

Workgroup: DETNET

Internet-Draft:

draft-eckert-detnet-criteria-assessment-00

Published: 6 March 2023

Intended Status: Informational

Expires: 7 September 2023

Authors: T. Eckert

Futurewei Technologies USA

Criteria and Assessment for DetNet Services Forwarding Plane Methods

Abstract

This memo proposes methods to define, document and assess common criteria for the evaluation of forwarding plane mechanisms intended to be used within the DetNet WG as well as by implementers, operators and vendors that need to compare and select such mechanisms.

This document is not intended to become an RFC but does at this point in time solely intend to help the process of documentation, assessment and comparison. The goals of this memo may change in later revisions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 7 September 2023.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Goals](#)
- [3. Assessment Criteria](#)
 - [3.1. General](#)
 - [3.2. Service Characteristics](#)
 - [3.2.1. Latency](#)
 - [3.2.2. Jitter](#)
 - [3.3. Network Applicability](#)
 - [3.4. Node Clocking](#)
 - [3.5. Traffic Model](#)
 - [3.6. Per-hop Processing State](#)
 - [3.7. Calculus](#)
 - [3.8. Packetization](#)
- [4. Example Assessment: TCQF](#)
 - [4.1. General](#)
 - [4.1.1. Name of the Mechanism: Tagged Cyclic Queuing and Forwarding \(TCQF\)](#)
 - [4.2. Reference: I-D.eckert-detnet-tcqf](#)
 - [4.3. Service Characteristics](#)
 - [4.3.1. Latency guarantee: Bounded \(deterministic\)](#)
 - [4.3.2. Jitter: on-time \(\$2 \times\$ cycle-time T\)](#)
 - [4.4. Network Applicability](#)
 - [4.4.1. Intended for IP, IPv6 and MPLS networks](#)
 - [4.4.2. Intended for 100Gbps networks and faster](#)
 - [4.4.3. Latency requirements for links: arbitrary](#)
 - [4.4.4. Wireline Jitter: supported \(at the cost of higher per-hop latency\)](#)
 - [4.4.5. Radio Jitter: NO / TBD](#)
 - [4.5. Node Clocking](#)
 - [4.5.1. Asynchronous: No](#)
 - [4.5.2. MTIE impact on jitter and bounded latency](#)
 - [4.6. Traffic Model](#)
 - [4.6.1. Traffic Model Per-Hop, Per-Flow Parameters](#)
 - [4.7. Per-hop Processing State](#)
 - [4.7.1. \(per-hop, per-flow\) Stateful: No](#)
 - [4.8. Algorithm parameters and mechanisms](#)
 - [4.9. Calculus](#)
 - [4.9.1. Published per-hop calculus: Yes](#)
 - [4.9.2. Published linear per-hop, per-flow calculus: Yes](#)
 - [4.9.3. Hop-by-hop jitter bounds: on-time](#)

[4.10. Packetization](#)

[4.10.1. Per-flow compatibility with per-hop source-routing: Yes](#)

[4.10.2. End-to-end packet header requirements](#)

[4.10.3. Hop-by-hop packet header requirements](#)

[5. Security Considerations](#)

[6. IANA Considerations](#)

[7. Changelog](#)

[8. Informative References](#)

[Author's Address](#)

1. Introduction

[[I-D.ietf-detnet-scaling-requirements](#)] defines a range of requirements in support of scaling DetNet deployment along various parameters including the size of the network and the number of DetNet flows to be supported. Understanding how the variety of currently existing as well as new, proposed forwarding mechanisms do match these requirements does require to start defining quantifiable criteria and assess how each specific mechanism does meet the criteria. This document proposes how to do this.

2. Goals

The criteria should be descriptive so that operators can assess the feasibility and benefits of an assessed method without necessarily having to understand the exact behavior of the mechanism. This implies that some duplication of assessments may be appropriate such as using benefits and challenges as criteria, even if they are coupled because of the same underlying root cause.

The criteria should be descriptive so that vendors of network and controller plane equipment can assess the feasibility and challenges of mechanisms for implementation, especially with respect to on-chip cost as well as scale aspects - speed of network links, latency/jitter of links, total number of flows to be supported, and so on. These criteria tend to require more detail assessment as hopefully necessary for operators.

One of the meta goals of assessments is to find the minimum number of higher-level criteria sufficient for operators, but currently this is not possible. For example, the authors consider it to be impossible to specify agreeable criteria for an assessment such as "can be supported in 400Gbps ++ switches/routers without additional device cost". But more explicit detail criteria which do add up to such assessments can be specified.

3. Assessment Criteria

3.1. General

Criteria: Name (/version/instances)

Description: Summary of the methods benefits and goals.

Criteria: Reference(s)

3.2. Service Characteristics

3.2.1. Latency

Criteria: Latency guarantee: NA / Bounded or Heuristic

Explanation: If the method assessed supports deterministically bounded latency, then this criteria is "Bounded". Else it is "NA" or "Heuristic".

Example: [[I-D.stein-srtsn](#)] proposes a method resulting in heuristic latency guarantees. the maximum latency can not 100% be guaranteed by the proposed forwarding mechanisms, but only with a very high probability (depending on traffic characteristics).

3.2.2. Jitter

Criteria: Hop-by-hop jitter bounds: in-time / on-time / <other>

Explanation: This criteria assesses coarsely the jitter behavior of the method. An on-time method does not delay packets on a per-hop basis to reduce jitter of packets across the hop. In the absence of competing traffic, packets of a flow that do not exceed the flows permitted traffic model parameters are forwarded without any delay. On-time methods have the largest possible jitter, but the shortest possible delay in no-load situations. On-time methods do reduce per-hop jitter through buffering. They have the lowest "possible" jitter, but a latency (mostly) independent on load. If neither in-time no on-time sufficiently characterizes the jitter behavior, new "<other>" classification terms should be introduced. If easily possible, the explanation should provide some numeric description of the jitter behavior.

Discussion: This criteria specifically asks for the hop-by-hop behavior, because every on-time mechanism can be converted into an in-time mechanism by add-on mechanisms such as receiver endpoint playout buffers using sender generated timestamps to delay processing up to the known bounded latency time. But such ad-on mechanisms would introduce for example the need for end-to-end clock

synchronization and may hence negate benefits of the chosen method
- and should therefore be evaluated separately.

Example: In [\[UBS\]](#), hop-by-hop jitter is on-time, and minimum latency is likely the minimum latency possible for any algorithm with linear hop-by-hop calculus. In [\[CQF\]](#) and [\[I-D.eckert-detnet-tcqf\]](#), jitter is in the order of the cycle time T and hence minimal, and latency therefore close to the bounded latency.

3.3. Network Applicability

Criteria: Intended / applicable for IP / IPv6 / MPLS (subset)

Criteria: Intended for 100Gbps networks and faster (Yes/No)

Criteria: Requires new on-the-wire packet header fields (Yes/No/Optional)

Criteria: Latency requirements for links

Description: This assessment is to specify and describe the latency requirements for links. Some methods can only support links with tight latency limits, such as [\[CQF\]](#).

Criteria: Jitter requirements/impact for wireline links

Description: Link jitter on wireline links may occur because of packet loss and recovery by FEC (such as on ATM links), or link length variation, such as "hanging" links on poles (up to 30% between noon and midnight in observed cases). This criteria should describe the impact of such known wireline jitter impacts (and their feasibility) on the method.

Criteria: Jitter requirements/impact for radio links

Description: Radio links typically are subject to high, unpredictable loss and the use of link-layer retransmissions which cause higher latency for such retransmitted packets. While these behaviors do typically not impact the assessed mechanism itself, they can impact the necessary parameters and limit of applicability of the method for such radio links. This criteria is meant to describe this.

Discussion: The distinction between wireline and radio links is somewhat artificial in so far as that all jitter mitigation techniques are applicable across transmission media, but practically, this distinction helps to keep the wireline descriptions simpler, and given how DetNet services over radio links is largely less well understood, this distinction matches also the difference in likelihood of initial adoption/relevance.

3.4. Node Clocking

Criteria: Asynchronous: Yes / No (No \approx Asynchronous)

Explanation: A forwarding mechanism can claim to be asynchronous if it can support unlimited "Maximum Time Interval Error" (MTIE) towards neighboring nodes. In other words, the clocks between adjacent nodes may drift/wander over time unlimited. If this is not the case, then the method should be called synchronous for this criteria.

TSN [[ATS](#)] is an example of an asynchronous mechanism. TSN [[CQF](#)] is an example of a synchronous mechanism.

Criteria: MTIE impact on jitter and bounded latency

Explanation: The MTIE of adjacent clock has a quantifiable impact on the achievable per-hop bounded latency and jitter. This criteria is intended to provide a characterization of this impact. At this point in time it is unclear what the most easily feasible absolute metric(s) are to perform this characterization, so this memo proposes an informal description, including absolute comparisons to the impact on pre-existing reference methods. Hopefully it will be possible to derive at more well defined metrics in later versions of this document.

Note: The MTIE will not be the only parameter of importance here, but also the frequency of clock variation within the MTI. The appropriate metric for this is also TBD.

For easier assessment, it can be helpful to provide minimum MTIE/drift-speed requirements for a 1 Gbps network vs. a 100 Gbps network to make it easy to compare numbers across mechanisms for these real-world numbers.

3.5. Traffic Model

Criteria: Traffic Model Per-Hop, Per-Flow Parameters

Explanation: Explanation for the parameters of the traffic model of the packets of one DetNet flow when being processed of the assessed mechanism. The mechanism may support more than one set of traffic model parameters.

Example: The [[UBS](#)] mechanism, as used in for example TSN [[ATS](#)] defines two possible traffic models: The Token Bucket model (TBE - Token Bucket Emulation), and the simpler Length Rate Quotient (LRQ) model, which could be ignored for assessment because it provides only a subset of functionality of TBE. For TBE, a flow (i) has two

parameters, a "leak rate" $r(i)$ (bits/sec) and a burstyness $b(i)$ (bits).

Example 2: In [\[RFC2210\]](#), the traffic model incurs a richer set of parameters compared to [\[UBS\]](#) (if definition here is desired: TBD), resulting in a higher processing complexity, but also more fine-grained options to differentiate the per-hop latency of flows compared to UBS.

3.6. Per-hop Processing State

Criteria: (per-hop, per-flow) Stateful: Yes / No (No \Rightarrow stateless)

Explanation: Assume a DetNet service with ingres/egres nodes called $PE(i)$, $i=1..N$. A forwarding plane flow $flow(i,j)$ is the totality of packets from one $PE(i)$ to one $PE(j)$ where the DetNet services, especially bounded latency and/or jitter will experience the same treatment. Such a flow can be a single DetNet flow, or it can be an aggregate of multiple DetNet flows that will be treated the same. Consider two or more forwarding plane flows $flow(i1,j)$ and $flow(i2,j)$ that arrive from different input interfaces onto the same router on the path towards $PE(j)$. If this router can process these packets without having to distinguish $i1$ and $i2$, then this criteria calls this method stateless, else stateful.

TSN [\[ATS\]](#) is an example of a stateful mechanism, TSN [\[CQE\]](#) is an example of a stateless mechanism.

Criteria: algorithm parameters and mechanisms

Explanation: This criteria describes the on-node state and processing mechanisms required by the method for the processing of packets.

The so-called "lookup-state" is the set of keys from a packet required to look up the so-called "on-node per-flow processing state" state required to process a packet. In addition, packets may also carry other "per-packet processing parameters"

In a stateful mechanism, the lookup state consists of the keys of a DetNet flow, depending on the DetNets forwarding plane (IP 5/6 tuple or MPLS label stack elements) or from the underlying L2 solution used to implement the mechanism, such as ethernet header fields for IEEE mechanisms.

Discussion: The [\[UBS\]](#) algorithm as used in TSN methods relies on ethernet header lookup-state, but could equally rely on DetNet flow parameters as lookup state. Hence the lookup-state for stateful methods is not an important comparison criteria and is therefore defined here explicitly to allow focussing on the processing state.

Example 1: In [\[UBS\]](#), the processing state for a flow(i) consists of the traffic models per-flow parameters $r(i)$ and $b(i)$ as well as the processing parameters $level(i)$, $timestamp(i)$ and a priority $p(i)$. $p(i)$ can be different on every hop and impacts the per-hop latency experienced by packets of flow(i).

Processing a packet consists of performing calculations over those five parameters as well as updating $level(i)$ and $timestamp(i)$ in time before another packet of flow(i) arrives, so that that following packets processing can take the updated parameters into account. Scheduling of the packet is determined by a calculation of these parameters and the presence of competing packets. The mechanisms proposed to support the scheduling of packets are a combination of "interleaved regulators" and "strict priority queueing".

Example 2: In [\[CQF\]](#), there is no (per-flow) lookup state. Instead, the arrival timestamp of packets is used to determine the scheduling of the packet into the appropriate cyclic queue.

Example 3: in [\[I-D.eckert-detnet-tcwf\]](#), there is no (per-flow) lookup state. Instead, a per-packet processing parameter called the cycle identifier 'c' is used to indicate the cycle into which the packet is to be enqueued.

3.7. Calculus

Criteria: Published per-hop calculus: Yes/No

Explanation: For this criteria to be assessed yes, there needs to be a published description for how to calculate in less than NP complete complexity whether a given set of flows with a given set of traffic model parameters and a required set of latency and (if offered) jitter requirement can fit onto an individual interface/link with a given set of parameters, speed, latency, buffers and resulting in a specific per-hop behavior: latency, jitter. If sufficiently easily feasible, the explanation of the assessed criteria should contain a summary of the calculus.

Description: It is hopefully sufficient that all methods of interest will allow to break down the resource consumption vs. achieved latency/jitter on a per-hop basis, so that the end-to-end characteristics can be calculated in linear time across the hops of a path.

Criteria: Published linear per-hop, per-flow calculus: Yes/No

Explanation: For this criteria to be assessed as yes, the prior requirement is extended by requiring for a calculus to add/delete an individual flow from the set of current flows and to require for

this calculus to have no more than linear complexity with the number of flows utilizing the link.

Description: The prior calculus criteria is assumed to be sufficient to permit admission of a set of flows during longer-term planning/provisioning, by allowing for the calculus to have significant complexity. Especially when the network has significantly more resources than required for DetNet services, it may only be necessary to calculate the overall DetNet resources, but not to quickly and dynamically reserve resources for new flows or release them. This criteria instead is written to support exactly this. Whether admission happens in an offline controller-plane, or on-path - both options are known to be easily supportable if this criteria can be met by a method.

Criteria: Hop-by-hop jitter bounds: in-time / on-time / <other>

Explanation: This criteria assesses coarsely the jitter behavior of the method. An on-time method does not delay packets on a per-hop basis to reduce jitter of packets across the hop. In the absence of competing traffic, packets of a flow that do not exceed the flows permitted traffic model parameters are forwarded without any delay. On-time methods have the largest possible jitter, but the shortest possible delay in no-load situations. On-time methods do reduce per-hop jitter through buffering. They have the lowest "possible" jitter, but a latency (mostly) independent on load. If neither in-time no on-time sufficiently characterizes the jitter behavior, new "<other>" classification terms should be introduced. If easily possible, the explanation should provide some numeric description of the jitter behavior.

Discussion: This criteria specifically asks for the hop-by-hop behavior, because every on-time mechanism can be converted into an in-time mechanism by add-on mechanisms such as receiver endpoint playout buffers using sender generated timestamps to delay processing up to the known bounded latency time. But such ad-on mechanisms would introduce for example the need for end-to-end clock synchronization and may hence negate benefits of the choosen method - and should therefore the evaluated separately.

Example: In [[UBS](#)], hop-by-hop jitter is on-time, and minimum latency is likely the minimum latency possible for any algorithm with linear hop-by-hop calculus. In [[CQF](#)] and [[I-D.eckert-detnet-tcqf](#)], jitter is in the order of the cycle time T and hence minimal, and latency therefore close to the bounded latency.

3.8. Packetization

Criteria: Per-flow compatibility with per-hop source-routing: Yes / No (SR-MPLS, SRv6, BIER-TE, ...)

Explanation: In a network deployment using source-routing, path selection is done not hop-by-hop through a distributed routing protocol, but decided by a packet header inserted by the ingress router / LSR into a domain. This is meant to reduce overall operations and signaling complexity and eliminating the need to update per-hop routing / forwarding state. Compatibility with this for DetNet services implies that the assessed mechanism does also provide the ability to perform the DetNet service (especially per-hop latency/jitter guarantees) purely through in-packet headers as opposed to per-hop, per-flow state in support of these services.

Discussion: This document is only aware of mechanisms meeting this criteria if they also are meeting the stateless criteria from earlier in this document. Nevertheless, this criteria is listed separate to not make the assumption that this is always necessary. In addition, stateless mechanisms for DetNet services may also be desirable in non-stateless routing deployments, such as especially when using stateless multicast with non source-routing for unicast. In such deployments, the compatibility with per-hop source-routing is irrelevant to the operator.

Criteria: End-to-end packet header requirements ($\sim N$ bits/bytes)

Description: This criteria assesses the end-to-end packet header requirements to support the assessed mechanism. This may depend on other packet encapsulation aspects such as whether IP or MPLS are being used, in this case multiple assessments can be provided, one for each context. If this is not applicable, then "none" should be used. This criteria should not include assessment of packet header requirements that are per-hop, including any possible end-to-end overhead for a per-hop header.

Example: In most stateful mechanisms such as [UBS](#), this criteria is "none", because the only packet header requirement is the flow-key headers, e.g.: DetNet flow identification. In a method such as TQF [\[I-D.eckert-detnet-tcqf\]](#) it is a small number such as 3 bits.

Criteria: Hop-by-hop packet header requirements ($M + N/\text{per-hop}$ bits/bytes)

Description: This criteria assesses the packet header requirements if the mechanism needs or supports per-hop in-packet header parameters. M should be the static overhead independent of the number of hops, N the per-hop cost. Descriptive text should be used as necessary to explain/refine the assessment.

Example: A mechanism such as [[I-D.stein-srtsn](#)] proposes per-hop latency deadline values in the packet header. Thus the per-hop header requirement is the number of bits required to represent these deadlines. If possible instances of such a mechanism might need different size header fields depending on size or speed of the network, it could be assessed using example 1Gbps and 100Gbps networks.

4. Example Assessment: TCQF

4.1. General

4.1.1. Name of the Mechanism: Tagged Cyclic Queuing and Forwarding (TCQF)

TCQF is a derivation of [[CQF](#)]. In CQF, the arrival time of a packet determines the cycle in which it is forwarded. This limits applicability of CQF to links/networks with short propagation latencies. TCQF replaces this meth with an in-packet cycle indicator and the use of 3 or more cycles to support arbitrary link latencies and link jitter. Due to the use of only few bits to indicate the cycle in a packet header TCQF could be implemented for IP and MPLS without change of packet headers by using DSCP or TC fields.

4.2. Reference: [*I-D.eckert-detnet-tcwf*]

4.3. Service Characteristics

4.3.1. Latency guarantee: Bounded (deterministic)

4.3.2. Jitter: on-time ($2 \times$ cycle-time T)

End-to-end jitter with TCQF is at most 2 times the cycle-time T . In a wireline 100 Gbps network, this cycle time could for example be 20...100 usec, resulting in a maximum end-to-end jitter of 40...200 usec.

4.4. Network Applicability

4.4.1. Intended for IP, IPv6 and MPLS networks

As mentioned in General section.

4.4.2. Intended for 100Gbps networks and faster

The faster the network, the shorter the cycle time can be, and hence the per-hop latency introduced by cyle forwarding. This method may not be a useful enhancement in those slower 1..10 Gbps networks where [[CQF](#)] can work well.

4.4.3. Latency requirements for links: arbitrary

4.4.4. Wireline Jitter: supported (at the cost of higher per-hop latency)

When links may incur jitter in propagation latency, the number N of cycles needs to be configured so that $\text{jitter} < T * (N - 3)$. In other words, in addition to the 3 cycles required to support arbitrary latencies, an additional $(N - 3)$ cycles are required so that the cycle time $T * (N - 3)$ exceeds the jitter: These additional cycles serve as a type of "per-hop" playout buffer converting the jitter into latency - and maintaining the end-to-end maximum Jitter of $2 * T$.

4.4.5. Radio Jitter: NO / TBD

Considerations of Jitter beyond those explained for Wireline are TBD.

4.5. Node Clocking

4.5.1. Asynchronous: No

4.5.2. MTIE impact on jitter and bounded latency

TCQF requires a known upper limit of the MTIE for cycles to not run out of synchronization. The impact of MTIE on the configuration of MTIE is like the impact of link jitter: Higher MTIE can be compensated for with larger cycle time / number of cycles. This allows to reduce MTIE requirements compared to [CQF] by a factor of 20 or more, thereby reducing the implementation cost and/or allowing to support faster links without increase of clock synchronization accuracy (which could otherwise be prohibitive at speeds ≥ 100 Gbps)..

4.6. Traffic Model

4.6.1. Traffic Model Per-Hop, Per-Flow Parameters

The per-hop traffic model for an end-to-end flow is the maximum number of bits in a cycle time. On ingress to a TCQF path, flows with other traffic models (such as Token Bucked models as in [UBS]) have to be shaped accordingly.

4.7. Per-hop Processing State

4.7.1. (per-hop, per-flow) Stateful: No

TCQF nodes only need to buffer packets in per-egress cycle buffers independent of flows. Therefore an arbitrary number of ingress to egress flows especially from different ingress can share the same

buffers. On an ingres-node, per-flow state may be necessary if the aforementioned shaping of flows is required. If a trusted application sender supports TCQF directly, no network node needs to have any per-flow state.

4.8. Algorithm parameters and mechanisms

TCQF only requires a mapping table on each egress interface:

(input-interface, ingres-cycle) -> (egres-cycle / egress-cycle-buffer)

4.9. Calculus

4.9.1. Published per-hop calculus: Yes

As per TCQF reference. See next criteria for explanation.

4.9.2. Published linear per-hop, per-flow calculus: Yes

As per TCQF reference

Bandwidth admission calculus is pretty much simply keeping track of the number of bits required for each flow on every hops cycle buffers in e.g.: an offline admission controller.

Latency calculus is simply adding up the per-hop cycle delay based on the number of cycles configured for each hop.

4.9.3. Hop-by-hop jitter bounds: on-time

The per-hop jitter as observed between enqueueing from the sending hop cycle queue to serving the next-hop cycle queue is as the end-to-end jitter $2 * T$. Note that this may be lower than the jitter of propagation latency across the traversed link, e.g.: the mechanism is compensating for that jitter as explained earlier.

4.10. Packetization

4.10.1. Per-flow compatibility with per-hop source-routing: Yes

TCQF can support SR-MPLS, SRv6, BIER and BIER-TE by use of the DSCP and/or TOS fields of the IP/IPv6 or MPLS header. This is explained in the TCQF reference.

Note that for MPLS, the number of possible values is less than 3 bits, so the amount of MTIE or link jitter than can be compensated will be limited

4.10.2. End-to-end packet header requirements

Basic TCQF can operate with three header values (less than 2 bits in a header). Depending on number of cycles needs 3 bits or more may be beneficial.

4.10.3. Hop-by-hop packet header requirements

TCQF does not propose any hop-by-hop headers in addition to the per-hop rewritten end-to-end header carrying the cycle identifier.

5. Security Considerations

TBD.

6. IANA Considerations

This document has no IANA considerations.

7. Changelog

[RFC-editor: please remove]

Initial draft name: draft-eckert-detnet-criteria-assessment-00

8. Informative References

- [ATS] Specht, J., "P802.1Qcr - Bridges and Bridged Networks Amendment: Asynchronous Traffic Shaping", IEEE , 9 July 2020, <<https://1.ieee802.org/tsn/802-1qcr/>>.
- [CQF] IEEE Time-Sensitive Networking (TSN) Task Group., "IEEE Std 802.1Qch-2017: IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks – Amendment 29: Cyclic Queuing and Forwarding", 2017.
- [I-D.eckert-detnet-tcqf] Eckert, T. T., Bryant, S., Malis, A. G., and G. Li, "Deterministic Networking (DetNet) Data Plane - Tagged Cyclic Queuing and Forwarding (TCQF) for bounded latency with low jitter in large scale DetNets", Work in Progress, Internet-Draft, draft-eckert-detnet-tcqf-01, 6 November 2022, <<https://datatracker.ietf.org/doc/html/draft-eckert-detnet-tcqf-01>>.
- [I-D.ietf-detnet-scaling-requirements] Liu, P., Li, Y., Eckert, T. T., Xiong, Q., Ryoo, J., zhushiyin, and X. Geng, "Requirements for Scaling Deterministic Networks", Work in Progress, Internet-Draft, draft-ietf-detnet-scaling-requirements-01, 1 March

2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-detnet-scaling-requirements-01>>.

[I-D.stein-srtsn] Stein, Y. J., "Segment Routed Time Sensitive Networking", Work in Progress, Internet-Draft, draft-stein-srtsn-01, 29 August 2021, <<https://datatracker.ietf.org/doc/html/draft-stein-srtsn-01>>.

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, DOI 10.17487/RFC2210, September 1997, <<https://www.rfc-editor.org/info/rfc2210>>.

[UBS] Specht, J. and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks", IEEE 28th Euromicro Conference on Real-Time Systems (ECRTS), 2016.

Author's Address

Toerless Eckert
Futurewei Technologies USA
2220 Central Expressway
Santa Clara, CA 95050
United States of America

Email: tte@cs.fau.de