IPPM Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 April 2026

R. Gandhi, Ed.
Cisco Systems, Inc.
T. Zhou
Huawei
Z. Li
China Mobile
W. Hawkins
University of Cincinnati

5 October 2025

Simple Two-Way Active Measurement Protocol (STAMP) Extensions for Reflecting STAMP Packet IP Headers
draft-ietf-ippm-stamp-ext-hdr-06

Abstract

The Simple Two-Way Active Measurement Protocol (STAMP) and its optional extensions can be used for Edge-To-Edge (E2E) active measurement. In Situ Operations, Administration, and Maintenance (IOAM) data fields can be used for recording and collecting Hop-By-Hop (HBH) and E2E operational and telemetry information. This document extends STAMP to reflect IP headers as well as IPv6 extension headers for HBH and E2E active measurements, for example, using IOAM data fields.

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 8 April 2026.

Table of Contents

1. Introduction

The Simple Two-Way Active Measurement Protocol (STAMP) provides capabilities for the measurement of various performance metrics in IP networks [RFC8762] without the use of a control channel to pre-signal session parameters. [RFC8972] defines optional extensions in the form of TLVs for STAMP. The STAMP test packets are transmitted along a path between a Session-Sender and a Session-Reflector to measure Edge-To-Edge (E2E) performance delay and packet loss along that path.

In Situ Operations, Administration, and Maintenance (IOAM) is used for recording and collecting operational and telemetry information while the packet traverses a path between two points in the network. The IOAM data fields are defined in [RFC9197]. Currently, there is no adopted method defined to reflect the collected IOAM data fields back to the Sender, where the Sender can use that information to support the hop-by-hop and edge-to-edge measurement use cases.

IPv6 packets may carry IPv6 extension headers containing IPv6 options headers for Hop-By-Hop (HBH) and Destination types as defined in [RFC8200]. The Hop-By-Hop options processing procedures are further specified in [RFC9673].

[RFC9486] defines option types for HBH and destination options headers to carry IOAM data fields [RFC9197] for the IPv6 data plane.

It may be desired to record and collect HBH and E2E operational and telemetry information using active measurement packets between two nodes in a network. This is achieved by augmenting STAMP [RFC8762] using optional STAMP extensions defined in [RFC8972] to reflect IP headers as well as IPv6 extension headers as specified in this document. The procedure defined in this document leverages the existing implementations on the midpoint nodes with IP data plane that supports the IPv6 extension headers used, without any additional requirements.

2. Conventions Used in This Document

2.1. 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.

2.2. Abbreviations

ECMP: Equal Cost Multi-Path

E2E: Edge-To-Edge

HBH: Hop-By-Hop

IOAM: In Situ Operations, Administration, and Maintenance

MTU: Maximum Transmission Unit

STAMP: Simple Two-way Active Measurement Protocol

TLV: Type-Length-Value

2.3. STAMP Reference Topology

In the "STAMP Reference Topology" shown in Figure 1, the STAMP Session-Sender S1 initiates a Session-Sender test packet, and the STAMP Session-Reflector R1 transmits a reply Session-Reflector test packet. Node M1 is a midpoint node that does not perform any STAMP processing.

T1 is a transmit timestamp, and T4 is a receive timestamp added by node S1. T2 is a receive timestamp, and T3 is a transmit timestamp added by node R1.

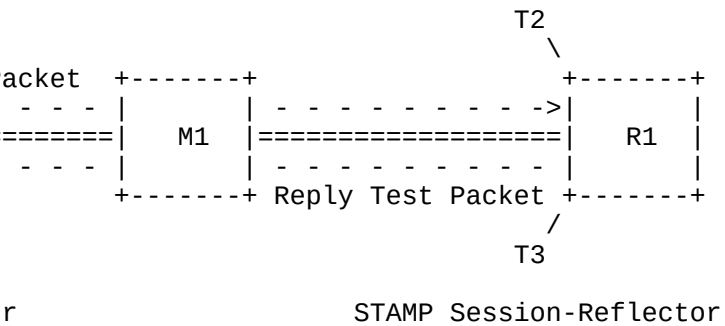

Figure 1: STAMP Reference Topology

3. Overview

[RFC8972] defines optional extensions for STAMP. The optional extensions are added to the base STAMP test packet defined in [RFC8762] in the form of TLVs. As specified in [RFC8972], both Session-Sender and Session-Reflector test packets are symmetric in size when including all optional TLVs. The Session-Reflector reflects all received STAMP TLVs from the Session-Sender test packets.

As specified in [RFC8762], STAMP test packets are transmitted with IP/UDP headers. Since midpoint nodes do not process the UDP headers in the packets, they are agnostic to the STAMP test packets in the payload.

3.1. IPv6 Data Plane

This document defines a new TLV option for STAMP, called "Reflected IPv6 Extension Header Data" (value TBA1). When a STAMP Session-Sender adds an IPv6 extension header, such as an IPv6 Hop-By-Hop options header or a Destination options header in the IPv6 header [RFC8200], it also adds a "Reflected IPv6 Extension Header Data" STAMP TLV in the Session-Sender test packet with the length set to the IPv6 extension header length that includes the lengths of all options (starting from the Next Header field) and the value field in the STAMP TLV initialized to zeros, in order to receive a copy of that IPv6 extension header back in the STAMP TLV. When adding multiple IPv6 extension headers in the Session-Sender test packet, corresponding "Reflected IPv6 Extension Header Data" STAMP TLVs MUST be added, with the matching length of the IPv6 extension header and in the same order, in order to receive a copy of that IPv6 extension header.

An example STAMP test packet for the IPv6 data plane carrying the IPv6 header and IPv6 extension headers and reflected IPv6 header data in STAMP TLVs, is shown in Figure 2.

Examples of IPv6 extension headers include: IOAM data fields IPv6 options header defined in [RFC9486], Performance and Diagnostic Metrics (PDM) IPv6 options header defined in [RFC8250], Maximum Path MTU IPv6 options header defined in [RFC9268], Alternate Marking Method IPv6 options header defined in [RFC9343], Routing Header for IPv6 including Segment Routing Header defined in [RFC8754], and any new IPv6 extension header that is defined in the future.

As the procedure defined in this document leverages the existing implementations on the midpoint nodes for the IPv6 extension headers, no additional requirements are specified when carrying these IPv6 extension headers in STAMP. The IPv6 extension header is processed by the nodes using the same procedures specified in the document that defined the IPv6 extension header.

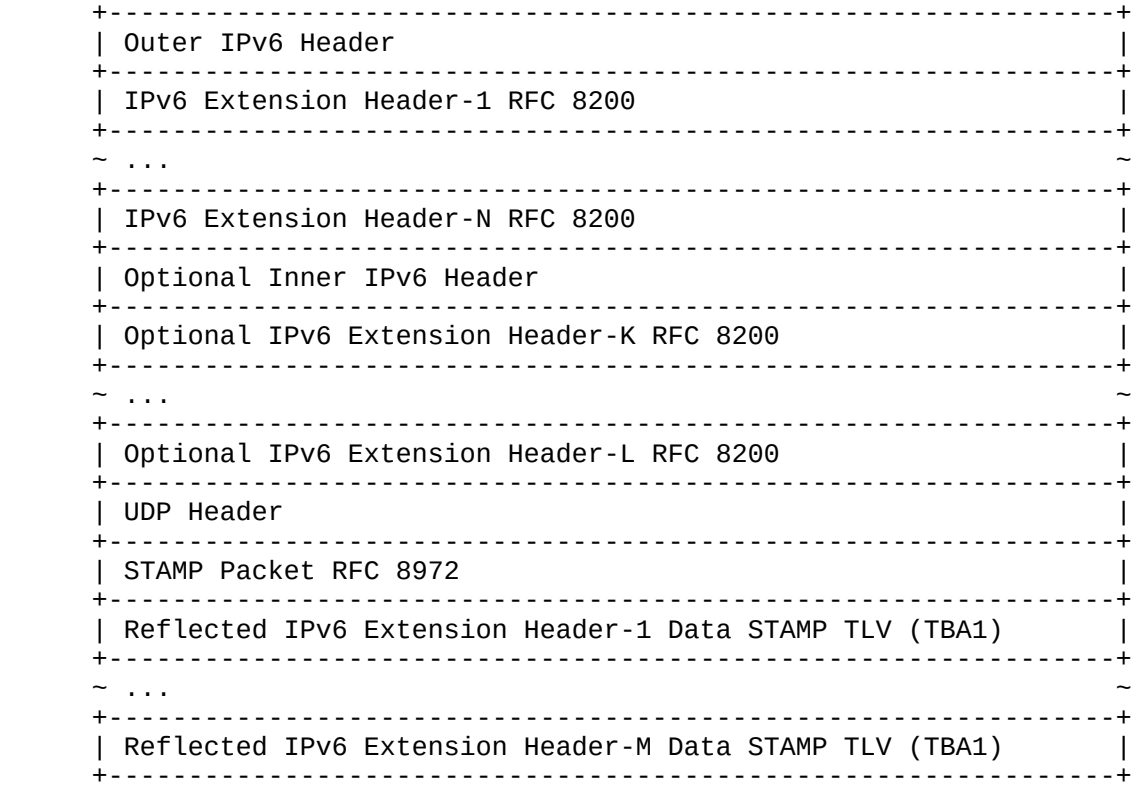

Figure 2: Example STAMP Test Packet with Reflected IPv6 Extension Header Data STAMP TLV

When the Session-Reflector receives a STAMP test packet with an IPv6 extension header and a STAMP TLV of "Reflected IPv6 Extension Header Data," the Session-Reflector that supports this STAMP TLV MUST copy the entire IPv6 extension header (i.e., all option headers), into the STAMP "Reflected IPv6 Extension Header Data" TLV in the Session-Reflector payload. When there are multiple IPv6 extension headers in the received Session-Sender test packet, each IPv6 extension header MUST be processed in order, starting from the outer header, and copied into the corresponding STAMP "Reflected IPv6 Extension Header Data" TLV in the reply Session-Reflector test packet, if that STAMP TLV exists.

When the Session-Reflector receives a STAMP test packet with an IPv6 extension header but without a "Reflected IPv6 Extension Header Data" STAMP TLV, the Session-Reflector does not copy the IPv6 extension header into the reply Session-Reflector test packet.

When the Session-Sender test packets carry an IPv6 extension header that it does not require the Session-Reflector to reflect in the Session-Reflector test packet, it does not add the matching "Reflected IPv6 Extension Header Data" TLV in the Session-Sender test packet.

If the Session-Reflector receives Session-Sender test packets with non-zero values in the first 4 bytes of the "Reflected IPv6 Extension Header Data" STAMP TLV, it MUST match the values in the corresponding IPv6 extension header before copying data into the STAMP TLV. This mechanism is employed in cases of ambiguity when there are multiple IPv6 extension headers with the same length present and not all need to be copied and reflected in the STAMP TLVs.

The Session-Sender and Session-Reflector test packets are symmetric in size, and hence the Session-Sender and Session-Reflector MUST ensure that the resulting test packets do not exceed the IPv6 MTU after adding the Reflected Data STAMP TLVs. If necessary, Reflected Data STAMP TLVs can be removed to avoid violating the IPv6 MTU limit.

If, for any reason, the Session-Reflector does not use the received "Reflected IPv6 Extension Header Data" STAMP TLV for reflecting data, it MUST return the STAMP TLV as unrecognized, i.e., with the U flag (Unrecognized) set in the STAMP TLV Flags using the procedure defined in [RFC8972].

The Session-Reflector adds the matching IPv6 extension header with IPv6 option(s) in the IPv6 header of the Session-Reflector test packets in the same order for the reverse direction measurements, as described in Section 5.3.

Note that the use case where the IPv6 extension header length changes in the Session-Sender test packets along the path is outside the scope of this document. Additionally, the use case where IPv6 extension headers are added or removed in the Session-Sender test packets along the path is outside the scope of this document.

3.2. Fixed Header

This document defines a new TLV option for STAMP, called "Reflected Fixed Header Data" (value TBA2). The STAMP TLV can be used to reflect any fixed size header received in the Session-Sender test packet, including IPv4 and IPv6 headers. When a STAMP Session-Sender adds an IP header, it also adds a "Reflected Fixed Header Data" STAMP TLV in the Session-Sender test packet with the length set to the IP header length and the value field in the TLV initialized to zeros, in order to receive a copy of that IP header back in the STAMP TLV. When adding multiple IP headers in the Session-Sender test packet,

multiple corresponding "Reflected Fixed Header Data" TLVs are added, each one with the matching length to the IP header and in the same order.

An example STAMP test packet carrying the IP header and reflected IP header in STAMP TLVs is shown in Figure 3.

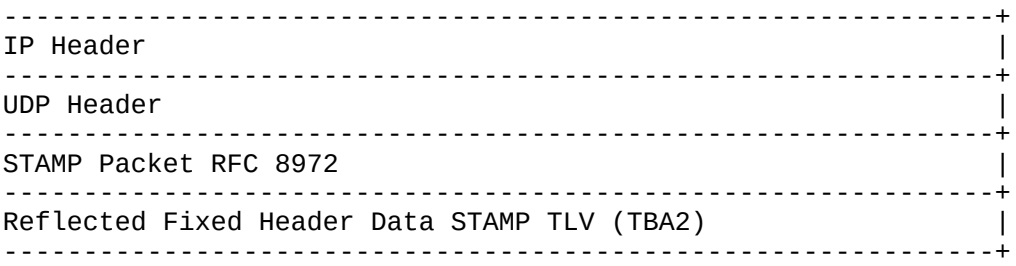

Figure 3: Example STAMP Test Packet with Reflected Fixed Header Data STAMP TLV

When the Session-Reflector receives a STAMP test packet with a STAMP TLV of "Reflected Fixed Header Data," the Session-Reflector that supports this STAMP TLV MUST copy the IP header into the "Reflected Fixed Header Data" TLV in the Session-Reflector payload. When there are multiple IP headers in the received Session-Sender test packet, all IP headers MUST be copied into the "Reflected Fixed Header Data" TLVs in the reply Session-Reflector test packet in the same order.

When the Session-Reflector receives a STAMP test packet with an IP header but without a "Reflected Fixed Header Data" STAMP TLV, the Session-Reflector does not copy the IP header into the reply Session-Reflector test packet.

When the Session-Sender test packets carry an IP header that it does not require the Session-Reflector to reflect in the Session-Reflector test packet, it does not add the matching "Reflected Fixed Header Data" TLV in the Session-Sender test packet.

If the Session-Reflector receives Session-Sender test packets with non-zero values in the first 4 bytes in the "Reflected Fixed Header Data" STAMP TLV, it MUST match the values in the corresponding IP header before copying data into the STAMP TLV. This mechanism is employed in case of ambiguity when there are multiple IP headers with the same length and not all need to be copied and reflected in the STAMP TLV.

The Session-Sender and Session-Reflector test packets are symmetric in size, and hence the Session-Sender and Session-Reflector MUST ensure that the resulting test packets do not exceed the IP MTU after adding the Reflected Data STAMP TLVs. If necessary, Reflected Data STAMP TLVs can be removed to avoid violating the IP MTU limit.

If, for any reason, the Session-Reflector does not use the received "Reflected Fixed Header Data" STAMP TLV for reflecting data, it MUST return the STAMP TLV as unrecognized, i.e., with the U flag (Unrecognized) set in the STAMP TLV Flags using the procedure defined in [RFC8972].

4. Use Case of Reflecting IOAM Data Fields

In Situ Operations, Administration, and Maintenance (IOAM) is used for recording and collecting operational and telemetry information while the packet traverses a path between two points in the network. The IOAM data fields are defined in [RFC9197]. Examples of data recorded by IOAM Trace Options include per-hop information, such as node ID, timestamp, queue depth, interface ID, interface load, etc. The information collected can be used for monitoring ECMP paths, proof-of-transit, and troubleshooting failures in the network. IOAM can be used with STAMP test packets for active measurement. The procedure and STAMP extensions defined in this document can be used to reflect the collected IOAM data fields back to the Sender, where the Sender can use that information to support the hop-by-hop and edge-to-edge measurement use cases.

IOAM Direct Exporting (DEX) [RFC9326] is applicable with STAMP test packets for on-path telemetry use cases [I-D.ietf-ippm-on-path-active-measurements]. In this case, the Session-Reflector does not reflect any IOAM data fields since no data is recorded in the test packets.

[RFC9486] defines types for HBH and destination options headers and is used to carry the IOAM option types defined in [RFC9197] for the IPv6 data plane. The STAMP Session-Sender and Session-Reflector test packets carry the IPv6 options headers with IOAM option types for recording and collecting HBH and E2E operational and telemetry information for active measurement, as shown in Figure 4. The Session-Sender, midpoints, and Session-Reflector nodes process the IOAM data fields as defined in [RFC9486]. Note that using the IOAM option type "Incremental Trace Option-Type" is not supported by [RFC9486].

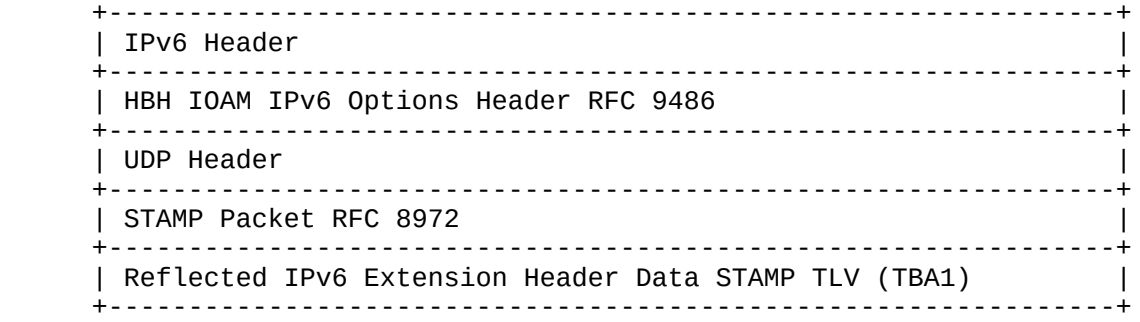

Figure 4: Example STAMP Test Packet with Reflected IPv6 Extension Header Data TLV

5. STAMP Extensions

5.1. Reflected IPv6 Extension Header Data STAMP TLV

The "Reflected IPv6 Extension Header Data" STAMP TLV is carried by Session-Sender and Session-Reflector test packets. STAMP test packets may carry multiple TLVs of this type. The same "Reflected IPv6 Extension Header Data" STAMP TLV Type is used for reflecting various IPv6 extension headers, including HBH and Destination IPv6 options headers. The format of the Reflected IPv6 Extension Header Data TLV is shown in Figure 5.

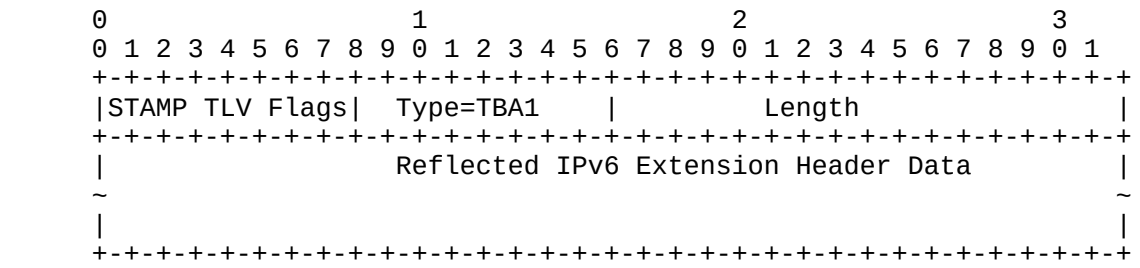

Figure 5: Reflected IPv6 Extension Header Data STAMP TLV

The TLV fields are defined as follows:

Type: Type (value TBA1)

STAMP TLV Flags: The STAMP TLV Flags follow the procedures described in [RFC8972].

Length: A two-octet field equal to the length of the Data in octets.

The Session-Reflector MUST return an error as unrecognized (U flag) in the STAMP TLV Flags when it determines that the length of the TLV does not match the length of the corresponding IPv6 extension header in the IPv6 header when processing in the same order.

5.2. Reflected Fixed Header Data STAMP TLV

The "Reflected Fixed Header Data" STAMP TLV is carried by Session-Sender and Session-Reflector test packets. STAMP test packets may carry multiple TLVs of this type. The format of the "Reflected Fixed Header Data" TLV is shown in Figure 6.

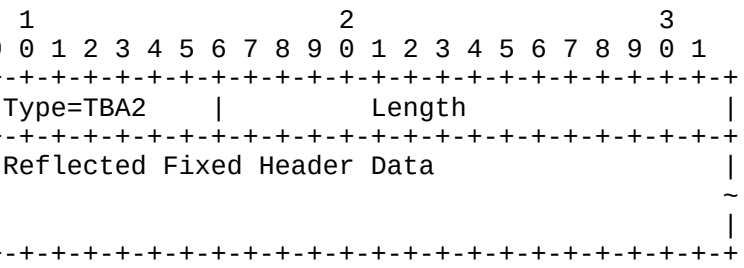

Figure 6: Reflected Fixed Header Data STAMP TLV

The TLV fields are defined as follows:

Type: Type (value TBA2)

STAMP TLV Flags: The STAMP TLV Flags follow the procedures described in [RFC8972].

Length: A two-octet field equal to the length of the Data in octets. For an IPv4 header, the length is set to 20, and for an IPv6 header, the length is set to 60.

The Session-Reflector MUST return an error as unrecognized (U flag) in the STAMP TLV Flags when it determines that the length of the TLV does not match the length of the corresponding IP header when processing in the same order.

5.3. One-Way Measurement Using Reflected Data STAMP TLVs

In the case of one-way HBH and E2E measurements for IPv6 data plane, the Session-Reflector does not need to add IPv6 extension headers in the reply Session-Reflector test packets matching the received IPv6 extension headers.

In this document, the Sub-TLV "IPv6 Extension Header Control" (Type TBA3) is defined for the STAMP TLV Type "Reflected Test Packet Control TLV" (Type TBA-ASYM) introduced in [I-D.ietf-ippm-asymmetrical-pkts].

When a Session-Sender test packet is received with the "IPv6 Extension Header Control" Sub-TLV, the Session-Reflector does not add the received IPv6 extension headers in the IPv6 header of the reply Session-Reflector STAMP test packet.

In the absence of this Sub-TLV in the received Session-Sender test packet, the Session-Reflector adds new IPv6 extension headers matching all received IPv6 extension headers (except the routing extension headers specific to the Session-Sender test packets) in the IPv6 header of the reply Session-Reflector test packet.

The IPv6 extension headers received in the Session-Sender test packets are still copied and reflected in STAMP TLVs to the Session-Sender.

6. Security Considerations

The security considerations specified in [RFC8762], [RFC8972], and [RFC8200] apply to the procedure and extensions defined in this document. In addition, the security considerations specified in [RFC9197] also apply when using the IPv6 options headers defined in that document.

7. Implementation Status

Editorial note: Please remove this section prior to publication.

An open-source implementation of the Simple Two-Way Active Measurement Protocol [RFC8762] is available in Teaparty.

https://github.com/cerfcast/teaparty

An implementation of the solution in this document is available at the following location:

https://github.com/cerfcast/teaparty/commit/393abf9357a6c2439877d9bcf2dc426dd89c7158

The features implemented are:

1. Extraction of Extension Headers from IPv6 packets from STAMP test packets.

2. Reflection of the headers in the reflected STAMP packet (with checks for matching length).

3. Reflection of the headers in the IP header.

4. Support for multiple IPv6 extension headers.

And there is also support for the Reflected IPv6 EH TLV in the Wireshark dissector:

https://github.com/cerfcast/teaparty/commit/fb74e2e02396e9bb3ead017e8d9a0c187e3573e2

And there is also support for tools for testing reflected IPv6 Extension Header Data:

https://github.com/cerfcast/teaparty/tree/main/testing_data#testing-reflected-ipv6-extension-header-data

Contact:

William Hawkins

University of Cincinnati

Email: hawkinsw@obs.cr

8. IANA Considerations

IANA has created the "STAMP TLV Types" registry for [RFC8972]. IANA is requested to allocate a value for the "Reflected IPv6 Extension Header Data" TLV Type and a value for the "Reflected Fixed Header Data" TLV Type from the IETF Review TLV range of the same registry.

Value	Description	Reference
TBA1	Reflected IPv6 Extension Header Data	This document
TBA2	Reflected Fixed Header Data	This document

Table 1: STAMP TLV Types

IANA is requested to allocate a value for the Sub-TLV Type "IPv6 Extension Header Control" (Type TBA3) for the STAMP TLV Type "Reflected Test Packet Control TLV" (Type TBA-ASYM) defined in [I-D.ietf-ippm-asymmetrical-pkts], from the "STAMP Sub-TLV Types" registry.

Value	Description	TLV Used	Reference
TBA3	IPv6 Extension Header Control	Reflected Test Packet Control	This document

Table 2: Sub-TLV Type for Reflected Test Packet Control STAMP TLV

9. References

9.1. 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>.

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8762] Mirsky, G., Jun, G., Nydell, H., and R. Foote, "Simple Two-Way Active Measurement Protocol", RFC 8762, DOI 10.17487/RFC8762, March 2020, <https://www.rfc-editor.org/info/rfc8762>.

[RFC8972] Mirsky, G., Min, X., Nydell, H., Foote, R., Masputra, A., and E. Ruffini, "Simple Two-Way Active Measurement Protocol Optional Extensions", RFC 8972, DOI 10.17487/RFC8972, January 2021, <https://www.rfc-editor.org/info/rfc8972>.

[RFC9673] Hinden, R. and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", RFC 9673, DOI 10.17487/RFC9673, October 2024, <https://www.rfc-editor.org/info/rfc9673>.

[I-D.ietf-ippm-asymmetrical-pkts]
Mirsky, G., Ruffini, E., Nydell, H., Foote, R. F., and W. Hawkins, "Performance Measurement with Asymmetrical Traffic Using STAMP", Work in Progress, Internet-Draft, draft-ietf-ippm-asymmetrical-pkts-08, 28 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ippm-asymmetrical-pkts-08>.

9.2. Informative References

[RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <https://www.rfc-editor.org/info/rfc8250>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

[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>.

[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>.

[RFC9268] Hinden, R. and G. Fairhurst, "IPv6 Minimum Path MTU Hop-by-Hop Option", RFC 9268, DOI 10.17487/RFC9268, August 2022, <https://www.rfc-editor.org/info/rfc9268>.

[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>.

[RFC9343] Fioccola, G., Zhou, T., Cociglio, M., Qin, F., and R. Pang, "IPv6 Application of the Alternate-Marking Method", RFC 9343, DOI 10.17487/RFC9343, December 2022, <https://www.rfc-editor.org/info/rfc9343>.

[I-D.ietf-ippm-on-path-active-measurements]
Fioccola, G., Zhu, K., Zhou, T., Zhu, Y., and X. Min, "On-Path Telemetry for Active Performance Measurements", Work

in Progress, Internet-Draft, draft-ietf-ippm-on-path-active-measurements-00, 22 September 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ippm-on-path-active-measurements-00>.

Acknowledgments

The authors would like to thank Greg Mirsky, Xiao Min, Tal Mizrahi, Cheng Li, Giuseppe Fioccola, Richard "Footer" Foote, and Jie Dong for reviewing this document and providing many useful comments and suggestions. Thank you William Hawkins for implementing the solution defined in this document in Teaparty.

Authors' Addresses

Rakesh Gandhi (editor)
Cisco Systems, Inc.
Canada
Email: rgandhi@cisco.com

Tianran Zhou
Huawei
China
Email: zhoutianran@huawei.com

Zhenqiang Li
China Mobile
China
Email: lizhenqiang@chinamobile.com

William Hawkins
University of Cincinnati
United States of America
Email: hawkinsw@obs.cr