Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Institute of Computer Science
Chair of Communication Networks
Prof. Dr. Tobias Hoßfeld

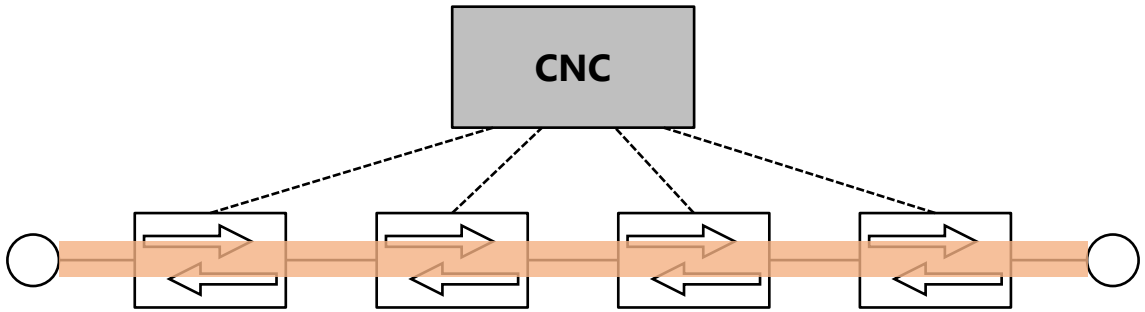
Measurement Points for Worst-Case Per-Hop Latency Computation

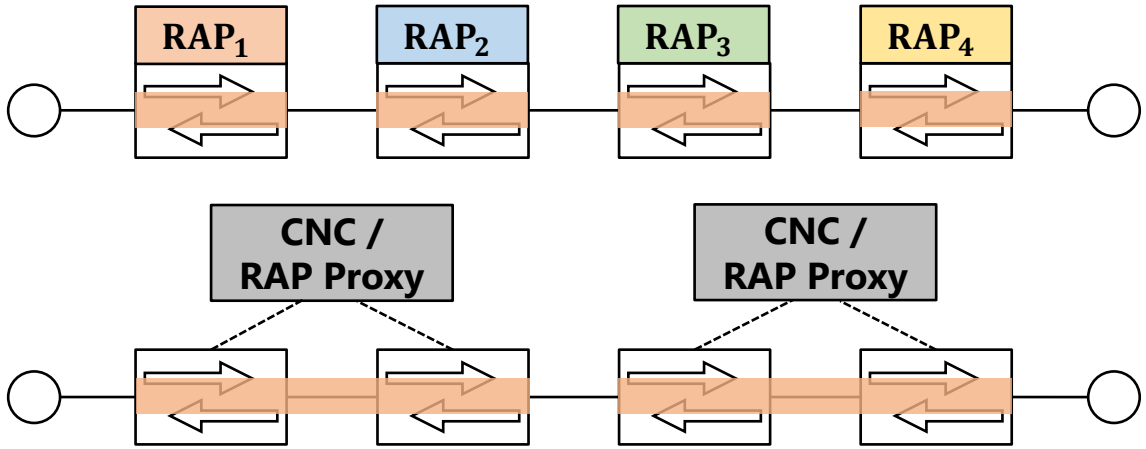
IEEE 802.1 Interim Session, May 2022

Alexej Grigorjew
alexej.grigorjew@uni-wuerzburg.de

Why are we talking about measurement points?

- Reservation protocols give latency guarantees based on traffic specifications
- We often talk vaguely about **end-to-end** latency and **per-hop** latency
- In various setups, end-to-end latency must be obtained by combining per-hop latencies

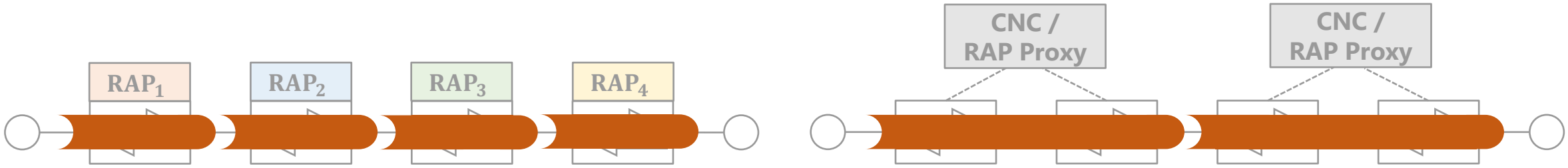




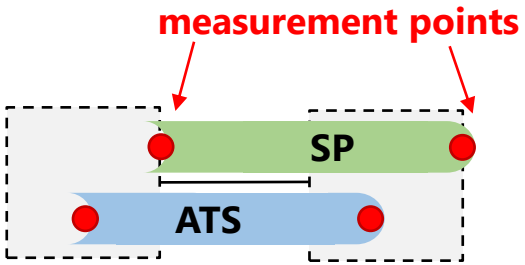
- Currently, multiple different interpretations of *"per-hop latency"* and measurement points exist
  - Edge to edge?
  - Egress queue pop (on bridge 1) to egress queue pop (on bridge 2)
  - Frame fully received (on bridge 1) to frame fully received (on bridge 2)
  - Shaper to shaper (based on eligibility times)

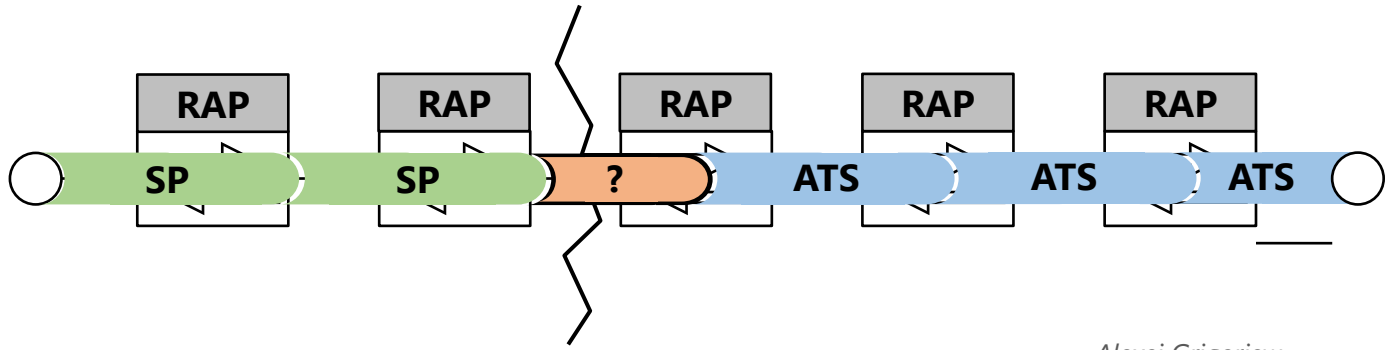
RAP needs a unified understanding of „per-hop latency“

- Different RAP implementations must be compatible
- The same measurement points for per-hop latency should be used by everyone



- But there is more:
  - What if different devices use different shapers?
  - Different shapers could currently have different latency models
  - The same measurement points should be used even with different shapers for full compatibility



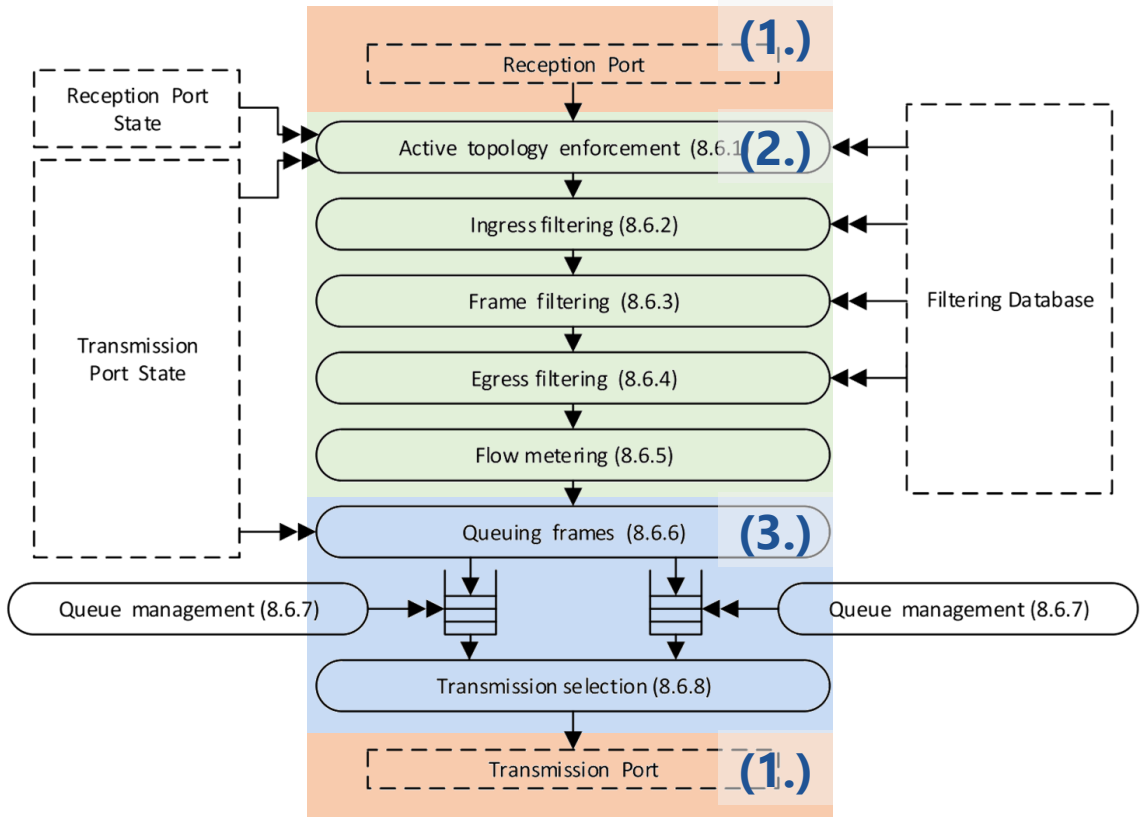


Sources of delay

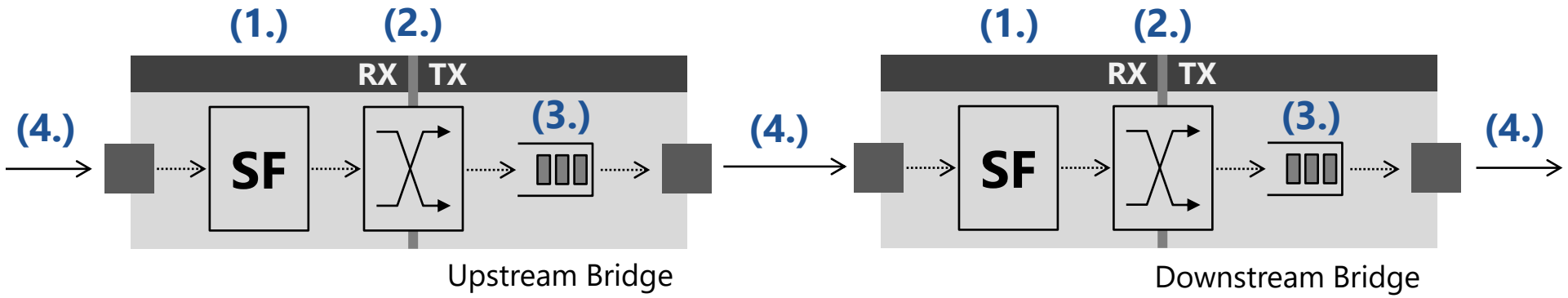
1. Store-and-Forward (→ frame size / link speed)
2. Processing (→ everything else)
3. Queuing
4. Propagation (→ distance)

- Note that these delays are not strictly disjunct
- E.g., processing may occur during store-and-forward

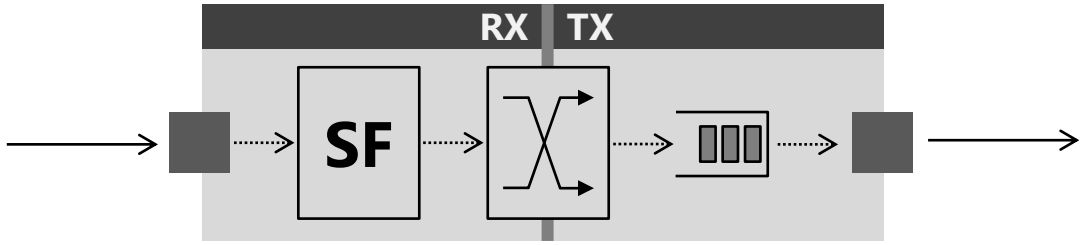
- Visualization for further reference:



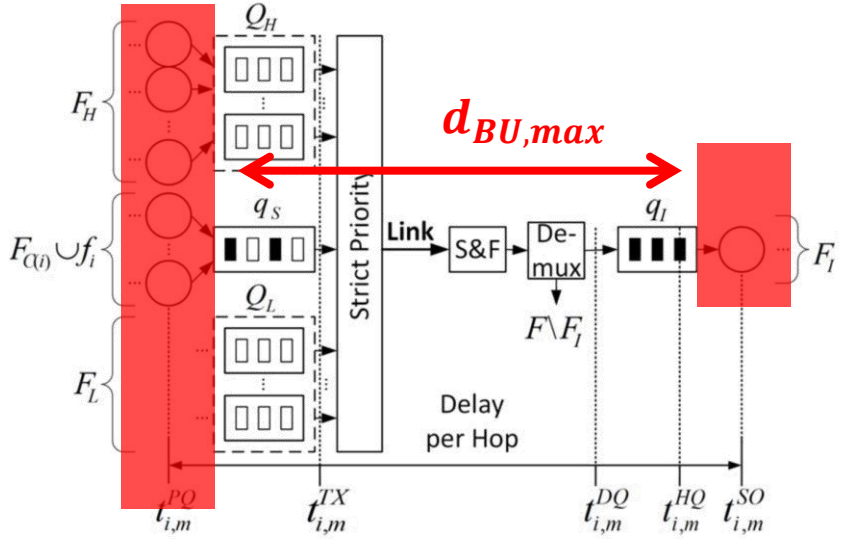
802.1Q-2018, Figure 8-12



Queuing and shaping (ATS)



Prop. SF Proc. Queue Prop.



Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks, Figure 5

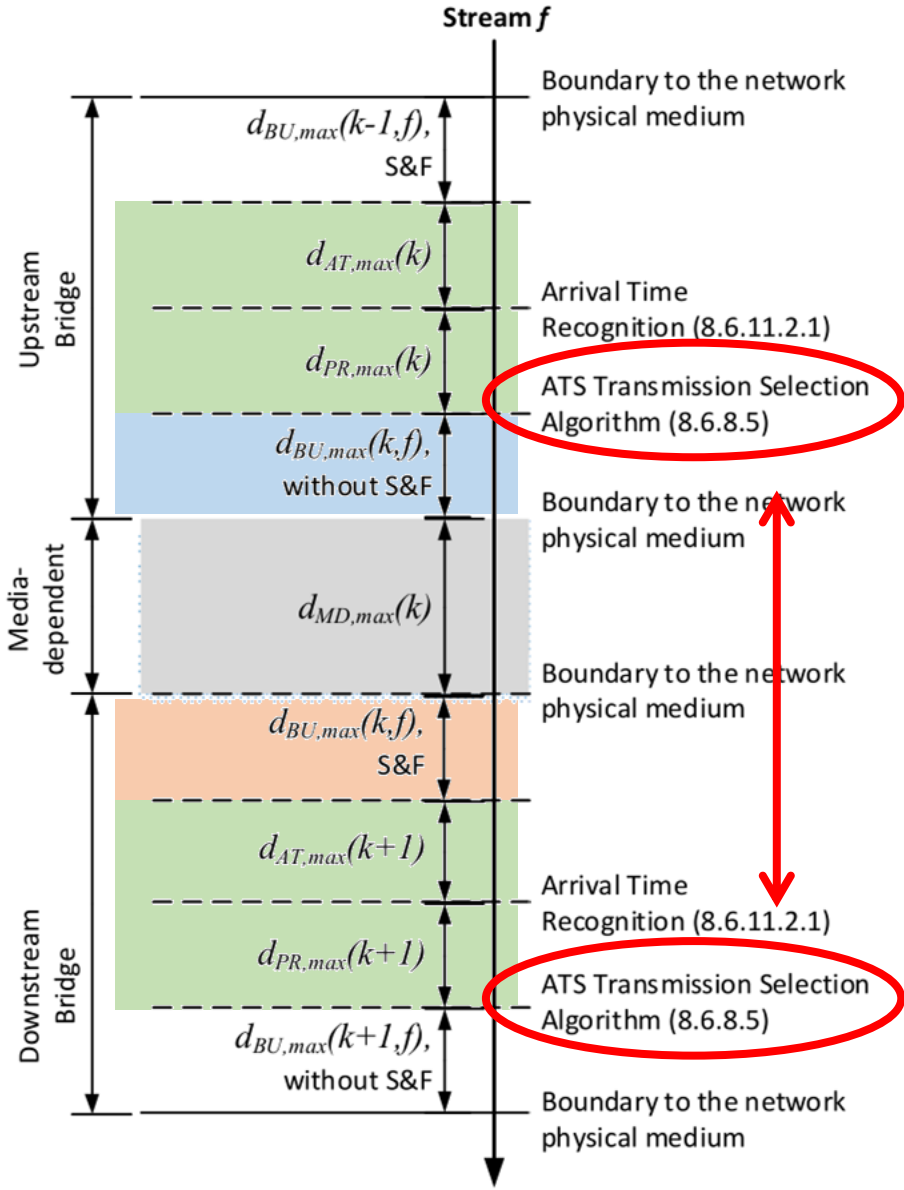
Processing

Queue

Prop.

SF

Processing



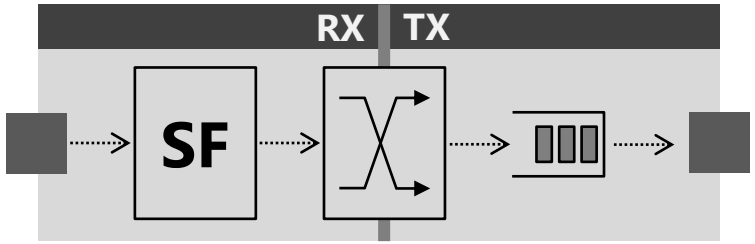
802.1Qcr-2020, Figure V-1

- ATS latency model includes downstream transmission selection!

Extended delay model, including transmission selection algorithm

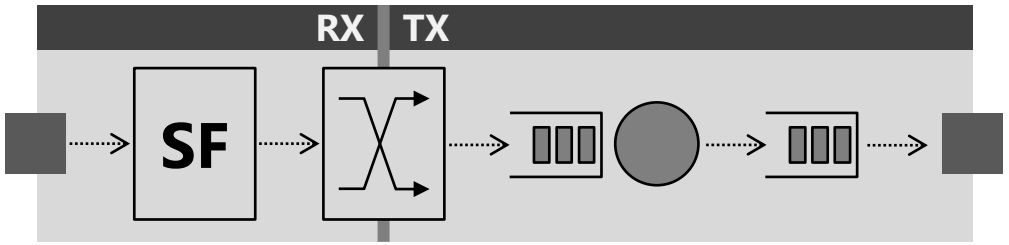
- Split “queuing” latency of formal latency models into…
  - Transmission Selection Algorithm (TSA)
  - Priority-Queuing, where only the eligible frames interfere

Previous model:



Queuing
(including TSA)

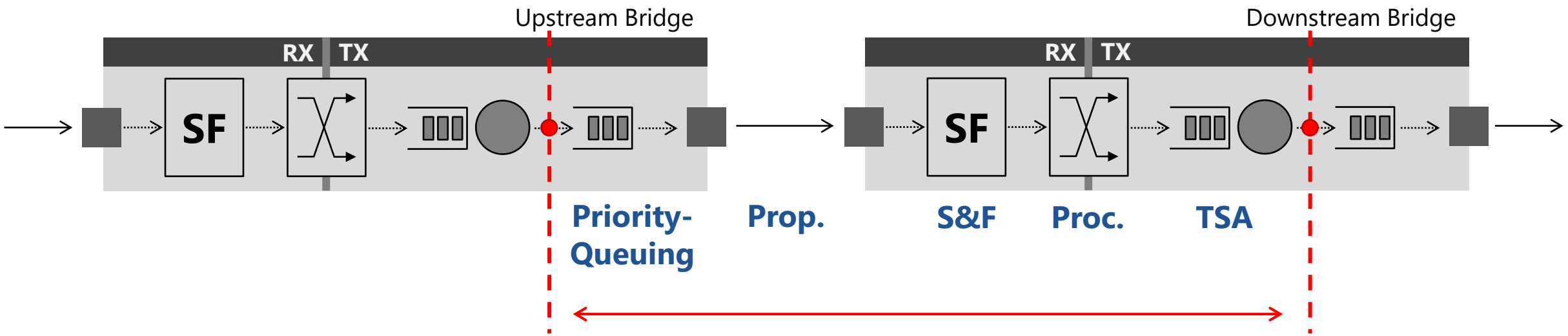
Extended model:



TSA **Priority-Queuing**

- Add measurement point during queuing when frame **becomes eligible for transmission**
  - SP: Immediately after enqueuing
  - CBSA: When credits >= 0, the head of the queue becomes eligible for transmission
  - ATS: When the defined eligibility time for that frame is reached (cf. Qcr)
  - CQF: When queues swap roles (receive → send), all frames in the send queue become eligible

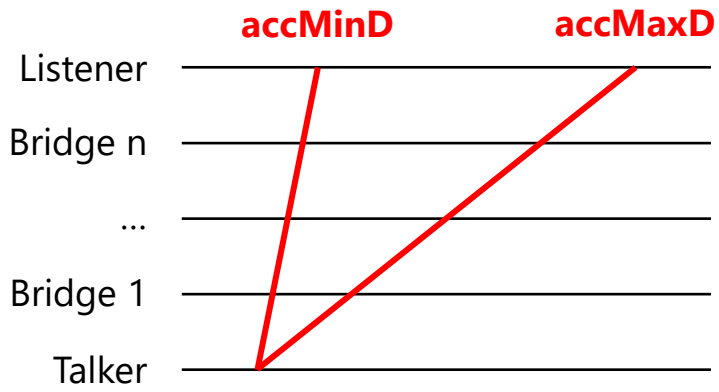
Suggestion: Use ATS measurement points for all shapers in RAP



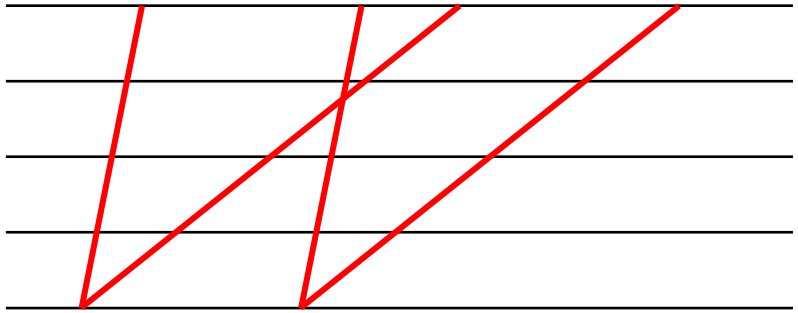
- Suggestion: Use the ATS measurement points for all TSAs & latency models in RAP
- Per-hop latency is given by…
  - Queuing, after eligibility time was reached (upstream)
  - Propagation
  - Store-and-Forward (downstream)
  - Processing (downstream)
  - Transmission selection / queuing, until eligibility time is reached (downstream)

Why is shaper-to-shaper latency beneficial?

Distributed latency model:



CQF (edge to edge measurement):



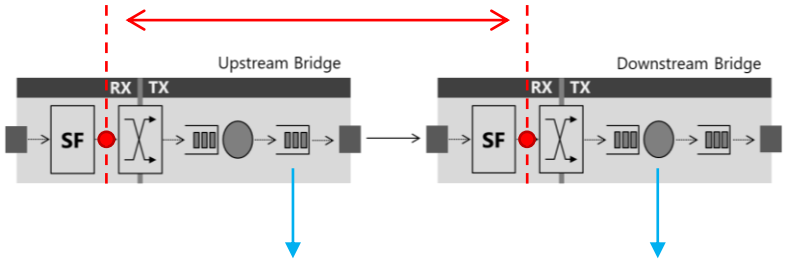
CQF (shaper to shaper):

[dd-grigorjew-strict-priority-latency-0320-v02.pdf]

Generally:

- Minimum delay and maximum delay accumulated per hop
- Accumulating bursts are calculated based on $(accMaxD - accMinD)$
- A lower latency variance is better for downstream delay computation

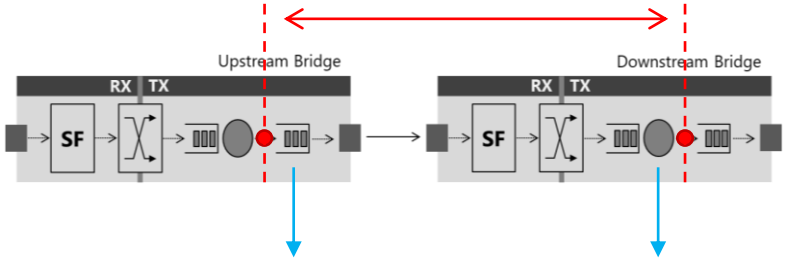
Fully-received to fully-received:



Upstream priority queue is a major **source of jitter**

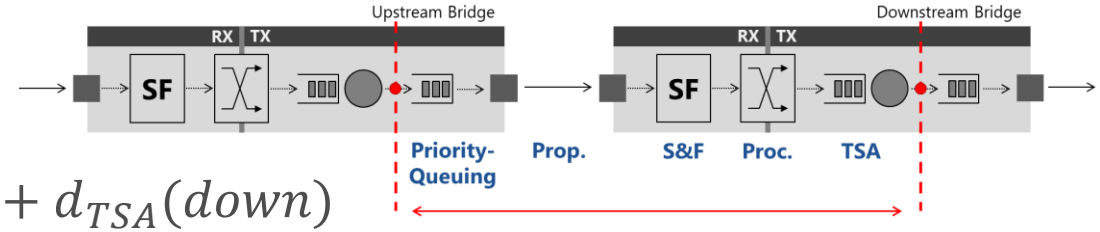
CQF removes this jitter – but only **after** the measurement point

Shaper to shaper:

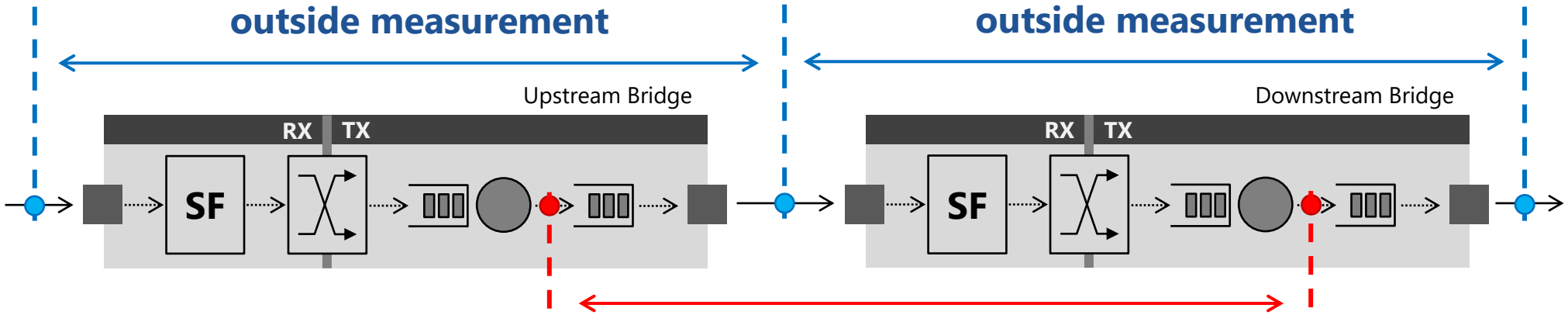


All sources of jitter can be removed; **Well-defined traffic pattern, as intended by the TSA, is measured directly after the TSA**.

More Implications



- $d_{hop} \leq d_{queue}(up) + d_{prop} + d_{SF}(down) + d_{Proc}(down) + d_{TSA}(down)$
- Downstream bridge must know some details about upstream bridge to compute $d_{queue}(up)$
  - All reserved streams of that egress (which should already be known by downstream)
  - Priority to traffic class mappings (in order to calculate worst-case priority queuing latency)
- Latency resource budget configuration (and admission control) can be more fine-grained
  - One threshold per class, per ingress (priority queuing), and per egress (TSA)
    `latency_guarantee[class][ingress_port][egress_port] = 1234µs`
  - For some shapers (e.g., SP), coarse-grained thresholds may suffice (d_{TSA} is always 0)
    `latency_guarantee[class][ingress_port][*] = 1234µs`
- " Verificaiton measurements" from the outside do not match these measurement points



Summary

- We should specify latency measurement points in RAP clearly and unambiguously
  - Interoperability between different RAP implementations
  - Interoperability between shapers
  - Enables simple transitions between CNC domains (use RAP between two CNCs)
  - Facilitates transitions between networks (DetNet)
- Suggestion: use the measurement points from the latency analysis of ATS for all TSAs / shapers
  - Per-hop latency from moment of upstream TSA eligibility to downstream TSA eligibility
  - Compatible with existing Qcr-2020 annex
  - Traffic pattern is well-defined at that point, maximizing the effect of TSA on latency math
  - Example: ATS (cf. latency analysis), CQF (max delay = min delay at that point)
- Other implications
  - Downstream bridge must be able to calculate priority queuing delay of upstream bridge
  - Threshold configuration can be more fine-grained (`threshold[class][ingress][egress]`)
  - Measurements from the outside represent different per-hop delays

THANK YOU!

Questions, comments, suggestions?