In-situ OAM Data Type Extension
draft-song-ippm-ioam-data-extension-00

Abstract

   This document describes a proposal which extends in-situ OAM to
   support potential future standard tracing data in addition to those
   currently defined. We provide use cases to motivate our proposal
and
   base the modifications on the latest in-situ OAM header format
   specification.

Table of Contents

1. Motivation for Data Type Extension

   In-situ OAM (iOAM) [I-D.brockners-inband-oam-requirements] records
   OAM information within user packets while the packets traverse a
   network. The data types and data formats for in-situ OAM data
   records have been defined in [I-D.ietf-ippm-ioam-data].

   Currently 12 data types and associated formats (including wide
format
   and short format of the same data) are defined in
   [I-D.ietf-ippm-ioam-data] . The presence of data is indicated by a
   16-bit bitmap in the "OAM-Trace-Type" field.

   In the current specification only five bits are left to identify new
   standard data types. Moreover, some data is forced to be bundled
   together as a single unit to save bitmap space and pack data to the
   ideal size (e.g., the hop limit and the node id are bundled, and the
   ingress interface id and the egress interface id are bundled),
   regardless of the fact that an application may only ask for a part
of
   the data. Last but not the least, each data is forced to be 4-byte
   aligned for easier access, resulting in waste of header space in
many
   cases.

Since the data plane bandwidth, the data plane packet processing, and
   the management plane data handling are all precious yet scarce
   resource, the scheme should strive to be simple and precise. The
   application should be able to control the exact type and format of
   data it needs to collect and analyze. It is conceivable that more
   types of data may be introduced in the future. However, the current
   scheme cannot support it after all the bits in the bitmap are used
   up.

   For example, when a flow traverses a series of middleboxes (e.g.,
   Firewall, NAT, and load balancer), its identity (e.g., the 5-tuple)

   is often altered, which makes the OAM system lose track of the flow
   trace. In this case, we may want to copy some of the original
packet
   header fields into the iOAM header so the original flow can be
   identified at any point of the network.

   For another example, in wireless, mobile, and optical network
   environments, some physical data associated with a flow (e.g.,
power,
   temperature, signal strength, GPS location) need to be collected to
   monitor the service performance.

   Both cases require new iOAM data types. More examples are listed in
   Section 2.2.

   Currently, bit 7 is used to indicate the presence of variable length
   opaque state snapshot data; Bit 5 and bit 10 are used to indicate
the
   presence of the application specific data. While these data fields
   can be used to store arbitrary data, the data is difficult to be
   standardized and another schema is needed to decode the data, which
   may lead to low data plane performance as well as interoperability
   issues.

2. Scalable Data Type Extension

   Based on the observation in Section 1, we propose a method for data
   type encoding which can solve the current limitation and address
   future data requirements.

2.1. Data Type Bitmap

   Bitmap is simple and efficient data structure for high performance
   data plane implementation. The base bitmap size is kept to be 16
   bits. We use one bit to indicate a single type of data in a single
   format. The last bit in the bitmap (i.e., bit 15), if set, is used
   to indicate the presence of the next data type bitmap, which is 32
   bits long. In the second bitmap, bit 31 is again reserved to
   indicate a third bitmap, and so on. With each extra bitmap, 31 more
   data types can be defined.

   Figure 1 shows an example of the in-situ OAM header format with two
   extended OAM trace type fields. Except the OAM Trace Type fields,
   all other fields remain the same as defined in
   [I-D.ietf-ippm-ioam-data].

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Base OAM Trace Type     |1|NodeLen| Flags | Octets-left |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Extended OAM Trace Type 1                |1|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Extended OAM Trace Type 2                |0|
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                                                             |
    |                    Node Data List []                        |
    |                                                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
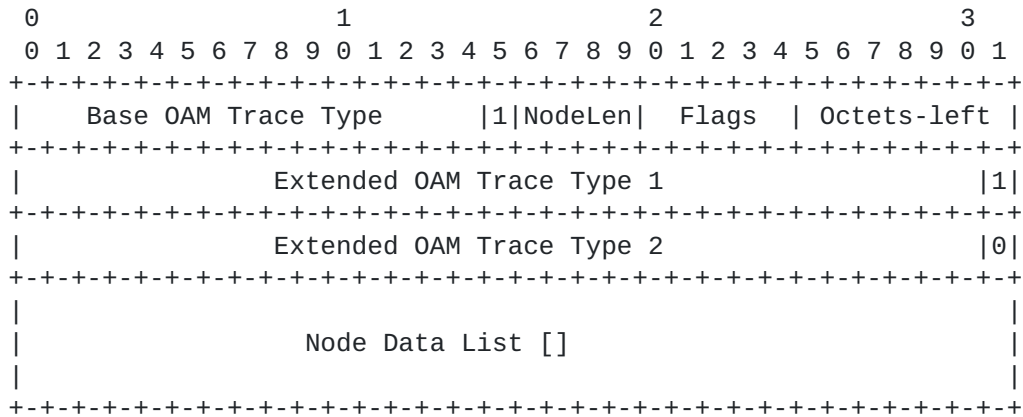
Figure 1: Extended OAM Trace Type Header Format

The specification of the Base OAM Trace Type is the same as the OAM
Trace Type in [I-D.ietf-ippm-ioam-data] except the last bit, which
is
defined as follows:

o Bit 15: When set indicates presence of next bit map.

The OAM trace type fields are labeled as Base OAM Trace Type,
Extended OAM Trace Type 1, Extended OAM Trace Type 2, and so on.
The
Base OAM Trace Type is always present. If no data type is asked by
the application in Extended OAM Trace Type n and beyond, then the
last bit in the previous bitmap is set to 1 and these extended
fields
are not included in the header. On the other hand, to eliminate
ambiguity, if any data is asked for by the application in Extended
OAM Trace Type n, then Extended OAM Trace Type 1 to (n-1) must be
included in the header, even though no data type in these bitmaps
are
needed (i.e., all zero bitmap except the last bit).

The actual data in a node is packed together in the same order as
listed in the OAM Trace Type bitmap. Each node is padded to be the
multiple of 4 bytes.

2.2. Use Cases

New types of data can be potentially added and standardized, which
demand new bits allocated in the OAM Trace Type bitmaps. Some
examples are listed here.

o Metered flow bandwidth.

o Time gap between two consecutive flow packets.

o Remaining time budget to the packet delivery deadline.

o Buffer occupancy on the Node.

o Queue depth on each level of hierarchical QoS queues.

o Packet jitter at the Node.

o Current packet IP addresses.

o Current packet port numbers.

o Time using different network timing protocol.

o Other node statistics.

2.3. Consideration for Efficient Data Packing

The length of each data must be the multiple of 2 bytes. However,
allowing different data type to have different length, while
efficient in storage, makes data alignment and packing difficult.

If we can define the maximum number of data types that can be
carried
per packet, the offset of each data in the node can be pre-
calculated
and carried in the iOAM header. The overhead can be justified by
the
overall space saving of the node data list. Otherwise, each data's
offset in the node must be calculated in each device, with the help
of a table which stores the size of each data type. We can also
arrange the bitmap to reflect the data availability order in the
system (e.g., the bit for egress_if_id must be after the bit for
ingress_if_id), so in a pipeline-based system, the required data can
be packed one after one.

2.4. Alternative Data Extension Possibilities

Bitmap is simple and support parallel processing in hardware.
However, it is not the only option to support data type extension.
For example, cascaded TLV can be used to support arbitrary number of
new data types. This can be implemented by using a flag bit to
indicate the presence of extra data types and packing the number of
types and the list of the type IDs after the trace option header.
The corresponding data is therefore added in each node data list in
the order as its type ID is listed in the extended trace option
header.

3. Security Considerations

   There is no extra security considerations beyond those have been
   identified by the original in-situ OAM proposals.

4. IANA Considerations

   This memo includes no request to IANA.

5. Acknowledgments

   We would like to thank Frank Brockners, Carlos Pignataro, and
Shwetha
   Bhandari for helpful comments and suggestions.

6. Contributors

   The document is inspired by numerous discussions with James N.
   Guichard. He also provided significant comments and suggestions to
   help improve this document.

7. Informative References

   [I-D.brockners-inband-oam-requirements]
              Brockners, F., Bhandari, S., Dara, S., Pignataro, C.,
              Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi,
              T., <>, P., and r. remy@barefootnetworks.com,
              "Requirements for In-situ OAM", draft-brockners-inband-
              oam-requirements-03 (work in progress), March 2017.

   [I-D.ietf-ippm-ioam-data]
              Brockners, F., Bhandari, S., Pignataro, C., Gredler, H.,
              Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov,
              P., Chang, R., and d. daniel.bernier@bell.ca, "Data
Fields
              for In-situ OAM", draft-ietf-ippm-ioam-data-00 (work in
              progress), September 2017.

Authors' Addresses

   Haoyu Song (editor)
   Huawei
   2330 Central Expressway
   Santa Clara, 95050
   USA

   Email: haoyu.song@huawei.com

Tianran Zhou
Huawei
156 Beiqing Road
Beijing, 100095
P.R. China

Email: zhoutianran@huawei.com