Bounded latency calculating method, using network calculus

www.huawei.com

Jiayi Zhang, Tongtong Wang, Norman Finn

January, 2019

IEEE 802.1 TSN Interim, Hiroshima

HUAWEI

Background and Motivation

- Bounded latency is a key factor in TSN techniques, required by many use cases, such as 5G uRLLC, automotive and industrial network.
- Network Calculus is a modeling theory for deterministic network, to compute bounds on queuing delays, buffers, burstiness of flows etc. [1]
  - Network calculus introduces arrival curve, service curve.
  - Network calculus has application in avionic Full duplex Ethernet (AFDX), as a latency evaluation approach, in the design and certification of Airbus A380 AFDX backbone.
  - IETF DetNet bounded-latency draft leverage network calculus. [2]
- TSN profiles may also take advantage of network calculus, to help calculate E2E latency and buffer bound.

[1] http://www.ieee802.org/1/files/public/docs2018/new-leboudec-network-calculus-for-tsn-0118-v04.pdf
[2] https://tools.ietf.org/html/draft-finn-detnet-bounded-latency-02

Network Calculus Basics

- Modeling and analyzing (for TSN forwarding)

TSN Queuing and forwarding | Arrival traffic model

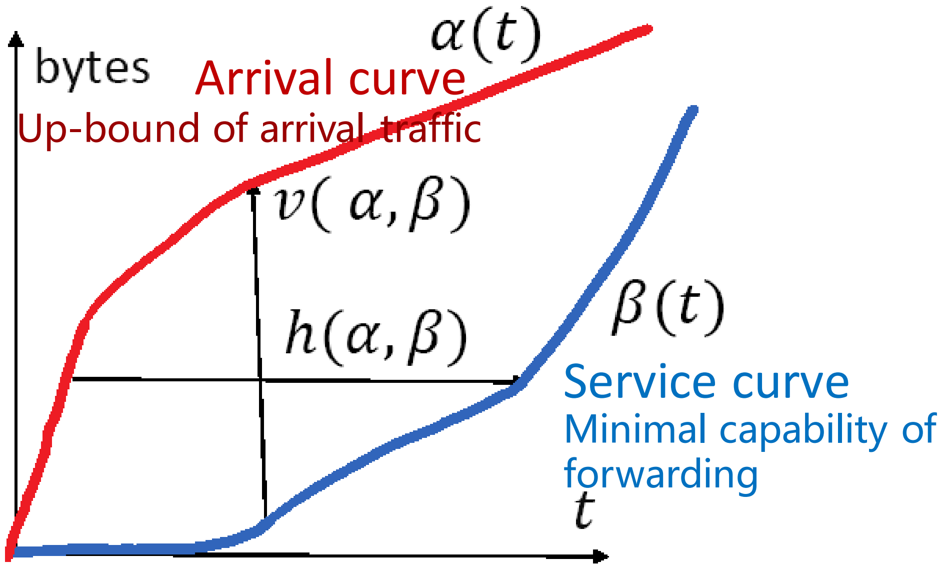
↓

Service curve / Arrival Curve

↓

Maximum horizontal distance → Delay bound (h)

Maximum vertical distance → Buffer bound (v)



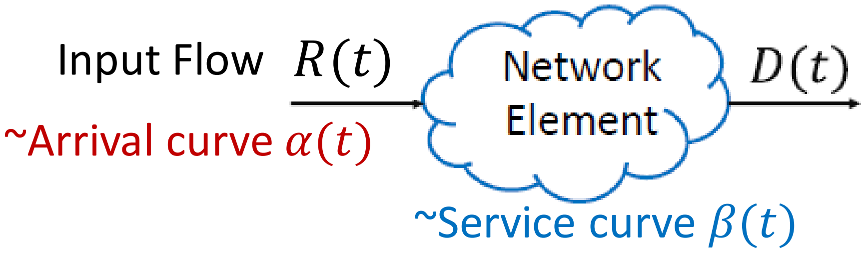
Data flow Input: $R(t)$, Output $R^*(t)$

Arrival curve $\alpha(t)$,

$$\text{s.t.}\quad R(t) - R(s) \le \alpha(t-s),\ \forall\, 0 \le s \le t$$

Service curve $\beta(t)$,

$$\text{s.t.}\quad R^*(t) \ge (R \otimes \beta)(t) = \inf_{t \ge t_0} \{R(t_0) + \beta(t - t_0)\}$$

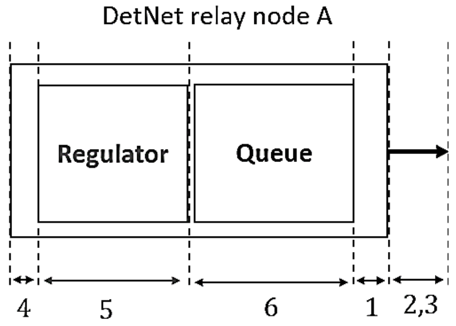


[1] http://www.ieee802.org/1/files/public/docs2018/new-leboudec-network-calculus-for-tsn-0118-v04.pdf

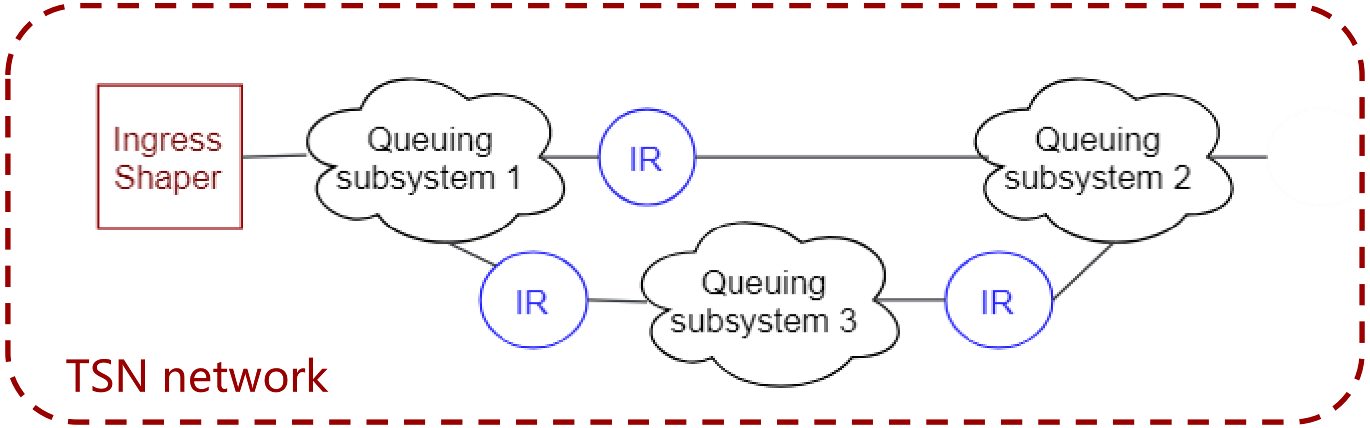
Network Latency Model

- IETF DetNet defines network timing model including different sources of delay, among which TSN mainly focuses on queuing delay.
- To calculate queuing delay, flow burstiness and aggregation causes burstiness cascade, resulting in TSN worst-case latency estimation pessimistic.
- To improve E2E latency bound, TSN may need to regulate the traffic
  - Ingress shaper at ingress node, to regulate flow burstiness;
  - Interleaved regulator (IR) at egress port per-hop (except the last hop), to avoid aggregation burstiness.

1) Output delay
2) Link delay
3) Preemption delay
4) Processing delay
5) Regulation delay
6) Queuing delay



https://tools.ietf.org/html/draft-finn-detnet-bounded-latency-02



* Queuing subsystem: queuing and forwarding method defined in IEEE 802.1 Qav, Qcr, Qbv, Qch…

Ingress Shaper

- The current method of specifying the bandwidth of a stream (maximum number of packets of a maximum size over an interval) is perfect for testing tools.
- But, it is a poor match to any of the queuing methods we have; gross overprovisioning is required. This is because unavoidable jitter causes the worst-case behavior to be significantly worse than the long-term worst-case.
- If one has a ‘maximum long-term worst-case bandwidth requirement’ for a stream, in addition to the current parameters, then an ingress shaper can be used, and very little, if any, overprovisioning is required.

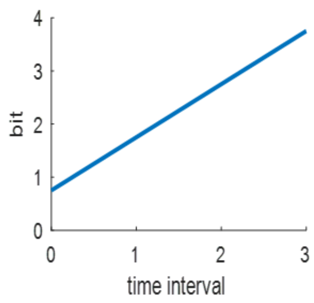
Ingress Shaper

Motivation: reshape the traffic before a flow enters TSN domain, in order to

1) Get precise traffic engineering (Users may have rough TSPEC, or BE, not fit for bridges' queuing methods)
2) Unify different TSPEC (Users may have different TSPEC)
3) In case of misbehaved talker

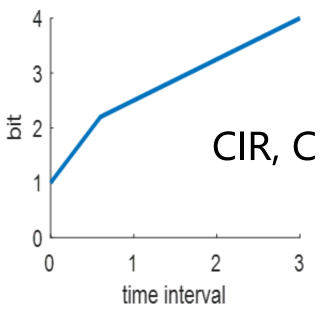
Possible Methods

- Single Leaky bucket



burst size (**b**)
rate (**r**)

- Double Leaky bucket



CIR, CBS, PIR, PBS,

Ingress shaping → Network calculus Arrival Curve

- Network calculus proved that, greedy shaping does not impact worst-case latency bound

TSN Queuing Subsystem

TSN Queuing and forwarding can be modeled as Rate-latency service curve.

1 - Strict Priority

- l_i^{max} – maximum packet length for priority i, R – link bandwidth

Higher — of i — Low

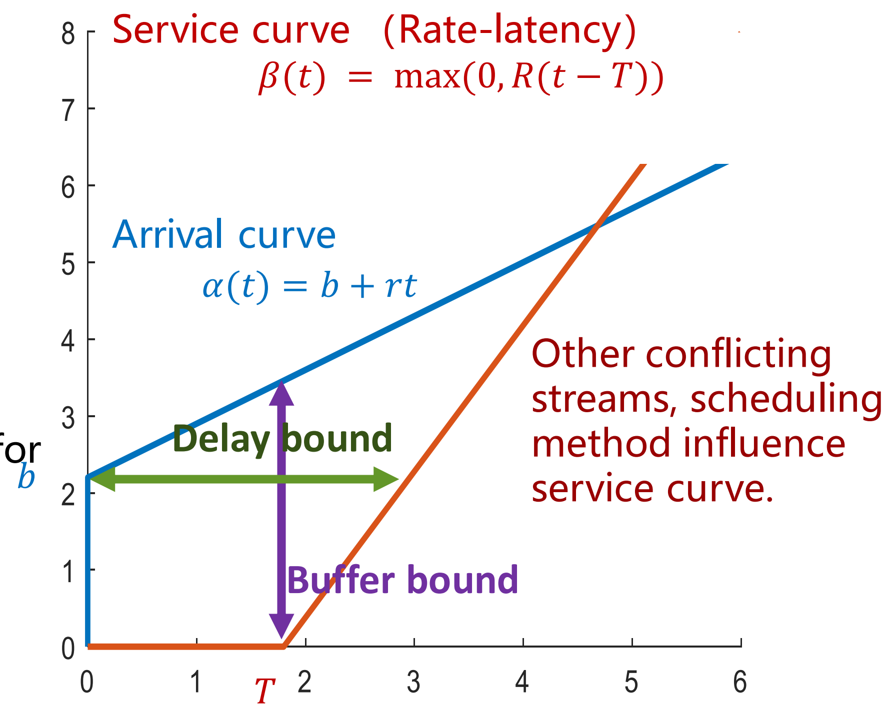
$$T_i^{max} = \frac{\Sigma_{k=1}^{j-1} l_k^{max} + \Sigma_{k=j}^{i-1} l_k^{max} + \max\{l_{LC}^{max}, l_{i+1}^{ma}}{R - \Sigma_{k=1}^{j-1} r_k} \qquad R_i = R - \sum_{j>i} r_j$$

Interfering Traffic

2 – Credit Based Shaping (CBS) *

- r^c, b^c-Control data traffic rate & burst , L^x – maximum packet length for class $x, x \in \{A, B, BE\}$. I^A - idle slope, S^A- send slope, determined by reservation.

$$T^A = \frac{1}{R - r^c}\left(L^A + b^c + \frac{r^c \bar{L}}{R}\right), \qquad R^A = \frac{I^A(R - r^c)}{I^A - S^A}$$



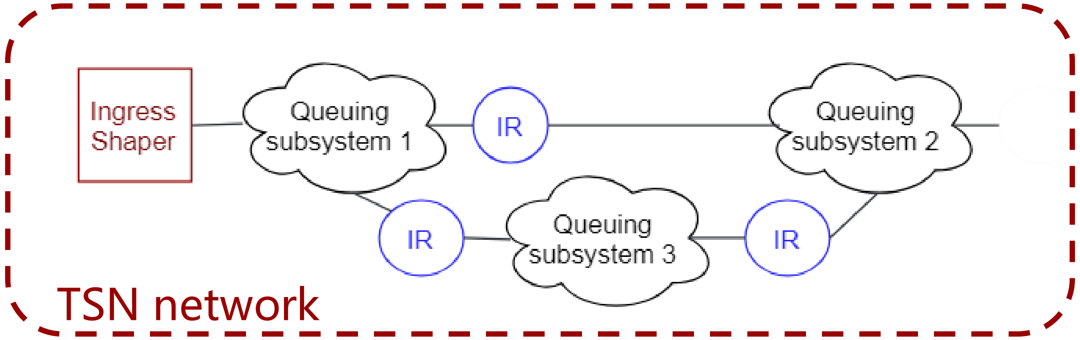
* https://arxiv.org/abs/1804.10608

Queuing subsystem → Service curve

- With arrival curve, we can calculate the per-hop delay up-bound: $T_i + b/R_i$ (Maximum horizontal distance)

HUAWEI

Interleave Regulator (IR)



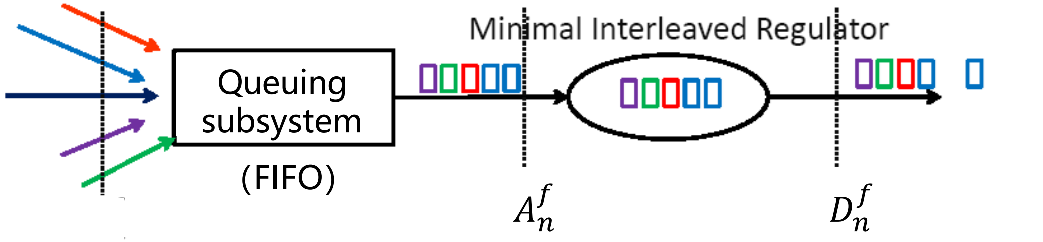
Motivation

- Avoid burstiness cascade caused by aggregation, regulate the flow at every hop (except for the last hop)

Method

- Aggregate packets of all flow in a FIFO queue
- Per-flow state is required. Each packet is assigned with a time for transmission D_n.
- Only head of the queue will be checked, whether it is allowed to transmit. Other packets should wait until they become the head of queue.
- Per-flow shaping with per-class queuing

Network Calculus proved that IR does not increase worst-case E2E latency. **

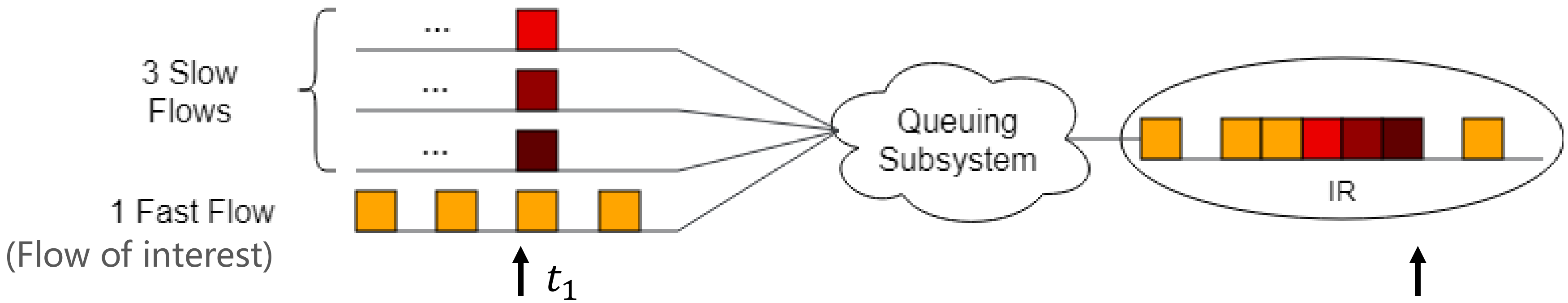


$$D_n = \max\left\{A_n, D_{n-1}, \Pi^{F_n}\left(D^{F_n}, L^{F_n}\right)_{I(n)}\right\}$$

Eligibility Time of packet at head of queue

* Johannes Specht, Soheil Samii, Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks.
** Le Boudec, Jean-Yves, "A Theory of Traffic Regulators for Deterministic Networks with Application to Interleaved Regulators." *arXiv preprint arXiv:1801.08477* (2018).

Example of IR

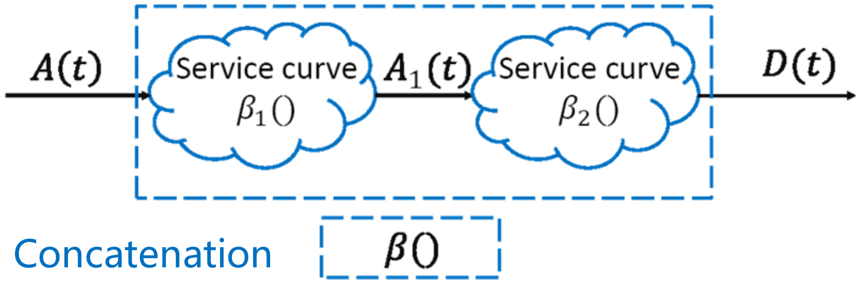


- All flows have the same priority, when packets arrive at the same time, at t_1, for the fast flow (Flow of interest), worst-case latency happens when all the packets of other flows comes ahead of me. Packets of the fast flow are delayed according to interleaved regulator.
  - Network calculus considers this worst-case.
- If we want to protect the fast flow, and to reduce the delay jitter, other scheduling methods may be designed, such as giving the fast flow a higher priority.
  - Network calculus can be used to calculate the corresponding latency bound.

End-to-end Latency Calculation

E2E service curve is concatenation of per-hop service curves

$$\beta(t) = \inf_{s \geq 0}\{\beta_1(s) + \beta_2(t-s)\}$$



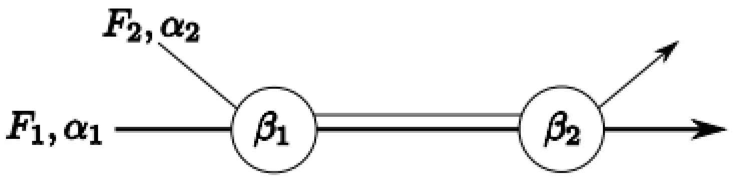
[1] http://www.ieee802.org/1/files/public/docs2018/new-leboudec-network-calculus-for-tsn-0118-v04.pdf

E2E Latency bound

- Maximum horizontal distance between arrival curve and concatenated service curve.

Improve the latency bound

- **Pay burst only once (PBOO):** The worst-case burstiness cannot happen at every hop, since the flow is shaped by node.
- **Pay multiplexing only once (PMOO):** Apply concatenation on a group of conflicting flows with the flow of interest, the worst-case burstiness caused by multiplexing happens only once.
- Introducing Linear Program



J. B. Schmitt, F. A. Zdarsky and I. Martinovic, "Improving Performance Bounds in Feed-Forward Networks by Paying Multiplexing Only Once," *14th GI/ITG Conference - Measurement, Modelling and Evalutation of Computer and Communication Systems*, Dortmund, Germany, 2008, pp. 1-15.

HUAWEI

Comparison of Shaping

Per-flow shaping	Per-class queuing, with per-flow state	Per-class shaping
• **Ingress Shaper** • Constrain the burstiness of a flow, at the ingress node. • Per-flow queuing	• **Interleaved Regulator** • With per-flow states, multiple flows are shaped individually within one class queue. • Per-class queuing.	• **Credit Based Shaper (CBS)** • Flows in one class are shaped all together. • Per-class queuing

Scheduling Granularity： High → Low

Complexity： High → Low

All the three shaping methods may take advantage of PBOO， PMOO, and other method to improve the tightness of latency bound.

Summary and Next Step

- We introduced two components in TSN, to help regulate traffic and ensure bounded latency
  - Ingress shaping, at edge node
  - Interleaved regulator, at every node except for the last node
- Calculate E2E latency by using network calculus
- This method may be useful in many TSN profiles, such as industrial, automotive, service provider network.
- We need to investigate the time-aware queuing techniques
  - Qbv, CQF, etc.
  - Timing model may be simple, in coordinated network

The Goal of This Work

- The authors intend to pursue this line of inquiry outside the regular IEEE 802.1 meetings.
- We want to provide the reader of IEEE Std 802.1Q with the means to calculate the end-to-end latency for at least:
  - CQF + Ingress shaper
  - CBS (no flow sharing in one queue) + Ingress shaper
  - ATS (hopefully, this is coming from Johannes Specht) + Ingress shaper + IR
- If the work progresses satisfactorily, the TSN TG may want to include this or similar work in 802.1Q.

Thanks

www.huawei.com