In-Band OAM

Frank Brockners, Shwetha Bhandari,
Sashank Dara, Carlos Pignataro (Cisco)
Hannes Gedler (rtbrick)
Steve Youell (JMPC)
John Leddy (Comcast)

IETF 96 – OPSWG; July 19th, 2016

draft-brockners-proof-of-transit-01.txt
draft-brockners-inband-oam-requirements-01.txt
draft-brockners-inband-oam-data-01.txt
draft-brockners-inband-oam-transport-01.txt

How to send OAM information in packet networks?

"On-Board Unit"

In-band OAM

- OAM traffic embedded in the data traffic but not part of the payload of the packet
- OAM "effected by data traffic"
- Example: IPv4 route recording

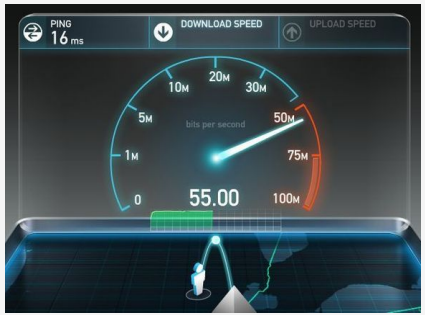
Speed control by police car

Out-of-band OAM

- OAM traffic is sent as dedicated traffic, independent from the data traffic ("probe traffic")
- OAM "not effected by data traffic"
- Examples: Ethernet CFM (802.1ag), Ping, Traceroute

In-Band/Passive OAM - Motivation

- Multipath Forwarding –
  debug ECMP networks
- Service/Path Verification –
  prove that traffic follows a pre-defined path
- Service/Quality Assurance –
  Prove traffic SLAs, as opposed to probe-traffic SLAs; Overlay/Underlay
- Derive Traffic Matrix
- Custom/Service Level Telemetry



"Most large ISP's prioritize Speedtest traffic and I would even go as far to say they probably route it faster as well to keep ping times low."

Source: https://www.reddit.com/r/AskTechnology/comments/2i1nxc/can_i_trust_my_speedtestnet_results_when_my_isp/

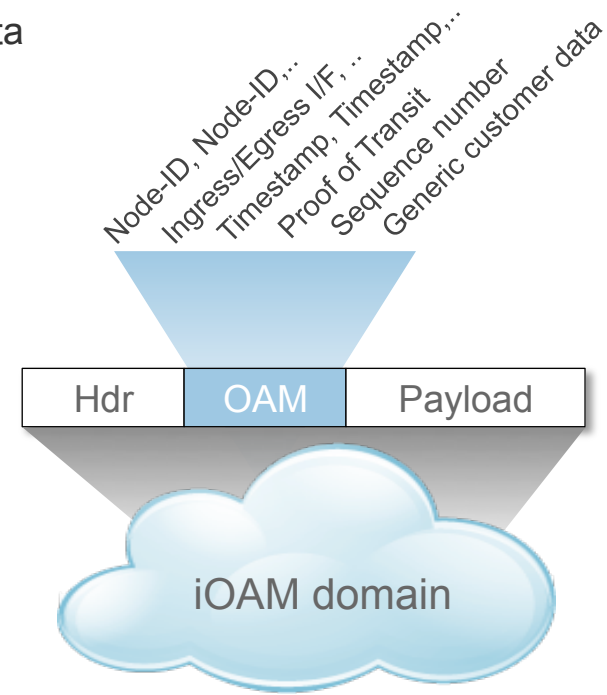
What if you could collect operational meta-data within your traffic?

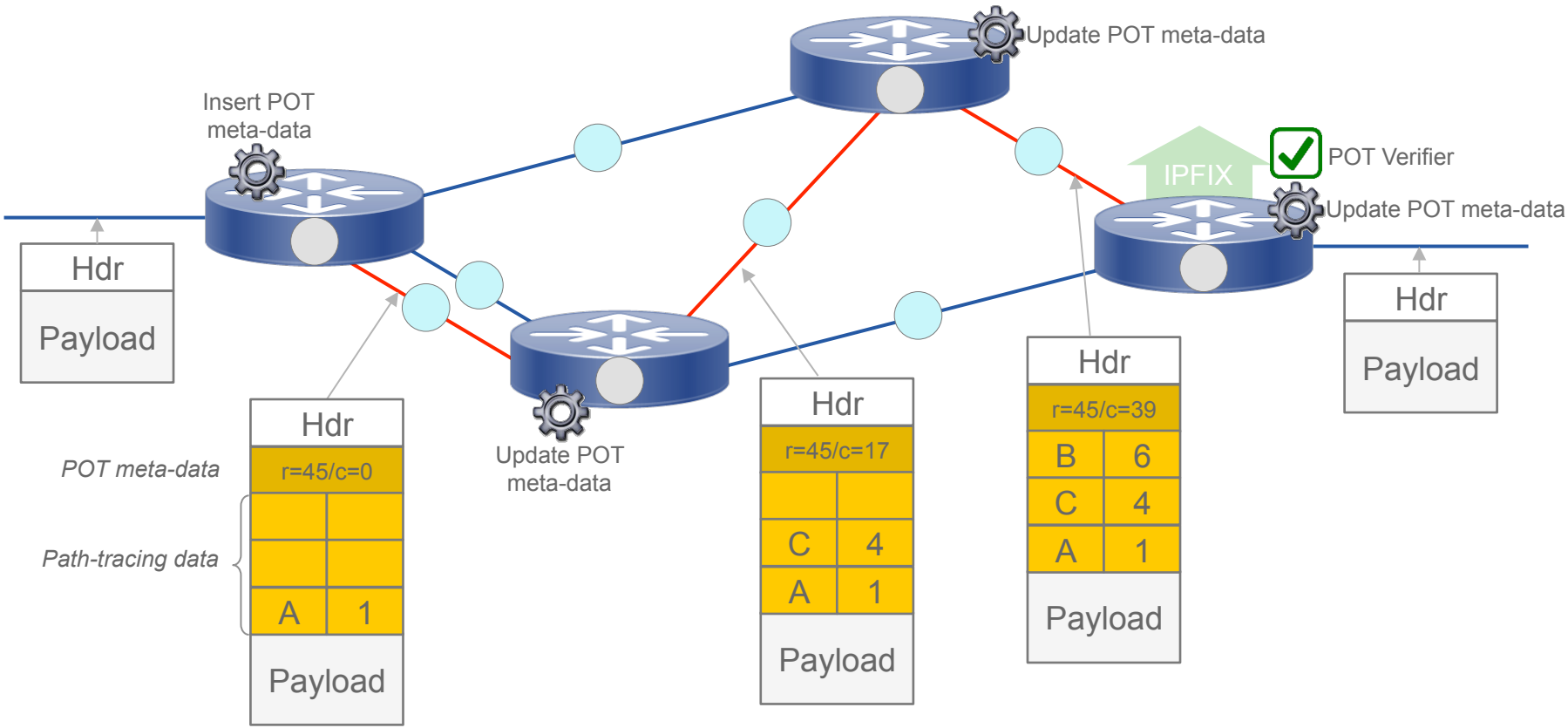
Example use-cases...	Meta-data required...
• Path Tracing for ECMP networks	• Node-ID, ingress i/f, egress i/f
• Service/Path Verification	• Proof of Transit (random, cumulative)
• Derive Traffic Matrix	• Node-ID
• SLA proof: Delay, Jitter, Loss	• Sequence numbers, Timestamps
• Custom data: Geo-Location,..	• Custom meta-data

In-Band OAM

- Gather telemetry and OAM information along the path within the data packet, as part of an existing/additional header
  - No extra probe-traffic (as with ping, trace, ipsla)
- Transport options
  - IPv6: Native v6 HbyH extension header or double-encap
  - VXLAN-GPE: Embedded telemetry protocol header
  - SRv6: Policy-Element (proof-of-transit only)
  - NSH: Type-2 Meta-Data (proof-of-transit only)

  ... additional encapsulations being considered (incl. IPv4, MPLS)
- Deployment
  - Domain-ingress, domain-egress, and select devices within a domaininsert/remove/update the extension header
  - Information export via IPFIX/Flexible-Netflow/publish into Kafka
  - Fast-path implementation

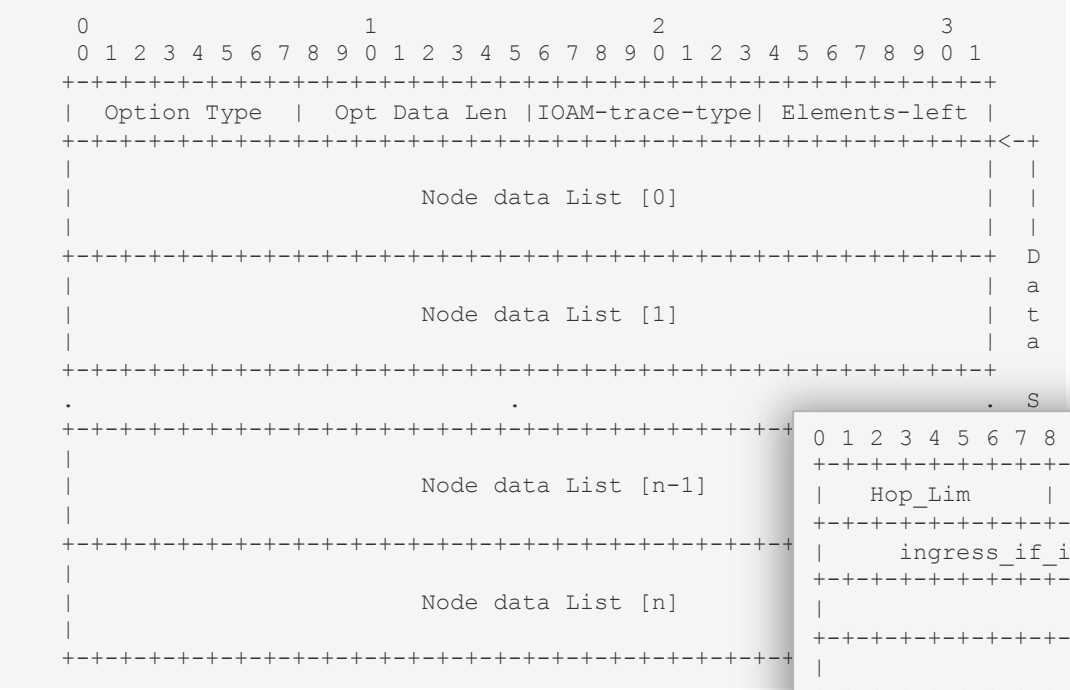


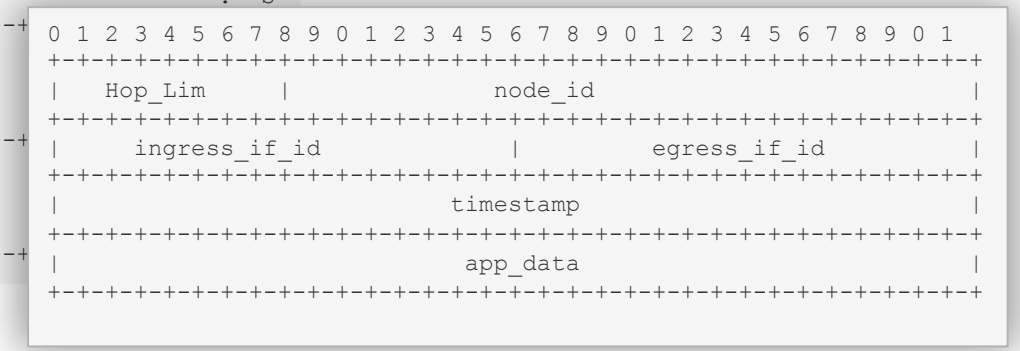


In-Band OAM: Information carried

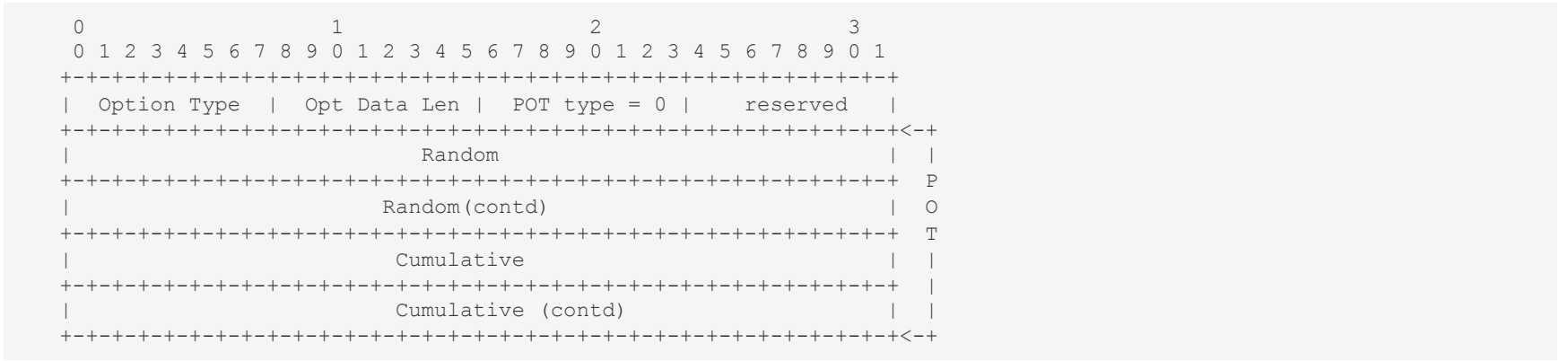
- Per node scope
  - Hop-by-Hop information processing
    - Device_Hop_L
    - Node_ID
    - Ingress Interface ID
    - Egress Interface ID
    - Time-Stamp
    - Application Meta Data
- Set of nodes scope
  - Hop-by-Hop information processing
    - Service Chain Validation (Random, Cumulative)
- Edge to Edge scope
  - Edge-to-Edge information processing
    - Sequence Number

Tracing Option

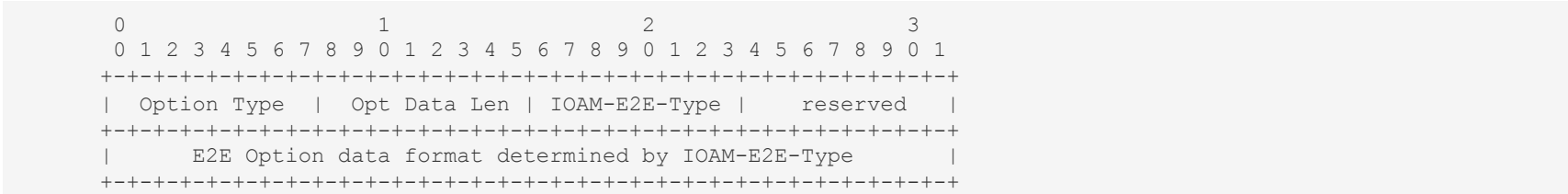




Proof-of-Transit Option



Edge-to-Edge Option



Option Type: 000xxxxxx 8-bit identifier of the type of option.

Opt Data Len: 8-bit unsigned integer. Length of the Option Data field of this option, in octets.

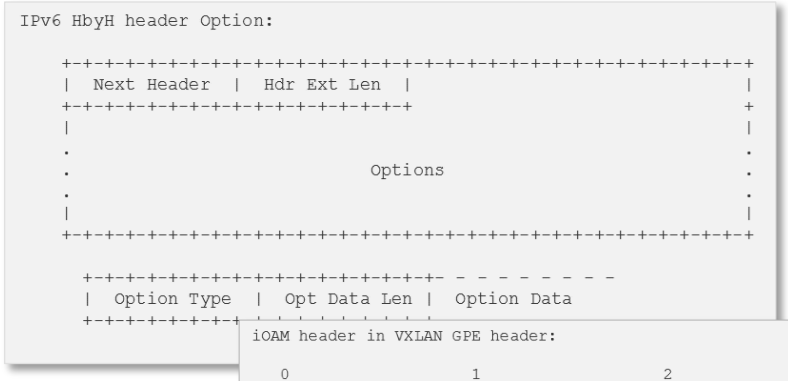
iOAM-E2E-Type: 8-bit identifier of a particular iOAM E2E variant.

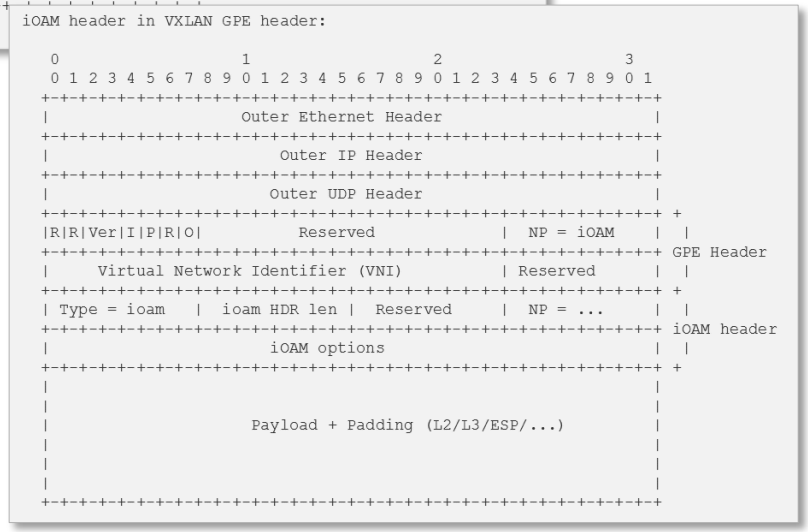
0: E2E option data is 64-bit Per Packet Counter (PPC) used to identify packet loss and reordering.

Reserved: 8-bit. (Reserved Octet) Reserved octet for future use.g

Transport Options – IPv6, VXLAN-GPE, SRv6, NSH...

- IPv6: v6 HbyH extension header
- VXLAN-GPE: Embedded telemetry protocol header; Combines with NSH etc.
- SRv6: In-Band OAM TLV in v6 SR-header SRH (proof-of-transit only)
- NSH: Type-2 Meta-Data (proof-of-transit only)
- ... more to come: MPLS, IPv4,...





Proof of Transit

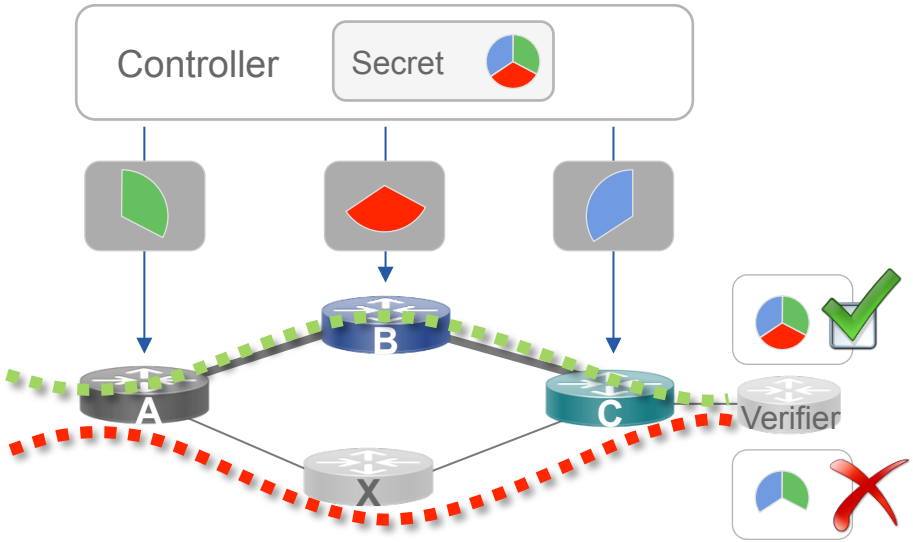
Consider traffic engineering, policy based routing, service chaining:

"How do you *prove* that traffic follows the suggested path?"

Ensuring Path and/or Service Chain Integrity Approach

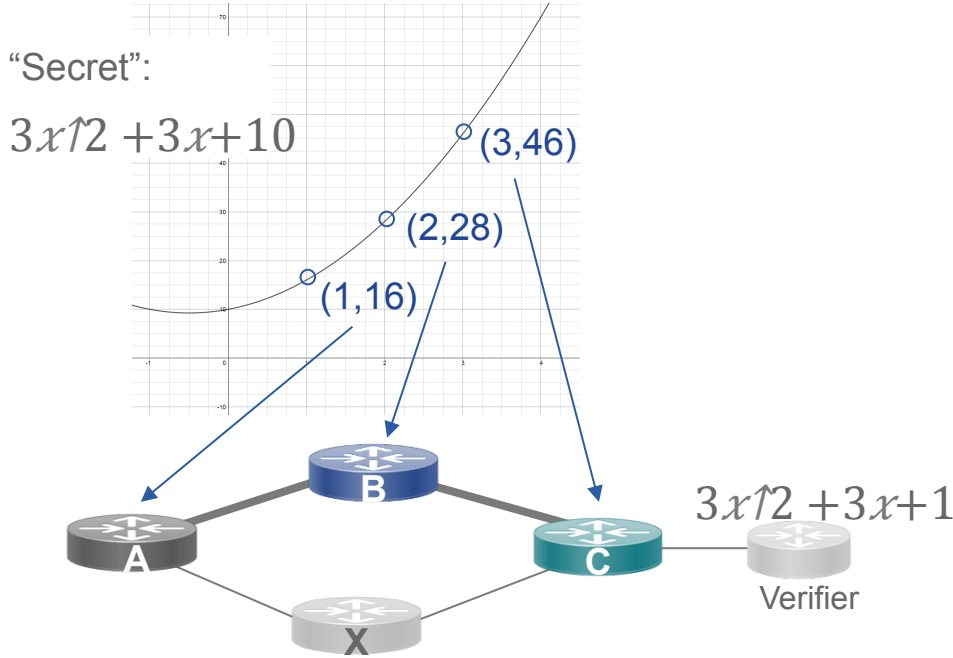
- Meta-data added to all user traffic
  - Based on “Share of a secret”
  - Provisioned by controller over secure channel to segment hops where “proof of transit” is required
  - Updated at every segment hop where proof of transit is required
- Verifier checks whether collected meta-data allows retrieval of secret

  ➡ “Proof of Transit”: Path verified



Solution Approach: Leverage Shamir's Secret Sharing
"A polynomial as secret"

- Each service is given a point on the curve
- When the packet travels through each service it collects these points
- A verifier can reconstruct the curve using the collected points
- Operations done over a finite field (mod prime) to protect against differential analysis



Running Code: Experimental OpenSource Implementation

- Open source experimental Implementation: FD.io/VPP (see fd.io)
- Demo Videos:
  Google+ In-Band OAM group:
  https://plus.google.com/u/0/b/112958873072003542518/112958873072003542518/videos?hl=en
  Youtube In-Band OAM channel: https://www.youtube.com/channel/UC0WJOAKBTrftyosP590RrXw

Next Steps

- The authors appreciate thoughts, feedback, and text on the content of the documents from the WG
- The authors also value feedback on where to progress the work?