DETERMINISTIC6G

IEEE 802.1 Interim Meeting

Control Plane Extensions for Wireless-Aware Traffic Engineering with Corresponding YANG Data Models

Hamburg, Germany,
September 20, 2024

Frank Dürr (University of Stuttgart)
Simon Egger (University of Stuttgart)
Joachim Sachs (Ericsson)

DETERMINISTIC6G has received funding from the European Union's Horizon Europe programme under grant agreement No 101096504.

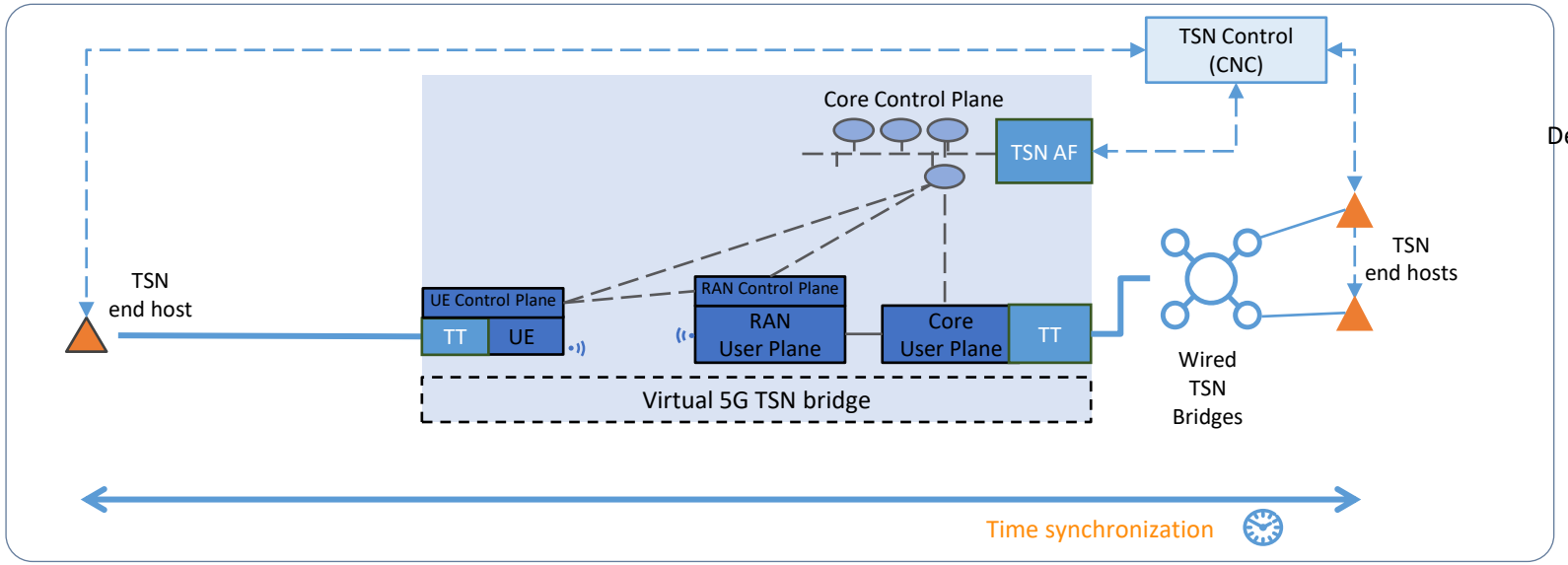
Agenda

- Virtual (wireless) TSN bridge
  - Characteristic port-to-port delay
- Wireless-friendly end-to-end scheduling
  - Requirements on port-to-port delay information
- Extended YANG Model for port-to-port delay
- NETCONF to retrieve port-to-port delay information

5G Standard Support for TSN

Standardized: 5G support for TSN (and for DetNet similarly)

- The 5G System represented as a **Virtual (Wireless) TSN bridge** in the end-to-end TSN view
  - External behavior (functionality) same for wired and wireless bridges
  - Including in particular gates and Gate-Control List (time table) for scheduled traffic (IEEE 802.1Qbv)
- Upper-bound latency via *ultra-reliable and low latency* (URLLC) communication

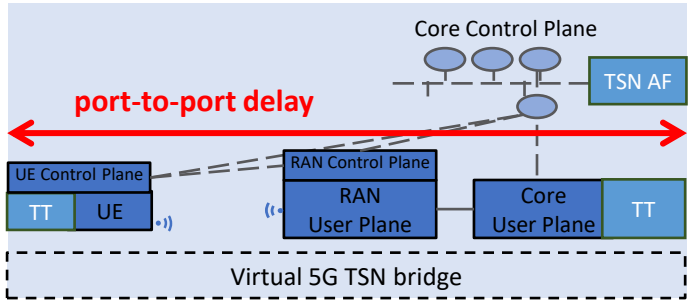


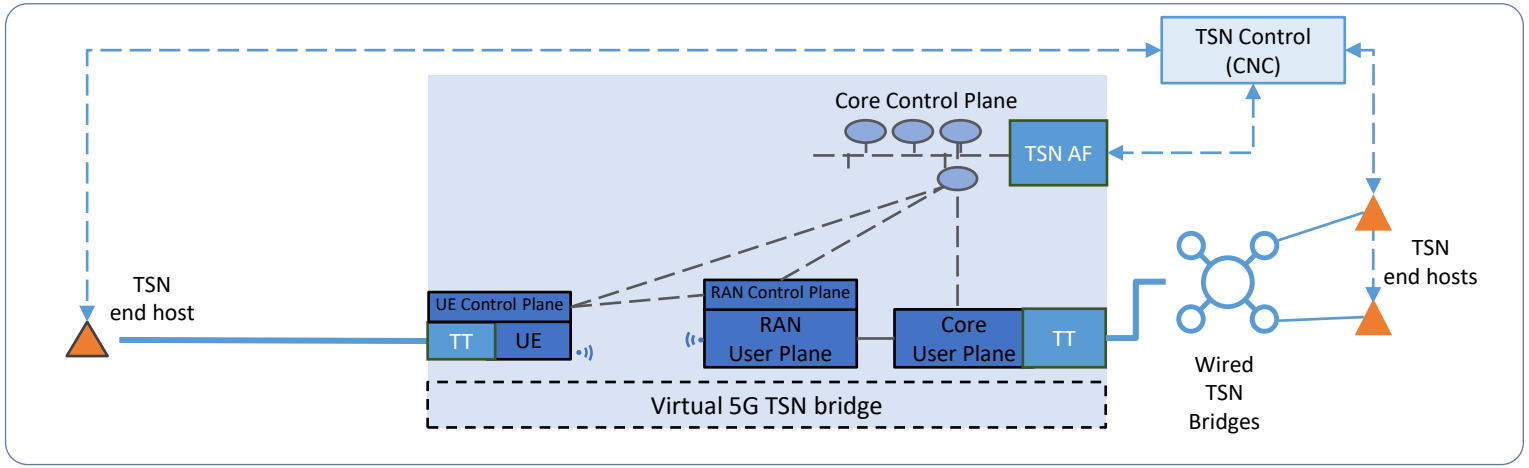
3GPP TS 23.501
5G-ACIA: 5G-TSN and 5G for IIoT Deliverables D1.4, D1.5, D5.1, D5.2, D5.3, D5.4, D5.5 and 5G-SMART Booklet [5G-SMART]

UE : User Equipment
RAN: Radio Access Network
CNC: Centralized Network Configuration
TT: TSN Translator
AF: Application Function

End-to-End Scheduling of Scheduled Traffic (802.1Qbv)

- CNC configures schedules (time tables = gate control lists) for scheduled traffic (IEEE 802.1Qbv)
  - Schedule controls **queuing delay** (only)
- Algorithms for schedule calculation require information about delay:
  - Propagation delay between bridges
  - Transmission delay of frames (from line rate and frame size)
  - **Port-to-port delay of bridge (aka Bridge Delay)**





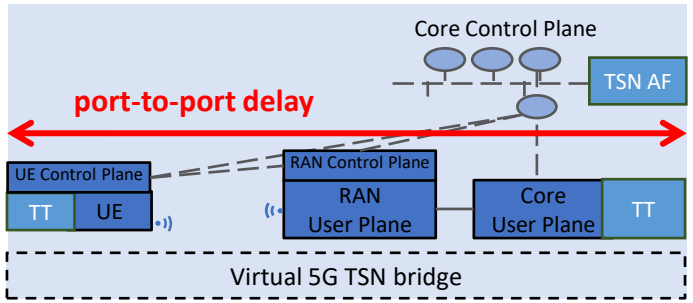
Port-to-Port Delay: Wired TSN Bridge vs. 5G Virtual TSN Bridge

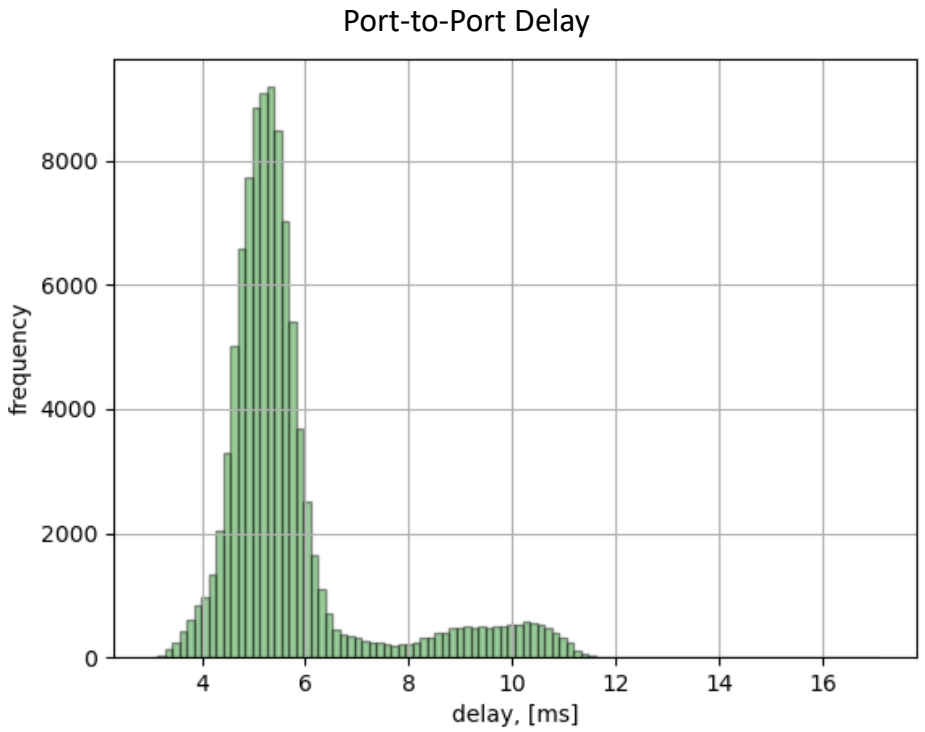
Port-to-Port Delay:

- Delay from ingress to egress port
- Without transmission selection
  - No queuing in egress queue (gates open)

Port-to-Port Delay Characteristics of Virtual Bridge:

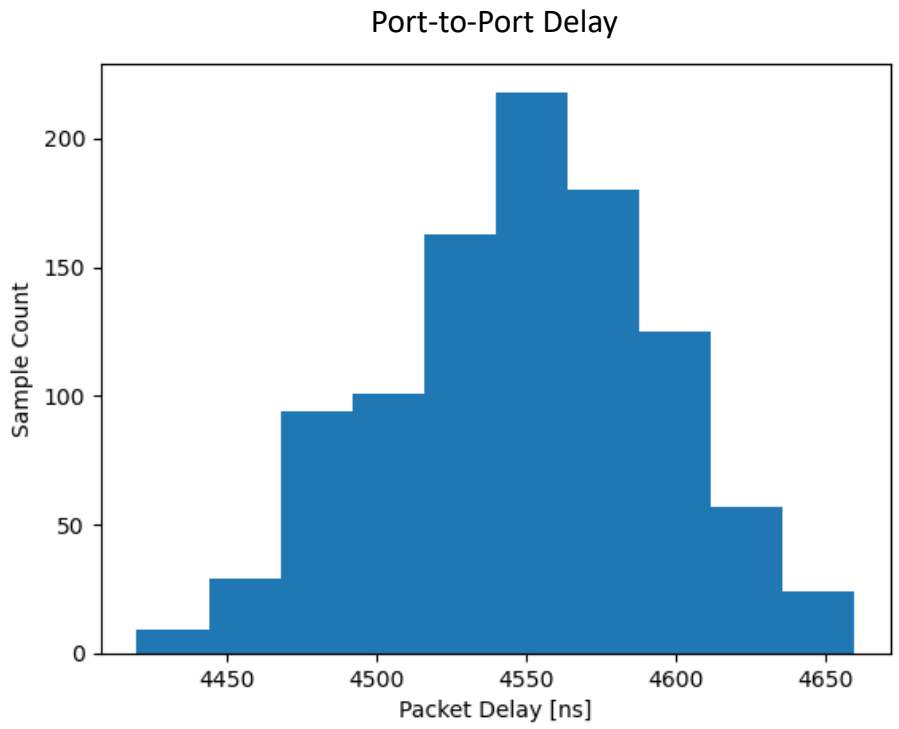
- Greater than for wired TSN bridges
  - But support for upper bound provided by URLLC
- Stochastic
  - Heavy-tailed



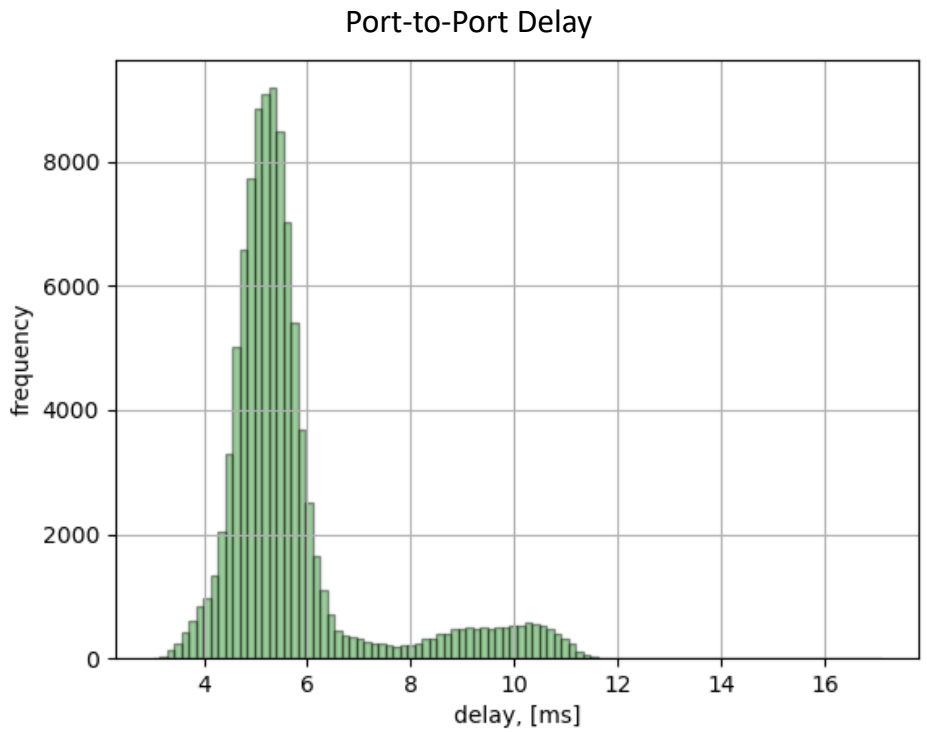


Port-to-Port Delay: Wired TSN Bridge vs. 5G Virtual TSN Bridge

Wired TSN Bridge



5G Virtual TSN Bridge



Delay data from measurements available here:
https://github.com/DETERMINISTIC6G/deterministic6g_data

Standard Support for Port-to-Port Delay Information (according to IEEE 802.1Q)

- "Each set of Bridge Delay attributes is accessed using three indices: **ingress Port**, **egress Port**, and **traffic class**."
- "The delays represent the **worst-case range per the design of the Bridge**, and are **not measured**."

Table 12-38—Bridge Delay attributes

Name	Data type	Operations supported[a]	Conformance[b]	References
independentDelayMin	unsigned integer	R	B	12.32.1.1
independentDelayMax	unsigned integer	R	B	12.32.1.1
dependentDelayMin	unsigned integer	R	B	12.32.1.2
dependentDelayMax	unsigned integer	R	B	12.32.1.2

[a] R = Read only access; RW = Read/Write access.
[b] B = Required for Bridge or Bridge component support of Stream reservation remote management; b = Optional for Bridge or Bridge component support of Stream reservation remote management.

[IEEE 802.1Q 2022]

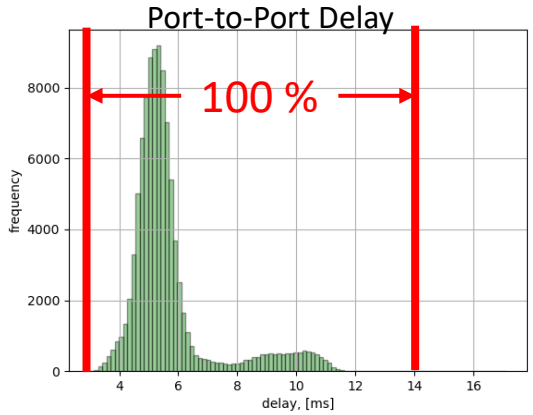
Agenda

- Virtual (wireless) TSN bridge
  - Characteristic port-to-port delay
- **Wireless-friendly end-to-end scheduling**
  - Requirements on port-to-port delay information
- Extended YANG Model for port-to-port Delay
- NETCONF to retrieve port-to-port delay information

Wired vs. Wireless-friendly Scheduling for 802.1Qbv

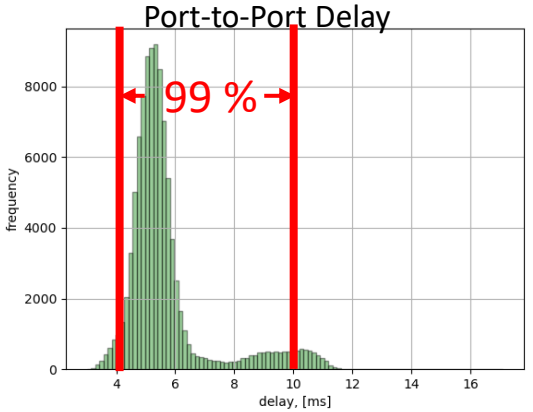
Wired TSN schedules

- **Reliability**(*1)**:** 100 % (**deterministic**)
  - Worst-case port-to-port delay



(Our) Wireless-friendly TSN schedules

- **Reliability:** a parameter (**dependable**)
  - Time window from delay distribution



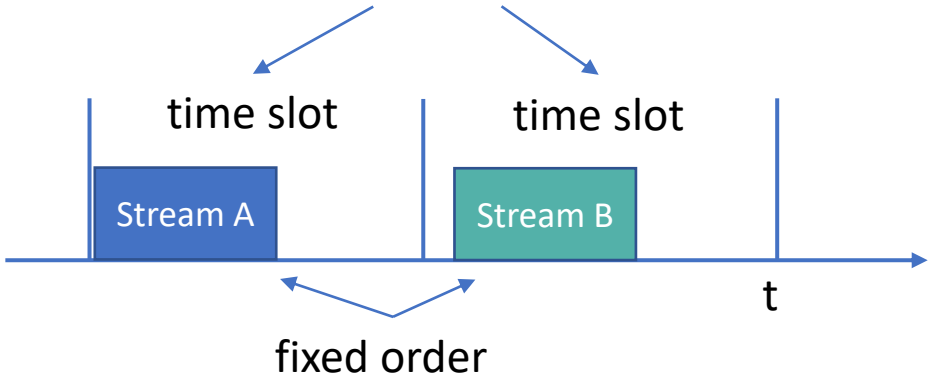
(*1) deliver frame within deadline

Wired vs. Wireless-friendly Scheduling for 802.1Qbv

Wired TSN schedules

- **No interleaving time slots**
  - Exclusive time slots per stream

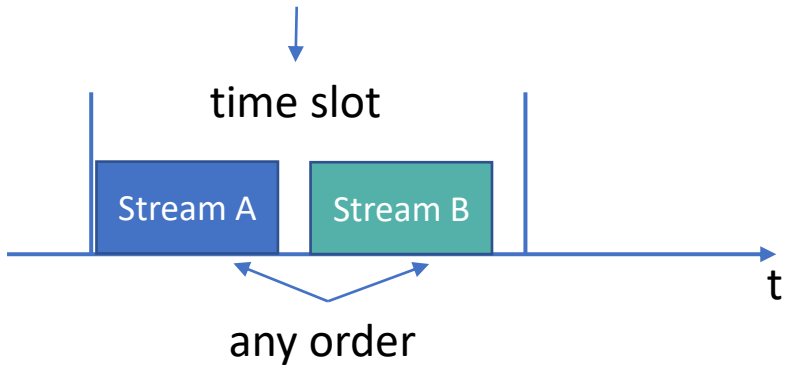
Time slots and stream assignment as defined by the calculated schedule



(Our) Wireless-friendly TSN schedules

- **Interleaving time slots**
  - Frames from multiple streams per time-slot

Time slots and stream assignment as defined by the calculated schedule



Wired vs. Wireless-friendly Scheduling: Performance

Evaluation Setup

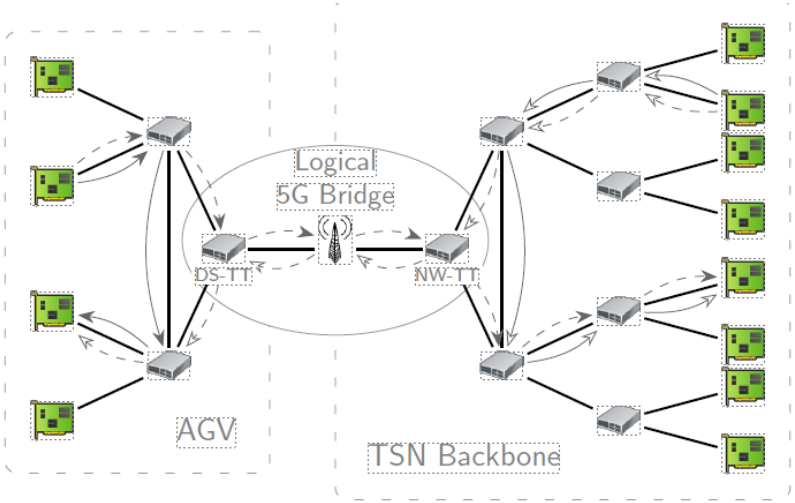
- Real 5G PD histograms
- 100 Mbps Ethernet links
- Frames per 20 ms hypercycle:
  - 60 wireless + 50 wired
- Simulation: 100k hypercycles

type	$f.size$	$f.period$	$f.latency$	$f.jitter$
wireless	100	20 ms	20 ms	5 ms
wired	100	4 ms	500 µs	0

Wired vs. Wireless-friendly Scheduling: Performance

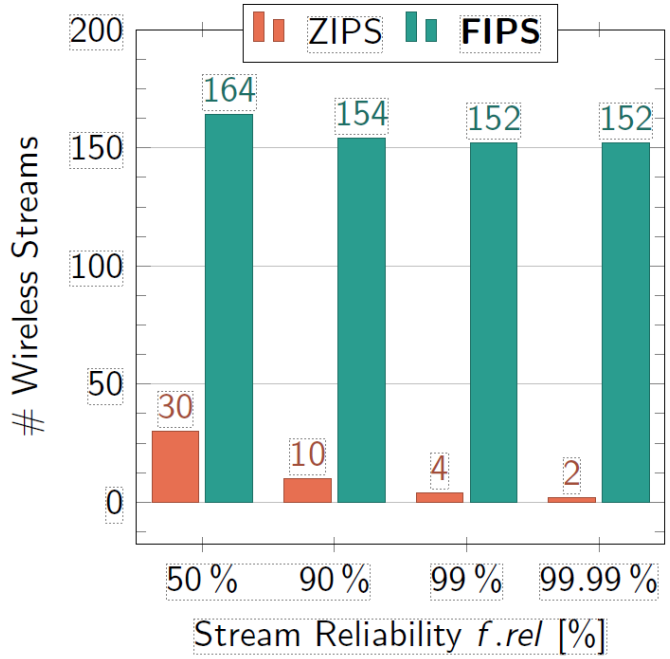
Evaluation Setup

- Real 5G PD histograms
- 100 Mbps Ethernet links
- Frames per 20 ms hypercycle:
  - 60 wireless + 50 wired
- Simulation: 100k hypercycles



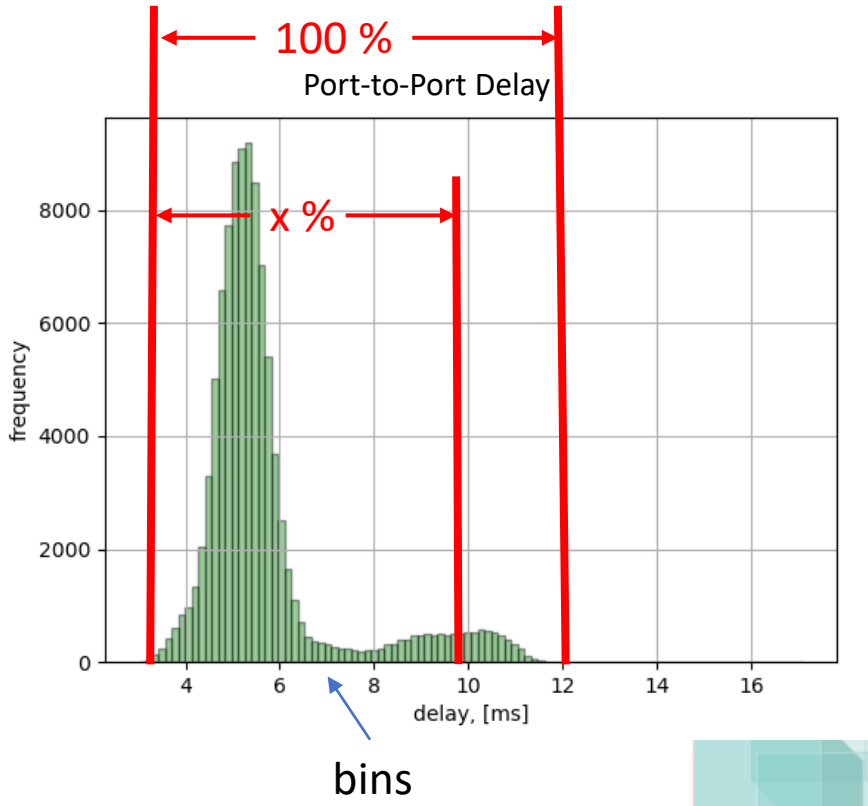
Efficiency: number of schedulable streams

- ZIPS: zero-interleaving (wired algorithm)
- FIPS: fully interleaving (wireless-friendly)



Requirement to Implement Wireless-friendly Scheduling: Extended Port-to-Port Delay Model

- Wireless-friendly scheduling needs information about **probabilistic port-to-port delay distribution**
- Our approach: **Histograms**
  - Binning of **online** (!) **delay measurements** by bridge
- Histograms are versatile including:
  - the standard worst-case bound model
    - 100 % between min and max delay (single bin)
    - **backward compatible** to standard min/max model
  - Quantiles
    - x % between min and max delay (single bin)
    - Difficult to specify the “best” quantile in a standard
  - Approximation of probability distribution functions
- Reasonably **simple to implement** on bridge

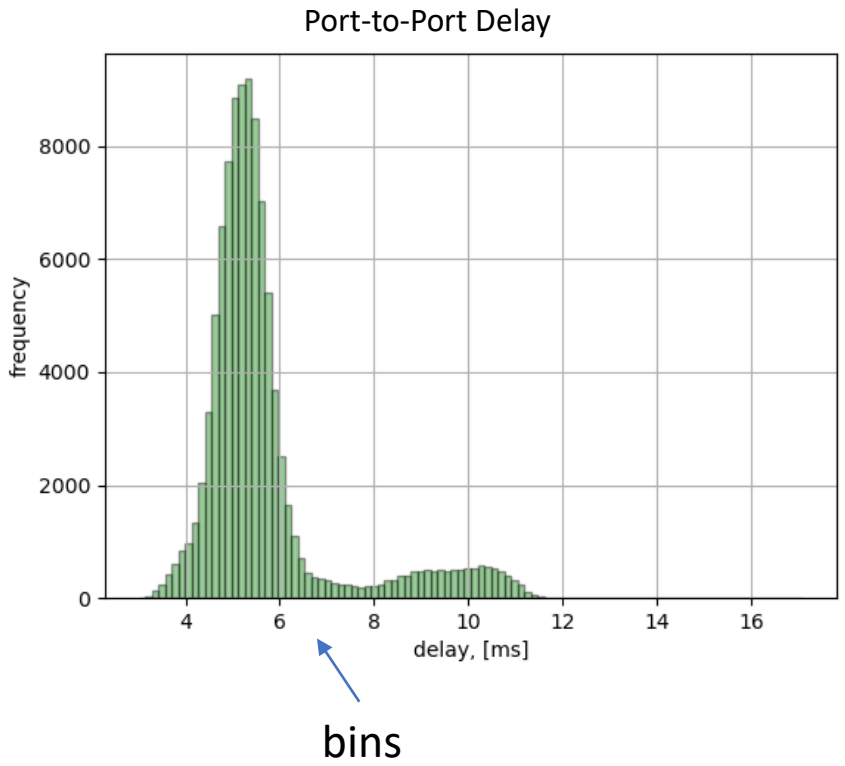


Agenda

- Virtual (wireless) TSN bridge
  - Characteristic port-to-port delay
- Wireless-friendly end-to-end scheduling
  - Requirements on port-to-port delay information
- **Extended YANG Model for port-to-port Delay**
- NETCONF to retrieve port-to-port delay information

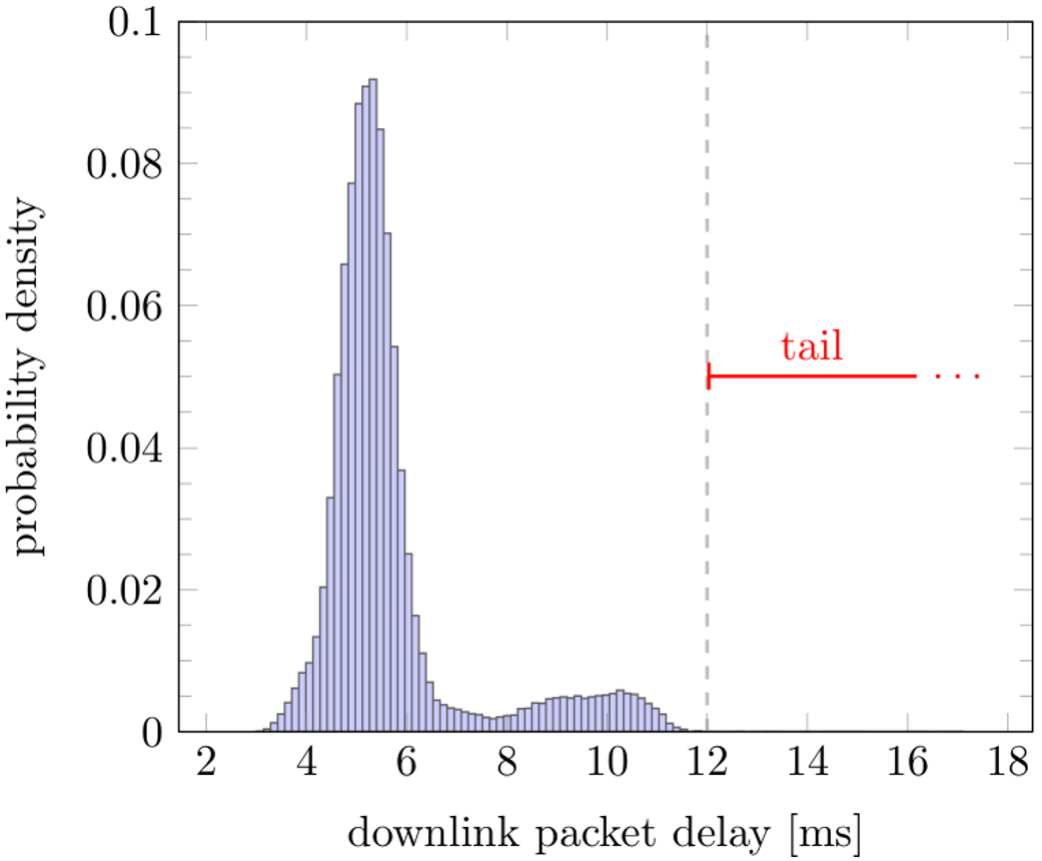
YANG Model for Port-to-Port Delay Histograms: Bins

```
grouping delay-histogram {
    description "Delay histogram";
    leaf start {
        type uint64;
        description
            "The start value of the first bin in nano-seconds.
            If not specified, the first bin starts at 0.";
    }
    leaf bin-count {
        type uint32;
        mandatory true;
        description "Number of bins.";
    }
    list bin {
        description "Bins of histogram.";
        key index;
        leaf index {
            type uint32;
            mandatory true;
            description "The index of this bin.";
        }
        leaf width {
            type uint64;
            mandatory true;
            description "The width of this bin in nano-seconds.";
        }
        leaf count {
            type uint32;
            mandatory true;
            description "Count of values in this bin.";
        }
    }
```



YANG Model for Port-to-Port Delay Histograms: Support for unbound delay distributions

```
grouping delay-histogram {
    …
    leaf tail {
        type uint32;
        description
            "Count of values in the
             tail of the histogram
             after the upper bound of
             last bin until infinity.
             Can be used to define an
             unbounded distribution.";
    }
```



YANG Model for Port-to-Port Delay Histograms: Added Below Bridge Component

- Added to **bridge component** (as for standard model; IEC/IEEE 60802)
- Per **port-pair** histograms (as for standard model)
  - Multiple histograms possible, e.g., for different frame sizes, channel conditions, etc.

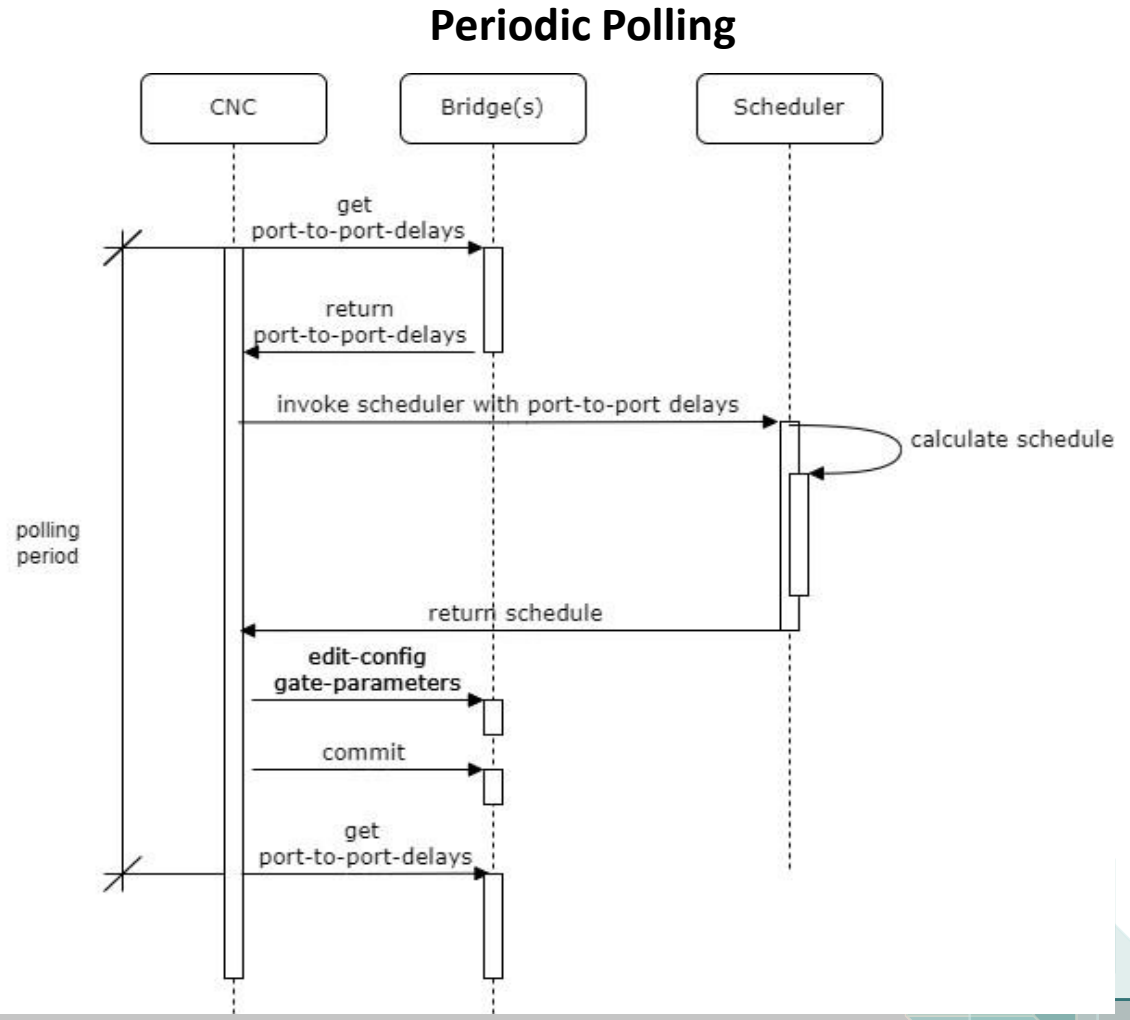
```
augment "/dot1q-bridge:bridges/dot1q-bridge:bridge/dot1q-bridge:component" {
    container port-to-port-delays {
        config false;
        list port-to-port-delay {
        key "ingress-port egress-port traffic-class index";
        leaf ingress-port {
            type dot1qtypes:port-number-type;
            config false;
            mandatory true;
            description "Unique number of ingress port.";
        }
        leaf egress-port {
            type dot1qtypes:port-number-type;
            config false;
            mandatory true;
            description "Unique number of egress port.";
        }
```

Agenda

- Virtual (wireless) TSN bridge
  - Characteristic port-to-port delay
- Wireless-friendly end-to-end scheduling
  - Requirements on port-to-port delay information
- Extended YANG Model for port-to-port Delay
- **NETCONF to retrieve port-to-port delay information**

NETCONF for Retrieving Port-to-Port Delay

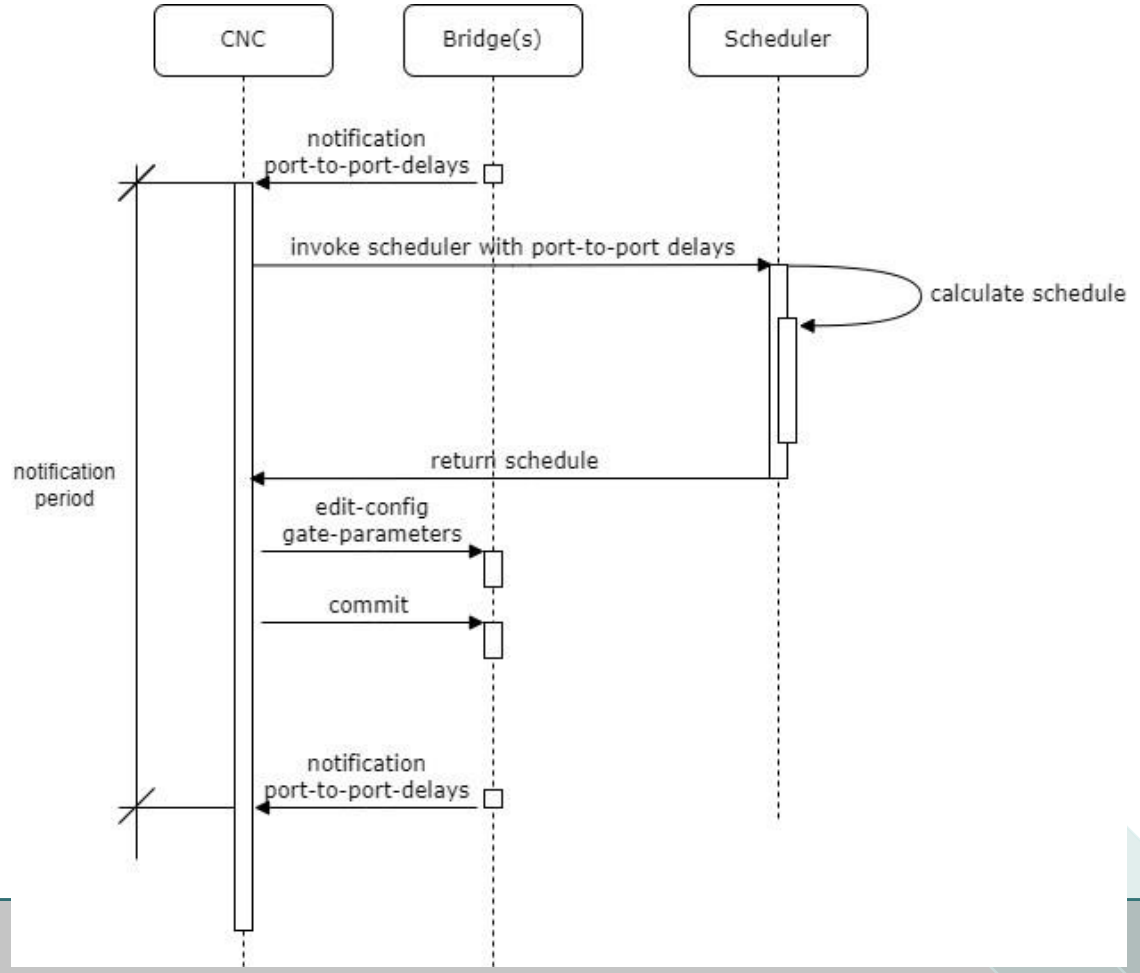
- Port-to-port delay might be non-stationary
- Different **NETCONF**-based schemes to retrieve current information by CNC:
- **Periodic polling** and **periodic subscriptions**
  - Difficult to define polling period



NETCONF for Retrieving Port-to-Port Delay

- Port-to-port delay might be non-stationary
- Different **NETCONF**-based schemes to retrieve current information by CNC:
- **Periodic polling** and **periodic subscriptions**
  - Difficult to define polling period
- **Event-based**
  - on-change subscriptions in YANG Push
- Proactively if delay prediction is available
  - Avoids invalid schedule due to slow schedule calculation

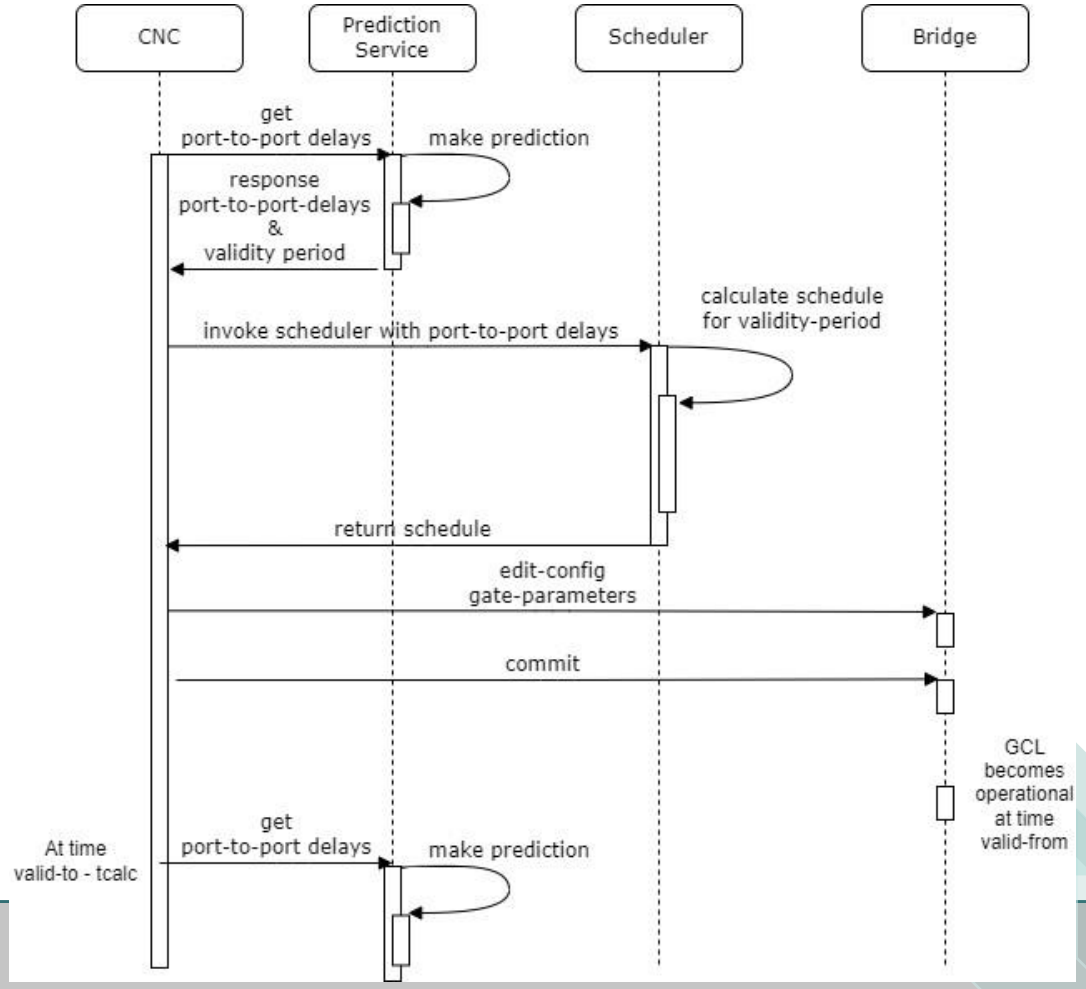
Event Notifications



NETCONF for Retrieving Port-to-Port Delay

- Port-to-port delay might be non-stationary
- Different **NETCONF**-based schemes to retrieve current information by CNC:
- **Periodic polling** and **periodic subscriptions**
  - Difficult to define polling period
- **Event-based**
  - on-change subscriptions in YANG Push
- **Proactively** if delay **prediction** is available
  - Avoids invalid schedule due to slow schedule calculation when reacting to out-dated distribution

Proactive



Conclusion

- **Virtual (wireless) bridge** has **characteristic port-to-port delay**
  - Greater magnitude, probabilistic, heavy-tailed
- **Wireless-friendly scheduling** can significantly improve efficiency of scheduled traffic
  - Interleaving of streams
  - Reliability as parameter of dependable schedule
- **Extended YANG models** to define histograms of port-to-port delay
- **NETCONF** to retrieve port-to-port delay information
  - Pull & event-based
  - Proactively with delay prediction

References / Contact

- Delay measurements of virtual TSN bridge (documentation and data):
  - D4.2: Latency measurement framework https://deterministic6g.eu/images/deliverables/DETERMINISTIC6G-D4.2_v1.0.pdf
  - Github: https://github.com/DETERMINISTIC6G/deterministic6g_data
- Wireless-friendly scheduling
  - D3.4: Report on Optimized Deterministic End-to-End Schedules for Dynamic Systems, https://deterministic6g.eu/images/deliverables/DETERMINISTIC6G-D3.4-v1.0.pdf
  - Contact authors for more information (pre-print): simon.egger@ipvs.uni-stuttgart.de
- YANG models, NETCONF integration (files and documentation)
  - D3.4: Report on Optimized Deterministic End-to-End Schedules for Dynamic Systems, https://deterministic6g.eu/images/deliverables/DETERMINISTIC6G-D3.4-v1.0.pdf
  - Github: https://github.com/DETERMINISTIC6G/deterministic6g_yang_models

… or directly contact: frank.duerr@ipvs.uni-stuttgart.de

DETERMINISTIC6G Grant Agreement No. 101096504

The DETERMINISTIC6G project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101096504.

If you need further information, please contact the coordinator:

Janos Harmatos, ERICSSON

E-Mail: coordinator@deterministic6g.eu

or visit: www.deterministic6g.eu

@DETERMINISTIC6G DETERMINISTIC6G