

Network working group
Internet Draft
Intended status: Informational

A. Tempia Bonda
G. Picciano
Telecom Italia
M. Chen
L. Zheng
Huawei Technologies Co., Ltd
October 25, 2010

Expires: April 25, 2011

Requirements for IP multicast performance monitoring
draft-bipi-mboned-ip-multicast-pm-requirement-02.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 25, 2011.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This document describes the requirement for an IP multicast performance monitoring system for service provider IP multicast networks. This system enables efficient performance monitoring in Service Providers' production networks and provides diagnostic information in case of performance degradation or failure.

Table of Contents

1. Introduction.....	2
2. Conventions used in this document.....	4
3. Terminologies.....	4
4. Functional Requirements.....	6
4.1. Topology discovery and monitoring.....	6
4.2. Performance measurement.....	6
4.2.1. Loss rate.....	6
4.2.2. One-way delay.....	7
4.2.3. Jitter.....	7
4.2.4. Throughput.....	7
4.3. Measurement session management.....	8
4.3.1. Segment v.s. Path.....	8
4.3.2. Static v.s. Dynamic configuration.....	8
4.3.3. Proactive v.s. on-demand.....	8
4.4. Measurement result report.....	9
4.4.1. Performance reports.....	9
4.4.2. Exceptional alarms.....	9
5. Design considerations.....	10
5.1. Inline data-plane measurement.....	10
5.2. Scalability.....	10
5.3. Robustness.....	11
5.4. Security.....	11
5.5. Device flexibility.....	11
5.6. Extensibility.....	12
6. Security Considerations.....	12
7. IANA Considerations.....	12
8. References.....	12
8.1. Normative References.....	12
8.2. Informative References.....	12
9. Acknowledgments.....	13

1. Introduction

Service providers (SPs) have been leveraging IP multicast to provide revenue-generating services, such as IP television (IPTV), video conferencing, as well as the distribution of stock quotes or news.

These services are usually loss-sensitive or delay-sensitive, and their data packets need to be delivered over a large scale IP network in real-time. Meanwhile, these services demand relatively strict service-level agreements (SLAs). For example, loss rate over 5% is generally considered unacceptable for IPTV delivery. Video conferencing normally demands delays no more than 150 milliseconds. However, the real-time nature of the traffic and the deployment scale of service make it very challenging for IP multicast performance monitoring in a SP's production network. With increasing deployment of multicast service in SP networks, it becomes mandatory to develop an efficient system that is designed for SPs to accommodate the following functions.

- o SLA monitoring and verification: verify whether the performance of a production multicast network meets SLA requirements.
- o Network optimization: identify bottlenecks when the performance metrics do not meet the SLA requirements.
- o Fault localization: pin-point impaired components in case of performance degradation and service disruption.

These functions alleviate the OAM cost of IP multicast network for SPs, and ensure the quality of services.

However, the existing IP multicast monitoring tools and systems, which were mostly designed either for primitive connectivity diagnosis or for experimental evaluations, do not suit an SP production network, given the following facts:

- o Most of them provide end-to-end reachability check only [2][4][6]. They cannot provide sophisticated measurement metrics such as packet loss, one-way delay, and jitter, for the purpose of SLA verification.
- o Most of them can perform end-to-end measurements only. For example, RTCP-based monitoring system [5] can report end-to-end packet loss rate and jitter. End-to-end measurements are usually inadequate for fault localization, which needs finer grain measurement data to pin-point exact root causes.
- o Most of them use probing packets to probe network performance [2][4]. The approach might yield biased or even irrelevant results because the probing results are sampled and the out-of-band probing packets might be forwarded differently from the monitored user traffic.

- o Most of them are not scalable in a large deployment like an SPs' production network. For example, in IPTV deployment, the number of group members might be in the order of thousands. In this scale, an RTCP-based multicast monitoring system [5] becomes almost unusable because RTCP report intervals of each receiver might be delayed up to minutes or even hours because of over-crowded reporting multicast channel [12].
- o Some of them rely on the information from external protocols, which make their capabilities and deployment scenarios limited by the external protocols. The examples are passive measurement tools that collect and analyze messages from protocols such as multicast routing protocols [7], IGMP [9], or RTCP [5], etc. Another example is a SNMP-based system [8] that collects and analyzes relevant multicast MIB information.

This document describes the requirement for an IP multicast performance monitoring system for service provider (SP) IP multicast networks. This system should enable efficient monitoring of performance metrics of any given multicast channel (*,G) or (S,G) and provides diagnostic information in case of performance degradation or failure, which help SPs to do SLA verification, network optimization, and fault localizations in a large production network.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [1].

3. Terminologies

- o SSM (source specific multicast): When a multicast group is operating in SSM mode, only one designated node is eligible to send traffic through the multicast channel. An SSM multicast group with the designated source address *s* and group address *G* is denoted by (*s*, *G*).
- o ASM (any source multicast): When a multicast group is operating in ASM mode, any node can multicast packets through the multicast channel to other group members. An ASM multicast group with group address *G* is denoted by (*, *G*).
- o Root (of a multicast group): In an SSM multicast group (*s*, *G*), the root of this group is the first-hop router next to the source node *s*. In an ASM multicast group (*, *G*), the root of this group is the selected rendezvous point router.

- o Receiver: The term receiver refers to any node in the multicast group that should receive multicast traffic.
- o Internal forwarding path: Given a multicast group and a forwarding node in the group, the internal forwarding path inside the node refers to the data path between the upstream interface towards the root and one of the downstream interfaces toward a receiver.
- o Multicast forwarding path: Given a multicast group, a multicast forwarding path refers to the sequence of the interfaces, links and internal forwarding paths from the downstream interface at the root until the upstream interface at a receiver.
- o Multicast forwarding tree: Given a multicast group G, the union of all multicast forwarding paths composes the multicast forwarding tree.
- o Segment (of multicast forwarding path): The segment of a multicast forwarding path refers to part of the path between any two given interfaces.
- o Measurement session: A measurement session refers to the period of time in which certain performance metrics over a segment of multicast forwarding path is monitored and measured.
- o Monitoring node: A monitoring node is a node on a multicast forwarding path that is capable of performing traffic performance measurements on its interfaces.
- o Active interface: An interface of a monitoring node that is turned on to start a measurement session is said to be active.
- o Measurement session control packets: The packets are used for dynamic configuration for active interface to coordinate measurement sessions.

Figure 1 shows a multicast forwarding tree rooted at a root's interface A. Within router 1, B-C and B-D are two internal forwarding paths. Path A-B-C-E-G-I is a multicast forwarding path, which starts at root's downstream interface A and ends at receiver 2's upstream interface I. A-B, B-C-E are two segments of this forwarding path. When a measurement session for a metric such as loss rate is turned on over segment A-B, interfaces A and B are active interfaces.

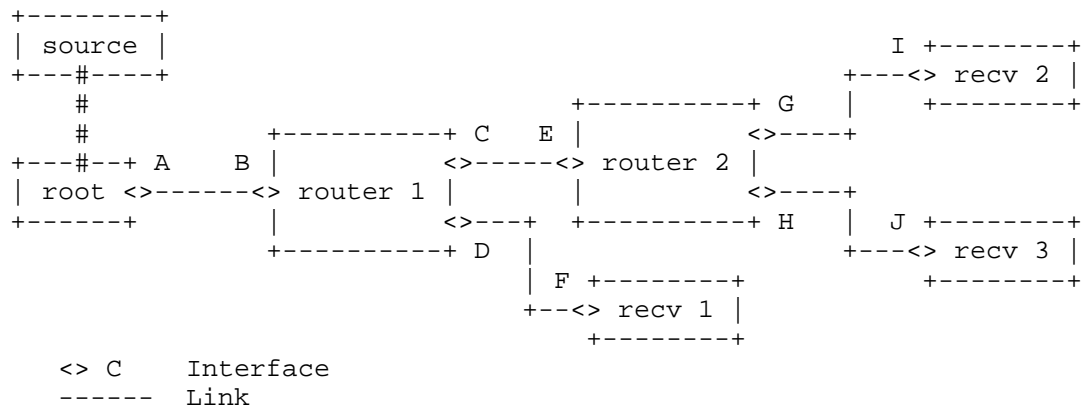


Figure 1. Example of multicast forwarding tree

4. Functional Requirements

4.1. Topology discovery and monitoring

The monitor system SHOULD have mechanisms to collect topology information of the multicast forwarding trees for any given multicast group. The function can be an integrated part of this monitoring system. Alternatively, the function might rely on other tools and protocols, such as mtrace [3], MANTRA[7], etc. The topology information will be referenced by network operators to decide where to enable measurement sessions.

4.2. Performance measurement

The performance metrics that a monitoring node needs to collect include, but are not limit to, the following.

4.2.1. Loss rate

Loss rate over a segment is the ratio of user packets not delivered to the total number of user packets delivered over this segment during a given interval. The number of user packets not delivered over a segment is the difference between the number of packets transmitted at the starting interface of the segment and received at the ending interface of this segment. Loss rate is crucial for multimedia streaming, such as IPTV, video/audio conferencing.

Loss rate over any segment of a multicast forwarding path MUST be provided. The measurement interval MUST be configurable.

4.2.2. One-way delay

One-way delay over a segment is the average time that user packets take to traverse this segment of forwarding path during a given interval. The time that a user packet traversing a segment is the difference between the time when the user packet leaves the starting interface of this segment and the time when the same user packet arrives at the ending interface of this segment. The one-way delay metric is essential for real-time interactive applications, such as video/audio conferencing, multiplayer gaming.

One-way delay over any segment of a multicast forwarding path SHOULD be able to be measured. The measurement interval MUST be configurable.

To get accurate one-way delay measurement results, the two end monitoring nodes of the investigated segments might need to have clock synchronized.

4.2.3. Jitter

Jitter over a segment is the variance of one-way delay over this segment during a given interval. The metric is of great importance for real-time streaming and interactive applications, such as IPTV, audio/video conferencing.

One-way delay jitter over any segment of a multicast forwarding path SHOULD be able to be measured. The measurement interval MUST be configurable.

Same as One-way delay measurement, to get accurate jitter, the clock frequencies at the two end monitoring nodes might need to be synchronized so that the clocks at two systems will proceed at the same pace.

4.2.4. Throughput

Throughput of multicast traffic for a group over a segment is the average number of bytes of user packets of this multicast group transmitted over this segment in unit time during a given interval. The information might be useful for resource management.

Throughput of multicast traffic over any segment of a multicast forwarding path MAY be measured. The measurement interval MUST be configurable.

4.3. Measurement session management

A measurement session refers to the period of time in which measurement for certain performance metrics is enabled over a segment of multicast forwarding path or over a complete multicast forwarding path. During a measurement session, the two end interfaces are said active. When an interface is activated, the interfaces start collecting statistics, such as number or timestamps of user packets which belongs to the given multicast group and pass through the interface. When both interfaces are activated, the measurement session starts. During a measurement session, data from two active interfaces are periodically correlated and the performance metrics, such as loss rate or delay, are derived. The correlation can be done either on the downstream interface if the upstream interface passes its data to it or on a third-party if the raw data on two active interfaces are reported to it. When one of the two interfaces is deactivated, the measurement session stops.

4.3.1. Segment v.s. Path

Network operators SHOULD be able to turn on or off measurements sessions for specific performance metrics over either a segment of multicast forwarding path or over a complete multicast forwarding path at any time. For example in Figure 1, network operator can turn on the measurement session of loss rate over path A-B-D-F and segment A-B-C as well as jitter over segment C-E-G-I simultaneously. This feature allows network operators to zoom into the suspicious components when degradation or failure occurs.

4.3.2. Static v.s. Dynamic configuration

A measurement session can be configured statically. In this case, network operators activate the two interfaces or configure their parameter settings on the relevant nodes either manually or automatically through agents of network management system (NMS).

Optionally, a measurement session can be configured dynamically. In this case, an interface may coordinate another interface on its forwarding path to start or stop a session. Accordingly, the format and process routines of the measurement session control packets need to be specified. The delivery of such packets SHOULD be reliable and it MUST be possible to secure the delivery of such packets.

4.3.3. Proactive v.s. on-demand

A measurement session can be started either proactively or on demand. Proactive monitoring is either configured to be carried out

periodically and continuously or preconfigured to act on certain events such as alarm signals. To save resources, operators may turn on measurement sessions proactively for critical performance metrics over the backbone segments of multicast forwarding tree only. This keeps the overall monitoring overhead minimal during normal network operations.

In contrast to proactive monitoring, on-demand monitoring is initiated manually and for a limited amount of time to carry out diagnostics. When network performance degradation or service disruption occurs, operators might turn on measurement sessions on-demand over the interested segments to facilitate fault localization.

4.4. Measurement result report

The measurement results might be present in two forms: reports or alarms.

4.4.1. Performance reports

Performance reports contain streams of measurement data over a period of time. A data collection agent MAY actively poll the monitoring nodes and collect the measurement reports from all active interfaces. Alternatively, the monitoring nodes might be configured to upload the reports to the specific data collection agents once the data become available. To save bandwidth, the content of the reports might be aggregated and compressed. The period of reporting SHOULD be able to be configured or controlled by rate limitation mechanisms (e.g., exponentially increasing).

4.4.2. Exceptional alarms

On the other hand, the active interfaces of a monitoring node or a third-party MAY be configured to raise alarms when exceptional events such as performance degradation or service disruption occur. Alarm thresholds and the management should be specified for each of the performance metric when the measurement session is configured on this interface. During measurement session, once the value of certain performance metric exceeds the threshold, alarm will be raised and reported to the configured nodes. To prevent huge volume of alarms from overloading the management nodes and network congestion, alarm suppression and aggregation mechanisms SHOULD be employed on the interfaces to limit the rate of alarm report and the volume of data.

5. Design considerations

To make the monitoring system feasible and optimal for a SP production network, the following considerations should take into account when design the system.

5.1. Inline data-plane measurement

Measurement results collected by probing packets might be biased or even totally irrelevant given the facts that (1) probing packets collect sampled results only and might not capture the real statistic characteristics of the monitored user traffic. Experiments have demonstrated that the measurement sampled by the probing packets, such as ping probes, might be incorrect if sampling interval is too long [10]; (2) probing packets introduce extra load onto the network. In order to improve accuracy, sampling frequency has to be high enough, which in turn increase network overhead and further bias the measurement results; (3) probing packets are usually not in the same multicast group as user packets and might take different forwarding path given that equal cost multi-path routing (ECMP) and link aggregation (LAG) have been widely adopted in SP network. An out-of-band probing packet might take a path totally different from the user packets of the multicast group that it is monitoring. Even if the forwarding path is the same, the intermediate node might apply different queuing and scheduling strategy for the probing packets. As a result, the measured results might be irrelevant.

The performance measurement should be "inline" in the sense that the measurement statistics are derived directly from user packets, instead of probing packets. At the same time, unlike offline packet analysis, the measurement is counting user packets at line-speed in real-time without any packet duplication or buffering.

To accomplish the inline measurement, some extra packets might need to be injected into user traffic to coordinate measurement across nodes. The volume of these packets SHOULD be keep minimal such that the injection of such packets will not impact measurement accuracy.

5.2. Scalability

The measurement methodology and system architecture MUST be scalable. A multicast network for an SP production network usually comprises of thousands of nodes. Given the scale, the collecting, processing and reporting overhead of performance measurement data SHOULD NOT overwhelm either monitoring nodes or management nodes. The volume of reporting traffic should be reasonable and not cause any network congestion.

5.3. Robustness

The measurements MUST be independent of the failure of the underlying multicast network. For example, the monitor SHOULD generate correct measurement result even if some measurement coordinating packets are lost; invalid performance reports should be able to be identified in case that the underlying multicast network is undergoing drastic changes.

If dynamic configuration is supported, the delivery of measurement session control packets SHOULD be reliable so that the measurement sessions can be started, ended and performed in a predictable manner. Meanwhile, the control packets SHOULD not be delivered based on the multicast routing decision. This multicast independent characteristic guarantees that the active interfaces are still under control even if the multicast service is malfunctioning.

Similarly, if an NMS is used to control the monitoring nodes remotely, the communication between monitoring nodes and the NMS SHOULD be reliable.

5.4. Security

The monitoring system MUST not impose security risks on the network. For example, the monitoring nodes should be prevented from being exploited by third parties to control measurement sessions arbitrarily, which might make the nodes vulnerable for DDoS attacks.

If dynamic configuration is supported, the measurement session control packets need to be encrypted and authenticated.

5.5. Device flexibility

Both the software and hardware deployment requirement for the monitoring system SHOULD be reasonable. For example, one-way delay measurement needs clock synchronization across nodes. To require the installation of expensive hardware clock synchronization devices on all monitoring nodes might be too costly to make the monitoring system infeasible for large deployment.

The monitor system SHOULD be incrementally deployable, which means that the system can enable monitoring functionality even if some of the nodes in the network are not equipped with the required software and hardware or does not meet the software and hardware deployment requirements.

The non-monitoring nodes without the monitoring capabilities SHOULD be able to coexist with monitoring nodes and function. The packets exchanged between monitoring nodes SHOULD be transparent to other nodes and MUST not cause any malfunction of the non-monitoring nodes.

5.6. Extensibility

The system should be easy to be extended for new functionalities. For example, the system should be easily extended to collect newly defined performance metrics.

6. Security Considerations

The security issues have been taken into account in design considerations (see Section 5.4).

7. IANA Considerations

There is no IANA action required by this draft.

8. References

8.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

- [2] Venaas, S., "Multicast Ping Protocol", draft-ietf-mboned-ssmping-07, December 2008.
- [3] Asaeda, H., Jinmei, T., Fenner, W., and S. Casner, "Mtrace Version 2: Traceroute Facility for IP Multicast", draft-ietf-mboned-mtrace-v2-03, March 2009.
- [4] Almeroth, K., Wei, L., and D. Farinacci, "Multicast Reachability Monitor (MRM)", draft-ietf-mboned-mrm-01, July 2000.
- [5] Bacher, D., Swan, A., and L. Rowe, "rtpmon: a third-party RTCP monitor", Conference 4th ACM International conference on multimediu, 1997.
- [6] Sarac, K. and K. Almeroth, "Application Layer Reachability Monitoring for IP Multicast", Journal Computer Networks Journal, Vol.48, No.2, pp.195-213, June 2005.

- [7] Rajvaidya, P., Almeroth, K., and k. claffy, "A Scalable Architecture for Monitoring and Visualizing Multicast Statistics", Conference IFIP/IEEE Workshop on Distributed Systems: Operations & Management (DSOM), Austin, Texas, USA, December 2000.
- [8] Sharma, P., Perry, E., and R. Malpani, "IP Multicast Operational Network Management: Design, Challenges and Experiences", Journal IEEE Network, Volume 17, Issue 2, Mar/Apr 2003 Page(s): 49 - 55, Mar/Apr 2003.
- [9] Al-Shaer, E. and Y. Tang, "MRMON: Remote Multicast Monitoring", Conference NOMS, 2004.
- [10] Sarac, K. and K. Almeroth, "Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring", Journal Journal of High Speed Networks, Vol.9, No.3-4, pp.191-211, 2000.
- [11] Sarac, K. and K. Almeroth, "Monitoring IP Multicast in the Internet: Recent Advances and Ongoing Challenges", Journal IEEE Communication Magazine, 2005.
- [12] Vit Novotny, Dan Komosny, "Optimization of Large-Scale RTCP Feedback Reporting in Fixed and Mobile Networks," icwmc, pp.85, Third International Conference on Wireless and Mobile Communications (ICWMC'07), 2007

9. Acknowledgments

The authors would like to thank Wei Cao, Xinchun Guo, and Hui Liu for their helpful comments and discussions.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Alberto Tempia Bonda
Telecom Italia
Via Reiss Romoli, 274
Torino 10148
Italy

Email: alberto.tempiabonda@telecomitalia.it

Giovanni Picciano
Telecom Italia
Via Di Val Cannuta 250
Roma 00166
Italy

Email: giovanni.picciano@telecomitalia.it

Mach(Guoyi) Chen
Huawei Technologies Co. Ltd.
Huawei Building, No.3 Xinx Road,
Hai-Dian District,
Beijing, 100085
China

EMail: mach@huawei.com

Lianshu Zheng
Huawei Technology Co. Ltd.
Huawei Building, No.3 Xinx Road,
Hai-Dian District,
Beijing, 100085
China

Email: verozheng@huawei.com

mboned
Internet-Draft
Intended status: Informational
Expires: February 25, 2011

T. Hayashi,
H. Satou,
H. Ohta
NTT
H.He
Nortel
S. Vaidya
Cisco Systems, Inc.
August 24, 2010

Requirements for Multicast AAA coordinated between Content Provider(s)
and Network Service Provider(s)
draft-ietf-mboned-macnt-req-10

Abstract

This memo presents requirements in the area of accounting and access control for IP multicasting. The scope of the requirements is limited to cases where Authentication, Accounting and Authorization (AAA) functions are coordinated between Content Provider(s) and Network Service Provider(s).

In order to describe the new requirements of a multi-entity Content Deliver System(CDS) using multicast, the memo presents three basic business models: 1) the Content Provider and the Network Provider are the same entity, 2) the Content Provider(s) and the Network Provider(s) are separate entities and users are not directly billed, and 3) the Content Provider(s) and the Network Provider(s) are separate entities and users are billed based on content consumption or subscriptions. The requirements of these three models are listed and evaluated as to which aspects are already supported by existing technologies and which aspects are not.

General requirements for accounting and admission control capabilities including quality-of-service (QoS) related issues are listed and the constituent logical functional components are presented.

This memo assumes that the capabilities can be realized by integrating AAA functionalities with a multicast CDS system, with IGMP/MLD at the edge of the network.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on February 25, 2011.

1. Introduction

Broadband access networks such as ADSL (Asymmetric Digital Subscriber Line) or FTTH (Fiber to the Home) have been deployed widely in recent years. Content Delivery Service (CDS) is expected to be a major application provided through broadband access networks. Because many services such as television broadcasting require huge bandwidth (e.g., 6Mbit/s) and processing power at the content server(s), IP multicast is used as an efficient delivery mechanism for CDS.

A single entity may design and be responsible for a system that covers the various common high-level requirements of a multicasting CDS such as 1) content serving, 2) the infrastructure to multicast it, 3) network and content access control mechanisms. For cases in which the business model includes the direct billing of users, the single provider of both content and network services has sufficient data in its control to bill users based on their content consumption. Furthermore it is possible to tie access to the network and QoS based on a user's contract status. Therefore current technologies support the single entity case.

Often, however, the content provision and network provision roles are

split between separate entities. Commonly, Content Providers (CP) do not build and maintain their own multicast network infrastructure as this is not their primary business area. Instead, CPs often purchase transport and management services from network service providers. This memo lists the requirements of a business model in which the NSP provides CDS using multicast as one such contractible service.

The direct revenue source for the multiple entity provider is a defining aspect of the business model which often has implications on requirements for the technologies that support the system. There are cases such as the the advertising-based model where billing end-users is not done and therefore accounting of content consumption can be anonymous and/or in aggregate. In these cases the requirements of the business model for accounting for billing purposes are already supported by existing technologies. However, the NSP can not guarantee high quality transmission on a per-content basis with existing technologies.

There is also the business model in which the individual user of multicasted contents is the source of revenue for both consumed content and network resources. In this model the NSP wants to receive the appropriate fees for multicast services and the NSP undertakes collecting bills as a proxy for the CPs. The NSP may provide high quality service by admission control. Current standards do not fully support this model and this memo will list the requirements which need to be supported.

2. Definitions and Abbreviations

2.1. Definitions

Authentication: action for identifying a user as a genuine one.

Authorization: action for giving permission for a user to access content or the network.

Eligible user: Users may be eligible (permitted) to access resources because of the attributes they have (e.g., delivery may require possession of the correct password or digital certificate), their equipment has (e.g., content may only be eligible to players that can decode H.264 or 3GPP streams), their access network has (e.g., HDTV content may only be eligible to users with 10 Mbps or faster access line), or because of where they are in network topology (e.g., HDTV content may not be eligible for users across congested links) or in actual geography (e.g., content may only be licensed for distribution to certain countries), and, of course, a mix of attributes may be required

for eligibility or ineligibility.

User: In this document user refers to a requester and a recipient of multicast data, termed a viewer in CDS.

User-based accounting: actions for grasping each user's behavior, when she/he starts/stops to receive a channel, which channel she/he receives, etc.

2.2. Abbreviations

AAA: Authentication, Accounting and Authorization

ASM: Any-Source Multicast

CDS: Content Delivery Service

CP: Content Provider

IGMP: Internet Group Management Protocol

MLD: Multicast Listener Discovery

NSP: Network Service Provider

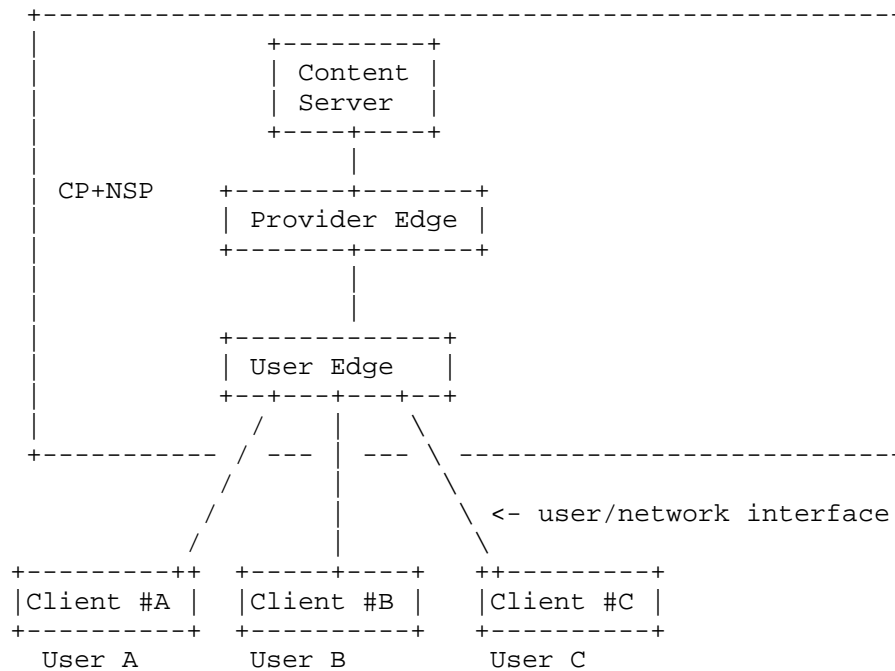
SSM: Source Specific Multicast

QoS: Quality of Service

3. Current Business Models

3.1. Single entity model where CP and NSP are the same entity

One existing business model is that of a single entity responsible for both content and network service provision which bills its users based on content provision. (See figure below.)



Example of CDS network configuration

Figure 1

In this model the network can query a content-policy-enabled AAA server within its own domain at the time a user requests content. The network can provide the AAA server with information such as user identity, device identity, the requested content (channel), geographic information, method of network connection, etc. that might be required for the content provision authorization decision. It is therefore possible to configure a network to deny network access based on the content policy decision.

In this model there are no issues of mapping user identities between different entity domains. The provider has access to the information on which user accessed from which point on what device. Furthermore as network provider they can record not only when a user joined or left a certain channel, but also if packets were actually delivered. Moreover, there are no inter-entity security and privacy concerns between the CP and NSP.

The single entity network service and content provider also knows the content schedules for various channels. This is important not only

for time and content-sensitive authorization decisions but also for providing meaningful billing details to end users.

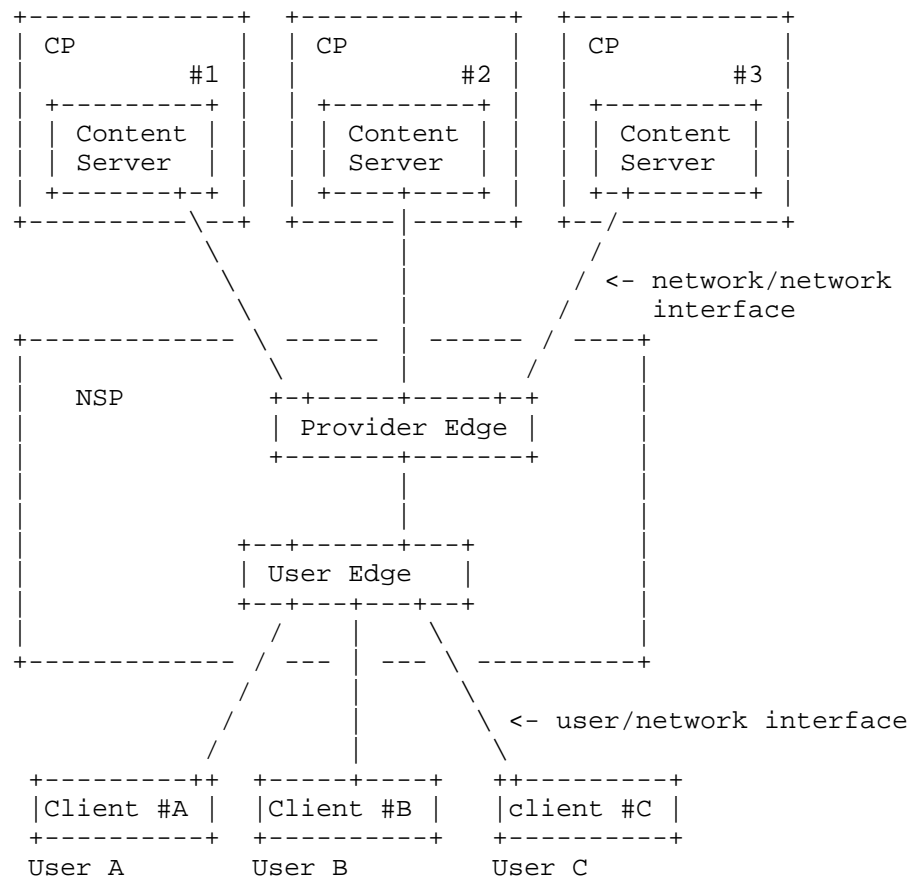
3.2. Multiple entity model without direct content-based billing

An additional model for delivering contents over a CDS is the advertising-based model where billing end-users is not done. In this model the four different roles may be filled by separate entities: Content Provider (CP), Network Service Provider (NSP), user clients, and advertising sponsors. In the general case of this business model, insofar as the advertiser does not require user-based metrics the accounting of content consumption can be anonymous and/or in aggregate and can be off-line from the multicast-with-AAA CDS system itself. Therefore this model does not require any new standards to provide user-based accounting for a multi-entity CDS using multicast with AAA. (Providing this data in near real-time and inline would entail further requirements which can be dealt with in a separate memo if necessary.)

A more complex version of this business model is conceivable in which a CP may require a user to enter into a subscription contract, even when the user does not get billed for content consumption. For example, a CP may value individual data because it allows it to supply the advertisers with rich, user-segmented data and charge a higher premium. In that case the requirements of the next section "CDS with direct billing of the end user" are generally applicable because of the need to link the user data which the CP has to the actual viewing (or stream downloading) data that the NSP has.

4. Proposed Model: Multity-entity CDS

In this model the networks for CDS contain three different types of entities: Content Provider (CP), Network Service Provider (NSP), and user clients. An NSP owns the network resources (infrastructure). It accommodates content providers on one side and accommodates user clients on the other side. NSP provides the network for CDS to two entities (i.e., CPs and user clients). A CP provides content to each user through the network of NSPs and charges users for content. NSPs are responsible for delivering the content to user clients, and for controlling the network resources. A NSP charges a user or a CP for network usage. A NSP may charge users for content as a proxy of the CP.



Example of CDS network configuration

Figure 2

The CP provides detailed channel information (e.g., Time table of each channel) to the information server which is either managed by the NSP or CP. An end-user client gets the information from the information server. In this model, multicasting is used in the NSP's CDS network, and there are two different contracts. One is the contract between the NSP and the user which permits the user to access the basic network resources of the NSP. Another contract is between the CP and user to permit the user to subscribe to multicast content. Because the CP and NSP are different entities, and the NSP generally does not allow a CP to control (operate) the network resources of the NSP, user authorization needs to be done by the CP and NSP independently. Since there is no direct connection to the

user/network interface, the CP cannot control the user/network interface. A user may want to move to another place, or may want to change her/his device (client) any time without interrupting her/his reception of services.

4.1. Information Required by Entities to Support the Proposed Business Model

User identification and Authentication:

The network should be able to identify and authenticate each user when they attempt to access the service requesting content. This user identification is required for:

- authorization for content consumption eligibility

- user tracking for billing based on actual content consumption and network resource usage

With current protocols (IGMP/MLD), the sender cannot distinguish which receivers (end hosts) are actually receiving the information. The sender must rely on the information from the multicasting routers. This can be complicated if the sender and routers are maintained by different entities. Furthermore, the current user associated with receiver must be identified.

User Authorization:

The network, at its option, should be able to authorize a user's access to content or a multicast group, so as to meet any demands by a CP to prevent content access by ineligible users.

Sharing Programming data:

NSP needs a mechanism to receive channel programming data from the CP in order to provide the information to the user at channel selection time and also for somehow logging or recording what programming content has been streamed to the user. In some cases the CP may contract the NSP to bill the user as a proxy for the CP. In this case there needs to be a mechanism for supplying the user-based viewing history with human-meaningful channel data to the end-user.

Content usage information by user:

For billing and auditing purposes the CP needs the NSP to provide it with detailed per-user usage behavior indicating what content was consumed from when to when. There needs to be a mechanism to

supply the user-based viewing history from the NSP to the CP. If the CP is selling on an on-demand model, or tiered subscription basis or supplies some sort of online account statement this history needs to be fed back to the CP in near real-time. To assemble such data on user behavior, it is necessary to precisely log information such as who (host/user) is accessing what content at what time (join action) until what time (leave action). The result of the access-control decision (e.g. results of authorization) would also be valuable information. The desired degree of logging precisions would depend on the application used.

Notification to Users of the Result of the Join Request:

It should be possible to provide information to the user about the status of his/her join request(granted/denied/other). Such information can be used to give meaningful feedback to the user.

5. Admission Control for Multicasting

In order to guarantee certain QoS it is important for network providers (at their option) to be able to protect their network resources from being wasted, (either maliciously or accidentally). The NSP should be able to apply appropriate access controlling actions based on user eligibility status:

The network should be able to apply necessary access controlling actions when an eligible user requests an action (such as a join or a leave.)

The network should be able to reject any action requested from an ineligible user.

In order to maintain a predefined QoS level, depending on the NSP's policy, a user edge should be able to control the number of streams it serves to a user, and total bandwidth consumed to that user. For example if the number of streams being served to a certain user has reached the limit defined by the NSP's policy, then the user edge should not accept a subsequent "join" until one of the existing streams is terminated. Similarly, if the NSP is controlling by per-user bandwidth consumption, then a subsequent "join" should not be accepted if delivery of the requested stream would push the consumed bandwidth over the NSP policy-defined limit.

The network may need to control the combined bandwidth for all channels at the physical port of the edge router or switch so that these given physical entities are not overflowed with traffic. This entails being able to control the number of channels delivered, the

bandwidth for each channel and the combined bandwidth for all channels.

6. Reauthorization/ deauthorization requirements

A mechanism for periodic reauthorization of users who have already joined a channel stream should be supported. The reauthorization could be an authorization check based on the NSP's eligibility requirements and/or could involve the NSP querying the CP for reauthorization of a user.

A mechanism for deauthorization should be supported for cases in which a user is deemed ineligible by the NSP and/or CP at the time of a reauthorization check. If a NSP revokes authorization for the network for a user it should force a leave, and record details of the leave (including the time and reason for the forced leave.) If a CP revokes authorization to content for a user the CP signals to the NSP to cease streaming to that user. An example usage case for deauthorizing a user is one where a user has a subscription or has paid for a certain amount of content and has reached that limit. In some models, it is conceivable that a CP could communicate the parameters for de-authorization to the NSP at the time of the original join's authorization so as to make NSP->CP reauthorization requests unnecessary.

7. Performance requirements

Channel Join Latency and Leave Latency

Commercial implementations of IP multicasting are likely to have strict requirements in terms of user experience. Join latency is the time between when a user sends a "join" request and when the requested data streaming first reaches the user. Leave latency is the time between when a user sends a "leave" signal and when the network stops streaming to the user. Leave and Join latencies impact the acceptable user experience for fast channel surfing. In an IP-TV application, users are not going to be receptive to a slow response time when changing channels. If there are policies for controlling the number of simultaneous streams a user may access then channel surfing will be determined by the join and leave latencies. Furthermore, leave affects resource consumption: with a low "leave latency" network providers could minimize streaming content when there are no audiences. It is important that any overhead for authentication, authorization, and access-control be minimized at the times of joining and leaving multicast channels so as to achieve join and leave latencies acceptable in terms of user experience. For

example this is important in an IP-TV application, because users are not going to be receptive to a slow response time when changing channels.

8. Concomitant requirements

Scalability

Solutions that are used for AAA and QoS enabled IP multicasting should scale enough to support the needs of content providers and network operators. NSP's multicast access and QoS policies should be manageable for large scale users. (e.g. millions of users, thousands of edge-routers)

Service and Terminal Portability:

Depending on the service, networks should allow for a user to receive a service from different places and/or with a different terminal device.

Deployable as Alternative to Unicast

IP Multicasting would ideally be available as an alternative to IP unicasting when the "on-demand" nature of unicasting is not required. Therefore interfaces to multicasting should allow for easy integration into CDS systems that support unicasting. Especially equivalent interfaces for authorization, access control and accounting capabilities should be provided.

Support of ASM and SSM

Both ASM (G), and SSM (S,G) should be supported as multicast models.

Support for Tunneled Multicast

The AAA requirements specified in this document should apply to both end-to-end native multicast and to tunnel-enabled multicast, such as AMT multicast: [I-D.ietf-mboned-auto-multicast]

Small Impact on the Existing Products

Impact on the existing products (e.g., protocols, software, etc.) should be as minimal as possible. Ideally the NSP should be able to use the same infrastructure (such as access control) to support commercial multicast services for the so called "triple play" services: voice (VoIP), video, and broadband Internet access services. When a CP requires the NSP to provide a level of QoS

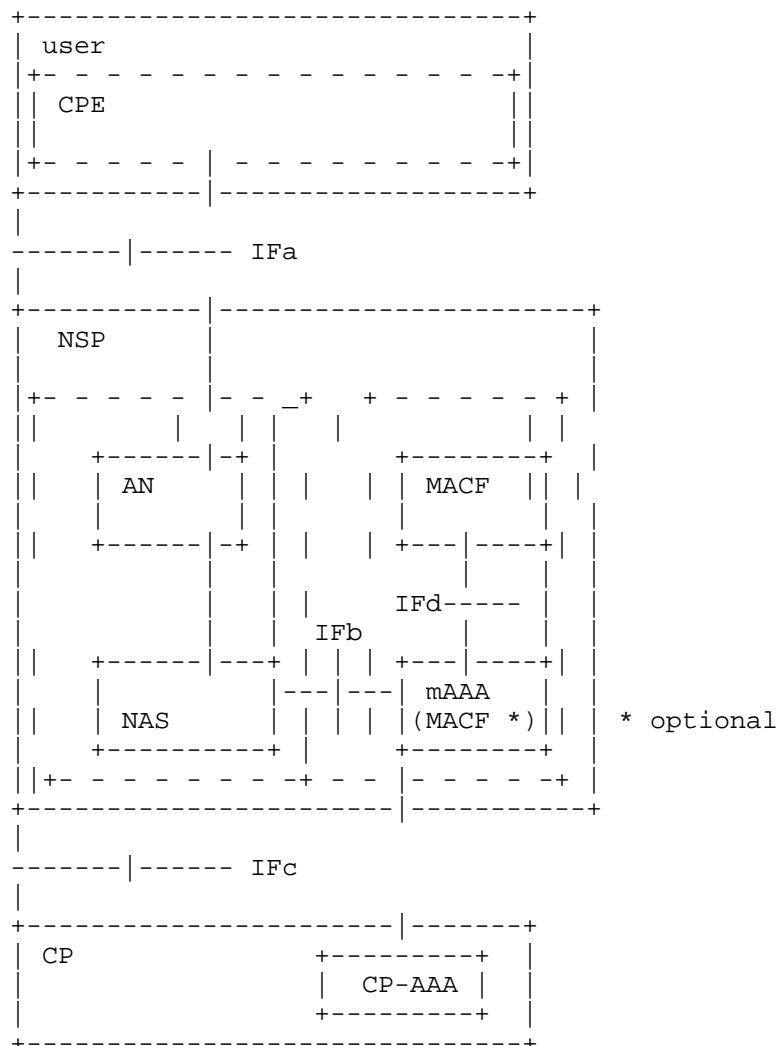
surpassing "best effort" delivery or to provide special services (e.g., to limited users with specific attributes), certain parameters of the CDS may be defined by a contractual relation between the NSP and the CP. However, just as for best-effort unicast, multicast allows for content sourced by CPs without a contractual relation with the NSP. Therefore, solutions addressing the requirements defined in this memo should not make obsolete multicasting that does not include AAA features. NSPs may offer tiered services, with higher QoS, accounting, authentication, etc., depending on contractual relation with the CPs. It is therefore important that Multicast AAA and QoS functions be as modular and flexible as possible.

Multicast Replication

The above requirements should also apply if multicast replication is being done on an access-node (e.g. DSLAMs or OLTs).

9. Constituent Logical Functional Components

Below is a diagram of a AAA enabled multicasting network, including the logical components within the various entities.



AAA enabled multicasting network with admission control

Figure 3

The user entity includes the CPE (Customer Premise Equipment) which connects the receiver (s).

The NSP (Network Service Provider) includes the transport system and a logical element for multicast AAA functionality. The TS (transport system) is comprised of the access node and NAS (Network Access Server) An AN (Access Node) may be connected directly to mAAA or a

NAS relays AAA information between an AN and a mAAA. Descriptions of AN and its interfaces are out of the scope for this memo. The multicast AAA function may be provided by a mAAA which may include the function that downloads Join access control lists to the NAS (this function is referred to as the conditional access policy control function.)

Interface between mAAA and NAS

The interface between mAAA and the NAS is labeled IFb in Figure 3. Over IFb the NAS sends an access request to the NSP-mAAA and the mAAA replies. The mAAA may push conditional access policy to the NAS.

CP-AAA

The content provider may have its own AAA server which has the authority over access policy for its contents.

Interface between user and NSP

The interface between the user and the NSP is labeled IFa in Figure 3. Over IFa the user makes a multicasting request to the NSP. The NSP may in return forward multicast traffic depending on the NSP and CP's policy decisions.

Interface between NSP and CP

The interface between the NSP and CP is labeled IFc. Over IFc the NSP requests to the CP-AAA for access to contents and the CP replies. CP may also send conditional access policy over this interface for AAA-proxying.

The NSP may also include a component that provides network resource management (e.g. QoS management), as described in section 5, "Admission Control for Multicasting". Resource management and admission control is provided by MACF (Multicast Admission Control Function). This means that, before replying to the user's multicast request, the mAAA queries the MACF for a network resource access decision over the interface IFd. The MACF is responsible for allocating network resources for forwarding multicast traffic. MACF also receives Leave information from NAS so that MACF releases corresponding reserved resources.

10. Acknowledgments

The authors of this draft would like to express their appreciation to Christian Jacquenet of France Telecom whose contributions to the "AAA

Framework for Multicasting" [draft-ietf-mboned-multiaaaa-framework] largely influenced this draft; Pekka Savola of Netcore Ltd.; Daniel Alvarez, and Toerless Eckert of Cisco Systems; Sam Sambasivan of AT&T; Sanjay Wadhwa, Greg Shepherd, and Leonard Giuliano of Juniper; Tom Anschutz and Steven Wright of BellSouth; Nicolai Leymann of T-Systems; Bill Atwood of Concordia University; Carlos Garcia Braschi of Telefonica Empresas; Mark Altom, Andy Huang, Tom Imburgia, Han Nguyen, Doug Nortz of ATT Labs; Marshall Eubanks in his role as mboned WG chair; Ron Bonica in his role as Director as the Operations and Management Area; Stephen Rife of Digital Garage and David Meyer in his former role as mboned WG chair as well as their thanks to the participants of the MBONED WG in general.

Funding for the RFC Editor function is currently provided by the Internet Society.

11. IANA Considerations

This memo does not raise any IANA consideration issues.

12. Security Considerations

Accounting capabilities can be used to enhance the security of multicast networks by excluding ineligible clients from the networks.

These requirements are not meant to address encryption issues. Any solution meeting these requirements should allow for the implementation of encryption such as MSEC on the multicast data.

13. Privacy considerations

Any solution which meets these requirements should weigh the benefits of user-based accounting with the privacy considerations of the user. For example solutions are encouraged when applicable to consider encryption of the content data between the content provider and the user in such a way that the Network Provider does not know the contents of the channel.

14. Conclusion

This memo describes general requirements for providing AAA and QoS enabled IP multicasting services in multi-entity models. A few models are evaluated with regard to their support by current technologies. The "multi-entity CDS with direct billing of the end

user" model is presented and requirements for information sharing between entities and requirements for admission control to enable guaranteeing of QoS are derived. Performance requirements and concomitant requirements are also presented.

15. References

15.1. Normative References

- [RFC2975] Aboba, B., Arkko, J., and D. Harrington, "Introduction to Accounting Management", RFC 2975, October 2000.
- [RFC3376] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

15.2. Informative References

- [I-D.ietf-mboned-auto-multicast]
Thaler, D., Talwar, M., Aggarwal, A., Vicisano, L., and T. Pusateri, "Automatic IP Multicast Without Explicit Tunnels (AMT)", draft-ietf-mboned-auto-multicast-09 (work in progress), June 2008.

Authors' Addresses

Tsunemasa Hayashi
Nippon Telegraph and Telephone Corporation
1-1 Hikarino'oka
Yokosuka-shi, Kanagawa 239-0847
Japan

Phone: +81 46 859 8790
Email: hayashi.tsunemasa@lab.ntt.co.jp

Hiroaki Satou
Nippon Telegraph and Telephone Corporation
3-9-11 Midoricho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 4683
Email: satou.hiroaki@lab.ntt.co.jp

Hiroshi Ohta
Nippon Telegraph and Telephone Corporation
3-9-11 Midoricho
Musashino-shi, Tokyo 180-8585
Japan

Phone: +81 422 59 3617
Email: ohta.hiroshi@lab.ntt.co.jp

Haixiang He
Nortel
600 Technology Park Drive
Billerica, MA 01801
USA

Phone: +1 978 288 7482
Email: haixiang@nortel.com

Susheela Vaidya
Cisco Systems, Inc.
170 W. Tasman Drive
San Jose, CA 95134
USA

Phone: +1 408 525 1952
Email: svaidya@cisco.com

Copyright and License Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

MBONED Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 1, 2019

H. Asaeda
NICT
K. Meyer

W. Lee, Ed.
July 31, 2018

Mtrace Version 2: Traceroute Facility for IP Multicast
draft-ietf-mboned-mtrace-v2-26

Abstract

This document describes the IP multicast traceroute facility, named Mtrace version 2 (Mtrace2). Unlike unicast traceroute, Mtrace2 requires special implementations on the part of routers. This specification describes the required functionality in multicast routers, as well as how an Mtrace2 client invokes a query and receives a reply.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 1, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Terminology	6
2.1. Definitions	6
3. Packet Formats	7
3.1. Mtrace2 TLV format	8
3.2. Defined TLVs	8
3.2.1. Mtrace2 Query	9
3.2.2. Mtrace2 Request	11
3.2.3. Mtrace2 Reply	11
3.2.4. IPv4 Mtrace2 Standard Response Block	12
3.2.5. IPv6 Mtrace2 Standard Response Block	16
3.2.6. Mtrace2 Augmented Response Block	19
3.2.7. Mtrace2 Extended Query Block	20
4. Router Behavior	21
4.1. Receiving Mtrace2 Query	21
4.1.1. Query Packet Verification	21
4.1.2. Query Normal Processing	22
4.2. Receiving Mtrace2 Request	22
4.2.1. Request Packet Verification	22
4.2.2. Request Normal Processing	23
4.3. Forwarding Mtrace2 Request	24
4.3.1. Destination Address	25
4.3.2. Source Address	25
4.3.3. Appending Standard Response Block	25
4.4. Sending Mtrace2 Reply	26
4.4.1. Destination Address	26
4.4.2. Source Address	26
4.4.3. Appending Standard Response Block	26
4.5. Proxying Mtrace2 Query	26
4.6. Hiding Information	27

5.	Client Behavior	27
5.1.	Sending Mtrace2 Query	27
5.1.1.	Destination Address	28
5.1.2.	Source Address	28
5.2.	Determining the Path	28
5.3.	Collecting Statistics	28
5.4.	Last Hop Router (LHR)	28
5.5.	First Hop Router (FHR)	29
5.6.	Broken Intermediate Router	29
5.7.	Non-Supported Router	29
5.8.	Mtrace2 Termination	29
5.8.1.	Arriving at Source	29
5.8.2.	Fatal Error	30
5.8.3.	No Upstream Router	30
5.8.4.	Reply Timeout	30
5.9.	Continuing after an Error	30
6.	Protocol-Specific Considerations	31
6.1.	PIM-SM	31
6.2.	Bi-Directional PIM	31
6.3.	PIM-DM	31
6.4.	IGMP/MLD Proxy	32
7.	Problem Diagnosis	32
7.1.	Forwarding Inconsistencies	32
7.2.	TTL or Hop Limit Problems	32
7.3.	Packet Loss	32
7.4.	Link Utilization	33
7.5.	Time Delay	33
8.	IANA Considerations	33
8.1.	"Mtrace2 Forwarding Codes" Registry	33
8.2.	"Mtrace2 TLV Types" Registry	34
8.3.	UDP Destination Port	34
9.	Security Considerations	34
9.1.	Addresses in Mtrace2 Header	34
9.2.	Verification of Clients and Peers	34
9.3.	Topology Discovery	35
9.4.	Characteristics of Multicast Channel	35
9.5.	Limiting Query/Request Rates	35
9.6.	Limiting Reply Rates	36
9.7.	Specific Security Concerns	36
9.7.1.	Request and Response Bombardment	36
9.7.2.	Amplification Attack	36
9.7.3.	Leaking of Confidential Topology Details	36
9.7.4.	Delivery of False Information (Forged Reply Messages)	37
10.	Acknowledgements	38
11.	References	38
11.1.	Normative References	38
11.2.	Informative References	39
	Authors' Addresses	39

1. Introduction

Given a multicast distribution tree, tracing hop-by-hop downstream from a multicast source to a given multicast receiver is difficult because there is no efficient and deterministic way to determine the branch of the multicast routing tree on which that receiver lies. On the other hand, walking up the tree from a receiver to a source is easy, as most existing multicast routing protocols know the upstream router for each source. Tracing from a receiver to a source can involve only the routers on the direct path.

This document specifies the multicast traceroute facility named Mtrace version 2 or Mtrace2 which allows the tracing of an IP multicast routing path. Mtrace2 is usually initiated from an Mtrace2 client by sending an Mtrace2 Query to a Last Hop Router (LHR) or to a Rendezvous Point (RP). The RP is a special router where sources and receivers meet in Protocol Independent Multicast - Sparse Mode (PIM-SM) [5]. From the LHR/RP receiving the query, the tracing is directed towards a specified source if a source address is specified and source specific state exists on the receiving router. If no source address is specified or if no source specific state exists on a receiving LHR, the tracing is directed toward the RP for the specified group address. Moreover, Mtrace2 provides additional information such as the packet rates and losses, as well as other diagnostic information. Mtrace2 is primarily intended for the following purposes:

- o To trace the path that a packet would take from a source to a receiver.
- o To isolate packet loss problems (e.g., congestion).
- o To isolate configuration problems (e.g., Time to live (TTL) threshold).

Figure 1 shows a typical case on how Mtrace2 is used. First-hop router (FHR) represents the first-hop router, LHR represents the last-hop router (LHR), and the arrow lines represent the Mtrace2 messages that are sent from one node to another. The numbers before the Mtrace2 messages represent the sequence of the messages that would happen. Source, Receiver and Mtrace2 client are typically hosts.

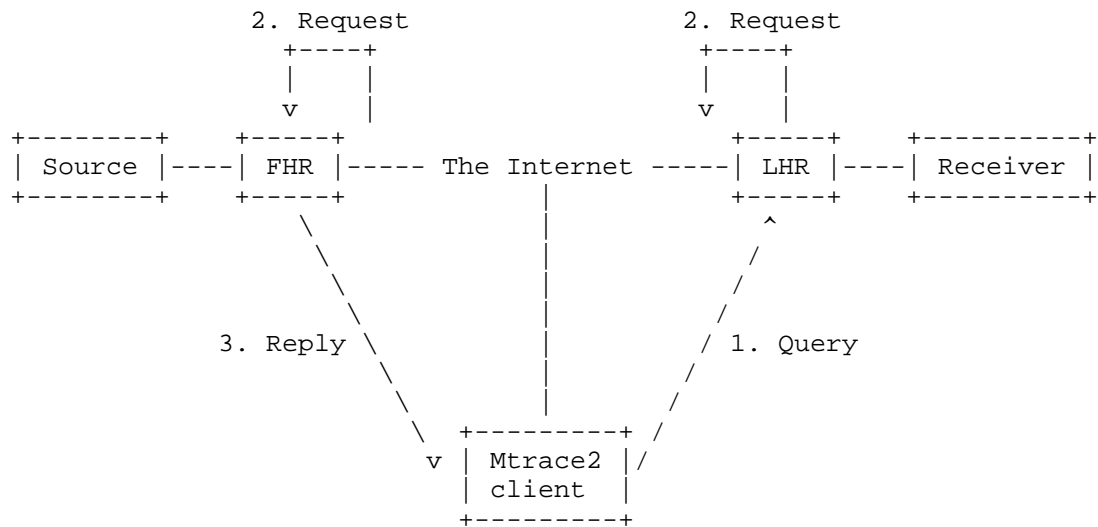


Figure 1

When an Mtrace2 client initiates a multicast trace, it sends an Mtrace2 Query packet to an LHR or RP for a multicast group and, optionally, a source address. The LHR/RP turns the Query packet into a Request. The Request message type enables each of the upstream routers processing the message to apply different packet and message validation rules than those required for handling of a Query message. The LHR/RP then appends a standard response block containing its interface addresses and packet statistics to the Request packet, then forwards the packet towards the source/RP. The Request packet is either unicasted to its upstream router towards the source/RP, or multicasted to the group if the upstream router's IP address is not known. In a similar fashion, each router along the path to the source/RP appends a standard response block to the end of the Request packet before forwarding it to its upstream router. When the FHR receives the Request packet, it appends its own standard response block, turns the Request packet into a Reply, and unicasts the Reply back to the Mtrace2 client.

The Mtrace2 Reply may be returned before reaching the FHR under some circumstances. This can happen if a Request packet is received at an RP or gateway, or when any of several types of error or exception conditions occur which prevent sending of a request to the next upstream router.

The Mtrace2 client waits for the Mtrace2 Reply message and displays the results. When not receiving an Mtrace2 Reply message due to network congestion, a broken router (see Section 5.6), or a non-

responding router (see Section 5.7), the Mtrace2 client may resend another Mtrace2 Query with a lower hop count (see Section 3.2.1), and repeat the process until it receives an Mtrace2 Reply message. The details are Mtrace2 client specific and outside the scope of this document.

Note that when a router's control plane and forwarding plane are out of sync, the Mtrace2 Requests might be forwarded based on the control states instead. In this case, the traced path might not represent the real path the data packets would follow.

Mtrace2 supports both IPv4 and IPv6. Unlike the previous version of Mtrace, which implements its query and response as Internet Group Management Protocol (IGMP) messages [8], all Mtrace2 messages are UDP-based. Although the packet formats of IPv4 and IPv6 Mtrace2 are different because of the address families, the syntax between them is similar.

This document describes the base specification of Mtrace2 that can serve as a basis for future proposals such as Mtrace2 for Automatic Multicast Tunneling (AMT) [9] and Mtrace2 for Multicast in MPLS/BGP IP VPNs (MVPN) [10]. They are therefore out of the scope of this document.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [1], and indicate requirement levels for compliant Mtrace2 implementations.

2.1. Definitions

Since Mtrace2 Queries and Requests flow in the opposite direction to the data flow, we refer to "upstream" and "downstream" with respect to data, unless explicitly specified.

Incoming interface

The interface on which data is expected to arrive from the specified source and group.

Outgoing interface

This is one of the interfaces to which data from the source or RP is expected to be transmitted for the specified source and group. It is also the interface on which the Mtrace2 Request was received.

Upstream router

The router, connecting to the Incoming interface of the current router, which is responsible for forwarding data for the specified source and group to the current router.

First-hop router (FHR)

The router that is directly connected to the source the Mtrace2 Query specifies.

Last-hop router (LHR)

A router that is directly connected to a receiver. It is also the router that receives the Mtrace2 Query from an Mtrace2 client.

Group state

The state a shared-tree protocol, such as PIM-SM [5], uses to choose the upstream router towards the RP for the specified group. In this state, source-specific state is not available for the corresponding group address on the router.

Source-specific state

The state that is used to choose the path towards the source for the specified source and group.

ALL-[protocol]-ROUTERS group

Link-local multicast address for multicast routers to communicate with their adjacent routers that are running the same routing protocol. For instance, the IPv4 'ALL-PIM-ROUTERS' group is '224.0.0.13', and the IPv6 'ALL-PIM-ROUTERS' group is 'ff02::d' [5].

3. Packet Formats

This section describes the details of the packet formats for Mtrace2 messages.

All Mtrace2 messages are encoded in the Type/Length/Value (TLV) format (see Section 3.1). The first TLV of a message is a message header TLV specifying the type of message and additional context information required for processing of the message and for parsing of subsequent TLVs in the message. Subsequent TLVs in a message, referred to as Blocks, are appended after the header TLV to provide additional information associated with the message. If an implementation receives an unknown TLV type for any TLV in a message, it SHOULD ignore and silently discard the entire packet. If the length of a TLV exceeds the available space in the containing packet, the implementation MUST ignore and silently discard the TLV and any remaining portion of the containing packet.

All Mtrace2 messages are UDP packets. For IPv4, Mtrace2 Query/Request/Reply messages MUST NOT be fragmented. Therefore, Mtrace2 clients and LHRs/RPs MUST set the IP header do-not-fragment (DF) bit for all Mtrace2 messages. For IPv6, the packet size for the Mtrace2 messages MUST NOT exceed 1280 bytes, which is the smallest Maximum Transmission Unit (MTU) for an IPv6 interface [2]. The source port is uniquely selected by the local host operating system. The destination port is the IANA reserved Mtrace2 port number (see Section 8). All Mtrace2 messages MUST have a valid UDP checksum.

Additionally, Mtrace2 supports both IPv4 and IPv6, but not mixed. For example, if an Mtrace2 Query or Request message arrives in as an IPv4 packet, all addresses specified in the Mtrace2 messages MUST be IPv4 as well. Same rule applies to IPv6 Mtrace2 messages.

3.1. Mtrace2 TLV format

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      Type      |      Length      |      Value ...      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Type: 8 bits

Describes the format of the Value field. For all the available types, please see Section 3.2

Length: 16 bits

Length of Type, Length, and Value fields in octets. Minimum length required is 4 octets. The length MUST be a multiple of 4 octets. The maximum TLV length is not defined; however the entire Mtrace2 packet length MUST NOT exceed the available MTU.

Value: variable length

The format is based on the Type value. The length of the value field is Length field minus 3. All reserved fields in the Value field MUST be transmitted as zeros and ignored on receipt.

3.2. Defined TLVs

The following TLV Types are defined:

Code	Type
====	=====
0x00	Reserved
0x01	Mtrace2 Query
0x02	Mtrace2 Request
0x03	Mtrace2 Reply
0x04	Mtrace2 Standard Response Block
0x05	Mtrace2 Augmented Response Block
0x06	Mtrace2 Extended Query Block

Each Mtrace2 message MUST begin with either a Query, Request or Reply TLV. The first TLV determines the type of each Mtrace2 message. Following a Query TLV, there can be a sequence of optional Extended Query Blocks. In the case of a Request or a Reply TLV, it is then followed by a sequence of Standard Response Blocks, each from a multicast router on the path towards the source or the RP. In the case more information is needed, a Standard Response Block can be followed by one or multiple Augmented Response Blocks.

We will describe each message type in detail in the next few sections.

3.2.1. Mtrace2 Query

An Mtrace2 Query is originated by an Mtrace2 client which sends an Mtrace2 Query message to the LHR. The LHR modifies only the Type field of the Query TLV (to turn it into a "Request") before appending a Standard Response Block and forwarding it upstream. The LHR and intermediate routers handling the Mtrace2 message when tracing upstream MUST NOT modify any other fields within the Query/Request TLV. Additionally, intermediate routers handling the message after the LHR has converted the Query into a Request MUST NOT modify the type field of the Request TLV. If the actual number of hops is not known, an Mtrace2 client could send an initial Query message with a large # Hops (e.g., 0xff), in order to try to trace the full path.

An Mtrace2 Query message is shown as follows:

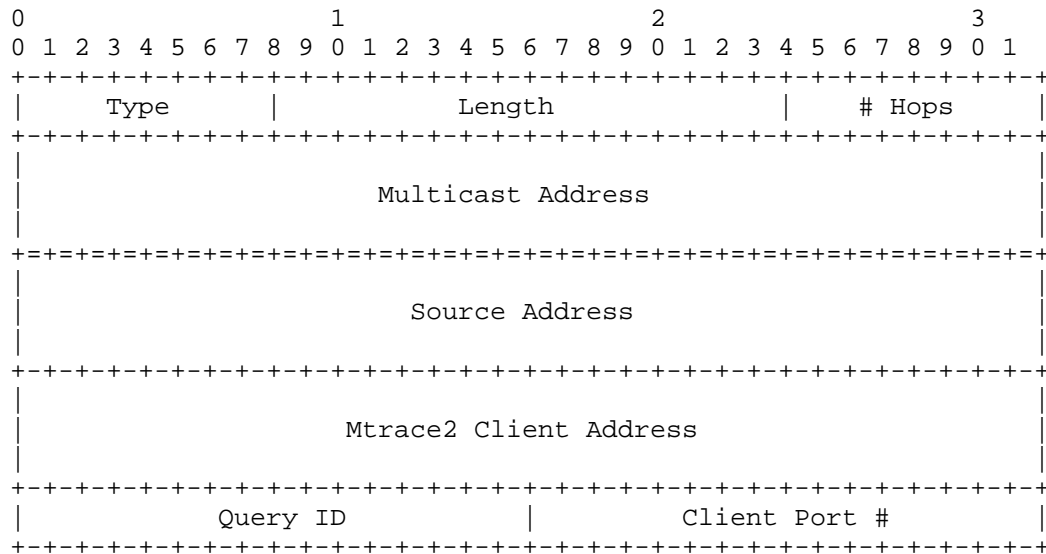


Figure 2

Length: 16 bits

The length field MUST be either 20 (i.e., 8 plus 3 * 4 (IPv4 addresses)) or 56 (i.e., 8 + 3 * 16 (IPv6 addresses)); if the length is 20, then IPv4 addresses MUST be assumed and if the length is 56, then IPv6 addresses MUST be assumed.

Hops: 8 bits

This field specifies the maximum number of hops that the Mtrace2 client wants to trace. If there are some error conditions in the middle of the path that prevent an Mtrace2 Reply from being received by the client, the client MAY issue another Mtrace2 Query with a lower number of hops until it receives a Reply.

Multicast Address: 32 bits or 128 bits

This field specifies an IPv4 or IPv6 address, which can be either:

m-1: a multicast group address to be traced; or,

m-2: all 1's in case of IPv4 or the unspecified address (::) in case of IPv6 if no group-specific information is desired.

Source Address: 32 bits or 128 bits

This field specifies an IPv4 or IPv6 address, which can be either:

s-1: a unicast address of the source to be traced; or,

s-2: all 1's in case of IPv4 or the unspecified address (::) in case of IPv6 if no source-specific information is desired. For example, the client is tracing a (*,g) group state.

Note that it is invalid to have a source-group combination of (s-2, m-2). If a router receives such combination in an Mtrace2 Query, it MUST silently discard the Query.

Mtrace2 Client Address: 32 bits or 128 bits

This field specifies the Mtrace2 client's IPv4 address or IPv6 global address. This address MUST be a valid unicast address, and therefore, MUST NOT be all 1's or an unspecified address. The Mtrace2 Reply will be sent to this address.

Query ID: 16 bits

This field is used as a unique identifier for this Mtrace2 Query so that duplicate or delayed Reply messages may be detected.

Client Port #: 16 bits

This field specifies the destination UDP port number for receiving the Mtrace2 Reply packet.

3.2.2. Mtrace2 Request

The Mtrace2 Request TLV is exactly the same as an Mtrace2 Query except for identifying the Type field of 0x02.

When a LHR receives an Mtrace2 Query message, it turns the Query into a Request by changing the Type field of the Query from 0x01 to 0x02. The LHR then appends an Mtrace2 Standard Response Block (see Section 3.2.4) of its own to the Request message before sending it upstream. The upstream routers do the same without changing the Type field until one of them is ready to send a Reply.

3.2.3. Mtrace2 Reply

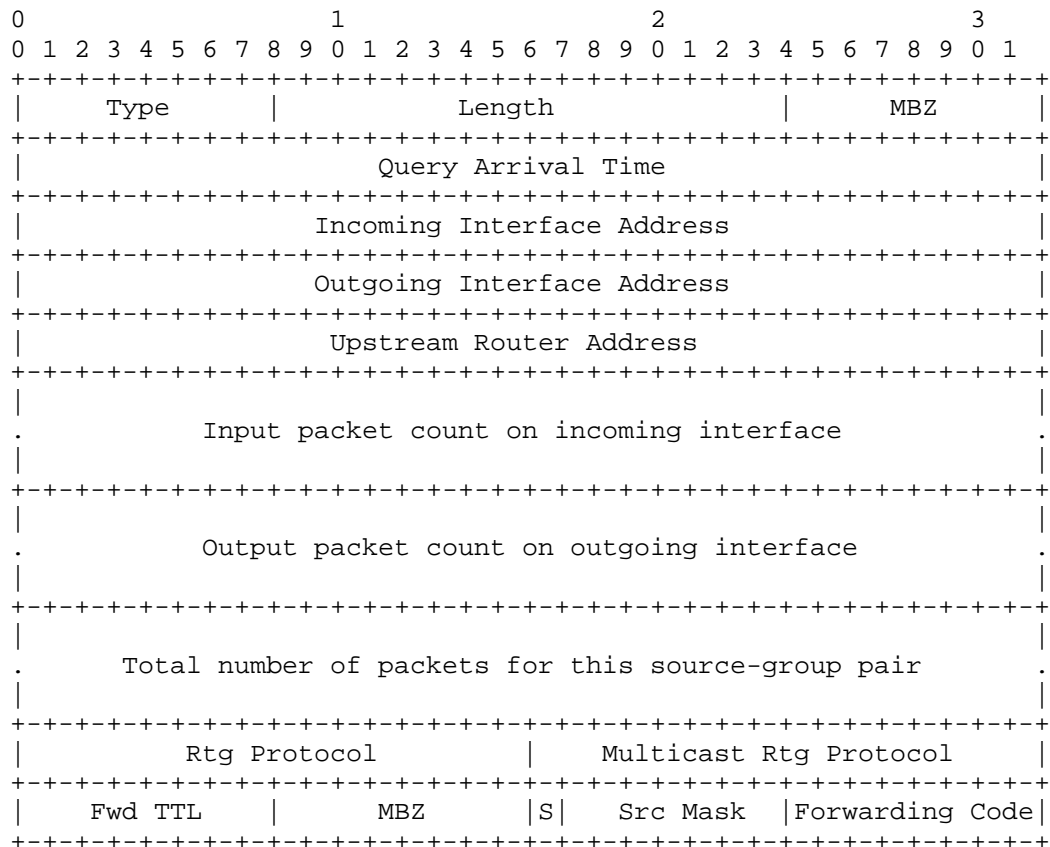
The Mtrace2 Reply TLV is exactly the same as an Mtrace2 Query except for identifying the Type field of 0x03.

When a FHR or an RP receives an Mtrace2 Request message which is destined to itself, it appends an Mtrace2 Standard Response Block (see Section 3.2.4) of its own to the Request message. Next, it turns the Request message into a Reply by changing the Type field of the Request from 0x02 to 0x03 and by changing the UDP destination port to the port number specified in the Client Port number field in the Request. It then unicasts the Reply message to the Mtrace2 client specified in the Mtrace2 Client Address field.

There are a number of cases in which an intermediate router might return a Reply before a Request reaches the FHR or the RP. See Section 4.1.1, Section 4.2.2, Section 4.3.3, and Section 4.5 for more details.

3.2.4. IPv4 Mtrace2 Standard Response Block

This section describes the message format of an IPv4 Mtrace2 Standard Response Block. The Type field is 0x04.



MBZ: 8 bits

This field MUST be zeroed on transmission and ignored on reception.

Query Arrival Time: 32 bits

The Query Arrival Time is a 32-bit Network Time Protocol (NTP) timestamp specifying the arrival time of the Mtrace2 Query or Request packet at this router. The 32-bit form of an NTP

timestamp consists of the middle 32 bits of the full 64-bit form; that is, the low 16 bits of the integer part and the high 16 bits of the fractional part.

The following formula converts from a timespec (fractional part in nanoseconds) to a 32-bit NTP timestamp:

```
query_arrival_time
= ((tv.tv_sec + 32384) << 16) + ((tv.tv_nsec << 7) / 1953125)
```

The constant 32384 is the number of seconds from Jan 1, 1900 to Jan 1, 1970 truncated to 16 bits. $((tv.tv_nsec \ll 7) / 1953125)$ is a reduction of $((tv.tv_nsec / 1000000000) \ll 16)$.

Note that synchronized clocks are required on the traced routers to estimate propagation and queueing delays between successive hops. Nevertheless, even without this synchronization, an application can still estimate an upper bound on cumulative one way latency by measuring the time between sending a Query and receiving a Reply.

Additionally, Query Arrival Time is useful for measuring the packet rate. For example, suppose that a client issues two queries, and the corresponding requests R1 and R2 arrive at router X at time T1 and T2, then the client would be able to compute the packet rate on router X by using the packet count information stored in the R1 and R2, and the time T1 and T2.

Incoming Interface Address: 32 bits

This field specifies the address of the interface on which packets from the source or the RP are expected to arrive, or 0 if unknown or unnumbered.

Outgoing Interface Address: 32 bits

This field specifies the address of the interface on which packets from the source or the RP are expected to transmit towards the receiver, or 0 if unknown or unnumbered. This is also the address of the interface on which the Mtrace2 Query or Request arrives.

Upstream Router Address: 32 bits

This field specifies the address of the upstream router from which this router expects packets from this source. This MAY be a multicast group (e.g., ALL-[protocol]-ROUTERS group) if the upstream router is not known because of the workings of the multicast routing protocol. However, it MUST be 0 if the incoming interface address is unknown or unnumbered.

Input packet count on incoming interface: 64 bits

This field contains the number of multicast packets received for all groups and sources on the incoming interface, or all 1's if no count can be reported. This counter may have the same value as ifHCInMulticastPkts from the Interfaces Group MIB (IF-MIB) [12] for this interface.

Output packet count on outgoing interface: 64 bit

This field contains the number of multicast packets that have been transmitted or queued for transmission for all groups and sources on the outgoing interface, or all 1's if no count can be reported. This counter may have the same value as ifHCOutMulticastPkts from the IF-MIB [12] for this interface.

Total number of packets for this source-group pair: 64 bits

This field counts the number of packets from the specified source forwarded by the router to the specified group, or all 1's if no count can be reported. If the S bit is set (see below), the count is for the source network, as specified by the Src Mask field (see below). If the S bit is set and the Src Mask field is 127, indicating no source-specific state, the count is for all sources sending to this group. This counter should have the same value as ipMcastRoutePkts from the IP Multicast MIB [13] for this forwarding entry.

Rtg Protocol: 16 bits

This field describes the unicast routing protocol running between this router and the upstream router, and it is used to determine the RPF interface for the specified source or RP. This value should have the same value as ipMcastRouteRtProtocol from the IP Multicast MIB [13] for this entry. If the router is not able to obtain this value, all 0's must be specified.

Multicast Rtg Protocol: 16 bits

This field describes the multicast routing protocol in use between the router and the upstream router. This value should have the same value as ipMcastRouteProtocol from the IP Multicast MIB [13] for this entry. If the router cannot obtain this value, all 0's must be specified.

Fwd TTL: 8 bits

This field contains the configured multicast TTL threshold, if any, of the outgoing interface.

S: 1 bit

If this bit is set, it indicates that the packet count for the source-group pair is for the source network, as determined by masking the source address with the Src Mask field.

Src Mask: 7 bits

This field contains the number of 1's in the netmask the router has for the source (i.e. a value of 24 means the netmask is 0xffffffff00). If the router is forwarding solely on group state, this field is set to 127 (0x7f).

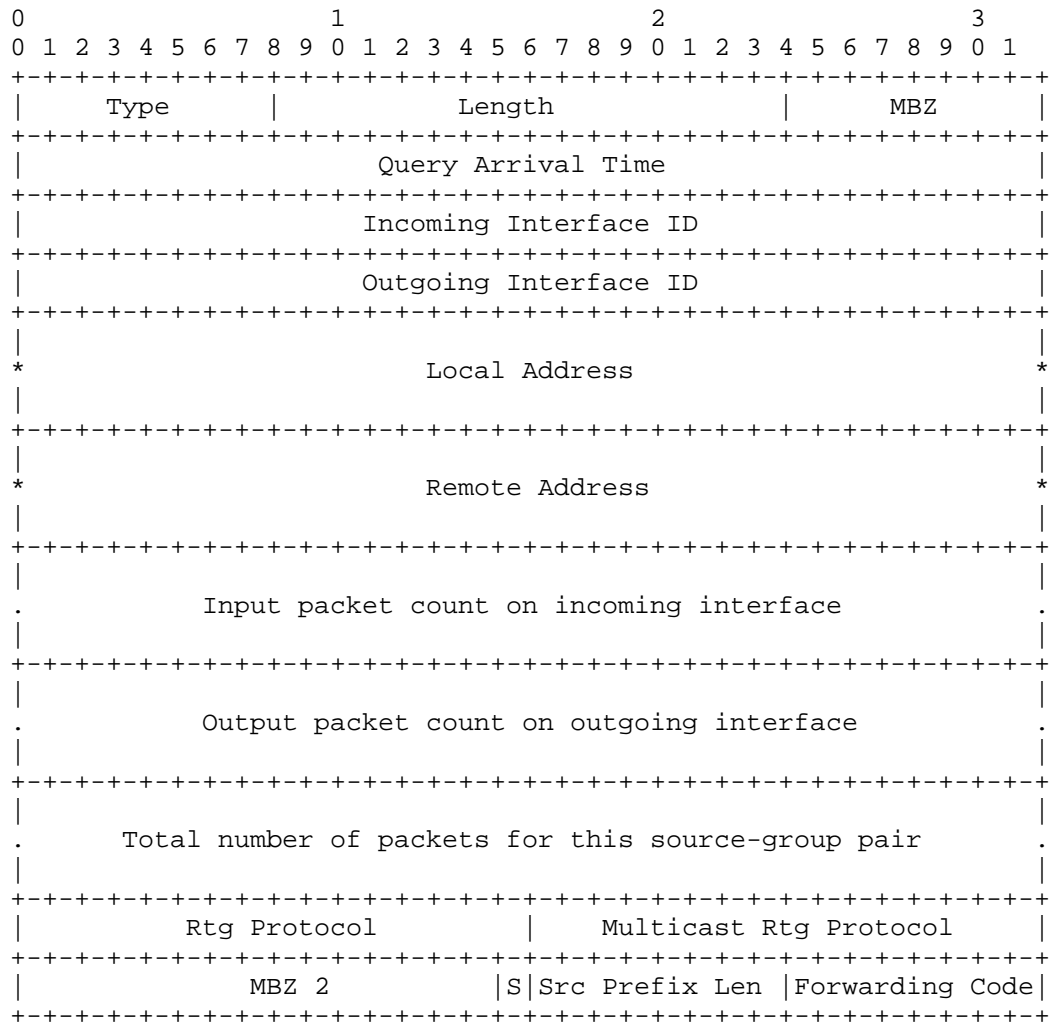
Forwarding Code: 8 bits

This field contains a forwarding information/error code. Values with the high order bit set (0x80-0xff) are intended for use with conditions that are transitory or automatically recovered. Other forwarding code values indicate a need to fix a problem in the Query or a need to redirect the Query. Section 4.1 and Section 4.2 explain how and when the Forwarding Code is filled. Defined values are as follows:

Value	Name	Description
-----	-----	-----
0x00	NO_ERROR	No error
0x01	WRONG_IF	Mtrace2 Request arrived on an interface to which this router would not forward for the specified group towards the source or RP.
0x02	PRUNE_SENT	This router has sent a prune upstream which applies to the source and group in the Mtrace2 Request.
0x03	PRUNE_RCVD	This router has stopped forwarding for this source and group in response to a request from the downstream router.
0x04	SCOPED	The group is subject to administrative scoping at this router.
0x05	NO_ROUTE	This router has no route for the source or group and no way to determine a potential route.
0x06	WRONG_LAST_HOP	This router is not the proper LHR.
0x07	NOT_FORWARDING	This router is not forwarding this source and group out the outgoing interface for an unspecified reason.
0x08	REACHED_RP	Reached the Rendezvous Point.
0x09	RPF_IF	Mtrace2 Request arrived on the expected RPF interface for this source and group.
0x0A	NO_MULTICAST	Mtrace2 Request arrived on an interface which is not enabled for multicast.
0x0B	INFO_HIDDEN	One or more hops have been hidden from this trace.
0x0C	REACHED_GW	Mtrace2 Request arrived on a gateway (e.g., a NAT or firewall) that hides the information between this router and the Mtrace2 client.
0x0D	UNKNOWN_QUERY	A non-transitive Extended Query Type was received by a router which does not support the type.
0x80	FATAL_ERROR	A fatal error is one where the router may know the upstream router but cannot forward the message to it.
0x81	NO_SPACE	There was not enough room to insert another Standard Response Block in the packet.
0x83	ADMIN_PROHIB	Mtrace2 is administratively prohibited.

3.2.5. IPv6 Mtrace2 Standard Response Block

This section describes the message format of an IPv6 Mtrace2 Standard Response Block. The Type field is also 0x04.



MBZ: 8 bits

This field MUST be zeroed on transmission and ignored on reception.

Query Arrival Time: 32 bits

Same definition as in IPv4.

Incoming Interface ID: 32 bits

This field specifies the interface ID on which packets from the source or RP are expected to arrive, or 0 if unknown. This ID should be the value taken from InterfaceIndex of the IF-MIB [12] for this interface.

Outgoing Interface ID: 32 bits

This field specifies the interface ID to which packets from the source or RP are expected to transmit, or 0 if unknown. This ID should be the value taken from InterfaceIndex of the IF-MIB [12] for this interface

Local Address: 128 bits

This field specifies a global IPv6 address that uniquely identifies the router. A unique local unicast address [11] SHOULD NOT be used unless the router is only assigned link-local and unique local addresses. If the router is only assigned link-local addresses, its link-local address can be specified in this field.

Remote Address: 128 bits

This field specifies the address of the upstream router, which, in most cases, is a link-local unicast address for the upstream router.

Although a link-local address does not have enough information to identify a node, it is possible to detect the upstream router with the assistance of Incoming Interface ID and the current router address (i.e., Local Address).

Note that this may be a multicast group (e.g., ALL-[protocol]-ROUTERS group) if the upstream router is not known because of the workings of a multicast routing protocol. However, it should be the unspecified address (::) if the incoming interface address is unknown.

Input packet count on incoming interface: 64 bits

Same definition as in IPv4.

Output packet count on outgoing interface: 64 bits

Same definition as in IPv4.

Total number of packets for this source-group pair: 64 bits

Same definition as in IPv4, except if the S bit is set (see below), the count is for the source network, as specified by the Src Prefix Len field. If the S bit is set and the Src Prefix Len field is 255, indicating no source-specific state, the count is for all sources sending to this group. This counter should have the same value as ipMcastRoutePkts from the IP Multicast MIB [13] for this forwarding entry.

Rtg Protocol: 16 bits

Same definition as in IPv4.

Multicast Rtg Protocol: 16 bits

Same definition as in IPv4.

MBZ 2: 15 bits

This field MUST be zeroed on transmission and ignored on reception.

S: 1 bit

Same definition as in IPv4, except the Src Prefix Len field is used to mask the source address.

Src Prefix Len: 8 bits

This field contains the prefix length this router has for the source. If the router is forwarding solely on group state, this field is set to 255 (0xff).

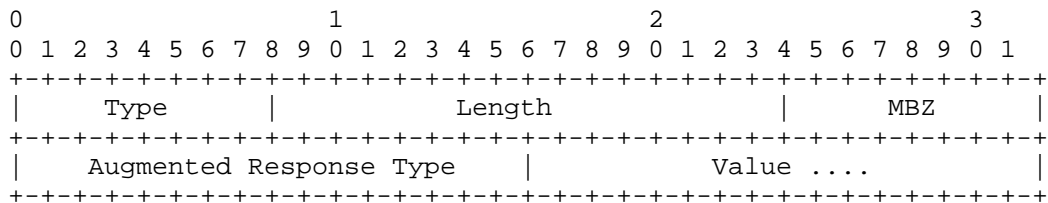
Forwarding Code: 8 bits

Same definition as in IPv4.

3.2.6. Mtrace2 Augmented Response Block

In addition to the Standard Response Block, a multicast router on the traced path can optionally add one or multiple Augmented Response Blocks before sending the Request to its upstream router.

The Augmented Response Block is flexible for various purposes such as providing diagnosis information (see Section 7) and protocol verification. Its Type field is 0x05, and its format is as follows:



MBZ: 8 bits

This field MUST be zeroed on transmission and ignored on reception.

Augmented Response Type: 16 bits

This field specifies the type of various responses from a multicast router that might need to communicate back to the Mtrace2 client as well as the multicast routers on the traced path.

The Augmented Response Type is defined as follows:

Code	Type
0x0001	# of the returned Standard Response Blocks

When the NO_SPACE error occurs on a router, the router should send the original Mtrace2 Request received from the downstream router as a Reply back to the Mtrace2 client and continue with a new Mtrace2 Request. In the new Request, the router adds a Standard Response Block followed by an Augmented Response Block with 0x01 as the Augmented Response Type, and the number of the returned Mtrace2 Standard Response Blocks as the Value.

Each upstream router recognizes the total number of hops the Request has been traced so far by adding this number and the number of the Standard Response Block in the current Request message.

This document only defines one Augmented Response Type in the Augmented Response Block. The description on how to provide diagnosis information using the Augmented Response Block is out of the scope of this document, and will be addressed in separate documents.

Value: variable length

The format is based on the Augmented Response Type value. The length of the value field is Length field minus 6.

3.2.7. Mtrace2 Extended Query Block

There may be a sequence of optional Extended Query Blocks that follow an Mtrace2 Query to further specify any information needed for the Query. For example, an Mtrace2 client might be interested in tracing the path the specified source and group would take based on a certain topology. In this case, the client can pass in the multi-topology ID as the Value for an Extended Query Type (see below). The Extended Query Type is extensible and the behavior of the new types will be addressed by separate documents.

The Mtrace2 Extended Query Block's Type field is 0x06, and is formatted as follows:

										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Type										Length										MBZ										T									
Extended Query Type										Value ...																													

MBZ: 7 bits

This field MUST be zeroed on transmission and ignored on reception.

T-bit (Transitive Attribute): 1 bit

If the TLV type is unrecognized by the receiving router, then this TLV is either discarded or forwarded along with the Query, depending on the value of this bit. If this bit is set, then the router MUST forward this TLV. If this bit is clear, the router MUST send an Mtrace2 Reply with an UNKNOWN_QUERY error.

Extended Query Type: 16 bits

This field specifies the type of the Extended Query Block.

Value: 16 bits

This field specifies the value of this Extended Query.

4. Router Behavior

This section describes the router behavior in the context of Mtrace2 in detail.

4.1. Receiving Mtrace2 Query

An Mtrace2 Query message is an Mtrace2 message with no response blocks filled in, and uses TLV type of 0x01.

4.1.1. Query Packet Verification

Upon receiving an Mtrace2 Query message, a router MUST examine whether the Multicast Address and the Source Address are a valid combination as specified in Section 3.2.1, and whether the Mtrace2 Client Address is a valid IP unicast address. If either one is invalid, the Query MUST be silently ignored.

Mtrace2 supports a non-local client to the LHR/RP. A router MUST, however, support a mechanism to drop Queries from clients beyond a specified administrative boundary. The potential approaches are described in Section 9.2.

In the case where a local LHR client is required, the router must then examine the Query to see if it is the proper LHR/RP for the destination address in the packet. It is the proper local LHR if it has a multicast-capable interface on the same subnet as the Mtrace2 Client Address and is the router that would forward traffic from the given (S,G) or (*,G) onto that subnet. It is the proper RP if the multicast group address specified in the query is 0 and if the IP header destination address is a valid RP address on this router.

If the router determines that it is not the proper LHR/RP, or it cannot make that determination, it does one of two things depending on whether the Query was received via multicast or unicast. If the Query was received via multicast, then it MUST be silently discarded. If it was received via unicast, the router turns the Query into a Reply message by changing the TLV type to 0x03 and appending a Standard Response Block with a Forwarding Code of WRONG_LAST_HOP. The rest of the fields in the Standard Response Block MUST be zeroed. The router then sends the Reply message to the Mtrace2 Client Address on the Client Port # as specified in the Mtrace2 Query.

Duplicate Query messages as identified by the tuple (Mtrace2 Client Address, Query ID) SHOULD be ignored. This MAY be implemented using a cache of previously processed queries keyed by the Mtrace2 Client Address and Query ID pair. The duration of the cached entries is implementation specific. Duplicate Request messages MUST NOT be ignored in this manner.

4.1.2. Query Normal Processing

When a router receives an Mtrace2 Query and it determines that it is the proper LHR/RP, it turns the Query to a Request by changing the TLV type from 0x01 to 0x02, and performs the steps listed in Section 4.2.

4.2. Receiving Mtrace2 Request

An Mtrace2 Request is an Mtrace2 message that uses TLV type of 0x02. With the exception of the LHR, whose Request was just converted from a Query, each Request received by a router should have at least one Standard Response Block filled in.

4.2.1. Request Packet Verification

If the Mtrace2 Request does not come from an adjacent router, or if the Request is not addressed to this router, or if the Request is addressed to a multicast group which is not a link-scoped group (i.e., 224.0.0.0/24 for IPv4, FFx2::/16 [3] for IPv6), it MUST be silently ignored. The Generalized TTL Security Mechanism (GTSM) [14] SHOULD be used by the router to determine whether the router is adjacent or not. Source verification specified in Section 9.2 is also considered.

If the sum of the number of the Standard Response Blocks in the received Mtrace2 Request and the value of the Augmented Response Type of 0x01, if any, is equal or more than the # Hops in the Mtrace2 Request, it MUST be silently ignored.

4.2.2. Request Normal Processing

When a router receives an Mtrace2 Request message, it performs the following steps. Note that it is possible to have multiple situations covered by the Forwarding Codes. The first one encountered is the one that is reported, i.e. all "note Forwarding Code N" should be interpreted as "if Forwarding Code is not already set, set Forwarding Code to N". Note that in the steps described below the "Outgoing Interface" is the one on which the Mtrace2 Request message arrives.

1. Prepare a Standard Response Block to be appended to the packet, setting all fields to an initial default value of zero.
2. If Mtrace2 is administratively prohibited, note the Forwarding Code of ADMIN_PROHIB and skip to step 4.
3. In the Standard Response Block, fill in the Query Arrival Time, Outgoing Interface Address (for IPv4) or Outgoing Interface ID (for IPv6), Output Packet Count, and Fwd TTL (for IPv4).
4. Attempt to determine the forwarding information for the specified source and group, using the same mechanisms as would be used when a packet is received from the source destined for the group. A state need not be instantiated, it can be a "phantom" state created only for the purpose of the trace, such as "dry-run."

If using a shared-tree protocol and there is no source-specific state, or if no source-specific information is desired (i.e., all 1's for IPv4 or unspecified address (::) for IPv6), group state should be used. If there is no group state or no group-specific information is desired, potential source state (i.e., the path that would be followed for a source-specific Join) should be used.

5. If no forwarding information can be determined, the router notes a Forwarding Code of NO_ROUTE, sets the remaining fields that have not yet been filled in to zero, and then sends an Mtrace2 Reply back to the Mtrace2 client.
6. If a Forwarding Code of ADMIN_PROHIB has been set, skip to step 7. Otherwise, fill in the Incoming Interface Address (or Incoming Interface ID and Local Address for IPv6), Upstream Router Address (or Remote Address for IPv6), Input Packet Count, Total Number of Packets, Routing Protocol, S, and Src Mask (or Src Prefix Len for IPv6) using the forwarding information determined in step 4.

7. If the Outgoing interface is not enabled for multicast, note Forwarding Code of NO_MULTICAST. If the Outgoing interface is the interface from which the router would expect data to arrive from the source, note forwarding code RPF_IF. If the Outgoing interface is not one to which the router would forward data from the source or RP to the group, a Forwarding code of WRONG_IF is noted. In the above three cases, the router will return an Mtrace2 Reply and terminate the trace.
8. If the group is subject to administrative scoping on either the Outgoing or Incoming interfaces, a Forwarding Code of SCOPED is noted.
9. If this router is the RP for the group for a non-source-specific query, note a Forwarding Code of REACHED_RP. The router will send an Mtrace2 Reply and terminate the trace.
10. If this router is directly connected to the specified source or source network on the Incoming interface, it sets the Upstream Router Address (for IPv4) or the Remote Address (for IPv6) of the response block to zero. The router will send an Mtrace2 Reply and terminate the trace.
11. If this router has sent a prune upstream which applies to the source and group in the Mtrace2 Request, it notes a Forwarding Code of PRUNE_SENT. If the router has stopped forwarding downstream in response to a prune sent by the downstream router, it notes a Forwarding Code of PRUNE_RCVD. If the router should normally forward traffic downstream for this source and group but is not, it notes a Forwarding Code of NOT_FORWARDING.
12. If this router is a gateway (e.g., a NAT or firewall) that hides the information between this router and the Mtrace2 client, it notes a Forwarding Code of REACHED_GW. The router continues the processing as described in Section 4.5.
13. If the total number of the Standard Response Blocks, including the newly prepared one, and the value of the Augmented Response Type of 0x01, if any, is less than the # Hops in the Request, the packet is then forwarded to the upstream router as described in Section 4.3; otherwise, the packet is sent as an Mtrace2 Reply to the Mtrace2 client as described in Section 4.4.

4.3. Forwarding Mtrace2 Request

This section describes how an Mtrace2 Request should be forwarded.

4.3.1. Destination Address

If the upstream router for the Mtrace2 Request is known for this request, the Mtrace2 Request is sent to that router. If the Incoming interface is known but the upstream router is not, the Mtrace2 Request is sent to an appropriate multicast address on the Incoming interface. The multicast address SHOULD depend on the multicast routing protocol in use, such as ALL-[protocol]-ROUTERS group. It MUST be a link-scoped group (i.e., 224.0.0.0/24 for IPv4, FF02::/16 for IPv6), and MUST NOT be the all-systems multicast group (224.0.0.1) for IPv4 and All Nodes Address (FF02::1) for IPv6. It MAY also be the all-routers multicast group (224.0.0.2) for IPv4 or All Routers Address (FF02::2) for IPv6 if the routing protocol in use does not define a more appropriate multicast address.

4.3.2. Source Address

An Mtrace2 Request should be sent with the address of the Incoming interface. However, if the Incoming interface is unnumbered, the router can use one of its numbered interface addresses as the source address.

4.3.3. Appending Standard Response Block

An Mtrace2 Request MUST be sent upstream towards the source or the RP after appending a Standard Response Block to the end of the received Mtrace2 Request. The Standard Response Block includes the multicast states and statistics information of the router described in Section 3.2.4.

If appending the Standard Response Block would make the Mtrace2 Request packet longer than the MTU of the Incoming Interface, or, in the case of IPv6, longer than 1280 bytes, the router MUST change the Forwarding Code in the last Standard Response Block of the received Mtrace2 Request into NO_SPACE. The router then turns the Request into a Reply and sends the Reply as described in Section 4.4.

The router will continue with a new Request by copying from the old Request excluding all the response blocks, followed by the previously prepared Standard Response Block, and an Augmented Response Block with Augmented Response Type of 0x01 and the number of the returned Standard Response Blocks as the value. The new Request is then forwarded upstream.

4.4. Sending Mtrace2 Reply

An Mtrace2 Reply MUST be returned to the client by a router if any of the following conditions occur:

1. The total number of the traced routers are equal to the # of hops in the request (including the one just added) plus the number of the returned blocks, if any.
2. Appending the Standard Response Block would make the Mtrace2 Request packet longer than the MTU of the Incoming interface. (In case of IPv6 not more than 1280 bytes; see Section 4.3.3 for additional details on handling of this case.)
3. The request has reached the RP for a non source specific query or has reached the first hop router for a source specific query (see Section 4.2.2, items 9 and 10 for additional details).

4.4.1. Destination Address

An Mtrace2 Reply MUST be sent to the address specified in the Mtrace2 Client Address field in the Mtrace2 Request.

4.4.2. Source Address

An Mtrace2 Reply SHOULD be sent with the address of the router's Outgoing interface. However, if the Outgoing interface address is unnumbered, the router can use one of its numbered interface addresses as the source address.

4.4.3. Appending Standard Response Block

An Mtrace2 Reply MUST be sent with the prepared Standard Response Block appended at the end of the received Mtrace2 Request except in the case of NO_SPACE forwarding code.

4.5. Proxying Mtrace2 Query

When a gateway (e.g., a NAT or firewall), which needs to block unicast packets to the Mtrace2 client, or hide information between the gateway and the Mtrace2 client, receives an Mtrace2 Query from an adjacent host or Mtrace2 Request from an adjacent router, it appends a Standard Response Block with REACHED_GW as the Forwarding Code. It turns the Query or Request into a Reply, and sends the Reply back to the client.

At the same time, the gateway originates a new Mtrace2 Query message by copying the original Mtrace2 header (the Query or Request without any of the response blocks), and makes the changes as follows:

- o sets the RPF interface's address as the Mtrace2 Client Address;
- o uses its own port number as the Client Port #; and,
- o decreases # Hops by ((number of the Standard Response Blocks that were just returned in a Reply) - 1). The "-1" in this expression accounts for the additional Standard Response Block appended by the gateway router.

The new Mtrace2 Query message is then sent to the upstream router or to an appropriate multicast address on the RPF interface.

When the gateway receives an Mtrace2 Reply whose Query ID matches the one in the original Mtrace2 header, it MUST relay the Mtrace2 Reply back to the Mtrace2 client by replacing the Reply's header with the original Mtrace2 header. If the gateway does not receive the corresponding Mtrace2 Reply within the [Mtrace Reply Timeout] period (see Section 5.8.4), then it silently discards the original Mtrace2 Query or Request message, and terminates the trace.

4.6. Hiding Information

Information about a domain's topology and connectivity may be hidden from the Mtrace2 Requests. The Forwarding Code of INFO_HIDDEN may be used to note that. For example, the incoming interface address and packet count on the ingress router of a domain, and the outgoing interface address and packet count on the egress router of the domain can be specified as all 1's. Additionally, the source-group packet count (see Section 3.2.4 and Section 3.2.5) within the domain may be all 1's if it is hidden.

5. Client Behavior

This section describes the behavior of an Mtrace2 client in detail.

5.1. Sending Mtrace2 Query

An Mtrace2 client initiates an Mtrace2 Query by sending the Query to the LHR of interest.

5.1.1. Destination Address

If an Mtrace2 client knows the proper LHR, it unicasts an Mtrace2 Query packet to that router; otherwise, it MAY send the Mtrace2 Query packet to the all-routers multicast group (224.0.0.2) for IPv4 or All Routers Address (FF02::2) for IPv6. This will ensure that the packet is received by the LHR on the subnet.

See also Section 5.4 on determining the LHR.

5.1.2. Source Address

An Mtrace2 Query MUST be sent with the client's interface address, which is the Mtrace2 Client Address.

5.2. Determining the Path

An Mtrace2 client could send an initial Query messages with a large # Hops, in order to try to trace the full path. If this attempt fails, one strategy is to perform a linear search (as the traditional unicast traceroute program does); set the # Hops field to 1 and try to get a Reply, then 2, and so on. If no Reply is received at a certain hop, this hop is identified as the probable cause of forwarding failures on the path. Nevertheless, the sender may attempt to continue tracing past the non-responding hop by further increasing the hop count in the hopes that further hops may respond. Each of these attempts MUST NOT be initiated before the previous attempt has terminated either because of successful reception of a Reply or because the [Mtrace Reply Timeout] timeout has occurred.

See also Section 5.6 on receiving the results of a trace.

5.3. Collecting Statistics

After a client has determined that it has traced the whole path or as much as it can expect to (see Section 5.8), it might collect statistics by waiting a short time and performing a second trace. If the path is the same in the two traces, statistics can be displayed as described in Section 7.3 and Section 7.4.

5.4. Last Hop Router (LHR)

The Mtrace2 client may not know which is the last-hop router, or that router may be behind a firewall that blocks unicast packets but passes multicast packets. In these cases, the Mtrace2 Request should be multicasted to the all-routers multicast group (224.0.0.2) for IPv4 or All Routers Address (FF02::2) for IPv6. All routers except

the correct last-hop router SHOULD ignore any Mtrace2 Request received via multicast.

5.5. First Hop Router (FHR)

The IANA assigned 224.0.1.32 as the default multicast group for old IPv4 mtrace (v1) responses, in order to support mtrace clients that are not unicast reachable from the first-hop router. Mtrace2, however, does not require any IPv4/IPv6 multicast addresses for the Mtrace2 Replies. Every Mtrace2 Reply is sent to the unicast address specified in the Mtrace2 Client Address field of the Mtrace2 Reply.

5.6. Broken Intermediate Router

A broken intermediate router might simply not understand Mtrace2 packets, and drop them. The Mtrace2 client will get no Reply at all as a result. It should then perform a hop-by-hop search by setting the # Hops field until it gets an Mtrace2 Reply. The client may use linear or binary search; however, the latter is likely to be slower because a failure requires waiting for the [Mtrace Reply Timeout] period.

5.7. Non-Supported Router

When a non-supported router receives an Mtrace2 Query or Request message whose destination address is a multicast address, the router will silently discard the message.

When the router receives an Mtrace2 Query which is destined to itself, the router returns an Internet Control Message Protocol (ICMP) port unreachable to the Mtrace2 client. On the other hand, when the router receives an Mtrace2 Request which is destined to itself, the router returns an ICMP port unreachable to its adjacent router from which the Request receives. Therefore, the Mtrace2 client needs to terminate the trace when the [Mtrace Reply Timeout] timeout has occurred, and may then issue another Query with a lower number of # Hops.

5.8. Mtrace2 Termination

When performing an expanding hop-by-hop trace, it is necessary to determine when to stop expanding.

5.8.1. Arriving at Source

A trace can be determined to have arrived at the source if the Incoming Interface of the last router in the trace is non-zero, but the Upstream Router is zero.

5.8.2. Fatal Error

A trace has encountered a fatal error if the last Forwarding Error in the trace has the 0x80 bit set.

5.8.3. No Upstream Router

A trace cannot continue if the last Upstream Router in the trace is set to 0.

5.8.4. Reply Timeout

This document defines the [Mtrace Reply Timeout] value, which is used to time out an Mtrace2 Reply as seen in Section 4.5, Section 5.2, and Section 5.7. The default [Mtrace Reply Timeout] value is 10 (seconds), and can be manually changed on the Mtrace2 client and routers.

5.9. Continuing after an Error

When the NO_SPACE error occurs, as described in Section 4.2, a router will send back an Mtrace2 Reply to the Mtrace2 client, and continue with a new Request (see Section 4.3.3). In this case, the Mtrace2 client may receive multiple Mtrace2 Replies from different routers along the path. When this happens, the client MUST treat them as a single Mtrace2 Reply message by collating the augmented response blocks of subsequent Replies sharing the same query ID, sequencing each cluster of augmented response blocks based on the order in which they are received.

If a trace times out, it is very likely that a router in the middle of the path does not support Mtrace2. That router's address will be in the Upstream Router field of the last Standard Response Block in the last received Reply. A client may be able to determine (via mrrinfo or the Simple Network Management Protocol (SNMP) [11][13]) a list of neighbors of the non-responding router. The neighbors obtained in this way could then be probed (via the multicast MIB [13]) to determine which one is the upstream neighbor (i.e., Reverse Path Forwarding (RPF) neighbor) of the non-responding router. This algorithm can identify the upstream neighbor because, even though there may be multiple neighbors, the non-responding router should only have sent a "join" to the one neighbor corresponding to its selected RPF path. Because of this, only the RPF neighbor should contain the non-responding router as a multicast next hop in its MIB output list for the affected multicast route.

6. Protocol-Specific Considerations

This section describes the Mtrace2 behavior with the presence of different multicast protocols.

6.1. PIM-SM

When an Mtrace2 reaches a PIM-SM RP, and the RP does not forward the trace on, it means that the RP has not performed a source-specific join so there is no more state to trace. However, the path that traffic would use if the RP did perform a source-specific join can be traced by setting the trace destination to the RP, the trace source to the traffic source, and the trace group to 0. This Mtrace2 Query may be unicasted to the RP, and the RP takes the same actions as an LHR.

6.2. Bi-Directional PIM

Bi-directional PIM [6] is a variant of PIM-SM that builds bi-directional shared trees connecting multicast sources and receivers. Along the bi-directional shared trees, multicast data is natively forwarded from the sources to the Rendezvous Point Link (RPL), and from which, to receivers without requiring source-specific state. In contrast to PIM-SM, Bi-directional PIM always has the state to trace.

A Designated Forwarder (DF) for a given Rendezvous Point Address (RPA) is in charge of forwarding downstream traffic onto its link, and forwarding upstream traffic from its link towards the RPL that the RPA belongs to. Hence Mtrace2 Reply reports DF addresses or RPA along the path.

6.3. PIM-DM

Routers running PIM Dense Mode [15] do not know the path packets would take unless traffic is flowing. Without some extra protocol mechanism, this means that in an environment with multiple possible paths with branch points on shared media, Mtrace2 can only trace existing paths, not potential paths. When there are multiple possible paths but the branch points are not on shared media, the upstream router is known, but the LHR may not know that it is the appropriate last hop.

When traffic is flowing, PIM Dense Mode routers know whether or not they are the LHR for the link (because they won or lost an Assert battle) and know who the upstream router is (because it won an Assert battle). Therefore, Mtrace2 is always able to follow the proper path when traffic is flowing.

6.4. IGMP/MLD Proxy

When an IGMP or Multicast Listener Discovery (MLD) Proxy [7] receives an Mtrace2 Query packet on an incoming interface, it notes a `WRONG_IF` in the Forwarding Code of the last Standard Response Block (see Section 3.2.4), and sends the Mtrace2 Reply back to the Mtrace2 client. On the other hand, when an Mtrace2 Query packet reaches an outgoing interface of the IGMP/MLD proxy, it is forwarded onto its incoming interface towards the upstream router.

7. Problem Diagnosis

This section describes different scenarios Mtrace2 can be used to diagnose the multicast problems.

7.1. Forwarding Inconsistencies

The Forwarding Error code can tell if a group is unexpectedly pruned or administratively scoped.

7.2. TTL or Hop Limit Problems

By taking the maximum of hops from the source and forwarding TTL threshold over all hops, it is possible to discover the TTL or hop limit required for the source to reach the destination.

7.3. Packet Loss

By taking multiple traces, it is possible to find packet loss information by tracking the difference between the output packet count for the specified source-group address pair at a given upstream router and the input packet count on the next hop downstream router. On a point-to-point link, any steadily increasing difference in these counts implies packet loss. Although the packet counts will differ due to Mtrace2 Request propagation delay, the difference should remain essentially constant (except for jitter caused by differences in propagation time among the trace iterations). However, this difference will display a steady increase if packet loss is occurring. On a shared link, the count of input packets can be larger than the number of output packets at the previous hop, due to other routers or hosts on the link injecting packets. This appears as "negative loss" which may mask real packet loss.

In addition to the counts of input and output packets for all multicast traffic on the interfaces, the Standard Response Block includes a count of the packets forwarded by a node for the specified source-group pair. Taking the difference in this count between two traces and then comparing those differences between two hops gives a

measure of packet loss just for traffic from the specified source to the specified receiver via the specified group. This measure is not affected by shared links.

On a point-to-point link that is a multicast tunnel, packet loss is usually due to congestion in unicast routers along the path of that tunnel. On native multicast links, loss is more likely in the output queue of one hop, perhaps due to priority dropping, or in the input queue at the next hop. The counters in the Standard Response Block do not allow these cases to be distinguished. Differences in packet counts between the incoming and outgoing interfaces on one node cannot generally be used to measure queue overflow in the node.

7.4. Link Utilization

Again, with two traces, you can divide the difference in the input or output packet counts at some hop by the difference in time stamps from the same hop to obtain the packet rate over the link. If the average packet size is known, then the link utilization can also be estimated to see whether packet loss may be due to the rate limit or the physical capacity on a particular link being exceeded.

7.5. Time Delay

If the routers have synchronized clocks, it is possible to estimate propagation and queuing delay from the differences between the timestamps at successive hops. However, this delay includes control processing overhead, so is not necessarily indicative of the delay that data traffic would experience.

8. IANA Considerations

The following new registries are to be created and maintained under the "Specification Required" registry policy as specified in [4].

8.1. "Mtrace2 Forwarding Codes" Registry

This is an integer in the range 0-255. Assignment of a Forwarding Code requires specification of a value and a name for the Forwarding Code. Initial values for the forwarding codes are given in the table at the end of Section 3.2.4. Additional values (specific to IPv6) may also be specified at the end of Section 3.2.5. Any additions to this registry are required to fully describe the conditions under which the new Forwarding Code is used.

8.2. "Mtrace2 TLV Types" Registry

Assignment of a TLV Type requires specification of an integer value "Code" in the range 0-255 and a name ("Type"). Initial values for the TLV Types are given in the table at the beginning of Section 3.2.

8.3. UDP Destination Port

IANA has assigned UDP user port 33435 (mtrace) for use by this protocol as the Mtrace2 UDP destination port.

9. Security Considerations

This section addresses some of the security considerations related to Mtrace2.

9.1. Addresses in Mtrace2 Header

An Mtrace2 header includes three addresses, source address, multicast address, and Mtrace2 client address. These addresses MUST be congruent with the definition defined in Section 3.2.1 and forwarding Mtrace2 messages having invalid addresses MUST be prohibited. For instance, if Mtrace2 Client Address specified in an Mtrace2 header is a multicast address, then a router that receives the Mtrace2 message MUST silently discard it.

9.2. Verification of Clients and Peers

A router providing Mtrace2 functionality MUST support a source verification mechanism to drop Queries from clients and Requests from peer router or client addresses that are unauthorized or that are beyond a specified administrative boundary. This verification could, for example, be specified via a list of allowed/disallowed client and peer addresses or subnets for a given Mtrace2 message type sent to the Mtrace2 protocol port. If a Query or Request is received from an unauthorized address or one beyond the specified administrative boundary, the Query/Request MUST NOT be processed. The router MAY, however, perform rate limited logging of such events.

The required use of source verification on the participating routers minimizes the possible methods for introduction of spoofed Query/Request packets that would otherwise enable DoS amplification attacks targeting an authorized "query" host. The source verification mechanisms provide this protection by allowing Query messages from an authorized host address to be received only by the router(s) connected to that host, and only on the interface to which that host is attached. For protection against spoofed Request messages, the source verification mechanisms allow Request messages only from a

directly connected routing peer and allow these messages to be received only on the interface to which that peer is attached.

Note that the following vulnerabilities cannot be covered by the source verification methods described here. These methods can, nevertheless, prevent attacks launched from outside the boundaries of a given network as well as from any hosts within the network that are not on the same LAN as an intended authorized query client.

- o A server/router "B" other than the server/router "A" that actually "owns" a given IP address could, if it is connected to the same LAN, send an Mtrace2 Query or Request with the source address set to the address for server/router "A". This is not a significant threat, however, if only trusted servers and routers are connected to that LAN.
- o A malicious application running on a trusted server or router could send packets that might cause an amplification problem. It is beyond the scope of this document to protect against a DoS attack launched from the same host that is the target of the attack or from another "on path" host, but this is not a likely threat scenario. In addition, routers on the path MAY rate-limit the packets as specified in Section 9.5 and Section 9.6.

9.3. Topology Discovery

Mtrace2 can be used to discover any actively-used topology. If your network topology is a secret, Mtrace2 may be restricted at the border of your domain, using the ADMIN_PROHIB forwarding code.

9.4. Characteristics of Multicast Channel

Mtrace2 can be used to discover what sources are sending to what groups and at what rates. If this information is a secret, Mtrace2 may be restricted at the border of your domain, using the ADMIN_PROHIB forwarding code.

9.5. Limiting Query/Request Rates

A router may limit Mtrace2 Queries and Requests by ignoring some of the consecutive messages. The router MAY randomly ignore the received messages to minimize the processing overhead, i.e., to keep fairness in processing queries, or prevent traffic amplification. The rate limit is left to the router's implementation.

9.6. Limiting Reply Rates

The proxying and NO_SPACE behaviors may result in one Query returning multiple Reply messages. In order to prevent abuse, the routers in the traced path MAY need to rate-limit the Replies. The rate limit function is left to the router's implementation.

9.7. Specific Security Concerns

9.7.1. Request and Response Bombardment

A malicious sender could generate invalid and undesirable Mtrace2 traffic to hosts and/or routers on a network by eliciting responses to spoofed or multicast client addresses. This could be done via forged or multicast client/source addresses in Mtrace2 Query or Request messages. The recommended protections against this type of attack are described in Section 9.1, Section 9.2, Section 9.5, and Section 9.6.

9.7.2. Amplification Attack

Because an Mtrace2 Query results in Mtrace2 Request and Mtrace2 Reply messages that are larger than the original message, the potential exists for an amplification attack from a malicious sender. This threat is minimized by restricting the set of addresses from which Mtrace2 messages can be received on a given router as specified in Section 9.2.

In addition, for a router running a PIM protocol (PIM-SM, PIM-DM, PIM Source-Specific Multicast, or Bi-Directional PIM), the router SHOULD drop any Mtrace2 Request or Reply message that is received from an IP address that does not correspond to an authenticated PIM neighbor on the interface from which the packet is received. The intent of this text is to prevent non-router endpoints from injecting Request messages. Implementations of non-PIM protocols SHOULD employ some other mechanism to prevent this attack.

9.7.3. Leaking of Confidential Topology Details

Mtrace2 Queries are a potential mechanism for obtaining confidential topology information for a targeted network. Section 9.2 and Section 9.4 describe required and optional methods for ensuring that information delivered with Mtrace2 messages is not disseminated to unauthorized hosts.

9.7.4. Delivery of False Information (Forged Reply Messages)

Forged Reply messages could potentially provide a host with invalid or incorrect topology information. They could also provide invalid or incorrect information regarding multicast traffic statistics, multicast stream propagation delay between hops, multicast and unicast protocols in use between hops and other information used for analyzing multicast traffic patterns and for troubleshooting multicast traffic problems. This threat is mitigated by the following factors:

- o The required source verification of permissible source addresses specified in Section 9.2 eliminates the origination of forged Replies from addresses that have not been authorized to send Mtrace2 messages to routers on a given network. This mechanism can block forged Reply messages sent from any "off path" source.
- o To forge a Reply, the sender would need to somehow know (or guess) the associated two byte Query ID for an extant Query and the dynamically allocated source port number. Because "off path" sources can be blocked by a source verification mechanism, the scope of this threat is limited to "on path" attackers.
- o The required use of source verification (Section 9.2) and recommended use of PIM neighbor authentication (Section 9.7.2) for messages that are only valid when sent by a multicast routing peer (Request and Reply messages) eliminate the possibility of reception of a forged Reply from an authorized host address that does not belong to a multicast peer router.
- o The use of encryption between the source of a Query and the endpoint of the trace would provide a method to protect the values of the Query ID and the dynamically allocated client (source) port (see Section 3.2.1). These are the values needed to create a forged Reply message that would pass validity checks at the querying client. This type of cryptographic protection is not practical, however, because the primary reason for executing an Mtrace2 is that the destination endpoint (and path to that endpoint) are not known by the querying client. While it is not practical to provide cryptographic protection between a client and the Mtrace2 endpoints (destinations), it may be possible to prevent forged responses from "off path" nodes attached to any Mtrace2 transit LAN by devising a scheme to encrypt the critical portions of an Mtrace2 message between each valid sender/receiver pair at each hop to be used for multicast/mtrace transit. The use of encryption protection between nodes is, however, out of the scope of this document.

10. Acknowledgements

This specification started largely as a transcription of Van Jacobson's slides from the 30th IETF, and the implementation in mroute 3.3 by Ajit Thyagarajan. Van's original slides credit Steve Casner, Steve Deering, Dino Farinacci and Deb Agrawal. The original multicast traceroute client, mtrace (version 1), has been implemented by Ajit Thyagarajan, Steve Casner and Bill Fenner. The idea of the "S" bit to allow statistics for a source subnet is due to Tom Pusateri.

For the Mtrace version 2 specification, the authors would like to give special thanks to Tatsuya Jinmei, Bill Fenner, and Steve Casner. Also, extensive comments were received from David L. Black, Ronald Bonica, Yiqun Cai, Liu Hui, Bharat Joshi, Robert Kebler, John Kristoff, Mankamana Mishra, Heidi Ou, Eric Rescorla, Pekka Savola, Shinsuke Suzuki, Dave Thaler, Achmad Husni Thamrin, Stig Venaas, Cao Wei, and the Mboned working group members.

11. References

11.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to indicate requirement levels", RFC 2119, March 1997.
- [2] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 8200, July 2017.
- [3] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [4] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 8126, June 2017.
- [5] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 7761, March 2016.
- [6] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, October 2007.

- [7] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 2006.

11.2. Informative References

- [8] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and A. Thyagarajan, "Internet Group Management Protocol, Version 3", RFC 3376, October 2002.
- [9] Bumgardner, G., "Automatic Multicast Tunneling", RFC 7450, February 2015.
- [10] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.
- [11] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [12] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [13] McWalter, D., Thaler, D., and A. Kessler, "IP Multicast MIB", RFC 5132, December 2007.
- [14] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.
- [15] Adams, A., Nicholas, J., and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)", RFC 3973, January 2005.

Authors' Addresses

Hitoshi Asaeda
National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi
Koganei, Tokyo 184-8795
Japan

Email: asaeda@nict.go.jp

Kerry Meyer

Email: kerry.meyer@me.com

Internet-Draft

Mtrace2

July 2018

WeeSan Lee (editor)

Email: weesan@weesan.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 8, 2011

T. Chown
University of Southampton
M. Eubanks
Iformata Communications
R. Parekh
G. Van de Velde
S. Venaas
cisco Systems
April 6, 2011

Multicast Addresses for Documentation
draft-venaas-mboned-mcaddrdoc-04.txt

Abstract

This document discusses which multicast addresses should be used for documentation purposes and reserves multicast addresses for such use. Some multicast addresses are derived from AS numbers or unicast addresses. This document also explains how these can be used for documentation purposes.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 8, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. IPv4 multicast documentation addresses	4
2.1. Administratively scoped IPv4 multicast addresses	4
2.2. GLOP multicast addresses	4
2.3. Unicast prefix based IPv4 multicast addresses	4
3. IPv6 multicast documentation addresses	5
3.1. Unicast prefix based IPv6 multicast addresses	5
3.2. Embedded-RP IPv6 multicast addresses	5
4. Security Considerations	6
5. IANA Considerations	7
6. Acknowledgments	8
7. Informative References	9
Authors' Addresses	10

1. Introduction

It is often useful in documentation, IETF documents, etc., to provide examples containing IP multicast addresses. For documentation where examples of general purpose multicast addresses are needed, one should use multicast addresses that never will be assigned or in actual use. There is a risk that addresses used in examples may accidentally be used. It is then important that the same addresses are not used by other multicast applications or services. It may also be beneficial to filter out such addresses from multicast signalling and multicast data sent to such addresses.

For unicast there are both IPv4 and IPv6 addresses reserved for this purpose, see [RFC5737] and [RFC3849] respectively. This document reserves multicast addresses for this purpose.

There are also some multicast addresses that are derived from AS numbers or unicast addresses. For examples where such addresses are desired, one should derive them from the AS numbers and unicast addresses reserved for documentation purposes. This document also discusses the use of these.

2. IPv4 multicast documentation addresses

The type of multicast addresses most commonly used today, are addresses used for so-called ASM (Any-Source Multicast). For ASM, the IPv4 multicast addresses allocated for documentation purposes are 233.252.0.0 - 233.252.0.255 (233.252.0.0/24).

Another type of multicast is SSM (Source-Specific Multicast). For SSM it is less important which multicast addresses are used, since a host/application joins a channel identified by both source and group. Any source addresses used in SSM examples should be unicast addresses reserved for documentation purposes, see [RFC5737].

Sometimes one wants to give examples where a specific type of address is desired. E.g. for text about multicast scoping, one might want the examples to use addresses that are to be used for administrative scoping. See below for guidance on how to construct specific types of example addresses.

2.1. Administratively scoped IPv4 multicast addresses

Administratively scoped IPv4 multicast addresses [RFC2365] are reserved for scoped multicast. They can be used within a site or an organization. Apart from a small set of scope relative addresses, these addresses are not assigned. There are no specific scoped addresses available for documentation purposes. Except for examples detailing the use of scoped multicast, one should avoid using them.

2.2. GLOP multicast addresses

GLOP [RFC3180] is a method for deriving IPv4 multicast group addresses from 16 bit AS numbers. For examples where GLOP addresses are desired, the addresses should be derived from the AS numbers reserved for documentation use. See [RFC5398].

2.3. Unicast prefix based IPv4 multicast addresses

IPv4 multicast addresses can be derived from IPv4 unicast prefixes, see [RFC6034]. For examples where this type of addresses are desired, the addresses should be derived from the unicast addresses reserved for documentation purposes, see [RFC5737].

3. IPv6 multicast documentation addresses

The type of multicast addresses most commonly used today, are addresses used for so-called ASM (Any-Source Multicast). For ASM, the IPv6 multicast addresses allocated for documentation purposes are TBD.

Another type of multicast is SSM (Source-Specific Multicast). For SSM it is less important which multicast addresses are used, since a host/application joins a channel identified by both source and group. Any source addresses used in SSM examples should be unicast addresses reserved for documentation purposes, see [RFC3849].

Sometimes one wants to give examples where a specific type of address is desired. E.g. for text about multicast scoping, one might want the examples to use addresses that are to be used for administrative scoping. See below for guidance on how to construct specific types of example addresses.

3.1. Unicast prefix based IPv6 multicast addresses

IPv6 multicast addresses can be derived from IPv6 unicast prefixes, see [RFC3306]. For examples where this type of addresses is desired, the addresses should be derived from the unicast addresses reserved for documentation purposes, see [RFC3849].

3.2. Embedded-RP IPv6 multicast addresses

There is a type of IPv6 multicast addresses called Embedded-RP addresses where the IPv6 address of a Rendezvous-Point is embedded inside the multicast address, see [RFC3956]. For examples where this type of addresses is desired, the addresses should be derived from the unicast addresses reserved for documentation purposes, see see [RFC3849].

4. Security Considerations

The use of specific multicast addresses for documentation purposes has no impact on security.

5. IANA Considerations

IANA is requested to assign "variable scope" IPv6 multicast addresses for documentation purposes. This should be a /96 prefix of the form FF0X:...

6. Acknowledgments

The authors thank Roberta Maglione for providing comments on this document.

7. Informative References

- [RFC2365] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, July 1998.
- [RFC3180] Meyer, D. and P. Lothberg, "GLOP Addressing in 233/8", BCP 53, RFC 3180, September 2001.
- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, August 2002.
- [RFC3307] Haberman, B., "Allocation Guidelines for IPv6 Multicast Addresses", RFC 3307, August 2002.
- [RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, November 2004.
- [RFC5398] Huston, G., "Autonomous System (AS) Number Reservation for Documentation Use", RFC 5398, December 2008.
- [RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010.
- [RFC6034] Thaler, D., "Unicast-Prefix-Based IPv4 Multicast Addresses", RFC 6034, October 2010.

Authors' Addresses

Tim Chown
University of Southampton
Highfield
Southampton, Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

Marshall Eubanks
Iformata Communications
130 W. Second Street
Dayton, Ohio 45402
US

Phone: +1 703 501 4376
Email: marshall.eubanks@iformata.com
URI: <http://www.iformata.com/>

Rishabh Parekh
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: riparekh@cisco.com

Gunter Van de Velde
cisco Systems
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 476 476 022
Email: gvandeve@cisco.com

Stig Venaas
cisco Systems
Tasman Drive
San Jose, CA 95134
USA

Email: stig@cisco.com

