6MAN Working Group
Internet-Draft
Intended status: Standards Track
Expires: 29 August 2026

X. Min
ZTE Corp.
G. Mirsky
Ericsson
R. Bonica
HPE

25 February 2026

IPv6 Query for Enabled In-situ OAM Capabilities
draft-ietf-6man-icmpv6-ioam-conf-state-10

Abstract

This document describes the application of the mechanism of discovering In-situ OAM (IOAM) capabilities, described in RFC 9359 "Echo Request/Reply for Enabled In Situ OAM (IOAM) Capabilities", in IPv6 networks. IPv6 Node IOAM Query functionality uses the ICMPv6 Query messages, allowing the IOAM encapsulating node to discover the enabled IOAM capabilities of each IOAM transit and IOAM decapsulating node.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

IPv6 encapsulation for In-situ OAM (IOAM) data is defined in [RFC9486], which uses the IPv6 hop-by-hop and destination options to carry IOAM data fields ([RFC9197], [RFC9326]).

As specified in [RFC9359], the echo request/reply can be used by the IOAM encapsulating node to discover the enabled IOAM capabilities at the IOAM transit and decapsulating nodes.

As specified in [RFC4443], the Internet Control Message Protocol for IPv6 (ICMPv6) is an integral part of IPv6. ICMPv6 messages include error messages and informational messages. [RFC4884] defines ICMP Extension Structure by which multi-part ICMPv6 error messages are supported. [I-D.xbm-intarea-icmp-query] updates [RFC4884] by adding two ICMPv6 informational messages, ICMPv6 Query Request message and ICMPv6 Query Response message, to the supporting list of ICMP Extension Structure. The two added ICMPv6 messages are used for a Querying node to query information of a Queried node.

This document describes the IPv6 Node IOAM Query functionality, which uses the ICMPv6 Query messages, allowing the IOAM encapsulating node to discover the enabled IOAM capabilities of each IOAM transit and IOAM decapsulating node.

The IOAM encapsulating node sends an IOAM Query Request to each IOAM transit and decapsulating node. Upon receiving the query, each node executes access control procedures. If access is granted, the node returns an IOAM Query Response indicating its enabled IOAM capabilities. The IOAM Query Request contains an ICMP Extension Structure including one IOAM Query Object, and the IOAM Query Response contains an ICMP Extension Structure including one or more IOAM Capabilities Objects.

Before the IOAM encapsulating node sends the IOAM Query Request, it must know the IPv6 address of each node along the transport path of the data packet to which IOAM data will be added. This can be achieved by executing an ICMPv6/UDP traceroute or by provisioning an explicit path at the IOAM encapsulating node. In an Equal-Cost Multipath (ECMP) scenario, the same values in any ECMP-affecting fields (e.g., the 3-tuple of the Flow Label, Source Address, and Destination Address fields as per [RFC6437]) of the IOAM data packets MUST be populated in the IOAM Query Request, ensuring fate sharing between the IOAM Query Request and the IOAM data packets.

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. IOAM Query Request

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

The IOAM Query 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        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |Version|       Reserved        |          Checksum             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     ~                       IOAM Query Object                       ~
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     ~                          Pad Object                           ~
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: IOAM Query 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:

* ICMPv6 header: The values of Type, Code, Checksum, Identifier, and Sequence Number are the same as specified in [I-D.xbm-intarea-icmp-query].

* Following the ICMPv6 header, it’s an ICMP Extension Structure ([RFC4884]) containing an IOAM Query Object and an Pad Object ([I-D.xbm-intarea-icmp-query]). The IOAM Query Object is also known as Query Request Object in [I-D.xbm-intarea-icmp-query].

3.1. IOAM Query Object

The IOAM Query 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      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                  List of IOAM Namespace-IDs                   ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: IOAM Query Object

Object fields:

* Class-Num: IOAM Query Object. The value is TBD1.

* C-Type: MUST be set to 0 and MUST be ignored upon receipt.

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

* Object payload: Following the IOAM Query Object Header is the IOAM Query Object Payload, which is defined in Section 3.1 of [RFC9359]..

3.2. Examples of the IOAM Query Request

The format of an IOAM Query Request can vary from deployment to deployment.

In a deployment where only the default Namespace-ID is used, the IOAM Query Request 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        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Version|       Reserved        |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |  Class-Num    |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Namespace-ID          |          Zero-padded          |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |  Class-Num    |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                         Extra Padding                         ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: IOAM Query Request of the Default IOAM Namespace

In a deployment where two Namespace-IDs (Namespace-ID1 and Namespace-ID2) are used, the IOAM Query Request 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        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Version|       Reserved        |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |  Class-Num    |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Namespace-ID1         |         Namespace-ID2         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |  Class-Num    |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                               |
    ~                         Extra Padding                         ~
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: IOAM Query Request of the Two IOAM Namespaces

When an IOAM Query Request message is received, the length of the message is determined by the Payload Length field in the IPv6 Header, as specified in [RFC8200].

4. IOAM Query Response

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

The IOAM Query Response 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        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|       Reserved        |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~               List of IOAM Capabilities Objects               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: IOAM Query Response Message

IPv6 Header fields:

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

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

ICMPv6 fields:

* ICMPv6 header: The values of Type, Code, Checksum, Identifier, and Sequence Number are the same as specified in [I-D.xbm-intarea-icmp-query]. Besides, two more values (4) No Matched Namespace-ID and (5) Exceed the minimum IPv6 MTU are defined for Code field. See Section 5 for details.

* Following the ICMPv6 header, it’s an ICMP Extension Structure ([RFC4884]) containing one or more IOAM Capabilities Objects. The IOAM Capabilities Object is also known as Query Response Object in [I-D.xbm-intarea-icmp-query].

4.1. IOAM Capabilities Objects

The 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               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: IOAM Capabilities Object

Object fields:

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

Value	Object Name
TBD2	IOAM Tracing Capabilities Object
TBD3	IOAM Proof of Transit Capabilities Object
TBD4	IOAM Edge-to-Edge Capabilities Object
TBD5	IOAM DEX Capabilities Object
TBD6	IOAM End-of-Domain Object

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

Class-Num	C-Type	C-Type Name
TBD2	0	Reserved
	1	Pre-allocated Tracing
TBD3	0	Reserved
TBD4	0	Reserved
TBD5	0	Reserved
TBD6	0	Reserved

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

* Object payload: Following the IOAM Capabilities Object Header is the IOAM Capabilities Object payload, which is defined in Sections 3.2.1, 3.2.3, 3.2.4, 3.2.5, and 3.2.6 of [RFC9359].

4.2. Examples of the IOAM Query Response

The format of an IOAM Query Response can vary from deployment to deployment.

In a deployment where only the default Namespace-ID is used, the IOAM Pre-allocated Tracing Capabilities and the IOAM Proof of Transit Capabilities are enabled at the IOAM transit node that received an IOAM Query Request, the IOAM Query Response 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        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|       Reserved        |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             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 7: Example 1 of the IOAM Query Response

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 the IOAM Proof of Transit Capabilities are enabled at the IOAM transit node that received an IOAM Query Request, the IOAM Query Response 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        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Version|       Reserved        |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             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 8: Example 2 of the IOAM Query Response

In a deployment where only the default Namespace-ID is used, the IOAM Pre-allocated Tracing Capabilities, the IOAM Proof of Transit Capabilities, and the IOAM Edge-to-Edge Capabilities are enabled at the IOAM decapsulating node that received an IOAM Query Request, the IOAM Query Response 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        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Version|       Reserved        |          Checksum             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             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  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             Length            |   Class-Num   |   C-Type      |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Namespace-ID          |         IOAM-E2E-Type         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |TSF|         Reserved          |           Reserved            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Example 3 of the IOAM Query Response

When an IOAM Query Response message is received, the length of the message is determined by the Payload Length field in the IPv6 Header, as specified in [RFC8200].

5. Code Field Processing

The Code field in the IOAM Query Response MUST be set to (4) No Matched Namespace-ID if any of the following conditions apply:

* The IOAM Query Request does not include any Namespace-ID.

* None of the contained list of IOAM Namespace-IDs is recognized.

* None of the contained list of IOAM Namespace-IDs is enabled.

The Code field in the IOAM Query Response MUST be set to (5) Exceed the minimum IPv6 MTU if the formatted IOAM Query Response packet exceeds the minimum IPv6 MTU (i.e., 1280 octets). In this case, all objects MUST be stripped before forwarding the IOAM Query Response to its destination.

6. IANA Considerations

This document requests the following actions from IANA:

* Add the following Codes to the "Type TBD - Query Response" subregistry:

  - (4) No Matched Namespace-ID

  - (5) Exceed the minimum IPv6 MTU

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

  - (TBD1) IOAM Query Object

* Add the following C-types to the "Sub-types - Class TBD1 - IOAM Query Object" subregistry:

  - (0) Reserved

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

  - (TBD2) IOAM Tracing Capabilities Object

* Add the following C-types to the "Sub-types - Class TBD2 - IOAM Tracing Capabilities Object" subregistry:

  - (0) Reserved

  - (1) Pre-allocated Tracing

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

  - (TBD3) IOAM Proof of Transit Capabilities Object

* Add the following C-types to the "Sub-types - Class TBD3 - IOAM Proof of Transit Capabilities Object" subregistry:

  - (0) Reserved

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

  - (TBD4) IOAM Edge-to-Edge Capabilities Object

* Add the following C-types to the "Sub-types - Class TBD4 - IOAM Edge-to-Edge Capabilities Object" subregistry:

  - (0) Reserved

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

  - (TBD5) IOAM DEX Capabilities Object

* Add the following C-types to the "Sub-types - Class TBD5 - IOAM DEX Capabilities Object" subregistry:

  - (0) Reserved

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

  - (TBD6) IOAM End-of-Domain Object

* Add the following C-types to the "Sub-types - Class TBD6 - IOAM End-of-Domain Object" subregistry:

  - (0) Reserved

All codes mentioned above are assigned on a First Come First Serve (FCFS) basis with a range of 0-255.

7. Security Considerations

Security issues discussed in [I-D.xbm-intarea-icmp-query] and [RFC9359] 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 IPv6 Node IOAM Query functionality. In order to enforce these policies, nodes that support IPv6 Node IOAM Query functionality MUST support the following configuration options:

* Enable/disable IPv6 Node IOAM Query functionality. By default, IPv6 Node IOAM Query 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 the IOAM Query Request messages are permitted.

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

8. Acknowledgements

The authors would like to acknowledge Eric Vyncke, Erik Kline, and Bob Hinden for their valuable suggestions.

The authors would like to acknowledge Chongfeng Xie, Zhenqiang Li, David Lamparter, and Daniel King for their review and helpful comments.

9. References

9.1. Normative References

[I-D.xbm-intarea-icmp-query]
Min, X., Bonica, R., and G. Mirsky, "ICMP Query for IP Node Information", Work in Progress, Internet-Draft, draft-xbm-intarea-icmp-query-00, 24 February 2026, <https://datatracker.ietf.org/doc/html/draft-xbm-intarea-icmp-query-00>.

[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>.

[RFC9359] Min, X., Mirsky, G., and L. Bo, "Echo Request/Reply for Enabled In Situ OAM (IOAM) Capabilities", RFC 9359, DOI 10.17487/RFC9359, April 2023, <https://www.rfc-editor.org/info/rfc9359>.

[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>.

9.2. Informative References

[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>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[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>.

[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>.

Authors' Addresses

Xiao Min
ZTE Corp.
Nanjing
China
Phone: +86 18061680168
Email: xiao.min2@zte.com.cn

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

Ron Bonica
HPE
United States of America
Email: ronald.bonica@hpe.com