On ATS

Johannes Specht (Self)

October 13, 2021

Introduction

Simple questions

- What is the best TSN shaper?
- Is ATS or TAS better for me?
- …

Giving Answers is tough

- What is your topology, in detail?
- What is your traffic, in detail?
  Path, pattern, quantitative and qualitative requirements of every stream!
- How dynamic is traffic and topology during runtime?
- How much planning and/or computation is ok?
- Do end station applications like the network timing?
- What is your Bridge failure model?
  They never fail, fail-silent only, or in a malicious manner?
- …

This Session

- Encourage discussion
  - Discussion ATS and other “TSN Shapers”
  - Discussion of aerospace use-cases

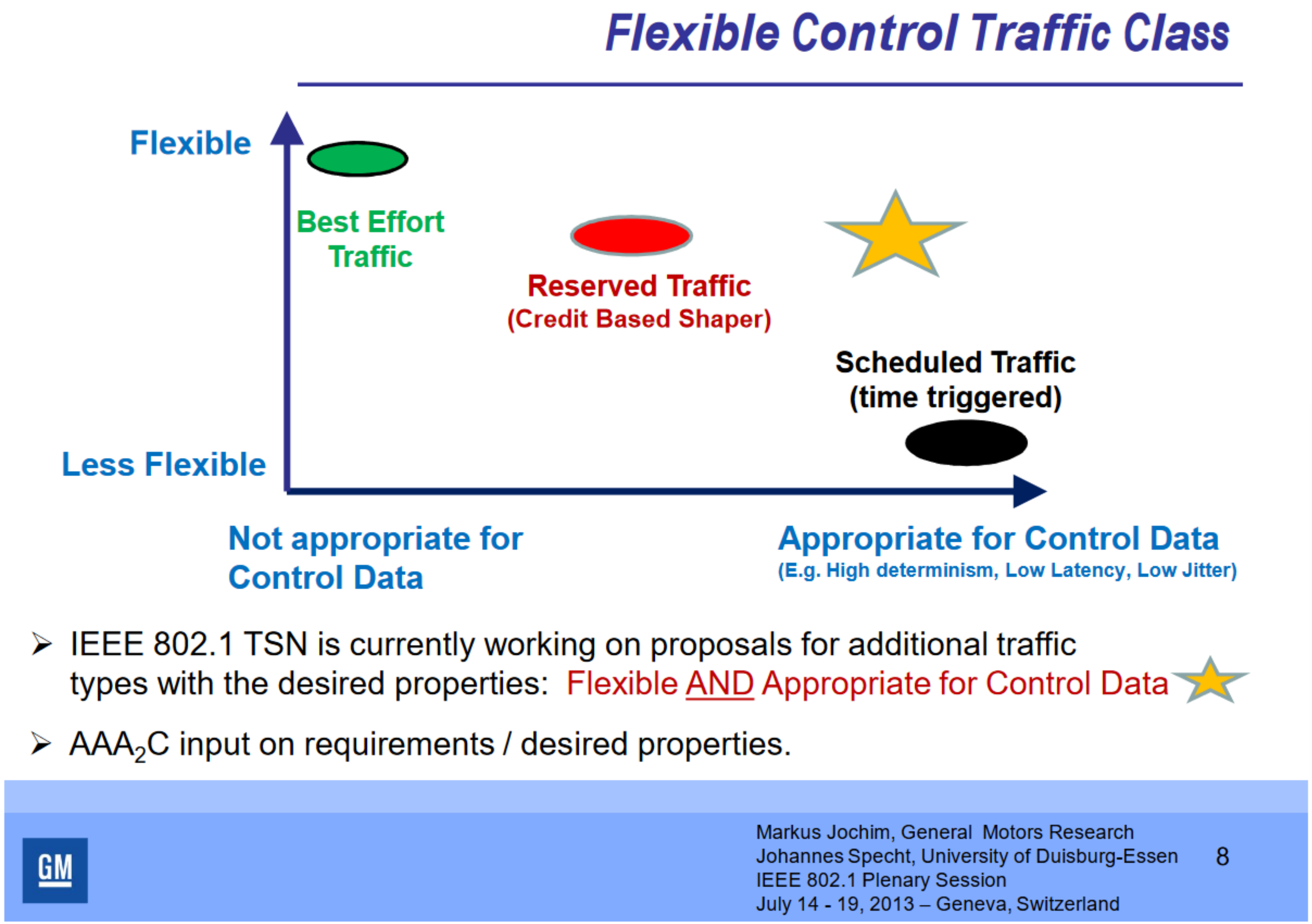
→ **Please just ask questions, interrupt me/add yourself to the queue, etc.**

This Slide Set

- Put ATS in context
- Properties of ATS/first thoughts on aerospace traffic
- ATS Math
- Explicit Pointers/References
  (in addition, look for “specht”, “ubs”, or “ats” in https://www.ieee802.org/1/files/public/docs2013 through https://www.ieee802.org/1/files/public/docs2021)

ATS Context

Background & Motivation: UBS → ATS



Flexible Control Traffic Class

Flexible

Less Flexible

Best Effort Traffic

Reserved Traffic
(Credit Based Shaper)

Scheduled Traffic
(time triggered)

Not appropriate for Control Data

Appropriate for Control Data
(E.g. High determinism, Low Latency, Low Jitter)

- IEEE 802.1 TSN is currently working on proposals for additional traffic types with the desired properties: Flexible AND Appropriate for Control Data ★
- AAA_2C input on requirements / desired properties.

Markus Jochim, General Motors Research
Johannes Specht, University of Duisburg-Essen
IEEE 802.1 Plenary Session
July 14 - 19, 2013 – Geneva, Switzerland

8

Source: https://www.ieee802.org/1/files/public/docs2013/new-tsn-jochim-aaa2c-requirements-for-control-traffic-0713-v01.pdf

The Standardized “TSN Shapers” @ Zero Congestion Loss

→ No “one size fits all”

- ***Different shapers are optimized for different areas in a multi-dimensional problem space***
- ***Performance Requirements, Reliability Requirements, Network Layout, etc.***

Shaper Std. and usage	**Bandwidth Efficiency** Converged Traffic	**Latency Bounds**	**Jitter Bounds**	**Global Clock Sync. Dependency**	**Configuration Complexity**	**Protection & Isolation** Per-stream filtering and policing (802.1Qci-2017)
Credit-based Shaper (CBS) IEEE Std 802.1BA-2011/AVB w. MSRP	High	High	Loose	No	Low, Dynamic	Loose
Time-Aware Shaper (TAS) IEEE Std 802.1Qbv-2016 /TDM & Zero Interference/Sync. Apps	Low	Ultra Low	Ultra Tight	Yes	High, Static	Tight
Cyclic Queuing and Forwarding (CQF) IEEE Std 802.1Qch-2016	Low	Medium	Tight	Yes	Low, Dynamic	Loose
Asynchronous Traffic Shaping (ATS) IEEE Std 802.1Qcr-2020	High	Medium	Loose	No	Low, Dynamic	Tight
Strict Priority (SP) IEEE Std 802.1Q, static usage	Medium	Medium	Loose	No	High, Static	Loose
Strict Priority (SP) IEEE Std 802.1Q, with a priori bounds	Medium	Medium	Loose	No	Medium, Dynamic	Loose

Results of abstraction, individual experiences, systems/use-cases in mind, etc.:

Ask *M* people to insert values, get *M* different tables! Ask *N* people for the important columns, get *N* sets of columns!

Interpretations, conclusions, translations to numbers (e.g., “Medium” v. “Medium”) are impossible without knowing all background details from the author!

The Standardized “TSN Shapers” @ Zero Congestion Loss

→ No “one size fits all”

- ***Different shapers are op in a multi-dimensional problem space***
- ***Performance Requirem ts, Network Layout, etc.***

Found in many, maybe all “TSN Switches”

Shaper Std. and usage	**B Conv raffic**	**nds**	**Jitter Bounds**	**Global Clock Sync. Dependency**	**Configuration Complexity**	**Protection & Isolation** Per-stream filtering and policing (802.1Qci-2017)
Credit-based Shaper (CBS) IEEE Std 802.1BA-2011/AVB w. MSRP	High	High	Loose	No	Low, Dynamic	Loose
Time-Aware Shaper (TAS) IEEE Std 802.1Qbv-2016 /TDM & Zero Interference/Sync. Apps	Low	Ultra Low	Ultra Tight	Yes	High, Static	Tight
Cyclic Queuing and Forwarding (CQF) IEEE Std 802.1Qch-2016			ight	Yes	Low, Dynamic	Loose
Asynchronous Traffic Shaping (ATS) IEEE Std 802.1Qcr-2020	High	Medium	Loose	No	Low, Dynamic	Tight
Strict Priority (SP) IEEE Std 802.1Q, static usage	Medium	Medium	Loose	No	High, Static	Loose
Strict Priority (SP) IEEE Std 802.1Q, with a priori bounds		Medium	Loose	No	Medium, Dynamic	Loose

If TAS is supported, and PSFP is *properly* implemented, CQF is supported.

Focus of this slide set

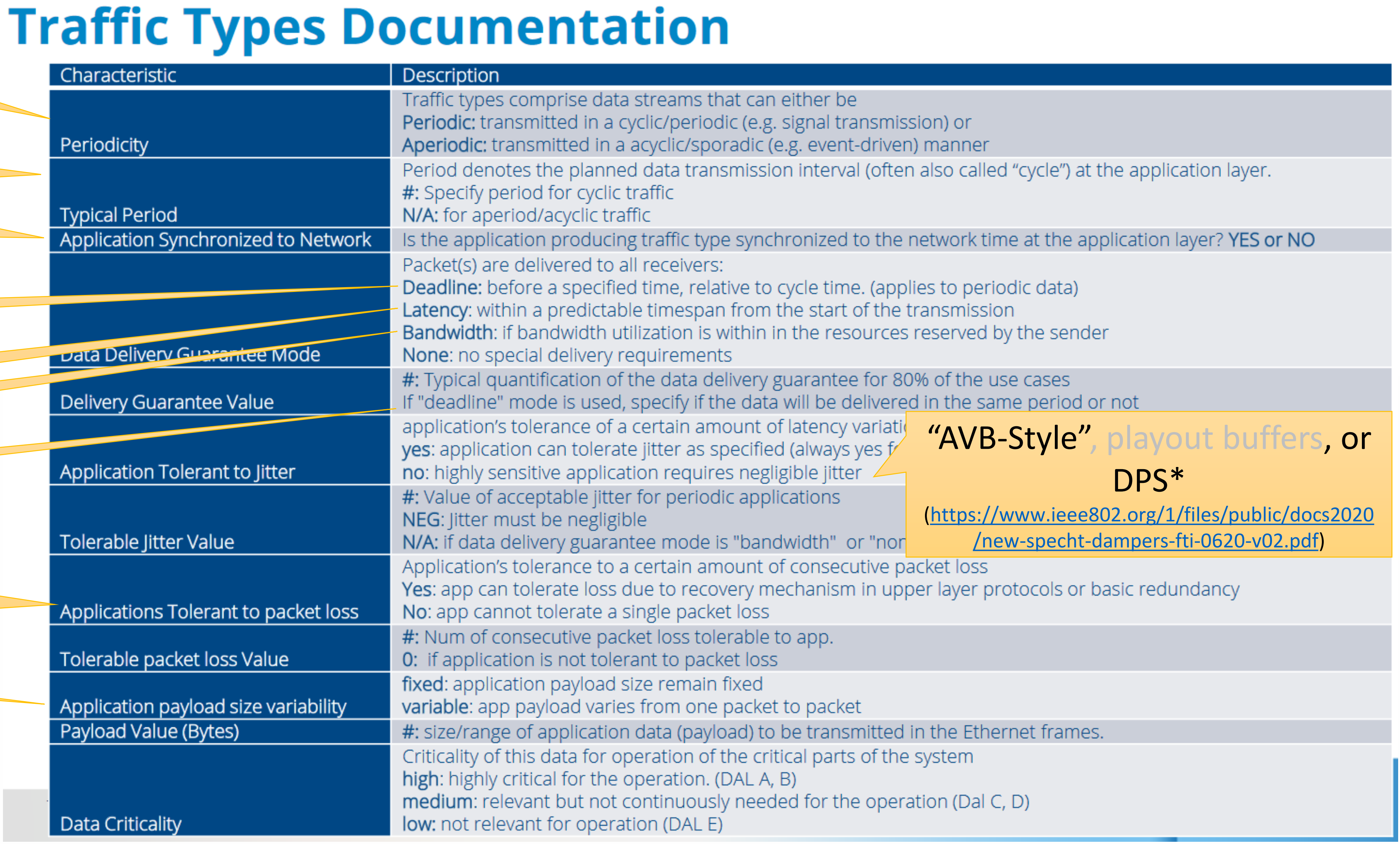
Always present

Results of abstraction, individual experiences, systems/use-cases in mind, etc.:

Ask *M* people to insert values, get *M* different tables! Ask *N* people for the important columns, get *N* sets of columns!

Interpretations, conclusions, translations to numbers (e.g., “Medium” v. “Medium”) are impossible without knowing all background details from the author!

Upfront, first thoughts: ATS for Aerospace Traffic?



Traffic Types Documentation

Characteristic	Description
Periodicity	Traffic types comprise data streams that can either be **Periodic:** transmitted in a cyclic/periodic (e.g. signal transmission) or **Aperiodic:** transmitted in a acyclic/sporadic (e.g. event-driven) manner
Typical Period	Period denotes the planned data transmission interval (often also called “cycle”) at the application layer. **#:** Specify period for cyclic traffic **N/A:** for aperiod/acyclic traffic
Application Synchronized to Network	Is the application producing traffic type synchronized to the network time at the application layer? **YES or NO**
Data Delivery Guarantee Mode	Packet(s) are delivered to all receivers: **Deadline:** before a specified time, relative to cycle time. (applies to periodic data) **Latency**: within a predictable timespan from the start of the transmission **Bandwidth**: if bandwidth utilization is within in the resources reserved by the sender **None**: no special delivery requirements
Delivery Guarantee Value	**#:** Typical quantification of the data delivery guarantee for 80% of the use cases If "deadline" mode is used, specify if the data will be delivered in the same period or not
Application Tolerant to Jitter	application’s tolerance of a certain amount of latency variati **yes**: application can tolerate jitter as specified (always yes f **no**: highly sensitive application requires negligible jitter
Tolerable Jitter Value	**#:** Value of acceptable jitter for periodic applications **NEG:** Jitter must be negligible **N/A:** if data delivery guarantee mode is "bandwidth" or "nor
Applications Tolerant to packet loss	Application’s tolerance to a certain amount of consecutive packet loss **Yes**: app can tolerate loss due to recovery mechanism in upper layer protocols or basic redundancy **No**: app cannot tolerate a single packet loss
Tolerable packet loss Value	**#:** Num of consecutive packet loss tolerable to app. **0:** if application is not tolerant to packet loss
Application payload size variability	**fixed**: application payload size remain fixed **variable**: app payload varies from one packet to packet
Payload Value (Bytes)	**#:** size/range of application data (payload) to be transmitted in the Ethernet frames.
Data Criticality	Criticality of this data for operation of the critical parts of the system **high**: highly critical for the operation. (DAL A, B) **medium**: relevant but not continuously needed for the operation (Dal C, D) **low**: not relevant for operation (DAL E)

Both supported

Irrelevant/per stream abstraction

NO => No significant delay penalty

End station perception

Supported

Supported

Numbers needed!

Designed for 0 loss in absence of errors on path

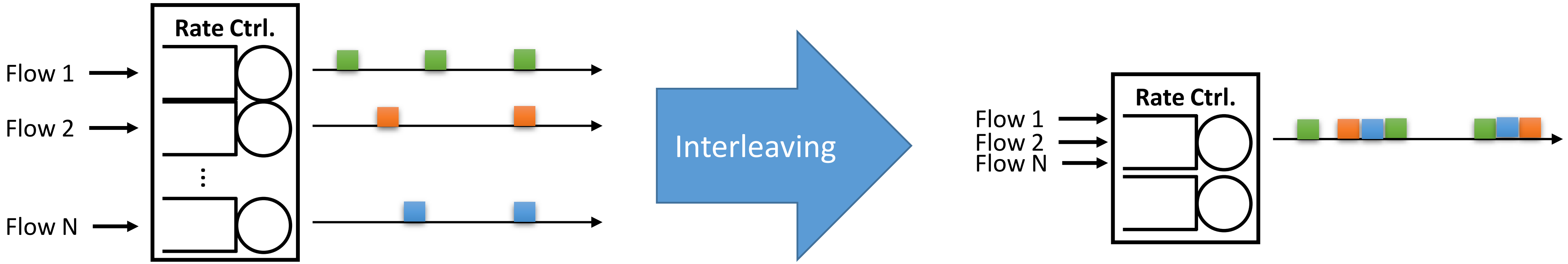
Supported

“AVB-Style”, playout buffers, or DPS*
(https://www.ieee802.org/1/files/public/docs2020/new-specht-dampers-fti-0620-v02.pdf)

Source: https://www.ieee802.org/1/files/public/docs2021/dp-Jabbar-Aerospace-TrafficTypes-Summary-0521-v02.pdf

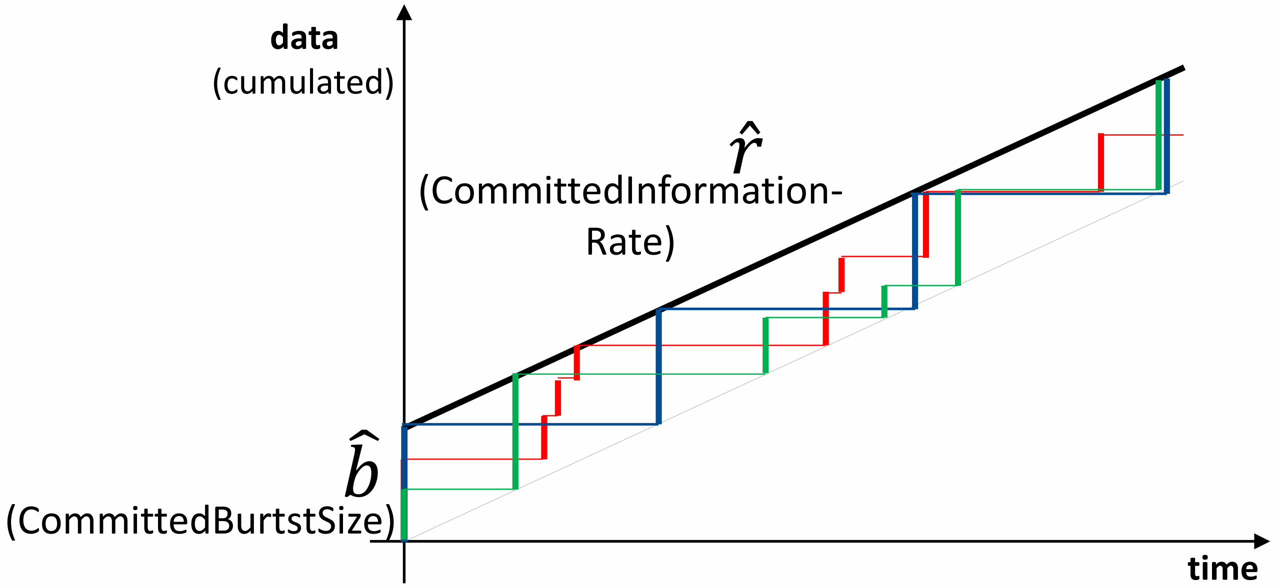
ATS Traffic Types/Streams

Token Bucket Traffic Model



Token Bucket Shaping in a Nutshell

- Buckets fills with tokens at *Flow Rate*
- Tokens consumed by *Packet Length*
- Delay, if not enough tokens



Traffic Types

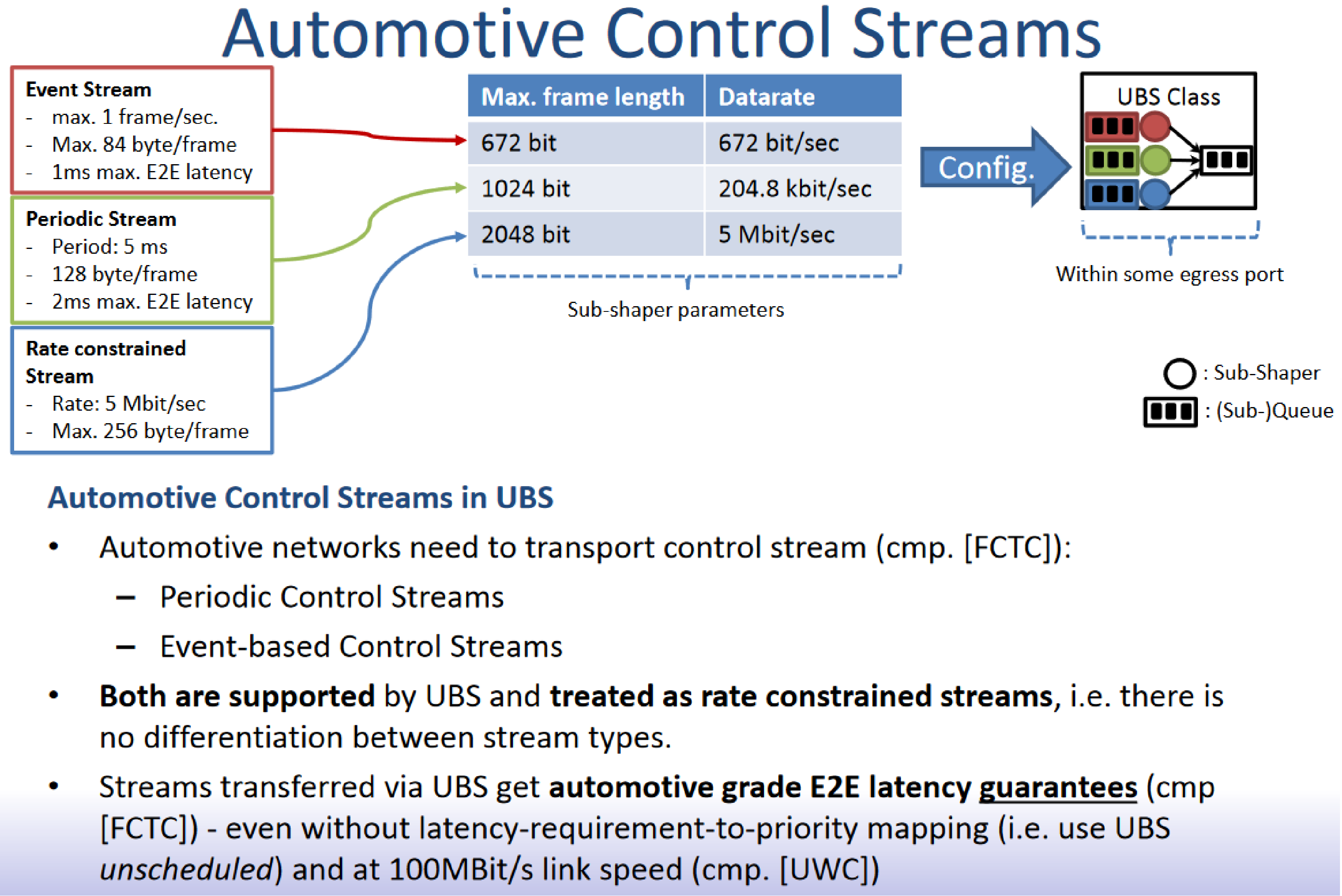
Min. Designflow

1. Per stream mapping to token bucket parameters (CommittedBurstSize & CommittedInformationRate)
2. Delay analysis and network configuration

Aspect to *not* think too much about

- Synchronizing end station timing and network timing (which simply does not exist for ATS)
- Harmonizing periods within a converged network/ mixtures on the same wires

Sidenote: Compared to UBS@2014, ATS “Interleaving” simplifies queuing



Automotive Control Streams

Event Stream
- max. 1 frame/sec.
- Max. 84 byte/frame
- 1ms max. E2E latency

Periodic Stream
- Period: 5 ms
- 128 byte/frame
- 2ms max. E2E latency

Rate constrained Stream
- Rate: 5 Mbit/sec
- Max. 256 byte/frame

Max. frame length	Datarate
672 bit	672 bit/sec
1024 bit	204.8 kbit/sec
2048 bit	5 Mbit/sec

Sub-shaper parameters

Within some egress port

○ : Sub-Shaper
▮▮▮ : (Sub-)Queue

Automotive Control Streams in UBS

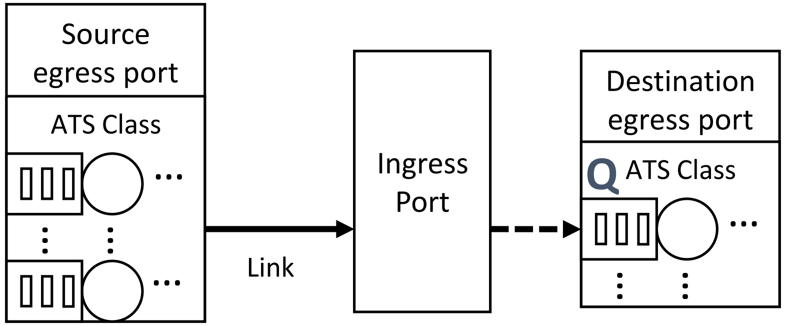
- Automotive networks need to transport control stream (cmp. [FCTC]):
  - Periodic Control Streams
  - Event-based Control Streams
- **Both are supported** by UBS and **treated as rate constrained streams**, i.e. there is no differentiation between stream types.
- Streams transferred via UBS get **automotive grade E2E latency guarantees** (cmp [FCTC]) - even without latency-requirement-to-priority mapping (i.e. use UBS *unscheduled*) and at 100MBit/s link speed (cmp. [UWC])

Source: https://www.ieee802.org/1/files/public/docs2014/new-tsn-specht-ubs-automotive-1114-v01.pdf

ATS Latency & Configuration

Per Hop Latency Math (Simplified)

$$W_f^{max} \le \underbrace{\max_{\forall f' \text{ in } Q(f)}}_{\substack{\text{Max. over} \\ \text{all streams sharing} \\ \text{The queue with } f \\ \text{in the destination port}}} \left(\underbrace{\frac{\sum_{i\in H} l_i^{max} + \sum_{i\in S} l_i^{max} + \max\limits_{i\in L}(l_i^{max})}{R - \sum_{i\in H} R_i}}_{\substack{\text{Interference by all competing streams} \\ \text{In the source egress port (not only} \\ \text{The ones in the same queue like } f\text{).}}} + \underbrace{\frac{l_{f'}^{max}}{R}}_{\text{S\&F}} \right)$$



Properties

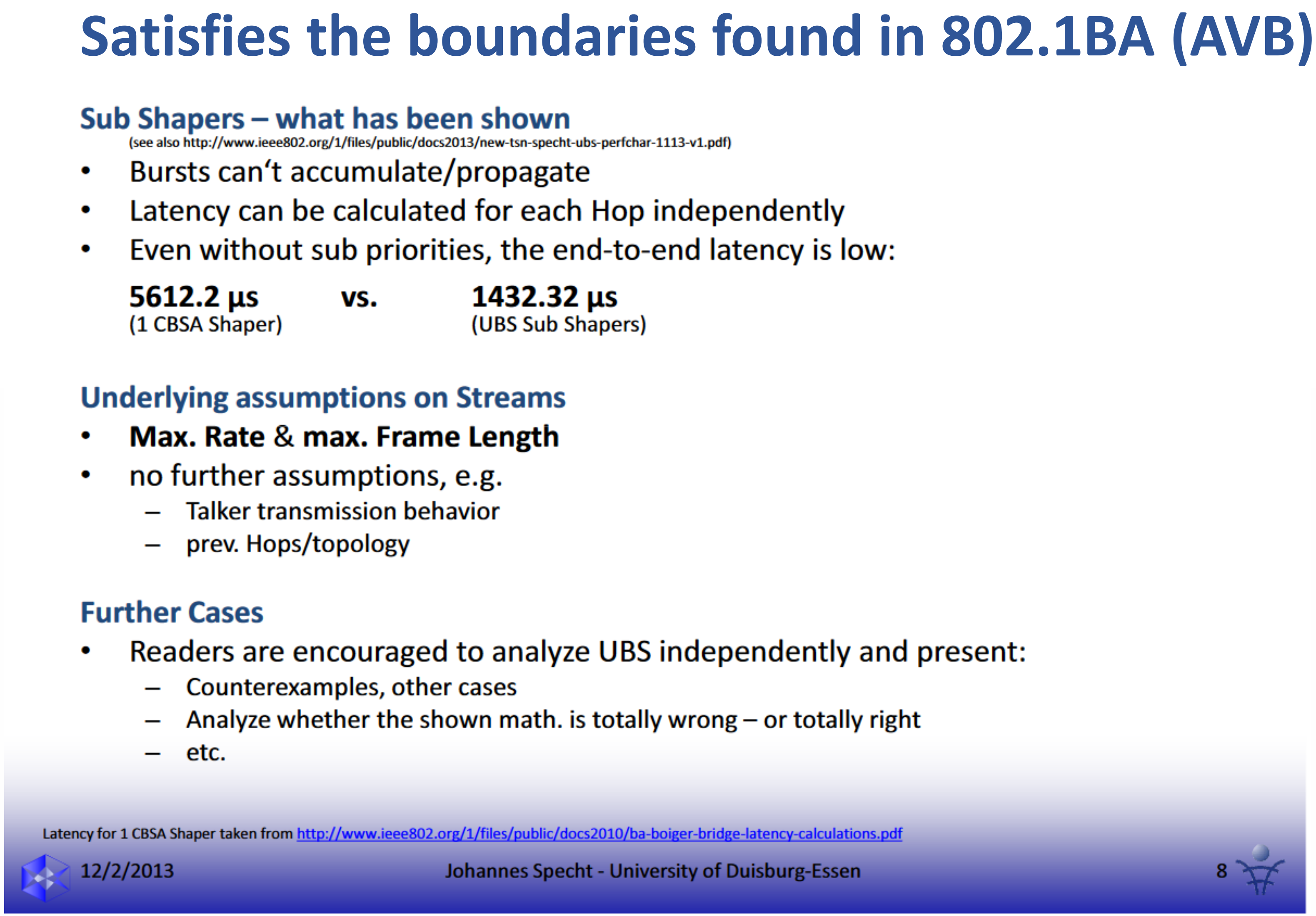
- Closed expression per hop
- Sum along the path from talker to listener

Simplification (won’t change the properties above)

- On this slide
  CommittedBurstSize = Max. packet length
- Key paper
  J. Specht and S. Samii, *Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks*, ECRTS 2016
- Full
  Annex V of IEEE Std 802.1Qcr-2020

Term	Description
W_f^{max}	Max. per hop delay of a stream *f*
$\sum_{i\in H} l_i^{max}$	Sum of max. packet lengths of streams with a higher sub-priority than *f*
$\sum_{i\in S} l_i^{max}$	Sum of max. packet lengths of streams with sub-priority equal to the sub-priority of *f*
$\max\limits_{i\in L}(l_i^{max})$	Maximum packet length of all streams with a lower sub-pririty than f, including lower priority traffic classes.
l_f^{max}	Maximum packet length of streams *f*.
R	Link speed.
$\sum_{i\in H} R_i$	Sum. of datarates of streams (i.e., CommittedBurstSize) with a higher sub-priority than f.

Latency Bounds: Order of Magnitude



Satisfies the boundaries found in 802.1BA (AVB)

Sub Shapers – what has been shown

(see also http://www.ieee802.org/1/files/public/docs2013/new-tsn-specht-ubs-perfchar-1113-v1.pdf)

- Bursts can't accumulate/propagate
- Latency can be calculated for each Hop independently
- Even without sub priorities, the end-to-end latency is low:

5612.2 µs (1 CBSA Shaper)	**vs.**	**1432.32 µs** (UBS Sub Shapers)

Underlying assumptions on Streams

- **Max. Rate** & **max. Frame Length**
- no further assumptions, e.g.
  - Talker transmission behavior
  - prev. Hops/topology

Further Cases

- Readers are encouraged to analyze UBS independently and present:
  - Counterexamples, other cases
  - Analyze whether the shown math. is totally wrong – or totally right
  - etc.

Latency for 1 CBSA Shaper taken from http://www.ieee802.org/1/files/public/docs2010/ba-boiger-bridge-latency-calculations.pdf

12/2/2013 Johannes Specht - University of Duisburg-Essen 8

Source: https://www.ieee802.org/1/files/public/docs2013/new-tsn-specht-ubs-avb1case-1213-v01.pdf

Configuration

There is a Range: How much computation/how optimized?

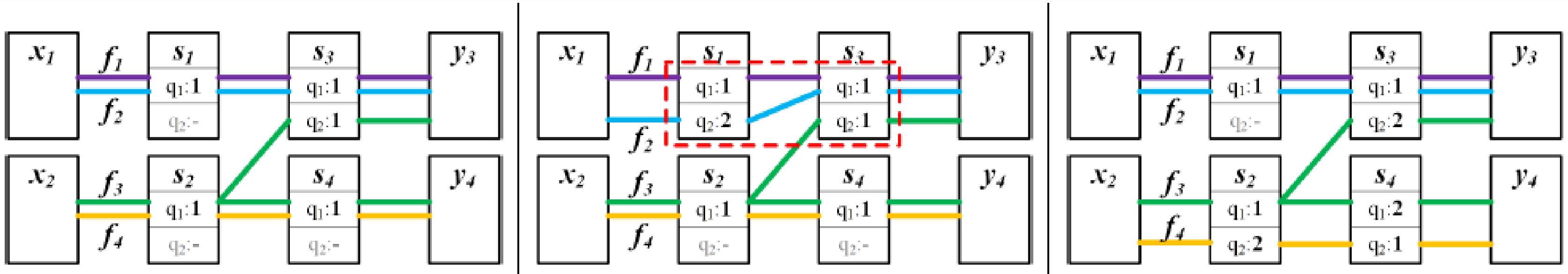
Simple (prev. Slides)

- Trivial latency calculation and setup
- All streams in one or more *global* traffic classes (aka priority level)
- “man-made” stream-to-class association
- Simple enough for distributed dynamic reservation without significant overprovisioning (e.g., P802.1Qdd)

…

Complex

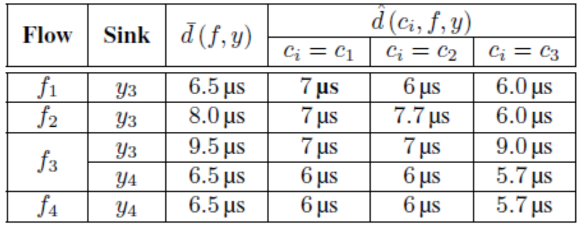
- Fine-tuning by assigning individual per stream per hop priority levels
- Optimize for matching tough/wide-spread per stream E2E latency requirements



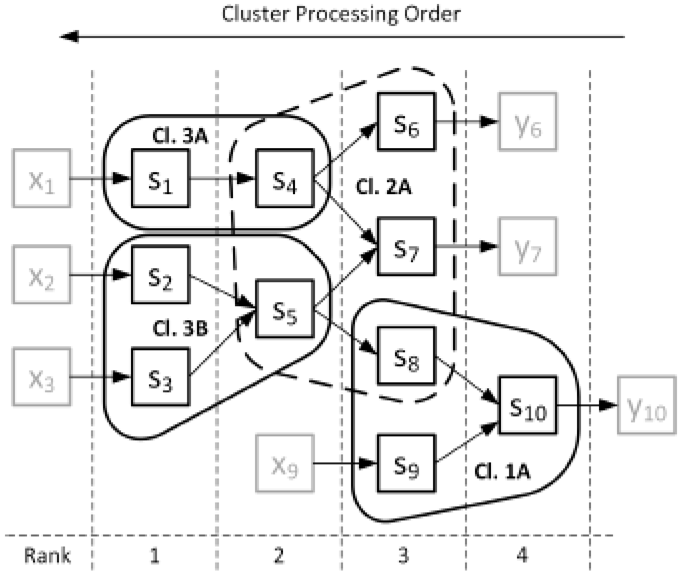
(a) Single priority level configuration c_1 (valid) violating deadline constraint CC2.

(b) Invalid configuration c_2 (violates $QC2$)

(c) Solution c_3 (and likewise a minimal configuration)



Flow	Sink	$\bar{d}(f,y)$	$\hat{d}(c_i,f,y)$		
			$c_i = c_1$	$c_i = c_2$	$c_i = c_3$
f_1	y_3	6.5 µs	**7 µs**	6 µs	6.0 µs
f_2	y_3	8.0 µs	7 µs	7.7 µs	6.0 µs
f_3	y_3	9.5 µs	7 µs	7 µs	9.0 µs
	y_4	6.5 µs	6 µs	6 µs	5.7 µs
f_4	y_4	6.5 µs	6 µs	6 µs	5.7 µs



Source of all figures and tables on this slide: J. Specht and S. Samii, *Synthesis of Queue and Priority Assignment for Asynchronous Traffic Shaping in Switched Ethernet*, RTSS 2017

ATS Robustness

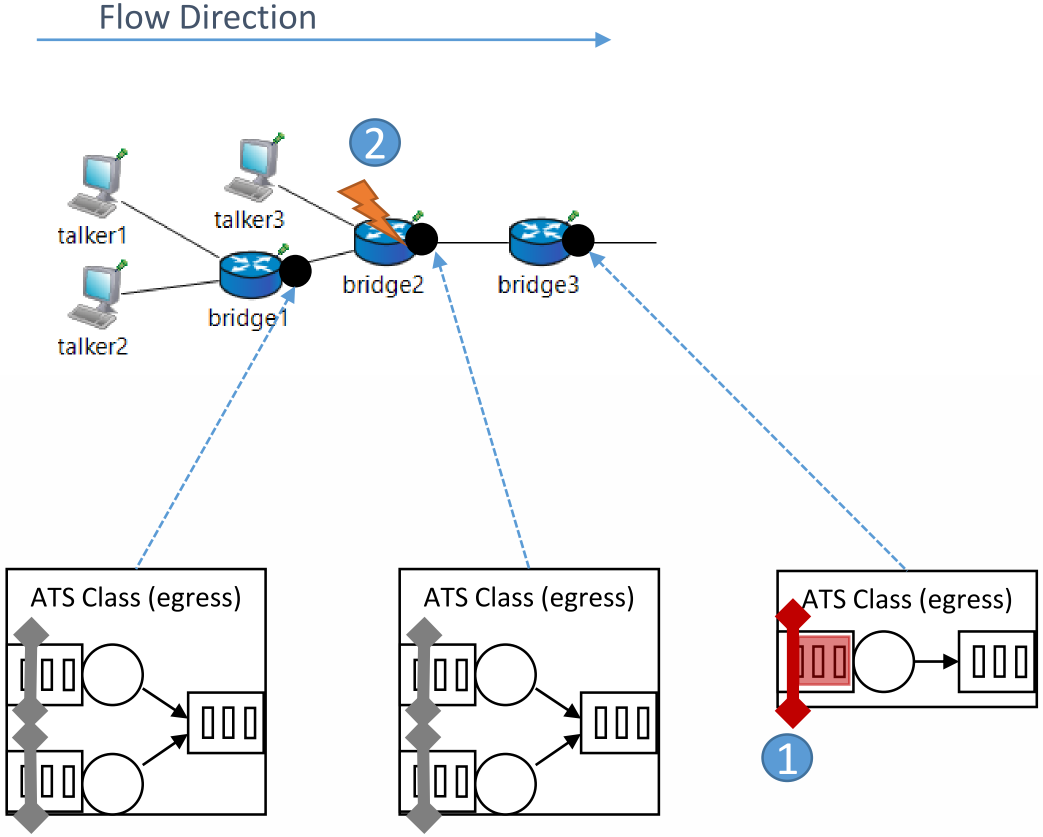
Robustness, Protection and Isolation

1. **Asynchronous**
   - No global clock sync. dependency
2. **Policing Included**
   - Token Bucket shaping
     1. Delaying (shaping), not only dropping (policing)
     2. Re-shaping per hop: No growing disturbance/burstiness along paths

        → No need for increasing CommittedBurstSize values
        (avoid false-positive policing reactions)

        → Low delay impact
        (no need to account for traffic form interfering babbling idiots maxing out increased CommittedBurstSize limits until policing reaction)
   - Possible mutual exclusion
     - Token Bucket state machines (shapers & flow meter) share similarities
     - (Re-)shaped traffic may not need extra flow meters
     - ASIC Implementers may design for this
3. **Traffic Isolation in (virtual) queues**
   - At least at per Port resolution
   - Mindset
     - Stations can break, not only in the nice way (i.e., become babbling idiots)
     - All traffic from a broken station is broken and lost (i.e., no separation in classes/streams)
     - Traffic on paths without broken boxes shall not be affected by interfering broken traffic

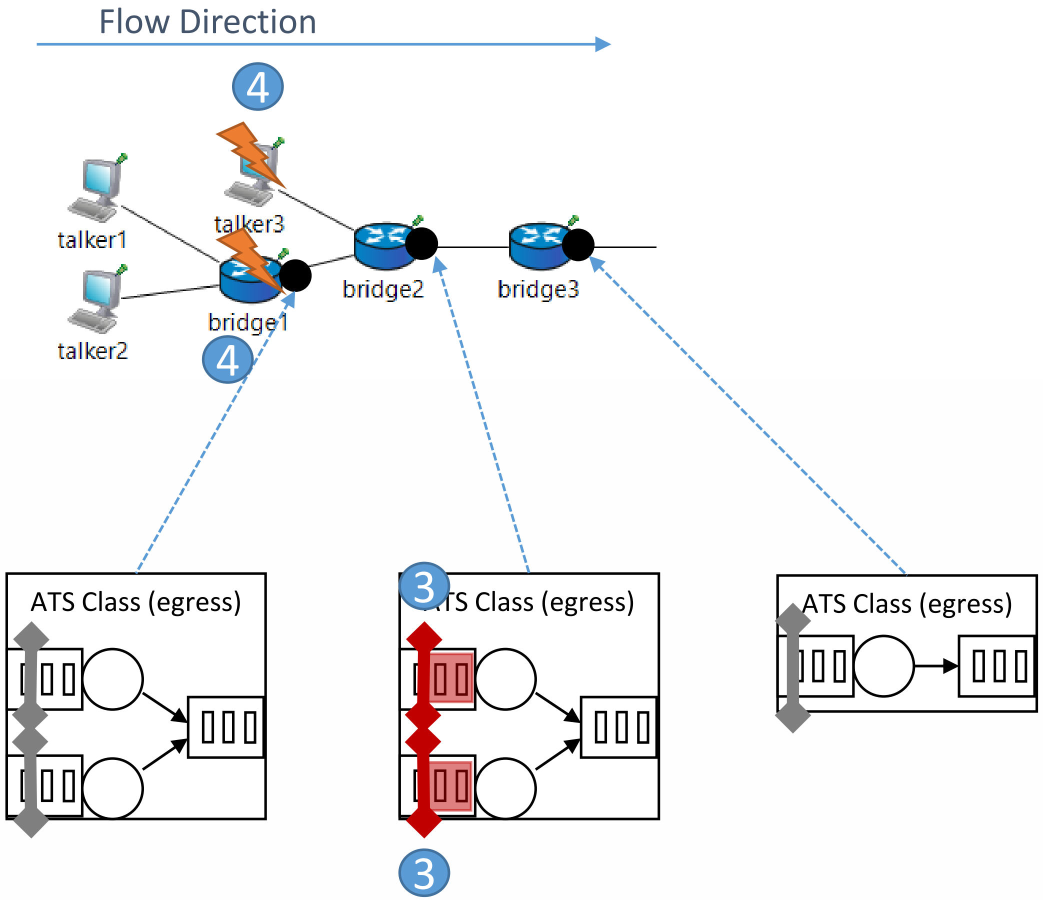
Separate (virtual) queuing at least on a per port resolution



Fault isolation Logic

1. If the queue limit in *bridge3* is exceeded…
2. … only *bridge2* can be the babbling idiot.

Separate (virtual) queuing at least on a per port resolution



Fault isolation Logic

1.If the queue limit in *bridge3* is exceeded…

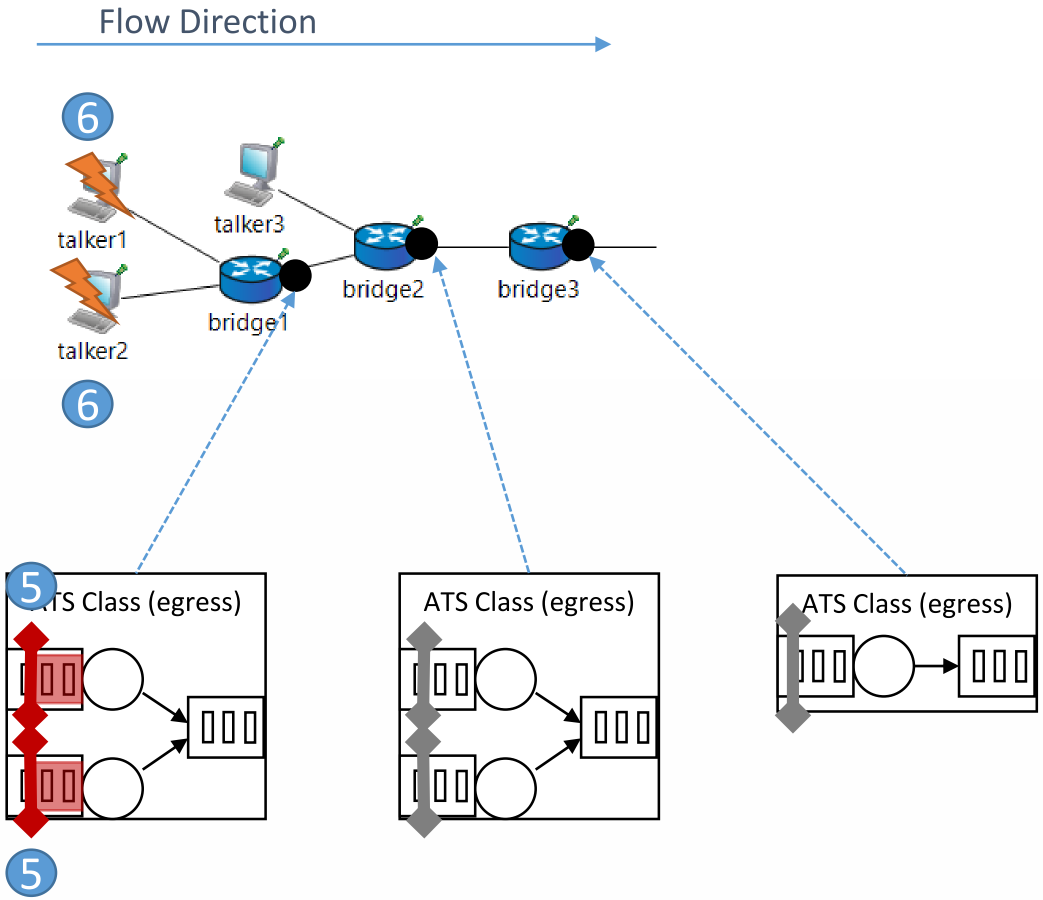
2.… only *bridge2* can be the babbling idiot.

Contradiction

3.If the queue limit in *bridge3* is exceeded and *bridge1* or *talker3* would be the babbling idiot…

4.… limits in *bridge2* would prevent the overload to propagate to *bridge3*.

Separate (virtual) queuing at least on a per port resolution



Fault isolation Logic

1. If the queue limit in *bridge3* is exceeded…
2. … only *bridge2* can be the babbling idiot.

Contradiction

3. If the queue limit in *bridge3* is exceeded and *bridge1* or *talker3* would be the babbling idiot…
4. … queue limits in *bridge2* would prevent the overload to propagate to *bridge3*.

… Continuing …

5. If a queue limit in *bridge2* is exceeded, *bridge1* would be fault free and *talker1* or *talker2* would be the babbling idiot…
6. … queue limits in *bridge1* would prevent the overload to propagate to *bridge2*.

Thank you for your Attention!

Time for Questions & Answers

Johannes Specht
Dipl.-Inform. (FH)

GERMANY
johannes.specht.standards@gmail.com