In-situ OAM

IPPM

November 6th, 2018

There are quite a few IOAM related drafts…

- IOAM data fields definition
  - ✶ draft-ietf-ippm-ioam-data
- IOAM data export
  - ✶ draft-spiegel-ippm-ioam-rawexport-01
- IOAM data fields encapsulation for different protocols:
  - ✶ Generic encap for protocols with Ethertype (GRE, etc): draft-weis-ippm-ioam-eth
  - VXLAN GPE: draft-brockners-ippm-ioam-vxlan-gpe
  - GENEVE: draft-brockners-ippm-ioam-geneve
  - IPv6: draft-ioametal-ippm-6man-ioam-ipv6-options
  - SRv6: draft-ali-spring-ioam-srv6
  - SR-MPLS: draft-gandhi-spring-ioam-sr-mpls
  - NSH encapsulation - draft-ietf-sfc-ioam-nsh
- Other IOAM related work
  - Proof of transit - draft-ietf-sfc-proof-of-transit
  - YANG Data Model for In-Situ OAM - draft-zhou-ippm-ioam-yang
  - ...

✶ Covered in this presentation

Data Fields for In-situ OAM

draft-ietf-ippm-ioam-data-04

IPPM

November 6th, 2018

draft-ietf-ippm-ioam-data -03 to -04 updates

1. Introduction of IOAM Namespace Identifier
    - Concept of Namespace ID was already discussed at IPPM WG meeting in IETF 102
    - Example use cases
        - Distinguish different operational domains
        - Provide additional context for IOAM data fields and thus ensure that IOAM data is unique (e.g. for node-id)
        - Identify different sets of devices (e.g., different types of devices) in a deployment

IOAM Namespace (see section 4.1)

IOAM data fields are defined within an IOAM namespace. An IOAM namespace is identified by a 16-bit namespace identifier (Namespace-ID). Namespace identifiers MUST be present and populated in all IOAM option headers. The Namespace-ID value is divided into two sub-ranges:

- An operator-assigned range from 0x0001 to 0x7FFF
- An IANA-assigned range from 0x8000 to 0xFFFF
- The Namespace-ID value of 0x0000 is defined as the default value and MUST be known to all the nodes implementing IOAM

Namespace ID as a data field

IOAM Trace Option headers (pre-allocated and incremental)

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          |NodeLen  | Flags | RemainingLen|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               IOAM-Trace-Type                 |  Reserved     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IOAM POT Option header

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Namespace-ID           |IOAM POT Type  | IOAM POT flags|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IOAM E2E Option header

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Namespace-ID          |         IOAM-E2E-Type         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

draft-ietf-ippm-ioam-data -03 to -04 updates (contd.)

2. IOAM Trace Options
   - Introduction and rearrangement of IOAM header fields allowed to also grow IOAM trace-type field to 24-bits
   - Need expressed for larger trace-type field to future proof definition in earlier IPPM meetings

IOAM Trace Option headers (pre-allocated and incremental)

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Namespace-ID           |NodeLen  | Flags | RemainingLen|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               IOAM-Trace-Type                 |  Reserved     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

draft-ietf-ippm-ioam-data -03 to -04 updates (contd.)

3. “app-data” data field (trace option) renamed to “namespace specific 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     namespace specific data                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4. “buffer occupancy” data field added

```
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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        buffer occupancy                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- indicate the current status of the buffer occupancy.
- current number of memory buffers used by the set of queues that share a common buffer pool.

IOAM Encapsulation for protocols which leverage Ethertype

draft-weis-ippm-ioam-eth-00

draft-weis-ippm-ioam-eth-00

- Evolution of draft-weis-ippm-ioam-gre
  - At IPPM WG meeting in Montreal (IETF102) it was suggested to make the draft generic to any protocol that uses Ethertype: Only one Ethertype codepoint required
- Approach
  - When the IOAM data fields are included within an encapsulation that identifies the next protocol using an EtherType (e.g., GRE or Geneve) the presence of IOAM data fields are identified with TBD_IOAM.
  - When the Ethernet Encapsulation for In-situ OAM Data is used, an additional IOAM header is also included. This header indicates the type of IOAM data that follows, and the next protocol that follows the 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   IOAM-Type   |  IOAM HDR len|          Next Protocol         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               |
!                                                               |
~                 IOAM Option and Data Space                    ~
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next Protocol definition reflects different parent protocols and their use of the Next Protocol field (e.g. for Geneve, things vary quite a bit depending on whether Geneve is used over UDP, IP, Ethernet, etc. See also draft-worley-nvo3-geneve-misc).

Next Protocol: 16 bits Next Protocol Type field contains the
    protocol type of the packet following IOAM protocol header. When
    the most significant octet is 0x00, the Protocol Type is taken to
    be an IP Protocol Number as defined in [IP-PROT]. Otherwise, the
    Protocol Type is defined to be an EtherType value from [ETYPES].
    An implementation receiving a packet containing a Protocol Type
    which is not listed in one of those registries SHOULD discard the
    packet.

Example:
IOAM data fields sequences with GRE header

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
   |C|       Reserved0         | Ver |   Protocol Type = <TBD_IOAM>  |  G
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  R
   |      Checksum (optional)      |       Reserved1 (Optional)    |  E
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
   |   IOAM Type   |  IOAM HDR len |         Next Protocol         |  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  I
   !                                                               |  O
   !                                                               |  A
   ~                 IOAM Option and Data Space                    ~  M
   |                                                               |  |
   |                                                               |  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
   |                                                               |
   |            Payload + Padding (L2/L3/ESP/...)                  |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example:
IOAM data fields sequences with Geneve header

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
|Ver|  Opt Len  |O|C|    Rsvd.  |  Protocol Type = <TBD_IOAM>   |  |G
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |E
|        Virtual Network Identifier (VNI)       |    Reserved   |  |N
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |E
|                    Variable Length Options                    |  |V
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+E
|   IOAM-Type   |   IOAM HDR len|         Next Protocol         |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  I
!                                                               |  O
!                                                               |  A
~                 IOAM Option and Data Space                    ~  M
|                                                               |  |
|                                                               |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
|                                                               |
|  Inner header + Payload + Padding (L2/L3/ESP/...)             |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IOAM Raw Data Export with IPFIX

draft-spiegel-ippm-ioam-rawexport-01

draft-spiegel-ippm-ioam-rawexport -00 to -01 updates

- ioamEncapsulationType
  - Updated references to encapsulations for different protocols
  - Added codepoint for *IPv6*
  - Renamed *GENEVE* codepoint to *GENEVE Option*
  - Added codepoint for *GENEVE Next Protocol*
- Added two new information elements that optimize use of IP packet section and Ethernet frame section
  - Templates can use existing *ipHeaderPacketSection* or new *ipHeaderPacketSectionWithPadding*
    - First example uses *ipHeaderPacketSection*
    - Other examples use *ipHeaderPacketSectionWithPadding*
  - Templates can use existing *dataLinkFrameSection* or new *ethernetFrameSection*
    - Example changed from *dataLinkFrameSection* to *ethernetFrameSection*

ipHeaderPacketSectionWithPadding Differences

sectionExportedOctets exists?	Fixed vs variable length	ipHeaderPacketSection: Padding allowed?	ipHeaderPacketSectionWithPadding: Padding allowed?
Yes	Fixed length	Yes	Yes
Yes	Variable length	?	Up to 3 octets of padding *
No	Fixed length	No	Yes, if the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding
No	Variable length	No	Up to 3 octets of padding *, if the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding

* used to preserve 4-octet alignment of subsequent IEs or subsequent records within the same set

ipHeaderPacketSectionWithPadding Text

- First 3 paragraphs (IP header contents, use of sectionOffset, payload and security considerations) are identical to *ipHeaderPacketSection*
- Use of padding with fixed or variable length IE is different:
  - When this Information Element has a fixed length, this MAY include padding octets that are used to fill out that fixed length.
  - When this information element has a variable length, the variable length MAY include up to 3 octets of padding, used to preserve 4-octet alignment of subsequent Information Elements or subsequent records within the same set.
  - In either case of fixed or variable length, the amount of populated octets MAY be specified in the sectionExportedOctets field corresponding to this Information Element, in which case the remainder (if any) MUST be padding. If there is no sectionExportedOctets field corresponding to this Information Element, then all octets MUST be populated unless the total length of the IP packet is less than the fixed length of this Information Element, in which case the remainder MUST be padding.

ethernetFrameSection

- Allows padding similar to *ipHeaderPacketSectionWithPadding*
  - Last 3 paragraphs are the same except for *IP packet* → *Ethernet frame*
- Eliminates need to include *dataLinkFrameType* and *dataLinkFrameSize*
- First paragraph text:

  This Information Element carries a series of n octets from the IEEE 802.3 Ethernet frame of a sampled packet, starting after the preamble and start frame delimiter (SFD), plus sectionOffset octets into the frame if there is a sectionOffset field corresponding to this Information Element.
- Second paragraph (payload and security considerations) is the same as *dataLinkFrameSection* and *ipHeaderPacketSection*

Next Steps

Next Steps

- IOAM data fields definition
  - draft-ietf-ippm-ioam-data – Draft requires several editorial fixes. With those, we should be ready for considering WGLC.
- IOAM data export
  - draft-spiegel-ippm-ioam-rawexport-01: Adopt as WG document?
- IOAM data fields encapsulation for different protocols:
  - VXLAN GPE: draft-brockners-ippm-ioam-vxlan-gpe
  - GENEVE: draft-brockners-ippm-ioam-geneve
  - Generic encap for protocols with Ethertype (GRE, etc): draft-weis-ippm-ioam-eth
  - IPv6: draft-ioametal-ippm-6man-ioam-ipv6-options
  - SRv6: draft-ali-spring-ioam-srv6
  - SR-MPLS: draft-gandhi-spring-ioam-sr-mpls

  Joint review of drafts between IPPM and working groups which “own” the parent protocol. Please review.

Thank you