Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 48 (2005) 195–213
Computer Networks
www.elsevier.com/locate/comnet

Application layer reachability monitoring for IP multicast

Kamil Sarac [a,*], Kevin C. Almeroth [b]

[a] Department of Computer Science, University of Texas at Dallas, 2601 N. Floyd Rd., Richardson, TX 75083, USA
[b] Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA 93106, USA

Received 11 December 2003; received in revised form 9 October 2004; accepted 1 November 2004
Available online 8 December 2004

Responsible Editor: I. Nikolaidis

Abstract

The successful deployment of multicast in the Internet requires the availability of good network management solutions. One of the first management tasks for multicast is to verify its availability in the network. This task is usually referred to as *reachability monitoring*. Reachability ensures that sources can reach all existing and potential group members. Reachability also implies that receivers have multicast connectivity and can reach all sources. As a result, verifying reachability becomes very important to maintain availability and robustness of the multicast service between sources and receivers.

In this paper, we present an application layer mechanism to monitor multicast reachability. First, we justify the need for reachability monitoring systems. Then, we present our monitoring system called *sdr-monitor*. *Sdr-monitor* leverages an existing application and provides close to real-time reachability monitoring for the multicast infrastructure. It is the first system that is developed and deployed for monitoring multicast reachability. We present the architecture of the system and then discuss its operation. Finally, we include our evaluations on a data set that we collected using this system. With this analysis, we present long term reachability characteristics of multicast infrastructure during a four year monitoring period between 1999 and 2003 and discuss potential causes for reachability problems.

Keywords: Multicast monitoring; Multicast management; Reachability; Sdr

1. Introduction

The successful deployment of multicast in the Internet requires the availability of good network monitoring and management solutions. Most of the work in multicast has been on developing necessary protocols [1]; deploying them in the Internet

* Corresponding author. Tel.: +1 972 883 2337.
E-mail addresses: ksarac@utdallas.edu (K. Sarac), almeroth@cs.ucsb.edu (K.C. Almeroth).

1389-1286/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2004.11.004

[2]; and providing a number of additional services on top of the infrastructure including reliability [3], security [4,5], and congestion control [6]. In order to reach global deployment, these efforts need to be complemented by developing necessary monitoring and management systems.

One of the first management tasks for multicast is to verify its availability to participating users. This task is referred to as *reachability monitoring*. Reachability ensures that sources can reach all existing and potential group members. Reachability also implies that receivers have multicast connectivity and can reach all sources.

Multicast is realized through the creation and maintenance of forwarding trees connecting sources and receivers in a multicast group. These trees are dynamically created and maintained by the routers, yet there is no feedback information built into the process. If a tree cannot be built because there is no path to the source, the receiver will never know. Local connectivity problems, inter-domain connectivity problems, link and/or node failures, and congestion related persistent data loss are potential reasons that contribute to reachability problems. Consequently, verifying reachability becomes very important to maintain availability and robustness of the multicast service between sources and receivers. Without this assurance, the multicast infrastructure may become disconnected and essentially unusable.

In this paper we present an application layer reachability monitoring system called *sdr-monitor*. With the deployment of native multicast in the inter-domain, the multicast community realized the need for a mechanism to monitor reachability as well as the quality of the multicast service in the Internet. Prior to *sdr-monitor*, there were no mechanisms for multicast users to automatically learn the reachability of their multicast data at receiver sites. When a multicast user wanted to have a multicast event, he/she would contact his/her friends in remote domains and ask them to join the group to verify reachability. Clearly, this mechanism is not scalable and can only give a very limited view for reachability. With *sdr-monitor*, we present the first mechanism to automate this operation and increase the scope of the monitoring region from one single remote site to potentially overall multicast infrastructure.

The development of *sdr-monitor* presents some unique architectural tradeoffs. On one hand, there was an immediate need for a system that would help multicast users to monitor multicast reachability in the inter-domain scale. On the other hand, a fully functional monitoring system that would support various types of monitoring services (including performance monitoring for multicast data transfer) required extensive design, development, and deployment efforts. Given the urgency of the problem, our solution favored the first direction and we designed *sdr-monitor* as an easy-to-develop and easy-to-deploy monitoring system.

Sdr-monitor is based on multicast session announcements exchanged by multicast users over a well-known session announcement channel, SAP.MCAST.NET. Using a session directory tool, called *sdr*, multicast users announce the availability of multicast audio, video, and/or text sessions on the SAP.MCAST.NET channel. *Sdr-monitor* has a number of participants and a centralized data collection site. Participants listen to the periodic session announcements sent by others and report the announcements seen at their local site to the *sdr-monitor* site. A manager program at the *sdr-monitor* site then processes these reports and builds a real-time web page displaying a reachability matrix for the global multicast infrastructure.

Being the first system for reachability monitoring, *sdr-monitor* provided a basic mechanism for multicast users to monitor reachability of their multicast data to remote domains. *Sdr-monitor* also motivated a number of additional research projects in the area including MRM [7], SMRM [8], MRMON [9], Multicast Beacon [10], RMP-Mon [11], and Mantra [12].

The remainder of this paper is organized as follows. In the next section, we motivate the importance of multicast monitoring. In Section 3, we present the *sdr-monitor* architecture, its components and the outputs it generates. In Section 4, we analyze long term reachability characteristics of the multicast infrastructure. In Section 5, by using additional network layer information, we classify reachability problems into two groups. In Section 6, we present the related work and the paper is concluded in Section 7.

2. Motivation

The ability to establish, monitor and maintain multicast reachability is an important requirement in today's hierarchical multicast infrastructure. For a globally-scoped application, a number of potential receivers may be located in other domains and the availability of data to these receivers may be affected by reachability. Different applications will be affected differently by multicast reachability problems. Network operators must have the ability to ensure multicast reachability to all potential receivers.

Multicast service is realized by running a set of protocols in the network. First, we use a protocol to construct a multicast forwarding tree connecting sources and receivers in a multicast group. Currently, Protocol Independent Multicast-Sparse Mode (PIM-SM) [13] is the most widely used protocol for multicast tree construction in the Internet. In addition, in order to provide inter-domain multicast service, we use Multiprotocol Border Gateway Protocol (MBGP) [14] to communicate multicast path availability, and Multicast Source Discovery Protocol (MSDP) [15] to communicate multicast source availability among different domains in the network. Finally, the Internet Group Management Protocol (IGMP) [16] is used by end-hosts to dynamically join and leave multicast groups. As a result, the success of multicast service in the Internet requires successful inter-operation of these protocols.

Soft-state based multicast applications are good examples that are particularly susceptible to reachability problems. A general characteristics of soft-state protocols is that sources periodically transmit refresh messages to one or more number of receivers over lossy communication channels [17]. On the other hand, receivers keep these refresh messages for a finite amount of time. If a receiver does not receive any refresh messages during a timeout period, it removes the state from its cache/memory. This behavior of soft-state protocols have an important implication for soft-state based multicast applications. In multicast, sources and receivers may not know existence of each other. That is, sources do not get any feedback from the receivers (to avoid implosion) and receivers assume no source in the absence of update messages (to avoid connection establishment complexities, etc.) In this situation, lack of update messages at a receiver site may be because of some type of reachability problems or it may be due to an in-active source. But the soft-state nature of the application makes the problem hard to detect and hard to isolate.

Multicast session announcements are a good example of a soft-state based multicast service that is affected by reachability problems. Before having a multimedia session, information is announced to receivers including what the session is about, media types, bandwidth, duration, etc. One of the announcement techniques that has been used since the original MBone is to send this information to a well-known multicast address [18]. This session announcement method is based on the soft-state concept. The person announcing the session does not know who receives the announcement. Furthermore, if some users do not receive the session announcement because of some reachability problems, they will never know that such a session existed. Tools need to exist to give session announcers confidence that the session is reaching most (if not all) potential receivers. Potential receivers need confidence that they are being informed of most (if not all) existing sessions.

3. *Sdr-monitor*: a global session monitoring tool

Sdr-monitor has been developed to monitor reachability in the global multicast infrastructure. In an ideal case, monitoring reachability in a global scale requires sources and receivers in all different domains to work together to collect this information. Even though it is difficult to achieve this ideal coverage, we have attempted to involve as many sites as possible in our study. In this section we first present the design tradeoffs. Then, we describe the *sdr*-based multicast session announcement mechanism and present the *sdr-monitor* architecture. Next, we describe the outputs generated by *sdr-monitor*. Finally, we present an evaluation of the *sdr-monitor* tool and point out potential improvements.

3.1. Design tradeoffs

Sdr-monitor was designed to be an inter-domain scale multicast reachability monitoring system. One main challenge in designing and deploying large scale monitoring systems is the difficulties in wide area deployment. Monitoring multicast reachability involves using agents that may function as multicast data source and/or sink. In both cases, the agents introduce additional monitoring traffic in the network. Due to this additional traffic overhead, it is generally difficult to achieve a wide area (i.e., inter-domain scale) deployment. First of all, due to privacy and performance reasons, it is extremely difficult to use network entities (i.e., routers) in remote domains for this type of monitoring [19]. Therefore, monitoring systems can only use end hosts in remote domains for their monitoring tasks. This requires finding a large number of end users in remote domains to participate in the monitoring effort. Experience [20] shows that people are usually reluctant to participate in such monitoring efforts due to the introduced processing and network overhead at/around their systems/networks.

From this perspective, *sdr-monitor* presents unique characteristics in that it introduces minimum overhead on the monitoring agents. This is achieved by leveraging an existing application for our monitoring purposes. As we will see in more detail in this section, in *sdr-monitor*, monitoring agents include (1) multicast users that use the *sdr* tool to announce their sessions and (2) *sdr-monitor* participants who run our script to send us the announcements collected at their sites. Our monitoring does not introduce any overhead on the session announcing sites. In addition, it only requires the participants to run our script once an hour to send an e-mail to the *sdr-monitor* site. This is the only monitoring overhead that we incur on the agent sites. We believe that it was mainly this characteristic of the *sdr-monitor* system that enabled us to attract around 120 participants during our monitoring efforts.

Our design above trades functionality for ease-of-development and ease-of-deployment. That is, the *sdr-monitor* system mainly focuses on reporting basic reachability information between a multicast source site and a large number of diversely located receiver sites in a binary form (i.e., reachable or not-reachable). It is not designed to monitor multicast data reception quality (i.e., loss and/or jitter information) between a given source and a given receiver. Please note that the latter monitoring task usually requires active monitoring support which limits the practical application domain of the monitoring task to intra-domain scenarios only. In addition, monitoring in the intra-domain is a relatively easier problem and there has been a number of systems developed for this purpose [8,9,21].

We believe that our design decisions that favored ease-of-development and ease-of-deployment enabled us to achieve a large scale deployment of our system and provided multicast community with an effective and convenient mechanism to monitor reachability of their multicast data.

3.2. Multicast session announcements and Sdr session directory tool

One mechanism to communicate session announcements in the network is to multicast them using the Session Announcement Protocol (SAP) [22]. In SAP, announcements are periodically sent to a well known multicast address (SAP.MCAST.NET) with a certain scope. SAP is a soft-state protocol in which reliability is achieved by periodically sending announcements. Acknowledgments are not used. Not every receiver is expected to receive every announcement every time it is sent, but enough should be received to build an accurate session list. From a reachability perspective, these SAP packets are a good source of one-way ping messages; sent from a widely scattered set of sources; and received by a potentially large number of receivers.

Sdr is the most commonly used tool for creating and communicating session announcements [23]. When a user wants to create an announcement entry, he/she uses the graphical user interface of *sdr* tool to provide necessary information for the entry. This information includes session name, multicast group addresses, media types, etc. *Sdr* then creates the entry using the Session Description Protocol (SDP) [24] and periodically announces it using SAP. In addition, *sdr* listens to the SAP address for announcements by other users. When an announcement is received, *sdr* caches the information and presents a continuously-updated list to the user. All the announcements that have been received within the previous hour are included in this list. To maintain robustness and keep its list up-to-date, *sdr* writes the current set of announcements to a cache directory periodically. This way, when a user starts *sdr*, the tool does not have to wait for new announcements to arrive from the network. Instead, it reads the available announcement entries from the cache, and uses them to populate its announcement list.

In addition to using SAP announcements as a heartbeat mechanism, *sdr* has a critical feature that enables us to easily collect feedback from remote participants. *Sdr* allows users to run customized code that executes when certain conditions occur. Each user puts its code into an "*sdr.tcl*" file. When *sdr* starts, it automatically reads the user-specified code and executes it. As we present in the Section 3.3, we use this mechanism as the basis of our multicast reachability monitoring task.

3.3. The sdr-monitor architecture

Sdr-based multicast session announcements provide a sufficient mechanism for reachability monitoring. *Sdr-monitor* uses available session announcements from topologically and geographically distributed sites to build a representation of the reachability status in the global multicast infrastructure. The *sdr-monitor* architecture includes the following components:

Session Announcement Originators. Any user that sends multicast session announcements on the SAP address (using *sdr* or any other tool) becomes a source for *sdr-monitor* heartbeat messages.

Sdr-Monitor Participants. Any *sdr* user can potentially be a part of our project. During our monitoring period, *sdr-monitor* had around 120 registered participants. On average, there were 25 active participants at a time. These participants use a *sender script* to deliver their *sdr* cache entries to the *sdr-monitor* collection site (see Fig. 1). This

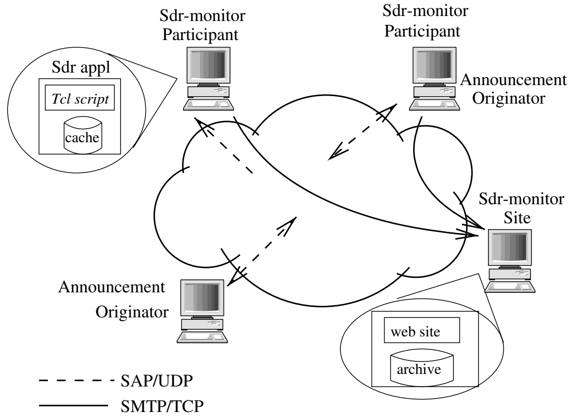

Fig. 1. The *sdr-monitor* architecture.

sender script is a small *Tcl* script that is appended to the *sdr.tcl* file. While *sdr* is running, the sender script runs parallel to *sdr*. Periodically, the sender script first forces *sdr* to write the current set of announcements to the cache directory and then sends these announcements to the *sdr-monitor* collection site via email. In order to limit the monitoring overhead at *sdr-monitor* participant sites, we set feedback period to one-hour. This mechanism provides a reliable method to collect the available announcements at remote sites. The email sent by the sender script also includes other useful information including a *sequence number*. This number is used to determine how long *sdr* has been running at the participant site.

Central Collection/Processing Site. At the *sdr-monitor* site, a manager receives emails from remote sites and processes them. The manager runs as a daemon process and periodically checks for incoming email messages. The manager uses these messages to generate a web page displaying a reachability matrix. The web page is continually updated as new information is received. In addition, the manager takes a snapshot of the reachability matrix every hour and archives it for long-term analysis. More details about each are described next.

3.4. Sdr-monitor outputs

Sdr-monitor produces two outputs: a real-time web page and an archival data set. The *sdr-monitor* web page displays the current view of global multicast reachability for all known global sessions for all *sdr-monitor* participants. The archival data set is a snapshot of this reachability taken once an hour and used for long term reachability analysis.

3.4.1. Sdr-monitor web page

The web page is used to give the multicast community a close to real-time picture of reachability in the multicast infrastructure. It consists of two parts: a session reachability matrix and a participant list. These two parts are further described as follows.

Session Reachability Matrix. The session matrix visualizes whether each globally announced session is known to each *sdr-monitor* participant. A snapshot of part of the matrix is shown in Fig. 2. The first column contains session information including name, time-to-live (TTL), IP address of the announcing host, and a time offset since the last time *sdr-monitor* received a report with this announcement in it. Each of the remaining columns corresponds to an active *sdr-monitor* participant. A white cell in a row means that the session announcement in this row is visible to the participant represented by the column. A black cell (red on the web page) means that the session announcement is not visible. Announcements on the matrix are sorted based on the number of current participants reporting these sessions. The most widely seen session is reported first.

Participant List. The participant list displays information about currently active *sdr-monitor* participants in a table. Each row in this table contains information about a participant including the email address, geographic location, IP address, and the number of global session advertisements seen and not seen. Entries in this table are sorted by the number of sessions visible to the participant. The participant seeing the most sessions is shown in the first column.

Assuming a large number of participants from diverse places around the world, the *sdr-monitor* web page displays the reachability status between a large number of networks. Because only globally scoped announcements are displayed on the web page, all participants should see all the announcements. By examining this real-time snapshot, the web page can be used to quickly detect reachability problems in the infrastructure. Over the course of this project we have become relatively adept at seeing patterns in the matrix. Some conclusions that can be drawn by looking at the web page include:

- A row with a single white cell indicates that the session announcement originator has local connectivity problems. Every row must have at least one white cell or otherwise *sdr-monitor* would not know about it. The one white cell for these types of sessions corresponds to either the session announcement originator or another participant close to it.

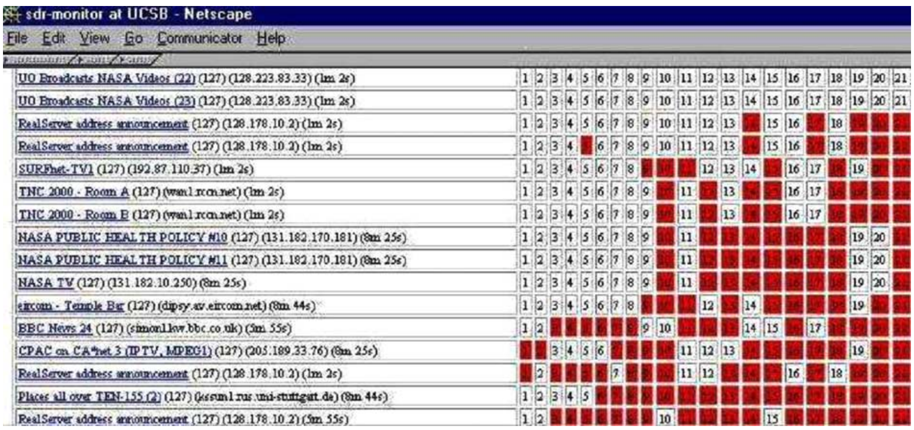

Fig. 2. A snapshot of the session reachability matrix from the *sdr-monitor* web page.

- A column with more than one but still only a few white cells is an indication of a local *reception* problem. If this site is also a sender, this result can be correlated with the appropriate row to determine if there are bi-directional reachability problems. However, we have frequently observed that connectivity is working in one direction, but not both. In most of these cases, sites experience reception problems.
- Because of the way the matrix is organized, white cells are concentrated in the upper-left corner and black/red cells are concentrated in the lower-right corner. If problems do occur, the reachability matrix will concentrate the negative results in the lower-right corner.
- One of the most interesting cases occurs when a group of white cells appears in a block of black/red or a group of black/red cells appears in a block of white cells. These cases may indicate potential connectivity problems within or between multicast capable domains. In general, since the multicast community works to ensure that the infrastructure is not split, these types of patterns should not occur. Therefore, this is likely to be an important error condition and should be correctable. However, understanding the actual causes of these problems require network layer monitoring and investigation and is currently left for future work. When conducting our analysis, we focus on quantifying and characterizing the duration of these types of events.
- For session announcement originators, if we knew the network they exist in and which networks are inter-domain peers, we could correlate black/red areas. This would allow us to identify peering problems between specific networks. Currently, we do this on an ad hoc basis. A future work in this direction is to incorporate the functionality into the web page automatically.

3.4.2. Archival data set

The archival data set contains information taken from the reachability matrix on a periodic basis. A snapshot of the reachability information contained in the web page is captured at 1 h intervals and stored for later use. Entries in the data set indicate which session announcements were received by which *sdr-monitor* participants. In the following section, we use this data to analyze long term reachability in the multicast infrastructure and quantify and characterize reachability problems.

3.5. Evaluation of sdr-monitor as a monitoring tool

As a monitoring tool, *sdr-monitor* has a number of areas that could be improved. In large part, many of the problems relate to the use of SAP as a heartbeat mechanism. These problems include:

- *Lack of flexible monitoring*. *Sdr-monitor* can only report reachability between sites that are advertising sessions and *sdr-monitor* participants. Furthermore, this reachability is in one direction only.
- *Lack of heartbeat message control*. *Sdr-monitor* cannot control the frequency of heartbeat messages sent by sources. Packets are sent periodically (approximately once every 5 min), and this may not be sufficient to establish the routing state necessary to measure reachability. Furthermore, periodic, single packet transmissions are not sufficient to give us a measure of the quality of the connections between sites.
- *Lack of consistent monitoring*. Because both announcement source sites and participants can come and go at will, the results can change dramatically even though overall reachability does not change significantly.

As we mentioned before, *sdr-monitor* is one of the first tools developed for inter-domain multicast reachability monitoring. The main goal of this system was to enable multicast users to monitor reachability of their multicast data as well as to help network administrators to identify and fix multicast related problems in their networks. Finally, some of the key findings that we have in this work are: (1) the overall reachability is generally poor and irregular, (2) if the reachability of a multicast site is divided into two parts as reachability-to-others and reachability-from-others, the reachability characteristics in these two directions are independent of each other, and (3) most of the reachability outages are short lived outages which suggest instability or existing of transient problems in the multicast infrastructure. The analysis presented in the rest of this paper will discuss some of these findings in more detail.

4. Reachability analysis

In this section, we present our analysis on four-year *sdr-monitor* archival data. *Sdr-monitor* was designed to help multicast users and network administrators to detect multicast reachability problems. It depends on the application layer data collected at a number of end points in the multicast infrastructure. From this point of view, it can report the existence of potential problems and their scope with respect to our monitoring coverage area. However, this information may not be enough to locate the exact problem spots or identify the causes/reasons of the problems. Problem isolation for multicast usually requires collecting additional network layer information. This has been addressed in a follow up project to *sdr-monitor* called Mantra [25]. Nevertheless, it was partly the analysis on an earlier set of *sdr-monitor* data set that helped multicast community to understand the reachability characteristics of the multicast service. This also motivated us as a community to pursue further studies in locating and identifying these problems. Therefore, we find it worthwhile to present our analysis on the collected data set to understand the long term reachability characteristics of the multicast infrastructure without necessarily looking at the causes/reasons of individual reachability problems. Historically, this has been the first step in attacking these problems and producing solutions to them.

In this section, we present a four-step analysis of *sdr-monitor* data. In the first step, data is processed to remove mis-formed and non-globally scoped *sdr* announcements. In the second step, we process the data further to remove artifacts caused by intermittent behaviors of *sdr* users, session announcements, and *sdr-monitor* participants. At the end of the second phase we hope to have eliminated all of the problems caused by using *sdr* as the underlying reachability mechanism. In the third step, we specifically focus on reachability problems and attempt to characterize their number and duration. Finally in the fourth step, we closely examine the reachability characteristics of a large number of session announcing sites and report our conclusions on them.

There are two types of reachability that could be considered: sender-to-receiver and receiver-to-sender. The session announcement mechanism used by *sdr* produces sender-to-receiver reachability information. Using *sdr*, we cannot monitor reachability in the reverse direction, i.e. receiver-to-sender reachability. Focusing only on source-to-receiver reachability, there are two perspectives that can be taken. They are:

- *Source-Based Reachability*. For each site announcing an *sdr* session, we compute the percentage of *sdr-monitor* participants who see announcements from that site. To calculate this, we count the number of *sdr-monitor* participants who see the announcement and divide it by the number of current *sdr-monitor* participants.
- *Receiver-Based Reachability*. For each *sdr-monitor* participant site, we compute the percentage of global sessions seen. We take the number of announcements seen by an *sdr-monitor* participant and divide it by the total number of currently announced global sessions.

The difference between the two is mostly semantic. Therefore, we only need to consider one type of reachability—source-based reachability.

4.1. Phase 0: data collection

Our analysis is based on a data set collected between April 1, 1999 and March 31, 2003. [1] During this time, as long as *sdr* was running at a participant site, our sender script (running in these sites) periodically packed the available session announcements into an email and sent it to the *sdr-monitor* collection site. Results reflect our esti-

[1] Due to an undetected problem, our system failed to archive reachability data between April 2002 and July 2002.

mate of what participants actually see at their remote site. However, this may not be the actual reachability at these sites. In the remainder of this section, we list problems we identified and how we processed the data set to remove those problems.

4.2. Phase 1: pre-processing

Our data set includes a number of entries that are not useful for global reachability monitoring. In general, either the data appears in the cache even though it is not being refreshed or the data is for a non-global session.

On receiving an announcement, *sdr* tool is expected to hold the announcement in its cache for 1 h. If no other refreshing announcements arrive within the next hour, the tool removes the announcement from its cache and forgets about it. Even though this is the default behavior of the *sdr* tool, early versions of the tool had a bug because of which they failed to remove such state announcements from their caches. In our preprocessing, we identified such state announcements and removed them from our data set.

In the second step, we filtered announcements with administrative scopes or with a TTL scope less than 127. These announcements are non-global announcements (i.e., announcements with TTL < 127 are considered to be non-global by *sdr* tool) and are not interesting for our analysis.

4.3. Phase 2: removing sdr artifacts

In the next phase, we deal with the artifacts of using *sdr* as the underlying mechanism for monitoring reachability. In particular, we must deal with the following problems:

Sdr-monitor Participant Behavior. In the data collection period, not all *sdr-monitor* participants were running *sdr* continuously. This means that not all participants were continuously reporting the sessions in their *sdr* caches. During the first three years, the number of active participants has been between 15 and 35 with average of 25 participants per hour and this number has dropped down to as low as 10 participants during the last one year. Since each participant has a potentially different picture of global reachability, their joining and leaving can cause dramatic changes in *sdr-monitor*'s results.

Behavior of Session Announcing Sites. Similar to the above problem, the number of sites *sourcing* session announcements is also dynamic. The number of sites sending announcements has been between 22 and 48 with an average of 35 sites per hour. The results show that sites frequently start and stop sending session announcements. Each time a site starts or stops advertising a session, it affects the perceived global reachability.

Reachability Changes at Announcement Start and End. When a site starts sending a session announcement, it takes some time until the announcement reaches all participants. During this startup period, the number of sites who immediately see a session will be relatively poor. It is not possible to take an accurate measure of reachability until all participant sites have had sufficient time to receive an announcement. Similarly, when a session announcing site stops advertising a session, inaccuracies can also occur. According to our analysis on a number of session announcement cases, it takes 2 h (two snapshot periods) for announcements to reach majority of the *sdr-monitor* participants (80% of the participants) and it takes 2 h (two snapshot periods) for announcements to be removed from the *sdr-monitor* data set when the source stops the announcements. Due to the coarse resolution of our snapshots in *sdr-monitor* system (hourly snapshots), this is in accordance with the expected behavior of the system. On the other hand, from the global reachability point of view, these cases should be removed from the data set in order not to mislead the results of our analysis. We first located the *sdr-monitor* archival snapshots for the beginning and end of each announcement session for the announcements that continued for more than 8 h. Then, we removed the information related to announcements from the first and the last two snapshots.

Short Lived Sessions. Due to reachability behavior at announcement start and end, sessions with a short lifetime particularly contribute to poor perceived reachability. Fig. 3 shows a breakdown of session announcement events of sites by the lifetime of their announcements. This figure shows that there are a lot of announcements with a very

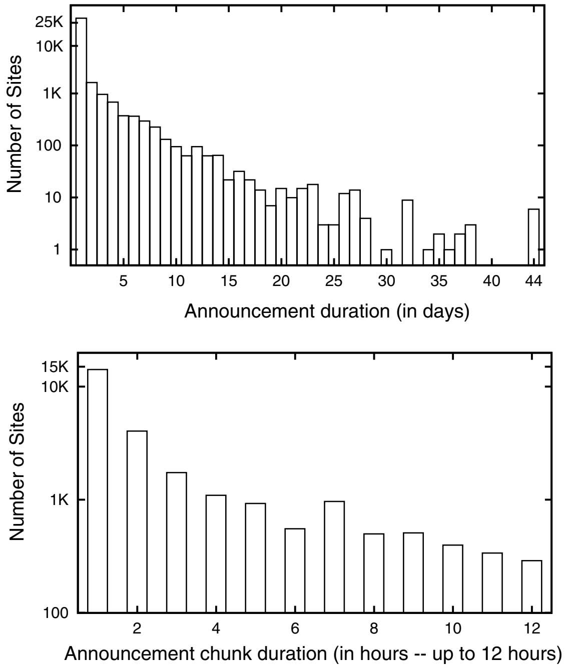

Fig. 3. Breakdown of global announcements based on lifetime.

short lifetime. These announcements contribute to poor perceived reachability because the announcement has started and ended before all *sdr-monitor* participants have had time to receive and cache the announcement.

Once we identified these types of problems, we filtered them out from the data set. Before presenting results after this processing, it is worthwhile to note that we consider reachability of announcement *sites* rather than that of individual announcements. Different sites are responsible for different numbers of session announcements. Some sites advertise as much as couple dozen sessions on the announcement channel SAP.MCAST.NET. However, we are only interested in reachability on a per-site basis and not per-announcement. That is, from an *sdr-monitor* participant point-of-view, receiving one or 10 announcements from the same announcement originator mean the same thing: there is multicast reachability from the originator site to the *sdr-monitor* participant site. Therefore, in order not to skew our results by arbitrarily weighting certain sites, we consider a site only once in our analysis.

For each session announcing site, we compute a daily average reachability. This is computed by averaging the reachability of sites for each day using our local time zoning (Pacific Standard Time). Reachability of a site is computed by dividing the number of participants receiving an announcement by the total number of active participants. We then divide announcing sites into four groups based on their daily average reachability. The four groups are: sites having reachability percentages of 0–25%, 26–50%, 51–75%, and 76–100%. Fig. 4 shows the breakdown of results over four year-long period. As an example, according to this figure, at the beginning of April 1999, 20% of announcement sites had less than 25%

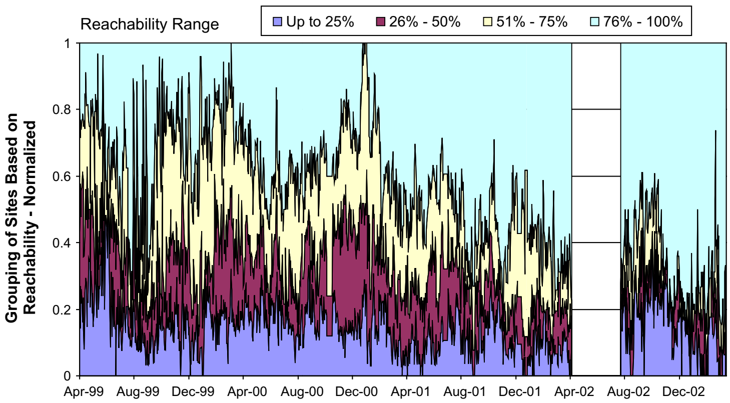

Fig. 4. Average reachability for session announcing sites: April 1999 to March 2003.

reachability; 55% of sites had less than 50% reachability and 75% of sites had less than 75% reachability. Noteworthy about these results are the following:

- Overall reachability during the first 2 years seems very poor then it improves gradually. There are a large percentage of announcing sites (approximately 30% during the first two years and 20% during the last 2 years) that send announcements seen by less than 50% of *sdr-monitor* participant sites.
- Reachability varies wildly. There are no distinctive trends and significant variability exists day-to-day.

4.4. Phase 3: frequency and duration of reachability problems

After Phase 2 processing we believe we have a data set that only includes end-to-end reachability problems. Our goal now is to analyze the frequency and duration of these problems.

The remaining analysis is based on characterizing a specific type of reachability problem. This analysis was conducted using the data set produced by Phase 2 processing. The specific event we are looking for can be described as follows: *an sdr-monitor participant site initially sees a session announcement and then does not; while at the same time other sdr-monitor participant sites continue to see the announcement.* This type of reachability problems occur only after an *sdr-monitor* participant first receives an announcement, and then does not. We call such events as *reachability loss events.* In order to compare the number of loss events to the total number of events we define a *successful reachability transition* event. This event occurs when a session announcement is seen by an *sdr-monitor* participant in two consecutive snapshots. By using these two types of events, we computed the percentage of loss events for each day during our monitoring to be around 5% (figure not shown). By reporting loss events as a percentage, we normalize the number of loss events over the number of participants and the number of source originating sites.

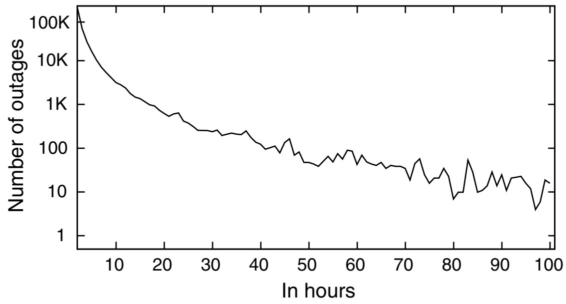

Fig. 5. Duration of reachability outages.

Having quantified the number of problems, we now attempt to characterize problems as short-lived or long-lived. Problems that lasted for only a short time partially contributed to the irregular reachability characteristics shown in Fig. 4. Our analysis consisted of first identifying all the cases in which an *sdr-monitor* participant saw a session, then did not see it, *and then saw it again*. If we were to use only reachability loss events, there would be cases when a session was seen and then never seen again. We would not be able to tell if the loss condition was permanent or it was a combination of a loss event and the end of a session. Fig. 5 shows a distribution of the reachability outages. The results, shown on a log-scale, exhibit characteristics of a heavy-tailed exponential distribution. Most reachability outages are short-lived. However, some outages lasted several days. Our own qualitative experience, based on continuously advertising

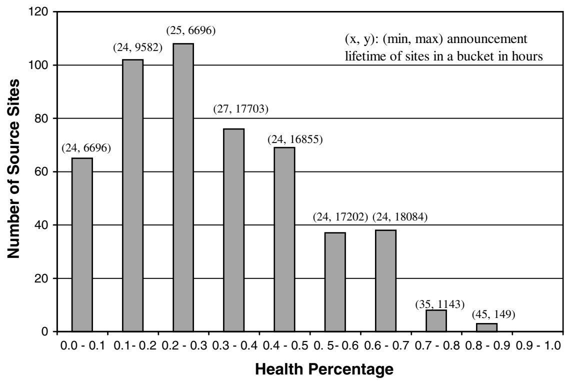

Fig. 6. Grouping of session announcing sites based on their health.

the Interactive Multimedia Jukebox (IMJ) sessions, suggests that outages can even last for weeks at a time.

We use the reachability characteristics of session announcing sites to analyze reachability characteristics for the global multicast infrastructure. In this part of our analysis, we classify session announcing sites based on their average reachability (V_{avg}) and their non-outage rates ($R_{n/o}$). Average reachability for a site is the average of its reachability ratios during its lifetime. The non-outage ratio for a site is the ratio of the number of time intervals without a reachability loss event to its lifetime. We define *health* of a site as the product of its average reachability and its non-outage ratio. A site with very good reachability and a high non-outage ratio will have a product close to one and is considered a healthy site. On the other hand, sites with poor reachability and/or low non-outage ratio will be unhealthy. Fig. 6 shows a grouping of sites based on their health. In this figure we only consider sites with a cumulative lifetime (L_{cum}) of more than a day. According to the figure, a majority of sites are not healthy (health <0.3). Most of the unhealthy sites are unhealthy because of a low average reachability. Only a few sites are unhealthy because of a poor non-outage ratio. A majority of the sites with relatively good health (over 0.6) are the ones with a relatively short lifetime (with a few exceptions). Popular/frequent session announcing sites have only average health. Table 1 shows the health ratios for the 10 most active session announcing sites.

Table 1
Health of the 10 most active session announcing sites

Announcement site	L_{cum}	V_{avg}	$R_{n/o}$	Health
University of Oregon	29421	0.764	0.880	0.672
ENST, (FR)	17703	0.392	0.809	0.318
Lulea University (SE)	17202	0.651	0.823	0.536
NASA, California	16855	0.559	0.853	0.476
UCSB	14703	0.707	0.774	0.547
CANARIE INC (CA)	14524	0.472	0.796	0.376
CISCO	10076	0.615	0.717	0.441
CRC (CA)	9594	0.506	0.753	0.381
George Mason University	9582	0.207	0.855	0.177
MulticastTech.com	8804	0.697	0.926	0.645

4.5. Phase 4: a closer look at individual sites

In this part of the analysis, we study the reachability characteristics of individual session announcement sites during a variety of time periods. Our focus in this analysis is to step through some interesting or abnormal cases to better understand what exactly is happening during a reachability outage. In all, we studied 50 cases. Each case corresponds to a session announcing site sending out continuous announcements during some time frame and the *sdr-monitor* site receiving continuous feedback information from at least 15–20 participants during this time. 28 of these cases correspond to session announcements from senders located in the United States or Canada and 22 correspond to announcements from senders in Europe. The duration of the announcements ranges from 122 h to 1035 h with an average of 516 h.

In this analysis, we computed two different hourly reachability values for the sender (session announcement) sites: one with respect to US receivers and the other with respect to European receivers. Then, we computed three-hour average, daily average, and overall average reachability val-

Table 2
Reachability performance for US senders

European reachability (%)	US reachability (%) Good (>85)	Fair (85–60)	Poor (<60)
Good (>85)	15	1	0
Fair (85–60)	6	2	0
Poor (<60)	1	0	3
Total US senders	22	3	3

Table 3
Reachability performance for European senders

US reachability (%)	European reachability (%) Good (>85)	Fair (85–60)	Poor (<60)
Good (>85)	8	1	3
Fair (85–60)	2	1	1
Poor (<60)	3	1	2
Total European senders	13	3	6

ues for each sender site. Tables 2 and 3 summarize our findings for US and European senders respectively. In these tables, we group sender sites based on their overall reachability characteristics with respect to US and European receivers. As an example, the bottom row in Table 2 indicates that out of 28 sender sites located in the US, 22 had good reachability with respect to US receivers, 3 had average, and 3 had poor reachability with respect to US receivers. In addition, the second column in the same table indicates that out of the 22 US senders with good reachability in US, 15 also had good reachability with respect to European receivers, 6 had average, and only one had poor reachability with respect to European receivers.

One observation from the above tables is that the intra-continental reachability for US senders is better than that of European senders. Another interesting result is that when the intra-continental reachability is poor for a US sender, the inter-continental reachability (with respect to European receivers) is also poor. However, this is not necessarily the case for European senders. There are several cases where the intra-continental reachability for a European sender is poor while the inter-continental reachability (with respect to US receivers) is good or fair. The tables also depict that the reachability is more unstable in Europe than in the US. One potential reason for this behavior is the fact that during our monitoring time some of the European sites were connected to each other via a connection that goes through the US.

In the rest of this subsection, we present results for three different cases as examples for reachability. Fig. 7a and b present hourly reachability of a US sender site (a host at Georgia Tech) for 846 h starting at 21:40 on Jan 13, 2001, with respect to US and European receivers respectively. According to the first figure, the reachability with respect to US receivers is quite good for the first 672 h. Then on Feb 10, 2001, it suddenly drops down to a 10% reachability level. According to our data, the 10% reachability corresponds to one US receiver that is an *sdr-monitor* participant located in Georgia Tech. On the other hand, according to the next figure, initially, European reachability was fair but degraded slowly. Then, starting from Feb 2, 2001 it improved significantly and stayed at 100% for around 192 h. Finally on Feb 10, 2001, it went down to 0% reachability. This case clearly shows that on Feb 10 a local connectivity problem occurred and the sender at Georgia Tech lost its connectivity to the outside world.

The second case is about a US sender (a Real.com server) with an unstable reachability pattern with respect to European receivers. Fig. 8a and b presents the hourly reachability of this sender with respect to US and European receivers between Oct 4 and Oct 19, 2001. According to the figures, the US reachability is quite good for the announcement duration. However, the European reachability has significant instability. The number of European receivers represented in this figure ranges between 5 and 8. The figure suggest that there were periodic reachability problems between the sender site and a number of receivers in Europe. A close examination of this behavior shows that this has been the case for the three individual receivers in Europe that were having alternating reachability behavior to this sender site. We believe that these reachability problems are caused by network congestion and/or multicast connectivity problems between the continents.

The final case is a European sender (a server at Lulea University in Sweden) with an interesting reachability pattern with respect to European and US receivers. Fig. 9a and b present the hourly reachability results. These figures correspond to close to 15 months of reachability data for this site. This case is of interest because for almost 100 days the reachability with respect to European receivers was nearly 0% while the reachability with respect to US receivers was fairly good. This is somewhat counter intuitive. We expect that sites within the same continent have better network connectivity to each other. From this perspective reachability among European senders and European receivers should be better than reachability among European senders and US receivers. However this particular European sender site as well as a number of others reported in Table 3 suggest that this is not necessarily so. As we mentioned above, this is partly because some of the European sites have direct connections to the US.

We end this section with some qualitative conclusions about the causes of these reachability problems. These include:

Local Connectivity Problems at Participant Sites. During the data collection period, we observed cases in which some participants reported only the announcements that were local to them. However, the data suggests that local problems are not permanent. When these local problems are solved and reoccur they create a significant number of reachability loss events. Our belief is that local connectivity problems occur frequently for some sites. For these sites, multicast is a relatively unstable service. Over time, sites become more experienced at correctly configuring the network and so multicast becomes more stable.

Inter-domain Connectivity/Peering Problems. Another observation is that a number of announcements are only reported by one or a few number of non-local participants. In these cases, announcement originating sites and *sdr-monitor* participant sites may not be on the same local network, but are topologically close to each other—likely within the same autonomous system (AS). Reachability problems to other domains can be linked either to inter-domain peering mis-configurations or more fundamental protocol problems. The limitations of the Multicast Source Discovery Protocol (MSDP) [15] is an example of a possible source of problems.

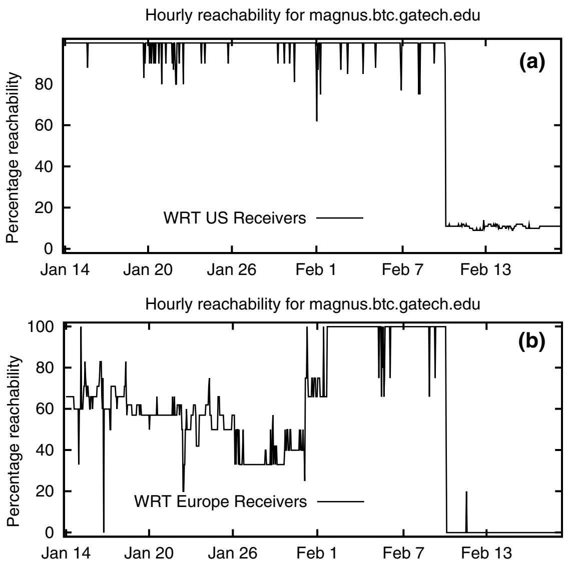

Fig. 7. A US sender located at Georgia Tech.

So far, we used our monitoring data to present the long term reachability characteristics of the multicast infrastructure. This information is collected at the application layer from the network end points. In the Section 5, we use additional information (network layer multicast path information) to identify potential reasons for reachability problems.

5. Classification of reachability problems

In Section 4.5, we presented a number of potential reasons for reachability problems. These analyses are based on application layer information collected by *sdr-monitor*. In this section, we use network layer monitoring information to classify reachability problems into two groups: multicast connectivity problems and other problems. For this, we use multicast path information collected from the network using a multicast version of the traceroute tool called *mtrace* [26]. In the rest of this section, we first briefly describe how *mtrace* works and then present our analysis.

5.1. Mtrace

Mtrace is a multicast version of the *traceroute* utility [26]. It is used to discover the multicast path between a given receiver and a source in a multicast group. The trace starts at the receiver site and works in the reverse direction toward the source site. On receiving an mtrace query, the last hop router at a receiver site starts the trace on the reverse path toward the source site. Each router on the path appends its response block to the request packet and forwards it to the next upstream router on the way to the source. When the request packet reaches the first hop router at the source site, it contains the complete path information. This information is then sent to the query originator via unicast. Mtrace allows users to run third party mtraces, i.e. the mtrace initiator need not be the source or the destination. In such a case, in order to start the trace, the mtrace initiator needs to reach the last hop router at the receiver site which in turn requires the initiator to know the IP address of the last hop router at the receiver site. This can be done by running an mtrace from the initiator site toward the receiver site. However, if this mtrace is not successful, then the initiator may not be able to start the actual trace.

5.2. Mtrace-based problem classification

As we mentioned previously, multicast depends on proper operation of several different protocols including PIM-SM, MBGP and MSDP. MBGP is used to communicate multicast path availability between multicast enabled domains. It is responsible for making sure that the global multicast infrastructure is connected and there exists a valid path between any two end points in the network. On the other hand, MSDP is used to communicate the addresses of active multicast sources to potential receivers in remote domains. This information is then used by receivers to join and receive data from these remote sources. Finally, PIM-SM is used to create multicast forwarding trees between sources and receivers.

Based on this protocol architecture, we can group the reachability problems that we observe at the application layer as follows:

(1) *Multicast connectivity problems*. This refers to the lack of multicast connectivity between the source site and the receiver sites in a multicast group. These problems are most likely MBGP problems. That is, MBGP does not provide a valid multicast path between the source domain and the receiver domain. When a receiver joins a source group, the join message cannot make its way to the source.

(2) *Non-connectivity related problems*. This refers to the case where there exist multicast connectivity between source and receiver domains but the receiver cannot get the source data or may not even know about the existence of the active source. This type of problem may have several causes including: (1) MSDP problems where active source information cannot be communicated to other domains, (2) policy and/or administrative issues where a network may be configured to block multicast data coming from a certain domain or source, or (3) multicast tree construction and maintenance problems due to buggy implementation or mis-behaving protocol functionality in routers [27] (early dropping

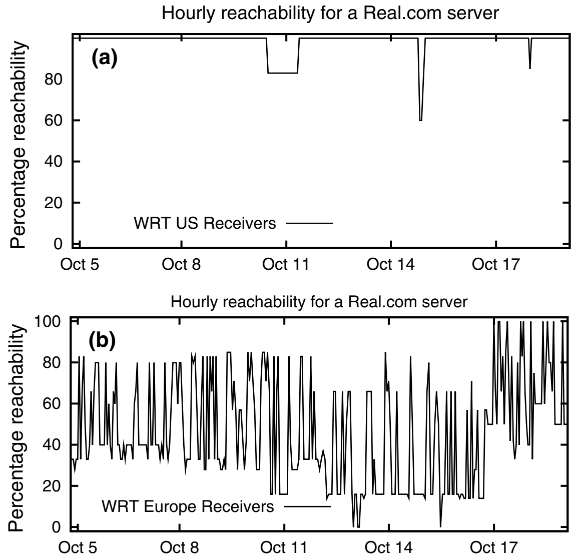

Fig. 8. A US sender from Seattle.

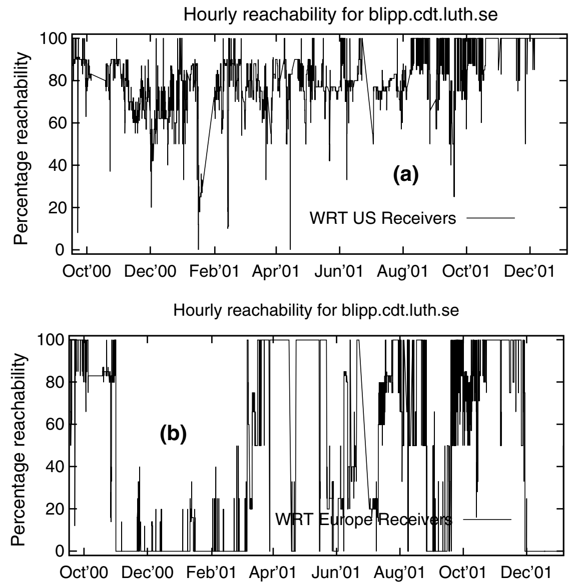

Fig. 9. A European sender from Sweden.

of forwarding state in routers, etc.). Please note that having network layer multicast connectivity (i.e., availability of a join path) from a receiver site to a sender/source site does not always guarantee reachability (i.e., successful reception of source data by the receiver).

At this point, we use *mtrace* to divide reachability problems into these two groups. Our reasoning is that if *mtrace* returns a valid path between a source and a receiver, multicast connectivity between the two sites does in fact exist. However, if mtrace does not return a valid path, we conclude that there is a multicast connectivity problem.

During our monitoring effort, we ran a total of 74,424 mtraces between session originating sites and *sdr-monitor* participant sites. Out of these traces, 73,128 were third party mtraces and only 1296 were between our local site (ucsb.edu) and 164 unique remote sites. We use the latter set of mtraces (1296 traces) to classify multicast problems into connectivity and non-connectivity problems. The reason why we do not use the third party mtraces for our analysis is that most of the time these traces were unsuccessful because we were not able to reach the last hop router at receiver sites to start the trace. Therefore, a majority of these traces resulted in a failure before starting the actual trace between the remote sites. However, we believe these failures do not necessarily indicate multicast connectivity and/or reachability problems between the remote sites.

Table 4 presents our classification of reachability problems between our local site and remote announcement sites. According to this figure, 24% of the reachability problems are non-connectivity related problems and 38% of the problems are local connectivity problems (mtrace failed before exiting our local domain). We argue that the local connectivity problems presented above can be easily fixed/removed with some amount of effort at the edges of the network. This leaves us with the non-local connectivity problems as the most important problems. If we assume that our local site is representative of the majority of multicast user sites, we can conclude that a significant portion of reachability problems (38%) can be easily corrected with some amount of monitoring and management effort at individual end networks. However, the rate of non-local connectivity problems (38%) suggest that the multicast infrastructure itself has a significant number of problems.

Table 4
Mtrace based classification of reachability problems

Mtrace-based problem classification	
Successful traces (non-connectivity problems)	24% (310 traces)
Local connectivity problems	38% (490 traces)
Other (non-local) connectivity problems	38% (496 traces)

Finally, the analysis in this section depends on the multicast connectivity characteristics between our local site (ucsb.edu) and a number of session announcing sites. Due to the lack of additional network layer information among remote multicast sites, our problem classification presents the picture as seen from our local site and the overall classification for the entire multicast infrastructure may be different from our findings.

6. Related work

Sdr-monitor helped multicast community realize the need for developing additional tools/systems to perform necessary monitoring and management functions for the IP multicast service. In this section, we present a summary of the follow up work that is most closely related to our *sdr-monitor* project and refer our readers to [28] for a more comprehensive survey of multicast monitoring and management tools.

We divide the related work into three main areas: (1) application-specific tools (e.g., *multicast beacon* [10]) that help end-users to monitor application perceived performance of the multicast service in the network, (2) intra-domain measurement tools (e.g., *MRM* [21], *RMPMon* [11], *SMRM* [8], and *MRMON* [9]) that help network operators to monitor and debug multicast service in their domains, and (3) inter-domain protocol monitoring tools (e.g., *mantra* [25]) that help multicast researchers to understand the operation of protocols in the multicast infrastructure. Most of these tools/systems have been developed with some specific monitoring or management functionality in mind and they have been successful in achieving their specific design goals. We briefly discuss the related work below.

6.1. Application specific tools

Multicast beacon [10] is developed as a follow-up project after *sdr-monitor*. Contrary to *sdr-monitor*, *multicast beacon* uses active monitoring probes to monitor multicast reachability characteristics among a number of multicast end points. Due to its active monitoring nature, end systems can measure data reception quality (e.g., loss and jitter values) of the incoming multicast data. Similar to *sdr-monitor*, end points are multicast users that volunteer in the monitoring effort.

6.2. Intra-domain level measurement tools

The second group of related work includes protocols (*MRM* [21]) and systems (*SMRM* [8], *RMPMon* [11], and *MRMON* [9]) that have been developed to monitor and manage IP multicast services in the intra-domain. These systems are powerful as they provide network administrators with necessary primitives to run various types of active and passive multicast monitoring and measurement tests. On the other hand, they are limited to usage in intra-domain environments. Since the main goal of *sdr-monitor* was to monitor reachability in the inter-domain scale, the monitoring scope becomes the key difference between *sdr-monitor* and the tools in this group.

The Multicast Reachability Monitor (*MRM*) [21] is a protocol that is used to create active and passive multicast monitoring and measurement scenarios. *MRM*-capable network devices can be configured to run an active multicast test session and collect performance information. Or, they can be configured to measure the quality of multicast service for an ongoing application in a passive mode. *SMRM* [8] is a follow up effort that incorporates *MRM* functionality into a Simple Network Management Protocol (SNMP)-based network management framework so as to provide a standard approach to perform multicast monitoring tests in the network.

RMPMon [11] proposes an SNMP-based architecture for remote management of multicast routing infrastructure. It uses the existing Real-Time Transport Protocol (RTP) MIB [29] and also introduces a RTP Sender MIB to achieve both active and passive monitoring within an administrative domain. The Remote Multicast Monitoring (*MRMON*) [9] project is a more recent attempt that uses an SNMP-based framework to collect various types of multicast performance metrics from multicast end systems. It defines several multicast MIB groups to collect a comprehensive set of information about ongoing multicast sessions. MRMON is a passive monitoring system and does not consume a large amount of network resources as in the case of active monitoring systems.

6.3. Inter-domain level protocol monitoring tools

In the third group, we present *mantra* [25] as the main example of a system developed to monitor the multicast routing infrastructure. *Mantra* collects multicast routing table information from a number of Internet backbone routers and processes this information to create a global view of the multicast routing infrastructure. The information collected by *mantra* has helped researchers and network administrators understand the functioning and interaction of various multicast routing protocols that make up the multicast routing infrastructure. *Mantra* uses around a dozen vantage points to collect its data and therefore can only present a partial picture of the global multicast routing infrastructure. Its ability to identify and isolate specific problems is also rather limited.

7. Conclusions

In this paper we have addressed reachability monitoring as an important multicast management task. We have stressed the importance of reachability monitoring and presented a system, *sdr-monitor*, to perform this task. *Sdr-monitor* is used to monitor the reachability status of the global multicast infrastructure and report results via a real-time web interface. Using this system, we have collected reachability information during the last four years (April 1999 to March 2003). With this data, we have analyzed long term reachability characteristics for the multicast infrastructure. Our results show that reachability was very irregular and generally poor in the first two years, but has slowly improved. We believe that the reasons for this include the complexity of the multicast service architecture and the burden of continuously operating multicast as a network service.

From a design point-of-view, *sdr-monitor* is a good example of a successful distributed monitoring system. During the design of our system, we had two alternative design approaches: (1) designing a monitoring system that would support a large set of monitoring functions (e.g., passive and active monitoring; reachability and connectivity monitoring; monitoring service quality in terms of packet loss and jitter, etc.) and (2) designing a system that would be easy-to-develop and easy-to-deploy on wide scale to perform basic reachability monitoring. Given the urgency of the need for a monitoring system, we chose ease-of-development and ease-of-deployment at the expense of limited functionality. Even though our system provided a limited view of reachability, it has received a widespread acceptance by the multicast community. During the last four years (1999–2003), there has been over 120 people participating in our monitoring effort. During this time, the *sdr-monitor* web site was receiving 300–400 hits per day. Multicast users were frequently using the web site to learn about the reachability status of their announcements as well as detecting potential multicast problems in the network. From this perspective, our system has served the multicast community well in detecting and correcting multicast problems and pioneered a number of additional research efforts for various multicast monitoring and management tasks.

We believe that monitoring and managing multicast have become key requirements for the success of deployment in the Internet. Since multicast continues to exist as an experimental Internet service, having a highly available and highly robust multicast service will encourage continued evolution. Internet Service Providers (ISPs) will want to deploy the service in their network and application providers to consider using multicast as the communication model in their applications. This exercise will then result in a globally deployed multicast service. In addition, since multicast has been one of the first value-added services to be deployed *in* the Internet, its success will help encourage other value-added network services, such as quality-of-service (QoS), to be deployed in the Internet.

References

[1] K. Almeroth, The evolution of multicast: from the MBone to inter-domain multicast to Internet2 deployment, IEEE Network 14 (1) (2000) 10–20.

[2] C. Diot, B. Levine, B. Lyles, H. Kassem, D. Balensiefen, Deployment issues for the IP multicast service and architecture, IEEE Network 14 (1) (2000) 10–20.

[3] S. Paul, K. Sabnani, J. Lin, S. Bhattacharyya, Reliable multicast transport protocol (RMTP), IEEE Journal on Selected Areas in Communications 15 (3) (1997) 407–421.

[4] P. Judge, M. Ammar, Gothic: a group access control architecture for secure multicast and anycast, in: Proceedings of IEEE INFOCOM, New York, NY, USA, June 2002.

[5] C. Shields, J. Garcia-Luna-Aveces, KHIP—a scalable protocol for secure multicast routing, in: Proceedings of ACM SIGCOMM, Cambridge, MA, USA, August 1999, pp. 53–64.

[6] S. Jagannathan, K. Almeroth, A. Acharya, Topology sensitive congestion control for real-time multicast, in: Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Chapel Hill, NC, USA, June 2000.

[7] K. Sarac, K. Almeroth, Providing scalable many-to-one feedback in multicast reachability monitoring systems, in: 4th IFIP/IEEE International Conference on Management of Multimedia Networks and Services (MMNS), Chicago, IL, USA, October 2001.

[8] E. Al-Shaer, Y. Tang, SMRM: SNMP-based multicast reachability monitoring, in: IEEE/IFIP Network Operations and Management Symposium (NOMS), Florence, Italy, April 2002.

[9] E. Al-Shaer, Y. Tang, MRMON: multicast remote monitoring, in: IEEE/IFIP Network Operations and Management Symposium (NOMS), Seoul, Korea, April 2004.

[10] NLANR, Multicast Beacon, National Laboratory for Applied Network Research, June 2000. Available from: <dast.nlanr.net/Projects/Beacon/>.

[11] J. Chesterfield, B. Fenner, L. Breslau, Remote multicast monitoring using the RTP MIB, in: IFIP/IEEE International Conference on Management of Multimedia Networks and Services, Santa Barbara, CA, USA, October 2002.

[12] P. Rajvaidya, K. Almeroth, A scalable architecture for monitoring and visualizing multicast statistics, in: IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM), Austin, TX, USA, June 2000.

[13] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, G. Liu, L. Wei, PIM architecture for wide-area multicast routing, IEEE/ACM Transactions on Networking 4 (2) (1996) 153–162.

[14] T. Bates, R. Chandra, D. Katz, Y. Rekhter, Multiprotocol extensions for BGP-4, Internet Engineering Task Force (IETF), RFC 2283, February 1998.

[15] D. Meyer, B. Fenner, Multicast source discovery protocol (MSDP), Internet Engineering Task Force (IETF), RFC 3618, October 2003.

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

[17] S. Raman, S. McCanne, A model, analysis, and protocol framework for soft state-based communication, in: Proceedings of ACM SIGCOMM, Cambridge, MA, USA, September 1999.

[18] S. Casner, S. Deering, First IETF Internet audiocast, ACM Computer Communication Review 22 (3) (1992) 92–97.

[19] K. Sarac, K. Almeroth, A distributed approach for monitoring multicast service availability, Journal of Network and Systems Management, Special Issue on Distributed Management 12 (3) (2004) 327–348.

[20] V. Paxson, J. Mahdavi, A. Adams, M. Mathis, An architecture for large-scale internet measurement, IEEE Communications Magazine 36 (8) (1998) 48–54.

[21] K. Almeroth, K. Sarac, L. Wei, Supporting multicast management using the multicast reachability monitor (MRM) protocol, Technical Report, University of California, Santa Barbara, March 2000.

[22] M. Handley, SAP: session announcement protocol, Internet Engineering Task Force (IETF), RFC 2974, October 2000.

[23] M. Handley, SDR. Session Directory Tool, University College London, November 1995. Available from: <cs.ucl.ac.uk/mice/sdr/>.

[24] M. Handley, V. Jacobson, SDP: session description protocol, Internet Engineering Task Force (IETF), RFC 2327, April 1998.

[25] P. Rajvaidya, K. Almeroth, Analysis of routing characteristics in the multicast infrastructure, in: Proceedings of IEEE INFOCOM, San Francisco, CA, USA, April 2003.

[26] W. Fenner, A 'traceroute' facility for IP multicast, Internet Engineering Task Force (IETF), Internet Draft draft-fenner-traceroute-ipm-mdast.txt, Work in progress, December 2003.

[27] D. Massey, B. Fenner, Fault detection in routing protocols, in: International Conference on Network Protocols (ICNP), Toronto, Canada, November 1999.

[28] K. Sarac, K. Almeroth, Supporting multicast deployment efforts: a survey of tools for multicast monitoring, Journal of High Speed Networks, Special Issue on QoS for Multimedia on the Internet 9 (3,4) (2000) 191–211.

[29] M. Baugher, B. Strahm, I. Suconick, Real-time transport protocol management information base, Internet Engineering Task Force (IETF), RFC 2959, p. 2000.

Kamil Sarac received his M.S. and Ph.D. degrees in computer science from the University of California 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, peer-to-peer networking and overlay networks; management and security of computer networks. He has co-chaired the Computer Networks special track in ACM SAC 2004 and has served as a reviewer for a number of conferences and journals. He is a member of both the ACM and IEEE.

Kevin C. Almeroth received the Ph.D. degree in computer science from the Georgia Institute of Technology, Atlanta, in 1997. He is currently an Associate Professor at the University of California in Santa Barbara where his main research interests include computer networks and protocols, multicast communication, large-scale multimedia systems, and performance evaluation. At UCSB, he is a founding member of the Media Arts and Technology Program (MATP), Associate Director of the Center for Information Technology and Society (CITS), and on the Executive Committee for the University of California Digital Media Innovation (DiMI) program. In the research community, he is on the Editorial Board of IEEE Network and ACM Computers in Entertainment; has co-chaired a number of conferences and workshops including the International Conference on Network Protocols (ICNP), the Network and System Support for Digital Audio and Video (NOSSDAV) workshop, the Network Group Communication (NGC) workshop, and the Global Internet Symposium; and has been on the program committee of numerous conferences. He is serving as the chair of the Internet2 Working Group on Multicast, and is active in several working groups of the Internet Engineering Task Force (IETF). He also serves on the boards or directors and/or advisory boards of several startups including Occam Networks, Techknowledge Point, NCast, and the Santa Barbara Technology Group. He has been a member of both the ACM and IEEE since 1993.