In-situ OAM – Update

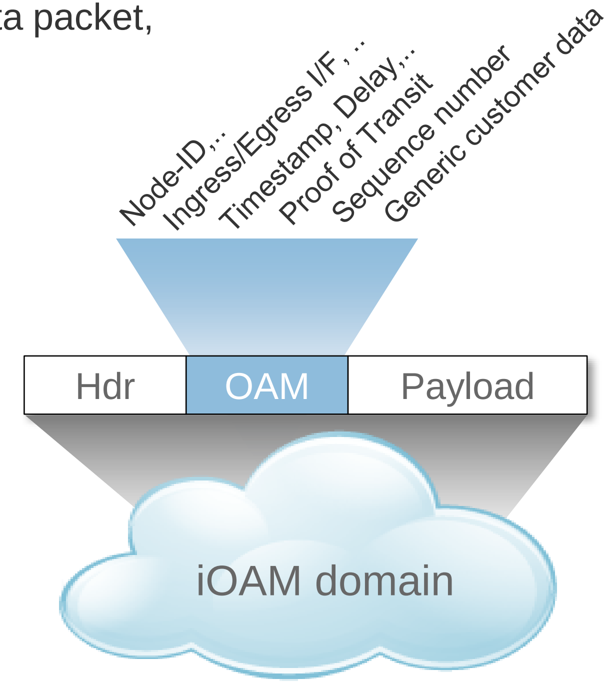
Frank Brockners, Shwetha Bhandari,
Sashank Dara, Carlos Pignataro (Cisco)
Hannes Gedler (rtbrick)
Steve Youell (JMPC)
John Leddy (Comcast)
David Mozes (Mellanox)
Tal Mizrahi (Marvell)
Petr Lapukhov (Facebook)
Remy Chang (Barefoot)

IETF 97 – OPSAWG; Nov 14th, 2016

draft-brockners-proof-of-transit-02.txt
draft-brockners-inband-oam-requirements-02.txt
draft-brockners-inband-oam-data-02.txt
draft-brockners-inband-oam-transport-02.txt

In-situ OAM – A brief recap

- 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)
- Transport options
  - IPv6: Native v6 HbyH extension header or double-encap
  - VXLAN-GPE: Embedded telemetry protocol header
  - SRv6: Meta-data in TLV format in SRH
  - NSH: Type-2 Meta-Data
    ... additional encapsulations being considered/WIP (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



Updates included in -02 version of the drafts

- General
  - Name change: “In situ OAM” (thanks to Erik Nordmark for proposing a the new name)
  - Proper classification per RFC 7799
  - Data-format alignment and content merged with I-D.lapukhov-dataplane-probe
  - Evolved requirements, fixes to in -00 versions of the drafts (thanks to Jen Linkova, Hemant Singh, Ignas Bagdonas)
- Proof of transit
  - Nested hashing as additional approach to POT (complementing Shamir’s Secret Sharing)
  - Evolved discussion of threat models and protection
    - RND as a hash across the payload to couple POT metadata and packet payload
- Data records
  - Short/long format of several data records (incl. node-id, app meta-data, etc.)
  - 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
  - Added constraint: All data 4 byte boundary aligned

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

In-situ OAM: Data Records

- Per node scope
  - Hop-by-Hop information processing
    - Device_Hop_L
    - Node_ID (long/short)
    - Ingress Interface ID (long/short)
    - Egress Interface ID (long/short)
    - Time-Stamp
      - Wall clock (ns/sec)
      - Transit delay
    - Queue length
    - Opaque data
    - Application Meta Data (long/short)

Two transport options*:

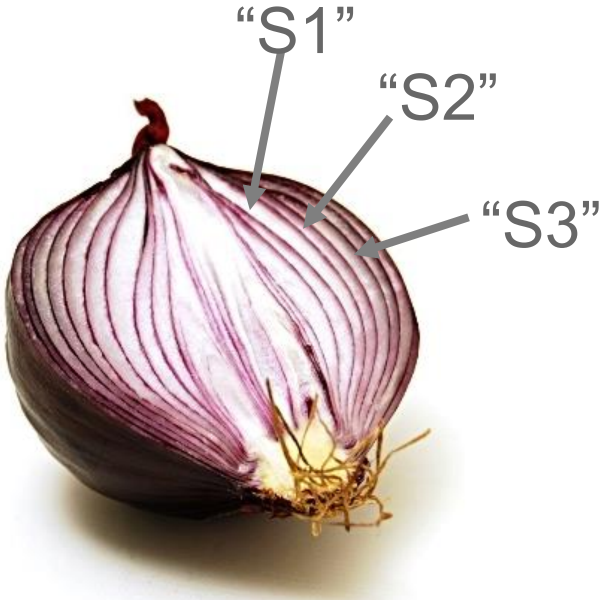
- Pre-allocated array (SW friendly)
- Incrementally grown array (HW friendly)

- 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

*See sections 3.1.2 and 3.1.3 in draft-brockners-inband-oam-data-02

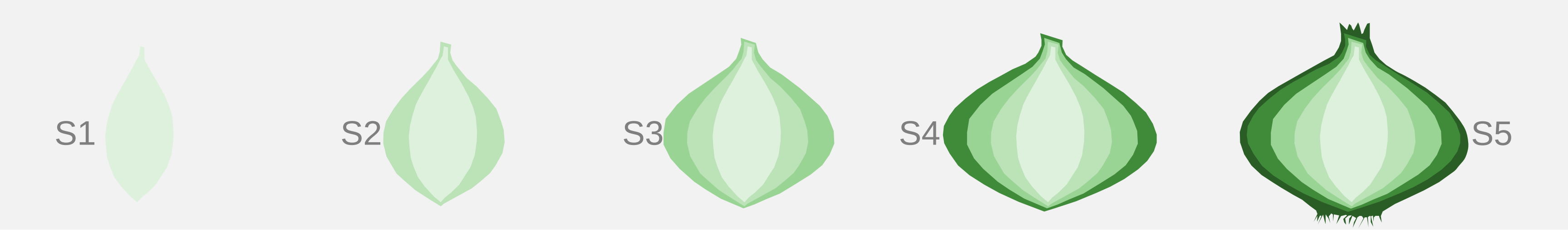
POT Solution Approach 2: Nested Crypto: “Compose an Onion”

- Approach
  - A service is described by a set of secrets, where each secret is associated with a service function. Service functions encrypt portions of the meta-data as part of their packet processing.
  - Only the verifying node has access to all secrets. The verifying nodes re-encrypts the meta-data to validate whether the packet correctly traversed the service chain.
- Notes
  - Nested encryption allows to check the order in which the nodes where traversed
  - To be used only when hardware assisted encryption is available. i.e. AES-NI instructions or equivalent. Otherwise this could be very costly operation to verify at line speed.



Service-Secrets are nested like layers of an onion

POT Solution Approach 2: “Compose the Onion”

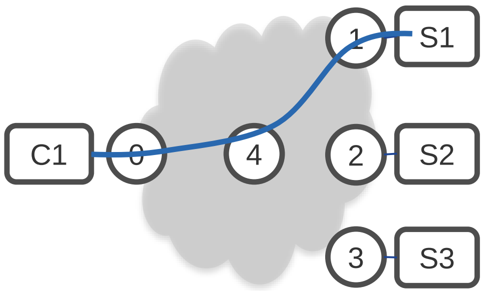


1. The controller provisions all the nodes with their respective secret keys.
2. The controller provisions the verifier with all the secret keys of the nodes.
3. For each packet, the ingress node generates a random number RND and encrypts it with its secret key to generate CML value
4. Each subsequent node on the path encrypts CML with their respective secret key and passes it along
5. The verifier is also provisioned with the expected sequence of nodes in order to verify the order
6. The verifier receives the CML, RND values, re-encrypts the RND with keys in the same order as expected sequence to verify.

In-situ OAM demos at Bits-n-bites

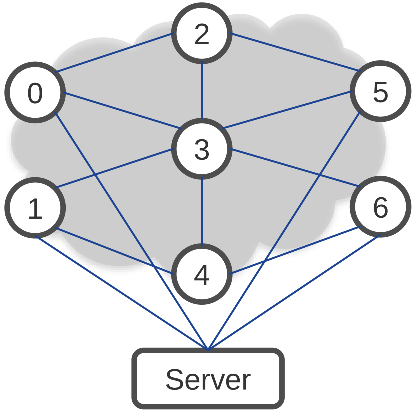
M-anycast

Smart service selection – combing SRv6 and in-situ OAM



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

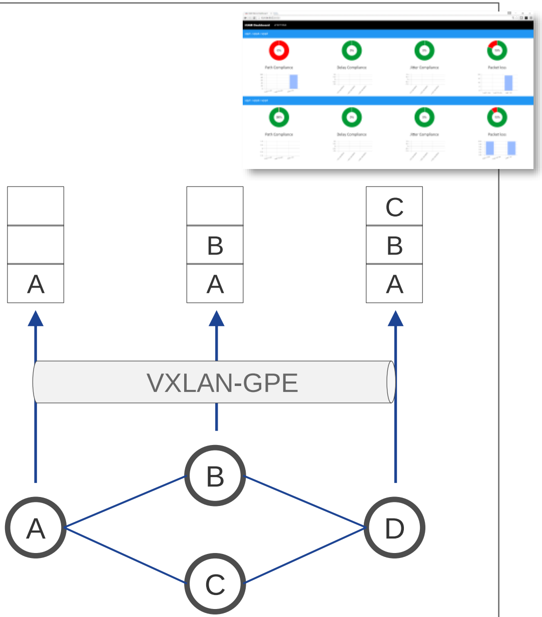
In-situ OAM based active network probing



UDP probe configured among all edge nodes (0,1,5,6). Server collects summarized probe info from all edge nodes

VXLAN-GPE

Overlay-Underlay Tracing and SLA Check



Next Steps

- The authors appreciate thoughts, feedback, and text on the content of the documents from the OPSAWG WG
- The authors believe OPSAWG is the right place to taking on the in-situ OAM / POT work.
  - In-situ OAM is an operational capability
  - In-situ OAM is applicable throughout various encapsulations and technologies, incl. IPv6.

*Consequently, is **OPSAWG** interested in taking on the in-situ OAM / POT work?*