Simple Scalable Internet Multicast

Tony Ballardie, Consultant, ballardie@dial.pipex.com
Radia Perlman, Sun Microsystems, Radia.Perlman@East.Sun.COM
Cheng-Yin Lee, Nortel, leecy@nt.com
Jon Crowcroft, UCL, jon@cs.ucl.ac.uk

4 February 1999

Abstract

Despite years of attempting to find a solution to Internet-wide IP multicast acceptable to both Internet Service Providers (ISPs) and end-users (applications), the solutions that have emerged are either relatively complex requiring significant administrative and protocol coordination, or have been found to be lacking in one or more crucial respects. This paper introduces Simple Multicast(SM), a new architecture and protocol borrowing attractive architectural and protocol features from existing ones, and introducing some new ones of its own. The architecture bases itself on the bi-directional shared tree architecture, which is acknowledged to have attractive scaling properties, and protocol similarities to CBT[1], PIM[2], and BGMP[3] are evident.

The thrust of Simple Multicast (SM) is twofold: firstly, SM identifies a group with 8 bytes - an ordered combination of the core IP address, and class D address, $< core, classD >$. The primary benefits of this are that multicast address space allocation becomes trivial; potentially, one class D space is administered by each distinct core. Secondly, an end-node (host) conveys the $< core, classD >$ parameters to a SM router, obviating the need for a "bootstrap" protocol[4] between routers, which potentially suffers from slow convergence.

This is a slight divergence from the traditional host service model[5] in which hosts register only a class D address to a local multicast router. Simple Multicast separates core/Rendezvous Point discovery from multicast routing; there exist many different means of discovering a group's parameters (e.g. core address, class D address, etc.) such as WWW, e-mail, telephone, a session directory (e.g. "sdr"[14]), and even DNS[6] has been proposed - any or all are valid. SM takes the view that core discovery introduces complexity that should remain orthogonal to the multicast protocol.

1 Introduction

This paper presents Simple Multicast (SM) which is a novel approach to providing many-to-many communication service.

The design goals of simple multicast represent a change of emphasis from the current Internet Standards for multicast. The motivation stemmed partly from anecdotal evidence that some Internet Service Providers were reluctant to deploy IP multicast in its current forms.

This reluctance seemed to stem from two problems: perceived complexity of IP multicast routing and addressing; lack of management tools for multicast routing and, in particular, lack of multicast traffic management.

The past several years has seen many IP multicast architectures/protocols emerge as candidates for Internet-wide multicasting, such as CBT[1] , PIM[2], BGMP/MASC[3], and others [6] [33] [34]. CBT and PIM have evolved from originally being inter-domain multicast proposed solutions to being confined to the intra-domain context due to aspects of their design which make them unsuited to the inter-domain environment. These aspects will be discussed in the section 5.6. The BGMP/MASC architecture remained as the only inter-domain candidate until Simple Multicast was proposed - the focus of this paper. BGMP/MASC is built upon the many lessons learnt over years designing and experimenting with a multitude of multicast protocols, but unfortunately BGMP/MASC is combined with a new and complex addressing architecture which requires a significant amount of administrative and protocol coordination; BGMP/MASC is a collection of protocols. BGMP/MASC too will be discussed in the next section.

A number of ancillary protocols have evolved around the inter/intra domain division characterised by PIM-DM and BGMP, to support address allocation[14], RP allocation/citebootrp, scoping[24] and scoped address discovery[25]. This plethora of infra-structural components also contribute to our unease with the overall architectural trend in IP multicast to date.

Simple Multicast is the product of combining the advantageous features of some existing protocols and incorporating them into the bi-directional shared tree architecture, and the reevaluation of some of the decisions made in the earlier years of multicast design. Consequently, it is suited to intra-domain as well as inter-domain multicasting. SM's design also allows it to flexibly support the data-driven dense-mode multicast distribution style. Shared tree - sparse mode - data distribution vs. dense-mode distribution is an application-specific choice which is discussed in section 5.8.

In addition, SM is designed to provide:

- separation of multicast group and core allocation from routing
- unification of intra and inter-domain routing
- possible exploitation of state reduction for single source trees
- partial NAT immunity
- integrated unicast and multicast policy route management
- expansion of usable address space so that group of group aggregation for profligate applications (DIS) is possible and
- hierarchical address use for filters
- an upgrade path to IPv6 multicast
- hooks for MPLS

2 Paper Organisation

In the next section, we give a brief overview of some of the history of IP multicast to date. Following that, we uncover the motivation for SM in more depth. Subsequently, we explore the design of SM in some depth. That is followed by a look at some possible variants on the SM theme. Open questions about SM are discussed next, followed by a discussion of related work, and some brief conclusions.

3 Background

There are many papers[7][9][3], that track the history and evolution of IP multicast as pioneered by Deering et al. in the late 1980s[5]. To avoid repetition, and for brevity, we describe only the more recent history here that is relevant to the emergence of Simple Multicast.

The Internet multicast backbone, the MBONE[10], was conceived in 1992 and, although it has evolved from a flat structure to a 2-level hierarchy of heterogeneous multicast "regions", the MBONE remains predominantly DVMRP[11] based. DVMRP is the oldest and most widely deployed multicast protocol in the Internet, the original version being one of Deering's contributions[5].

Issues related to the scalability of DVMRP in the wide-area context motivated the development of CBT. CBT originated the bi-directional shared tree architecture[1] in 1992. A tree rooted at a distinguished node, the "core", and spanning only those networks and links leading to interested receiver subnets is created by receivers' first-hop routers explicitly joining the tree. A first-hop router "grafts" itself onto a the shared tree by unicasting a join message hop-by-hop towards the core. If the join message hits a router that belongs to the corresponding tree the new branch is "grafted" on at that point, terminating the join. This router becomes a data distribution "fan-out" point in the tree. Only those routers along a tree's branches need keep state about the group - the group-identifier and a list of associated on-tree ports (interfaces). The data forwarding model is simple: if a packet arrives via an on-tree port (interface) forward a copy of the packet over all outgoing ports listed in the forwarding entry for the group-identifier.

The original principle behind PIM's design was to have receivers explicitly join towards source(s) as a means of supporting sparse groups - these early ideas evolved in parallel to (independently of) CBT. These ideas later developed into Protocol Independent Multicast (PIM), which comes in two modes: sparse (PIM-sm)[12] and dense (PIM-dm)[13]. PIM-sm is based on a uni-directional shared tree rooted at a Rendezvous Point (RP), synonymous with CBT's core. The motive behind PIM-sm's uni-directional approach to shared trees was to safeguard against looping data packets; data packets must arrive via a router's incoming interface for the relevant group which is always the upstream interface (closest to the RP). Hence, data originated on a member subnet (always downstream from the RP) must be first encapsulated and unicast to the RP where it distributed over downstream interface(s).

PIM-sm also allows each on-tree last-hop router to independently switch to shortest-path data distribution based on received data rate. If a switch is triggered the last-hop router explicitly joins towards the source. When data begins arriving via the new path, the old (shared tree) path is pruned from the distribution for the (source, group) pair. Thus, PIM-sm incurs state that tends from O(g) towards O(s, g) as more last-hop routers make the switch.

PIM-dm builds source-rooted trees and is based on a data-driven "broadcast and prune" distribution model. It is thus similar in many respects to DVMRP, and is therefore best suited to, and most scalable in, environments with a dense receiver population.

A key issue relating to CBT and PIM-sm is how they discover $< core/RP, group >$ mappings; PIM-sm defines a supplementary "bootstrap" protocol[4] which all PIM-sm routers must participate in; some subset of the PIM-sm routers advertise themselves at regular intervals to all PIM-sm routers as Candidate-RPs. Last-hop routers use a hash function to map a new group address - discovered by one of a number of possible means - to a Candidate-RP. This method of core discovery does not guarantee any correlation between an RP, its location, and the member set it is interconnecting. It also potentially suffers from slow convergence. CBT has left the use of a "bootstrap" protocol as optional - last-hop router manual (or network management) configuration being the other likely option.

No protocol other than BGMP/MASC and Simple Multicast (SM) has defined a scalable means of core discovery, and a multicast addressing architecture that deals with the issue of on-demand "clash-free" multicast address allocation/assignment. This, and the poor scaling properties of the "bootstrap" protocol, have resulted in the relegation of PIM and CBT from inter-domain multicast proposed solutions to intra-domain multicast solutions.

The confinement of CBT and PIM to intra-domain led to the proposal of the BGMP/MASC architecture and protocols. BGMP (Border Gateway Multicast Protocol) builds a bi-directional shared tree of domains, and supports shortest-path joins towards sources, similar to PIM-sm. The Multicast Address Set Claim (MASC) [14] protocol deals with hierarchical block allocation of Class D address space to routing domains (autonomous systems (ASs)). Essentially, MASC creates a prefix structure in the multicast address space in a way similar to unicast address space. Multicast address allocation has to be dynamic due to the limited multicast address space. MASC incorporates mechanisms for detecting collisions between different allocations and also deals with de-allocation. Once a block of multicast addresses is allocated, and no collision is detected for a period of time, the address block is passed to Multicast Address Allocation Server(s) (MAAS servers) for actual assignment to multicast groups. The domain "owning" an address block is said to be the "root domain" for all group addresses within the block.

A multicast address block's association with an AS, i.e. unicast prefix, is achieved by propagating the multicast prefix, and associated path attributes, throughout the routing system in BGP-4+[15]. Any router which subsequently sees a multicast data packet for which there exists no group (tree) state, or a join message for group g, knows in which direction to forward the packet.

4 Motivation for Simple Multicast

In the previous section we described the history and evolution of PIM and CBT. As part of their evolution we explained why these protocols are now confined to intra-domain multicasting. It remains for us to present a critique of BGMP/MASC as a means of introducing the motivation for Simple Multicast.

As explained in the previous section, BGMP/MASC is an architecture that relies on a set of protocols: BGMP - the tree building protocol, MASC - the protocol used by MASC servers (some small number per Autonomous System (AS)) for requesting/relinquishing a group address block from a top-level address block allocation domain (e.g. an Address Registry), and the protocol(s) that hosts will use to request group addresses on-demand from the local MAAS server.

BGMP builds a bi-directional shared tree of domains (ASs), and is to be deployed on AS Border Routers. As such it must be categorized explicitly as a multicast Exterior Gateway Protocol (EGP); it cannot run on multicast routers deployed inside a domain since interior routers do not keep the same granularity, if any, of the routing information - in particular the $< group - to - AS >$ mappings discussed in the previous section - propagated by BGP-4+. Therefore, there are protocol interworking mechanisms that must be implemented on BGMP routers each time a group spans more than one domain. Note that we are not criticising the intra- / inter-domain split which is advantageous and necessary for many reasons, including network manageability and information-hiding (scalability), but rather the design of BGMP which imposes multicast protocol interworking at every AS boundary. It would be an advantage to have a multicast protocol that can, where appropriate, be deployed within a domain as well as between domains to avoid the protocol interworking issues, which are often cumbersome and complex. Simple Multicast is one such design.

We also consider the other component of the BGMP/MASC architecture - MASC, to be the most complex part of the architecture; it requires the deployment of a hierarchy of servers, with different protocols required for communication between the different levels of the hierarchy. This will undoubtedly be very difficult to deploy in an Internet of today's proportions. If we assume a 3-level hierarchy - matching today's unicast routing hierarchy - the top-level servers are those deployed in some top-level domain (e.g. a Routing Registry such as RIPE), the mid-level servers are deployed within ASs (perhaps one per AS), and finally MAAS servers at the subnet level. It is not clear at this stage whether a different protocol will be used for communication between successive levels of the hierarchy; at least two protocols are proposed - one for AS MASC servers to request/relinquish an address block from a top-level domain, the other to be used by hosts to request/relinquish group assignments from a local MAAS server. MDHCP[16] has been proposed for the latter.

The MASC architecture also raises many questions, such as how is fair allocation assured, and by what means are allocations controlled - a small domain is unlikely to need as many multicast addresses as a large domain, so are maximum allocations to be imposed for some/all domains? Also, large ISPs are likely to act as multicast address registries to their smaller ISP customers, and with a limited class D address space conflicts of interest could arise. It is clearly very difficult to manage dynamic address allocations from a relatively limited address space and ensure fairness and availability at all times.

5 SM Design Features

This section discusses SM's principle design features. This section, and paper generally, omits detailing SM's protocol mechanisms for brevity, as they are very similar to those of CBT, which are well documented[1]. We do however, summarise protocol operation, and also fully explain aspects of the SM protocol which differ from CBT.

- Bi-directional Shared Trees: SM's Foundation

Simple Multicast's design is based on a bi-directional shared tree distribution model. The shared tree is built and maintained by protocol mechanisms very similar to CBT's; tree building and maintenance belongs to the protocol's control plane - SM control packets are directly encapsulated by IP. SM is awaiting an IP Protocol number from IANA[17].

CBT[7] demonstrated that, compared with source-rooted multicast trees, bi-directional shared trees have attractive scaling characteristics particularly for the many-sender case, incurring O(G) state in the network. In the same scenario they are also lower cost to the network; the number of nodes comprising a tree does not increase with the number of senders as is the case with source-rooted trees.

Bi-directional shared trees are more robust than their uni-directional counterparts. Data from any sender (member or not) must be encapsulated to the core in a uni-directional tree, and therefore the success of data distribution is reliant on the availability of the core. The success of data distribution is not wholly reliant on the availability of the core in bi-directional shared trees. More generally, a shared tree whose core is the sender's first-hop router is as robust as a data-driven source tree distribution - the most robust of all multicast distribution styles.

An unfounded misconception, now widely acknowledged, of bi-directional shared trees is that the core is a "bottleneck"; once the tree is formed, the core is just another node in the tree and has no special significance. Data from a sender on a member subnet is not sent via the core before being forwarded to other receivers, as is the case with uni-directional shared trees, but is distributed over the tree from the first-hop (on-tree) router.

Topologically proximate receivers' join messages for a topologically distant core are likely to converge relatively close to the receiver set producing a high "fan-out" in the tree near its edges. Trees with high edge "fan-out" have been shown in [18] to have good delivery delay properties in the many-many distribution scenario. Thus, even the simplest heuristic for electing a core, such as the group initiator electing his own host or local router, is likely to produce a perfectly adequate distribution tree for most applications. Nevertheless, there are cases where the receiver distribution may be sufficiently dense to warrant the use of dense-mode distribution - the Reverse Path Forwarding (RPF)[5] of multicast data by a router over all of its outgoing interfaces (except "leaf" subnetworks with no group memberships). Simple Multicast can support this type of data packet distribution (see below).

5.1 Trivial Multicast Address Allocation

With the growing competition for multicast addresses from the relatively limited class D address space, any Internet multicast architecture should have an accompanying addressing architecture; BGMP/MASC, Simple Multicast, and EXPRESS[19] all satisfy this requirement. EXPRESS is, in fact, similar in many respects to Simple Multicast - for example, both identify a group by means of the concatenation of a unicast IP address and a class D address (totalling 8 bytes in all). This facilitates very simple address allocation and management; potentially, each core (or source in EXPRESS) is able to assign a whole class D space, almost infinitely increasing the available multicast address space. Besides the obvious, there are several advantages to this. To illustrate one such example, a radio station provider such as the BBC could have a well-known $< core, classD >$ identifier, where the class D address represents the BBC's "home", or "base", address - perhaps one of BBC's News stations. Receivers could intuitively deduce other stations, such as Radio 1, 2, 3, 4, or 5, by simply joining the group identified by the same core, but with a class D address that is a positive offset from the "base" class D address. The EXPRESS model is further discussed in section X.

5.2 Data Packets

SM data packets must carry a SM header[1], directly encapsulated by IP. This header is used to carry the group identifier (core, class D) as well as other semantic information relating to the group. If we presume the use of masks to allow the specification of core and class D prefixes, we expect the SM header not to consume more than 20 bytes (fixed length); in some SM variations (see below) this could be significantly less than 20 bytes. All SM data packets are IP destination addressed to the "all-SM-nodes" group address (224.0.0.x), currently requested from IANA[17].

Non-member senders must unicast data packets to the group's core router for dissemination over the tree. A point of concern is: how does a non-member sender know if the core is alive/reachable to avoid a "black hole" scenario? One way would be for a sender (or first-hop router) to join the tree by creating a uni-directional branch. This would require a new SM join option to support the feature.

5.3 Incrementally Deployable

SM control messages are directly encapsulated by IP. Join messages are unicast to the core router, but joins (and all SM control messages) contain the Router Alert IP option[20], causing all intervening routers on the

[1] It would be possible to carry this information as a new IP option, but we consider the SM header approach to be "cleaner" and more flexible, and probably makes for more efficient (hardware) implementations.

path to the tree/core to inspect the contents of the packet. The last SM-aware router visited inserts its IP address (of the forwarding interface) into a field of the SM control packet header. The next SM-aware router that processes the packet inspects this field to see if the previous SM hop is adjacent - if not, the router marks the interface as a tunnel interface for the group and remembers the previous SM hop. The previous SM hop can derive similar tunnel end-point information from the corresponding join-ack. Data packets flowing between non-adjacent SM hops are "tunnelled" - IP-in-IP encapsulation[21] is not necessary - rather, the data packet's IP destination address is simply re-written to the IP address of the remote "tunnel" end point. This "auto-tunnelling" mechanism allows for transparent incremental deployment of SM.

5.4 Host Behaviour

A SM end-node (host) conveys the $< core, classD >$ parameters to a SM router, obviating the need for a "bootstrap" protocol[4] between routers, which does not scale. This is a slight divergence from the traditional host service model[5] in which hosts register only a class D address to a local multicast router. If the end-node takes responsibility for explicitly joining the tree - rather than modifying IGMP[22] to carry core as well as class D address - advantages could be gained from seamless incremental deployment viewpoint; this strategy increases the likelihood that a host can participate in a SM group irrespective of the presence of a local SM-aware router.

When a join hits the tree it is terminated and acknowledged - the acknowledgement follows, in the reverse direction, the state created by the join, all the way back to where the join originated, "confirming" that state in the process.

In the worst case (many hosts join the same group but there is no local SM router), many SM "tunnels" emanate from the LAN - one per joined host. This will cause multiple (unicast) copies of a group's packets to be forwarded over the LAN. In the best case (the presence of a local SM router), all but the first join is terminated at the LAN router. When the join-ack for the first join is received by the LAN router, it and the remaining joins can be ack'd back to the host(s). On receiving multicast packets, the LAN SM router will forward one native copy for the group members onto the LAN. Further discussion of SM host behaviour is provided in section X.

5.5 Tree Maintenance: Heartbeats & Keepalives

SM's tree maintenance is similar to CBT's - parents monitor their children, and children monitor their parents; a SM child periodically sends "keepalive" messages to its parent on a group-specific basis. The parent router expects these from each of its children at certain intervals; the failure to hear a keepalive from a child after several keepalive intervals results in the parent removing the relevant port (interface)from the distribution tree, provided no other children are reachable over it. If the parent router has no more children, it can remove itself from the tree by issuing a SM "quit" message to its parent.

"Heartbeat" messages travel from the parent to the child - they serve to indicate the core's "liveness" to each of a parent's child routers. If the core fails, its adjacent child router(s) indicate "core unreachable" in the heartbeats they generate to their children. Thus, the failure of the core does not prevent the flow of heartbeats between parent and child routers. The purpose of serving notice of core unreachability to a child is so that child can take appropriate action, such as attempt to rejoin the same tree via a different route, or prune itself off the tree and graft itself onto a "backup" tree. Multiple trees - pre-formed or dynamically created - per group may be relevant to applications requiring a very high degree of reliability/availability.

5.6 Applicable to Inter-and Intra-Domain Multicasting

SM can be deployed within and between domains, obviating the need to implement cumbersome and complex protocol interworking mechanisms on domain boundaries. It is obviously simpler to have one multicasting mechanism than many.

5.7 Support for Access Control

Internet Service Provider's (ISPs) will be catalytic in the global success of IP multicast; for them, IP multicast must be economically and technically viable. Amongst other things, an IP multicast architecture must be robust, incrementally deployable, scalable wrt to network state, and incorporate an easily-manageable multicast addressing architecture. More generally, the ability to control access to a group will contribute to the success of multicast; if an ISP's charging model charges the content provider (the source), then it would not be unreasonable for the source to expect there are no interruptions during the send period. A shared tree's core is an obvious point of control; an access control list (allowed/include or disallowed/exclude senders list) is configured on the group's core router and then propagated to other on-tree routers in "heartbeat" messages. In a strictly controlled tree all senders might have to unicast all data to the core for authenticating, before the core forwards the packets. The Simple Multicast architecture supports these features.

5.8 Support for Dense-Mode Distribution

A sender (belonging to a member subnet) can indicate to the first-hop SM router that it wishes a data packet to be forwarded using RPF style distribution - it does so by inserting a core address of 0xFF:FF:FF:FF in the SM header of the data packet.

5.9 Join Forwarding "Aggregation"

The forwarding of SM join messages, based on the core IP address portion of a group-id, follows unicast routing information already present in routers. For BGMP joins to do the same, group prefixes have to be associated with unicast prefixes which requires BGMP-specific supplementary routing information propagation.

5.10 Multicast Scoping

At present, IP multicast scoping[23] is achieved by configuring multicast border routers (M-BRs) on a scope boundary with a boundary scope address range - so-called Administratively Scoped address range. Multicast traffic flows which are to be confined within a range must use a class D address which is within the range. M-BRs are an impermeable boundary to any multicast packet with a class D destination address that falls within any of its configured Administratively Scoped address ranges.

It is perfectly feasible for SM to use exactly the same mechanism for achieving multicast scoping. However, multicast scoping as it is currently defined requires a significant amount of configuration, as well as co-ordination of the address space for defining scope boundary ranges. Any mis-configurations can lead to multicast packets "leaking" across boundaries they shouldn't. Multicast scope boundary configurations must conform to certain rules, such as the rule that boundaries must be completely contained within one another (the term "nesting", or "convex", are often used). The MZAP protocol[24] is implemented on M-BRs to detect inconsistent administratively scoped boundary configurations. As such it is essentially a network management tool, it does not correct mis-configurations.

SM has the unique ability to take advantage of the unicast routing system boundaries (e.g. subnet, area, AS, AS-Confederation etc.) and use these as "natural" boundaries for multicast traffic, obviating the need for the configuration of explicit multicast boundaries. Furthermore, one group identifier ($< core, classD >$) can be used with multiple scopes. It works as follows: assume a $< core, classD >$ group identifier is to be used for scopes A and B, with A nested inside B. A and B are natural unicast routing boundaries, e.g. area, and AS. A unicast routing system boundary is implicitly identified by a router aggregating routing information before propagating it over outgoing interfaces; this is achieved by shortening a prefix mask. For example, routing information inside boundary A has an associated mask of 24 bits. The boundary router between A and B reduces this is to 16 bits before propagating inside B.

Now, if a SM data packet carried a "scope mask(len)" in the SM header, the data packet would not pass beyond any unicast routing system boundary that itself propagates a shorter mask in unicast route updates it sends. The general rule is: a SM data packet carrying a "scope mask(len)" is only forwarded over those interfaces that aggregate unicast routing information using a mask which is equal length or longer than that specified in the SM data packet header.

Figure 1 illustrates a router with 4 interfaces, a), b), c) and d), each with the respective prefix length. If a SM data packet arrives on interface b) carrying a “scope-mask(len)a” of 12, it is forwarded only over interfaces c) and d).

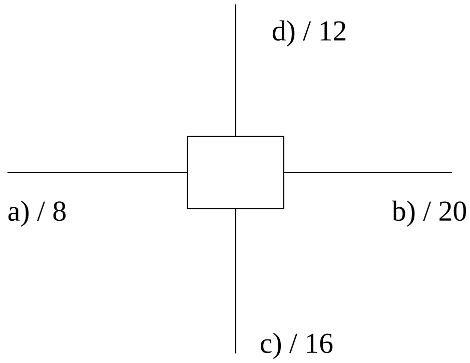

Figure 1: Operation of SM Scoping

5.11 Loop Detection

SM tree building is almost identical to that of CBT, and similar to BGMP and PIM-sm. Like these other protocols, SM takes care to avoid loops during tree creation. Nevertheless, SM has the opportunity to make use of flags in the SM header included with data packets, and one such flag could be used by data packets to detect loops. One bit in the SM data packet header could indicate to a receiving router that this packet's IP TTL should not reach zero. Note that not all zero-TTL packets are necessarily looped; expanding-ring searches[5] and some multicast diagnostic programs (e.g. mtrace[26]) purposefully use low IP TTLs. The SM ”loop bit” could be set by the relevant applications.

6 SM Variations

Throughout the paper we have presented a uniform view of Simple Multicast, but there are variations on the theme which we now discuss. These variations are proposed solely with the intention of making SM backwards compatible with existing multicast router behaviour - perhaps we should call this ”conventional” multicast router behaviour. Multicast router designs are currently optimised to expect a multicast data packet to have a class D address as its IP destination address and be able to uniquely identify the relevant distribution tree using this value, usually combined with the IP source address (RPF). As we have seen, SM does not subscribe to this conventional model.

There are two SM variations which allow multicast routers to uniquely identify a SM group's data packets using the class D address alone. These variations relate to how class D addresses are used - in the uniform view we proposed the use of one class D space per core:

- one class D space shared between all core routers
- one class D space per data link

In this first variation, the positive implication of this is that data packets need not carry a SM header. Data packets would be sent as native multicast data packets across multicast-capable links, or tunnelled IP-in-IP otherwise. In all SM variations control packets need the SM header to identify control packet type (e.g. join-request,join-ack, quit, keepalive, etc.).

A significant disadvantage of this variation is that the problem of class D address space management resurfaces - primarily, how can addresses be guaranteed to be unique, and if multicast becomes ubiquitous, the relatively limited address space could stifle its success. Nevertheless, we still consider this method an easier and more manageable option than MASC; multicast address only need to be unique, they do not need to be pre-allocated in blocks, nor do they have to be propagated by a routing protocol. Uniqueness is achievable - for example, if all dynamic multicast address were assigned by the same mechanism (like ”sdr”[14]).

The second variation again removes the class D space management problems - in this variation one class D space is shared only across a data-link. When a host sends a join to the first-hop SM-aware router, the class D address contained within the join is relevant only to that link. The first-hop router maps the incoming class D to a unique outgoing class D before forwarding the join with the new class D on the outgoing data link. Subsequent routers on the path to the tree (core) do the same.

This variation too, eliminates the need for data packets to carry a SM header - data packets are multicast natively across each data link. The class D portion of the group identifier is guaranteed to be unique across each data link, and therefore the right tree can be identified using it alone.

A perceived disadvantage might be the IP destination address re-writing that happens on a link-by-link basis on the packet's journey - in this respect, this variant is somewhat ATM-like. However, like ATM, an incoming address and port (interface) can be mapped to an outgoing address and port very quickly because it involves an exact-match lookup.

It may not be possible to accommodate tunnels in this SM variant and still identify a tree using the class D address alone; a router that is serving as multiple tunnel end-points could be sent the same class D address (for different trees) by two different virtual SM neighbours. These are distinguishable by incoming port, but, if both joins are propagated upstream over the same (virtual) port, subsequent data packets arriving via the upstream port will not be uniquely mappable (using class D alone) to the branch(es) of one particular tree. This could be overcome by having the router "reject" a join and request a new join for a class D address already seen, but this ultimately impacts join latency.

6.1 Network Address Translation

A multicast session can be scoped within a private network if the core/source address belongs to the private address space and is not translated to any global address. It is the address separation and filtering feature of NAT that is of interest to scoping, not it's address translation function , this does not require any changes to NAT router. Note that the NAT [38] router require changes (ALG) only if we want the multicast session to work across NAT, i.e. some of the IP addresses in SM packets must be translated in that case.

As we have discussed before, you can scope within the same address space at higher granularity (i.e. areas , subnets) by including the core and/or source masks in the data packets. This requires the Border Router to be SM BR. However without NAT, the SM BR alone only won't be able to filter private addresses, it doesn't have the notion of private address or address separation.

7 SM Questions

This section discusses the primary open, or controversial, questions about the SM architecture.

7.1 Host Implications of SM

The role of hosts in the SM architecture requires the incorporation of host kernel software. The extent of host kernel changes depends on whether a host takes responsibility for sending SM joins, and in this respect the host is acting similarly to a SM router, or whether IGMP is extended to carry a core address as well as a class D address. Extending IGMP is perfectly feasible - it has been done before in upgrading from IGMPv1 to IGMPv2, and changes will be required for IGMPv3[27] if it gains wider acceptance. We do not, therefore, consider host changes as barriers to SM deployment.

Both cases require changes to the host Application Programming Interface (API). Taking the BSD Sockets API[28] as an example, joining a group is achieved using a system call; the data structure passed with the system call as an argument only supports the specification of a class D address and interface (IP) address. For SM this data structure needs modifying to include a core address element, which can be concatenated with the class D address to form SM's 8 byte group identifier. The kernel SM software, or IGMP software, can then make use of this information to generate a SM join message, or IGMP Report, respectively. Note that, since all hosts implement IGMP, if the host issues a SM join message it must suppress the issuing of any non-SM-modified IGMP Reports[2].

For the case where SM is implemented on the host, an obvious question is: if a local SM-aware router is present, how does this router monitor group membership presence/absence in a timely fashion so as to not forward group traffic onto the LAN unnecessarily? As we noted above, SM hosts act like SM routers attached to a multi-access link. As child "routers" on a multicast tree, they send SM Keepalives to their parent. The adjacent upstream router (locally attached in this scenario) can thus establish the presence or absence

[2] The sending of an IGMP Report may cause a local non-SM multicast router to install multicast state, which would be superfluous and could cause an error condition - multiple trees for the group.

of group members by the presence or absence of SM Keepalives. However, the Keepalive interval is unlikely to provide the same fine granularity of leave latency that can be achieved with IGMPv2[22].

7.2 Layer 2 Filtering

The SM architecture does not recommend any modifications to aspects of layer 2 sending or reception of multicast frames. The low-order 4 bytes of the SM group identifier - the class D address - is statically mapped to a 6 octet multicast frame destination address[29].

SM potentially imposes higher layer filtering of multicast packets due to the potential for each of a subnet's hosts to join different SM groups, with their group-ids differing only in the core address portion of the group-id. In this worst-case scenario the transmission of packets to one group will be received by hosts belonging to all other SM groups on the subnet; a group's packets only become distinguishable at the hosts' network layers. In a more realistic case we might reasonably expect only a small percentage of a subnet's hosts to receive packets unnecessarily.

One possible way of guaranteeing layer-2 multicast destination address uniqueness would be for an IGMP-joining host to indicate in its membership report that the layer-2 address is currently unknown, and then for the local SM router to assign a currently unused layer-2 multicast address. Obviously, this requires changes to IGMP (discussed above); for instance, a membership report would have to be layer-2 addressed to the well-known "all-routers" group address.

Another possibility would be to statically map the core address to a multicast layer 2 address if we assume groups associated with a core are likely to be related (such as the BBC Radio 1, 2, 3... example described above). This would still potentially incur higher layer filtering of undesired groups, but only those hosts subscribed to group(s) associated with a particular core would be affected.

As a final note on this topic, the problem of mapping a larger-than-usual network identifier to a layer 2 address is not unique to SM - the problem manifests itself in IPv6[30] and EXPRESS.

7.3 Performance issues

Although data packets carry a SM header in the "default" variation of SM, from a purely technical viewpoint there is no reason why SM data packet forwarding performance should be inferior to that of other schemes. Today's production routers' multicast forwarding engines are primarily hardware-based. This hardware has been designed to forward multicast packets using the RPF technique[5] - the most widely deployed protocols, DVMRP and PIM, are RPF-based. SM forwarding involves a multicast forwarding entry look-up based on the contents of the SM header, then forwarding a packet copy over each outgoing interface listed in the entry. This forwarding model is probably the simplest of all forwarding models - there is no need to perform a source address check. Deploying SM on routers designed to forward using a different forwarding paradigm will inevitably result in SM exhibiting poorer performance - all data packets would go to the "slow path", which is software based.

7.4 Interoperability

For domains not deploying SM, the issue of border router interoperability arises. This issue is not unique to SM - all protocols have defined interoperability specifications for interworking with most other protocols[30]. The interoperability requirements for SM will be similar to those of BGMP[3] and CBT[32]. One of the most challenging aspects of interoperability is that between shared and RPF-based source trees; a shared tree of heterogeneous domains requires each domain to attach via a single border router. The implication for routers inside an RPF-based domain (e.g. DVMRP domain) is that externally sourced data must arrive via the path the router would use to reach the data source. However, the domain ingress point - the attachment point to the shared tree - is not the RPF path for all external sources. A commonly recommended solution to this problem is to have the domain ingress router tunnel data to the correct RPF ingress router so the data appears to have arrived via an acceptable path.

Another aspect of interoperability unique to SM and EXPRESS is: how to map the 8-byte group identifier to a "conventional" multicast address, and vice-versa. At the time of writing this is a topic of ongoing research.

7.5 Domain-Level Control ("3rd Party" Independence)

For the case in which SM is used both within and between domains, joins from different parts of the domain might only converge (merge) outside the domain. It is not desirable for a domain to depend on another, "3rd party", domain for the distribution of internally sourced traffic to other internal receivers. It is therefore necessary to ensure that joins from different internal receivers merge at a common point inside the domain.

BGP-4 operates on border routers (BRs) of transit domains, and ensures that all BRs know which of them acts as egress for a particular unicast prefix. Some transit domains (the elected egress router) inject external route information internally, and therefore, internal routers know in which direction to forward packets destined to a particular unicast prefix. In other cases, and in stub domains, external route information is not injected inside the domain. Nevertheless, the BRs of these domains know for which unicast prefix(es) each of them is acting as egress. Thus, domain BR routing knowledge ensures that joins originated inside a domain converge at a common point inside the domain.

This principle can be applied recursively across a multiple levels of routing hierarchy.

7.6 Transit policy

Shared trees are not as inherently flexible at supporting transit routing policies as multicast source trees or unicast sink trees. For example, there may be instances when policy might prohibit packets from A to D transmitting domain B for a group (C,M). With a core (C) in domain B, or just due to the shared tree that was formed, packets from senders in A to receivers in D might traverse domain B.

One solution to this problem takes inspiration from the PIM-SM concept of using the shared tree to find out about per-source trees. The way it works is that the sender in domain A, say S, sends a message to the core C telling it that it would like to create a "spin-off" group, (S,M'). Then the core C, in the heartbeat messages for group (C,M) advertises the spin-off tree that members of (C,M) should also join for receiving packets from S. Although this does allow creation of multiple trees to support a single group, this is less expensive than the PIM-sm scheme in which receivers source-join to senders based on source data rate; SM creates additional trees only when it has to, such as when transit policy prevents a source reaching the group's receivers.

7.7 Group State Aggregation

The ability to aggregate routing state information is a desirable property of any widely deployed routing system. Aggregation of multicast forwarding information is desirable to reduce memory consumption and improve lookup (i.e. forwarding) performance. Achieving multicast forwarding state aggregation however, is difficult; most multicast groups are transient (created on-demand and short-lived) and group membership of these group types tends to be highly dynamic over the lifetime of the group, as shown in YAM[18]. There is no evidence to suggest, for a set of contiguous groups flowing through a router - relative to a source (or core) - the receiver set for each group will be related in the slightest. This applies equally to any multicast architecture, including BGMP/MASC, despite BGMP/MASC associating blocks of group addresses with domains.

8 Related Work

The proposal here involves a change to router and host systems. There have been several other multicast researchers who have proposed changes to both end systems and infrastructure, usually motivated by special high level protocol requirements. For example, Pragmatic General Multicast[37] is a modification to host and router dynamic subtree filtering functions to enable more scalable reliability mechanisms.

The Addressable Internet Multicast model[36] envisages changes to the infrastructure to support a wide range of enhanced multicast services including more scaleable anycast, sub-cast, and layered multicast. EXPRESS[19] multicast is more closely aligned to SM and we discuss this in more detail in the next subsection.

8.1 Comparison with EXPRESS

Simple Multicast exhibits many similarities to the recently proposed EXPRESS model [19]; EXPRESS uses 8-byte group identifiers: the combination of source IP address and an EXPRESS identifier - a class D address. Similar to SM, this makes address management trivial by greatly expanding the available address space, with the ability to easily ensure uniqueness per source (core in SM). The host behaviour is also similar, with EXPRESS hosts issuing subscribe requests to join a tree that is built using protocol operations similar to CBT's. However, the tree built is a source-rooted tree - joins are sent towards a source, and there are controls implemented that only allow the source to inject data onto the tree. Like SM, EXPRESS considers access control an important aspect of a successful multicast business model.

Where SM diverges from EXPRESS is in the nature of the multicast tree - SM builds a bi-directional shared tree which can be shared by multiple senders whereas EXPRESS imposes one tree per sender. In EXPRESS multiple senders define a multicast session, but the a session is an application layer notion. We believe that a single tree supporting multiple senders is best suited for the purpose of Internet multicasting because a large number of applications are multi-sender (e.g. conference applications, multi-player games, DIS[33] applications). In multi-sender scenarios it is less overhead, and therefore more efficient, to build a single shared tree than multiple trees.

9 Conclusion

This paper introduced Simple Multicast (SM) - a new multicast architecture and protocol suited to intra-or/and inter-domain multicasting. The principles of SM are that its architecture centres on a bi-directional shared tree model, an 8-byte group identifier comprising a core IP address and a class D address, and the role of the host in informing other SM routers of the core's identity. We motivated SM, in part, by explaining the unsuitability of other proposals, including BGMP/MASC.

10 Acknowledgements

Besides the co-authors listed, there are many who have contributed ideas to Simple Multicast. The fact that SM is partly a combination of other architectures and protocols [x,x,x,x,x...] means that the authors of those deserve some of the credit. Those who have contributed more directly to SM include ....

References

[1] "Core Based Trees Multicast Routing (CBTv2)"; A. Ballardie, Sept. 1997. RFC 2189;

[2] Protocol Independent Multicast (PIM) Web Site; `http://netweb.usc.edu/pim`

[3] "The MASC/BGMP Architecture for Inter-domain Multicast Routing", Satish Kumar (ISI/USC), Pavlin Radoslavov (ISI/USC), David Thaler (Merit Network, Inc) Cengiz Alaettinoglu, (ISI/USC), Deborah Estrin (ISI/USC), and Mark Handley (ISI/USC) in Proc ACM SIGCOMM 1998, Vancouver, volume 28, number 4, October 1998. ISSN # 0146-4833.

[4] "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing", D. Estrin, M. Handley, A. Helmy, P. Huang, D. Thaler. To appear in Infocom 99 , NY, April, 1999.

[5] Steve Deering, PhD thesis, 1988, Stanford University.

[6] "Static Multicast", Work in Progress, Masataka Ohta and Jon Crowcroft.

[7] "Core Based Trees", A. Ballardie, J. Crowcroft, P. Francis, in proc ACM Sigcomm'93, pp85-95

[8] "A New Approach to Multicast Routing in a Datagram Internetwork", A. Ballardie, PhD thesis, `ftp://cs.ucl.ac.uk/darpa/ballardie-thesis.ps.Z`

[9] "An Architecture for Wide Area Multicast Routing", Deering et al, In proceedings ACM Sigcomm'94, pp126-135

[10] "MBone Provides Audio and Video Across the Internet" Michael R. Macedoniaand Donald P. Brutzman, IEEE COMPUTER, pp. 30-36, April 1994.

[11] "Distance Vector Multicast Routing Protocol", version 3, DVMRPv3, T. Pusateri, Juniper Networks, Work in Progress 1997.

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

[13] "Protocol independent multicast-dense mode (pim-dm) : Protocol specification", Estrin, D., D.Farinacci, V.Jacobson, C.Liu, L.Wei, P.Sharma, and A.Helmy. Work in Progress.

[14] "Session Directories and Scalable Internet Multicast Address Allocation", M. Handley Proc.ACM Sigcomm 98, September 1998, Vancouver, Canada.

[15] "A Border Gateway Protocol 4 (BGP-4) ", Y. Rekhter, T. Li. March 1995. RFC 1771.

[16] "Multicast address allocation based on the Dynamic Host Configuration Protocol", Baiju V. Patel, Munil Shah, Stephen R. Hanna, Work in progress, November 1998

[17] 2200 INTERNET OFFICIAL PROTOCOL STANDARDS. J. Postel. June 1997.

[18] "Examining Shared Trees with Group Migration and Different Metrics", K. Carlberg, J. Crowcroft; Submitted to IEEE IWQoS 99, University College London, 1999.

[19] "EXPRESS Multicast", H. Holbrook, D. Cheriton, E-Mail Communication on RMRG list.

[20] "IP Router Alert Option", RFC 2113, D. Katz, Feb 1997

[21] "IP Encapsulation within IP", C. Perkins, Oct.1996. RFC 2003,

[22] "Internet Group Management Protocol", version 2; W. Fenner, Nov.1997. RFC 2236,

[23] "Administratively Scoped IP Multicast" D. Meyer, July 1998, RFC 2365

[24] "Multicast-Scope Zone Announcement Protocol (MZAP)", Handley, M., Work in Progress, December 1997.

[25] Scoped Address Discovery Protocol (SADP) Roger Kermode, Work in Progress 1998

[26] mtrace, unix tool, available at `http://www.mbone.com/`

[27] IGMPv3, Fenner et al Work in progress, 1998

[28] "UNIX Network Programming", W. Richard Stevens, pp55, Prentice-Hall, ISBN: 0-13-490012-X

[29] "TCP/IP Illustrated, Vol.1"., W. Richard Stevens, pp177, Addison-Wesley, ISBN: 0-201-63346-9

[30] "IPv6 Specification" , S. Deering, R. Hinden, Dec.1995 RFC 1883,

[31] "Multicast Interopability Specification", D. Thaler, Work in Progress, 1997/98

[32] "CBT Multicast Border Router Specification", A. Ballardie, B. Cain, Z. Zhang; Work in progress, March 1998.

[33] "Protocols for Distributed Interactive Simulation Applications", IEEE Standard for Information Technology - IEEE Std. 1278-1993.

[34] "The Ordered Core Based Trees Protocol", C. Shields and J.J.Garcia-Luna-Aceves in Proceedings infocomm'97, pp 884-91, April 1997.

[35] "Simulation of the YAM Protocol", K.Carlberg, Submitted to ACM CCR, 1999. available at `ftp://cs.ucl.ac.uk/darpa/ccr_yam99.ps.gz`

[36] "Improving Internet Multicast with Routing Labels," B. N. Levine and J. J. Garcia-Luna-Aceves, IEEE International Conference on Network Protocols (ICNP-97), October 28 - 31, 1997. p. 241-250.

[37] Pragmatic General Multicast A. Speakman et al Work in Progress, 1998.

[38] Network Address Translators "The IP Network Address Translator (Nat)", P. Francis, K. Egevang, RFC 1631, 05/20/1994