P4 Applications Working Group

Kick-off Meeting, November 13, 2017

https://goo.gl/K7uNb3

Introductions

Attendees signed up for kick-off meeting

Xilinx - Gordon Brebner

Mellanox - Alan Lo

Alibaba Group - Jianwen Pi

Bell Canada - Daniel Bernier

Ixia Communications - Chris Sommers

xFlow Research - Shabbir Khan

Cisco - Andy Fingerhut, Rajesh Sharma

Postech - Jonghwan Hyun

Netcope - Albert Ross

Working Group Charter

- Facilitate development and interoperability of P4 application components
- Forum for sharing open source modules and specifications
- Identify application requirements for language, architecture and API working groups
- Deliverables
  - Specifications
  - Open Source Modules
  - Test Cases

Processes

- Specifications will be posted to https://github.com/p4lang/p4-spec
  - Telemetry Specifications
    - Dataplane specification:
      p4-spec/applications/telemetry/SPEC_Inband_Network_Telemetry.pdf
    - Report specification:
      p4-spec/applications/telemetry/DRAFT_Telemetry_Report_Format.pdf
  - Create github ‘issues’ to request changes/additions to the specifications
- Repository for Applications WG: https://github.com/p4lang/p4-applications.git
  - Meeting Minutes, Source Code, Test Cases
- Mailing List: p4-apps@lists.p4.org
  - Subscribe at http://lists.p4.org/mailman/listinfo/p4-apps_lists.p4.org

Specifications for interop of P4 apps

- What a spec means for p4 application?
  - Header formats
  - P4 expressions of the headers
  - Semantics of header fields
- Not a standard, rather open-source s/w library
  - We encourage fast revision cycles
  - Use version numbering to ensure interop
- Not necessarily common denominator across all architectures/targets
  - Model arch/target-dependent features as option or runtime-selectable
  - Avoid explosion of version numbers

P4 applications to discuss after telemetry

- Criteria
  - Apps that need interop
  - Design is not trivial, need community effort
- List of apps from the member survey
  - Connection load balancing
  - Segment/source routing
  - Security (stateless firewall, whitelisting, SYN-cookie authentication)
  - Packet broker (aggregation, filtering, tool load-balancing, protocol stripping)

Telemetry Application

What should we consider for dataplane telemetry?

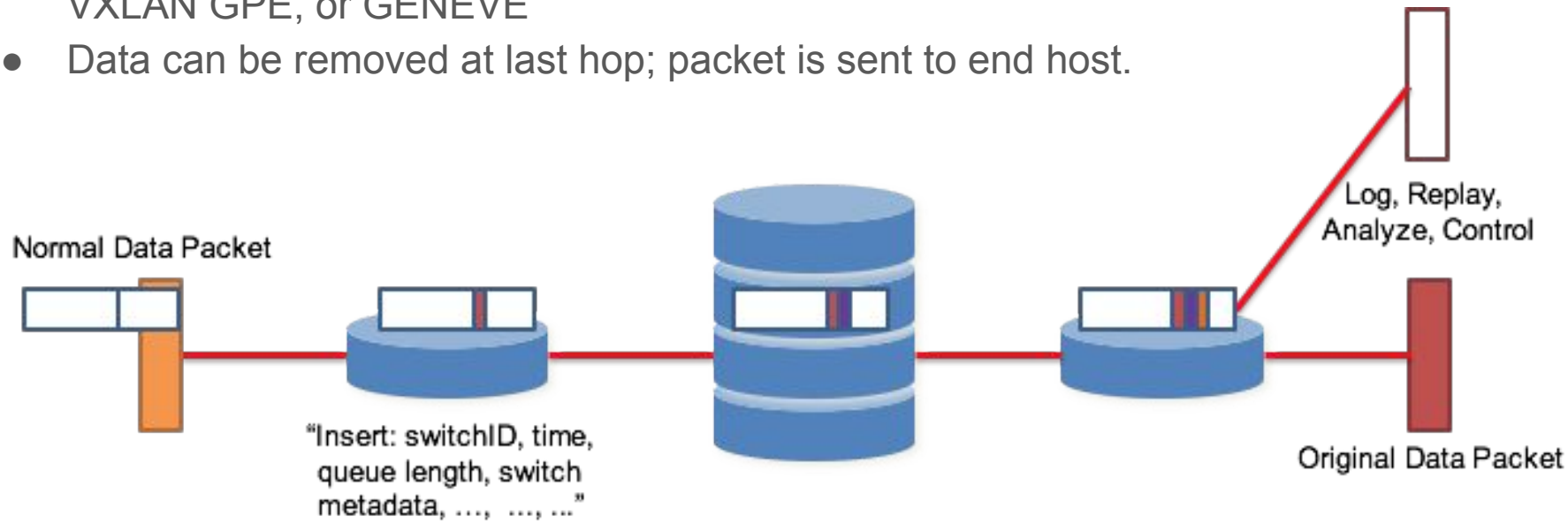
- Dataplane telemetry is centered around data packets
- Which packets/flows to monitor?
  - ACL-like table to match on packet headers (e.g., flow spec)
- What metadata to report?
  - Slice of packet header, switch ID, timestamp, in/out port IDs, ...
- Where to monitor the packets?
  - switch, port, queue
- When to generate reports?
  - Events/conditions experienced by packets to generate telemetry reports
  - E.g., flow forwarding path change event, queue congestion event
- How to monitor?
  - In-band (INT, iOAM), per-pkt postcard, dedicated probes (DPP)

Essential aspects for dataplane interoperability

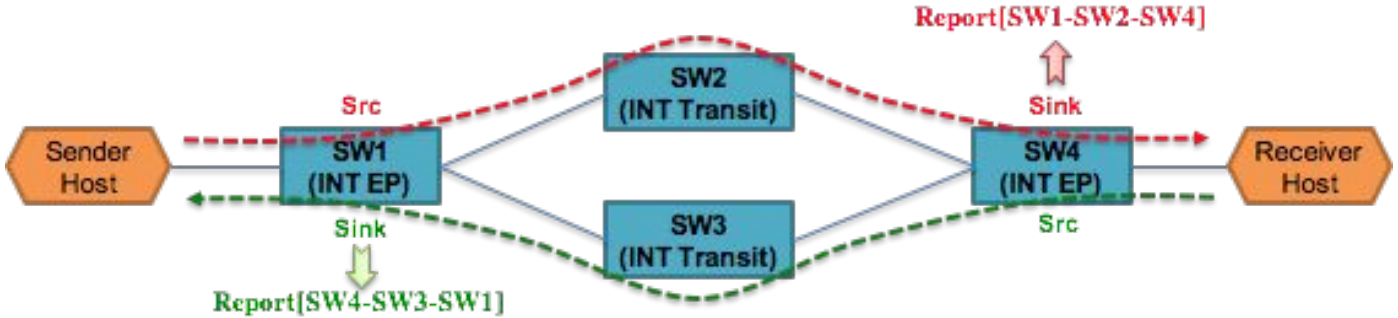
- Telemetry wire format, semantics and network-wide operations
  - Essential for interop between network devices in the same telemetry domain
  - In-band telemetry: P4.org INT, IETF iOAM
  - Out-of-band telemetry: IETF DPP (Data-Plane Probe)
- Report format and semantics
  - Crucial for interop between reporting devices and collectors
  - May need query APIs for device-specific semantics: h/w-type, timestamp unit, queue length unit, ...
- In addition, unified configuration API can help interop between layers
  - e.g., Switch OS <-> dataplane
  - Not essential for dataplane interop
  - Being discussed in OCP SAI

In-band Network Telemetry (INT)

- Packets carry instructions to insert state into packet header
- Metadata added at any point in packet header, for example, after L4 header, VXLAN GPE, or GENEVE
- Data can be removed at last hop; packet is sent to end host.



INT: device-level capabilities

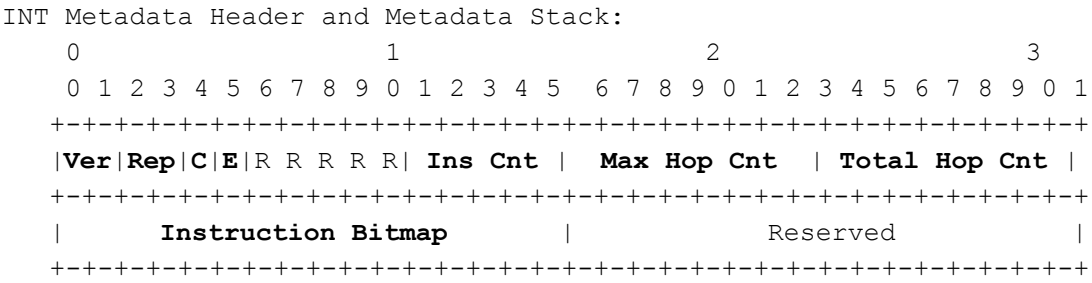


- INT Src device
  - Initiates INT by inserting “instruction header” and prepending its own local metadata to packets (that get matched on ACL-like “watch list”, outside of current spec)
- INT Transit device
  - Prepends its own local metadata per INT instruction header
- INT Sink device
  - Terminates INT and, if necessary, generates a report (upon event of interest, outside of current spec)

INT: little history

- Sep 2015: initial release
  - Hop-by-hop type and Destination type
  - VXLAN-GPE, GENEVE as transportation protocols
- June 2016: minor revision
  - Correct length field of VXLAN GPE shim header
- Oct 2017: latest spec
  - Introduce INT over TCP/UDP
  - Removed BOS (Bottom-of-Stack) bit in each 4B metadata
  - INT header formats and reference code for INT Transit in p4_16

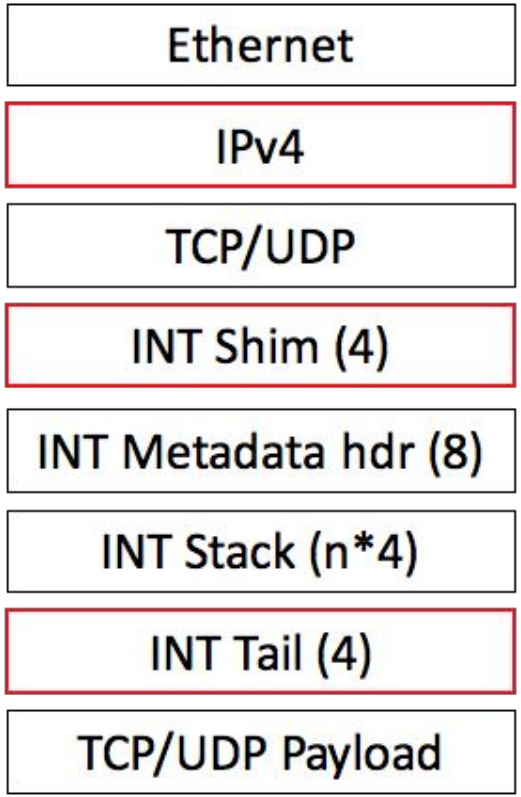
INT Metadata Header, added by INT Src



Instruction bitmap

- bit0 (MSB): Switch ID
- bit1: Ingress port ID + egress port ID
- bit2: Hop latency
- bit3: Queue ID + Queue occupancy
- bit4: Ingress timestamp
- bit5: Egress timestamp
- bit6: Queue ID + Queue congestion status
- bit7: Egress port tx utilization
- The remaining bits are reserved.

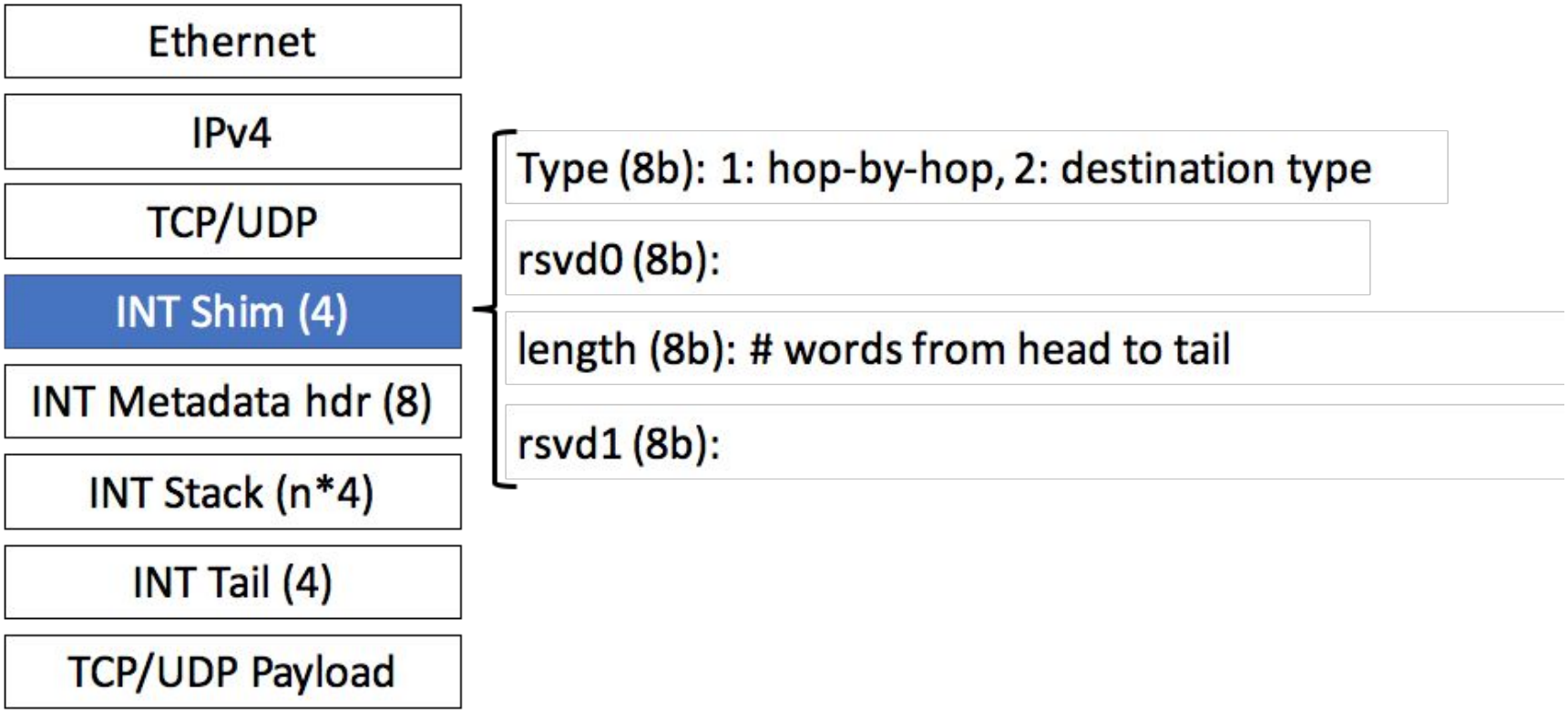
INT over L4



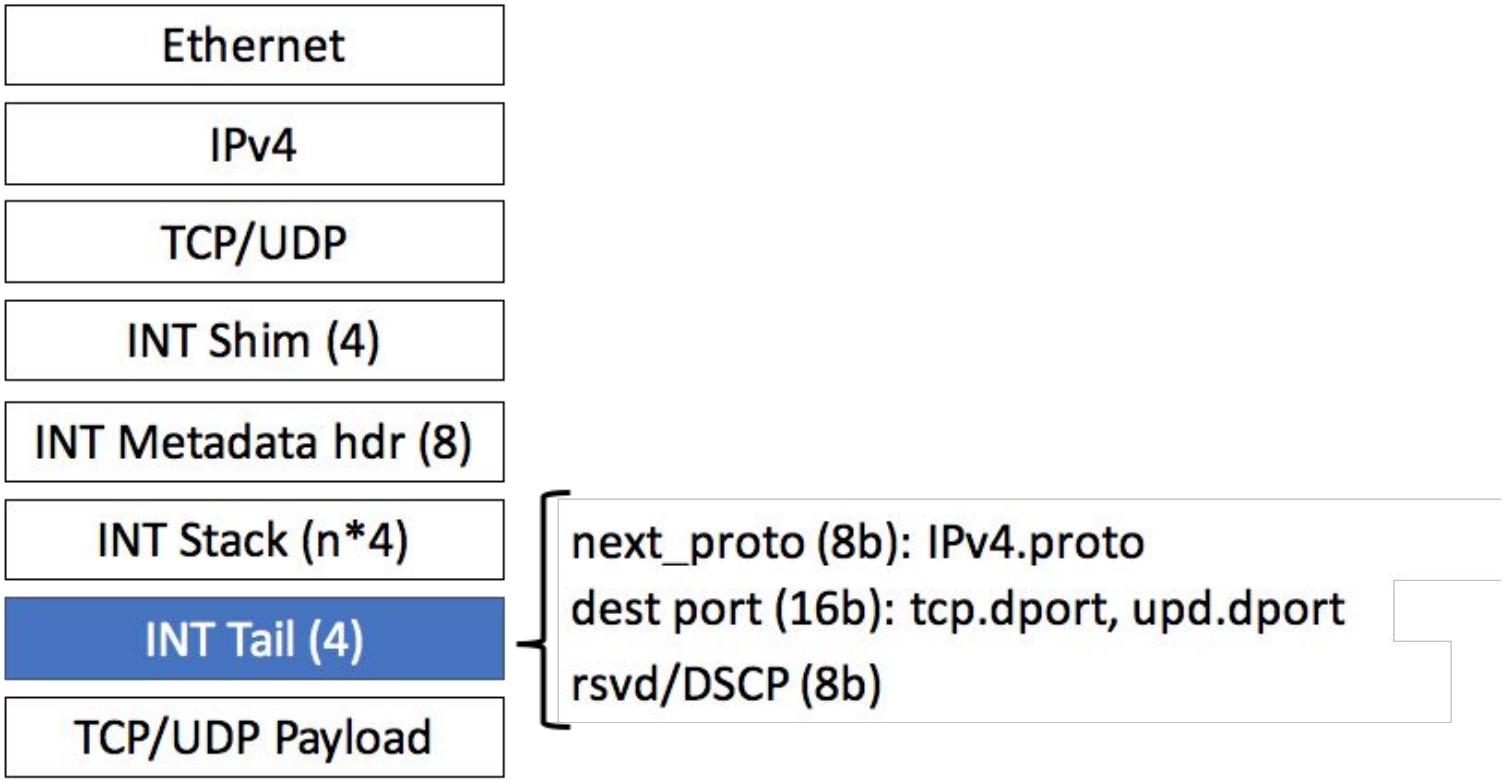
If IPv4.DSCP && bitmask == value

- Benefits:
  - Monitor both native and virtualized traffic
  - Easy to add INT stack into outer, inner, or even both layers
- Limitations:
  - May interrupt middlebox/proxy looking into L4 payload

INT over L4: Head



INT over L4: Tail



Feedbacks, questions on INT spec (1/2)

- Path MTU
  - Application must set MTU size as “PMTU *minus* max INT stack size”
  - INT Src must set **Max Hop Cnt** not to exceed PMTU
  - Basically same problem as VXLAN GW
- Overlay-underlay
  - Opt1) Encap and decap INT at tunnel encap/decap points
  - Opt2) two layers of INT: inner AND outer
    - E.g, inner INT after Geneve/VxlanGPE (or inner L4); outer INT over outer UDP
- VNF as application server or INT transit?
  - Domain-specific
  - The answer determines whether SmartNIC/vswitch is INT src/sink or transit

Feedbacks, questions on INT spec (2/2)

- Port IDs (16b) can be either physical or logical, how to differentiate?
  - Opt1) API to query a device of the semantic of each field
  - Opt2) Separate INT metadata type for **tunnel** interfaces
- Destination-type metadata headers may need a different instruction bitmap
  - SmartNIC or vswitch may want to add application-specific info
- And more..

Logistics & Next Steps

Logistics & Next steps

- Communication over github and mailing list
- Working Group Meeting every two weeks
  - Doodle Poll to get an idea of day and time that works for most people
- P4.org Calendar - https://goo.gl/Zyo87T
- Meeting minutes and reference INT dataplane code will be posted to p4-applications repository
- Next Meeting:
  - Continue where we left off on Telemetry Datapalne Spec
  - Discuss Telemetry Report Format

    p4-spec/applications/telemetry/DRAFT_Telemetry_Report_Format.pdf