Deployment Issues for the IP Multicast Service and Architecture

Christophe Diot, Sprint Advanced Technology Labs
Brian Neil Levine, University of Massachusetts
Bryan Lyles, Sprint Advanced Technology Labs
Hassan Kassem, SprintLink
Doug Balensiefen, Sprint

Abstract

IP multicast offers the scalable point-to-multipoint delivery necessary for using group communication applications on the Internet. However, the IP multicast service has seen slow commercial deployment by ISPs and carriers. The original service model was designed without a clear understanding of commercial requirements or a robust implementation strategy. The very limited number of applications and the complexity of the architectural design — which we believe is a consequence of the open service model — have deterred widespread deployment as well. We examine the issues that have limited the commercial deployment of IP multicast from the viewpoint of carriers. We analyze where the model fails and what it does not offer, and we discuss requirements for successful deployment of multicast services.

Since its introduction [1], IP multicast has seen slow commercial deployment in the Internet. Although it has been available through the experimental Mbone for a number of years, it is just beginning to see commercial support from carriers, Internet service providers (ISPs), and common operating systems. IP-based networks offer point-to-multipoint and multipoint-to-multipoint best-effort delivery of datagrams by means of the IP multicast service and architecture.[1] The current service model in IP multicast was defined without a commercial service explicitly in mind, which is one possible reason for its slow deployment. Although each of these issues is the subject of current research efforts, the service model and architecture do not efficiently provide or address many features required of a robust commercial implementation of multicast. Some of these issues include:

- Group management, including authorization for group creation, receiver authorization, and sender authorization
- Distributed multicast address allocation
- Security, including protection against attacks on multicast routes and sessions, as well as support for data integrity mechanisms
- Support for network management

Consequently, the current IP-multicast architecture deployed by carriers and ISPs to compensate for these issues is complex and has limited scalability. Trying to generalize and commercialize multicast from the current service model and protocol architecture is difficult, and, in the worst case, adversely impacts the long-term success of multicast.

In this article we examine, from the viewpoint of ISPs and carriers, the current IP multicast service model and the issues that have limited the commercial deployment of IP multicast. We discuss the motivations of ISPs and users for using multicast. We show where the architecture has become too complex, which services are not addressed by the model, and what is required for long-term successful deployment of multicast service.

The goal of this article is not to prove or show that the current model is wrong. Rather, it is to show that the open multicast service model and the complexity in providing the necessary functionality for ISPs are limiting the possibility of Internet-wide multicast.

In the next section we review the current service model and the architecture that supports it. We then analyze the motivations of ISPs and customers for using a multicast service. Next, we examine the difficulties ISPs have had with the current model and architecture. We discuss the functionalities that are lacking from the service model, and propose alternate services models that are more aligned with commercial deployment. Finally, we offer our concluding remarks.

IP Multicast

The Current Service Model

IP multicast is based on an *open* service model. No mechanism restricts the hosts or users from creating a multicast

[1] *By architecture, we mean the set of protocols supported by the IETF and vendors to realize the service model.*

group, receiving data from a group, or sending data to a group. The notion of group membership is only a reachability notion for receivers and is not meant to provide any kind of access control. As with all IP datagrams, multicast datagrams are best-effort and unreliable. Each multicast group is named by a class-D multicast address (which is, in fact, a name [2]).

To receive data from the multicast group, hosts must join the group by contacting their routers using the Internet Group Management Protocol version 2 (IGMPv2) [3]. Once a host joins a group, it receives all data sent to the group address regardless of the sender's source address.

Hosts can send to a multicast group without becoming a receiver; such hosts are often referred to as *non-member senders*. Multiple senders may share the same multicast address; whether those sources share a single multicast routing tree or have separate trees leading to the receivers is dependent on the multicast routing protocol. Senders cannot reserve addresses or prevent another sender from choosing the same address. The number of hosts joined to a group as receivers is dynamic and unknown. The status of entities (i.e., sender, receiver, or both) is unknown. In sum, an IP multicast group is not managed.

The connections between the routers that form the multicast spanning tree are maintained by a multicast routing protocol. Many such protocols have been proposed and are in use today on the Internet. They include (but are not limited to) Distance Vector Multicast Routing Protocol (DVMRP) [4], Multicast Open Shortes Path First (MOSPF) [5], Protocol Independent Multicast Sparse Mode (PIM-SM), PIM Dense Mode (PIM-DM) [6–9], Core-Based Trees (CBT) [10], Ordered CBT (OCBT) [11], HIP [12], and Border Gateway Multicast Protocol (BGMP) [13]. As we will see next, the deployed architecture has tended toward just a few protocols.

The differences in these protocols lies mainly in the type of multicast routing trees they build. DVMRP, MOSPF, and PIM Dense Mode build multicast spanning trees that are shortest path from each source. PIM-SM, CBT, OCBT, and HIP build multicast spanning trees that are shortest path from a known central core, also called a *rendezvous point* (RP), where all sources in the session share the same spanning tree. (PIM-SM is a complicated protocol that at times builds source-rooted shortest path trees.) CBT, OCBT, BGMP, and HIP build *bidirectional shared* trees: packets from each source are disseminated along the tree starting from any point. PIM Sparse Mode uses a *unidirectional shared* tree, where packets are sent first to the core, which then sends packets down the multicast spanning tree to all participants of the session.

The Current Architecture

The de facto architecture in routers today is based on IGMPv2, DVMRP, MOSPF, and PIM-SM, coupled with the Multicast Source Discovery Protocol (MSDP) [14] or Multicast Border Gateway Protocol (MBGP) [15]. DVMRP, MOSPF, and PIM-SM are limited in applicability to autonomous systems and administrative domains. Interdomain multicast routing is largely managed by MSDP.

IGMP is used by hosts to announce their interest in receiving a multicast group to edge routers. These edge routers use multicast routing protocols to form multicast spanning trees through the Internet. IGMPv1 [1] was proposed in conjunction with DVMRP, the first multicast routing protocol. IGMPv2 [3] adds fast termination of group subscriptions and is an IETF standard. IGMPv3 [16] is a work in progress. It allows receivers to subscribe to specific sources of a particular multicast group.

DVMRP is a *flood-and-prune* protocol. The source of a multicast group floods the entire domain with multicast datagrams, which also serve to announce the existence of the group. Datagrams that do not arrive at a router on the reverse path interface back to the source are ignored, and a prune message is sent in reply to the neighboring router. End routers that do not service any hosts interested in receiving the multicast group also prune back the spanning tree. DVMRP was never meant to work beyond a small autonomous domain because its flooding mechanism does not scale to the entire Internet. PIM Dense Mode is very similar in operation to DVMRP, except it is independent of the underlying unicast routing protocol.

MOSPF is based on OSPF routing mechanisms. Group membership information is flooded throughout the network, and per-source trees are computed by each router using link-state routing information available from OSPF. Similar to DVMRP, MOSPF is regulated to intradomain scenarios.

PIM-SM (which is similar to PIM Dense Mode only in name) is based on the concept of RPs, predefined points in the network known by all edge routers. Edge routers with attached hosts interested in joining the multicast group start a multicast tree by sending *join* messages on the shortest reverse path to the RP, which instantiates a new branch of the RP's unidirectional shared tree. After forming a branch to the RP of a session, the newly joined edge routers learn of each source joined in the same session (i.e., member senders). The edge routers then switch to a shortest path tree for sources that transmit over a certain threshold. PIM-SM builds shortest path trees by sending join messages to each source in the session. The edge routers then prune back on the RP's tree for that source. This results in per-source-per-group routing table entries in the multicast tree. As we discuss later, the current operation of PIM is different from its intended design.

If receivers using PIM-SM wish to join multicast groups with sources located in remote domains (with remote RPs), PIM-SM requires that the group-to-RP mapping be advertised to all edge routers in PIM-SM domains. When crossing provider domains, an interdomain multicast routing solution is required. Currently, the most commonly employed solution is the Multicast Source Discovery Protocol (MSDP), which distributes this mapping and announces sources via TCP connections between RPs. MSDP runs over a multicast-capable Border Gateway Protocol (commonly known as BGP4+ or MBGP) [15], which is a set of multicast extensions for BGPv4 that separates unicast and multicast policy. We discuss MSDP in detail later.

Because there is no standard, globally recognized method of allocating addresses uniquely in the current model, the Internet Engineering Task Force (IETF) is experimenting with static allocation of blocks of multicast addresses. This scheme is often referred to as *GLOP* [17]. This experiment should last until May 2000, when it is expected that protocols developed under the Multicast Address Allocation Architecture (MAAA) [18] will be implemented.

In the near future, interdomain multicast is expected to be managed by BGMP [13], an interdomain protocol used to manage interoperability between multicast routing protocols in different domains. It uses bidirectional shared trees between domains and relies on MAAA protocols or GLOP to designate the *core domains* of multicast groups, and to solve address allocation and core placement. (In BGMP and HIP, entire domains act as cores.)

The right side of Fig. 1 illustrates the IP multicast architecture. Interdomain support, if present, is based on MSDP or BGMP, which rely on MBGP. Intradomain multicast routing trees are built by CBT, PIM-SM, or PIM Dense Mode, which rely on the presence of an underlying unicast routing protocol. MOSPF relies specifically on OSPF. DVMRP includes its own unicast routing protocol. Hosts ask routers to join multicast groups with IGMP. Multicast address allocation is not defined in the IP multicast service model. Presently, allocation defaults to the static GLOP model. An alternative proposed option is the combination of MADCAP, AAP, and MASC that makes up MAAA. Session announcement may be performed with SAP. Reliable multicast protocols provide error correction and congestion control for multicast sessions. Not shown are group-key distribution protocols, which manage shared encryption keys across large receiver sets to provide receiver authorization services. On the left side of Fig. 1 are corresponding unicast protocols.

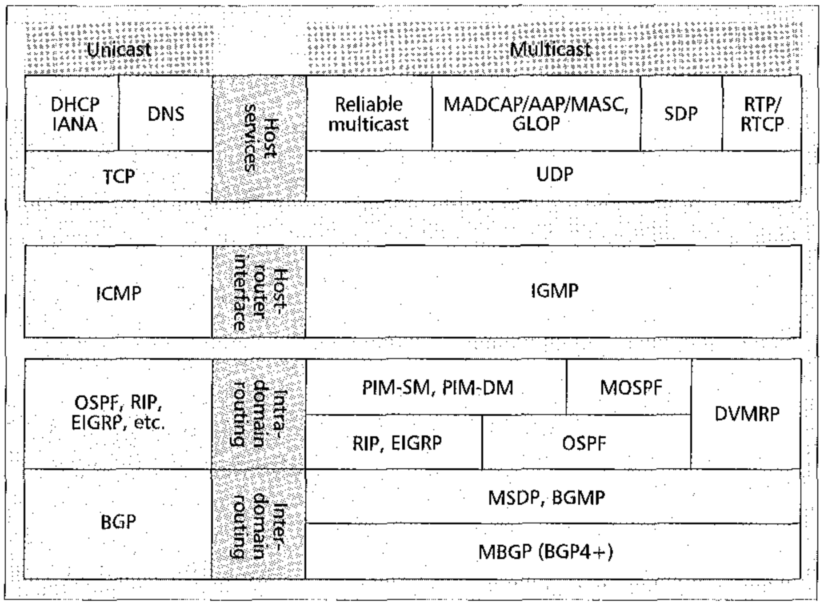
An in-depth description of the IP multicast architecture, including the history of its design and deployment, is available elsewhere [19].

Motivations and Requirements

Multicast is included with the standard set of protocols shipped with most commercial routers, but most IP carriers have not yet enabled the service in their networks. A number of issues have stalled the widespread use of multicast. We preface a discussion of what has stalled multicast deployment with a review of the applications that are driving multicast and the requirements of ISP customers.

Market Motivations

Businesses have been encouraged to connect to the Internet ISPs by the phenomenal success of unicast-based e-mail and Web applications. However, general users of the Internet (i.e., receivers) will not drive Internet-wide multicast connectivity. The use of multicast results in bandwidth savings that make it an attractive service mainly to sources and administrators of low-capacity domains, such as corporate networks. Receivers do not care whether they receive their audio streams from unicast or multicast. As receivers, they require the same amount of bandwidth that they would obtain with unicast transmission (this argument may be generalized to other aspects such as real-timeliness or quality of service). Moreover, users will find unicast delivery a more stable service at this point.



■ Figure 1. *A comparison of protocol components for IP unicast and IP multicast architectures.*

Sources require multicast so that they may scale their services to extremely large audiences. Low-capacity domains require multicast only when many redundant high-bandwidth streams threaten the capacity of incoming links. For example, many employees in a company may choose to all receive unicast streams of a popular video event, overwhelming the incoming bandwidth capacity.

The current set of applications driving multicast deployment are typically one-to-many or few-to-few, and fall into four categories:

- Audio and video distribution, also referred to as Webcasting, involves one source sending real-time audio and video over the Internet to one or more receivers simultaneously. Many Web sites have already made video distribution an integral part of their content.
- Push applications (information delivery) allow individual users to select from a variety of information or content bundles, called *channels*. This information is then automatically spooled and pushed to them at regular intervals. PointCast is an example of an existing push application. This application is always downloading information, up to 100 kbytes/hr, even if the user is otherwise occupied and not actively reading the information. Because of this, the impact of unicast bandwidth for PointCast and others like it has been substantial and troublesome for corporate networks. Companies that provide push technology are looking for ways to conserve bandwidth to keep corporations from banning these applications entirely.
- Audio and videoconferencing and group collaboration applications build on the capabilities used for Webcasting, but allow users to interact with each other. However, because of social issues, these applications, which appear to be many-to-many, are in reality likely to be few-to-few, or multiple instances of one-to-many.
- File transfer involves sending data (typically large amounts of data) from one location to one or more locations. As the amount of data grows and the number of recipients increases, the bandwidth requirements and the time to complete file transfers can become unmanageable. Multicast file transfer services support Web caching, distributed databases, and remote logging.

In the longer term, more applications with more interaction among users will appear. We believe such interaction will appear first at a low level, in streaming applications (e.g., interaction with the content), and then with the deployment of shared virtual worlds and distributed games. Multicast is then a mandatory technology to allow such interaction due to its scalable dissemination of data and because it minimizes delay among participants [20]. The scale of the multicast groups for these applications is likely to be tightly tied to social and human factors issues, and should not automatically be assumed to require large-scale many-to-many multicast.

Customer Requirements

Customer requirements and market motivations dictate to carriers and ISPs which functions to provide, and consequently which service model to implement. Commercial use of multicast will require at least the same level of availability and maintainability as unicast.

The following customer requirements can be extrapolated from the market motivations and experiences with IP unicast services. They are partially motivated by the fact that multicast is not a service which adds value for the receiver:

- ISP customers must have ubiquitous global access to multicast services. This requires scalable interdomain access to multicast services.
- Multicast will be an attractive service only if it is easy and transparent to install. The ISP's ability to install, manage, and maintain the multicast service is an important customer criterion for selecting service providers. Similarly, setup and configuration of a multicast session must have low latency and be straightforward. Network management for customers should be easy. Corporate customers regularly rely on management services to provide granular usage statistics and billing information that can be used to plan network expansion, bill back users, and verify service-level agreements.
- Senders expect group membership to be controlled for both senders and receivers. For senders it is important that only authorized sources send to a multicast group; either because a content provider wishes to be the only source of data being sent to the group, or because of concerns about denial-of-service attacks via flooding. Likewise, the set of receivers, or scope, of the group must be controlled. Note that this may be more complex than simple time-to-live or domain scoping. Sources may wish to authorize receivers in several domains without delivering content to the entire Internet.
- Similarly, content providers will expect that their assigned multicast addresses are unique (minimally, for the duration of their session). This is for several reasons. First, applications will not expect data from separate sessions to arrive on the same multicast address. Second, separate sessions may have different bandwidth requirements, and if they are on the same multicast tree, a high-bandwidth session will drown out a low-bandwidth session unnecessarily. Finally, placing separate sessions on separate multicast addresses makes network management easier (for tracking of problems). Other reasons can be found.
- Finally, reliable transmission may be required. Today it is provided experimentally at the application level, but it is unclear whether a robust, reliable multicast can be built without support from the network.

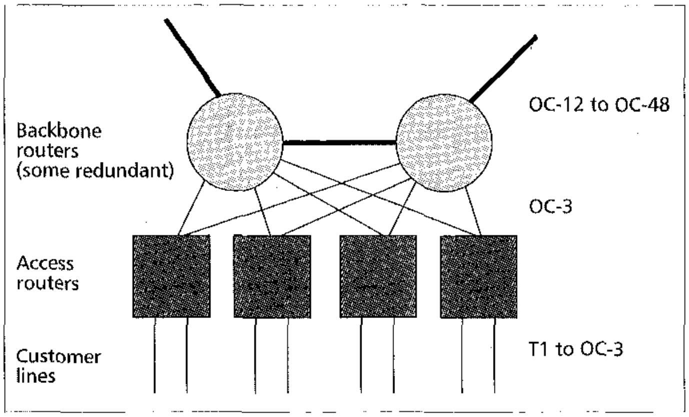
We show why these requirements are not easily provided to customers with the current service model.

Deployment Issues

Multicast currently relies on a protocol architecture that requires more setup and administration than the unicast architecture. In this section we report and analyze experiences in deploying the multicast architecture for commercial use. It has been noticed by major carriers that the current architecture is unstable [21]. We try to understand whether this is the result of bugs in protocol implementations or the architecture is broken.

Router Migration

Multicast deployment at a customer's premises is not a simple issue due to the legacy of existing network infrastructure. A long-term problem for multicast deployment is that it upsets the *router migration model* ISPs follow, which is where routers are initially deployed in the backbone and, over time, pushed toward customer access points. Figure 2 illustrates a typical ISP point of presence (POP). Customer access lines are fed into edge routers, which in turn are connected to higher-capacity backbone routers. As customers acquire higher-speed access lines, backbone routers are migrated toward customers' access points to handle the higher-speed access lines. Newer routers that support even higher bandwidth are added to the backbone. In other words, routers are generally installed in the backbone and pushed toward customer access lines as technology moves forward.



■ Figure 2. *A typical ISP POP structure.*

Multicast upsets this model because older hardware generally does not support multicast. When there are no software upgrades offered, the routers are forced into early retirement. Companies rely on the depreciation of their hardware's value in their business models. However, removing hardware for upgrades prevents a normally available tax write-off of the depreciation. Furthermore, the natural cycle of cost of migration results in the use of equipment longer than a simple model of its value would predict. Hardware is typically removed when the cost to remove and replace it is less than or equivalent to the cost to maintain or upgrade vital components that would make the hardware support new features.

For example, deploying native support for multicast for dialup customers might require replacing dialup servers before their fully depreciated value can be written off, and before their planned longevity as part of the network infrastructure. In some cases, multicast is provided by forcing dialup customers to send multicast datagrams encapsulated in User Datagram Protocol (UDP) packets to a proxy, which then multicasts the data to receivers.

Router migration has another implication for multicast architecture designs. New routers that are deployed in the backbone are generally less intelligent routers, lacking complicated services such as congestion and admission control. Routers that are simple and unintelligent can handle higher-capacity traffic more efficiently. Therefore, complex services like multicast would be better deployed in the edge routers, but replacing such routers upsets the business model. Therefore, both backbone routers and edge routers resist multicast deployment. And despite frequent software updates, multicast will not be fully deployed in networks managed by carriers before a new generation of routers has been installed at all levels of the network architecture.

Domain Independence

For applications with many low-rate sources, such as distributed games and DIS applications, it might be more efficient to have all sources share a tree. Such trees are more efficient in terms of the amount of state at routers (although not with the data carried to receivers [22]). Protocols like PIM-SM and CBT were designed to support shared trees.

However, ISPs using PIM-SM or other RP/core-based protocols face a number of problems regarding domain independence. Many problems are present when RPs and their associated sources are in distinct domains:

- Traffic sources in other domains potentially require traffic controls, such as rate or congestion control.
- An ISP that relies on an RP located in another domain has very little control over the service its customers receive via the remote RP.
- ISPs do not want to be the core of a session for which they have no receivers or sources since it is a waste of their resources.
- Advertisement of the address of the RP or core must occur in a scalable fashion with low latency.

MSDP was introduced to announce PIM-SM source-to-group mapping information so that trees could be built directly toward the source's domain without third-party dependencies. (This also amounts to solving the problem of announcing RP-to-group mappings.) In MSDP, neighboring domains (i.e., peers) announce sources to each other using *source active* messages. MSDP floods source information to all other RPs on the Internet using TCP links between RPs. RPs servicing receivers that are interested in a particular source then join on the shortest path to the source.

MSDP occasionally carries multicast data within the source active message to avoid the delay in transmitting data while these messages are propagated, and to avoid the timeout of bursty sources at remote RPs. TCP is used in MSDP for two reasons: first, to ensure that source active messages containing encapsulated multicast data are delivered in order; and second, because the information sent may be too large for a single UDP datagram and may arrive out of order. Unfortunately, RTCP and many reliable multicast applications perform multicast round-trip time estimation with low-rate session messages, but if a TCP retransmit timer is used, RTCP will return unrepresentative results for the high-rate data flows. For this reason, MSDP is being modified by the IETF to use unreliable GRE tunnels between peers. Unfortunately, MSDP does not scale due to its periodic flood-and-prune mechanism. It also has dramatic effects on the transmission delay and breaks the IP multicast service model by carrying data over TCP. However, it does eliminate the problems related to RPs that are not located in a source's domain.

Specific to PIM-SM are problems due to the difference in its deployment from its intended design. PIM-SM uses RPs so that applications with multiple low-rate sources can benefit from shared trees. However, *deployed* PIM-SM never uses shared trees for transport for two reasons: MSDP and incorrect variable settings.

First, MSDP prevents the use of shared trees between domains. This is because when remote RPs receive source active messages, they join directly to the source and not to the RP of the source. Even when two sources are collocated in the same domain, RPs in remote domains will form two separate per-source branches, one to each source. Accordingly, MSDP defeats the shared tree support in PIM-SM between domains.

Second, although PIM-SM specifies that receivers should only switch to a per-source tree when the rate of a source passes a threshold, in practice major vendors have set the default setting of the threshold to zero kb/s. With such a setting, the following steps occur in deployed PIM. Receivers begin by joining an RP's unidirectional shared tree. Next, receivers immediately learn of all other participants in the session. Finally, receivers immediately form per-source trees to each participant in the session. Therefore, in practice, PIM does not construct shared trees for any source with more than ephemeral traffic.

PIM-SM was designed to support both per-source trees and unidirectional shared trees. The deployment of PIM-SM and MSDP defeats these design goals: deployed PIM-SM is de facto a per-source protocol that also suffers from the problems of shared tree protocols, such as third-party dependencies and RP advertisement. This set of problems disconnects the deployed architecture from the intended designs.

An alternative solution to the problems of core-based protocols is given by the Simple Multicast [23] model. Core advertisement is left to a session discovery tool, e-mail, or Web pages. The core of a multicast group is told to routers by hosts by including a core-multicast (C, M) tuple in the header of all packets destined for a particular group. The use of a (C, M) tuple makes the architecture simpler.

Management

Due to the complexity of the protocol architecture described in the previous section and to the poor interoperability with existing services, multicast is extremely difficult to install and manage.

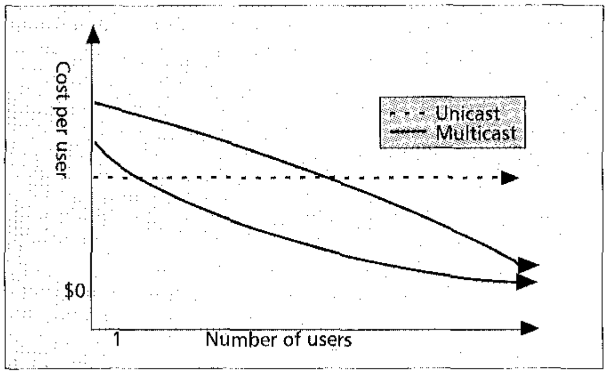
Multicast deployment at a customer's premises is not a simple issue due to the legacy of existing network infrastructure. Problems common to multicast deployment today include firewalls and a lack of support for network address translation (NAT). An Internet draft on NAT has been issued [24], but is not yet implemented as a standard solution in commercial equipment. The main problem with most firewalls is that multicast (i.e., class D) addresses are not recognized. The only solution to this problem is to tunnel multicast packets through the firewall. This creates a serious security hole in the system where multicast is deployed. Until these problems can be fixed by vendors, the solution supported by vendors to solve firewall incompatibilities is to use static routes to all multicast-enabled routers.

ISPs are having a difficult time managing multicast, although it has yet to become a popular service. While intradomain multicast is relatively easier to deploy, providing interdomain multicast is complicated. Interdomain multicast precludes complete control of the network, which makes it difficult to debug problems. This is important with protocols like BGMP or MSDP, which involve contact with other domains. We review multicast management tools later.

Justifying the Cost of Multicast

Multicast is currently a service that reduces the amount of bandwidth required to transport data to multiple recipients. In the longer term, it may also be used to minimize network delays in interactive application sessions. Multicast services are currently significantly more expensive than unicast service, in terms of deployment, installation at customer premises, and management. Consequently, multicast makes sense today for an ISP or corporate customer only when the bandwidth savings are higher than the deployment and management costs. Cain suggests that this is multicast deployment's *sweet spot* [25]. Because multicast is more complex and more difficult to manage than unicast, it is only cost effective for an ISP to deploy multicast and manage it for customers when doing so saves significant bandwidth. The sweet spot is where the additional cost of providing the service is outweighed by the gained performance benefit.

The *cost* of a network service can be defined as the sum of network-related costs (router state, processing, and signaling; interdomain routing scalability) and management costs (ease of deployment and maintenance in terms of human resources and infrastructure). Figure 3 illustrates this principle. Unicast is represented as a straight line because each new receiver adds a new cost (mostly network cost). Multicast, however, has an initial cost higher than unicast; but the cost of adding new receivers should not be as high as in the unicast case. This scenario is ideal, and is represented as the downward-sloping curved lines in Fig. 3. It corresponds to a group where all members would join the same source in the same



■ Figure 3. *The multicast sweet spot occurs when the performance benefits of a new service outweigh the costs as compared to unicast.*

autonomous system (AS). Therefore, there is an opportunity to amortize the cost of multicast over each receiver. To date, multicast is in fact very costly. A more accurate representation of multicast may be the less optimistic second curved line: each additional multicast receiver may exist in a different domain, causing management and network costs that exceed the benefit of efficient multicast routing. In the best case, multicast is advantageous to use over unicast services for low numbers of receivers (occurring at the intersection point on the graph). Less optimistically, multicast requires a larger receiver set (possibly an order of magnitude larger than the optimistic case) before there exists a benefit over unicast.

Consequently, there is an incentive for ISPs and content providers for supporting small group sessions with unicast rather than multicast. Broadcast.com, for example, follows this philosophy. Web events where the expected audience is small are supported by unicast connections because the bandwidth saved is not worth the overhead of multicast management. For events as large as the Victoria's Secret fashion show, which attracted 1.5 million visitors [26], multicast bandwidth savings were sought whenever possible. Such a large audience has the potential to overwhelm any large collection of servers and available network bandwidth, making multicast a profitable and useful service for both servers and the network.

Cain's sweet spot principle predicts that while the multicast architecture remains complex and difficult to manage, it will face difficulty in reaching wide deployment.

The next section enumerates additional carrier and ISP requirements not yet met by the multicast service model. These additional requirements raise questions about whether the complexity of a commercially viable Internet-wide multicast architecture will ever be simple enough to inspire Internet-wide connectivity.

Functionality Not Addressed

Earlier, we reviewed market motivations that drive customer requirements of multicast services. Then we reviewed the difficulties faced by ISPs when deploying multicast. In addition to these difficulties, there are functionalities not well addressed by the current model and architecture, which we analyze in this section. Many customer requirements concern missing components of the IP multicast service model. These components are a prerequisite to successful commercial deployment of multicast. We review the seriousness of these concerns and the complexity each adds to the current model. For most of these functionalities, solutions exist that are either currently proposed for IETF standardization with no modification to the service model, or being studied in the context of a new service model.

Group Management

The current service model does not consider group management, including receiver authorization, transmission authorization, and group creation. Group management may also include billing policy and address discovery. We address such issues separately, choosing to define group management as access control functions that limit who may send and receive on a particular multicast address.

The lack of access control functions presents a danger for companies providing content over multicast groups as well as for receivers that pay for a given service. Just as Web sites require protection from hackers attempting to change the content of a Web site, multicast-based content providers require access controls as protection from outsiders launching a number of possible attacks, including:

- Flooding attacks, where high-rate useless data is transmitted on the same multicast group, causing congestion and packet loss. Flooding attacks prevent reception of data by valid receivers. Although this is a problem for unicast as well, multicast affords the opportunity for attacks of much larger magnitude and scope.
- Collisions of sessions. Due to the lack of group creation controls, two sessions using the same address can interleave their data.
- Unauthorized reception of multicast data, including pay-for content, such as pay-per-view events. This represents a source of lost revenue for content providers. This problem exists for unicast; however, the solutions for multicast require group-key management, a topic which is just beginning to see solutions.
- Drowning out of authentic sources with alternate data, changing the content of the session. This is also a source of lost revenue.

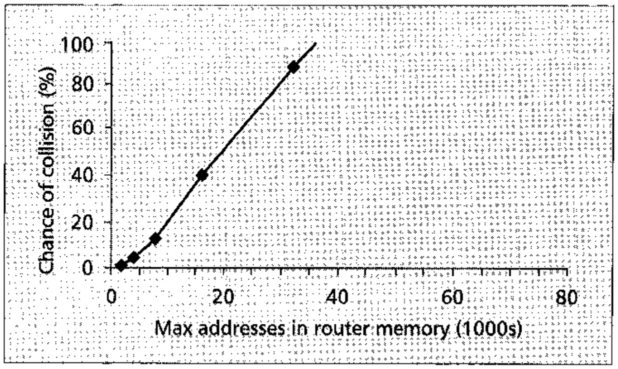
Without access control mechanisms, such attacks are trivial to implement.

One enhancement over current IP group management is IGMPv3 [16], which provides source pruning for specific multicast groups, as well as source-specific joins. IGMPv3 prevents data from entering the backbone when the routing protocols support this option. Unfortunately, it is not possible to prune sources or have source-specific joins in shared tree protocols, such as CBT or BGMP (although BGMP is compatible with Express-style multicast groups). In addition, attacks may still be possible in the backbone with IGMPv3 when even one receiver does not prune all noisy or malicious sources. To prevent such a scenario, receivers would have to explicitly subscribe to a known source list (as occurs in Express) rather than prune noisy sources after the fact. Note that IGMPv3 is still under development.

Sprint and UUNet have deployed multicast as a commercial service. However, nothing prevents receivers from joining any particular group other than restricting access to the Web page that lists the multicast group address.

Multicast Security

Providing security for multicast-based communication is inherently more complicated than for unicast-based communication because multiple entities participate, most of which will not have trusted relationships with each other. Future multicast security should provide four distinct mechanisms: authentication, authorization, encryption and data integrity. Authentication is the process of forcing hosts to prove their identities so that they may be authorized to create, send data to, or receive data from a group. Authorization is the process of allowing authenticated hosts to perform specific tasks. Encryption ensures that eavesdroppers cannot read data on the network. Data integrity mechanisms ensure that the datagram has not been altered in transit.



■ Figure 4. *The chance of address collision in IPv4 with restricted available router memeory, when all memory is allocated.*

The current IP multicast service and architecture do not mandate any authentication. Source authentication and data integrity is possible through the services provided in IPsec, but not receiver authentication. Furthermore, IPsec does not prevent sources from sending; it just allows receivers to drop unauthenticated packets after they are received. IPsec is not widely deployed and is currently under study by the IETF. Other solutions to this problem have been proposed end-to-end and at the network level.

Encryption is often cited as the appropriate mechanism to preserve data privacy at the application level. Unfortunately, for large heterogeneous groups, application-level key management is at best a partially solved problem. To maintain scalability in the presence of a large receiver set, rekeying must be done on portions of the tree [27–29]. For example, the Iolus protocol [29] protects data from unauthorized receivers with data encryption. Unlike normal multicast delivery, Iolus has the drawback that packets may cross some links more than once.

Secure multicast services are network-level solutions to ensure that multicast tree construction and delivery services are restricted to authenticated and authorized hosts. Such protocols are therefore more resistant to attacks, such as denial-of-service and theft-of-service attacks. For example, Keyed-HIP (KHIP) [27] is a network-level security scheme that restricts subbranch construction to unauthorized domains and hosts. KHIP provides transmissions and receiver access control, as well as data integrity. Each packet is encrypted to ensure data integrity.

KHIP uses a bidirectional, core-based multicast routing tree, and lacks facilities for excluding specific sources from the tree. In fact, all bidirectional shared tree protocols break down into the same state as per-source trees when individual sources must authenticate for each receiver. Cain has suggested placing authorization mechanisms at the edge of the network to maintain group state in the presence of receiver-specific source prunes [30]. However, such a scheme maintains security at the edge routers. If the edge routers are bypassed, unauthorized transmissions will enter the backbone. The advantage of such as scheme is that it keeps complexity on edge routers and out of the backbone.

One very serious unresolved issue with multicast security is the location of access lists. One simple model is to place control of authorized receivers at the (primary) source of a session. Such a model does not resolve who authorizes sources within domains - presumably, it would be handled by system administrators - or interdomain authorization. One alternative to source-based authentication would be to use authentication servers.

Note that security mechanisms are often at odds with application requirements of fast joins, pointing toward the use of multicast groups within multicast groups [22, 31], or authentication of blocks of addresses.

Address Allocation

Because the current multicast address space is unregulated, nothing prevents applications from sending data to any multicast address. Members of two sessions will receive each other's data if separate addresses are not chosen. A lack of address allocation mechanisms poses no threat to ISPs, other than that of dealing with angry customers and carrying unwanted data. However, address collision poses a serious inefficiency risk for multicast receivers and can create application inconsistencies. This is because packets from other sessions must be processed and dropped.

This problem could be partially solved with proper access controls for group creation, which would limit collisions via sender access lists.

A proper allocation scheme would have a number of properties:

- No single user could disrupt service to other users, for example, by allocating all addresses
- No, or negligible, delay in address allocation so as not to delay applications
- Low complexity of implementation
- High scalability to interdomain environments
- Efficient utilization of the address space
- Long-term scaling to millions of multicast groups

The chance of an address collision is very limited right now only because multicast has yet to become a popular interdomain service. The average multicast-capable router sold and deployed today has memory available for only 1000–2000 (source address, group address) entries.[2] The limited memory of routers in the current deployed Internet limits the chance of address collision because new groups cannot be created after memory runs out. Deriving the chance of address collision is a simple application of the "birthday problem," often applied to hash collisions. The chance of no collisions for X addresses is simply $(2^{28})(2^{28} - 1) \cdots (2^{28} - X + 1)/(2^{28})$. (Therefore, the chance of a collision is one minus this value.) For the 268 million class D multicast addresses available, the chances of collision are limited to 0.78 percent for memory that can hold 2K addresses. However, if multicast becomes more popular (and routers reserve more memory for multicast addresses), the problem of multicast allocation will become a serious issue. For routers with memory that can store just 8K addresses, the chances of a collision when all addresses are used increases to about 12 percent. Figure 4 shows the graph of the probability of a collision of addresses given a limited amount of router memory (in units of addresses).

Currently, there are four alternatives to the current model for address allocation:

- MAAA [18]
- Static allocation and assignment [17] (referred to as GLOP)
- Per-source (or channel) allocation as proposed by the Express [32] model (or in a similar way by the Simple Multicast [23] protocol)
- IPv6 addressing [33]

MAAA's design emphasized the efficient use of a dynamically allocated address space at the cost of complex design. GLOP uses AS numbers as the basis for restricting addresses available to domains. GLOP is a short-term experiment to be reviewed in May 2000. IPv6 drastically increases the address space at the expense of changing IP packet structures, although IPv6 was designed to be incrementally deployed in the Internet. IPv6 is a major rework of IP, but does provide

[2] *As discussed earlier, deployed PIM-SM shared-tree (*,g) entries do not save state because of the automatic switchover to shortest-path (s,g) by receivers upon joining the tree.*

sufficient unique addresses to make address allocation easy. IPv6 and Express solve all requirements, requiring a change in current packet header formats for IPv6, and the deployment of IGMPv3 for Express. (Simple Multicast would also require changes to packet-header formats.)

MAAA is the most complex of these choices. It consists of three protocols connecting hosts, domains, and multicast address allocation servers. Hosts request addresses from servers using the Multicast Address Dynamic Client Allocation Protocol (MADCAP) [34]. The servers inform each other of claimed address blocks using the Address Allocation Protocol (AAP) [35]. The allocation of addresses between domains is handled by the Multicast Address Set Claim (MASC) [36] protocol. Even if MAAA scalability issues can be solved by an appropriate implementation, MAAA does not address whether enough multicast addresses are available in the current addressing scheme if multicast becomes a popular interdomain service. MAAA and GLOP could also create the same kind of problems as class-based allocation of IP addresses, i.e., fragment the address space and create starvation.

Express is an alternative to the IP-multicast model that uses a per-source, *channel-based* model [32]. Each channel is a service identified by a tuple (S,E) where S is the sender's source address and E is the express destination address (i.e., a class-D address). Only S may send to (S,E) because receivers subscribed to (S,E) are not subscribed to (S′,E), for some other host S′. Thus, data transmitted from two sources to the same address E is only sent to receivers subscribing to both sources. Similarly, Simple Multicast proposes designating addresses as a (core, class-D address) model for the purposes of core advertisement for shared trees. The scheme in Simple Multicast is not meant to address source authorization. Regardless of purpose, such a tuple solves the allocation problem since address allocation is local to the core or source S listed.

One small problem with Express results from each host using a different multicast address (unlike the current model, where even per-source trees have the same class D address). The session can no longer be identified by a common address among sources. For example, in a distributed game, many users are the source of data. This problem can be solved at the application level by using an alternate identifier. A more serious limitation of Express is that receivers must explicitly learn of every source in the session (whereas this is taken care of implicitly by routers in the case of PIM-SM or CBT using the traditional IP architecture).

The IPv6 addressing scheme offers 2^{120} multicast addresses for world use, driving the chances of collision to near zero. IPv6 offers a number of other advantages, and is already supported by an application programming interface (API) in UNIX and Microsoft WindowsNT operating systems. For an IPv6 router with space for 1024K addresses, the chance of a collision is less than 10^{-24} percent.

Furthermore, an Express-like scheme can be used in IPv6. If a domain of the source aggregator (i.e., the first part of the IPv6 address) is placed in the first part of the 120-bit multicast address, domains can claim implicit ownership of address spaces. Ownership of multicast addresses within a domain can be managed with AAP or a similar protocol.

Using IPv6 satisfies most, if not all, of the properties for a good allocation scheme, and is already supported by vendors and the IETF.

Network Management

Network management refers to the debugging of problems that occur with the multicast tree during transmission, and the monitoring of utilization and operation patterns for the purpose of network planning. The current tools for debugging multicast are all freeware developed as needed by MBone users. Commercial toolkits for multicast network management await widespread deployment of multicast. However, such tools are a crucial part of multicast deployment since deploying multicast without them is likely to generate less than satisfactory customer experiences.

The current set of programs available for multicast management includes Simple Network Management Protocol (SNMP)-based applications, Mrinfo, Mtrace, RTPmon, Mhealth, Multimon, and Mlisten. Almeroth has an excellent survey of these tools and of the issues involved in multicast network management [37].

Also available is the RouteMonitor, a tool that measures the stability of routes on the MBone [38]. RouteMonitor counts the number of times distance metrics for each DVMRP router change in a given period. An MBGP RouteMonitor is under development.

Finally, the Multicast Route Monitoring (MRM) protocol [39] is under development by the IETF. MRM is an SMNP-based tool that has special provisions for collection of SMNP management information base (MIB) data over a multicast tree in a scalable fashion. Most of these tools are academic prototypes; none of them are robust enough to support commercial deployment. They only partially address the various issues in monitoring and debugging, and cannot identify all problems related to the current protocol architecture.

Billing for Multicast Services

Although the multicast service model does not define any support for multicast billing, it is not clear that there is a need in the short term. Today, Sprint provides multicast to its customers at no charge. This makes sense to the extent that it provides savings on backbone costs as compared to multiple unicast streams in one-to-many applications. As discussed earlier, multicast is a service that is useful mostly to content providers and not to general Internet receivers. Pricing schemes and business strategies reflect this.

UUnet advertises its multicast pricing as a comparison against flat rate unicast pricing [40]. UUnet multicast is priced as a flat rate service that is independent of the number of receivers. Customers of UUnet (i.e., sources) choose among six discrete bandwidths and monthly charges: 5 kb/s at $2200; 10 kb/s at $4300; 25 kb/s at $10,900; 35 kb/s at $15,200; 64 kb/s at $27,000; and 128 kb/s at $54,000 [40]. UUnet should be deploying multicast data streams up to 1.5 Mb/s shortly. The UUnet multicast service is called UUcast and is not a native multicast implementation. UUcast sources unicast data to a proxy, which then multicasts the data over UUnet's (or a partner's) backbone on to receivers. By mixing unicast and per-source multicast, UUcast solves some of the deficiencies of the service model. However, UUcast is not interoperable with native multicast services, such as those implemented by Sprint and other carriers.

Additional Services

Additional services that might be offered by a commercial multicast service and support architecture, although not as vital as the above requirements, includes the following; these services are often analogous to existing unicast services:

- Service-level agreement (SLA) and virtual private network (VPN) management. SLAs include guarantees on network availability and latency, and notification of when SLAs are not met. VPNs use a public infrastructure such as the Internet to provide secure communication.
- Network performance measurement. Providing measurements to senders allows applications to adjust properly to network conditions; for example, measurements of the highest transmission delay among members of the group.

- Subcasting. Many efficient reliability and congestion control protocols rely on or make use of subcasting. Subcasting is useful for receiver-based scoping [22].
- Congestion control. Without congestion control, multicast sessions threaten to unfairly overwhelm well-behaved TCP connections. Many proposed solutions address this problem at the transport layer, or directly at the application layer (e.g., layered multicast). It might be the case that network-level congestion control is the best solution; this issue requires more study.
- Low-latency interdomain routing. Routing between domains should be as immediate as intradomain routing from a data transmission standpoint.
- Unidirectional links. Multicast should work efficiently on unidirectional links and with unicast topologies. Satellite links are unidirectional and form asymmetric routing paths. They are already an important element in the delivery of audio and video content.

Alternate Service Models

The current multicast service model is inherently complex. Many of the features that invoke problems are designed to support applications which are not widely popular today, such as multiplayer games and distributed simulations. On the other hand, the service model does not support well the applications we know to be of immediate interest, such as the distribution of streaming media. This is because the model is not restrictive enough. For example, the service model allows multiple senders but does not provide authorization mechanisms. One consequence of this is that we have seen nonstandard deployments, which may eventually discourage software developers from writing multicast applications. For example, UUnet has not deployed standard PIM; they have deployed a proxy-based version in order to control sources.

In order for multicast services to remain manageable by ISPs, and for multicast to remain a standards-based service, we support breaking the deployment of the model into single-source and multipeer parts. We view such a separation as temporary, and it would ease ISP issues with deployment until multipeer services mature to a point where their designs are scalable and manageable. Furthermore, some common functions that do not exist in the current model must be added to both of their parts. These functions have been discussed in the previous section:

- Address allocation
- Access control
- Interdomain management

The ability of the proposed model to easily implement each is discussed in the following sections. Note that solutions to the problem of address allocation is independent of the choice of single-source or multipeer models.

A Single-Sender Service Model

Single-source Internet multicast is a much simpler paradigm to support than multipeer services, and can be deployed successfully right now. Moreover, the driving applications to date are one-to-many, including file transfers and streaming multimedia. Multicast services should initially be deployed around these applications. Additionally, single-source, source-rooted multicast is well supported by ATM networks, whereas shared trees are not.

The single-source service model requires a simpler architecture. There is no third-party problem, and scalability can be maintained by protocols that build routing by means of explicit-join signaling to the source, as suggested by Express. With only one source, routing can always be shortest path back to that source. Complex protocols like the automatic PIM-SM changeover or MSDP peering are unnecessary for single-source applications. RPs or cores are not necessary. Pricing should be easier to manage since it can be compared against unicast streams, which is not the case with the multipeer service model. Authorization of the source can be provided and checked by border routers in remote domains and edge routers in the source's domain. Receiver authorization can be provided by group-key distribution protocols.

Single-source multicast is well supported by the source-rooted Express model. Express is compatible with the current Internet, since its required functions have been well anticipated by IGMPv3. Edge routers can send source-specific (S,G) joins using IGMPv3 for designated Express multicast groups. IGMPv3 is still under development, but Express has already been allocated a space of experimental addresses by the Internet Assigned Numbers Authority (IANA) for which joins from receivers are expected on a per-source basis [41]. The convention of forcing receivers to specifiy exact sources must be enforced by routers for Express to work properly.

Interdomain issues are also simple to implement by this model, since the notion of a core or RP does not apply to the single-source model.

The Multipeer Service Model

Architectures for multisender applications that require multipeer multicast are not as well understood as single-source models. Multipeer sessions based on shared multicast trees are either bidirectional or unidirectional from a known core. Because such trees are not shortest path to a main source, they must be centered at some advertised core, or at a domain acting as a core. This presents a number of problems not present in the single-source tree scenario:

- The core must be advertised or discovered.
- The core must be "well located."
- Secondary cores must exist so that one ISP is not responsible for the robustness of the entire session.

The current architecture addresses these problems with MSDP and GLOP and, in the future, with BGMP and MAAA.

An alternate idea is to use a core-multicast (C,M) tuple, as proposed by the Simple Multicast [23] protocol. Simple Multicast decouples core allocation from routing, and relies on application-level mechanisms to choose and advertise core information. The (C, M) tuple in packet headers would also require some protocol to allocate addresses from a remote core. Presumably, this would also occur out-of-band and at a higher level.

It is not clear whether multisender applications will require a shared-tree model; the trend in such applications is that all data is not usually wanted by or useful to all receivers [22]. Remember that shared trees carry all data to all receivers, and therefore waste bandwidth on unwanted data.

Sender authorization and authentication are more difficult in the multipeer shared-tree model, and are not addressed by any implementation. In the simplest case, once a sender is authorized to send, all receivers in the group must accept the sender. If not, receiver-specific prunes cause the amount of state in the tree to increase toward per-source state. An alternate solution is to prune sources at the end routers using the IGMPv3 protocol [16]. Nevertheless, such mechanisms still allow data to travel through the network, and would not truly prevent denial-of-service attacks or unauthorized senders. Placing gateways at border routers would prevent traffic from entering domains, but would not prevent this traffic from congesting the backbone.

An additional unknown is who controls sender access to the group. If it is a centralized site, that site represents a single point of failure. It is likely that such functionality will be collocated with the core or distributed with a set of cores, adding additional overhead and coordination among remote domains.

	IP multicast	Simple Multicast	Express
Routing tree type	Any	Shared bidirectional only	Per-source unidirectional only
Address allocation	Large address space in IPv6 or proposed in MAAA	(core, class D) model	(Source, class D) model
Sender authentication and authorization (IP-sec allowed in all)	None	None	none, but multiple senders are nor allowed in same group
Receiver authentication and authorization	None	None	(hooks provided)
Protocols required to manage interdomain core/RP Allocation discovery	PIM, MSDP, & BGP + (or BGMP/MASC)	None (Stored in packets)	Cores not used.
Group creation controls	None (proposed with MAAA)	Yes, at core	Yes, at source
Requires modification to packet formats	No (Yes for Ipv6)	Yes	No, but requires IGMPv3 and router cooperation

■ Table 1. *A comparison of multicast service models.*

The multipeer service model is consequently more complex to realize, and seems to offer less robustness and scalability to carriers and ISPs.

Conclusion

After a long period of very useful experimentation using the MBone, commercial deployment of multicast services has begun. In this article we examine the issues that are limiting deployment.

The initial design of multicast was motivated by the need to support one-to-many and many-to-many applications in a scalable fashion. Such applications cannot be serviced efficiently with unicast delivery. The commercial design of multicast must now include the market requirements of ISPs and their customers. ISPs require a service and a protocol architecture that are easy to deploy, control, and manage, and scale well with the growing Internet. ISP customers expect to be the sole owners of multicast addresses, if only temporarily, to be protected from malicious network attacks and thefts of service and content, and to be able to correct network problems quickly. A deployable architecture should be driven by these concerns.

The current architecture does not consider these concerns well. It lacks simple and scalable mechanisms for supporting:

- Access controls, including group creation and membership
- Security, for protection against attacks to the routing and data integrity of multicast datagrams
- Address allocation, including all the properties listed earlier
- Network management; such tools are not well developed at this stage

Many of the mechanisms in the current architecture that address these issues do so too broadly because they consider both the multipeer and single-source models. The applications most popular today are one-to-many, such as file transfer, streaming media, and information push. Many-to-many applications at this point mainly consist of less popular DIS and serverless multiplayer games. (Currently, serverless architectures are not a credible commercial model). Conferencing over the Internet remains few-to-few but is currently better supported by unicast, as Cain's *sweet spot* predicts. By attempting to support many-to-many applications, the architecture has become cumbersome and at times defeated itself. For example, MSDP supports PIM RPs, but prevents the creation of bidirectional shared trees across domains.

We have shown that from a carrier standpoint, deployment that supports the per-source model makes more sense for robust, simple, and scalable multicast services to all customers. We are not suggesting that efforts toward multipeer multicast halt. We suggest only that commercial deployment begin with the well-understood source-rooted one-to-many model and architecture, even if the implication is an increase in multicast routing tables at routers.

Table 1 shows a comparison of the features offered by IP multicast and two recent proposals that rely on a different service model. As stated, Express supports the single-source model, and Simple Multicast supports the multipeer model, which we discussed in the previous section. Both solve the address allocation problem by using an extended address space. Alternatively, a transition to IPv6 multicast would also solve address allocation problems by reducing the chance of address collision to near zero.

We propose that a service model for multicast be defined that supports carrier, ISP, and market requirements. A new protocol architecture, eventually based on emerging solutions, could be designed and deployed, coexisting with the current deployment of IP multicast. Interoperability with the current protocol architecture, and with PIM and IGMP in particular, should be preserved.

Otherwise, the current deployment strategy threatens to compromise the success of multicast as a service that adds value to the Internet and significantly delay the deployment of applications that would benefit from multicast, such as media streaming and interactive applications.

Acknowledgments

We thank the following people for their insightful comments on the issues presented in this article: Kevin Almeroth, Brad Cain, Jon Crowcroft, Preston Grant, Mark Handley, Edward Kress, Manuel Oliveira, Ramanan Shanmugan, and Clay Shields.

References

[1] S. Deering and D. Cheriton, "Multicast Routing in Datagram Inter–Networks and Extended LANs," *ACM Trans. Comp. Sys.*, vol. 8, no. 2, May 1990, pp. 85–110.

[2] J. F. Shoch, "Inter–networking Naming, Addressing & Routing," *Proc. IEEE COMPCON*, Fall 1978, pp. 72–79.

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

[4] D. Waitzman, C. Partridge, and S. Deering, "Distance Vector Multicast Routing Protocol," IETF RFC 1075, Nov. 1988.

[5] J. Moy, "Multicast Extensions to OSPF," IETF RFC 1584, Mar. 1994.

[6] S. Deering *et al.*, "An Architecture for Wide–Area Multicast Routing," *Proc. ACM SIGCOMM '94*, 1994, pp. 126–35.

[7] S. Deering *et al.*, "PIM Architecture for Wide-Area Multicast Routing," *IEEE/ACM Trans. Net.*, Apr. 1996, pp. 153–62.

[8] S. Deering *et al.*, "Protocol Independent Multicast Version 2 Dense Mode Specification," IETF Internet draft, draft_ietf_pim_v2_dm_*.txt, Nov. 1998.

[9] D. Estrin *et al.*, "Protocol Independent Multicast Sparse_Mode (PIM_SM): Protocol Specification," RFC 2362, June 1998.

[10] T. Ballardie, P. Francis, and J. Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Multicast Routing," *Proc. ACM Sigcomm*, Sept. 1995, pp. 85–95.

[11] C. Shields and J. J. Garcia-Luna-Aceves, "The Ordered Core Based Tree Protocol," *Proc. IEEE INFOCOM '97*, Kobe, Japan, Apr. 1997.

[12] C. Shields and J. J. Garcia-Luna-Aceves, "HIP – A Protocol for Hierarchical Multicast Routing," *Proc. 17th Annual ACM Symp. Principles of Dist. Comp.*, Puerto Vallarta, Mexico, June 1998.

[13] K. Kumar *et al.*, "The MASC/BGMP Architecture for Interdomain Multicast Routing," *Proc. ACM SIGCOMM '98*, Sept. 1998.

[14] D. Farinacci *et al.*, "Multicast Source Discovery Protocol (MSDP)," Internet draft, draft_farinacci_msdp_*.txt, June 1998.

[15] T. Bates *et al.*, "Multiprotocol Extensions for BGP_4," RFC 2283, Feb. 1998.

[16] B. Cain, S. Deering, and A. Thyagarajan, "Internet Group Management Protocol, Version 3," Internet draft, draft_ietf_idmr_igmp_v3_*.txt, Feb. 1999.

[17] D. Meyer and P. Lothberg, "Static Allocations in 233/8," Internet draft, draft-ietf-mboned-static-allocation-*.txt, May, 1999.

[18] M. Handley, D. Thaler, and D. Estrin, "The Internet Multicast Address Allocation Architecture," Internet draft. draft-handley-malloc-arch-*.ps, Dec. 1997.

[19] K. Almeroth, "The Evolution of Multicast: From the Mbone to Interdomain Multicast to Internet2 Deployment," *IEEE Network*, this issue.

[20] C. Diot and L. Gautier, "A Distributed Architecture for Multiplayer Interactive Applications on the Internet," *IEEE Network*, July/Aug. 1999.

[21] "Maestro BOF Meeting Minutes," *45th IETF Meeting*, Oslo, Norway, IDMR Working Group, July 1999; http://www.ietf.org/Proc./99jul/45th-99jul-ietf-101.html.

[22] B. N. Levine *et al.*, "Consideration of Receiver Interest in Content for IP Delivery," *Proc. IEEE INFOCOM*, Apr. 2000.

[23] R. Perlman *et al.*, "Simple Multicast: A Design for Simple, Low-Overhead Multicast," Internet draft, Feb. 1999.

[24] R. Finlayson, "IP Multicast and IETF Internet Drafts," draft-ietf-mboned-mcast-firewall-*.txt. Nov. 1998

[25] B. Cain, Nortel Networks, Private Communication, June 1999.

[26] J. Borland, "Net Video Not Yet Ready for Prime Time," *CNET News.com*, Feb. 5, 1999; http://www.news.com/News/Item/0,4,32033,00.html.

[27] C. Shields and J. J. Garcia-Luna-Aceves, "KHIP – A Scalable, Efficient Protocol for Secure Multicast Routing," *Proc. ACM SIGCOMM '99*, Sept. 1999.

[28] C. Wong, M. Gouda, and S. Lam, "Secure Group Communications Using Key Graphs," *Proc. ACM SIGCOMM '98*, Sept. 1998, pp. 68–79.

[29] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting," *Proc. ACM SIGCOMM '97*, Sept. 1997.

[30] B. Cain, "Source Access Control for Bidirectional Trees," *41st IETF Meeting*, Los Angeles, CA, Mar. 30–Apr. 3, 1998.

[31] B. N. Levine and J. J. Garcia-Luna-Aceves, "Improving Internet Multicast with Routing Labels," *Proc. IEEE Int'l. Conf. Network Protocols*, Oct. 1997, pp. 241–50.

[32] H. Holbrook and D. Cheriton, "IP Multicast Channels: EXPRESS Support for Large-Scale Single-Source Applications," *Proc. ACM SIGCOMM '99*, Sept. 1999.

[33] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," IETF RFC 2460, Dec. 1998.

[34] B. Patel, M. Shah, and S. Hanna, "Multicast Address Dynamic Client Allocation Protocol," Internet draft raft-ietf-malloc-madcap-*.txt, May 1999.

[35] M. Handley, "The Address Allocation Protocol," Internet draft, draft-ietf-malloc-aap-*.txt. Aug. 1998.

[36] D. Estrin *et al.*, "The Multicast Address Set Claim (MASC) Protocol," Internet draft, draft-ietf-malloc-masc-*.txt, Aug. 1998.

[37] K. Almeroth, "Managing IP Multicast Traffic: A First Look at the Issues, Tools, and Challenges," IP Multicast Initiative white paper, Feb. 1999.

[38] D. Massey and W. Fenner, RouteMonitor, Available from http://ganef.cs.ucla.edu/~masseyd/Route

[39] L. Wei and D. Farinacci, "Multicast Reachability Monitor (MRM)," Internet draft, Oct. 1999.

[40] UUnet multicast: http://www.uu.net/lang.en/products/access/uucast

[41] Internet Assigned Numbers Authority, http://www.isi.edu/in-notes/iana/assignments/multicast-addresses; see also http://www.isi.edu/in-notes/iana/assignments/ single-source-multicast

Additional Reading

[1] T. Speakman *et al.*, "Pragmatic General Multicast," Internet draft, Jan. 1998.

Biographies

CHRISTOPHE DIOT [M] (cdiot@sprintlabs.com) received a Ph.D. degree in Computer Science from INP Grenoble in 1991. He was a research scientist at INRIA Sophia Antipolis, working on new Internet architecture and protocols, for the last 5 years. He moved to Sprint Advanced Technology Laboratory in October 1998 to take the lead of the IP research group. His current interest is in deployable multicast and Internet resource control. He is a member of ACM and a member of the COST 264 action, and serves as an editor for ACM/Transactions on Networking.

BRIAN NEIL LEVINE [M] (brian@cs.umass.edu) is currently an assistant professor of computer science at the University of Massachusetts, Amherst. He received his B.S. in applied mathematics and computer science from the State University of New York at Albany in 1994. He received his Master's and Ph.D. degrees in computer engineering from the University of California, Santa Cruz, in 1996 and 1999, respectively. His research interests include networked group communication, multicast, and security. He is a member of ACM and Phi Beta Kappa.

BRYAN LYLES (lyles@sprintlabs.com) received his undergraduate degree from the University of Virginia, Charlottesville, and his Ph.D. from the University of Rochester, New York, in 1972 and 1983, respectively. He was a member of the research staff at the Xerox Palo Alto Research Center from 1988 until the beginning of 1998. He is currently senior scientist at Sprint Corporation, where his current research interests include aspects of the evolution of the Internet related to providing public services.

HASSAN KASSEM (hassank@sprint.net)received an M.S. in communications from George Washington University in 1994. He also received an M.S.E.E. from Lvov Polytechnic Institute in 1985. He joined the Sprintlink Data Engineering Team in 1995 after working at UUNET Technologies.

DOUG BALENSIEFEN (doug.balensiefen@sprint.com) had no biography available at the time this issue went to press.