ARTICLE IN PRESS

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Computer Networks xxx (2004) xxx–xxx

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

An overlay framework for provisioning differentiated services in source specific multicast ☆

Ning Wang *, George Pavlou

Centre for Communication Systems Research, University of Surrey, Guildford, Surrey GU2 7XH, UK

Abstract

Scalability in QoS-aware multicast deployment has become an important research dimension in recent years. In this paper we propose a new scheme named Differentiated QoS Multicast (DQM) based on the Source Specific Multicast (SSM) model in order to provision limited qualitative QoS channels for supporting heterogeneous end users. In a similar fashion to the Differentiated Services paradigm, in DQM the network is configured to provide unified QoS classes to both content provider and receivers. Based on Service Level Agreements, both sources and group members should select a specific QoS channel available from the network for group data transmission, and hence arbitrarily quantitative QoS states are eliminated. Moreover, we use the group address G contained in the (S, G) tuple in the SSM service model to encode QoS channels, and data packets belonging to the same QoS channel identified by a common class D address can be treated aggregately within core networks. Hence the proposed DQM scheme can be regarded as an overlay solution to the DiffServ paradigm, specifically for single source multicast applications.

Keywords: Differentiated services; Multicast routing; Source specific multicast

1. Introduction

In contrast to the current state of things, multiparty applications based on group communication are expected to become widespread in the Internet in the future. Among these, applications with Quality-of-Service (QoS) requirements will play an important role. Given this expected evolution, the situation in which the Internet is uniquely dominated by point-to-point communications based on the Best Effort (BE) service model should change soon.

Multicasting is an efficient paradigm for group communications thanks to its capability for bandwidth conservation. The recently proposed Source Specific Multicast (SSM [2]) model has been considered to be a promising solution for the development of one-to-many applications on a large scale. 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. Direct join requests

☆ A preliminary version of this paper was published in the Proc. International Workshop on QoS in Multiservice IP Networks (QoSIP 2003) [14].

* Corresponding author.

E-mail addresses: n.wang@eim.surrey.ac.uk (N. Wang), g.pavlou@eim.surrey.ac.uk (G. Pavlou).

1389-1286/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/j.comnet.2003.12.005

from individual subscribers create a unique multicast tree rooted at the well-known information source and SSM defines (S, G) channels on a per-source basis. In this model, the scalability problems of IP multicast such as address allocation and inter-domain source discovery are not deployment obstacles any more. Due to its simplicity and scalability, SSM is expected to see significant deployment on the Internet in the future, for applications such as Internet TV/radio, content distribution, etc. In this context, both the Internet group management protocol (IGMPv3 [6]) and multicast routing protocol (PIM-SM [8]) have been adapted to support the SSM model.

On the other hand, the provisioning of QoS capabilities in a scalable manner is another major research direction towards the next generation of the Internet. The Differentiated Services (DiffServ [3,11]) architecture is seen as a promising scheme for service differentiation on 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 (i.e., the bandwidth broker). Admission control and traffic conditioning are performed at edge routers, while core routers simply treat traffic aggregates on a Per Hop Behaviour (PHB) basis according to the Differentiated Services Code Point (DSCP) in each packet header.

Recent research efforts have extensively addressed the issue of multicast applications with heterogeneous QoS requirements given the potentially different capacity of individual receivers. Yang et al. proposed Multicast with QoS (MQ) [17] as an integrated framework with the consideration of QoS routing, resource reservation and user heterogeneity. This genuine receiver-initiated approach inherits some basic characteristics of RSVP [5], such as quantitative QoS guarantees and resource reservation merging from heterogeneous end users. It should be noted that MQ also requires that on-tree routers maintain state on a per-flow basis for end-to-end QoS guarantees, and this aspect still has problems of scalability. Other schemes propose to use dedicated multicast groups to carry multimedia information (e.g., video) with different QoS levels to heterogeneous receivers [7,10]. These approaches avoid maintaining quantitative QoS states in the network, but it is difficult to perform traffic aggregation, since QoS classes are exclusively defined and configured by individual external sources. More recently, schemes for developing multicast services in the DiffServ environment have been proposed [1,4,12,16], and multicast traffic belonging to the same QoS class can be treated in an aggregate fashion at core routers. However, these solutions need extension of the underlying multicast routing protocols (e.g., PIM-SM) as well as core router's forwarding infrastructure for the inclusion of QoS state, in order to route replicated traffic with different QoS treatment to heterogeneous receivers. The basic reason for this undesired situation is that the DiffServ framework caters mostly for sender-based unicast communication in which *Service Level Agreements* (*SLAs*) with a provider specify traffic entering the network at a particular ingress router. In the inherently receiver-initiated multicast paradigm, it is individual group members that demand various classes of service. These additional extensions not only raise new scalability and backwards compatibility problems in incremental deployment, but also violate the per-flow QoS stateless requirement at core routers in the DiffServ model.

In this paper, we propose a new framework called Differentiated QoS Multicast (DQM) for supporting qualitative service levels (e.g., Olympic Services) based on the Source Specific Multicast model. The basic characteristics of DQM are as follows: First, qualitative QoS states are directly encoded in the class D address and are centrally managed by the ISP, so that core routers inside the network remain stateless regarding QoS service classes. Second, differentiated levels of QoS demands for the specific information source are merged in one unique distribution tree, and data packets from different sources belonging to the same QoS service level, identified by a common multicast group address, can be treated aggregately. Moreover, a pre-defined number of classes of service by the ISP make it easier to provision network resources for each QoS aggregate, and this is in the same fashion to the classical Differentiated Services. From this point of view, the proposed DQM model can be regarded as an overlay solution of DiffServ, specifically for source specific multicast applications. Finally we apply per class routing to each QoS channel such that bandwidth requirements can be guaranteed for heterogeneous end users.

The rest of the paper is organized as follows. In Section 2 we survey some existing solutions for providing QoS heterogeneity in multicast applications. Section 3 presents a brief introduction to the proposed DQM framework. Section 4 is dedicated to the detailed description of DQM tree maintenance, including routing, data forwarding and group dynamics. We present an extended performance analysis of our scheme through simulation in Section 5, and finally we present a summary in Section 6.

2. Supporting multicast applications with QoS heterogeneity

2.1. The multicast with QoS (MQ) approach

Being an integrated solution, MQ sets up a multicast distribution tree with quantitative QoS requirements, and makes explicit bandwidth reservation for each group member during the phase of tree construction. When there exist heterogeneous receivers, resources are reserved up to the point where the paths to different receivers diverge. When a join request propagates upstream towards the source, it terminates at the point where there is already an existing QoS reservation that is equal to or greater than the one being requested. Fig. 1(a) illustrates how different resource reservations are merged along the multicast join procedure. Suppose the requests from receivers A, B and C demand 10 Mbps, 512 and 56 kbps bandwidth respectively, their reservations are merged to the highest request at each hop as shown in the figure. MQ can also adapt to resource consumption with dynamic group membership. For example, if an on-tree router detects that the departing receiver originally requested the highest QoS, it will automatically shrink its reservation or even reshape the distribution tree to exactly satisfy the remaining participants. In Fig. 1(b), we can see that when receiver A with the bandwidth requirement of 10 Mbps wants to leave the multicast session, the remaining receiver B with 512 kbps requirement will switch from the original "shared" path ($S \rightarrow R1 \rightarrow R2 \rightarrow R4$) with the capacity of 10 Mbps to a shorter one ($S \rightarrow R3 \rightarrow R4$) which still satisfies its QoS demand for bandwidth optimisation purposes.

On the other hand, the mechanism for network resource allocation works in an accumulative fashion, i.e., bandwidth is reserved in sequence for various incoming QoS requests until the link becomes saturated. This approach is straightforward and simple, but might not be efficient in bandwidth allocation, especially in case of highly dynamic group membership. From the deployment point of view, each on-tree router needs to maintain not only group states but also the quantitative QoS demands for its downstream receivers, and this imposes heavy overhead, in a similar fashion to RSVP.

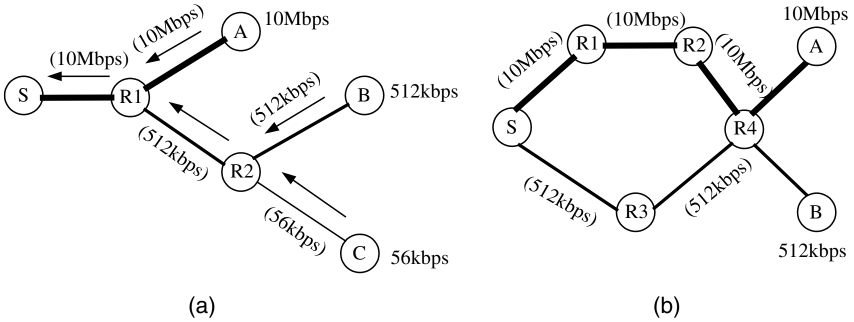


Fig. 1. MQ group join and tree reshaping.

2.2. Layered/replicated transmission

The layered transmission approach is particularly useful for Internet TV applications since it relies on the ability of many video compression technologies to divide their output stream into layers: a *base layer* as well as one or more *enhancement layers*. The base layer is independently decoded and it provides a basic level of quality. The enhancement layers can only be decoded together with the base layer to improve the video quality. The source can send individual layers to different multicast groups and a receiver can join the group associated with the base layer and as many layers for enhancement as its capability allows. Receiver-Driven Layered Multicast (RLM) [10] is a typical example for layered video transmission. Fig. 2 briefly illustrates the basic working scenario of RLM, and Fig. 2(b) describes how receiver *R2* "probes" to subscribe to additional enhanced layers for higher video quality.

It should be noted that not all types of multimedia streams can be encoded into layers as described above, and hence RLM has its own limitations in handling all types of applications. An alternative approach is replicated transmission that is applicable to generalized type of multimedia applications. In this approach, the information source keeps a finite number of streams carrying the same content, but each targeted at receivers with different capabilities. In a similar fashion to layered transmission, the data source assigns independent multicast groups to each of the maintained streams, and receivers may move among them by subscribing to the corresponding group address. A typical example of replicated transmission is Destination Set Grouping (DSG) [7]. Fig. 3 illustrates how DSG works in a heter-

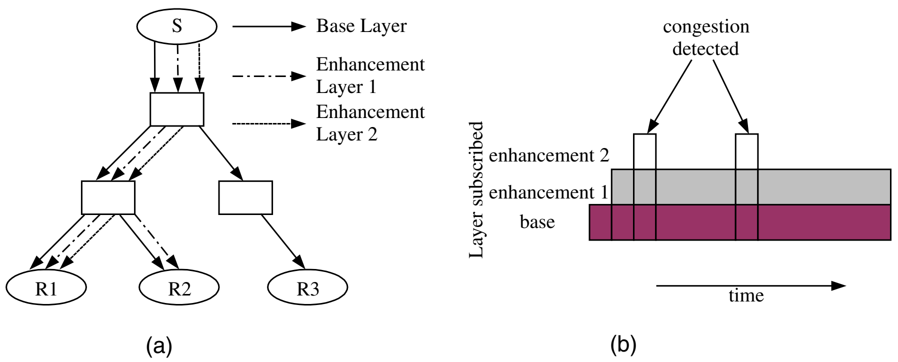


Fig. 2. Layered transmission example: RLM. (a) Layered multicast tree; (b) layer probing for R2.

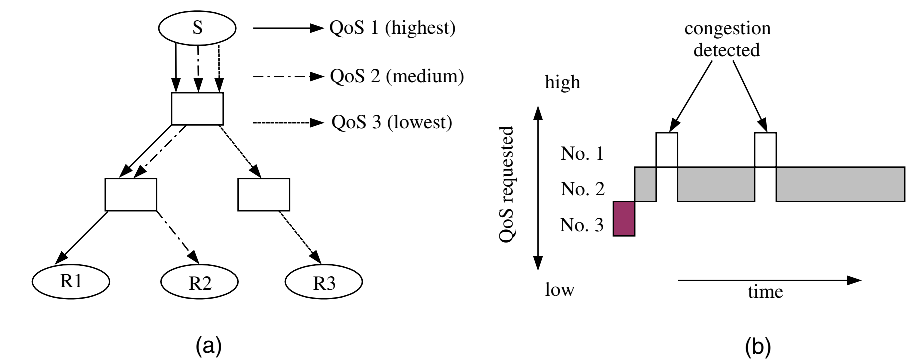


Fig. 3. Replicated transmission example: DSG. (a) Replicated multicast tree; (b) QoS probing for R2.

ogeneous environment. It should be noted that DSG receivers always subscribe to only one unique group for receiving data at any time, which is a key difference with RLM.

2.3. Diffserv-based multicast

Recently, research efforts have targeted the provisioning of multicast services in DiffServ networks [1,4,12,16], exploring a new research direction towards scalable deployment of QoS-aware multicast. In these approaches, one single delivery tree is constructed that encapsulates multiple classes of service, with individual tree branches reflecting heterogeneous QoS requirements from downstream users. The key idea in this type of hybrid QoS tree is that branches with lower classes can be directly grafted from those with higher classes for the same group session. We will take the *QUASIMODO* approach [1] as an example and illustrate how the multicast tree can support receiver heterogeneity. From Fig. 4(a) we can observe some similarities between MQ and *QUASIMODO*: upstream tree links always reflect the highest QoS requirements from individual downstream links for a particular multicast group. However, since *QUASIMODO* is a DiffServ-oriented scheme, only a set of Per Hop Behaviours (PHBs) and not arbitrary QoS requirements should be exhibited in the tree. In *QUASIMODO*, the Designated Router (DR) close to a receiver first embeds its QoS requirements in the PIM-SM join request, which should be extended for the inclusion of the associated DSCP value. When this join request is delivered, each of the core routers it has passed through should record the DSCP information and associate it with the outgoing interface (*oif*) from which the join request has been received. This is necessary because otherwise the DSCP contained in the group data cannot be modified when the packet reaches the branching point where heterogeneous QoS classes meet each other. By recording DSCP values at core routers, when group data comes from the incoming interface (*iif*), the on-tree router knows exactly through which QoS class it should forward the packets to its different outgoing interfaces. This can be done by checking the DSCP value on individual *oif*s.

We next calculate the memory overhead of core router extensions assuming Source Specific Multicast. Let us define first the following notations:

- $|S|$ Length of source address, which is 32 bits in IPv4.
- $|G|$ Length of SSM group address, which can be distinguished by 24 bits in IPv4 (232.*.*.*).
- X Total number of interfaces per router.
- Y Total number of QoS classes an ISP provides.

According to [9], the length of a typical forwarding entry for each (S, G) group session in a

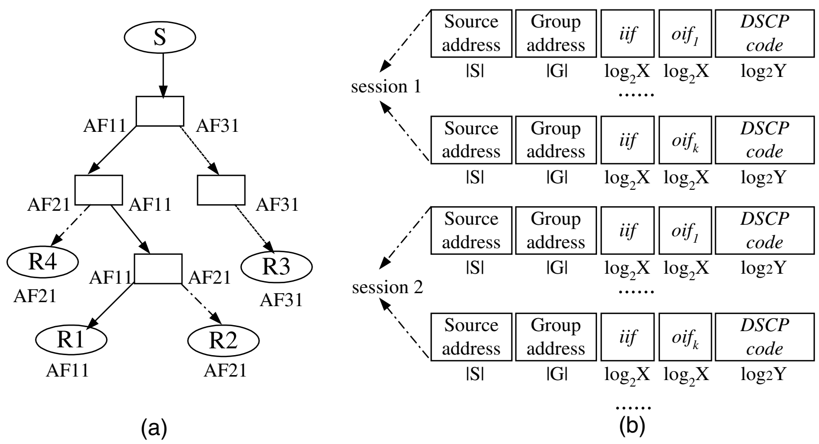


Fig. 4. *QUASIMODO* Multicast. (a) Tree structure; (b) forwarding table structure.

conventional SSM-aware router can be expressed as

$$E_{\mathrm{SSM}} = |S| + |G| + \log_2 X + X.$$

Assuming that the maximum number of interfaces per router is 16, then the SSM forwarding entry is 76 bits.

In *QUASIMODO* and other approaches [4,16], since a single tree is used to handle all QoS classes for a particular group, each outgoing interface (*oif*) needs to be associated with a unique PHB accordingly, and thus the most straightforward solution is to append an encoded DSCP value to each forwarding entry (shown in Fig. 4(b)). The length of such type of entry is

$$E_{QUASIMODO} = |S| + |G| + 2 \times \log_2 X + \log_2 Y.$$

Given a maximum of 64 classes of service, the length of a SSM forwarding entry is 70 bits. Note that this value is smaller than that of plain SSM (76 bits), but it should be noted that the *QUASIMODO* forwarding entries are outgoing interface instead of group session specific, i.e., a core router has to maintain k forwarding entries for a given session group where k is the total number of *oif*s associated with it.

3. Basic DQM framework

From the previous section, we can see that MQ provides a type of arbitrary bandwidth guaranteed services, while DSG and RLM offer qualitative services to heterogeneous end users. From a scalability viewpoint, the latter two approaches incur lighter state overhead at on-tree routers. On the other hand, neither RLM nor DSG provides the mechanism of multicast traffic aggregation within the network given that the QoS definition and configuration of each group session is done externally. *QUASIMODO* and other DiffServ-based approaches improve the situation by aggregating multicast traffic within each QoS class. However, additional extensions for QoS information are required for routing protocols as well as for the underlying forwarding table structure, which raises new problems in scalability and backwards compatibility.

In this paper we propose a new multicast transmission scheme, called Differentiated QoS Multicast (DQM), which can be regarded as the integration of Source Specific Multicast (SSM) and the Olympic Service model in DiffServ. From an ISP's viewpoint, it provides external customers (including both sources and receivers) with a finite set of classes of service (e.g., gold service, silver service and bronze service, etc.), each of which is uniquely encoded into an SSM-based class D address. 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 stands for the QoS service level (we name it QoS channel) that is available from S. Once receivers have decided the source address S and the desired QoS class, they will directly send *conventional* SSM (S,G) join requests towards the source, where the group address G identifies the QoS class being requested. In this scenario, when core routers receive group join requests, they do not need to maintain additional QoS states on per group session basis, as is the case in *QUASIMODO*. On the other hand, since the unified service levels are centrally managed by the ISP instead of individual sources, the proposed DQM framework (Fig. 5) still makes it possible for traffic aggregation on a QoS channel basis, without introducing any scalability problem when the number of external sources increases (i.e., total number of QoS channels is independent of the number of external sources).

It should be noted that the DQM tree is in effect an evolved version from source specific trees in the SSM model with additional heterogeneous QoS capability. Fig. 6 presents the basic structure of a DQM tree with three classes of service. In this tree, upstream links reflect the highest QoS class requirements, in a similar fashion to *QUASIMODO*. However, since QoS information has been embedded into the multicast class D address, the maintenance of a DQM tree is achieved exclusively by using group states, which conforms to the conventional SSM model. Using Fig. 6 for an example, we can see that individual QoS classes are encoded with SSM group addresses respectively, e.g., $G3$ identifies Gold service, $G2$ for Silver service, etc. Tree branches with $(S,G1)$ state can be grafted from those with either $(S,G2)$ or $(S,G3)$ states, which implies that Bronze tree branches are

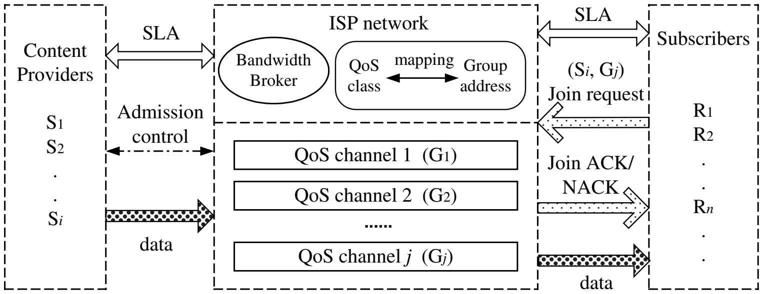


Fig. 5. The DQM framework.

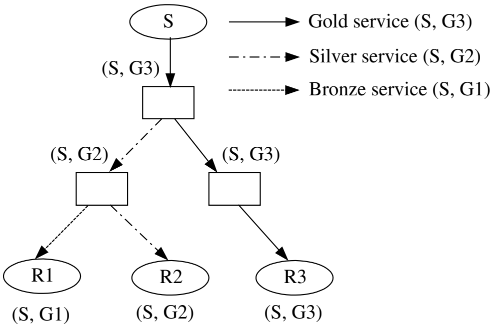


Fig. 6. DQM group state maintenance.

allowed to be extracted from Gold and Silver ones. This is in contrast to the scenario in Fig. 4(a), where tree maintenance still needs the aid of extra QoS states (i.e., DSCP values) kept at core routers.

The advantages of the proposed DQM scheme are as follows. First, it solves the fundamental conflict between the stateless DiffServ service model and the state-based IP multicast mechanism. In the DiffServ model, core routers do not maintain QoS state for individual applications/flows, and data treatment is according to the DSCP value in each packet header. On the other hand, the basic mechanism of IP multicast is to keep group states within the network in order to route data to active group members. In DQM, QoS state is directly encoded into the multicast group address and maintained within the network, as in IP multicast; hence, no additional QoS information needs to be kept at core routers. This indicates that the current conventional router structure is still applicable to the proposed DQM approach. Furthermore, the maximum number of QoS classes provided by an ISP needs not be bounded by the length of the DSCP field, which allows up to 64 levels (see Section 4.1). Second, service differentiation is centrally defined and managed by the ISP instead by individual sources, as it is done in DSG and RLM, so that traffic from different sources with identical QoS classes can be treated in an aggregate fashion within the network. Finally, in contrast to the "come and use" strategy of bandwidth allocation in MQ, DQM allows an ISP to allocate network resources specifically to individual QoS channels according to the forecasted traffic demands, so that the traffic distribution can be improved by the a flexible bandwidth configuration scheme. However, there is also a restriction with 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 conventional SSM model, an information source can be simultaneously involved in multiple group sessions because $(S, G1)$ and $(S, G2)$ are independent of each other. A short-term solution to this restriction is to use multiple unicast IP source addresses, with each dedicated to a particular group session.

4. DQM QoS channel maintenance

4.1. Data forwarding mechanism

Current router implementation for service differentiation adopts priority or weighted queuing technologies such as Class Based Queuing (CBQ) and Weighted Fair Queuing (WFQ). For example, in DiffServ networks data packets marked with different DSCP values are treated in queues with different priority for scheduling. Similarly, in

DQM the core network bandwidth is allocated to each QoS channel and data packets from different channels (distinguished by class D addresses instead of DSCP values) are scheduled in the corresponding priority queues. In this section we describe the working mechanism of routers that supports QoS channel differentiation.

Once a router receives (S, G) join requests with different values of G that are associated with various QoS channels from subscribers, it will merge all of them and only send a single (S, G_m) join request towards S, where G_m is the class D address associated with the highest QoS channel being requested. Using this approach, a single tree is constructed for all QoS channels of a group session. Detailed description of DQM group join and leave procedures is presented next in Section 4.2. In accordance with the conventional SSM terminology, we still define the interface from which a join request is received as the *outgoing interface* (*oif*) and the one used to deliver unicast data to the source as the *incoming interface* (*iif*). When the router receives group data from its *iif*, it will take the following steps to forward the packets (shown in Fig. 7(a), assuming $\mathrm{QoS}(G) > \mathrm{QoS}(G')$:

1. Check the group state(s) associated with the source S on each outgoing interface and replicate the packet where necessary.
2. Copy the value of G contained in the (S, G) state of each outgoing interface to the IP destination field in the replicated packet (if the two are not consistent).
3. Assign the data packet to the priority queue associated with the relevant QoS channel at the outgoing interface based on the (S, G) channel state.

Step 2 is necessary because the value of G contained in the packet indicates how this packet will be treated in the next on-tree router. Remember that the group states are created by (S, G) join requests for different QoS channels, and the way data packets are treated at each router is uniquely identified by the value of G contained in the (S, G) state; in this way, data can be forwarded according to the QoS requirements of individual users. On the other hand, data packets from different sources but with the same class D address in their (S, G) address tuples are treated aggregately in the corresponding queues. To achieve this, core routers should be configured so that each priority queue is associated with a group address at outgoing interfaces (see Fig. 7(b)). This figure also illustrates how data from different sources but with a common group address is treated aggregately in a specific queue of a core router.

4.2. Routing with dynamic group membership

In this section we discuss how a source specific DQM tree is constructed through dynamic group

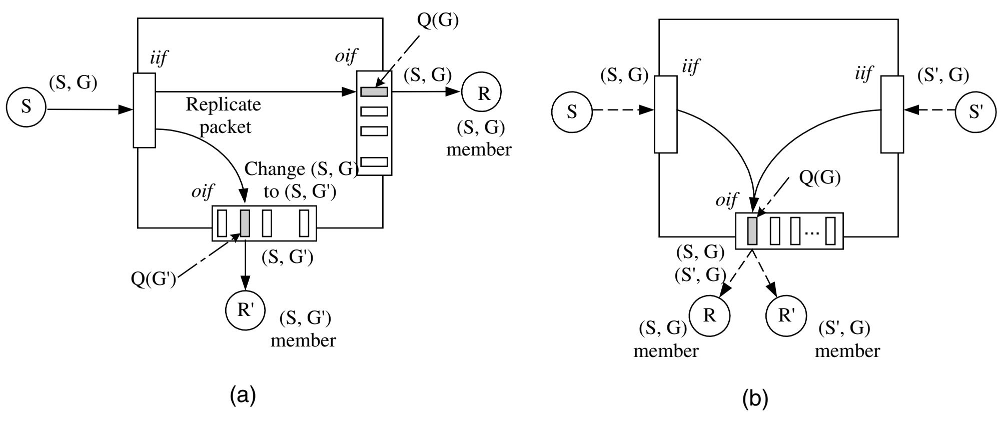


Fig. 7. DQM forwarding behaviour at core routers. (a) Data replication, (b) data aggregation.

membership. Individual priority queues should be configured with proper bandwidth allocation for each QoS channel, thus the joining path of the same source–destination pair might not be exactly the same for all (S, G) channels. This is because path computation also considers the specific bandwidth availability of the subscribed QoS channel; we name this Per Channel QoS routing.

4.3. QoS channel subscription

Once an end user R decides to join the DQM tree rooted at source S with a desired QoS class, it first negotiates with the Bandwidth Broker (BB) on the bandwidth availability for that QoS channel. If successful, the user will send an IGMPv3 [6] (S, G) group membership request to its Designated Router (DR) at the edge of the DiffServ domain, where G is the associated DQM group address mapped to the negotiated QoS channel. If the BB finds that there is not sufficient bandwidth for admitting the traffic invoked by the join request, the user may adaptively choose to select a lower (S, G) QoS channel and renegotiate. On receiving the membership report, the DR will submit to the source a plain SSM-based PIM-SM (S, G_i) join request that does not contain any extra QoS class information from the new group member R. This join request will follow a feasible path with sufficient available bandwidth for supporting channel G_i towards the source S. When the (S, G_i) join request reaches a router that has already received traffic from the source S with the same or higher QoS channel, i.e., with group state (S, G_j) where $G_i \leqslant G_j$, [1] then the join procedure terminates and this interface is added to the *oif* list of group (S, G_i). Thereafter, data packets from S are replicated and forwarded to this interface with the class D address of the new packets modified from G_j to G_i. This way, a new tree branch is grafted from the current QoS channel that has equal or higher service level.

If the (S, G_i) join request reaches a router with the highest available QoS channel (S, G_j) where $G_i \geqslant G_j$ (i.e., a router with lower QoS channel for S), the join will continue to explore a new path that satisfies the new requirement of the (S, G_i) channel subscription. Once a path with desired QoS channel has been set up and this particular router has received traffic from the (S, G_i) channel, it will tear down the (S, G_j) channel on the original path with lower QoS level. It should also be noted that the procedure of tearing down the (S, G_j) channel might invoke another internal join request from an on-tree router, where (S, G_j) is the highest local channel it maintains and there exist other channels with lower QoS channel (see the example below). The flowchart for group join is presented in Appendix A, and it is worth mentioning that this flowchart also includes the steps for handling internal group joins invoked by QoS channel unsubscriptions, which is specified later.

In Fig. 8, we assume that initially there already exists a single QoS channel constructed by $(S, G2)$ subscriptions from both receivers $R1$ and $R2$ (Fig. 8a). After some time router D receives a $(S, G1)$ subscription from $R3$ where $G1 < G2$, i.e., a subscription with a lower QoS channel. In this case D will send a join request towards S and this request will terminate at router B that has already received group data from S for a higher QoS channel (shown in Fig. 8b). In Fig. 8c, we assume that router E receives a $(S, G3)$ join request from $R4$ where $G3 > G2$. In this case a new path with a higher QoS channel is constructed, shown with the solid line in the figure. When router C receives data traffic from S in $(S, G3)$ channel, it will tear down the original $(S, G2)$ channel back to S. When router B has detected the pruning, it finds that it has also maintained a lower QoS channel for $R3$, namely $(S, G1)$. Therefore, it will first send a $(S, G1)$ join request back to S. When detecting that group data from S comes in the new channel $(S, G1)$, router B will tear down the original $(S, G2)$ channel on link AB as shown in Fig. 8d.

4.4. QoS channel unsubscription

Suppose that a particular router is currently receiving traffic from source S with QoS channel (S, G_i). When it detects no (S, G_i) subscribers attached and wants to leave the channel, it will stop

[1] We assume that higher class D address is associated with higher QoS channel, i.e., $G_i > G_j \leftrightarrow \mathrm{QoS}(G_i) > \mathrm{QoS}(G_j)$

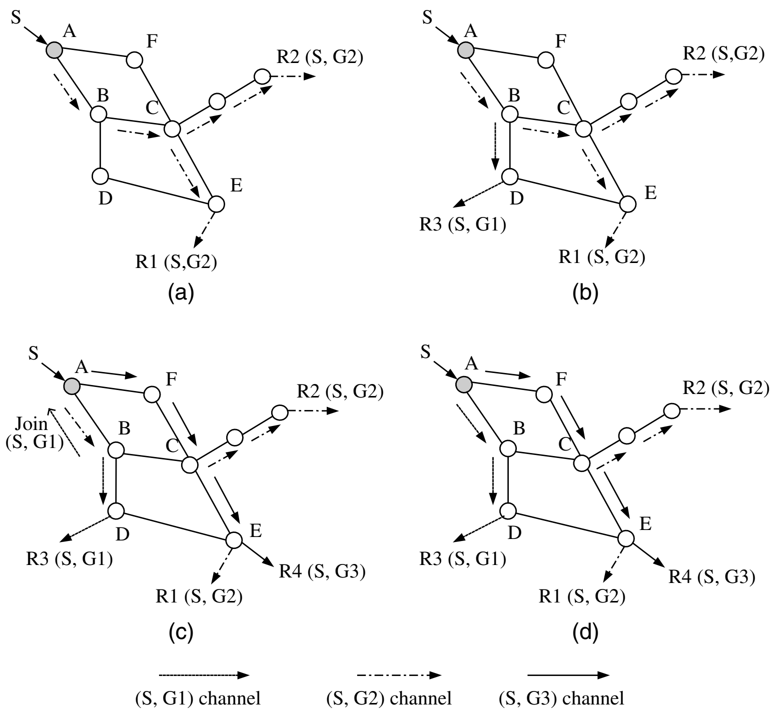


Fig. 8. Dynamic QoS channel subscription.

sending (S, G_i) join requests towards the source S. When the (S, G_i) state times out at the *oif*, the upstream router will check all its *oif*s with QoS channels associated with S. There exist three possible cases as follows (illustrated in Fig. 9):

a. There exists at least one (S, G_j) state where $G_j \geqslant G_i$, *or* there are other *oif*s for (S, G_i); then the router will simply stop forwarding traffic on the (S, G_i) channel at this timed out *oif*, and it will not need to take any further actions;
b. There does not exist any (S, G_j) state where $G_j \geqslant G_i$, *and* this interface is the only *oif* for (S, G_i); then the router will check the status of all the remaining QoS channels associated with S, it will select the class D address G_m that is associated to the highest QoS channel currently requested and it will send an internal (S, G_m) join request towards the source S. Once this router has received data traffic from the (S, G_m) channel, it will stop sending (S, G_i) join requests on its incoming interface. Special considerations are required for internal join requests invoked by channel unsubscriptions, and we will discuss this issue in detail using an example;
c. If this is the last subscriber for S, the router will simply stop sending any (S, G) join requests towards the source and hence it will break from the tree.

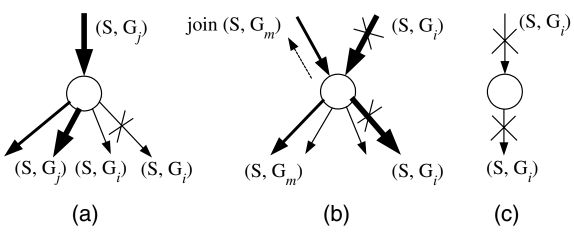


Fig. 9. Dynamic QoS channel unsubscription.

The flowchart for QoS channel unsubscription is provided in Appendix B.

We still follow the example in Fig. 8 to illustrate the QoS channel unsubscription procedure. Starting from Fig. 8d, we assume that receiver $R4$ unsubscribes from the $(S, G3)$ channel, and we will show how DQM efficiently adapts the tree for the remaining group members. When router E notices this unsubscription, it finds out that the highest remaining active channel is $(S, G2)$ for $R1$, and hence it first sends a $(S, G2)$ join request towards S. When the upstream router C detects that currently there exists a higher QoS channel $(S, G3)$ on the interface from which this lower $(S, G2)$ join request is received, it assumes that this downstream router E is downgrading its QoS requirement due to a high QoS channel unsubscription it has noticed. [2] Meanwhile router C finds out that, after E has downgraded its requirement to $(S, G2)$, the remaining highest channel becomes exactly $(S, G2)$, and hence it will send an internal $(S, G2)$ request towards S (This procedure is also described in Appendix A for internal group joins invoked by the relevant QoS channel unsubscription, identified by the * branch). Let us assume that router B is the next hop, and that its highest QoS channel is $(S, G1)$, which is lower that what has been requested from router C. In this case, router B first forwards the $(S, G2)$ request to the upstream router A, and once $(S, G2)$ traffic comes from A, it will stop sending $(S, G1)$ join requests on the same path, so that the $(S, G1)$ channel will be deleted on link AB after the channel state times out. Once the $(S, G2)$ traffic reaches router C from B, router C will stop sending $(S, G3)$ join request to router F, so that in a similar fashion is pruned from the $(S, G3)$ channel. Similarly, router F will prune itself from the tree since it is not receiving any join request for the source S. As a result, the adapted DQM tree is restored to that of Fig. 8b after receiver $R4$ unsubscribes from the $(S, G3)$ QoS channel.

It should be noted that the basic mechanism in this routing with loop-freedom guarantees applies also to *QUASIMODO* and other DiffServ-based schemes that follow an approach of building hybrid QoS trees. However, none of those schemes have investigated a detailed per-class QoS routing scenario according to group dynamics as we consider here. Moreover, QoS routing in those schemes needs not only group states but also extra QoS information (i.e., DSCP). This requirement introduces additional overhead to DiffServ core routers, as we have also previously indicated in Section 2. Finally, it is worth mentioning that in DQM, a boundary router issues join/leave requests only when the first receiver for a new (S, G) session joins or the last member leaves the group. This strategy of pushing group management to the edge of the network reduces significantly the frequency of reshaping delivery trees within the domain.

[2] This assumption is only valid for point-to-point router connections, and further research is required for operations on multiaccess networks.

5. Simulation results

5.1. Simulation model

In this section, we evaluate the proposed scheme through simulation. We adopt the GT-ITM topology generator for constructing our network models. This approach distributes the nodes randomly on the rectangular grid and nodes are connected with the probability function

$$P(u, v) = \lambda \exp\left(\frac{-d(u, v)}{\rho L}\right),$$

where $d(u, v)$ is the distance between node u and v and L is the maximum possible distance between any pair of nodes in the network. The parameters λ and ρ ranging $(0, 1]$ can be modified to create the desired network model. A larger value of λ gives node with a high average degree, and a small value of ρ increases the density of shorter links in comparison to longer ones. In our simulation we set the values of λ and ρ to be 0.3 and 0.2 respectively, and generate a random network with 100 nodes with the source node being randomly selected.

In order to generate group dynamics, a sequence of events for QoS subscription/unsubscription are also generated. A probability model is used to determine whether a request is for QoS subscription or unsubscription. The function

$$P_c = \frac{\alpha(N - m)}{\alpha(N - m) + (1 - \alpha)m}$$

is defined for this purpose [15]. The function P_c is the probability that a QoS subscription is issued. In the function, m indicates the current number of subscribers while N identifies the network size. α ranging (0, 1) is the parameter that controls the density of the group (i.e., the average number of subscribers). When a QoS subscription is issued, a node that is not in the multicast group is randomly selected for joining the session. Similarly, a node is randomly removed from the current multicast group when a QoS unsubscription request is triggered.

In our simulation, the average number of subscribers varies from 10 to 40 in steps of 5 (by setting the value of α). For simplicity we assume that there is at most one subscriber attached at each edge router. In addition, we also assume that the ISP provides three qualitative QoS channels, namely Gold Channel (GC), Silver Channel (SC) and Bronze Channel (BC), and that the subscription bandwidths for these three channels are 8, 4 and 2 Mbps per receiver respectively. Within the network, the bandwidth capacity of each link varies from 10 to 45 Mbps in an even distribution. The bandwidth capacity of each link is partitioned in the following proportion: 50% for GC, 30% for SC and 20% for BC respectively. Among all the receivers, we assume that 20% of them subscribe to GC, 30% to SC and 50% to BC. In our simulation we adopt the *QOSPF* algorithm as the receiver-initiated DQM routing paradigm on per QoS channel basis, i.e., the DR of individual group members is responsible for computing a feasible join path with bandwidth requirement for the subscribed channel. According to [13], *QOSPF*-based multicast routing does not support user heterogeneity within a particular group, but in DQM such type of QoS heterogeneity is reflected by different (S, G) group identification. In this sense, *QOSPF* can still apply to per QoS channel routing in DQM, and different tree branches can be merged if the same source address S is discovered.

5.2. Performance analysis

First of all, we investigate bandwidth conservation performance, and comparisons are made between DQM and that of building independent trees for each service level with disjoined QoS channels (e.g., DSG [7] in which the source maintains independent data streams for heterogeneous users simultaneously). We did not include the performance of RLM because it is not a solution for the general QoS requirement, but only for some specific multimedia applications (e.g., layered video). Following that we focus on the capability of traffic engineering in terms of load balancing between the DQM and MQ approaches. We also compare network utilization between DQM and MQ, but overhead for group state maintenance is incomparable since the latter involves quantitative states for user heterogeneity. Finally we study the performance of scalability in terms of memory overhead for group state maintenance.

In order to evaluate the network utilization, we define the bandwidth conservation overhead for a particular channel C (C could be gold, silver or bronze service) as follows:

$$O_C = 1 - \frac{U^C_{\mathrm{DQM}}}{U^C_{\mathrm{DSG}}},$$

where U^C_{DQM} is the bandwidth utilization of channel C by DQM, and U^C_{DSG} is that by using the schemes with independent QoS tree maintenance such as DSG. Similarly, we define the overhead for all channels as

$$O_T = 1 - \frac{U_{\mathrm{DQM}}}{U_{\mathrm{DSG}}},$$

where U_{DQM} is the overall link utilization by DQM and U_{DSG} is that by DSG.

Fig. 10 illustrates the overhead performance for both individual QoS channels and overall bandwidth conservation. We observe that in DQM bandwidth for non-gold channels can always be saved and the corresponding overhead varies from 0.33 to 0.46. Obviously, bandwidth for gold channels is not conserved at any time since it cannot be merged into any other QoS channel. Regarding the overall bandwidth conservation, we observe that the aggregated overhead varies from 0.19 to 0.23, i.e., by using QoS channel merging in DQM, the average bandwidth consumption is 81.3–84% that of non-QoS merging approaches.

Another interesting empirical study is the traffic engineering capability of DQM and MQ. In DQM, network bandwidth is pre-allocated to specific traffic aggregates of individual QoS channels, and this is very similar to the general strategy of DiffServ. In contrast, MQ and RSVP allow the

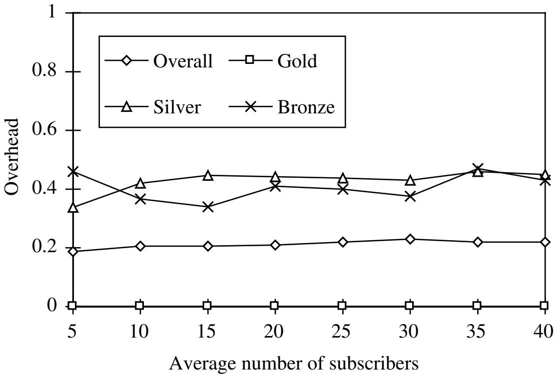


Fig. 10. Overhead of bandwidth conservation.

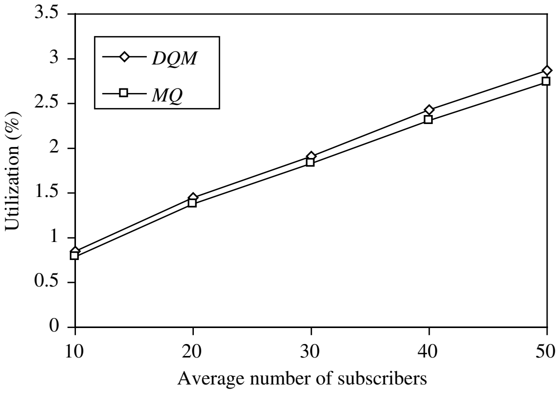


Fig. 11. Overall link utilization comparison with MQ.

overall bandwidth to be accumulatively reserved by QoS demands until the link has become saturated. In the following simulation, we examine the performance of load balancing in DQM and MQ/RSVP. According to bandwidth utilisation, we classify network links into the following three categories: (1) High load link with overall utilization above 50%; (2) Medium load link with overall utilization between 20% and 50%; and (3) Low load link with overall utilization below 20%. Table 1 presents the proportion of these three types of links in the network with the average number of subscribers varying from 10 to 50. From the table we can see that DQM performs better in terms of load balancing since traffic is more evenly distributed. For example, when the average number of subscribers is below 30, none of the network links become highly loaded in DQM. In contrast, MQ always results in hotspots with utilization above 50% even when the average number of subscribers is 10. From the table we can also see that the proportion of low load link in DQM is consistently higher than that in MQ.

We also investigate the overall link utilization of DQM and MQ, and the simulation results are presented in Fig. 11. From the figure we can see that the average link utilization of DQM is consistently higher than that of MQ by a small margin, e.g., when the average number of subscribers is fixed at 50, the link utilization of DQM is 4.7% higher than that of MQ. From the empirical results in Table 1 and Fig. 11, we can infer that the better performance of DQM's load balancing is in effect at the expense of higher bandwidth consumption, but the relevant cost is very small (i.e., up to maximum 5% higher than MQ).

In addition to the previous evaluation based on traffic characteristics, we also investigate the scalability aspect in terms of memory consumption for group state maintenance. We scope the comparison between approaches that use group states to identify differentiated QoS classes, i.e., DQM and DSG. The reason we exclude RLM and *QUASIMODO* is as follows: in RLM, receivers need multiple group subscriptions to obtain high QoS classes for a single session, which obviously

Table 1
Traffic distribution comparison with MQ

		10	20	30	40	50
DQM	High load link (%)	0.00	0.00	0.00	0.07	0.12
	Medium load link (%)	1.2	2.6	4.1	4.9	5.4
	Low load link (%)	98.8	97.4	95.9	95.0	94.5
MQ	High load link (%)	0.23	0.41	0.86	1.33	1.58
	Medium load link (%)	1.7	3.1	4.1	4.7	6.7
	Low load link (%)	98.1	96.5	95.0	94.0	91.7

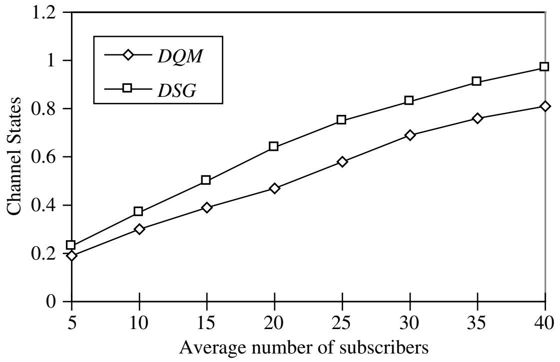


Fig. 12. Overall channel states maintenance.

results in significantly heavier memory overhead than both DQM and DSG. Moreover, RLM is specific to layered video applications, which means that it is not a general solution for multicast QoS heterogeneity. On the other hand, group states in *QUASIMODO* do not contain QoS information, and routing functionality also needs the aid of DSCP values kept at core routers. In this sense, comparison only in terms of group state overhead does not reflect the scalability of the approach.

Fig. 12 shows the average number of channel states that are maintained at each router, i.e., the total number of states in the network divided by the number of routers. The simulation configuration in this experiment is the same as that of Fig. 10. From the figure we can see that the number of channel states needed per router increases as the group size grows. On the other hand, by using QoS channel merging in DQM, the burden of maintaining group states can be alleviated significantly, e.g., when the number of subscribers is fixed at 40, the router memory overhead of using DQM channel merging is 83.5% that of DSG, which needs dedicated trees for each QoS class. In the extreme case, in DQM the ingress router for the source S only maintains one (S, G) state, where G corresponds to the highest QoS channel requested from all the downstream receivers. In contrast, DSG requires that the first hop router of the source maintain as many group states as the total number of QoS classes being subscribed. Fig. 13 depicts the memory overhead for maintaining individual QoS channel states in both schemes. We can see in this figure that the total number of states for the Gold service is exactly the same in DSG and DQM. This is because the branches for the top QoS channel cannot be grafted from any other tree, and given the same group subscription scenario, DSG and RLM always form an identical tree shape for the Gold channel. On the other hand, by comparing (a) and (b) in Fig. 13, we also notice that DQM is able to conserve group states for non-top class channels, namely the Silver and Bronze classes in our simulation. For example, when the average number of subscribers reaches 40, the number of group states for the Silver channel in DQM is 80% that in DSG, and for the Bronze channel the corresponding value is 72.1%.

6. Summary

In this paper we proposed a novel overlay scheme called DQM that provides differentiated QoS channels based on the Source Specific Multicast (SSM) service model. This approach efficiently supports heterogeneous QoS requirements applications on a qualitative basis, without extensions to the multicast routing protocol and

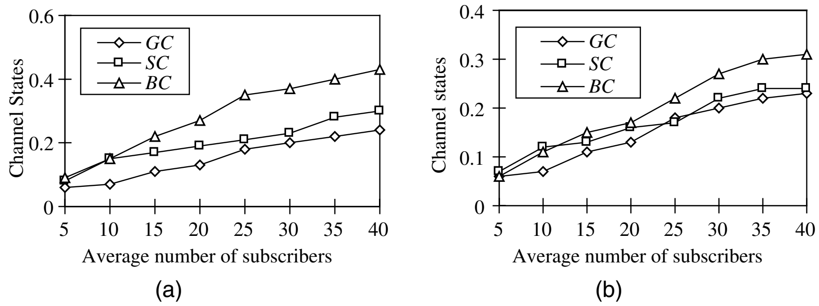
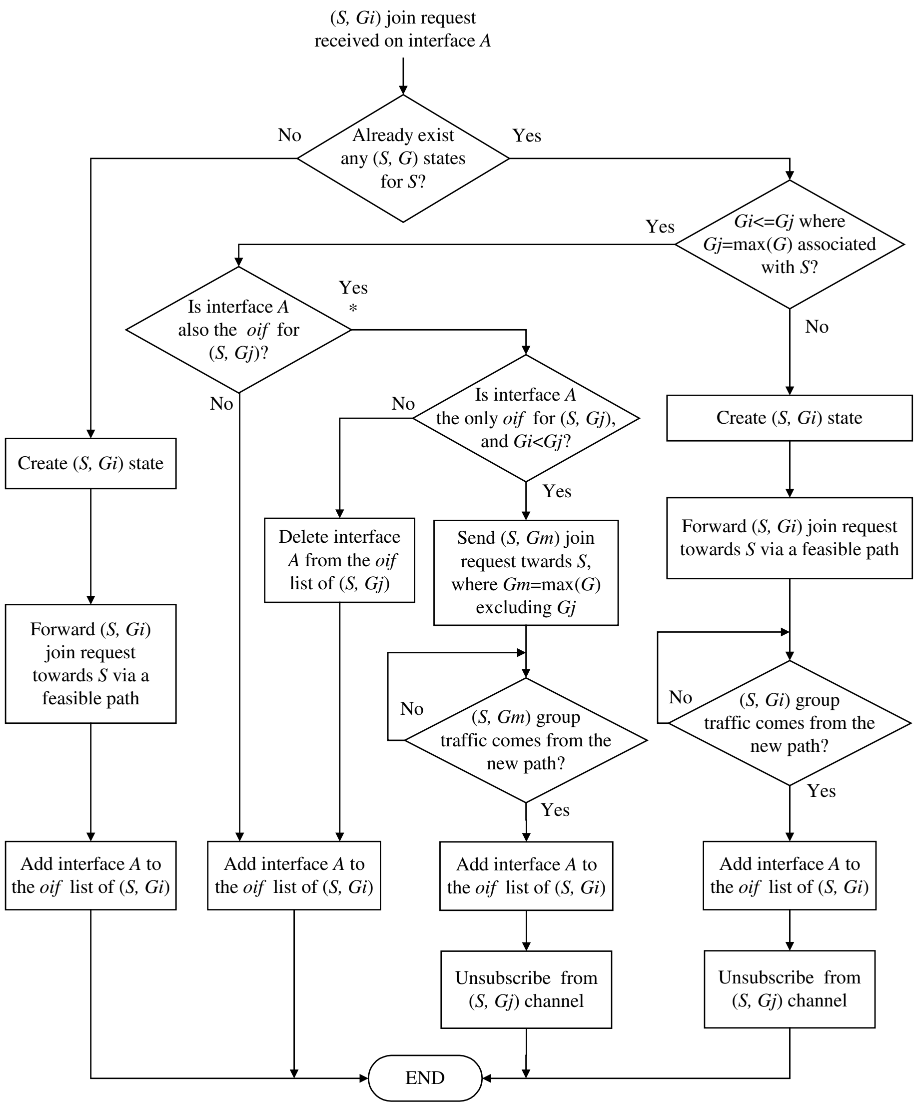


Fig. 13. Individual channel states maintenance (a) DSG, (b) DQM.

router forwarding infrastructure. By means of per channel QoS routing and merging mechanism, not only router overhead for maintaining group states is alleviated, but also network bandwidth consumption is reduced compared with traditional solutions such as multicast layered and replicated transmission. Moreover, per QoS channel bandwidth management contributes to improvements in terms of traffic load distribution in comparison to the MQ/RSVP approaches.

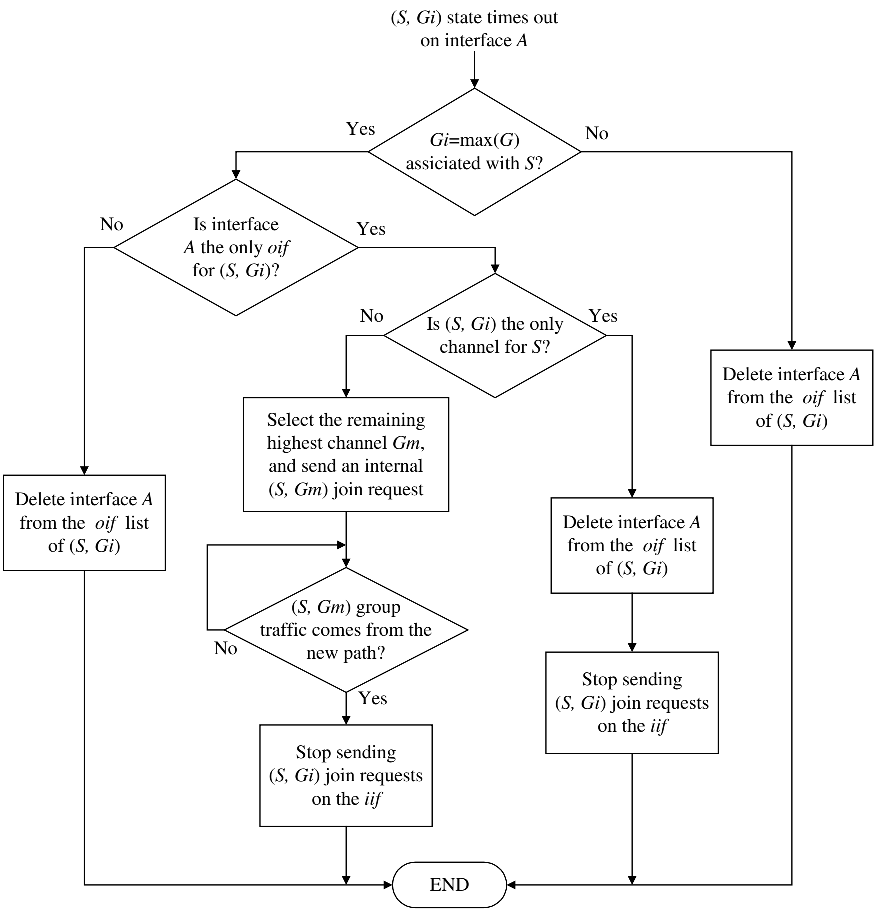
Our future work will address dynamic configuration and management of network resources (e.g., bandwidth pre-emption among QoS channels) based on forecasted traffic condition of QoS aggregates, and also algorithms for DiffServ-aware multicast traffic engineering.

Appendix A. Flowchart for handling group join



* This branch is only for handling internal group joins invoked by downstream *QoS* channel unsubscriptions

Appendix B. Flowchart for handling group leaving



References

[1] G. Bianchi et al., QUASIMODO: Quality of service-aware multicasting over diffserv and overlay networks, IEEE Network, Special Issue on Multicasting 17 (1) (2003) 38–45.

[2] S. Bhattacharyya, An Overview of Source Specific Multicast (SSM), RFC 3569, July 2003.

[3] S. Blake et al., An Architecture for Differentiated Services, RFC 2475, December 1998.

[4] R. Bless, K. Wehrle, Group communication in differentiated services networks, in: Proc. IQ2001, pp. 618–625.

[5] R. Braden et al., Resource ReSerVation Protocol (RSVP), Version 1 Functional Specification, RFC 2205, September 1997.

[6] B. Cain et al., Internet Group Management Protocol, Version 3, RFC 3376, October 2002.

[7] S. Cheung et al., On the use of destination set grouping to improve fairness in multicast video distribution, in: Proc. IEEE INFOCOM 1996, pp. 553–560.

[8] B. Fenner, Protocol Independent Multicast—Sparse Mode (PIM-SM): Protocol Specification (Revised), draft-ietf-pim-sm-v2-new-08.txt, October 2003, work in progress.

[9] H.W. Holbrook, D.R. Cheriton, IP multicast channels: EXPRESS support for large-scale single-source applications, in: Proc. ACM SIGCOMM 1999.

[10] S. McCanne et al., Receiver-driven layered multicast, in: Proc. ACM SIGCOMM 1996.

[11] K. Nichols et al., Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, RFC 2474, December 1998.

[12] A. Striegel, G. Manimaran, A scalable approach for DiffServ multicasting, in: Proc. IEEE ICC 2001.
[13] B. Wang, J. Hou, QoS-based multicast routing for distributing layered video to heterogeneous receivers in rate-based networks, in: Proc. IEEE INFOCOM 2001.
[14] N. Wang, G. Pavlou, DQM: an overlay scheme for quality of service differentiation in source specific multicast, in: Proc. International Workshop on QoS in Multiservice IP Networks, Lecture Notes in Computer Science, Vol. 2601, Springer, Berlin, 2003, pp. 306–319.
[15] B.M. Waxman, Routing of multipoint connections, IEEE J. Select. Areas Commun. 6 (9) (1988) 1617–1622.
[16] B. Yang, P. Mohapatra, Multicasting in differentiated service domains, in: Proc. IEEE GLOBECOM 2002.
[17] D. Yang et al., MQ: an integrated mechanism for multimedia multicasting, IEEE Trans. Multimedia 3 (1) (2001) 82–97.

Ning Wang received his B.Eng degree from Changchun University of Science and Technology, China in 1996 and his M.Eng degree from Nanyang Technological University, Singapore in 2001 respectively. He is currently a PhD student in the Centre for Communication Systems Research (CCSR), Department of Electronic Engineering, University of Surrey, UK. His research interests include multicast communication, QoS provisioning, and Internet traffic engineering.

George Pavlou is a Professor of Communication and Information Systems at the Centre for Communication Systems Research, Department of Electronic Engineering, University of Surrey, UK, where he leads the activities of the Networks Research Group. He holds a BEng in Electrical and Mechanical Engineering from the National Technical University of Athens, Greece, and MSc and PhD degrees in Computer Science from University College London, UK. He is responsible for a number of European and UK research projects and industrial collaborations. His research interests include network planning and dimensioning, traffic engineering, multicast, mobile ad hoc networks, multimedia service control, programmable networks and communications middleware. He has published about 100 papers in refereed international conferences and journals and has contributed to standardization activities in ISO, ITU-T, TMF, OMG and IETF.