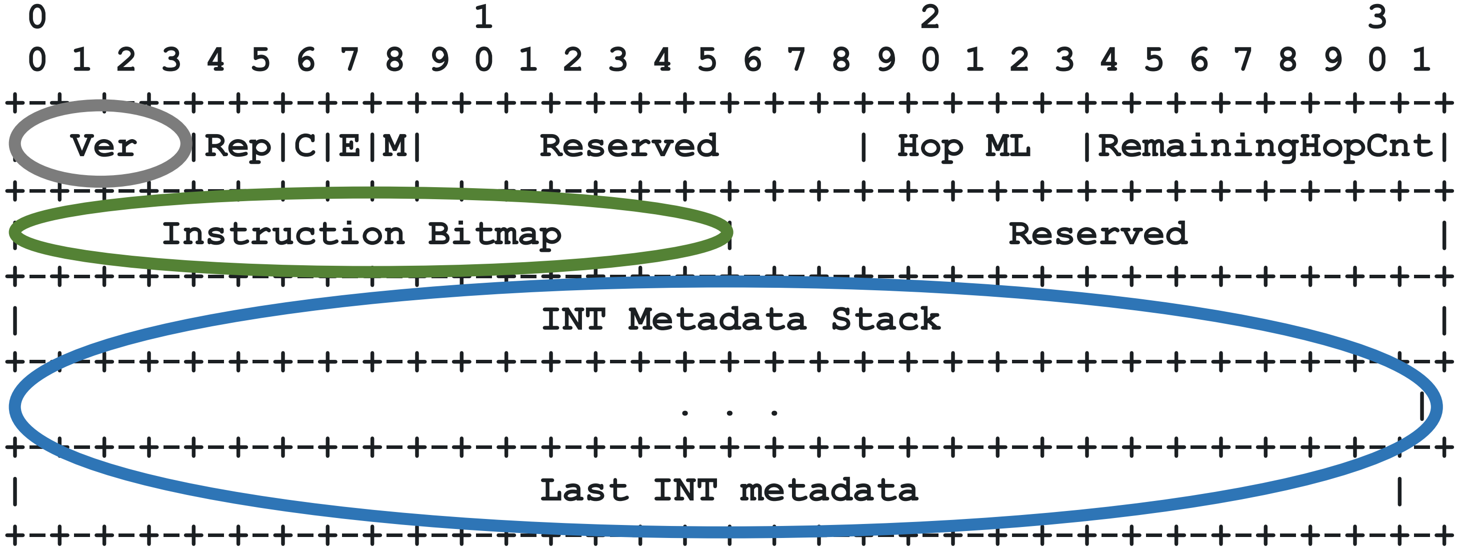
Conveying Metadata Semantics to Monitors

P4 Apps working group

May 17, 2018

INT v1.0 Hop-by-Hop Metadata Header Format



```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Ver  |Rep|C|E|M|      Reserved       | Hop ML   |RemainingHopCnt|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Instruction Bitmap       |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      INT Metadata Stack                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             . . .                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Last INT metadata                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

INT instructions are encoded as a bitmap in the 16-bit INT Instruction field:

bit0 (MSB): Switch ID

bit1: Level 1 Ingress Port ID (16 bits) + Egress Port ID (16 bits)

bit2: Hop latency

bit3: Queue ID (8 bits) + Queue occupancy (24 bits)

bit4: Ingress timestamp

bit5: Egress timestamp

bit6: Level 2 Ingress Port ID + Egress Port ID (4 bytes each)

bit7: Egress port Tx utilization

bit15: Checksum Complement

The remaining bits are reserved.

Telemetry Report v1.0 Header

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Ver  |Length |NProt|RepMdBits  | Reserved  |D|Q|F|   hw_id   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Switch id                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Ingress Timestamp                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Variable Optional Metadata                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

RepBits: Report Metadata Bits indicate which optional metadata (4 octets each) is present

- bit 0 (MSB): Ingress port id (16 bits) + Egress port id (16 bits)
- bit 1: Hop latency
- bit 2: Queue id (8 bits) + Queue occupancy (24 bits)
- bit 3: Egress Timestamp (32 bits)
- bit 4: Queue id (8 bits) + Drop reason (8 bits) + Padding (16 bits)
- bit 5: Egress port tx utilization

Aspects of Metadata Semantics

- Supported or not supported
  - May vary by version
  - May vary for INT metadata vs Telemetry Report metadata
- Units
  - Queue occupancy in bytes, cells, or packets?
    Cell size?
  - Hop latency in nanoseconds, microseconds, or 2^(-32) seconds?
  - Utilization in percent?
  - Drop reason enum definition? By reference?
- Boundary locations
  - Queue occupancy at enqueue time or at dequeue time? Includes this packet?
  - Hop latency is from where to where?
- Anything to say about ID definitions?
  - Switch
  - Port levels 1 + 2
  - Queue
- Report location for last hop metadata
  - embedded INT
  - telemetry report metadata in same report, or
  - telemetry report metadata in separate report

Proposed Telemetry Extensions Header

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| NProt |        Length         |           Schema ID           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                        Extensions data                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Presence is indicated by new *NProt* codepoint in Telemetry Report Header

- NProt: Next Protocol: Same definition as *NProt* in the Telemetry Report Header.
- Length: Indicates the length of the telemetry extensions header in multiples of 4 octets.
- Schema ID: 2-octet unsigned integer identifying the schema of the telemetry extensions data:
  - 0 - 0xEFFF : Available for private or experimental use.
  - 0xF000 - 0xFEFF : Reserved.
  - 0xFF00 - 0xFFFE : Reserved for specification by P4.org. This range is set aside for future uses such as specification of field units in INT and telemetry reports.
  - 0xFFFF : No schema, indicates that the telemetry extensions header has no content.
- Extensions data: Variable length field. This field is interpreted as specified by the schema identified by the Schema ID.

Encoding Within Telemetry Extensions

- Generated by CPU on switch, consumed by monitor
  → ease of implementation in hardware is not a concern
- Encoding options
  - List of fixed-length key value pairs
  - TLVs
  - Strings, e.g. JSON
- Dedicate one entire Schema ID value for metadata semantics?

Encoding using fixed-length key value pairs

Key	Meaning	Value Syntax	Comments

1	Hop latency	enum {
		nanoseconds,
		microseconds,
		2^(-32) seconds }	

2	Queue occupancy	enum {
		bytes,
		cells,
		packets }
			If cells, what is the cell size?
			Subdivide value?
			Or cell size is part of enum definition?
			e.g. 80 byte cells, 100 byte cells, …

3	Tx utilization	enum {
		percent }	

- 16 bit key + 16 bit value for each key value pair?
- How to convey semantics other than units?
  Separate key value pair for queue occupancy location?
- Is there anything that requires a value syntax other than enum?

Encoding using nested TLVs

Type			Meaning	Length	Value Syntax
1			Metadata	6	pad

	1		Hop latency	1	enum {
					nanoseconds,
					microseconds,
					2^(-32) seconds }

	2		Queue occupancy	3	enum {
					bytes,
					cells,
					packets }

		1	Cell size	1	integer

		2	Packet timing	1	enum {
					enqueue,
					dequeue }

	3		Tx utilization	1	enum { percent }

2			Last hop metadata	1	enum {
					embedded INT,
					same telemetry report,
					different telemetry report }

1 byte type

1 byte length in units of 4 octets (inclusive)

2 + 4*n bytes fixed part of value arbitrary nested TLVs

Encoding using JSON (JSON Schema)

```
{
    "metadata" :  {
        "type" : "object",
        "properties" : {
            "hop latency" : {
                "type" : "object",
                "properties" : {
                    "units" : { "type" : "string”,
                                "enum" : ["nanoseconds", "milliseconds", "2^(-32) seconds"] }
                }
            },
            "queue occupancy" : {
                "type" : "object",
                "properties" : {
                    "units" : { "type" : "string",
                                "enum" : ["bytes", "cells", "packets"] },
                    "cell size" : { "type" : "integer" },
                    "packet timing" : { "type" : "string",
                                        "enum" : ["enqueue", "dequeue"] }
                }
            }
        }
    },
    "last hop metadata" : { "type" : "string",
                            "enum" : ["embedded INT", "same telemetry report", "different telemetry report"] }
}
```

Encoding using JSON (JSON example)

```
{
    "metadata" : {
        "hop latency" : {
            "units" : "nanoseconds"
        },
        "queue occupancy" : {
            "units" : "cells"
            "cell size" : 80
            "packet timing" : "enqueue"
        },
        "hop latency" : {
            "units" : ”percent"
        }
    },
    "last hop metadata" : "same telemetry report"
}
```