In-situ OAM (IOAM)

Frank Brockners, Shwetha Bhandari,
Sashank Dara, Carlos Pignataro (Cisco)
Hannes Gedler (rtbrick)
Steve Youell (JPMC)
John Leddy (Comcast)
David Mozes (Mellanox)
Tal Mizrahi (Marvell)
Petr Lapukhov (Facebook)
Remy Chang (Barefoot)

IETF 98 – IPPM; March 27th, 2017

draft-brockners-inband-oam-data-03.txt

See also:
draft-brockners-inband-oam-requirements-03.txt
draft-brockners-inband-oam-transport-03.txt
draft-brockners-proof-of-transit-03.txt

Motivation and Requirements - IOAM

Use-Cases

- Service/Quality Assurance
  - Prove traffic SLAs, as opposed to probe-traffic SLAs; Overlay/Underlay
  - Service/Path Verification (Proof of Transit) – prove that traffic follows a pre-defined path
- Micro-Service/NFV deployments
  - Smart service selection based on network criteria
- Operations Support
  - Network Fault Detection and Fault Isolation through efficient network probing
  - Path Tracing – debug ECMP, brown-outs, network delays
  - Custom/Service Level Telemetry

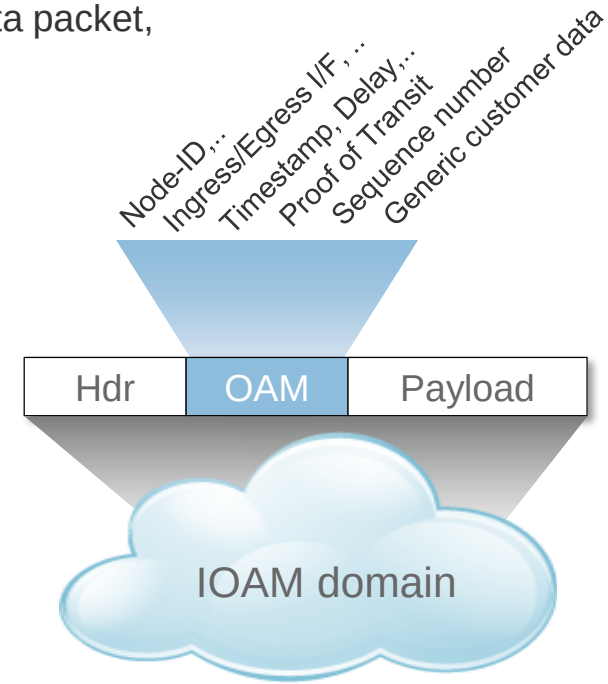
High Level Requirements

- Add OAM data-fields to live user traffic
  - *Data-fields*: Path-tracing and path verification information (node-ids, ingress/egress interfaces),timestamps, transit-delay, transit jitter, sequence numbers, application-defined metadata; dedicated namespaces for data-fields
  - *Domain-specific operation*: Classifier to select the set of traffic that IOAM is applied to
  - *Transport independence*: Data-fields definition independent from underlying transport protocol
- Consider operational aspects
  - Security/Vulnerability due to hop-by-hop information added to live user-traffic
  - Data-fields suitable for both, SW and HW implementations
  - ECMP processing, path MTU, ICMP message handling
  - Management, control, and export of IOAM information
  - Layering of IOAM information / Nesting of IOAM domains

More information: [draft-brockners-inband-oam-requirements-03.txt]

In-situ OAM in a nutshell

- Gather telemetry and OAM information along the path within the data packet, (hence “in-situ OAM”) as part of an existing/additional header
  - No extra probe-traffic (as with ping, trace, ipsla)
  - “Hybrid, Type-1 OAM” per RFC 7799
- Generic, Transport independent data-fields for IOAM
  - Scope: Per-hop, specific-hops only, end-to-end
  - Data fields include: Node IDs, interface IDs, timestamps, sequence numbers, ...
- Deployment
  - IOAM data fields can be embedded into a variety of transports, incl. IPv6, SRv6, NSH, GRE, ...
  - Domain focused: Domain-ingress, domain-egress, and select devices within a domain insert/remove/update the IOAM data fields
  - Information export via IPFIX/publish into Kafka/etc.
  - Fast-path implementation (reference implementation as open source)



IOAM Status

IOAM History

- IOAM introduced at IETF 96 and evolved in IETF 97 – informational sessions and/or mailing list discussions in OPSAWG, SFC, NVO3, RTG, SPRING, IPPM.
- OPSAWG originally voiced interest in taking on IOAM work, but after discussions IESG recommended for IPPM as the appropriate place.
- Bits-n-Bites demo at IETF 97

IOAM related drafts

- draft-brockners-inband-oam-data-03.txt
- draft-brockners-inband-oam-requirements-03.txt
- draft-brockners-inband-oam-transport-03.txt
- draft-brockners-proof-of-transit-03.txt

Open source reference implementation

- FD.io/VPP (see fd.io) – initial support in 16.09, enhanced on 17.01
- OpenDaylight Control App (for proof of transit / tracing) – Carbon release

Additional information: https://github.com/CiscoDevNet/iOAM

Incorporated IETF discussion feedback: Key updates since -00 version

- Name: “In situ OAM”
- Proper classification of IOAM per RFC 7799
- Data fields alignment and content merged with I-D.lapukhov-dataplane-probe, included loopback option
- Short/long format of several data records (e.g. node-id, interface-id)
- Timestamps: Wall-clock (in ns and sec), transit-delay
- Queue length: Capture egress queue depth when packet is being processed
- Two options for data record allocation for trace data: Pre-allocated and incremental
- All data 4-byte boundary aligned
- Cleaned-up nomenclature (data fields, data types, …)
- Data-fields definition independent from container to carry data fields (container assumed to be transport specific)

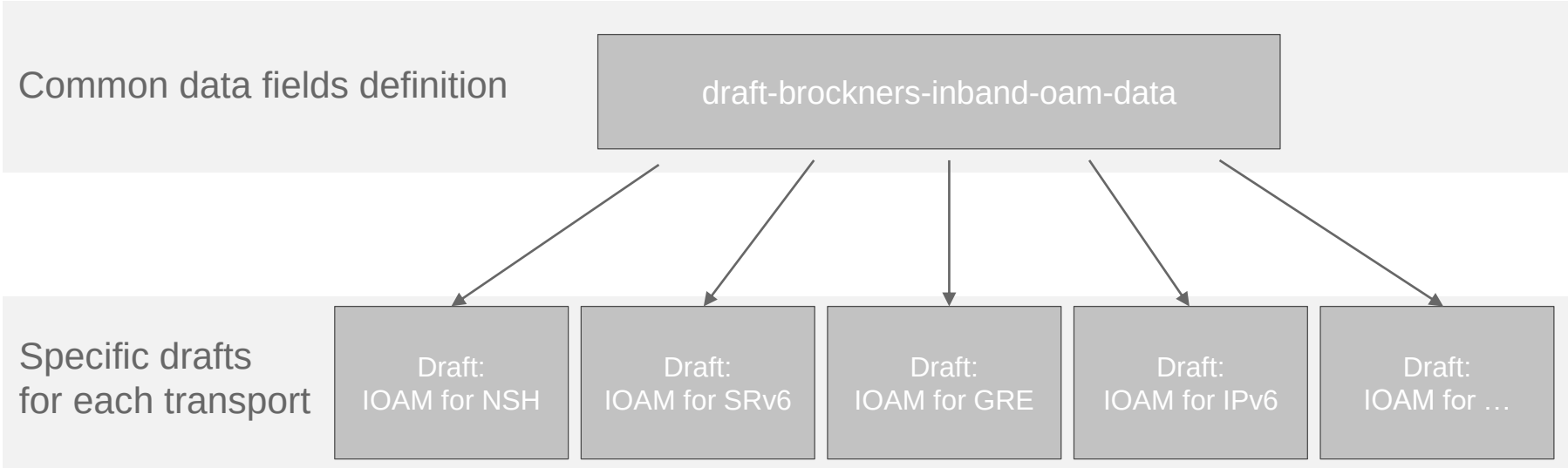
See also: https://github.com/inband-oam/ietf/issues?q=is%3Aissue+is%3Aclosed

IOAM Data Draft:

draft-brockners-inband-oam-data-03.txt

IOAM Data Draft
(draft-brockners-inband-oam-data-03)

IOAM Data Draft for Common Data Fields Definition: Will be complemented by transport specific drafts to carry the data fields



Draft for suggestions for a variety of IOAM transports: [draft-brockners-inband-oam-transport-03.txt]

In-situ OAM Data Fields Overview

- Per node scope
  - Hop-by-Hop information processing
    - Hop Limit
    - Node_ID (long/short)
    - Ingress Interface ID (long/short)
    - Egress Interface ID (long/short)
    - Timestamp
      - Wall clock (seconds, nanoseconds)
      - Transit delay
    - Queue length
    - Opaque data
    - Application Data (long/short)

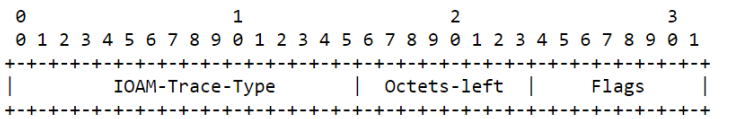
Two transport options:
- Pre-allocated array (SW friendly)
- Incrementally grown array (HW friendly)

- Set of nodes scope
  - Hop-by-Hop information processing
    - Service Chain Validation (Random, Cumulative)

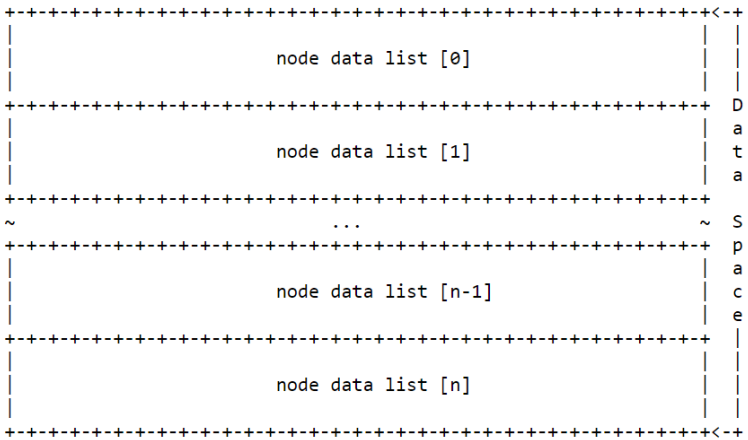
- Edge to Edge scope
  - Edge-to-Edge information processing
    - Sequence Number

Pre-Allocated & Incremental Trace Option Header (per-node info)

Pre-allocated Trace Option header:

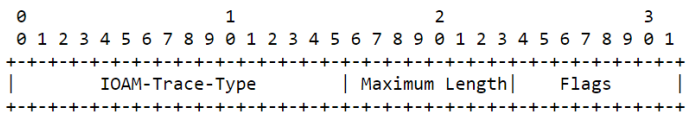


Pre-allocated Trace Option Data MUST be 4-byte aligned:

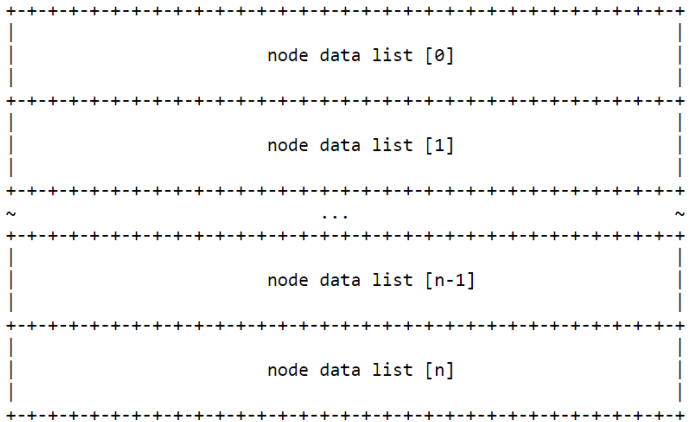


Software friendly

In-situ OAM Incremental Trace Option Header:



In-situ OAM Incremental Trace Option Data MUST be 4-byte aligned:



Hardware friendly

Trace Types

Bit 0 When set indicates presence of Hop_Lim and node_id in the node data.

Bit 1 When set indicates presence of ingress_if_id and egress_if_id in the node data.

Bit 2 When set indicates presence of timestamp seconds in the node data.

Bit 3 When set indicates presence of timestamp nanoseconds in the node data.

Bit 4 When set indicates presence of transit delay in the node data.

Bit 5 When set indicates presence of app_data in the node data.

Bit 6 When set indicates presence of queue depth in the node data.

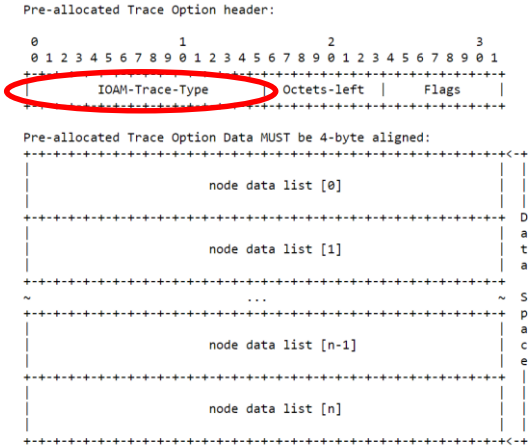
Bit 7 When set indicates presence of variable length Opaque State Snapshot field.

Bit 8 When set indicates presence of Hop_Lim and node_id wide in the node data.

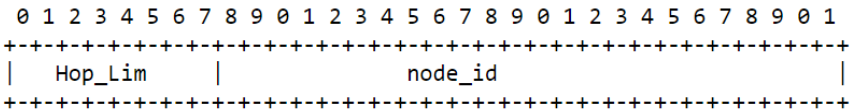
Bit 9 When set indicates presence of ingress_if_id and egress_if_id wide in the node data.

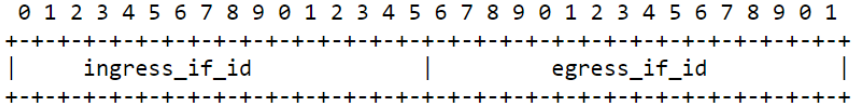
Bit 10 When set indicates presence of app_data wide in the node data.

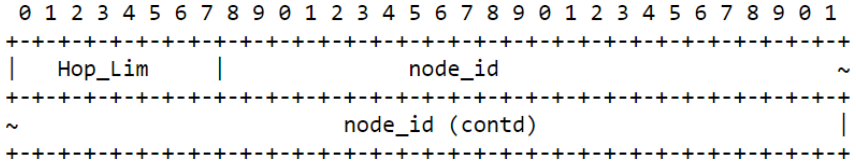
Bit 11-15 Undefined in this draft.

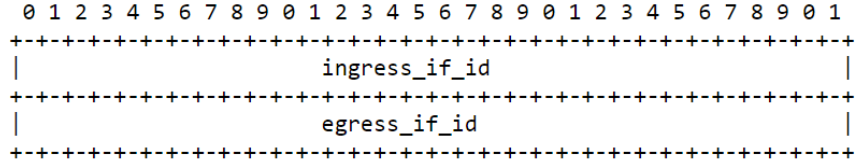


Data Fields:
Node-ID and Interface-IDs

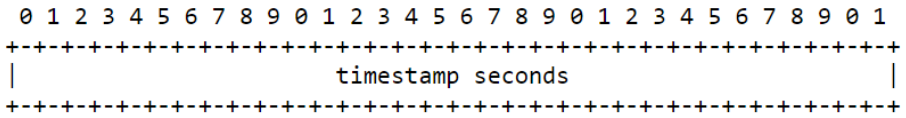


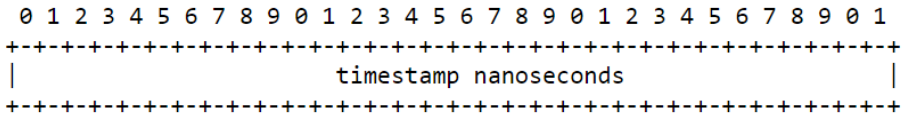


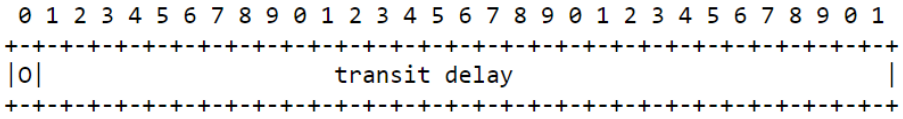




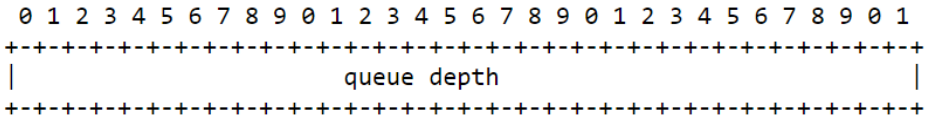
Data Fields: Timestamps, Delay

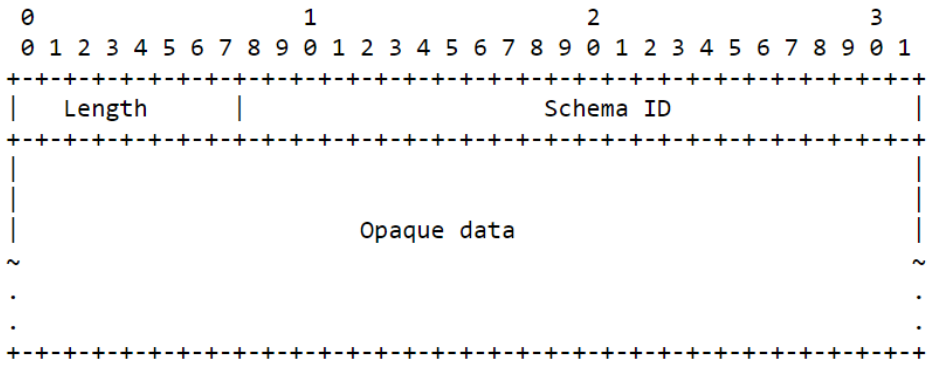






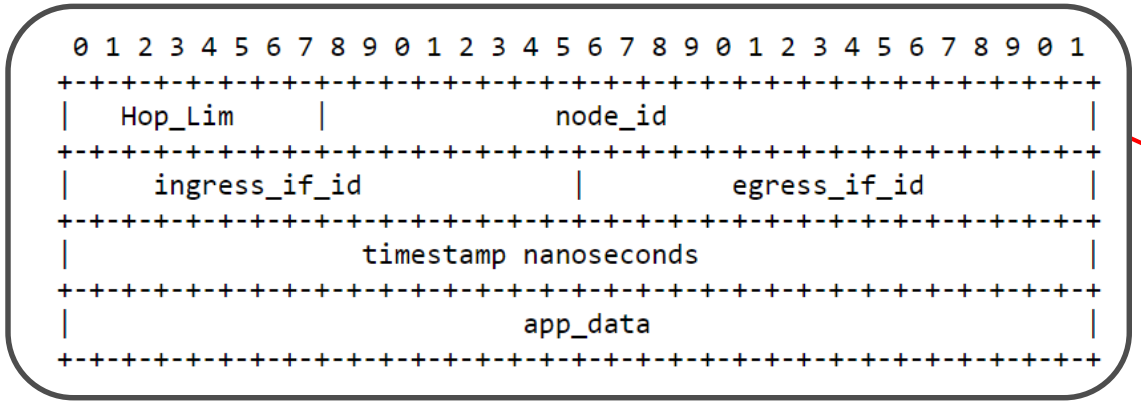
Data Fields:
Queue Depth, Opaque Data

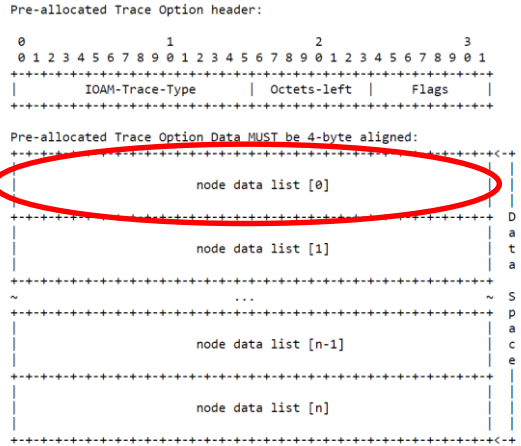




IOAM Trace Data Fields Example:

0x002B: IOAM-Trace-Type is 0x2B then the format of node data is:

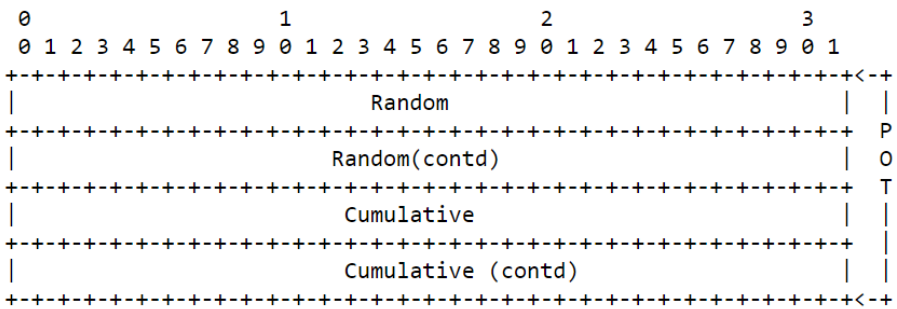




IOAM Data which in only updated by selected nodes: Proof-of-Transit

In-situ OAM Proof of Transit Option Header:

In-situ OAM Proof of Transit Option Data MUST be 4-byte aligned:

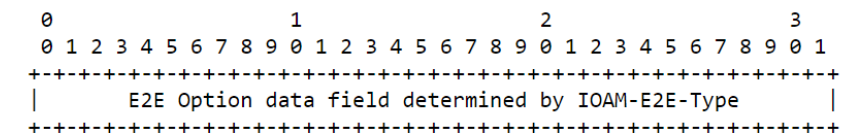


IOAM Data with E2E focus: Sequence Numbers

In-situ OAM Edge-to-Edge Option:

In-situ OAM Edge-to-Edge Option Header:

In-situ OAM Edge-to-Edge Option Data MUST be 4-byte aligned:



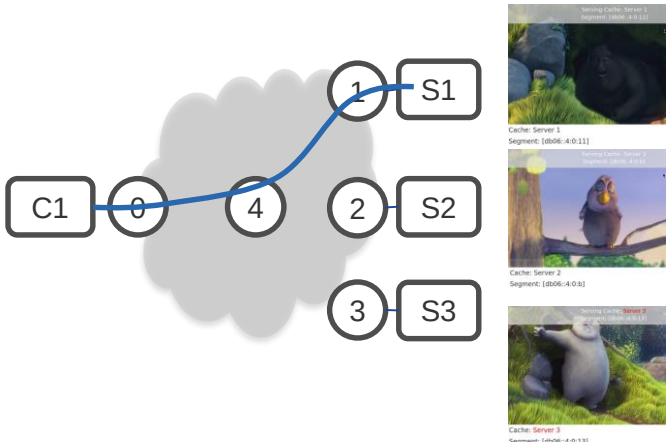
IOAM-E2E-Type: 8-bit identifier of a particular in situ OAM E2E variant.

0: E2E option data is a 64-bit sequence number added to a specific tube which is used to identify packet loss and reordering for that tube.

In-situ OAM demos at Bits-n-bites

M-anycast

Smart service selection – combing SRv6 and in-situ OAM for micro-service based video delivery (6CN)



Measure transit delays, server loads, choose optimal service for client and steer connection using SRv6

https://youtu.be/-jqww8ydWQk

In-situ OAM based active network probing

UDP probe configured among all edge nodes.
Server collects summarized probe info from all edge nodes

Discussion and next steps

- IOAM has been discussed in various IETF WGs (e.g. OPSAWG, RTG, SPRING, NVO3). After discussions, it was suggested to have IPPM take on the protocol independent part of the IOAM work and faciliate integration of IOAM into different transport protocols.
- IOAM has been discussed on the IPPM WG mailing list:
  -03 version of the drafts already include feedback received from IPPM WG.
- Proposal: Adopt draft-brockners-inband-oam-data-03.txt as a WG draft in IPPM WG

Thanks!

References

[1] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Data Formats for In-situ OAM", draft-brockners-inband-oam-data-03 (work in progress), March 2017, https://tools.ietf.org/html/draft-brockners-inband-oam-data.

[2] Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., and R. Chang, "Encapsulations for In-situ OAM Data", draft-brockners-inband-oam-transport-03 (work in progress), March 2017, https://tools.ietf.org/html/draft-brockners-inband-oam-transport.

[3] Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mozes, D., Mizrahi, T., Lapukhov, P., and R. Chang, "Requirements for In-situ OAM", draft-brockners-inband-oam-requirements-03 (work in progress), March 2017, https://tools.ietf.org/html/draft-brockners-inband-oam-requirements.

[4] Brockners, F., Bhandari, S., Dara, S., Pignataro, C., Leddy, J., Youell, S., Mozes, D., and T. Mizrahi, "Proof of Transit", draft-brockners-proof-of-transit-03 (work in progress), March 2017, https://tools.ietf.org/html/draft-brockners-proof-of-transit-03.

[5] Morton, A., "Active and Passive Metrics and Methods (with Hybrid Types In-Between)", RFC 7799, DOI 10.17487/RFC7799, May 2016, https://www.rfc-editor.org/info/rfc7799.

[6] https://github.com/CiscoDevNet/iOAM.