Echo Request/Reply for Enabled In-situ OAM Capabilities
draft-xiao-ippm-ioam-conf-state-10

Abstract

   This document describes an extension to the echo request/reply
   mechanisms used in IPv6, MPLS, SFC and BIER environments, which can
   be used within an IOAM domain, allowing the IOAM encapsulating node
   to acquire the enabled IOAM capabilities of each IOAM transit node
   and/or IOAM decapsulating 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 December 9, 2021.

the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction

   The Data Fields for In-situ OAM (IOAM) [I-D.ietf-ippm-ioam-data]
   defines data fields that record OAM information within the packet
   while the packet traverses a particular network domain, which is
   called an IOAM domain. IOAM can be used to complement OAM mechanisms
   based on, e.g., ICMP or other types of probe packets, and IOAM
   mechanisms can be leveraged where mechanisms using, e.g., ICMP do not
   apply or do not offer the desired results.

   As specified in [I-D.ietf-ippm-ioam-data], within the IOAM-domain,
   the IOAM data may be updated by network nodes that the packet
   traverses. The device which adds an IOAM data container to the
   packet to capture IOAM data is called the "IOAM encapsulating node".
   In contrast, the device which removes the IOAM data container is
   referred to as the "IOAM decapsulating node". Nodes within the
   domain that are aware of IOAM data and read and/or write or process
   the IOAM data are called "IOAM transit nodes". Both the IOAM

encapsulating node and the decapsulating node are referred to as
domain edge devices, which can be hosts or network devices.

In order to add the correct IOAM data container to the packet, the
IOAM encapsulating node needs to know the enabled IOAM capabilities
at the IOAM transit nodes and/or the IOAM decapsulating node as a
whole, e.g., how many IOAM transit nodes will add tracing data, and
what kinds of data fields will be added. A centralized controller
could be used in some IOAM deployments. The IOAM encapsulating node
can acquire these IOAM capabilities info from the centralized
controller, through, e.g., NETCONF/YANG, PCEP, or BGP. In the IOAM
deployment scenario where there is no centralized controller,
NETCONF/YANG or IGP may be used for the IOAM encapsulating node to
acquire these IOAM capabilities info, however, whether NETCONF/YANG
or IGP has some limitations as follows.

o When NETCONF/YANG is used in this scenario, each IOAM
   encapsulating node (including the host when it takes the role of
   an IOAM encapsulating node) needs to implement a NETCONF Client,
   each IOAM transit node and IOAM decapsulating node (including the
   host when it takes the role of an IOAM decapsulating node) needs
   to implement a NETCONF Server, the complexity can be an issue.
   Furthermore, each IOAM encapsulating node needs to establish
   NETCONF Connection with each IOAM transit node and IOAM
   decapsulating node, the scalability can be an issue.

o When IGP is used in this scenario, the IGP domain and an IOAM
   domain don't always have the same coverage. For example, when the
   IOAM encapsulating node or the IOAM decapsulating node is a host,
   the availability can be an issue. Furthermore, it might be too
   challenging to reflect IOAM capabilities at the IOAM transit node
   and/or the IOAM decapsulating node if these are controlled by a
   local policy depending on the identity of the IOAM encapsulating
   node.

This document describes an extension to the echo request/reply
mechanisms used in IPv6, MPLS, SFC and BIER environments, which can
be used within an IOAM domain where no Centralized Controller exists,
allowing the IOAM encapsulating node to acquire the enabled IOAM
capabilities of each IOAM transit node and/or IOAM decapsulating
node.

The following documents contain references to the echo request/reply
mechanisms used in IPv6, MPLS, SFC and BIER environments:

o [RFC4443] ("Internet Control Message Protocol (ICMPv6) for the
   Internet Protocol Version 6 (IPv6) Specification"), [RFC4884]

("Extended ICMP to Support Multi-Part Messages") and [RFC8335] ("PROBE: A Utility for Probing Interfaces")

o [RFC8029] ("Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures")

o [I-D.ietf-sfc-multi-layer-oam] ("Active OAM for Service Function Chains in Networks")

o [I-D.ietf-bier-ping] ("BIER Ping and Trace")

This feature described in this document is assumedly applied to explicit path (strict or loose), because the precondition for this feature to work is that the echo request reaches each IOAM transit node as live traffic traverses.

2. Conventions

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

BIER: Bit Index Explicit Replication

BGP: Border Gateway Protocol

E2E: Edge to Edge

ICMP: Internet Control Message Protocol

IGP: Interior Gateway Protocol

IOAM: In-situ Operations, Administration, and Maintenance

LSP: Label Switched Path

MPLS: Multi-Protocol Label Switching

MBZ: Must Be Zero

MTU: Maximum Transmission Unit

   NTP: Network Time Protocol

   OAM: Operations, Administration, and Maintenance

   PCEP: Path Computation Element (PCE) Communication Protocol

   POSIX: Portable Operating System Interface

   POT: Proof of Transit

   PTP: Precision Time Protocol

   SFC: Service Function Chain

   TTL: Time to Live

3. IOAM Capabilities Formats

3.1. IOAM Capabilities Query TLV in the Echo Request

   In echo request IOAM Capabilities Query uses TLV (Type-Length-Value
   tuple) which have 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 = IOAM Capabilities Query|            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                                                               .
   .                      List of Namespace-IDs                    .
   .                                                               .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
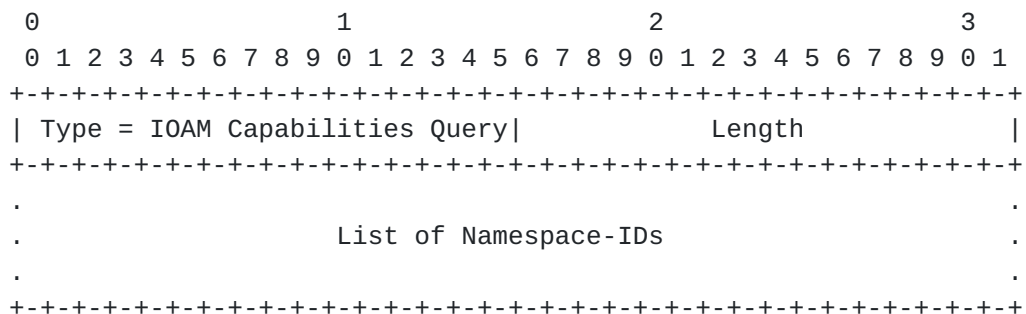
        Figure 1: IOAM Capabilities Query TLV in the Echo Request

   When this TLV is present in the echo request sent by an IOAM
   encapsulating node, it means that the IOAM encapsulating node
   requests the receiving node to reply with its enabled IOAM
   capabilities. If there is no IOAM capability to be reported by the
   receiving node, then this TLV SHOULD be ignored by the receiving
   node, which means the receiving node SHOULD send echo reply without
   IOAM capabilities or no echo reply, in the light of whether the echo
   request includes other TLV than IOAM Capabilities Query TLV. List of
   Namespace-IDs MAY be included in this TLV of the echo request. In
   that case, the IOAM encapsulating node requests only the IOAM
   capabilities that match one of the Namespace-IDs. The Namespace-ID
   has the same definition as what's specified in
   [I-D.ietf-ippm-ioam-data].

Type is set to the value that identifies it as an IOAM Capabilities
Query TLV.

Length is the length of the TLV's Value field in octets, including a
List of Namespace-IDs.

Value field of this TLV is zero-padded to align to a 4-octet
boundary.

3.2. IOAM Capabilities Response TLV in the Echo Reply

In echo reply IOAM Capabilities Response uses TLV which have 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 = IOAM Capabilities Resp |            Length             |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  .                                                               .
  .                        List of Sub-TLVs                       .
  .                                                               .
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
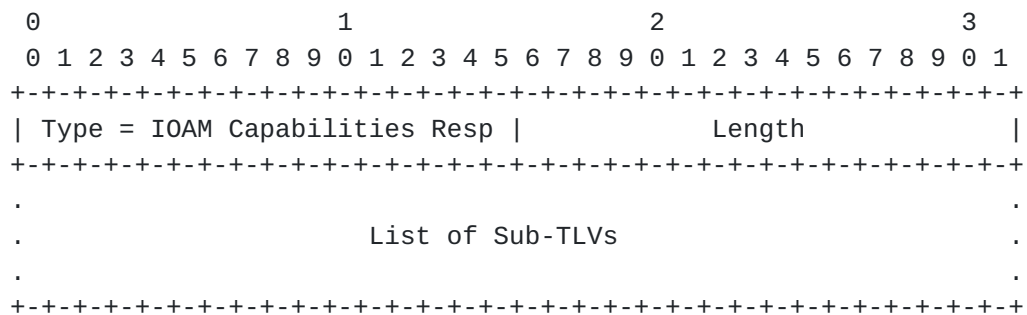
        Figure 2: IOAM Capabilities Response TLV in the Echo Reply

When this TLV is present in the echo reply sent by an IOAM transit
node and/or an IOAM decapsulating node, it means that the IOAM
function is enabled at this node, and this TLV contains the enabled
IOAM capabilities of the sender. A list of Sub-TLVs which contains
the IOAM capabilities SHOULD be included in this TLV of the echo
reply. Note that the IOAM encapsulating node or the IOAM
decapsulating node can also be an IOAM transit node.

Type is set to the value that identifies it as an IOAM Capabilities
Response TLV.

Length is the length of the TLV's Value field in octets, including a
List of Sub-TLVs.

Value field of this TLV or any Sub-TLV is zero-padded to align to a
4-octet boundary. Based on the data fields for IOAM, specified in
[I-D.ietf-ippm-ioam-data] and [I-D.ietf-ippm-ioam-direct-export], six
kinds of Sub-TLVs are defined in this document. The same type of the
sub-TLV MAY be in the IOAM Capabilities Response TLV more than once
only if with a different Namespace-ID.

3.2.1. IOAM Pre-allocated Tracing Capabilities sub-TLV

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Sub-type = Pre-allocated trace |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               IOAM-Trace-Type                 |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          |          Egress_MTU           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Egress_if_id (short or wide format)        ......            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

        Figure 3: IOAM Pre-allocated Tracing Capabilities Sub-TLV

   When this sub-TLV is present in the IOAM Capabilities Response TLV,
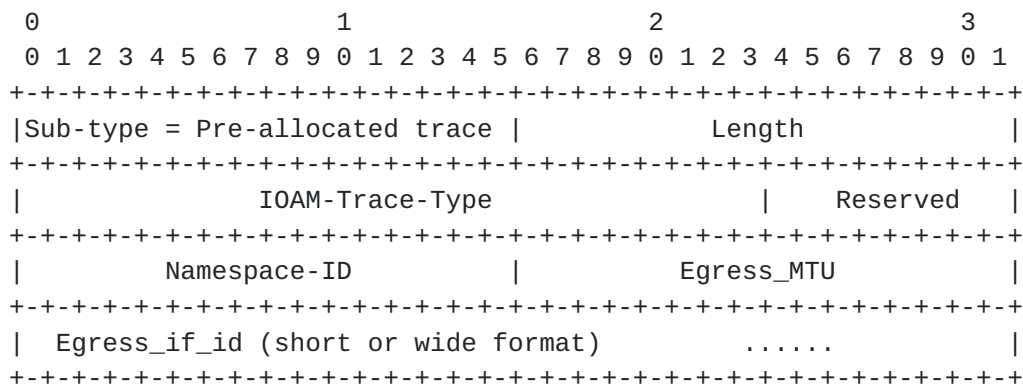   it means that the sending node is an IOAM transit node and IOAM pre-
   allocated tracing function is enabled at this IOAM transit node.

   Sub-type is set to the value that identifies it as an IOAM Pre-
   allocated Tracing Capabilities sub-TLV.

   Length is the length of the sub-TLV's Value field in octets. If
   Egress_if_id is in the short format, which is 16 bits long, it MUST
   be set to 10. If Egress_if_id is in the wide format, which is 32
   bits long, it MUST be set to 12.

   IOAM-Trace-Type field has the same definition as what's specified in
   section 5.4 of [I-D.ietf-ippm-ioam-data].

   Reserved field is reserved for future use and MUST be set to zero.

   Namespace-ID field has the same definition as what's specified in
   section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the
   Namespace-IDs listed in the IOAM Capabilities Query TLV of echo
   request.

   Egress_MTU field has 16 bits and specifies the MTU of the egress
   direction out of which the sending node would forward the received
   echo request, it should be the MTU of the egress interface or the MTU
   between the sending node and the downstream IOAM transit node.

   Egress_if_id field has 16 bits (in short format) or 32 bits (in wide
   format) and specifies the identifier of the egress interface out of
   which the sending node would forward the received echo request.

3.2.2. IOAM Incremental Tracing Capabilities sub-TLV

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sub-type = Incremental trace  |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               IOAM-Trace-Type                 |   Reserved    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Namespace-ID         |          Egress_MTU           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Egress_if_id (short or wide format)          ......          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
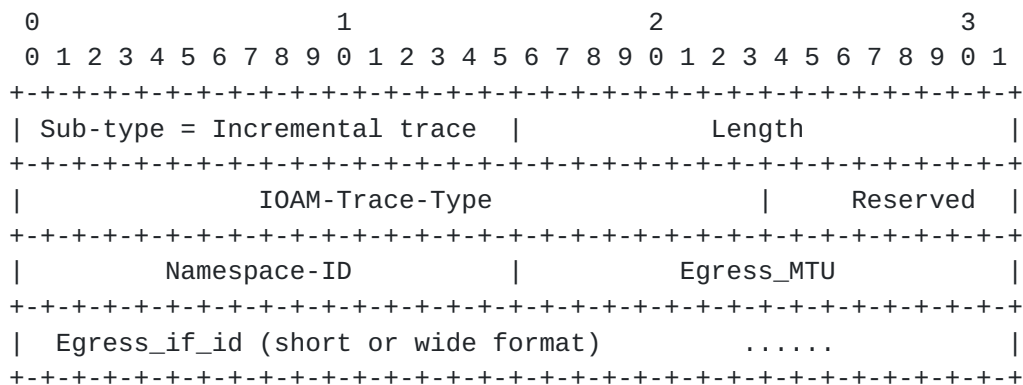
Figure 4: IOAM Incremental Tracing Capabilities Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV, it means that the sending node is an IOAM transit node and IOAM incremental tracing function is enabled at this IOAM transit node.

Sub-type is set to the value that identifies it as an IOAM Incremental Tracing Capabilities sub-TLV.

Length is the length of the sub-TLV's Value field in octets. If Egress_if_id is in the short format, which is 16 bits long, it MUST be set to 10. If Egress_if_id is in the wide format, which is 32 bits long, it MUST be set to 12.

IOAM-Trace-Type field has the same definition as what's specified in section 5.4 of [I-D.ietf-ippm-ioam-data].

Reserved field is reserved for future use and MUST be set to zero.

Namespace-ID field has the same definition as what's specified in section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

Egress_MTU field has 16 bits and specifies the MTU of the egress direction out of which the sending node would forward the received echo request, it should be the MTU of the egress interface or the MTU between the sending node and the downstream IOAM transit node.

Egress_if_id field has 16 bits (in short format) or 32 bits (in wide format) and specifies the identifier of the egress interface out of which the sending node would forward the received echo request.

3.2.3. IOAM Proof of Transit Capabilities sub-TLV

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sub-type = POT Capabilities  |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          | IOAM-POT-Type |P|SoR|Reserved |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
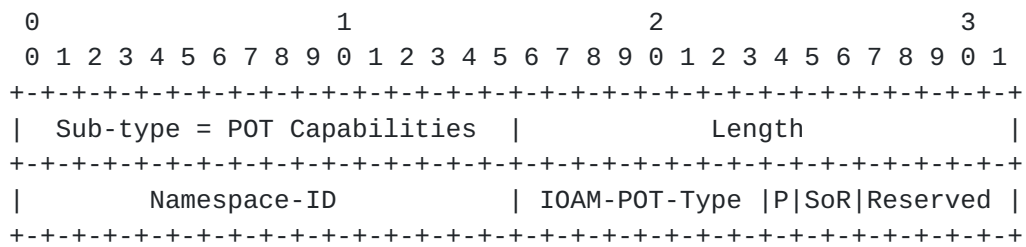
        Figure 5: IOAM Proof of Transit Capabilities Sub-TLV

   When this sub-TLV is present in the IOAM Capabilities Response TLV,
   it means that the sending node is an IOAM transit node and IOAM proof
   of transit function is enabled at this IOAM transit node.

   Sub-type is set to the value that identifies it as an IOAM Proof of
   Transit Capabilities sub-TLV.

   Length is the length of the sub-TLV's Value field in octets and MUST
   be set to 4.

   Namespace-ID field has the same definition as what's specified in
   section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the
   Namespace-IDs listed in the IOAM Capabilities Query TLV of echo
   request.

   IOAM-POT-Type field and P bit have the same definition as what's
   specified in section 5.5 of [I-D.ietf-ippm-ioam-data]. If the IOAM
   encapsulating node receives IOAM-POT-Type and/or P bit values from an
   IOAM transit node that are different from its own, then the IOAM
   encapsulating node MAY choose to abandon the proof of transit
   function or to select one kind of IOAM-POT-Type and P bit, it's based
   on the policy applied to the IOAM encapsulating node.

   SoR field has two bits, which means the size of "Random" and
   "Cumulative" data that are specified in section 5.5 of
   [I-D.ietf-ippm-ioam-data]. This document defines SoR as follow:

      0b00 means 64-bit "Random" and 64-bit "Cumulative" data.

      0b01~0b11: Reserved for future standardization

   Reserved field is reserved for future use and MUST be set to zero.

3.2.4. IOAM Edge-to-Edge Capabilities sub-TLV

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sub-type = E2E Capabilities  |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          |         IOAM-E2E-Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|TSF|TSL|       Reserved        |             MBZ               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
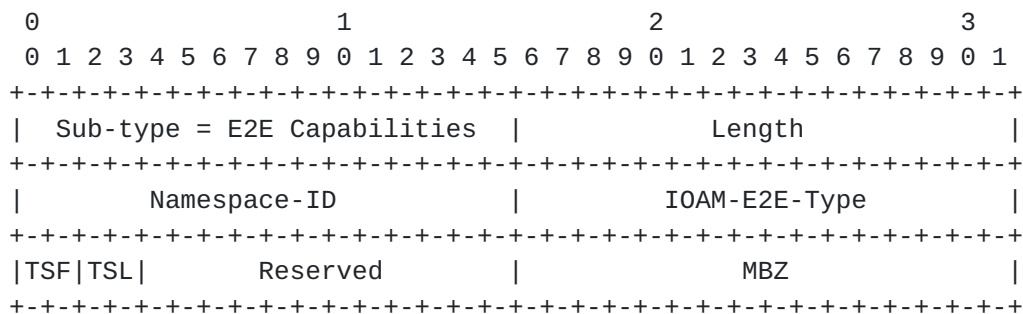
             Figure 6: IOAM Edge-to-Edge Capabilities Sub-TLV

   When this sub-TLV is present in the IOAM Capabilities Response TLV,
   it means that the sending node is an IOAM decapsulating node and IOAM
   edge-to-edge function is enabled at this IOAM decapsulating node.
   That is to say, if the IOAM encapsulating node receives this sub-TLV,
   the IOAM encapsulating node can determine that the node which sends
   this sub-TLV is an IOAM decapsulating node.

   Sub-type is set to the value that identifies it as an IOAM Edge-to-
   Edge Capabilities sub-TLV.

   Length is the length of the sub-TLV's Value field in octets and MUST
   be set to 8.

   Namespace-ID field has the same definition as what's specified in
   section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the
   Namespace-IDs listed in the IOAM Capabilities Query TLV of echo
   request.

   IOAM-E2E-Type field has the same definition as what's specified in
   section 5.6 of [I-D.ietf-ippm-ioam-data].

   TSF field specifies the timestamp format used by the sending node.
   This document defines TSF as follow:

      0b00: PTP timestamp format

      0b01: NTP timestamp format

      0b10: POSIX timestamp format

      0b11: Reserved for future standardization

   TSL field specifies the timestamp length used by the sending node.
   This document defines TSL as follow.

When the TSF field is set to 0b00, which indicates the PTP
timestamp format, the values of the TSL field are interpreted as
follows:

0b00: 64-bit PTPv1 timestamp as defined in IEEE1588-2008
[IEEE1588v2]

0b01: 80-bit PTPv2 timestamp as defined in IEEE1588-2008
[IEEE1588v2]

0b10~0b11: Reserved for future standardization

When the TSF field is set to 0b01, which indicates the NTP
timestamp format, the values of the TSL field are interpreted as
follows:

0b00: 32-bit NTP timestamp as defined in NTPv4 [RFC5905]

0b01: 64-bit NTP timestamp as defined in NTPv4 [RFC5905]

0b10: 128-bit NTP timestamp as defined in NTPv4 [RFC5905]

0b11: Reserved for future standardization

When the TSF field is set to 0b10 or 0b11, the TSL field would be
ignored.

Reserved field is reserved for future use and MUST be set to zero.

3.2.5. IOAM DEX Capabilities sub-TLV

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Sub-type = DEX Capabilities  |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              IOAM-Trace-Type                  |    Reserved   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          |            Reserved           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
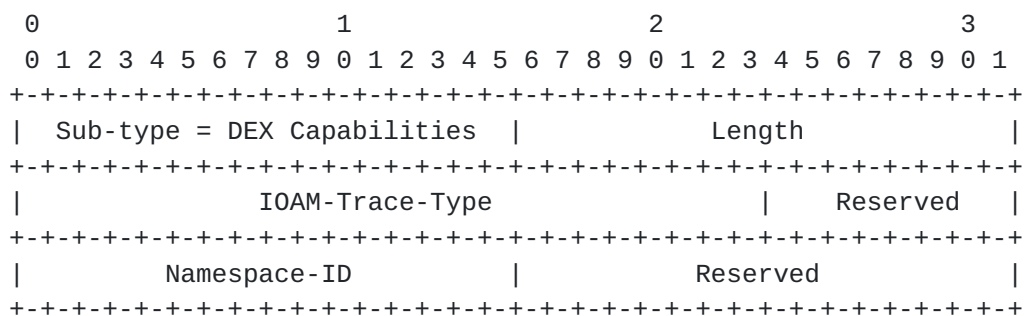
Figure 7: IOAM DEX Capabilities Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV,
it means that the sending node is an IOAM transit node and the IOAM
DEX function is enabled at this IOAM transit node.

Sub-type is set to the value that identifies it as an IOAM DEX
Capabilities sub-TLV.

Length is the length of the sub-TLV's Value field in octets and MUST
be set to 8.

IOAM-Trace-Type field has the same definition as what's specified in
section 3.2 of [I-D.ietf-ippm-ioam-direct-export].

Namespace-ID field has the same definition as what's specified in
section 3.2 of [I-D.ietf-ippm-ioam-direct-export], it should be one
of the Namespace-IDs listed in the IOAM Capabilities Query TLV of
echo request.

Reserved field is reserved for future use and MUST be set to zero.

3.2.6. IOAM End-of-Domain sub-TLV

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Sub-type = End of Domain    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          |             MBZ               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
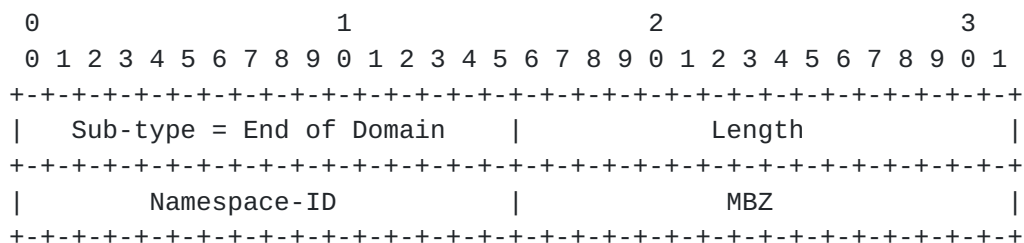
Figure 8: IOAM End of Domain Sub-TLV

When this sub-TLV is present in the IOAM Capabilities Response TLV,
it means that the sending node is an IOAM decapsulating node. That
is to say, if the IOAM encapsulating node receives this sub-TLV, the
IOAM encapsulating node can determine that the node which sends this
sub-TLV is an IOAM decapsulating node. When the IOAM Edge-to-Edge
Capabilities sub-TLV is present in the IOAM Capabilities Response TLV
sent by the IOAM decapsulating node, the IOAM End-of-Domain sub-TLV
doesn't need to be present in the same IOAM Capabilities Response
TLV, otherwise the End-of-Domain sub-TLV MUST be present in the IOAM
Capabilities Response TLV sent by the IOAM decapsulating node. Both
the IOAM Edge-to-Edge Capabilities sub-TLV and the IOAM End-of-Domain
sub-TLV can be used to indicate that the sending node is an IOAM
decapsulating node. It's recommended to include only the IOAM Edge-
to-Edge Capabilities sub-TLV if IOAM edge-to-edge function is enabled
at this IOAM decapsulating node.

Sub-type is set to the value that identifies it as an IOAM End of
Domain sub-TLV.

Length is the length of the sub-TLV's Value field in octets and MUST be set to 4.

Namespace-ID field has the same definition as what's specified in section 5.3 of [I-D.ietf-ippm-ioam-data], it should be one of the Namespace-IDs listed in the IOAM Capabilities Query TLV of echo request.

[4]. Operational Guide

Once the IOAM encapsulating node is triggered to acquire the enabled IOAM capabilities of each IOAM transit node and/or IOAM decapsulating node, the IOAM encapsulating node will send echo requests that include the IOAM Capabilities Query TLV. First with TTL equal to 1 to reach the nearest node, which may be an IOAM transit node or not. Then with TTL equal to 2 to reach the second nearest node, which also may be an IOAM transit node or not. And further, increasing by 1 the TTL every time the IOAM encapsulating node sends a new echo request, until the IOAM encapsulating node receives an echo reply sent by the IOAM decapsulating node, which should contain the IOAM Capabilities Response TLV including the IOAM Edge-to-Edge Capabilities sub-TLV or the IOAM End-of-Domain sub-TLV. Alternatively, if the IOAM encapsulating node knows exactly all the IOAM transit nodes and/or IOAM decapsulating node beforehand, once the IOAM encapsulating node is triggered to acquire the enabled IOAM capabilities, it can send an echo request to each IOAM transit node and/or IOAM decapsulating node directly, without TTL expiration.

The IOAM encapsulating node may be triggered by the device administrator, the network management system, the network controller, or even the live user traffic. The specific triggering mechanisms are outside the scope of this document.

Each IOAM transit node and/or IOAM decapsulating node that receives an echo request containing the IOAM Capabilities Query TLV will send an echo reply to the IOAM encapsulating node, and within the echo reply, there should be an IOAM Capabilities Response TLV containing one or more sub-TLVs. The IOAM Capabilities Query TLV contained in the echo request would be ignored by the receiving node that is unaware of IOAM.

[5]. Security Considerations

Queries and responses about the state of an IOAM domain should be processed only from a trusted source. An unauthorized query MUST be discarded by an implementation that supports this specification. Similarly, unsolicited echo response with the IOAM Capabilities TLV MUST be discarded. Authentication of echo request/reply that

includes the IOAM Capabilities TLV is one of methods of the integrity
protection. Implementations could also provide a means of filtering
based on the source address of the received echo request/reply. The
integrity protection for IOAM capabilities information collection can
also be achieved using mechanisms in the underlay data plane. For
example, if the underlay is an IPv6 network, IP Authentication Header
[RFC4302] or IP Encapsulating Security Payload Header [RFC4303] can
be used to provide integrity protection.

Information about the state of the IOAM domain collected in the IAOM
Capabilities TLV is confidential. An implementation can use secure
transport to provide privacy protection. For example, if the
underlay is an IPv6 network, confidentiality can be achieved using
the IP Encapsulating Security Payload Header [RFC4303].

6. IANA Considerations

This document requests the following IANA Actions.

IANA is requested to create a registry group named "In-Situ OAM
(IOAM) Capabilities Parameters".

This group will include the following registries:

o IOAM SoR Capability

o IOAM TSF+TSL Capability

New registries in this group can be created via RFC Required process
as per [RFC8126].

The subsequent sub-sections detail the registries herein contained.

Considering the TLVs/sub-TLVs defined in this document would be
carried in different kinds of Echo Request/Reply message, such as
ICMPv6 or LSP Ping, it is intended that the registries for Type and
sub-Type would be requested in subsequent documents.

6.1. IOAM SoR Capability Registry

This registry defines 4 code points for the IOAM SoR Capability field
for identifying the size of "Random" and "Cumulative" data as
explained in section 5.5 of [I-D.ietf-ippm-ioam-data]. The following
code points are defined in this draft:

```
   SoR         Description
   ----        -----------
   0b00        64-bit "Random" and 64-bit "Cumulative" data
```

0b01 - 0b11 are available for assignment via RFC Required process as
per [RFC8126].

6.2. IOAM TSF+TSL Capability Registry

This registry defines 3 code points for the IOAM TSF Capability field
for identifying the timestamp format as explained in section 6 of
[I-D.ietf-ippm-ioam-data].

o When the code point for the IOAM TSF Capability field equals 0b00
   which means PTP timestamp format, this registry defines 2 code
   points for the IOAM TSL Capability field for identifying the
   timestamp length.

o When the code point for the IOAM TSF Capability field equals 0b01
   which means NTP timestamp format, this registry defines 3 code
   points for the IOAM TSL Capability field for identifying the
   timestamp length.

The following code points are defined in this draft:

```
TSF         TSL         Description
----        ----        -----------
0b00                    PTP Timestamp Format
            0b00        64-bit PTPv1 timestamp
            0b01        80-bit PTPv2 timestamp
0b01                    NTP Timestamp Format
            0b00        32-bit NTP timestamp
            0b01        64-bit NTP timestamp
            0b10        128-bit NTP timestamp
0b10                    POSIX Timestamp Format
```

Unassigned code points of TSF+TSL are available for assignment via
RFC Required process as per [RFC8126].

7. Acknowledgements

The authors would like to acknowledge Tianran Zhou, Dhruv Dhody,
Frank Brockners and Cheng Li for their careful review and helpful
comments.

The authors appreciate the f2f discussion with Frank Brockners on
this document.

The authors would like to acknowledge Tommy Pauly and Ian Swett for
their good suggestion and guidance.

8. References

8.1. Normative References

[I-D.ietf-ippm-ioam-data]
          Brockners, F., Bhandari, S., and T. Mizrahi, "Data Fields
          for In-situ OAM", draft-ietf-ippm-ioam-data-12 (work in
          progress), February 2021.

[I-D.ietf-ippm-ioam-direct-export]
          Song, H., Gafni, B., Zhou, T., Li, Z., Brockners, F.,
          Bhandari, S., Sivakolundu, R., and T. Mizrahi, "In-situ
          OAM Direct Exporting", draft-ietf-ippm-ioam-direct-
          export-03 (work in progress), February 2021.

[IEEE1588v2]
          IEEE, "IEEE Std 1588-2008 - IEEE Standard for a Precision
          Clock Synchronization Protocol for Networked Measurement
          and Control Systems", IEEE Std 1588-2008, 2008,
          <http://standards.ieee.org/findstds/
          standard/1588-2008.html>.

[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>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,
          "Network Time Protocol Version 4: Protocol and Algorithms
          Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010,
          <https://www.rfc-editor.org/info/rfc5905>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for
          Writing an IANA Considerations Section in RFCs", BCP 26,
          RFC 8126, DOI 10.17487/RFC8126, June 2017,
          <https://www.rfc-editor.org/info/rfc8126>.

[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>.

8.2. Informative References

[I-D.ietf-bier-ping]
          Kumar, N., Pignataro, C., Akiya, N., Zheng, L., Chen, M.,
          and G. Mirsky, "BIER Ping and Trace", draft-ietf-bier-
          ping-07 (work in progress), May 2020.

[I-D.ietf-sfc-multi-layer-oam]
          Mirsky, G., Meng, W., Khasnabish, B., and C. Wang, "Active
          OAM for Service Function Chaining", draft-ietf-sfc-multi-
          layer-oam-10 (work in progress), March 2021.

[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>.

[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>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N.,
          Aldrin, S., and M. Chen, "Detecting Multiprotocol Label
          Switched (MPLS) Data-Plane Failures", RFC 8029,
          DOI 10.17487/RFC8029, March 2017,
          <https://www.rfc-editor.org/info/rfc8029>.

[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
ZTE Corp.
USA

Email: gregory.mirsky@ztetx.com

Lei Bo
China Telecom
Beijing
China

Phone: +86 10 50902903
Email: leibo@chinatelecom.cn