Internet Engineering Task Force
INTERNET-DRAFT
draft-ietf-mboned-mrm-use-00.txt
Expires August 1999

MBONED Working Group
Kevin Almeroth
UCSB
Liming Wei
cisco Systems, Inc
February 26, 1999

Justification for and use of the Multicast Routing Monitor (MRM) Protocol

Status of this Memo

Abstract

This document motivates the need for the Multicast Routing Monitor (MRM) [MRM] protocol by describing the niche that exists for a router-based multicast management protocol. Using the "sufficient and necessary" argument, we suggest that existing protocols and techniques lack important management functionality. This document briefly describes the methodology used by MRM, justifies the existence of MRM, and describes some of the scenarios in which MRM will be of value.

1. Introduction

The Multicast Routing Monitor (MRM) protocol has been designed to assist in the detection and isolation of network faults related to the delivery of multicast traffic[MRM99]. In particular, management functions offered by MRM are specifically designed to monitor routing

operation, and assist in the investigation of routing anomalies and connectivity problems.

MRM has been designed with consideration for the other types of multicast management protocols and tools that are available. As

we will show, even though there are a wide variety of tools available today, there is a need for a router-based monitoring protocol. The justification for MRM as a new protocol has followed the ``necessary and sufficient'' premise. MRM is being developed because it is necessary when comparing its functions to those offered by alternatives like the Real Time Control Protocol (RTCP) [RTP] and the Simple Network Management Protocol (SNMP) [SNMPV1,SNMPV2].

Furthermore, MRM is being developed because it is sufficient in providing the functions needed by its target class of applications. Using this reasoning, MRM will offered functions and provide multicast traffic management that no other protocols currently offer.

2. Overview of MRM

MRM is a protocol intended to be implemented in both routers and end stations. The operation of MRM is based on communication and coordination between three types of network entities.

* MRM Manager: The MRM manager provides an interface enabling a user to configure and execute tests, and then collect and present results. The MRM manager communicates with MRM testers who are instructed to source and/or sink multicast traffic. The MRM manager, through beacon messages, also maintains and modifies the set of MRM testers.

* Test Sender (TS): A test sender is basically responsible for sourcing multicast traffic. TSs will receive authenticated requests from the MRM manager and will send a specified number of multicast packets to a specified multicast address with a specified inter-transmission time between packets.

* Test Receiver (TR): Based on instructions from an MRM manager, a test receiver is expected to either explicitly join a multicast group or simply monitor traffic on a specified group address. Based on thresholds specified by the MRM manager, a TR will report faults. Additionally, the MRM manager may request TR reports regardless of whether any thresholds were

violated. One of the keys to scalability is ensuring that a large number of TRs don't overwhelm the MRM manager with traffic. Scalability is handled using a combination of techniques including report suppression and aggregation.

As the MRM protocol specification indicates about itself, ``it only specifies the types of information a MRM manager can obtain, and the protocol used to acquire such information. How an MRM manager processes or presents the diagnostic information is an implementation issue.'' These functions are expected to be provided using companion management tools. Furthermore, the MRM protocol specification does not fully describe the scenarios in which MRM is expected to be useful. Such functions and scenarios are described in Section 4 of this document.

---

3. Justification

MRM provides a set of functions not provided by any of the commonly used MBone debugging protocols and tools. Most of the tools used in the MBone today fall into one of three categories: (1) SNMP-based tools like Mview [MVIEW] or Mstat [MSTAT]; (2) RTCP-based tools like Mhealth [MHEALTH], RTPmon [RTPMON], or MultiMON [MULTIMON]; or (3) multicast route tracing tools like Mtrace [MTRACE1,MTRACE2]. MRM, in addition to being an independent management tool, can be used in conjunction with these other tools to provide a richer set of management functions. Some of the reasons why the above protocols or tools fail are discussed in the following paragraphs.

* SNMP: SNMP provides a mechanism to poll devices for information or to have alarms generated when certain events occur. The problem with SNMP is that a wide ranging failure could potentially overwhelm a management station. For example, consider a scenario in which SNMP agents in a particular multicast tree are configured to generate an alarm if the packet loss exceeds a certain level. Then consider the implosion that would occur if a link close to the root becomes congested, and a majority of group members generated alarms. This scenario demonstrates the basic drawbacks of SNMP: a general lack of scalability especially when considering that large number of devices/hosts that may be involved in a multicast group. Scalability arguments do not preclude the use of SNMP, but a manager using SNMP to manage multicast would have to be extremely careful in deciding how to configure the network. In fact, properly configuring network devices to provide sufficient

management information while avoiding management-induced congestion or implosion may be prohibitive in most networks.

* RTCP: RTCP has a much more scalable feedback mechanism but it has its own deficiencies. The scalability of RTCP is based on a random wait time chosen from an interval calculated by each group member and based on an estimate of the overall group size. The larger the group, the larger the wait interval, and the longer the average inter-packet time between RTCP feedback messages. The goal of the RTCP feedback mechanism to is consume bandwidth equal to 5% of the data traffic rate. While this algorithm seems reasonable it can be problematic as a tool to management multicast traffic. Some reasons include:

  o RTCP feedback is multicast to all group members, and given that receivers will have heterogeneous bandwidth capabilities, even scalable feedback has the potential to overwhelm some receivers.

  o In many management applications there is no need for feedback data to be transmitted to all group members. And if privacy is an issue, the group-wide delivery of RTCP is even less desirable.

  o RTCP feedback provides only a single, end-to-end loss and jitter value. More generally, RTCP contains only a very small amount of information useful for debugging purposes. MRM is designed to include a broader range of information, including packet duplication statistics, and also to be extensible.

While some of the problems with RTCP are being addressed by redefining the standard to allow more flexibility in the use of RTCP [RTPNEW], these efforts do not solve all of the problems. In particular, the most critical deficiency of RTCP is a lack of detailed routing information. In particular, when trying to isolate routing faults, the end-to-end style feedback provided by RTCP is unlikely to have sufficient granularity. To address this problem, some RTCP-based tools are used in combination with other tools. For example, RTPmon and mtrace are commonly used together. However, the major drawback of this solution is that

it fails to provide the sort of traffic origination and flexible group membership services offered by MRM.

* Mtrace: Mtrace is a tool designed to provide hop-by-hop path information for a specific source and destination. It is a useful tool for figuring out a multicast path and round trip information. For a specific group, mtrace will also tell a user hop-by-hop packet loss. Coupled with RTCP feedback, mtrace can be used to monitor many of the relevant factors for an active source and group including per-receiver loss, hop-by-hop loss, tree topology, jitter, and round trip time. Several tools in development, including Mhealth, provide a graphical real-time display of group statistics. However, mtrace (coupled with other tools) only provides information about active groups. Attempting to do fault detection, or more specifically, fault pre-detection, is nearly impossible. The common paradigm today is to gather a set of willing participants who then join a ``debugging'' session. Further complicating the problem is that sometimes, starting an MBone tool in a remote location to receive and transmit RTCP reports is not possible. One solution to this problem is a ``dumb'', non-GUI tool that simply receives and responds to an RTP stream. While tools like this have been discussed, but none are widely available, and even if they were, attempting to rapidly configure and change group membership would be laborious at best. MRM is designed with the specific purpose of facilitating on-the-fly, adhoc test multicast senders and receivers to test a variety of multicast group configurations.

4. Scenarios for Use of MRM

MRM is designed to provide automated fault detection and isolation services for multicast traffic. In order to support these services with any kind of automation, MRM must be both flexible and scalable. MRM scalability implies the ability detect faults without raising so many alarms that additional problems are caused from the delivery of alarm messages. One problem, in particular, is response implosion at the MRM manager. MRM flexibility implies the ability to isolate faults by sourcing traffic from anywhere in the network and collecting statistics from any node or subset of nodes. In addition to basic fault detection and isolation, MRM is intended to provide more advanced functions. These extended functions include:

* Fault logging and real-time (passive) monitoring functions.

* Pro-active test (fault isolation) include service provisioning and impact analysis.

The remainder of this section is dedicated to the description of scenarios in which MRM functions are expected to be used.

* Pre-Event Testing: One of the best examples of this type of scenario is the MBone delivery of two audio/video channels from the IETF meetings held three times a year all over the world. Preceding the week of meetings for each IETF, staff members install a terminal room and establish network connectivity including multicast capability. In some cases, setup activities occur weeks, days, or hours before the first meeting Monday morning. Verifying that multicast routing is working both into and out of the IETF meeting rooms can be a challenge. Verification is especially challenging because the IETF meetings have a world-wide audience. Ensuring that multicast is working at even a small number of remote sites is difficult. One problem that sometimes occurs is that the MBone equipment, including cameras and workstations, may not be available when the network is first turned on. In these cases, there are no multicast-capable sources or receivers inside the IETF network. MRM would alleviate this problem by allowing testing of multicast in both directions. Furthermore, MRM would also allow someone not yet on site to test multicast connectivity. Relatively extensive testing can be performed by choosing a set of Test Receivers representative of the world-wide distribution of actual IETF participants. MRM would allow the IETF staff and the ISP to observe where major network bottlenecks are occurring. In some cases, early discovery of problems could lead to fixes in time for the event.

These techniques, used for pre-event testing at ``nomadic events'', would also be appropriate for estimating the quality of transmissions events in ``non-nomadic'' networks. Instead of the IETF or an academic conference, an MRM manager might want to estimate the loss, delay, and jitter for a frequently scheduled

event like an MBone lecture or company event. Instead of waiting until the event starts and using a tool like RTPmon, an MRM manager can set up a test session any time before the session starts, and evaluate the quality to most, if not all of the critical company locations. In the MBone today, if a transmitter wants to perform this kind of testing, the transmitter will, out-of-band, have to ask several friends to join a test session and then send a multicast stream and monitor RTCP reports. Obviously, this method is not very compelling.

* Classic Fault Isolation: A second scenario that MRM is designed to assist a network manager in is classic fault isolation. Like unicast routing, multicast routing problems can be very difficult to debug. And unlike unicast routing, the additional complexities of providing efficient, one-to-many delivery can introduce additional bugs that are difficult to find. To date, a significant number of strategies, tools, and techniques have been developed, built, and proposed [MDH]. However, these attempts generally require a significant level of multicast routing expertise and experience, characteristics not always found among NOC personnel. As a result, MRM is designed to offer a layer of abstraction between multicast route management and the intricacies of multicast routing. MRM is also designed not to be completely independent of the strategies, tools, and techniques already in use today. MRM and existing tools can work in concert to isolate multicast routing problems.

  MRM's design offers some important flexibility in isolating multicast routing faults. In particular, the ability to specify a transmission rate allows a manager to closely inspect single, infrequently transmitted packets. Also, the ability to easily add and remove members from the group of Test Receivers allows a manager to quickly and efficiently affect the topology of the multicast tree.

* Session Monitoring: The scenarios discussed so far followed logically, from verifying multicast connectivity to isolating any potential faults. The next key scenario is monitoring of existing, active sessions. Such groups will have a well-known multicast address, and might be exchanging group membership information via RTCP reports or some other out-of-band mechanism. If the group is small, and feedback from each receiver is

important, the set of test receivers can be configured to send reports to the MRM manager via unicast. If the group is large and complete feedback is not necessary, the set of test receivers can be frequently adjusted to represent some statistical sampling of the group. The ability to send statistical reports via unicast helps to improve the scalability of session monitoring by not overwhelming all receivers with all reports. Finally, if the group is using multicast tools that do not use RTCP and use no real-time signaling, generation of a real-time list of group members may be difficult to create. Other techniques will have to be used. One network-layer approach might be to use SNMP information to find the set of links in the multicast tree. A more simple approach might depend on other available information like the fact that most users start the multicast tool via a WWW page. In this case, HTTP server logs can be used to estimate group membership.

* Fault Logging: In the case when session monitoring identifies the existence of a fault, a range of fault logging functions may be required. At one extreme, the MRM manager may simply need to be alerted when faults occur so that appropriate investigative measures can be taken. At the other extreme, service contracts may depend on the provision of service with certain guarantees. Any outages will need to be closely tracked. These two extremes again demonstrate the need for MRM to be flexible. In particular, when faults need to be closely monitored and logged, a wide-scale outage may itself cause a heavy load on the network. While identifying the exact load capable of being supported by a distressed network is beyond the scope of MRM, MRM does and will support scalability and aggregation functions.

8. Security

Security issues are discussed in the MRM protocol description [MRM].

9. Authors' Addresses

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

Liming Wei
cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA
lwei@cisco.com

10. References

[MRM] L. Wei, and D. Farinacci, "Multicast Routing Monitor (MRM)", IETF Internet-Draft, draft-ietf-mboned-mrm-*.txt, February 1999.

[RTP] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 1889, January 1996.

[SNMPV1] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "Simple Network Management Protocol", IETF RFC 1157, May 1990.

[SNMPV2] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF RFC 1905, January 1996.

[MVIEW] D. Thaler, "Mview Tool", http://www.merit.edu/~mbone/mviewdoc/Welcome.html.

[MSTAT] B. Fenner, et al., "Mstat", Available as part of mrouted at ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/.

[MHEALTH] D. Makofske, and K. Almeroth, "Mhealth -- Real-Time Multicast Tree Health Monitoring Tool", http://imj.ucsb.edu/mhealth/, August 1998.

[RTPMON] A. Swan, and D. Bacher, "RTPmon", ftp://mm-ftp.cs.berkeley.edu/pub/rtpmon/, January 1997.

[MULTIMON] J. Robinson, and J. Stewart, "MultiMON 2.0 -- Multicast Network Monitor", http://www.merci.crc.ca/mbone/MultiMON/, August 1998.

[MTRACE1] B. Fenner, et al., "Multicast Traceroute (mtrace) 5.2", ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/ September 1998.

[MTRACE2] B. Fenner, and S. Casner, "A `traceroute' Facility for IP Multicast", IETF Internet-Draft, draft-ietf-idmr-traceroute-ipm-*.txt, November 1995.

[RTPNEW] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", IETF Internet-Draft, draft-ietf-avt-rtp-new-*.txt", November 1998.

[MDH] D. Thaler, and B. Aboba, "Multicast Debugging Handbook", IETF Internet-Draft, draft-ietf-mboned-mdh-*.txt, October 1998.