P4 Applications Working Group

December 4, 2017

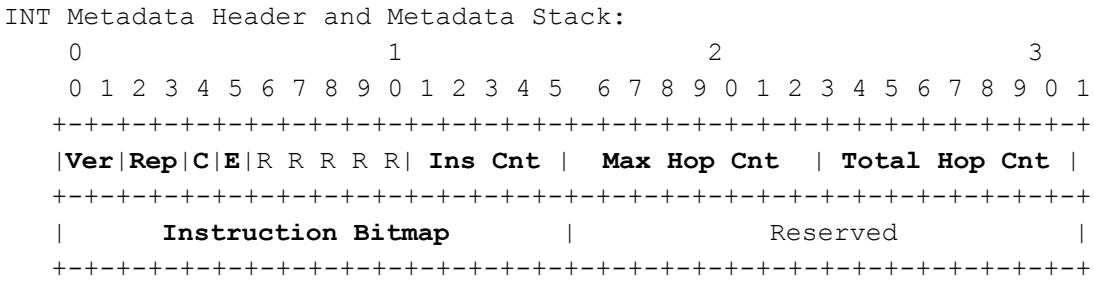
Agenda

- INT continued, revision items
- Dataplane Telemetry config model
- (Telemetry Report Format, deferred to the next meeting)

INT continued

\+ feedback and proposals

INT Metadata Header, added by INT Src



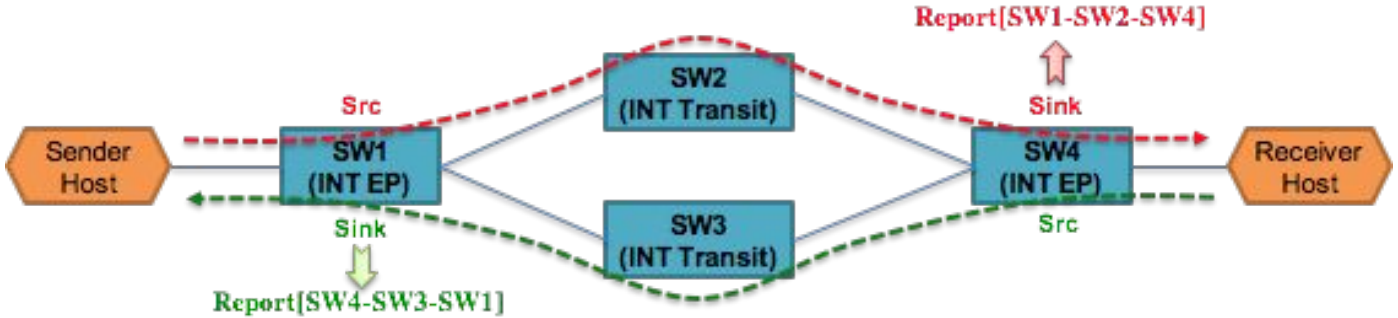
Instruction bitmap

- 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.

Revision item: increase VER bits

- Apps WG aims to adopt changes at the speed of s/w revision cycles
- 2bit Version field is too small to accommodate future revisions
- For backward compatibility, Version field size/location should be set right and fixed prior to production deployments
- **Proposal**
  - Increase Version bits from 2b to **4b or 5b** (Group decision: 4b)
  - Drop or move 2bit Rep + 1b C fields (Group decision: keep them and shift position)
    - Rep: optional field, replicate INT packets and explore valid multiple forwarding paths
    - C: indicate a copy of original packet

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)

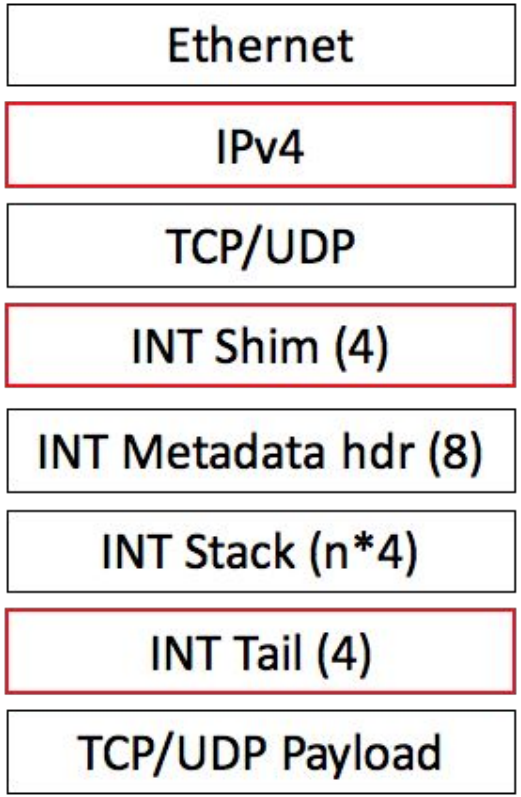
Feedback: INT initiation & termination decision

- In general, INT Src/Sink capability must reside at network edges
- INT Src must insert INT header only when INT Sink avail at the egress edge towards the destination
  - INT Src configuration can match on ‘watchlist’
- INT Sink can selectively terminate INT based on egress interface
  - INT Sink configuration can take ‘termination interface’, which is logical
- Proposal
  - Config is out of dataplane spec, but can **mention as optional feature**
  - **Group decision: this is related to network-wide config. Some usecase may require per-flow/pkt termination decision. Create a github issue and continue discussion over there.**

Feedback: accounting for INT packet drops

- INT packet may get dropped at switches other than INT Sink or at Sink
  - May lose telemetry info accumulated from the upstream switches
- Switch dropping packets can deliver INT reports even though it is not an INT sink
  - Packet drop case differs from actual INT Sink, which should forward the original data packet to the destination end-host or application stack after stripping out INT stack
  - Desirable behavior for the drop case: deliver the INT stack to the monitor
- Proposal
  - Specify the desired behavior in the spec
  - Group decision: do as proposed. Detailed how part is implementation-specific. efer to the report format document.

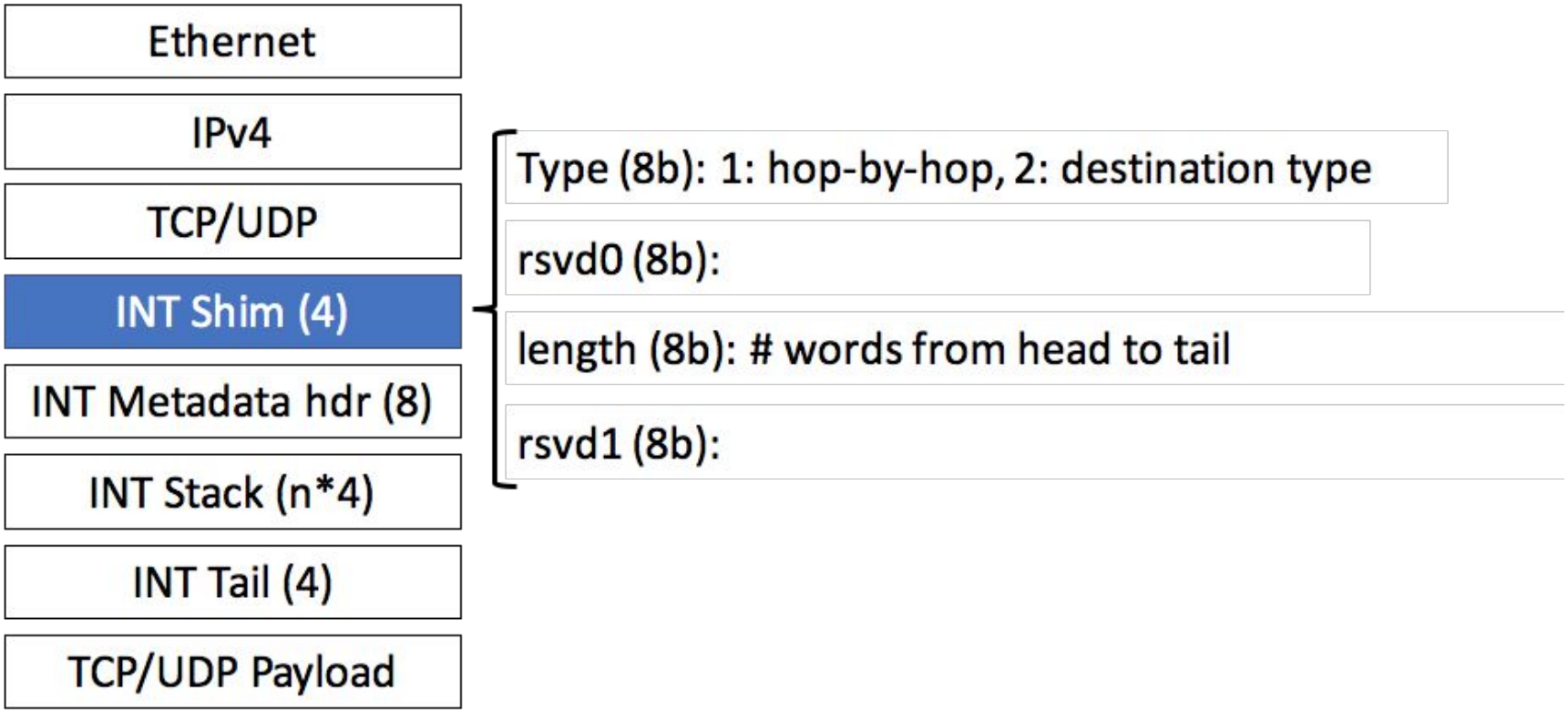
INT over L4



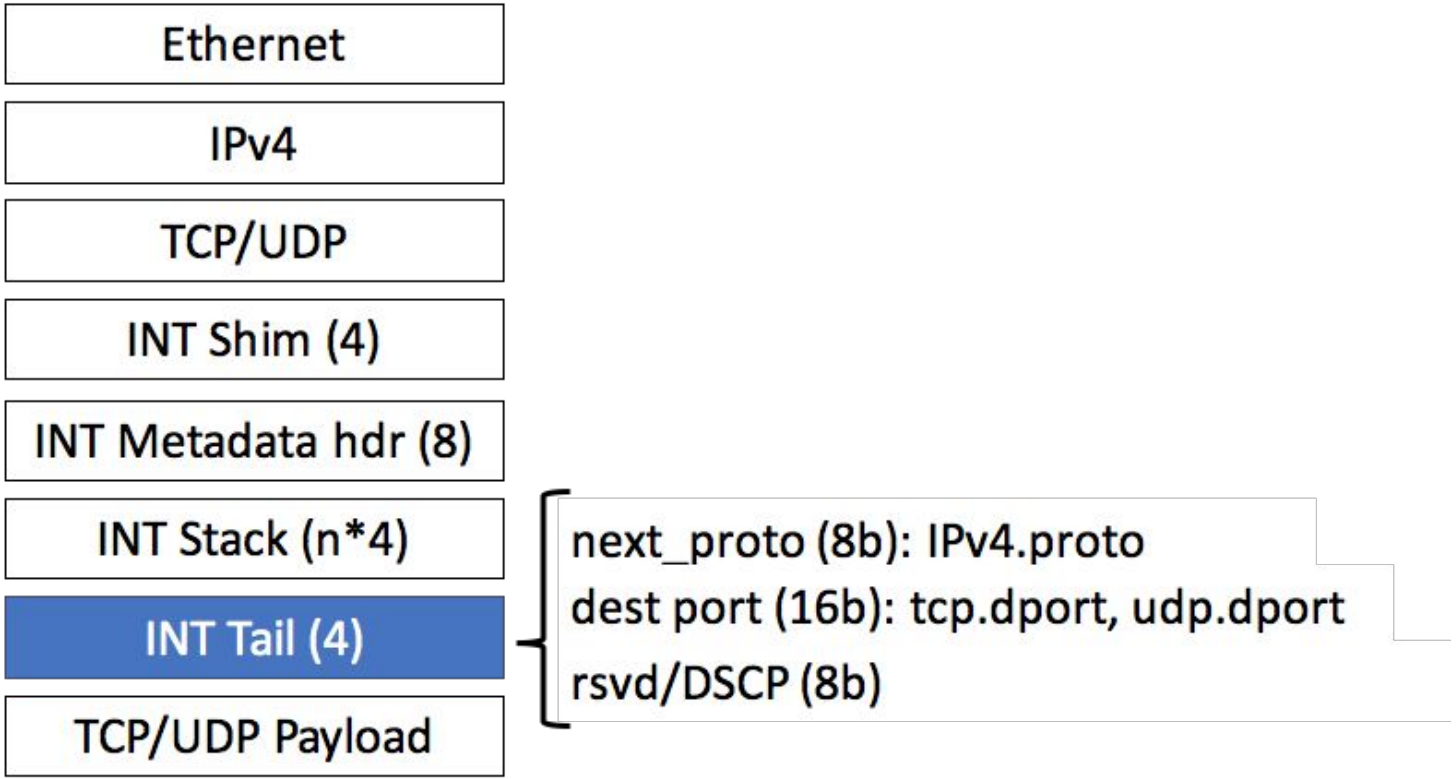
If IPv4.DSCP && bitmask == value

- 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



More feedback

- Path MTU
  - Basically same problem as VXLAN GW. The spec currently talks about:
  - 1) End-host must set MTU size as “PMTU *minus* max INT stack size”
  - 2) INT Src must set Max Hop Cnt not to exceed PMTU
  - **Proposal: 3) if MTU was about to exceed, INT transit device can set E bit, stop adding INT**
  - **Group decision: do as proposed. INT transit can also notify the Src and End-host via ICMP**
- Overlay-underlay monitoring
  - In general, encap and decap INT at tunnel encap/decap points
  - Two layers of INT is possible: inner AND outer
    - E.g, inner INT after Geneve/VxlanGPE (or inner L4); outer INT over outer UDP
    - Assumption: underlay will update INT stack only at outer

More feedback, questions

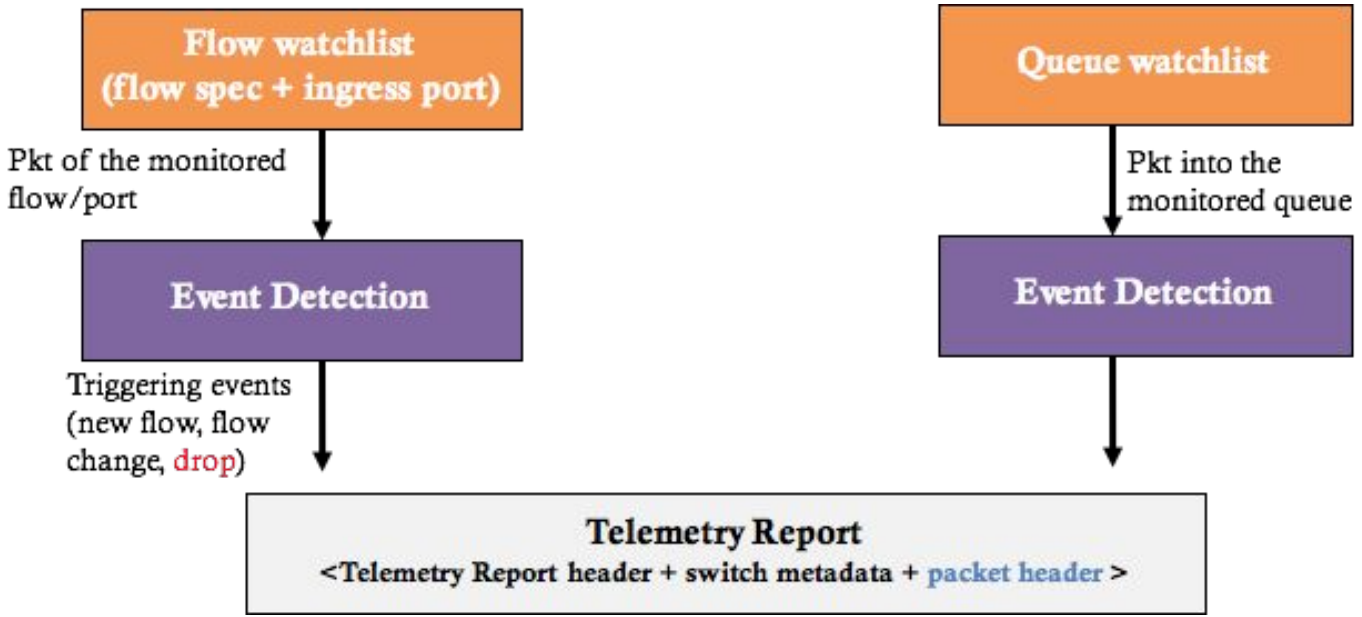
- Port IDs (16b) can be either physical or logical, how to differentiate?
  - Opt1) API to query device-specific semantics of a port field (physical, logical, tunnel interface?)
  - Opt2) Separate INT metadata instruction for **logical/tunnel** interfaces
  - Group decision: gather use cases when
- Destination-type metadata headers may need a different instruction bitmap
  - SmartNIC or vswitch may want to add application-specific info
  - We welcome proposals from NIC, vswitch vendors

Dataplane Telemetry config model

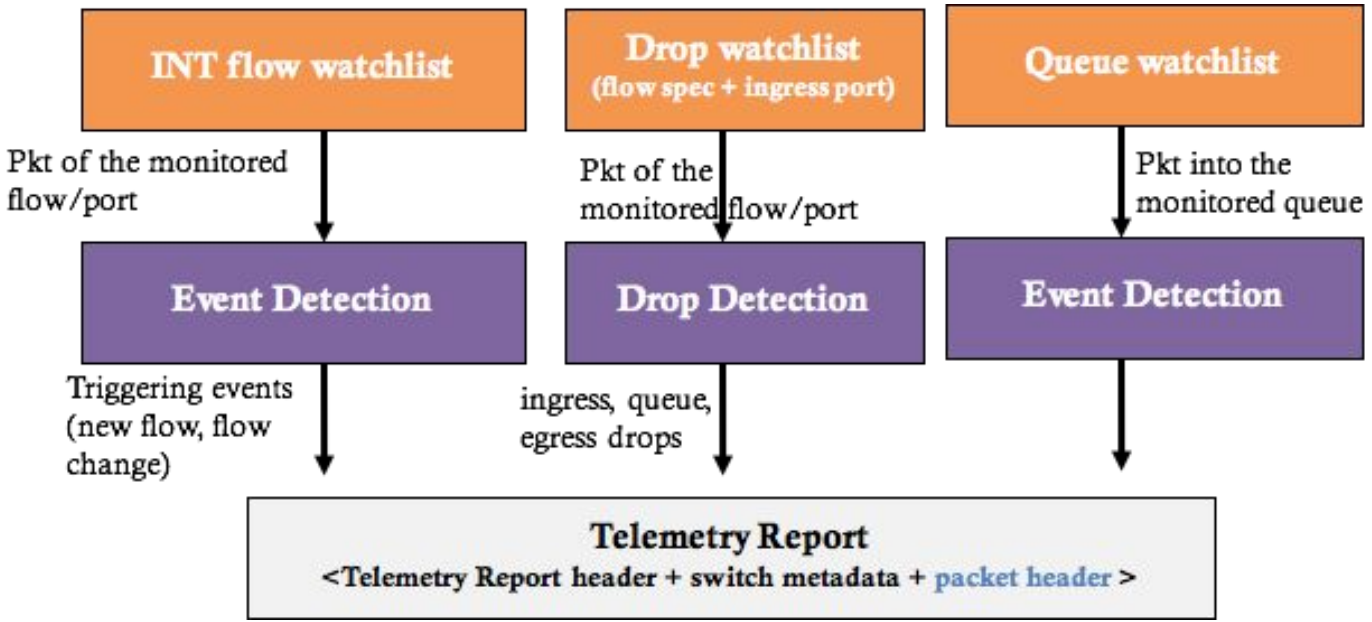
Questions for dataplane telemetry config

- Dataplane telemetry is centered around data **packets**
  - Config model closely interacts with other dataplane table/object models
- Where to monitor the packets at?
  - switch, port, queue
- Which packets/flows to monitor?
  - ACL-like watchlist table to match on packet headers (e.g., flow spec)
  - Where to put this table in L2/L3 processing pipeline?
- What metadata to report?
  - Slice of packet header, switch ID, timestamp, in/out port IDs, ...
- How to monitor?
  - In-band (INT, iOAM), per-pkt postcard, dedicated probes (DPP)
- When/where to generate the reports?
  - Events/conditions experienced by packets to generate telemetry reports
  - Load balancing of the report traffic?

Abstract model



Config model



- Treat packet drop as a first-class citizen, Drop Watchlist at every switch
  - INT watchlist at INT Src device
- 3 report types: Flow, Drop, Queue

Next Steps

INT and Report Format specs

- Moving from `p4-specs` to `p4-applications` github repo
  - There would be lots of specs, one for each application needing interop
  - Track specs, reference codes and test cases in one place
- Will convert them to editable formats (.mdk), with Makefile
  - Travis CI will auto-gen PDF, will be posted to p4.org
- Will create **github issues** for the revision items discussed today
- Will create Pull Requests for the items we close on today

- If you have any suggestion/proposal, pls go ahead and open a github issue!