Deterministic QoS for MPLS data plane considerations:

draft-eckert-detnet-mpls-tc-tcqf-01 and beyond

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

Stewart Bryant, University of Surrey ICS (s.bryant@surrey.ac.uk)

Andy Malis (agmalis@gmail.com)

MPLS WG, 11/08/2021

Why ?

- Bounded latency QoS solution for DetNet. Debrief to MPLS WG of risk
  - Please get involved, voice your opinion (hint: support for our draft is a great option!)

- Current MPLS forwarding plane allows to only support (per-hop,per-flow) bounded latency
  - Aka: „per-flow shaper (interleaved regulator)“ on every MPLS P-node hop
  - IntServ Guaranteed Service (RFC2212) or IEEE TSN „Asynchronous Traffic Shaping“
  - Both only need to identify flow (by MPLS label)
  - Both them „only“ need for management plane to set up a shaper per-label
    - *No yang model for management plane defined though*
  - Also likely no high-speed routers supporting such shapers at scale
  - Not in-line with preferred Service Provider MPLS designs == Segment-Routing Networks
    - Scalability / low-churn control-plane prefers per-hop, per-flow stateless operations.

- RFC2212/TSN-ATS have maximum jitter of bounded latency options
  - Minimum competing traffic: 0 queuing latency. Maximum competing traffic: maximum latency
  - This makes applications difficult, require more clock synchronization
  - Strong statements from industry / industrial that jitter need to be minimized
    - Aka: better to always have maximum latency, independent of competing traffic.

What options do we have ?

- Without extensions/changes to MPLS packet header / processing

  - draft-eckert-detnet-mpls-tc-tcqf

  - Per-hop, per-flow stateless
  - Very low jitter, independent of path properties (#hops, per-hop latency)
  - Relies on defining a new per-hop-qos behavior.
    - Utilizes 3..5 MPLS TC value (Traffic Classes).

  - ?! Only short term option - No MPLS packet header change
    - For Service-Provider Class MPLS Networks

  - PoC validation deployment 2000 KM WAN network
    - 100Gbps+, high-speed WAN routers, simple QoS FPGA code

What options do we have ?

- With extensions/changes to MPLS packet header / processing
  - Various options.
    - IMHO dangerous to standardize a NON-extensible solution now
  - IHMO should work through potentially desirable better QoS options
    - This could go well beyond “bounded latency”
  - IMHO subject for QoS packet header design team
  - Would be great if we would not do this just for MPLS
    - See ideas MPLS-DT to have common extension header MPLS/IPv6

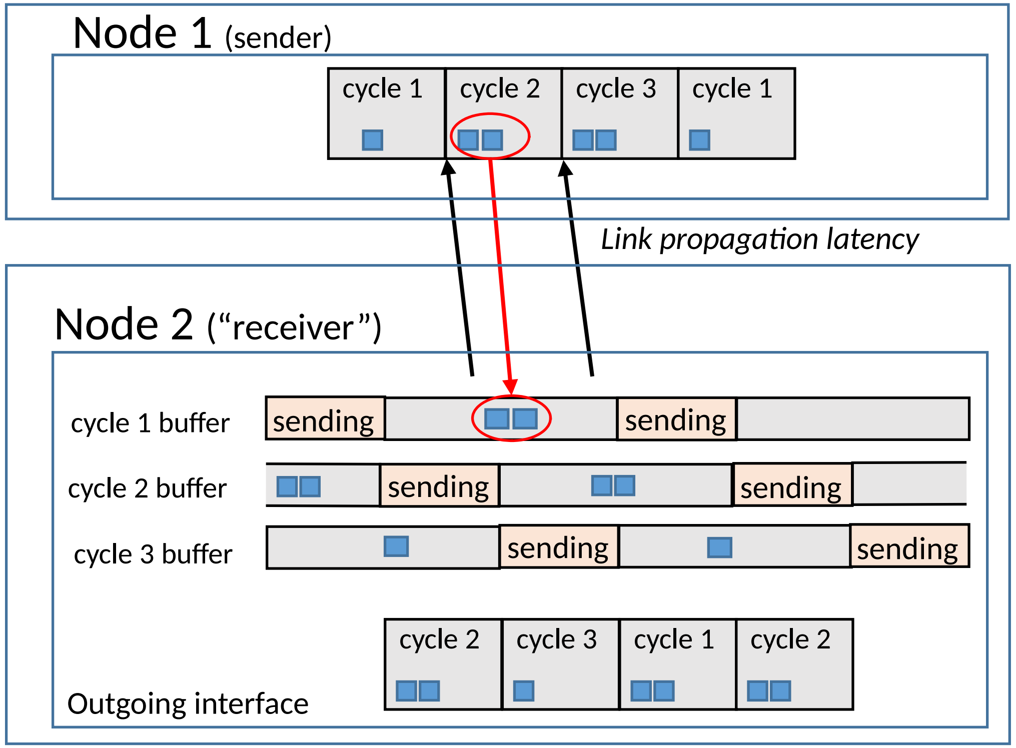
Excerpt: How does it work

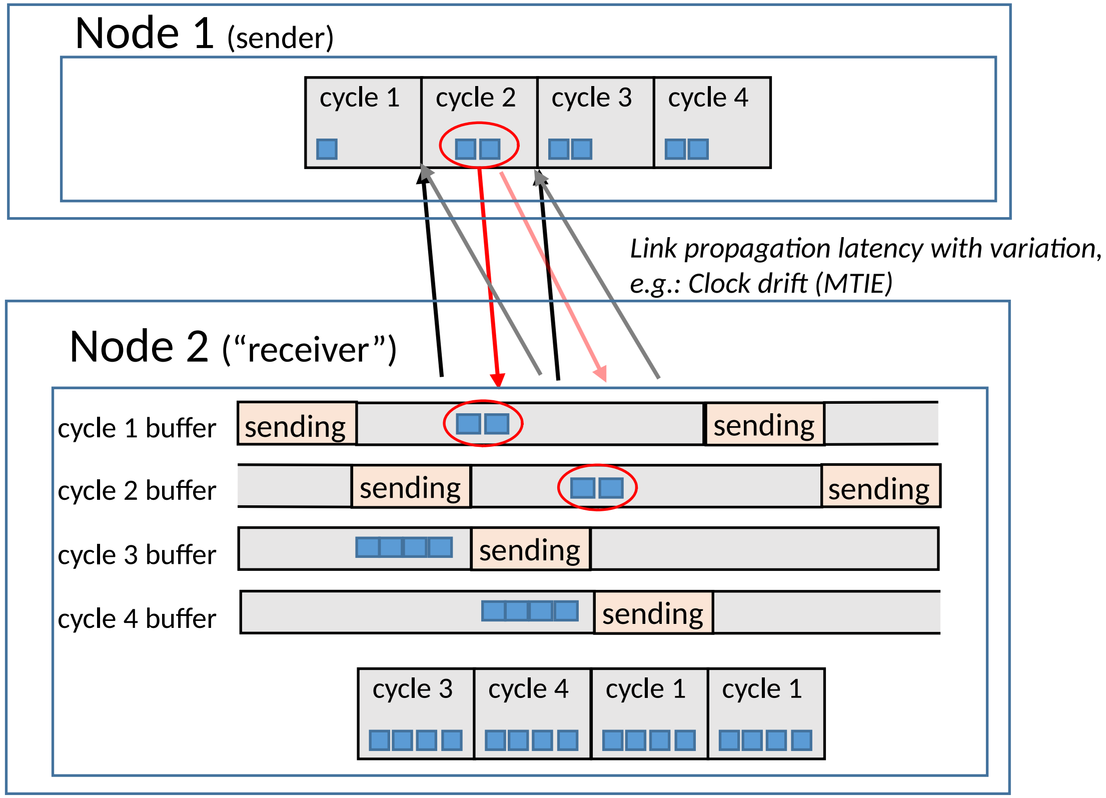
Packets are forwarded in periodic (e.g. 100usec cycles). Minimum: 3 cycles

- Existing concept: TSN Cyclic Queing and Forwarding (2 Cycles) – Available in <= 10 Gbps ethernet switches for many years.
- In TSN cycle not carried in packet, but derived from packet arrival time: Strict time-sync and short links are result

Tagged solution: Cycle number is indicated in TC field

- With more than 3 cycles we can overcome more jitter of clocks and on links.





More… come to DetNet WG…

We even have QoS forwarding pseudocode!

```
tcqf = ietf-detnet-tcqf

void receive(pak) {
  // Receive side TCQF - remember cycle in
  // packet internal header
  iif = pak.context.iif
  if(tcqf.tcqf-if-config[iif]) { // TCQF enabled
    if(tcqf.tcqf-mpls-tc-tag[iif]) { // TC-TCQF
      pak.context.tcqf_cycle =
        map_tc2cycle(pak.mpls_header.lse[tos].tc,
             tcqf.tcqf-mpls-tc-tag[iif])
    } else // future encap/tagging options for TCQF
  }
  // Forwarding including any label stack operations
  oif = pak.context.oif = forward_process(pak)

  // ... optional  DetNet PREOF functions here
  // ... if router is DetNet service node
```

```
// Output interface TCQF enqueuing

if(pak.context.tcqf_cycle && // non TCQF packets value is 0
  tcqf.tcqf-if-config[oif]) { // TCQF enabled
  // Map tcqf_cycle for iif to oif mapping table

  cycle = pak.context.tcqf_cycle =  map_cycle(cycle,
       tcqf.tcqf-if-config[oif].tcqf-iif-cycle-map[[iif])

  // Map cycle to TC value of ToS LSE
  if(tcqf.tcqf-mpls-tc-tag[iif]) { // TC-TCQF
     pak.mpls_header.lse[tos].tc =
       map_cycle2tc(cycle, tcqf.tcqf-mpls-tc-tag[oif])
  } else // future encap/tagging options for TCQF

  // Enqueue into cycle buffer
  tcqf_enqueue(pak, oif.cycleq[cycle])
```

Scheme does of course work equally for IP via 3..5 DSCP values. And can mix & match easily
But MPLS (With SR) is most likely, most desirable immediate forwarding plane option.

Why extensible QoS header ?

- Deterministic QoS
  - PREOF: Would like to have ‘sequence-number’ header field not only for MPLS, but also IP
    - Else we end up unnecessary encapsulating IP in MPLS just for that header.
  - Bounded latency: Alternative (to T-CQF): Per-hop, per-flow - stateless, ’zero-jitter’ deterministic
    - E.g.: http://dl.ifip.org/db/conf/cnsm/cnsm2021/1570754857.pdf (gLBF)
    - Ideally requires a ‘timestamp’ packet header field (e.g.: 16 bit).
    - Ideally requires per-hop sequence of priorities (sequence of 4 bit values)
    - Different solutions from the past 30 years to review, which may want different parameters
- Stochastical bounded latency
  - draft-stein-srtsn – sequence of per-hop ‘deadlines’ (usec time values – 16..32 bit?)
- Other QoS – per-hop, per-flow stateless!
  - DSCP: BIER header (RFC8296) already allows MPLS with DSCP
    - Given how MPLS payload is most often IP, using DSCP instead of TC can make ops easier
  - Weighted bandwidth:
    - draft-stoica-diffserv-dps (bandwidth weight parameter)
  - More than 2 bits for ECN ? (See L4S – Issue not trickled into MPLS world much yet!)
  - Several more interesting per-hop, per-flow stateless QoS
    - Which now becomes more interesting:
    - Better / more flexible QoS hardware / programmability

Summary

- Would love to see one well working, proven Service Provider Class bounded latency for DetNet with MPLS that is useable NOW
  - draft-eckert-detnet-mpls-tc-tcqf-01
  - Please come to DetNet if you are interested!

- Would also be great to also start longer-term QoS header effort
  - IMHO, most important to see if we can get common MPLS (MPLS-DT) and IP solution
    - Does not need to be 100% same encoding, but same data model parameters/semantics
  - Should be easily extensible
  - Also need more QoS experts involved
    - Alas, very few QoS experts in IETF interested today beyond AQM for best-effort!

The End