Monitoring Reachability in the Global Multicast Infrastructure

Kamil Saraç and *Kevin C. Almeroth*
Department of Computer Science
University of California - Santa Barbara
{ksarac, almeroth}@cs.ucsb.edu

Abstract

The multicast infrastructure has transitioned to a topology that now supports hierarchical routing. Instead of a flat virtual topology originally called the Multicast Backbone (MBone) there now exists a hierarchy where routing information is exchanged between autonomous systems (ASes). In today's multicast infrastructure reachability problems are common. Unlike in the MBone, the possibility of limited connectivity between domains is now possible. In this paper, we present a system called sdr-monitor. This tool collects session directory information from numerous places around the world and presents an application-layer view of reachability. Using data collected over the last year, we present an analysis of long term reachability characteristics for the global multicast infrastructure. Our findings are that overall reachability is generally quite poor. However, having identified some of the reasons, we believe there is not a fundamental infrastructure problem, but rather protocol bugs and a lack of management tools.

1 Introduction

The IP multicast infrastructure, formerly known as the Multicast Backbone(MBone), is transitioning from the original flat, virtual network topology to a hierarchical topology[1, 2]. Today, most router vendors support native multicast routing and Internet Service Providers (ISPs) are starting to deploy multicast in their networks. In this hierarchical topology, ISPs run potentially different multicast routing protocols within their domain and use a particular set of protocols to provide inter-domain multicast support[3].

Commercial ISPs have been experiencing difficulties in deploying multicast in their networks[4]. These difficulties are mainly related to the instability and complexity of multicast routing. It is believed that the availability of good management tools has become a crucial roadblock to successful multicast deployment.

Multicast reachability is one of the key multicast management issues. Reachability ensures that sources can reach all existing and all potential group members. Reachability also implies that receivers also have multicast connectivity. Some protocols and services rely on multicast connectivity to sources as well as among group members[5]. We have undertaken the task of developing techniques to monitor reachability in the global multicast infrastructure and then to develop a quantitative analysis of global multicast reachability.

Reachability monitoring in the original MBone was relatively straightforward. The multicast 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[6]. 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. For example, with deployment of a multicast version of the Border Gateway Protocol (BGP)[7], inter-domain peering problems can exist. In general, inter-domain connectivity is a much more fragile communications path. As a result, compared to the previous flat topology, the hierarchical multicast infrastructure is more scalable and more robust but we must now pay more attention to reachability monitoring.

In this paper we present a system to monitor multicast reachability. Our system, called *sdr-monitor*, is based on multicast session announcements sent by *sdr*, the session directory tool[1]. *Sdr* is a distributed tool used by researchers around the world to announce the availability of multicast audio, video, whiteboard, and/or text sessions. Sdr-monitor has a number of *participating sites* and a centralized *data collection site*. Participants listen to the periodic session announcements sent by *sdr* and report which announcements are seen at their site. Sdr-monitor then processes these reports and builds a real-time web page displaying a reachability matrix for the global multicast infrastructure. Sdr-monitor also archives the collected information for long term analysis. Using archived data col-

[1] http://www-mice.cs.ucl.ac.uk/multimedia/software/

lected over the last year, we have conducted an analysis of global reachability patterns. As a result of our analysis, we have found that reachability in the multicast infrastructure is relatively poor. We have identified a number of possible causes. The root problem seems to be that multicast is not considered an equivalent service to unicast. There is relatively less management and monitoring resources dedicated to maintaining high availability of multicast connectivity.

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. The paper is concluded in Section 5.

2 Motivation

Multicast traffic management is somewhat similar to unicast traffic management, but there are differences. One of the key differences is that multicast traffic can be received by a potentially large number of destinations. This difference can make some management functions more complex. It also changes the end-user management semantics. In a multicast session, sources do not know who the group receivers are, and receivers do not necessarily know all sources.

In this paper we focus on multicast *reachability*. Compared to multicast reachability, unicast reachability is an easier problem. Unicast deals with communication between two known end points. Thus, unicast reachability involves only the links along the path between these two sites. In the multicast model, any sender can send data to a multicast channel and any receiver can join to the channel without explicitly notifying the senders or any other administrative entity[1]. This means that the number of receivers and senders may be unknown to individual sites. Monitoring reachability in this environment becomes more complex. Among the reasons: the underlying data path changes into a tree topology, and this tree can change over time.

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.

Soft-state based protocols are a good example of a multicast application particularly susceptible to reachability problems. In general, soft-state based protocols are characterized as a number of sources transmitting periodic state-refresh messages over lossy communication channels to one or more receivers that maintain a copy of the state[8]. In these protocols, sources and receivers may not know the existence of each other. That is, sources do not get any feedback from receivers (to avoid implosion) and receivers assume no sources in the absence of update messages (to avoid connection establishment complexities, etc.). In addition, receivers keep a timeout interval. If no refresh messages arrive within this timeout interval, receivers expire their state information. Therefore, a lack of updates due to reachability problems not only interferes with the service but the nature of the application makes problems 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. 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[2]. Users run the session directory tool *sdr* which listens to this address; receives session announcements; and displays them. 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 running *sdr* do not receive the session announcement because of some reachability problem, they will never know that such a session existed. Tools need to exist to give the person announcing a session confidence that the session is reaching most (if not all) potential receivers. Potential receivers need confidence that they are being informed of all existing sessions.

3 Monitoring Multicast Reachability

In this section, we describe our tool, sdr-monitor, which has been developed to monitor reachability. In the ideal case, monitoring global reachability 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 sessions they see to a centralized site. The centralized site would then use 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.

One way that we attempt to improve coverage is to make becoming an sdr-monitor participant as easy as possible. Therefore, our approach is 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 *sdr* session announcements as heartbeat mechanism. This heartbeat serves as a way of monitoring reachability. In this section we first describe *sdr's* operation and then present the sdr-monitor architecture. Finally, we describe the outputs generated by sdr-monitor.

3.1 Multicast Session Announcements

Multicast session announcements have traditionally been used in the MBone as a way of informing the community about the availability of multicast conferences and broadcasts. One mechanism to communicate session announcements in the MBone is the *sdr* tool. *Sdr* is based on the Session Announcement Protocol (SAP)[9]. 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. Acknowledgements 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 tool most commonly used for creating and communicating session announcements. When a session announcement is created, *sdr* requests information about the session from the user. This information includes session name, multicast addresses, media types, etc. *Sdr* then creates an announcement entry using the Session Description Protocol (SDP)[10] 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 sessions for which an announcement has 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 a cache, and uses them to populate the *sdr* session 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 their code into an "*sdr.tcl*" file. When *sdr* starts, it automatically reads the user-specified code and executes it. This enables users to add customized functionality. By developing code that sdr-monitor participants can run, we can easily build functionality into *sdr* to monitor and report on session visibility. These results form the basis for monitoring reachability.

3.2 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. Currently sdr-monitor has around 100 registered participants. On average, there are 26 active participants. 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 every hour. When invoked, 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 what sessions are seen at a remote site. 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 particular site.

Central Collection/Processing Site: At the sdr-monitor collection 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.

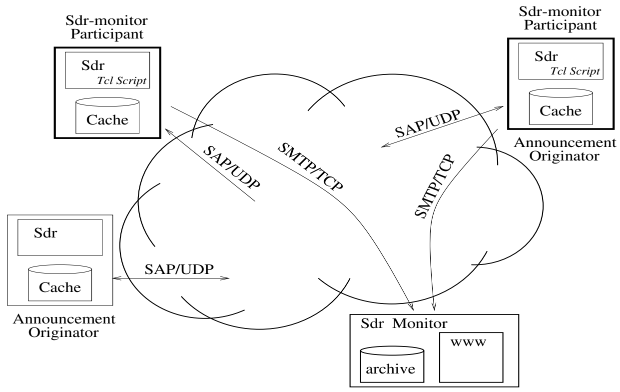

Figure 1. The sdr-monitor architecture.

3.3 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.

3.3.1 Sdr-Monitor Web Page

The web page is used to give the multicast community a 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 a 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 sdr-monitor report. 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.

Participant List: The participant list displays information about currently active sdr-monitor participants. Figure 3 shows a snapshot of the participant list. Each row contains information about a participant including the email address, geographic location, IP address, and the number of global session advertisements seen and not seen.

Only globally scoped announcements are displayed on the web page, so the sdr-monitor web page should reasonably display reachability between a large number of networks. By examining this real-time snapshot, the web page can be used to quickly identify 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 does 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 concentrates the negative results in the lower-right corner.
- The most interesting results occur 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 likely indicate potential reachability problems. In general, since the multicast community works to ensure that the infrastructure is not split, these types of patterns should not occur. When conducting our analysis we focus on counting 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. We do this on an ad hoc basis now but are working to incorporate the functionality into the web page automatically.

3.3.2 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 to count and characterize reachability problems.

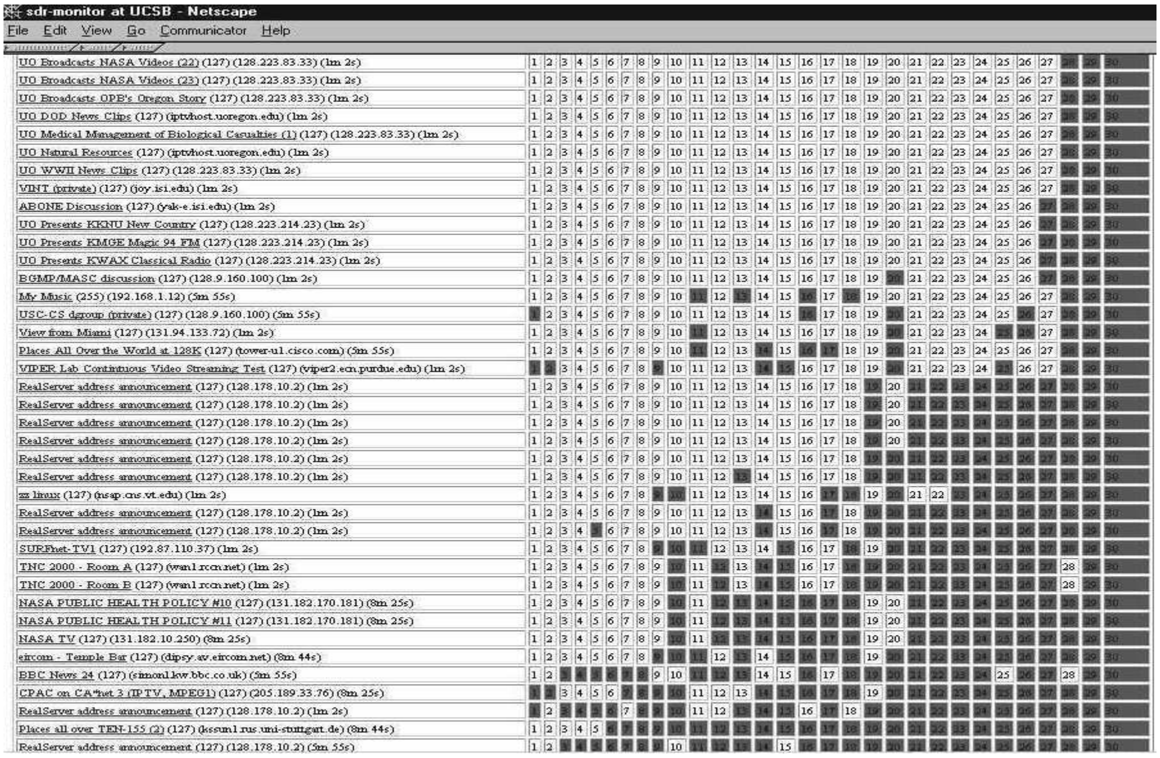

Figure 2. A snapshot of the session reachability matrix from the sdr-monitor web page.

No	Participant e-mails	Their Locations	IP Addresses	Last Heard	Participations	Sessions They See	Don't but SHOULD see
1	dax@inf.enst.fr	Paris_FRANCE	137.194.160.3	30m47s	8d1h	71	12
2	Thierry.Turletti@sophia.inria.fr	Sophia_Antipolis_FRANCE	138.96.24.64	8m44s	2d11h	70	13
3	esj@cs.fiu.edu	Miami_FL_USA	131.94.125.231	20m39s	1d22h	66	17
4	tward@marmot.nsg.nwu.edu	Evanston_IL_USA	129.105.26.22	33m13s	3d3h	66	17
5	damon@Porsche.it.iupui.edu	Indianapolis_IN_USA	134.68.11.41	36m55s	1d23h	65	18
6	gcook@ecn.purdue.edu	West_Lafayette_IN_USA	128.46.200.203	28m44s	19d22h	65	18
7	hak@raven.uoregon.edu	Eugene_OR_USA	128.223.214.21	13m47s	2d1h	65	18
8	llynch@pythia.uoregon.edu	Eugene_OR_USA	128.223.60.21	32m43s	20d21h	64	19
9	Jan.Novak@dante.org.uk	Cambridge_ENGLAND	193.63.211.19	21m14s	9d11h	62	21
10	he@vader.runit.sintef.no	Trondheim_NORWAY	129.241.100.134	5m55s	3d4h	61	22
11	kevin@cc.gatech.edu	Atlanta_GA_USA	130.207.3.207	8m25s	0d8h	61	22
12	tengi@CS.Princeton.EDU	Princeton_NJ_USA	128.112.136.10	17m56s	8d7h	61	22
13	guchardi@pisces.scar.utoronto.ca	Scarborough_ON_CANADA	142.150.160.86	32m41s	14d4h	60	23
14	esther@sideral.rediris.es	Madrid_SPAIN	130.206.1.32	58m18s	9d11h	59	24
15	leinen@babar.switch.ch	Zurich_SWITZERLAND	130.59.4.2	25m57s	0d2h	59	24
16	wmaton@ryouko.dgim.crc.ca	Ottawa_ON_CANADA	142.92.39.75	1m2s	3d8h	58	25
17	Niels.denOtten@surfnet.nl	Radboudburcht_NETHERLANDS	192.87.109.128	58m23s	8d10h	57	26
18	pavlin@ISI.EDU	Los_Angeles_CA_USA	128.9.128.128	14m36s	3d2h	56	27
19	pgp@pgp1.cit.cornell.edu	Ithaca_NY_USA	128.253.64.34	43m53s	8d23h	55	28
20	fenner@www.aciri.org	Berkeley_CA_USA	192.150.187.11	21m11s	280d18h	52	31
21	mbone@bfbi.net	UNKNOWN	192.249.24.22	8m26s	3d9h	49	34
22	yozo@aohakobe.ipc.chiba-u.ac.jp	Chiba_JAPAN	133.82.241.137	33m40s	14d16h	49	34
23	almeroth@cs.ucsb.edu	Santa_Barbara1_CA_USA	128.111.52.10	11m45s	17d3h	48	35
24	ksarac@cs.ucsb.edu	Santa_Barbara1_CA_USA	128.111.52.19	53m49s	6d18h	48	35
25	Loris.Marchetti@CSELT.IT	Turino_ITALY	163.162.41.5	6m55s	14d6h	47	36
26	slevy@ncsa.uiuc.edu	Urbana_IL_USA	141.142.2.9	9m56s	3d3h	47	36
27	aronson@nlm.nih.gov	Bethesda_MD_USA	130.14.35.128	43m44s	14d21h	41	42
28	alex@nunes.uminho.pt	Braga_PORTUGAL	193.136.9.202	44m42s	0d10h	8	75
29	jfletche@shetland.xkl.com	Redmond_WA_USA	192.94.202.202	7m33s	3d6h	1	82
30	houle@zeppo.acns.fsu.edu	Tallahasse_FL_USA	146.201.3.10	31m51s	1d3h	0	83

Figure 3. A snapshot of the participant table from the sdr-monitor web page.

4 Reachability Analysis

In this section, we present a three-step analysis of one 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 in *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.

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: either a source- or receiver- based calculation of reachability. The difference between the two is mostly semantic. Therefore, we only need to consider one type of reachability—source-based reachability. Reachability is calculated by counting the number of sdr-monitor participants who see the session and divide it by the number of current sdr-monitor participants.

4.1 Phase 0: Data Collection

Analysis in this paper is based on data collected between April 1, 1999 and March 31, 2000. During this time, the sdr-monitor collection site received approximately 220,000 emails. 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.

4.2 Phase 1: Pre-Processing and Initial View

Our data set includes a number of entries that are not useful for monitoring. 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 (non-global) are filtered.

Administratively scoped announcements: All administratively scoped session announcements are filtered. Even though these sessions may have a global TTL, they will likely be blocked at administrative boundaries.

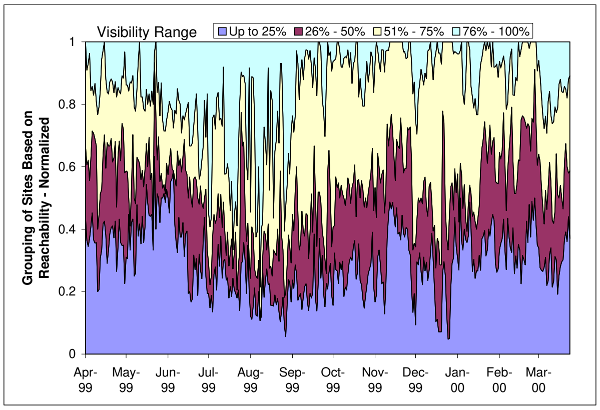

Figure 4. Average visibility for session announcing sites averaged daily.

Stale announcements: All sessions 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 one of 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. By looking at the last time an announcement was actually received, we can decide whether to remove it.

For each session announcing site, we compute a daily average reachability. This is computed by averaging the visibility of sites for each day. Visibility 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 visibility. The four groups are: 0%-25%, 26%-50%, 51%-75%, and 76%-100%. Figure 4 shows the breakdown of results over the year-long period. According to this figure, at the end of March 2000, 40% of announcement sites had less than 25% visibility; 60% of sites had less than 50% visibility and 90% of sites had less than 75% visibility. Noteworthy about these results are the following:

- **Overall reachability seems poor**. There are a large percentage of announcing sites (approximately 30%) that send announcements seen by less than 25% of sdr-monitor participant sites.

- **Reachability varies wildly.** There are no trends and significant variability exists day-to-day.

In trying to understand why the results are like they are, 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 mechanism.

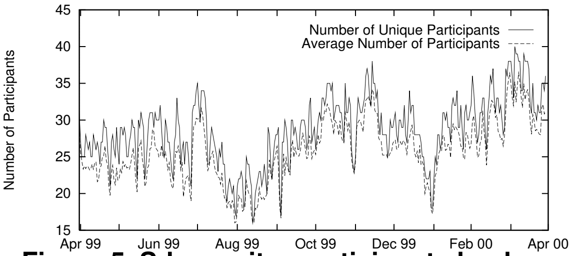

Figure 5. Sdr-monitor participants by day.

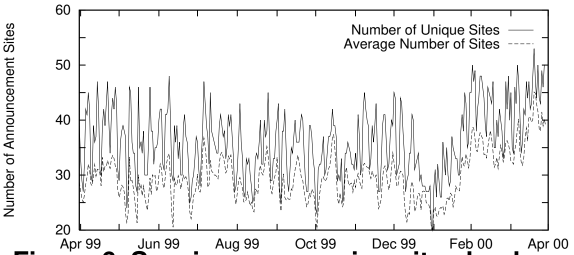

Figure 6. Session announcing sites by day.

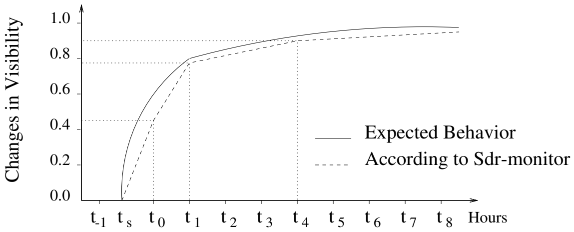

Figure 7. Observed behavior at session start.

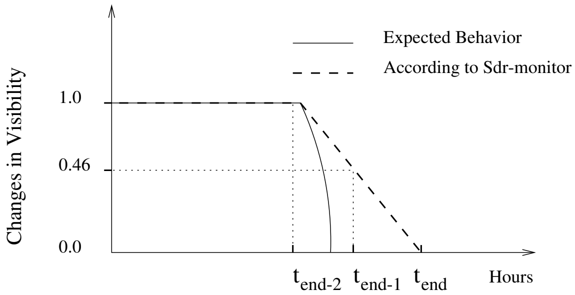

Figure 8. Observed behavior at session end.

4.3 Phase 2: Removing Sdr Artifacts

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

Sdr-monitor Participant Behavior. Not all sdr-monitor participants run *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 time. Since each participant has a potentially different picture of global reachability, their joining and leaving can cause dramatic changes in sdr-monitor's results. Figure 5 shows daily participation during the data collection period. The solid line in the figure shows the number of unique sdr-monitor participants. The dotted line gives the daily average of the number of active participants. Figure 5 shows that the average number of participants varies enough to cause some of the variability seen in Figure 4.

Session Announcing Site Behavior. Similar to the above problem, the number of sites *sourcing* session announcements is also dynamic. Figure 6 shows the average and unique numbers of sites sending announcements. 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.

Visibility at Announcement Start/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 see a session will be a small percentage of all sites. 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. In order to estimate what the start and end behavior is, we have isolated a set of cases from the data set. Figure 7 shows the average reachability at the beginning of a session announcement. According to Figure 7, it takes at least one hour for a newly started session announcement to reach approximately 80% of reachable sdr-monitor participants and 4 hours to reach 90%. We conducted a similar analysis for behavior at the end of a session announcement. Figure 8 shows the expected behavior and the observed behavior. The difference between the two is a result of there being a period of time in which some sdr-monitor participants report having heard from the session in the last hour while others report not seeing the session.

Short Lived Sessions. Visibility behavior at announcement start and end can affect perceived reachability especially for short-lived sessions. Figure 9 shows a breakdown of session announcements by lifetimes. This figure shows there are many announcements with a short lifetime. These announcements contribute to poor perceived reachability because the announcement has started and ended before the announcement has been cached by all sdr-monitor participants.

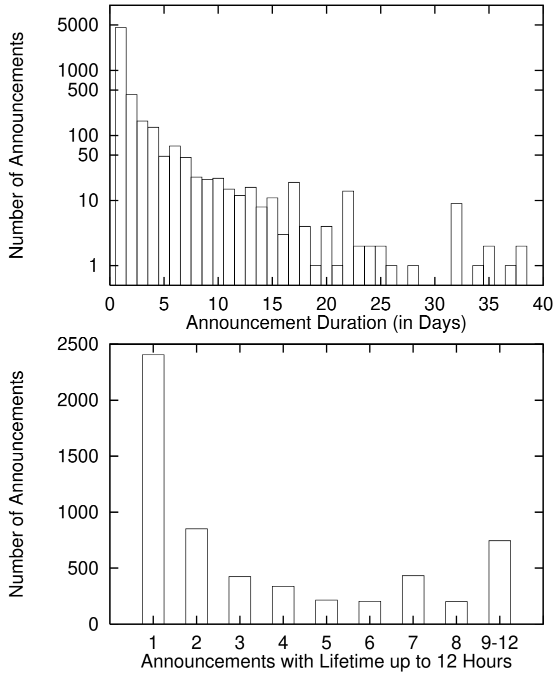

Figure 9. Global announcement lifetimes.

The three reasons just discussed clearly affect the reachability characteristics displayed in Figure 4. 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 10 displays the reachability characteristics in a way similar to Figure 4 but after the Phase 2 filtering process. According to this figure, overall reachability improves but the same general patterns still exist.

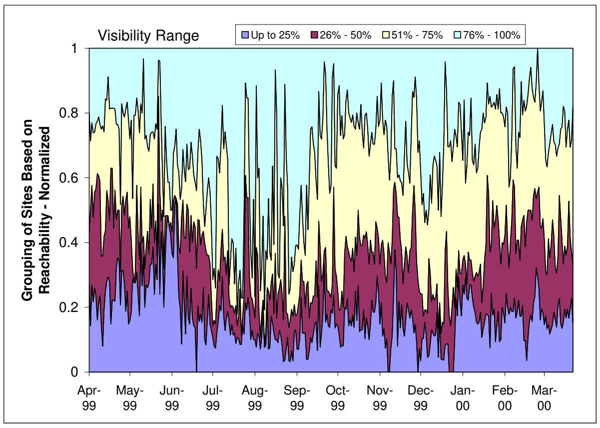

Figure 10. Average visibility after Phase 2.

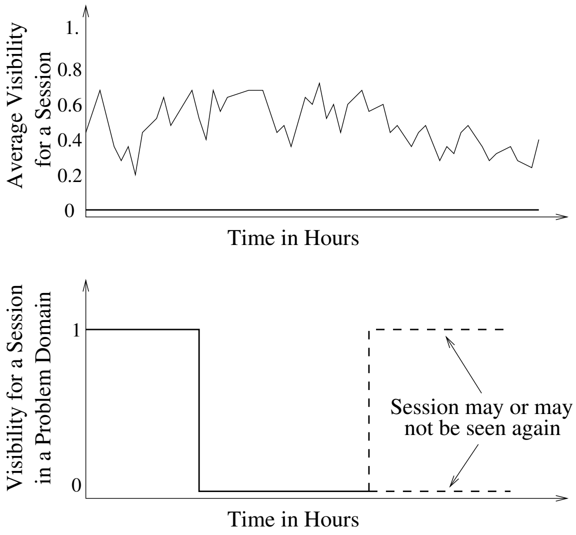

Figure 11. Visibility loss in a problem domain.

4.4 Phase 3: Characterizing Problems

After Phase 2 processing we believe we have a data set that only includes end-to-end reachability problems. Our goal is now 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 can not be received, then criteria for connectivity are not being met. Other research efforts are underway that analyze network layer statistics[11, 12].

Our 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 session announcement.* This type of reachability problem occurs only after an sdr-monitor site first receives an announcement, and then does not. This type of event is logically represented using the graphs shown in Figure 11.

When an event of the type in Figure 11 occurs we count it as a *visibility loss event*. In order to compare the number of loss events to the total number of events we define a *successful visibility transition* event. This event occurs when a session announcement is seen by an sdr-monitor participant in two consecutive snap-

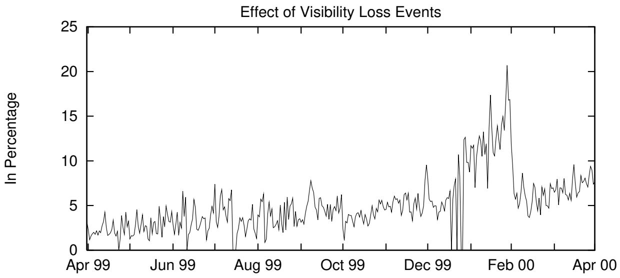

Figure 12. Loss events as a percentage of total events.

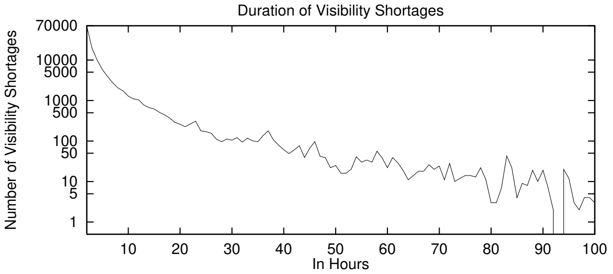

Figure 13. Duration of reachability outages.

shots. Figure 12 shows the loss events as a percentage of the total number of possible events. 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. The results in Figure 12 show an odd surge in errors during the month of January. We do not have an explanation for this. Otherwise, the percentage of problems seems to have increased slightly during the year.

Having quantified the number of problems, we now attempt to characterize whether problems are short-lived or long-lived. Problems that lasted for only a short time partially contributed to the irregular visibility characteristics shown in Figures 4 and 10. 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 visibility 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 13 shows a distribution of the visibility 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 continue with a qualitative analysis and suggest a number of reasons why reachability problems are so common. These include the following:

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 or re-occur they create a significant number of visibility loss events. Our belief is that local connectivity problems occur frequently for some sites; that multicast is a relatively unstable service, especially for new sites. 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 AS. Reachability problems to other domains can be linked either to inter-domain peering misconfigurations or more fundamental protocol problems. The newness and limitations of the Multicast Source Discovery Protocol (MSDP)[13] is an example of a possible source of problems.

Trans-oceanic Connectivity Problems: Problems are also frequently caused by trans-oceanic links. In the data set, we observed many cases in which announcements originating from a site in Asia or Europe were only visible to participants in countries in those regions. Furthermore, announcements originating in the United States were only visible to US participants. We believe that these reachability problems are caused by network congestion and/or multicast connectivity problems between continents.

In this paper, we use the reachability characteristics of session announcing sites to analyze reachability characteristics for the global multicast infrastructure. In this last part of our analysis, we classify session announcing sites based on their average visibility (V_{avg}) and their non-outage rates ($R_{n/o}$). Average visibility for a site is the average of its visibility ratios during its lifetime. The non-outage ratio for a site is the ratio of the number of time intervals without a visibility loss event to its lifetime. We define *health* of a site as the product of its average visibility and its non-outage ratio. A site with very good visibility 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

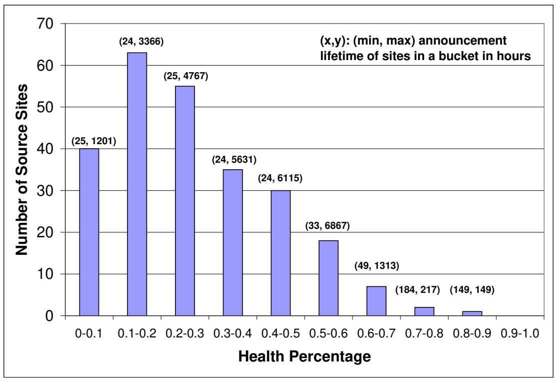

Figure 14. Grouping of session announcing sites based on their health.

Site	L_{cum}	V_{avg}	$R_{n/o}$	Health
U. Oregon	6867	0.743	0.797	0.592
NASA-Calif.	6202	0.694	0.749	0.520
UCSB	6115	0.638	0.675	0.431
George M.U.	5631	0.381	0.791	0.302
Lulea U. (SE)	5607	0.632	0.723	0.457
ENST (FR)	5513	0.472	0.801	0.378
Chiba U. (JP)	4767	0.355	0.666	0.258
ISI-Calif.	4433	0.355	0.698	0.248
LIVE-NET	4098	0.548	0.847	0.464
CISCO	4036	0.556	0.456	0.253

Table 1. Health of the 10 most active session announcing sites.

poor visibility and/or low non-outage ratio will be unhealthy. Figure 14 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 visibility. Only a few sites are unhealthy because of a poor non-outage ratio. Sites with relatively good health (over 0.6) are usually the ones with a relatively short lifetime (less than 500 hours). However, there is one exception: University of Washington has good health and a cumulative lifetime of 1313 hours. Popular/frequent session announcing sites have only average health. Table 1 shows health ratios for 10 most active session announcing sites.

5 Conclusion

In this paper we have addressed the topic of reachability monitoring for multicast. First, we have defined reachability for multicast and discussed how it is different from unicast reachability. We stress the importance of reachability monitoring in the growing hierarchical multicast infrastructure. Based on our desire to monitor reachability, we have developed a system to accomplish this function. 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 past year. With this data, we have analyzed long term reachability characteristics. Our results show that reachability has been very irregular and generally poor. We believe that the reasons for this include the newness of multicast routing protocols, and the complexity and burden of continuously monitoring the operation of multicast as a network service. Future work will target finding ways to automate reachability fault detection and to more narrowly isolate the source of discovered problems.

References

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

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

[3] 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.

[4] 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.

[5] X. Li, M. Ammar, and S. Paul, “Video multicast over the Internet,” *IEEE Network*, April 1999.

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

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

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

[9] M. Handley, “SAP: Session announcement protocol.” Internet Engineering Task Force (IETF), draft-ietf-mmusic-sap-*.txt, March 2000.

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

[11] P. Rajvaidya and K. Almeroth, “A scalable architecture for monitoring and visualizing multicast statistics,” tech. rep., University of California–Santa Barbara, 2000.

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

[13] D. Farinacci, Y. Rekhter, P. Lothberg, H. Kilmer, and J. Hall, “Multicast source discovery protocol (MSDP).” Internet Engineering Task Force (IETF), draft-farinacci-msdp-*.txt, June 1998.