Stream: Internet Engineering Task Force (IETF)
RFC: 9630
Category: Standards Track
Published: August 2024
ISSN: 2070-1721
Authors: H. Song, *Futurewei Technologies*; M. McBride, *Futurewei Technologies*; G. Mirsky, *Ericsson*; G. Mishra, *Verizon Inc.*; H. Asaeda, *NICT*; T. Zhou, *Huawei Technologies*

RFC 9630
Multicast On-Path Telemetry Using In Situ Operations, Administration, and Maintenance (IOAM)

Abstract

This document specifies two solutions to meet the requirements of on-path telemetry for multicast traffic using IOAM. While IOAM is advantageous for multicast traffic telemetry, some unique challenges are present. This document provides the solutions based on the IOAM trace option and direct export option to support the telemetry data correlation and the multicast tree reconstruction without incurring data redundancy.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9630.

Copyright Notice

Table of Contents

1. Introduction

IP multicast has had many useful applications for several decades. [MULTICAST-LESSONS-LEARNED] provides a thorough historical perspective about the design and deployment of many of the multicast routing protocols in use with various applications. We will mention of few of

these throughout this document and in the Application Considerations section (Section 5). IP multicast has been used by residential broadband customers across operator networks, private MPLS customers, and internal customers within corporate intranets. IP multicast has provided real-time interactive online meetings or podcasts, IPTV, and financial markets' real-time data, all of which rely on UDP's unreliable transport. End-to-end QoS, therefore, should be a critical component of multicast deployments in order to provide a good end-user experience within a specific operational domain. In multicast real-time media streaming, if a single packet is lost within a keyframe and cannot be recovered using forward error correction, many receivers will be unable to decode subsequent frames within the Group of Pictures (GoP), which results in video freezes or black pictures until another keyframe is delivered. Unexpectedly long delays in delivery of packets can cause timeouts with similar results. Multicast packet loss and delays can therefore affect application performance and the user experience within a domain.

It is essential to monitor the performance of multicast traffic. New on-path telemetry techniques, such as IOAM [RFC9197], IOAM Direct Export (DEX) [RFC9326], IOAM Postkcard-Based Telemetry - Marking (PBT-M) [POSTCARD-TELEMETRY], and Hybrid Two-Step (HTS) [HYBRID-TWO-STEP], complement existing active OAM performance monitoring methods like ICMP ping [RFC0792]. However, multicast traffic's unique characteristics present challenges in applying these techniques efficiently.

The IP multicast packet data for a particular (S, G) state remains identical across different branches to multiple receivers. When IOAM trace data is added to multicast packets, each replicated packet retains telemetry data for its entire forwarding path. This results in redundant data collection for common path segments, unnecessarily consuming extra network bandwidth. For large multicast trees, this redundancy is substantial. Using solutions like IOAM DEX could be more efficient by eliminating data redundancy, but IOAM DEX lacks a branch identifier, complicating telemetry data correlation and multicast tree reconstruction.

This document provides two solutions to the IOAM data-redundancy problem based on the IOAM standards. The requirements for multicast traffic telemetry are discussed along with the issues of the existing on-path telemetry techniques. We propose modifications and extensions to make these techniques adapt to multicast in order for the original multicast tree to be correctly reconstructed while eliminating redundant data. This document does not cover the operational considerations such as how to enable the telemetry on a subset of the traffic to avoid overloading the network or the data collector.

1.1. Requirements Language

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Requirements for Multicast Traffic Telemetry

Multicast traffic is forwarded through a multicast tree. With PIM [RFC7761] and Point-to-Multipoint (P2MP), the forwarding tree is established and maintained by the multicast routing protocol.

The requirements for multicast traffic telemetry that are addressed by the solutions in this document are:

- Reconstruct and visualize the multicast tree through data-plane monitoring.
- Gather the multicast packet delay and jitter performance on each path.
- Find the multicast packet-drop location and reason.

In order to meet all of these requirements, we need the ability to directly monitor the multicast traffic and derive data from the multicast packets. The conventional OAM mechanisms, such as multicast ping [RFC6450], trace [RFC8487], and RTCP [RFC3605], are not sufficient to meet all of these requirements. The telemetry methods in this document meet these requirements by providing granular hop-by-hop network monitoring along with the reduction of data redundancy.

3. Issues of Existing Techniques

On-path telemetry techniques that directly retrieve data from multicast traffic's live network experience are ideal for addressing the aforementioned requirements. The representative techniques include IOAM Trace option [RFC9197], IOAM DEX option [RFC9326], and PBT-M [POSTCARD-TELEMETRY]. However, unlike unicast, multicast poses some unique challenges to applying these techniques.

Multicast packets are replicated at each branch fork node in the corresponding multicast tree. Therefore, there are multiple copies of the original multicast packet in the network.

When the IOAM trace option is utilized for on-path data collection, partial trace data is replicated into the packet copy for each branch of the multicast tree. Consequently, at the leaves of the multicast tree, each copy of the multicast packet contains a complete trace. This results in data redundancy, as most of the data (except from the final leaf branch) appears in multiple copies, where only one is sufficient. This redundancy introduces unnecessary header overhead, wastes network bandwidth, and complicates data processing. The larger the multicast tree or the longer the multicast path, the more severe the redundancy problem becomes.

The postcard-based solutions (e.g., IOAM DEX) can eliminate data redundancy because each node on the multicast tree sends a postcard with only local data. However, these methods cannot accurately track and correlate the tree branches due to the absence of branching information. For instance, in the multicast tree shown in Figure 1, Node B has two branches, one to Node C and the other to node D; further, Node C leads to Node E, and Node D leads to Node F (not shown). When applying postcard-based methods, it is impossible to determine whether Node E is

the next hop of Node C or Node D from the received postcards alone, unless one correlates the exporting nodes with knowledge about the tree collected by other means (e.g., mtrace). Such correlation is undesirable because it introduces extra work and complexity.

The fundamental reason for this problem is that there is not an identifier (either implicit or explicit) to correlate the data on each branch.

4. Modifications and Extensions Based on Existing Solutions

We provide two solutions to address the above issues. One is based on IOAM DEX and requires an extension to the instruction header of the IOAM DEX Option. The second solution combines the IOAM trace option and postcards for redundancy removal.

4.1. Per-Hop Postcard Using IOAM DEX

One way to mitigate the postcard-based telemetry's tree-tracking weakness is to augment it with a branch identifier field. This works for the IOAM DEX option because the IOAM DEX option has an instruction header which can be used to hold the branch identifier. To make the branch identifier globally unique, the branch fork node ID plus an index is used. For example, as shown in Figure 1, Node B has two branches: one to Node C and the other to Node D. Node B may use [B, 0] as the branch identifier for the branch to C, and [B, 1] for the branch to D. The identifier is carried with the multicast packet until the next branch fork node. Each node **MUST** export the branch identifier in the received IOAM DEX header in the postcards it sends. The branch identifier, along with the other fields such as Flow ID and Sequence Number, is sufficient for the data collector to reconstruct the topology of the multicast tree.

Figure 1 shows an example of this solution. "P" stands for the postcard packet. The square brackets contains the branch identifier. The curly braces contain the telemetry data about a specific node.

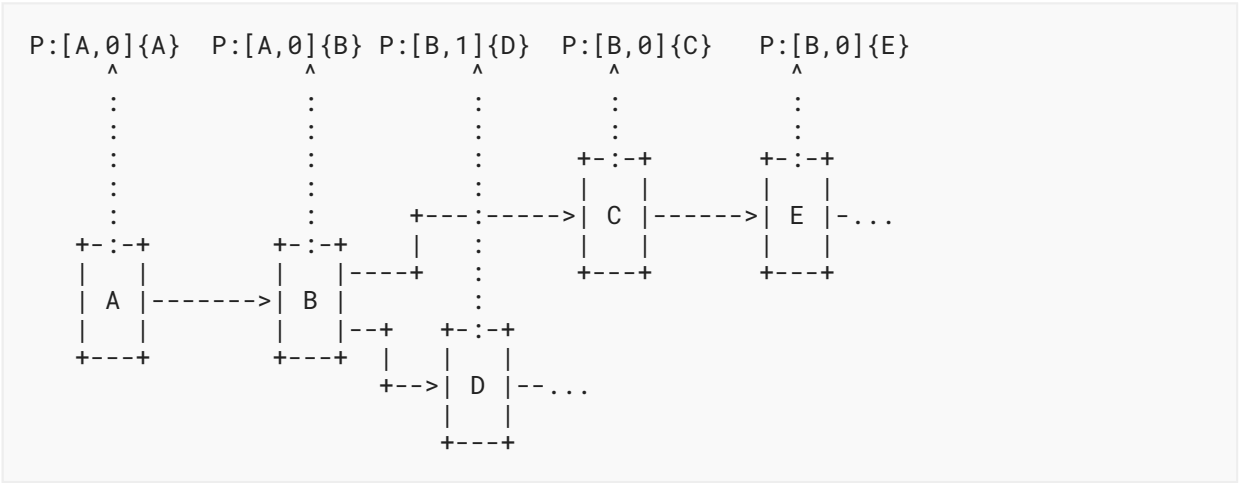

Figure 1: Per-Hop Postcard

Each branch fork node needs to generate a unique branch identifier (i.e., branch ID) for each branch in its multicast tree instance and include it in the IOAM DEX option header. The branch ID remains unchanged until the next branch fork node. The branch ID contains two parts: the branch fork node ID and an interface index.

Conforming to the node ID specification in IOAM [RFC9197], the node ID is a 3-octet unsigned integer. The interface index is a two-octet unsigned integer. As shown in Figure 2, the branch ID consumes 8 octets in total. The three unused octets **MUST** be set to 0; otherwise, the header is considered malformed and the packet **MUST** be dropped.

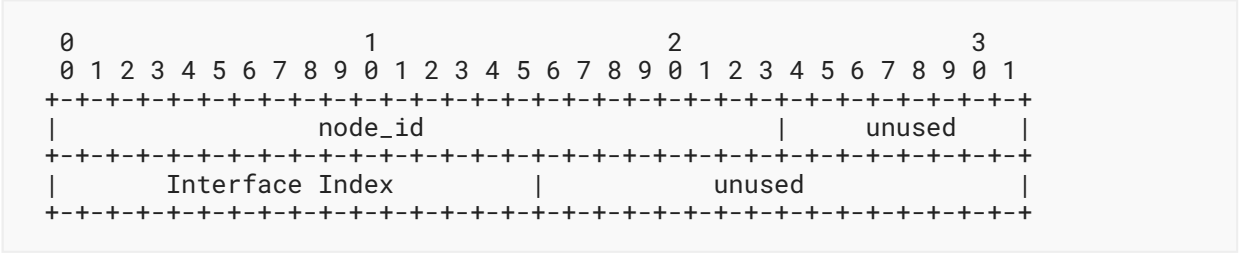

Figure 2: Multicast Branch ID Format

Figure 3 shows that the branch ID is carried as an optional field after the Flow ID and Sequence Number optional fields in the IOAM DEX option header. Two bits "N" and "I" (i.e., the third and fourth bits in the Extension-Flags field) are reserved to indicate the presence of the optional branch ID field. "N" stands for the Node ID, and "I" stands for the interface index. If "N" and "I" are both set to 1, the optional multicast branch ID field is present. Two Extension-Flag bits are used because [RFC9326] specifies that each extension flag only indicates the presence of a 4-octet optional data field, while we need more than 4 octets to encode the branch ID. The two flag bits **MUST** be both set or cleared; otherwise, the header is considered malformed and the packet **MUST** be dropped.

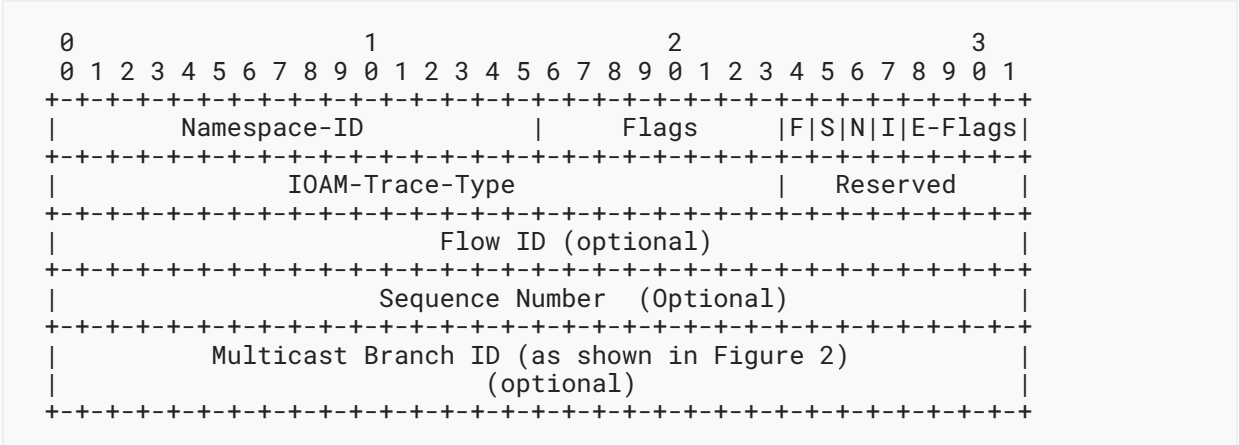

Figure 3: Carrying the Branch ID in IOAM DEX Option Header

Once a node gets the branch ID information from the upstream node, it **MUST** carry this information in its telemetry data export postcards so the original multicast tree can be correctly reconstructed based on the postcards.

4.2. Per-Section Postcard for IOAM Trace

The second solution is a combination of the IOAM trace option [RFC9197] and the postcard-based telemetry [IFIT-FRAMEWORK]. To avoid data redundancy, at each branch fork node, the trace data accumulated up to this node is exported by a postcard before the packet is replicated. In this solution, each branch also needs to maintain some identifier to help correlate the postcards for each tree section. The natural way to accomplish this is to simply carry the branch fork node's data (including its ID) in the trace of each branch. This is also necessary because each replicated multicast packet can have different telemetry data pertaining to this particular copy (e.g., node delay, egress timestamp, and egress interface). As a consequence, the local data exported by each branch fork node can only contain the common data shared by all the replicated packets (e.g., ingress interface and ingress timestamp).

Figure 4 shows an example in a segment of a multicast tree. Node B and D are two branch fork nodes, and they will export a postcard covering the trace data for the previous section. The end node of each path will also need to export the data of the last section as a postcard.

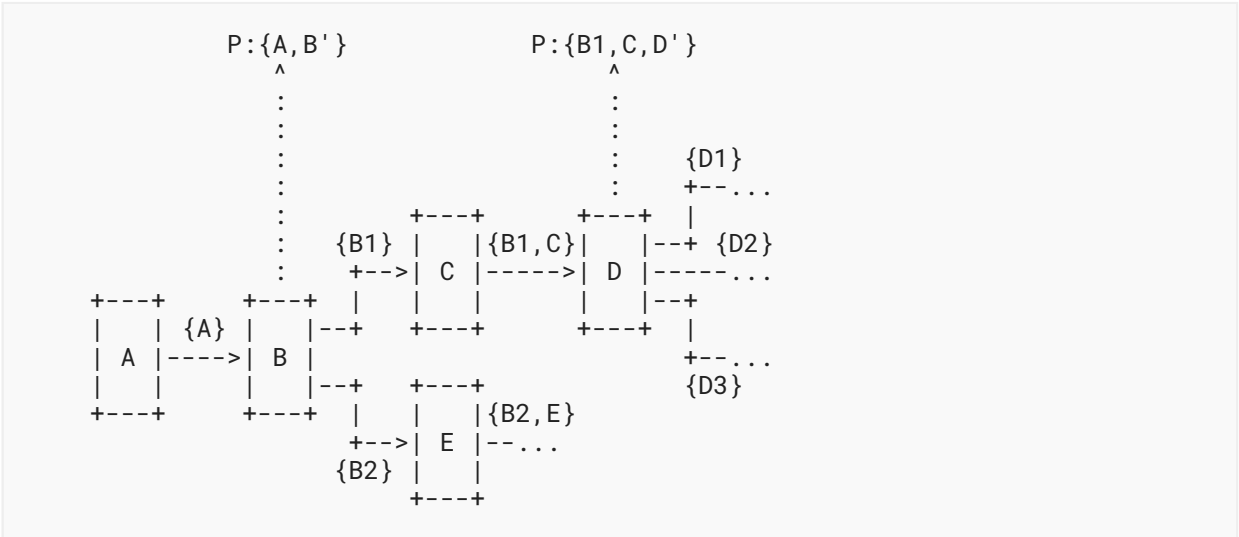

Figure 4: Per-Section Postcard

There is no need to modify the IOAM trace option header format as specified in [RFC9197]. We just need to configure the branch fork nodes, as well as the leaf nodes, to export the postcards that contain the trace data collected so far and refresh the IOAM header and data in the packet (e.g., clear the node data list to all zeros and reset the Remaining Length field to the initial value).

5. Application Considerations for Multicast Protocols

5.1. Mtrace Version 2

Mtrace version 2 (Mtrace2) [RFC8487] is a protocol that allows the tracing of an IP multicast routing path. Mtrace2 provides additional information such as the packet rates and losses, as well as other diagnostic information. Unlike unicast traceroute, Mtrace2 traces the path that the tree-building messages follow from the receiver to the source. An Mtrace2 client sends an Mtrace2 Query to a Last-Hop Router (LHR), and the LHR forwards the packet as an Mtrace2 Request towards the source or a Rendezvous Point (RP) after appending a response block. Each router along the path proceeds with the same operations. When the First-Hop Router (FHR) receives the Request packet, it appends its own response block, turns the Request packet into a Reply, and unicasts the Reply back to the Mtrace2 client.

New on-path telemetry techniques will enhance Mtrace2, and other existing OAM solutions, with more granular and real-time network status data through direct measurements. There are various multicast protocols that are used to forward the multicast data. Each will require its own unique on-path telemetry solution. Mtrace2 doesn't integrate with IOAM directly, but network management systems may use Mtrace2 to learn about routers of interest.

5.2. Application in PIM

PIM - Sparse Mode (PIM-SM) [RFC7761] is the most widely used multicast routing protocol deployed today. PIM - Source-Specific Multicast (PIM-SSM), however, is the preferred method due to its simplicity and removal of network source discovery complexity. With PIM, control plane state is established in the network in order to forward multicast UDP data packets. PIM utilizes network-based source discovery. PIM-SSM, however, utilizes application-based source discovery. IP multicast packets fall within the range of 224.0.0.0 through 239.255.255.255 for IPv4 and ff00::/8 for IPv6. The telemetry solution will need to work within these IP address ranges and provide telemetry data for this UDP traffic.

A proposed solution for encapsulating the telemetry instruction header and metadata in IPv6 packets is described in [RFC9486].

5.3. Application of MVPN X-PMSI Tunnel Encapsulation Attribute

IOAM, and the recommendations of this document, are equally applicable to multicast MPLS forwarded packets. Multipoint Label Distribution Protocol (mLDP), P2MP RSVP-TE, Ingress Replication (IR), and PIM Multicast Distribution Tree (MDT) SAFI with GRE Transport are all commonly used within a Multicast VPN (MVPN) environment utilizing MVPN procedures such as multicast in MPLS/BGP IP VPNs [RFC6513] and BGP encoding and procedures for multicast in MPLS/BGP IP VPNs [RFC6514]. mLDP LDP extensions for P2MP and multipoint-to-multipoint (MP2MP) label switched paths (LSPs) [RFC6388] provide extensions to LDP to establish point-to-

multipoint (P2MP) and MP2MP LSPs in MPLS networks. The telemetry solution will need to be able to follow these P2MP and MP2MP paths. The telemetry instruction header and data should be encapsulated into MPLS packets on P2MP and MP2MP paths.

6. Security Considerations

The schemes discussed in this document share the same security considerations for the IOAM trace option [RFC9197] and the IOAM DEX option [RFC9326]. In particular, since multicast has a built-in nature for packet amplification, the possible amplification risk for the DEX-based scheme is greater than the case of unicast. Hence, stricter mechanisms for protections need to be applied. In addition to selecting packets to enable DEX and to limit the exported traffic rate, we can also allow only a subset of the nodes in a multicast tree to process the option and export the data (e.g., only the branching nodes in the multicast tree are configured to process the option).

7. IANA Considerations

IANA has registered two Extension-Flags, as described in Section 4.1, in the "IOAM DEX Extension-Flags" registry.

Bit	Description	Reference
2	Multicast Branching Node ID	This RFC
3	Multicast Branching Interface Index	This RFC

Table 1: IOAM DEX Extension-Flags

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <https://www.rfc-editor.org/info/rfc6388>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

[RFC7761] 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)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <https://www.rfc-editor.org/info/rfc9197>.

[RFC9326] Song, H., Gafni, B., Brockners, F., Bhandari, S., and T. Mizrahi, "In Situ Operations, Administration, and Maintenance (IOAM) Direct Exporting", RFC 9326, DOI 10.17487/RFC9326, November 2022, <https://www.rfc-editor.org/info/rfc9326>.

8.2. Informative References

[HYBRID-TWO-STEP] Mirsky, G., Lingqiang, W., Zhui, G., Song, H., and P. Thubert, "Hybrid Two-Step Performance Measurement Method", Work in Progress, Internet-Draft, draft-ietf-ippm-hybrid-two-step-01, 5 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-ippm-hybrid-two-step-01>.

[IFIT-FRAMEWORK] Song, H., Qin, F., Chen, H., Jin, J., and J. Shin, "Framework for In-situ Flow Information Telemetry", Work in Progress, Internet-Draft, draft-song-opsawg-ifit-framework-21, 23 October 2023, <https://datatracker.ietf.org/doc/html/draft-song-opsawg-ifit-framework-21>.

[MULTICAST-LESSONS-LEARNED] Farinacci, D., Giuliano, L., McBride, M., and N. Warnke, "Multicast Lessons Learned from Decades of Deployment Experience", Work in Progress, Internet-Draft, draft-ietf-pim-multicast-lessons-learned-04, 22 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-pim-multicast-lessons-learned-04>.

[POSTCARD-TELEMETRY] Song, H., Mirsky, G., Zhou, T., Li, Z., Graf, T., Mishra, G., Shin, J., and K. Lee, "On-Path Telemetry using Packet Marking to Trigger Dedicated OAM Packets", Work in Progress, Internet-Draft, draft-song-ippm-postcard-based-telemetry-16, 2 June 2023, <https://datatracker.ietf.org/doc/html/draft-song-ippm-postcard-based-telemetry-16>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC3605] Huitema, C., "Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)", RFC 3605, DOI 10.17487/RFC3605, October 2003, <https://www.rfc-editor.org/info/rfc3605>.

[RFC6450] Venaas, S., "Multicast Ping Protocol", RFC 6450, DOI 10.17487/RFC6450, December 2011, <https://www.rfc-editor.org/info/rfc6450>.

[RFC8487] Asaeda, H., Meyer, K., and W. Lee, Ed., "Mtrace Version 2: Traceroute Facility for IP Multicast", RFC 8487, DOI 10.17487/RFC8487, October 2018, <https://www.rfc-editor.org/info/rfc8487>.

[RFC9486] Bhandari, S., Ed. and F. Brockners, Ed., "IPv6 Options for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9486, DOI 10.17487/RFC9486, September 2023, <https://www.rfc-editor.org/info/rfc9486>.

Acknowledgments

The authors would like to thank Gunter Van de Velde, Brett Sheffield, Éric Vyncke, Frank Brockners, Nils Warnke, Jake Holland, Dino Farinacci, Henrik Nydell, Zaheduzzaman Sarker, and Toerless Eckert for their comments and suggestions.

Authors' Addresses

Haoyu Song
Futurewei Technologies
2330 Central Expressway
Santa Clara, CA
United States of America
Email: hsong@futurewei.com

Mike McBride
Futurewei Technologies
2330 Central Expressway
Santa Clara, CA
United States of America
Email: mmcbride@futurewei.com

Greg Mirsky
Ericsson
United States of America
Email: gregimirsky@gmail.com

Gyan Mishra
Verizon Inc.
United States of America
Email: gyan.s.mishra@verizon.com

Hitoshi Asaeda
National Institute of Information and Communications Technology
Japan
Email: asaeda@nict.go.jp

Tianran Zhou
Huawei Technologies
Beijing
China
Email: zhoutianran@huawei.com