A journey into time-triggered communication protocols with a focus on Ethernet TSN

Nicolas NAVET, University of Luxembourg

http://labex-digicosme.fr/GT+OVSTR Working Group
Paris | June 11, 2018

Outline

1. Landscape of real-time (wired) communication networks
2. Time-triggered (TT) protocols evolution: TTP, FlexRay, TTEthernet, TSN/TAS (IEEE802.1Qbv)
3. Misconceptions about TT communication
4. Takeaways & what is ahead of us

We focus here on TT protocols with a fine-grained global clock (typ. sub-us precision) but there is a spectrum of solution for TT communication: master-slave protocols, traffic shaping based 1) on local offsets (i.e., locally synchronous – globally asynchronous) or 2) coarse-grained global clocks (ms), etc

Trends in real-time networks

from **Proprietary protocols** to **Industry standards** to **Cross-industry standards**

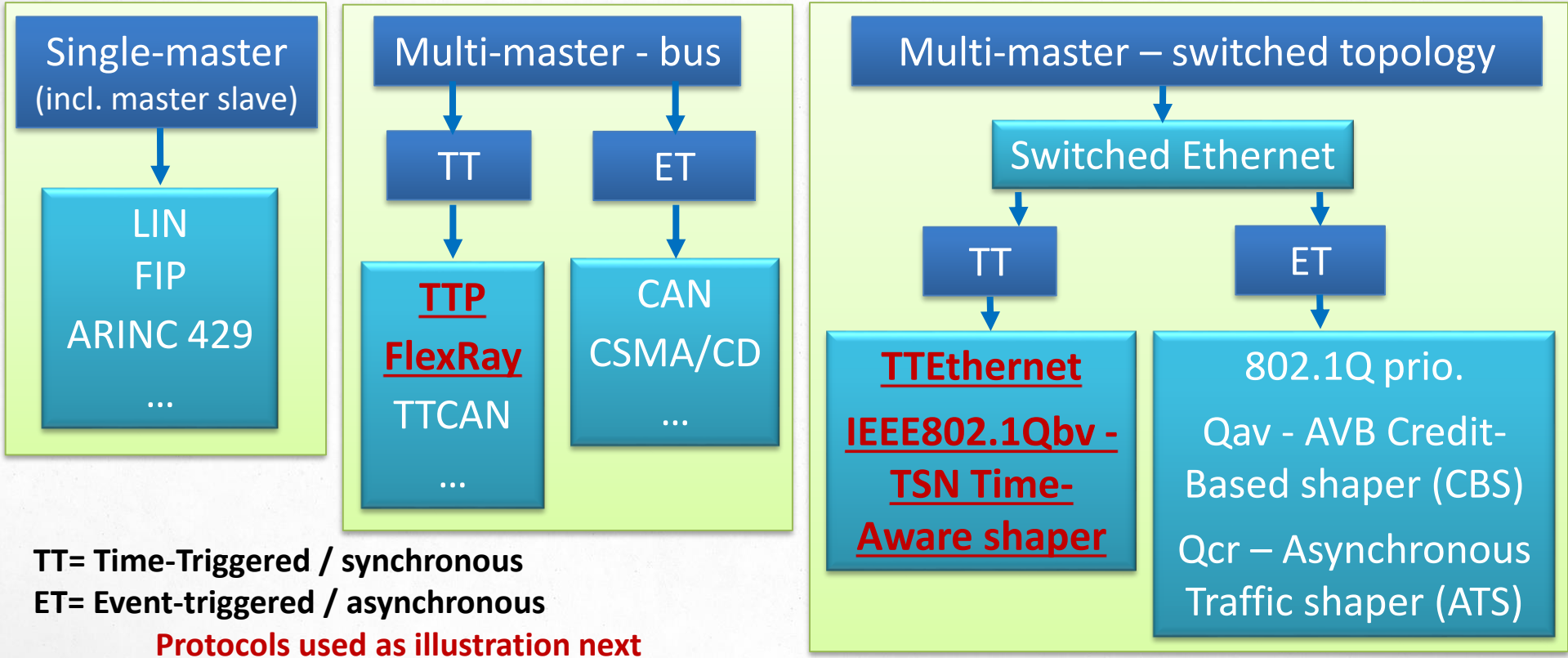
Figure from [13]

- ✓ **Automotive industry**: CAN, LIN, MOST, FlexRay
- ✓ **Aerospace**: ARINC664 (AFDX), ARINC429, ARINC659, MIL-STD-1553
- ✓ **Automation:** IEC standards

- ✓ **TTEthernet** (SAE6802)
- ✓ **IEEE802.1 Time-Sensitive Networking (Ethernet TSN)** now with industry specific profiles, e.g. IEC/IEEE 60802 TSN Profile for Industrial Automation

✓Networking is usually considered as outside the field of competition by OEMs
✓Not all standards are self-contained and up-to-date, and ensure interoperability
✓Cross-domain standardization easier from OSI layer 2 to 4 (Ethernet + TCP-UDP/IP)
✓Personal view: Ethernet TSN will shape the landscape for the next decades

Wired Real-Time Networks



TT= Time-Triggered / synchronous
ET= Event-triggered / asynchronous

Protocols used as illustration next

Main TSN QoS protocols on top of Ethernet

Temporal QoS = managing interfering traffic

Priority-based

IEEE802.1Q

8 priority levels for streams

Benefits:

✓standard and simple

✓efficient at the highest priority

✓can be used with shaping in transmission (“pre-shaping”)

Limitations:

✓ not fine-grained enough to for all kinds of requirements

✓starvation at lowest priority levels with bursty traffic

Traffic Shaping

AVB / Credit-Based Shaper (CBS)

Two egress queues shaped + 6 priority levels below

Benefits:

✓ Perf. guarantee for AVB classes

✓ No starvation for best-effort traffic

Limitations:

✓ Per class (not stream) shaping

✓ Not for control traffic

✓ Not flexible enough with standard configuration (CMI)

Time-triggered (TT)

TSN / Time-Aware Shaper (TAS)

TAS defines egress ports’ gate schedule (open/close)

Benefits:

✓ Strong time constraints can be met

✓ Can be combined with AVB

Limitations:

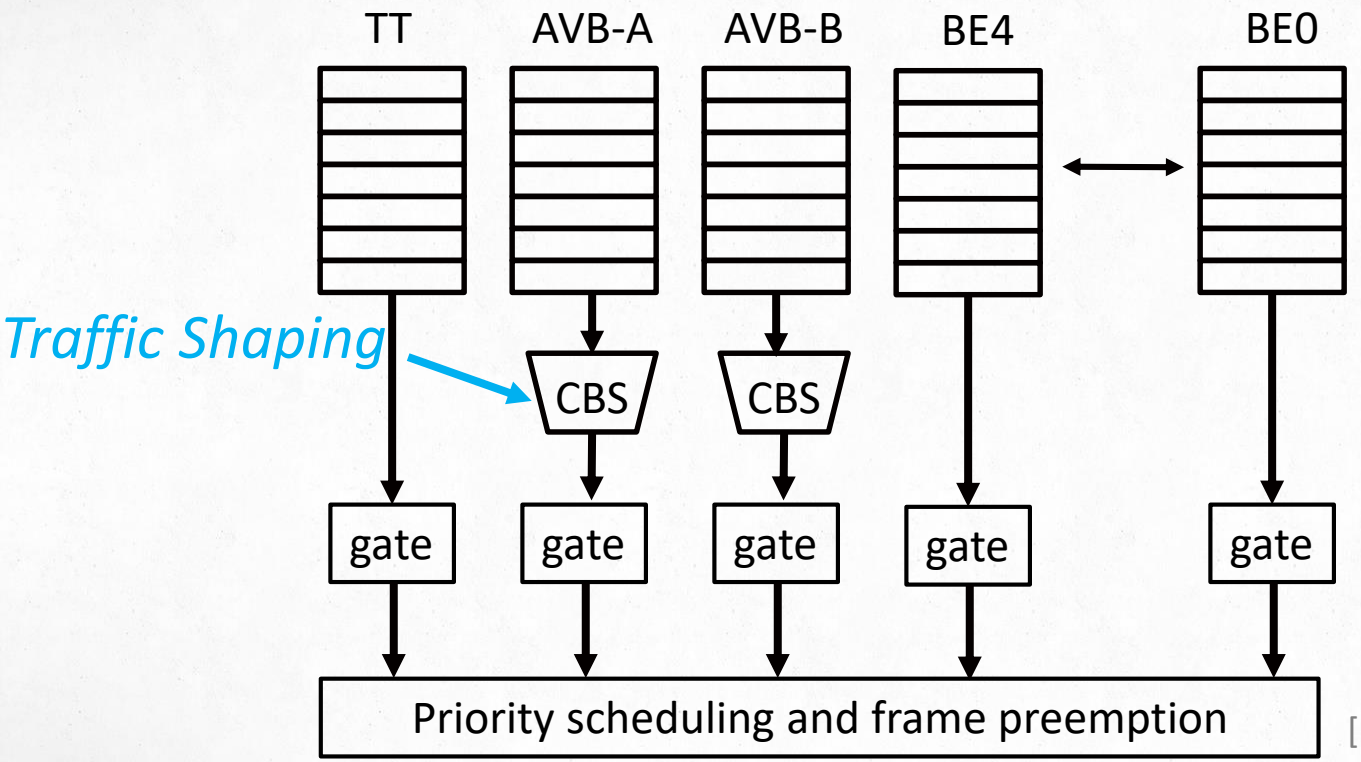
✓ Hard to configure

✓ Rely on a global clock

✓ Task sched. must be tailored to communication for best perf.

See [4] for further information

Typical TSN switch configuration – view of an egress port



Traffic classes up to 8

3 QoS schemes: priority, shaping, blocking (TT windows & preemption)

Gates can be open or closed

[Figure inspired from 12]

The evolution of TT communication protocols

TT protocols: recap of pros & cons

PROS

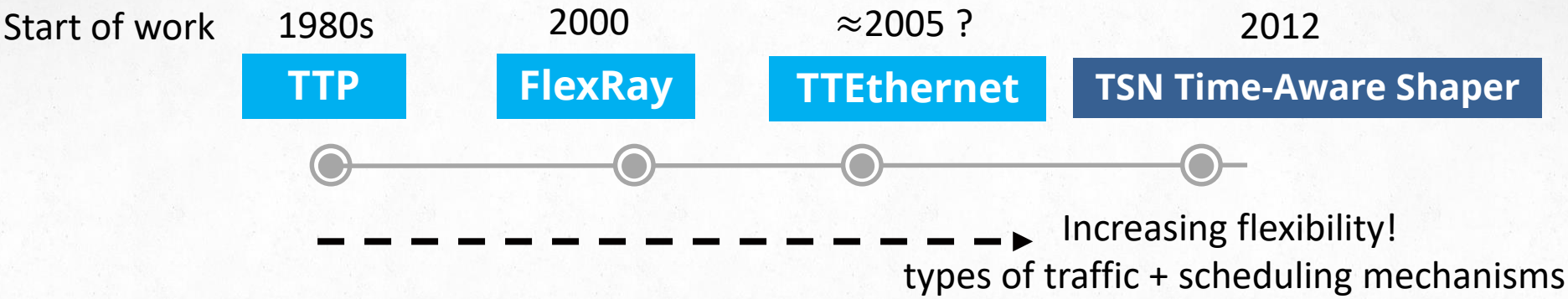
- ✓ Conceptual simplicity for designer
- ✓ Easier to check timing/safety correctness
- ✓ Small jitters
- ✓ “heart beats” and dependability services
- ✓ Adding frames/nodes do not change system behavior if properly planed in advance

CONS

- ✓ Bandwidth utilization usually not optimal because TT slots may not be fully used
- ✓ Clock synchronization needed
- ✓ Coupling between task schedule and message schedule for best data freshness
- ✓ The need for flexibility increases protocol complexity ..

Efficiency of the off-line message and task pre-runtime schedule generation tool is key

TT protocols timeline



TTP

- ✓ Mode changes support

FlexRay

- ✓ A static segment under TDMA, and a dynamic segment under FTDMA
- ✓ Slot-multiplexing: different messages in same slots

TTEthernet

- ✓ TT, Rate-Constrained (RC) and best-effort traffic
- ✓ Per flow TT slots
- ✓ AFDX-like transmissions for RC
- ✓ Priority-based for Best-Effort (BE) and RC traffic

TSN TAS

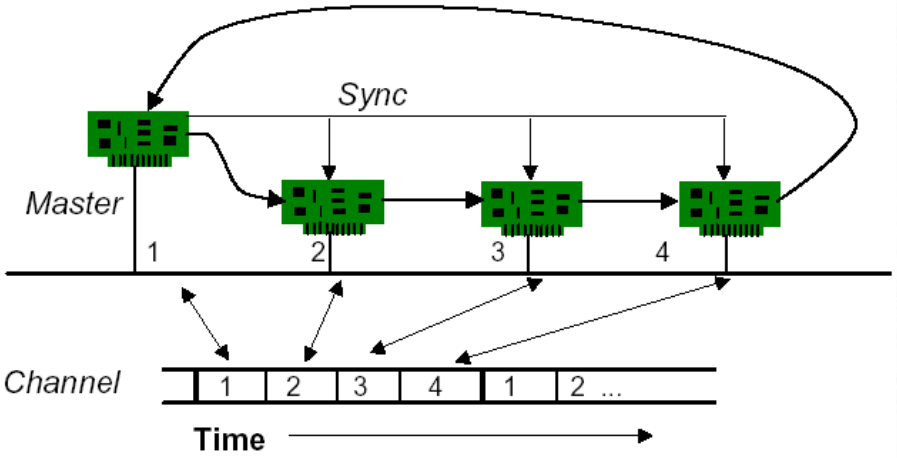
- ✓ Per traffic-class TT slots
- ✓ Traffic shapers, priority scheduling, frame preemption and TT scheduling

Time-Triggered Protocol (TTP)

Time-Triggered Protocol (TTP)

- ✓ Developed from 1980s at T.U. Vienna by H. Kopetz and colleagues, then at TTTech - now SAE AS6003 standard
- ✓ Was considered for use in cars in early 2000s but found its market in aerospace applications (e.g. pressure control system of A380, used in 787 Dreamliner)
- ✓ Characteristics: determinism, fault-tolerance (e.g., clique detection, redundant channels), support for mode changes
- ✓ Data rate up to 20Mbps

Time-Triggered Protocol (TTP)



- ✓ Slot: time window given to a station for a transmission
- ✓ TDMA Round: sequence of slots s.t. each station transmits exactly once
- ✓ Cluster Cycle: sequence of the ≠ TDMA rounds
- ✓ Support for bus guardians to avoid “babbling idiots”

Bounded response times and « heartbeats » but

- ✓ not optimal in terms of bandwidth usage
- ✓ Max refresh rate depends on # of stations - e.g. 5ms achievable with 200bit frames if less than 12 stations or 6 Fault-Tolerant Units (FTU) of two replicated nodes each

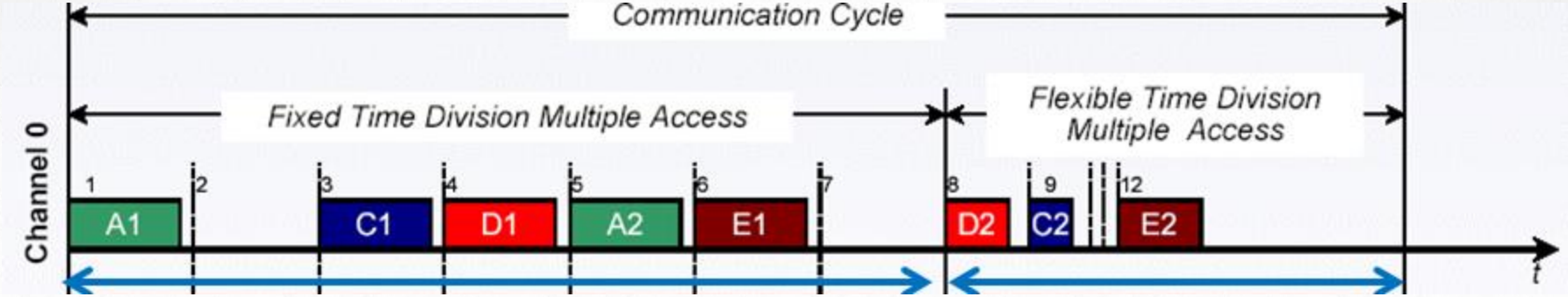
FlexRay

FlexRay

- ✓ Developed from 2000 by an automotive consortium that disbanded in 2009, now maintained as a set of ISO standards
- ✓ Designed as an automotive-specific alternative to TTP
- ✓ FlexRay's dynamic segment operates according to ByteFlight protocol developed by BMW
- ✓ Has been used in 25+ (high-end) series car models, first time in 2006
- ✓ Obsolete technology, will be progressively replaced by Ethernet
- ✓ Meant to support X-by-Wire app. with dependability-related services/features
- ✓ But was merely used as a high-speed CAN for control applications (e.g., chassis)

Design limitations in hindsight: was neither conceived to support audio/video streams nor to fit into TCP/IP stack

FlexRay basics



- ✓ Data rate: between 500kbit/s and 10Mbit/s
- ✓ Typically ST segment: 3 ms and DYN: 2ms
- ✓ Frames: up to 254 bytes (typ. 16bytes), slot size is fixed in the static segment
- ✓ 64 ≠ communication schedules max.

Flexibility through

- ✓ Different comm. schedules for static segment
- ✓ ET traffic in dynamic segment
- ✓ Slot multiplexing in dynamic segment

TTEthernet (TTE)

TTEthernet

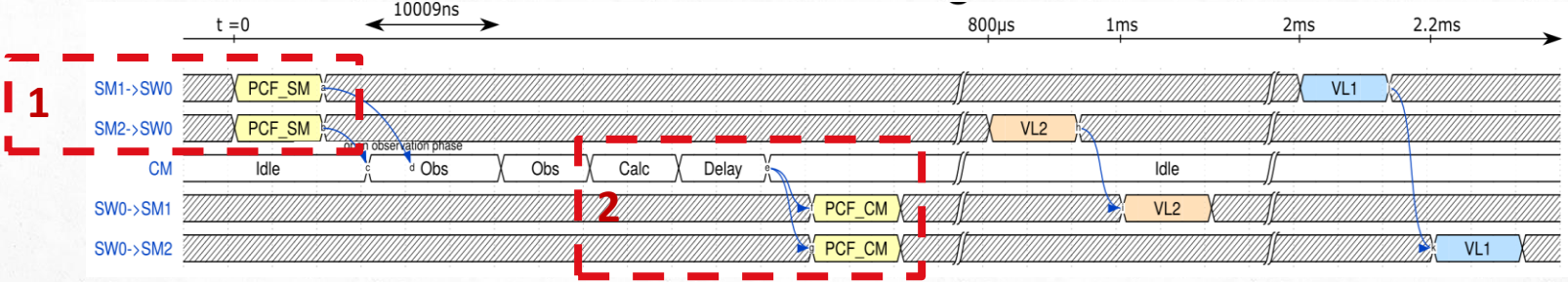
- ✓ TTEthernet (TTE) is a switched Ethernet technology marketed by TTTech and based on SAE6802 standard (2011)
- ✓ TTE is considered for use as high-speed data rate in future launchers (MIL-STD-1553B replacement) and in satellites
- ✓ TTE used in NASA's Orion Multi-Purpose Crew Vehicle
- ✓ TTE provides excellent support the design of applications with strong dependability constraints

Link-Layer protocol supports 3 types of traffic:
Time-Triggered (TT) + (AFDX-like) Rate-Constrained (RC) + Best-effort (BE)

- ✓ Nb: in switched TT networks, each link has its own schedule but all schedules are synchronized
- ✓ In TTE, a single TT frame is transmitted in a TT slot (“per-flow TT schedule”)

A primer on TTE and its clock-synchronization algorithm

✓ Clock synchronization through the exchange of ***Protocol Control Frames*** (PCF)
- Step 1: ***Synchronization Masters*** (SM) “send“ local clock to ***Compression Master(s)*** (CM)
- Step 2: CM calculates new clocks based on received SMs clocks and sends back to SMs
- Step 3: SMs adjust their local clock

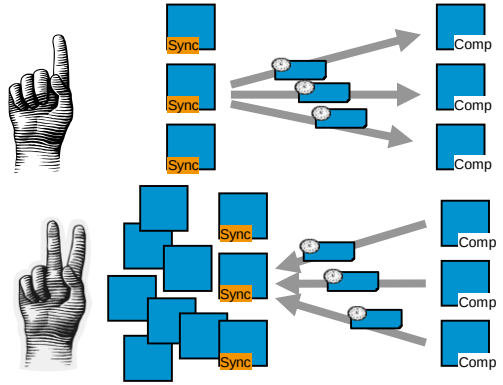


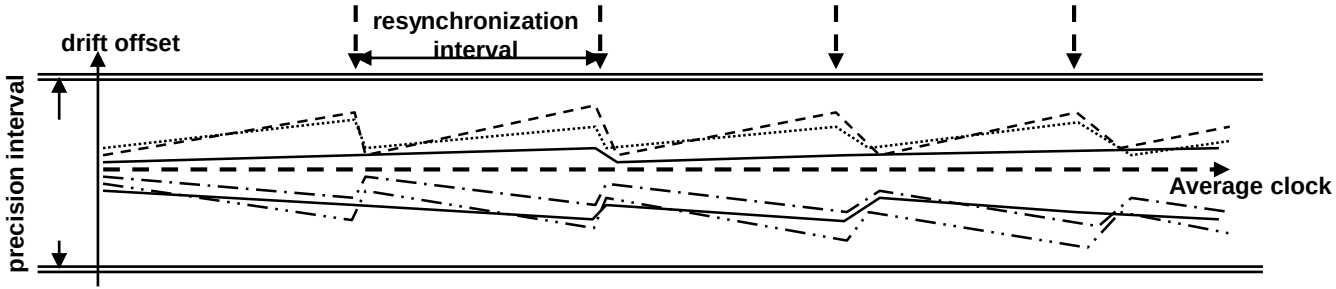
Time-triggered Ethernet – two step synchronization

Protocol Control Frames called “Integration Frames” are used to perform all synchronization functions.

They are transmitted accordingly:

1. The Synchronization Masters send Integration Frames at the beginning of each Integration Cycle. The timing of these frames is used for the “voting”
2. The Compression Masters send Integration Frames to everybody, timing them in a special way so that everybody can correct their clocks

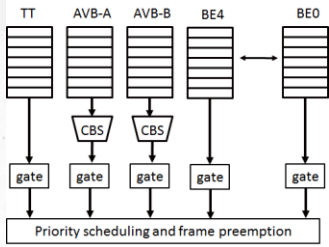




Slide courtesy TTTech – all rights reserved

IEEE Time-Sensitive Networking (TSN)
Time-Aware Shaper (TAS)

TSN & TAS : a primer



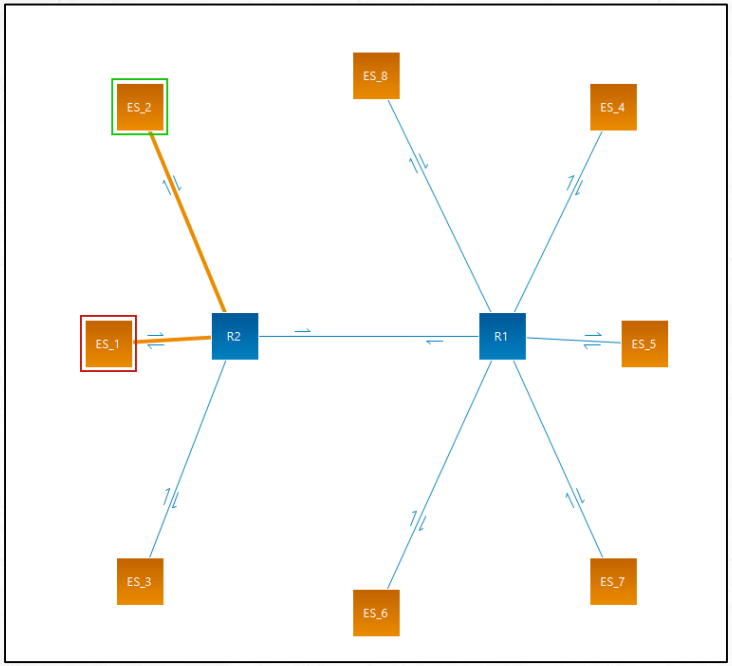
- ✓ IEEE TSN: follow-up initiative to AVB started in 2012 – driven by companies
- ✓ IEEE TSN a set of about 15 standards, most already approved
- ✓ IEEE 802.1qbv: Time-Aware Shaper (TAS) associates a gate with each egresss queue which says whether the queue can transmit (open) or not (closed)
- ✓ A guard band (GB) ensures that no non-allowed frames overlaps with a reserved interval – GB set to max frame size with an optional mechanism that allows a best-effort frame to be transmitted if it can fit in the GB
- ✓ If multiple queues are open, priority scheduling applies
- ✓ Applies on traffic class, not flow - interferences remain from same priority traffic in a FIFO manner
- ✓ Suggested but not required: every critical traffic class has link access only during *scheduled* time intervals with exclusive bus access

TSN: a wealth of possibilities at Link Layer

IEEE 802.1Q	8 priority levels	Standard
IEEE 802.1Qav	AVB Credit Based Shaper (CBS)	Approved
IEEE 802.1Qbv	**Time Aware Shaper (TAS)**	Approved
IEEE 802.1Qcr	Asynchronous Traffic Shaper (ATS)	Ongoing
IEEE 802.1Qbu	Frame Preemption	Approved
IEEE 802.1Qci	Per stream ingress policing	Approved
802.1ASrev	Clock synchronization protocols	Ongoing
802.1CB	Redundancy, Frame Replication	Approved
802.1Qca	Path Control and Reservation	Approved
802.1Qcc	Central Configuration Management	Approved

TAS illustration : setup

All links at 100Mbps except inter-switch link at 1Gbps



Focus on stream CC_1 that goes from ES_1 to ES_2 via switch R2

[RTaW-Pegase screenshot]

ClassConfigurations

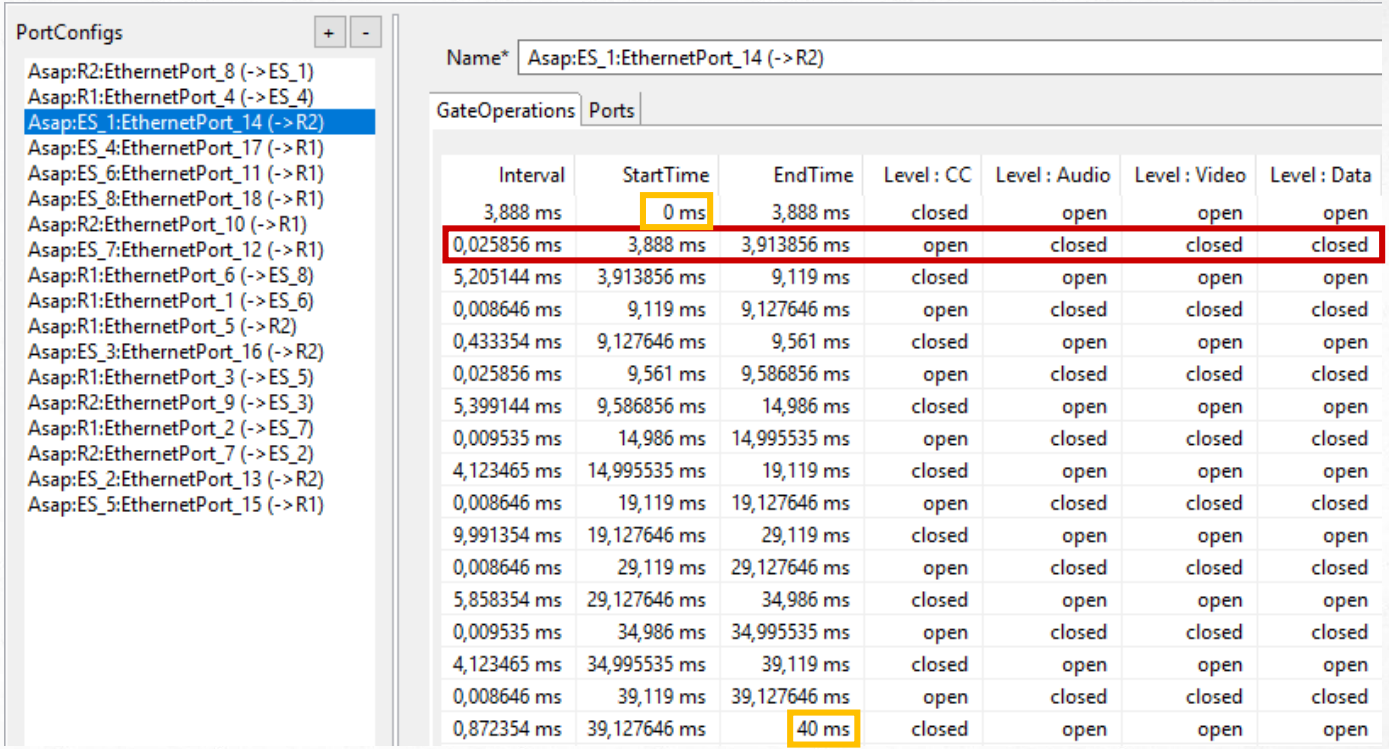
TrafficClass	Priority	SchedulingPolicy
C&C	6	FIFO
Audio	5	FIFO
Video	4	FIFO
Data	3	FIFO

4 traffic classes with Command & Control (C&C) under TAS

Algorithm *ASAP* in RTaW-Pegase: minimize latencies for one traffic class having exclusive bus access
- other algorithms in [8,9,10,11]

TAS schedule: Per egress-port Gate Control List

GCL starts at zero and repeats in cycles after 40ms here (LCM of stream periods)

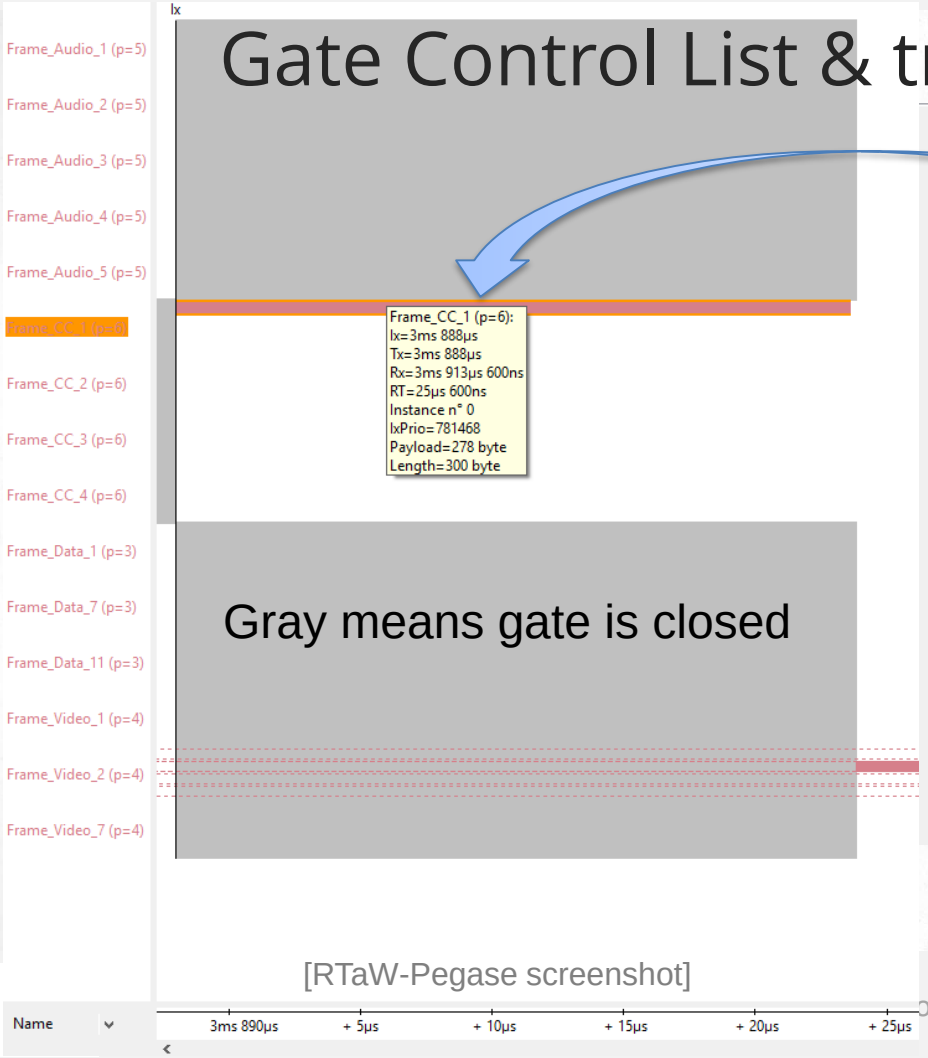


Interval	StartTime	EndTime	Level : CC	Level : Audio	Level : Video	Level : Data
3,888 ms	0 ms	3,888 ms	closed	open	open	open
0,025856 ms	3,888 ms	3,913856 ms	open	closed	closed	closed
5,205144 ms	3,913856 ms	9,119 ms	closed	open	open	open
0,008646 ms	9,119 ms	9,127646 ms	open	closed	closed	closed
0,433354 ms	9,127646 ms	9,561 ms	closed	open	open	open
0,025856 ms	9,561 ms	9,586856 ms	open	closed	closed	closed
5,399144 ms	9,586856 ms	14,986 ms	closed	open	open	open
0,009535 ms	14,986 ms	14,995535 ms	open	closed	closed	closed
4,123465 ms	14,995535 ms	19,119 ms	closed	open	open	open
0,008646 ms	19,119 ms	19,127646 ms	open	closed	closed	closed
9,991354 ms	19,127646 ms	29,119 ms	closed	open	open	open
0,008646 ms	29,119 ms	29,127646 ms	open	closed	closed	closed
5,858354 ms	29,127646 ms	34,986 ms	closed	open	open	open
0,009535 ms	34,986 ms	34,995535 ms	open	closed	closed	closed
4,123465 ms	34,995535 ms	39,119 ms	closed	open	open	open
0,008646 ms	39,119 ms	39,127646 ms	open	closed	closed	closed
0,872354 ms	39,127646 ms	40 ms	closed	open	open	open

[RTaW-Pegase screenshot]

Gate Control List for link ES_1 to R2

Gate Control List & transmission window



Asap:ES_1:EthernetPort_14 (->R2)

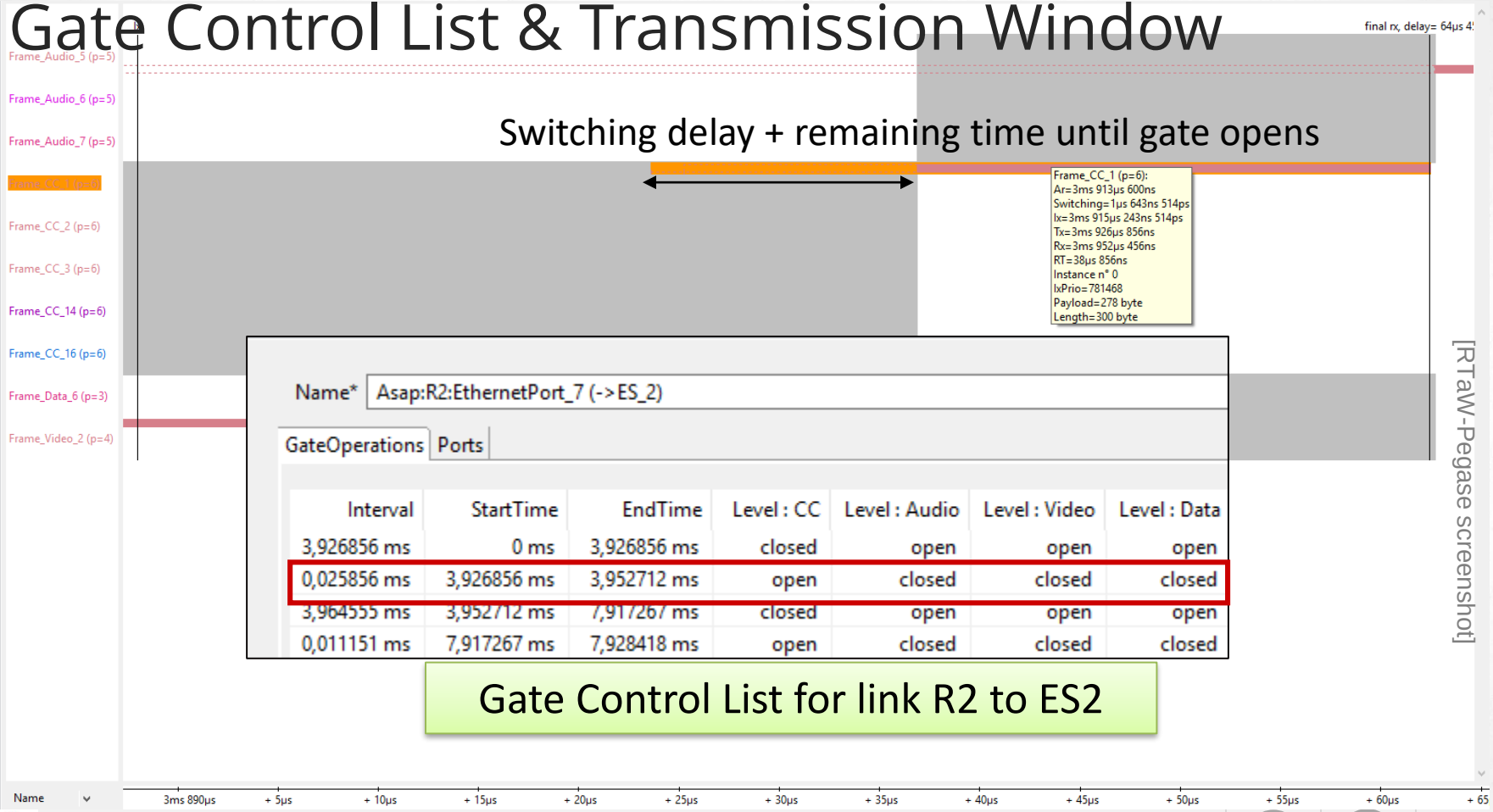
GateOperations | Ports

Interval	StartTime	EndTime	Level : CC	Level : Audio	Level : Video	Level : Data
3,888 ms	0 ms	3,888 ms	closed	open	open	open
0,025856 ms	3,888 ms	3,913856 ms	open	closed	closed	closed
5,205144 ms	3,913856 ms	9,119 ms	closed	open	open	open
0,008646 ms	9,119 ms	9,127646 ms	open	closed	closed	closed
0,433354 ms	9,127646 ms	9,561 ms	closed	open	open	open
0,025856 ms	9,561 ms	9,586856 ms	open	closed	closed	closed
5,399144 ms	9,586856 ms	14,986 ms	closed	open	open	open
0,009535 ms	14,986 ms	14,995535 ms	open	closed	closed	closed
4,123465 ms	14,995535 ms	19,119 ms	closed	open	open	open
0,008646 ms	19,119 ms	19,127646 ms	open	closed	closed	closed
9,991354 ms	19,127646 ms	29,119 ms	closed	open	open	open
0,008646 ms	29,119 ms	29,127646 ms	open	closed	closed	closed
5,858354 ms	29,127646 ms	34,986 ms	closed	open	open	open
0,009535 ms	34,986 ms	34,995535 ms	open	closed	closed	closed
4,123465 ms	34,995535 ms	39,119 ms	closed	open	open	open
0,008646 ms	39,119 ms	39,127646 ms	open	closed	closed	closed
0,872354 ms	39,127646 ms	40 ms	closed	open	open	open

Gate Control List for link ES_1 to R2

[RTaW-Pegase screenshot]

Gate Control List & Transmission Window



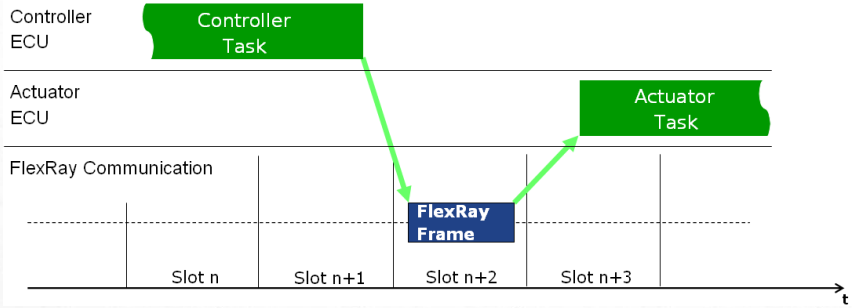
Gate Control List for link R2 to ES2

[RTaW-Pegase screenshot]

A number of misconceptions ?

#1 - TT ensures short communication latencies

- ✓ Tasks run either synchronously or asynchronously wrt the comm. cycle:
  1. Fully asynchronously : data produced at arbitrary points in time
  2. Weakly synchronously : task startup triggered by the networks but task periods are arbitrary
  3. Synchronously : task periods multiple of the cycle length



Picture from [1]

Asynchronisms between data production and data transmission may drastically reduce data freshness (even if communication latencies are small)

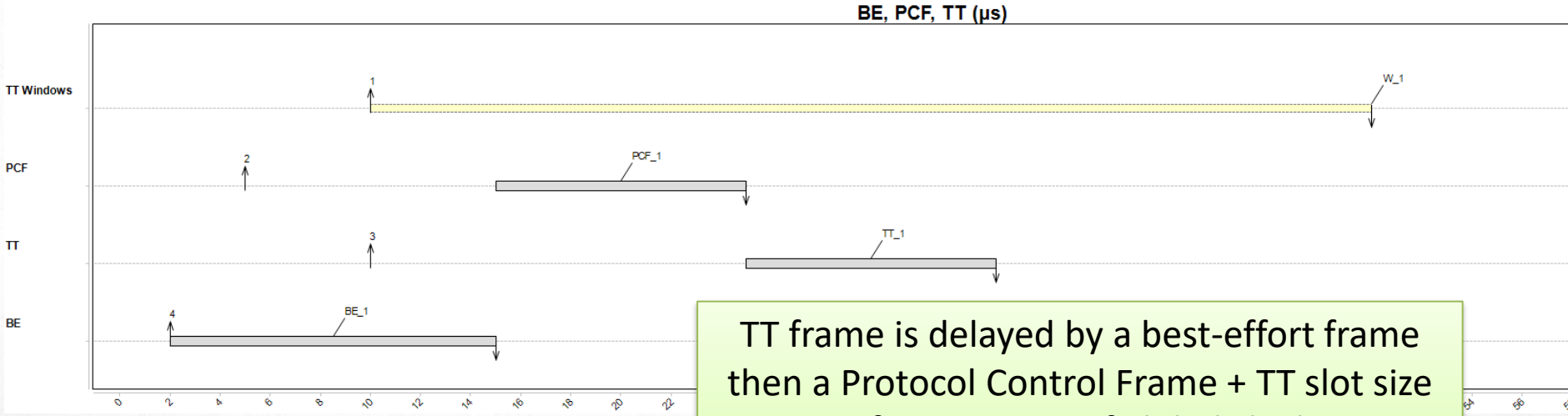
#2 - There are no jitters in TT networks

- ✓ Jitters in reception is usually what matters
- ✓ Jitters can be suppressed by buffering in reception .. at the expense of latencies
- ✓ 2 distinct cases:
  - Per-stream TT schedule like in TTEthernet, TTP and FlexRay
  - Per-class TT schedule like in TSN/TAS

1. True, jitters are reduced by comparison with ET networks
2. But there are jitters due to interfering traffic, limited clock precision, variable switching delays, traffic sharing same TT windows, etc.
3. In some contexts, jitters can be significant if not paid attention to

Per-stream schedule, e.g. TTEthernet

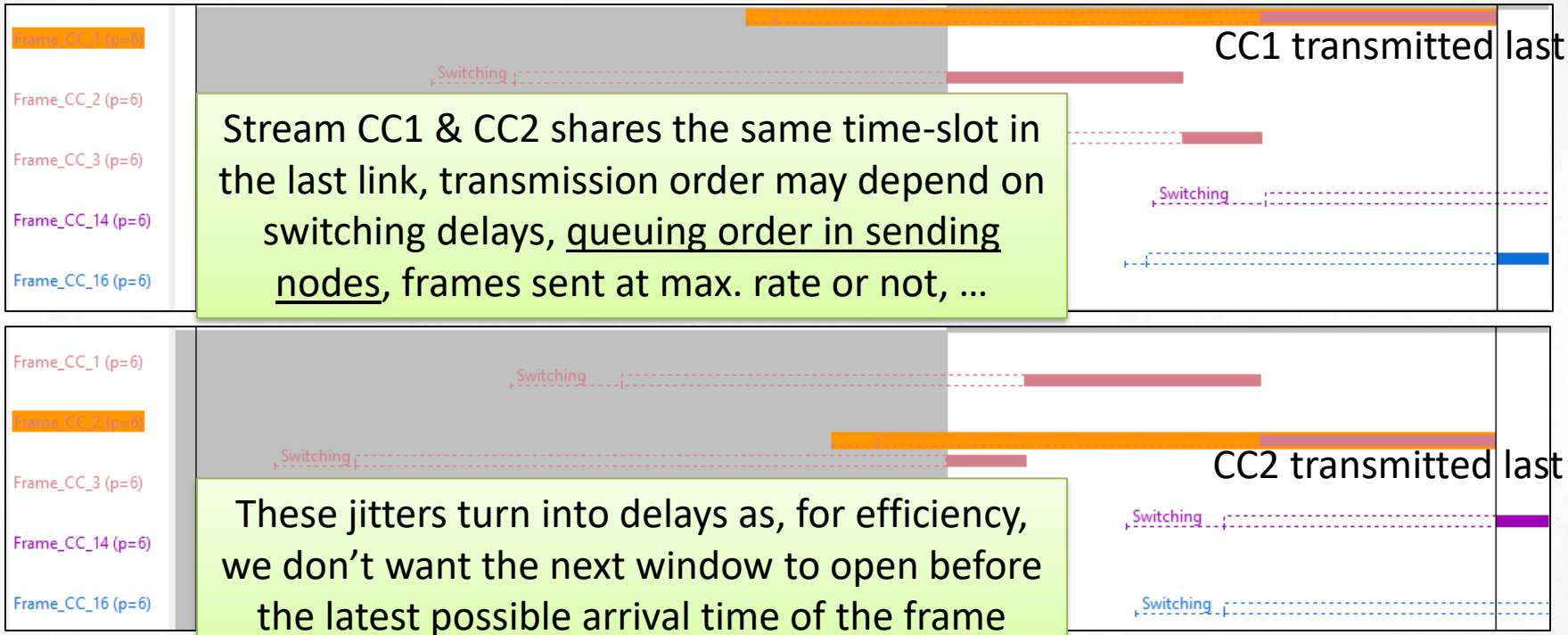
- ✓ Depends on the traffic “integration policy”: media reservation window (=guard band under TSN/TAS) or shuffling like below:



Max. BE frame size @100Mbps = 123us
PCF frame size = 6us

TT frame is delayed by a best-effort frame then a Protocol Control Frame + TT slot size accounts for precision of global clock – jitters in the 130-140us range

Per-class schedule, e.g. TSN/TAS



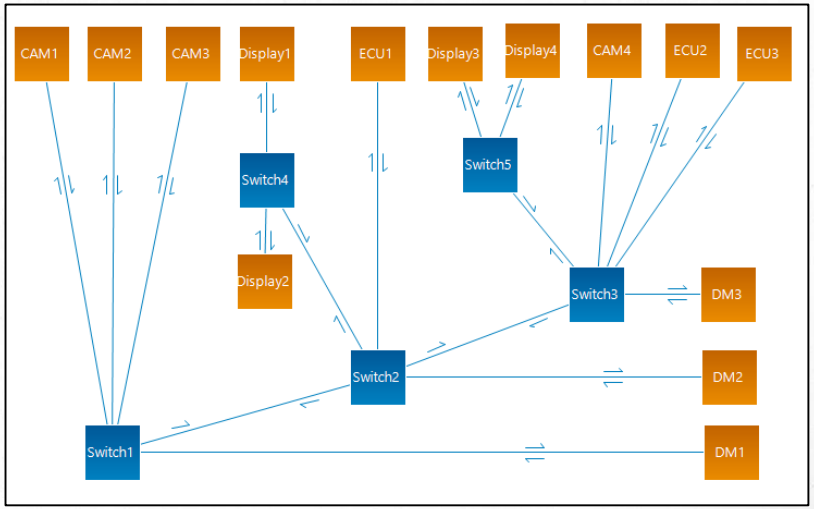
Two possible transmission schedules on the last link of a path

#3 - Jitters grow along with communication latencies

Max jitter = worst-case latency – best-case latency

Yes, in ET networks
like CAN or non-TT Ethernet

No, in TT networks and delaying transmission in the final link will reduce/suppress jitter .. at the expense of additional delay



#4 - TT protocols are bandwidth-efficient at high load

- ✓ If TT slots are not used by TT traffic, they cannot be used by other types of traffic (in FlexRay/Qbv/TTP – slot reallocation possible in TTE)
- ✓ "Guard bands" before TT slots are lost transmission times (in TTE – optional mechanism to use them in Qbv)
- ✓ TT slots are bigger than packet size due to clock precision (all networks), and possibly an unfinished transmission at the beginning (shuffling in TTE) or urgent transmissions during TT slot (PCFs in TTE).

1. But bandwidth has become cheaper
2. And pre-runtime scheduling is more efficient .. provided traffic characteristics are well known

#5 - TT protocols are proven correct

Indeed, key correctness properties of TTP/TTE/FlexRay have been formally established (e.g. clock synchronization, clique detection) but

- ✓ Formal models do not cover all properties of interest
- ✓ Proofs are made with assumptions (e.g., simplifications) not always met by actual systems
- ✓ Proofs usually do not go beyond the design fault-hypotheses, but what happens outside? → Simulation Based Fault Injection helpful here
- ✓ Proofs are based on standards/specifications but implementation may not fully comply and implementation choices may matter

Verification and comprehensive understanding of new technologies in critical systems best achieved through combined use of testbeds, formal verification and simulation (see[5])

#6 - TT protocols support composability

- ✓ The ability to design a system by integration of sub-systems
- ✓ Correctness in the time, value and safety domain of a sub-system is not invalidated after integration
- ✓ In theory it is enough to provision empty TT slots to integrate sub-systems
- ✓ But requirements evolve over time and sub-systems are developed in parallel, how to conceive the schedule so as to avoid conflicting requirements?

Personal view: TT protocols may introduce coupling between sub-systems – imagine a new/updated function requiring new data or a different timing QoS:

- ✓ The transmission schedule may have to be updated globally
- ✓ The scheduling of task on each station should be adjusted accordingly

Conclusions

Takeaways

- ✓ After 40 years, TT vs ET communication remains a controversial question!
- ✓ Historically, TT protocols have evolved towards an **increasing flexibility** – TT transmission becomes one possibility among other QoS strategies (shaping, priority, preemption) --> networks have become multi-protocols
- ✓ Today: most important TT protocols for real-time communication are TTEthernet (in markets like aerospace) and TSN/TAS (in automotive and industrial domains)
- ✓ Personal view: **building efficient (static) TT schedules is well mastered** – just like more generally the problem of configuring mixed TT/ET protocols (e.g., through design-space exploration, see *ZeroConfig-TSN* [13])
- ✓ Academics have disappeared from the landscape, industry is the driving force

Ahead of us

- ✓ Challenge: **TT protocols for dynamically evolving systems** (see [14]) such as production line reconfiguration for customized production in Smart Manufacturing or TSN-based fog node communication [9]
- ✓ In the spirit of SDN, **application-specific / functioning-mode-specific link-level protocols** (see[7]) should lead to more flexibility & more adaptable networks

Thank you for your attention!

Any questions or feedback? Contact: nicolas.navet@uni.lu

References

Click to download

1. B. Schätz, C. Kühnel, M. Gonschorek, “The FlexRay Protocol”, Automotive embedded Handbook, N. Navet, F. Simonot-Lion editors, 2008.
2. N. Navet, M. Grenier, L. Havet, "Configuring the communication on FlexRay: the case of the static segment", Proc. ERTS 2008, 2008.
3. B. Gaujal, N. Navet, "Maximizing the Robustness of TDMA Networks with Applications to TTP/C", Real-Time Systems, vol 31, nº1-3, pp5-31, 2005.
4. J. Migge, J. Villanueva, N. Navet, M. Boyer, “Insights on the performance and configuration of AVB and TSN in automotive networks”, Proc. ERTS 2018, 2018.
5. L. Fejoz, B. Régnier, P. Miramont, N. Navet, “Simulation-Based Fault Injection as a Verification Oracle for the Engineering of Time-Triggered Ethernet networks”, Proc. ERTS 2018, 2018.
6. M. Boyer, H. Daigmorte, N. Navet, J. Migge, "Performance impact of the interactions between time -triggered and rate-constrained transmissions in TTEthernet", Proc. ERTS 2016, 2016.
7. M. Grenier, N. Navet, "Fine Tuning MAC Level Protocols for Optimized Real-Time QoS", IEEE Transactions on Industrial Informatics, vol 4, nº1, 2008.

FlexRay

TTP

TSN

TTE

Application-specific MAC protocols

References

8. J.B. Chaudron, "TTEthernet: Theory and Concepts", TORRENTS working group, 28/06/2013.
9. M.L. Raagaard, P. Pop, M. Gutiérrez, W. Steiner, Runtime reconfiguration of time-sensitive networking (TSN) schedules for Fog Computing, IEEE Fog World Congress (FWC), 2017.
10. R.S. Oliver, S.S. Craciunas, W. Steiner, "IEEE 802.1Qbv Gate Control List Synthesis using Array Theory Encoding", Proc. RTAS, 2018.
11. S.S. Craciunas, R. Serna Oliver, M. Chmelik, and W. Steiner, "Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks", Proc. RTNS, 2016.
12. P. Pop, M.L. Raagaard, S.S. Craciunas, W. Steiner, "Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks", IET Cyber-Physical Systems: Theory & Applications, 1(1), 2016.
13. N. Navet, J. Villanueva, J. Migge, "Automating QoS protocols selection and configuration for automotive Ethernet networks", oral presentation at WCX18: SAE World Congress Experience (WCX018), Detroit, USA, April 10-12, 2018.

Excellent presentation on TTE

Schedule construction for TSN/TAS

Design Space Exploration for TSN/TAS

References

13. J.P. Thomesse, “Fieldbus Technology in Industrial Automation”, slides of keynote speech at ETFA 2005.
14. W. Qiu, “Future Industrial Network Requirement Discussion for TSN”, slides of presentation at the IEEE802.1 Interim meeting in Saint-Paul’NL, 2017.