Monitoring IP Multicast in the Internet: Recent Advances and Ongoing Challenges

Kamil Sarac, University of Texas at Dallas
Kevin C. Almeroth, University of California

ABSTRACT

Multicast was one of the first "value-added" services to be developed and deployed in the Internet. In evaluating the success of multicast, if ubiquitous deployment has been the goal, multicast has not been successful. However, if widespread use of multicast as a bandwidth-saving technique has been the goal, multicast has indeed been successful. Upon closer investigation, one of the reasons for only partial success is a lack of support for *service management*. Multicast is particularly hard to manage *interdomain* where it has been less successful, but easier to manage within a domain where network administrators have more control and smaller networks to manage. In this article we survey some of the recent service management efforts, efforts that have been successful intradomain, but fall short for interdomain. In particular, we focus on important topics like monitoring multicast reachability between sources and receivers; understanding the different challenges and solutions between inter- and intradomain service management; and surveying existing solutions to determine whether multicast capability exists on an end-to-end path. Our investigation shows that while not much attention was initially given to multicast service management, more recent efforts have been successful at developing good solutions and tools.

INTRODUCTION

Multicast is one of the first value added services to be developed and deployed in the Internet [1]. The initial deployment of the service was in the form of an overlay network called the multicast backbone (MBone), a network that has since been replaced with native support in much of the Internet [1]. Recently, much attention has been given to the *interdomain* deployment of multicast and its relative "failure" due to less than universal service availability [2]. However, commonly ignored or unknown is that *intradomain* deployment has been much more successful. The key reason for any lack of knowledge about intradomain multicast is the difficulty of monitoring private enterprise-based networks. No published works exist that examine the extent of deployment and use within private networks. We are therefore left to use anecdotal evidence to qualitatively estimate the use and impact of multicast. Fortunately, there is much such evidence. Suffice it to say that a large number of streaming multimedia players, including those by Microsoft and Real Networks, as well as data delivery tools, including solutions from companies like Digital Fountain and Multicast Technologies, make use of multicast. Furthermore, there are numerous articles in newspapers and trade magazines about the use of multicast to deliver popular content. For example, the British Broadcasting Company (BBC) recently announced that Olympic events would be delivered to home users via multicast technology.

Interdomain deployment and use of the initial IP multicast service, now called Any Source Multicast (ASM), has experienced difficulties [2] partly due to the complexity of the protocol architecture implementing the service [3]. More recently, the introduction of Source Specific Multicast (SSM) [1] has eliminated most of these difficulties. Today, IP multicast is at a crossroads, and the continued use of multicast within private networks and across the public Internet depends on effective service management.

Compared to unicast, multicast is a more complex service, and therefore requires additional mechanisms to manage it in the network. More specifically, due to its one-to-many or many-to-many nature, fault and performance management for multicast requires additional tools and systems. In addition, access control (security management), pricing (accounting management), and configuration management require additional support mechanisms in existing network management systems. In fact, there are very few traditional management functions that do not require either augmented tool support or completely new solutions to work for multicast.

0163-6804/05/$20.00

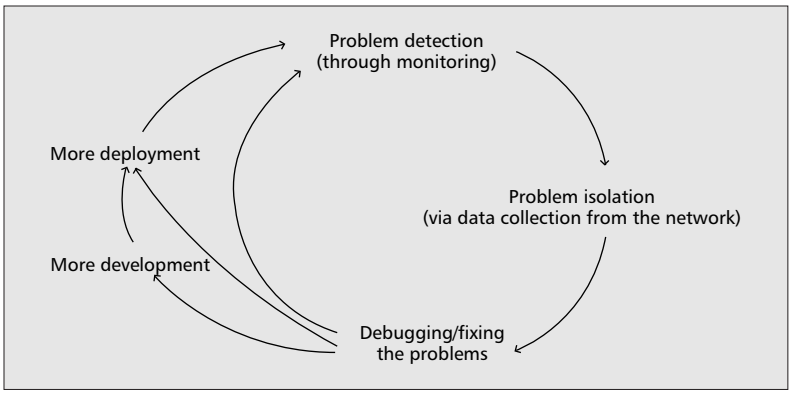

Figure 1. *The role of monitoring in service management.*

Multicast service is realized through creation and maintenance of forwarding trees connecting sources and receivers in a multicast group. These trees are dynamically created and maintained by the routers. There is no feedback information in the process. Therefore, *monitoring* becomes very important to verify the availability of multicast in the network (Fig. 1). Application developers cannot be expected to use an unreliable service in their programs. As a result, our focus in this article is *service monitoring* as the first and one of the most important functions of service management.

In this article we first present a review of the recent studies on monitoring multicast reachability (i.e., end-to-end availability of multicast service between senders and receivers) in the Internet. Then we identify the need for further work in the area and propose new mechanisms to improve the current state of the art in multicast monitoring and management. We divide the monitoring task into three parts:

- Interdomain-level reachability monitoring
- Intradomain-level reachability monitoring
- End-user-level reachability verification

The goal of interdomain reachability monitoring has been to inform multicast researchers and protocol developers about the operation and performance of multicast Internet-wide. These efforts have been useful in observing the robustness of the service and understanding the interaction among the various multicast protocols.

The goal in intradomain reachability monitoring is to help network administrators to monitor and verify proper multicast operation in their networks. Today, IP multicast is successfully used in enterprise network environments to support multireceiver network applications. In addition, standardization efforts are underway to extend multicast usage to multisite enterprise network environments using virtual private network (VPN) technologies. These standardization efforts clearly indicate that multicast usage is continually increasing.

Finally, the goal in end-user-level reachability verification is to help multicast users verify the existence of multicast between themselves and remote hosts. Currently, there is no effective mechanism to provide this feedback. Similarly, the existence of multiple service options (e.g., ASM or SSM) and the lack of a single application programming interface (API) make it difficult for application developers to determine whether multicast is available. In this article we identify the required functionality for hosts to verify service availability and propose mechanisms to support them.

The remainder of this article is organized as follows. We give a brief overview of multicast and the protocols in use today. In the next two sections we describe multicast monitoring work for both inter- and intradomains. We focus on additional tools and systems for multicast service management. The article then concludes.

IP Multicast: A Brief Overview

The original multicast service model aims to support a wide range of multicast applications including one-to-many and many-to-many applications. This service model is ASM, and is currently implemented using a set of protocols [1] including:

- A protocol to construct multicast forwarding trees, called Protocol Independent Multicast — Sparse Mode (PIM-SM) [4]
- A protocol to advertise routes to multicast-enabled networks, called the Multiprotocol Border Gateway Protocol (MBGP) [5]
- A protocol for disseminating information about active sources across domains called the Multicast Source Discovery Protocol (MSDP) [6]

In addition, end hosts use the Internet Group Management Protocol (IGMP) for group management between the receivers and their designated multicast edge routers.

The above protocol architecture also works for IPv6 environments with an exception that instead of IGMP, IPv6 multicast uses the Multicast Listener Discovery (MLD) protocol for group management. In addition, having a large address space in IPv6 simplifies some of the protocol details for ASM. Most important is the ability to replace MSDP and instead embed rendezvous point (RP) information in IPv6 addresses [1].

Soon after the deployment of ASM, a number of problems were identified [2]. These include:

- Protocol complexity
- A number of security vulnerabilities
- Address scarcity for IPv4 networks
- Interdomain scalability problems
- Single points of failure in the architecture

Researchers have since developed an alternative service model specifically to support one-to-many and few-to-many applications. This service model is SSM [1], derived from the EXPRESS protocol [7]. SSM provides solutions to all of the above mentioned problems and requires only a small set of changes to the existing ASM multicast infrastructure.

Interdomain Multicast Monitoring

The multicast service model is an open access service model where senders and receivers may not be known to each other, and there is no implicit group coordination or management.

Therefore, there is no simple way of knowing group membership or verifying that all hosts who wish to receive group data can or will. In addition, multicast researchers and protocol developers need a mechanism to understand protocol operation, protocol interaction, and service behavior. As a result, several tools and systems have been developed to monitor multicast operation in the interdomain. In this article we briefly discuss `sdr-monitor`, the `multicast beacon`, and `mantra` as representative work in the area. None of these monitoring systems require any changes to the existing multicast protocol architecture. We refer the reader to an earlier survey of additional tools, including `mtrace`, `rtpmon`, and `mhealth` [8].

The main goal among the above systems has been to provide a high-level picture of the *global reachability* characteristics of multicast interdomain. In this context, global reachability is used to indicate the availability of multicast data from a source to all or at least some remote receivers. Reachability becomes the ultimate test of whether multicast is working between two sites. The information provided by these tools presents a snapshot of the reachability between a number of sources and receivers rather than between two given multicast sites. While network-wide reachability verification would be ideal, these tools can only scalably provide a partial snapshot. Given the nature of multicast, this is probably the best that interdomain reachability monitoring tools can provide.

SDR-MONITOR AND MULTICAST BEACON

`Sdr-monitor` [9] was the first tool developed to monitor multicast reachability interdomain. `Sdr-monitor` is based on multicast session announcements exchanged by multicast users. `Sdr-monitor` used a number of participants and a centralized data collection site. Participants listened to periodic session announcements sent by others and reported the announcements seen at their local site to the `sdr-monitor` site (Fig. 2). A manager program then processed the reports and built a real-time Web page displaying a reachability matrix for the global multicast infrastructure.

`Sdr-monitor` provided a basic mechanism to monitor the overall status of multicast on an interdomain scale. The scope of the monitoring effort was limited in that reachability monitoring could only be performed between sites that sent announcements and sites that volunteered in the `sdr-monitor` effort. Information collected during the four-year monitoring effort (1999–2003) was used to analyze the reachability characteristics of multicast during the monitoring period. More information about `sdr-monitor` and the analysis of reachability monitoring data can be found in related work [9].

The `multicast beacon` (http://dast.nlanr.net/Projects/Beacon/) is a followup effort to `sdr-monitor`. Instead of relying on passive monitoring of session announcements, the `multicast beacon` uses active monitoring probes to record multicast reachability among a number of participating multicast-enabled hosts. In the `multicast beacon` project, participants both send and receive active probes to/from a well-known multicast group address. Due to its active nature, hosts can compute the reception quality (e.g., loss and jitter values) of incoming multicast data. Similar to `sdr-monitor`, participants are multicast users who volunteer to be in the monitoring effort, and monitoring information is limited to the participants' locations. The `multicast beacon` project does not archive its data, so there are no long-term studies of its reachability data.

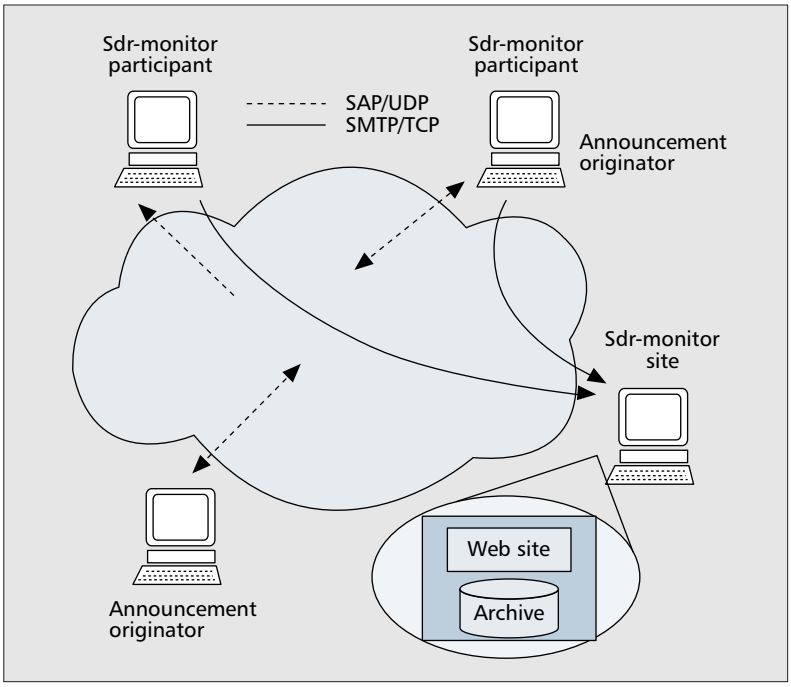

Figure 2. *The* `sdr-monitor` *architecture.*

MANTRA

In addition to the *application layer* monitoring efforts presented above, there have been efforts to monitor multicast by using *network layer* information. `Mantra` [8] is the main example of a system developed to monitor multicast by collecting multicast routing information from the Internet. `Mantra` periodically collected multicast routing information (e.g., MSDP and MBGP tables) from approximately a dozen multicast-enabled backbone routers in the Internet. It then processed this information to generate useful statistics about the deployment and availability of multicast across the interdomain. The information collected by `mantra` has helped researchers and network administrators understand multicast operation, routing protocol interaction, and evolution of the infrastructure.

One important result from `mantra` is shown in Fig. 3. It shows the change in the size of the multicast infrastructure through a display of the number of IP addresses advertised in MBGP over a three-year period. In other words, if a PIM join message were sent to one of these IP addresses, the infrastructure would have a route for the join message to follow. However, like unicast and BGP, just because an IP address is part of an advertised prefix does not necessarily mean that a physical machine exists and has that IP address. Therefore, the results in Fig. 3 are a relative estimate of the size of the multicast infrastructure and its

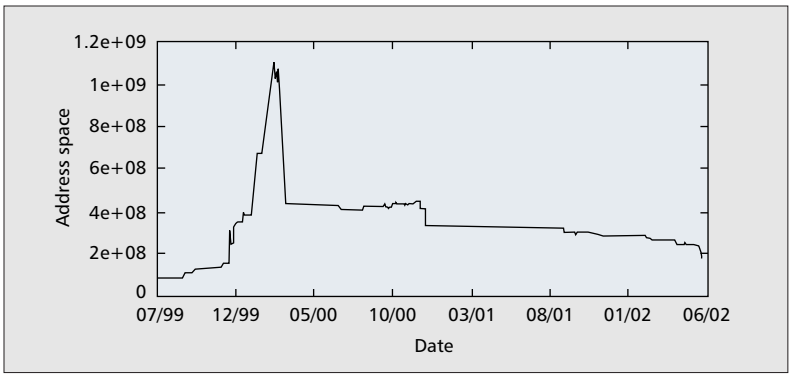

Figure 3. *Results from* `mantra` *showing the number of multicast-capable IP addresses.*

growth over time. The results show a general trend that has the multicast infrastructure either in steady state or slightly shrinking. The spike in address space in the middle of 1999 shows a protocol anomaly that occurred during the transition from the MBone to the infrastructure as it exists today. Additional analysis of these results plus additional graphs are presented in a more comprehensive paper [10].

As with `sdr-monitor` and the `multicast beacon`, `mantra` is only a monitoring solution and not a true service *management* solution. Actual management support for interdomain multicast is a very different and difficult problem. As a result, much more progress has been made in developing systems to provide intradomain management. However, we discuss several recent proposals to add management support for interdomain multicast.

Intradomain Multicast Monitoring

In this section we present an overview of the recent work in developing tools and systems to monitor and manage multicast in intradomain environments. The goal in this area has been to develop the necessary support mechanisms for network administrators to effectively monitor and manage the multicast service in their networks.

Reachability Monitoring Protocols for Multicast

In this subsection we present three systems developed to provide a standard mechanism for multicast reachability monitoring. Each of these systems introduce new protocol(s) or management information bases (MIBs) to be supported by the monitoring entities in the network. The first system is based on the Multicast Reachability Monitor (MRM) [8] protocol. MRM was an effort to define a standard mechanism for conducting reachability monitoring sessions both intra- and interdomain. In addition, MRM supports both passive and active monitoring of multicast sessions. The received traffic at the test receiver's site can be used to verify basic reachability or compute end-to-end capability. Tests can be conducted in advance of an event to confirm proper operation or during an event to monitor quality. In addition to individual tests, a suite of tests can be conducted in which a frequently changing set of network devices is used. In this way statistical testing can be performed across even large networks.

MRM is designed to be used in end-to-end monitoring scenarios using hosts and in-network monitoring scenarios using network devices (e.g., routers). MRM uses a proprietary IPSec-based secure message exchange mechanism to communicate configuration messages.

The second system we discuss is also based on a new protocol. SNMP-based MRM (SMRM) [11] is a followup effort to the MRM protocol and is based on the Simple Network Management Protocol (SNMP). The main goal in SMRM is to integrate MRM functionality into an SNMP-based management framework. Since SNMP-based management is the most widely used management platform, this integration improves the likelihood that MRM will be deployed and used. SMRM defines a number of MIB classes to implement MRM functionality within the SNMP-based management framework. These include McastInfoMIB, smrmMIB, and Schedule-MIBs to include group-specific configuration information (e.g., multicast IP address and port numbers) for active measurements; to define MRM-relevant configuration parameters for the test sessions; and to schedule, initiate, and coordinate MRM experiments. Sallay *et al.* propose another network management system that also uses MRM [12].

`RMPMon` [13] is another SNMP-based system designed and developed to monitor end-to-end performance of multicast. `RMPMon` was built on two key protocols, SNMP and the Real-Time Transport Protocol (RTP). RTP includes a companion protocol, called the Real-Time Transport Control Protocol (RTCP), to communicate information on data reception quality (including packet loss and delay jitter) among the sources and receivers of an active multicast group. `RMPMon` uses two MIBs, RTP MIB and RTP Sender-MIB, to perform the required monitoring. Using SNMP, an `RMPMon` agent can be configured to join and passively monitor multicast performance. In addition, `RMPMon` agents can be configured to create a test session, and act as test senders and receivers. The main difference between `RMPMon` and MRM is that `RMPMon` depends on SNMP for control message exchanges and uses RTP MIBs to implement the required agent functionality, whereas MRM defines its own communication protocol and defines its agent functionality independent of SNMP.

Other Monitoring and Management Systems for Multicast

In this subsection we present several systems that have been developed to provide more comprehensive monitoring and management support for operational network environments.

From an operational network management point of view, `mmon` [14] was one of the first multicast monitoring and management systems.

Mmon is designed to be a unified multicast monitoring, traffic surveillance, and multicast fault detection and isolation system. It uses a graphical user interface to support ease of use and intuitive presentation of results. It utilizes IGMP queries and several Internet Engineering Task Force (IETF)-standardized multicast routing MIBs to collect information about tree topology and multicast-capable routers; information on individual multicast groups including traffic rates, identities, and location of sources and receivers; and multicast status of individual routers. In addition, mmon can perform MRM-based active monitoring tests. Mmon was developed in HP Labs and has been integrated in the HP OpenView network management platform.

MRMON [15] is a more recent system designed to capture, analyze, and present multicast session, traffic, and membership information in real time. MRMON defines and uses several MIB modules to passively collect and monitor multicast services in intradomain environments. More specifically, MRMON uses a *multicast statistics group* MIB to record multicast traffic information useful for configuration, fault, security, and performance management. Network operators can identify individual groups and their sources; measure traffic rates on per group and per source bases; and identify individual sources to filter or block. The *multicast history group* MIB stores information that can be used for debugging and performance monitoring purposes. MRMON can be used to observe long-term traffic distribution and service usage characteristics that can then be used for fault management and multicast service provisioning. MRMON uses IGMP reports to collect information about host multicast usage characteristics. The MRMON agent architecture is designed to work on a host placed within individual subnets in a domain and can receive and process SNMP Get and Set requests. Compared to mmon, MRMON defines and uses several new MIBs to collect a richer set of monitoring information to monitor and manage multicast services in an Internet service provider (ISP) or enterprise network environment.

MAFIA [16] is a multicast management solution with the specific aim of strengthening multicast security through multicast access control, multicast traffic filtering, and the prevention of denial-of-service (DoS) attacks. MAFIA achieves these tasks by using information about multicast group membership available at different locations in a network without requiring any changes to the network. The most important of these locations is at intradomain network boundaries. At this level, a MAFIA server would monitor and possibly limit outbound group join requests as well as rate control or filter inbound multicast data traffic. MAFIA can also have additional servers installed within specific networks to provide a finer granularity of network control. While solutions presented so far have been predominantly passive monitoring systems with some limited amount of proactive control, MAFIA is very much designed for active service management through access control and traffic monitoring.

Management Support for the Interdomain Use of IP Multicast

In this section we discuss several important additional management issues. The interdomain use of multicast faces two practical problems:

- The lack of a mechanism to verify multicast service availability between sources and receivers in a group
- The lack of a common application programming interface

In this section we present an overview of recent work in designing management tools and solutions to address these problems.

Monitoring the End-to-End Availability of Multicast

Whether multicast exists or not is an important and practical concern for applications attempting to use it. From a multicast receiver's point of view, end-to-end availability of multicast indicates that the host can join the multicast group and receive data from the source sending to the group. Due to several reasons, including local or interdomain connectivity problems, node failures, link failures, configuration errors, policy incompatibilities, and congestion-related persistent errors, multicast may not be available between sources and receivers. Since the current multicast routing protocols do not provide any feedback to group participants, there is no mechanism to communicate the lack of service availability to end users. This lack of robustness has serious consequences. If applications cannot rely on multicast to exist, how can it be used as a reliable basis for communication? Hence, there is a practical need for a mechanism to verify multicast service availability between receivers and sources. In the rest of this section we present an overview of a solution, called *mcping*, to implement the functionality necessary to allow applications and users to test the availability of multicast.

Ping is one of the most basic yet most useful network diagnostic tools used for network management purposes. In unicast, ping provides a convenient way to verify unicast service availability between two systems in a network (Fig. 4a). On the other hand, multicast ping (mping) requests are sent to a multicast group address, and these requests trigger group receivers to send ping responses to the pinging host via unicast (Fig. 4b). The returned information tells the pinging host which other hosts received the request and responded. Mping does not say who did not receive the request but should have. A real-world analogy is an instructor in a classroom asking the question, "If you cannot hear me, raise your hand." This information is not useful for debugging and there exist other problems, like feedback implosion. As a result, compared to unicast ping, mping does not really help verify multicast availability between a local receiver and a remote source.

The above discussion suggests that there is a need for a tool truly analogous to unicast ping. Mcping has been recently proposed to address this need [17]. Mcping is designed for a multicast user to test the availability of multicast ser-

Since the current multicast routing protocols do not provide any feedback to group participants, there is no mechanism to communicate the lack of service availability to end-users. This lack of robustness has serious consequences.

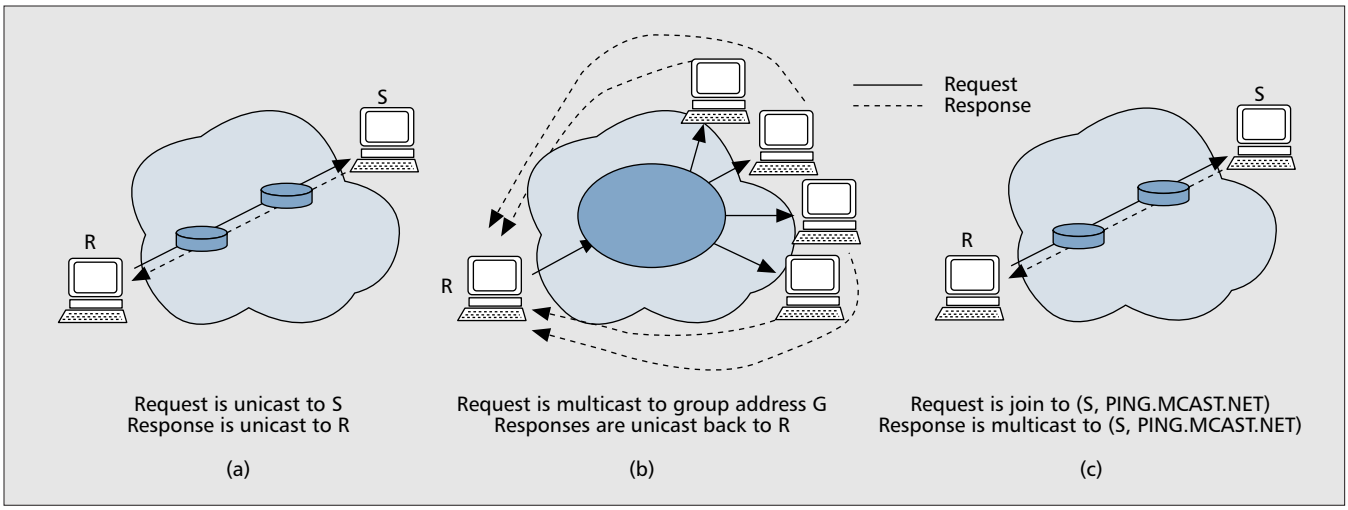

Figure 4. *Comparison of a)* `ping`*, b)* `mping`*, and c)* `mcping`.

vice between its site as a receiver and a specified remote site as a sender. In `mcping`, a positive response to an `mcping` request indicates that a local host can successfully join and receive multicast data from a remote host. By using a dedicated multicast group address, PING.MCAST.NET, in the SSM address range (232/8), a host, R, sends an `mcping` request to a remote host, S, and expects to receive an `mcping` reply on the (S, PING.MCAST.NET) multicast channel. Since the overall mechanism uses the existing multicast service architecture between the two hosts, the result of the test gives a definitive answer about the availability of multicast between the two hosts.

The proposed `mcping` mechanism works as follows (Fig. 4c). `Mcping` first sends an IGMP join request on the multicast channel (S, PING.MCAST.NET). Upon receiving this message, the designated router (DR) at the pinging site creates a PIM join message for (S, PING.MCAST.NET) and forwards it toward the pinged host, S. Each router on the R-to-S reverse shortest path creates a forwarding entry for the multicast channel (S, PING.MCAST.NET) and forwards the join message toward S. When the join request reaches the DR at S's subnet, the local router forwards a message to S informing it about the `mcping` request. On receiving the `mcping` request, S creates a reply message and sends it to the (S, PING.MCAST.NET) multicast channel. This message propagates on the multicast forwarding path and reaches the pinging host, R. During the operation, any problem that prevents the PIM-Join message from reaching S's site or the `mcping` reply from reaching R's site indicates the lack of service. As a result, `mcping` provides R with the ability to test the availability of multicast to a remote host, S. The mechanism does not depend on any other application, and it does not require any user intervention or interaction.

`Mcping` can be used to test the availability of service from a local site to a remote site. From a network administrator point of view, it helps detect potential multicast problems and fix them before a multicast event. From an end-user point of view, it helps to test the availability of multicast and consider alternative communication mechanisms (e.g., unicast or application layer multicast) when the service is not available end-to-end. `Mcping` requires pinged end systems to be modified to support this functionality, and a working implementation of `mcping` can be found at http://www.venaas.no/multicast/ssmping/. Finally, contrary to `ping` and `mping`, `mcping` cannot be used in launching flooding-based denial-of-service attacks on third-party Internet sites.

Facilitating Robust Multicast Group Management

A key problem in multicast is the fact that from an application point of view, there is no feedback on whether a group has been successfully joined. Normally, applications open a multicast socket. This operation then initiates a join process. However, if the join fails, no feedback is returned to the application. This lack of feedback is particularly problematic given that the multicast join process is too fragile and prone to failure at so many places in the network. This creates serious problems for applications wishing to rely on multicast for data communication.

A recently proposed solution, called the Multicast Detective [18], attempts to develop a robust solution to determine the existence of multicast in the network. Not only does the Multicast Detective attempt to determine if multicast exists, but it tries to determine what kind of multicast is available. For example, only ASM could be available, only SSM, or a combination of the two. The Multicast Detective tool has been developed considering two options:

- Using existing protocol features to extract information from the network about what kind of multicast is supported
- Introducing new protocol extensions to query the network and obtain the information directly

The two different options have been considered because while a truly robust group join is possible only with additional network support, changes to the infrastructure make deployment more challenging.

Without making changes to the network, the Multicast Detective cannot determine whether a particular group join request has succeeded, but it can determine whether such a request is likely to succeed. The difference is that instead of getting feedback about a specific request, the Multicast Detective can attempt to trace the set of steps that will be performed and determine if any are likely to fail. The Multicast Detective follows a series of steps to determine which parts of the multicast join process are likely to function correctly. These steps are:

- The Multicast Detective joins a well-known group consistently transmitting traffic and then checks to see if any data packets are received from the group. If no data packets are received, the group join is assumed to have failed. This is a good indication that there is no multicast connectivity.
- The Multicast Detective issues an SSM join request and listens for data packets. If the join fails, the host IP stack does not support SSM. This is a common problem while we wait for operating system developers to add SSM support.
- The Multicast Detective sends ICMP ping messages to a variety of multicast groups in order to determine whether the first-hop router supports multicast or not. In addition, a response also indicates that switches along the path to the first-hop router are performing snooping correctly.
- In this step the Multicast Detective looks more closely at the message exchange for a group join request. The tool sends a membership message for an arbitrary group and then verifies the functional correctness of IGMP by examining the content of the response message(s).
- In the final step, the Multicast Detective sends an ICMP ping message to some of the multicast routing control groups in an attempt to determine if the first-hop multicast router has routing information to other multicast networks. If this information is returned, it is a good indication that multicast is working in at least the local domain.

These five steps cover most of the set of steps that need to occur for a host to successfully join a multicast group. Not only can these steps be integrated into an application to ensure proper operation of multicast, but network administrators can periodically run these steps from hosts located throughout their network.

Conclusions

In this article we have focused on service monitoring as one of the most important management functions for IP multicast. We have presented an overview of the recent work in multicast monitoring in three different dimensions: interdomain, intradomain, and end-user-level. An important conclusion we have reached from this study is the fact that even though we have sufficient systems for intradomain monitoring and management of IP multicast, there still exists a need for additional primitives and tools to help application developers to interact with the underlying multicast service to make the most effective use of it.

References

[1] K. Almeroth, "The Evolution of Multicast: From the MBone to Interdomain Multicast to Internet2 Deployment," *IEEE Network*, vol. 14, Jan./Feb. 2000, pp. 10–20.
[2] C. Diot *et al.*, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network*, vol. 14, Jan./Feb. 2000, pp. 78–88.
[3] S. Deering, "Host Extensions for IP Multicasting," IETF RFC 1112, Aug. 1989.
[4] D. Estrin *et al.*, "Protocol Independent Multicast Sparse-mode (PIM-SM): Protocol Specification." IETF RFC 2362, June 1998.
[5] T. Bates *et al.*, "Multiprotocol Extensions for BGP-4," IETF RFC 2283, Feb. 1998.
[6] D. Meyer and B. Fenner, "Multicast Source Discovery Protocol (MSDP)." IETF RFC 3618, Oct. 2003.
[7] H. Holbrook and D. Cheriton, "IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications," *ACM Sigcomm*, Cambridge, MA, Aug. 1999.
[8] K. Sarac and K. Almeroth, "Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring," *J. High Speed Networking*, Special Issue on QoS for Multimedia on the Internet, vol. 9, no. 3, 4, 2000.
[9] K. Sarac and K. Almeroth, "Application Layer Reachability Monitoring for IP Multicast," *Elsevier Comp. Net.*, vol. 48, June 2005, pp. 195–213.
[10] P. Rajvaidya and K. Almeroth, "Analysis of Routing Characteristics in the Multicast Infrastructure," *IEEE INFOCOM*, San Francisco, CA, Apr. 2003, pp. 1532–42.
[11] E. Al-Shaer and Y. Tang, "SMRM: SNMP-Based Multicast Reachability Monitoring," *IEEE/IFIP NOMS*, Florence, Italy, Apr. 2002, pp. 467–82.
[12] H. Sallay, R. State, and O. Festor, "A Distributed Management Platform for Integrated Multicast Monitoring," *IEEE/IFIP NOMS*, Florence, Italy, Apr. 2002.
[13] J. Chesterfield, B. Fenner, and L. Breslau, "Remote Multicast Monitoring Using the RTP MIB," *IFIP/IEEE Int'l. Conf. Mgmt. of Multimedia Nets. and Svcs.*, Santa Barbara, CA, Oct. 2002.
[14] P. Sharma, E. Perry, and R. Malpani, "IP Multicast Operational Network Management: Design, Challenges, and Experiences," *IEEE Network*, vol. 17, Mar./Apr. 2003, pp. 49–55.
[15] E. Al-Shaer and Y. Tang, "MRMON: Multicast Remote Monitoring," *IEEE/IFIP NOMS*, Seoul, Korea, Apr. 2004, pp. 585–98.
[16] K. Ramachandran and K. Almeroth, "MAFIA: Multicast Management Solution for Access Control and Traffic Filtering," *IFIP/IEEE Int'l. Conf. Mgmt. of Multimedia Nets. and Svcs.*, Belfast, N. Ireland, Sept. 2003.
[17] P. Namburi, K. Sarac, and K. Almeroth, "Practical Utilities for Monitoring Multicast Service Availability," *Comp. Commun.*, Fall 2005.
[18] A. Mazumder, K. Almeroth, and K. Sarac, "Facilitating Robust Multicast Group Management," *NOSSDAV*, Skamania, WA, June 2005.

Biographies

KAMIL SARAC [M] (ksarac@utdallas.edu) received his M.S. and Ph.D. degrees in computer science from the University of California at Santa Barbara in 1997 and 2002, respectively. He is currently an assistant professor in the Department of Computer Science at the University of Texas at Dallas. His research interests include computer networks and protocols; group communication, including IP multicast and overlay networks; and management and security of computer networks. He has co-chaired the Computer Networks special track at ACM SAC 2004, and served as a reviewer for a number of conferences and journals. He is a member of the ACM.

KEVIN ALMEROTH [SM] (almeroth@cs.ucsb.edu) is a professor in the Department of Computer Science at the University of California at Santa Barbara. His research interests include computer networks and protocols, wireless networking, multicast communication, large-scale multimedia systems, and performance evaluation. He is chair of the Steering Committee for the ACM Network and System Support for Digital Audio and Video workshop; on the Editorial Board of *IEEE/ACM Transactions on Networking*, *IEEE Network*, and *ACM Computers in Entertainment*; has co-chaired a number of conferences and workshops, including IEEE ICNP, the IFIP/IEEE International Conference on Management of Multimedia Networks and Services, the Network Group Communication workshop, and the Global Internet Symposium.

Even though we have sufficient systems for intra-domain monitoring and management of IP multicast, there still exists a need for additional primitives and tools to help application developers to interact with the underlying multicast service to make the most effective use of it.