Application Layer Reachability Monitoring for IP Multicast

Kamil Sarac, *Member*, IEEE and Kevin C. Almeroth, *Senior Member*, IEEE

Abstract—**Monitoring and management have become key requirements for the success of multicast deployment in the Internet. One of the most important monitoring tasks for multicast is to verify the availability of service in the network. This task is usually referred to as *reachability monitoring*. In this paper, we present an application layer multicast reachability monitoring system called *sdr-monitor*. *Sdr-monitor* has emerged in response to the practical need of verifying service availability and detecting potential problems during the early years of native multicast deployment in the inter-domain. *Sdr-monitor* leverages an existing application and provides close to real-time reachability monitoring for the multicast infrastructure. Since its initial deployment in 1998, *sdr-monitor* has been serving the multicast community in detecting and correcting multicast reachability problems in the Internet. In addition, *sdr-monitor* pioneered a number of additional research projects in multicast monitoring and management. In this paper, we first present the architecture of the *sdr-monitor* system and its outputs. Then, by using a four-year reachability monitoring data set, we present a long term analysis of the reachability characteristics of the multicast infrastructure. Next, by using additional network layer information, we classify reachability problems. Finally, we evaluate *sdr-monitor* as a reachability monitoring system and identify a number of ways in which it could be improved.**

Keywords— **Multicast monitoring and management, reachability, sdr.**

I. Introduction

Traffic generated by multimedia-based applications has evolved into a significant portion of Internet traffic[1]. As a result, there is a need to develop better mechanisms to support multimedia data delivery. New network-services, such as multicast delivery[2], quality-of-service[3], [4], and in-the-network processing[5] have all been proposed as potential solutions.

The focus of this paper is multicast. Multicast offers mechanisms to reach tens, thousands, even millions of receivers simultaneously in a scalable and bandwidth efficient way. The fundamental service offered by multicast is to solve the bandwidth bottleneck problem at the content server. Multicast allows one copy of each packet to be sent from a source. These packets are then replicated at key branching points along a tree connecting all interested receivers[6].

Most of the work in multicast has been on developing necessary protocols[7]; deploying them in the Internet[8]; and providing a number of additional services on top of the infrastructure including reliability[9], security[10], [11], and congestion control[12]. On the other hand, in order to achieve global deployment, we need the ability to monitor and manage multicast infrastructure-wide.

One of the most important monitoring tasks for multicast is to verify its availability to participating users. This task is referred to as *reachability monitoring*. 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. That is, if a tree cannot be built because there is no path to the source, the receiver will never know. 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. Consequently, verifying reachability becomes very important to maintain availability and robustness of the multicast service between sources and receivers. Without it, the multicast infrastructure becomes disconnected and essentially unusable.

In this paper we present an application layer reachability monitoring system called *sdr-monitor*. *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, whiteboard, 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 *sdr* and report which announcements are 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.

In addition to the web-based real time interface, *sdr-monitor* archives the collected reachability information for long term analysis. Using the archived data collected over the past four years, we have conducted an analysis of global reachability patterns. As a result of our analysis, we have found that reachability in the multicast infrastruc-

Kamil Sarac is with the Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, (email: ksarac@utdallas.edu).

Kevin Almeroth is with the Department of Computer Science, University of California, Santa Barbara, CA 93106, (email: almeroth@cs.ucsb.edu).

ture was initially poor but has seen noticeable improvement during the last two year. In our analysis, we have identified a number of possible causes for this trend. One important reason seems to be that during the early deployment of multicast, it was not considered an equivalent service to unicast. There were almost no traffic monitoring or management efforts dedicated to maintaining robustness and high availability of the multicast service.

Sdr-monitor emerged in response to the practical need of detecting multicast reachability problems in the Internet. 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. With this goal in mind, *sdr-monitor* has been designed as a convenient mechanism to monitor multicast reachability on an inter-domain scale. Prior to *sdr-monitor*, there were no mechanisms for multicast users to automatically learn the reachability of their multicast data at receiver sites. On the other hand, being an application layer reachability monitoring system, *sdr-monitor* is not necessarily the most effective way of performing reachability monitoring for multicast (see Section VI). From this perspective, it motivated a number of additional research projects for performing related monitoring and management tasks for multicast including MRM[13], RMPMon[14], HPMM[15], Multicast Beacon[16], Mantra[17], and MCPM[18].

The remainder of this paper is organized as follows. In the next section, we motivate the importance of multicast monitoring. In Section III, we present the *sdr-monitor* architecture, its components and the outputs it generates. In Section IV, we analyze long term reachability characteristics of the multicast infrastructure. In Section V, by using additional network layer information, we classify reachability problems into two groups. In Section VI, we provide an evaluation of *sdr-monitor* as a monitoring tool and the paper is concluded in Section VII.

II. 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. Reachability monitoring in the original multicast network topology (known as MBone[19]) was relatively straightforward. The MBone network topology was a virtual, flat network. Reachability, in most cases, was all or nothing. Cases of only partial connectivity existed but were not typical[20]. As the MBone has evolved into a native network service, and as the multicast topology has become hierarchical, reachability monitoring has become more complicated. The opportunity for reachability problems to exist has increased. In the current hierarchical model, multicast service is realized by running a set of protocols. 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)[21] 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)[22] to communicate multicast path availability and Multicast Source Discovery Protocol (MSDP)[23] to communicate multicast source availability among different domains in the network. Finally, the Internet Group Management Protocol (IGMP)[24] 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[25]. 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[26]. 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.

III. *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. That is, a sender in each domain should first send periodic heartbeat messages to a multicast channel. Second, receivers located in all other domains should be listening to this channel. And finally, these receivers should be reporting what messages they receive to a centralized site[1]. The centralized site then uses this information to generate a real-time visualization of global reachability. Even though it is difficult to achieve this ideal coverage, we have attempted to involve as many sites as possible in our study.

One way that we attempt to improve coverage is to make becoming an *sdr-monitor* participant as easy as possible. Therefore, our approach has been to build a system based on existing mechanisms. This has saved development time and is easier to deploy on a wide scale. Our system is based on the use of multicast session announcements as a heartbeat mechanism. This heartbeat serves as a way of monitoring reachability. In this section we first describe the *sdr*-based multicast session announcement mechanism and then present the *sdr-monitor* architecture. Finally, we describe the outputs generated by *sdr-monitor*.

A. *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)[27]. In SAP, announcements are periodically sent to a well known multicast address (SAP.MCAST.NET) with a certain scope. SAP is a soft-state based 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[28]. 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)[29] 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 next subsection, we use this mechanism as the basis of our multicast reachability monitoring task.

B. *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 Figure 1). This 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*. At every hour, 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. 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

[1]Using a centralized server may create a potential bottleneck or a single point of failure for the application. Extending this architecture to a distributed one is possible but is not necessarily the focus of this work With *sdr-monitor*, using a centralized approach did not cause any scalability problems during our monitoring efforts

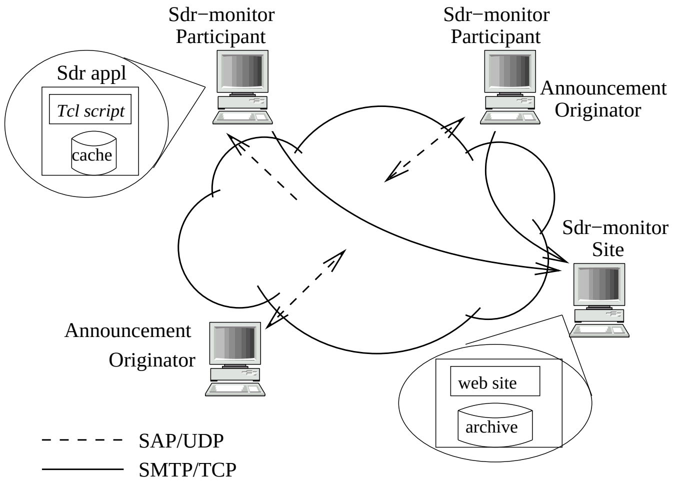


Fig. 1. The *sdr-monitor* architecture.

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.

C. 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.

C.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 Figure 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.
- 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.

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

• 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.

C.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 one hour 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.

IV. Reachability Analysis

In this section, we present a four-step analysis of four year's worth 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.

A. Phase 0: Data Collection

Our analysis is based on a data set collected between April 1, 1999 and March 31, 2003[2]. 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 estimate 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.

[2]Due to an un-detected problem, our system failed to archive reachability data between April 2002 and July 2002.

B. Phase 1: Pre-Processing and Initial View

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. The specific types of filtering we perform in Phase 1 are as follows:

- **Announcements with TTL less than 127**: All announcements with a TTL of less than 127 are filtered. This is done because it is difficult to determine which *sdr-monitor* sites should actually see these "less-than-global" session announcements.
- **Administratively scoped session announcements**: All administratively scoped session announcements including those announced with a global scope are filtered. Even though these sessions may have a global TTL, they will likely be blocked at administrative boundaries.
- **Stale announcements**: All announcements that have not received a soft-state update in the previous hour are considered "stale" and filtered. Stale announcements might be sent by *sdr-monitor* participants for several reasons. First, old versions of *sdr* do not expire stale announcements properly. Second, when a user starts *sdr*, the tool reads in the cached announcements and treats them as newly received announcements. When the sender script code is invoked, it will pack *all* announcements into a file and send them to the *sdr-monitor* collection site. In the first email received, it will look like announcements for some sessions have been received even though this is not the case. By looking at the last time an announcement was actually received, we can decide whether it is stale and should be removed.

Before presenting results after Phase 1 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. However, we are only interested in reachability on a per-site basis and not per-announcement. 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%. Figure 3-a shows the breakdown of results over three year-long period. As an example, according to this figure, at the beginning of April 1999, 38% of announcement sites had less than 25% reachability; 62% of sites had less than 50% reachability and 95% of sites had less than 75% reachability. Noteworthy about these results are the following:

- Overall reachability seems very poor. There are a large percentage of announcing sites (approximately 30% during the first two years and 20% during the last year) that send announcements seen by less than 25% of *sdr-monitor* participant sites.
- Reachability varies wildly. There are no distinctive trends and significant variability exists day-to-day.

In trying to understand the results, we have found that dynamic behavior among *sdr* users, session announcements, and *sdr-monitor* participants contributes significantly. In the next section we look to process the data in such a way to eliminate all problems related to using *sdr* as the underlying reachability monitoring mechanism.

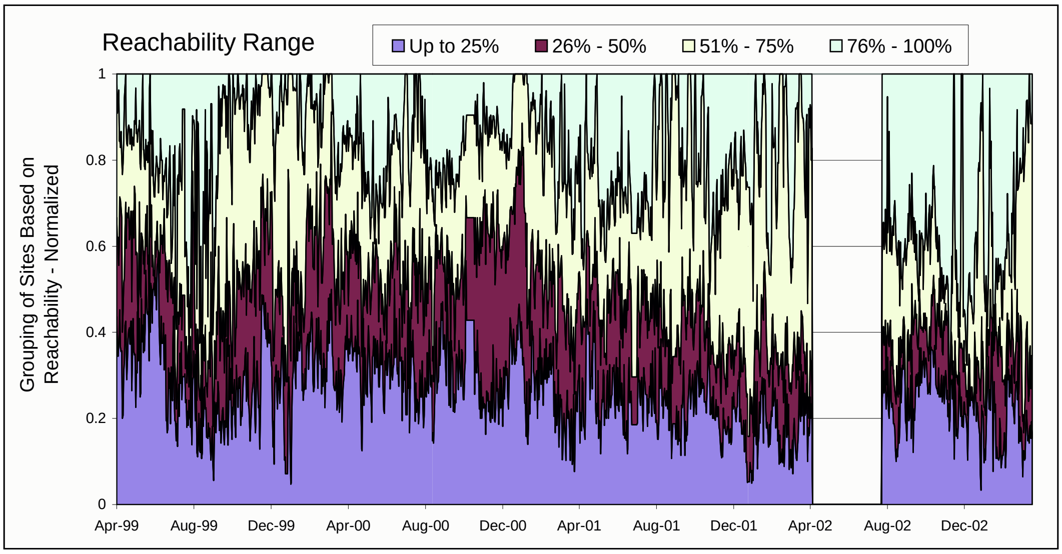
C. Phase 2: Removing Sdr Artifacts

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

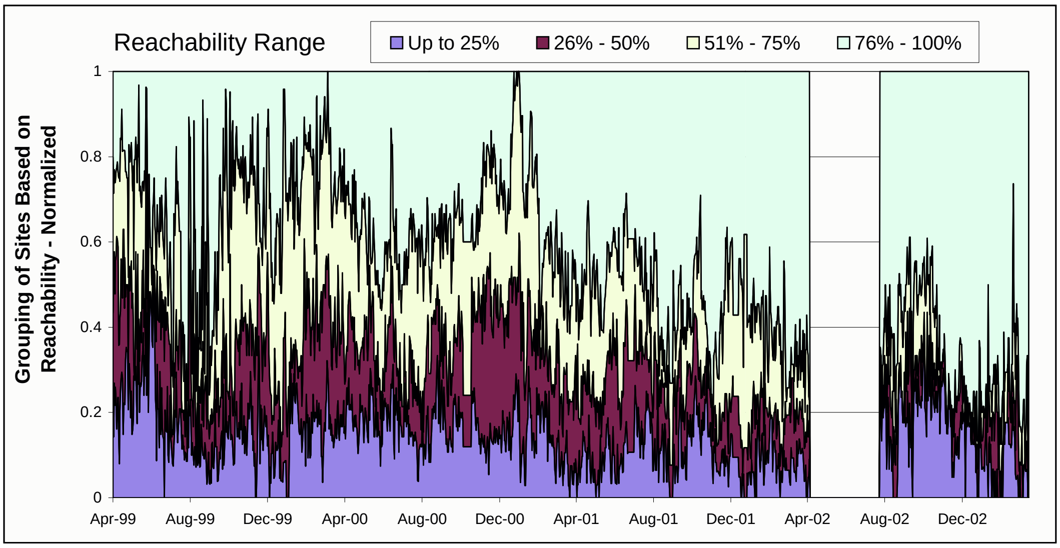
***Sdr-monitor* Participant Behavior.** In the data collection period, not all *sdr-monitor* participants were running *sdr* continuously. This means that not all participants are continuously reporting the sessions in their *sdr* caches. Therefore, the number and identity of participants actively sending their reports is not constant over long periods of time. During the first three years, the number of active participants has been between 15 and 35 with average of 26 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. In some cases, even though a session has not ended, the *sdr* tool advertising the session may be stopped. Like participants who see different sets of sites, session announcing sites will be seen by different sets of participants. 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 oc-



(a) Before removing sdr artifacts.



(b) After removing sdr artifacts.

Fig. 3. Average reachability for session announcing sites: April 1999 to March 2003.

cur. In order to estimate what the start and end behavior is, we have isolated a set of cases from the data set. Data with the following properties was used: 1) a session lasted longer than 12 hours, 2) it had at least 10 participants reporting it as visible at the end of the first 12 hours, and, 3) all these participants were continuously reporting their *sdr* caches during this 12 hours. We identified announcement chunks with these properties and computed their average visibilities at each hour during this 12 hour period. Figure 4-a shows the average reachability at the beginning of a session announcement. According to this figure, it takes two hours (two snapshot periods) for announcements to reach majority of the *sdr-monitor* participants (80% of the participants) and then the figure presents a heavy-tail distribution.

We conducted a similar analysis for behavior at the end of a session announcement. There are two reasons why a session would be removed from an *sdr* cache. First, the end of a session's lifetime may be reached. In this case, *sdr* should be able to use the wall-clock-time to determine that the session advertisement lifetime has ended. Second, a session may be prematurely terminated. Either the *sdr* tool announcing the session could be terminated or the particular session could be deleted. In either case, the session is no longer announced. Other caches should remove the session after not receiving an announcement for an hour. Figure 4-b shows the expected behavior and the observed behavior. The difference between the two is as a result of our archival process and does not conflict with the expected behavior.

Short Lived Sessions. Due to reachability behavior at announcement start and end, sessions with a short lifetime

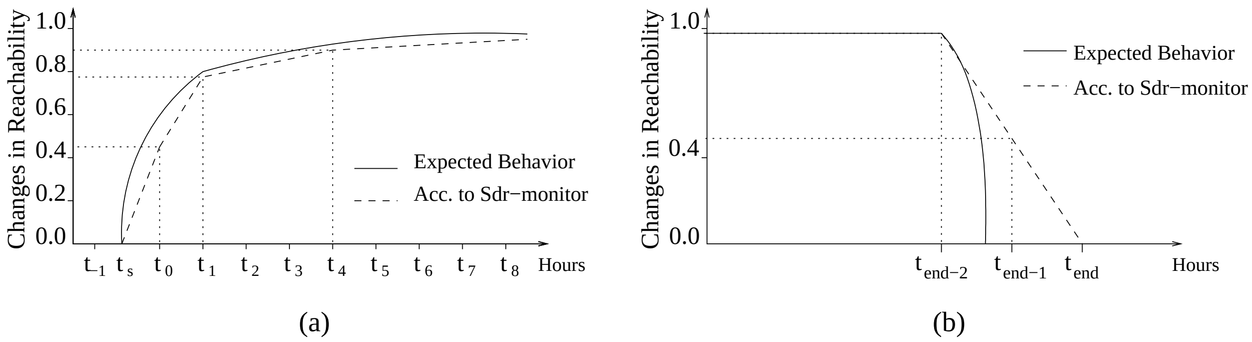


Fig. 4. Reachability at start and end of a session announcement.

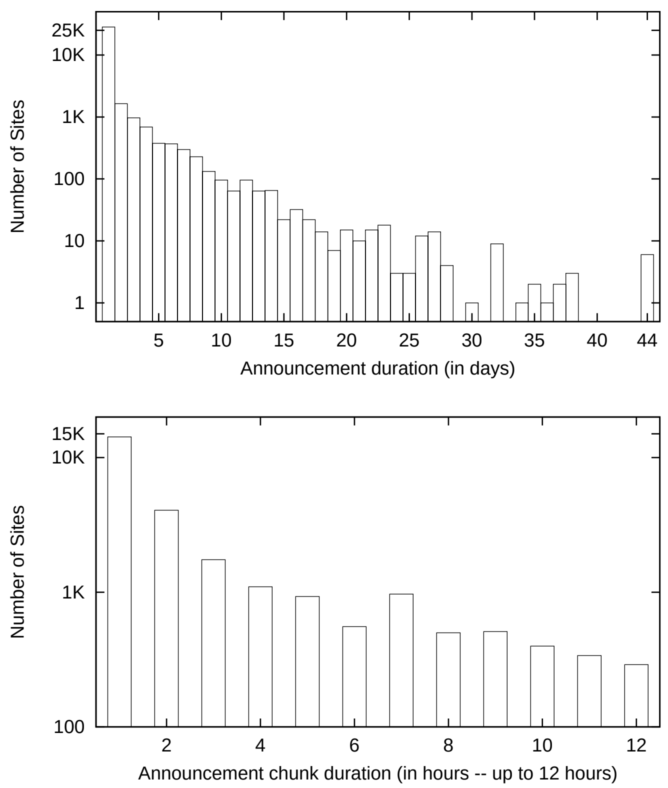


Fig. 5. Breakdown of global announcements based on lifetime.

particularly contribute to poor perceived reachability. Figure 5 shows a breakdown of session announcements by lifetimes. This figure shows that there are a lot of announcements with a very 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.

The above reasons clearly affect the reachability characteristics displayed in Figure 3-a. However, they are all related to the mechanism that we use to collect reachability information. From a multicast reachability perspective, they are not the true reachability problems that we are interested in identifying and characterizing. Once we identified these types of problems, we were able to filter them out from the data set. Figure 3-b displays the reachability characteristics after the Phase 2 filtering process. According to this figure, overall reachability improves but the same general patterns still exist.

D. 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. However, before continuing, it is worth making one final comment about the use of *sdr* traffic as a reachability heartbeat. *Sdr* traffic is bursty, sent infrequently, and susceptible to loss. So while network connectivity between two sites may exist, there is no guarantee that *sdr* traffic is actually received. We accept this as a characteristic of our system and even embrace it. Our sense is that if periodic traffic over the course of an hour cannot be received, then criteria for connectivity are not being met. Other research efforts are underway that analyze network layer statistics[30], [31].

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.

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 con-

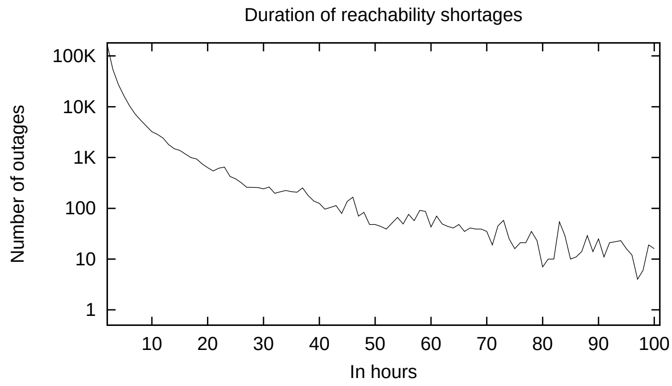


Fig. 6. Duration of reachability outages.

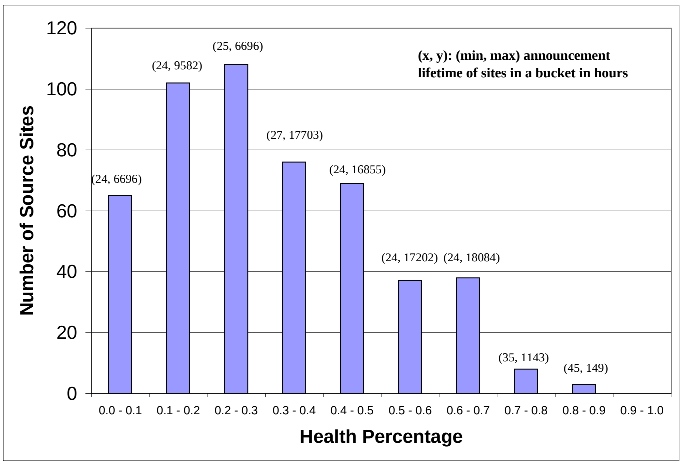


Fig. 7. Grouping of session announcing sites based on their health.

tributed to the irregular reachability characteristics shown in Figures 3-a and 3-b. 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. Figure 6 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 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. Figure 7 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 I shows the health ratios for the 10 most active session announcing sites.

Announcement Site	L_{cum}	V_{avg}	$R_{n/o}$	**Health**
Univ Oregon	29421	0.764	0.880	0.672
ENST, (FR)	17703	0.392	0.809	0.318
Lulea Univ, (SE)	17202	0.651	0.823	0.536
NASA, Calif.	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 Univ	9582	0.207	0.855	0.177
MulticastTech.com	8804	0.697	0.926	0.645

TABLE I

HEALTH OF THE 10 MOST ACTIVE SESSION ANNOUNCING SITES.

E. 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 hours to 1035 hours with an average of 516 hours.

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 values for each sender site. Tables II and III summarize

	US Reachability (in %)		
European Reachability (in %)	**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 II
REACHABILITY PERFORMANCE FOR US SENDERS.

	European Reachability (in %)		
US Reachability (in %)	**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

TABLE III
REACHABILITY PERFORMANCE FOR EUROPEAN SENDERS.

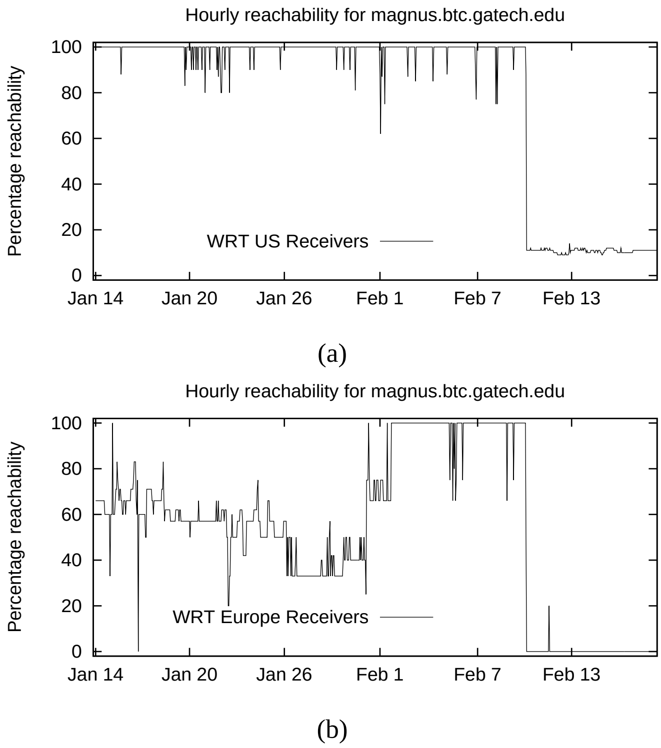


Fig. 8. A US sender located at Georgia Tech

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 II 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. Figures 8-a and 8-b present hourly reachability of a US sender site (a host at Georgia Tech) for 846 hours 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 hours. 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 hours. 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. Figures 9-a and 9-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.

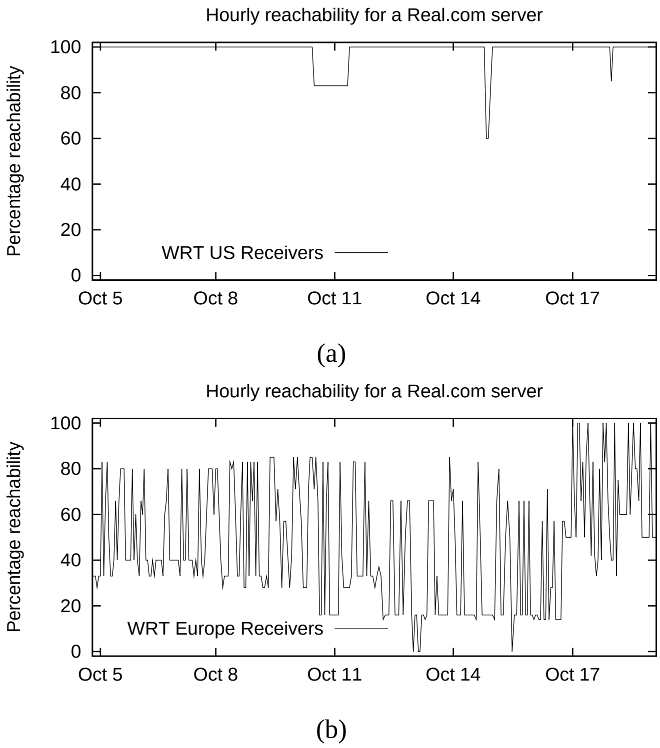


Fig. 9. A US sender from Seattle

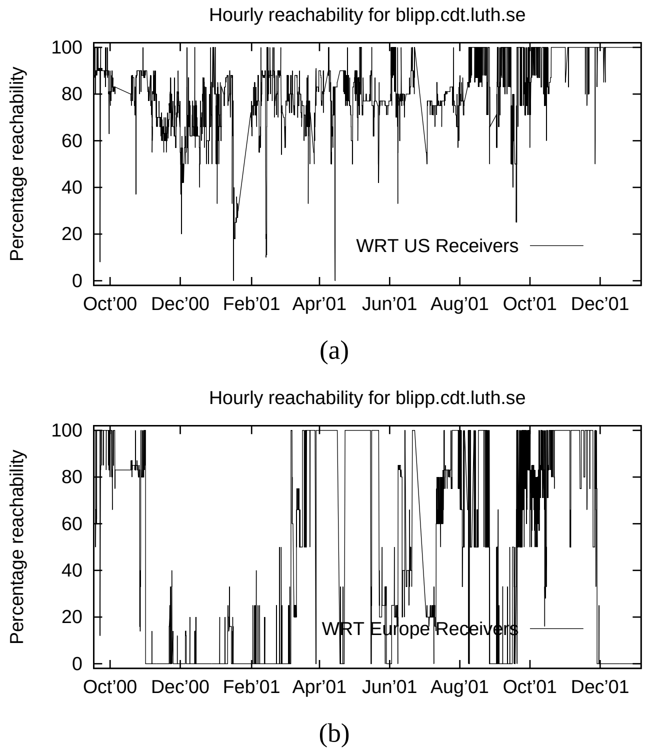


Fig. 10. A European sender from Sweden

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. Figures 10-a and 10-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 III 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 re-occur 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)[23] is an example of a possible source of problems.

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 next section, we use additional information (network layer multicast path information) to identify potential reasons for reachability problems.

V. Classification of Reachability Problems

In Section IV-D, 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*[32]. In the rest of this section, we first briefly describe how *mtrace* works and then present our analysis.

A. Mtrace

Mtrace is a multicast version of the *traceroute* utility[32]. 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. 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. 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.

B. 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[20] (early dropping of forwarding state in routers, etc.).

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 1,296 were between our local site (ucsb.edu) and 164 unique remote sites. We use the latter set of mtraces (1,296 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 IV 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.

VI. 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 reachabil-

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)

TABLE IV
MTRACE BASED CLASSIFICATION OF REACHABILITY PROBLEMS

ity 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 minutes), 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 source sites and participants can come and go at will, the results can change dramatically even though overall reachability does not change significantly (see Figure 3-b).

As we mentioned before, *sdr-monitor* is one of the first tools developed for inter-domain multicast reachability monitoring. *Sdr-monitor* has received widespread acceptance by the multicast community. During the last four years (since April 1999), there has been over 120 people participating in our monitoring effort. During this time, the *sdr-monitor* web site has been receiving 300-400 hits per day. Multicast users have been frequently using the web site to learn about the reachability status of their announcements as well as detecting potential multicast problems in the network. More recently, many of the above mentioned problems have been fixed in a follow-on project to *sdr-monitor* called the Multicast Beacon[16].

VII. 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. Finally, from a historical perspective, *sdr-monitor* has become one of the first widespread multicast monitoring systems. It has served the multicast community 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] S. McCreary and K. Claffy, “Trends in wide area IP traffic patterns, a view from Ames Internet exchange,” in *ITC Specialist Seminar*, (Monterey, California, USA), September 2000.

[2] S. Deering and D. Cheriton, “Multicast routing in datagram internetworks and extended LANs,” *ACM Transactions on Computer Systems*, pp. 85–111, May 1990.

[3] M. Carlson, W. Weiss, S. Blake, Z. Wang, D. Black, and E. Davies, “An architecture for differentiated services.” Internet Engineering Task Force (IETF), RFC 2475, December 1998.

[4] S. Shenker, C. Partridge, and R. Guerin, “Specification of guaranteed quality of service.” Internet Engineering Task Force (IETF), RFC 2212, September 1997.

[5] S. Bhattacharjee, K. Calvert, and E. Zegura, “Active networking and end-to-end arguments,” *IEEE Networking Magazine*, 1998.

[6] Y. K. Dalal and R. M. Metcalfe, “Reverse path forwarding of broadcast packets,” *Communications ACM*, vol. 21, pp. 1040–1048, Dec. 1978.

[7] K. Almeroth, “The evolution of multicast: From the MBone to inter-domain multicast to Internet2 deployment,” *IEEE Network*, vol. 14, pp. 10–20, January/February 2000.

[8] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen, “Deployment issues for the IP multicast service and architecture,” *IEEE Network*, vol. 14, pp. 10–20, January/February 2000.

[9] S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya, “Reliable multicast transport protocol (RMTP),” *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 407–421, April 1997.

[10] P. Judge and M. Ammar, “Gothic: A group access control architecture for secure multicast and anycast,” in *IEEE Infocom*, (New York City, NY, USA), June 2002.

[11] C. Shields and J. Garcia-Luna-Aveces, “KHIP - a scalable protocol for secure multicast routing,” in *SIGCOMM*, pp. 53–64, 1999.

[12] S. Jagannathan, K. Almeroth, and A. Acharya, “Topology sesitive congestion control for real-time multicast,” in *Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Chapel Hill, North Carolina, USA), June 2000.

[13] K. Sarac and 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.

[14] J. Chesterfield, B. Fenner, and L. Breslau, "Remote multicast monitoring using the RTP MIB," in *IFIP/IEEE International Conference on Management of Multimedia Networks and Services*, (Santa Barbara, California, USA), October 2002.

[15] J. Walz and B. Levine, "A hierarchical multicast monitoring scheme," in *International Workshop on Networked Group Communication (NGC)*, (Palo Alto, California, USA), November 2000.

[16] NLANR, *Multicast Beacon*. National Laboratory for Applied Network Research, June 2000. Available from http://dast.nlanr.net/Projects/Beacon/.

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

[18] A. Kanwar, K. Almeroth, S. Bhattacharyya, and M. Davy, "Enabling end-user network monitoring via the multicast consolidated proxy monitor," in *SPIE ITCom Conference on Scalability and Traffic Control in IP Networks*, (Denver, Colorado, USA), August 2001.

[19] H. Eriksson, "The multicast backbone," *Communications of the ACM*, vol. 8, pp. 54–60, 1994.

[20] D. Massey and B. Fenner, "Fault detection in routing protocols," in *International Conference on Network Protocols (ICNP)*, (Toronto, CANADA), November 1999.

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

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

[23] D. Meyer and F. B., "Multicast source discovery protocol (MSDP)." Internet Engineering Task Force (IETF), draft-ietf-mboned-msdp-*.txt, November 2002.

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

[25] S. Raman and S. McCanne, "A model, analysis, and protocol framework for soft state-based communication," in *ACM Sigcomm*, (Cambridge, Massachusetts, USA), September 1999.

[26] S. Casner and S. Deering, "First IETF Internet audiocast," *ACM Computer Communication Review*, pp. 92–97, July 1992.

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

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

[29] M. Handley and V. Jacobson, "SDP: Session description protocol." Internet Engineering Task Force (IETF), RFC 2327, April 1998.

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

[31] T. Wong and R. Katz, "An analysis of multicast forwarding state scalability," in *International Conference on Network Protocols (ICNP)*, (Osaka, JAPAN), November 2000.

[32] W. Fenner and S. Casner, "A 'traceroute' facility for IP multicast." Internet Engineering Task Force (IETF), draft-ietf-idmr-traceroute-ipm-*.txt, July 2000. Work in progress.

Kamil Sarac is currently an assistant professor in the Department of Computer Science at the University of Texas at Dallas. He obtained his M.S. and Ph.D. degrees in Computer Science from the University of California Santa Barbara in 1997 and 2002 respectively. He received his B.S. in Computer Engineering from Middle East Technical University, Turkey, in 1994. His main research interests focus on multimedia networking; IP multicast and multicast routing protocols; inter-domain network monitoring and management and distributed systems. Particularly, he has been working on developing tools and techniques for monitoring data transfer operations between multicast-enabled networks. He is a member of both the ACM and IEEE.

Kevin C. Almeroth 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, Dr. Almeroth 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, Dr. Almeroth is on the Editorial Board of IEEE Network, has co-chaired Global Internet, NGC, NOSSDAV, and ICNP; has served as tutorial chair for several conferences, and has been on the program committee of numerous conferences. Dr. Almeroth is serving as the chair of the Internet2 Working Group on Multicast, and is a member of the IETF Multicast Directorate (MADDOGS). He is also serving on the advisory boards of several startups including Occam Networks, NCast, and the Santa Barbara Technology Group. He has been a member of both the ACM and IEEE since 1993.