Workgroup: 6MAN Working Group
Internet-Draft:
draft-xiao-6man-icmpv6-ioam-conf-state-01
Updates: 4884 (if approved)
Published: 25 April 2022
Intended Status: Standards Track
Expires: 27 October 2022
Authors: X. Min G. Mirsky
ZTE Corp. Ericsson

ICMPv6 Echo Request/Reply for Enabled In-situ OAM Capabilities

Abstract

This document describes the ICMPv6 IOAM Echo functionality, which uses the ICMPv6 IOAM Echo Request/Reply messages, allowing the IOAM encapsulating node to discover the enabled IOAM capabilities of each IOAM transit and decapsulating node.

This document updates RFC 4884.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

IPv6 encapsulation for In-situ OAM (IOAM) data is defined in [I-D.ietf-ippm-ioam-ipv6-options], which uses IPv6 hop-by-hop options and destination option to carry IOAM data.

As specified in [I-D.ietf-ippm-ioam-conf-state], echo request/reply can be used for the IOAM encapsulating node to discover the enabled IOAM capabilities at IOAM transit and decapsulating nodes.

As specified in [RFC4443], the Internet Control Message Protocol for IPv6 (ICMPv6) is an integral part of IPv6, and the base protocol MUST be fully implemented by every IPv6 node. ICMPv6 messages include error messages and informational messages, and the latter are referred to as ICMPv6 Echo Request/Reply messages. [RFC4884] defines ICMPv6 Extension Structure by which multi-part ICMPv6 error messages are supported. [RFC8335] defines ICMPv6 Extended Echo Request/Reply messages, and the ICMPv6 Extended Echo Request contains an ICMPv6 Extension Structure customized for this message. Both [RFC4884] and [RFC8335] provide sound principles and examples on how to extend ICMPv6 error messages and echo request/reply messages.

This document describes the ICMPv6 IOAM Echo functionality, which uses the ICMPv6 IOAM Echo Request/Reply messages, allowing the IOAM encapsulating node to discover the enabled IOAM capabilities of each IOAM transit and decapsulating node.

The IOAM encapsulating node sends an ICMPv6 IOAM Echo Request message to each IOAM transit and decapsulating node, then each receiving node executes access control procedures, and if access is granted, each receiving node returns an ICMPv6 IOAM Echo Reply message which indicates the enabled IOAM capabilities of the receiving node. The ICMPv6 IOAM Echo Reply message contains an ICMPv6 Extension Structure exactly customized to this message, and the ICMPv6 Extension Structure contains one or more IOAM Capabilities Objects.

Note that before the IOAM encapsulating node sends the ICMPv6 IOAM Echo Request messages, it needs to know the IPv6 address of each node along the transport path of a data packet to which IOAM data would be added. That can be achieved by executing ICMPv6 traceroute or provisioning explicit path at the IOAM encapsulating node.

2. Conventions Used in This Document

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. ICMPv6 IOAM Echo Request

The ICMPv6 IOAM Echo Request message is encapsulated in an IPv6 header [RFC8200], like any ICMPv6 message.

The ICMPv6 IOAM Echo Request message has the following 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |Sequence Number| Num of NS-IDs |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.           IOAM Capabilities Query Container Payload           .
.                         as specified in                       .
.         Section 3.1 of draft-ietf-ippm-ioam-conf-state        .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: ICMPv6 IOAM Echo Request Message

IPv6 Header fields:

*Source Address: The Source Address identifies the IOAM encapsulating node. It MUST be a valid IPv6 unicast address.

*Destination Address: The Destination Address identifies the IOAM transit or decapsulating node. It MUST be a valid IPv6 unicast address.

ICMPv6 fields:

*Type: IOAM Echo Request. The value is TBD1.

*Code: MUST be set to 0 and MUST be ignored upon receipt.

*Checksum: The same as defined in [RFC4443].

*Identifier: An Identifier aids in matching IOAM Echo Replies to IOAM Echo Requests. It may be zeroed.

*Sequence Number: A Sequence Number to aid in matching IOAM Echo Replies to IOAM Echo Requests. It may be zeroed.

*Num of NS-IDs: Number of Namespace-IDs within the payload.

*Following the IOAM Echo Request header, it's a List of Namespace-IDs, which is also called IOAM Capabilities Query Container Payload in Section 3.1 of [I-D.ietf-ippm-ioam-conf-state]. If the payload would not otherwise terminate on a 4-octet boundary, it MUST be padded with zeroes.

4. ICMPv6 IOAM Echo Reply

The ICMPv6 IOAM Echo Reply message is encapsulated in an IPv6 header [RFC8200], like any ICMPv6 message.

The ICMPv6 IOAM Echo Reply message has the following 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |Sequence Number| Num of NS-IDs |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.         IOAM Capabilities Response Container Payload          .
.                         as specified in                       .
.         Section 3.2 of draft-ietf-ippm-ioam-conf-state        .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: ICMPv6 IOAM Echo Reply Message

IPv6 Header fields:

- *Source Address: Copied from the Destination Address field of the invoking IOAM Echo Request packet.
- *Destination Address: Copied from the Source Address field of the invoking IOAM Echo Request packet.

ICMPv6 fields:

- *Type: IOAM Echo Reply. The value is TBD2.
- *Code: Values are (0) No Error, (1) Malformed Query, (2) No Matched Namespace-ID, and (3) Exceed the minimum IPv6 MTU.
- *Checksum: The same as defined in [RFC4443].
- *Identifier: Copied from the Identifier field of the invoking IOAM Echo Request message.
- *Sequence Number: Copied from the Sequence Number field of the invoking IOAM Echo Request message.
- *Num of NS-IDs: Number of different Namespace-IDs within the payload, its value MUST be no more than the Num of NS-IDs field of the invoking IOAM Echo Request message.
- *Following the IOAM Echo Reply header, it's a List of IOAM Capabilities Objects, which is also called IOAM Capabilities Response Container Payload in Section 3.2 of [I-D.ietf-ippm-ioam-conf-state].
- *Section 7 of [RFC4884] defines the ICMP Extension Structure. As per RFC 4884, the Extension Structure contains exactly one Extension Header followed by one or more objects. When applied to the ICMPv6 IOAM Echo Reply message, the ICMP Extension Structure MUST contain one or more IOAM Capabilities Objects.

4.1. IOAM Capabilities Objects

All ICMPv6 IOAM Capabilities Objects are encapsulated in an ICMPv6 IOAM Echo Reply message.

Each ICMPv6 IOAM Capabilities Object has the following 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length            |   Class-Num   |   C-Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                IOAM Capabilities Object Payload               .
.                        as specified in                        .
.        Section 3.2.x of draft-ietf-ippm-ioam-conf-state       .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: IOAM Capabilities Object

Object fields:

*Class-Num: IOAM Capabilities Objects. The values are listed as the following:

Value	Object Name
TBD3	IOAM Tracing Capabilities Object
TBD4	IOAM Proof-of-Transit Capabilities Object
TBD5	IOAM Edge-to-Edge Capabilities Object
TBD6	IOAM DEX Capabilities Object
TBD7	IOAM End-of-Domain Object

*C-Type: Values are listed as the following:

Class-Num	C-Type	C-Type Name
TBD3	0	Reserved
	1	Pre-allocated Tracing
	2	Incremental Tracing
TBD4	0	Reserved
TBD5	0	Reserved
TBD6	0	Reserved
TBD7	0	Reserved

*Length: Length of the object, measured in octets, including the Object Header and Object Payload.

*Following the IOAM Capabilities Object Header, it's the IOAM Capabilities Object Payload, which is defined respectively in Section 3.2.1, Section 3.2.2, Section 3.2.3, Section 3.2.4, Section 3.2.5 and Section 3.2.6 of [I-D.ietf-ippm-ioam-conf-state].

4.2. Examples of IOAM Echo Reply

The format of ICMPv6 IOAM Echo Reply can vary from deployment to deployment.

In a deployment where only the default Namespace-ID is used, the IOAM Pre-allocated Tracing Capabilities and IOAM Proof-of-Transit Capabilities are enabled at the IOAM transit node that received ICMPv6 IOAM Echo Request message, the ICMPv6 IOAM Echo Reply message is depicted as the following:

```
  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      |     Code      |          Checksum             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Identifier          |Sequence Number| Num of NS-IDs |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |             Length            |   Class-Num   |   C-Type      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |               IOAM-Trace-Type                 |  Reserved   |W|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         Namespace-ID          |          Ingress_MTU          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Ingress_if_id (short or wide format)         ......          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |             Length            |   Class-Num   |   C-Type      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         Namespace-ID          | IOAM-POT-Type |SoP| Reserved  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Example 1 of IOAM Echo Reply

In a deployment where two Namespace-IDs (Namespace-ID1 and Namespace-ID2) are used, for both Namespace-ID1 and Namespace-ID2 the IOAM Pre-allocated Tracing Capabilities and IOAM Proof-of-Transit Capabilities are enabled at the IOAM transit node that received ICMPv6 IOAM Echo Request message, the ICMPv6 IOAM Echo Reply message is depicted as the following:

```
 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      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Identifier          |Sequence Number| Num of NS-IDs |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Length           |   Class-Num   |   C-Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                IOAM-Trace-Type                |  Reserved   |W|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID1         |          Ingress_MTU          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Ingress_if_id (short or wide format)         ......          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Length           |   Class-Num   |   C-Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID1         | IOAM-POT-Type |SoP| Reserved  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Length           |   Class-Num   |   C-Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                IOAM-Trace-Type                |  Reserved   |W|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID2         |          Ingress_MTU          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Ingress_if_id (short or wide format)         ......          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Length           |   Class-Num   |   C-Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID2         | IOAM-POT-Type |SoP| Reserved  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Example 2 of IOAM Echo Reply

In a deployment where only the default Namespace-ID is used, the IOAM Pre-allocated Tracing Capabilities, IOAM Proof-of-Transit Capabilities and IOAM Edge-to-Edge Capabilities are enabled at the IOAM decapsulating node that received ICMPv6 IOAM Echo Request message, the ICMPv6 IOAM Echo Reply message is depicted as the following:

```
     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      |     Code      |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Identifier          |Sequence Number| Num of NS-IDs |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |   Class-Num   |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |               IOAM-Trace-Type                 |  Reserved   |W|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Namespace-ID          |          Ingress_MTU          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |  Ingress_if_id (short or wide format)         ......          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |   Class-Num   |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Default Namespace-ID      | IOAM-POT-Type |SoP| Reserved  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |   Class-Num   |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Namespace-ID          |         IOAM-E2E-Type         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |TSF|         Reserved          |              MBZ              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Example 3 of IOAM Echo Reply

Note that when an ICMPv6 IOAM Echo Request message or IOAM Echo Reply message is received, the Payload Length field of IPv6 Header [RFC8200] indicates the message length.

5. ICMPv6 Message Processing

When a node receives an ICMPv6 IOAM Echo Request and any of the following conditions apply, the node MUST silently discard the incoming message:

- The node does not recognize the ICMPv6 IOAM Echo Request message.

- The node has not explicitly enabled ICMPv6 IOAM Echo functionality.

- The incoming ICMPv6 IOAM Echo Request carries a Source Address that is not explicitly authorized.

- The Source Address of the incoming message is not a unicast address.

- The Destination Address of the incoming message is a multicast address.

Otherwise, when a node receives an ICMPv6 IOAM Echo Request, it MUST format an ICMPv6 IOAM Echo Reply as follows:

- Set the Hop Limit to 255.

- Set the DiffServ codepoint to CS0 [RFC4594].

- Copy the Destination Address from the IOAM Echo Request to the Source Address of the IOAM Echo Reply.

- Copy the Source Address from the IOAM Echo Request to the Destination Address of the IOAM Echo Reply.

- Set the Next Header to (58) ICMPv6.

- Set the ICMPv6 Type to (TBD2) IOAM Echo Reply.

- Copy the Identifier from the IOAM Echo Request to the IOAM Echo Reply.

- Copy the Sequence Number from the IOAM Echo Request to the IOAM Echo Reply.

- Set the Code field as described in Section 5.1.

- If the Code field is equal to (0) No Error, then add one or more objects as described in Section 4.1.

- Set the Checksum appropriately.

- Forward the ICMPv6 IOAM Echo Reply to its destination.

5.1. Code Field Processing

The Code field MUST be set to (1) Malformed Query if any of the following conditions apply:

- The ICMPv6 IOAM Echo Request does not include any Namespace-ID.

- The value of Num of NS-IDs field does not match the contained list of Namespace-IDs.

- The query is otherwise malformed.

The Code field MUST be set to (2) No Matched Namespace-ID if none of the contained list of Namespace-IDs is recognized.

The Code field MUST be set to (3) Exceed the minimum IPv6 MTU if the formatted ICMPv6 IOAM Echo Reply exceeds the minimum IPv6 MTU (i.e., 1280 octets). In this case, all objects MUST be stripped before forwarding the ICMPv6 Echo Reply to its destination.

Otherwise, the Code field MUST be set to (0) No Error.

6. Updates to RFC 4884

Section 4.6 of [RFC4884] provides a list of extensible ICMP messages (i.e., messages that can carry the ICMP Extension Structure). This document adds the ICMPv6 IOAM Echo Request message and the ICMPv6 IOAM Echo Reply message to that list.

7. IANA Considerations

This document requests the following IANA actions:

- Add the following to the "ICMPv6 'type' Numbers" registry:

  - TBD1 IOAM Echo Request

  - As ICMPv6 distinguishes between informational and error messages, and this is an informational message, the value must be assigned from the range 128-255.

- Add the following to the "Type TBD1 - IOAM Echo Request" sub-registry:

  - (0) No Error

- Add the following to the "ICMPv6 'type' Numbers" registry:

  - TBD2 IOAM Echo Reply

  - As ICMPv6 distinguishes between informational and error messages, and this is an informational message, the value must be assigned from the range 128-255.

- Add the following to the "Type TBD2 - IOAM Echo Reply" sub-registry:

  - (0) No Error

  - (1) Malformed Query

  - (2) No Matched Namespace-ID

-(3) Exceed the minimum IPv6 MTU

*Add the following to the "ICMP Extension Object Classes and Class Sub-types" registry:

-(TBD3) IOAM Tracing Capabilities Object

*Add the following C-types to the "Sub-types - Class TBD3 - IOAM Tracing Capabilities Object" sub-registry:

-(0) Reserved

-(1) Pre-allocated Tracing

-(2) Incremental Tracing

-C-Type values are assigned on a First Come First Serve (FCFS) basis with a range of 0-255.

*Add the following to the "ICMP Extension Object Classes and Class Sub-types" registry:

-(TBD4) IOAM Proof-of-Transit Capabilities Object

*Add the following C-types to the "Sub-types - Class TBD4 - IOAM Proof-of-Transit Capabilities Object" sub-registry:

-(0) Reserved

-C-Type values are assigned on an FCFS basis with a range of 0-255.

*Add the following to the "ICMP Extension Object Classes and Class Sub-types" registry:

-(TBD5) IOAM Edge-to-Edge Capabilities Object

*Add the following C-types to the "Sub-types - Class TBD5 - IOAM Edge-to-Edge Capabilities Object" sub-registry:

-(0) Reserved

- -C-Type values are assigned on an FCFS basis with a range of 0-255.

*Add the following to the "ICMP Extension Object Classes and Class Sub-types" registry:

- -(TBD6) IOAM DEX Capabilities Object

*Add the following C-types to the "Sub-types - Class TBD6 - IOAM DEX Capabilities Object" sub-registry:

- -(0) Reserved
- -C-Type values are assigned on an FCFS basis with a range of 0-255.

*Add the following to the "ICMP Extension Object Classes and Class Sub-types" registry:

- -(TBD7) IOAM End-of-Domain Object

*Add the following C-types to the "Sub-types - Class TBD7 - IOAM End-of-Domain Object" sub-registry:

- -(0) Reserved
- -C-Type values are assigned on an FCFS basis with a range of 0-255.

All codes mentioned above are assigned on an FCFS basis with a range of 0-255.

8. Security Considerations

Securiy issues discussed in [I-D.ietf-ippm-ioam-conf-state] apply to this document.

This document recommends using IP Authentication Header [RFC4302] or IP Encapsulating Security Payload Header [RFC4303] to provide integrity protection for IOAM Capabilities information.

This document recommends using IP Encapsulating Security Payload Header [RFC4303] to provide privacy protection for IOAM Capabilities information.

This document recommends that the network operators establish policies that restrict access to ICMPv6 IOAM Echo functionality. In order to enforce these policies, nodes that support ICMPv6 IOAM Echo functionality MUST support the following configuration options:

- *Enable/disable ICMPv6 IOAM Echo functionality. By default, ICMPv6 IOAM Echo functionality is disabled.

- *Define enabled Namespace-IDs. By default, all Namespace-IDs except the default one (i.e., Namespace-ID 0x0000) are disabled.

- *For each enabled Namespace-ID, define the prefixes from which ICMPv6 IOAM Echo Request messages are permitted.

When a node receives an ICMPv6 IOAM Echo Request message that it is not configured to support, it MUST silently discard the message. See Section 5 for details.

In order to protect local resources, implementations SHOULD rate-limit incoming ICMPv6 IOAM Echo Request messages.

9. Acknowledgements

TBA.

10. References

10.1. Normative References

[I-D.ietf-ippm-ioam-conf-state] Min, X., Mirsky, G., and L. Bo, "Echo Request/Reply for Enabled In-situ OAM Capabilities", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-conf-state-03, 26 January 2022, <https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-conf-state-03.txt>.

[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>.

10.2. Informative References

[I-D.ietf-ippm-ioam-ipv6-options] Bhandari, S. and F. Brockners, "In-situ OAM IPv6 Options", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-ipv6-options-07, 6 February 2022, <https://www.ietf.org/archive/id/draft-ietf-ippm-ioam-ipv6-options-07.txt>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC4594] Babiarz, J., Chan, K., and F. Baker, "Configuration Guidelines for DiffServ Service Classes", RFC 4594, DOI 10.17487/RFC4594, August 2006, <https://www.rfc-editor.org/info/rfc4594>.

[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>.

Authors' Addresses

Xiao Min
ZTE Corp.
Nanjing
China

Phone: +86 25 88013062
Email: xiao.min2@zte.com.cn

Greg Mirsky
Ericsson
United States of America

Email: gregimirsky@gmail.com