Challenges and solution directions or deterministic QoS in service provider networks, large scale and wide area networks

Academic Salon on Time-Sensitive Networking and Deterministic Applications, October 13-14, 2021

Toerless Eckert, Futurewei USA, tte@cs.fau.de
Stewart Bryant, sb@stewartbryant.com
From: draft-eckert-detnet-bounded-latency-problems-00
DetNet background

- IETF DetNet WG was created to define deterministic services for IP/MPLS
- Ongoing work, ca. 5 years in the making.
- Architecture/Use-cases/Forwarding (IP/MPLS Encapsulation) plane became RFCs 2019 – 2021
  - [https://datatracker.ietf.org/wg/detnet/documents/](https://datatracker.ietf.org/wg/detnet/documents/)
- Management / Controller-Plane ongoing – QoS hopefully coming

- Core DetNet Services:
  - No packet-loss: Packet Replication Elimination and Ordering Function (PREOF)
    - Specified for MPLS packet encapsulation only
  - Bounded Latency / (Jitter)
    - Informational overview of existing/standardized bounded latency QoS mechanisms draft.
    - No standard targets yet: “use existing QoS technologies” (which were never implemented for IP/MPLS) ?!
      - IETF GS/TSN-ATS (UBS) ?! CQF ? (several others…)

- WG process / charter challenge
  - DetNet (claims to be) not chartered to improve packet headers / QoS (yet)
  - IMHO Absence of enough deterministic network experts to work closing the gaps

- September 2021 DetNet Internim
  - Several presentation including yours truly on the DetNet bounded latency challenges
  - Result: WG agreed that it wants to work on QoS.
  - Have to see if/how WG chairs are willing to re-charter to make it happen.
Various DetNet Issues (IMHO!)

- **Scope / modularity?**
  - Some think think bounded latency always a MUST for a DetNet service
  - Some think stochastic bounded latency will suffice (and should be worked on)
  - Some think modular is important: Use-cases may only require PREOF but not bounded latency

- No PREOF for IP/IPv6 yet
  - IMHO: Easy to spec: new extension header. Deploy? (IP changes very slow in industry)

- DetNet Network design architecture expectations
  - Architecture seems to be built on the same premise / from application expertise as TSN / IntServ
    - Intserv QoS: per-hop, per-flow forwarding (DetNet flows)
    - But use-cases are meant to support wide-area, high-speed IP or MPLS networks
      - Also problem of service provider networks – aggregating multiple subscribers

- No DetNet / IETF specifications for bounded latency/jitter for IP/MPLS (beyond RFC2212)
  - Some of the mechanisms would require new IP/MPLS header fields as well
  - Would be good to have a single DetNet extension header for IP PREOF+Latency-QoS

- No use-case discussion about requirements for bounded jitter
  - Therefore unclear if there is agreement on what queuing mechanisms should be standardized?!

- Focus for main part of this presentation: deterministic bounded latency in IP networks
  - draft-eckert-detnet-bounded-latency-problems
The three enemies of simple/scalable bounded latency !?

• Per-hop, per-flow state

• Clock synchronization

• Jitter
System model of latency guarantee in networks (e.g. TSN-ATS)

Flow k requirements \( \hat{r}_k/\hat{b}_k \)  
End-to-end Min..max latency

Flow REQUEST
ACK/ NAK

Sender e.g. Flow k

IntServ/ TSN-ATS Edge

IntServ/ TSN-ATS Forwarder

IntServ/ TSN-ATS Forwarder

IntServ/ TSN-ATS Forwarder

Egress PE

Aggregating traffic from 10x PE routers,  
Very fast, cost/operational-sensitive (no-per-flow control wanted)
Desirable system model of latency guarantee in networks

Sender e.g. Flow k
Flow k requirements $r_k$, $b_k$
End-to-end Min..max latency

Ingress PE
Policing Ede
Per-flow stateless Forwarder
Per-flow stateless Forwarder
Per-flow stateless Forwarder
Egress PE

Receiver e.g. Flow k

Flow REQUEST
ACK/NAK

Network resource Database
Per-link/hop, per-queue free space
link free bandwidth

Per-flow, per-hop Database
Flow $k$: $r_k$, $b_k$, $(Q_i)$

(Calculate path or use shortest path
(slide does not show path steering)

Flow fits? Admit else reject

Install flow $k$ params ONLY into first hop router
To ensure traffic from sender complies with its parameters

(1) Look up Flow $k$
(2) Shaper param / state vars
(3) Calculate path or use shortest path
(Calculate best queue for each hop to meet min..max latency
Flow fits? Admit Else reject

Flow REQUEST
ACK/NAK

<table>
<thead>
<tr>
<th>Flow# $k$</th>
<th>Next Hop</th>
<th>$r_k$</th>
<th>$b_k$</th>
<th>time stamp</th>
<th>level</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S1,D1,Sport1, Dport1, Prot1)</td>
<td>1</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>5</td>
</tr>
<tr>
<td>(S2,D2,Sport2, Dport2, Prot2)</td>
<td>2</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>3</td>
</tr>
<tr>
<td>(550k,D50k,Sport50k, Dport50k, Prot50k)</td>
<td>50,000</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>
Feeds & Speeds considerations:

- Per-hop, per-flow state of UBS / TSN-ATS
  - Better than per-flow shaper because of Interleaved Regulators (IR)
  - But: IR reduces scheduling complexity from $O(\text{flows})$ to $O(\#\text{IIF}*\text{PRIO})$
    - IIF = Input Interfaces, PRIO = number of latency priorities (e.g.: 8)
    - In embedded/industrial switches with e.g.: 12 port big saving.
    - Aggregation routers may have hundreds of interfaces.

- TSN-ATS: target ?! 1..10 Gbps interfaces
- DetNet/SP routers: ++ 100 Gbps interfaces (400Gbps interfaces now).
  - Good news: per-packet queuing latencies accordingly shorter (serialization latency 1/40th).
    - Packet routers/switches less and less at disadvantage over optical/TDM alternatives.
  - Bad news:
    - Any clock synchronization needs to be 40 times more accurate (with same technology)
    - Any shaper/interleaved-regulator needs to operate at 40x speed
      - Very fast per-packet state table read/write cycles (very uncommon function today)
      - Edge interfaces from PE may be factor 10..40 slower (easier to do shaping there)
    - Even if only small percentage of traffic is DetNet:
      Cost of line-rate speed of shaper does not reduce (only size of state table)
Experience from IP/MPLS Multicast

• **IP-what-or-why-the-heck-would-i-care?**
  - TSN/DetNet also supposedly scoped to support multicast (actually used in TSN AFAIK)
  - Reason to discuss: We learned the crucial problems only after we solved all other problems – 10-20 years later
  - Multicast is the only IP/MPLS network service with dynamic, user-application created per-hop, per-flow state
    - … that is deployed quite widely in SP networks
    - … and that replication state is (IMHO) even less problematic than per-flow shaper state (prior slide)

• **Initial concerns: HW-state-scale**
  - Early 2000… Vendors in IETF bidding war -> 150,000 states declared working

• **Second concern: control plane performance: reliability, reconvergence speed, etc…**
  - Towards end of 200x: Lot of control plane optimizations (fast/make-before-break reconvergence,..)

• **Third / Ongoing issue:**
  - Operational concerns dealing with dynamic, user-created per-hop state.
    - Core different to traditional industrial TSN: flows are created/deleted all the time by users – same to expect/support for DetNet
  - Churn of control plane of P-routers when many states are created/deleted.
    - Even simple cases: SP’s own IPTV server rebooting: 2000 IP Multicast flows go away, return.
    - Bugs / Misconfigs: Networks went down because of state issues on core-routers.
  - Troubleshooting considered extremely difficult.
  - Often SPs decided to use unicast-workarounds to avoid having to deal with new functionality on P nodes.
    - And for deterministic bounded latency we do not even have workarounds.

• **Result: BIER (Bit-Indexed Explicit Replication) Multicast**
  - No per-hop, per-flow state on P-routers anymore. Replication through bits in packet header
  - Working on finalizing Traffic Steering for BIER (BIER-TE)
    - IMHO best current technology for DetNet multicast in SP networks (not solving latency of course)
Experience / History from Unicast

• Original per-flow protocol RSVP and in SP: RSVP-TE (MPLS)
  • Could support per-hop latency (Guaranteed Services)
  • But AFAIK no implementation that does support per-hop GS shapers.

• Deterioration of RSVP-TE requirements through use-cases
  • Guaranteed Bandwidth only traffic: no per-hop, per-flow QoS needed, just admission control
  • Network capacity optimization: Not even admission control needed, just PCE calculation of best set of steered paths to load-split traffic in network. This is today's 99% of reason for most SP to do Traffic Engineering

• Replacement of RSVP-TE by Segment Routing
  • Source-routing of packets via source-routing header (MPLS, IPv6 (SRH))
  • Eliminates any signaling to P routers when flows/tunnels are added/removed
    • Solutions in IETF for flow-signaling to P-routers also met with little interest by most operators
  • For capacity optimization, short headers suffice (today's use-cases)
  • “Easily” used for strict hop-by-hop steered Deterministic traffic flows
    • Just add admission controller to PCE (as done with RSVP-TE)
    • But may want to have more compact source routing header for IPv6 (ongoing work in IETF)

• Summary: should have a hop-by-hop bounded latency solution for SR and BIER
  • Same stateless QoS would work for both (IMHO)
Clock Synchronization

• HW cost factor
  • PTP common in some type of industrial networks and “Fronthaul” 3/4/5G networks
  • Quite uncommon / undesirable in any larger / faster networks
  • Ca. 8 years ago: “All ethernet MAC will support PTP”
    • Did not seem to have come true ubiquitously
  • Only NTP (msec) WAN network clock synchronization fairly ubiquitous

• Operational Cost (especially wide area network)
  • Network wide PTP setup / management yet another specialty OPS requirement
  • Need to measure link latency asymmetries, temperature drifts (copper), etc. pp.

• Cost of synchronization goes up with required accuracy
  • Temperature controlled oscillators etc..
  • Shapers for faster links require higher accuracy
  • Or account for higher bounded latency from clocking inaccuracy.
    • And we where so happy that faster links reduce queuing latency…

• Ideally reduce required clock synchronization accuracy to “free”
  • E.g. Whatever the interfaces themselves need already to reclock on receiver side.
  • SyncE can be in this ball-park.
**Jitter**

Shaper based solutions (GS/UBS/TSN-ATS) have MAXIMUM jitter
- No-competing traffic: no queuing latency
- Max-competing traffic: maximum queuing/shaping latency

Deterministic app MUST accept any packet to arrive with maximum latency
- AFAIK?! Only few application MAY benefit from opportunistic earlier arriving packets
  - Some telemetry, financial market data
- MOST?! Applications use playout buffers ro “re-sync” traffic to guaranteed latency
  - Most control loops, Most media playout
  - Requires upfront knowledge of network size/hop == maximum jitter!
  - Jitter may be acceptable, but is rarely a benefit for deterministic service ?!

Synchronous (no-jitter) packet delivery allows to operate client devices without synchronized clocks on the clients
  - PLL, (IoT) streaming media clients

One reason for TSN to develop ATS was to eliminate need for clock synchronization
- But if the client devices of the network need clock synchronization then we have not eliminated the need for it from the network.
- If we are lucky, the clients clock-synchronization may just be less accurate than what a "synchronous" deterministic latency network service would require

---

**Diagram:**

- **Sensor**
  - **Measurement Packet**
- **Actor**
  - **Command Packet**
- **Network latency and jitter compensation**
- **PLC Programable Logic Controller**
  - **Measurement Packet**
  - **Dejitter Packet**
  - **Dejitter Packet**
  - **Command Packet**
- **PLL Phase Locked Loop compute**

---

**Jitter problem example:**
When network introduces jitter,
Sensors and actors in a tight PLC Control loop need to have their own accurately synchronized clocks so the PLC knows/controls when their sensor Data / action happen(ed)
Proposed immediately possible solution: Tagged CQF

• IEEE 802.1Qch – Cyclic Queuing and Forwarding
  Hop-by-hop forwarding via 2 gated queues
  Pro
  • Per-flow, per-hop stateless !
  • Path independent low jitter in order of cycle time (e.g.: 100…20 usec at 100Gbps).
  Con
  • Packets are assigned to cycle queue based on arrival time
  • Requires highly accurate clock synchronization
    • E.g.: better than 1% of cycle time.
  • Propagation latency of link eats into throughput
    • Propagation latency has to be << 10% of cycle time
    • With 100 usec cycle times, feasible distance of links < 3Km

• Tagged CQF
  • draft-dang-queuing-with-multiple-cyclic-buffers (earlier: draft-qiang-detnet-large-scale-detnet)
  • Carry cycle identifier in packet header
  • Eliminates link distance / delay-variation limitations
  • Reduces required clock accuracy by factor 10 or more over CQF
    • Depending on number of cycles
  • Implemented and prototype deployed on 100Gbps++ routers an 2000km size network.
  • Packet header only needs to care cycle identifier, e.g.: 1..4 – possible to do with existing headers
    • draft-eckert-detnet-mpls-tc-tcqf
T-CQF example

Node 1 (sender)

Node 2 ("receiver")

With Time Interval Error (TIE) (clock drift)

Outgoing interface
Summary / outlook

• High-speed, low-cost, large scale network bounded latency
  • Better without per-hop, per-flow state, no or “relaxed” clock synchronization

• Deterministic bounded latency applications
  • More often benefit from low jitter than opportunistic earliest arrival

• IETF DetNet WG wants to start exploring queuing options / work
  • Great opportunity to start engaging with DetNet if you are interested in this
  • Also very interested in collaboration on this.

• Advertisement:
  • See also CNSM 2021, HIPNET workshop Oct 29h:
    • gLBF: Per-flow stateless packet forwarding with guaranteed latency and near-synchronous jitter
Thank You.