On 802.1Qcr D0.0

Johannes Specht, University of Duisburg-Essen

Overview

Purpose of this Slide Set

- Discuss outstanding technical issues from Annex Z
  - Explanation
  - Specific proposals …
    … but proposals will not be integrated if the 802.1 TSN group disagrees, agrees on different solutions for the issues, etc.
  - Discussions appreciated
- If we have time, discuss further technical issues
- If we have time, discuss rather editorial issues (not listed in this slide set)

Why?

- Decision points for the editor
- Affects further content, e.g. managed objects

Precision of the Shaper State Machine Algorithm

Cmp. http://www.ieee802.org/1/files/private/cr-drafts/d0/802-1Qcr-d0-0.pdf, clause Z.3

Current Algorithm in 802.1Qcr-D0.0

Description

- The Shaper State Machine algorithm isa C-like pseudo code at an abstract level (real numbers, etc.)
- Abstraction level in 802.1Q varies:
  - Credit-bases Shaper: Verbally explained
  - Enh. for Scheduled Traffic: State machines, encodings, etc.
  - …

```
ProcessFrame(frame) {
    lengthRecoveryDuration      = length(frame)/
                                  CommittedInformationRate;
    emptyToFullDuration         = CommittedBurstSize/
                                  CommittedInformationRate;
    shaperEligibilityTime       = BucketEmptyTime +
                                  lengthRecoveryDuration;
    bucketFullTime              = BucketEmptyTime +
                                  emptyToFullDuration;
    eligibilityTime             = max(arrivalTime(frame),
                                      GroupEligibilityTime,
                                      shaperEligibilityTime);

    if (eligibilityTime <= (arrivalTime(frame) + MaxResidenceTime)){
        // The frame is valid
        GroupEligibilityTime = eligibilityTime;
        BucketEmptyTime       = (eligibilityTime < bucketFullTime) ?
            lengthRecoveryDuration + BucketEmptyTime :
            lengthRecoveryDuration + eligibilityTime - emptyToFullTime;
        AssignAndProceed(frame,eligibilityTime);
    } else {
        // The frame is invalid
        Discard(frame);
    }
}
```

Observations

- If the algorithm remains abstract, implementations may do something fundamentally wrong (see next slide). If, on the other hand, the algorithm gets very specific, it could be misunderstood as a restriction on implementation freedom.
- Implementations *will* deviate from the shown shaper state machine code: Imprecision. In either case, it is necessary to introduce well defined and externally available imprecision parameters per device for timing analysis.
- However, it may be complex to specify imprecision parameters on the current abstraction level, because the space of possible implementations seems … quiet large.

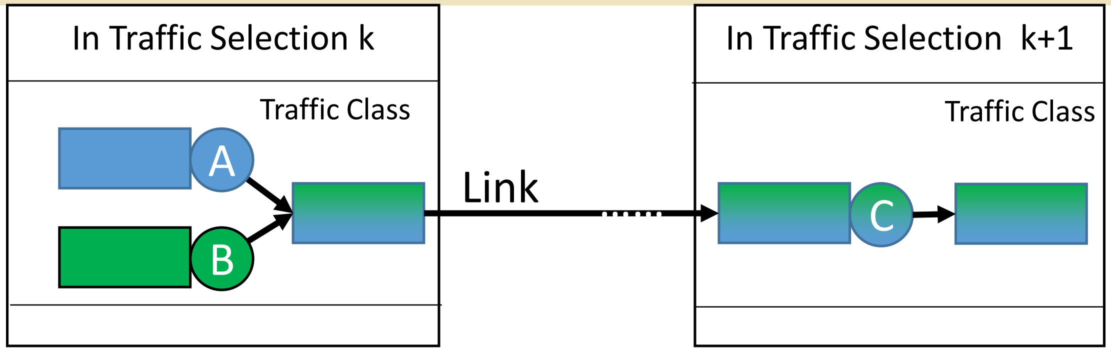
Example of an imprecise Implementation

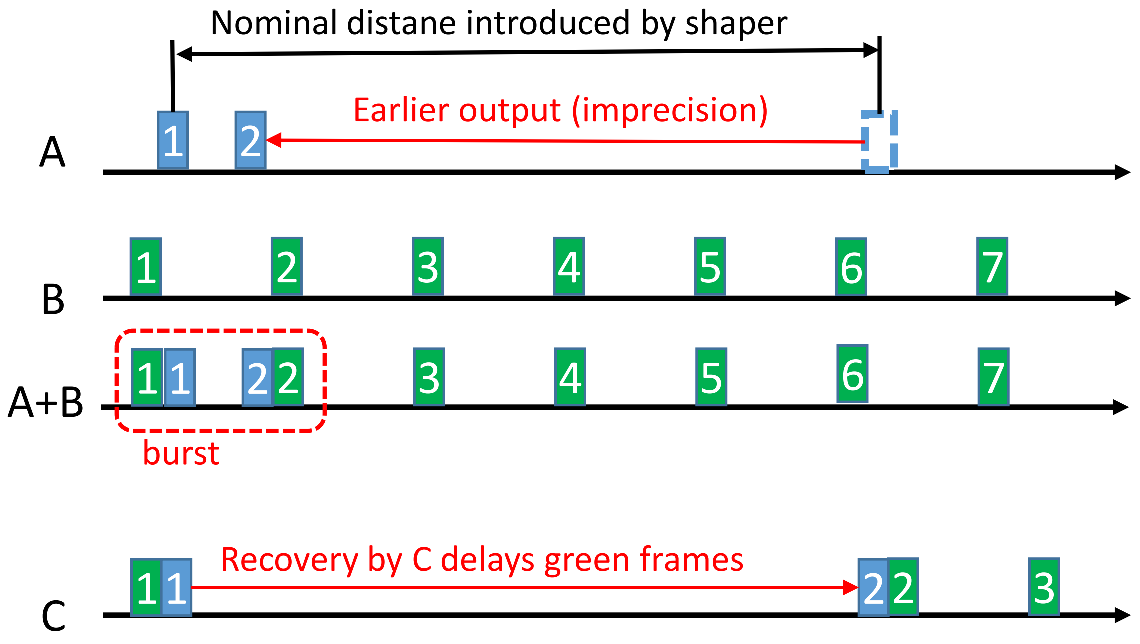
Description

- Two-level queuing
  (cmp. http://www.ieee802.org/1/files/public/docs2015/new-tsn-specht-ubs-queues-0521-v0.pdf)
  - Shaped queues
    (Nx per traffic class)
  - Unshaped queue per traffic class collecting the output from the shaped queues (1x per traffic class)
- Shaped Queues
  - A: *One* shaper state machine for blue streams, affected by imprecision
  - B: *One* shaper state machine for green streams, not affected by imprecision
  - C: *Two* shaper state machines, one for blue and one for green streams, not affected by imprecision

Observations

1. Frame #2 (blue) output happens earlier due to imprecision.
2. The *burst* in the unshaped queue (A+B) is larger than nominal.
   → **All lower priority traffic classes (not explicitly shown), including ATS classes, will experience a larger delay, because it takes longer until the unshaped queue drained.**
3. At C, the shaper state machine of blue streams recovers the nominal distance between frames #1 and #2.
   → **Green frames are shifted/experience unexpected large delays.**




Nominal distane introduced by shaper
Earlier output (imprecision)
A
1
2
B
1
2
3
4
5
6
7
A+B
1 1
2 2
3
4
5
6
7
burst
Recovery by C delays green frames
C
1 1
2 2
3

Example of an imprecise Implementation

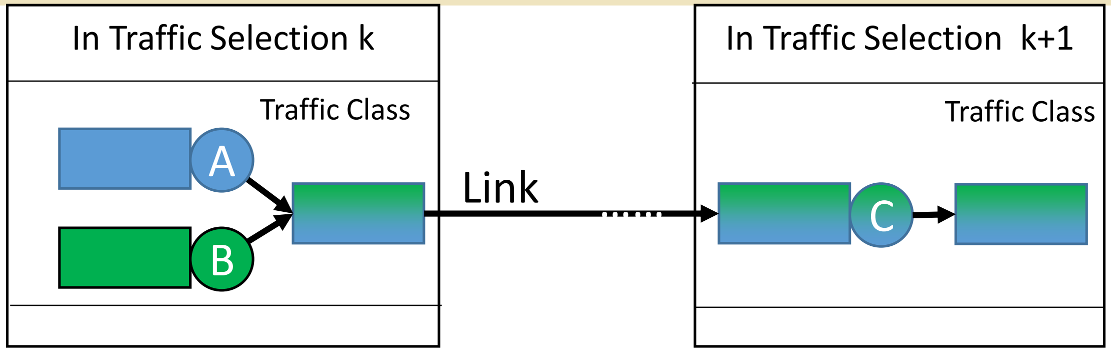
On a Higher Level…

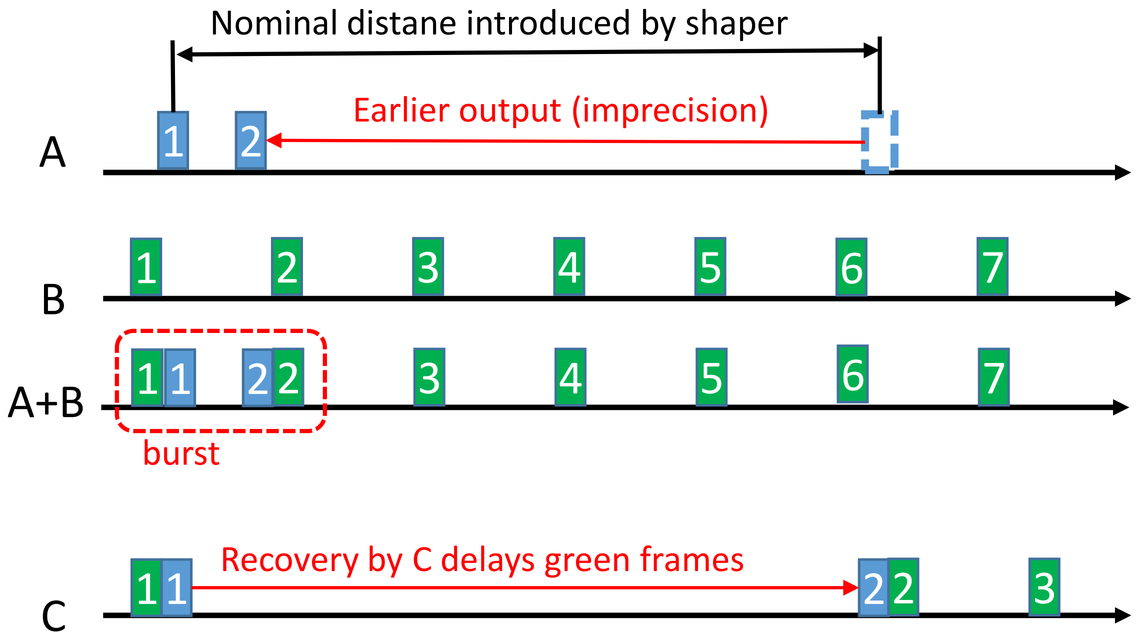
- The shown effect is non-accumulative, as long as shaped queues output frames with some jitter.
  I.e., this jitter would not prevent bounded delays, but the impact can get large if shaper state machine are implemented in a naïve manner.
- In contrast, If a shaped queue would outputs permanently to fast (>> committed information rate), calculating bounded delays gets hard to impossible. This comes to no surprise:
  If, for example, the shaper state machine belonging to A would output blue frames with 10 Mbit/s, AND
  the shaper state machine belonging to C would output blue frames with 5 Mbit/s, the result would be equal to a broken network configuration.
- Nonetheless, the author is unsure whether a set of well-defined imprecision parameters that fits all potential implementations can be found easily.

Proposal/Discussion

Unless decided otherwise by the TSN group:

1. Specify the shaper state machine algorithm more explicit, closer to implementation, and for which the author believes the negative impact on delay bounds is low.
2. Derive imprecision parameters referring to this algorithm



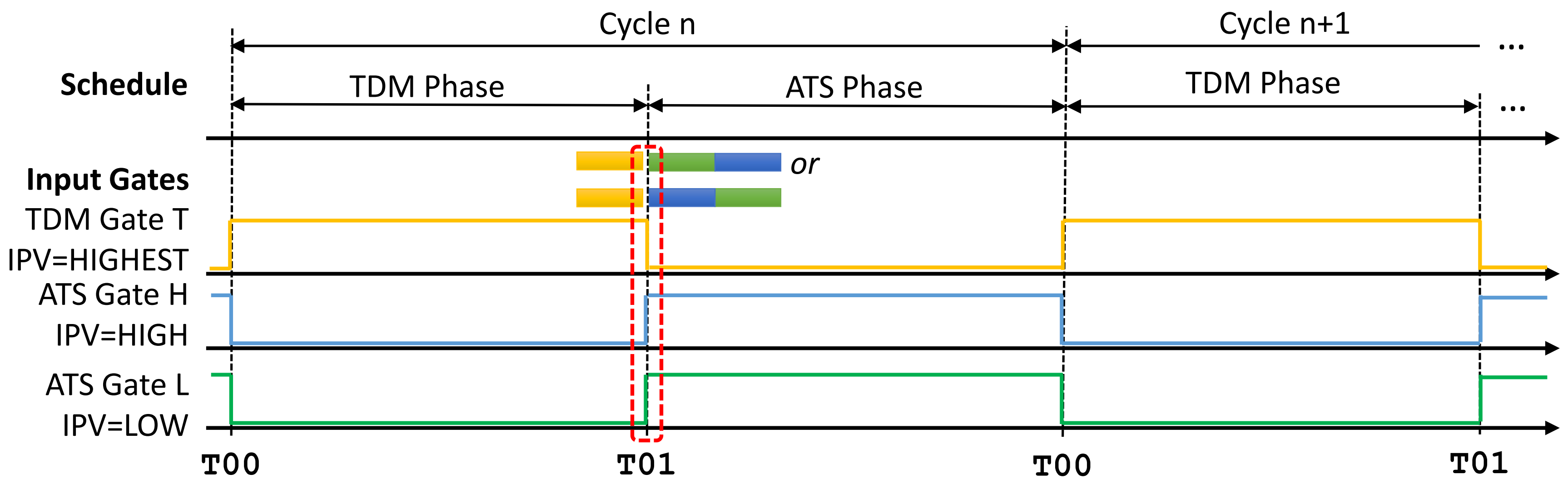


Nominal distane introduced by shaper
Earlier output (imprecision)
A
1
2
B
1 2 3 4 5 6 7
A+B
1 1 2 2 3 4 5 6 7
burst
Recovery by C delays green frames
C
1 1 2 2 3

Opening and Closing Input Gates

Cmp. http://www.ieee802.org/1/files/private/cr-drafts/d0/802-1Qcr-d0-0.pdf, clause Z.1

Closing ATS Gates



Gate T control list
```
T00: o,HIGHEST
T01: c,HIGHEST
```

Gate H control list
```
T00: c,HIGH
T01: o,HIGH
```

Gate L control list
```
T00: c,LOW
T01: o,LOW
```

Description

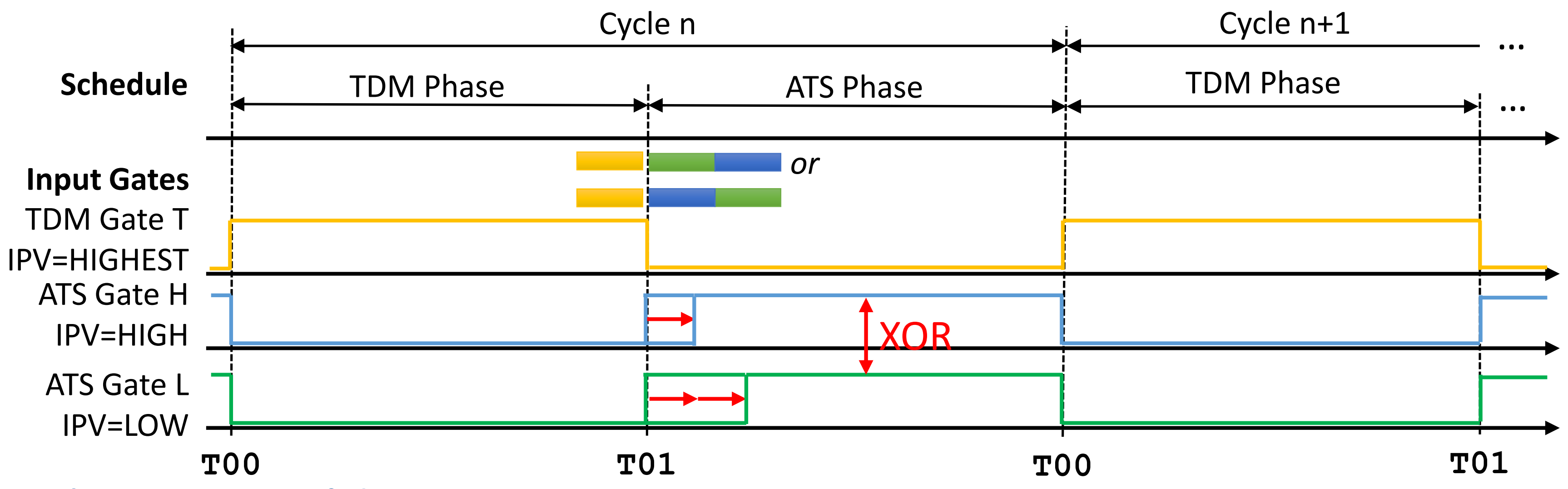
- In D0.0, ATS uses Input Gates for Transmission Priority Level Assignments via IPV: Each IPV requires a dedicated Input Gate
- Example of a mixed TDM and ATS setup with TDM and ATS phases:
  - HIGHEST priority level: Strict Priority Transmission Selection (TDM)
  - HIGH and LOW priority level: ATS Transmission Selection (2 Traffic Classes)
- **A**TS is **A**synchronous:
  - The first ATS frame in an ATS phase is either IPV=HIGH or IPV=LOW, but this cannot be pre-determined

  → Two Input Gates (H and L) must be opened (or closed) in ATS phases simultaneously, while simultaneously Gate T is closed (or opened)
- **Important**: Input Gates, as specified in 802.1Qci, allow this!
  Each gate has a separate control list, multiple control lists can contain events at the same times ***However …***

The Issue

UNIVERSITÄT DUISBURG ESSEN

Open-Minded



Gate T control list
```
T00: o,HIGHEST
T01: c,HIGHEST
```

Gate H control list
```
T00: c,HIGH
T01: o,HIGH
```

Gate L control list
```
T00: c,LOW
T01: o,LOW
```

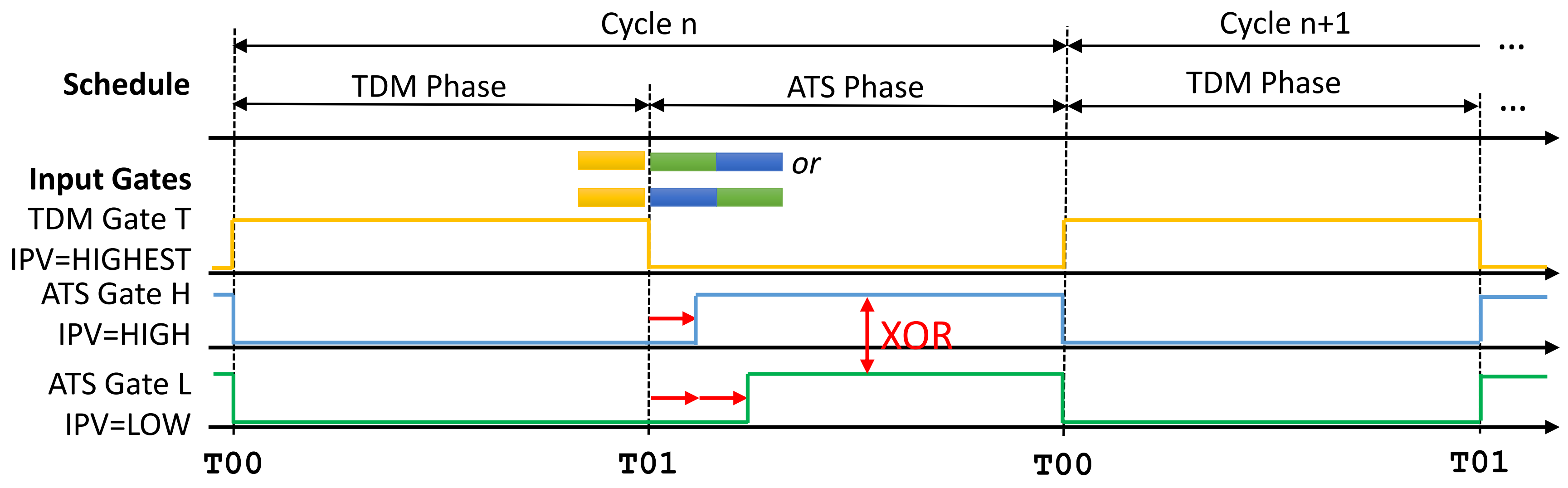
Implementation Specific?

- Implementations may exploit the fact that, in a pure TDM setup with multiple Traffic Classes (no ATS), only a very limited number of Input Gates is opened simultaneously
- Example 1
  - 1 Input Gate is open at a time, all other Input Gates are closed
  - Gate ID can be encoded in *log2(Max. Gate ID)* Bits
  - **Problem:** Impossible to have several (2 in the example) simultaneously opened gates
- Example 2
  - Explicit open and close events for a single gate, but no simultaneous open and close events
  - **Problem:** Event processing will take some time (`d`) - no simultaneous gate transitions.
    → false positives, false negative, both

Merged gate control list (Ex. 1)
```
T00: T=o,HIGHEST
T01: H=o,HIGH     ; L=o,LOW
```

Merged gate control list (Ex. 2)
```
T00:      T=o,HIGHEST
T00+1*d: H=c,HIGH
T00+2*d: L=c,LOW
T01:      T=c,HIGHEST
T01+1*d: H=o,HIGH
T01+2*d: L=o,LOW
```

Proposal



Gate T control list

```
T00: o,HIGHEST
T01: c,HIGHEST
```

Gate H control list

```
T00: c,HIGH
T01: o,HIGH
```

Gate L control list

```
T00: c,LOW
T01: o,LOW
```

Merged gate control list (Ex. 1)

```
T00: T=o,HIGHEST
T01: H=o,HIGH     ; L=o,LOW
```

Merged gate control list (Ex. 2)

```
T00:      T=o,HIGHEST
T00+1*d: H=c,HIGH
T00+2*d: L=c,LOW
T01:      T=c,HIGHEST
T01+1*d: H=o,HIGH
T01+2*d: L=o,LOW
```

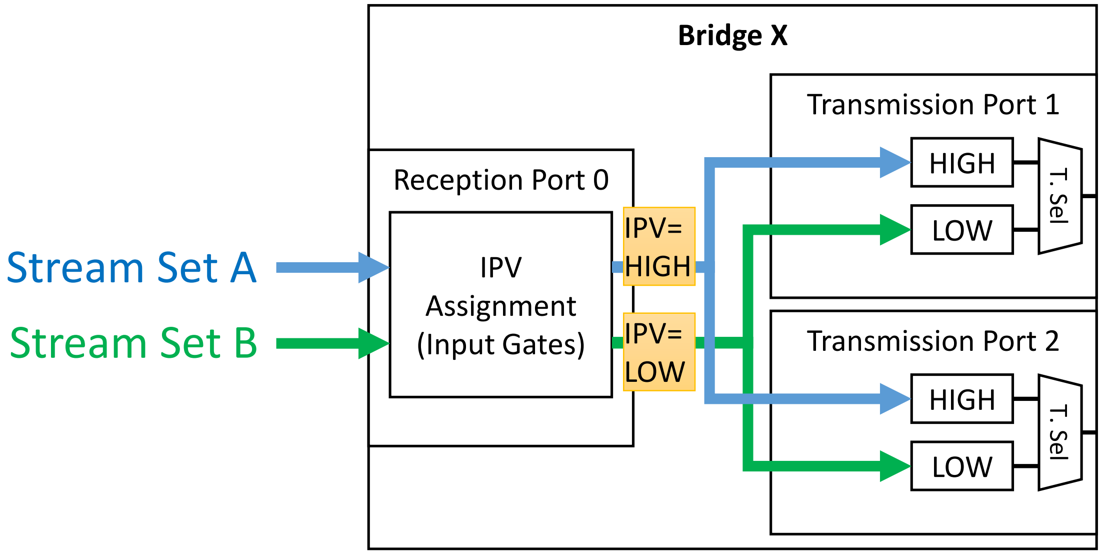
Observations, Options and Discussions

- This issue is implementation specific, as said - the Input Gate model in the standard seems ok to the author!
- However, is it nonetheless considered reasonable to address it?
- If Yes, there could be (at least) two Options:
  - **Option #1**
    Insert normative content in clause 8.6.5.2 (ATS), which ensures that, if multiple Gate Control Lists contain events with identical associated times, then all these events become effective simultaneously.
  - **Option #2**
    The proposed solution for Issue Z.2 (read on…)

Transmission Priority Assignment with Input Gates

Cmp. http://www.ieee802.org/1/files/private/cr-drafts/d0/802-1Qcr-d0-0.pdf, clause Z.2

Transmission Priority Assignment



From IEEE 802.1Qcr-D0.0

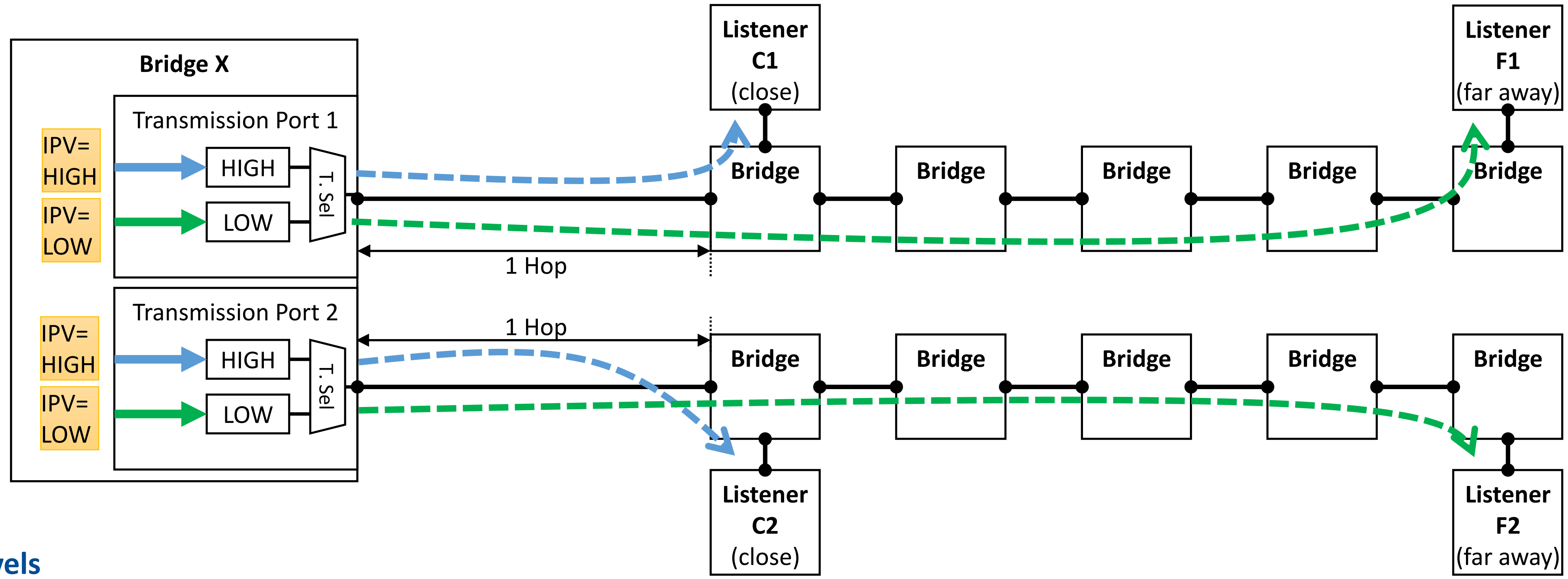
- IPV assigned to frames by PSFP Input Gates for transmission (cmp. San Antonio 2016). I.e., IPV assigned in reception ports.
- If traffic, received from one reception port are transmitted by multiple transmission ports (multi-cast, P802.1CB, etc.), all transmission ports transmit at the same priority level (Traffic Class)

Delays in a Two-Priority Setup

Stream Set A = Stream Set B
- 10 identical Streams per set
- Per Stream: Max. 125 Byte/Frame
- Per Stream: 25 Mbit/s per Stream
- Best Effort: Max. 125 Byte/Frame
- Links: 1 Gbit/s

Max. Per-Hop Delay @ Tx. Port 1
- Stream Set A: 12.0 µs
- Stream Set B: 27.7 µs

Max. Per-Hop Delay @ Tx. Port 2
- Stream Set A: 12.0 µs
- Stream Set B: 27.7 µs



Delays and Transmission Priority Levels

- Transmission Priority Levels affect max. delay bounds experienced along the path:
  - Higher transmission priority level → smaller delay bound
  - Lower transmission priority level → greater delay bound

  For math background, see http://ieeexplore.ieee.org/document/7557870/
- Transmission Priority Level Assignment are the "knobs" to control per Hop Delays and to satisfy end-to-end deadlines

The Issue

Stream Set A = Stream Set B

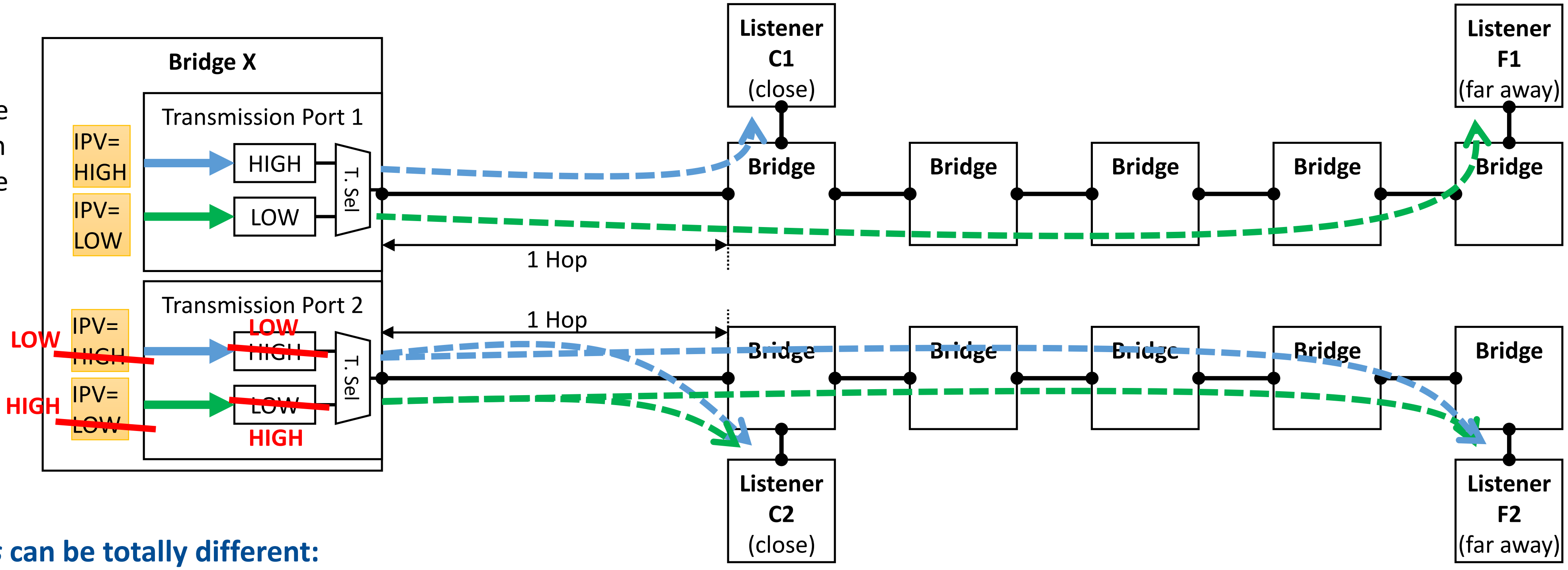
- 10 identical Streams per set
- Per Stream: Max. 125 Byte/Frame
- Per Stream: 25 Mbit/s per Stream
- Best Effort: Max. 125 Byte/Frame
- Links: 1 Gbit/s

Max. Per-Hop Delay @ Tx. Port 1

- Stream Set A: 12.0 µs
- Stream Set B: 27.7 µs

Max. Per-Hop Delay @ Tx. Port 2

- Stream Set A: ~~12.0 µs~~ **27.7 µs**
- Stream Set B: ~~27.7 µs~~ **12.0 µs**



End to end deadline *requirements* can be totally different:

- In the shown example, Stream Sets A and B exchanged their Paths at Port 2
- Now, it makes more sense to send Set A with LOW priority and Set B with HIGH priority, but only at Port 2
- This would require two different IPVs to be assigned: One for Port 1 and another one for Port 2
- Note: The author believes that this is issue is not limited to ATS and affects other traffic types as well.

Proposal

Assumption

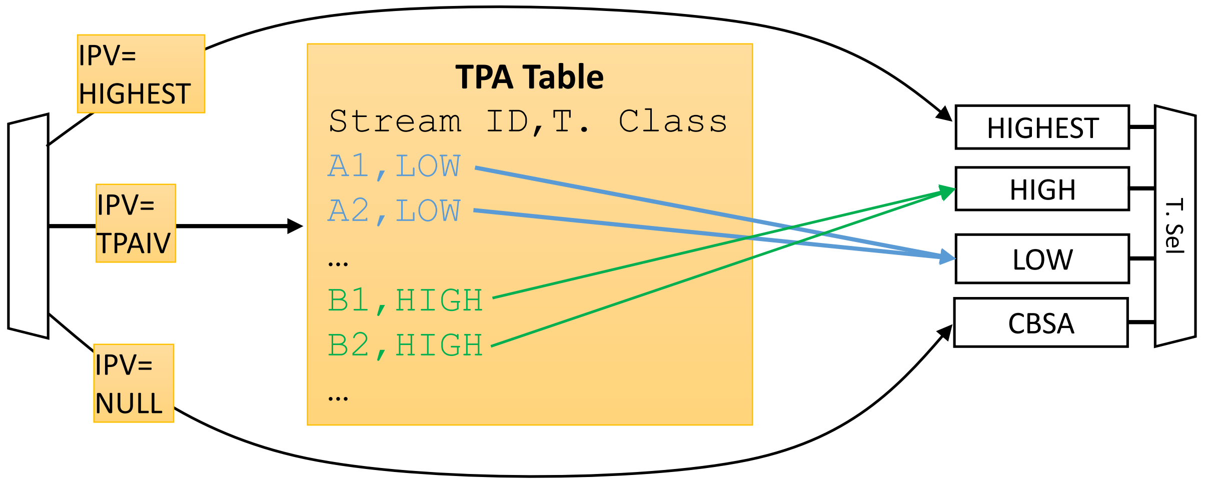
- Solutions desired? **Yes**
- Make it Mandatory? **No**
  IPV assignment by Input Gates works, **but** puts higher constraints on the user side (i.e., implies lower performance). In addition, devices that support only a single ATS traffic class do not need this.

Potential Solution

1. A new special IPV value, <u>co-existent</u> with the null value of PSFP (clause 8.6.5): *Transmission Priority Assignment Indicator Value* (TPAIV)
   - Symbolic value (implementation-specific encoding)
   - Implementations *may* support this

   Re-using the null value is not possible: Assume ATS co-exists with other traffic classes (e.g., strict priority class with highest priority for TDM traffic). Transmission Priority Assignment for ATS would fallback to the Tag-based traffic class mapping (802.1Q, table 8-4). However, priority levels of ATS traffic are changed along the path, ATS frames would be forwarded to the wrong classes in several cases.

2. An optional Transmission Priority Assignment Table (TPA Table, clause 8.6.6)
   - Per transmission port
   - Maps stream ID → Traffic Class ID
   - **if** IPV(frame) == TPAIV
     - **then**: Traffic Class = <*Transmission Priority Assignment Table lookup for Stream ID*>
     - **else**: Traffic Class = IPV(frame)



Example for illustration

Implications …

… from an Implementer Perspective

- If more than one ATS transmission priority level are supported (>1 ATS traffic class) **AND** per transmission port priority levels are supported:
  - Stream ID has to be carried from reception ports to transmission ports
  - However, this seems be already required for 802.1CB capable bridges anyways
- If at most one ATS transmission priority level is supported (=1 ATS traffic class):
  - Case IPV=TPAIV:
    - The Transmission Priority Assignment Table would contain the same value (traffic class ID) for all keys (Stream ID)
    - The Transmission Priority Assignment Table effectively not needed!
  - Case (IPV /= TPAIV) AND (IPV /= NULL):
    - IPV = Traffic Class ID for
    - The Transmission Priority Assignment Table effectively not needed!

Implications …

… from a User Perspective

- If more than one ATS transmission priority level are supported (>1 ATS traffic class) AND per transmission port priority levels are supported
  - On, more, or even all Streams can get be transmitted in different priority levels per transmission port
- In addition, if all Input Gates set (IPV=TPAIV) OR (IPV=N), with N being a single transmission traffic class
  - Each Input Gate can now correspond to one ATS phase in a cycle
  - Input Gates for ATS can be opened and closed in ATS phases in a mixed TDM+ATM Setup
  - Unexpected ATS frames in TDM phases (and vice versa) can be detected and trigger further error handling reaction
  - ➔ ***This would not only solve issue Z.2, but also issue Z.1!***

Co-existence of Flow Meters and Shaper Instances

Meters vs. Shapers

Description

- Meter ID are effectively replaced with a Shaper ID in Stream Filters.
- Shaper Instances operate on the same level as Meters would do.

What is not explicitly stated (current formulations are intentionally inaccurate)

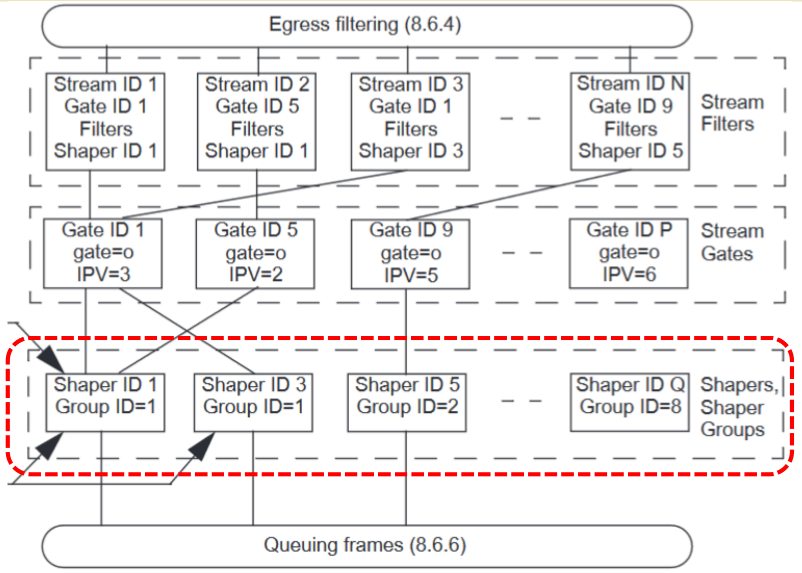
- Can Meters and Shapers co-exist?
- I.e., shall it be possible to process frames by Meters, followed by processing by Shaper State Machines?

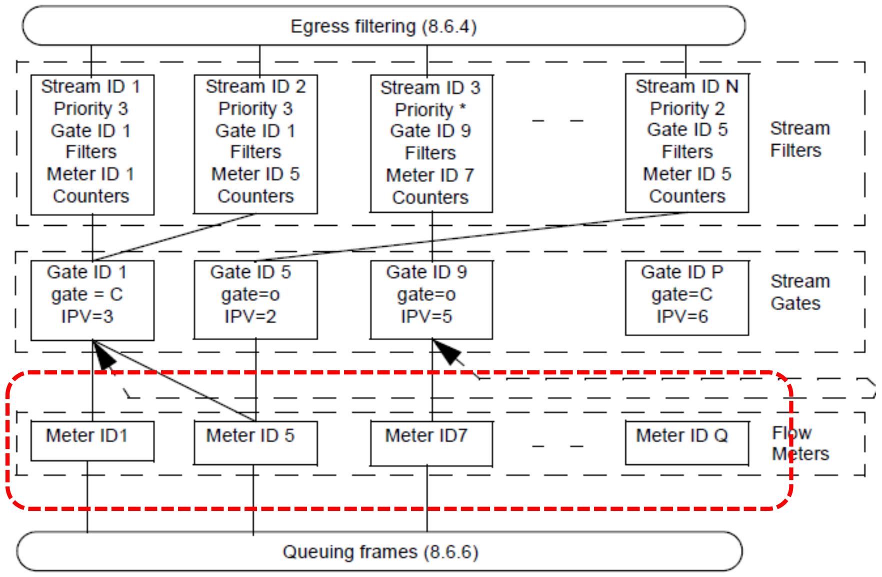
Observations

- Shaping can be considered as a stronger form of Metering:
  - Meters drop frames that violate the configured traffic envelope
  - Shapers would introduce delays to comply with the configured traffic envelope.
- Co-existence would allow to, for example, meter subsets streams that are processed by one Shaper. If co-existence is not desired, then this is not possible.

Proposal

- Unless decided differently by the TSN group, it remains inaccurate with a tendency against co-existence. The author can change this during the course of the project, but would appreciate presentations about use-cases for co-existing Meters and Shapers.





Egress filtering (8.6.4)
Stream ID 1 Priority 3 Gate ID 1 Filters Meter ID 1 Counters
Stream ID 2 Priority 3 Gate ID 1 Filters Meter ID 5 Counters
Stream ID 3 Priority * Gate ID 9 Filters Meter ID 7 Counters
Stream ID N Priority 2 Gate ID 5 Filters Meter ID 5 Counters
Stream Filters
Gate ID 1 gate = C IPV=3
Gate ID 5 gate=o IPV=2
Gate ID 9 gate=o IPV=5
Gate ID P gate=C IPV=6
Stream Gates
Meter ID1
Meter ID 5
Meter ID7
Meter ID Q
Flow Meters
Queuing frames (8.6.6)

Talker/End Station Behavior

Talker/End Station Behavior

From IEEE 802.1Qcr-D0.0

- [clause 8.6.5.2] *Bridges and end stations that support ATS shall provide filtering and assignment functions to allow subsequent queuing and transmission selection decisions (8.6.6.2, 8.6.8.4).*
- [clause T.2, Editor‘s Note] *This clause is intended to describe the transmission behavior of talkers in order to comply with ATS in bridges. Clause 8.6.5.2 specifies that end stations implement the specified functions, but (technically) it is likewise possible to support further traffic types sent by end stations. Which of this content should be normative is subject to discussions in the TSN group.*

Observation

- For end stations, statement 8.6.5.2 can easily be misunderstood with (speaking technically) high probability, given that clause 8.6.5.2 is located in reception ports.
- Clause T.2 is a better place to describe the talker behavior. However, clause T.2 is in an informative annex (i.e., no normative content).
  → Annex B (PICS proforma – End Station Implementations) cannot refer it.

Proposal

- For the moment, and unless decided differently by the TSN group, talker behavior is collected in T.2. This must not be the final decision – it should not be too problematic (or is it?) to move the content of T.2 to a new normative clause.

Thank you for your Attention!

Questions, Opinions, Ideas?

Johannes Specht
Dipl.-Inform. (FH)

Dependability of Computing Systems	Schuetzenbahn 70
Institute for Computer Science and	Room SH 502
Business Information Systems (ICB)	45127 Essen
Faculty of Economics and	GERMANY
Business Administration	T +49 (0)201 183-3914
University of Duisburg-Essen	F +49 (0)201 183-4573

Johannes.Specht@uni-due.de
http://dc.uni-due.de