

Extension Headers for QoS - discussion

6MAN / DetNet WG, IETF119 Brisbane, v1.0

draft-eckert-6man-qos-exthdr-discuss-00

Toerless Eckert, Futurewei USA (tte@cs.fau.de) (Editor)

Jinoo Joungh (jjoung@smu.ac.kr)

Shaofu Peng (peng.shaofu@zte.com.cn)

Xuesong Geng (gengxuesong@huawei.com)

Goals with 6MAN perspective

Admin: Work towards an actual draft-ietf-6man-qos-exthdr draft

... Currently several open technical IPv6 EH questions

... Is there enough support to do the real draft work.

... Provide summary of the use-cases (specs have likely multiple hundreds of pages in other drafts)

Technical: Single (set of) common EH to support multiple different QoS schemes

Promote instead of roadblock innovation in IPv6!

Break through core problem: Each single QoS scheme will fail to argue for new header alone (proof: 25 years IETF history)

Do not have arbitrary/enough EH code points

Do not want open-ended packet parsing if more QoS schemes show up

Differentiated by new “Method” code point in header

Define Standards / Experimental / Information(External) allocation – and extensibility

Each (set of) QoS Methods to use this header would be new doc, referring/complying to qos-exthr RFC

Common qos-exthdr RFC defines

Common processing, payload (“metadata) encoding & information model requirements

Goals with DetNet perspective

... Research/advanced HW development background: Programmable QoS

PIFO (Push In First Out) queues. Packets enqueued/dequed with desired time of departure

First scaleable PIFO algorithms for ASIC/FPGA shown recently/'24.

FPE (e.g.: P4) can calculate this time for many queuing algorithms.

Also several simpler methods for existing / easy HW (e.g.: cyclic queuing based)

... Multiple “large-scale DetNet” QoS mechanism proposals

Not using per-flow QoS, but per-packet QoS (eliminates state management, per-packet flexibility)

Requires additional in-packet metadata (which else is bound to per-flow state in routers)

... Difficult IETF /DetNet selection process. Each method has different Pro/Cons

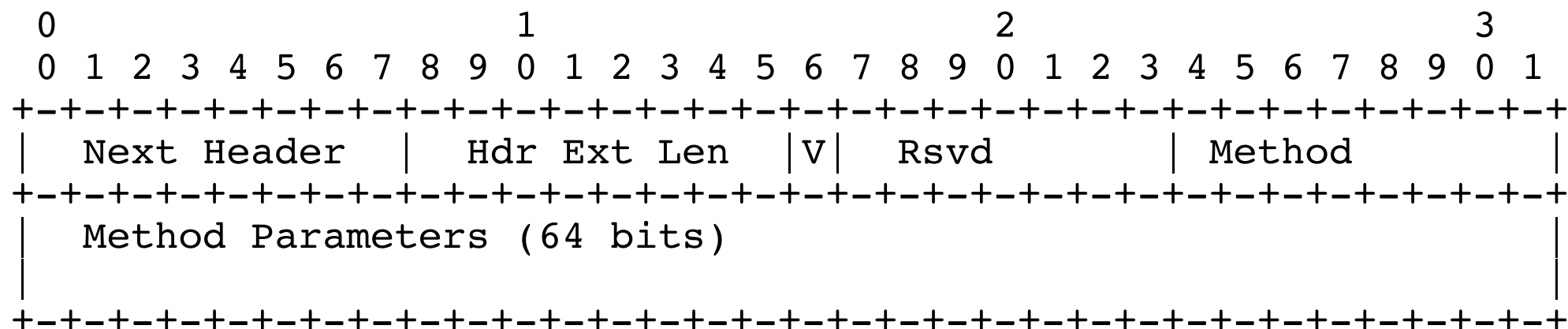
... Chicken and Egg problem: Absence of easy-to-use EH makes deployment experimentation difficult.
Absence of experimentation makes standardization difficult

... Common header / ease of getting “Method” codepoint would allow experimentation

And DetNet problem could reduce to labelling (standard / experimental / informational) the methods.

“Congestion Control” based traffic / use-cases may (hopefully) also chime in.

Possible proposal – HbH header



- Method: New registry, each Method value defines syntax/semantic of Method parameters
 - Sub-Range 1: IETF standard Methods
 - Sub-Range 2: IETF all (std, exp, inf), non-IETF (expert review, spec required)
 - Sub-Range 3: local experiment
 - Sub-Range 4: network wide configured
- Fixed size for all Methods to minimize HW-packet-parser-cost to skip unknown Methods at linerate
 TBD: will 64 bits suffice for „all“ interesting/future Methods – collect likely Methods
 - Rather waste some bits than make it harder to deploy!
- V=0. Support extensibility of Methods with V=1, only after 5 years of initial allocation of Method.

Possible proposal – Further details

- Also “Destination Option” header
 - Same format, separate Method registry.
 - See DetNet example
- Specification requirements / guidelines
 - For all spec types: Standard, Experimental, Information, external):
 - Format / Semantic of Method Parameters
 - Packet processing when Method parameter encountered
 - Limitations such as “limited to per-hop packet scheduling”
 - Permissible semantic of Method parameters
 - To address e.g. Privacy concerns upfront
 - Further details from draft-ietf-6man-hbh-processing
 - Use of additional metadata fields, e.g.: per-flow behavior with IPv6 flow-label, parameter fields in IPv6 SID/Addresses (RFC6554, RFC8754)

Examples - DetNet – see draft for summary of functionality

DetNet Destination Option Header for PREOF and timed playout

```

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Reserved      |P|                End-to-End Timestamp                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|0|0|0|0|      |                Sequence Number                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

HbH Method for per-hop bounded latency with C-SCORE Method (guaranteed latency stateless WFQ)

```

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Reserved                        | Service Rate (16 bit)      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Finish Time (usec)              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

HbH Method for per-hop bounded latency and jitter with TCQF Method (cyclic queuing with explicit cycle Identifier)

```

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Reserved                        |Prio | Cycle                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Reserved                        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

HbH Method for per-hop bounded latency and relaxed Jitter with TQF Method (orchestration period scale cyclic queuing and slip tracking)

```

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Reserved                        | Timeslot                  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                | Deviation (#slot slips)  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

HbH Method for per-hop bounded latency with minimum jitter with gLBF Method (UBS calculus with Damper)

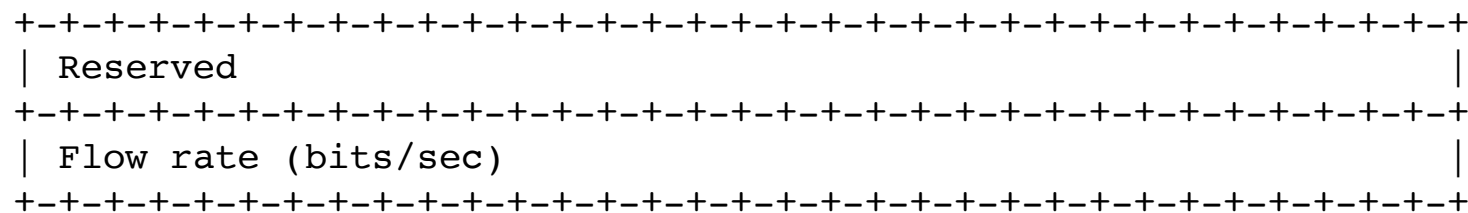
```

+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Reserved                        | PPrio | Prio |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Damper Time (usec)              |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Examples – Mechanisms for “AQM” with Congestion Controlled Traffic

- HbH Method for per-hop stateless weighted fair queuing – draft-stoica-diffserv-dps (2000)



- Possible other per-packet metadata that was considered for enhanced CC
 - In research paper or IETF discussions (but AFAIK never made it into drafts)
- More than 1 ECN bit to indicate level of congestion (e.g: LIVE research paper ca. 2010)
- More than 1/2 bit to indicate type of congestion control (as required today when ECN is used to distinguish L4S from non-L4S traffic.
- More than 1 bit to indicate degree of “Pre Congestion” (PCN vs. ECN – currently impossible to distinguish with 2 ECN bits).
- Parameters to help “Upspeeding” – aka: maximum free bandwidth on slowest congestion point along path.

From 6MAN discussions (Thanks TimC / Tom Herbert)

- Authentication metadata (Tim Chown -> draft-herbert-fast)
 - Crypto authentication like FAST should be separate HbH extension header from HbH QoS:
 - QoS HbH header must be acted on on every hop (for e.g.: per-hop latency)
 - FAST type authentication would only be acted upon on domain ingress router (no need to support on every hop)
 - FAST style cryptographic authentication not necessary in single-domain when e.g.: 802.1X is used.
 - Likely in industrial DetNet deployments.
- Accounting/Billing metadata e.g.: draft-cc-v6ops-wlwg-flow-label-marking in EH instead of flow-label
 - Strong rejection from INTAREA / TSVWG chairs/ADs in 2013 for similar (broader) proposal
 - draft-eckert-intarea-flow-metadata-framework , draft-choukir-tsvwg-flow-metadata-encoding
 - Argument: Applications can not be trusted to send unencrypted privacy critical metadata such as “Application Type” or “User related” metadata into the network. Network can/must always be considered untrustworthy.
 - Author did not believe arguments back in 2013, but will not try to have the same fight again.
 - Best to try to go with encrypted metadata with FAST for any such potentially privacy sensitive metadata
 - Need experiments with hardware forwarding plane feasible crypto operation.

Questions ?

- Please contact us when you are interested in this!