

Providing Multicast Communication in a Differentiated Services Network Using Limited Branching Techniques*

Minaxi Gupta, Mostafa Ammar
Networking and Telecommunication Group
College of Computing
Georgia Institute of Technology, USA
minaxi, ammar@cc.gatech.edu

Abstract

The differentiated services (DS) paradigm has emerged as a scalable approach to provide QoS for unicast communication in the Internet. Utilizing the DS to provide QoS for multicast poses several challenges because of the multi-point aspect, dynamic group membership, and heterogeneous receiver resource requirements of multicast. Existing proposals to accomplish this goal either require extra per packet overhead or changing multicast routing tables. This paper proposes an alternate architecture called M-DS (multicast-DS), utilizing two inter-operable limited branching techniques, *edge-router branching* and *limited-core branching*, as appropriate within each DS domain. Our focus is on defining the signaling protocols for each technique, one for resource allocation on membership discovery and another for resource deallocation on membership termination in subnetworks. Signaling sets up state in each DS domain so that the data can flow through them as it would in the case of the unicast DS. M-DS preserves the DS scalability, does not incur any per packet overhead due to extra headers in data packets, and routes packets using the IP multicast routing tables that are already set up in individual domains, albeit by enhancing the router functionality. We evaluate the performance (signaling overhead and extra bandwidth required) of our architecture and show that it is practical to include it in the current DS framework.

*This work is supported by the AFOSR MURI grant F49620-00-1-0327, NSF grant ANI-9973115, and by a research grant from Bellsouth.

1 Introduction

Multicast communication accomplishes one-to-many and many-to-many delivery of data in an Internet environment. Some examples [1] of one-to-many applications are: scheduled multimedia distribution, file distribution, announcements, and stock price dissemination. Examples of many-to-many applications include multimedia conferencing, collaborative documents, chat groups, distance learning, and multi-player games. Multicast is scalable and efficient because it outperforms unicast even for a small number of receivers. A recent study [2] shows that even when there are 20-40 receivers, multicast can be 60-70% more efficient than unicast in the Internet.

The differentiated Services (DS) architecture [3] is a scalable method for implementing service differentiation for unicast communication in the Internet. It allocates network resources to traffic streams by service provisioning policies. Traffic classification state is conveyed by means of IP-layer packet marking. The architecture is scalable because it aggregates flows with similar characteristics. Further, the core of the network is kept simple and traffic classification, marking, policing, and shaping operations are implemented only at network boundaries or hosts.

Dynamic join/leave of multiple potentially heterogeneous receivers poses unique challenges in providing support for multi-point multicast communication in a DS network. This is because when new receivers join the multicast group, branches may get added to the existing multicast tree with-

out prior resource allocation and this can adversely affect the unicast and multicast traffic for which resources have previously been reserved. Existing proposals for providing support for multicast in the DS framework either incur extra per packet overhead by introducing extra headers in data packets or require changing multicast routing tables.

This paper proposes an alternate architecture called M-DS, a scalable architecture to provide QoS for multicast communication utilizing the DS framework. The architecture uses one of two limited branching techniques, namely, *edge-router branching* and *limited-core branching*. For each of the techniques, we focus on defining two signaling protocols; one for resource allocation on membership discovery and another for resource deallocation on membership termination on subnetworks. The signaling for resource allocation configures state in appropriate routers to be used during multicast data transmission. After this phase, data can begin to flow for multicast with QoS, as it would be in the case of DS for unicast. The signaling for resource deallocation resets the configuration changes made by the signaling for resource allocation. All the four signaling protocols have low message overhead. *Edge-router branching* achieves scalability by moving all the complexity to the edges of the domain. It disallows multicast branching points in the core of any domain. Keeping the core simple slightly increases the packet hops for the overall multicast tree. *Limited-core branching* allows a limited number of branching points to exist within each domain to reduce the bandwidth overhead in terms of packet hops introduced by moving the branching points to domain edges, while still maintaining scalability. For both the techniques, flows are aggregated for scalability, just as in the DS framework for unicast. Also, both techniques use the multicast state already set up in individual domains for routing packets and interoperate with each other. Not changing multicast tables also facilitates the co-existence of IP multicast without the QoS requirements. These techniques have no per packet overhead during data transmission in terms of extra headers in individual data packets. We elaborate later on the changes required in the routers to accommodate these techniques in a DS framework.

The remainder of this paper is organized as follows. The next section provides a discussion of related work. Section 3 discusses the challenges involved in providing support for multicast communication in a DS network. Section 4 explains the components of the architecture and outlines the assumptions. Sections 5 and 6 describe the signaling protocols for both the techniques that comprise the M-DS architecture. Routing details for M-DS are described in section 7. Performance results of our simulations are presented in section 8. Finally, section 9 presents the conclusions.

2 Background And Related Work

A DS domain is comprised of *boundary nodes* and *core nodes*. Boundary nodes interconnect the DS domain to other DS or non-DS capable domains while core nodes only connect to other core or boundary nodes within the same DS domain. Traffic enters a DS domain at an ingress node and leaves at an egress node. Figure 1 shows a DS domain $DS1$, boundary nodes $B1$ and $B2$, and core nodes $C1$, $C2$, and $C3$. $B2$ connects $DS1$ to a second domain $DS2$.

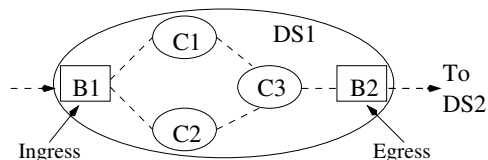


Figure 1: Various Entities in a DS Domain

The DS framework uses a six bit *DS field* from the IP header to define DS codepoints. All packets with the same codepoint that cross a link in a particular direction form a behavior aggregate. The DS boundary nodes at the customer egress set the appropriate codepoint in each packet in accordance with the customers' service level agreement (SLA) and the packet joins the correct behavior aggregate. From this point on, subsequent boundary or core nodes in various DS domains have no information about a particular customer's flow, they only deal with behavior aggregates. This contributes significantly to the scalability of the architecture. The DS architecture specifies various components

needed to treat packets belonging to various behavior aggregates at the boundary nodes of the DS domains. Traffic classifiers separate submitted traffic into different classes. The packets from different classes matching some specified rule are subjected to metering, shaping, policing and/or remarking to ensure that traffic entering a DS domain conforms to the SLA. Traffic forwarding in the core DS nodes is very simple; the core nodes apply the appropriate per hop behavior to each behavior aggregate.

The DS framework is specified with unicast in mind. Recently, there has been some work in the direction of utilizing the DS framework for providing service differentiation for multicast. Bless and Wehrle [4] have pointed out some challenges in using the existing DS architecture for multicast communication. We describe these and other challenges in detail in section 3. They propose to extend multicast routing tables to include codepoints to provide QoS for multicast in the DS framework. This involves changing IP multicast protocols. Striegel and Manimaran [5] have proposed an encapsulation-based approach called *DSMCast* for providing multicast support in a DS domain. Their approach consists of adding a *DSMCast* header to each packet at the edge of the DS domain by the ingress router. Upon receiving such a packet, a core router will inspect the packet to determine which interfaces the packet should be replicated on based on the information contained in the *DSMCast* header. This solution keeps the core routers simple but incurs bandwidth overhead for every data packet. This approach is scalable in terms of number of multicast groups, but not in terms of bandwidth overhead because the *DSMCast* header size is dependent on the number of receivers in each DS domain. Our approach, M-DS, is scalable both in terms of number of multicast groups, as well in terms of number of receivers. There are two kinds of overheads in both of our techniques. First, the signaling overhead to set up state in appropriate routers. This is a small *one-time* overhead incurred for each receiver join and is independent of the duration of data flow. Second, the bandwidth overhead in terms of extra packet hops incurred because of possibly moving the branching point. This overhead is topology dependent and is

controllable in the case of *limited-core branching*. There is no bandwidth overhead in terms of extra headers in individual data packets.

3 Challenges

Providing quality of service (QoS) for multicast communication using the DS framework poses unique challenges. Multicast sources generally do not know the identity of receivers. The source sends out one copy of the data and the IP layer multicast routers make duplicate copies where needed in the network to reach all receivers of the group. Also, group membership in multicast is dynamic and the receivers are heterogeneous. Out of these issues, heterogeneity is not a stumbling block because of the manner in which multicast can be accomplished at the application layer. At the application layer, receivers with different resource requirements join different multicast groups [6], so within a particular multicast group, the resource requirements are homogeneous. However, dynamic group membership makes it challenging to provide QoS for multicast communication.

The DS architecture for unicast can not be used as is to accomplish QoS for multicast without affecting other traffic adversely. The main reason for this is because scalability in the DS architecture for unicast is achieved by distinguishing between the functionality of core and boundary routers in each domain and by traffic aggregation. Except for the routers near the source, all routers along the way deal only with behavior aggregates in the DS architecture. In multicast, however, new members may join a multicast group dynamically and as a result, several routers in the core of the network may duplicate packets to reach the new receivers. Keeping the core routers simple to preserve the DS scalability would imply core routers assign the same codepoint to the duplicated packets as to the original packets. As a result, new branches can get added to the existing multicast tree without prior resource reservation. This problem is termed as *non-reservation subtree (NRS) problem* in [4]. NRS can potentially lead to violation of SLAs between the DS peers and hence compromises QoS for one or more classes of traffic. Hence, dynamic resource allocation is required to add the new re-

ceivers.

Dynamic group membership poses another challenge for adapting DS for multicast communication because multicast does not lend itself to defining static SLAs like unicast does. This can be explained by describing a strawman proposal to define static multicast SLAs for providing QoS for multicast communication. For every domain, one can define a superset multicast SLA for each ingress node in the domain. This SLA would cover all possible egress nodes to account for dynamic receiver join downstream of each egress node. The problem with this approach is that depending on how many receivers join the multicast group, there may be unused bandwidth that can just be used by the best effort traffic, but it can not be reserved for other unicast flows.

4 Architecture Components And Assumptions

An example of various entities of the M-DS architecture is shown in figure 2. It shows two DS domains, with the multicast source attached to domain $DS1$ and receivers $R1$ and $R2$ on the same subnetwork attached to domain $DS2$ through the designated router (DR). Each domain has boundary and core routers. The figure also shows the bandwidth brokers $BB1$, and $BB2$, for domains $DS1$ and $DS2$ respectively. The DRs initiate the signaling with the BB of their domain upon each multicast group membership discovery and termination on their respective subnetworks. The bandwidth brokers (BB) handle resource allocation and deallocation requests in their domain [7] by contacting their peers.

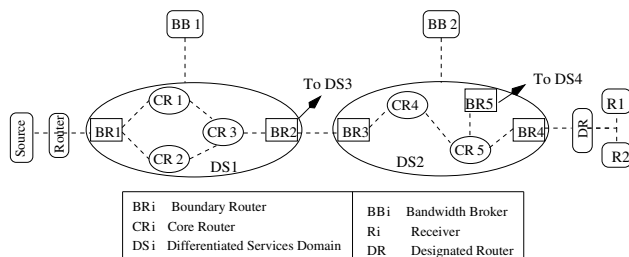


Figure 2: Various Components of the M-DS Architecture

The architecture makes the following assumptions about the infrastructure. First, each DS domain has static unicast SLAs with its peer domains for each of its ingress and egress router pairs. These SLAs are for aggregates of flows. Admission requests for both unicast and multicast are handled dynamically by the BB of each domain by contacting the BB in the peer domain. These assumptions are consistent with those of the *QBone* BB work group [7]. Second, each BB has TCP connections for communication with all its peer BBs and the DRs of its domains for communication during signaling. It also has access to unicast multicast routing information. BGP-4 [8] routers have TCP connections with each other and they exchange routing information, hence the BBs can be configured as BGP routers to satisfy this requirement. Third, each BB has a data repository containing router configurations and policy information. It needs this to be able to make decisions to allocate and deallocate resources. Fourth, before sources start sending data, they register with the BB of their domain, which can then propagate this information to peer BBs. In this manner, the BBs know about the active multicast sources.¹ Fifth, we assume that all receivers know what QoS to ask for. The resource request can be carried in a manner similar to the one prescribed by the RSVP specification [11].

5 Edge-Router Branching Technique

We now describe the details of the edge-router branching technique. It uses existing unicast SLAs and exploits multicast scalability at domain granularity because it allows branching to occur only at the ingress and egress routers. If there are any branching points in the core of the domains, they are moved to the ingress of the domain. This keeps all the complexity confined to the edge of the network and hence the core nodes are kept scalable as in the case of DS for unicast.

¹This is essentially similar to how MSDP [9] requires the RPs [10] in each domain to advertise active sources.

5.1 Signaling Protocol for Admitting Receivers

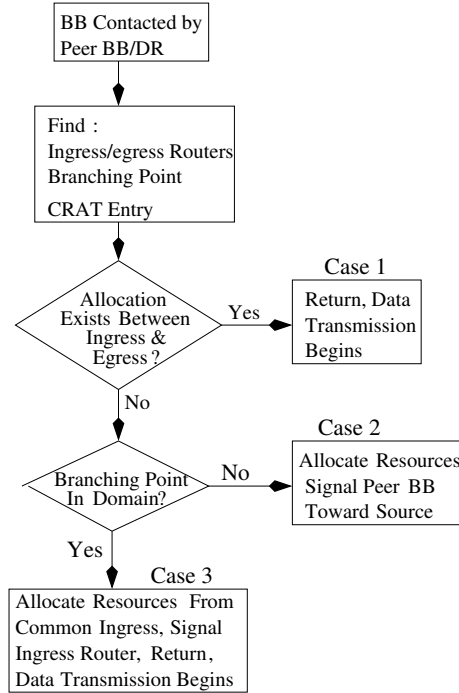
Signaling is initiated by the DRs of the domains upon a multicast group discovery on their respective subnetworks. To begin with, the DR sends a message to the BB of its group, giving it its own IP address, the QoS requirements and the multicast group address G . Subsequently, this BB may contact its peer BB and so on. Notice that for each subsequent receiver joining the same subnetwork served by this DR, no action needs to be taken as long as the QoS requirements and G are the same.

The BB extracts the QoS and G and then finds out if the resources for this request have already been allocated in its domain. Because if they are, no new resources are to be allocated. The BB needs two pieces of information to make that decision.

1. Appropriate entries from the *current resource allocation table*. This table contains resource allocation information for each pair of ingress and egress routers in the domain along with the multicast groups they serve. It also contains the number of different subnetworks (identified by the IP addresses of the DRs) each ingress-egress router pair serves, directly or indirectly. Keeping the number of the subnetworks served corresponding to each ingress-egress router pairs helps the BB know when to deallocate the resources in its domain. Entries in this table are filled after making a decision that resources can be granted. To find out the appropriate entry, the BB first finds out the pair of ingress and egress routers in its domain in the path from the multicast source for group G to the DR. It does that using the routing information that it has access to as a BGP-4 router.
2. If the *branching point* for G lies in its domain. While the signaling protocol is in progress, multicast state is being set up in the domain using the IP multicast protocol in use in that domain. The core routers in every domain that determine that they are going to be the branching point for group G communicate this information to their BB by sending the corresponding multicast forwarding table en-

try. The BB sets a timer to get this information from the branching point router(s) in its domain. If the timer expires without the BB getting a reply, it assumes that no branching point exists in its domain.

Based on the above information, three cases arise for the ingress-egress router pair needed to satisfy this request, as outlined in figure 3.



CRAT → Current Resource Allocation Table

Figure 3: Signaling Steps in Edge-Router Branching Technique upon Membership Discovery

5.1.1 Case 1: Allocation Exists

This case arises if the BB finds an entry in the current resource allocation table with requested QoS corresponding to the ingress and egress routers needed to serve the new request for group G . This implies that adequate resources have already been allocated. Nothing needs to be done other than updating the number of subnetworks served by this ingress-egress router pair in the current resource allocation table and the BB replies in the path to the DR affirmatively and signaling is terminated. The new receivers are grafted to the existing multicast tree and since multicast state is already set

up, they can start getting data. Figure 4 depicts the signaling for the case when $R1$ is present and $R2$ wants to join group G . The signaling for this case returns from $DS2$ and does not need to go all the way to domain $DS1$.

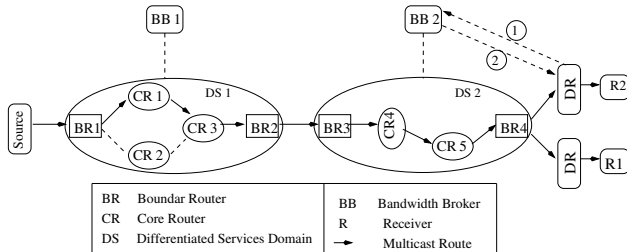


Figure 4: Signaling for the Case 1

5.1.2 Case 2: No Allocation, No Branching Point

The second case arises if no resources have been allocated for G in this domain with requested QoS. In this case, the BB does not find any entry with the requested QoS in the current resource allocation table corresponding to the ingress and egress routers needed to serve the new request. Also, no branching point router replies to the BB. The BB makes a decision to grant resources in the form of an SLA between the ingress-egress router pair based on policy information and SLA availability. Note that to be able to use the existing multicast routing state during actual data transmission, the SLAs have to be multicast routed unicast SLAs, i.e., unicast SLAs that follow the same path from ingress to egress routers as the multicast route would take. But in case such an SLA is not available, there are two options that BBs in individual domains can employ. The first involves communicating with all routers in the unicast and multicast routes between ingress and egress routers and *re-installing* the multicast routing state to reflect the new route according to the available unicast SLA. The second option is to use IP tunneling [12] to use normal unicast SLAs.

If resources are granted, the BB updates the current resource allocation table for the latest bandwidth allocation corresponding to the ingress-egress router pair involved and sets the subnetwork

count for the corresponding entry to one. The signaling does not terminate here for this case and the BB signals the upstream peer BB. Upon receipt of the message, the peer BB runs the same protocol as this BB. Figure 5 depicts the signaling for the case when $R1$ joins group G in a two domain network. No resource allocations exist for G in either domain $DS1$ or $DS2$, so new resources are allocated in both domains when $R1$ joins. The signaling goes all the way to the multicast source's domain before returning back to the DR in this case.

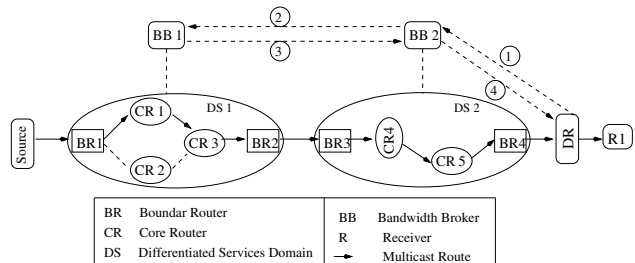


Figure 5: Signaling for the Case 2

5.1.3 Case 3: Branching Needed

The third case arises if one or more core routers send the corresponding multicast forwarding table entry to the BB of their domain, informing that they would be the branching point for group G . This implies that the multicast tree passes through this domain but branching is required to graft the new receivers to the existing multicast tree. If the allocated resources are adequate for the new QoS request, new receivers can be grafted to the existing tree. The edge-router branching technique does not allow any branching point in the middle of any domain, hence to be able to graft new receivers to group G , an additional SLA from ingress router to the new egress is used if one is available. Figure 6 shows the use of two separate SLAs in a domain when the branching point exists in a domain.

Since all SLAs are unicast SLAs, to be able to use the new SLA, the BB moves all branching points from its core to the ingress of the domain and enhances the role of the ingress router to act as a branching point in addition to being an ingress router. This technique makes the core router functionality very scalable, just as in the DS for unicast

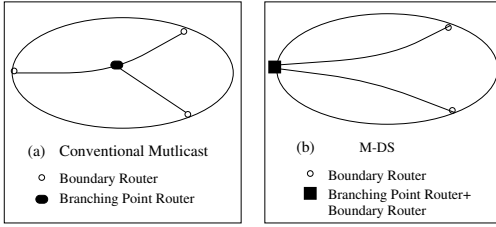


Figure 6: Two different SLAs when branching point exists in the domain

communication but doing so amounts to introducing some changes in routing. The BB extracts the new routing information using the multicast forwarding table entry sent to it by the core branching point router(s) and conveys the appropriate routing information to the ingress router. The details on how the ingress router performs routing during data transmission and how core branching point router(s) use their existing multicast routing state are explained in section 7. As the section explains, there is no per packet bandwidth overhead during data transmission and some changes are required to be made to router functionality.

After moving the branching point to the ingress, the BB creates a new entry for this ingress-egress router pair in the current resource allocation table and sets the subnetwork count for the corresponding entry to one. To complete the signaling, the BB signals affirmatively in the path toward the DR and data transmission begins for the new receivers since multicast state is already set up. Figure 7 depicts the signaling for the case when $R1$ and $R2$ are present and branching is needed in $DS2$ for data to reach $R3$. New resources are allocated in $DS2$ and signaling returns to the DR without going all the way to domain $DS1$.

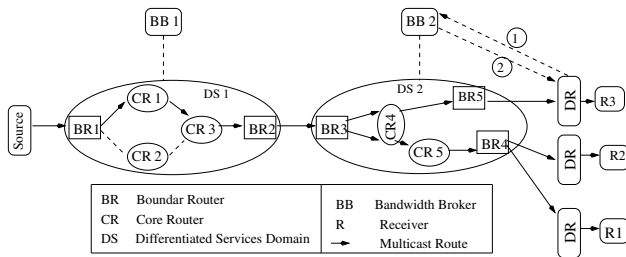


Figure 7: Signaling for the Case 3

At any point in the path to the source, if re-

sources are denied, signaling returns from that BB all the way back to the DR, freeing all the resources and canceling all configuration changes and table updates. Most IP multicast routing protocols are based on soft state protocols and if data transmission does not begin, routing state vanishes automatically due to lack of a refresh. In our case, since the source will not be sending data to the new receivers unless resources are allocated all the way, the IP multicast routing state would vanish automatically.

5.1.4 Additional Considerations

In addition to the signaling overhead described above that is incurred when DRs make resource allocation/deallocation requests, depending on the protocol in use in a domain, there may be some state maintenance overhead also for our protocols. Many multicast protocols like PIM-SM [10] are based on the notion of soft state. Hence, routers periodically send state join/prune messages to reflect the latest receiver population for various multicast groups. If the underlying routing changes are reflected in these periodic messages, the action of moving the branching points from the core of a domain to the ingress (as in the third) case described above will have to be carried out in that domain. In addition, depending on the amount of traffic from particular sources, PIM-SM switches from shared tree to source specific tree, which can lead to a change in the branching points in one or more domains. In this case also, the action of moving the branching point from the core to the ingress router will have to be carried out.

5.2 Signaling Protocol When Receivers Leave

The protocol for the case when resources need to be deallocated for multicast group G is very similar to that of the case when they need to be allocated; only allocation of resources become deallocations. As in the case of allocation of resources, the DRs first contact the BB. The BB uses the routing information it has to look up the ingress-egress router pair serving this receiver in its domain. There are two possible scenarios. First is that the sub-

network count corresponding to the ingress-egress router pair is one. This means this was the last receiver group served from this branch of multicast tree. Second is that the subnetwork count is greater than one, implying that there are more receivers on subnetworks in this domain or other domains that are being served by this ingress-egress router pair. In the first case, the resources can be deallocated and the corresponding entry from the current resource allocation table can be removed. Also, the changes made to the functionality of the ingress router of this domain are undone and peer BB in the path toward the source is signaled to carry out the leave protocol as well. In the second case, the subnetwork count is decremented by one because resources can not be deallocated yet and the adequate changes are made to the functionality of the ingress router. For the second case, the signaling can return from this point back to the DR confirming that the last receiver's leave from group G is complete.

5.3 Detailed Description

In this section, we describe the basic state transitions of all the entities of the M-DS architecture (with respect to a particular multicast group G and QoS requirements) that undergo changes with respect to their functionality in the case of the DS for unicast. Such entities in each domain are: the receivers, DRs, core routers, ingress routers, and the BB.

Receivers A receiver R is in one of the three states at any point of time: READY (ready to receive data), WAITING (has contacted the DR with group G and QoS information and is waiting to receive data), or RECEIVING (receiving data). If t denote the time and T the event of a time-out that prompts the receiver to contact its DR again and N the maximum number of time-outs, the transitions for R are as shown in figure 8. We use *done* to denote the transition onto the starting state for all the entities.

Designated Routers The state transitions for a DR look almost identical to those of a receiver. Upon multicast discovery (DISCOVERY), The DR transitions to state WAITING by providing its

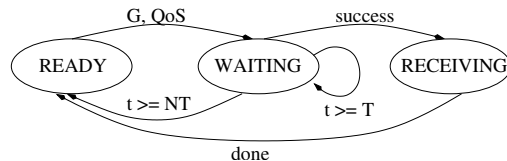


Figure 8: State Transition Diagram for a Multicast Receiver

IP address, multicast group G , and R 's QoS requirements to the BB of its domain. It transitions to state RECEIVING when it starts getting data for R . Assuming the time-out period is T , the transitions are as shown in figure 9.

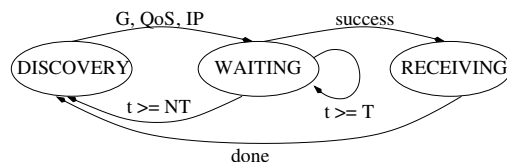


Figure 9: State Transition Diagram for DR

Core Routers While the IP multicast routing state is being set-up in a particular core router, it is in state SETUP. If we denote the event of core router discovering it is a branching point router by BP , depending on whether or not it is a branching point router, it transitions into state CONTACTING, whereby it contacts the BB of domain to give the BB its appropriate multicast forwarding table entry. The state when it is receiving data and processing it according to the M-DS routing rules described in section 7 is called RECEIVING. These states yield the state transition diagram shown in figure 10.

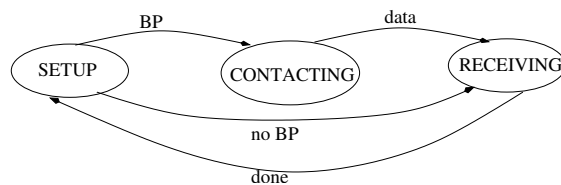


Figure 10: State Transition Diagram for Core Router

Ingress Routers An ingress router starts out in state CONTACTED, when it is contacted by the

BB for duplicating packets for multicast group G , instead of the original core branching point router. While in this state, the ingress router records information sent by the BB for use during data transmission for G . Upon receiving multicast data, the ingress router processes and forwards them according to the rules described in section 7 and transitions into state RECEIVING. The state transition diagram is as shown in figure 11.

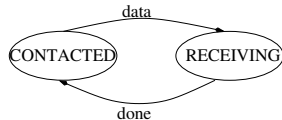


Figure 11: State Transition Diagram for Ingress Router

Bandwidth Broker We begin by describing the ten states of a BB that involve either contacting a peer BB or a DR of its domain, or the states BB is in after being contacted by a peer BB or the DR of its domain. We use *downstream* to indicate the direction of data flow from the source to the receiver and *upstream* for the direction from the receiver to the source. The BB is in state CONTACTED/RA when it is contacted by its downstream peer BB or DR for resource allocation. It is in state CONTACTED/RD when it is contacted for resource deallocation by the downstream peer BB or DR. The states where it is contacted for successful allocation/successful deallocation/failed signaling by upstream BB are CONTACTED/SA, CONTACTED/SD, and CONTACTED/FS respectively. The corresponding five states for the case when the BB has to contact its peer BBs or DRs are given by CONTACT/RA, CONTACT/RD, CONTACT/SA, CONTACT/SD, and CONTACT/FS, with the upstream becoming downstream and vice versa. The transitions between these states occur because of the following events:

- looking up the routing information (RI) to find out the ingress and egress routers in its domain that the SLA for group G requires, denoted by L -RI.
- looking up the current resource allocation table (CRAT) to find out if appropriate resource

allocations exist between ingress and egress routers for group G , denoted by L -CRAT.

- expiration of timer that was started to get a response from all the core routers that act as branching point routers for group G , denoted by T .
- getting a response from the core branching point router (CR) before the expiration of the timer, denoted by R -CR.
- availability of SLA between the ingress and egress routers for G , denoted by A -SLA.
- using the forwarding table entry(entries) sent by the core branching point router(s) to send M-DS routing information to the new branching point router (the ingress router), denoted by S -RI.
- updating the count of networks that a particular ingress/egress router pair serves (for deallocation purposes), denoted by C .
- BB discovering it does not have any peer BB on the upstream (implying that the source is in the BB's domain), denoted by S
- undoing the routing changes sent to the branching point router (ingress router) and deallocating resources, denoted by U .

Figure 12 shows the transitions between the ten BB states. The BB can start out in any of the five *CONTACTED* states and end up in any of the five *CONTACT* states. The figure shows dashed lines for the case when the BB is contacted by the *upstream* router and solid lines for when the BB is contacted by the *downstream* router.

6 Limited-Core Branching Technique

The edge-router branching technique defined in section 5 uses available unicast SLAs in each domain and concatenates them to make up end to end multicast SLAs. It exploits multicast scalability only at the domain level, and hence incurs extra

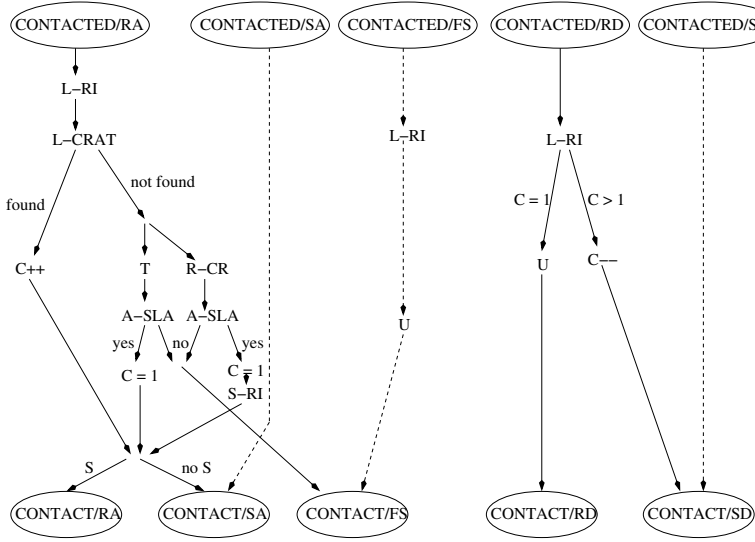


Figure 12: State Transition Diagram for BB

packet hops. This simplifies the core of the domain but introduces extra packet hops during data transmission. A good trade-off between making all core routers that are branching points complex versus incurring extra packet hops is to limit the number of core routers that would be allowed to be branching point routers (section 8 will quantify these trade-offs). This is the motivation behind *limited core branching*.

For the edge-router branching technique, we assumed that SLAs in a domain existed only between ingress and egress router pairs, for the purposes of the limited-core branching technique, we assume that additional SLAs have been defined from certain special *core routers* to some egress points in addition to the usual unicast SLAs. These special core routers are the only ones that can be used as branching points in the domain. We do not address the issue of optimal placement strategy of special core routers, optimal number of such routers, and whether special core routers can be dynamically changed when receiver population changes. That remains an issue of future investigation.

The rest is a simple extension of three cases we described for edge-router branching. The manner in which the first two cases, namely when no resource allocations exist and when exact resource allocations exist and no branching is needed, are dealt with as before. The only difference is in the

way the third case is handled. For every new membership discovery, if the BB finds out that one or more branching point(s) exist in its domain, before enhancing the functionality of the ingress router to duplicate packets to reach all downstream receivers, it first finds out if there is a special core router configured in its domain that is allowed to serve as a branching point. If there is, then for the path subsequent to that core router in that domain, that core's functionality is enhanced to be similar to that of the ingress router in edge-router branching. Only when the configured limit on the allowed branching point routers in the core of the domain is exhausted that the ingress router's functionality is enhanced to duplicate packets in addition to perform the DS specified functions.

Figure 13 shows an example of a domain that runs limited-core branching with one special core router. It shows the situation when receivers have joined a particular multicast group G that requires exiting at 5 egress routers. The role of branching point $B4$ has been upgraded to that of special core router SC , while the behavior of other branching points $B1$, $B2$, and $B3$ remains the same as in edge-router branching. As a result, the ingress router will be instructed to make 4 duplicate copies of packets, one to reach receivers $R1$ and $R2$, and one each to reach $R3$, $R4$, and $R5$. The special core router will be instructed to make two copies to reach $R1$ and $R2$. The special core router has become more complex than $B1$, $B2$ because it performs functionality similar to ingress router (but for fewer traffic flows), and $B3$ but the number of packets traversing the path between the ingress I and special core router has been reduced.

Limited-core branching reaps the benefits of minimized packet hops during data transmission while keeping the technique scalable by limiting the number of special core routers allowed at the cost of a slight increase in complexity, which is controllable by individual domains. Any domain can opt to using limited-core branching independently of other domains and can configure as many special core routers as it chooses. This is because both the edge-router branching and limited-core branching techniques inter-operate.

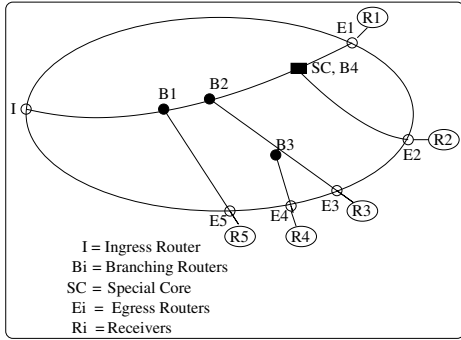


Figure 13: Different Roles of Routers in Limited-Core Branching

7 Routing Under M-DS Architecture

This section describes the routing details for the edge-router branching technique for the case when branching is required to graft the new receivers to the existing multicast tree. The details can easily be extended to limited-core branching technique because the only difference between edge-router branching and limited-core branching techniques is the presence of special core routers that act as branching point routers. For edge-router branching, only ingress routers act as branching point routers.

The basic idea behind the new routing is to keep the core simple. As a result, the branching point(s) is(are) moved from the core of the domain to the ingress. The ingress router would now need to duplicate packets (and hence act as the new branching point router) to reach all receivers instead of the original core branching point router(s). The complication arises when these identical duplicate packets reach the original branching point router(s) in the core of the domain. The technique described in this section helps the ingress router to put the information in the packets (without introducing any extra headers) that guides the original core branching point router(s) to use the multicast routing information they have and forward the data packets on its correct interfaces without duplicating the packets.

Routing Entry Sent by Core Branching Point Routers to BB

During the signaling for resource

allocation, when one or more core routers of a domain determine that they are going to be the branching point routers to graft the new receivers of group G to the existing multicast tree, they send their corresponding forwarding table route entry to the BB of its domain. A forwarding table route entry in an IP multicast router includes fields such as source address, multicast group address, incoming interface from which the packets are accepted, list of outgoing interfaces to which packets are sent, and the corresponding TTLs (Time To Live).

Routing Information Sent by BB to New Branching Point

The BB uses the multicast forwarding entry sent by the core branching point routers to deduce and communicate the routing information to the new branching point router(s) in its domain. In the case of edge-router branching, it would just be the ingress router of this domain, but in the case of limited-core branching, a few special core routers can be configured as new branching point routers as well. The routing information contains multicast address for which packets have to be duplicated, outgoing interface number for the new branching point router (same as the incoming interface of the original core branching point router), and the number of entries; one for each original core branching point router. Each entry contains the IP address of the core branching point router for identity purposes, the number of duplicate packets that the core branching point router would have made, TTL, and the individual bit vectors corresponding to each outgoing interface of the original core branching point router (explained in section 7; one each corresponding to the number of duplicate packets (see figure 14).

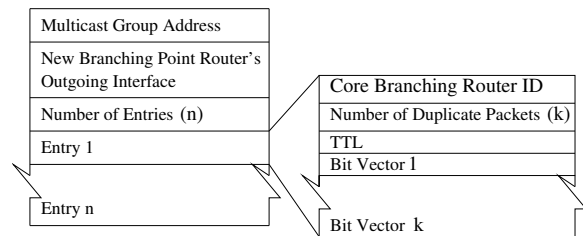


Figure 14: Routing Information for the New Branching Point Router

New Branching Point Router Functionality

The new branching point router extracts the entries for each core branching point router from the information sent by the BB. For each entry, it makes as many packets as indicated by the *number of duplicate packets* field. To help the original core branching point router process each of these duplicate packets, we propose a solution that utilizes the IP options [13] field as shown in figure 15. In each of the duplicate packets, the new branching point router copies the *core branching router ID*, and the *bit vector*.

11000101	Length	Core Branching Router ID	Bit Vector
----------	--------	--------------------------	------------

Figure 15: New IP option

The first bit in the first octet of figure 15 is to ensure copying of this new field in all fragments. The next two bits define a new option class for DS for multicast, and the last 5 bits define a new option type with *number* = 1. The second octet contains the length of this IP option. Following that are the core branching point router identity and the bit vector.

Core Branching Point Router Functionality

Upon seeing a multicast packet, the original core branching point router extracts the ID field from the IP option field along with the bit vector. If the ID field matches its own IP address, it processes the packets, otherwise, it just forwards it without doing anything. The processing of the packets is simple. The bit vector (assume of size one octet) can take 256 values, one corresponding to each outgoing interface of core branching point router. The core branching point router uses this bit vector and the relevant multicast forwarding table entry for group G , and maps the outgoing interface in the multicast entry to the bit vector. After the mapping, instead of making duplicate packets using the multicast entry, it forwards the packet on that interfaces.

8 Performance

Bless and Wehrle [4] have proposed extending the multicast routing tables, while the DSMCast [5]

proposal appends an extra header in each data packet. The M-DS architecture differs in its overheads from these proposals for accomplishing the same goal, making it hard to do a comparative evaluation of overheads. The primary overheads of the M-DS architecture are in the form of signaling and increased packet hops for the multicast paths for some receivers. After the signaling phase, multicast data flows as it would in the case of DS for unicast. Therefore, we evaluate the signaling and bandwidth overhead of *edge-router branching* and *limited-core branching* techniques. The main results of this section can be summarized as follows. First, the total signaling overhead per membership discovery and termination on subnetworks under both techniques is similar for all the topologies tested and is small compared to the average per second routing overhead of 23 messages/second per BGP-4 router [14]. Second, the simplicity of the edge-router branching technique comes at the cost of extra bandwidth consumption in terms of packet hops during data transmission. The effect is more pronounced when receivers are clustered together. Third, the minimal controllable additional complexity of limited-core branching technique compared to edge-router branching saves the extra bandwidth consumption compared to edge-router branching by about 40%.

8.1 Simulation Environment

We used GT-ITM [15] to generate various topologies comprising of 744, 2646, and 6384 nodes each. Compared to the size of the Internet, these topologies seem small, but we believe that our simulations on these topologies produce results that prove the feasibility of deployment of both the techniques. The reason for this is the following. Signaling overhead depend on the number of domains in the topologies, not the number of nodes. In choosing the number of domains in the topologies, we used the results of a simple traceroute experiment we performed using random destinations across the globe. Our results indicated that most packets cross between 3 and 5 domains between source and destination. All our topologies have these properties. The details about the number of transit, stub, and total domains in these topologies

are shown in figure 16.

# Nodes	# Transit Domains	# Stub Domains Per Transit Domain	# Total Domains
744	4	3	12
2646	6	4	24
6384	7	5	35

Figure 16: Simulation Topologies

We implemented a custom simulator in *C* to conduct the simulations. Since the techniques move the multicast branching points to the edge of the domains for scalability, the simulations tested the extra packet hops incurred in the multicast tree by doing so. Also, since the techniques require signaling protocols to be run for membership discovery and termination on subnetworks, the simulations also estimated signaling overhead in terms of the number of messages. To estimate the extra packet hops and the signaling overhead for both the techniques, we experimented with two types of placement of receivers, random and clustered. For random placement of receivers, we only specify the number of receivers to be placed in the entire topology. For clustered placement, we specify the number of stub domains the receivers should be placed in and the number of receivers per stub domain. Our simulator implements Dijkstra’s shortest path algorithm for multicast routing. To estimate bandwidth overhead in terms of extra packet hops compared to multicast routing under both techniques, we placed receivers statically. To estimate signaling overhead for both techniques, we experimented with dynamic join/leave of receivers. We varied the mean time in group for receivers and experimented with uniform and exponential distributions of receivers’ time in the multicast group. We assumed a maximum of one receiver per subnetwork for all the test topologies, which in fact makes the results presented in this section conservative because in real scenarios, there will be many receivers on each subnetwork and signaling takes place once per membership discovery and termination for each multicast group. The simulations assume that all receivers join the same multicast group, and that a receiver join request is never denied for lack of resources.

8.2 Bandwidth Overhead

The graphs in figure 17 show the percentage extra packet hops (compared to the total hops using shortest path multicast routing) for various topologies for edge-router branching. The receivers are placed randomly and in clusters. The graph for clustered placement of receivers is not smooth because the extra hops are closely tied to the actual placement of receivers. We ran the tests on various topologies for any given number of nodes, with similar results. The overhead is less for random placement of receivers, about 10% extra packet hops compared to total hops for 140 receivers; compared to about 40% when the same number of receivers are clustered. This is expected because random placement of receivers does not lend itself to savings in packet hops because of lack of commonality in the path to the receivers. The overhead does not increase substantially beyond a certain percentage when more receivers are added to the clustered placement. This is evident from the fact that in going from 140 to 450 clustered receivers, the overhead only goes from 40% to 50%. This seems logical because adding more receivers to the same domain after a point would not increase the overhead further.

The graphs in figure 18 show the percentage extra packet hops for random and clustered placement of receivers for limited-core branching. The plots show three cases, when 1, 2, and 4 actual branching points are configured as special routers in each domain for the topology with 6384 nodes. For clustered placement of receivers, configuring 4 branching points as special routers in each domain brings down the percentage of extra hops to about 30% of actual packet hops under multicast routing, which is about 40% saving compared to edge-router branching technique. Also, note that configuring more branching points as special routers for random placement of receivers does not make a difference in the savings. This implies that decision about configuring branching point depends on the placement of receivers.

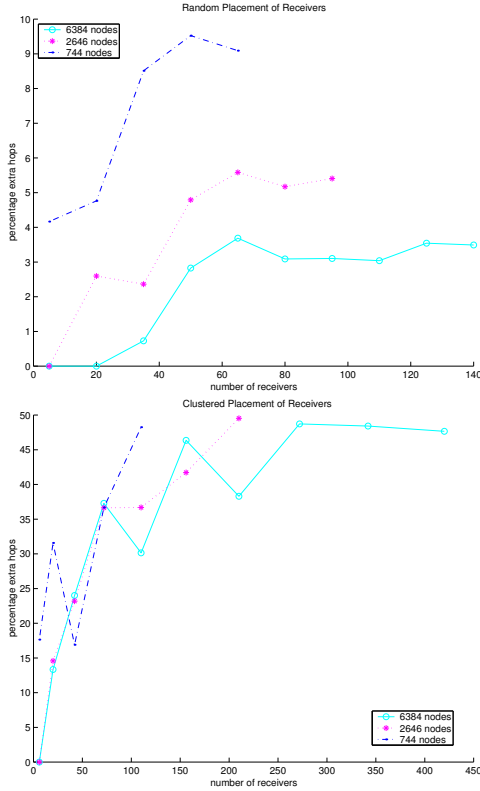


Figure 17: Percentage Extra Hops for Various Topologies for Edge-Router Branching

8.3 Signaling Overhead

To estimate the signaling overhead, the graphs in figure 19 show the number of signaling messages for when the receivers join. They show it for random and clustered placement of receivers for all three topologies for edge-router branching. The mean time between the arrival of receivers in the group for these plots is 60 minutes and the distribution of receivers' time in group is uniform. Since signaling is at the domain level, and all the simulation topologies have a similar number of domains, the signaling message overhead is almost independent of the number of nodes in the topology. As expected, the signaling overhead is more for random placement of receivers than for the clustered placement. The message overhead for 140 clustered placement receivers is about 50% less compared to similar number of randomly placed receivers. This is so because for clustered receivers, the signaling does not have to go all the way to the sender because of the presence of other receivers.

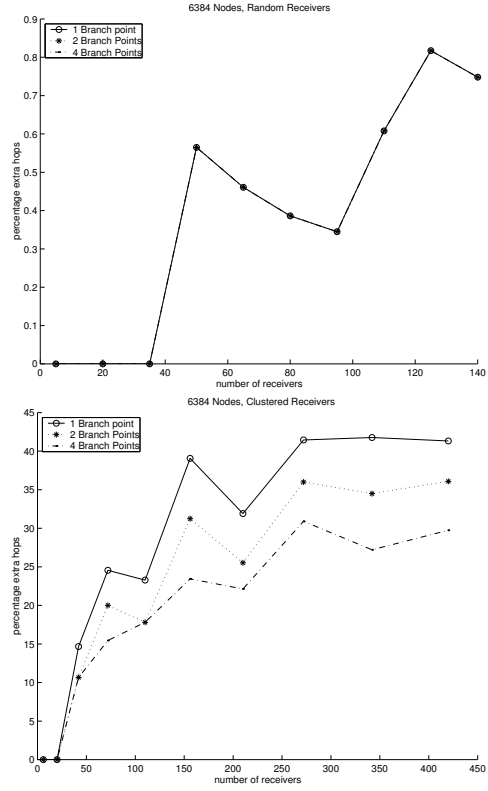


Figure 18: Percentage Extra Hops for Various Allowed Branching Points in Each Domain for Limited-Core Branching

We experimented with exponential distribution as well with the mean time in group ranging from 2 minutes to 2 hours, the results were almost the same. Since the protocols involve a similar amount of signaling overhead for both join and leave of each receiver, the corresponding plots for the overhead when receivers leave were identical. The signaling in the case of limited-core branching involves only as many additional messages as the number of branching points in each domain. Hence the signaling overhead for it is expected to be similar to that in the case of edge-router branching.

Assume 100 receivers join the same multicast group, one every 2 minutes. From figure 19, for random placement of receivers, the approximate number of join messages per receiver is about 7 (700 messages/100 receivers). The corresponding figure for clustered placement is $375/100 = 3.75$. The number of receivers per second is about approximately 1 for both cases. Using these two

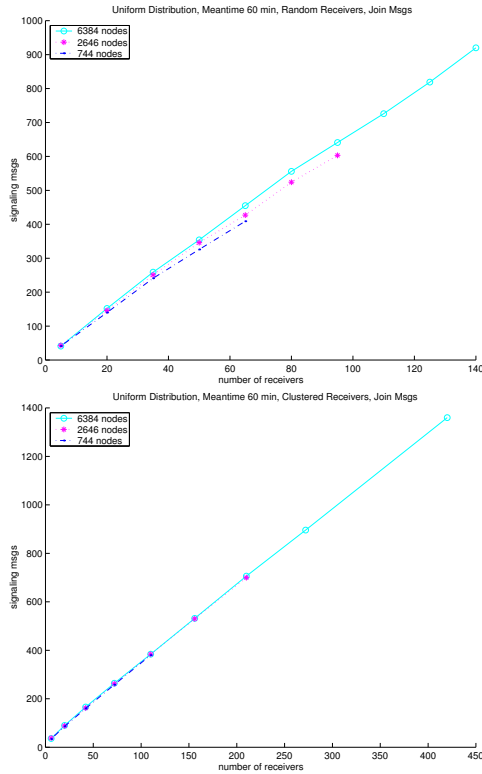


Figure 19: Signaling Messages for Receiver Join for Various Topologies

numbers for each of random and clustered case, we find that the number of messages per second is about 7 for the random case and 3.75 for the clustered case. Each BGP-4 router processes about 23 messages/second on an average [14]. For edge-router branching, all domains combined that are involved in getting data to the receiver process between 3.75 to 7 messages per second for our topologies, a small overhead. The results are expected to be similar for limited-core branching also because of the similarity of the signaling protocols. This implies that the signaling overhead is minimal for both techniques in our architecture.

As mentioned before, the above results on the signaling are conservative because we assume only a maximum of one receiver per subnetwork. In reality, because of the manner in which IGMP works, signaling would be carried out once for every membership initiation and termination on a subnetwork. For all other receivers on the same subnetwork, there is no signaling overhead.

9 Conclusions

The DS framework is a scalable way to provide QoS for unicast communication. Dynamic join/leave of receivers in multicast poses unique challenges in utilizing the DS framework to provide support for multicast communication. This paper proposes M-DS, a scalable architecture that consists of two techniques for providing support for multicast communication in a DS network. Both the techniques, *edge-router branching* and *limited-core branching* define two signaling protocols to be run upon membership discovery and termination on subnetworks. Signaling allocates and deallocates resources and sets up state in the relevant routers to be used during data transmission. Upon the completion of signaling, data flows in as in the case of DS for unicast.

Both the techniques keep the core of the network simple. Traffic aggregation gives the architecture its scalability just as in the DS framework. The edge-router branching technique exploits the multicast scalability at a domain granularity and moves all the branching points required to graft new receivers to the existing multicast tree to the ingress of the domain. The limited-core branching technique allows a limited number of branching points to exist in a domain. This introduces a small amount of extra complexity but exploits multicast scalability in a more effective manner. These two techniques are inter-operable and can be used individually in various domains. None of the techniques involve any per packet bandwidth overhead during data transmission. This is because they do not introduce any extra headers in the data packets. The architecture allows each domain to run its individual IP multicast protocol. Both the techniques use the IP multicast routing state already set up in individual domains for forwarding packets during actual data transmission.

Finally, incorporating these techniques in the DS framework requires that the functionality of all the routers be enhanced. But for actual data transmission, only a few routers would need to use this enhanced functionality. For example, for edge-router branching technique, only ingress routers of each domain need to be able to serve as a branching point and the original branching point routers

in the domain need to send messages to the BB of their domain and know how to forward the multicast packets using the existing IP multicast forwarding table. All other routers can continue to function as before. In case of limited-core branching, a few core routers may also serve as a branching points, hence their functionality is also similar to the ingress router in edge-router branching.

References

- [1] B. Quinn and K. Almeroth. IP Multicast Applications: Challenges and Solutions. IETF Draft. June 1999.
- [2] K. Almeroth. Validating the Multicast Mystique. IEEE Infocom. Apr 2001.
- [3] S. Blake et. al. An Architecture for Differentiated Services. RFC 2475. Dec 1998.
- [4] R. Bless and K. Wehrle. IP Multicast in Differentiated Services Network. Internet Draft. Sept 1999.
- [5] A. Streigel and G. Manimaran. A Scalable Approach to Diffserv Multicasting. ICC 2001.
- [6] X. Li, M. Ammar, and S. Paul. Video Multicast over the Internet. IEEE Network Magazine. April 1999.
- [7] B. Teitelbaum P. Chimento. QBone Bandwidth Broker Architecture (work in progress). QBone Bandwidth Broker Work Group.
- [8] Y. Rekhter and T. Li. A Border Gateway Protocol 4. RFC 1771. Mar 1995.
- [9] D. Meyer and B. Fenner. Multicast Source Discovery Protocol (MSDP). Internet Draft. May 2001.
- [10] D. Estrin et. al. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. RFC 2362. June 1998.
- [11] R. Braden et. al. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205. September 1997.
- [12] C. Perkins. IP Encapsulation within IP. RFC 2003. Oct 1996.
- [13] J. Postel. Internet Protocol Specification. RFC 791. Sept 1981.
- [14] Internet Performance Measurement and Analysis. <http://www.merit.edu/ipma.trends>.
- [15] K. Calvert, M. Doar, and E. Zegura. Modeling Internet Topology. IEEE Communication Magazine. July 1997.