MBone Deployment Working Group
Internet Engineering Task Force
Internet Draft
October 1999
Expires: April 1999

Kevin Almeroth (ed)
UCSB
Liming Wei
Siara Systems, Inc.
Dino Farinacci
Cisco

Multicast Reachability Monitor (MRM)
<draft-ietf-mboned-mrm-00.txt>

Status of this Memo

Abstract

MRM facilitates automated fault detection and fault isolation in a large multicast routing infrastructure. It is designed to alarm a network administrator of multicast reachability problems in close to real-time.

There are two basic types of components in MRM, MRM manager and MRM testers. This document specifies the protocol with which the two MRM components communicate, the types of operations the testers perform, and information an MRM manager can obtain.

Table of Contents

1. Introduction

The Multicast Reachability Monitor (MRM) is a network fault detection and isolation mechanism for administering a multicast routing infrastructure. It is proposed in response to requests from network managers and users who need more systematic ways to get up-to-date multicast reachability status. For these purposes, existing tools are inefficient and inconvenient to use across large numbers of systems. The companion document [mrm-use] contains additional information on justification and usage guidelines for MRM.

The design goals for MRM include:

(1) Close to real-time detection and alarm of network problems, independent of user input;

(2) Good coverage over the network, both in terms of the number of systems to be monitored, and the types of diagnostics to be performed;

(3) Good extensibility and relative independence of other specific diagnostic tools and protocols (we borrow packet formats from RTPv2, but almost nothing else from the RTP protocol). This makes it easy to incorporate newer diagnostic tools as they become available.

1.1 Partitioning Network Monitoring Tasks

Functionally, the task of monitoring a multicast domain can be divided into two subtasks:

(1) Fault detection
(2) Fault isolation

In the fault detection phase, the participating MRM systems do not need much detail about the nature of the fault. The mechanism can be very simple and brute force. Data packets can be originated from designated locations in the network and reception conditions monitored from other locations.

In the fault isolation phase, depending on the types of fault identified, the MRM manager can use proper tools to isolate the fault and hopefully pin-point the location or reasons of the fault.

The rest of this document is organized as follows, Section 2 describes the MRM framework and details of the MRM protocol; Section 3 describes the usage of the well known MRM addresses and ports; Section 4 specifies packet formats; Sections 5 discusses the MRM authentication mechanisms; Section 6 discusses a few security issues; and Section 8 gives an example of MRM setup.

2. Functions of the MRM Mechanism

An MRM based fault monitoring system consists of two types of components: (1) an MRM manager that configures tests, collects and presents fault information, and (2) MRM testers that source or sink test traffic. These components collaborate to accomplish the two functions of MRM: fault detection and fault isolation.

The MRM testers can be any routing devices or trusted end hosts. They provide statistics about received data packets, to be used to derive the network reachability status. These data packets can be sourced by a router acting as an MRM tester, in response to a request from the MRM manager. A system originating MRM data packets for testing purposes is also called a Test Source (TS). A configured set of MRM testers receiving the test traffic, and collecting receiver statistics are also called Test Receivers (TRs).

An MRM manager initiates configuration requests to the MRM testers and assigns the roles of TSs and TRs. The MRM manager informs the TSs and TRs the types of monitoring or diagnostic tests to run. The MRM manager also specifies the type of reports the TRs should send.

To guard against attacks on the MRM systems, IPsec Authentication Header (AH) [AH] is used with HMAC-MD5 transformation as the standard authentication algorithm. Authentication should always be enabled, especially when MRM is used to monitor production services.

Note that this document only specifies the types of information an 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. An MRM manager can be as simple as a command line wrapper of requests with simple display functions, it can also be more sophisticated and incorporated as part of a operational network monitoring tool in daily use by a network operation center (NOC).

2.1 Fault Detection

Multicast routing can behave abnormally in different ways. The following are a few common types of faults:

(1) Topological disconnectivity

The network topology for multicast routing is disconnected. For example when a route for a subset of the networks are not in the topology table.

(2) Black holes in forwarding path

No multicast packets can get through to certain receivers, even though the network topology is perhaps intact. A possible cause could be disabled multicast forwarding. Another possibility is pruning errors,n e.g. due to inconsistent actions and timer values on a multi-access LAN.

(3) Excessive/persistent Losses

Packets flow, but with excessive losses over extended period of time. The possible causes include heavy congestion, line errors or misuse of forwarding modes, etc.

(4) Excessive duplicates

Packets arrive at the receivers, but with large numbers of duplicates.

(5) Others

Other types of fault that can be detected, e.g. non-pruners as a failure mode. A non-pruning neighbor can be a sink for all multicast traffic at all times, even if no receivers exist behind that neighbor. This is "outlawed" by the "MBONE-community" [jhawk]. Detecting the existence of such system in an inter-domain scenario, however, is not trivial. We leave this task to the next iteration of MRM refinement.

2.2 Fault Isolation

Fault isolation is initiated by the MRM manager. For different types of faults detected, various tools can be used to isolate the faults to small areas in the network. Currently, the tools available for this purpose includes but not limited to mtrace [MTRACE}, MIBs based debugging tools based, http-based status report mechanism and remote execution mechanisms.

When one tool is not sufficient, a combination of tools can be applied. In general, MRM is designed to be flexible about the types of tools it can utilize. Integrating the functionality of other tools into MRM is an implementation issue for the MRM manager.

2.3 The Protocol

As stated above, the task of monitoring multicast reachability is accomplished by letting an MRM manager configure the MRM testers to perform fault detection and isolation tests. The MRM manager summarizes or displays the collected reports for the network operators, in an implementation specific way.

The MRM manager keeps a list of tester addresses. The relevant routing devices are administratively configured as candidate MRM testers. These testers will become active TSs and TRs once they accept and process requests from an MRM manager.

We chose to use RTPv2 encapsulation for the following MRM messages: fault report messages from TRs and optionally some test data packets. This is to allow re-use of existing RTP based reception mechanisms. Note that despite the use of the RTPv2 packet format, the design goals and rules for the MRM message exchange protocol are entirely different from those specified in RTP.

2.3.1 MRM Manager Requests

An MRM manager sends Test Sender requests to the TSs, and Test Receiver requests to the TRs.

The MRM manager optionally transmits periodic beacon requests to the well-known MRM multicast address MRM-ADDR (224.0.1.111) that all TSs and TRs listen to. This beacon mechanism has three purposes:

(1) For the TSs and TRs to learn the liveness of the MRM manager;

(2) As a medium to periodically refresh requests, in order for testers to recover lost MRM messages, configurations or state (e.g. across reboots).

(3) Inform a large group of test participants that an MRM session has been changed or cancelled.

The use of beacon messages by the manager is optional primarily because multicast connectivity between the manager and TSs and TRs may not exist. As a result, while beacon messages may add robustness, they should not be relied on to provide critical functionality. While the manager chooses whether or not to send beacon messages, TSs and TRs must be prepared to handle these messages.

The MRM manager may send a request to either a unicast address, or multicast address 224.0.1.111. When the message is sent via unreliable unicast transport (UDP), the recipient must send a positive acknowledgement after it has received that request. Unacknowledged request messages are retransmitted.

2.3.1.1 MRM Manager Beacon Message

The MRM manager periodically transmits beacon messages to advertise its liveness to all MRM testers. This message is UDP-encapsulated. The sender's timestamp can be used to calculate the jitters in delay between subsequent beacon messages.

The recommended default Beacon message interval is 1 minute. The MRM manager may piggyback the manager requests on the beacon messages. This potentially reduces the need to individually check and repair each tester's setup state, while still able to provide reliability through a soft-state refresh mechanism.

2.3.1.2 Test Sender Requests (TSRs)

A Test Sender request is first unicast delivered to a TS, then refreshed through multicast delivery via the MRM beacon mechanism. A Test Sender request specifies one of the following two ways to generate test packets:

(1) Local packet trigger. This request includes the following parameters:

(a) intervals between two consecutive test packets;
(b) format and length of test packets (e.g. RTP/UDP);
(c) multicast address for the test group.

If a TS accepts this local packet trigger, it will start sending periodic test packets, at intervals specified in the MRM request message. The IP address of the MRM manager will be used as the ID for all test packets originated by the TS under this request. To detect loops and packet losses, all test packets also contain a monotonically increasing sequence number (if encapsulated in RTP, this would be the RTP sequence number).

(2) Proxy packet trigger (see Section 5 for security impacts).

This request lets a TS send a (sequence of) MRM test packet(s), using the IP source address provided by the manager request message. This request contains all parameters a local packet trigger has, plus a proxy-source address.

This request is useful for monitoring intra-domain multicast connectivity for external sources. A proxy packet trigger can be used to inject packets into the local domain, pretending there is an active source external of the local domain. Inside the domain, as far as forwarding is concerned, these packets are indistinguishable from packets originated from a real external source. For security reasons, proxy packet triggers should be enabled very carefully.

TSR messages are also used to stop ongoing tests. By re-sending the original TSR packet, but with a holdtime of zero, a test can be stopped. NOTE: TRR messages with a holdtime of zero should also be sent to each test receiver participating in the test.

2.3.1.3 Test Receiver Requests (TRRs)

An MRM status request is first addressed to a unicast address of a TR, and subsequently should be carried in the MRM manager beacon messages sent to 224.0.1.111.

Each such request carries a holdtime of the request, after which the TR can safely discard any information collected. A TRR with a holdtime of zero implies that an ongoing test should be terminated. The TRR specifies how each TR should collect the reception data.

The following are the request types for the TRs:

(1) Monitor multicast group. This request has the following fields:

(a) J-bit. If set, the TR will join the specified group, as if it were a host with a member of that group.

If a tester did an IGMP join at the beginning of a test, when the MRM request expires, the IGMP group membership should be withdrawn.

When a TR is instructed to join a data group of an existing application (e.g. a heartbeat [heartbeat] group), it is wise to assess the impact on the TR system if the data rate is non-trivial.

Furthermore, the use of existing groups introduces uncertainty as to whether the source is actually transmitting. Because TRs expect a constant flow of packets, using existing group traffic, which may be bursty, introduces uncertainty at the receiver as to whether traffic is flowing but is being lost or not being sent.

(b) The address of the group to be monitored;

(c) List of source addresses to record reception quality information;

(d) Threshold description for triggering fault reports.

This draft revision only specifies packet loss based threshold. A fault is detected if the packet loss percentage has reached the threshold during the specified time window for measurement. Once set, the width of this window is fixed. But the starting point (or left edge) of the window keeps moving forward.

Reception quality data within the measurement window should be kept so that threshold calculations can be made continuously as the window moves forward in time.

(e) Maximum and minimum delays to trigger fault report. The report is sent at a randomized delay between the minimum and the maximum value.

(f) Type of error reports solicited. It is possible to specify an RTCP report (as if the test session uses RTP), or a native MRM report. Currently, MRM only supports RTP-based reports.

(2) Fault isolation request. This request is sent after a fault is detected and identified by the MRM manager. It specifies the tool and its associated parameters.

Details about this request message will be added in a future revision of the MRM specification.

(3) Poll for receiver statistics. This instructs the TR to report the statistics (historic data) it has collected via Status Reports. The TR will send Status Reports, even if the fault threshold has not been reached. Section 2.3.2 describes the status report mechanism in detail.

When large numbers of TRs are activated, a fault in the upstream of a tree may result in many TRs sending reports at the same time. To address the issue of possible report implosion, each TR may use one of the following two strategies:

(1) Report via unicast message. The MRM manager assigns a pre-determined report-delay (as part of the configuration design task) to each TR. Each TR upon detecting a fault, will randomly delay the sending of its report based on the pre-set delay period. This would allow an MRM system to monitor networks with up to thousands of systems without unreasonable compromises in detection response times.

(2) Each TR may be instructed to report the detected faults to the well-known MRM group address 224.0.1.111 using the RTCP format [RFC1889] and does back-off or suppression when duplicate reports from other Testers are seen. If using this strategy the manager should realize that using multicast to report a problem with

multicast may not be particularly robust.

This method allows the use of existing RTP-based monitoring tools in the initial deployment and experiments with MRM. However, it will prevent the MRM manager from learning a complete list of receivers affected by a specific fault. When multicast routing is not working correctly, these reports may not be heard by the MRM manager, leaving faults undetected and not alarmed by the MRM manager. It is recommended that all designs include at least a subset of TRs that (take turns to) unicast their reports.

There is ambiguity in MRM not hearing any fault report from a certain TR. It could be due to fault-free network status, the crash of the TR, or problems in the transport mechanism between the TR and the MRM manager. Requiring each TR to frequently report its liveness and to only do unicast fault report may work for a moderate number of testers, but may put undue burden on the network for larger numbers of testers. A compromising solution is to only report liveness from a critical portion of the network and do unicast fault report from a subset of the testers. The periodic liveness reports serve two purposes: (1) it provides evidence that the tester is still alive; (2) it indicates the conditions of the tester functions. The request-ack messages are used as tester liveness reports.

Note that the fault isolation phase does not necessarily require the MRM manager to send a Fault Isolation Request to a TR. E.g, in a typical network today, a third party mtrace issued by the MRM manager may be sufficient to identify the faulty hop excessively dropping packets if the tester is not completely blacked out.

2.3.2 Status Reports

These reports are sent by the TRs to the MRM manager, in response to a status request.

For now, we use RTP [RFC1889] "receiver report (RR)" packet format to carry receiver's status reports. It is expected that the MRM-native report format (to be defined in future draft revisions) will carry more useful information about the routing state and statistics.

Please refer to RFC1889 for details on the packet formats. Here we define the few RTCP items used by MRM (or loosely referred to as RTP profile for MRM):

SSRC (Synchronization source) of packet sender:
IP address of the Test Sender.

Extended highest sequence number received:
Highest sequence number seen by the Test Receiver.

Fraction loss:
Percent loss of Test Sender data.

Cumulative number of packets lost:
Total number of RTP data packets from SSRC lost within this reception window period.

Inter-arrival Jitter:
Set to zero when sent, ignored when received.

When this report is UDP encapsulated and unicast addressed to the MRM manager, it is explicitly acknowledged. The acknowledgement packet contains the RTCP header portion of the original packet after the MRM header.

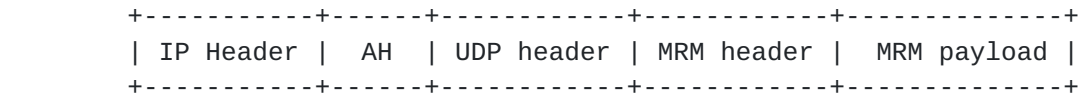
3. Use of MRM Well Known Addresses and Ports

Once all TS and TR systems are configured, they join the well-known MRM control group MRM-ADDR (224.0.1.111) and listen to the well-known MRM UDP port MRM-MANAGER-PORT (679).

The MRM beacon messages are periodically sent to 224.0.1.111 UDP port 679.

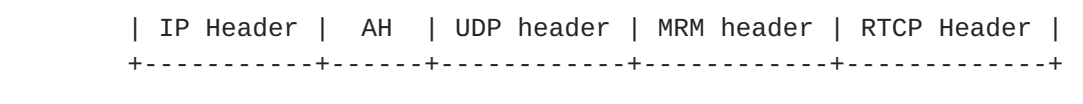
4. Message Formats

By default, MRM control messages are encapsulated inside UDP, and an IP authentication header (AH) [KA98], is inserted in between the IP header and the UDP header, as shown below:

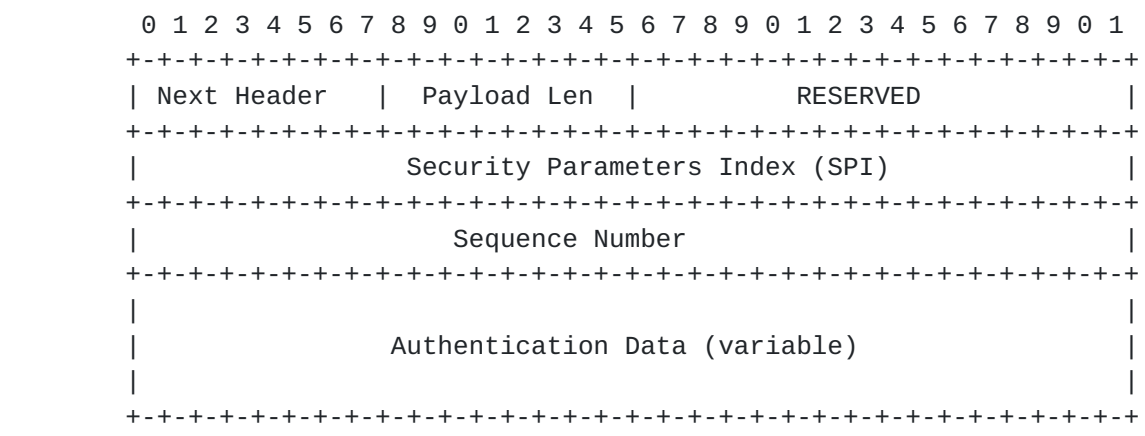


The MRM status report in RTCP format is:

The MRM ACK packet format is:



The inserted AH is reproduced below:



As specified in [KA98], the following are the default values for the fields above:

Next Header: 17, the value for UDP protocol.

Payload Len: 5, when MD5 is used (total length is 7 32-bit words).

RESERVED: 0 when sent, ignored when received.

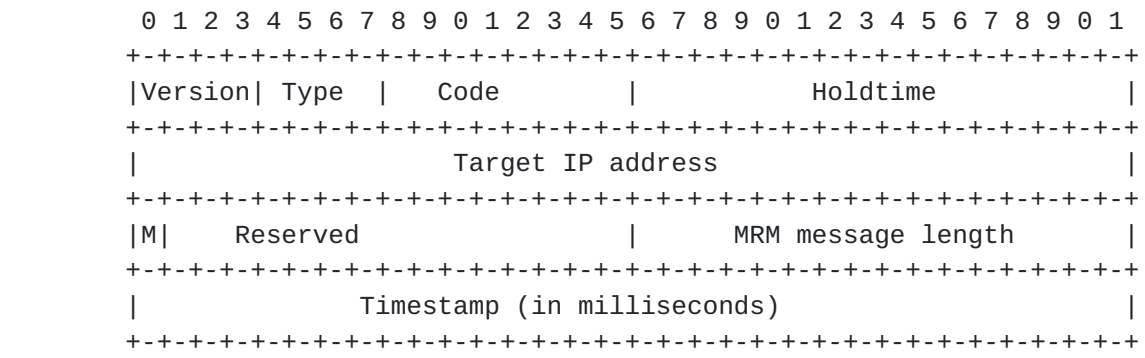
SPI: 0 - 50, when using configured MD5 keys

Sequence Number: the sequence number

Authentication Data: message digest

4.1 MRM Message Header

The MRM message header contains 4 32-bit words.



Version: 4 bits
This revision defines version 1 of MRM.

Type: 4 bits
The defined message types are:

0 = Beacon (from MRM manager to all testers)
1 = TS Request (from MRM manager to Test Senders)
2 = TR Request (from MRM manager to Test Receivers)
3 = Status Response (from TR to the MRM manager)
4 = TS Request Ack (from TS to MRM manager)
5 = TR request Ack (from TR to MRM manager)
6 = Status Response Ack (from MRM manager to TR)

Code: 8 bits
Defined according to each packet type.

Holdtime: 16 bits
Maximum duration in seconds this message should be honored.

Target IP address: 32 bits
The unicast address of the intended recipient of this message.

M: 1 bit,
0: last MRM request message in this packet.
1: more MRM request messages follow in the same packet.

When multiple MRM messages are grouped into one packet, the IP/AH/UDP headers of the second and all subsequent MRM messages are omitted. The total length of the IP packet will reflect the the sum of lengths of all MRM messages in the packet.

4.2 MRM Manager Beacon Message

This message is UDP encapsulated, addressed to UDP port MRM-MANAGER-PORT. The outstanding Test Sender Requests and Test Receiver Requests are included in the beacon message. The individual MRM headers are included with these TSR/TRRs.

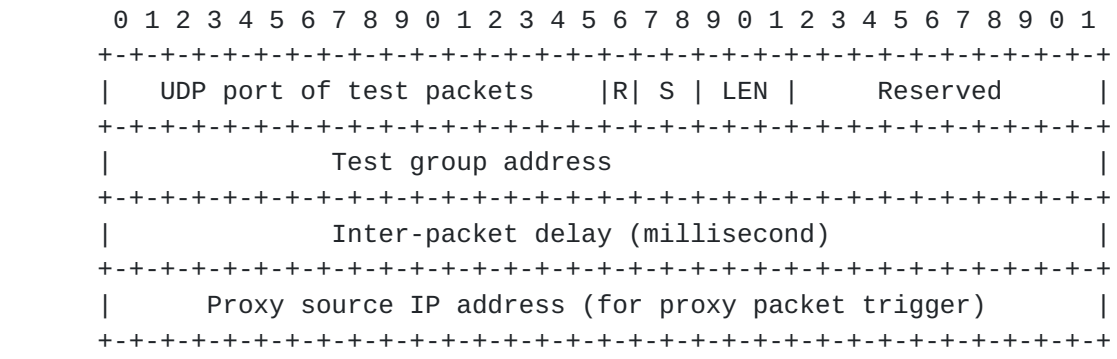
4.3 Test Sender Request (TSR)

There are two code values for a TSR:

0: Local packet trigger
1: Proxy packet trigger

NOTE: A host-based implementation is not expected to provide proxy packet capability.

Following the MRM message header are the fields for the source specification request:



UDP port of test packets: 16 bits
   UDP port of test packets.

R: 1 bit
   0: Tester will originate RTP/UDP encapsulated test packets
   1: Tester will originate another kind of packet (not used)

S: 2 bits
   00: send on the targeted interface only
   01: send on all the multicast enabled interfaces
   10: send on test-send enabled interfaces
   11: Unused

LEN: 3 bits (optional)
   Size of the packets to be sourced. The length field represents
   a multiple of 16 bytes. The range of possible packet sizes is
   16 bytes to 2048 bytes (2^7)*(16 bytes). The LEN field is
   optional. If ignored, test senders should send 16 byte packets.

Reserved: 10 bits
   Set to zero when sent. Ignored with received.

Inter-packet delay: 32 bits
   Number of milliseconds between consecutive test packets.

Test group address: 32 bits
   Multicast address of the test group.

Proxy source IP address: 32 bits
   IP address of the source to proxy packet for. This field
   exists only for a proxy packet trigger request.

4.4 Test Receiver Requests (TRR)

The following are code values for status request messages:

   0: Monitor multicast group (Monitor request)
   1: Poll for receiver statistics (Poll request)
   2: Fault isolation request (not used in this revision)

Message format for monitor and poll requests:

J: 1 bit
0: Don't join the multicast group to be monitored.
1: Join the multicast group to be monitored.

R: 1 bit
0: Fault report should be sent in RTCP format
1: Fault report should be sent in native MRM format (not used).

Reserved:
Zeroed when sent, ignored when received.

Number of sources to monitor: 16 bit
The number of sources this target tester should monitor. When all sources for the test group are monitored, this field is set to 1, and the corresponding source address field is set to 0.0.0.0.

Thres index: 8 bits
Always 0. Index of the criteria for determining a threshold for a fault. The value of this index determines the content

for the "Threshold description Block".

Pkt loss (%): 8 bits
Percentage of packet loss. A criteria to determine whether a fault has occurred.

Max report delay (seconds): 16 bits
Maximum number of seconds within which a fault report must be sent after it is detected.

Min report delay (seconds): 16 bits
Minimum number of seconds a fault report needs to be sent after it is detected. A report should not be sent in less than this delay.

Max startup delay (seconds): 16 bits
Max number of seconds the TR can wait before the start of the test. The test is considered started if a test packet is received, or the "max startup delay" has passed after the receipt of this request.

Reception window (seconds): 16 bits
Number of seconds used for calculating packet loss percentage.

UDP port of data packets: 16 bits
UDP port test data packets use.

UDP port of status report packets: 16 bits
UDP port of status report packets.

Threshold description block: 0 bit
Variable length, depending on "Thres index". This revision only defines threshold index 0, with no threshold description block.

Test group address: 32 bits
The IP multicast address for the test group.

IP address of source 1 .. n: 32 bits
The IP address of the sources the targeted tester should monitor. When the address is 0.0.0.0, all sources to this group will be monitored.

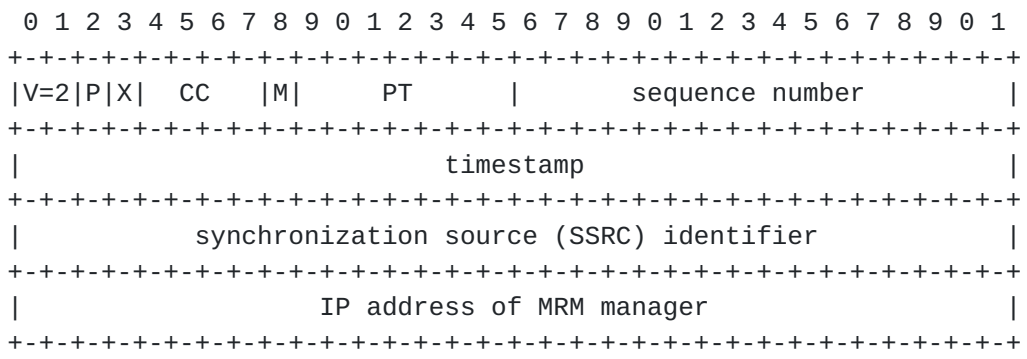
Inter-packet delay from source 1 .. n: 32 bits
Intervals between consecutive packets from the source (milliseconds).

4.5 Status Report to the MRM Manager

This MRM revision uses the reception report (RTCP) format based on Section 2.3.2. Future revisions will define MRM specific report formats.

4.6 MRM Test Packet

MRM test packets are RTPv2/UDP encapsulated. The RTPv2 packet header is replicated below for easy of description.



CC:
Set to 0 when sent, ignored when received.

M:
Set to 0 when sent, ignored when received.

PT:
Set to 0 when sent, ignored when received.

Sequence number:
Sequence number. Set to 0, when a tester is activated.

Timestamp:
System timestamp, in milliseconds.

SSRC:
IP address of the tester, or a configured 32-bit number that uniquely identifies the tester.

4.7 MRM Request-Ack Messages

The Acknowledgement messages for the Test Sender Request and the Status Request provide guarantees that the requests are indeed received by the testers, instead of being lost. The acknowledgement packets contain the MRM header and trailer for the respective messages, except that the message length and authentication data fields are recalculated.

5. Authenticating MRM Messages

All MRM messages should be authenticated with the MD5 mechanism specified here. The fields in the messages are transmitted in the clear. Packets that fail the authentication check are discarded by the receivers.

5.1 Generating Authenticated Messages

The sender of the MRM message decides which authentication key is used.

(1) The MRM message length field is filled with the number of bytes in the message;

(2) The rest of the message is composed;

(3) The IPSEC AH is constructed;

(4) The "authentication data" field is zeroed;

(5) The MRM authentication Key (16 byte long) is appended to the MRM message.

(6) The pad for the key is added. The digest is calculated and written into the "authentication data" field.

The part with the MD5 secret is not transmitted.

5.2 Receiving Authenticated Messages

The receiver follows the following steps when processing an incoming message:

(1) The digest is stored away and the "authentication data" field zeroed;

(2) It finds the key according to the value of "Key ID", and the key is appended and the packet properly padded;

(3) A new digest is calculated.

A message is discarded if the new digest is different from the one carried in the packet.

5.3 Key Management

We expect to rely on manual key distribution in the initial stages. And MRM should be able to utilize the standard secure key management mechanism when it becomes available.

6. Security Considerations

The strength of the security mechanism here depends on the strength of the key and the MD5 algorithm.

Insufficiently protected TSs and TRs (e.g. by weak keys) can be subject to attacks that can cause the TSs and TRs to take actions causing harm to the network.

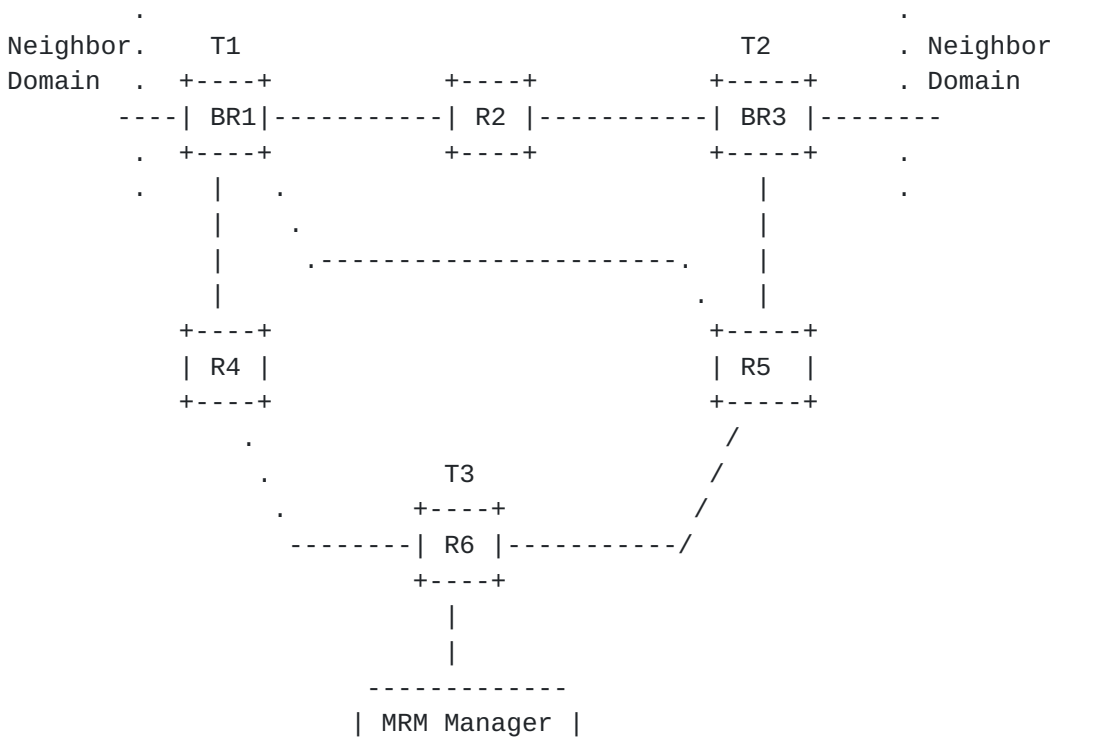
7. Different Approaches to Implement MRM

MRM is originally targeted at two types of users: network operation centers that provide production quality services; and network administrators who oversee semi-production or experimental multicast services. The former often rely on SNMP-based tools for management tasks and typically desire all types of monitoring functionalities to be wrapped into the same set of tools. While the later, who usually set the stage for production quality offerings, do not normally rely on SNMP-based tools and favor task-oriented tools.

For this reason, this document specifies the native MRM messages and operations. A companion document will define the MRM MIB that can accomplish the majority of the native MRM tasks.

8. Example of an MRM Setup

The example shown in this section is for illustration purpose only, and does not cover all possible functionalities of the MRM framework.



The above is a simple topology used to demonstrate the use of various MRM features. Border routers BR1, BR3 and an internal router R6 are administratively configured as candidate MRM Testers. The MRM manager configures T1 to be a TS, and T2,T3 to be TRs. The following are the messages sent by the MRM components.

(1) MRM manager sends Test Sender request (TSR) to T1.
Req1 = {Local packet trigger,
test packet interval = 60,000 (ms),
RTP/UDP test packet = TRUE,
Test group = 239.255.255.2}

T1 acknowledges receipt of Req1.

(2) MRM manager sends TR request Req2 to T2. Req2 has the following content:

```
J-bit                          = TRUE,
list of source addresses       = {T1's IP address},
threshold for fault detection = {20% loss over 10 minutes},
max delay for fault report     = 10 seconds,
min delay for fault report     = 0 seconds,
Test group                     = 239.255.255.2,
```

T2 acknowledges receipt of Req2. Req2 is retransmitted if the retransmission timer expires.

(3) MRM manager sends TR request Req3 to T3. Similar to Req2, except the target is T3, and,

```
max delay for fault report    = 20 seconds,
min delay for fault report    = 10 seconds
```

By using different (min, max) report times, it can avoid report implosion at the MRM manager, when a fault is detected by T2 and T3 at the same time.

(4) MRM manager periodically sends beacon messages, carrying Req1 and Req2, Req3. The holdtime is set to the remaining lifetime of the original request.

Assume T1 has a fault such that it can only forward 1% of all multicast packets, the fault is detected by T2 and T3. T2 randomly delays between 0-10 seconds, and sends a fault report to the MRM manager. The MRM manager acknowledges this report. T3 randomly delays between 10-20 seconds, and sends its fault report to the MRM manager, which is also acknowledged. This concludes the fault detection phase.

In the fault isolation phase, assume the MRM manager sends a third party mtrace request to T2 or T3, and isolates the fault to between T1, R2 and T1, R4. The MRM manager can then issue an an alarm to the network operator, with proper descriptions of the problem.

The operation for fault isolation phase might be more complicated for other types of fault, e.g. if T1 has lost the ability to forward multicast packets completely, T2 and T3 wouldn't have any multicast routing state or statistics for mtrace to work, some other mechanisms would have to be put in use.

9. Acknowledgment

We'd like to thank John Meylor, Beau Williamson, Stephen Deering, Ishan Wu, Louis Mamakos, Manoj Leelanivas, David Meyer, Bill Fenner and Dave Thaler for their comments and suggestions. We'd like to especially TY Lin and Kamil Sarac for filling in missing details from the previous version of the specification.

10. Authors addresses

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

Liming Wei
Siara Systems, Inc.
300 Ferguson Drive
Mountain View, California 94043
lwei@siara.com

Dino Farinacci
cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
dino@cisco.com

11. References

[mtrace] Steven Casner, Bill Fenner et al. The mtrace tool.

[mrm-use] Kevin Almeroth, Liming Wei, "Justification and Use of MRM", draft, Jan 15, 1999.

[aboba] Bernard Aboba, "The Use of SNTP as a Multicast Heartbeat", Draft, draft-ietf-mboned-sntp-heart-02.txt.

[ping] Jon Postel, "Internet Control Message Protocol", RFC792, Information Sciences Institute, 1981.

[UDP] Jon Postel, "User Datagram Protocol", RFC768. Information Sciences Institute.

[scope] Dave Meyer, "Administratively Scoped IP Multicast", Draft, draft-ietf-mboned-admin-ip-space-03.txt.

[MD5] R. Rivest, "The MD5 Message-Digest Algorithm", RFC1321, April, 1992

[KA98] Kent Stephen, Randall Atkinson, "IP Authentication Header", "draft-ietf-ipsec-auth-header-07.txt", July 1998

Appendix A - Change History

October 1999 -- revisions since draft-ietf-mboned-mrm-00.txt

(1) Added a TS length field to allow test send packets to be specified between 16 bytes and 2048 bytes in 16 byte increments.

(2) Made usage of beacon messages by the manager optional. Test agents are required to be able to process beacon messages.

(3) Monitoring existing groups is relegated to a later version because of the difficulty in monitoring the source to determine if it is sending a packet. When an MRM Test Source is used, Test Receivers know when, how many, and for how long packets will be sent. If no packets are received the test receiver knows to report 100% loss. This assumption is not possible when monitoring existing groups.

(4) Added additional detail about packet formats and packet handling procedures to reduce ambiguity.