# A Router-Based Technique for Monitoring the Next-Generation of Internet Multicast Protocols

*Prashant Rajvaidya*
Department of Computer Science
University of California
Santa Barbara, CA 93106-5110
prash@cs.ucsb.edu

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

*Abstract*— **Most of the tools available for monitoring multicast have only marginal utility in today's multicast infrastructure. These tools lack the ability to handle the latest routing protocols because they rely on application layer data like the Realtime Transport Protocol (RTP). Our goal is to develop a system that monitors multicast at the network layer; that can provide functionality for all of the network management tasks; and that can be easily extended to monitor the evolving set of multicast protocols. To this end, we have developed Mantra, a tool for collecting data from multicast routers, analyzing the collected data and presenting realtime results. Using collected data, Mantra creates interactive graphs depicting the state of the multicast infrastructure. The results from Mantra are being used to monitor multicast usage and deployment, as well as routing protocol deployment and performance. In addition, the results are being used to detect common anomalies and to troubleshoot frequent routing problems. In this paper, we discuss the issues related to multicast monitoring, describe the existing efforts in the field and present the design of Mantra. In addition, we present results based on data collected over a 6 months period by monitoring the multicast traffic at two points: (1) at the Federal IntereXchange–West (FIXW) router, and (2) an on-campus router.**

## I. Introduction

Over the last decade, a myriad of new Internet applications have evolved that require transfer of high bandwidth media streams to a large number of users. Because traditional Internet protocols are mostly unsuitable for such traffic flows, several solutions have been proposed as efficient alternatives. Among these is multicast communication[1]. Multicast is a mechanism for one-to-many and many-to-many delivery of data over the Internet in a bandwidth efficient manner.

During the first five years of its deployment multicast existed as a virtual and experimental network on top of the existing Internet. This topology called, the Multicast Backbone (MBone)[2], used the Distance Vector Multicast Routing Protocol (DVMRP). However, more recently, multicast has evolved beyond the MBone into a native multicast infrastructure. Protocols like Protocol Independent Multicast - Dense Mode (PIM-DM), Protocol Independent Multicast - Sparse Mode (PIM-SM), the Multicast Border Gateway Protocol (MBGP) and the Multicast Source Discovery Protocol (MSDP) are in use[3].

The rapid evolution of multicast has been accomplished with only a few monitoring and debugging tools[4]. Tool development and monitoring efforts have not kept pace with the growth in deployment. Several existing tools, including *mrinfo*[5], *mwatch*[6] and *mrtree*[7], were developed in the context of the MBone and its flat routing topology. As a result, their utility has been marginalized as the topology has shifted towards hierarchical interdomain routing. Existing tools also provide only limited data collection, analysis and display functionality. In addition, many tools collect monitoring data based on application layer protocols, e.g., the Realtime Transport Protocol (RTP)[8]. Their weakness in this respect is that if the application does not send the right data, or connectivity is broken, the data either is not representative or is not available at all.

Limitations of existing monitoring tools and the need for newer monitoring mechanisms has motivated us to develop *Mantra*, a tool to monitor multicast at the network layer. We have developed Mantra to monitor the latest set of routing protocols. Mantra can perform network management tasks like monitoring the network, generating real-time results, presenting results intuitively, and providing useful analysis. Mantra is a Java-based tool that can be easily deployed across different platforms. Its basic operations include (1) collecting data by periodically capturing multicast router state and (2) processing collected data to generate monitoring results. Other tasks include, data logging and Web-based presentation of results. Mantra's ability to collect data at the router level enables us to monitor activity based on what the router sees instead of what users see. Results from Mantra are useful for monitoring several aspects of multicast including multicast usage and deployment as well as protocol deployment and performance. Mantra results are also useful for debugging routing problems.

We have been monitoring two locations using Mantra: (1) a major multicast exchange point, Federal IntereXchange–West (FIXW) and (2) a UCSB campus router. The results presented in this paper are based on six-month worth of data collected from these two collection points between October of 1998 and March of 1999. With the help of these results and their preliminary analysis, we describe the types of results that Mantra generates and how can they be used for monitoring and debugging purposes.

The rest of this paper is organized as follows. In Section 2, we discuss challenges in monitoring multicast, review different types of multicast monitoring and list some of the existing multicast monitoring tools. In Section 3, we present an overview of the Mantra design. Section 4 describes some useful results from Mantra and the pertinent analysis. The paper is concluded in Section 5.

## II. Monitoring Multicast

The rapid pace of multicast deployment has increased the importance of multicast monitoring. There is a strong need to monitor multicast usage as well as different aspects of multicast networks like deployment, protocol performance and routing problems. However, due to the shift in the multicast infrastructure from a DVMRP-based tunnel topology to native deployment, monitoring multicast has become more difficult. Conventional monitoring techniques have lost some effectiveness. Consequently, in the current infrastructure, not only has the utility of the existing monitoring mechanisms diminished, but developing new ones has become a challenging task as well.

In the remainder of this section we identify the challenges in monitoring multicast and discuss our goals for monitoring multicast. We also classify multicast monitoring systems into two different categories; *application layer monitoring* and *network layer monitoring*; and describe several monitoring tools.

### A. Challenges in Monitoring

Challenges in monitoring multicast can primarily be attributed to the following factors:

- *Data delivery via distribution trees.* Multicast data travels in the network along a *distribution tree.* The sender is the root of the tree and each receiver is a leaf. Monitoring data delivery in a multicast session becomes hard as it involves monitoring the distribution trees for each of the senders and all the branches of each tree.
- *Lack of information about receivers.* Any host that is sending data to a multicast group only knows about the nodes that form the next level of the tree. As a result, a sender will likely not know about who or how many receivers there are. Although, it is possible to get this information at the application level, it requires all members of a multicast session to adhere to the same application layer protocol.
- *Limited SNMP support.* Simple Network Management Protocol (SNMP)[9] has evolved into a standard mechanism for gathering monitoring information from various types of devices in the network. However, there is lack of updated standards and Management Information Bases (MIBs) for the newer multicast protocols. In cases of protocols like MSDP, proper MIBs do not even exist. As a result, SNMP has been only marginally effective for multicast.

The factors mentioned above also attribute to the deviation of multicast monitoring from that of unicast monitoring. In contrast to the multicast service model, in unicast networks, data distribution takes place via a single path from a sender to a receiver and each sender knows about the hosts receiving data from it. Consequently, it is hard to directly use unicast monitoring techniques for multicast.

### B. Goals of Monitoring

Effective monitoring is essential for gathering information about and solving problems in the multicast infrastructure. Moreover, it is very important to monitor all aspects of multicast including, deployment, usage, performance and problems. The different characteristics of multicast that need to be monitored can be grouped into two basic categories: *usage characteristics* and *routing characteristics.* Following is a description of monitoring these two types of characteristics.

**Usage Monitoring.** Usage monitoring can be further classified into the following two categories:

- *Session Monitoring.* Session monitoring aims to monitor the session characteristics like: availability of sessions, reachability of sessions, session densities and membership join patterns. Analysis of results from session monitoring play a key role in estimating availability of multicast content. In addition, by analyzing the global reachability of sessions it is also possible to monitor the robustness of data delivery over multicast.
- *Participant Monitoring.* Participant monitoring involves monitoring characteristics of senders as well as receivers. Results from participant monitoring can be used to analyze the characteristics of data flows from senders and discovering topological and geographical characteristics of receivers.

**Route Monitoring.** The main aim behind route monitoring is to provide a snapshot of performance and usage of multicast routing protocols. Route monitoring involves processing routing state information available from the routers and examining characteristics of the actual network paths that the packets take. Processing routing state results in statistics like the number of reachable multicast networks, frequency of changes in the routing table, route lifetimes and individual route stability characteristics. In addition, paths in the network can be monitored to obtain results about losses in intermediate links, location of bottleneck links and traffic flow. Results from route monitoring provide key information needed for analyzing and troubleshooting multicast routing problems.

### C. Characteristics of Existing Tools

Characteristics of a monitoring tool depend on where it is collecting data from, application-layer or network-layer. The type of data that a monitoring tool uses largely decides the kind of results that can be produced and the quality of these results. While application layer data relies on information that is collected from end-user applications, network layer data relies on information collected from network layer protocols like DVMRP, MBGP and PIM. Next, we describe the two types of monitoring data, discuss their merits and limitations and present some of the existing multicast monitoring tools that use them.

**Application Layer Data.** Monitoring data can be collected at the application layer by observing data and control traffic generated by multicast applications. Results from application layer monitoring give a snapshot of the status of multicast from the point of view of the end-user. The results available at the application layer are more suitable for analyzing the characteristics of sessions, participant hosts and data streams. Currently there are two main application layer protocols that are being used to gather monitoring information: the Realtime Transport Protocol (RTP)[8] and the Session Announcement Protocol (SAP)[10]. Use of these protocols for monitoring purposes is further described below:

- *Using RTP data for monitoring.* RTP and its control protocol, the Realtime Transport Control Protocol (RTCP), are used for the realtime delivery of data streams like audio and video. Packets from RTP and RTCP contain usage and quality information as observed by each participant. This information is sent to all group members. These statistics form the basis for monitoring results. Some of the tools that fall in this category include *mlisten*[11] for monitoring multicast group membership and *rtpmon*[12] for collecting data about realtime loss and jitter statistics.
- *Using SAP data for monitoring.* SAP is a common mechanism for advertising current and future multicast sessions across the multicast infrastructure. SAP announcements can be used as a basis for measuring reachability between sources and receivers. Sdr-monitor[13] is a tool that uses SAP packets corresponding to these announcements to monitor reachability.

One of the biggest drawbacks of application layer monitoring is that the successful reception of control and data packets relies on the end-to-end operation of multicast. When multicast is not operating correctly, there is no feedback. Another problem is that application layer data gives little or no information about what is happening at the network layer. When the goal of a monitoring tool is to understand multicast protocol performance, application layer data tends to be of little value. Some concerns also exist regarding the quality of RTCP data itself. One of the most serious concerns is that not all the multicast applications adhere to the RTCP standard and, hence, RTCP data is not transmitted by all of the session participants. In addition, the RTCP scalability mechanism increases the interval between packet transmissions as the number of participants increase. This, in turn, decreases the temporal granularity and accuracy of results.

**Network Layer Data.** State information maintained by multicast routers and the protocol messages that they generate form the core of network layer monitoring data. Results from network layer monitoring depict the network state from the perspective of the collection point(s). The information available at the network layer can be used for usage monitoring as well as route monitoring. Data available at the network layer can be classified into two categories, *routing tables* and *forwarding tables*. Routing tables are usually byproducts of the operation of routing protocols like DVMRP and MBGP. They provide information about the routes in the multicast networks. On the other hand, forwarding tables provide information for all the participant-group pairs for which the router has state. These tables are maintained by routers as a result of routing protocols like PIM and DVMRP. In addition to routing, these protocols are responsible for creation and management of data distribution trees. The process of data collection at the network layer involves recording state at various points in the multicast network. Following are three mechanisms that existing monitoring tools employ for collecting data at the network layer:

- *Using Special Implementation in the Routers.* Some monitoring tools need special mechanisms to be deployed in routers. *Mtrace*[14] and *mrinfo*[5] are two such tools. *Mtrace* uses forwarding state from intermediate routers to provide a path trace tool and hop-by-hop packet flow statistics. *Mrinfo* collects routing information about a router's multicast capable interfaces. Additionally, there are a set of tools that are built on top of *mtrace* and *mrinfo*. For example, tools like *mhealth*[15] and *mantaray*[16] provide useful front-ends for *mtrace*. Similarly, *mwatch* recursively calls *mrinfo* and aims to find all the multicast routers across all the multicast networks.
- *Snooping of data packets.* This involves capturing data packets on individual links in the network. Route Monitor is a tool that snoops DVMRP route updates issued by a local multicast router[17]. This data is used to collect data for monitoring the stability of DVMRP. Another tool, MultiMON[18] collects RTCP data at the location of *multimon server*. Results obtained from the collected RTCP statistics are then available via a remote display. This tool is used to monitor multicast traffic on the local network segments.
- *Reading router tables.* This involves capturing internal state information maintained by multicast routers. Here, data collection can be accomplished either by logging onto routers and capturing the tables directly or by using mechanisms like SNMP. Data is captured from state information cached by the routers, and it is presented to the user. Depending on monitoring requirements, only selective information can be captured. However, accuracy of the information and its temporal validity is limited to that of the accuracy and refresh rate of the routers cache respectively. Merit Networks has developed a suite of tools that rely on state information maintained by routers for gathering monitoring data[19]. This suite includes tools like *mstat*, *mrtree*, and *mview*. These tools use SNMP to collect all manner of data from routers. *Mstat* queries SNMP-enabled router for information about various routing tables and packet statistics. *Mrtree* performs

cascaded SNMP queries on routers to discover a particular multicast session's distribution tree. Finally, *mview* is a GUI front-end to *mstat* and *mrtree* as well as to other diagnostic utilities such as *mtrace*.

Because routing information and details of the network topology are available only at the network layer, network layer data is more suitable for debugging and isolating network problems. In addition, forwarding state is maintained in the network for all sessions and participants. This information can be used to monitor participants even if they are not using RTCP. Furthermore, state will be maintained for sessions even if they are not advertised via SAP. However, there are several limitations of network layer monitoring. The most prominent of these is the difficulty of collecting network layer data. Collecting state information directly from the routers usually requires special privileges. A password is needed on a router if the data is to be collected by logging into it, or a valid community string is required for accessing router state via SNMP. Another factor that limits the use of SNMP for data collection is the lack of updated standards for the newer multicast protocols. This makes SNMP unsuitable for monitoring newer routing protocols. This is one of the main reasons that we do not use SNMP for data collection in Mantra.

## III. Design of Mantra

The design of Mantra is based on five major modules each of which is responsible for performing a major task in Mantra's monitoring cycle. Figure 1 displays these modules and the flow of data in Mantra. In the remainder of the section, we discuss the design and development of Mantra by describing each of these modules.

**Data Collector.** With the help of this module, Mantra collects data from multicast routers and pre-processes the collected data. Acquiring data from a router involves, logging on to the router, reading its internal memory tables and transferring these tables to the local system. Mantra achieves this with the help of a set of *expect* scripts, which it launches at frequent intervals to collect the latest monitoring data from the router. Preprocessing, on the other hand, involves removing unwanted information, excess white-spaces and delimiters from the collected table.

**Router-Table Processor.** In this step, collected data is converted to Mantra's local data format. We have designed a set of tables that provide a standard framework for storing the monitoring information we collect. Thus, the main task of the router-table processor is to map each of the pre-processed data sets to the corresponding local table(s). There are four kinds of tables that define the internal data format for Mantra:

- *Pair Table*: Entries for source-group pairs, called (S,G)s, lists all the session-participant tuples. Other information available in these tables include the current and the average bandwidth used by each entry.
- *Participant Table*: Information about hosts participating in a session including the host name (if available), number of groups it is participating in and the time period for which Mantra has had state for it.
- *Session Table*: Information about the multicast sessions including the group's name (if available), its density, packets received, packets sent, and the protocol that first advertised it.
- *Route Table*: Information about the current set of live routes including, next hop, uptime for the route and the metric associated with it.

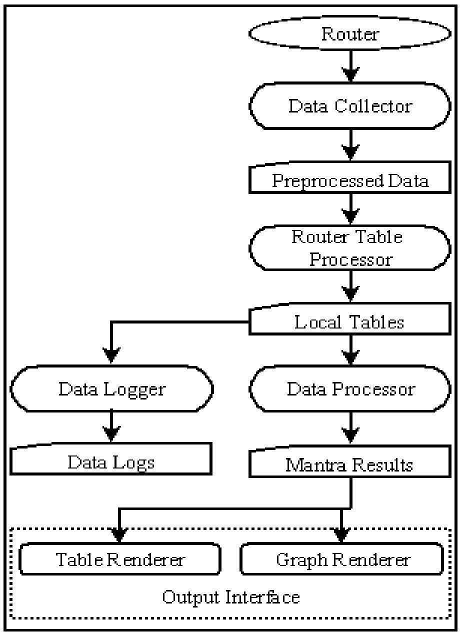


Fig. 1. Mantra modules and information flow.

**Data Logger.** The data logger stores processed data for detailed off-line analysis and long-term trend analysis. While our main goal is to be able to store as many data elements as possible, issues related to conservation of storage space are also a major concern. Therefore, we have developed some techniques to conserve space without losing any information. These techniques include:

- *Storing Only Deltas.* Instead of storing the entire data table, we only store the changes that have occurred since the last measurement. For tables containing infrequently changing data (for example, route table data), this is a very effective way of conserving storage space.
- *Avoiding Redundancy.* Some of Mantra's tables can be constructed with the help of information from other tables. For example, the pair table can be constructed with the help of the source table and the group table. In such cases Mantra only stores the constituent tables.

**Data Processor.** The data processor is a collection of modules that take as input tables represented in Mantra's local data format, processes these tables and generates final results. The results that are generated show the most recent snapshot of multicast state. Results from the data processor can be classified into two categories.

The first contains raw statistical results represented as x-y coordinate data, which is used to plot charts. The other category comprises multi-column text tables containing summary results. The information contained in summary tables ranges from description of the busiest multicast sessions to the raw count of networks available via DVMRP.

**Output Interface.** Results from Mantra are available via the web using interactive Java applets. These applets provide two kinds of output interfaces: interactive tables and interactive graphs. Each of these interfaces is describe in further detail below.

- *Interactive Tables.* This is the interface for presenting summary-tables, which Mantra generates at the end of each monitoring cycle. The top snapshot in Figure 2 shows this interface. This interface provides functions that perform several useful operations on the tables. Some of the functionality available thorough this interface include searching and sorting; algebraic manipulation of numeric columns; and several date and time conversion operations.
- *Interface for Displaying Graphs.* Mantra uses this interface to display results in the form of two-dimensional line graphs. The bottom snapshot in Figure 2 displays this interface. This interface provides functions to interactively tailor the graph display. Two of the most important of these features include: (1) ability to overlay multiple graphs and (2) the ability to manipulate graph axes and scale. The former allows users to plot multiple graphs on the same display. This function is useful for analyzing the relationship among different variables. The scaling feature allows users to zoom in and out of the graph by specifying new X-axis and Y-axis ranges either by entering new values in the text-box or by selecting the area using click-and-drag mouse operation.

## IV. Results from Mantra

The results from Mantra can be used for analyzing multicast usage as well as routing state. We have been using Mantra to collect data from different points in the network for over two years now. Thus, the data archives from Mantra are also suitable for long term analysis and for gaining useful insight into the developments in multicast, especially those associated with the shift of infrastructure from a DVMRP tunnel-based topology to native multicast. In the remainder of the section we describe the characteristics of the data we have been collecting. In addition, we present some of the results and relevant analysis.

### A. Characteristics of Data

Mantra primarily collects two types of data sets from multicast routers, DVMRP route tables and multicast forwarding tables. Mantra uses route tables to monitor the deployment of DVMRP and reachability of DVMRP-networks. In addition, Mantra uses information available in the forwarding tables to obtain results pertaining to session and host characteristics, session-membership patterns and bandwidth used by the multicast traffic through the router.

Results that we present in this paper are based on data sets collected from two routers. The first is a UCSB router running mrouted and the other is a major multicast exchange point, FIXW. Data from the UCSB router gives a snapshot of the state of multicast from the point of view of the campus network. In contrast, data from FIXW gives a snapshot that is more representative of the multicast infrastructure. This is the case because FIXW is an exchange point and the information available at it is an aggregation of traffic from different networks.

An added advantage of FIXW is the utility of it's data in depicting the transition of the multicast infrastructure towards native multicast. In the early stages of multicast deployment, DVMRP tunnels constituted most of the multicast infrastructure. In this infrastructure, FIXW was a central peering location and it acted as a core MBone router. However, as deployment of native multicast increased, the role of FIXW transitioned into being a border router for the DVMRP networks. It currently acts as an interface between DVMRP-based multicast networks and the domains with native multicast. Therefore, the history of FIXW represents a major change in the infrastructure. Mantra data for this period is very useful for analyzing and tracking changes in infrastructure. Even though we have been collecting data using Mantra since November of 1998, the results that we present in this paper are based on the data collected in the first 6 months. The main reason behind this is to show the changes in the level of activity at FIXW that occurred during the infrastructure transition period.

### B. Usage Monitoring by Mantra

In this section, we present results specific to session monitoring and participant monitoring–the two subcategories of usage monitoring. We also analyze the effects of the transition on these results. The results are based on the data collected from FIXW between November of 1998 and April of 1999. This is followed by a general analysis of results from session monitoring and participant monitoring.

**Classification of Sessions and Participants.** One of the important aspects of usage monitoring is to identify those sessions and participants that have consistent content/data associated with them. To this end we use bandwidth usage as a metric to classify sessions and participants. We classify participants into two categories *senders* and *passive-participants*. Similarly sessions are classified as either *active-sessions* or *inactive-sessions*. Participants are classified on the basis of the kind of data they are sending to their corresponding sessions, all the participants that are sending content-data are termed as senders and others fall in the category of passive-

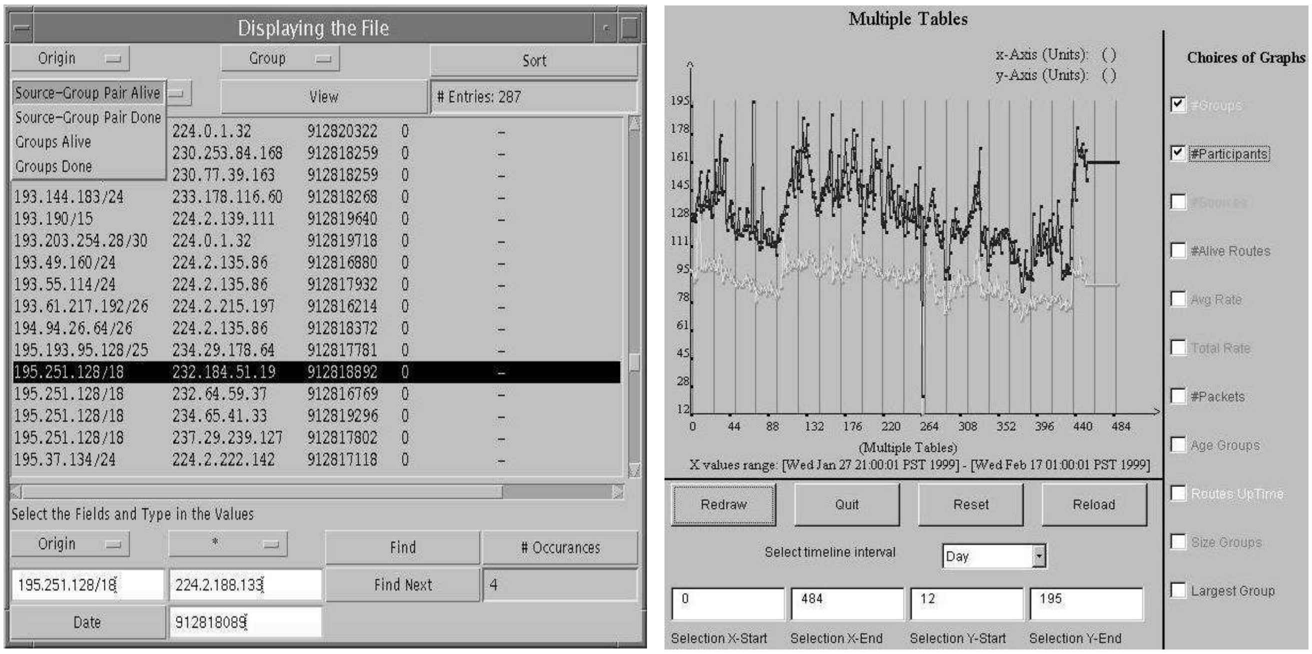


Fig. 2. Mantra interface for accessing data using a table format (left) and an interface graph (right).

participants. On the other hand, classification of a session depends on the constitution of its participant population. While active-sessions are the sessions with at least one participant that is a sender, all the participants of inactive-sessions are passive-participants. As there is no straightforward way to categorize participants, Mantra uses bandwidth used by a participant as a metric to distinguish senders from passive-participants. Any participant host that is sending data at a rate greater than the threshold of 4kbps is identified as a sender. Our decision to choose a threshold of 4 kbps is primarily based on two reasons: (1) bandwidth used by control traffic rarely exceeds this rate; and (2) content data usually uses bandwidth at a rate greater than 4 kbps. Most of the participant hosts send control traffic to a session (example: RTCP traffic) as feedback to the data they are receiving.

**Characteristics from Sessions and Participants.** Graphs in Figure 3 show session and participant monitoring results from Mantra. While the top two plots show the change in the number of sessions and participants over time, the bottom ones plot change in the number of active-sessions and senders over time. The following observations can be made: (1) number of sessions is low; i.e. the number of sessions and active sessions available at any time is very small; (2) participation is scanty; i.e. the number of participants and active participants is very low; and (3)variations are high: i.e. the availability of sessions and the number of participants vary significantly. Some inferences that can be made about the usage of multicast on the basis of these observations include the following:

- *Experimental Usage.* The high frequency of variations implies that there is a large number of sessions that are short lived. One of the main factors that can be attributed to this behavior is experimental usage. This behavior was still very common in early 1999. An often observed scenario consists of a single host initiating several sessions simultaneously. This conclusion is based on results from a detailed off-line analysis which shows that at any point of time when the number of sessions is more than 500, more than 85% of sessions have only a single member.
- *Sessions Without Data.* A wide gap between the graphs for number of sessions and that of active-sessions shows that most of the sessions present at any time have either had no data streams associated with them or the data streams were using very little bandwidth. Several reasons can be attributed to such an observation: the presence of experimental sessions without contents, routing problems causing loss of data within the network, and/or the lack of data flow through the router because of absence of downstream participants.
- *Diverse Session Densities.* Figure 3 plots the change in average density of sessions over time. A detailed off line analysis shows that the densities vary significantly across sessions. At any given instance, on average, more than 65% of sessions do not have more than two participants. In contrast, in several data sets, total participants in less than 6% of sessions account for about 80% of participants. Another interesting point is the correlation of density results with those for the number of sessions and participants. Comparison of results presented in Figure 4 with those presented in Figure 3 show that, each spike in number of sessions corresponds to a dip in average density. In contrast, each spike in number of participants causes the average density to increase

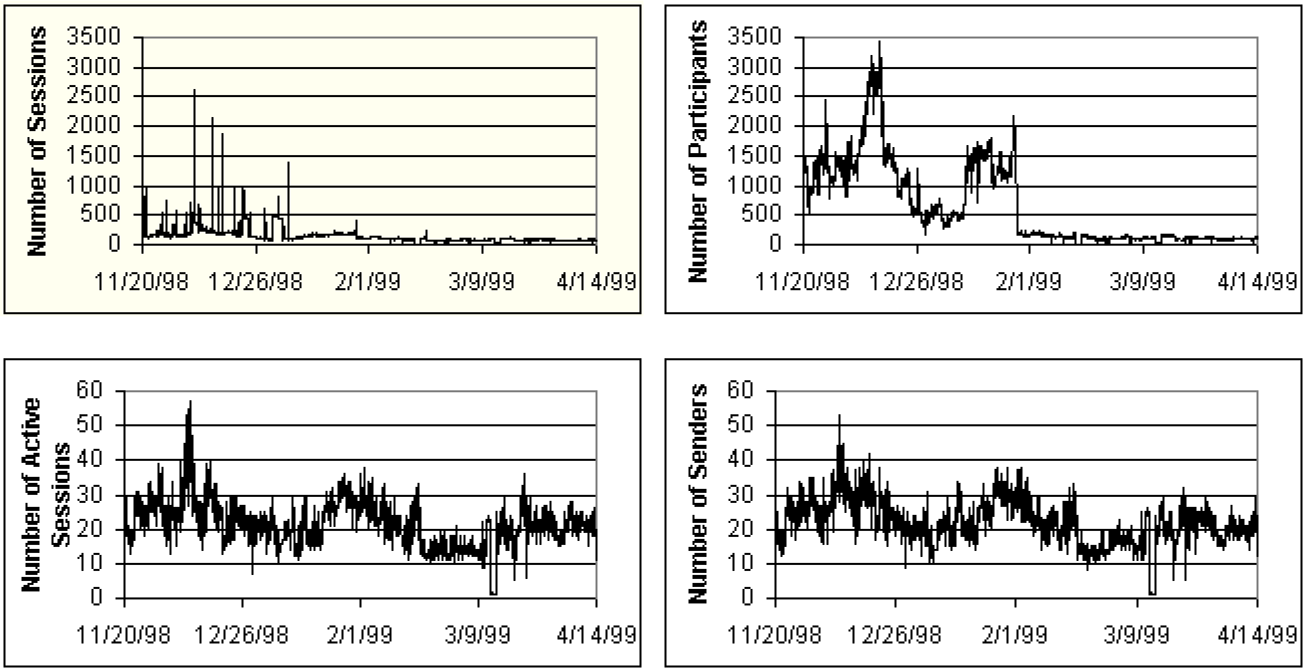


Fig. 3. **Session and Participant Statistics (Total Counts):** Sessions(top-left); Participants(top-right); Active Sessions(bottom-left); and Senders (bottom-right)

significantly. There are two reasons that might be causing this phenomenon. First, spikes in the number of sessions represent short-lived sessions and before any participant can join them they cease to exist. Second reason is that when there is a surge of new participants, participants tend to join already present groups. This might very well be the case, as some sessions are more popular and well advertised than others. Delivery of IETF meetings is a good example. In fact, the peak in early December seen in Figure 4 corresponds to the 43rd IETF meeting in Orlando, Florida and falls in line with our conclusions.

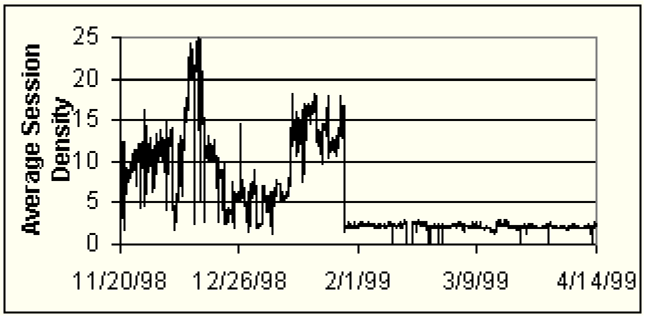


Fig. 4. Session Densities.

- *Low Bandwidth Usage.* The left plot in the Figure 5 shows the amount of bandwidth for all multicast traffic that passed through FIXW. In general the bandwidth used by multicast traffic is very small. Except for some peak periods, the average bandwidth requirements remain around 4 Mbps. However, a standard deviation of about 2.2 Mbps over a median 2.9 Mbps indicate that variations in this rate are very high. This can, in turn, be attributed to the presence of short-lived high bandwidth data streams. The bandwidth usage results also depict the amount of bandwidth saved by the use of multicast. The Right plot in Figure 5 shows the amount of bandwidth saved. In order to obtain results shown in this graph, we find out the density of an active session and bandwidth being used by its senders. Assuming that every unicast path from senders to all the receivers would pass through the router, then the density multiplied by the rate of the stream gives an estimate of the bandwidth that would have been used if unicast were used for content distribution.

**Effect of Transition.** Results from both session monitoring as well as participant monitoring show the prominent effects of the multicast topology transition. After the transition, while the total number of participants dropped considerably, availability of sessions at the router stabilized. However, the number active sessions and number of senders remained almost the same. One of the reasons for this can be attributed to the increased use of sparse mode multicast routing. Sparse mode protocols enabled FIXW to start filtering out entries corresponding to sessions for which there were no downstream participants. In addition, this eliminated all the sessions with single participants who were not downstream of FIXW. The contrast between changes in the number total participants and the number of senders is evident with the help of the right plot in Figure 6. This plot shows the ratio of the number of senders and total participants. This

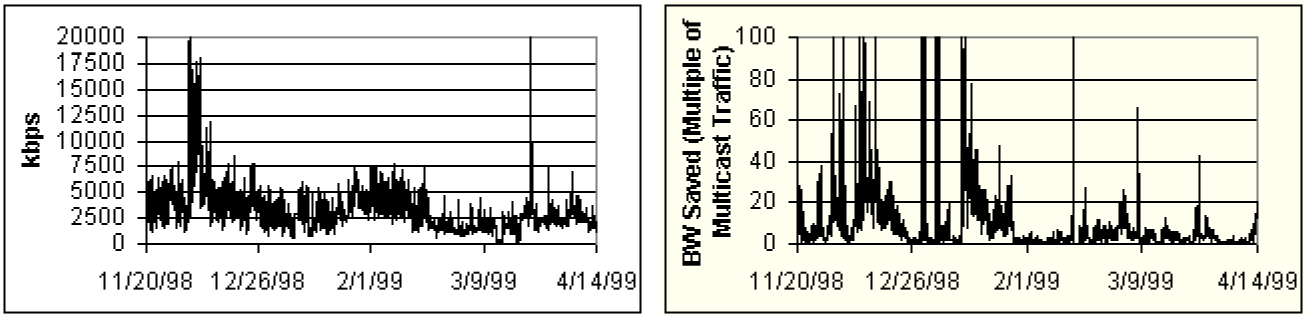


Fig. 5. **Bandwidth Usage:** Multicast Traffic from all the Senders, in kbps(left); Amount of Bandwidth Saved, Represented as Multiple of that Used by Multicast Traffic(right).

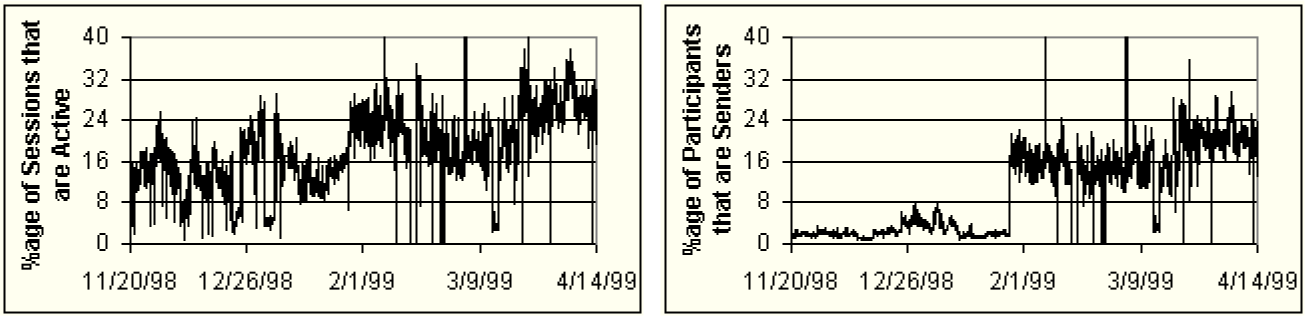


Fig. 6. **Percentage Active:** % Sessions that are Active(left); % Participants that are Senders(right)

ratio clearly starts to increase after the transition. The plot on the left in Figure 6 plots the ratio between the number of active sessions and total sessions. The trend in this plot confirms that while the ratio increases only marginally after the transition, variations decrease considerably. This confirms our conclusion that after the transition availability of sessions at FIXW had stabilized considerably.

### *C. Route Monitoring by Mantra*

Route monitoring in Mantra consists of monitoring the characteristics of DVMRP route tables. Plots in Figure 7 show the change in the number of routes as seen at the UCSB router and at FIXW. Following are the three important inferences that can be derived from these results:

- *Unstable Routes.* The number of routes as seen at either of the routers, varies significantly over time. As each route in the DVMRP table represents network that are reachable via multicast, variation in the number of routes implies that the reachability of these networks varies significantly and multicast routing is unstable.
- *Inconsistent State.* Ideally every DVMRP router should have similar DVMRP tables and should be able to reach the same DVMRP networks. However, comparing DVMRP route tables from UCSB router and FIXW shows that this is not the case and the routing state that they have are inconsistent. Some of the factors that could cause this inconsistency include, loss of route updates messages, inconsistent route aggregation and problems with route exchanges due to incompatibility among routers.
- *Effect of Transition.* The effects of the transition in multicast infrastructure are obvious in the results from route monitoring. Unlike the transition of multicast towards sparse-mode protocols, transition towards removing existing DVMRP did not gain momentum until the end of 1999. Figure 8 plots the number DVMRP networks as seen at FIXW in the last two years. The results show that use of DVMRP has declined and seems to be almost nonexistent today.

Route monitoring results from Mantra can also be very helpful in debugging routing problems. While a more detailed analysis can aid in debugging, it is also possible to easily detect the routing problems. Figure 9 shows one such problem. This figure plots a snapshot of the number of DVMRP routes seen at the UCSB router on October 14th, 1998. As seen in the graph, there was a sharp increase in the number of routes at around 1400 hours. A detailed off-line analysis shows that this was caused because of unicast route injection into the DVMRP route tables[20].

## V. Conclusions

In this paper we have argued that there is a lack of monitoring tools for monitoring multicast at the network layer. This argument depends on first identifying multicast monitoring requirements in the current infrastructure and recognizing that different types of data sources

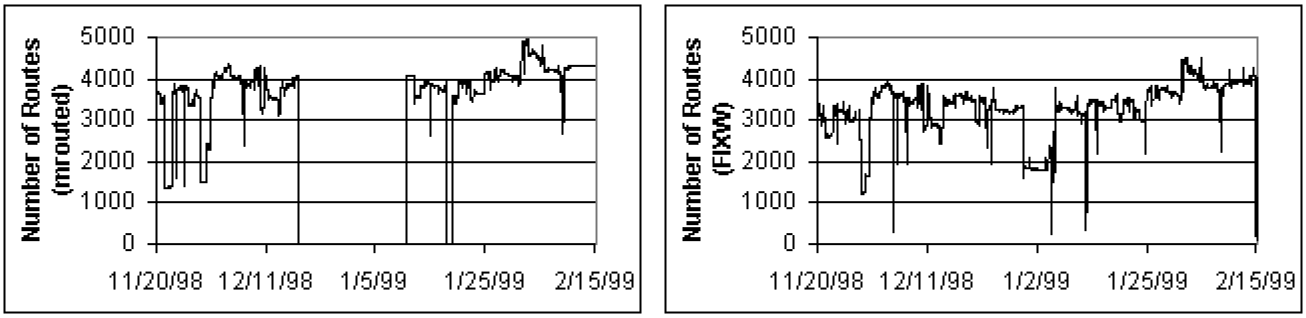


Fig. 7. **DVMRP-Routes Statistics:** Number of Routes, at UCSB Router–mrouted(top) and at FIXW (bottom).

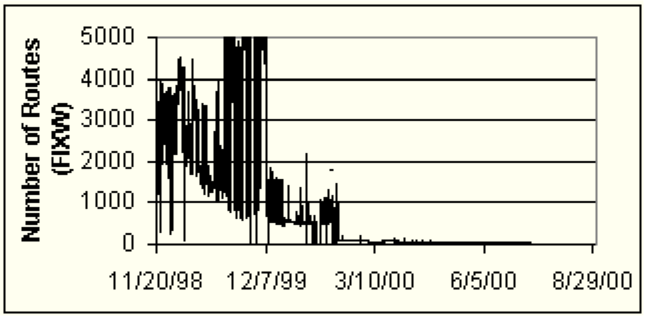


Fig. 8. DVMRP at FIXW: Long Term Results.

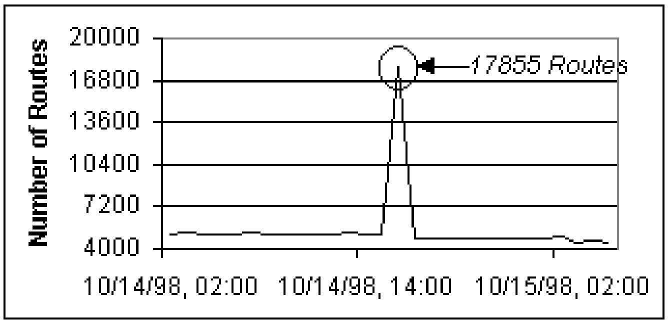


Fig. 9. Unicast route injection into mrouted routes-table.

exist. We have focused on using the two categories: *network layer* and *application layer* statistics. We believe that significant work exists in investigating and then developing new types of monitoring tools. To this end we have designed and prototyped Mantra, a tool to collect state from multicast routers. Mantra is portable and provides the functions of data collection, processing, basic analysis, data logging, and real-time online presentation.

Mantra is currently being used to monitor FIXW, a major multicast exchange point. FIXW acts as a core router for the MBone. Results based on its data give a fare estimate of the global state, but only when the multicast infrastructure was primarily comprised of DVMRP-tunnels. Results from Mantra show that the number of multicast sessions and participant hosts has been very low. The small number of sessions that have participants sourcing actual data traffic shows that the usage of multicast is far from what is expected from a production-level service. Experimental usage of multicast is still very common and a high frequency of variations in the availability of sessions indicate that multicast routing is unstable. Results pertaining to DVMRP monitoring based on data from FIXW affirm the second conclusion because route availability has been observed to change frequently. Through comparison with DVMRP monitoring results collected from a UCSB router, we have seen that route availability varies between domains.

A final observation is that there has been a significant drop in the numbers of inactive sessions and passive participants. This is a strong evidence of the transition of the multicast infrastructure towards native sparse-mode. However, these results also indicate that in the sparse mode scenario, irrespective of a routers topological location, it is becoming difficult to track global usage statistics. It has become extremely important to generate global results by collecting data at multiple points. Results should then be based on aggregate views. Considering the importance of this issue, works is in progress to enhance Mantra such that it can not only collect data from multiple routers concurrently, but also aggregate different data sets and generate combined results in real-time.

## 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*, January/February 2000.

[4] K. Sarac and K. Almeroth, “Supporting multicast deployment efforts: A survey of tools for multicast monitoring,” *Journal of High Speed Networking–Special Issue on Management of Multimedia Networking*, March 2001.

[5] 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/.

[6] A. Ghosh and P. Brooks, *MWATCH 3.6.2.* University College London, June 1994. Available from http://www.cl.cam.ac.uk/mbone/index.html#Mrouted.

[7] D. Thaler and A. Adams, *Mrtree.* Merit Network, Inc. and University of Michigan. http://www.merit.edu/net-research/mbone/mrtree_man.html.

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

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

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

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

[12] A. Swan and D. Bacher, *rtpmon 1.0a7.* University of California at Berkeley, January 1997. Available from ftp://mm-ftp.cs.berkeley.edu/pub/rtpmon/.

[13] K. Sarac and K. Almeroth, "Monitoring reachability in the global multicast infrastructure," in *International Conference on Network Protocols (ICNP)*, (Osaka, JAPAN), November 2000.

[14] W. Fenner and S. Casner, "A 'traceroute' facility for IP multicast." Internet Engineering Task Force (IETF), draft-ietf-idmr-traceroute-ipm-*.txt, August 1998.

[15] D. Makofske and K. Almeroth, "Real-time multicast tree visualization and monitoring," *Software–Practice & Experience*, vol. 30, pp. 1047–1065, July 2000.

[16] B. Huffaker, K. Claffy, and E. Nemeth, "Tools to visualize the internet multicast backbone," in *Proceedings of INET '99*, (San Jose, California, USA), June 1999.

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

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

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

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