A Scalable Architecture for Monitoring and Visualizing Multicast Statistics

Prashant Rajvaidya and *Kevin C. Almeroth*
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5110
{prash,almeroth}@cs.ucsb.edu

k Claffy
CAIDA
University of California, San Diego
San Diego, CA 92093
kc@caida.org

April 03, 2000

Abstract

An understanding of certain network functions is critical for successful network management. Managers must have insight into network topology, protocol performance and fault detection/isolation. The ability to obtain such insight is even more critical when trying to support evolving technologies. Multicast is one example of a new network layer technology and is the focus of this paper. For multicast, the pace of change is rapid, modifications to routing mechanisms are frequent, and faults are common. In this paper we describe a tool, called Mantra, we have developed to monitor multicast. Mantra collects, analyzes, and visualizes network-layer (routing and topology) data about the global multicast infrastructure. The two most important functions of Mantra are: (1) monitoring multicast networks on a global scale; and (2) presenting results in the form of intuitive visualizations. To achieve accurate monitoring, Mantra collects data from several topologically and geographically diverse networks. For the purpose of presentation, Mantra uses several interactive visualization mechanisms to present statistics, topology maps and geographic properties. Another noteworthy feature of Mantra is its flexible and scalable architecture. This architecture helps keep our monitoring efforts current with the fast pace of multicast developments. It also enables us to expand Mantra's monitoring scope to more networks and larger data sets.

1 Introduction

Several useful network monitoring mechanisms have evolved over the years to support operational debugging and troubleshooting. The Internet Control Message Protocol (ICMP) and the Simple Network Management Protocol (SNMP)[1] are the original control and management protocols of the TCP/IP protocol suite. They form the basis for many monitoring tools. Despite these developments, monitoring the global Internet is still a formidable task. Its ever-increasing size and heterogeneity show the scalability weaknesses of existing management solutions. Generically, we believe that the basic challenges in global monitoring include: collection of data from a variety of networks; aggregation of these heterogeneous data sets; useful data mining of these data sets; and presentation of results. These challenges are applicable to almost any global monitoring system, and are applicable for almost any kind of data collected.

Recent network technologies such as multicast and Quality-of-Service (QoS) impose new requirements for network monitoring. Current deployment of these technologies in the infrastructure is far less than that of traditional unicast delivery. This can theoretically make monitoring easier–at least there is less data. We can also leverage experience and lessons learned with traditional monitoring tools in designing systems to help monitor newer technologies. Nevertheless, the rapid pace of development, the lack of standards, and the lack of widespread understanding of these new technologies pose challenges for developing systems to monitor next-generation networks.

In this paper, we focus on multicast monitoring. Multicast is a particularly rapidly evolving new network technology. Multicast provides a scalable and bandwidth-conserving solution for one-to-many and many-to-many delivery of packets in the Internet. Delivery of high-bandwidth streaming media via multicast not only improves the scalability of the streaming server (i.e., allows it to serve more clients) but also reduces the number of redundant data streams. Multicast was first widely deployed in 1992. Since then, the multicast infrastructure has transitioned from an experimental tunnel-based (virtual overlay) architecture based on the Distance Vector Multicast Routing Protocol (DVMRP)[2] to pervasive deployment of native multicast. In the current infrastructure, stable Internet multicast relies on a complex system of protocols operating in harmony: legacy DVMRP and Protocol Independent Multicast Dense Mode (PIM-DM)[3] for intra-domain multicast routing; PIM Sparse Mode (PIM-SM) for both intra- and inter-domain multicast routing; the Multicast Border Gateway Protocol (MBGP)[4] for policy-based route exchange; and the Multicast Source Discovery Protocol (MSDP)[5] for exchanging information about active sources. Increased commercial interest and associated growth in multicast deployment makes monitoring both more important and more difficult. Systems that can gauge the performance of various multicast protocols, delineate various aspects of current multicast infrastructure, and predict future trends in workload are of tremendous value.

Our goal is to design and develop a system to monitor multicast on a global scale by collecting data at the network layer. We aim to use monitoring results to provide intuitive views of the multicast infrastructure. In this paper, we present Mantra, a tool that we have developed for this purpose. Mantra collects network-layer data by capturing internal tables from several multicast routers. Data is processed to depict global and localized views of the multicast infrastructure. Presentation mechanisms include topological and geographic network visualizations and interactive graphs of various statistics. Results from Mantra are useful for several purposes including assessing the amount of network activity, evaluating routing stability, and detecting and diagnosing problems. Another important feature of Mantra is its scalable and flexible architecture. Mantra provides mechanisms to easily support growth in the network as well as support for new data collection activities.

The rest of this paper is organized as follows. We review related work in Section 2. In Section 3, we describe the goals and challenges of our work. Section 4 describes the design and architecture of Mantra. Section 5 provides an example of Mantra being used to identify network problems. The paper is concluded in Section 6.

2 Related Work

Monitoring the current Internet infrastructure on a global scale is challenging because it consists of a complex topology of numerous heterogeneous networks. Moreover, there is little interest for commercial Internet Service Providers (ISPs) to provide monitoring data to external organizations. Nevertheless, there is an array of useful work for monitoring the Internet beyond a single administrative domain. The earliest such tools include *traceroute* and *ping*. There are also several ongoing efforts in the field of end-to-end Internet monitoring, most involving active probe traffic sent from a source to one or several hosts and subsequent evaluation of response time, throughput, or path changes. Paxson's work[6] is one example. However, most end-to-end monitoring tools and related analysis efforts lack intuitive visualization of results. As a consequence, proper interpretation requires an in-depth knowledge of the infrastructure and protocol operation. Most existing visualization efforts are limited to front-ends to existing tools, e.g. GTrace[7] for *traceroute*. These tools tend to be less than sufficient for detailed problem identification, isolation, and resolution. A variety of useful work in the field of Internet monitoring and visualization is taking place at the Cooperative Association for Internet Data Analysis (CAIDA). Projects include: skitter[8] for macroscopic topology discovery and depiction; Mapnet[9] for visualization of Internet backbone topologies; and MantaRay[10] for interactive graphing of MBone topology. MantaRay is a follow-up to Munzer et al.'s 1996 collaboration on 3D geographic visualization of the global MBone tunnel topology[11]. CAIDA has also developed and maintains more general-purpose topology visualization tools such as Otter[12] and GeoPlot[13].

A second challenge of monitoring the Internet beyond the complexities of the topology is the difficulty of monitoring multicast traffic. The difficulty arises primarily because of the differences between the unicast and multicast service models. In unicast networks, data transfer is between only two hosts. In contrast, in multicast networks, data is delivered to logical groups of hosts and data transfer takes place via a dynamic distribution tree. Consequently, monitoring multicast usually involves monitoring either the whole or a part of such distribution trees. In addition, a multicast sender does not typically know about all of a group's receivers. Therefore, even monitoring at the source is not straightforward.

The differences between unicast and multicast also reduce the effectiveness of using existing unicast monitoring mechanisms for multicast. In general, unicast tools provide only limited functionality and do not perform well for multicast-related network management tasks like data collection, data processing, presentation of results and provision for analysis. The solution has been to ignore existing unicast tools and develop new tools specifically for multicast.

There are a number of monitoring tools that have been developed specifically for multicast. One of the most widely used examples is *mtrace*[14]. It is an end-to-end tool that characterizes multicast paths between hosts. *MHealth*[15] provides a useful visualization front-end for *mtrace*, and MantaRay attempted to do the same for mwatch/mrinfo[16, 17] information. However, both *mtrace*, and necessarily *MHealth*, suffer from scalability problems. The primary problem is that *mtrace* provides only a source-to-receiver trace and must be repeated for each group member. Large groups require large numbers of traces. Other tools, such as *mstat*, *mrtree*, and *mview*[18], collect data directly from routers via SNMP[1]. The limitation with SNMP-based tools is that they are typically only useful for intra-domain monitoring. Still another class of monitoring tools, including *mlisten*[19], *rtpmon*[20] and *sdr-monitor*[21], collect data at the application layer. While these tools provide important results, they provide little information about the network, router state, and network protocol operation.

3 Goals and Challenges

Monitoring multicast networks on a global scale requires mechanisms for collecting, analyzing, and presenting results. In this section we describe both our goals and the challenges of meeting these goals. We specifically frame this discussion in the context of Mantra but believe that our experiences are applicable to the construction of other, similar tools.

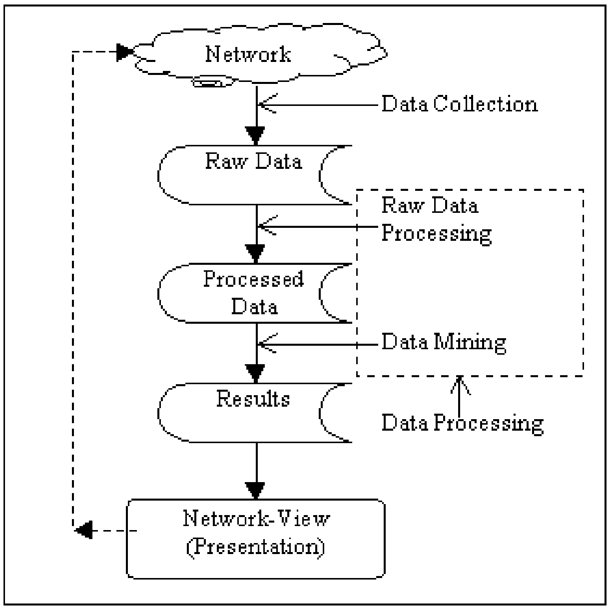

Figure 1: Network Monitoring Model.

3.1 Goals

Design goals pertain to Mantra's architecture for data collection and analysis; presentation goals reflect the need to provide intuitive and useful visualization.

Design Goals. We have attempted to develop an appropriate generic architecture for collecting and processing data from multiple networks. Figure 1 depicts a simple model and the necessary stages. We need a flexible and scalable architecture for performing a wide range of monitoring tasks. As shown in the model, monitoring involves both data collection and data processing. Mantra's data collection occurs at the network layer, acquiring memory tables from multicast routers that are geographically and topologically dispersed throughout the world. Data processing requirements include: removing noise from raw data; converting raw data to Mantra's local data format; aggregating data collected from different networks; and analyzing these data sets to generate useful results. We elaborate on these tasks in later sections.

We also need Mantra's architecture to be flexible enough to accommodate the rapidly evolving multicast infrastructure. Frequent changes are common and may require modifications to the monitoring process. In addition, Mantra needs to be able to adapt to potential variations in the monitored environment, e.g., inconsistent raw data formats, unreliable data sources, and an unstable topology. Finally, Mantra needs to be able to handle a large, and increasing, volume of data; an inevitable consequence of the growing number of networks and protocols, as well as increased use in currently monitored networks.

Presentation Goals. We need to use collected and processed data to generate useful views of various aspects of multicast. We visualize results using several tools: Otter[12], for interactive topology visualizations; GeoPlot[13], for visualization of the geographic placement of various multicast entities; and MultiChart, a tool we have developed for interactive graphing. These mechanisms add to Mantra's usefulness for tasks such as: measuring performance of networks; estimating the extent of multicast deployment; debugging protocol implementations; detecting faults; identifying problems spots; and planning growth in the multicast infrastructure.

3.2 Challenges

As multicast has grown, so have the challenges associated with each step of the monitoring process. Some of the specific challenges include:

Challenges in data collection. Data collection from multiple sites poses a number of problems. The two most important are temporal variations in the allowed frequency of data collection and data format incompatibility. First, data collection is an invasive activity and will always add overhead to the router being polled. In the worst case, this additional overhead might contribute to overload, causing congestion and possibly the failure to handle the current traffic load. Second, different routers may be from different vendors and even routers from the same vendor will likely be running different versions of routing code. Each difference will likely affect the format of the data. Although protocols like SNMP exist to standardize the process of data collection as well as the format of collected data, there is a lack of SNMP support for multicast. Management Information Bases (MIBs) for the newer multicast protocols either do not exist or are not up to date. Consequently, SNMP is not suitable for monitoring newer multicast routing protocols.

Challenges in data processing. Data processing involves parsing raw data into well-structured tables and removing various types of errors from these tables. The first task requires keeping the parsing modules current with changes in raw data formats. The second task, error reduction/elimination, is extremely difficult to automate. Data can be noisy and unrepresentative of the true picture for several reasons, including: effect of test users joining and leaving sessions very quickly; incorrect data due to bugs in protocol implementations; and corrupt data because of problems during collection. Mechanisms to mitigate the effects of errors vary with the cause of the problem. While removing noise due to experimental user behavior involves developing heuristics to identify anomalies in data sets, managing data corruption might involve ignoring the entire data set.

Challenges in data mining. Challenges in data mining involve keeping our analysis techniques current with the rapid pace of multicast technology developments, as well as generating a representative global view of the multicast infrastructure. Problems with generating a global view are two-fold: (1) protocols such as PIM-SM and MBGP do not keep detailed global information, instead, they keep hierarchical information, i.e. they only keep information about reaching a domain and not how to reach hosts within the domain; (2) the lack of sufficient worldwide monitoring locations, data format compatibility, and temporal congruity makes it difficult to develop a consistent global view.

4 Design of Mantra

Mantra's architecture follows the basic model introduced in Section 3. Figure 2 depicts the information flow at different stages–from data collection to data processing, analysis and storage of results. We classify different entities that constitute this model into two broad categories: information (data) formats and module groups. In this section we describe these two categories in further detail.

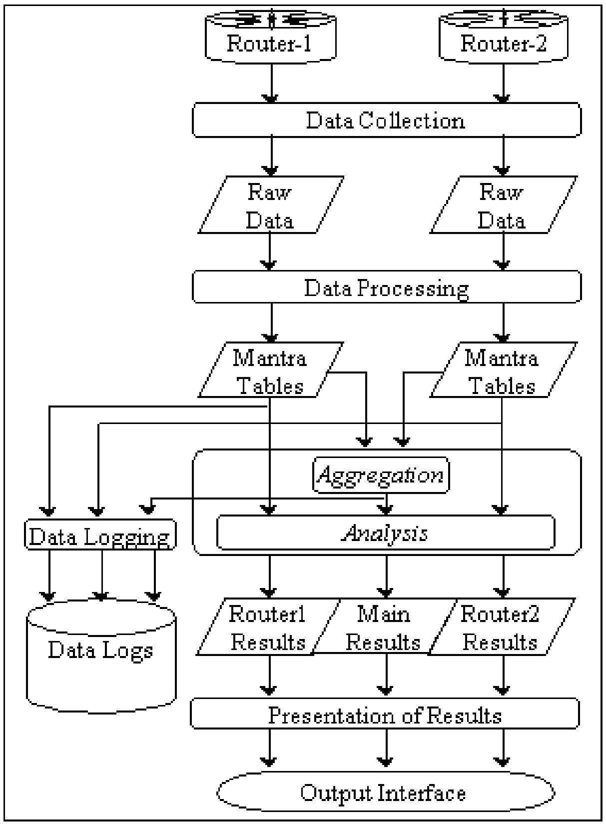

Figure 2: Architecture of Mantra.

4.1 Information Formats

At any stage of a monitoring cycle, data can belong to one of the following three classes: intermediate results, data logs, or monitoring results. Intermediate results refer to the transient information passed on from one module-group to another during different stages of processing. Data logs refer to the final form of the data. These data sets are archived and used for future analysis. Monitoring results refer to the data that has been prepared for use as input for the visualization tools.

We have designed a set of tables, referred to as *mantra-tables*, which provide a standard framework for formatting different types of monitoring information collected from various sources. The two main benefits of such a framework:

- Analysis and aggregation modules remain transparent to the different raw data formats. We can make Mantra adopt to such changes simply by modifying the existing data processing modules or creating new ones. This process is further explained later in the section.

- Efficient data aggregation provides scalability by reducing processing requirements. It also facilitates a more accurate global view of various aspects of multicast by having a more consistent data set.

Based on their key data field(s), mantra-tables can be classified into two types: base tables and composite tables. Base tables hold information about the characteristics of basic multicast entities: groups, hosts, networks and Autonomous Systems (ASes). Composite tables hold data from multiple base tables, related to either the state of different protocols, or multicast routes. Figure 3 illustrates an example use of mantra-tables: a composition of route tables holds information about multicast routes between two network endpoints. Initial processing of raw data yields two types of route tables: MBGP route tables and traversed-paths route tables. An MBGP route table contains information about AS paths, i.e. each intermediate hop in a path is an AS. A traversed-path route table holds information related to actual paths that multicast packets take; i.e. intermediate hops are actual nodes, either routers or hosts. Both types are composite tables, composed of fields representing basic entities such as multicast networks, ASes, hosts and routers. A combination of these two types of route tables provides a further level of aggregation. While Mantra uses the original route tables for archival purposes, it uses aggregated route tables for analyzing the routing data.

4.2 Module-Groups: Mantra Tasks

We divide Mantra functionality into four phases, each with a module group that performs the corresponding task. These four phases are represented in Figure 2 and each is discussed below.

4.2.1 Data Collection

Data collection involves capturing state tables from multicast routers. As mentioned above due to lack of updated standards, SNMP data can not yet be used for monitoring newer multicast protocols. Consequently, Mantra obtains router state by logging into the routers and capturing router's memory tables directly. The module group for data collection in Mantra constitutes of two modules: the launcher-thread and the data-acquirer. The launcher-thread initiates data collection from routers and passes the data to the next phase of operations; the data-acquirer module is responsible for the actual data capture. At the start of each monitoring cycle, the launcher-thread starts multiple instances of the data-acquirer module and then waits for all of them to finish before passing the data to the next module group. Collection from multiple routers thus occurs in parallel. This not only reduces the overall time required for collection but also increases the temporal vicinity of data from different sources. Figure 4(a) illustrates the launcher thread.

Collection from a router involves three steps: (l) logging on to the router; (2) launching appropriate router commands to acquire memory tables; and (3) transferring these tables to the local system. The data-acquirer module also verifies completeness of collected data sets. Collected data sets may be incomplete due to either router or network unavailability. In the face of incomplete data, the data-acquirer will repeat its attempt to capture data periodically until it either reaches the limit of permissible attempts or a timeout period. The number of permissible attempts is router-specific and is always less than four. This limit is decided based on a router's workload and resources (memory and processing power). The timeout period is the same for all the routers and is usually half the time between two consecutive monitoring cycles. Figure 4(b) shows this algorithm.

4.2.2 Raw Data Processing

Data processing consists of converting raw data captured from external sources to mantra-tables. We have developed a conversion module for each type of data set collected. These modules act as plug-in parsers for converting associated data types to appropriate mantra-table(s). Using separate modules for different data types makes Mantra easily adaptable to changes in formats. New parsers can quickly and easily be substituted for existing ones. The level of processing in these modules varies. Two important tasks that these modules perform are:

- *Rectifying Erroneous Information*: Collected data can be erroneous and/or unrepresentative of the true picture for several reasons, including: implementation bugs in the routers; anomalous user behavior; and incompatibility among adjacent routers. We need to detect inaccurate information and either correct or remove erroneous values. For example, one problem often encountered is routers with bogus state for source-group pairs. This results in exaggerated counts for active multicast hosts and groups. Mantra eliminates these bogus entries using another router table to verify packet counts processed by a router for a particular source-group pair. All entries corresponding to source-group pairs with a zero packet count are considered bogus and ignored.

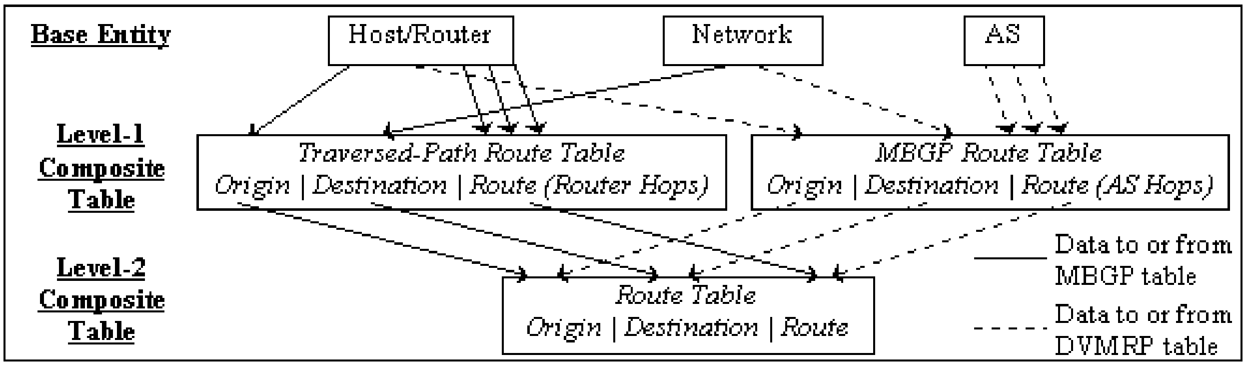

Figure 3: Architecture of Mantra-Tables.

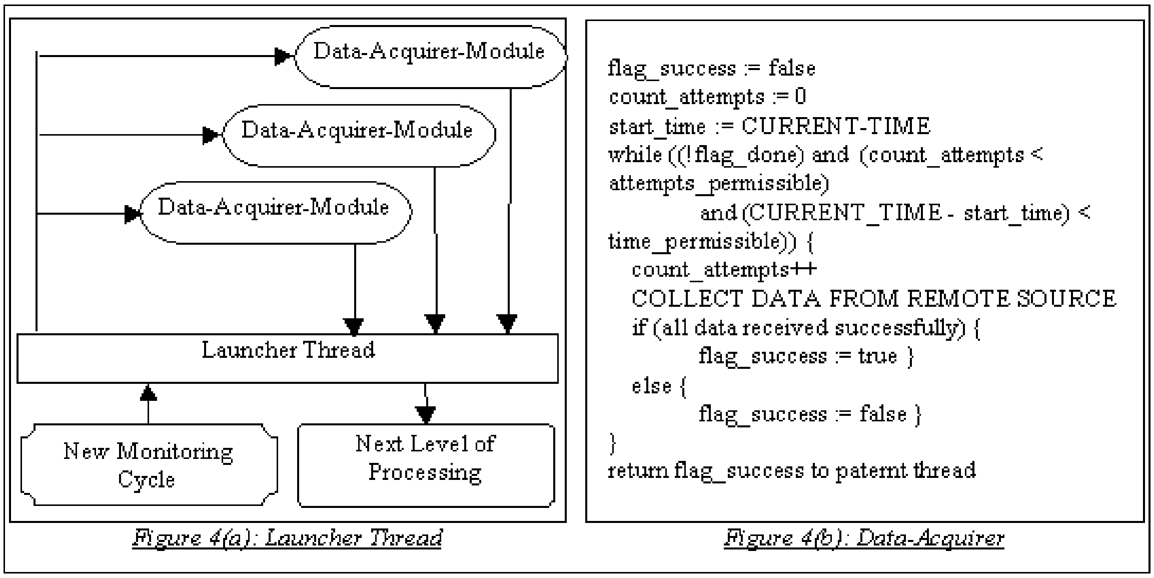

Figure 4: Data Collection.

- *Generating Mantra Tables*: During this stage, raw data modified during the previous phase is converted to mantra-tables. The conversion procedure is straightforward, and is typically a simple mapping of fields from raw tables to mantra-tables.

4.2.3 Data Logging

During this phase we archive mantra-tables containing processed data. These archives can later be used for indepth offline analysis. Our primary goal is to minimize storage space requirements without loss of information. Techniques used include:

- *Storing Only the Deltas* : Mantra stores only the entries that have been either withdrawn or added since the last monitoring cycle. This technique is very useful for storing MBGP or DVMRP tables; tables that do not change often.

- *Utilizing the Relational Nature* : Many mantra tables can be grouped into sets such that combining tables yields data on some important entity. In some cases, such as when the primary key constitutes most of the information in the table, we merge tables into a single table and store only that table.

- *Splitting the Tables* : The opposite of joining tables is also a useful technique. Mantra may split a composite table into constituent base tables for archival. For example, we may split an mroute table into two tables: the sources table and the groups table. The advantage of table-splitting is increased ability to store deltas, since the possibility of temporal consistency between base tables is higher.

4.2.4 Data Analysis and Aggregation

During this phase, Mantra further processes data for analysis. Some aspects of multicast that Mantra analyzes include membership patterns, usage of multicast address space, MSDP performance, routing stability, host characteristics, and network characteristics. The format of these results is optimized for use with different output interfaces. For example, Mantra stores results from group size analysis in simple tabular format-primarily useful for graphing. Other results represent topology trees and are stored for use in topology visualizations.

Mantra also performs two types of data aggregation during this phase: (1) aggregation of various types of data sets; and (2) aggregation of similar data sets from different sources. The first type of aggregation allows us to broaden the scope of monitoring beyond the analysis of individual protocols. For example, consider the case of MBGP and MSDP. Both tables are monitored individually by Mantra, but which are often needed together, e.g., to assess propagation of Source Active (SA) messages or density of MSDP sources in MBGP domains. The second type of aggregation is critical to obtaining a global picture of the infrastructure and relating various types of data.

5 Presenting Mantra Results

We use a set of static as well as interactive visualization mechanisms for presenting results. The types of results Mantra can produce include topological characteristics of the multicast infrastructure, geographical characteristics of its components, and statistics about various aspects of multicast. These visualizations support both a cursory examination of multicast statistics as well as detailed analysis of routing problems. In general, they allow study of multicast deployment, traffic load, protocol performance, and fault detection/isolation. In this section we describe these visualization mechanisms and demonstrate their utility with a case study of Mantra's use in detecting and isolating a routing problem.

5.1 Visualization Mechanisms

Mantra uses five output interfaces for presentation of results: (1) tables, (2) static graphs, (3) interactive graphs, (4) interactive topology maps and (5) interactive geographical representations. Of these, the interactive presentations offer important functionality and flexibility for enhancing the utility of our visualization efforts. We describe these interactive interfaces below and then present a case study using these interfaces in the next section.

Topology Maps. These provide graphical illustrations of different MBGP topology views. Mantra uses a Java-based, interactive topology visualization tool, Otter, for this purpose. Two types of views are: local views–the MBGP topology as seen from an individual router, and a global view–the MBGP topology obtained by aggregating data from different routers. Otter provides functionality through which user can interactively customize the colors of links and nodes based on values associated with them. Mantra can display statistics about various characteristics, including: node degree, link traffic, MSDP statistics, and distribution of participant hosts across administrative systems (ASes).

Geographic Placements. Placement provides a mapping of various components of the multicast infrastructure according to geographic location. Mantra uses the interactive Java-based tool, GeoPlot, to provide geographic placement of MBGP networks, DVMRP networks, participant hosts and RPs on a world map.

Interactive Graphs. Statistics are presented in the form of customizable graphs, using the MultiChart tool that we developed for Mantra. MultiChart provides a user-friendly interface for controlling different visualization aspects of the graphs, e.g., overlaying different graphs on the same display, choosing temporal range of data, and scaling graphs.

5.2 Isolating an Outage: A Case Study

In this section we present a case study of the use of Mantra to detect a routing problem, discover its cause, and evaluate its effects. The case we present pertains to a MBGP routing problem that we noticed on August 21, 1999 at ORIX, one of the routers that we collect data from. Below we present a step-by-step analysis.

Observation–The Unusual Results. Figure 5 graphs the number of session participants graphed over time. The most striking feature of this plot is the unusual drop in the number of sources at 1:56am on August 21, 1999. During this snapshot, the number of sources decreased by 23%. Such a severe and sudden drop is unlikely to be normal user behavior. It is likely the result of a routing problem.

Problem Solving. MBGP routing statistics derived from the data collected in the same time frame confirm that a routing problem occurred. Figure 6 shows the distribution of the number of MBGP routes as seen from ORIX. Here we noticed a sharp drop, about 22.2%, in the number of MBGP routes in the snapshot taken at 1:56am on August 21, 1999. This drop correlates with the number of participants (shown in Figure 5).

The number of routes in a router's MBGP table should typically remain relatively constant, so a large change is a strong indication of a potential routing problem. However, it is difficult to derive an exact correlation between the loss of MBGP routes and a decrease in the number of participants. Other factors may conspire to make drops caused by a single event look less synchronized. For example, a large number of joins in another part of the topology may minimize the perceived impact. Our efforts to visualize MBGP topology help to provide additional data for verifying outages. Figure 7 shows a screen shot of two consecutive snapshots of the MBGP topology overlaid on the same display. Links common to both topology snapshots are in light gray; those seen only in the second snapshot are black. The figure shows that an entire portion of the multicast infrastructure reachable via AS-704 is absent from the second snapshot.

Analysis of the Effects of the Problem. A detailed offline analysis showed that AS-704 provides links to several networks in Europe. Consequentially, loss in connectivity for AS-704 resulted in lost connectivity to most European networks. This confirms the loss in participant-hosts shown in Figure 6. The bar chart in Figure 8 shows statistics about the first-order domain of these hosts. Each bar reflects the number of hosts lost from that domain. Loss in connectivity to a large number of hosts present in Europe, especially in Germany (domain name suffix "de"), Czech Republic (domain name suffix "cz") and Greece (domain name suffix "gr") is evident.

Our efforts to place participants on a geographical map offers another useful result. Figure 9 shows geographic placement of participant hosts on a world map for both before and after. Figure 9(a) displays the hosts present before the drop, Figure 9(b) depicts the scenario after the drop. The difference in the density of the hosts in Europe between the two figures confirms the loss of connectivity to the domains "de", "cz" and "gr".

6 Conclusions

Mechanisms for monitoring the Internet infrastructure on a global scale hold great value. However, developing such mechanisms is challenging due to the relentless growth in deployment, heterogeneity among networks, fast pace of developments, and lack of support for inter-domain monitoring. Current monitoring systems provide only limited functionality, and are only marginally successful at intuitive visualization of results. With the emergence of the next generation of networking technologies, the need for new types of monitoring mechanisms has become urgent. Multicast is one such rapidly growing networking technology that requires effective monitoring to promote deployment and stable evolution. However, progress in multicast monitoring is hindered by several factors, including rapid changes in the field, incompatible standards, routing instability, and bugs in protocol implementations.

We have introduced Mantra, a tool developed for monitoring multicast on a global scale. Mantra collects network-layer data by capturing internal memory tables from routers across topologically and geographically diverse networks. Through Mantra we have developed a useful system for analyzing multicast behavior, including session characteristics, membership patterns, routing stability and MSDP performance. We have designed Mantra to be flexible; by keeping different modules independent of each other and by defining a standard data format for information flow amongst them we have created a model that can sustain intensive processing even as the number of networks and volume of monitored data grows. This model also enhances the scalability of Mantra as the processing of data sets can be easily distributed amongst different hosts or can be done at the source itself. Processed data can then be aggregated hierarchically and results can be generated based on a global snapshot.

We have described the visualization of monitoring results from Mantra with tools for interactive graphing of various statistics, topology visualizations, and geographic placement of different multicast subnets. We have also described how realtime results from Mantra can be used for gauging the current state of multicast and longer term results can be used for detecting faults, and discovering the cause of these faults. Finally, we have provided a case study to illustrate the utility of Mantra in troubleshooting a routing problem.

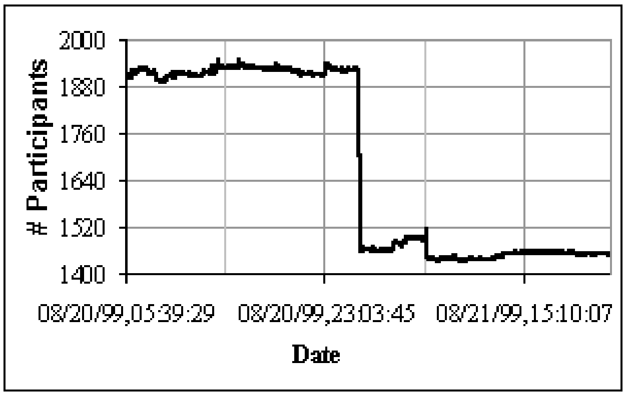

Figure 5: Number of Participants.

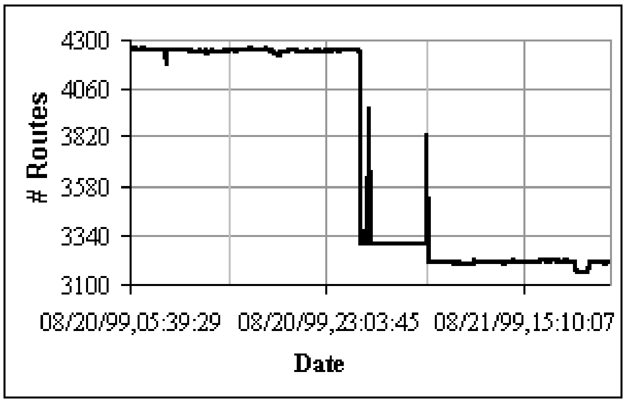

Figure 6: Number of MBGP Routes.

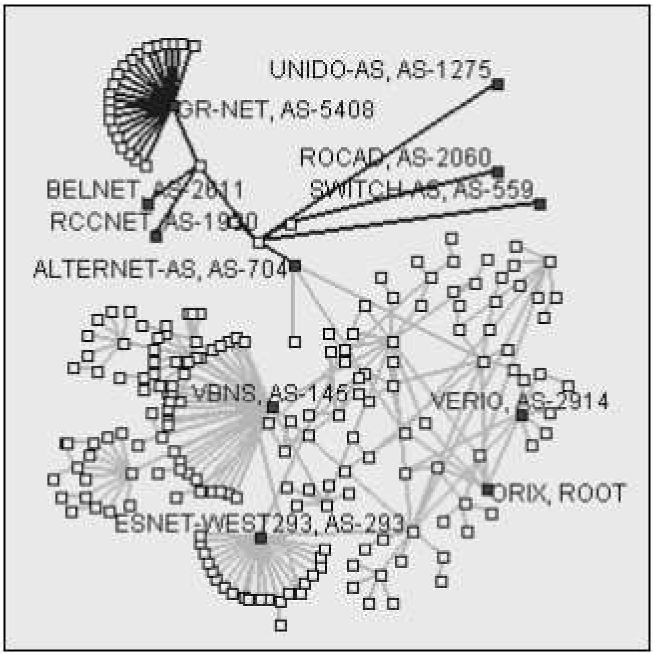

Figure 7: Loss in MBGP Connectivity.

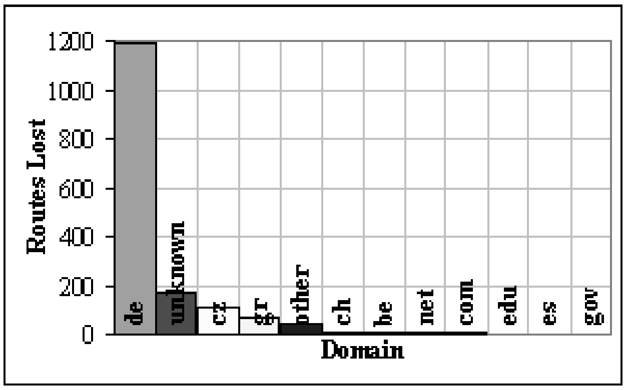

Figure 8: Domains that Lost Connectivity.

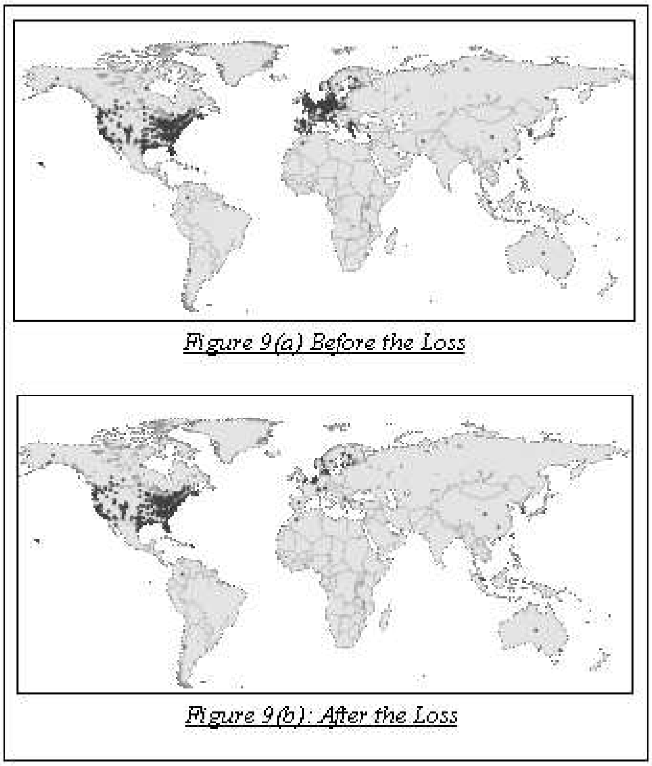

Figure 9: Affects of Loss in Connectivity.

References

[1] 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.

[2] D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol (DVMRP)." Internet Engineering Task Force (IETF), RFC 1075, November 1988.

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

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

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

[6] V. Paxson, "End-to-end routing behavior in the internet.," in *ACM Sigcomm*, (Stanford, California, USA), August 1996.

[7] R. Periakaruppan and N. E., "Gtrace - a graphical traceroute tool.," in *USENIX LISA*, (Seattle, Washington, USA), Novembeer 1999.

[8] D. McRobb, K. Claffy, and T. Monk, *Skitter: CAIDA's macroscopic Internet topology discovery and tracking tool.*, 1999. Available from http://www.caida.org/tools/skitter/.

[9] K. Claffy and B. Huffaker, *Macroscopic Internet visualization and measurement.* Available from http://www.caida.org/Tools/Mapnet/summary.html.

[10] 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.

[11] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner, "Visualizing the global topology of the mbone.," in *IEEE Symposium on Information Visualization*, (San Francisco, California, USA), October 1996.

[12] B. Huffaker, E. Nemeth, and K. Claffy, "Otter: A general-purpose network visualization tool.," in *INET*, (San Jose, California, USA), June 1999.

[13] R. Periakaruppan, *GeoPlot - A general purpose geographical visualization tool.* Available from http://www.caida.org/Tools/GeoPlot/.

[14] B. Fenner, *Multicast Traceroute (mtrace) 5.2*, September 1998. ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/.

[15] D. Makofske and K. Almeroth, "MHealth: A real-time graphical multicast monitoring tool for the MBone," in *Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, (Basking Ridge, New Jersey, USA), June 1999.

[16] 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.

[17] 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/.

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

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

[20] 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/.

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