Extending BGMP for Shared-Tree Inter-Domain QoS Multicast [⋆]

Aiguo Fei and Mario Gerla

Department of Computer Science
University of California, Los Angeles, CA 90095, USA
{afei,gerla}@cs.ucla.edu

Abstract. QoS support poses new challenges to multicast routing especially for inter-domain multicast where network QoS characteristics will not be readily available as in intra-domain multicast. Several existing proposals attempt to build QoS-sensitive multicast trees by providing multiple joining paths for a new member using a flooding-based search strategy which has the draw-back of excessive overhead and may not be able to determine which join path is QoS feasible sometimes. In this paper, first we propose a method to propagate QoS information in bidirectional multicast trees to enable better QoS-aware path selection decisions. We then propose an alternative "join point" search strategy that would introduce much less control overhead utilizing the root-based feature of the MASC/BGMP inter-domain multicast architecture. Simulation results show that this strategy is as effective as flooding-based search strategy in finding alternative join points for a new member but with much less overhead. We also discuss extensions to BGMP to incorporate our strategies to enable QoS support.

1 Introduction

Over the years, many multicast routing algorithms and protocols have been proposed and developed for IP networks. Several routing protocols[20, 17, 3, 12] have been standardized by the IETF (Internet Engineering Task Force), with some of them having been deployed in the experimental MBone and some Internet Service Providers'(ISP) networks[1]. Some of these protocols[20, 17] are for intra-domain multicast only while the others[3, 12] have scalability limitation and/or other difficulties when applied to inter-domain multicast though they were designed for that as well.

To provide scalable hierarchical Internet-wide multicast, several protocols are being developed and considered by IETF. The first step towards scalable hierarchical multicast routing is Multiprotocol Extensions to BGP4(MBGP)[4] which extends BGP to carry multiprotocol routes. In the MBGP/PIM-SM/MSDP architecture[1], MBGP is used to exchange multicast routes and PIM-SM(Protocol Independent Multicast - Sparse Mode) [12] is used to connect group members across domains, while another protocol, MSDP(Multicast Source Discovery Protocol)[15], is developed to exchange

[⋆] © Springer-Verlag. In Proceedings of the Ninth International Workshop on Quality of Service (IWQoS 2001), Karlsruhe, Germany, June 2001. This work was supported in part by grants from Cisco Systems and NSF.

information of active multicast sources among RP (Rendezvous Point) routers across domains.

The MBGP/PIM-SM/MSDP architecture has scalability problems and other limitations, and is recognized as a near-term solution[1]. To develop a better long-term solution, a more recent effort is the MASC/BGMP architecture[16, 21]. In this architecture, MASC(Multicast Address Set Claim)[13] allocates multicast addresses to domains in a hierarchical manner and BGMP(Border Gateway Multicast Protocol)[21] constructs a bidirectional group-shared inter-domain multicast tree rooted at a root domain. The bidirectional shared tree approach is adopted because of its many advantages[16].

In recent years, great effort has been undertaken to introduce and incorporate quality of service(QoS) into IP networks. Many multicast applications are QoS-sensitive in nature, thus they will all benefit from the QoS support of the underlying networks if available. The new challenge is how to build multicast trees subject to multiple QoS requirements. However, all the existing multicast protocols mentioned above are *QoS-oblivious*: they build multicast trees based only on connectivity and *shortest-path*. QoS provision is thus “opportunistic”: if the default multicast tree can not provide the QoS required by an application then it is out of luck.

Several QoS sensitive multicast routing proposals [8, 14, 22, 10] attempt to build QoS-sensitive multicast trees by providing multiple paths for a new member to choose in connecting to the existing tree through a “local search” strategy which finds multiple connecting paths by flooding TTL(Time-To-Live)-controlled search messages. In these alternate path search strategies, connecting path selection is solely based on QoS characteristics of the connecting paths. This may not necessarily yield the best connecting path selection and sometimes may not provide a QoS feasible path at all. Moreover, the flooding nature of local search strategy can cause excessive overhead if the TTL gets large.

In this paper, first we argue that the limitations of path selection with current proposals originates from the lack of QoS information concerning the existing multicast tree (for example, maximum inter-member delay and residual bandwidth of the existing tree). We then propose a method to propagate QoS information in group-shared bi-directional multicast trees so that better routing decisions can be made. In the mean time, we propose a new alternative “join point” search strategy that can significantly reduce the number of search messages utilizing the root-based feature of the recent MASC/BGMP inter-domain multicast architecture, which feature is also shared by other multicast routing protocols including CBT (Center Based Trees)[3] and PIM-SM[12].

In our new “scoped on-tree search” strategy, the on-tree node receiving a *join* message from a new member starts a search process to notify nearby on-tree nodes to provide alternative connecting points for the new member if the default one is not QoS feasible. This strategy starts a search only if the default route is not feasible and the on-tree search nature will only introduce limited overhead. Simulation results show that this strategy is as effective as flooding-based search strategy in finding alternative connecting “points” for a new joining member but with much less overhead.

Our work was motivated by the realization of the limitations of current alternate path selection method and excessive overhead of flooding based search strategies. The purpose of this paper is not to propose a new multicast routing protocol, but rather to show how our QoS information propagation method and alternative join point search strategy can be incorporated into BGMP to extend it for QoS support. Our methods are general in principle and can be applied to other protocols such as PIM-SM or other future routing protocols for QoS support as well.

The rest of this paper is organized as follows. Section 2 reviews the current Internet multicast architecture especially the MASC/BGMP architecture, QoS-aware multicast routing and several existing proposals. Section 3 discusses the limitations with current local-search-based QoS multicast routing proposals and Section 4 proposes a QoS information propagation mechanism for shared-tree multicast to address those limitations. Section 5 proposes our improved alternative join point search strategy with simulation results. Section 6 discusses modifications and extensions to BGMP to support QoS and Section 7 concludes the paper.

2 Background and Related Work

2.1 Current IP Multicast Architecture

In the current IP multicast architecture, a host joins a multicast group by communicating with the designated router using Internet Group Membership Protocol (IGMP[9]) (by sending a membership report or answering a query from the router). To deliver multicast packets for a group, IP multicast utilizes a tree structure which is constructed by multicast routing protocols. In MOSPF[17], routers within a domain exchange group membership information. Each router computes a source-based multicast tree and corresponding multicast tree state is installed. In CBT[3] or PIM-SM[11, 12], a group member sends an explicit join request towards a core router or an RP router. The request is forwarded hop-by-hop until it reaches a node which is already in tree or the core/RP, and multicast states are installed at intermediate routers along the way.

The Internet consists of numerous Autonomous Systems (AS) or domains, which may be connected as service provider/customers in a hierarchical manner or connected as peering neighbors, or both. Normally a domain is controlled by a single entity and can run an intra-domain multicast routing protocol of its choice. An inter-domain multicast routing protocol is deployed at border routers of a domain to construct multicast trees connecting to other domains. A border router capable of inter-domain multicast communicates with its peer(s) in other domain(s) via inter-domain multicast protocols and routers in its own domain via intra-domain protocols, and forwards multicast packets across the domain boundary. Currently there are two prominent inter-domain multicast protocol suits: MBGP/PIM-SM/MSDP and MASC/BGMP[1, 16, 21].

In this paper, we are concerned with multicast routing at the inter-domain level. By **receiver** or **multicast group member**, we refer to a router that has multicast receiver(s) or multicast group member(s) in its domain. We are not concerned with how a receiver host joins or leaves a group and how an intra-domain multicast tree is constructed. Similarly, by **source**, we refer to a router that has a multicast traffic source host in its domain.

2.2 The MASC/BGMP Architecture

In the MASC/BGMP[16, 21] architecture, border routers run BGMP to construct a bidirectional "shared" tree for a multicast group. The shared tree is rooted at a root domain that is mainly responsible for the group (e.g., the domain where the group communication initiator resides). BGMP relies on a hierarchical multicast group address allocation protocol (MASC) to map a group address to a root domain and an inter-domain routing protocol (BGP/MBGP) to carry "group route" information (i.e., how to reach the root domain of a multicast group).

MASC is used by one or more nodes of a MASC domain to acquire address ranges to use in its domain. Within the domain, multicast addresses are uniquely assigned to clients using intra-domain mechanism. MASC domains form a hierarchical structure in which a child domain (customer) chooses one or more parent (provider) domains to acquire address ranges using MASC. Address ranges used by top-level domains (domains that don't have parents) can be pre-assigned and can then be obtained by child domains. This is illustrated in Fig. 1, in which A, D, and E are backbone domains, B and C are customers of A while B and C have their own customers F and G, respectively. A has already acquired address range 224.0.0.0/16 from which B and C obtain address ranges 224.0.128.0/124 and 224.0.1.1/25, respectively.

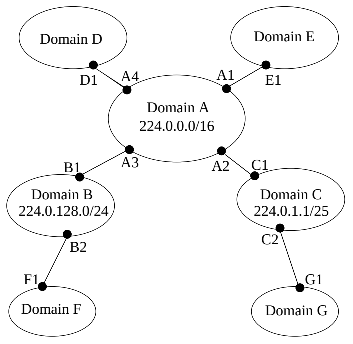

Fig. 1. Address allocation using MASC, adopted from [16].

Using this hierarchical address allocation, multicast "group routes" can be advertised and aggregated much like unicast routes. For example, border router B1 of domain B advertises *reachability* of root domains for groups in the range of 224.0.128.0/124 to A3 of domain A, and A1(A4) advertises the aggregated 224.0.0.0/16 to E1(D1) in domain E(D). Group routes are carried through MBGP and are injected into BGP routing tables of border routers. BGMP then uses such "group routing information" to construct shared multicast tree and distribute multicast packets.

BGMP constructs a bidirectional shared tree for a group rooted at its root domain through explicit join/prune as in CBT and PIM-SM. A BGMP router in the tree maintains a *target list* that includes a *parent target* and a list of *child targets*. A parent target is the next-hop BGMP peer towards the root domain of the group. A child target is either a BGMP peer or an MIGP (Multicast Interior Gateway Protocol) component of this router from which a join request was received for this group. Data packets received for the group will be forwarded to all targets in the list except the one from which data packet came. BGMP router peers maintain persistent TCP connections with each to exchange BGMP control messages (join/prune, etc.).

In BGMP architecture, a source doesn't need to join the group in order to send data. When a BGMP router receives data packets for a group for which it doesn't have forwarding entry, it will simply forward packets to the next hop BGMP peer towards the root domain of the group. Eventually they will hit a BGMP router that has forwarding state for that group or a BGMP router in the root domain. BGMP can also build source-specific branches, but only when needed (i.e., to be compatible with source-specific trees used by some MIGPs), or to construct trees for source-specific groups. A source-specific branch stops where it reaches either a BGMP router on the shared tree or the source domain. This is different from the source trees by some MIGPs in which source-specific state is setup all the way up to the source.

2.3 QoS-Aware Multicast Routing

Currently the Internet only provides shortest-path routing based on administrative costs and policies. Providing QoS in this routing architecture is “opportunistic”: if the default route can not provide the QoS required by an application then it is out of luck. To effectively support QoS, QoS routing may be a necessity in future QoS-enabled IP networks. Two main issues involved in QoS routing are how to compute or find QoS specific paths and how to enforce their use. Because of the network-stateless nature of conventional IP networks, the second issue (i.e., enforcing specific paths, which are different from the default unicast path, for different connections) might be a more difficult one to tackle within the current Internet architecture. In this regard, multicast is more suited for QoS-aware routing than unicast connections because the network maintains multicast states which provide a natural way to “pin-down” a QoS-feasible path or tree.

A multicast tree grows and shrinks through joining and pruning of members. Several QoS sensitive multicast routing proposals [8, 14, 22, 10] attempt to provide multiple paths for a new member to choose in connecting to the existing tree, hoping that one of them is QoS feasible. We call a join path QoS feasible if it satisfies the specific QoS requirements (for example, has enough bandwidth). In these proposals, some kind of search mechanism is used to explore multiple paths for the new member. YAM[8] proposes an inter-domain join mechanism called “one-to-many join”. In QoSMIC[14], there are two ways for a new router to select an on-tree router to connect to the tree. One is called “local search” which is very similar to “one-to-many join” in YAM[8] and is further studied in [22]. In “local search”, the new joining router floods a BID-REQUEST message with scope controlled by the TTL(Time-To-Live) field, an on-tree routers receiving search messages unicasts BID messages back to the new member. The other one is called “multicast tree search” in which the new joining router contacts a manager which then starts a “bidding” session by sending BID-ORDER messages to the tree, a subset of routers which receive the message become candidates. A candidate router unicasts a BID message to the new member. In both cases, a BID message “picks up” QoS characteristics of the path and the new member collects BID messages and selects the “best” candidate according to the QoS metrics collected, then it sends JOIN message to setup the connection. One drawback of both YAM and QoSMIC is the large amount of messages flooded, and consequently the large number of nodes which have to participate in the route selection process. In another proposal, QMRP[10], a more elaborate search strategy is proposed to restrict the number of nodes involved and the amount of search messages flooded. In YAM and QoSMIC, messages are only controlled by TTL. While in QMRP, they are also controlled by restricting branching degree – the number of REQUESTs that can be sent to neighbors by a node; more importantly, a REQUEST message is only forwarded to nodes that have the *required resources*.

The “multicast tree search” in QoSMIC[14] was also proposed to limit the overhead of flooding: a joining node starts a local search with a small TTL first, it contacts a tree manager to start a tree search if the local search fails. However this tree search strategy has its own drawbacks: (1)centralized manager may have to handle many join requests and be a potential hot spot; (2)there is no easy scalable way to distribute the information of mapping of multicast groups to tree managers for inter-domain multicast – this is one motivation for the development of inter-domain multicast protocols instead of simply using PIM-SM; (3)a manager has to multicast a search message to the group which is considerably expensive since each on-tree router has to decide whether it should be a candidate[14] (unless the manager has global network topology and group membership information so it can select a few connecting candidates for the new member itself – the problem is, though group membership information can be obtained and maintained, it is not possible with global network information in inter-domain multicast), and it is not easy for an existing tree node to do so because it needs some kind of distance information [14] (to the new member) which the inter-domain routing protocol (BGP) is unable to provide or such information is very coarse.

Among these proposals, YAM’s one-to-many join is specifically proposed for inter-domain multicast while the others are targeted for both intra-domain and inter-domain. For routing within an AS domain, a link-state routing protocol is often used. It is possible that intra-domain QoS-sensitive routing in future QoS-enabled networks is still link-state-based[2]. A link-state routing protocol gives all network nodes the complete knowledge of network connectivity, and QoS characteristics in case of QoS-sensitive routing. Thus a local search strategy might be neither necessary, nor technically attractive, in intra-domain QoS multicast. One might argue that, QoS information propagated by a link-state protocol wouldn’t be as up-to-date and accurate as what discovered by local search. The counter argument is that, if QoS multicast gains widespread use, then the overhead caused by local search of the many new joins would be more than that of flooding QoS link-state information to achieve the same timeliness and accuracy. However, for inter-domain multicast, link-state-based routing is not an option today and a local search or other search strategies must be pursued in constructing QoS multicast trees.

3 Limitations of Current QoS Multicast Routing Proposals

Besides the drawback of control overhead, there are other limitations with the several current QoS multicast routing proposals discussed in Section 2.3. Consider a multicast tree and a new joining member as illustrated in Fig. 2. Discussions here apply to both intra-domain and inter-domain multicast. In inter-domain multicast this paper is focused on, a link between two on-tree neighbors in the figure can be an actual physical link if these two are just one hop away, but can also be a logical link which may have several other nodes in between (for example, two intra-domain peers as to be discussed later). Existing QoS-sensitive multicast routing proposals[8, 14, 22, 10] provide multiple connecting paths for a new joining member to choose. For example, new member N can connect to the tree by connecting to node A or B or C. Also note that there are two connecting paths to node B. This is possible if a search strategy as in [10] is used, or if BID-REQUEST messages in “local search”[14] are allowed to “pick up” multiple paths to an on-tree node and BID messages are sent back hop-by-hop according to the paths picked up.

In these routing schemes, a connecting candidate is selected according to the QoS properties of the connecting paths. Now consider the four available paths in Fig. 2. Assume that the QoS metric we care about here is delay and the connecting path from A to N has the shortest delay. Thus A would be chosen as the candidate for N to connect to.

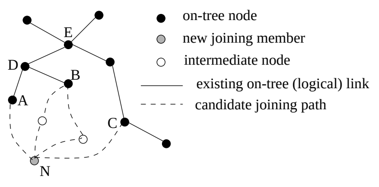

Fig. 2. Multiple connecting paths.

However, A might have the shortest delay to N, but might not be the best candidate after all; for example, the link between A and D is a slow one with long delay. Now assume the QoS metric in question is bandwidth. It is reasonable to assume that an on-tree node can easily learn how much traffic it is receiving from the group (say, 256kb/s) and intermediate nodes can detect bandwidth availability. So this will not a be a problem for the local search scheme if the new joining member is a receiver only, though it is arguable how a local *resource availability* check would work in [10] before a REQUEST reaches an on-tree node and learns the bandwidth requirement. The problem arises if the new member is a source as we assume bidirectional shared-tree multicast here. It is easy to detect if the connecting path can accommodate the amount of traffic that is going to be injected by N, but how does N or A(or any other on-tree node) know if the existing tree can accept at node A(or any other on-tree node) the amount of traffic that is going to be injected by N?

In the above discussion, we illustrate two problems with candidate selection based solely on QoS properties of connecting paths: (1)the “best” connecting path might not necessarily be the best choice; (2)QoS information collected by local search itself is not enough to determine how and whether a new member can be connected to the existing tree if the new member is a source. We need some mechanism to convey the QoS properties and resource (e.g., bandwidth) availability of the existing multicast tree to member nodes to address the above two problems. Without such information readily available, a local search strategy would have to resort to a brutal-force flooding of connection request to the whole tree to find out.

One may think of using RSVP[7]. But because of its reliance on the routing protocol to construct a tree first, it is not a viable solution here. In RSVP, a multicast source periodically sends out PATH messages downstream along the tree, carrying traffic specifications. Receivers can then send RESV messages upstream to reserve required resources. As a signalling protocol, RSVP requires a tree already in place, while the problem here is how to construct a tree that is capable of supporting the QoS required.

4 QoS Information Propagation

In this section we discuss how QoS information can be propagated along a multicast tree to address the problems we have with local-search-based or other alternative path routing strategies. Here first we use bandwidth as an example, which is likely to be the most important metric to consider if QoS support is ever going to be realized. Other metrics, such as end-to-end delay and buffer spaces, can be easily accommodated by our scheme with little or no modification. The procedures discussed here apply to both BGMP-based inter-domain multicast and other bidirectional shared-tree multicast.

For an on-tree node to determine if it is a "feasible" connecting point for a new member, it maintain the following information: total bandwidth of traffic injected by existing members (B^t_{recv}) and the bandwidth that can be injected to the tree "at this point" (B^t_{feasb}, "feasible bandwidth"). Let's look at a very simple tree with four nodes as illustrated in Fig. 3. Assume A and D are source nodes and each has 0.2Mb/s data to transmit, and every (logical) link has available bandwidth 1Mb/s on both directions before the multicast group is using any bandwidth. So A and D are receiving 0.2Mb/s from the group and B and D are receiving 0.4Mb/s from the group. It is also easy to calculate that B^t_{feasb} at node B is 0.6Mb/s: the residual bandwidth from B to A or D is 0.8Mb/s and it is 0.6Mb/s from B to C. B^t_{feasb} at node A and D is also 0.6Mb/s while B^t_{feasb} at node C is 0.8Mb/s. With such information, when receiving a connection request from a new joining receiver member (say, N), a node (say, C) can inform the new member the required available bandwidth and determine whether the connecting path (from C to N) has enough bandwidth. If the new member is also going to be a source, node C also has to check if it can accept the amount of traffic that is going to be injected by N.

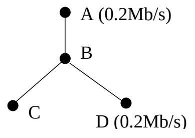

Fig. 3. A tree with four nodes.

Now we describe how such information can be maintained at each node and propagated along the tree. An on-tree router will maintain B_{recv} and B_{feasb} for each neighbor. B_{recv} is the bandwidth of traffic received from that neighbor, and B_{feasb} is the amount of traffic that neighbor can accept (into the tree towards downstream). In BGMP, a neighbor is a *target* in the *target list* if it is a peer in other domain, or another BGMP router within the same domain as to be discussed in more details later. A node periodically sends messages to a neighbor with bandwidth information summarized from B_{recv} and B_{feasb} information of all other neighbors: B_{recv} sent will be the sum of B_{recv} for all other nodes plus bandwidth of local sources, and B_{feasb} sent will be the minimum of all other nodes,

$$\begin{cases} B_{recv} = B(local) + \sum_j B_{recv}(j), \forall neighbor\ j\ except\ x \\ B_{feasb} = min\{B_{feasb}(j), \forall neighbor\ j\ except\ x\}. \end{cases} \tag{1}$$

When node j receives a B_{feasb} from neighbor i, it should replace it with the minimum of B_{feasb} and $B_{available}(link\ j \rightarrow i)$:

$$B_{feasb}(i) = min\{received\, B_{feasb}(i),\ B_{available}(link\ j \rightarrow i)\} \tag{2}$$

In the example discussed above, node A sends B:

$$\begin{cases} B_{recv} = 0.2Mb/s \\ B_{feasb} = \infty, \end{cases} \tag{3}$$

since A has no other neighbor nodes. Information B stores for A will be:

$$\begin{cases} B_{recv}(A) = 0.2Mb/s \\ B_{feasb}(A) = min\{\infty,\ 1Mb/s - 0.2Mb/s\} = 0.8Mb/s. \end{cases} \tag{4}$$

Node B sends C:

$$\begin{cases} B_{recv} = B_{recv}(A) + B_{recv}(D) = 0.4Mb/s \\ B_{feasb} = min\{B_{feasb}(A), B_{feasb}(D)\} = 0.8Mb/s. \end{cases} \tag{5}$$

In the above example, we illustrated how *group shared* bandwidth information is maintained at tree nodes and propagated along the tree. The same can be easily done with delay information. Now assume what we care is the maximum inter-member delay. A node i maintains the following information for each neighbor j: $D^{fr}(j)$=maximum delay from the farthest member to i through j, and $D^{to}(j)$=maximum delay to reach the farthest member through j in the branch down from j. Summary information is maintained as:

$$\begin{cases} D^{fr}_{max} = max\{D^{fr}(j), \forall neighbor\ j\} \\ D^{to}_{max} = max\{D^{to}(j), \forall neighbor\ j\}. \end{cases} \tag{6}$$

The information for i to propagate to neighbor x is:

$$\begin{cases} D^{fr} = max\{D^{fr}(j) + delay(i \to x), \forall neighbor\ j\ except\ x\} \\ D^{to} = max\{D^{to}(j), \forall neighbor\ j\ except\ x\}. \end{cases} \tag{7}$$

When neighbor x receives the D^{to} from i, it gets

$$D^{to}(i) = received\ D^{to}(i) + delay(x \to i). \tag{8}$$

4.1 Discussions

Source-specific QoS information. BGMP supports source-specific branch which is recursively built from a member sending such a request and stops where it reaches either a BGMP router on the shared tree or a BGMP router in the source domain. To facilitate QoS management on the source-specific branches, source-specific QoS information can be propagated to and maintained at BGMP routers where source-specific group states reside. To do so, a source can send a source-specific QoS update down the tree; all members forward this update with QoS information related to this source in addition to group shared summary to downstream nodes. Any node with source-specific state for this source will store such source-specific QoS information locally while all other nodes don't need to do so and only need to pass it down.

Intra-domain BGMP peers. In BGMP, internal BGMP peers (BGMP border routers within the same multicast AS) participate in the intra-domain tree construction through an intra-domain multicast protocol. They don't exchange *join* or *prune* messages with each other as with external peers. To support QoS, they would have to exchange QoS update information as above. They connect to each other through an intra-domain multicast tree and can be considered as tree "neighbors" connected through virtual links.

When exchanging QoS update information with an internal peer, a tree node should only send “summary” for updates received from external neighbors. While it should send summary for all updates received including that via virtual links when exchanging QoS updates with external peers. Fig. 4 illustrates portion of a multicast tree with an intra-domain tree shown. When node A2 sends QoS information to node A1 and A3, it should only send summary for what received from node C1 and F1; but when it sends QoS information to node C1 and F1, it should include what received from A1 and A3.

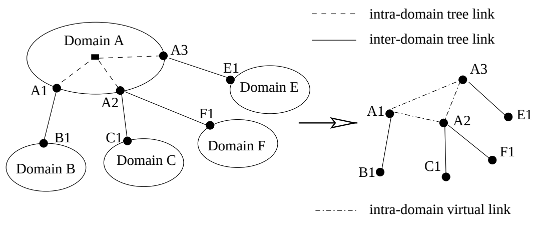

Fig. 4. Portion of an inter-domain multicast tree with an intra-domain tree shown.

In the above discussion we assumed that QoS information is propagated by periodical update message exchange between neighbors. The longest propagation delay for a change to reach all routers in the tree is about $D \times T$ in average and $2 \times D \times T$ for worst case, where T is the update timer and D is “diameter” of the tree (i.e., hop-count between two farthest nodes). This delay can be significant if the group gets large or spreads out far. When a member newly joins, especially if it is a source, such change should be reflected at all nodes as soon as possible. To do so, the new member could “order” an immediate flush of update down the whole tree. Any other significant change in the tree such as leaving of a source node or loss of connectivity should also prompt such immediate update flush.

So far we are only concerned with how QoS information is propagated and maintained. Apparently we need some mechanism to provide such information. For example, available bandwidth information on a “virtual link” between two internal peers may be obtained through a *bandwidth broker* as in DiffServ[5]. Availability of bandwidth and other resources may also subject to policy constraints. For example, even physically the available bandwidth between two external BGMP peers is 10Mb/s, a policy or service level agreement may specify that at most 2Mb/s can be used for a multicast or at most 1Mb/s can be used by a single multicast group. Some other metric (for example, delay) can be measurement-based.

5 Alternative Join Point Search

In the existing BGMP, to join a group, the new member sends a join message towards the root domain of the group. It is forwarded hop-by-hop and corresponding group states are installed at each hop, until it reaches a router which already has state for that group or a BGMP router of the root domain. This way, the join path to the existing tree is the unique default route, which may not be able to provide the QoS required. To effectively support QoS, here we propose a search strategy that can provide alternative “join points” in case the default one doesn’t work out. This strategy utilizes the “root-based” feature of the MASC/BGMP architecture: mapping of a multicast group to a root domain and the ability to provide a default route to the root domain.

5.1 Scoped On-Tree Search

Our proposed strategy is called *scoped on-tree search*. We discuss this strategy within the BGMP routing architecture and the discussion actually applies to CBT or PIM-SM architecture with little modifications. In this strategy, a new joining member sends join request for the group towards the root domain as usual and it is forwarded by other BGMP routers hop-by-hop. The join request collects QoS information along the way and will eventually hit a BGMP router on the tree or in the root domain. The on-tree (or root domain) router checks QoS feasibility based on the group-shared QoS information concerning the current tree and the QoS information collected by the join request. It sends an acknowledgment message to the new member if QoS requirements are met; otherwise it sends a search request to all neighbors in the tree, with a TTL (time-to-live) field to control the search depth. For example, if the TTL is 2, then a search request is forwarded at most 2 hops away from its sender. An on-tree nodes receiving a search request forwards the request to all neighbors except the one from which the request came, unless TTL reaches 0. It also sends back (by unicast) a reply message to the new member whose address is carried in the search request. Before sending a reply, a BGMP router does a QoS feasibility check if possible and sends reply only if the check is passed.

The way that a reply message is forwarded depends on the functionalities of the routing protocol. To effectively use this search technique, the routing protocol must be able to provide a *multicast path*, which can be different from unicast forwarding path especially in inter-domain cases, back to the joining new member. BGMP provides such functionalities. The protocol specification[21] specifies that: “For a given source-specific group and source, BGMP must be able to look up the next-hop towards the source in the Multicast RIB (routing information base)”. This means that a reply message can be effectively sent hop-by-hop by BGMP routers to the joining member since the new member’s address can essentially be treated as a source address and the next (multicast) hop can be looked up from multicast RIB. Along the way, BGMP routers can insert the corresponding QoS information. If the joining member doesn’t get an acknowledgment, it collects reply messages and selects one satisfying the QoS requirement (or the best one) to send a renewed join. This join message will not be forwarded along the default route; instead it will be forwarded along the route picked up by the reply message.

In Fig. **??**, a new joining member (N) sends a join message towards root domain until it reaches on-tree node B. B discovers that the default route cannot provide the required QoS (e.g., delay on that path is too much) or this connecting point cannot provide the required QoS (e.g., B cannot accept the amount of traffic that is going to be injected by N), it then sends a *tree-search* request to its neighbors with TTL=2. This message reaches nodes A, C, D and E. They then send reply messages back to N for N to select one to connect to.

5.2 Discussions

Our strategy is proposed to work with the existing routing protocol BGMP (may also work with others that provide similar functionalities), thus it relies on a hierarchical multicast addressing and forwarding (MASC/MBGP) architecture as BGMP. This hierarchical routing architecture eliminates the need and drawbacks of requiring a tree manager in the *multicast tree search* strategy proposed in [14], and provides a nature way to start an on-tree search. In our strategy, a search process is started by an on-tree node that a join request first reaches on its way towards the root domain, and is started only if the default route is not QoS feasible. The search process also only involves a small set of nodes in its neighborhood. Both these features help improve the scalability – or, say, "hurt" the scalability less as every QoS proposal imposes this and that additional requirements and ours is no exception.

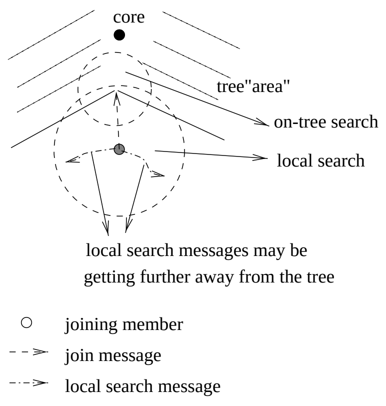

Fig. 6. Rationale behind on-tree search.

This strategy is based on the rationale that, the first on-tree node "hit" by a join request is the one close to the joining member, and other on-tree nodes close to this new member must be around the neighborhood. Flooding-based local search is "directionless": it floods search messages to all directions. It is the most effective strategy in the sense it can discover all on-tree nodes within a certain range. But it is also very expensive because it forwards many messages "blindly". This rationale is illustrated in Fig. 6. Simulation results in the next subsection shows that this rationale is indeed very reasonable and the search strategy based on it can find most of the on-tree nodes discovered by flooding-based local search.

5.3 Simulation

We compare the effectiveness and overhead of our strategy with flooding-based local search strategy using simulation. The metric used to compare *effectiveness* is the number of valid alternative join on-tree nodes (that can serve as a connecting point for the new member) being discovered, which is a good indication of the effectiveness since more alternative join nodes being discovered means a better chance for one of them to provide a QoS feasible join path. We only count nodes that are *valid* join points; for example, in Fig. **??**, if the join path from N to D has to go through node B, then D is not a valid connecting point (while B is) and will not be counted. This number is called *number of hits* in the following presentation.

One may attempt to compare the *quality* of the corresponding join paths. However, this would require to make assumptions about different QoS characteristics(available bandwidth, delay, etc.) of the networks at inter-domain level. Given the current status of QoS support in real networks, it is extremely difficult to make any realistic or representative assumptions to conduct such simulation.

We use the number of search messages being forwarded counted on a per-hop basis as the overhead. For example, in Fig. **??**, a message is forwarded by node B to D and then to A and E, the overhead is counted 3. We do not count the number of reply messages which is proportional to the number of on-tree nodes discovered in our on-tree search and is far less a concern compared with the number of search messages in flooding-based search.

In unrestricted local search, forwarding of a message is terminated only if its TTL reaches 0 or it hits an on-tree node. Thus a node may receive multiple search messages for the same search and forward them multiple times: in Fig. 7, joining node A sends a message to B and C, B and C would again forward the message to each other; message looping may also happen: node A forwards message to B then B to C, which then forwards it back to A, if initial TTL>3. This is necessary if we want to discover not only all nearby on-tree routers but also all the possible paths from the new member to them. This can generate significant overhead and may be hardly necessary. So we also simulate a restricted local search in which a node will forward a search message (for a given joining member) only once (to all neighbors except the one from which it came) and do so only if that neighbor is not already in the path the message travelled so far (loop prevention).

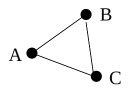

Fig. 7. A loop.

The topology used in our simulation consists of 3325 nodes obtained from a BGP routing table dump available from [18], which was collected by the Oregon Route Views Project[19]. A similar set of data was used for the simulation topology in [16]. For each simulation instance, we first construct a shortest-path tree (from the root domain) of a random multicast group of a given size. Then a random non-tree node is chosen as the new member and search strategies are simulated to find alternative join points. Each data point in the figures shown is the average of 1000 instances.

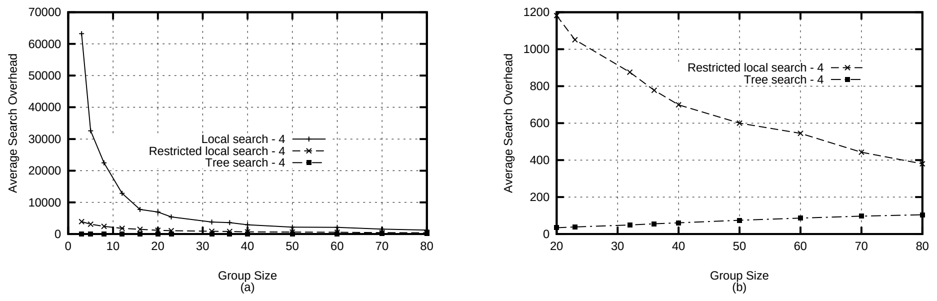

Fig. 8. Average overhead of local search and scoped on-tree search vs. group size. The number following a search method name is the TTL or search depth.

Fig. 8 shows the comparison of average search message overhead with varied group size. Group size varies from 3 to 80; 80 members at inter-domain level represents a fairly large group. The overhead of unrestricted local search is significantly larger than restricted local search and scoped on-tree search with the same TTL, especially when the multicast group is small in which case search messages forwarded may hardly "hit" any on-tree routers before the TTL reaches 0. When the group size increases, there is a better chance for a search message to reach an on-tree node and terminates earlier thus the overhead is significantly reduced. Restricted local search does help reduce the overhead significantly while requiring much more sophisticated message handling (i.e., a node needs to "remember" if it has already forwarded search message for a new joining node). But its overhead is still much larger than scoped on-tree search with the same search depth. Another fine point is that, on-tree search is only conducted when the default *shortest-path* route is not QoS feasible. So if the default route is always "good" enough (say, has enough bandwidth), then the search will not be necessary at all.

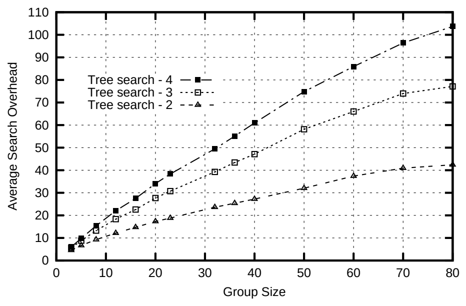

Fig. 9. Average overhead of scoped on-tree search with different search depths vs. group size.

Fig. 9 shows the overhead vs. group size of on-tree search with different search depths. One can see that, when group size increases, so does the overhead. The reason is apparent: when the group becomes larger, so does the *branch degree* of on-tree nodes and thus the number of search messages being forwarded increases.

Fig. 10 presents the average number of hits of local search and on-tree search vs. group size. Both restricted and unrestricted search with the same TTL will discover the same number of join points since both will reach all nodes within the specified TTL range. Fig. 10 shows that the number of hits by on-tree search is a little less than, but very close to, that of by local search with the same search depth (4). The average number of hits that are discovered by both local search and on-tree search with search depth 4 is also shown in Fig. 10. That number is a little less but very close to that of by local search or on-tree search. This is very interesting and very encouraging for on-tree search: with much less overhead, on-tree search are essentially discovering the same set of alternative "join points" that are discovered by local-search. The results for on-tree search also shows that, it can find more alternative join points when the search depth is increased. This comes at no surprise.

One may also observe that when group size is large enough, the number of valid alternative join points discovered will no longer increase but actually decrease a little when group size becomes larger. This is surprising at first thought, but actually there is a reason behind. When multicast group becomes larger, it tends to "spread" closer to the new joining member. A search strategy may be able to find more on-tree nodes, but many of them go through some other on-tree nodes to reach the new member, and the number of on-tree nodes that provide connecting paths without going through any other on-tree node still remains a small number. What that number will be for a specific router depends a lot on its connectivity and location on the network topology. For example, a router of a stub domain may always have only one connecting point to connect to while a router at the backbone may be able to find a fairly large number of alternative connecting points in its neighborhood. When a multicast group becomes sufficiently large (say, almost all nodes are members), then the average number of alternative connecting points will become close to the average node out-degree (when all neighbors are already in the tree). In topology used in our simulation, the average node out-degree is about 3.5.

6 Extending BGMP for QoS Support

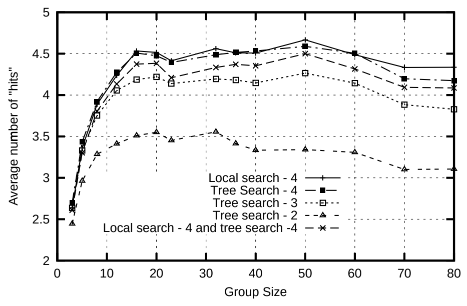

Fig. 10. Average number of “hits” vs. group size.

Here we discuss the modifications and extensions that need to be made to BGMP to support QoS. First of all, we need to use the method described in Section 4 to propagate QoS information along the tree. In BGMP, peers periodically exchange *join* messages to keep multicast group state (soft-state protocol). Thus periodical QoS update information can be piggybacked with join messages. Of course, message processing at each node now becomes more complicated and more state information has to be stored. While it is still an open question how often QoS update should be exchanged so that QoS information is accurate within a certain error margin or “useful” to a meaningful extend. To quickly convey some important QoS update (for example, a new joining source is transmitting at 100Kb/s), a new type of QoS update message should be introduced. This update message originates from a node where such update occurs (e.g., the new source node) and is multicasted along the tree. Any on-tree node receiving this message should process and forward it promptly. In order to scale to support large group, use of such update messages should be limited. In practice, dramatic changes that will trigger this message may not be very frequent after all.

In BGMP, a source can send data to a group without joining that group. For multicast with QoS support, there are reasons to require a source to join the group first before it transmits data, especially if some kind of guaranteed service is required. First of all, without joining a group, a source may not know if the rate of data it is going to transmit can be supported by the network or the existing tree (or delay requirement can be met). At the same time, it is not desirable for other members to see a new source disrupt the service quality in the existing tree. Access control, authentication requirement are among the other reasons: e.g., a group may not want to receive data from any others or only only want to communicate with someone with an authentic identity (for example, a company’s internal conference).

A couple of modifications to the join process are also necessary. First of all, to support alternate path routing, a BGMP must be able to support the on-tree search strategy and be able to forward a join message along a join path specified by the new joining member – this is necessary when the default shortest-path route is not feasible and a search procedure finds a feasible one. Moreover, distinctions between a join update and a new join should also be introduced: a new join involves searching for alternative join points if necessary and possibly resource reservation, admission control and an authentication process while a join update is to keep the state and renew the reservation. To determine whether the specified QoS requirements can be met, a new join also involves join acknowledgement and thus be a multiple-phase process. The leaving of a group member can be handled as the same (by sending prune messages). However, if the leaving node is a source node, it should order an immediate QoS information update before it leaves so the the corresponding resource used can be released and state information (about available bandwidth etc.) can be updated as soon as possible.

QoS support is a multi-facet problem involving traffic classification and/or prioritizing, buffering and scheduling, resource reservation, signaling, routing, and others. Some other modifications and extensions may also be necessary in addition to what we addressed so far, especially in the areas concerning admission control and resource reservation. These are beyond the scope of this paper in which we are focusing on the routing problem.

7 Conclusions

In this paper we have presented mechanisms to support QoS-aware multicast routing in group-shared bidirectional multicast trees within the BGMP inter-domain routing architecture. First we reviewed several existing QoS multicast routing proposals and discussed their drawbacks. We then analyzed some important limitations of these proposals imposed by making join path selection decisions based solely on the QoS characteristics of the alternative join paths, and proposed a QoS information propagation mechanism to address such limitations. After that, we proposed a new scoped on-tree search strategy to discover alternative "join points" for a new joining member if the default join path is not QoS feasible. Simulation results demonstrates that this strategy is as effective as the flooding-based local search strategy but with much less overhead. Finally we briefly discussed some modifications and extensions to BGMP that are necessary to effectively support QoS.

Contributions of our work lie in the followings: (a)a QoS information propagation method in bidirectional shared multicast tree to enable better routing decision making, (b)a low-overhead on-tree search strategy for alternative "join points", (c)applying these to BGMP-based multicast. Though they were presented within the BGMP architecture, both the QoS information propagation method and the alternative join point search strategy can be applied to other similar routing protocol for QoS support as well.

References

1. K. Almeroth, "The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet2 Deployment", IEEE Network, January/February 2000.
2. G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, and T. Przygienda, "QoS routing mechanisms and OSPF extensions", IETF RFC 2676, August 1999.
3. A. Ballardie, "Core based trees (CBT version 2) multicast routing: protocol specification", RFC2189, Septermber 1997.
4. T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol extensions for BGP-4", IETF RFC 2283, Feburary 1998.
5. S. Blake, D. Black, et al., "An architecture for differentiated services", RFC 2475, December 1998.

6. R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview", IETF RFC 1633, 1994.
7. R. Braden, L. Zhang, S. Berson, et al., "Resource reservation protocol (RSVP) – version 1 functional specification", IETF RFC 2205, September 1997.
8. K. Carlberg and J. Crowcroft, "Quality of multicast service (QoMS) by yet another multicast (YAM) routing protocol ", in *Proceedings of HIPARCH'98*, June 1998.
9. B. Cain, S. Deering, and A. Thyagarajan, "Internet group management protocol, Version 3", Internet draft: draft-ietf-idmr-igmp-v3-01.txt, Feburary 1999.
10. S. Chen, K. Nahrstedt, and Y. Shavitt, "A QoS-aware multicast routing protocol", in *Proceedings of IEEE INFOCOM'00*, March 2000.
11. S. Deering, D. Estrin, D. Farinacci, et al., "The PIM architecture for wide-area multicast routing", *IEEE/ACM Transaction on Networking*, Vol.4(2), pp.153-162, April 1996.
12. S. Deering, D. Estrin, D. Farinacci, et al., "Protocol independent multicast-sparse mode (PIM-SM): motivation and architecture", Internet draft: draft-ietf-idmr-pim-arch-05.txt{ps}, August 1998.
13. D. Estrin, M. Handley, and D. Thaler, "Multicast-Address-Set advertisement and Claim mechanism", IETF Internet draft: draft-ietf-malloc-masc-05.txt, July 2000.
14. M. Faloutsos, A. Banerjea, and R. Pankaj, "QoSMIC: quality of service sensitive multicast internet protocol", in *Proceedings of ACM SIGCOMM'98*, pp.144-153, September 1998.
15. D. Farinacci, Y. Rekhter, D. Meyer, P. Lothberg, H. Kilmer, and J. Hall, "Multicast source discovery protocol (MSDP)", IETF Internet draft: draft-ietf-msdp-spec-06.txt.
16. S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoğlu, D. Estrin, and M. Handley, "The MASC/BGMP architecture for inter-domain multicast routing", in *Proceedings of ACM SIGCOMM'98*, pp.93-104, September 1998.
17. J. Moy, "Multicast routing extensions to OSPF", RFC 1584, March 1994.
18. Measurement and Operations Analysis Team, National Laboratory for Applied Network Research: http://moat.nlanr.net/AS/.
19. Oregon Route Views Project: http://www.antc.uoregon.edu/route-views/.
20. C. Partridge, D. Waitzman, and S. Deering, "Distance vector multicast routing protocol", RFC 1075, 1988.
21. D. Thaler, D. Estrin, and D. Meyer, "Border gateway multicast protocol (BGMP): protocol specification", Internet Draft, draft-ietf-bgmp-spec-02.txt, November 2000.
22. D. Zappala, "Alternate path routing in multicast", in *Proceedings of IEEE INFOCOM'00*, March 2000.