

In-situ OAM (IOAM)

IETF 100 Hackathon

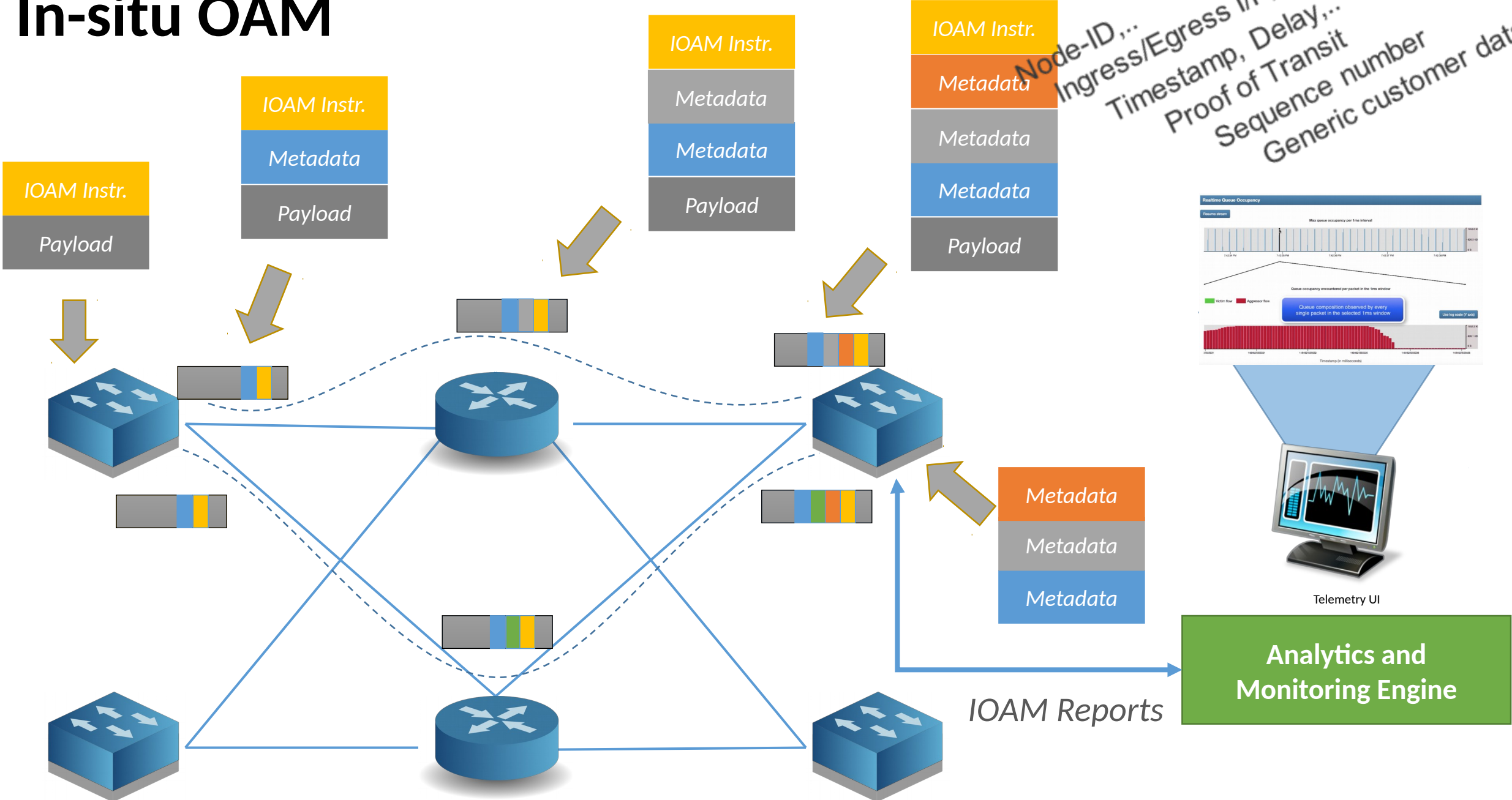
<https://tools.ietf.org/html/draft-ietf-ippm-ioam-data>

November 11-12 2017

Team

- Aditya Dogra @ Cisco << *First time at hackathon and IETF*
- Kai Lu @Huawei
- Manish Kumar (Remote) @ Fire eye << *First time at hackathon*
- Mickey Spiegel @ Barefoot << *First time at hackathon*
- Shwetha Bhandari @ Cisco
- Yap Chern Nam @ Temasek Polytechnic

In-situ OAM



Benefits of IOAM

Complete network visibility

No black boxes, silent drops and guessing.

Real-time monitoring

No need to depend on polling

Per-packet metadata

No sifting through aggregated counters and information

Full line rate, zero switch CPU involvement

Not limited by control plane bandwidth

<https://tools.ietf.org/html/draft-ietf-ippm-ioam-data>

<https://tools.ietf.org/html/draft-brockners-ioam-vxlan-gpe>

<https://tools.ietf.org/html/draft-brockners-inband-oam-transport>

Goals

- Encapsulation in multiple protocols
- Hardware implementation
- Interop between P4 and VPP data plane

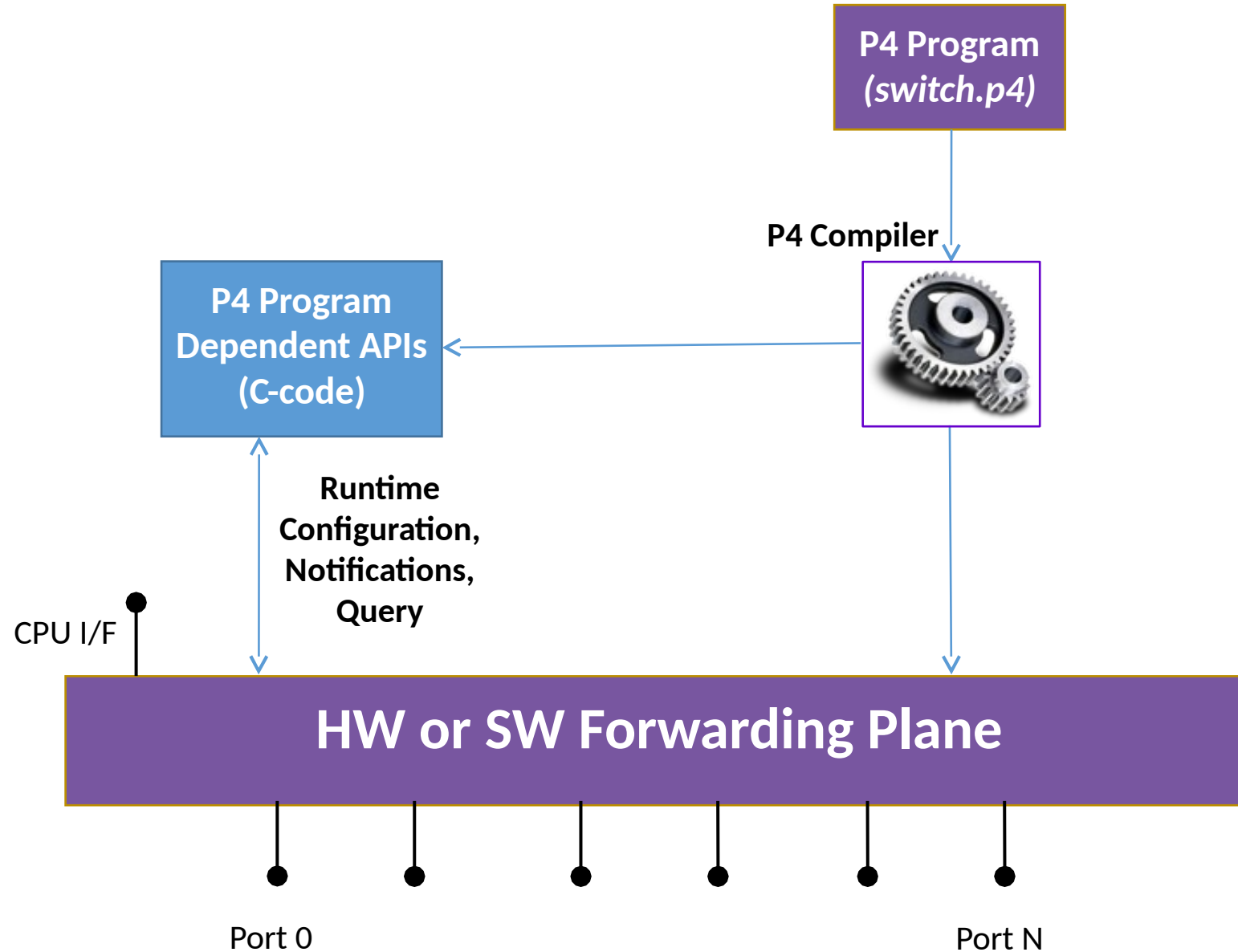
Results

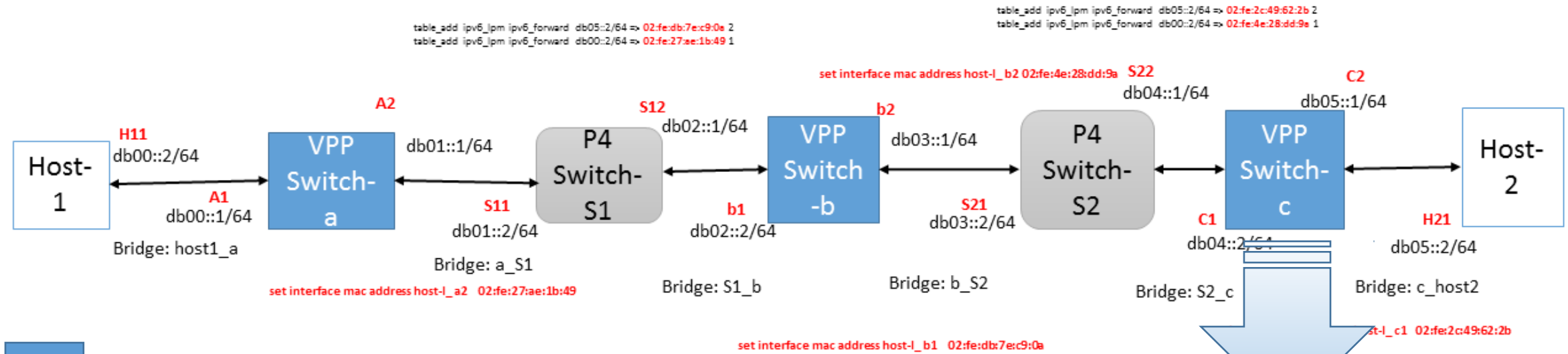
- IPv6 implementations
 - P4
 - VPP
 - Interop
- VXLAN-GPE implementation in P4
 - Performance tested on Barefoot Tofino





Programmable Forwarding Plane





VPP

P4

```

00:20:00:308461: ip6-hop-by-hop
IP6_HOP_BY_HOP: next index 12 len 64 traced 64 (Incr) Trace Type 0x9 , 2 max elts
[0] ttl 0x3d node id 0x4 ts 0x81c0b6b

[1] ttl 0x3f node id 0x2 ts 0xaff214c

(Pre-alloc) Trace Type 0x9 , 0 elts left
[0] ttl 0x3c node id 0x5 ts 0x5a0767ef

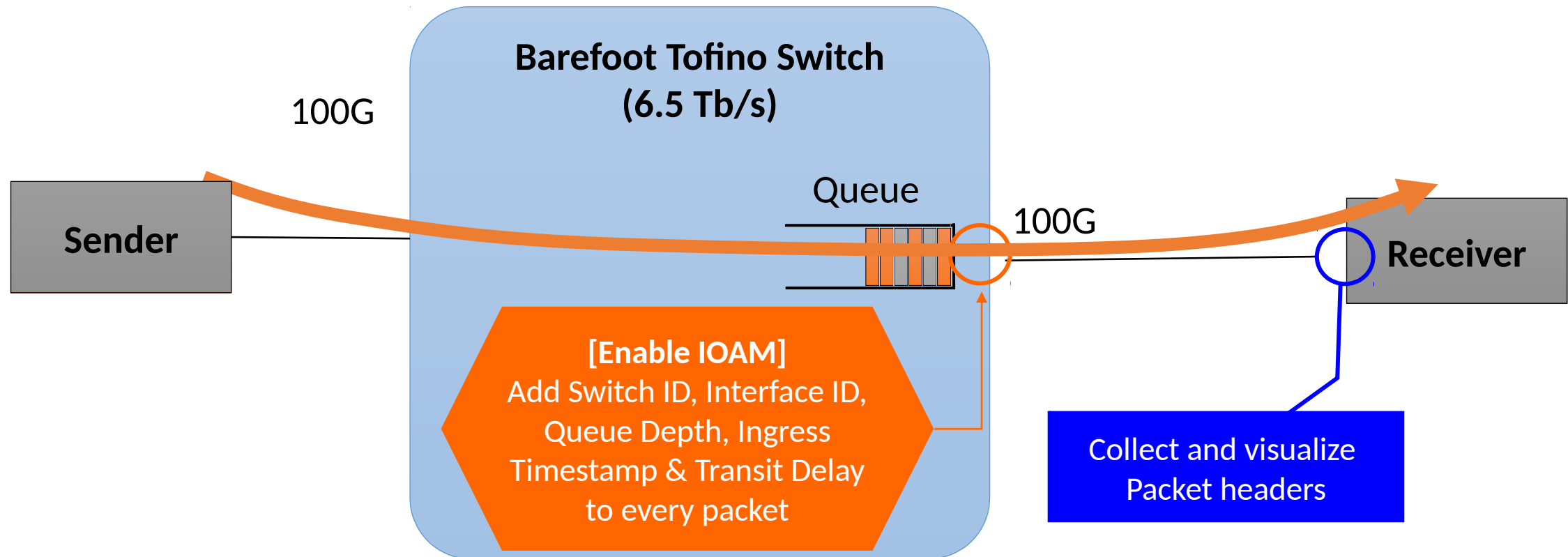
[1] ttl 0x3e node id 0x3 ts 0x5a076801

[2] ttl 0x40 node id 0x1 ts 0x5a0767f6
  
```


IOAM Incremental Trace on **BAREFOOT** Tofino Switch

NETWORKS

- Modified existing implementation of P4.org In-band Network Telemetry (INT) spec
- Support IOAM data and transport encapsulation formats
- Change in p4 code \Rightarrow change in HW behavior \Rightarrow IOAM at 6.5 Tb/s



```
###[ Ethernet ]###
dst   = 00:11:22:33:44:55
src   = 00:77:66:55:44:33
type  = IPv4
###[ IP ]###
version = 4L
ihl    = 5L
tos    = 0x0
len    = 160
id     = 1
flags  =
frag   = 0L
ttl    = 63
proto  = udp
chksum = 0x5d36
src    = 10.11.0.1
dst    = 10.10.10.1
\options \
###[ UDP ]###
sport  = 101
dport  = 4790
len    = 140
chksum = 0x0
###[ VXLAN_GPE ]###
flags  = I+P
reserved1 = 0x0
next_proto= 0x6
vni    = 4660
reserved2 = 0x0

[...]
```

```
[...]
###[ VXLAN_GPE_IOAM_header ]###
ioam_type = 0x1
length   = 0x6
rsvd     = 0x0
next_proto= 0x3
###[ IOAM_inc_trace_metadata_header ]###
ioam_trace_bits= 0x8200
node_len  = 2L
o         = 0L
l         = 0L
rsvd     = 0L
max_len   = 127L
###[ IOAM_hop_info ]###
val      = 0x3f111111L
###[ IOAM_hop_info ]###
val      = 0x3L
###[ IOAM_hop_info ]###
val      = 0x40222222L
###[ IOAM_hop_info ]###
val      = 0x5L

[...]
```

```
[...]
###[ Ethernet ]###
dst   = 00:33:33:33:33:33
src   = 00:11:11:11:11:11
type  = IPv4
###[ IP ]###
version = 4L
ihl    = 5L
tos    = 0x0
len    = 86
id     = 108
flags  =
frag   = 0L
ttl    = 64
proto  = tcp
chksum = 0x7431
src    = 2.2.2.2
dst    = 1.1.1.1
\options \
sport  = 1234
dport  = http
seq    = 0
ack    = 0
dataofs = 5L
reserved = 0L
flags  = S
window = 8192
chksum = 0x887a
urgptr = 0
options = []
###[ Raw ]###
load   =
"\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#%&'()*+,-'
```

Source code

- <https://github.com/vpp-dev/vpp.git> - ioam-ietf100
- [P4 program git](#)
- [Scripts to create p4 simple switch + vpp setup](#)

Feedback on draft-ietf-ippm-ioam-data

- Proposed encaps are feasible in both hardware and software implementations
- 5 issues raised
 - [IOAM-Trace-Type undefined bits: clarify behavior for future compatibility](#)
 - [IOAM Incremental Trace Max Length Definition](#)
 - [Queue ID and Queue Depth](#)
 - [Checksum complement effect on sink nodes?](#)
 - [Transit Delay O bit: Is it really necessary?](#)
- Ease of building open source dataplane reference implementation in VPP and P4