Challenges of Integrating ASM and SSM IP Multicast Protocol Architectures

Kevin C. Almeroth[1], Supratik Bhattacharyya[2], and Christophe Diot[2]

[1] Department of Computer Science
University of California
Santa Barbara, CA 93106-5110
almeroth@cs.ucsb.edu
[2] Sprint Advanced Technology Labs
One Adrian Court
Burlingame, CA 94010
{supratik,cdiot}@sprintlabs.com

Abstract. The Source Specific Multicast (SSM) service model and protocol architecture have recently been proposed as an alternative to the currently deployed Any Source Multicast (ASM) service. SSM attempts to solve many of the deployment problems of ASM including protocol complexity, inter-domain scalability, and security weaknesses. However, the SSM protocol architecture is not radically different from that of ASM. This has created opportunities for integrating it into the currently deployed ASM infrastructure. In this paper, we first describe the ASM and SSM service models and associated protocol architectures, highlighting the relative merits and demerits of each. We then examine the network infrastructure needed to support both of them. Our conclusion is that integration is relatively straightforward in most cases; however there is one case—supporting ASM service over an SSM-only protocol architecture—for which it is difficult to design elegant solutions for an integrated SSM/ASM infrastructure.

1 Introduction

The original IP multicast service model was developed with the goal of creating an interface similar to that of best-effort unicast traffic[1]. For transmitters, the goal was to provide scalable transmission by allowing sources to simply transmit without having to register with any group manager or having to perform connection setup or group management functions. The application programming interface was similar to that for UDP packet transmission— an application would simply have to open a socket to a destination and begin transmitting. For receivers, the goal was to provide a way to join a group and then receive all packets sent by all transmitters to the group. In this service model, each multicast host group was identified by a class-D IP address so that an end-host could participate in a multicast session without having to know about the identities of other participating end-hosts. This eventually led to a a triumvirate of protocols to build multicast trees and forward data along them: a tree construction protocol

(the most widely deployed of which is) called Protocol Independent Multicast–Sparse Mode (PIM-SM), the multicast equivalent of the Border Gateway Protocol (BGP) for advertising reverse paths towards sources called the Multiprotocol Border Gateway Protocol (MBGP), and a protocol for disseminating information about sources called the Multicast Source Discovery Protocol (MSDP)[2]. In addition, the Internet Group Management Protocol (IGMP) was designed for end-hosts to dynamically join and leave multicast groups.

The wide-scale commercial deployment of this service model and protocol architecture has run into significant barriers [3]. Many of these barriers are rooted in the problem that building efficient multicast trees for dynamic groups of receivers is a non-trivial problem. As a result, the existing set of protocols is fairly complex and the learning curve is quite steep. Furthermore the "any-to-any" design philosophy of the current Any Source Multicast (ASM) service model is not suitable for commercial services. Most applications today need tighter control over who can transmit data to a set of receivers.

By trying to improve both the efficiency and reduce the complexity of current multicast protocols, the goal is to reduce the barriers to deployment. However, developing and deploying yet another set of protocols creates the additional burden of yet another round of modifications to the existing infrastructure. This may itself become an impediment to deployment efforts. Therefore, it is important to to identify the technical problems that needs to be solved before designing and deploying a new protocol architecture. For the current ASM architecture and service model, the problems include:

1. attacks against multicast groups by unauthorized transmitters
2. deployment complexity
3. problems of allocating scarce global class-D IP addresses
4. lack of inter-domain scalability
5. single point of failure problems

A new service model, Source Specific Multicast (SSM), and an associated protocol architecture have been proposed as a solution to the above problems and is beginning to be deployed[4]. In the SSM service model a receiving host explicitly specifies the address of the source it wants to receive from, in addition to specifying a class-D multicast group address.

From a deployment standpoint, the fundamental advantage of SSM is that protocol complexity can be removed from the network layer and implemented more easily, simply, and cheaply at the application layer. The tradeoff, which itself has both advantages and disadvantages, has the potential to fundamentally change how IP multicast service is provided in the Internet. Moreover there is significant overlap in the protocol architectures for ASM and SSM, thereby facilitating the rapid integration of SSM support in networks that already support ASM.

The key difference between the two service models lies in the way a receiving host joins a multicast group. As a result, there are a number of questions that arise about supporting the two. Should the two models exist simultaneously?

Should the existence of two models be made visible to the user? What part of the multicast infrastructure should be responsible for dealing with interoperability between the two service models? Answering these questions is a critical step in providing seamless interoperability between ASM and SSM.

In this paper, we describe the differences between the ASM and SSM protocol architectures and service models. We then study the challenges of deploying an integrated ASM/SSM infrastructure. We believe that SSM can solve many of the technical problems without making the existing infrastructure obsolete. But, technical challenges exist in seamlessly integrating the two without creating "black holes". Fortunately, we find that in most cases, the problems faced in integrating the two architectures are neither many nor insurmountable; they simply need to be identified and then the appropriate solutions implemented.

The remainder of the paper is organized as follows. In Section 2 we describe the multicast service models. Section 3 describes the ASM and SSM protocol architectures. Section 4 discusses the challenges in integrating ASM and SSM. The paper is concluded in Section 5.

2 IP Multicast Service Models

In order to understand the implication of offering different types of IP multicast services, we first need to make a distinction between a *protocol architecture* and a *service model*. A multicast protocol architecture refers to a set of protocols that together allow end-hosts to join/leave multicast sessions, and allows routers to communicate with each other to build and forward data along inter-domain forwarding trees. An IP multicast service model refers to the semantics of the multicast service that a network provides an end-user with. It is embodied in the set of capabilities available to an end-user at the application interface level, and is supported by a network protocol architecture. Any multicast service model is realized through:

- an application programming interface (API) used by applications to communicate with the host operating system.
- host operating system support for the API.
- protocol(s) used by the host operating system to communicate with the leaf network routers (referred to as designated routers or edge-routers).
- protocol(s) for building inter-domain multicast trees and for forwarding data along these trees.

With multiple service models and protocol architectures, the challenge therefore lies in bridging the gap between the protocol architecture deployed in the network and the service model expected by end-user applications.

Currently, there are two main IP multicast service models plus a third deriving from a combination of the two. The details of the protocol architecture supporting each of them is described in Section 3.

(a) **Any-Source Multicast (ASM)**: This is the traditional IP multicast service model defined in RFC 1112[1]. An IP datagram is transmitted to a

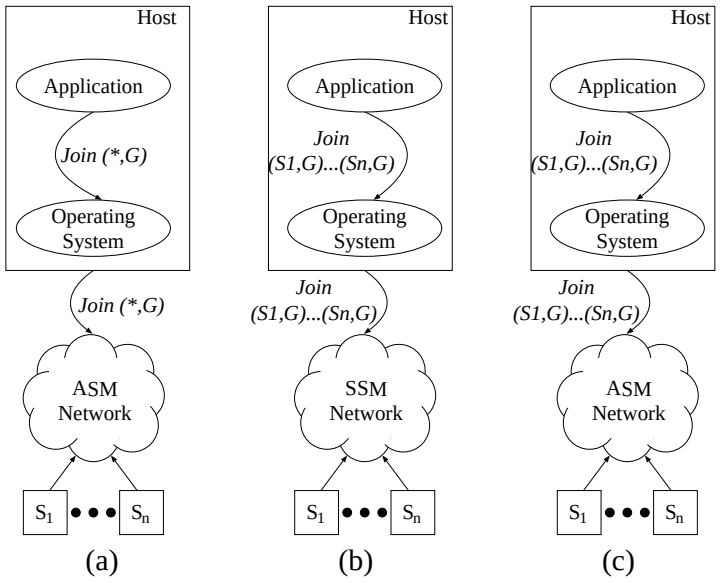

Fig. 1. Three choices for the IP multicast service model.

"host group", a set of zero or more hosts identified by a single IP destination address (224.0.0.0 through 239.255.255.255 for IPv4). This model supports one-to-many and and many-to-many multicast communication. Hosts may join and leave the group at any time. There is no restriction on the location or number of receivers, and a source need not be a member of the host group it transmits to. Host-to-network communication support for ASM is provided by the Internet Group Management Protocol (IGMP) version 2. IGMPv2 allows a receiver to specify a class-D group address for the host group it wants to join, but does not allow it to specify the sources that it wants (or does not want) to receive traffic from. This service model is shown in Figure 1(a).

(b) **Source-Specific Multicast (SSM)**: This is the multicast service model defined in [5]. An IP datagram is transmitted by a source S to an SSM address G, and receivers can receive this datagram by subscribing to channel (S,G). SSM is derived from EXPRESS[6] and supports one-to-many multicast. The address range 232/8 has been assigned by IANA[7] for SSM service in IPv4. In IPv6, an address range ($FF2x$ :: and $FF3x$ ::) already exists[8] for SSM services. IGMP version 3, which allows a receiver to specify explicitly the source address, provides host-to-network communication support for SSM. This requires upgrading most host operating systems and edge routers from IGMPv2 to IGMPv3. This also implies that the host operating system's API must now allow applications to specify a source and a group in order to receive multicast traffic. This service model is shown in Figure 1(b).

A variant of the ASM service model is known as the **Source-Filtered Multicast (SFM)** model. In this case, a source transmits IP datagrams to a host group address in the range of 224.0.0.0 to 239.255.255.255. However, each application can now request data sent to a host group G for *only* a specific set of sources, or can request data sent to host group G from *all except* a specific set of sources. In other words, applications can apply "source filtering" to the multicast data being transmitted to a to a given host group. Host-to-network support for source filtering is provided by IGMPv3 for IPv4, and version 2 of the Multicast Listener Discovery (MLD) protocol for IPv6[9].

3 IP Multicast Protocol Architectures

In this section we describe in detail the protocol architectures for supporting the ASM and SSM service models, and the relative merits and demerits of each.

3.1 ASM Protocol Architecture

The current inter-domain multicast architecture is based on the ASM service model. To become a member of a particular group, end-hosts register their membership with querier routers handling multicast group membership functionality using the IGMP version 2 (IGMPv2) protocol[10] for IPv4 or the MLD version 1 (MLDv1) protocol[11] for IPv6. With IGMPv2 and MLDv1, source-filtering capabilities are not available to receivers.

Multicast-capable routers then construct a distribution tree by exchanging messages with each other according to a routing protocol. A number of different protocols exist for building multicast forwarding trees. These protocols differ mainly in the type of delivery tree constructed[1, 12–15]. Of these, the Protocol Independent Multicast Sparse-Mode (PIM-SM) protocol[14] is the most widely deployed in today's public networks. PIM-SM, by default, constructs a single spanning tree rooted at a core Rendezvous Point (RP) for all group members within a domain. Local sources then send their data to this RP which forwards the data down the shared tree to interested local receivers. A receiver joining a host group can only specify interest in the entire group and therefore will receive data from any source sending to this group. Distribution via a shared tree can be effective for certain types of traffic, e.g., where the number of sources is large since forwarding on the shared tree is performed via a single multicast forwarding entry. However, there are many cases (e.g., Internet broadcast streams) where forwarding from a source to a receiver is more efficient via the shortest path. PIM-SM also allows a designated router serving a particular subnet to switch to a source-based shortest path tree for a given source once the source's address is learned from data arriving on the shared tree. This capability provides for distribution of data from local sources to local receivers using a common RP inside a given PIM domain.

It is also possible for RP's to learn about sources in other PIM domains by using the Multicast Source Discovery Protocol (MSDP)[16]. Once an active remote source is identified, an RP can join the shortest path tree to that source and obtain data to forward down the local shared tree on behalf of interested

local receivers. Designated routers for particular subnets can again switch to a source-based shortest path tree for a given remote source once the source's address is learned from data arriving on the shared tree.

The IGMPv2/PIM-SM/MSDP-based inter-domain multicast architecture supporting ASM has been deployed in IPv4 networks. It has been particularly effective for groups where sources are not known in advance; when sources come and go dynamically; or when forwarding on a common shared tree is found to be operationally beneficial.

However, there are several problems hindering the commercial deployment of these protocols. Some of these are inherent in the service model itself, while others are due to the complexity of the protocol architecture:

- **Attacks by unauthorized transmitters**: In the ASM service model, a receiver cannot specify which specific sources it would like to receive data from when it joins a given group. A receiver is forwarded data sent by *all* group sources. This lack of access control can be exploited by malicious transmitters to disrupt data transmission from authorized transmitters.
- **Deployment complexity**: The ASM protocol architecture is complex and difficult to manage and debug. Most of the complexity arises from the RP-based infrastructure needed to support shared trees, and from the MSDP protocol used to discover sources across multiple domains. These challenges often make network operators reluctant to enable IP multicast capabilities in their networks, even though most of today's routers support the IGMP/PIM-SM/MSDP protocol suite.
- **Address allocation**: This is one of the biggest challenges in deploying an inter-domain multicast infrastructure supporting ASM. The current multicast architecture does not provide an adequate solution to prevent address collisions among multiple applications. As a result two entirely different multicast sessions may pick the same class-D address for their multicast groups and interfere with each other's transmission. The problem is more serious for IPv4 than IPv6 since the total number of multicast addresses is smaller. A static address allocation scheme, GLOP[17], has been proposed as an interim solution for IPv4. GLOP addresses are allocated per registered Autonomous System (AS). However, the number of addresses per AS is inadequate when the number of sessions exceeds an AS's allocation. Proposed longer-term solutions such as the Multicast Address Allocation Architecture (MAAA)[18] are generally perceived as being too complex (with respect to the dynamic nature of multicast address allocation) for widespread deployment. Another long term solution, the unicast-prefix-based multicast architecture of IPv6[8] expands on the GLOP approach; simplifies the multicast address allocation solution; and incorporates support for source-specific multicast addresses.
- **Inter-domain scalability**: MSDP has always been something of an ugly solution. The protocol has weaknesses in terms of security and scalability. For security, it is susceptible to denial-of-service attacks by domains sending out a flood of source announcements. For scalability, MSDP is not well designed to handle large numbers of sources. The primary reason is because the

source announcements were designed to be periodically flooded throughout the topology *and* to carry data. As the number of sources in the Internet increases, MSDP will generate greater amounts of control traffic.
- **Single point of failure**: When multicast data distribution takes place over a shared tree via a core network node (RP in the case of PIM-SM), failure of the core can lead to complete breakdown of multicast communication[1]. In the ASM protocol architecture, a receiver is always grafted on to an RP-based shared tree when it first joins a multicast group. This reliance on the shared-tree infrastructure makes the ASM protocol architecture fundamentally less robust.

3.2 SSM Protocol Architecture

As mentioned before, Source Specific Multicast (SSM) defines a service model for a "channel" identified by an (S,G) pair, where S is a source address and G is an SSM address. This model can be realized by a protocol architecture, where packet forwarding is restricted to shortest path trees rooted at specific sources, and channel subscriptions are described using a group management protocol such as IGMPv3 or MLDv2.

The SSM service model alleviates all of the deployment problems described earlier:

- The distribution tree for an SSM channel (S,G) is always rooted at the source S. Thus there is no need for a shared tree infrastructure. In terms of the IGMPv2/PIM-SM/MSDP architecture, this implies that neither the RP-based shared tree infrastructure of PIM-SM nor the MSDP protocol is required. Hence the protocol architecture for SSM is significantly less complex than that for ASM, making it easy to deploy. In addition, SSM is not vulnerable to RP failures or denial-of-service attacks on RP(s).
- SSM provides an elegant solution to the access control problem. Only a single source S can transmit to a channel (S,G) where G is an SSM address. This makes it significantly more difficult to spam an SSM channel than an ASM host group. In addition, data from unrequested sources need not be forwarded by the network, which prevents unnecessary consumption of network resources[19].
- SSM defines channels on a per-source basis; hence SSM addresses are "local" to each source. This averts the problem of global allocation of SSM addresses, and makes each source independently responsible for resolving address collisions for the various channels that it creates.
- It is widely held that point-to-multipoint applications such as Internet TV will dominate the Internet multicast application space in the near future. The SSM model is ideally suited for such applications. Thus the deployment of SSM will provide tremendous impetus to inter-domain Internet multicasting and will pave the way for a more general multipoint-to-multipoint service in the future.

[1] Multiple cores can certainly be used to alleviate this problem, but redundancy comes at the price of extra overhead and complexity.

A protocol architecture for SSM requires the following:

- **Source specific host membership reports**: The host-to-network protocol must allow a host to describe specific sources from which it would like to receive data.
- **Shortest path forwarding**: DR's must be capable of recognizing receiver-initiated, source-specific host reports and initiating (S,G) joins directly to the source.
- **Elimination of shared tree forwarding**: In order to achieve global effectiveness of SSM, all networks must agree to restrict data forwarding to source trees (i.e., prevent shared tree forwarding) for SSM addresses. The address range 232/8 has been allocated by IANA for deploying source-specific IPv4 multicast (SSM) services. In this range, SSM is the sole service model. For IPv6, a source-specific multicast address range has been defined[8], as a special case of unicast prefix-based multicast addresses.

We now discuss the framework elements in detail:

- **Channel discovery**: In the case of ASM, receivers need to know only the group address for a specific session. In the IGMPv2/PIM-SM/MSDP architecture, designated routers discover an active source via the RP infrastructure and MSDP, and then graft themselves to the multicast forwarding tree rooted at that source. In the case of SSM, an application on an end-host must know both the SSM address G and the source address S before subscribing to a channel. Thus the function of channel discovery becomes the responsibility of applications. This information can be made available in a number of ways, including via web pages, sessions announcement applications, etc.
- **SSM-aware applications**: The advertisement for an SSM session must include a source address as well as a group address. Also, applications subscribing to an SSM channel must be capable of specifying a source address in addition to an group address. In other words, applications must be *SSM-aware*. Specific API requirements are identified in [20].
- **Address Allocation**: For IPv4, the address range of 232/8 has been assigned by IANA for SSM. Sessions expecting SSM functionality must allocate addresses from the 232/8 range. To ensure global SSM functionality in 232/8, including in networks where edge routers run IGMPv2 (i.e., do no support source filtering), operational policies are being proposed[4] which prevent data sent to 232/8 from being delivered via shared trees.
  Note that it is possible to achieve the benefit of direct and immediate (S,G) joins in response to IGMPv3 reports in other ranges than 232/8.However, non-SSM address ranges allow for concurrent use of both the ASM and SSM service models. Therefore, while we can achieve the PIM join efficiency in the non-SSM address range with IGMPv3, it is not possible to prevent the creation of shared trees or shared tree data delivery, and thus cannot provide for certain types of access control or assume per-source unrestricted address use as with the SSM address range.

In the case of IPv6, [8] has defined an extension to the addressing architecture to allow for unicast prefix-based multicast addresses. In this case, bytes 0-3 (starting from the least significant byte) of the IP address is used to specify a multicast group id, bytes $4-11$ is be used to specify a unicast address prefix (of up to 64 bits) that owns this multicast group id, and byte 12 is used to specify the length of the prefix. A source-specific multicast address can be specified by setting both the prefix length field and the prefix field to zero. Thus IPv6 allows for 2^{32} SSM addresses per scope for every source, while IPv4 allows 2^{24} addresses per source.

– **Host-to-network communication**: The currently deployed version of IGMP (IGMPv2) allows end-hosts to register their interest in a multicast group by specifying a class-D IP address for IPv4. However in order to implement the SSM service model, an end-host must specify a source's unicast address as well as an SSM address. This capability is provided by IGMP version 3 (IGMPv3). IGMPv3 supports "source filtering", i.e., the ability of an end-system to express interest in receiving data packets from only a set of *specific* sources, or from *all except* a set of specific sources. Thus IGMPv3 provides a superset of the capabilities required to realize the SSM model. Hence an upgrade from IGMPv2 to IGMPv3 is an essential change for implementing SSM.
IGMPv3 requires the API to provide the following operation (or its logical equivalent)[21]:

$$IPMulticastListen(Socket, IF, G, filter-mode, source-list)$$

As explained in the IGMPv3 specification[21], the above IPMulticastListen() operation subsumes the group-specific join and leave operations of IGMPv2. Performing (S,G)-specific joins and leaves is also trivial. A join operation is equivalent to:

$$IPMulticastListen(Socket, IF, G, INCLUDE, S)$$

and a leave operation is equivalent to

$$IPMulticastListen(Socket, IF, G, EXCLUDE, S)$$

There are a number of backward compatibility issues between IGMP versions 2 and 3 which have to be addressed. There are also some additional requirements for using IGMPv3 for the SSM address range. A detailed discussion of these issues is provided in [22].
The Multicast Listener Discovery (MLD) protocol is used by an IPv6 router to discover the presence of multicast listeners on its directly attached links, and to discover the multicast addresses that are of interest to those neighboring nodes. Version 1 of MLD[11] is derived from IGMPv2 and allows a multicast listener to specify the multicast group(s) that it is interested in. Version 2 of MLD[9] is derived from, and provides the same support for source-filtering as, IGMPv3.

– **PIM-SM modifications for SSM**: PIM-SM[14] itself supports two types of trees, a shared tree rooted at a core (RP), and a source-based shortest path tree. Thus PIM-SM already supports source-based trees; however, PIM-SM is not designed to allow a router to choose between a shared tree and a source-based tree. In fact, a receiver always joins a PIM shared tree to start with, and may later be switched to a per-source tree by its adjacent edge router.

  A key to implementing SSM is to eliminate the need for starting with a shared tree and then switching to a source-specific tree. This involves several changes to PIM-SM as described in [14]. The resulting PIM functionality is referred to as PIM-SSM. The most important changes to PIM-SM with respect to SSM are as follows:

  - When a DR receives an (S,G) join request with the address G it must initiate a (S,G) join and *never* a (*,G) join.
  - Backbone routers (i.e. routers that do not have directly attached hosts) must be capable of receiving (S,G) joins and forwarding them based on correct RPF information. In addition, they must not propagate (*,G) joins for group addresses in the SSM address range.
  - Rendezvous Points (RPs) must not accept PIM Register messages or (*,G) join messages.

In summary, the ASM service model and protocol architecture suffer from a number of serious deployment problem[3]. The SSM service model addresses many of the needs of today's commercial multicast applications. Also, the associated protocol architecture is simpler, and easy to deploy in networks that already supports ASM. The challenge then becomes integrating the two.

4 Integrating ASM and SSM

In this section, we examine interoperability issues between ASM and SSM. From our discussion so far, it is clear that there is significant overlap between the two protocol architectures. Therefore, it is possible to integrate the two. The task is to then investigate the interoperability issue from a host perspective, i.e. if a host is connected to a network, what does it have to do to properly utilize whatever multicast service is present?

Given that the two service models and two protocol architectures form a set of four combinations, the challenge is to identify any problems in providing a seamless multicast service—including both intra- and inter-domain operation. As we have discovered, most of these scenarios are trivially workable. Of those that remain, one requires minor changes to existing protocols, and one is quite challenging. Our goal is to (1) identify what the interoperability problems are, (2) identify solutions to these problems, and (3) understand the relative complexities of deploying these solutions.

In the next section we describe the four combinations of service models and protocol architectures. Following the overview, we focus specifically on solutions for the most difficult case.

4.1 Service Model and Protocol Architecture Combinations

There are four combinations of service models and protocol architectures. These are shown in Figure 2. The key challenge naturally occurs because the host does not know how the network is configured. This is actually a reasonable abstraction. The host should simply join a multicast group. If IGMPv3 is available and the application knows who the source is, this information should be passed to the network. If this information is not available or if IGMPv3 is not supported, the network should still respond in a predictable manner.

The challenge of deploying a multicast service is to provide correct operation for all kinds of multicast no matter if (1) the host is limited to only IGMPv2 and/or (2) the network is limited only to SSM support. In fact, the ability to interoperate with ASM was one of the key requirements for SSM[5]. This requirement was critical because the development of any completely new protocol architecture would mean that multicast deployment efforts would have to start completely over. Furthermore, given the near infinite lifetime of legacy architectures, ASM would in all likelihood continue to exist and need to be supported. Therefore, a new protocol architecture that did not integrate with ASM would not reduce complexity but rather increase it. Therefore, SSM was designed to interoperate with ASM. However, because SSM implements a subset of ASM functionality, there needs to be additional work to properly integrate the two.

The challenge with integrating ASM and SSM is how to handle the discontinuities between the two. If the two are not integrated properly, multicast does not work. Even more problematic is that there is no feedback from any part of the network or host that says multicast is not working. Cases when this kind of behavior occurs are called "black holes". Black holes occur when both the network and the host are operating correctly, but no multicast flows because there is a disconnect between the service model and the protocol architecture. For example, the network only allows hosts to specify an explicit list of sources but the host sends a (*,G) join. The network cannot process this kind of message and ignores it. There is no feedback to the host that the join message was not properly handled.

Before describing the specific black hole scenarios, we first describe the scenarios that are more straightforward. Figure 2 shows all four combinations. They are:

- **ASM service model and ASM network**: The upper-left scenario is the service model and protocol architecture that has been running in the Internet since 1997[2]. Theoretically, there are no black holes in this combination, though in practice, problems often occur[23, 24].
- **SSM service model and ASM network**: The upper-right scenario is essentially the same protocol architecture that has been running since 1997, but with IGMPv3 support. Through IGMPv3, users are given the ability to specify a subset of all group receivers, thereby refining the granularity of join and leave messages to more than just one choice for all sources. In the section listing the service models, this combination is also called Source-Filtered

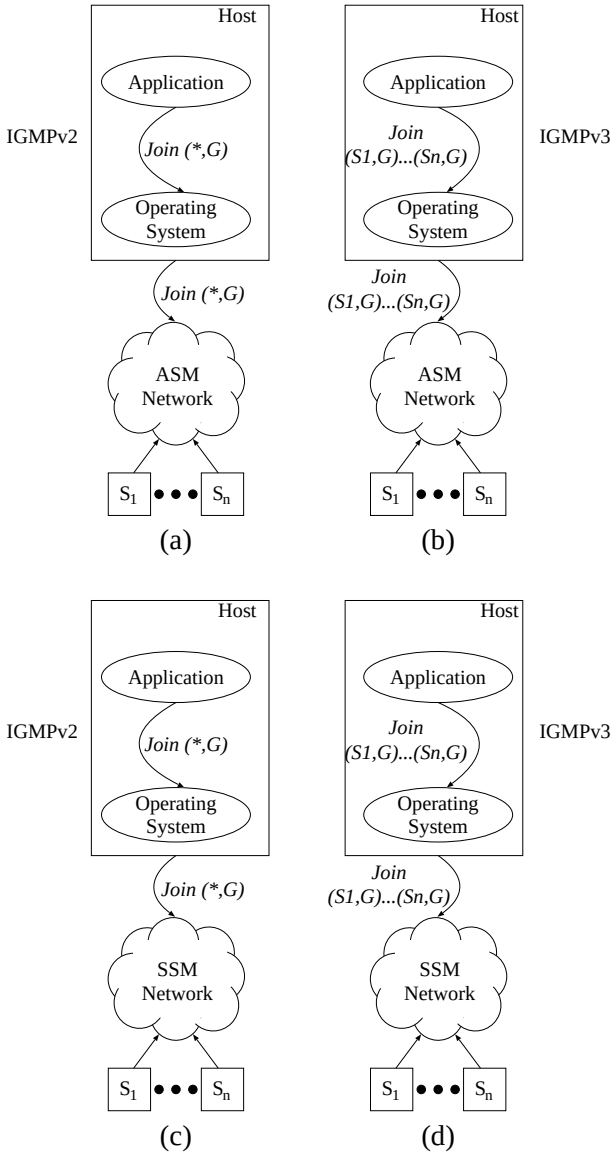

Fig. 2. The four combinations of service models and protocol architectures.

Multicast (SFM). SSM is supported in this combination in the address range 232/8[25]. Theoretically, there are no black holes in this combination.

- **ASM service model and SSM network**: The lower-left scenario is the most problematic of the four combinations. From the network point-of-view, the service provider has opted to only provide support for SSM. But for any of a variety of reasons, a host either chooses to send or is only capable of sending (*,G) join messages. The dilemma is how to solve this problem. Several possible solutions are discussed in the next section.
- **SSM service model and SSM network**: The lower-right scenario is more straightforward than the previous case, but there is still one problem. Because the multicast address space is divided into SSM and non-SSM ranges, the straightforward behavior is when the group address is in the SSM range (232/8)[25]. Uncertainty occurs when handling (S,G) joins for the non-SSM range. There are two considerations to understand:
  - One of the dependencies here is whether the network is providing SSM support only in the 232/8 address range or whether it has been extended to cover the entire multicast address range (224/4). If the choice is only to support the 232/8 range, how should the network, host, and application handle (S,G) joins for addresses outside this range? The currently accepted practice seems to be to not allow SSM support outside of the 232/8 range and embed these semantics in the operating system.
  - If the network provider instead chooses to provide SSM support for the entire address range, a problem is created for sources. Sources transmitting on a non-SSM address will *not* have their existence announced throughout the inter-domain infrastructure. To understand why this situation occurs, consider the behavior in an ASM domain. When a source sends its first packet, the network encapsulates it and sends it to the RP. Since MSDP runs in the RP, an SA message is generated and flooded on the MSDP peering topology. Because the domain chooses to run SSM for the entire address space, there is no RP, no initial packet encapsulation, and no MSDP peer. Receivers in ASM domains will never discover the existence of this particular source. Again, the current accepted practice seems not to provide SSM-style service for addresses outside of the 232/8 range.

There are a number of solutions to solving the problem of an ASM service model running in an SSM network. These solutions are discussed in the next section.

4.2 Handling ASM Hosts in an SSM Infrastructure

For hosts that can only speak IGMPv2, operation in an SSM-only network is difficult. Even for hosts that do speak IGMPv3 there are inter-domain consistency problems if SSM behavior is enforced beyond the 232/8 address range. The first step to solving these interoperability problems is to understand more clearly what the problems are. At a minimum, applications need deterministic, predictable behavior. Ideally, applications should be able to maintain some level

of abstraction from the type of multicast service. However, implementing this in the Internet looks to be quite difficult. The problem is that because there are different semantics tied to the multicast address space (232/8 vs. the rest of 224/4), different behavior is expected depending on the address used. Therefore, hosts need to have some awareness of these semantics and what the network supports. While a host does not need a complete understanding of what the network protocol architecture is, it needs to know whether its join messages are going to be processed properly.

One of the fundamental problems is that a host sending a (*,G) join into an SSM network will have the join message ignored. Therefore, some additional action must be taken by the network if IGMPv2 hosts are to be supported. The first two solutions attempt to resolve the (*,G) into a set of sources. Two choices of this type are shown in the top half of Figure 3 and described below.

- **Run an MSDP peer**: An SSM-only domain can run MSDP. Leaf routers would query the MSDP cache for source information. This solution is shown in Figure 3(a). A straightforward implementation based on existing protocols would be to run an MSDP peer at the domain boundary and then use a new protocol for communicating between the peer and the leaf routers. The obvious disadvantage of this solution is that it appears to have almost as much complexity as ASM. One savings is in not having to run an RP. A second savings is being able to run the source discovery protocol at the application layer and not embed it in the network layer. By implementing an MSDP peer as an application and then adding some basic functionality just to the leaf routers, this solution can make sense.
- **URL Rendezvous Directory (URD)**: Similar to running an MSDP peer, URD involves translating network-layer complexity into application-layer management overhead. The idea is that a host will use the web to gather information about the multicast group and will then initiate the group join based on this information. This solution is shown in Figure 3(b). When a user clicks on a link, the response is to send both the source and group information back to the client as an HTTP re-direct to the URD port (Port 659). For example, the returned link might look like the following:

  ```
  http://content-source.com:659/source-address,group-address/
  ```

  The router is intercepting traffic sent to port 659, and in combination with the IGMPv2 join message sent by the user's application, the router will be able to issue an (S,G) join to the source. The goal of URD is to be able to put all of the additional functionality necessary for SSM at the content source site and in the leaf routers. This objective is achieved because no additional modifications are required in the application or the host operating system. The application responds normally to the HTTP re-direct and the operating system issues an IGMPv2 (*,G) join. There is even support to avoid black holes in the case where a content site uses URD but the leaf router does not support it. What happens is that the router does not intercept the URL

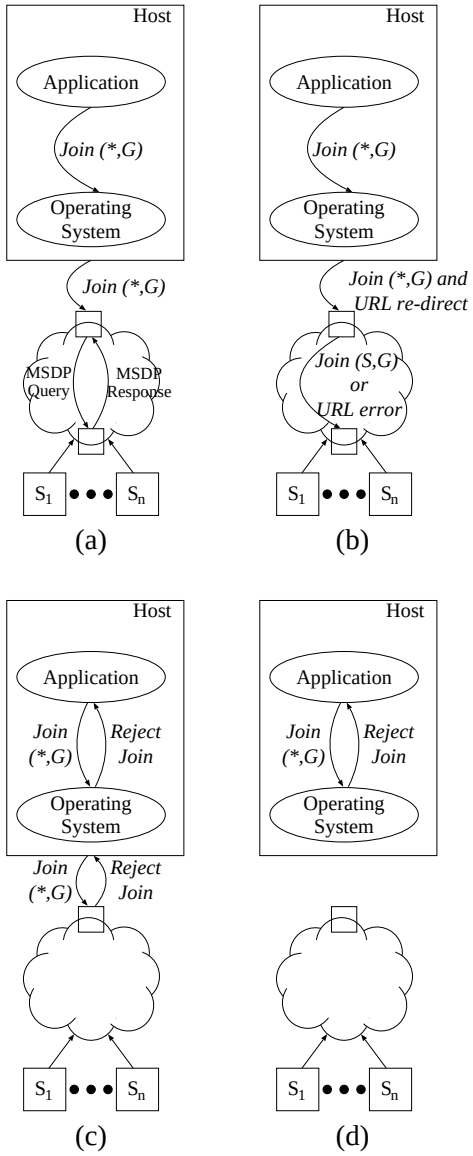

Fig. 3. Four possible solutions for ASM hosts in an SSM infrastructure.

re-direct and so it reaches the content source. If this re-direct appears, the content source knows the leaf router did not intercept the re-direct and can inform the user via a web page that the join did not work. Of course, the major disadvantage is that this requires routers to snoop on port 659 and intercept traffic—not an acceptable requirement for most network providers.

Given that these two solutions require application-layer mechanisms to replace the functionality of PIM RPs and MSDP, neither solution is particularly elegant. A better solution would be to simply limit what the user can and cannot do. For example, the idea would be to prohibit joins to non-232/8 group addresses in SSM-only networks. Black holes would be avoided by letting the user/application/host know that the unsupported join actions had failed. Details on two particular solutions are shown at the bottom of Figure 3 and described below.

- **IGMPv3 with reject capability**: The idea is to modify IGMPv3 to create a more robust control path. The solution is to allow the leaf router to "reject" an IGMP join. This solution is shown in Figure 3(c). A number of possible reasons could exist for rejecting a join. The obvious case is when a (*,G) join is set for an address configured only to allow (S,G) joins. Another example might be joins sent to a group that has been listed in a site-controlled reject list. The IGMP reject message could include a return code providing a reason for the rejection. This solution would require either an addition to IGMPv3 or a new version. While this solution seems quite reasonable, it requires revising IGMPv3 which creates yet another deployment delay.
- **Host discovery of network capability**: Hosts could be given the capability to discover for themselves how the network is configured. This solution is shown in Figure 3(d). In this way, hosts could determine what group addresses require a specific set of sources and what group addresses allow the use of the "*" in join messages. This discovery provides the host with enough information to reject an application's join request. Like many of the other solutions, the problem is host behavior needs to be modified.

Either of the last two solutions is the most reasonable. But the changes require another round of deployment. The practical solution is for all routers, operating systems, and applications to assume that SSM runs only in the 232/8 range, and that some networks might only support SSM. Therefore, any (S,G) join should be expected to work, but any (*,G) join should be suspect. The simple policy should be that if an application has access to the set of one or more group sources, it should use them. Otherwise, the possibility exists that (*,G) joins will not be successful. This creates a certain amount of non-determinism but seems easy to characterize: IGMPv2 joins might not work. The incentive is to upgrade to IGMPv3 as quickly as possible.

5 Conclusions

In this paper, we have considered the integration of the two service models for IP multicast: Any Source Multicast (ASM) and Source Specific Multicast (SSM).

We have described the protocol architecture for each and discussed their advantages and disadvantages. ASM is the traditional service model; however, it suffers from a number of serious problem from a commercial deployment standpoint. SSM solves most of these which makes it suitable for rapid deployment. The important advantage of SSM is the significant overlap of its protocol architecture with that of SSM. We have explored the interoperability of these two service models, and found that in most cases, the challenges are not insurmountable. In the near term, we expect these two service models to co-exist in a unified IP multicast infrastructure.

From a broader perspective, successfully integrating ASM and SSM should have a positive impact on the use and deployment of multicast. IP multicast has long suffered from the "chicken and egg" problem. The lack of popular applications has given very little incentive to ISPs to enable multicast in their networks. Furthermore, ASM has been plagued by a number of deployment problems. On the other hand, the lack of widespread deployment has resulted in very limited interest in developing new applications. At the same time, the popularity of application-layer multicast has further slowed down deployment of IP multicast. It is hoped that SSM will spur the deployment and use of IP multicast by virtue of its simplicity, ease of deployment, and its ability to be integrated into the existing infrastructure. While SSM is ideally suited for point-to-multipoint, multi-peer applications such multi-party games can easily be supported by building relays at the application level over an SSM-capable network. Such an approach represents an attractive compromise between the efficiency of network-level multicast and the ease of manageability of application-level multicast.

References

1. S. Deering, "Host extensions for IP multicasting." Internet Engineering Task Force (IETF), RFC 1112, August 1989.
2. K. Almeroth, "The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment," *IEEE Network*, January/February 2000.
3. C. Diot, B. Lyles, B. Levine, and H. Kassem, "Requirements for the definition of new IP-multicast services," *IEEE Network*, January/February 2000.
4. S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockwell, J. Meylor, D. Meyer, G. Shepherd, and B. Haberman, "An overview of source-specific ip multicast(ssm) deployment." Internet Engineering Task Force (IETF), draft-ietf-bhattach-ssm-*.txt, May 2001.
5. H. Holbrook and B. Cain, "Source-specific multicast for IP." Internet Engineering Task Force (IETF), draft-holbrook-ssm-arch-*.txt, March 2001.
6. H. Holbrook and D. Cheriton, "IP multicast channels: EXPRESS support for large-scale single-source applications," in *ACM Sigcomm*, (Cambridge, Massachusetts, USA), August 1999.
7. Z. Albanna, K. Almeroth, D. Meyer, and M. Schipper, "IANA guidelines for IPv4 multicast address assignments." Internet Engineering Task Force (IETF), draft-ietf-mboned-iana-ipv4-mcast-guidelines-*.txt, April 2001.
8. B. Haberman and D. Thaler, "Unicast-prefix-based IPv6 multicast addresses." Internet Engineering Task Force (IETF), draft-ietf-ipngwg-uni-based-mcast-*.txt, January 2001.
9. R. Vida, L. Costa, S. Fdida, S. Deering, B. Fenner, I. Kouvelas, and B. Haberman, "Multicast listener discovery version 2 (MLDv2) for ipv6." Internet Engineering Task Force (IETF), draft-vida-mld-v2-*.txt, February 2001.

10. W. Fenner, "Internet group management protocol, version 2." Internet Engineering Task Force (IETF), RFC 2236, November 1997.

11. S. Deering, B. Fenner, and B. Haberman, "Multicast listener discovery (MLD) for IPv6." Internet Engineering Task Force (IETF), RFC 2710, October 1999.

12. S. Deering, D. Estrin, D. Farinacci, V. Jacobson, G. Liu, and L. Wei, "PIM architecture for wide-area multicast routing," *IEEE/ACM Transactions on Networking*, pp. 153–162, Apr 1996.

13. D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol independent multicast sparse-mode (PIM-SM): Protocol specification." Internet Engineering Task Force (IETF), RFC 2362, June 1998.

14. B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol independent multicast - sparse mode (PIM-SM): Protocol specification (revised)." Internet Engineering Task Force (IETF), draft-ietf-pim-sm-v2-new-*.txt, March 2001.

15. S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, D. Meyer, and L. Wei, "Protocol independent multicast version 2 dense mode specification." Internet Engineering Task Force (IETF), draft-ietf-pim-v2-dm-*.txt, June 1999.

16. D. Meyer and B. Fenner, "Multicast source discovery protocol (MSDP)." Internet Engineering Task Force (IETF), draft-mboned-msdp-spec-*.txt, May 2001.

17. D. Meyer and P. Lothberg, "GLOP addressing in 233/8." Internet Engineering Task Force (IETF), RFC 2770, February 2000.

18. M. Handley, D. Thaler, and D. Estrin, "The internet multicast address allocation architecture." Internet Engineering Task Force (IETF), draft-ietf-malloc-arch-*.txt, December 1997.

19. B. Fenner, H. Holbrook, and I. Kouvelas, "Multicast source notification of interest protocol (msnip)." Internet Engineering Task Force (IETF), draft-ietf-idmr-msnip-*.txt, February 2001.

20. D. Thaler, B. Fenner, and B. Quinn, "Socket interface extensions for multicast source filters." Internet Engineering Task Force (IETF), draft-ietf-idmr-msf-api-*.txt, June 2000.

21. B. Cain, S. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan, "Internet group management protocol, version 3." Internet Engineering Task Force (IETF), draft-ietf-idmr-igmp-v3-*.txt, March 2001.

22. H. Holbrook and B. Cain, "Using IGMPv3 for source-specific multicast." Internet Engineering Task Force (IETF), draft-holbrook-idmr-igmpv3-ssm-*.txt, March 2000.

23. K. Sarac and K. Almeroth, "Monitoring reachability in the global multicast infrastructure," in *International Conference on Network Protocols (ICNP)*, (Osaka, JAPAN), November 2000.

24. P. Rajvaidya and K. Almeroth, "A router-based technique for monitoring the next-generation of internet multicast protocols," in *International Conference on Parallel Processing*, (Valencia, Spain), September 2001.

25. G. Shepherd, E. Luczycki, and R. Rockell, "Source-specific protocol independent multicast in 232/8." Internet Engineering Task Force (IETF), draft-ietf-mboned-ssm232-*.txt, April 2001.