802.1Qcc findings

Astrit Ademaj

astrit.ademaj@tttech.com

Sept 2018

Background

- Present Qcc findings - mainly related to but not limited to centralized configuration model
- Information sharing, with the WG
- Discussion how to handle these findings (maintenance requests, additional PAR,…)

CNC Interfaces

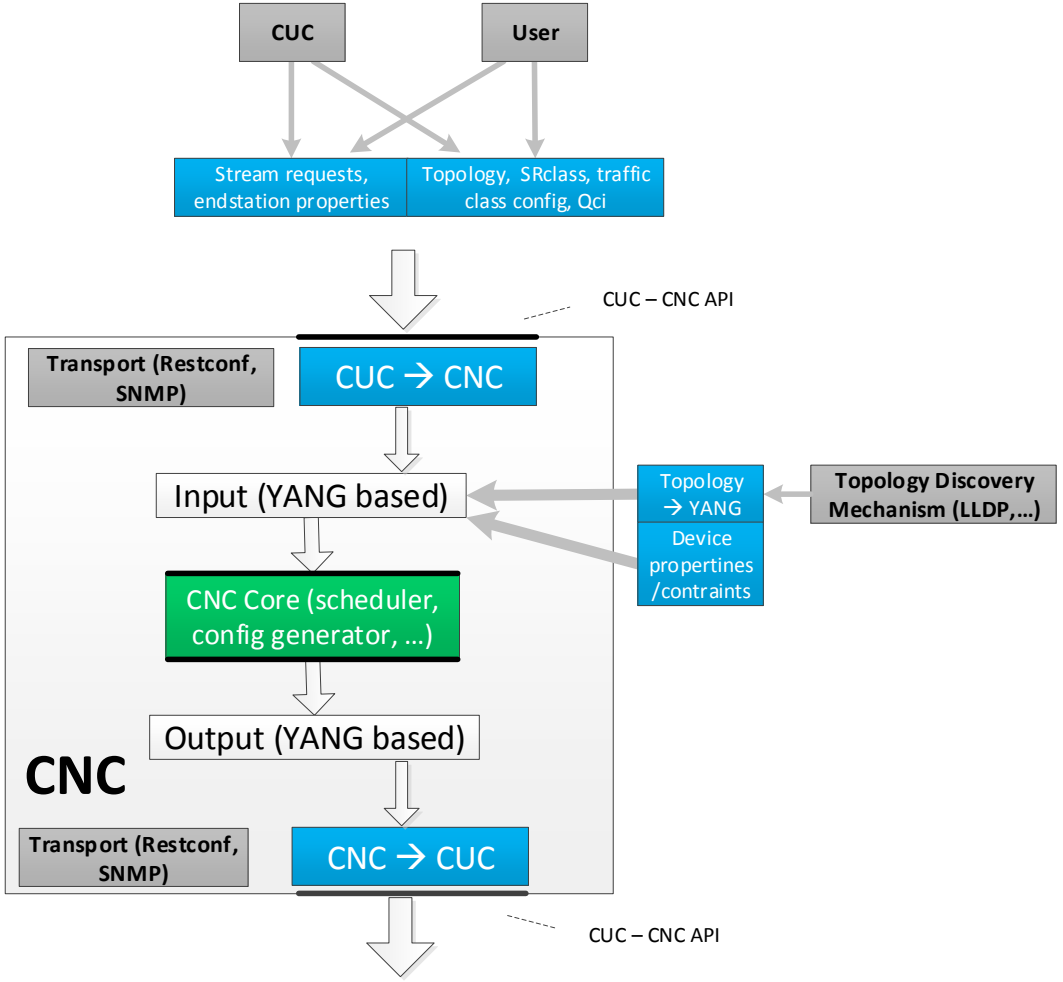
TTTech

CNC Input:

1. User input from CUC
   - stream configuration requests, end-station capabilities
2. From network services
   - topology, clock sync precision, bridge capabilities,…
3. From a user (or off-line CUC or Network Manager)
   - desired topology, SRclass configuration, reservation for best-effort (non-stream traffic), …

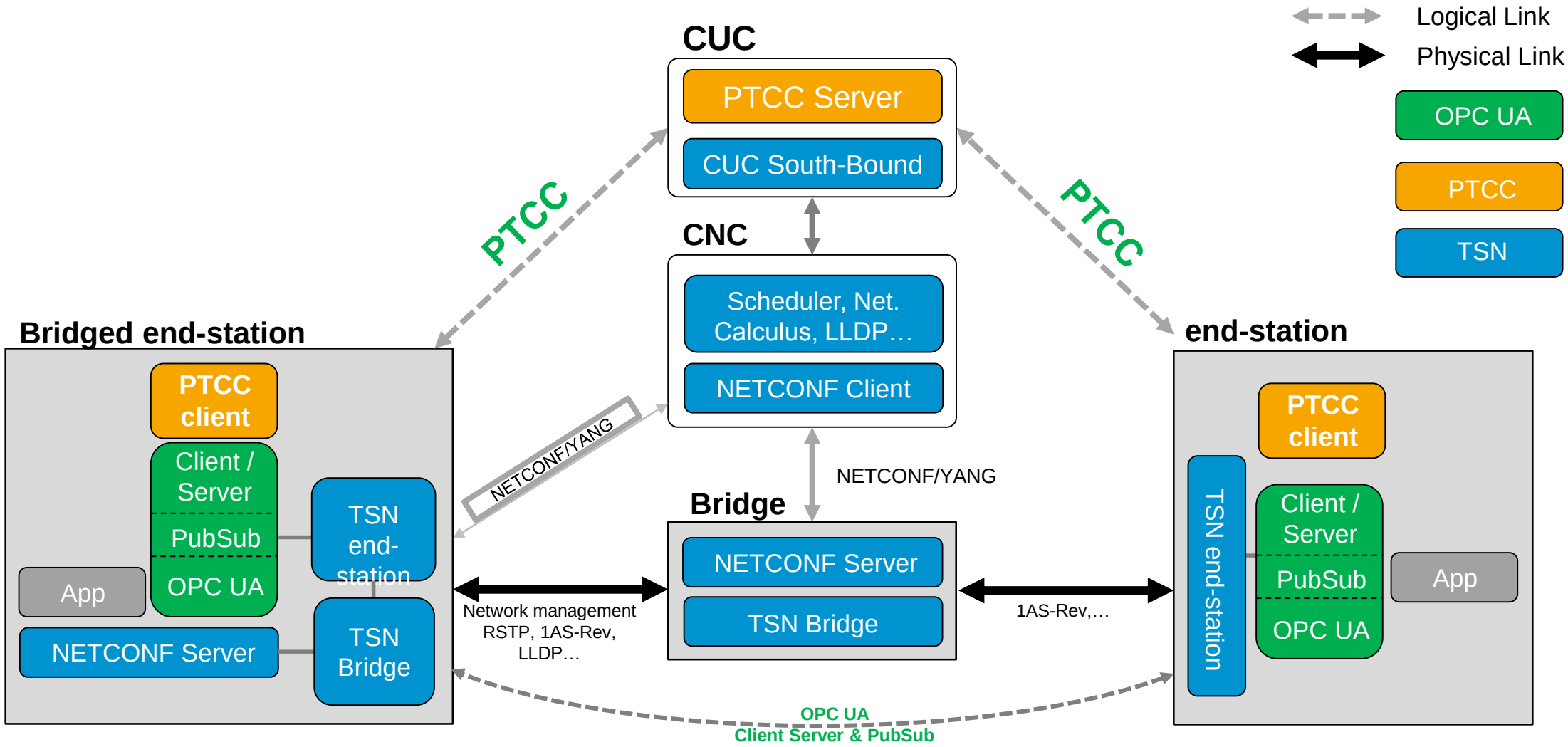
CNC Output:

1. Bridge configuration
2. Output to CUC
   - end-station configuration
   - discovered topology,
   - bridge configuration status, clock sync



OPC UA PubSub TSN Working Group work on PubSub TSN Centralized Configuration (PTCC)

TTTech



PTCC – and mapping to 802.1Qcc YANG snippets

TTTech

- PTCC specification – part of the OPC UA TSN Working Group specification
- PTCC is a request/response protocol between the *end-stations* and the *CUC*
  - PTCC Client (end-station)
  - PTCC Server (CUC)
  - Stream Object
    - State machine at the PTCC client and PTCC server

- PTCC message based interface between PTCC client and server
  - interface parameters with respect to *stream configuration* mapped to the YANG snippets from the IEEE 802.1Qcc ?!

Discussion points

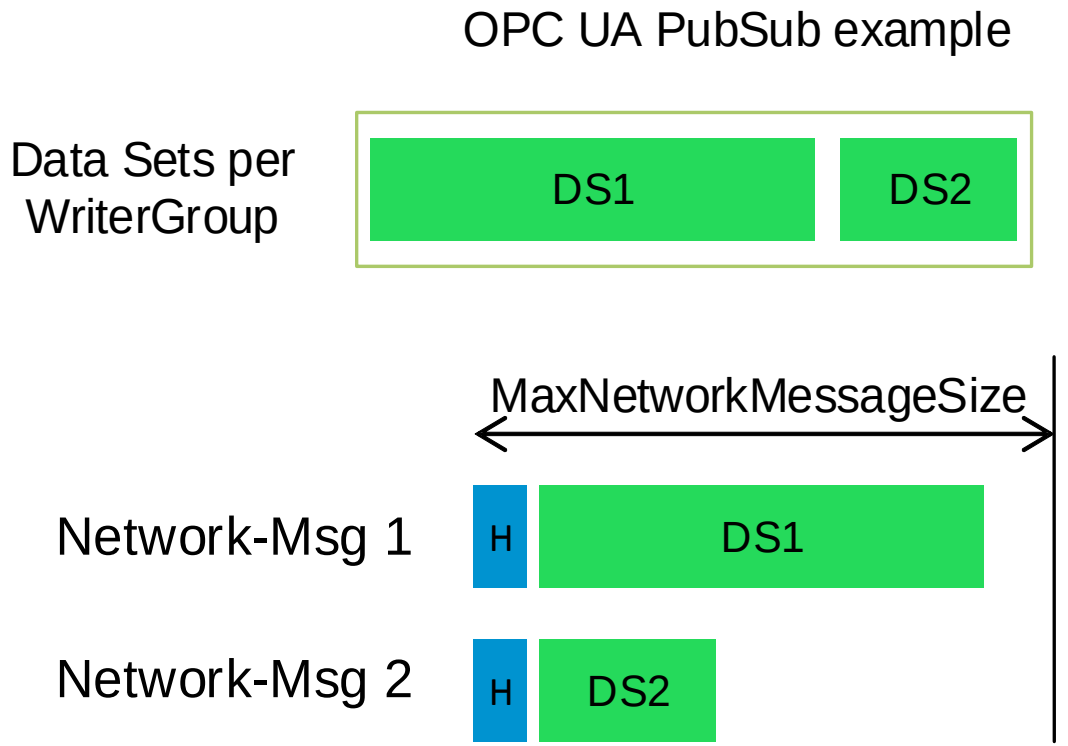
Some parameters from the PTCC interface with respect to *stream configuration request* cannot be mapped to the YANG snippets defined in IEEE 802.1Qcc

1. Bytes per transmission Interval (for streams consisting of more than one frame)
2. Missing parameter for latency requirement for scheduled traffic
3. Talker latency requirements
4. Number of Seamless Trees at talker
5. Maximum transmission GAP between two successive frames of same stream
6. Status information on clock synchronization for network components along the path of a given stream
7. Missing model to express user requirements for Qbu, Qci, …

1. Bytes per transmission Interval - for streams consisting of more than one frame

- Use case where the payload of a stream does not fit in 1 (one) Ethernet frame
  - Currently **max-frame-size** and **max-frames-per-interval** are available
  - In case of e.g., stream with 1800 payload bytes
    - `max-frame-size` = 1522 (1500 Payload)
    - `max-frames-per-interval` = 2
    - With current parameters it will results on a reservation of 3000 bytes , which is a waste of bandwidth for 1200 Bytes

      (2 x 1500 – 1800 = 1200 Bytes)



Proposal 1: Introduce additional parameter indicating the exact number of Bytes per interval (period): ***bytes-per-interval***

2. Latency requirements for scheduled traffic at listeners (1)

TTTech

§3.118 Latency – is defined as:

> The delay experienced by a frame in the course of its propagation between two points in a network, measured from the time that a known reference point in the frame passes the first point to the time that the reference point in the frame passes the second point.

§46.2.3.6.2 MaxLatency – is defined as:

Shall be 3.118

> Latency shall use the definition of 3.102, with additional context as follows: The 'known reference point in the frame' is the message timestamp point specified in IEEE Std 802.1AS for various media (i.e. start of the frame). The 'first point' is in the Talker, at the reference plane marking the boundary between the network media and PHY (see IEEE Std 802.1AS). The 'second point' is in the Listener, at the reference plane marking the boundary between the network media and PHY.

46.2.3.6.2 MaxLatency – redefinition in case of Tspec time-aware is present

> When TSpecTimeAware is present:
>
> The 'first point' is assumed to occur at the start of the Interval, as if the Talker's offsets (EarliestTransmitOffset and LatestTransmitOffset of 46.2.3.5) are both zero.

2. Latency requirements for scheduled traffic at listeners (2)

TTTech

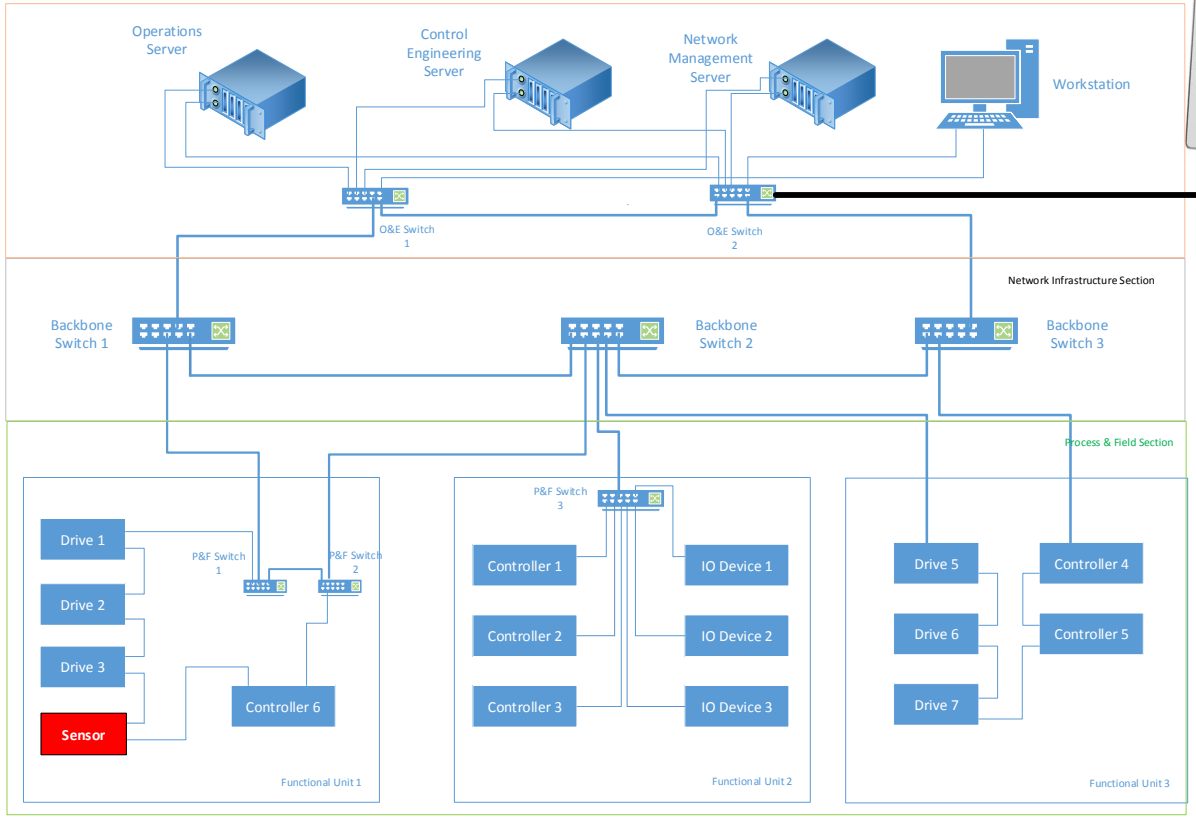
- *Latency definition* is “transformed” to *Deadline definition* for “time-aware” streams
  - With this definition, latency requirement in case of “time-aware” streams cannot be requested by the listener
  - *Latency constraints* and *Deadline constraints* are different values – they can be related if sending offset is known - but this is not known at the listeners
  - There are use cases when both requirements are needed (Deadline and Latency)

Proposal 2.a: Fix the text, to keep the *latency definition* as “latency“

Proposal 2.b: Introduce additional parameter for deadline requirement in case of time-aware Tspec at listeners

3 & 4. Latency and Redundancy requirements for talkers

- *Latency* and Number of Seamless Trees at talker(s)
  - Some listeners may have same latency and redundancy requirements some not (e.g., a monitoring HMI device)
  - In some use cases setting the redundancy and latency requirements at the talker may cause conflicts with the listener requests



Proposal 3:

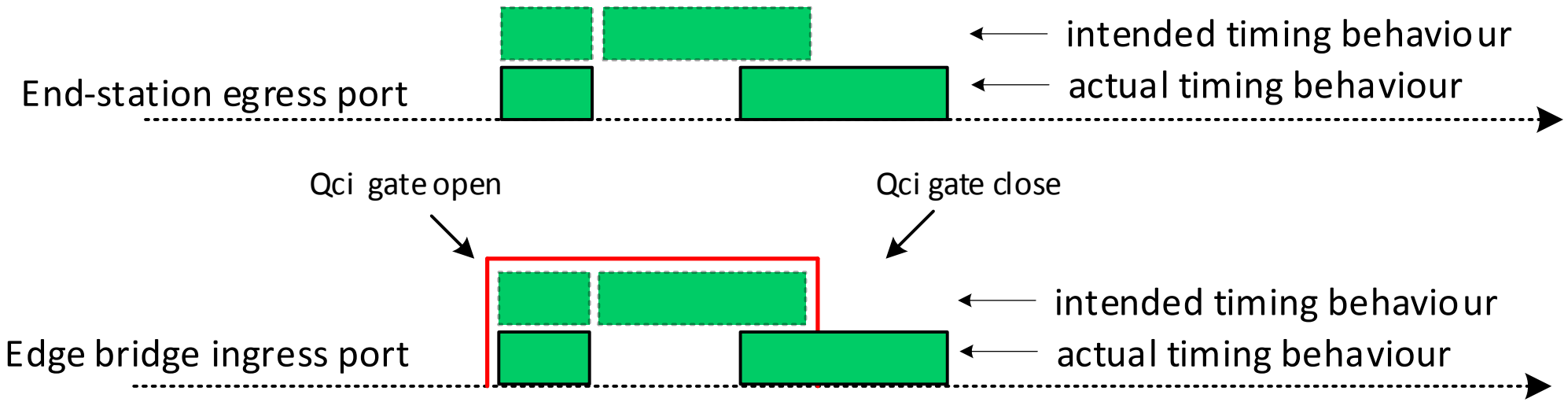
- **Either remove the latency requirement at the talker,**
- **or add informative text to explain the possible implications if used**

Proposal 4:

- **either remove the Number of Seamless Trees from the talker**
- **or add informative text to explain the possible implications if used**

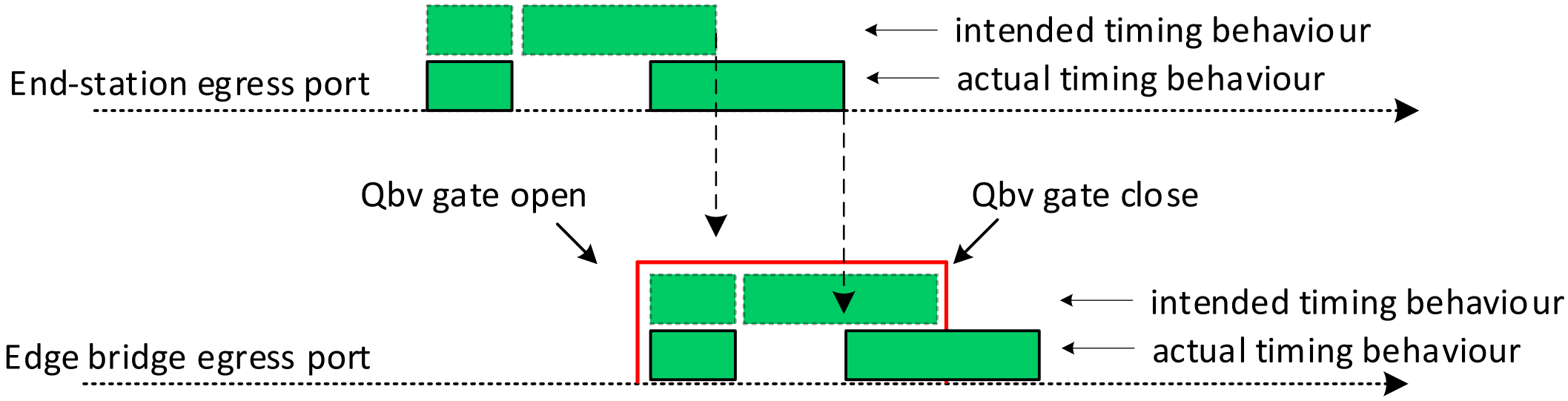
5. Maximum Transmission GAP between two successive frames

- Use case where the payload of a scheduled stream does not fit in 1 (one) Ethernet frame
  - Some end-stations will not be able to send the frames back-to-back (because of encoding, security,…)
  - The incoming time for these frames are important for calculation of the schedule
  - Without taking that into consideration, the Qci may police the frames



5. Maximum Transmission GAP between two successive frames (2)

- Without taking that into consideration, the CNC make generate bridge schedules that may lead that a frames from a scheduled stream miss a gating “slot”



Proposal 5:

- **Add a parameter to specify Maximum Transmission GAP between two successive frames as a part of end-station capabilities (constraints) – similar to the jitter parameter.**
- **If the parameter is not set, Missing info about this parameter – use standard value of IFG**

6. Status information on clock synchronization for devices that are “part” of a stream

- CUC needs a status information if the bridges are synchronized, in order to inform the talker(s) about the start of periodic transmission of scheduled traffic
- Otherwise, network can be brought in such a state that the latency requirements for scheduled streams cannot be meet, messages are stored in the bridge queues and the state cannot be recovered

Proposal 6:

- **Add a parameter to indicate the clock synchronization status of the Bridges along the path of the given datastream during start-up**

Next Steps

Open for a proposal how to proceed with the findings

- Option a): Maintenance requests
- Option b): new PAR

In case of new PAR – additional topics

TTTech

I. Addressing Qbu and Qci user requirements

- Details to be discussed in upcoming meeting/calls

II. CNC-CUC interface

- CUC – CNC interface (Parameters in Requests/Responses)
- Avnu: selecting RESTCON
  - Draft/Prototype API implementations

III. Topology Model

- More on that in the Bangkok meeting

IV. CNC – related topics

- CNC responsibilities
- Multiple CNC - initial work started at Avnu