INT Change Proposal

Michael Orr

michael.orr@intel.com

Who's This Guy?

- Currently: Intel DC Group/Connectivity Group (DCG/CG) CTO office
  - Years of switching chip (ASICs, NPU's) and switch control-plane experience
    - Marvell (Incl. Xelerated), Ericsson Spider, some BRCM XGS and Dune …
    - Wrote F. Specs and I/F Specs (CLI, GUI) for many commercially successful "legacy" switches
- Used to be active in ONF/OpenFlow (ahhh, the good old bad old days …)
  - (Actually was part of Stanford OpenFlow group even before ONF, starting ~ v0.9)
  - Extensibility WG, Hybrid WG, Testing WG
  - Chair of CAB for the 3 years it existed (Chipmakers Advisory Board)
    - If you have any cats to herd, I'm your man!
- Specified 1st commercially released Integrated-Hybrid OpenFLow Switch

ONF/OpenFlow Experience (Partial List, and - (very) Personal Views)

- OpenFlow Standard Defined the required Functionality of switches in PDF spec
  - Per-vendor/per-model nuances ➔ Push to (Lowest) Common Denominator as mandatory, rest "Optional" or out

- "Extensions" and revisions of main standard done by ONF WG's
  - Active members got their stuff in, sometimes as few as 3-4 people caused a standard change
    - SOME purpose/use-case cited, technically OK, PoC implementation = Accepted. Feature-creep very common
    - No requirement to show wide-spread need/expected adoption/actual adoption, no tracking of usage later, and no way to retract a proven-near-useless item
    - "Ronin" Extensions – not quite ONF standard, not quite single-player items

- Results
  - Untestable, "compliant" = technically meaningless, lots of Unused stuff both in Extensions and in Standard
  - (almost) un-Implementable in fixed-function switches ➔ Most of the market unserved, Forklift Upgrade asked ➔ O/F Adoption hindered
  - O/F today Widely Disparaged, as well as widely available as only de-facto multi-vendor SBI solution

- Solution/Mitigation (TTP's) came too late
  - (Machine Readable) TEMPLATES specifying/describing a specific O/F pipeline instance
    - including Constrains: Ordering, Mutual Exclusions, Mandatory Couplingt, scale, etc.
  - Could be used as either a Func. Spec of an existing switch to derive usage, or a Requirement Spec to implement, auto-generation of API's and test-specs, …
  - "P3" Programming Language …

INT Current Standard – OpenFlow Deja-Vu ?

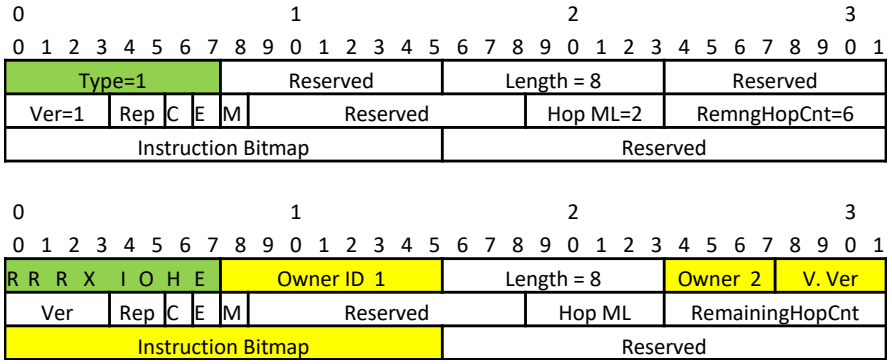
- It is Hard/Impossible to know what MD a switch supports, or to specify what it is required to support
  - Precision, accuracy, unit-of-measure, wrap-around, scope, SEMANTICS, …
    - Switch HW ALWAYS has nuances/corner cases/Gotcha's
- Current INT tries to have it both ways – Specify MD's to support, but leaves their semantics OOB
  - Pressure towards (lowest) Common Denominator
- No support for Extensibility, Staging-until-filed-proven and per-vendor MDs
- Possible future Roadblocks
  - Users will find we are missing their one favorite/crucial MD, and will want/need it added to the Standard, forcing a revision
  - Added MD's may be relevant to only a small subset of users/vendors, but and MD is either in or out of Standard
    - As soon as an MD is "in the standard", all Vendors have to upgrade, or explain why they are behind
    - Added MD's may bring side-effects (scale, mutual exclusion, load/rate limits, etc.)

Proposal principles

- Focus on defining Frame-format to carry MD's, ANY MD's
- Leave MD definition to an OOB mechanism
  - Semantics of MD's already defined to be OOB anyway, so we are half-way there
- Still, a "Standard set" of MD's is useful and SHOULD be supported
  - Keep Backward compatibility with the already-done V1.0
  - Support adding MD's to the Standard "cheaply"
  - Allow Staging of proposed-standard MD's until field-proven
- Allow Experimental and Private/per-vendor MD's

Proposal (WIP, Draft, YMMV, Broken …)

1. TYPE field becomes Bit-Flag
   - **E** – E2E Service, **H** – Hop-by-Hop
   - **O** – MD's to be carried are OWNED by "Owner ID" organization
   - **I** – Interim Switch to end-point (Optional, separate proposal)
   - **X** – Extension Header Follows (For the inevitable "we need more bits, let's add a header" time (Optional, separate proposal)
2. SETS of MD's as unit
   - Instruction bitmap interpreted as "record Set number X" as opposed to current "record MD item # X" (at least if "O" is on)
   - Have to figure out how to handle MD set Length
3. Per-"Owner" Sets
   Each vendor/user organization assigned an Owner ID
   - Owner defines numbered MD **sets,** whose Semantics are defined OOB to INT standard.
   - 4K Owners, each with 64 versions, each 16 **SETS** of MDs
     - Should cover per-model/per-SW version variations



INT header structure remains as is, but new interpretation

4. P4.Org reserves at least 2 Owner ID's (probably more)
   - One Owner ID used to define Today's V1.0 MD's as a set
   - Versions to account for different Semantics (E.g. Bytes/packets counted)
5. Standards-track MD Staging
   - Add'l P4.Org Owner ID's to define "Staging" MD sets.
   - If/when these sets are found to be "widely" adopted (for a TBD definition of "widely" and "Adopted" …) they move to the "main" Owner-ID of P4.Org, and take on the Halo of "standard", or at least "P4.Org Approved/recommended"
     - Inspiration from IEEE 802.1[?] system

While we are here … (Feature creep?)

- Current INT allows E2E or HbH.
  - I have a feeling it can be useful to have INT-PATH start or end at some middle Switch. I propose a “I” bit in the TYPE filed that works like a Router-alert in IP – causes DP of the switch to bump packet to Ctl-plane handling
    - My busy gut further thinks a “router-alert” kind of facility can prove useful in the future in some TBD way

- Always add an X bit …
  - We are likely to find, in future, we need more bits in the Header. So, this Bit will signal this header followed by an Extension Header, format/semantics of which are TBD
  - This proposal eats a lot of Previously “reserved” bits

(Asbestos Suit On)


intel®
experience
what's inside™