On non-FIFO Queues

Johannes Specht

2021-07-16

Introduction

Background

- Norman Finn proposed one or more PAR(s) for the following (https://www.ieee802.org/1/files/public/docs2021/new-finn-pulsed-queuing-0121-v02.pdf):
  1. Multi-CQF (more than 2 alternating CQF cycles)
  2. Paternoster (introduced by Mick Seaman)
  3. Bin rotation scheme (“Pulsed Queues”) to realize the former
  4. Bundling (a.k.a. flow aggregation)

Assumption

- All aforementioned proposals are intended for “*shaping for bounded latency*”
- Latency bounds shall be easy to compute and tight

Goals of this slide set

- Symmetries with ATS
- Some technical cross-checking
- Author’s thoughts/proposals/recommendations

Queues, Bins and Implementations

Some Insights

Insights

ClockOffsetVariationMax = ClockOffsetMax – ClockOffsetMin

NOTE 1—ClockOffsetMin and ClockOffsetMax capture implementation specific properties such as the resolution of the associated clocks, associated rounding errors, constant offsets between clocks, Bridge-internal synchronization inaccuracies in presence of different underlying oscillators, and similar.

A pair of a scheduler clock instance and a transmission selection clock instance has an implementation specific nominal rate, and a maximal absolute deviation from this nominal rate during operation, as characterized by the ClockRateDeviationMax parameter (12.31.8.4).

NOTE 2—ClockRateDeviationMax captures implementation specific properties such as oscillator rate deviation, numeric resolution for the operations specified in 8.6.11.3, and similar.

NOTE 3—ATS scheduler clocks and transmission selection clocks provide a model to express different sources of delay, delay variation, and inaccuracy. It is not required to implement different multiple physical oscillators/clocks (i.e., ATS scheduler clocks and transmission selection clocks can actually be the same physical clock or can be generated from the same oscillator), but the model captures the properties of implementations with and without different physical oscillators/clocks in a unified manner.

!

!

!

Source: 8.6.11.2 of IEEE Std 802.1Qcr-2020

A good Idea to think about

- Avoiding the “everything is a FIFO” paradigm provides new options
- ATS, as Standardized in IEEE Std 802.1Qcr-2020, exploited this:
  - Frames are sent in order of associated internal eligibility times
  - Eligibility times can “jitter” internally, modelled in the Standard by two internal clocks

  → **The jitter band can include the width (duration) of a <u>pulsed queue bin</u>** ☺

More Insights

"Pulsed Queues" are common

- 1969, E. G. Ulrich:
  Time-Sequenced Logical Simulation Based on Circuit Delay and Selective Tracing of Active Network Paths ["Δt-Loop"]
- 1987, G. Varghese and T. Lauck*:*
  Hashed and Hierarchical Timing Wheels: Data Structures for the Efficient Implementation of a Timer Facility
- 1988, R. Brown:
  Calendar Queues: A Fast O(1) Priority Queue Implementation for the Simulation Event Set Problem
- 2021, N. Finn:
  Towards a PAR (or PARs) for Pulsed Queues

Alternatives: "Big boxes" may use other implementations than "small boxes"

- **Timing wheels**, or whatever we call this (see above)
- **A few FIFOs**, for a few ports (interleaved shaping), with head-of-line frame eligibility time comparison
- **Heaps**, each node containing a frame with eligibility time + some extra for in-order delivery
- **Combinations** (e.g., heaps, each node pointing to a FIFO)
- …

→ Choosing the right implementation depends on the (ASIC) design under consideration!

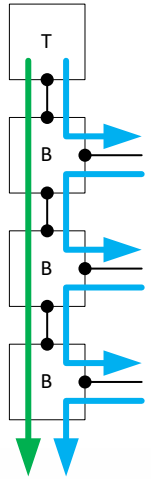
Syntonized CQF

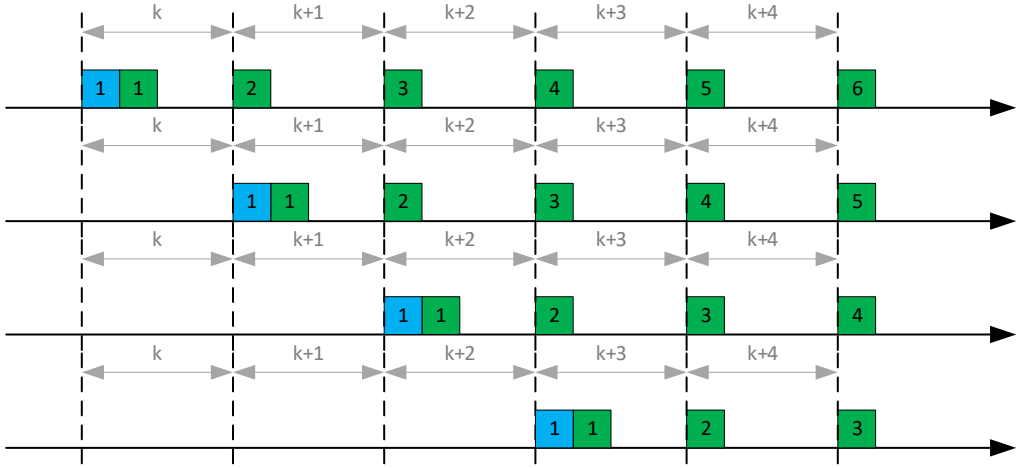
https://www.ieee802.org/1/files/public/docs2021/new-finn-pulsed-queuing-0121-v02.pdf, slide 14, 4th bullet

Synchronized CQF

Background

- Standardized by P802.1Qch
- Uses (fully) synchronized cycles in all Bridges, potentially with a controlled phase-shift (higher link delays)
- Frames received in a cycle *k* are transmitted in cycle *k+x* (*x≥1*)

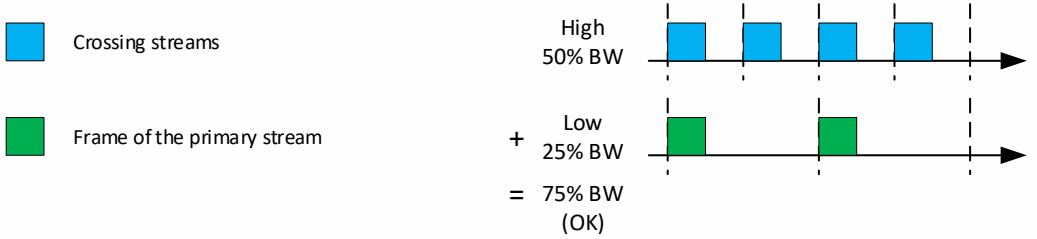




Trivial Assumptions

(more just makes the picture bigger)

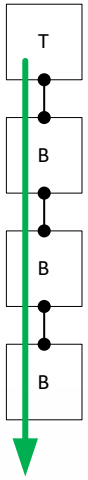
- 2 Alternating Queues (*x=1*)
- Constant Frame Size
- Two classes
- No (!) lower priority interference
- Zero link delay

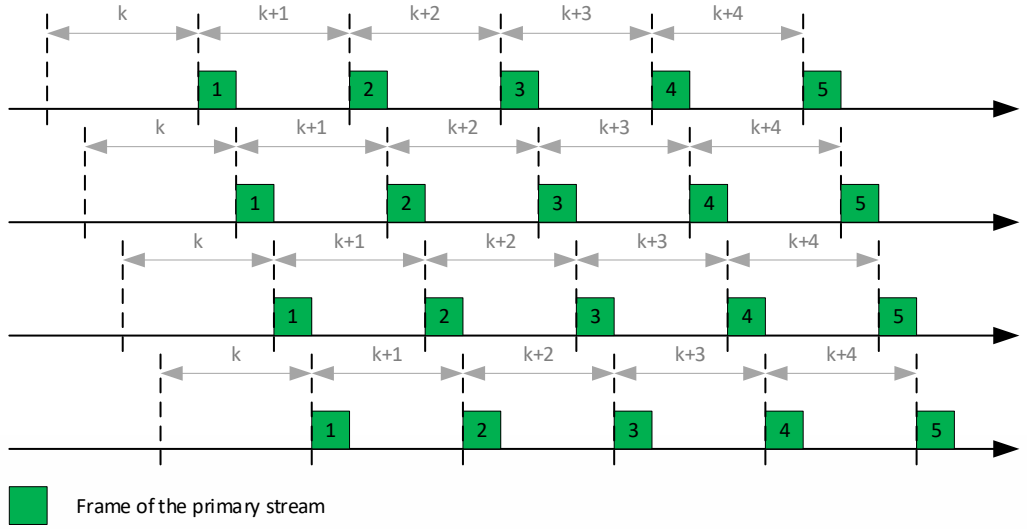


Syntonized CQF: The Nominal Case

Trivial Assumptions

- 2 Alternating Queues (like sync. CQF)
- Constant Frame Size
- One class
- No (!) lower priority interference
- Zero link delay





Operation

- Syntonized only
  (nodes run at the same frequency,
  but with random phase shifts)
- Downstream node
  - Buffer ingress cycle k, egress in cycle k+1
  - Buffer ingress cycle k+1, egress in cycle k+2
  - Buffer ingress cycle k+2, egress in cycle k+3

  …

Syntonized CQF: With Interference

Trivial Assumptions

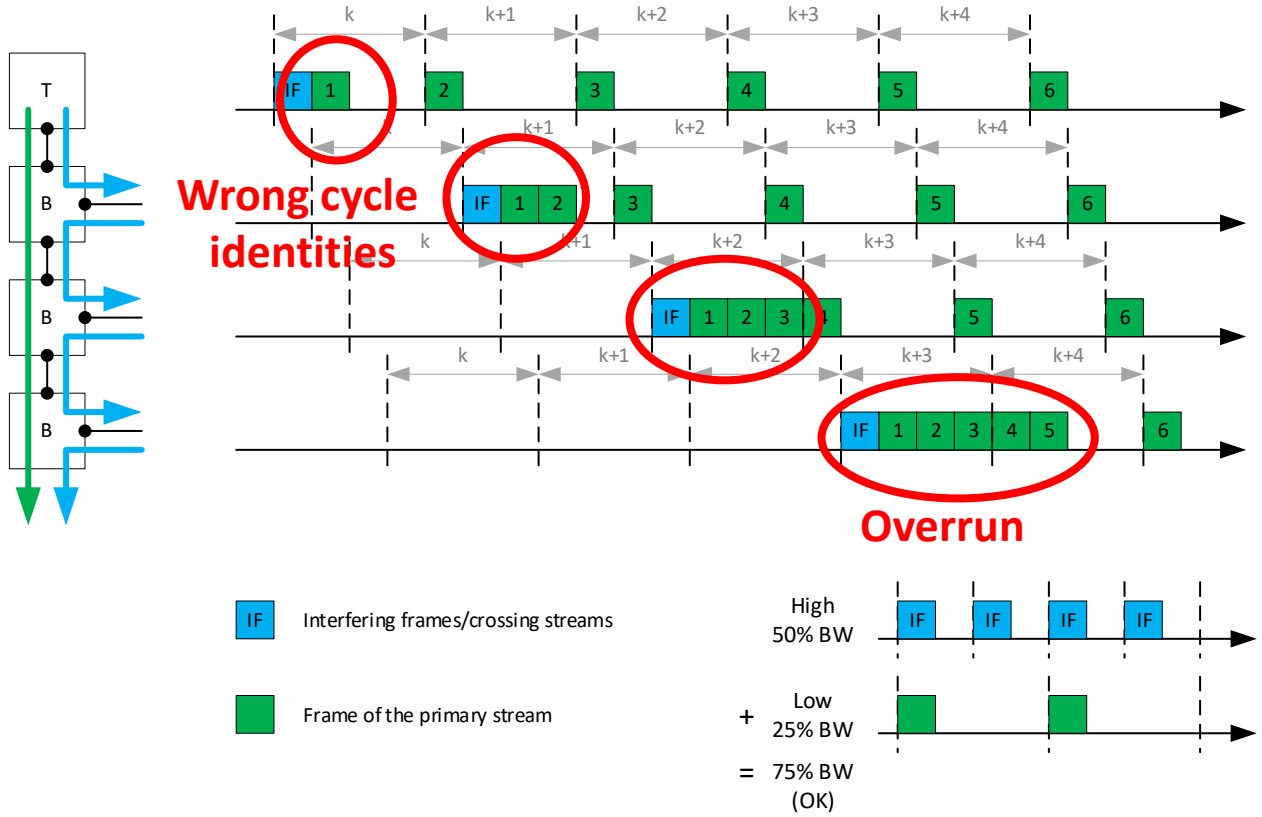
- 2 Alternating Queues (like sync. CQF)
- Constant Frame Size
- Two classes
- No (!) lower priority interference
- Zero link delay

Issue

- Overrun due to cross traffic, a.k.a. burst accumulation
  (this is not a “corner case”)
- An effect like with plain FIFO queuing (a.k.a. strict priority transmission selection algorithm), but on a “macroscopic” level
- Preemption won’t help here [period]

Conclusion

- **<u>Syntonization is not enough, CQF requires synchronization!</u>**



Note: Just for illustration – there are many other setups (including single class), but the author considered this example more intuitive.

More Thoughts

Can we fix it differently?

- Adding dynamic packet state (DPS) with cycle IDs[1] *looks promising*
- **But …** this implies new challenges[2]

About synchronized Multi-CQF

- *Looks very promising* for path latency balancing with 802.1CB
- Link delays must be low, otherwise cycle identities are also lost with synchronized CQF

On Paternoster

- *Looks promising* as a “reduced ATS”
- Bundling may be quite complex in detail

What means “looks promising”?

- The author is **not aware** of a *clean* analysis/formal proof of the desired easy to compute and tight latency bounds
- What means “clean”: Self-contained, math sound, complete, broad applicable/generic (e.g., applicability to single hop paths only appears insufficient)

1): See also https://www.ietf.org/archive/id/draft-qiang-detnet-large-scale-detnet-05.txt
2): See also https://www.ieee802.org/1/files/public/docs2020/new-specht-dampers-fti-0620-v02.pdf

Summary & Conclusions

Summary

- Core operation of pulsed queues
  - A.k.a. calendars, a.k.a. timing wheels, …
  - Just **one implementation**, but there are alternatives (heaps, etc.)
  - 802.1Qcr introduced an abstract model
    → covers **(hopefully) all implementations**
  - The most **efficient implementation**
    → **depends** on the ASIC **design**
  - Efficient ASIC **design**
    → **no “one size fits all”**
- **Syntonized CQF** “feels” trivial to understand (no analytic proof needed…), but it **has issues, as pointed out**
- **Paternoster** is missing clean analytic proof (with/without bundling)

Conclusions

Standardizing new shapers in IEEE 802.1 is a good idea, in general

There is **no "one size fits all"**

→Different shapers address different areas in a multi-dimensional problem space.

Recommendation: Do the math first

- The analytic work to proof the desired latency properties of new shapers should be done before standardizing.
- This is not new, it was stated earlier, and is based on experiences the IEEE 802.1 TSN TG made in the AVB days.
- **Affected**: **Syntonized CQF** and **Paternoster (with/without bundling)**

Abstract models can cover multiple implementations

- ATS introduced an abstract model to describe the externally visible behavior.
- Extending/generalizing this model appears more reasonable than limiting implementation(s).
- Again: Implementing standardized behavior efficiently is no rocket science, but not primary objective of IEEE 802 standards.

If we limit to synchronized [Multi-]CQF, here is a rough outline how this could look like

1. Add at least **one more pair of internal clocks**
   - Running on synchronized time
   - Co-existent to the present pair of (asynchronous) internal clocks – ATS and CQF can be used in parallel
2. Add a **new eligibility time assignment function** for CQF
3. Potentially **generalize the standardized ATS transmission selection algorithm** (i.e., transmission in order of eligibility times appears identical)
4. **Clean-up Annex T (CQF)** of IEEE Std 802.1Q, maybe some other CQF locations

Thank you for your Attention!

Questions, Opinions, Ideas?

Johannes Specht
Dipl.-Inform. (FH)
M +49 (0)170 718-4422
johannes.specht.standards@gmail.com