The MASC/BGMP Architecture for Inter-domain Multicast Routing*

Satish Kumar†, Pavlin Radoslavov
Information Sciences Institute
University of Southern California
{kkumar,pavlin}@isi.edu

David Thaler
Electrical Engineering and Computer Science Dept
University of Michigan
thalerd@eecs.umich.edu

Cengiz Alaettinoğlu, Deborah Estrin‡, Mark Handley
Information Sciences Institute
University of Southern California
{cengiz,estrin,mjh}@isi.edu

Abstract

Multicast routing enables efficient data distribution to multiple recipients. However, existing work has concentrated on extending single-domain techniques to wide-area networks, rather than providing mechanisms to realize inter-domain multicast on a global scale in the Internet.

We describe an architecture for inter-domain multicast routing that consists of two complementary protocols. The Multicast Address-Set Claim (MASC) protocol forms the basis for a hierarchical address allocation architecture. It dynamically allocates to domains multicast address ranges from which groups initiated in the domain get their multicast addresses. The Border-Gateway Multicast Protocol (BGMP), run by the border routers of a domain, constructs inter-domain bidirectional shared trees, while allowing any existing multicast routing protocol to be used within individual domains. The resulting shared tree for a group is rooted at the domain whose address range covers the group's address; this domain is typically the group initiator's domain. We demonstrate the feasibility and performance of these complementary protocols through simulation.

This architecture, together with existing protocols operating within each domain, is intended as a framework in which to solve the problems facing the current multicast addressing and routing infrastructure.

1 Introduction

IP Multicast provides efficient one-to-many data distribution in an Internet environment, and also provides the functionality to logically group (and hence identify) a set of hosts/routers in a distributed fashion. Hence, it is an important mechanism to support many applications such as multimedia teleconferencing, distance learning, data replication and network games. The current infrastructure for global, Internet-wide multicast routing however faces some key problems.

Firstly, the existing multicast routing mechanisms broadcast some information and therefore do not scale well to groups that span the Internet. Multicast routing protocols like DVMRP[1, 2, 3] and PIM-DM[4] periodically flood data packets throughout the network. MOSPF[5] floods group membership information to all the routers so that they can build multicast distribution trees. Protocols like CBT[6, 7] and PIM-SM[8, 9] scale better by having the members explicitly join a multicast distribution tree rooted at a core router. However, the mechanism for distributing the mapping of a group to its corresponding core router requires flooding of the set of all routers that are willing to be cores.

Secondly, the current scheme used to assign multicast addresses to groups does not scale well. An IP multicast group is identified by a single IP address. Senders to the group use this address as the destination of packets to reach all the members of the group. The multicast addresses do not have any structure to them. A multicast group initiator typically contacts an address allocation application (e.g., the session directory tool, sdr[10]) and an address is randomly assigned from those not known to be in use. The assigned address is unique with high probability when the number of addresses in use is small, but the probability of address collisions increases steeply when the percentage of addresses

*This work was supported by NSF under grant NCR-9321043, DARPA under contract number DABT63-96-C-0054 and a Predoctoral Merit Fellowship from the University of Southern California.

†Kumar was the primary author of the paper, Radoslavov collaborated in the design of MASC and conducted all the simulations in this paper. Alaettinoglu, Estrin, Handley, and Thaler collaborated on the design of the overall architecture and protocol details. This work built upon earlier work by Handley et. al. on HPIM, Ballardie et. al. on CBT, and many of the key ideas came from discussions with Steve Deering, Dino Farinacci, Van Jacobson, and David Meyer.

‡Also with the Computer Science Department at the University of Southern California.

in use crosses a certain threshold and as the time to notify other allocators grows. Hence, a need has been recognized for a hierarchical multicast address allocation scheme[10] for the Internet.

The Internet today is an interconnection of networks administered by different organizations. The set of networks under administrative control of a single organization is referred to as an *Autonomous System* or *domain*. In this paper we describe an architecture for inter-domain multicast routing comprising two complementary protocols: the Multicast Address-Set Claim (MASC)[11] protocol and the Border Gateway Multicast Protocol (BGMP)[12]. Together with existing protocols operating within each domain, this is intended as a framework in which to solve the problems facing multicast addressing and routing.

MASC forms the basis for a hierarchical address allocation architecture. The domains running MASC form a hierarchy based on the structure of the existing inter-domain topology (e.g., campus area networks have a regional network as their parent, regionals have backbone networks as parent). MASC then dynamically allocates address ranges to domains using a *listen and claim with collision detection* approach. In this approach, child domains listen to multicast address ranges selected by their parent, select subranges from their parent's range and propagate the claims to their siblings. The claimers wait for a suitably long period to detect any collision, before communicating the *acquired* range to the domain's *Multicast Address Allocation Servers* (MAAS's)[13] and to other domains through the inter-domain routing protocol, BGP[14, 15] as *group* routes. MAAS's can then allocate individual multicast addresses to groups initiated in their domain.

BGMP (run by the border routers of a domain) uses the BGP group routes to construct multicast distribution trees for active multicast groups. These are known as *bidirectional shared trees*, and consist of the BGMP border routers of domains that lie on the path between the sender/receiver domains and the group's *root domain*. Each shared tree is rooted at the domain whose address allocation includes the group's address. Thus, the domain that injected a multicast address range into BGP is the *root domain* for those groups. Since an initiator of a group normally gets a multicast address from the range allocated to its local domain, the shared tree is rooted there. Intra-domain routing protocols such as PIM and DVMRP are used to forward data between the domain's group members and its BGMP border routers. These border routers in turn forward the data along the group's shared tree to reach members in other domains. Data can flow in either direction along this bidirectional tree[1].

Since inter-domain routing involves the use of resources in autonomously administered domains, the routing policy constraints of such domains need to be accommodated. In our architecture, polices for multicast traffic can be realized through selective propagation of group routes in BGP. This mechanism should be of operational benefit as it is the same as that used for unicast routing policy expression. In addition, bidirectional trees minimize *third-party dependency* policy issues. In other words, the communication between group members in two domains along the bidirectional tree does not depend, as much as possible, on the quality of paths to a third domain that does not lie on the path between the two domains.

In section 2, we describe the current inter-domain unicast routing infrastructure and how it relates to our proposed architecture for inter-domain multicast routing. We then enumerate the design requirements for efficient inter-domain multicast routing in Section 3. In section 4, we describe MASC, with simulations of its performance, and describe how the allocations from MASC are distributed using BGP. We describe BGMP in section 5 with simulations comparing the quality of bidirectional shared trees built by BGMP to distribution trees built by other multicast routing protocols.

[1] This is in contrast to *unidirectional* trees built by protocols like PIM-SM where data can flow only in a specified direction on each branch of the tree.

2 Background and Motivation

Inter-domain unicast routing To enable communication between domains in the Internet, border routers run an inter-domain unicast routing protocol. The one used in the Internet today is BGP. BGP border routers of neighboring domains establish TCP peerings with each other to exchange routing information in *update* messages. Update messages contain a set of *routes*, each comprising an address prefix of a destination network that is reachable from the border router together with attributes of the path to that destination network. When a router X advertises a route for R to a router Y, it means that the router Y can use X to reach the destination network R.

All the border routers of a domain peer with each other to exchange the routes received by them from external peers (i.e., peers in other domains). They then locally select the best route to each destination. The chosen route is then advertised by the border routers to the external peers.

BGP is currently being extended[16] to carry multiple types of routes for destinations and consequently allow multiple logical views of the routing table corresponding to each route type. This helps in supporting other network layer protocols like QoS and multicast routing protocols. For example, the multicast routing information in the logical view of the routing table called the Multicast Routing Information Base (M-RIB) would be used for RPF checks [2] so that the multicast routing protocols perform correctly even if the multicast topology is not congruent to the unicast topology.

The architecture presented in this paper uses a type of route in BGP that we call a *group route*. Group routes, injected into BGP by a MASC speaker, contain the multicast address ranges allocated to the domain by MASC, and hence implicitly bind each group address to its root domain. We refer to the portion of the routing table holding group routes as the Group Routing Information Base (G-RIB). BGMP uses the G-RIB information to construct a shared multicast distribution tree for a group rooted at the root domain for that group address. Hence BGP serves as a glue between MASC and BGMP by distributing the address allocation information from MASC to border routers of domains so as to enable BGMP to construct inter-domain multicast distribution trees for groups.

Address Allocation and Aggregation The number of networks in the Internet is growing at an exponential rate. Since the BGP unicast routing tables had entries on the order of the number of networks (address prefixes) in the Internet, it too was growing at an exponential rate. The above scaling problem was reduced by the deployment of Classless Inter-Domain Routing (CIDR)[17], which allows consecutive address prefixes to be combined into a single prefix thereby

[2] An RPF (Reverse Path Forwarding) check refers to the check made by some multicast routing protocols that a packet received from a source S came from the neighboring router that is the shortest path back to S.

reducing the number of routing table entries. For example, the address prefixes 128.8.0.0/16[3] and 128.9.0.0/16 can be *aggregated* to 128.8.0.0/15 as they differ only in their last (i.e. 16th) bit. When a border router X advertises an aggregate to border router Y, Y can reach hosts in all the component address prefixes of the aggregate via X. CIDR only achieves efficient aggregation to the extent that the unicast address prefixes are assigned to domains in a structured manner. With unicast address prefixes, this is currently achieved by static address assignments with limited success.

Unlike unicast addresses where an address represents a host or router, the number of multicast group addresses required in a domain is far more volatile (due to their logical nature). Hence, MASC dynamically allocates multicast address ranges to domains based on the usage patterns so as to achieve efficient address space utilization. In addition, dynamic allocation of address ranges makes it possible to achieve better aggregation (compared to the unicast case) of the group routes that MASC injects into BGP.

Routing policies The Internet is composed of a number of *provider* domains that offer as a service to facilitate data exchange between other domains. Provider domains vary in size; some operate in a single metropolitan area while others span continents. Larger provider domains offering national or inter-continental transit are typically known as *backbone* domains. Smaller provider domains are often customers of larger provider domains.

Provider-customer relationships in the Internet define policy constraints of domains to limit traffic that they are willing to carry. Typically, a provider domain allows transit traffic to or from its customer domains to pass through its networks. These policies are realized by BGP through selective propagation of the unicast routes. For example, if an unicast route R is not propagated by a border router X to its peer Y, then Y will not be aware that it can use X to reach the destinations represented by R. The border routers of a provider domain would typically advertise only routes corresponding to networks in its own domain and its customer domains. This ensures that only traffic destined either to a host in its own networks or in the networks of one of its customer domains will transit through it (apart from the traffic originated by hosts in these networks).

We propose to realize multicast policies through selective propagation of the group routes in BGP so that use of the provider's networks can be suitably restricted (similar to the unicast case). However, there is a limit to how many heterogeneous policies can be supported in the construction of a single multicast distribution tree for a group across domains. Multicast gains its efficiency by distributing packets over a common tree. The fragmentation of the tree by policy might at some point lead to the communication between the group members essentially devolving to unicast. Providers specifying multicast policy should be aware of the impact of baroque policies.

We next enumerate the requirements for efficient inter-domain multicast.

3 Requirements for inter-domain multicast

The key requirements for inter-domain multicast routing concern scaling, stability, policy, conformance to the IP Service Model [18, 19] and intra-domain multicast routing protocol independence.

[3] 128.8.0.0/16 refers to the set of addresses whose first 16 bits are 10000000 00001000 (128.8)

Scaling:

- **Multicast forwarding state:** The amount of state which must be distributed to permit global multicast forwarding should be minimal and scale well as the Internet expands. Where there are no receivers or senders, state for the group should be minimized.
- **Address allocation:** The address allocation scheme should scale well as the number of groups increases. The probability of address collision, as well as the delay in obtaining an address to assign to a group, should be small, consistent with best-effort architectural principles. An application- or session-level protocol should be able to detect and drop packets that it receives due to infrequent collisions to the extent required by that application.

Stability: Distribution trees should not be reshaped frequently, since this causes both additional control traffic as well as potential packet loss on sessions in progress. We believe that reducing protocol overhead is more important than maintaining optimal distribution trees.

Policy:

- **Policy model:** It is important for an inter-domain multicast routing protocol to have a policy model to control the flow of multicast traffic if it is to be widely deployed in the Internet.
- **Third-party dependency issues:** As much as possible, the communication between two domains should not rely on the quality of paths to a third domain if that third domain does not lie on the path between the two domains. Such dependencies are possible in protocols that build shared trees when the root of the shared tree lies in a third-party domain and the protocol requires that all packets go via the root before reaching group members along the shared tree. In addition, for administrative reasons it is desirable that the root of the shared tree not be in a domain that has neither group members nor senders.
- **Incongruent multicast and unicast topologies:** The multicast routing protocol should work even if the unicast and multicast topologies are not congruent. This can be achieved by using the M-RIB information in BGP.

Conformance to IP service model: In the IP Multicast service model, senders need not be members of a group to send data. This accommodates a wide range of applications; for example many small sensors reporting data to a set of servers without facing the overhead of receiving each other's traffic. Moreover, IP does not require signaling in advance of sending data. This has contributed to its applicability to bursty-source applications that expect to send whenever data is available but for which long-term per-source state is inefficient. The combination of the above requirements implies that any router must be able to forward a data packet towards group members if there is a multicast group in existence. It is therefore important that any required computation at the router to forward data packets to groups be fast enough so that data is not lost due to buffer overflows. This cannot be addressed by caching the information obtained from a remote lookup since data packets can be lost deterministically if the packet inter-arrival time is greater than the router state timeout period [20].

Intra-domain multicast routing independence: Intra-domain multicast routing protocol independence allows each domain the choice of which multicast routing protocol to run inside the domain. This allows each domain the autonomy to run a protocol that is best suited for its needs. It also allows a domain to upgrade to a newer version of a protocol while minimizing the effects on other domains.

In the context of these requirements, we will describe the MASC protocol for address allocation. In section 4.2 we describe the distribution of MASC address allocations through BGP, followed by a description of how BGMP uses this information to build multicast distribution trees in Section 5.

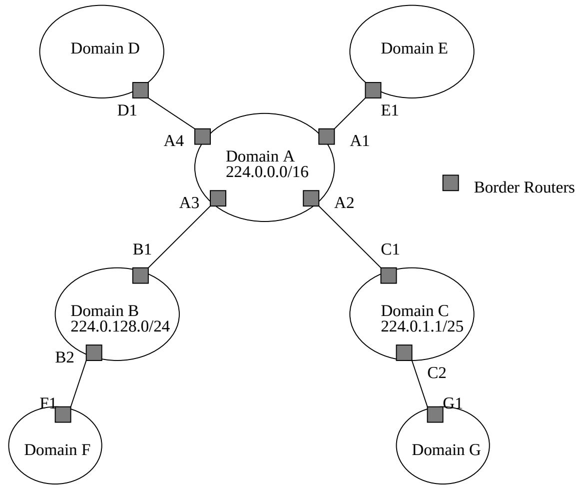


Figure 1: Address Allocation using MASC

4 Multicast address allocation to domains using MASC

One or more nodes (typically border routers) in a domain use MASC to acquire address ranges for use by Multicast Address Allocation Servers (MAAS's) in that domain. These servers coordinate with each other using intra-domain mechanisms [13] to assign unique multicast addresses to clients in their domain from address ranges provided, and to monitor the domain's address space utilization. When necessary, they communicate to the MASC nodes the need for more address space or to relinquish some of the acquired space.

We term a domain that has at least one node running MASC a *MASC domain*. MASC domains form a hierarchy that reflects the structure of the inter-domain topology. A domain that is a customer of other domains will choose one or more of those provider domains to be its MASC parent. Backbone MASC domains that are not customers of other domains typically do not have a parent MASC domain. We refer to a MASC domain that does not have a parent as a *top-level* domain. The hierarchy can be configured, or heuristics can be used to select the parent. For example, the MASC node could look up the routing table on one of its border routers to determine who its provider domain is (typically indicated by the default routing entry, if any, in the unicast routing table)

4.1 MASC address allocation overview

Consider a hierarchy of MASC domains as shown in figure 1. Domains A, D, and E are backbone providers. Domains B and C are regional providers and are customers of A. The regional providers, B and C have F and G as their customers respectively. We will show how B acquires an address range for use by its domain. We assume that backbone domain A has already acquired the address range 224.0.0.0/16 using MASC. Backbone domains acquire address ranges by a process similar to that of any other domain, which is explained at the end of this subsection.

MASC domains B and C have A as their parent. A advertises its address range, 224.0.0.0/16, to all its children. Child domain B *claims* an address range, say 224.0.1.0/24, from its parent domain's (A's) address space and informs its parent, as well as any directly-connected siblings, of the claim. A then propagates this claim information to its other children (the algorithm used by B to decide the address range it should claim is discussed in section 4.3.3).

In case any of B's siblings are using the address range that B chose, they send back collision announcements. For example, if C is already using the address range, 224.0.1.1/25 it sends a collision announcement to B. In general, if two domains make a claim for the same address range, one of them will "win" the claim[4]. When B hears a collision announcement, it gives up its current claim and makes a claim for a different address range, say 224.0.128.0/24, from its parent's space. Domain B listens for collision announcements for a waiting period long enough to span network partitions that might prevent B's claim from reaching all its siblings. Based on feedback from network service providers, we believe 48 hours to be a realistic period of time to wait. If no collisions occur, B communicates the acquired address range to the local MAASes and to other domains through BGP as group routes. We refer to the proposed mechanism used to obtain address ranges as the *claim-collide* mechanism. The motivation for the claim-collide mechanism is elucidated in section 4.3.4.

This decoupling of inter- and intra-domain address allocation mechanisms allows multicast addresses to be internally allocated very quickly, just as a local unicast address can be acquired quickly relative to acquiring a new unicast subnet prefix for the domain. It is expected that MASC will keep ahead of the demand for multicast addresses in its domain, but if there is a sudden increase in demand, addresses could be obtained from the parent's address space. If this is done, the root of the shared tree for these groups would simply be the parent's domain, which might be sub-optimal for the group if no senders or receivers were outside the child domain.

The parent domain, A, keeps track of how much of its current space has been allocated to itself and to its children. It claims more address space when the utilization exceeds a given threshold. Since A is a backbone provider domain, it does not have any parent MASC domain from which it can claim address ranges. However, domain A still uses the claim-collide mechanism to acquire address ranges by making claims from the global multicast address space,

[4] The winner may be based on domain IDs or IP addresses of the claimant MASC nodes or timestamps on the requests.

224.0.0.0/4. Its sibling domains correspond to the other top-level (backbone) domains that do not have a parent domain.

4.2 Distribution of MASC address allocation information through BGP

Once a MASC router[5] in B successfully obtains an unique address range, it is sent to the other border routers of the domain, which then inject the address range into BGP as a group route. When a border router X advertises a group route R to a border router Y, it means that the border router Y can use X as the next hop to forward data packets (as well as control messages) *towards the root domain* for the address range represented by R. For example, the border router B1 advertises the group route corresponding to the address prefix 224.0.128.0/24 to A3 in domain A. Since all BGP border routers of a domain peer with each other to exchange routes received by them from external peers, the border routers A1, A2, A3, and A4 learn of the group route received from B1. As only one route is received by A for this address prefix, it is chosen for use. If multiple routes are received, a preference function based on the attributes of the group route is used to pick one route among them, according to normal BGP behavior.

The chosen group route is stored by A3 in its G-RIB as (224.0.128.0/24, B1), indicating that B1 is the next hop from A3 to reach the root domain for range 224.0.128.0/24. The other border routers of A (i.e., A1, A2 and A4) store (224.0.128.0/24, A3) in their G-RIBs, as they use A3 as the next hop to reach the root domain for 224.0.128.0/24.

BGP group route aggregation then functions the same as for unicast routes. Since the address range allocated to A, 224.0.0.0/16, subsumes B's address range, 224.0.128.0/24, A's border routers need not propagate 224.0.128.0/24 to other domains. The border routers in these other domains that need to reach the root domain for 224.0.128.0/24 can forward their packets following the group route corresponding to 224.0.0.0/16 that A is already advertising. Such packets reach a border router in A that then uses its *more specific* G-RIB entry for 224.0.128.0/24 to direct packets to the root domain, B.

In the above fashion, the routing information to reach the root domain for the address ranges allocated by MASC is distributed to the border routers of other domains. Multicast policies are realized by the selective propagation of the group routes in BGP. For example, if border router X does not advertise group route R to neighbor Y then Y will not be aware that it can use X to reach the root domain for the address range represented by R. Thus, a provider domain could restrict the use of its resources by advertising only the group routes pertaining to its claimed address ranges and propagating only those group routes received from its customer domains (whose address ranges are not subsumed by the provider's address range) to other domains.

4.3 MASC design choices

The design choices we made for the MASC address allocation mechanism are intended to achieve efficient address space utilization, aggregation of the injected group routes, and robustness. In addition, we wanted to make use of existing Internet mechanisms as much as possible. In this section, we will examine details of the protocol within the context of these goals.

4.3.1 Address space utilization

To achieve good address utilization, the multicast address range claims made by a MASC domain are driven by its own and its children's usage patterns in a bottom-up fashion. In addition, each claimed address range is associated with a *lifetime*. The address range claimed by the domain becomes invalid once the lifetime expires unless the request is *renewed* before expiration. Once the lifetime expires, the address range is treated as unallocated by the parent domain and can be claimed by the children or by itself for its own use. A domain should claim address ranges with appropriate lifetimes according to its needs, but it may only claim a range for a lifetime less than or equal to the lifetime of the parent's range. If an appropriate lifetime range is not available, a domain should pick the address range with the longest lifetime that meets its needs. Consequently, it is possible that some applications within the domain may obtain a multicast address that has a shorter lifetime than needed for their sessions. Applications should be prepared to cope with this hopefully infrequent event by either explicitly renewing the address before it expires, or getting a new address once the old one expires.

We expect to have at least two pools of multicast addresses with different lifetimes - one associated with lifetimes on the order of months and the other with lifetimes on the order of days. The former would be useful for a domain to meet the "steady-state" demand for multicast addresses while the latter could take care of short-term increases in demand. As we gain more experience with multicast usage patterns, heuristics for lifetime selection should be refined.

The above restrictions posed by the address lifetimes allow address allocations of domains to organize themselves based on the usage patterns. This enables us to achieve efficient address space utilization as well as aggregation of group routes so that the G-RIB size scales well.

4.3.2 Aggregation of group routes

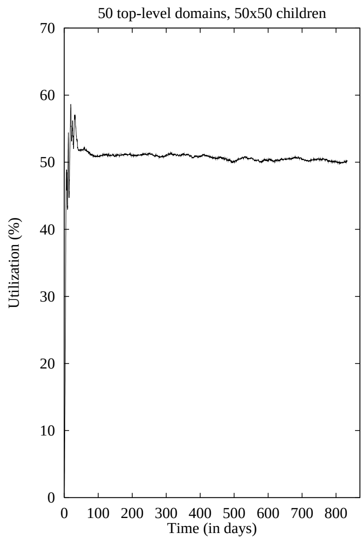
A MASC domain claims addresses by picking an address range out of its parent's address space. If no collisions occur, the claimed address prefixes are then injected by the domain as group routes into BGP. The group routes injected by the parent would hence cover the prefixes claimed by its children. For this reason, the border routers of the parent domain need not propagate their children's group routes explicitly to the rest of the world. This helps in reducing the number of routes in the G-RIB at the border routers of domains.

In addition, the address prefixes claimed by a domain should be aggregatable so that the number of group routes injected by the domain into BGP is minimal, thus further improving the scaling of the G-RIB sizes. *It is challenging to design the MASC claim algorithm to achieve both aggregation and efficient utilization of the address space given the dynamic nature of the demand patterns for multicast addresses.* We now present a MASC address claim algorithm and evaluate its performance in terms of address space utilization and the size of the G-RIB routing tables that result.

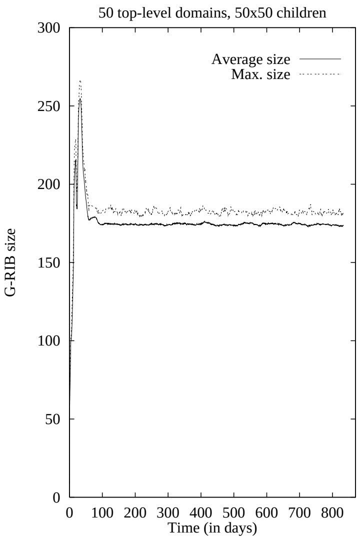
4.3.3 MASC claim algorithm

We have considered and simulated a number of claim algorithms for MASC. The algorithm described below is rela-

[5] Typically the MASC nodes are border routers of the domain, but this is not a requirement. If this is not the case, a BGP peering session has to be set up between the MASC node and one of the Border Routers of the domain to inject the address range.



(a) Address space utilization



(b) G-RIB size

Figure 2: Simulation results for the MASC claim algorithm

tively simple and performs well[6].

The number of significant bits of an address prefix is known as the *mask length* of the prefix. For example, the address range 224.0.1.0 to 224.0.1.255 has a mask length of 24, written 224.0.1/24. When a domain desires a new prefix, it looks at its local record of those prefixes that have already been claimed by its siblings. After removing these from consideration, it finds all the remaining prefixes of the shortest possible mask length, and randomly chooses one of them. The prefix it then claims is the first sub-prefix of the desired size within the chosen space.

[6] We expect further simulation, experimentation, and deployment experience to result in versions that perform even better.

For example, assume that 224.0.1/24 and 239/8 have been allocated out of 224/4. The largest sub-prefixes of 224/4 that do not overlap with either allocated prefix are 228/6 (1110 01...) and 232/6 (1110 10...); no non-overlapping masks of length 5 exist, and all other non-overlapping prefixes have longer length. If a domain requires 1024 addresses this requires a mask length of 22 (known as a "/22"). It randomly chooses either 228.0/22 or 232.0/22 as these are the first /22 prefixes inside each unallocated /6 range).

Choosing the first sub-prefix from the chosen prefix allows the greatest potential for future growth since the next domain would then choose from the other /6, rather than taking space from the same /6. This maximizes the chance that a domain will be able to expand its prefix in the future, rather than be forced to obtain a new, un-aggregatable one, and hence reduces the number of prefixes that need to be advertised globally.

Finally, collisions may result if several (say, n) domains make new claims simultaneously, and choose the same sub-prefix. In the worst case, the nth domain might have to make up to n claims before it obtains a prefix, although in the absence of network partitions, the difference in delay is negligible. Even in the presence of temporary partitions, choosing randomly among the /6 ranges provides a lower chance of a collision than if claims were deterministic. Note that since the obtained address ranges have expiry dates, the addresses allocated to a domain have to be given up once the lifetime expires unless the addresses are explicitly renewed. This helps us adapt continually to usage patterns so that better aggregation can be achieved.

The results below are with the use of *contiguous masks* for address prefixes. We are also investigating the use of *non-contiguous* masks as in Francis' Kampai[21] scheme. The use of non-contiguous masks in the Internet may face operational resistance (due to difficulty in understanding the scheme) but would provide even better address space utilization.

Simulation

When simulating claim algorithms, two properties are of primary concern: the G-RIB size (number of prefixes in use) and the address space utilization (how many allocated addresses are actually used). To examine these properties, we simulated a network with 50 top-level domains, each with 50 child domains. We also examined more heterogeneous topologies with similar results. Each child domain's allocation server requests blocks of 256 addresses with a lifetime of 30 days for local usage. The inter-request times for each child domain are chosen uniformly and randomly from between 1 and 95 hours, so that the total number of addresses allocated to a domain varies over time.

We define address space utilization to be the fraction of the total addresses obtained from 224/4 that were actually requested by the local address allocation servers. The prefixes claimed by the top-level domains are advertised globally (as they do not have a parent MASC domain to aggregate their claims with). The G-RIB size at a top-level domain is the sum of the number of globally advertised prefixes and the number of prefixes of its children domains. The G-RIB size at a child domain is the sum of the number of globally advertised prefixes and the number of prefixes claimed by its sibling domains.

To implement the above algorithm, we also need to specify thresholds for address space expansion. Our target occupancy for a domain's address space is 75% or greater. At the same time, we attempt to keep the number of prefixes per domain to no more than two. We term a domain's prefix to be *active* if addresses from the prefix's range will be assigned to new groups in the domain. Otherwise the prefix is termed *inactive*. When a domain receives a demand for more addresses that it cannot satisfy, we allow it to either double one of its active prefixes, or to claim an additional small prefix that is just sufficient to satisfy the demand. We double an active prefix if the total demand for addresses is such that after doubling this prefix, utilization of the domain's entire address space will be at least 75%. Typically this means that when we have more than one active prefix, we double the smallest one. If a domain has two or more active prefixes and none of them can be expanded, a single new prefix large enough to accommodate the current usage is claimed, if possible. If the domain succeeds in claiming this new prefix, the old prefixes are made inactive and will timeout when the currently allocated addresses timeout.

The results of this simulation are shown in figure 2(a). They indicate the utilization and G-RIB size over time from the start of the simulation. At the left hand side of both graphs a startup transient is observed caused by the very rapid increase in demand for addresses. After 30 days, the total demand for addresses has stabilized, and the G-RIB size then reduces rapidly as prefixes are recycled and aggregation can take place. Utilization rapidly converges to 50%, and the G-RIB size reaches a mean of 175 group routes, and does not exceed 180 group routes. It should be noted that the total number of child domains is 2500, and in the steady state there are 37500 requests for address blocks being satisfied, so this indicates extremely good aggregation. The 50% utilization is due in part to the choice of a 75% threshold at each level in this two level hierarchy. Results in this range should be acceptable for wide-scale deployment in the Internet.

4.3.4 Robustness

We have avoided centralized control in the MASC protocol. We believe this decentralization contributes to the overall robustness of the scheme, just as it does to the robustness of the Internet. The MASC hierarchy configuration rules are simple and are decided locally by bilateral agreements between domains, just as in BGP. For example, a domain can choose any of its provider domains to be its parent. Within this context, there were two options for the claim mechanism. One was the claim-collide mechanism we described in section 4.1, and the other was a *query-response* mechanism where a domain would acquire an unique address prefix by querying its parent domain.

We chose the claim-collide mechanism for reasons of policy, simplicity, and robustness. The top-level domains in the MASC hierarchy are typically backbone domains that are not customers of other domains. For policy reasons, the architecture should not require that any one of the top-level domains be specified as the root of the hierarchy. At the top-most level, there is no clear reason for a domain to be the parent of all the other top-level domains or for the top-level domains to accept someone as their parent. Using a query-response mechanism would however require a single root and introduce a third-party dependency at the top-level. Therefore, we chose to make the top-level domains claim from the entire multicast address space, 224/4. Lower in the hierarchy, parent-child relationships do exist, and follow the provider-customer relationships where there are payment agreements to enable the customer domain access the Internet. Given the use of a claim-collide mechanism among the top-level domains, and given the fluid nature of network topologies, it appears simpler and more robust to have a common mechanism at all levels.

A query-response mechanism would also require redundant servers within a domain for robustness, introducing additional problems of synchronization. Besides, this mechanism would still need to handle address collisions that might occur due to network partitions. Hence, our proposed claim-collide mechanism appears simpler and more robust than a query-response mechanism.

4.4 Start-up phase behavior

Start-up behavior is based on the same rules as those used in steady state. The entire multicast address space is initially partitioned among one or more Internet exchange points [7] (say, one per continent). MASC nodes at each exchange are bootstrapped to advertise its portion of the address space. All the MASC domains hear these advertisements and pick an address subrange, the size of which depends on their current address requirements. Backbone providers with no parent then pick the prefix of a nearby exchange (either one to which they connect, or one which they are configured to use) as their "parent's" prefix. Since this involves no parent-child MASC peerings at the top level, this approach minimizes third-party dependencies.

Top-level providers can then claim a small amount of space, which then grows as their children issue claims as described earlier. Alternatively, a top-level provider could initially wait for some period of time before claiming space, and just propagate its parent's group route to its own children. After listening to children's claims, it could then estimate the amount of address space needed to satisfy its children's requirements and then claim address ranges sufficiently large to satisfy their needs. However, to achieve aggregatability, the parent domains would then need to send back collision announcements to any children whose claims fall outside the parent's newly acquired space, forcing the children to pick up new address ranges.

We next describe how BGMP uses the G-RIB information to build inter-domain multicast distribution trees for a group.

5 Inter-domain distribution tree construction using BGMP

BGMP is run by domain border routers to construct an inter-domain shared tree for a group. The border routers also participate in protocols such as DVMRP and PIM running *within* each domain. Such intra-domain multicast routing protocols are also known as Multicast Interior Gateway Protocols (MIGPs). The portion of the border router running an MIGP is referred to as the *MIGP component* and the portion running BGMP as the *BGMP component*.

As in the example in section 4.2, we assume that BGP's route selection algorithm ensures that one border router is chosen as the *best exit router* for each group route. This router has an external peer as its next hop towards a group's root domain, while all other border routers have the best exit router as their BGP next hop. When a host in the domain joins a new group whose root domain is elsewhere in the Internet, the BGMP component of the best exit router is informed by the domain's MIGP. The rules to achieve this are

[7] The Internet exchange points are neutral locations where the larger providers such as backbones interconnect with each other (e.g., MAE-East in Washington, D.C. and LINX in London).

MIGP-specific. For example, in DVMRP, a *Domain Wide Report*[22] is sent to the MIGP components of the domain's border routers, including the exit border router, when a host joins a new group. The MIGP component of the best exit router passes a join request to the BGMP component, which sends a BGMP group join message to the next hop towards the root domain. These join messages set up multicast forwarding state in the intermediate border routers as they propagate across various domains towards the root domain, establishing the shared tree for the group.

When a group's root domain is external, multicast data packets reach the group's best exit router using MIGP specific rules. In DVMRP, for example, data packets are initially flooded throughout the domain and so reach all the border routers. The border routers that are not the group's exit router send DVMRP prune messages back towards the source, ensuring that only the exit border router continues to get data packets for that group. The best exit router forwards the packets along the inter-domain shared tree to reach group members in other domains. Since in IP Multicast, senders need not be members of the group, the best exit router might receive data packets destined to a group even though it has not previously received a group join request. In this case, the border router simply forwards the data packets towards the root domain, and when they reach a router that is on the group's shared tree, they are distributed to the members.

In the following sections, we discuss the key design choices made for BGMP, and present some protocol details. The design choices made in BGMP address the requirements for inter-domain multicast which are typically different from the intra-domain case. Protocols like CBT and PIM-SM build shared trees among their group members, but the mechanisms used to build these trees are better suited for the *intra*-domain case, and do not apply as well when used for inter-domain multicast. We explain in the following sections why some of the corresponding choices made for intra-domain multicast routing do not apply well to inter-domain multicast. We also present simulation results to compare the quality of the distribution trees built by the different multicast routing protocols.

5.1 Location of the root of the shared-tree

The choice of a group's shared-tree root has important policy and performance implications in the inter-domain case. In the intra-domain case, all routers that are willing to be the root are treated the same, and one is chosen, typically by hashing the group address over the set of routers. This is well suited to the intra-domain case where the emphasis is more on load sharing, and where the penalty of non-optimally rooted trees is not significant. In the inter-domain case, however, all potential roots cannot be treated as equivalent, since there are administrative issues concerning the ownership of the root, and there is also a much greater chance of poor locality.

We therefore adopt an approach where the root of the shared tree is selected based on administrative and performance constraints. The shared tree built by BGMP is rooted at the root domain for the group, which is the domain whose multicast address allocation includes the group address. Since a group initiator typically gets a group address from its domain's address range, the group initiator's domain is normally also the group's root domain.

The root's location affects performance since a root that is located poorly with respect to the senders and group members can lead to long paths between them. If the group's initiator sources a significant portion of the data, the root domain in BGMP is located reasonably optimally because the shortest-path tree from the receivers to the most significant sender now coincides with the shared tree for the group[8]. For example, the multicast session for a NASA space shuttle broadcast would have the shared tree rooted in NASA's domain. The root would be reasonably optimal for all receivers as they would receive packets from NASA along the shortest path from them to the sender.

A disincentive for a domain to claim an excessive amount of addresses in MASC is that it would then be the root domain for all the covered groups. There is also an incentive for domains to claim sufficiently large address ranges, which is that groups initiated locally can get multicast addresses from the domain's range, thereby making them locally rooted. We believe that these two competing factors will lead to some equilibrium for the size of the address ranges claimed by a domain; however, this is an area that requires further investigation.

We do not try to have BGMP form *optimally* rooted distribution trees as the multicast group memberships are fairly dynamic. Frequent reshaping of the distribution trees as members join and leave can cause data loss and high control traffic overhead.

5.2 Bidirectional trees

BGMP, like CBT[6, 7], builds bidirectional group-shared trees to minimize third-party dependencies and improve performance. For example, in figure 3(a), members in domains C and D can communicate with each other along the bidirectional tree without depending on the quality of their connectivity to the root domain, B. This is also more efficient because of the shorter paths taken. In contrast, PIM-SM builds unidirectional shared trees for a group, where data from senders has to travel up to the root and then down the shared tree to all the members. This approach would introduce third-party dependencies and potentially poor performance if applied at the inter-domain level.

In section 5.4, we present simulation results to compare the path lengths from senders to receivers in unidirectional trees built by PIM-SM to the bidirectional trees built by BGMP. We next illustrate BGMP bidirectional tree construction through an example.

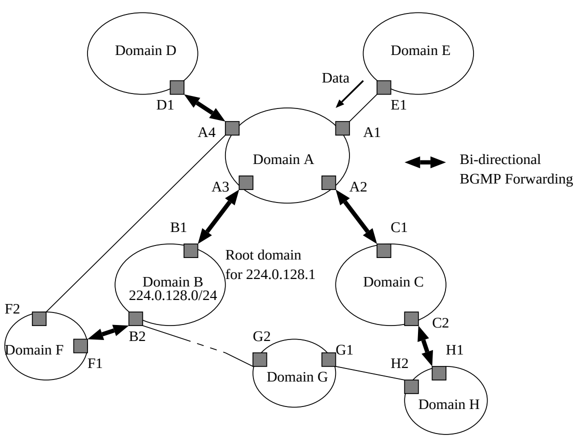
Establishing the bidirectional shared tree

Consider a multicast group created by a host in domain B in figure 3(a). Since the host acquires the address 224.0.128.1 from B's address range, B will be the group's root domain. When a host in domain C now joins this group, a join request is received from the MIGP by the BGMP component of the best exit router for 224.0.128.1, namely C1.

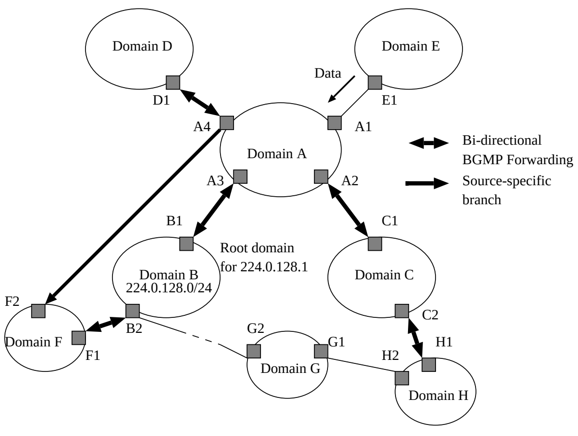
C1 looks up 224.0.128.1 in its G-RIB, finds (224.0.0.0/16, A2), and creates a multicast-group forwarding entry consisting of a *parent target* and a list of *child targets*. The parent target identifies the BGMP peer that is the next hop towards the group's root domain[9]. A child target identifies either a BGMP peer or an MIGP component from which

[8] Note that the shortest-path trees built by multicast routing protocols are actually *reverse* shortest-path trees; i.e., data from the senders flows along the shortest path from the receivers to the senders. If unicast routing is symmetric, this would be the shortest path from the senders to the receivers. Using reverse paths avoids the need to enumerate all the receivers.

[9] In case the BGMP peer is internal to the domain, the parent target is the MIGP component of the border router.



(a) Bidirectional shared tree



(b) Source-specific branch

Figure 3: Multicast distribution trees in BGMP

a join request was received for this group. The parent and child targets together are called the *target list*. In the case of C1, the parent target is A2 and the only child target is its MIGP component. This multicast forwarding entry, also known as a (*,G) entry, denotes that data packets sent to the group G, received at C1 from any source are to be forwarded to all the targets in the target list except the target from which the data packet came.

BGMP border routers have persistent TCP peering sessions with each other for the exchange of BGMP control messages (in this case, group joins and prunes). After creating the (*,G) entry for 224.0.128.1, C1 sends a group join message over the connection to the parent target, A2.

On receiving the group join message from C1, router A2 looks up 224.0.128.1 in its G-RIB and finds the entry (224.0.128.0/24, A3) indicating that A3 is the next hop to reach the root domain for 224.0.128.1. It then instantiates a (*,G) entry with the MIGP component to reach A3 as the parent target and C1 as the child target. A2 then transmits the join request to its MIGP component because A3 is an *internal* BGMP peer. The MIGP component of the border router performs the necessary actions to enable data packets to transit through the domain between A2 and A3. For example, if PIM-SM were the MIGP in the domain, it might make exit router A3 the Rendezvous-Point for the distribution tree within the domain and send a PIM join message towards A3, setting up forwarding state from A3 to A2.

On receiving the join request from its MIGP component, A3 creates a (*,G) entry with the MIGP component as the child target to enable the exchange of data packets with A2 through the MIGP. The parent target is B1, since B1 is the next hop to reach the root domain according to its G-RIB entry, (224.0.128.0/24, B1).

On receiving the join from A3, router B1, which is in the root domain for the group, creates a (*,G) entry with its MIGP component as the parent target (since it has no BGP next hop) and A3 as the child target. A join request is sent to its MIGP component, which joins as a member of the group 224.0.128.1 within the domain using the MIGP rules. For example, in DVMRP, a Graft message might be sent towards all pruned sources for the group in the domain.

To illustrate how data reaches the shared tree from domains not on the tree, suppose a host in domain E that has no members of the group sends data to the group 224.0.128.1. The data packets are transmitted through the MIGP to the best exit router, E1. Since E1 has no forwarding state for the group, it simply forwards the packets to the next hop towards the root domain (A1). Since A1 also has no forwarding state for the group, it transmits the packet through the MIGP of A to reach the next hop border router to the root domain, A3. For example, if DVMRP was the MIGP, the data packet would be broadcast throughout the domain and hence reach all the border routers of the domain. Since the border routers, A2, A3, and A4, are on the shared tree for the group, they each forward the data packets they receive to all the targets in their (*,G) entry except the target from which the packet was received (i.e., their MIGP component). The data packets thus reach group members in domains B, C, D, F and H along the shared tree.

When a BGMP router or an MIGP component no longer leads to any group members, it removes itself from the child target list of its parent target by sending a prune message or notification to its parent target. When the child target list becomes empty, the BGMP router removes the (*,G) entry and sends a prune message upstream towards the root domain. In this way, the multicast distribution tree is torn down as members leave the group.

5.3 Source-specific "branches"

BGMP can build *source-specific branches* in cases where the shortest path to a source from the domain does not coincide with the bidirectional tree from the domain (e.g. domain F in figure 3(b) for sources in domain D). This is useful for domains running MIGPs like DVMRP and PIM-DM which attempt to build source-rooted trees within the domain. In such domains, if the border router receiving packets on the shared tree is not on the shortest path to the source, it normally must send them encapsulated to the appropriate border router where they can be injected into the domain's MIGP. Otherwise the packets would be dropped by routers inside the domain due to failure of the RPF checks towards the source.

If a source-specific branch is built, data can be brought into the domain from the source via the appropriate border router so that the data encapsulation overhead can be avoided. This is done by allowing the decapsulating border router the option of sending a source-specific join towards the relevant source once data is flowing. The joins then propagate until they hit either a branch of the bidirectional tree or the source domain. A source-specific prune is sent back to the encapsulating border router, which can then propagate it up the shared tree to prevent unnecessary copies of the packet arriving.

Source-specific branches differ from source-specific shortest path trees built by some MIGPs in that the source-specific branch stops where it reaches either a BGMP router on the bidirectional tree or the source domain. In shortest-path trees, the source-specific state is set up all the way back to the source. BGMP does not support source-specific trees because of their incompatibility with bidirectional shared trees. There are some scenarios in which persistent duplication of data packets can occur when both source-specific trees and bidirectional shared trees intersect [10]. Fortunately, the difference in path lengths between source-specific distribution trees and bidirectional trees is less significant for the inter-domain case. The inter-domain topology is sparser than the intra-domain topologies, so that the path lengths and traffic concentration properties of the bidirectional shared tree are more acceptable (see Section 5.4). Source-specific branches are thus used primarily to stop data encapsulation.

We next illustrate the establishment of a source-specific branch through an example (see figure 3(b)).

Establishing a source-specific branch from F

Suppose there are members of a group 224.0.128.1 in domains B, C, D, F, and H, and B is the root domain for the group (see figure 3(b)). The bidirectional shared tree is set up as shown in the figure. Domain F has an inter-domain link to domain A via border router F2. Hence, the shortest path from domain F to hosts in D is through F2. F runs DVMRP as its MIGP, which implies that internal routers will only accept packets from a source which they receive from their neighbor towards that source. Since only F1 is on the bidirectional shared tree, data from a source S in domain D will be received by F1. F1 must then encapsulate the data packets to F2[11] in order to avoid internal RPF check failures. F2 then injects the data into the DVMRP domain so that group members in F receive the data packets.

If F2 decides to stop the above encapsulations, F2 may send a source-specific join towards the source S. It also instantiates a multicast forwarding entry called a (S,G) entry, with the parent target being the next hop towards S (A4), and the child target list consisting of its MIGP component. Data packets that arrive from A4 will thus be accepted and forwarded to other targets listed in the (S,G) entry. The source-specific join from F propagates towards the source (in similar fashion to a shared-tree join propagating towards the root domain) setting up (S,G) state in the intermediate border routers until it reaches a border router that is on the shared tree for the group. In our example, A4 is on the shared tree for the group. A4, on receiving the source-specific join, creates an (S,G) entry, copies the existing target list from the (*,G) entry, and adds F2 to the child target list. The source-specific join is not propagated further by A4.

Subsequent data packets sent by S and received by A4 are forwarded to all other targets in the (S,G) entry, including F2. Once it begins receiving data from A4, F2 sends a source-specific prune to F1, and starts dropping the encapsulated copies of S's data packets flowing via F1. Since F1 has no other child targets for (S,G), it propagates the prune up the shared tree to B2 so as to stop receiving packets from S along the shared tree.

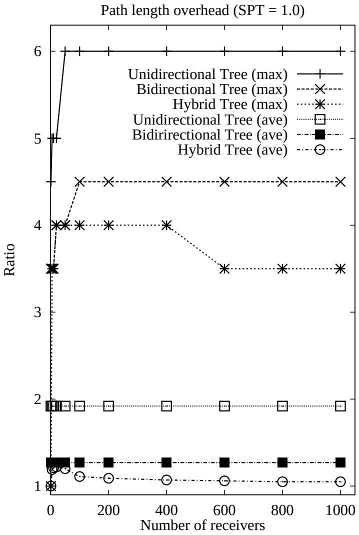


Figure 4: Comparison of path lengths on different multicast distribution trees

5.4 Simulation results

We performed simulations to compare the quality of trees built by BGMP to those built by other multicast routing protocols. In DVMRP, PIM-DM, and MOSPF, the data is delivered to the group members along source-specific shortest path trees. PIM-SM builds unidirectional shared trees where data from senders travels up to the Rendezvous Point (RP) and then down the distribution tree to the receivers. PIM-SM also allows receivers to switch to shortest-path trees to receive packets from sources. CBT builds a bidirectional tree rooted at the core router. BGMP builds bidirectional trees rooted at the root domain. In addition, BGMP allows

[10] Duplication or loss of data packets may occur depending on the rules used in forming the outgoing interface (oif) list for a source-specific (S,G) entry when a shared tree (*,G) entry also exists at the router for the group. When both (S,G) and (*,G) entries exist at a router, the router will forward packets from source S that arrive on the RPF interface to S only to the oifs listed in the (S,G) entry. Hence if the oif list from a (*,G) entry is not copied to a (S,G) entry then data packets from S will consistently not be delivered to some receivers. Let us consider the case where the oif list of the (*,G) entry is copied to the (S,G) entry. Suppose domain D (in figure 3(a)) establishes source-specific state to domain H so that receivers in D can receive packets along the shortest path (according to unicast routing) from a source, S in domain H. The shortest path between domains H and D happens to be via domains A, B and G (note that inter-domain unicast routing is policy based and hence the shortest path between domains according to unicast routing may not be the same as the path with the smallest number of inter-domain hops). This action will cause receivers in some domains like C to get duplicate packets from the source in H - one copy from source in H on the bidirectional tree and another copy while the packet travels along the source-specific tree via G and B to A and then down the bidirectional tree to C. The rules required for an appropriate border router of domain C to generate prunes to stop getting such duplicates seem too complex or have high state or forwarding overhead in the more general case. Hence BGMP allows only source-specific branches when the source-specific trees intersect with the bidirectional shared tree for a group. In BGMP, the source-specific join received at D1 from internal receivers is not propagated further by D1 since it is already on the shared tree for the group.

[11] F1 knows that the path through F2 is the shortest path to reach the source in D from its BGP routing tables, since F2 is the best exit router for the route to S.

source-specific branches to be built by the domains. We refer to the tree built by BGMP consisting of the bidirectional tree and the source-specific branches as a *hybrid* tree.

We compare the path lengths on the different types of inter-domain multicast distribution trees constructed by the above protocols (i.e., shortest path, unidirectional shared, bidirectional shared and hybrid trees). The path length between a source and a receiver of a group in the simulations is the number of inter-domain hops in the path between them.

Our topology[12] of 3326 nodes was derived from a dump[13] of the BGP routing tables at 15 routers located at the major backbones and network exchange points in the Internet. We studied the variation in path length from a source selected randomly to all the receivers of the group as the group size was increased from 1 to 1000 (see figure 5.4). The average path lengths from the source to the receivers were less than 20% longer (with a maximum difference of 4 times in the worst case) on a hybrid tree where source-specific branches were established from the receivers to the source, compared to that on the shortest-path tree. The bidirectional trees without the source-specific branches had path lengths that were less than 30% longer (maximum difference of 4.5 times) to that on the shortest-path trees. The unidirectional trees performed much worse compared to the bidirectional trees, with their average path lengths being about twice that of the shortest-path trees (maximum worst case path lengths up to 6 times that of the shortest-path trees).

6 Related work

HPIM[23] builds on PIM-SM by using a hierarchy of RPs for a group. A receiver would send joins to the lowest level RP, which in turn would join a RP at the next level, and so on. The number of levels in the hierarchy depends on the scope of the multicast group. Data from senders flows along the branches of this tree in a bidirectional manner to reach receivers. However, as HPIM uses hash functions to choose the next RP at each level, the trees can be very bad in the worst case, especially for global groups.

OCBT[24] is a proposed extension to CBT where a hierarchy of cores is used for a group on the lines of HPIM. Both HPIM and OCBT do not allow MIGP independence; i.e., it is not possible to run an arbitrary MIGP inside the domain, and run HPIM or OCBT only for inter-domain multicast. HPIM and OCBT also do not explicitly address policy constraints of domains.

HDVMRP[25] has been proposed as an inter-region (or domain) routing protocol that interconnects regions running any of the existing multicast protocols. HDVMRP floods data packets to the boundary routers of all regions and boundary routers that are not part of the group send prunes towards the source region to stop getting packets. Like DVMRP, HDVMRP still suffers from the overhead of broadcasting packets to parts of the network where there are no members. In addition, the memory requirements are high, as each boundary router must maintain state for each source sending to each group. HDVMRP also requires encapsulating data packets for them to transit a domain, which adds additional undesirable overhead.

In the area of multicast addressing, Sassan et al.[26] have proposed assigning group addresses based on the IP address of the host and the port number of the application on that host initiating the group. The resulting group address is 6 bytes long. In order to perform multicast routing, all the multicast routers would have to be changed to recognize these extended addresses.

Braudes and Zabele have outlined an hierarchical address allocation scheme in [27]. However they use the query-response mechanism with a single root for the hierarchy that we believe is not well suited for the Internet.

7 Open Issues and Conclusions

The primary open issues in our architecture that require further investigation involve incentives in MASC, the MASC claim algorithm, address allocation interface, scaling BGMP forwarding state and authentication mechanisms.

Incentives in MASC:

Restricting number of top-level domains: There are many open issues with respect to who becomes a top-level domain, and the incentives to be provided for appropriate self-selection. Poor behaviour can lead to poor scaling.

"Fair" address space utilization: We would like MASC to provide disincentives to domains to prevent them from claiming too large an address range, as this may starve other domains of addresses. A possible enforcement mechanism is for a parent domain to send back explicit collisions when a child claims too large a range. At the top-level, collisions could be sent by the sibling domains whenever a top-level domain claims too large a range. However, we lack an appropriate definition for "too large". We similarly need incentives for top-level domains to choose address ranges with reasonable lifetimes.

MASC claim algorithm: The algorithm used by domains to claim address ranges, while achieving good address space utilization and aggregation of group routes, needs to be studied more. The performance of the algorithm could be improved by the use of *non-contiguous* masks as in Tsuchiya's Kampai[21] scheme.

Address allocation interface: It may be desirable to allow a group initiator to pick a group address such that the resulting tree is rooted in another domain. This might be the case if, for example, the initiator knew that either the dominant sources would be located elsewhere, or that the initiator would be moving to another domain by the time the session begins. If there is sufficient demand for this capability, the interface by which a group initiator could obtain an address from another domain would need to be designed.

Scaling forwarding entries: We need mechanisms to enable the size of the multicast forwarding tables scale well to large numbers of groups. BGMP has provisions for this by allowing (*,G-prefix) and (S-prefix, G-prefix) state to be stored at the routers wherever the list of targets are the same. Its effectiveness will depend on the location of the group members and sources to those groups.

Authentication mechanisms: In general, authentication mechanisms are needed in our architecture. These

[12] We have performed simulations with other generated topologies as well. The simulation results for these topologies are available at http://netweb.usc.edu/bgmp/sim-results.

[13] The routing table dumps were obtained from the Oregon Exchange BGP Route Viewer, route-views.oregon.ix.net

mechanisms are especially important in the enforcement of disincentives in MASC.

We have presented an architecture for global Internet multicast addressing and routing. The MASC protocol dynamically allocates address ranges to domains with set lifetimes. These are distributed as group routes by BGP to all the domains, where they are used by BGMP to determine the root of the inter-domain multicast distribution tree for each group. By default, these shared trees are rooted at the group initiator's domain.

Our architecture allows MIGP independence; within each domain, any multicast routing protocol can be used. In addition, mechanisms are provided to realize policies to control the multicast traffic flowing through a domain. There is no centralized control in our architecture. The peerings between domains required by MASC and BGMP are decided locally by a domain through bilateral or payment agreements with other domains, and hence conform to the current Internet administrative model.

8 Acknowledgements

Many people have contributed to the basic ideas and have provided detailed comments on the design of this architecture. We particularly appreciate the valuable comments and suggestions from Steve Deering, George Eddy, Dino Farinacci, Bill Fenner, Ramesh Govindan, Jeremy Hall, Van Jacobson, David Meyer and the anonymous SIGCOMM reviewers.

References

[1] S. Deering. *Multicast Routing in a Datagram Internetwork.* PhD thesis, Stanford University, 1991.

[2] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, pages 85–111, May 1990.

[3] D. Waitzman, S. Deering, C. Partridge. Distance Vector Multicast Routing Protocol. *RFC-1075*, November 1988.

[4] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, L. Wei. Protocol Independent Multicast Version 2, Dense Mode Specification. *Internet Draft*, May 1997. Work in progress.

[5] J. Moy. Multicast Extensions to OSPF. *RFC-1584*, March 1994.

[6] A. Ballardie, P. Francis, J. Crowcroft. Core Based Trees. In *Proc. of the ACM SIGCOMM, San Francisco*, September 1993.

[7] A. Ballardie. Core Based Trees (CBT Version 2) Multicast Routing - Protocol Specification. *RFC-2189*, September 1997.

[8] S. Deering, D. Estrin, D, Farinacci, V. Jacobson, C. Liu, L. Wei. An architecture for wide-area multicast routing. In *Proc. of the ACM SIGCOMM, London, UK*, September 1994.

[9] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification. *RFC-2117*, June 1997.

[10] M. Handley. *On Scalable Internet Multimedia Conferencing Systems.* PhD thesis, University of London, 1997.

[11] D. Estrin, M. Handley, S. Kumar, D. Thaler. The Multicast Address Set Claim (MASC) Protocol. *Internet Draft*, November 1997. Work in progress.

[12] D. Thaler, D. Estrin, D. Meyer. Border Gateway Multicast Protocol (BGMP): Protocol Specification. *Internet Draft*, March 1998. Work in progress.

[13] M. Handley, D. Thaler, D. Estrin. The Internet Multicast Address Allocation Architecture. *Internet Draft*, December 1997. Work in progress.

[14] Christian Huitema. *Routing in the Internet.* Prentice Hall, 1995.

[15] Y. Rekhter and T. Li. A border gateway protocol 4 (bgp-4). *RFC-1771*, March 1995.

[16] T. Bates, R. Chandra, D. Katz, Y. Rekhter. Multiprotocol Extensions for BGP-4. *Internet Draft*, January 1998. Work in progress.

[17] V. Fuller, T. Li, J. Yu, K. Varadhan. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy. *RFC-1519*, September 1993.

[18] J. Postel. Internet Protocol. *RFC-791*, September 1981.

[19] S. Deering. Host extensions for IP Multicasting. *RFC-1112*, August 1989.

[20] D. Estrin, M. Handley, A. Helmy, P. Huang, D. Thaler. A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing. Technical Report USC CS TR97-644, University of Southern California, 1997.

[21] Paul Tsuchiya. Efficient and Flexible Hierarchical Address Assignment. *INET92*, pages 441–450, June 1992.

[22] W. Fenner. Domain Wide Multicast Group Membership Reports. *Internet Draft*, November 1997. Work in progress.

[23] M. Handley, J. Crowcroft, I. Wakeman. Hierarchical Protocol Independent Multicast. ftp://cs.ucl.ac.uk/darpa/hpim.ps.

[24] C. Shields and J. J. Garcia-Luna-Aceves. The Ordered Core Based Tree Protocol. In *Proc. of IEEE INFOCOM, Kobe, Japan*, April 1997.

[25] A. Thyagarajan and S. Deering. Hierarchical Distance-Vector Multicast Routing for the Mbone. In *Proc. of the ACM SIGCOMM, Cambridge, Massachusetts*, August 1995.

[26] S. Pejhan, A. Eleftheriadis, D. Anastassiou. Distributed Multicast Address Management in the Global Internet. *IEEE Journal on Selected Areas in Communications*, pages 1445–1456, October 1995.

[27] R. Braudes and S. Zabele. Requirements for Multicast Protocols. *RFC-1458*, May 1993.