

UNI Requirements: CNC Dynamics & End Station Capabilities

Rodrigo Ferreira Coelho [Siemens AG]

Contributors:

Günter Steindl [Siemens AG]

Contributions by Johannes Specht [self]

Motivation

Theses

1. **CNC must know the end station (ES) transmission capabilities** in order to know how streams are transmitted by ES and consequently to compute stream latencies and to configure the ES
2. **CNC needs the means to determine the transmission order of streams** at the end stations in order to avoid overly pessimistic stream latency computations

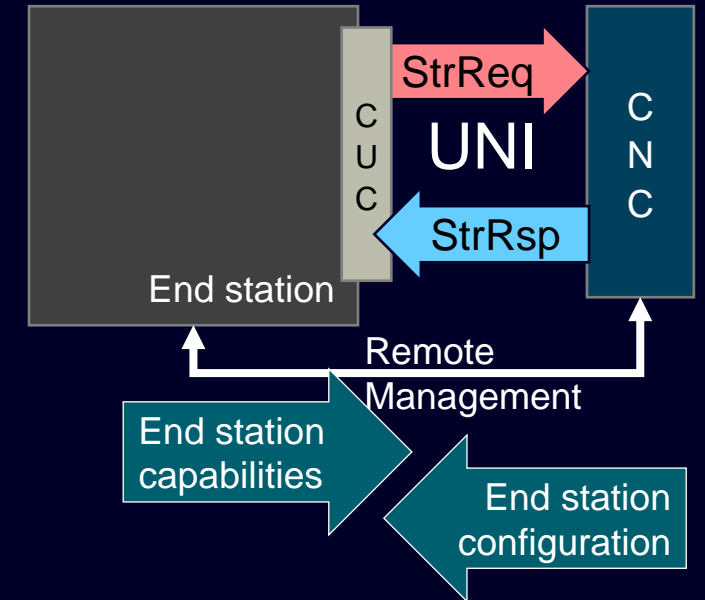
Similar requirements also presented in the contribution [new-specht-dev-caps-and-limits-1121-v01.pdf](#)

Thesis 1: CNC must know the end station (ES) transmission capabilities

CNC plans frame **transmission at end stations**

- **CNC requires end station capabilities in order to compute stream latency**
 - E.g. **streams sent as burst**, achievable **inter packet gap (IPG)**, **transmission time jitter**
- **Naïve CNC assumptions hold for all end stations?**
 - E.g. up to 255 streams back-to-back, minimal IPG
- End stations with limited SWaP* **do not cope** with these naïve **assumptions**
 - Yet, these products should also participate in the communication
 - Therefore, **thesis 1 proven**

* Size, Weight and Power

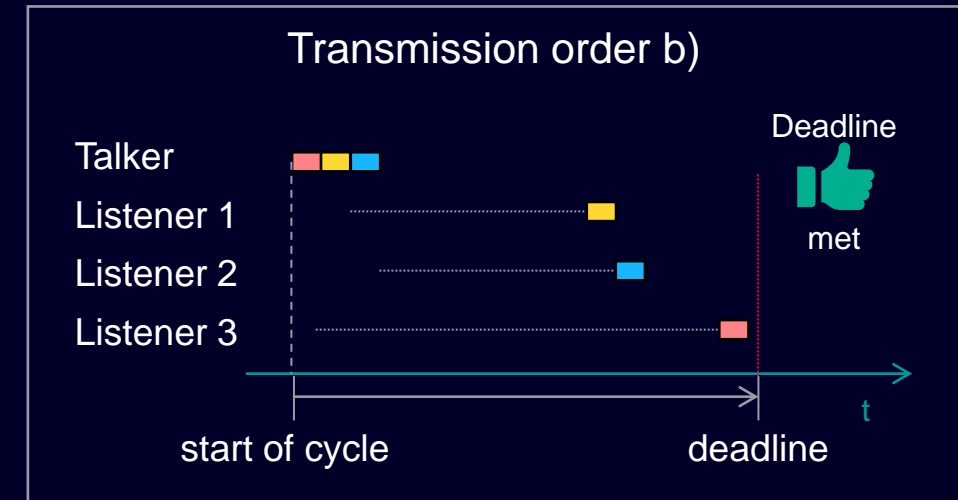
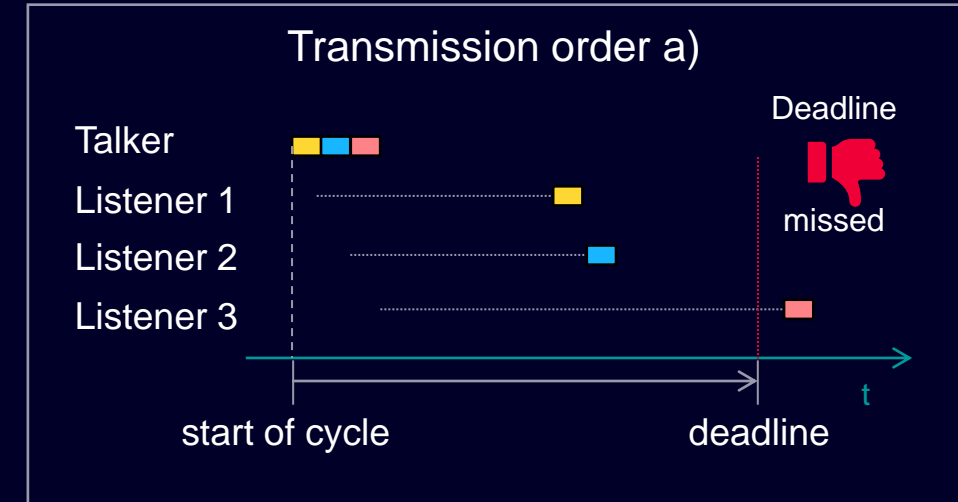


**Currently, CNC can only use naïve assumptions.
There is no means to identify end station's capabilities**

Thesis 2: CNC needs the means to determine the transmission order of streams at the end stations

Transmission time at end station impacts

- Latency computation
- Ignoring sending order leads to large latency/ deadline computation pessimism
- Frame buffer usage at bridges
- Qbv time window usage at bridges (if Qbv is used)



Thesis 2: CNC needs the means to determine the transmission order of streams at the end stations

Why transmission order matters for IA

- Scenario 1: a few streams transmitted per IA Station as burst

- Number of streams: 3
- Frame size: 400 octets
- Deadline range: 500 μ s to 64 ms
- Octet time: 8ns

Maximum delay at transmitting IA Station

- $2 \times 400 \times 8\text{ns} = 6.4 \mu\text{s}$

Transmission order irrelevant for very simple scenarios

- Scenario 2: many streams transmitted per IA Station as burst

- Number of streams: 256 (per gating cycle)
- Frame size: 400 octets
- Deadline range: 500 μ s to 64 ms
- Octet time: 8ns

Maximum delay at transmitting IA Station

- $255 \times 400 \times 8\text{ns} = 816.0 \mu\text{s}$

depending on the TX order, 500 μ s stream is rejected!

Transmission order matters for IA scenarios (described in IEEE/IEC 60802)

Thesis 2: CNC needs the means to determine the transmission order of streams at the end stations

According to Qcc, CNC can **only** determine the transmission **point in time** of **each requested stream** in a burst transmission if the following hold:

1. End station supports **per stream scheduling (time aware offset)**
2. CNC knows **all streams at calculation time (all at once)**
3. **Topology is fixed and known** when streams are requested



Current Qcc

- No means to modify transmission time of a scheduled stream

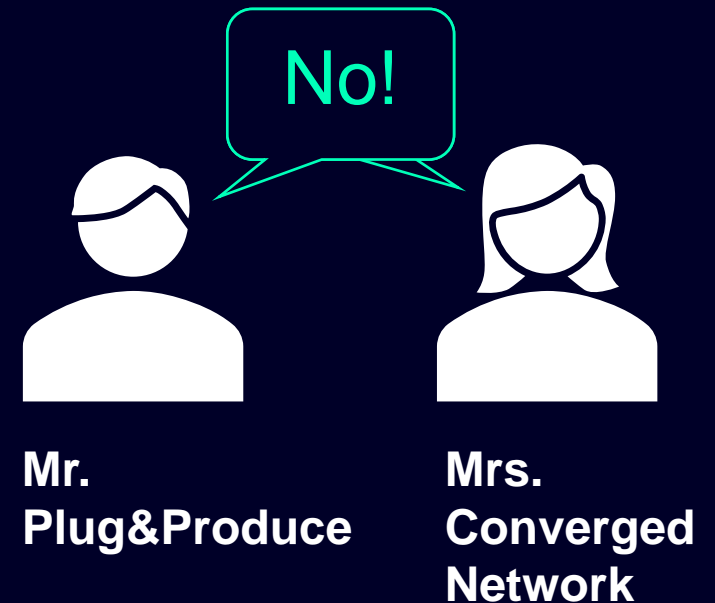
Thesis 2: CNC needs the means to determine the transmission order of streams at the end stations

According to Qcc, CNC can **only** determine the transmission **point in time** of **each requested stream** in a burst transmission if the following hold:

1. End station supports **per stream scheduling (time aware offset)**
2. CNC knows **all streams at calculation time (all at once)**
3. **Topology is fixed and known** when streams are requested

Are these CNC **assumptions** about **end stations** and **topology realistic** for **IA** use case?

- If not, current Qcc does **not allow** CNC to **determine a meaningful transmission order of streams** in a burst



Thesis 2: CNC needs the means to determine the transmission order of streams at the end stations

According to Qcc, CNC can **only** determine the transmission **point in time** of **each requested stream** in a burst transmission if the following hold:

1. End station supports **per stream scheduling (time aware offset)***
2. CNC knows **all streams at calculation time (all at once)**
3. **Topology is fixed and known** when streams are requested

Are these CNC **assumptions** about **end stations** and **topology realistic** for **IA** use case?

- If not, current Qcc does **not allow CNC to determine a meaningful transmission order of streams** in a burst
- Therefore, **thesis 2 proven**

* Per **stream scheduling not suitable** for dynamic network with large number of streams

Consequently, only **calculations** based on **pessimistic** transmission order are possible. Leading to, e.g.

- **Resource waste**
- **Less established streams**



Currently, there is no means for CNC to determine (i.e. inform via UNI) the order of transmitted frames in a burst

Summary

- Current Qcc assumptions on end station capabilities do not match a practical (real) end station
- Current Qcc assumptions on use cases do not match a practical (real) IA use case
- Both presented theses are proven
 1. CNC must know the end station (ES) transmission capabilities
 2. CNC needs the means to determine the transmission order of streams

Summary

- Missing pieces to be standardized, e.g. in Qdj
 1. Define a model for end station capabilities
 1. Define means for CNC to identify ESs capabilities
 2. Define the configuration parameters exchanged via UNI to allow for the CNC to determine the transmission order of streams
- Next contribution: textual
 - Cooperation with experts more than welcome



This amendment specifies procedures, interfaces, and managed objects to enhance the three models of ‘Time-Sensitive Networking (TSN) configuration’. It specifies enhancements to the User/Network Interface (UNI) to include new capabilities to support bridges and end stations in order to extend the configuration capability. This amendment preserves the existing separation between configuration models and protocol specifications. This amendment also addresses errors and omissions in the description of existing functionality.

The management models and User/Network Interface (UNI) already described in Clause 46: Time-Sensitive Networking (TSN) configuration of IEEE Std 802.1Q include only the concepts (e.g. in form of a YANG types module) for managing bridged LANs using Time-Sensitive Networking (TSN) features. In order to be able to fully manage such bridged LANs with TSN features, comprehensive interfaces and management modules are required that are currently not available. Enhancements are especially needed for the ‘fully centralized’ and ‘centralized network/distributed user’ configuration models. The proposed amendment will address these issues.

<https://1.ieee802.org/tsn/802-1qdj/>

| Further questions ?

| Contact

Dr. Rodrigo Ferreira Coelho

System Architect

DI FA CTR ICO ARC

Siemenspromenade 1

91058 Erlangen

Deutschland

Phone: +49 9131 17-45546

E-mail: rodrigo.ferreira_coelho@siemens.com