

6MAN
Internet-Draft
Intended status: Standards Track
Expires: 14 February 2026

T. Mizrahi
Huawei
X. He
China Telecom
T. Zhou
Huawei
R. Bonica
HPE
X. Min
ZTE Corp.
13 August 2025

Internet Control Message Protocol (ICMPv6) Reflection
draft-ietf-6man-icmpv6-reflection-09

Abstract

This document describes the ICMPv6 Reflection utility. The ICMPv6 Reflection utility is a diagnostic tool, similar to Ping and the ICMPv6 Probe utility. It is similar to Ping and Probe in that it relies on a stateless message exchange between a probing node and a probed node. The probing node sends a request to the probed node and the probed node responds to the request.

The ICMPv6 Reflection utility differs from Ping and Probe because, in the ICMPv6 Reflection utility, the probing node requests a snapshot of the message that it sent, as it was when arrived at the probed node. The probed node returns the requested snapshot.

The ICMPv6 Reflection utility is useful because it allows the user to see how the network modified the request as it traveled from the probing node to the probed node.

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 14 February 2026.

Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Requirement Language	3
3. Use Cases	3
4. Theory of Operation	4
5. New ICMP Extension Object	6
6. IANA Considerations	7
7. Security Considerations	8
8. Acknowledgements	8
9. References	8
9.1. Normative References	8
9.2. Informative References	9
Contributors	10
Authors' Addresses	11

1. Introduction

The ICMPv6 Reflection utility is an IPv6 [RFC8200] diagnostic tool. It is similar to Ping [RFC2151] and the ICMPv6 Probe [I-D.ietf-intarea-rfc8335bis] utility in the following respects:

- * A probing node sends an ICMPv6 [RFC4443] message to a probed node. This ICMP message requests that it be reflected back to the probing node.
- * The probed node receives the above-mentioned message, encodes it into another ICMPv6 message, and sends that ICMPv6 message back to the probing node.

For the purposes of this document, the ICMPv6 message that the probing node sends is called the "request message" and the ICMPv6 message that the probed node sends is called the "reply message".

The reply message includes a copy of the request message, starting from its IPv6 header, as was when it arrived at the probed node.

The ICMPv6 Reflection utility uses the ICMPv6 Extended Echo Request and Extended Echo Reply message types [I-D.ietf-intarea-rfc8335bis]. Each of these message types includes an ICMP Extension Structure [RFC4884]. The ICMP Extension Structure includes one or more extension objects. This document defines the 'Reflect All' object, which is used for reflecting the request message, as it arrived at the probed node.

An alternative approach involves the probing node sending a UDP packet with an unused destination port to the probed node. This causes the probed node to send an ICMPv6 Destination Unreachable message, which includes "as much of invoking packet as possible without the ICMPv6 packet exceeding the minimum IPv6 MTU" [RFC4443]. However, middle boxes modify the invoking packet in ICMPv6 Destination Unreachable messages (e.g. [RFC3022]). By contrast, middle boxes do not modify the Reflect All object.

2. Requirement 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.

3. Use Cases

The following protocols define mutable IPv6 extension headers that can be used to trace a path and its performance:

- * IPv6 Options for In Situ Operations, Administration, and Maintenance (IOAM) [RFC9486]
- * Inband Flow Analyzer [I-D.kumar-ippm-ifa]
- * Path Tracing in SRv6 networks [I-D.filsfils-ippm-path-tracing]

These extensions are used to collect information along a packet's delivery path. Currently, the collected information is sent to a controller for processing. However, in some cases this information is relevant to the sender.

The ICMPv6 Reflection utility provides a mechanism by which this information can be returned to the probing node.

The ICMPv6 Reflection utility can also be used to determine how the probe message's IPv6 header has changed along its delivery path. For example, it can be used to determine how middleboxes have changed the Source Address, Destination Address, and Flow Label.

4. Theory of Operation

The probing node sends an ICMPv6 Extended Echo Request message [I-D.ietf-intarea-rfc8335bis] to the probed node. The ICMPv6 Extended Echo Request message contains an ICMP Extension Structure [RFC4884] and the ICMP Extension Structure contains an Extension Header. The ICMP Extension Header MUST include the Length field, as defined in [I-D.ietf-intarea-icmp-exten-hdr-len]. The ICMP Extension structure includes a 'Reflect All' object, which is defined in this document.

The 'Reflect All' object contains an object payload field whose length SHOULD be sufficient to carry the IPv6 and ICMP header of the reflected request message. The total length of both the request and reply packets SHOULD NOT exceed the IPv6 minimum MTU (1280 bytes), to avoid triggering fragmentation.

The probed node receives the ICMPv6 Extended Echo Request and formats an ICMPv6 Extended Echo Reply message. The total length of the ICMPv6 Extended Echo Reply message MUST be equal to the total length of the corresponding request message. The main body of the ICMPv6 Extended Echo Reply message reflects the status of an interface on the probed node.

The ICMPv6 Extended Echo Reply message also contains an ICMP Extension Structure. The ICMP Extension Header MUST include the Length field, as defined in [I-D.ietf-intarea-icmp-exten-hdr-len]. The value of the Length field in the reply message MUST be equal to the value of the Length in the ICMP Extension Header of the request message. The ICMP Extension Structure also MUST contain the 'Reflect All' object that the ICMPv6 Extended Echo Request message contained. The length of the 'Reflect All' object in the reply message MUST be equal to the length of the 'Reflect All' object in the request message. If the following requirements are satisfied, the probed node populates the payload field of the 'Reflect All' object:

- * The probed node supports the 'Reflect All' object.
- * Responding to the request message does not violate the probed node's security policy.

Otherwise, the probed node sets the object payload field to all zeros.

An example of a request and a reply is provided in Figure 1. In this example the request message includes the 'Reflect All' object. The reply also includes the 'Reflect All' object, containing the IPv6 header, ICMPv6 header and ICMP Extension Header of the request message. The request and reply messages have the same length. The length of the IPv6 header of the request message is at least 40 octets, depending on whether there are extension headers, followed by an 8-octet ICMPv6 header, a 4-octet ICMP Extension Header, and a 4-octet Object Header. For example, if the length of the IPv6 header is H octets, and the length of the object payload is $H+12$ octets, the reply includes the reflected request message starting from the IPv6 header and up to and including the ICMP Extension Header, as shown in Figure 1.

Middle boxes must not modify the Reflect All extension object. This ensures that the reflected information reaches the probing node exactly as sent by the probed node.

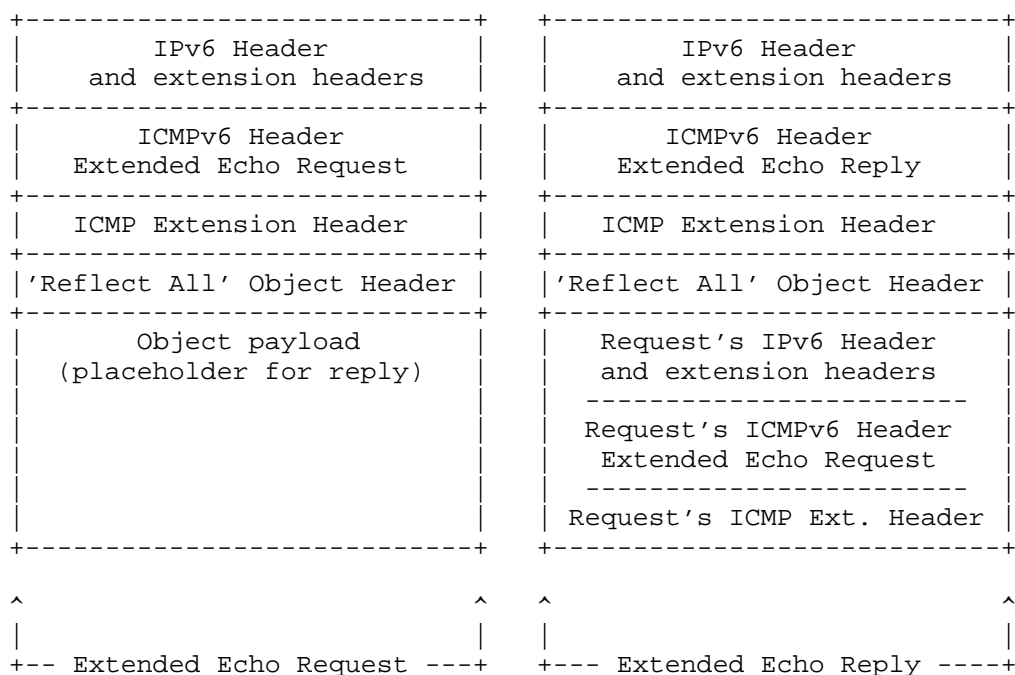


Figure 1: ICMPv6 Reflection Message Formats

If a node that does not support the 'Reflect All' object receives an ICMP Extended Echo Request containing this object, the expected behavior according to [I-D.ietf-intarea-rfc8335bis] and [RFC8335] is to respond with an ICMP Echo Reply message that includes the "Malformed Query" code in the Code field.

5. New ICMP Extension Object

This document defines the 'Reflect All' object.

An implementation that supports ICMPv6 Reflection MUST support the 'Reflect All' object.

In the ICMPv6 Reflection utility, the 'Reflect All' object MUST be included in the Extension Structure. If there is more than one object, the 'Reflect All' object MUST be the first object.

The structure of the 'Reflect All' object follows the specification of ICMP Extension Objects as defined in [RFC4884] and MUST include the following fields:

- * The Length of the 'Reflect All' object.
- * An object class (as specified in Section 6).
- * C-Type as described below.
- * An object payload field.

The Length field specifies the number of octets in the object. The Length in the 'Reflect All' object of a reply message MUST be equal to the Length in the 'Reflect All' object of the respective request message.

The C-Type value is used for indicating whether the probed node was able to process the object. The following C-Type values are supported:

- * (0) Request
- * (1) Reply - No Error
- * (2) Reply - Unsupported Object

The C-Type field of a Reflection object in a request message MUST be set to the 'Request' value. If the probed node is able to process the 'Reflect All' object, it MUST copy the request message into the object payload and update the C-Type field to the 'Reply - No Error'

value. If the probed node is not able to process the object, it MUST update the C-Type value of the object in the Extended Echo Reply to 'Reply - Unsupported Object'.

The object payload field in the ICMPv6 Extended Echo Request message is a placeholder for the corresponding reply message. Its length determines the number of bytes, starting from the beginning of the IPv6 header, that the probed node includes in the object payload of the reply. The object payload field in a request message MAY contain all zeros. Alternatively, this field MAY contain arbitrary data. In reply messages the object payload field MUST contain the received request message starting from the beginning of the IPv6 header and according to the length of the object payload.

If the 'Reflect All' object is sufficiently long, the reply message includes the initial octets of the 'Reflect All' object. A notable use case for including arbitrary data in the object payload is the inclusion of a transmission timestamp, similar to how conventional Ping utilities incorporate timestamps into the ICMP payload. If the initial octets of the 'Reflect All' object payload contain a timestamp and the object is long enough, the timestamp is reflected back to the probing node, enabling round-trip time calculations.

6. IANA Considerations

IANA is requested to allocate the following values in the "ICMP Extension Object Classes and Class Sub-types" registry.

Class-Num	Object Name	C-Type	C-Type Name	Reference
TBD1	Reflect All	0-2	See below	[This document]

Figure 2: IANA Allocation

For each of the object classes above the following C-Type values are defined:

- * (0) Request
- * (1) Reply - No Error
- * (2) Reply - Unsupported Object

7. Security Considerations

Since this document uses technologies from [RFC4443], [RFC4884], and [I-D.ietf-intarea-rfc8335bis], it inherits security considerations from those documents.

The Reflection procedure that is defined in this document is symmetric in terms of the length of the request and reply messages. This symmetry mitigates the potential for amplification attacks, which would be possible if the reply message was longer than the request message.

Depending on the network policy and the location of nodes in the network, ICMPv6 informational and/or error messages are sometimes filtered or not supported. For example, some nodes do not reply to ICMPv6 Echo or do not send ICMPv6 Time Exceeded messages (used in Traceroute), due to policy considerations that may be related to DoS mitigation or to privacy. Network operators SHOULD apply similar considerations to ICMPv6 Extended Echo messages when they are used for Reflection. For example, an operator can choose to disable support for ICMPv6 Reflection in networks or in nodes that do not respond to ICMPv6 Echo and/or do not generate ICMPv6 Time Exceeded messages.

As specified in [I-D.ietf-intarea-rfc8335bis], in order to protect local resources, implementations SHOULD rate-limit incoming ICMP Extended Echo Request messages. Moreover, as per [I-D.ietf-intarea-rfc8335bis], by default, ICMP Extend Echo functionality is disabled.

8. Acknowledgements

The authors gratefully acknowledge Sebastian Moeller, Zafar Ali, Bob Hinden, Jen Linkova, Jeremy Duncan, Greg Mirsky, Nick Buraglio and Maciej Zenczykowski for their insightful comments.

9. References

9.1. Normative References

[I-D.ietf-intarea-icmp-exten-hdr-len]
Bonica, R., hexiaoming, X., Min, X., and T. Mizrahi, "ICMP Extension Structure Length Field", Work in Progress, Internet-Draft, draft-ietf-intarea-icmp-exten-hdr-len-05, 5 August 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-intarea-icmp-exten-hdr-len-05>>.

[I-D.ietf-intarea-rfc8335bis]

Fenner, B., Bonica, R., Thomas, R., Linkova, J., Lenart, C., and M. Boucadair, "PROBE: A Utility for Probing Interfaces", Work in Progress, Internet-Draft, draft-ietf-intarea-rfc8335bis-01, 21 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-intarea-rfc8335bis-01>>.

[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>>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <<https://www.rfc-editor.org/info/rfc4884>>.

[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>>.

[RFC8335] Bonica, R., Thomas, R., Linkova, J., Lenart, C., and M. Boucadair, "PROBE: A Utility for Probing Interfaces", RFC 8335, DOI 10.17487/RFC8335, February 2018, <<https://www.rfc-editor.org/info/rfc8335>>.

9.2. Informative References

[I-D.filsfils-ippm-path-tracing]

Filsfils, C., Abdelsalam, A., Camarillo, P., Yufit, M., Su, Y., Matsushima, S., Valentine, M., and Dhamija, "Path Tracing in SRv6 networks", Work in Progress, Internet-Draft, draft-filsfils-ippm-path-tracing-04, 4 July 2025, <<https://datatracker.ietf.org/doc/html/draft-filsfils-ippm-path-tracing-04>>.

[I-D.kumar-ippm-ifa]

Kumar, J., Anubolu, S., Lemon, J., Manur, R., Holbrook, H., Ghanwani, A., Cai, D., Ou, H., Li, Y., and X. Wang, "Inband Flow Analyzer", Work in Progress, Internet-Draft, draft-kumar-ippm-ifa-08, 26 April 2024, <<https://datatracker.ietf.org/doc/html/draft-kumar-ippm-ifa-08>>.

[RFC2151] Kessler, G. and S. Shepard, "A Primer On Internet and TCP/IP Tools and Utilities", FYI 30, RFC 2151, DOI 10.17487/RFC2151, June 1997, <<https://www.rfc-editor.org/info/rfc2151>>.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<https://www.rfc-editor.org/info/rfc3022>>.

[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>>.

Contributors

Shahar Belkar
Huawei
8-2 Matam
Haifa 3190501
Israel
Email: shahar.belkar@huawei.com

Chongfeng Xie
China Telecom
Email: xiechf@chinatelecom.cn

Zhenqiang Li
China Mobile
Email: li_zhenqiang@hotmail.com

Justin Iurman
Universite de Liege
10, Allee de la decouverte (B28)
4000 Sart-Tilman
Belgium
Email: justin.iurman@uliege.be

Authors' Addresses

Tal Mizrahi
Huawei
25 Matam
Haifa 3190501
Israel
Email: tal.mizrahi.phd@gmail.com

Xiaoming He
China Telecom
Email: hexm4@chinatelecom.cn

Tianran Zhou
Huawei
156 Beiqing Rd.
Beijing
100095
China
Email: zhoutianran@huawei.com

Ron Bonica
HPE
United States of America
Email: rbonica@juniper.net

Xiao Min
ZTE Corp.
Email: xiao.min2@zte.com.cn