Proof of Transit & In-Band OAM

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

IETF 96 – SFC WG; July 18th, 2016

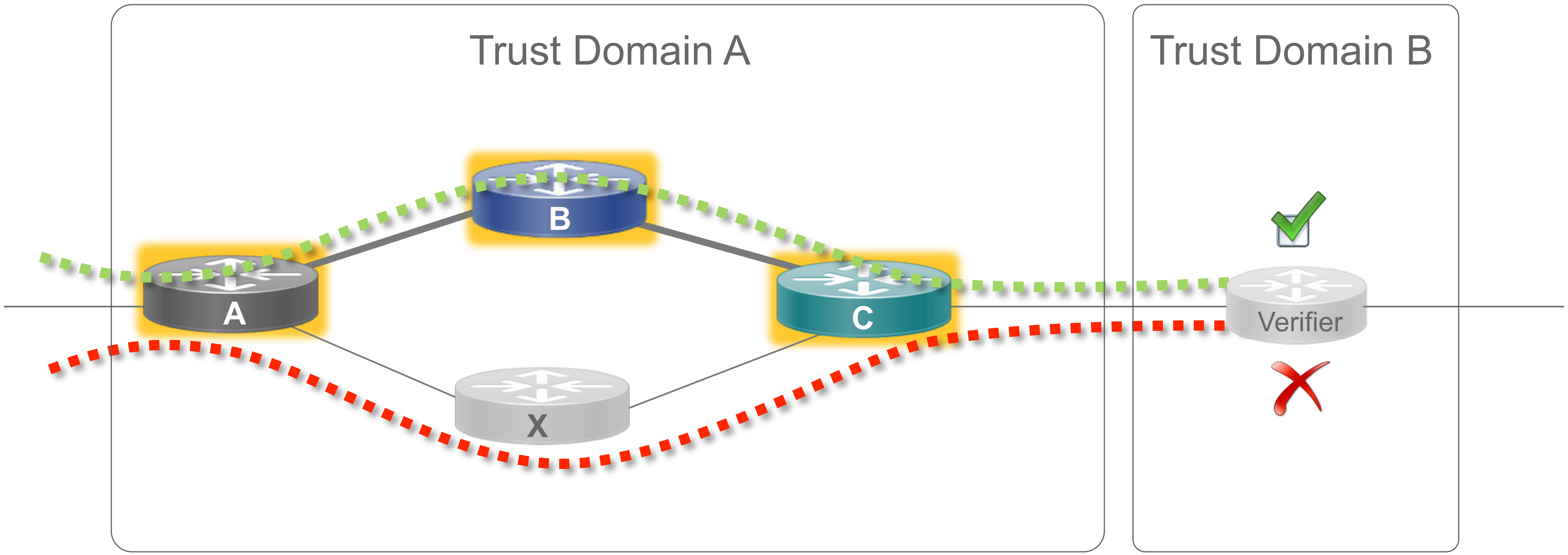
draft-brockners-proof-of-transit-01.txt
draft-brockners-inband-oam-requirements-01.txt
draft-brockners-inband-oam-data-01.txt
draft-brockners-inband-oam-transport-01.txt

© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

Consider a service chain:

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

© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

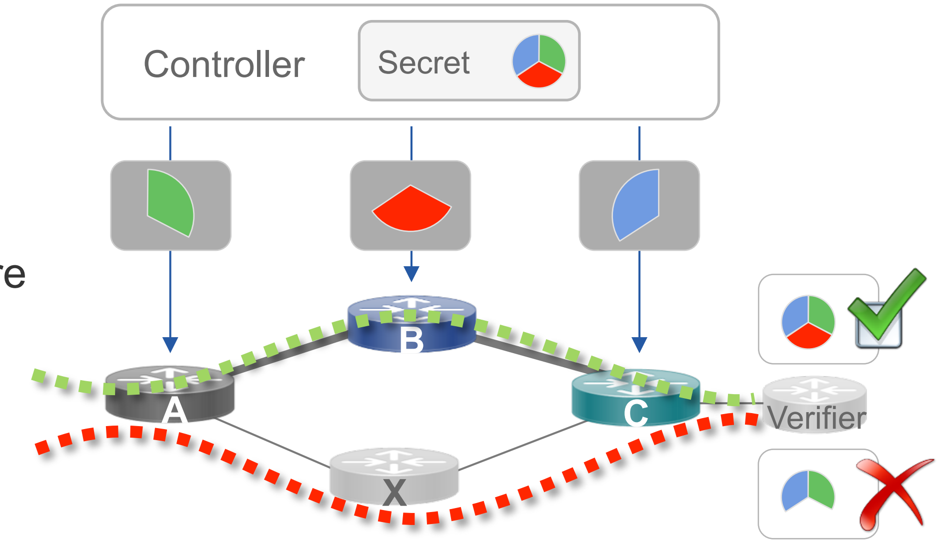


© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

Ensuring Path and/or Service Chain Integrity
Approach

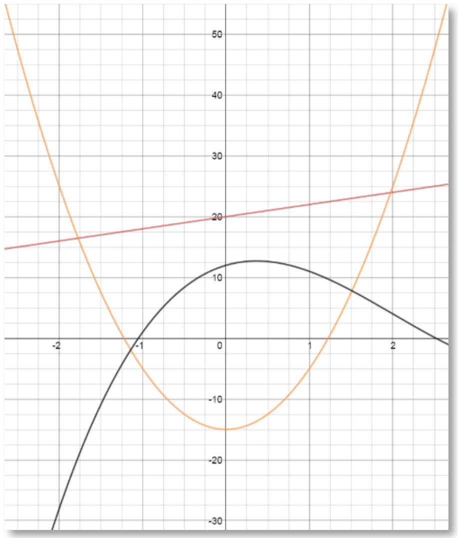
- Meta-data added to all user traffic
  - Based on “Share of a secret”
  - Provisioned by controller over secure channel to segment hops where “proof of transit” is required
  - Updated at every segment hop where proof of transit is required
- Verifier checks whether collected meta-data allows retrieval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

➡ “Proof of Transit”: Path verified



© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

Solution Approach: Leveraging Shamir's Secret Sharing
Polynomials 101



$f2(x) = 10x^2 - 15$ - Parabola: Min 3 points

$f1(x) = 2x + 20$ - Line: Min 2 points

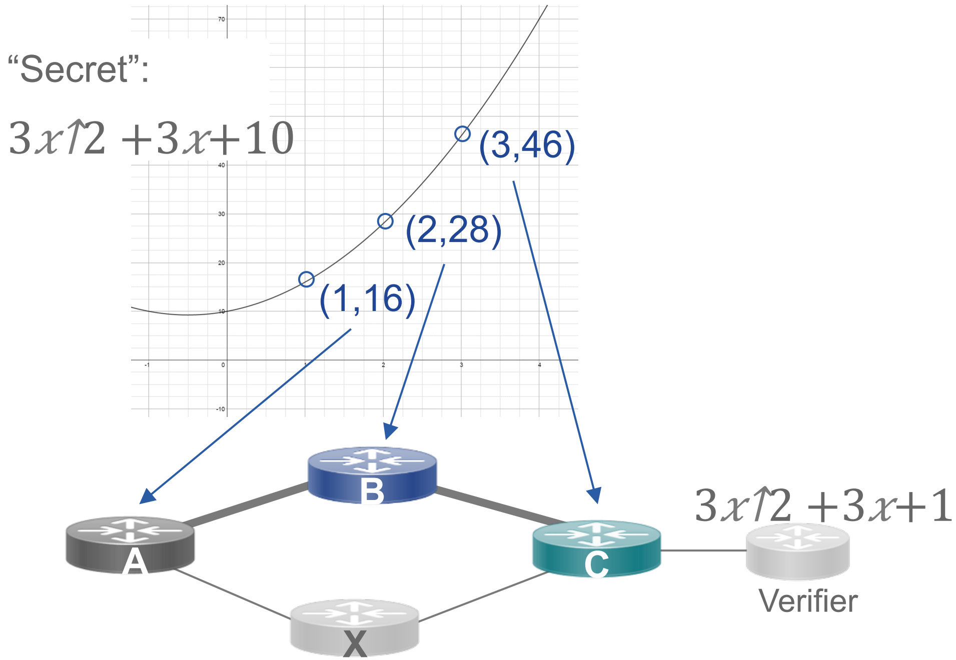
$f3(x) = x^3 - 6x^2 + 4x - 12$ - Cubic function: Min 4 points

General: It takes k+1 points to defines a polynomial of degree k.

© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

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



© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

Operationalizing the Solution

- Leverage two polynomials:
  - POLY-1 secret, constant: Each hop gets a point on POLY-1
    Only the verifier knows POLY-1
  - POLY-2 public, random and per packet.
    Each hop generates a point on POLY-2 each time a packet crosses it.
- Each service function calculates (Point on POLY-1 + Point on POLY-2) to get (Point on POLY-3) and passes it to verifier by adding it to each packet.
- The verifier constructs POLY-3 from the points given by all the services and cross checks whether POLY-3 = POLY-1 + POLY-2
- Computationally efficient: 3 additons, 1 multiplication, mod prime per hop

POLY-1
Secret – Constant

+

POLY-2
Public – Per Packet

=

POLY-3
Secret – Per Packet

© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

Meta Data for Service/Path Verification

- Verification secret is the independent coefficient of POLY-1
  - Computation/retrieval through a cumulative computation at every hop ("cumulative")
- For POLY-2 the independent coefficient is carried within the packet (typically a combination of timestamp and random number)
  - n bits can service a maximum of 2^n packets
- Verification secret and POLY-2 coefficient ("random") are of the same size
  - Secret size is bound by prime number

Transfer Rate	RND/ Secret Size	Max # of packets (assuming 64 byte packets)	Time that "random" lasts at maximum
1 Gbps	**64**	$2^{64} \approx 2 * 10^{19}$	approx. 10^{13} seconds, approx. 310000 years
10 Gbps	**64**	$2^{64} \approx 2 * 10^{19}$	approx. 10^{12} seconds, approx. 31000 years
100 Gbps	**64**	$2^{64} \approx 2 * 10^{19}$	approx. 10^{11} seconds, approx. 3100 years
10 Gbps	56	$2^{56} \approx 7 * 10^{16}$	approx. 10^{9} seconds, approx. 120 years
10 Gbps	48	$2^{48} \approx 2 * 10^{14}$	approx. 10^{7} seconds, approx. 5.5 months
10 Gbps	40	$2^{40} \approx 1 * 10^{12}$	approx. $5 * 10^{4}$ seconds, approx. 15 hours
1 Gbps	32	$2^{32} \approx 4 * 10^{9}$	2200 seconds, 36 minutes
10 Gbps	32	$2^{32} \approx 4 * 10^{9}$	220 seconds, 3.5 minutes
100 GBps	32	$2^{32} \approx 4 * 10^{9}$	22 seconds

© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

Proof of Transit: Meta-Data Transport Options

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Random                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Random (contd)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Cumulative                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Cumulative (contd)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2nd Polynomial

Interative computation of secret

- 16* Bytes of Meta-Data for SCV
  - Random – Unique random number (e.g. Timestamp or combination of Timestamp and Sequence number)
  - Cumulative (algorithm dependent)

- Transport options for different protocols
  - Segment Routing: New TLV in SRH header
  - Network Service Header: Type-2 Meta-Data
  - In-band OAM for IPv6:
    Proof-of-transit extension header
  - VXLAN-GPE
    Proof-of-transit embedded-telemetry header
  - ... more to be added (incl. IPv4)

*Note: Smaller numbers are feasible, but require a more frequent renewal of the polynomials/secrets.

NSH Type 2 Meta Data for POT

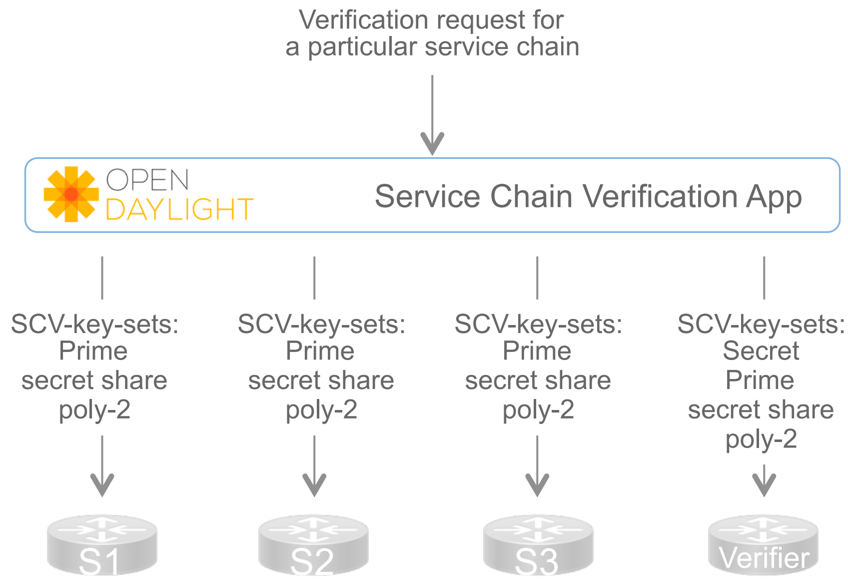
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TLV Class=Cisco (0x0009) |C|    Type=POT |F|R|R| Len=4   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
|                            Random                             |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  S
|                         Random(contd)                         |  C
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  V
|                          Cumulative                           |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                          Cumulative (contd)                   |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+<-+
```

TLV Class: Describes the scope of the "Type" field. In some cases, the TLV Class will identify a specific vendor, in others, the TLV Class will identify specific standards body allocated types. POT is currently using the Cisco (0x0009) TLV class.
Type: The specific type of information being carried, within the scope of a given TLV Class. Value allocation is the responsibility of the TLV Class owner. An experimental implementation currently uses a type value of 0x94 is used for proof of transit.
Reserved bits: Two reserved bit are present for future use. The reserved bits MUST be set to 0x0.
F: One bit. Indicates which POT-profile is active. 0 means the even POT-profile is active, 1 means the odd POT-profile is active.
Length: Length of the variable metadata, in 4-octet words. Here the length is 4.
Random: 64-bit Per packet Random number.
Cumulative: 64-bit Cumulative that is updated by the Service Functions.

© 2016 Cisco and/or its affiliates. All rights reserved.   IETF 96

Meta-Data Provisioning

- Meta-Data for POT provisioned through a controller (e.g. OpenDaylight App)
- Netconf/YANG based protocol
- Provisioned information from Controller to Service Function / Verifier
  - Service-Chain-Identifier
  - Service count (number of services in the chain)
  - 2 x POT-key-set
    - Secret (in case of communication to the verifier)
    - Share of a secret, service index
    - 2nd polynomial coefficients
    - Prime number



© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

Enter…

In-Band OAM

© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96

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

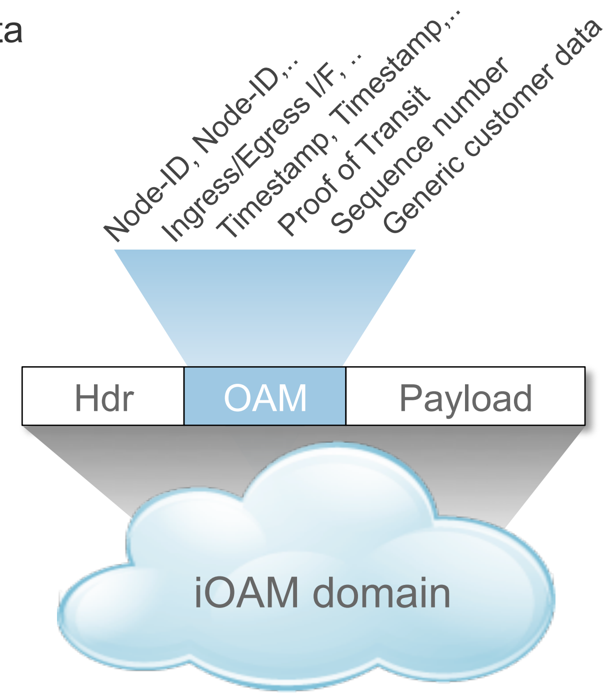
Example use-cases...	Meta-data required...
• Path Tracing for ECMP networks	• Node-ID, ingress i/f, egress i/f
• Service/Path Verification	• Proof of Transit (random, cumulative)
• Derive Traffic Matrix	• Node-ID
• SLA proof: Delay, Jitter, Loss	• Sequence numbers, Timestamps
• Custom data: Geo-Location,..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

© 2016 Cisco and/or its affiliates. All rights reserved.

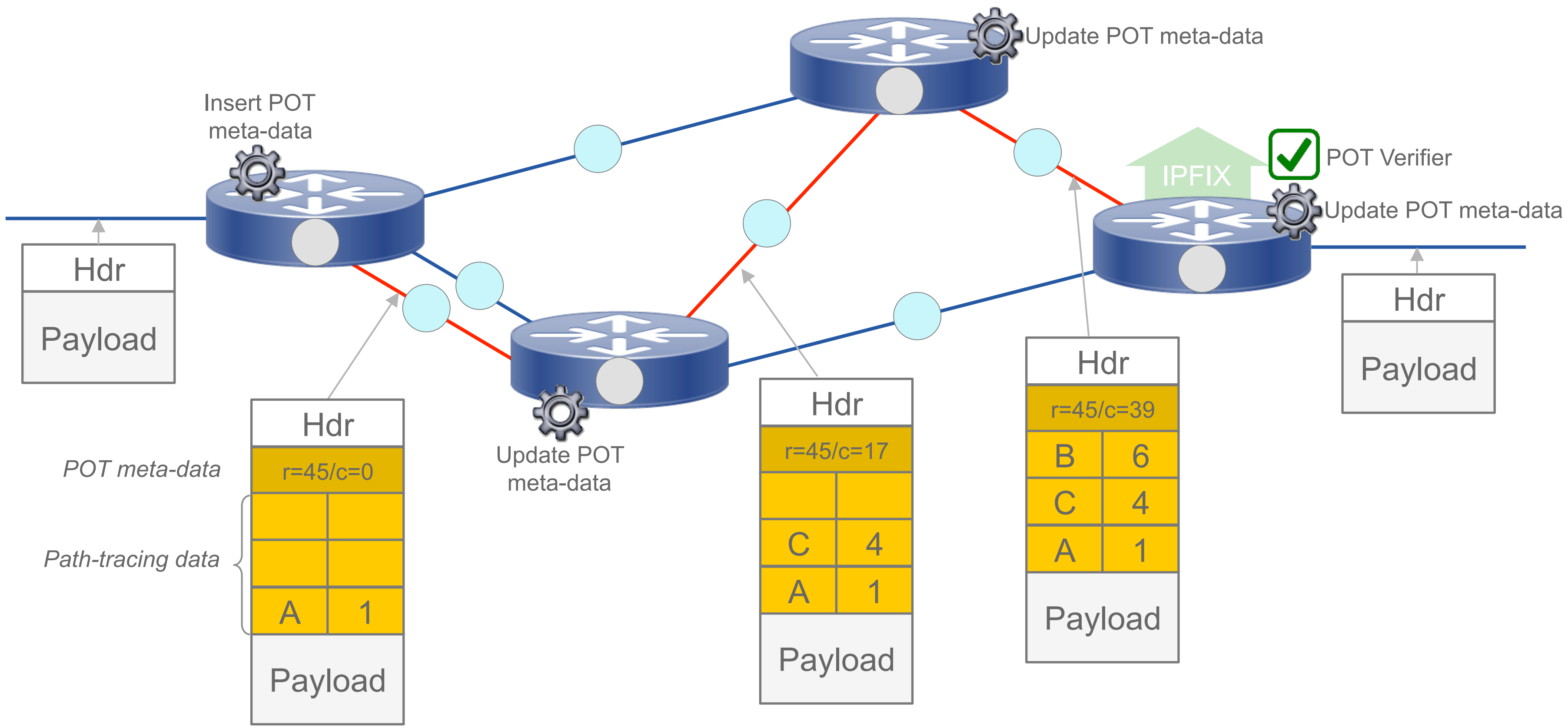
In-Band OAM

- 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



© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96



© 2016 Cisco and/or its affiliates. All rights reserved.

Next Steps

- The authors appreciate thoughts, feedback, and text on the content of the documents from the SFC WG
- The authors also value feedback on where to progress the work (in particular the POT)?
- Consider dedicated draft to specify TLV class and Type for POT TLV for Type-2 Meta-Data

© 2016 Cisco and/or its affiliates. All rights reserved. IETF 96