Proposed evaluation template

For draft-ietf-detnet-scaling-requirements-03

NOT (YET) APPROVED BY ANYONE EXCEPT THE AUTEHOR

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

V0.0, 09/07/2023

What is being evaluated

- Mechanisms that support bounded latency across a DetNet domain
- This document proposes for requirements to be distinguised to where they apply:
  - “P”: P router / Hop-by-hop processing
  - “PE”: ingress PE router processing

PE functions

- On ingress to a DetNet domain, all flows need to be processed on a per-flow basis because received traffic flows need to be policed to comply with their admitted traffic envelope.
  - Complexity / scalability on PE node can therefore never be as high as on P node.
  - *Note: in simple ring topologies, every node is a P and PE node. If a ring has 20 nodes, then a node will typically be PE for only 1/20'th of the overall flows. So P scalability is even important in these topologies. Larger ring topologies will have ring nodes that are hubs for a larger number of spokes, and hence ring nodes are only P nodes.*
- Hop-by-hop forwarding mechanism may need specific ingress PE processing beyond policing. This needs to be evaluated.
  - Example: cycle mechanisms need per-flow timing of admitting packets of a flow into specific cycles. This is a specific version of a generic policer.
- draft-eckert-detnet-flow-interleaving makes the argument that policing and timing of flows on ingress PE is a generic function that DetNet is currently missing, but which if defined could be combined with all hop-by-hop mechanisms and act as the common additional PE function
  - This would be a simplified version of the TSN timed gates function (which in TSN is allowed to be used on every hop, but in "scaling" DetNet only on ingress PE
- In current sheet, only one requirement is identified as "only applicable to PE". Other requirements have not been identified to have specific different requirements for PE, so only P is indicated.

Enhancements to evaluation sheet

- New column to indicate whether requirement is per-hop (P), or ingress (PE).
- Split 3.1 into its four constituent requirements (3.1.1 - 3.1.4)
  - No evaluation for 3.1 overall, just the four individual ones
  - No evaluation for 3.1.4, because synchronous hop-by-hop such as *CQF can also provide support for asynchronous flows, for example by burs shaping as a domain ingress function. Has benefits over per-hop burst shaping as in rfc2211 or multi-priority burst-shaping as in Qcr/gLBF. Need to revisit requirement.
- 3.2.1 Added requirement for single-hop transmission jitter (NEW)
- 3.8(1) List explicit requirement for “support tight jitter (per-hop)”
  - as mentioned in requirements document as one possible option of traffic (e.g.: industrial control loops)

<Mechanism> evaluation (<notes>)

Sec.	P/PE	Requirements	Eval	Notes
3.1.1	PE	Async across TSN subdomains		changes from overall section 3 eval to individuals
3.1.2	P	Tolerate Clock Jitter/Wander		
3.1.3	P	No Full Time Sync required		
3.2	P	Large Single-hop Propagation Latency		
3.2.1	P	Support single-hop propagation jitter		New
3.3	P	Support Higher Link Speed		
3.4(1)	P	Scalable to Large Number of Flows		Details from prior reviews
3.4(2)	P	Tolerate High Utilization		Details from prior reviews
3.5	P	Link/Node failures, Topo Changes		
3.6	P	Prevent Flow Fluctuation from Disrupting Service		
3.7	P	Scalable to Large Number of Hops with Complex Topology		
3.8(1)	P	Support tight jitter/sync-control loops		New

Possible evaluation points

3.1.1	YES if hop-by-hop mechanism has defined ingress function, and/or assumed to be combined with some pre-existing ingress function. Describe or reference to ingress function in notes.
3.1.2	Suggest YES if permitted drift (short term variation of clock frequency) on a 100Gbps (or faster) link can be equal or less than 1 usec.
3.1.3	YES if MTIE (maximum time interval error) over arbitrary periods can be infinite. Or (simpler): if clock wander over 24 hours can be larger than 100ppm ~= 8 seconds.
3.1.4	Not evaluated, see prior discussion why.
3.2	YES if 1000km links can be supported (on every hop)
3.1.2	NEW: YES if jitter in link latency can be supported (length variation, L2 retransmission jitter on mobile/radio links,...)
3.3	YES if 100Gbps or faster links can be supported without loss of performance (e.g.: higher dead times) than for slower links.
3.4(1)	YES if mechanism does not require per-hop, per-flow state requiring per-packet lookup and read/write cycles.
3.4(2)	YES if 100Gbps or faster long latency links permit same close to 100% DetNet traffic
3.5	YES if Segment Routing and fast-reroute options with resource reservation can be supported
3.6	YES if mechanism does not have burst-accumulation across multiple network hops (no per-hop increase in flow burstyness) AND if mechanism allows to add/delete flows without impacting prior latency/jitter guarantees of pre-existing flows
3.7	YES if 3.6 is YES AND if flow interleaving can be applied to the mechanism across flows from different ingress/egress PE
3.8(1)	YES if end-to-end jitter is independent of number of hops.

Example: TCQF/CQSF evaluation **(comparison to CQF/ECQF/gIBF highlighted)**

Sec.	P/PE	Requirements	Eval	Notes
3.1.1	PE	Async across TSN subdomains	Yes	TCQF defines async ingres function.
3.1.2	P	Tolerate Clock Jitter/Wander	Yes	E.g.: with 4 cycles, max jitter/wander 1 cycle time. **Main benefit over (E)CQF** *ECQF can not do this: arrival time inaccuracy leads to wrong cycle assumption*
3.1.3	P	No Full Time Sync required	No/ “NA”	Same as CQF, ECQF, TAS. **Main benefit of gLBF (which has YES)** **Lower accuracy clock sync requirements than CQF/ECQF**
3.2	P	Large Single-hop Propagation Latency	Yes	**Main benefit over TAS, CQF**. Same as ECQF.
3.2.1	P	NEW: Support single-hop propagation jitter	Yes	**Benefit over ECQF , CQF, TAS** (radio, retransmissions, length deviation)
3.3	P	Support Higher Link Speed	Yes	200km/100Gbps proven, can scale well beyond that. **Not proven for ECQF**
3.4(1)	P	Scalable to Large Number of Flows	Yes	No per-hop, per-flow state, read/write memory access requirement
3.4(2)	P	Tolerate High Utilization	Yes	**Solves CQF issue, equal/better than ECQF**: No dead times.
3.5	P	Link/Node failures, Topo Changes	NA? / Yes	**CSQF**: Could support all Segment Routing re-route soutions (path, tunnel,...)
3.6	P	Prevent Flow Fluctuation from Disrupting Service	Yes	Like CQF/ECQF: No burst accumulation/jitter-increase due to per-hop cycle based reshapeper-hop reshaping.
3.7	P	Scalable to Large Number of Hops with Complex Topology	Good (TCQF) Best (CSQF)	Assuming flow interleaving on edge, like CQF/ECQF **CSQF more flexible** than TSQF, CQF/ECQF
3.8(1)	P	Support tight jitter/sync-control loops	Yes	Network size independent hop-by-hop/end-to-end jittter ~ O(0)