Protocol Analysis of Any-Source Multicast and Source-Specific Multicast

Hitoshi Asaeda

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Protocol Analysis of Any-Source Multicast and Source-Specific Multicast

Hitoshi Asaeda

N° 5080

January 2004

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--5080--FR+ENG

Protocol Analysis of Any-Source Multicast and Source-Specific Multicast

Hitoshi Asaeda *

Thème 1 — Réseaux et systèmes
Projet Planete

Rapport de recherche n° 5080 — January 2004 — 24 pages

Abstract: It is in general recognized that IP multicast routing protocols are fairly complex and non-scalable and its property to make it work ideally requires additional maintenance cost to network administrators and operators. Hence although the Internet community has been doing a significant amount of research works on IP multicast over the last decade and most router vendors already support basic IP multicast routing protocols, there are still deployment problems in the Internet.

In this document, we analyze the difficulties of traditional many-to-many communication and the benefit of one-to-many or few-to-many communication. While the former communication model called *Any-Source Multicast (ASM)* does not require the source address specification, the later communication model called *Source-Specific Multicast (SSM)* requires the source address specification, when an application joins to the group. We mainly study the difference of the routing protocols between these models.

Key-words: IP multicast, multicast routing protocol, ASM, SSM

* Hitoshi.Asaeda@sophia.inria.fr

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Les protocoles de transmission de données multipoint ASM et SSM

Résumé : Il est bien connu que les protocoles de routage IP multipoint sont assez complexes et non scalables et qu'ils nécessitent un coût de maintenance aux opérateurs et administrateurs réseaux. Bien que la communauté Internet ait effectué un grand nombre de travaux de recherches sur les protocoles multipoint IP ces dix dernières années et que plupart des fabricants de routeurs supportent le service de base IP multipoint, il reste toujours des problèmes de déploiement de ce service dans l'Internet.

Dans ce document, nous analysons les problèmes de la communication multipoint "plusieurs-vers-plusieurs" traditionnelle et les bénéfices de la communication multipoint "un-vers-plusieurs". Alors que l'ancien modèle de communication appelé *Any-Source Multicast (ASM)* n'exige pas de spécifier l'adresse de la source, le nouveau modèle de communication appelé *Source-Specific Multicast (SSM)* requiert la spécification de l'adresse de source, lorsqu'une application rejoint un groupe multipoint. Dans ce rapport, nous étudions principalement les différences entre les protocoles de routage de ces deux modèles.

Mots-clés : IP multipoint, les protocoles de routage IP multipoint, ASM, SSM

1 Overviews of Multicast Routing Protocols

1.1 Broadcast-and-Prune Type and Explicit-Join Type Multicast Routing Protocols

According to several early works [1, 2], original and conceptual properties conceived for multicast communication can be defined as follows:

- Host group model conformance
  Host group model [3] defines what the multicast service looks like to users of the network service interface within a host, but it doesn't how that service should be implemented. Further, it lists a set of properties a multicast routing protocol should exhibit, that contribute to its flexibility and generality.

- High probability of delivery
  The probability of successful delivery of multicast packets decreases when sending those packets over the wide-area network. However, the successful delivery rate should remain high enough to allow for the recovery of lost or damaged packets by end-to-end protocols.

- Low delay
  Low delay is an important property for many multicast applications. Small networks impose very little delay on the delivery of multicast packets, but the delays over the wide-area network are higher due to the greater geographic extent. Therefore, optimizing multicast routes can be an important factor in minimizing delay exacerbation.

In order to fulfill these conditions, multicast routing protocols used in the Internet must have the essential aim of wide-area routing, which establishes a reasonably optimal path between a multicast data sender and the other members of the group with keeping following conditions:

- Flexibility
  We may use many kinds of multicast applications over the routing tree. Therefore, a multicast routing tree should be built so as to reflect the nature of the application.

- Scalability
  In order to deploy multicast application in the Internet, it is no doubt that the routing protocols must be scale well. It would be necessary that routers not on the multicast routing tree require no knowledge of the tree whatsoever.

- Simplicity
  A multicast routing algorithm should be as simple as possible, since complexity might break the protocol robustness and causes maintenance costs to network administrators.

- Independency
  A multicast routing algorithm should not interfere with unicast routing protocols and constructed routing topologies. It is important for policy control.

But fundamental properties of multicast routing protocols, on the contrary, bring complexities and difficulties than unicast routing protocols, mainly includes;

- Reverse Path Forwarding
  Each multicast router registers incoming interface (iif) and outgoing interface(s) (oif(s)) for each data stream [4]. Such information is needed for a mechanism called Reverse Path Forwarding (RPF) in order to effectively help avoiding multicast routing loop. However, RPF requires more information than unicast routing protocols, therefore it consumes additional performance costs and resources on each router. In addition, because of the adaptation of RPF, for multicast routing tree, asymmetric routing path is not allowed.

- Aggregation of routing entries
  Address aggregation of unicast routing entries is well-known technique. It is achieved by careful unicast address assignment. However, conceptually, multicast address doesn't have any network topological dependency. This implies that aggregated routing information exchange for multicast is currently impossible. Uncompressing corresponding multicast address ranges on a routing table would affect not only resource consumption on each router but also management cost to network administrators.

Before discussing about concrete multicast routing architecture, we remember that each unicast routing protocol is categorized by the way to exchange routing information between their neighbor routers and the decision which routing protocol should be used in a network basically depends on the target network size and policy.

For example, RIP [5, 6] is "distance vector" routing protocol, which includes in its routing updates a vector of distance (i.e., hop counts). The algorithm of this kind of protocol works on the concept that routers exchange all the network numbers they can reach via periodic broadcasts of the entire routing table, therefore in large networks, the routing table exchanged between routers becomes very large and hard to maintain. OSPF [7] is a "link state" routing protocol, which computes each link state. This protocol is more advanced routing protocol that have addressed the deficiencies of distance vector protocols, because each router inside the network doesn't exchange routing tables, but calculates and builds its own routing table based on a shortest path algorithm. However, In very large networks and in case of route fluctuation caused by link instabilities, link state retransmission and recomputation will become too large for any router to handle [8].

RIP and OSPF can be useful routing protocols with in a limited domain. These routing protocols are abbreviated to Interior Gateway Protocols (IGPs). What we have gained by segregating the world into administrations is the capability to have one large network, i.e., the Internet, divided into smaller and more manageable networks. These networks called Autonomous Systems (ASes) can have their own set of policies including the decision which IGP is used. But each negotiation for routing exchanges between ASes is obviously necessary. Moreover, such routing protocol called Exterior Gateway Protocols (EGPs) should impose no restrictions on the underlying Internet topology and must be scalable [8]. Currently, BGP [9] is selective and de facto EGP in the Internet. BGP is a "path vector" routing protocol

carrying a sequence of AS numbers over TCP. It doesn't send periodic routing messages and doesn't send any routing table unless routing topology is changed.

Reasonable unicast routing classification has been providing smooth Internet communication and manageable environment. Now we can understand such classification should be also adapted to multicast routing protocols in order to harmonize the same philosophy of routing concept, even with the architectural differences of unicast and multicast. An *"intra-domain multicast routing protocol"* and an *"inter-domain multicast routing protocol"* are the keywords for this purpose.

When MBone [10, 11] was established as an experimental multicast backbone, irrespective of native connections or by virtual point-to-point links called "tunnels" using IP over IP technique [12], every multicast router in the MBone used Distance Vector Multicast Routing Protocol (DVMRP) [13] for the routing exchange. DVMRP is a "broadcast-and-prune type" protocol since the first datagram for any source address and group address pair (hereafter referred to as (S,G)) is forwarded across the entire network. Upon receiving this traffic, leaf routers may transmit prune messages back toward the source if there are no group members on their directly-attached leaf subnetworks. The prune messages remove all branches that do not lead to group members from the tree, leaving a source-based Shortest-Path Tree (SPT). After a period of time, the prune state for each (S,G) expires to reclaim router memory that is being used to store prune state pertaining to groups that are no longer active. If those groups happen to still in use, a subsequent datagram for the (S,G) will be broadcast across all downstream routers. This will result in a new set of prune messages, serving to regenerate the SPT for the (S,G) [14].

MOSPF [15] is another broadcast-and-prune type protocol. It is based on a link-state routing algorithm. When the initial packet arrives, the source subnetwork is located in the MOSPF link state database. The MOSPF link state database is simply the standard OSPF link state database with the addition of Group-Membership LSAs. Whenever a new group appears or an old group disappears from a link, the designated router on that link floods the new state to all other routers in the network. Protocol Independent Multicast - Dense Mode (PIM-DM) [16] is also broadcast-and-prune type protocol. It is similar to DVMRP, though, to find routes back to sources, PIM-DM relies on the presence of an existing unicast routing table, therefore PIM-DM is independent of the mechanisms of any specific unicast routing protocol.

These protocols are designed for areas with a high density of listeners. Furthermore, these protocols exhibit another scaling characteristic that is routers which do not intend to send and receive multicast packets also need to take account of the routing messages. Hence, these protocols don't fulfill above stated properties especially for "High probability of delivery" and "Low delay", if they are used in a wide-area.

On the other hands, Core Based Tree (CBT) [2, 17] and Protocol Independent Multicast - Sparse Mode (PIM-SM) [18, 19] are explicit-join type protocols, which are suitable when multicast receivers are sparse in a wide area with many hop counts. This type of multicast routing protocol builds a shared tree rooted at a Core Router or a Rendezvous Point (RP) per group prefix. The coordination of the shared tree is *receiver-based*, i.e., no router is

involved in becoming part of the tree for a particular group unless that router is intent on becoming a member of the group. This gives a significant benefit to all routers out of the routing tree, since they are incurred no tree-building overhead. Then these explicit-join type protocols are recognized as more scalable routing protocols than broadcast-and-prune type protocols.

This shared tree, however, causes traffic concentration on the core router, and the packets sending from source to the receivers does not travel via the shortest path. There are several tradeoffs for using a shared tree and an SPT [20], therefore, PIM-SM introduces to optionally switch to an SPT per source.

PIM-SM is currently the most popular multicast routing protocol. Many router venders have supported it recently, and IETF has agreed PIM-SM is an effective multicast routing protocol as mostly a de facto standard, so why it has not widely deployed in the Internet yet? One of the main reasons is, in a nutshell, PIM-SM still cannot completely fulfill above all conditions. It should be categorized as an intra-domain multicast routing protocol, not an inter-domain multicast routing protocol.

In a next section, we study the detail specification of PIM-SM in order to investigate the technology of native IP multicast. It would provide a good hint to make general research contribution for multicast routing protocol deployment.

1.2 Protocol Independent Multicast - Sparse Mode (PIM-SM)

In this section, we summarize PIM-SM's behavior. PIM relies on an underlying topology-gathering protocol to populate a routing table with routes. This routing table is called Multicast Routing Information Base (MRIB). The routes in this table may be taken directly from the unicast routing table, or it may be different and provided by a separate routing protocol such as MBGP [21], which is explained later. This simply means PIM doesn't exchange any routing table between neighbor routers. It only acts to construct multicast routing tree by sending a message including a joined or pruned incoming interface address and by receiving a message including a joined or pruned outgoing interface address for appropriate multicast data.

PIM-SM routes data packets from sources to receivers without either the sources or receivers knowing a-priori of the existence of the others. For the routing tree formation as seen on Figure 1, this is essentially done in following three phases:

- Construct RP tree
- Stop Register message encapsulation
- Switch to shortest-path tree

In a first phase, a multicast receiver expresses its interest in receiving traffic destined for a multicast group. Typically, Internet Group Management Protocol (IGMP) [24, 25] for IPv4 or Multicast Listener Discovery (MLD) [26, 27] for IPv6 is used for it. One of the receiver's local routers is elected as the Designated Router (DR). DR election is performed using PIM

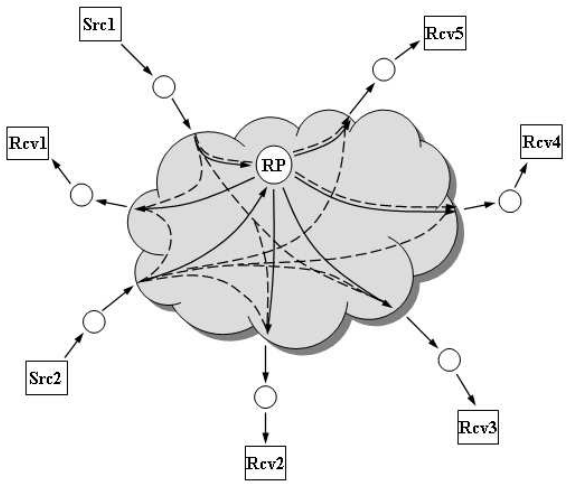

Figure 1: Multicast routing tree rooted at a Rendezvous Point

Hello messages, which are sent periodically on each PIM-enabled interface, in order to learn about the neighboring PIM routers on each interface. On receiving the receiver's expression of interest, the DR then sends a PIM Join message towards the RP for that multicast group. This Join message requesting to receive group G's data coming from any sources is known as a PIM (*,G) Join. The PIM (*,G) Join travels hop-by-hop towards the RP for the group, and in each router along the path, multicast tree state for group G, (*,G) state, is instantiated.

Eventually, an RP is a router that has been selected among several candidate RPs in PIM domain, where the boundary is defined by some policies or by Autonomous Systems (ASes). This selection mechanism is done either through a bootstrap mechanism or through static configuration. One dynamic way to do this is to use the Bootstrap Router (BSR) mechanism [34]. One router in each PIM domain is elected the BSR through a simple election process using own priority value. Candidate RPs periodically unicast their candidacy to the BSR. From the candidates, the BSR picks an RP-set, and periodically announces this set in a Bootstrap message. Bootstrap messages are flooded hop-by-hop throughout the domain until all routers in the domain know the RP-Set. An RP is defined per multicast address or multicast address prefix. As a view of the availability, multiple candidate RPs can be configured to cover same multicast address prefixes.

When many receivers join the group, their Join messages converge on the RP, and form a distribution tree for group G that is rooted at the RP. This shared tree which is shared by all sources sending to that group is known as the RP tree (RPT). Join messages are resent periodically so long as the receiver remains in the group.

When a multicast data sender starts sending data destined for a multicast group, the sender's local router, designated router (DR), takes those data packets, encapsulates in

unicast packets, and sends them directly to the RP. The RP receives these encapsulated data packets, decapsulates them, and forwards them onto the RPT natively. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are known as PIM Register packets.

Since normally Register-encapsulation of data packets is inefficient, in a second phase, an RP chooses to switch to native forwarding. When the RP receives a register-encapsulated data packet from source S on group G, it can initiate a PIM (S,G) source-specific Join towards S. This Join message travels hop-by-hop towards S, instantiating (S,G) multicast tree state in the routers along the path. (S,G) multicast tree state is used only to forward packets for group G if those packets come from source S.

While the RP is in the process of joining the source-specific tree for S, the data packets continue being encapsulated to the RP. Then, when packets from S also start to arrive natively at the RP, the RP receives two copies of each of these packets. At this point, the RP starts to discard the encapsulated copy of these packets, and it sends a RegisterStop message back to S's DR to prevent the DR unnecessarily encapsulating the packets. A sender may start sending before or after a receiver joins the group, thus this phase may happen before the RPT to the receiver is constructed.

Although having the RP join towards the source removes the encapsulation overhead, it does not completely optimize the forwarding paths. For many receivers the route via the RP may involve a significant detour when compared with the shortest path from the source to the receiver. To obtain lower latencies, in a third phase, the receiver site DR may optionally initiate a transfer from the RPT to a source-specific shortest-path tree (SPT) with issuing a PIM (S,G) Join towards S. This instantiates (S,G) state in the routers along the path to S. After this join either reaches S's subnet, or reaches a router that already has (S,G) state, data packets from S start to flow following the (S,G) state until they reach the receiver.

Finally, the receiver receives traffic from S along the SPT between the receiver and S. The RP is receiving the traffic from S, but this traffic is no longer reaching the receiver along the RPT. It is just kept for serving the data to remaining receivers which are using the RPT and to new receivers which will use the RPT. Since senders and receivers may come and go at any time, all three phases may occur simultaneously.

Additionally, it should be noted that there may be the case more than one upstream router with join state for the same group or source-group pair. In this situation, when duplicate data packets appear on the LAN from different routers, these routers notice this, and then elect a single forwarder. This election is performed using PIM Assert messages, which resolve the problem in favor of the upstream router which has (S,G) state, or if neither or both router has (S,G) state, then in favor of the router with the best metric to the RP for RP trees, or the best metric to the source to source-specific trees.

2 Inter-domain Multicast Routing Protocols

2.1 Condition for Inter-domain Multicast Routing Protocols

PIM-SM has been designed to be used in a wide area where each multicast receiver is sparsely distributed. However, not only the specification complexities, still it has several problems stated below when it is used in the Internet.

- Traffic concentration
  As described above, PIM-SM uses RP to collect both senders and receivers information. In other words, every data traffic and join/leave messages for RPT must reach to the selected RP. It is obvious that this traffic concentration is not scalable if the size of PIM domain is large enough.

- Third-party resource dependency
  Although every source data goes through the RP in the PIM domain, an available RP may be located far from the source. In this case, it may happen the data doesn't come to each receiver or the quality is bad even the sender is close to the receiver.

- Bootstrap scalability
  One router in each PIM domain is elected the BSR, though, the election messages are broadcasted to whole PIM domain. In addition, candidate RPs periodically unicast their candidacy to the BSR, and the BSR periodically announces an RP-set in a Bootstrap message. Bootstrap messages are also flooded hop-by-hop throughout the PIM domain until all routers in the domain know the RP-Set.

According to these problems, PIM-SM itself could not be recognized as an appropriate multicast routing protocol used in a very wide network, then it would not be an inter-domain multicast routing protocol. When IETF arose these problems, two interesting multicast routing protocols were proposed. One was Border Gateway Multicast Routing Protocol (BGMP) [30] and the other was Simple Multicast [32].

BGMP is a hierarchical multicast routing algorithm that uses *border routers* acting as nodes in the routing tree. The concept comes from BGP, which means multicast address prefix is managed by own Group Routing Information Base (GRIB). BGMP avoids global knowledge of all active groups, sources and receivers. Main concerning point is BGMP requires new multicast addressing architecture. Because of this reason, working in conjunction with the complementary Multicast Address Set Claim (MASC) architecture [31] is the requirement. Such conceptual change leads a big impact for the deployment.

Simple Multicast uses core router (or rendezvous point) technique and a routing algorithm based on an RPF. It proposes a *channel* identifier used by Source-Specific Multicast, which is detailed in later sections, in order to make it scale, but the biggest change, which is that an end-node joins the group by sending a special join message towards the core, creating state in the routers along the path until the join packet hits the core or a router that is already on the tree for this multicast group, also leads a big impact.

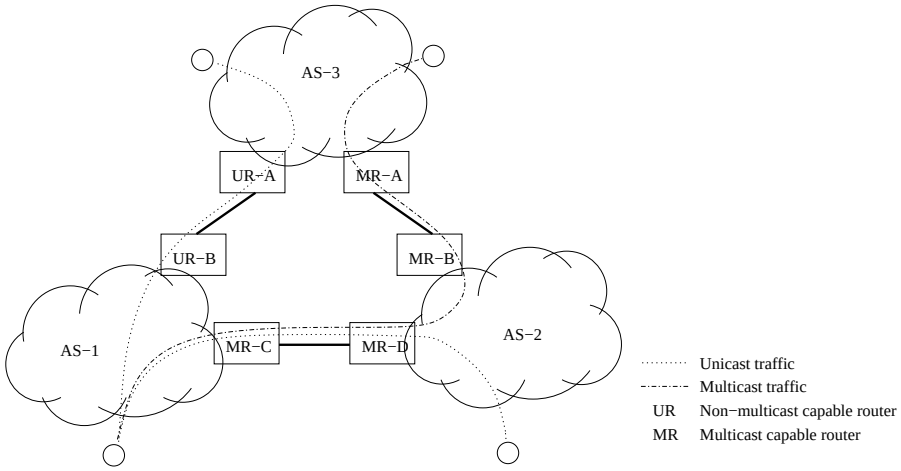

Figure 2: Policy-based multicast routing path controled by MBGP

Therefore, it was decided that each of BGMP and Simple Multicast should be a long-term solution in IETF, and as a short-term solution, Multicast Source Discovery Protocol (MSDP) [33], which can collaborate with PIM-SM, has been discussed. But before studying MSDP, Multiprotocol Extensions for BGP-4 (MBGP) [21], which is another component to make PIM-SM as an inter-domain multicast routing protocol is shown next.

2.2 Multiprotocol Extensions for BGP-4 (MBGP)

As described above, PIM uses underlying topology so called MRIB for multicast packets forwarding. In order to decide the topology, regular unicast routing table can be used, but network administrators may want to distinguish each routing policy. Multiprotocol Extensions for BGP-4 (MBGP) [21] adds the capability to enable multicast routing policy throughout the Internet.

The primary role of the MRIB in the PIM protocol is to provide the next hop router along a multicast-capable path to each destination subnet. The MRIB is used to determine the next hop neighbor to which any PIM Join/Prune message is sent. Data flows along the reverse path of the Join messages. Thus, in contrast to the unicast RIB which specifies the next hop that a data packet would take to get to some subnet, the MRIB gives reverse-path information, and indicates the path that a multicast data packet would take from its origin subnet to the router that has the MRIB. Thus MBGP allows having a unicast routing topology different from a multicast routing topology, and network administrators have more control over their networks and resources.

In Figure 2, three ASes are connected by each backbone router. Some of them only support unicast (UR), and others support multicast (MR). Among UR, BGP is used for

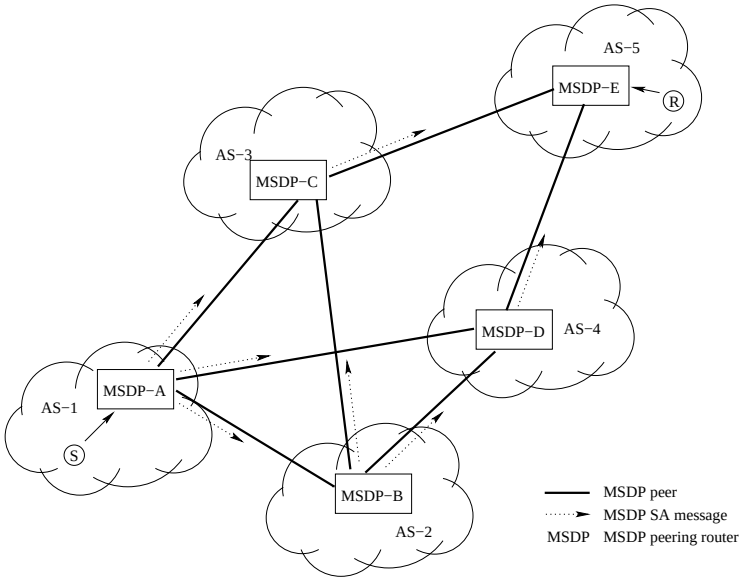

Figure 3: MSDP peering example

unicast routing exchange. For MR, BGP and MBGP may be used for unicast RIB and MRIB constructions. These combinations provide the possibility to flow the data shown on this figure.

2.3 Multicast Source Discovery Protocol (MSDP)

In the PIM-SM model, multicast sources and receivers must register with their local RP. Actually, a source may register with one RP and receivers may join to a different RP, but the key point is that the RP *must* know all the source addresses for any particular group. RPs in other domains have no way of knowing about sources located in other domains, so some method is needed for the RPs to exchange information about active sources. This information exchange is done with Multicast Source Discovery Protocol (MSDP) [33].

MSDP is a mechanism to connect multiple PIM domains together. Each PIM domain uses its own independent RP(s) and does not have to depend on RPs in other domains. By dividing PIM domain, network administrators can control reasonable size of the number of PIM routers. This brings several benefits, which are that policy control becomes easier than before, Bootstrap message can be restrained, and so on.

MSDP routers in a PIM-SM domain have an MSDP peering relationship with MSDP routers in another domain (Figure 3). Every MSDP routers must be Candidate RP, statically configured RP or Anycast RP [22]. The peering relationship is made up of a TCP connection

in which control information is exchanged. Each domain has one or more connections to this virtual topology. The purpose of this topology is to allow domains to discover multicast sources from other domains. If the multicast sources are of interest to a domain which has receivers, the normal source-tree building mechanism in PIM-SM will be used to deliver multicast data over an inter-domain distribution tree. It would be thought that this virtual topology would essentially be congruent and reasonable to the existing MRIB topology constructed by MBGP or regular unicast routing protocol.

The key advantage of the combination of PIM-SM/MSDP/MBGP is that it is a functional solution largely built on existing protocols. The key disadvantage is that, as a long-term solution, MSDP protocol suite may be susceptible to scalability problems. Further discussion of two particular problems follows.

One reason: the timescales for change of active source indications are much different than BGP was designed to carry. BGP wants to carry data that doesn't change very often, e.g. see route dampening. Sources can come and go at an arbitrary rate, so the rate of change of the information is potentially much higher. This situation has been well-recognized by the trouble, so called *Ramen Worm* [35] and *Sapphire* [36].

Another reason is, unfortunately, MSDP just can be a tentative patchwork of bypassing impending scalability problem. In other words, traffic concentration problem and third party problem, which are stated before, are still sitting on our side, because in any case all PIM router need to exchange the information of every data sender and receiver.

3 Source-Specific Multicast (SSM)

3.1 Advantages of SSM Architecture

According to the traditional multicast communication model, especially from the stand point of IP multicast routing protocol, the many-to-many communication model and protocol architecture has run into significant barriers for the wide-scale deployment. Mainly, these barriers are rooted in the problem to build efficient multicast routing trees for dynamic group memberships. More precisely, like PIM-SM, explicit-join type multicast routing protocols providing many-to-many communication use core routers (i.e. Rendezvous Point) and maintain core-rooted multicast routing tree (i.e. RPT). They require complex routing algorithms to construct and maintain multicast routing tree including the mechanism to switch to optimized source-rooted routing tree (i.e. SPT).

Let's recall Figure 1. In this PIM-SM network, we have one RP, two multicast data senders, and five receivers, and assume these two senders use same multicast address. When each receiver sends Join messages with specifying multicast address as known as (*,G) Join to each upstream router, each upstream router tries to construct RPT (a solid arrow) with sending PIM (*,G) Join message towards the RP.[1] Finally, first hop router for receiver, DR, switches to construct SPT (a dotted arrow) based on defined condition. Unlike this figure,

[1] Hereafter, (*,G) Join/Leave and (S,G) Join/Leave mean IGMP or MLD Join/Leave. For other Join/Leave, e.g., PIM (*,G) Join, it is mentioned precisely.

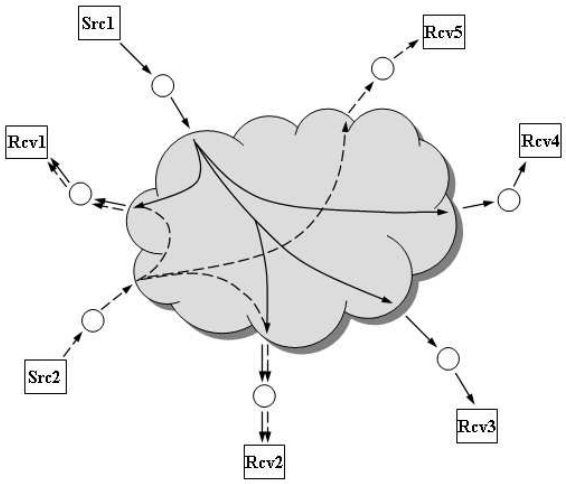

Figure 4: Multicast routing tree rooted at each source

however, if network widely spreads, or if the large number of receivers exist, then it causes scalable problems, even with MSDP.

Such complexity of multicast routing tree coordination comes to focusing traditional many-to-many multicast communications, called Any-Source Multicast (ASM). ASM has been designed any kinds of multicast applications can work well over it. However, supporting *any* kinds of applications by single multicast communication architecture is not realistic, and it *is* the reason that general multicast routing algorithm becomes pretty complex and the scalable problems have never solved.

We can categorize multicast applications to two types; one is many-to-many communication, and the other is one-to-many or few-to-many communication. With regard to an application which requests synchronous group communication over the Internet, one-to-many or few-to-many communication model is mostly sufficient. When we consider one-to-many or few-to-many communication, it can be assumed that the application clients already know each sender address(es) as well as multicast address advertised via the World Wide Web, DNS, SDP/SAP model [39, 40] and Channel Reflector [38]. In this case, each client can specify and send interesting source address(es) with group address to the upstream router as a group membership information, when he request to receive the data. This becomes a good *collaborative work* for the upstream router because the router does not need to try to find source address. It is easy to understand that eliminating source address discovery procedure from multicast routing protocols is highly beneficial if we remember the problems caused by MSDP stated in a previous section. Moreover, the router can eliminate the process to coordinate and maintain a shared tree because it can directly construct an SPT from the initial phase, therefore, a core router like an RP can be eliminated from the network and

routing protocol itself, then multicast routing tree becomes simple, and finally scalability problem would be effectively reduced.

Now we can understand the advantages of one-to-many or few-to-many multicast communication model also from the routing protocol's point of view. Because an end-node works on this communication model can specify the interesting source addresses, this architecture is called Source-Specific Multicast (SSM) [23]. Since SSM solely maintains explicit source-based routing tree, it eliminates many multicast routing complexities, and then it is recognized as a feasible communication model to deploy multicast services in the Internet successfully.

Let's see Figure 4. This figure shows SSM communication environment. A solid line shows the SPT whose sender is S1, and a broken line shows the SPT whose sender is S2. Each first hop multicast router for each receiver just coordinates appropriate SPTs for each data receiver. Comparing with Figure 1, a routing tree totally becomes simple.

Talking the behavior of an end-node, in an ASM environment, each end-node sends Join or Leave message only indicating multicast address referred to as (*,G) Join/Leave message, but in SSM environment, it must send Join or Leave message specifying source address(es) as well as multicast address referred to as (S,G) Join/Leave message. This makes the *collaborative work* stated above.

3.2 SSM Adaptation to Current Environment

SSM improves scalability of multicast routing protocols with focusing a realistic multicast communication model. The concerning point is, however, implementations of Internet Group Management Protocol Version 3 (IGMPv3) [25] for IP Version 4 and Multicast Listener Discovery Version 2 (MLDv2) [27] for IP Version 6 on every end-node are indispensable as well as on every router in order to specify interesting source address and multicast address pairs called multicast channels. Therefore, in order to not make SSM deployment delay caused by the implementation costs, having an effective and accurate IGMPv3 and MLDv2 reference implementations is urged and desired. In [29], while we have illustrated the concept, design, and logic of MLDv2 host-side kernel implementations for BSD operating systems, in order to deploy SSM in the Internet smoothly, we summarize IGMPv3 and MLDv2 protocol concept in a next section.

4 IGMPv3 and MLDv2 Protocol Concepts

4.1 End-node Behavior

By trying to design SSM communication, both of host-side and router-side extensions are required in order to make a join or leave process to the pair of interesting source address(es) and group address as known as (S,G) channels. As a protocol level, each extension is done by IGMPv3 implementation for IPv4 and by MLDv2 implementation for IPv6.

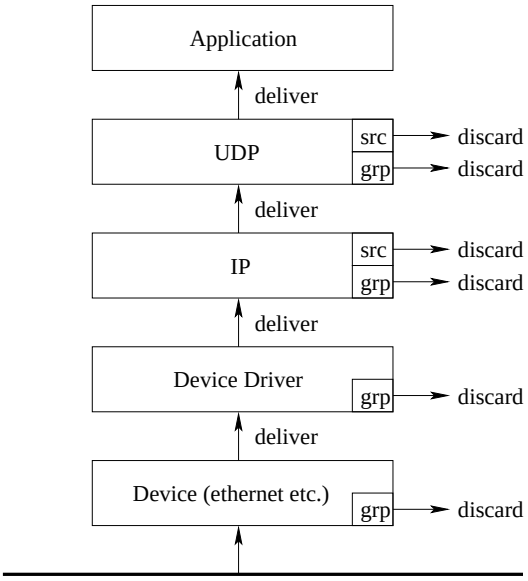

Figure 5: Multicast packet filtering based on (S,G) information

IGMPv3 and MLDv2 provide the capability of handling (S,G) notification. When an application working on an end-node triggers (S,G) Join, it gives "socket state", which consists of following four items, to IGMPv3 and MLDv2 capable kernel.

(interface, multicast address, filter mode, source address list)

An "interface" indicates an incoming interface of multicast data. If an application doesn't specify it, kernel chooses a default interface to be used. A "multicast address" is the destination address of the data and a "source address list" shows single or multiple data sender(s). With regard to the protocol specification of IGMPv3 and MLDv2, an end-node needs to specify "filter mode", which is either `INCLUDE` or `EXCLUDE`. In `INCLUDE` mode, reception of packets sent to the specified multicast address is requested only from those IP source addresses listed in the source address list parameter, and in `EXCLUDE` mode, reception of packets sent to the given multicast address is requested from all source addresses except those listed in the source address list parameter. Based on the semantics of SSM [23], only the request of `INCLUDE` filter mode prompts SSM communication.

Due to these concepts, additional filtering implementation is embedded in each kernel. Comparing the original implementation, an SSM-capable end-node makes additional source address filtering on IP layer and socket (UDP) layer (Figure 5).

In addition to the per-socket multicast reception state, kernel must also maintain and compute multicast reception state for each interface. This state is called "interface state" and consists of following three items.

(multicast address, filter mode, source address list)
This per-interface state is derived from the per-socket state, but may differ from the per-socket state when different applications have differing filter modes and/or source lists for the same multicast address and interface. This state reflects the aggregate behavior of the information maintained by all sockets to the kernel.

4.2 New Group Membership Report Message

After an end-node sends new Group Membership Report to an upstream router, the router can handle which (S,G) pairs are of interest to neighboring systems with gathering the information that is provided to whichever multicast routing protocol is used.

This new Group Membership Report message is defined to "State-Change Record". It is classified two types of records, "Filter-Mode-Change Record" and "Source-List-Change Record". The former is sent by kernel whenever some application causes a filter mode change from previous interface state entry for a particular multicast address. This record type is shown by `CHANGE_TO_INCLUDE_MODE` or `CHANGE_TO_EXCLUDE_MODE` (hereafter, referred to as `TO_IN` and `TO_EX` respectively). The later is sent by kernel whenever some application causes a change of source address list that is not coincident with a filter mode change from previous interface state entry for a particular multicast address. This record type is shown by either `ALLOW_NEW_SOURCES` to contain a list of the additional sources that the system wishes to listen from, or `BLOCK_OLD_SOURCES` to contain a list of the sources that the system no longer wishes to listen from (hereafter, referred to as `ALLOW` and `BLOCK`).

As another Report message, IGMPv3 and MLDv2 prepares "Current-State Record". This Report message is sent for the response to each Query message sent by router. It reports the current reception state of the interface where the Query is received.

4.3 Multicast Source Filters (MSF)

When an application requests new (S,G) Join, it uses embedded Application Program Interface (API), which controls socket operations. In SSM definition, new Application Program Interfaces (APIs) for `setsockopt()`, `getsockopt()` and `ioctl()` are used [28]. These APIs called Multicast Source Filters (MSF) extensions are classified to "IPv4 MSF API" and "Protocol-Independent MSF API". In IPv6 application, Protocol-Independent MSF API should be used. Each API provides an another taxonomy, "Basic API" and "Advanced API".

Basic API, which implies using `setsockopt()` and `getsockopt()` operations, can minimize changes needed in existing IP multicast application source code to add MSF operations, like following example.

Case-1: IPv4 Basic MSF API

```
bcopy(&in_grp, &ims.imr_multiaddr, sizeof(in_grp));
```

```
bcopy(&in_src, &ims.imr_sourceaddr, sizeof(in_src));

if (setsockopt(socket, IPPROTO_IP, IP_ADD_SOURCE_MEMBERSHIP,
        (char *)&ims, sizeof(ims)) < 0)
    perror("cannot listen group");
```

Case-2: IPv6 (Protocol Independent) Basic MSF API

```
bcopy(&grp, &gsr.gsr_group, grp.sin6_len);
bcopy(&src, &gsr.gsr_source, src.sin6_len);

if (setsockopt(socket, IPPROTO_IPV6, MCAST_JOIN_SOURCE_GROUP,
        (char *)&gsr, sizeof(gsr)) < 0)
    perror("cannot listen group");
```

However, an application using this API cannot specify multiple source addresses by one request and cannot change filter mode without leaving previous Join request. For example, if an application initially made `INCLUDE` request, then following `EXCLUDE` request without leaving previously joined `INCLUDE` source list is not permitted. In this case, kernel returns error to the application. It is obvious that Basic API is really effective for current multicast applications to be changed to support SSM communication.

On the contrary, Advanced API, which implies using `ioctl()` operation, can specify multiple sources simultaneously and can change filter mode without leaving previous Join request. Following is a part of a sample code.

Case-3: IPv6 (Protocol Independent) Advanced MSF API

```
if ((gf = malloc(GROUP_FILTER_SIZE(numsrc))) == NULL)
    perror("memory allocation error");

bzero(gf, GROUP_FILTER_SIZE(numsrc));
gf->gf_interface = index;
gf->gf_fmode = mode;
gf->gf_numsrc = numsrc;
bcopy(&grp, &gf->gf_group, grp.sin6_len);
for (i = 0; i < numsrc; i++)
    bcopy(&src[i], &gf->gf_slist[i], src[i].sin6_len);
```

```
if (ioctl(socket, SIOCSMSFILTER, gf) != 0)
    perror("cannot listen group");
```

With Advanced API, the legality is not dependent on the previous state, since new request overwrites existing filtering condition. So it is only when a request is completely same of the previous one that Advanced API returns an error.

Although this modification is not so hard, it is obvious that Basic API is rather simple and familiar to general application programmers.

4.4 Interface State Transition

In this section, we study the interface state transition mechanisms using application examples working on top our our kernel implementations. At first, let's assume that socket `s1` makes an `INCLUDE` mode (S,G) join request to indicate that it wants to receive multicast data, whose multicast address is `m` and source address is either `src-1` or `src-2` through interface `i`:

`s1: (i, m, INCLUDE, {src-1, src-2})`

This requiest expresses the following state to the kernel;

`(INCLUDE, {src-1(1), src-2(1)}),`
`(EXCLUDE, {null})`

Because the interface state is changed by this join, the kernel triggers `ALLOW(m, {src-1, src-2})` message transmission.

When another socket `s2` requests the following join;

`s2: (i, m, INCLUDE, {src-2, src-3})`

The group and source list maintained in the kernel is merged to the existing list. Since source address `src-2` is used by both `s1` and `s2`, the reference count of `src-2` is changed to "2".

`(INCLUDE, {src-1(1), src-2(2), src-3(1)}),`
`(EXCLUDE, {null})`

If all socket states have a filter mode of `INCLUDE`, then the filter mode of the interface state is `INCLUDE` and the source list of the interface record is the union of the source lists of all the socket states. Hence new interface state is;

`(m, INCLUDE, {src-1, src-2, src-3})`

and the kernel sends `ALLOW(m, {src-3})` message to tell the difference from previous interface state.

When the kernel receives a new socket request;

`s3: (i, m, EXCLUDE, {src-3, src-4})`

the interface state is changed to;

`(INCLUDE, {src-1(1), src-2(2), src-3(1)}),`
`(EXCLUDE, {src-3(1), src-4(1)})`

If some socket state has `EXCLUDE` filter mode, the interface state is also `EXCLUDE` mode and

the source list of the interface state is the intersection of the source lists of all socket states in EXCLUDE mode minus those source addresses that appear in any socket state in INCLUDE mode. Hence new interface state is changed to;

(m, EXCLUDE, {src-4})

and the kernel sends TO_EX(m, {src-4}) message. This state indicates that the kernel receives multicast data whose destination address is m, except from the source whose address is src-4.

Additional socket request;

s4: (i, m, EXCLUDE, {src-1, src-3})

changes the interface state to;

(INCLUDE, {src-1(1), src-2(2), src-3(1)}),
(EXCLUDE, {src-1(1), src-3(2), src-4(1)})

This join is the second EXCLUDE request for (i, m) pair, thus the interface state given by an intersection of EXCLUDE source list is;

(EXCLUDE, {src-3})

Therefore the final interface state becomes;

(m, EXCLUDE, {null})

and the kernel sends ALLOW(m, {src-4}) message. At this moment, the interface state is equivalent to (*,G) join state.

Next, let's see (S,G) leave request. When s2 leaves from the multicast group, the kernel makes (S,G) leave procedure to handle the filter mode and the group and source list and to reduce the reference count of joined source list. As the result, each source list becomes;

(INCLUDE, {src-1(1), src-2(1), src-3(0)}),
(EXCLUDE, {src-1(1), src-3(2), src-4(1)})

then the interface state is changed to;

(m, EXCLUDE, {src-3})

Since the kernel releases the memory of the source address whose reference count is "0", src-3 is removed from the INCLUDE source list. BLOCK(m, {src-3}) message is sent afterward.

If remaining applications are finished, i's state is gone back to the initial state;

(null, INCLUDE, {null})

and the kernel sends a TO_IN(m, {null}) message, which indicates there is no application receiving the data whose multicast address is m from interface i any more.

4.5 Query Response and Host Compatibility Mode

Upon reception of an IGMP/MLD message containing a Query, after the validity of the message is verified, the node starts to process the Query. Instead of responding immediately, the node delays its response by a random amount of time derived from the "Maximum Response Code" in the received Query message. If the message is an IGMPv3 Query, a host kernel works for the new timer management with allocating the pending Report. If its message type is a General Query, the kernel starts the random timer for the "Interface Timer", and if others, the kernel starts the random timer for the "Group Timer".

As the same thoughts of the State-Change Report message retransmission, if a new Query message is arrived before the Interface Timer or the Group Timer is expired, pending Report must be compared with new response and merged to a new Current-State Report. The difference from the mergence of the State-Change Report is that each Query message expects the coverage of the response given by the Report because of the following relation between each Report message;

A Report message for a General Query
$\supseteq$ A Report message for a Group-Specific Query
$\supseteq$ A Report message for a Group-and-Source-Specific Query

Based on this condition, pending response or newly arrived Query may be canceled. This consideration minimizes the amount of the traffic of the Report messages.

In addition to such timer management and pending Report transmission, both IGMPv3 and MLDv2 provide backward compatibility with previous versions of each protocol. The kernel modifications interoperate with applications and routers that have not yet been upgraded to IGMPv3/MLDv2. This compatibility is controlled with the value of the "*host compatibility mode*" assigned per interface, so as to provide appropriate actions depending on the versions of other hosts and upstream routers on the LAN. When an older version General Query is received, the host sets its compatibility mode timer called "Older Version Querier Present Timer" to the defined interval. And when the host's interface is operating in, for instance, IGMPv2 compatibility mode, it sends an IGMPv2 Reports through the interface until its Older Version Querier Present Timer is expired. If the upstream router stops transmitting an IGMPv2 General Query and speaks IGMPv3, or if another IGMPv3 capable router becomes the querier, the host can gracefully change to speak IGMPv3.

However, such *dynamic* host compatibility mode adaptation might be meddlesome in some occasion. For example, if some node or router sends an IGMPv2 General Query, all IGMPv3 capable hosts attached on the same LAN start IGMPv2 host compatibility mode and behave as IGMPv2 hosts. At this moment, these hosts do not request SSM join within the timer expiration. After the timer expired, each host tries to go back to IGMPv3 host, but if it receives an IGMPv2 Query again before its timer is expired, it restarts the compatibility mode timer. This indicates that it is fairly easy to make end-nodes stop acting as IGMPv3 hosts intentionally or accidentally.

Because of these observations, our implementations have introduced to configure the host compatibility mode statically by `sysctl` command. After the end-node operates the host compatibility mode to IGMPv3/MLDv2 by `sysctl`, it just ignores older version General Query and keeps transmitting IGMPv3/MLDv2 Report. Although this capability requires the attention that all legitimate neighbor nodes and routers must be upgraded, it would be efficient to avoid the denial of service.

5 Conclusion

When we talk about multicast deployment, we have many times heard about "chicken and egg" problem. The use of multicast is still driven more by the academic community than

by customer demand. This view might be correct. But we believe new architecture, SSM, would break this kind of story.

In this document, we summarize the difficulties of traditional ASM communication and the benefit of SSM communication. While we mainly study the difference of the routing protocols between these models, we also mention the protocol specifications of IGMPv3 and MLDv2. Since we have opened our IGMPv3 and MLDv2 host-side kernel implementations, we hope this document would become the additional contribution for the multicast communication deployment. As well as making experiences to use SSM applications on our kernel over the Internet, studying the performance advantage would be a next research items. It may be able to contribute another deployment scenario.

References

[1] S. Deering, "Multicast Routing in Internetworks and Extended LANs", ACM Symposium on Communication Architectures and Protocols, pp.55-64, August 1988.

[2] T. Ballardie, P. Francis and J Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing", Proceedings of ACM SIGCOMM 93, pp.85-95.

[3] S. Deering, "Host Extensions for IP Multicasting", RFC1112, August 1989.

[4] S. Deering and D. P. Cheriton, "Multicast Routing in Datagram Internetwork and Extended LANs", ACM Transactions on Computer Systems, vol.8, no.2, May 1990.

[5] C. Hedrick, "Routing Information Protocol", RFC1058, June 1988.

[6] G. Malkin, "RIP Version 2 - Carrying Additional Information", RFC1723, November 1994.

[7] J. Moy, "OSPF Version 2", RFC2328, April 1998.

[8] Bassam Halabi, "Internet Routing Architectures", Cisco Press, 1997.

[9] Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC1771, March 1995.

[10] S. Casner and S. Deering, "First IETF Internet Audiocast", ACM SIGCOMM Computer Communication Review, vol.22, no.3, pp.92-97, July 1992.

[11] H. Eriksson, "MBONE: The Multicast Backbone", Communications of the ACM, vol.37, no.8, pp.54-60, August 1994.

[12] C. Perkins, "IP Encapsulation within IP", RFC2003, October 1996.

[13] D. Waitzman and C. Partridge, "Distance Vector Multicast Routing Protocol", RFC1075, November 1988.

[14] T. Maufer, "Deploying IP Multicast in the Enterprise", Prentice-Hall, 1998.

[15] J. Moy, "Multicast Extensions to OSPF", RFC1584, March 1994.

[16] A. Adams, J. Nicholas and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", Internet Draft - work in progress, February 2003.

[17] A. Ballardie, "Core Based Trees (CBT version 2) Multicast Routing", RFC2189, September 1997.

[18] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei, "An Architecture for Wide-Area Multicast Routing", Proceedings of ACM SIGCOMM '94, pp.126-135, August 1994.

[19] B. Fenner, M. Handley, H. Holbrook and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", Internet Draft - work in progress, March 2003.

[20] L. Wei and D. Estrin, "A Comparison of Multicast Trees And Algorithms", Technical Report USC-CS-93-560, University of Southern California, September 1993.

[21] T. Bates, R. Chandra, D. Katz and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC2283, February 1998.

[22] D. Kim, D. Meyer, H. Kilmer and D. Farinacci, "Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)", RFC3446, January 2003.

[23] H. Holbrook and B. Cain, "Source-Specific Multicast for IP", Internet Draft - work in progress, November 2001.

[24] W. Fenner, "Internet Group Management Protocol, Version 2", RFC2236, November 1997.

[25] B. Cain et al., "Internet Group Management Protocol, Version 3", RFC3376, May 2002.

[26] S. Deering, W. Fenner and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC2710, October 1999.

[27] R. Vida et al., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", Internet Draft - work in progress, November 2002.

[28] D. Thaler, B. Fenner and B. Quinn, "Socket Interface Extensions for Multicast Source Filters", Internet Draft - work in progress, June 2003.

[29] H. Asaeda and S. Suzuki, "MLDv2 Protocol Design, Implementation and Evaluation for Source-Specific Multicast over IPv6", Proceedings of SAINT 2003 Workshops, pp.244-249, January 2003.

[30] D. Thaler, D. Estrin and D. Meyer, "Border Gateway Multicast Protocol (BGMP): Protocol Specification", Internet Draft - work in progress, November 2000.

[31] P. Radoslavov, D. Estrin, R. Govindan, M. Handley, S. Kumar and D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol", RFC2909, September 2000.

[32] R. Perlman, C. Y. Lee, A. Ballardie, J. Crowcroft, Z. Wang, T. Maufer, C. Diot, J. Thoo and M. Green, "Simple Multicast: A Design for Simple, Low-Overhead Multicast", Internet Draft - work in progress, October 1999.

[33] B. Fenner and D. Meyer, "Multicast Source Discovery Protocol (MSDP)", RFC3618, October 2003.

[34] B. Fenner, M. Handley, R. Kermode and D. Thaler, "Bootstrap Router (BSR) Mechanism for PIM Sparse Mode", Internet Draft - work in progress, February 2003.

[35] P. Rajvaidya, K. Ramachandran and K. Almeroth, "Detection and Deflection of DoS Attacks Against the Multicast Source Discovery Protocol", Appeared in Proceedings of IEEE INFOCOM 2003.

[36] "Sapphire Worm", <http://www.nmsl.cs.ucsb.edu/mantra/ries/sapphire>

[37] "Multicast Security (msec) Charter", <http://www.ietf.org/html.charters/msec-charter.html>

[38] H. Asaeda and V. Roca, "Consideration of Multicast Channel Announcement Architecture", INRIA Research Report, RR-4762, March 2003.

[39] M. Handley and V. Jacobson, "SDP: Session Description Protocol", RFC2327, April 1998.

[40] M. Handley, C. Perkins and E. Whelan, "Session Announcement Protocol", RFC2974, October 2000.

Contents

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399