Managing IP Multicast Traffic: A First Look at the Issues, Tools, and Challenges

Kevin C. Almeroth (almeroth@cs.ucsb.edu)
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5110

September 1, 1998

1 Introduction

The management of network traffic has grown into a hot area, both for research and commercial interest. And up until recently, much of the emphasis has been on managing *unicast* traffic, leaving consideration of multicast traffic lost in the shuffle. Tasks associated with multicast traffic management seem to have been equated with those of general traffic management. In other words, the thinking seems to be that if the challenges of general network management can be met, any challenges associated specifically with multicast management should, de facto, also be solved. While many of the same principles certainly apply, the premise of this paper is that multicast traffic requires different tools and strategies. This issue is addressed by looking at three broad questions, each of which is discussed in detail in the rest of this paper.

While similar techniques may be used for multicast and general network traffic management, fundamental differences in the communication paradigm for unicast and multicast suggest different management approaches are needed. Using a common starting point, a network manager should first ask the question, "What does it means to manage traffic in general, and multicast traffic in particular?" This question is addressed in detail in Section 2, but the basic answer is that the management of multicast traffic includes all of the functions associated with traditional network management, e.g. fault isolation and detection, configuration management, performance monitoring, etc. Consider an ideal situation: network management personnel monitoring network conditions from a Network Operations Center (NOC); identifying problems before, or as they are happening; and either fixing the problem from a NOC management station or dispatching repair personnel to the proper location. The end result is that network management efforts meet the goal of increasing general network reliability, and providing users with a network they can expect to function correctly.

The ideal suite of multicast management tools and strategies likely does not exist today. This raises our next question. Why has multicast-specific network management received so little attention? The basic answer is that much of the effort being devoted to multicast is focused on developing multicast protocols, maintaining reasonable quality connectivity in the Multicast Backbone (MBone)[1], evolving the MBGP multicast infrastructure[2], and providing connectivity to users in end networks. As multicast begins to mature as an Internet service, more attention will be given on how to manage it. This statement alludes to the third and final question which asks, "How long before there are tools to help manage multicast traffic?" The basic answer is that there are some freeware tools available today. A set of tools has been developed for the MBone, but they are generally not available as commercial software. The drawback of freeware tools is that they typically are not as refined as commercial tools. They tend to have a lack functionality, can be difficult to use, and can often have little customer support. However, there are several companies beginning to address the multicast management issue, and tools should soon follow. The bottom line is that multicast-specific management tools will be available when the use of multicast applications increases and users demand better multicast connectivity.

This white paper is intended to give an overview of what it means to manage multicast traffic, what tools are available today, and what tools might be available in the future. It is organized as follows: Section 2 discusses what it means to manage multicast traffic. Section 3 provides an overview of the Multicast Backbone. Section 4 lists the existing freeware tools and some techniques in use today for managing multicast traffic. Section 5 describes some recent commercial efforts. Section 6 describes some trends that will likely affect the future direction of management for multicast.

2 What Does it Mean to Manage Multicast?

The goal of network management is to organize and highlight relevant information about the network including protocols, addressing, data flow, statistics, and *especially* anomalies. This allows people without in-depth knowledge of a network's configuration to (1) monitor its operation, (2) easily identify problems when they occur, and (3) solve problems based on the availability and presentation of relevant information. The management of multicast traffic is somewhat similar to the management of unicast traffic; after all, traffic is traffic, but there are differences. The key difference derives from the simple fact that multicast traffic can be destined for multiple receivers. This difference necessarily affects the types of questions asked by network managers. With multicast, this level of abstraction carries additional importance because of the added complexity associated with delivering a packet to multiple receivers. Instead of managing/monitoring connectivity between pairs of users, multicast deals with potentially very large groups of users. And instead of managing/monitoring the links along a single path, multicast deals with links organized into a tree.

In the ideal case, multicast functions for an enterprise network should take place in a Network Operations Center (NOC). The NOC is generally a centralized facility for receiving, processing, and displaying network status information. Managing and monitoring multicast traffic is a function that can and should be conducted in the NOC and conducted in conjunction with general network management. Successful management means asking the right questions, collecting the right data, and drawing the correct conclusions about problems and events. Many of the most important issues that need to be addressed with regard to multicast management fall into one of several categories. These categories and specific questions in each include the following:

- **Traffic Management**
  - What is the total amount of multicast traffic flowing across various links in the network?
  - How much multicast traffic is flowing into and out of the network?
  - How many groups and group members are there in the network?

- Which group or source within a group is responsible for the recent jump in multicast traffic?
  - If multicast is overwhelming a particular network, how can traffic be limited?

- **Performance Monitoring**
  - Is there significant loss across links in the network?
  - Are there any links that are congested due to multicast traffic?
  - Are there any routers suffering from resource limitations (CPU or memory) due to multicast traffic?

- **Capacity Planning**
  - What is the composition of unicast and multicast traffic on a network?
  - How has the amount and usage of multicast traffic changed over time?
  - What is the usage trend for multicast traffic on a network?
  - How has the addition of multicast as a service changed (hurt or improved) network usage?

- **Fault Detection**
  - Are hosts in the network receiving the multicast traffic they are supposed to receive?
  - Is multicast traffic limited only to the network links necessary to reach group receivers (are there multicast black holes)?
  - Are there receivers in a multicast group who do not see some other receivers?

- **Fault Isolation**
  - User X just called and is not receiving traffic for a multicast group known to currently be sending data. Why not?
  - User Y just called and said traffic was being received but other group members could not see User Y's source traffic. Why not?

- **Fault Prevention**
  - The company CEO is planning to make an important announcement today. Can we evaluate reception quality to a few potential receiver locations?

3 A Brief Overview of the Multicast Backbone

Before discussing multicast management, some readers might find a brief overview of the MBone and its freeware tools useful. The Multicast Backbone (MBone) is the Internet infrastructure for delivering multicast data to Internet users. The MBone is a virtual network overlaying parts of the Internet that interconnects multicast-capable users via a series of multicast-capable routers and tunnels. The MBone was originally created as an experimental virtual network created to provide the means for multicasting data to any number of connected hosts. The motivation was to connect together a number of sites so that they could receive audio and video from the Internet Engineering Task Force (IETF) meetings. Much of the original research focus was on the deployment of multicast capabilities in routers and routing daemons running on workstations. Current work is on extending the multicast routing capability to include all of the Internet. Various protocols are being developed including the Distance Vector Multicast Routing Protocol (DVMRP)[3], Multicast Open Shortest Path First (MOSPF)[4], Protocol Independent Multicasting–Dense Mode (PIM-DM)[5], Protocol Independent Multicasting–Sparse Mode (PIM-SM)[6], and Core Based Trees (CBT)[7]. More recent work has focused on the development of an inter-domain multicast routing infrastructure including the deployment of the Multicast Border Gateway Protocol (MBGP)[8], the Multicast Source Discovery Protocol (MSDP)[9], and the Border Gateway Multicast Protocol (BGMP)[10]. As the MBone has evolved it has seen an increasingly diverse set of applications. Since the first audio conference in 1992 the MBone has seen the development of new streaming media applications using audio, video, whiteboard, and text. Other applications using non-streaming media like multicast-based caching, bulk file transfer and push-based applications require additional network services like multicast-based congestion control, reliability, and quality of service. Many of the global, publicly accessible MBone *sessions* are program broadcasts based on the common streaming media types. While there are a number of commercially available applications and tool sets (see the IPMI WWW site[11] for a list of companies), the "MBone tools" refer to the suite of freeware tools commonly used for conferencing and program broadcasts. Information about the public sessions are periodically transmitted across the MBone on a well-known multicast address. The Session DiRectory (sdr) tool[12] allows users to receive these announcements, and it creates a list of advertised sessions. Through sdr, an MBone user can choose from this list and launch the MBone tools (including audio[13], video[14], whiteboard[15], or text[16]) required to receive the component streams of a session. For each of these tools there is a multicast group which the user *joins* when the tool is started. Once part of the group, members will receive group transmissions and they can actively participate or simply listen. Joining a group means that a user must be *grafted* into the multicast tree. The existing multicast routing protocols are capable of seamlessly providing both join (graft) and leave (prune) functions.

For additional information about the MBone see the MBone list of Frequently Asked Questions[1], the unofficial MBone home page[17], and the IP Multicast Initiative (IPMI) home page[11]. MBone freeware can be found either at the home pages of individual tools (see references throughout Section 3), or at mirror sites like Merit's home page[18], the University College London home page for PC versions of the MBone tools[19], and the MBone FreeBSD home page[20].

4 Today's Tools for Managing Multicast Traffic

Much of the focus of today's multicast management tools is on debugging problems in the MBone. Many of the tools in use today have been developed by people integrally involved in the ongoing deployment and management of the MBone. Furthermore, the actual day-to-day management of the MBone is relatively unique effort in that management functions are handled informally with the main responsibilities distributed among a few dedicated individuals. As the reader will come to understand, the debugging strategies and management tools in use today have been influenced significantly by the requirement of dealing with the day-to-day problems that have arisen as the MBone grows. The key disadvantage of this effect is that today's multicast management tools require an in-depth understanding of how multicast works. The problem with this requirement is that a high proficiency in multicast operation is a difficult skill to find in most NOC personnel, and realistically should not really be needed. (NOTE: Regardless of how much abstraction a management tool might provide, there is no substitute for a good working knowledge of multicast communication. See [21] for an excellent reference on multicast operation.) Furthermore, many of the tools for multicast management are actually freeware-style debugging tools that do not offer commercial-level support, functionality, ease-of-use, or reliability. But even with these disadvantages there are a number of tools that are invaluable in aiding someone interested in observing multicast protocol operation or tracking multicast data flow.

The remainder of this section focuses on some of the most useful tools in use today. This section provides only a quick overview and the actual use of these tools is left to other resources. In particular, the "Multicast Debugging Handbook"[22], is a Internet Draft offering a comprehensive source of information on both techniques for isolating common problems, and tools to aid the multicast network manager. Furthermore, additional information on the key debugging tools listed below can be found at the respective WWW sites and URLs given for each tool.

- **Mrinfo**: shows the multicast tunnels and routes for a router/mrouted.
- **Mtrace**: traces the multicast path between two hosts.
- **RTPmon**: displays receiver loss collected from RTCP messages.
- **Mhealth**: monitors tree topology and loss statistics.
- **Multimon**: monitors multicast traffic on a local area network.
- **Mlisten**: captures multicast group membership information.
- **Dr. Watson**: collects information about protocol operation.

4.1 Mrinfo

The mrinfo[23] command gives information about the current status of a multicast router or mrouted. Information returned by mrinfo includes the set of tunnels and/or interfaces on which multicast is enabled or disabled. The mrinfo command can be used to determine if certain tunnels are up and functioning properly. The mrinfo command also provides information about the metric (first number after the host name) and time to live threshold (second number after the host name) for each interface. A sample output from an mrinfo command is:

```
130.207.244.30 (feta-fddi.gatech.edu) [version 3.255]:
  130.207.166.214 -> 0.0.0.0 (local) [1/1/querier/leaf]
  130.207.244.30 -> 0.0.0.0 (local) [1/1/disabled]
  130.207.244.30 -> 4.0.35.20 (f0.atlanta1-mbone1.bbnplanet.net) [1/32/tunnel/leaf]
  130.207.244.30 -> 198.79.7.99 (bstfirewall.atglab.bls.com) [1/16/tunnel/leaf]
  130.207.244.30 -> 198.79.12.196 (198.79.12.196) [1/16/tunnel/down/leaf]
  130.207.244.30 -> 199.77.254.6 (199.77.254.6) [1/16/tunnel/leaf]
  130.207.244.30 -> 170.140.150.17 (mathsunf.mathcs.emory.edu) [1/16/tunnel/down/leaf]
  130.207.244.30 -> 199.77.249.10 (robin.fernbank.edu) [1/16/tunnel/leaf]
```

4.2 Mtrace

The mtrace[24] command is used to return a snapshot of the set of links used to connect a particular source with a particular destination. Additional mtrace options allow a user to see the number of multicast packets per second flowing across each hop. When mtrace is given a particular multicast group address it will return losses per hop for that particular multicast address. The mtrace tool is one of the best ways of discovering how multicast packets are flowing through a network and determining heavily congested tree links. However, mtrace does not work 100% of the time. There are a number of reasons, one of the biggest being improper mtrace support in multicast routers. Another common reason for failure is heavily congested links which leads to lost mtrace packets, missing routing state, and a general inability of mtrace to get at information it needs to get at. One of the dangers of mtrace is that the tool itself can cause additional congestion. The additional load required of a router responding to an mtrace packet may increase congestion. For this reason, many router vendors give a low priority to mtrace requests. So while packets may be flowing and congestion seems low, an mtrace request may be ignored by a router in favor of committing resources to better routing performance. A sample mtrace is as follows:

```
Mtrace from 192.9.9.71 to 132.180.15.11 via group 224.2.172.238
Querying full reverse path...
  0  btr0xb.rz.uni-bayreuth.de (132.180.15.11)
 -1  btr0xb.rz.uni-bayreuth.de (132.180.15.11)  DVMRP  thresh^ 1   [default]
 -2  btrzw3.dvmrp.uni-bayreuth.de (132.180.11.3)  PIM/Special  thresh^ 1   [default]
 -3  ds9.gate.uni-erlangen.de (131.188.6.3)  PIM/Special  thresh^ 24  Reached RP/Core
 -4  mr-nuernberg1.win-ip.dfn.de (188.1.207.1)  PIM/Special  thresh^ 32
 -5  mr-stuttgart1.win-ip.dfn.de (188.1.200.5)  PIM/Special  thresh^ 32
 -6  dec3800-1-fddi-0.Washington.mci.net (204.70.2.13)  DVMRP  thresh^ 64
 -7  dec3800-1-fddi-1.WestOrange.mci.net (204.70.64.45)  DVMRP  thresh^ 1
 -8  dec3800-2-fddi-1.WestOrange.mci.net (204.70.64.77)  DVMRP  thresh^ 1
 -9  e1.cambridge1-mbone1.bbnplanet.net (199.94.207.2)  PIM/Special  thresh^ 32
-10  f0.paloalto-mbone1.bbnplanet.net (131.119.0.197)  PIM/Special  thresh^ 32
-11  mbone.Sun.COM (192.9.9.71)  DVMRP  thresh^ 4
-12  mbone.Sun.COM (192.9.9.71)
Round trip time 389 ms; total ttl of 70 required.

Waiting to accumulate statistics... * Results after 12 seconds:

  Source        Response Dest    Overall     Packet Statistics For Traffic From
192.9.9.71      224.0.1.32       Packet      192.9.9.71 To 224.2.172.238
     v       __/  rtt  377 ms     Rate       Lost/Sent = Pct  Rate
192.9.9.71      mbone.Sun.COM
     v     ^      ttl    5         4 pps        0/50   =  0%   4 pps
192.42.110.249
131.119.0.197   f0.paloalto-mbone1.bbnplanet.net
     v     ^      ttl   34       118 pps        2/50   =  4%   4 pps
199.94.207.2    e1.cambridge1-mbone1.bbnplanet.net
     v     ^      ttl   35       109 pps        0/48   =  0%   4 pps
204.70.64.61
204.70.64.77    dec3800-2-fddi-1.WestOrange.mci.net
     v     ^      ttl   35        52 pps        0/48   =  0%   4 pps
204.70.64.45    dec3800-1-fddi-1.WestOrange.mci.net
     v     ^      ttl   35       114 pps        1/48   =  2%   4 pps
```

```
204.70.2.13     dec3800-1-fddi-0.Washington.mci.net
     v     ^      ttl   70       235 pps        8/47   = 17%   3 pps
193.174.226.254
188.1.200.5     mr-stuttgart1.win-ip.dfn.de
     v     ^      ttl   70       279 pps        0/39   =  0%   3 pps
188.1.200.6
188.1.207.1     mr-nuernberg1.win-ip.dfn.de
     v     ^      ttl   70       286 pps        0/39   =  0%   3 pps
131.188.6.3     ds9.gate.uni-erlangen.de Reached RP/Core
     v     ^      ttl   70       102 pps        0/39   =  0%   3 pps
132.180.13.3
132.180.11.3    btrzw3.dvmrp.uni-bayreuth.de
     v     ^      ttl   70         6 pps        0/39   =  0%   3 pps
132.180.15.11   btr0xb.rz.uni-bayreuth.de
     v      \__   ttl   70         7 pps        ?/39           3 pps
132.180.15.11   128.111.52.10
  Receiver      Query Source
```

Notice in the trace that because we included a specific multicast group we see loss percentages for each hop. The trace shows that there is 17% loss along a path between Washington DC and Stuttgart, Germany. This makes sense since trans-oceanic links are typically congested. The first part of the mtrace also shows the various protocols that are being used at each router, and the TTL threshold required for packets to make it past that link. Finally, notice that the trace uses a reverse path approach and starts at the receiver and traces back to the source. Since most multicast algorithms use a reverse shortest path algorithm this trace technique is appropriate.

4.3 RTPmon

The RTPmon tool[25], coupled with mtrace, is one of the most powerful tools currently available for monitoring active multicast groups. RTPmon joins a particular multicast group address and receives feedback reports from all receivers. These feedback reports are generated by the Real-Time Control Protocol (RTCP) which is part of the Real-Time Protocol (RTP)[26]. The loss rates for each receiver for each source are displayed in a real-time table (see Figure 1). By “clicking” on a particular cell in the table, additional information about loss and jitter are displayed. Buttons in the RTPmon interface allow a user to execute an mtrace for the particular group, source, and destination. Together, these two tools allow a user to monitor the quality of a multicast transmission.

4.4 Mhealth

The mhealth[27] is a soon-to-be-released tool that takes RTPmon a step further by displaying a real-time, graphical representation of a particular group’s multicast tree including loss information. The addition of tree structure information allows a network manager to better “see” where loss in the network is occurring. The mhealth tool presents its results in real-time using Java and also archives topology and loss information for future analysis similar to what was conducted in [28]. The mhealth tool is written in Java and so is relatively platform independent but it relies on an installed version of mtrace. Figure 2 is a snapshot of an early version of the mhealth tool.

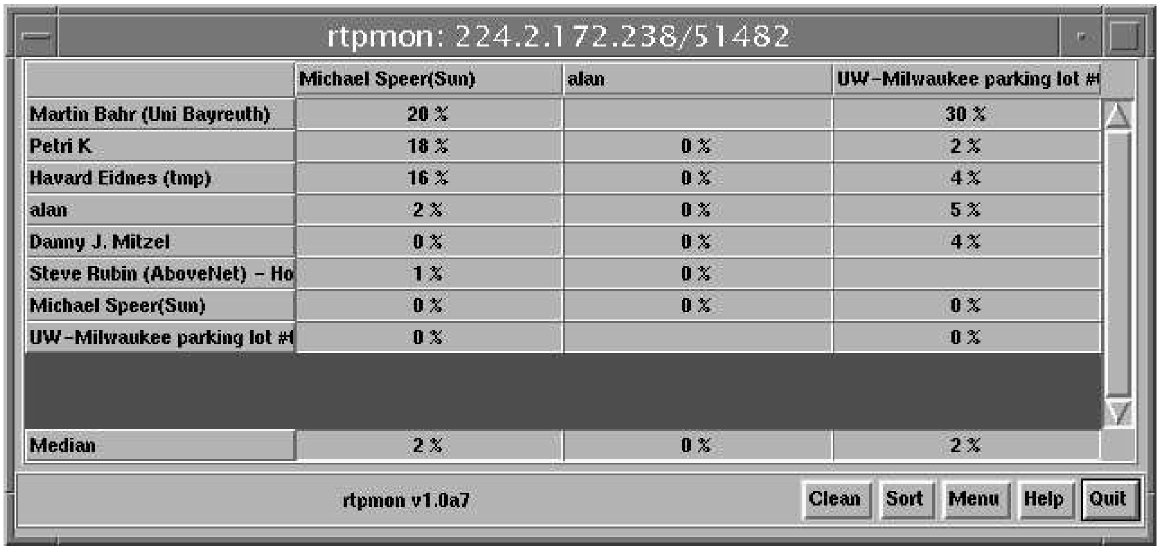


Figure 1: A snapshot of the rtpmon tool.

4.5 Multimon

The multimon tool[29] takes the opposite approach to mhealth. Instead of presenting information about the end-to-end performance of a particular multicast group, it gives information on all of the multicast traffic flowing on a particular LAN. Using a modified version of TCPdump[30], which snoops packets from the network, multimon collects statistics about the amount and types of traffic flowing across a LAN. Multimon is quite useful in monitoring traffic flowing across particular LANs (see Figure 3. However, multimon was developed primarily for use on UNIX-based systems. Furthermore, getting multimon running, even on UNIX systems, can be something of a challenge. The tool suite requires Tcl 8.0, Tk 4.2, a modified TCPdump (included), xplot (included), tcl-dp (included), and stooop (included). While many of the required files are included, it can be time consuming to install all the tools and ensure everything is working properly.

4.6 Mlisten

The mlisten tool[31] is more of a research tool, but it has potential uses for monitoring group membership in an enterprise. Its basic purpose is to collect group membership data for a specific set of groups. This set of groups is gathered from the sessions advertised via sdr. Mlisten typically collects data for all advertised groups, but it can be tailored to listen to only specific groups with a certain characteristics (like TTL values that are less than 16). This tool could be useful for debugging if group membership within a specific domain needs to be known. The danger of the mlisten tool is that it joins all groups that it monitors and could therefore become a traffic sink. Currently, there are only binary versions available for Sun UNIX (both SunOS and Solaris). One version has a graphical interface and allows a manager to see group size in real-time (see Figure 4). A user can also click on a group and see a list of the group's members by IP address.

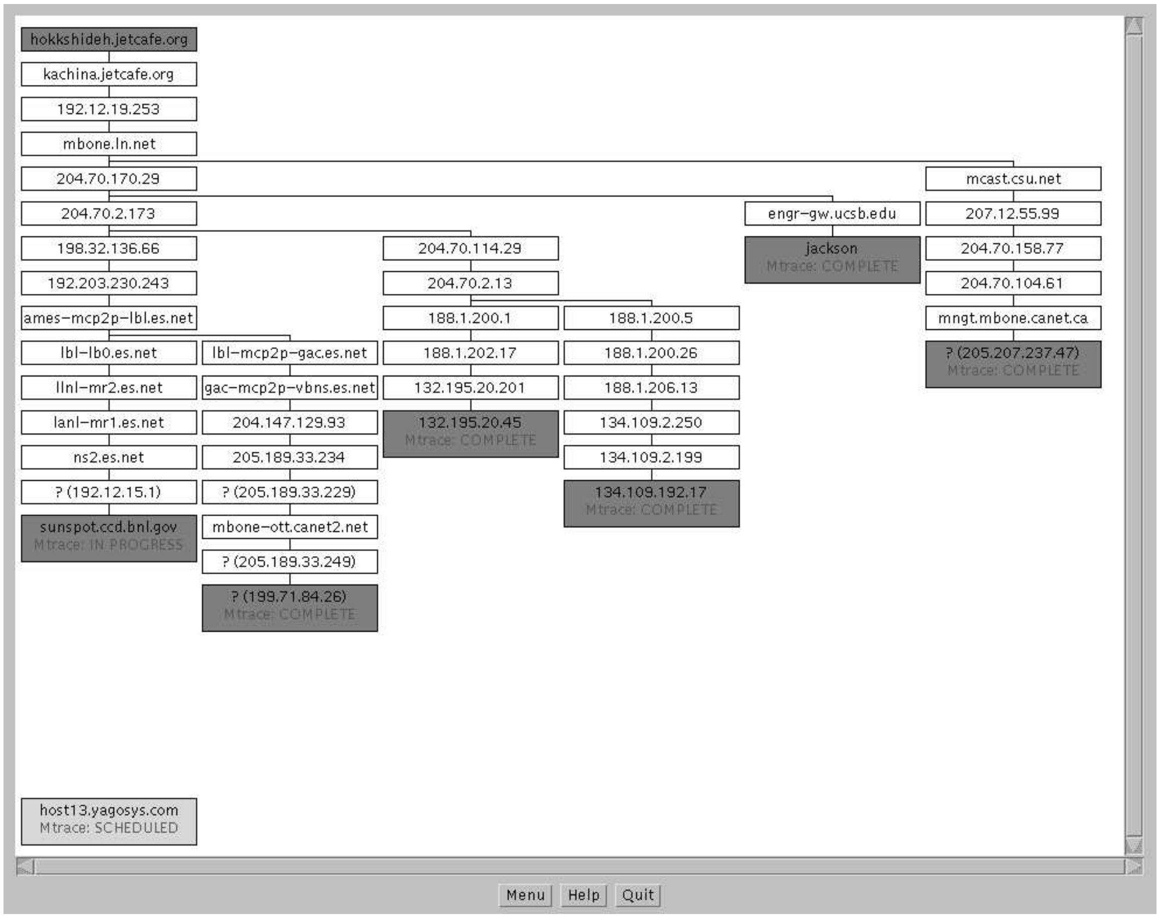


Figure 2: A snapshot of the mhealth tool.

4.7 Dr. Watson

The Dr. Watson tool[32] is a multi-purpose diagnostic tool. It is able to perform reachability tests, generate SNMP queries, view network traffic, and examine routing table information. Dr. Watson implements a number of unicast and multicast protocols. Its usefulness for multicast debugging is its ability to monitor traffic, source multicast traffic (as a test source), and transmit IGMP messages to the router. This last function is particularly useful in that a user can test whether the router and other LAN hosts are responding properly to multicast group join and prune messages. One advantage of Dr. Watson is that it is a commercially supported tool and comes with reasonable documentation and support. The disadvantage of Dr. Watson is that it is a very low level tool and addresses a very specific set of potential network problems. As such, Dr. Watson requires a fair amount of expertise to use and may not be useful for larger-scale network management or multicast routing problems.

4.8 SNMP-Based Tools and Multicast Related MIBs

In addition to the management and debugging tools already described, there are also Management Information Bases (MIBs) for multicast[33]. MIBs are accessed using the Simple Network Management Protocol

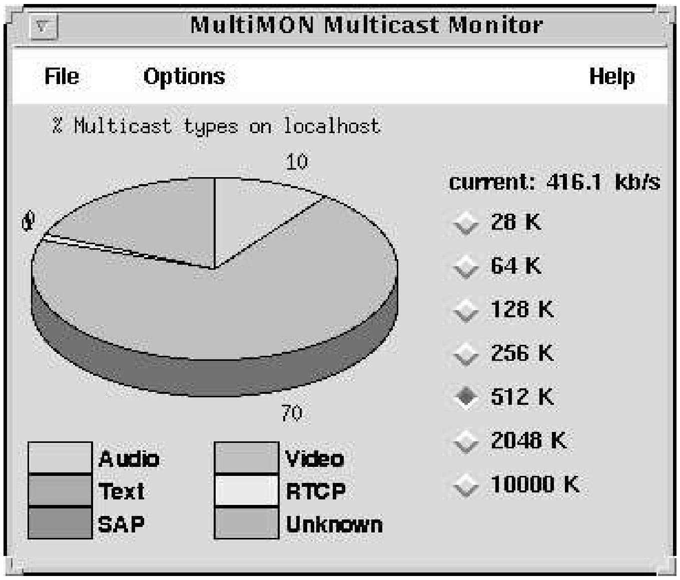
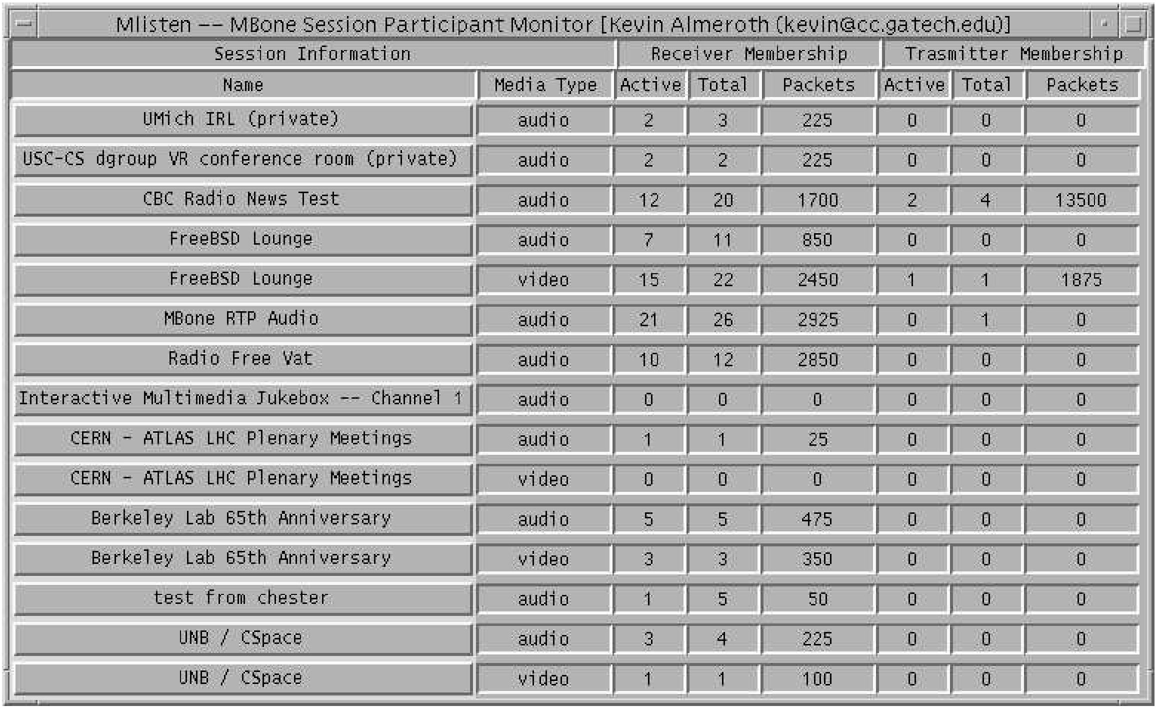


Figure 3: A snapshot of the multimon tool.

(SNMP)[34, 35]. The reader should understand that much of the management and debugging functions carried on in the MBone are done without the assistance of SNMP and MIBs. The reason for this again has to do with the evolution of the MBone. People working on the MBone were more focused on establishing basic connectivity than developing robust management tools. For this reason, there are more non-SNMP-based tools than there are SNMP-based tools. However, this trend will likely change as the demand for better multicast management tools is made. The primary reason why SNMP-based tools could potentially be useful is that SNMP is a well understood protocol and paradigm for managing networks. And while SNMP tools are not commonly used in the MBone, several MBone researchers have come to understand the importance of specifying multicast MIBs. There are now a number of MIBs that have been proposed and exist in IETF Internet Draft form. These MIBs are undergoing revisions and have not yet progressed to RFCs.

- **RTP MIB**: The RTP MIB[36] is designed to be used by either host running RTP applications or intermediate systems acting as RTP monitors. There are tables defined for each type of user. The monitor portion of the MIB is designed to collect statistical data about RTP sessions. The RTP MIB document states that this data can be used for "capacity planning and other network management purposes." The RTP MIB can also be used to diagnose and isolate faults.
- **Basic Multicast Routing MIB**: The IP multicast routing MIB[37] includes only general data about multicast routing. Data for specific protocols are included in other MIBs (see next bullet). This MIB contains information about multicast group and source pairs; next hop routing state, forwarding state for each of a router's interfaces, and information about multicast routing boundaries.
- **Protocol-Specific Multicast Routing MIBs**: The protocol-specific MIBs provide information specific to a particular routing protocol. In addition to MIBs for true routing protocols there is also a MIB for multicast tunnels and a MIB for IGMP. The list of available MIBs is as follows:



Mlisten -- MBone Session Participant Monitor [Kevin Almeroth (kevin@cc.gatech.edu)]

Session Information		Receiver Membership			Trasmitter Membership		
Name	Media Type	Active	Total	Packets	Active	Total	Packets
UMich IRL (private)	audio	2	3	225	0	0	0
USC-CS dgroup VR conference room (private)	audio	2	2	225	0	0	0
CBC Radio News Test	audio	12	20	1700	2	4	13500
FreeBSD Lounge	audio	7	11	850	0	0	0
FreeBSD Lounge	video	15	22	2450	1	1	1875
MBone RTP Audio	audio	21	26	2925	0	1	0
Radio Free Vat	audio	10	12	2850	0	0	0
Interactive Multimedia Jukebox -- Channel 1	audio	0	0	0	0	0	0
CERN - ATLAS LHC Plenary Meetings	audio	1	1	25	0	0	0
CERN - ATLAS LHC Plenary Meetings	video	0	0	0	0	0	0
Berkeley Lab 65th Anniversary	audio	5	5	475	0	0	0
Berkeley Lab 65th Anniversary	video	3	3	350	0	0	0
test from chester	audio	1	5	50	0	0	0
UNB / CSpace	audio	3	4	225	0	0	0
UNB / CSpace	video	1	1	100	0	0	0

Figure 4: A snapshot of the mlisten tool.

– **PIM MIB**: The PIM MIB[38] contains information about the PIM interfaces that are configured; the router's PIM neighbors; the set of of rendezvous points and an association for the multicast address prefixes; the list of groups for which this particular router should advertise itself as the candidate rendezvous point; the reverse path table for active multicast groups; and component table with an entry per domain that the router is connected to.

– **CBT MIB**: The CBT MIB[39] contains information about the configuration of the router including interface configuration; router statistics for multicast groups; state about the set of group cores, either generated by automatic bootstrapping or by static mappings; and configuration information for border routers.

– **DVMRP MIB**: The DVMRP MIB[40] contains configuration information; interface configuration and statistics; peer router configuration states and statistics; the state of the DVMRP routing table; and information about key management for DVMRP routes.

– **Tunnel MIB**: The Tunnel MIB[41] includes information about the tunnels that might be supported by a router or host. The table supports tunnel types including Generic Routing Encapsulation (GRE) tunnels, IP-in-IP tunnels, minimal encapsulation tunnels, layer two tunnels (LTTP), and point-to-point tunnels (PPTP).

– **IGMP MIB**: The Internet Group Management Protocol (IGMP) only deals with determining if packets should be forwarded over a particular leaf router interface. IGMP is not a routing protocol, but manages group membership between hosts and routers. The IGMP MIB[42] contains information about the set of router interfaces that are listening for IGMP messages, and a table with information about which interfaces currently have members listening to particular multicast groups.

Two important freeware tools which work with multicast MIBs are mstat[43] and mview[44]. Both are tools produced by the Merit SNMP-Based Management Project[45]. Mstat queries a router or SNMP-capable mrouted to generate various tables of information including routing tables, interface configurations, cache contents, etc. Mview is an, "application for visualizing and managing the MBone. Mview allows the user to display and interact with the topology of the MBone in various ways, to collect and monitor performance statistics on routers and links, and can aid in diagnosing network problems."

5 Recent Commercial Efforts

The evolution of commercial tool suites to support multicast management is only just beginning. The tools that are being deployed are evolving from traditional, unicast-based management platforms. Developers are slowly beginning to add functionality to support multicast communication. Two efforts profiled in this section are extensions to OpenView by Hewlett-Packard Labs and Chariot by Ganymede Software.

5.1 OpenView: MMap, MMon, and Policy-based Control

Researchers at HP Laboratories are investigating IP multicast network management and are currently building a prototype integrated with HP OpenView. It is intended for use by the network operators who are not experts in IP multicast. The prototype provides discovery, monitoring and fault detection capabilities.

The prototype automatically discovers and monitors the status of IP multicast routers and topology, including tunnels. From a visual map, an operator can see at a glance the state of the multicast infrastructure. An operator can query to display the multicast traffic activity on the network by multicast group and in aggregate across network interfaces. This allows an operator to isolate faults in multicast transmissions and to note the loading that multicast traffic has on the network. Various faults in multicast topology are automatically identified and presented to the operator as alerts in OpenView.

The research prototype uses standard IGMP and SNMP protocols to collect it's information. It will be suitable for test deployment in late February 1999. If you are interested in serving as a test site, or if you wish to discuss this or other issues in IP multicast management, contact Radhika Malpani (radhika@hpl.hp.com).

HP is also developing policy-based product to control IP multicast transmissions, allowing a network manager to limit the impact of IP multicast traffic. Through a common policy-based network management GUI, a network operator would instruct routers (via TELNET/CLI, SNMP, etc.) to restrict the forwarding of IP multicast packets.

5.2 Chariot Tool Suite

Chariot, developed by Ganymede Software (http://www.ganymedeSoftware.com/), is a network performance test tool that allows distributed, end-to-end performance tests anywhere in a network. It provides a highly flexible way to test a wide variety of local and wide area networking environments and infrastructures. With

Chariot, it is possible to run remote multi-protocol tests between many different operating systems, all from a single console that can be located anywhere in the enterprise. Chariot provides an accurate view of how applications and equipment will perform in today's complex network environments.

Chariot provides support to test networks with real-world network traffic (applications such as e-mail, database updates, multimedia conferencing, Web). Unlike packet generators, which produce a steady stream of unchanging data, Chariot tests generate application traffic that is bi-directional, variable, and interacts with the protocol stack. Chariot tests run over multiple protocols (including several multicast protocols). Because the application scripts interact directly with the network protocol stack, tests measure performance degradation caused by lost frames, timeouts, and congestion control mechanisms.

Ganymede Software's Network Performance Endpoint technology includes two key components: Network Performance Endpoints and Application Scripts. A Network Performance Endpoint (NPE) is a "skinny" software agent installed on computers throughout a network. The endpoint receives instructions from the Chariot console and executes tests using the application scripts. Application scripts define how application network traffic is generated by the test. Application scripts make the same API calls to the network protocol stacks and invoke the same load on the stack as production applications. Each script consists of communication commands, such as SEND and RECEIVE, along with script variables, such as buffer size and data type, that can be modified by the Chariot user. Application scripts can emulate anything from a simple file transfer to a complicated SAP/R3 transaction or a streaming multimedia application. Chariot tells an endpoint how to emulate a particular application by sending it an application script and other test setup information. The script describes the type and amount of data to send and receive, when to connect and disconnect, and what delays should be used to emulate end-user or application overhead. The receiving endpoint (NPE 1) keeps its half of the test and sends the other half to its partner (NPE 2). The endpoints then run the test. At the end of the test, NPE 1 sends the test results back to the Chariot console. The end-user can then use that data to evaluate the network performance or export it to text, spreadsheet, and HTML for later use.

6 The Future of Multicast Traffic Management

The management of multicast traffic is obviously an important function. While MBone engineers have had to debug problems and manage multicast traffic for a number of years, the strategy has been to build tools on an as-needed basis. These tools have mostly been designed for a specific function, and used by people who have in-depth knowledge of the topology, functions, and limitations of multicast. There have been only a few concerted efforts by companies to build commercial tool suites (see Section 5). One reason for this lack of commercial interest might be the well known chicken-and-egg problem – without user/customer demand there are no commercial management products and without products, network managers are less willing and able to support multicast as a network service. What is just now beginning to happen is that companies with multicast experience or some company with network management experience will step forward and offer a product that will integrate itself into an existing network management product. This has happened with the MBone tools themselves. Originally the MBone tools were developed and supported by individuals in the research community, but as demand for multicast-based services has grown, it has spurred the development of commercial tools. Now, companies like Precept/Cisco, Icast, Real Networks, and Microsoft (to name but a few) have started offering commercial, fully supported versions of the MBone tools.

In the near term, users can expect a growing effort to integrate and improve SNMP-based management functions into commercial management packages. As this effort gains momentum and as more groups (ISPs, companies, etc) deploy multicast in their enterprise, there will be more demand for these types of functions. Furthermore, as our understanding of what it means to manage multicast grows, we will be able to develop better ways of using the data available. Also in the near term, new debugging tools will be developed for multicast. For example, consider the development of new tools like mhealth and the focused attention on monitoring and tracking MBone growth[46] and performance[28].

In the longer term, users can expect companies to focus strongly on providing an integrated management package that combines debugging tools and network management tools. In some cases, these tools will be integrated into the framework of existing management systems. One example of work like this already in progress is Hewlett Packard's attempts to add multicast functionality to OpenView. Using interface features familiar to many users will go a long way in making multicast traffic easier to manage.

Finally, in addition to end-user products there is also a growing interest in adding additional functionality to internal network components. In particular, an effort being led by Cisco Systems will standardize a protocol for facilitating multicast fault detection, isolation, and prevention capabilities to routers. The Multicast Routing Monitor (MRM) protocol[47] is being developed and standardized through the IETF. MRM protocol provides on-demand multicast group creation (including sources and receivers) to detect and isolate faults. MRM is designed to fill a niche not covered by SNMP, RTCP, or existing debugging tools. While MRM has great potential, it is still evolving. However, the need for this type of protocol has created some pressures to move quickly and a standard along with beta code may be available as early as mid-1999.

References

[1] S. Casner, *Frequently Asked Questions(FAQ) on the Multicast Backbone(MBone)*. USC/ISI, December 1994. Available from ftp://ftp.isi.edu/mbone/faq.txt.

[2] H. LaMaster, S. Shultz, J. Meylor, and D. Meyer, "Multicast-friendly internet exchange (MIX)," Tech. Rep. draft-ietf-mboned-mix-*.txt, Internet Engineering Task Force (IETF), December 1998.

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

[4] J. Moy, "Multicast extensions to OSPF," Tech. Rep. RFC 1584, Internet Engineering Task Force (IETF), March 1994.

[5] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, A. Helmy, D. Meyer, and L. Wei, "Protocol independent multicast version 2 dense mode specification," Tech. Rep. draft-ietf-pim-v2-dm-*.txt, Internet Engineering Task Force (IETF), November 1998.

[6] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol independent multicast sparse-mode (PIM-SM): Protocol specification," Tech. Rep. RFC 2362, Internet Engineering Task Force (IETF), June 1998.

[7] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT): An architecture for scalable multicast routing," in *ACM Sigcomm 93*, (California, USA), pp. 85–95, September 1995.

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

[9] D. Farinacci, Y. Rekhter, P. Lotherberg, H. Kilmer, and J. Hall, "Multicast source discovery protocol (MSDP)," Tech. Rep. draft-farinacci-msdp-*.txt, Internet Engineering Task Force (IETF), June 1998.

[10] D. Thaler, D. Estrin, and D. Meyer, "Border gateway multicast protocol (BGMP): Protocol specification," Tech. Rep. raft-ietf-idmr-gum-*.txt, Internet Engineering Task Force (IETF), November 1998.

[11] *IP Multicast Initiative (IPMI) Home Page.* http://www.ipmulticast.com/.

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

[13] V. Jacobson and S. McCanne, *VAT: Visual Audio Tool.* Lawrence Berkeley Laboratory (LBL), February 1992. Available from ftp://ee.lbl.gov/conferencing/vat/.

[14] S. McCanne and V. Jacobson, *VIC: Video Conference Tool.* Lawrence Berkeley National Laboratory (LBL), November 1994. Available from ftp://ee.lbl.gov/conferencing/vic/.

[15] V. Jacobson and S. McCanne, *WB: Whiteboard Tool.* Lawrence Berkeley Laboratory (LBL), July 1994. Available from ftp://ee.lbl.gov/conferencing/wb/.

[16] M. Handley and J. Crowcroft, "Network text editor (NTE): A scalable shared text editor for the MBone," in *ACM Sigcomm 97*, (Canne, FRANCE), August 1997.

[17] *MBone Home Page.* http://www.mbone.com/.

[18] *Merit's MBone Home Page.* http://www.merit.edu/~mbone/index/titles.html.

[19] *PC-specific MBone Tools.* http://scrappy.cs.ucl.ac.uk/shrimp/update.html.

[20] *FreeBSD MBone Information and Tools.* http://www.cs.ubc.ca/spider/mjmccut/freebsd.html.

[21] T. Maufer, *Deploying IP Multicast in the Enterprise.* Upper Saddle River, New Jersey: Prentice Hall, Inc., 1998.

[22] D. Thaler and B. Aboba, "Multicast debugging handbook," Tech. Rep. draft-ietf-mboned-mdh-*.txt, Internet Engineering Task Force (IETF), March 1997.

[23] B. Fenner and et al., *mrouted 3.9-beta, mrinfo, and other tools*, March 1998. Available from ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/.

[24] B. Fenner and et al., *Multicast Traceroute (mtrace) 5.1*, December 1996. Available from ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/.

[25] A. Swan and D. Bacher, *rtpmon 1.0a7*, January 1997. Available from ftp://mm-ftp.cs.berkeley.edu/pub/rtpmon/.

[26] H. Schulzrinne, S. Casner, R. Frederick, and J. V., "RTP: A transport protocol for real-time applications," Tech. Rep. RFC 1889, Internet Engineering Task Force, January 1996.

[27] D. Makofske and K. Almeroth, *Mhealth – Real-Time Multicast Tree Health Monitoring Tool*, August 1998. Available from http://imj.ucsb.edu/mhealth/.

[28] M. Handley, "An examination of MBone performance," Tech. Rep. ISI/RR-97-450, University of Southern California (USC), Information Sciences Institute (ISI), January 1997. Available from ftp://north.east.isi.edu/~mjh/mbone.ps.

[29] J. Robinson and J. Stewart, *MultiMON 2.0 – Multicast Network Monitor*, August 1998. Available from http://www.merci.crc.ca/mbone/MultiMON/.

[30] e. a. Jacobson, V., *TCPdump 3.4*. Network Research Group–Lawrence Berkeley Laboratory (LBL), January 1998. Available from ftp://ee.lbl.gov/tcpdump.tar.Z.

[31] K. Almeroth, *Multicast Group Membership Collection Tool (mlisten)*. Georgia Institute of Technology, September 1996. Available from http://www.cc.gatech.edu/computing/Telecomm/mbone/.

[32] K. Auerbach, *DWTNDA: Dr. Watson, The Network Detective's Assistant*, November 1997. Available from http://www.cavebear.com/dwtnda/spd.html.

[33] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Structure of management information for version 2 of the simple network management protocol (SNMPv2)," Tech. Rep. RFC 1902, Internet Engineering Task Force (IETF), January 1996.

[34] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Protocol operations for version 2 of the simple network management protocol (SNMPv2)," Tech. Rep. RFC 1905, Internet Engineering Task Force (IETF), January 1996.

[35] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple network management protocol," Tech. Rep. RFC 1157, Internet Engineering Task Force (IETF), May 1990.

[36] M. Baugher, J. Du, and S. Naudus, "Real-time transport protocol management information base," Tech. Rep. draft-ietf-avt-rib-mib-*.txt, Internet Engineering Task Force (IETF), November 1997.

[37] K. McCloghrie, D. Farinacci, and D. Thaler, "IP multicast routing MIB," Tech. Rep. draft-ietf-idmr-multicast-routmib-*.txt, Internet Engineering Task Force (IETF), November 1997.

[38] K. McCloghrie and D. Thaler, "Protocol independent multicast MIB," Tech. Rep. draft-ietf-idmr-pim-mib-*.txt, Internet Engineering Task Force (IETF), November 1997.

[39] A. Ballardie and D. Thaler, "Core based trees (CBT) multicast routing MIB," Tech. Rep. draft-ietf-idmr-cbt-mib-*.txt, Internet Engineering Task Force (IETF), January 1998.

[40] D. Thaler, "Distance-vector multicast routing protocol MIB," Tech. Rep. draft-thaler-dvmrp-mib-*.txt, Internet Engineering Task Force (IETF), May 1998.

[41] D. Thaler, "IP tunnel MIB," Tech. Rep. draft-thaler-tunnel-mib-*.txt, Internet Engineering Task Force (IETF), November 1997.

[42] K. McCloghrie, D. Farinacci, and D. Thaler, "Internet group management protocol MIB," Tech. Rep. draft-ietf-idmr-igmp-mib-*.txt, Internet Engineering Task Force (IETF), February 1998.

[43] *Mstat utility*. Available as part of mrouted at ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/.

[44] *Mview utility*. http://www.merit.edu/~mbone/mviewdoc/Welcome.html.

[45] *Merit SNMP-Based MBone Management Project*. http://www.merit.edu/net-research/mbone/.index.html.

[46] K. Almeroth and M. Ammar, "Multicast group behavior in the Internet's multicast backbone (MBone)," *IEEE Communications*, vol. 35, pp. 224–229, June 1997.

[47] L. Wei and D. Farinacci, "Multicast routing monitor (MRM)," Tech. Rep. draft-ietf-mboned-mrm-*.txt, Internet Engineering Task Force (IETF), February 1998.

Author's Biography

Kevin Almeroth [http://www.cs.ucsb.edu/~almeroth/] is an assistant professor at the University of California in Santa Barbara where he is part of the Networking and Multimedia Systems Group. His research interests include computer networks and protocols, multicast communication, large-scale multimedia systems, and performance evaluation. Some specific research projects include the Interactive Multimedia Jukebox (IMJ) [http://imj.gatech.edu/], a WWW-based scheduling system for the transmission of on-demand content using multicast communication. A paper describing the IMJ won the Best Paper Award at the WWW7 conference. Another project is the long-term collection of multicast group membership data for use in measuring the growth and health of multicast communication in the Internet. In addition to these research projects, Dr. Almeroth is involved in the design of the California Research and Education Network (CalREN-2), a high-speed backbone to connect University of California schools and California research labs. He is working to develop the Media Arts and Technology Program (MATP) at UCSB. Outside of UCSB, Dr. Almeroth's work with multicast has lead to participation in a number of organizations. He is an active participant in several Internet Engineering Task Force (IETF) working groups; has helped manage multicast for Networld+Interop as part of the Network Operations Center (NOC) team; is a Senior Technologist for the IP Multicast Initiative (IPMI) sponsored by Stardust Forums; and is the multicast working group chair for Internet2.