



TSN the new super hero?

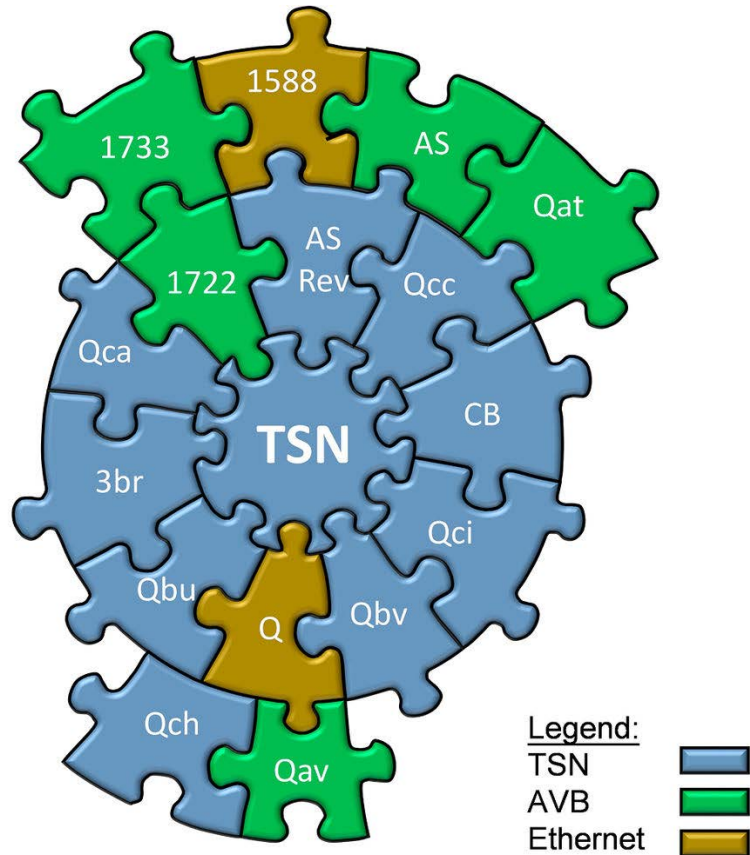
TSN according to Wiki

Time-Sensitive Networking (TSN) is a set of standards under development by the Time-Sensitive Networking task group of the IEEE 802.1 working group. The TSN task group was formed in November 2012 by renaming the existing Audio Video Bridging Task Group and continuing its work. The name changed as a result of extension of the working area of the standardization group. The standards define mechanisms **for the time-sensitive transmission of data over deterministic Ethernet networks.**

The majority of projects define extensions to the IEEE 802.1Q – Bridges and Bridged Networks, which describes Virtual LANs and network switches. These extensions in particular address the transmission of very low transmission latency and high availability.

Applications include converged networks with real time **Audio/Video Streaming** and real-time control streams which are used in **automotive** or **industrial control facilities.**

The IEEE name and number puzzle



Source: Bosch-Rexroth AG

Traffic shaping

- Preemption IEEE 802.3br & 802.1Qbu
- Time aware shaper (TAS) IEEE 802.1Qbv
- Credit Based Shaper IEEE 802.1Qav
- Cyclic Queuing and Forwarding IEEE 802.1Qch
- Asynchronous Traffic Shaping IEEE 802.1Qcr

Reliability

- Frame Replication & Elimination IEEE 802.1CB

Dedicated Resources

- Stream Reservation Protocol IEEE 802.1Qat
- Extensions (LRP & RAP) IEEE 802.1Qdd
- Data Path Discovery and Control IEEE 802.1Qca
- Per Stream Filtering & Policing IEEE 802.1Qci
- Resource Mgt Models IEEE 802.1Qcc

Synchronization (base service)

- IEEE 802.1ASRev

“Ready” by 2023/2024

TSN definition

TSN is defined as

- **Guaranteed data transport** with
 - **bounded low latency,**
 - **low delay variation,**
 - and **extremely low loss**

How do we achieve this?

- Bounded low latency and low delay variation
- Extremely low loss

traffic shaping

dedicated resources & reliability

Using TSN

TSN will be a key component for Industrie 4.0 and IoT

Automation requirements

- Huge L2 networks (up to 1024 devices and even more)
- Huge network diameter (up to 64 hops) due to line wiring
- Fast isochronous applications (250 μ s and below)
 - E.g. printing machines, wood/metal processing machines, wind craft (down to 31.25 μ s cycle)
 - E2E latency lower than 125 μ s needed (even down to 16 μ s and less)

TSN and automation

- TSN will not substitute fieldbus protocols
- Fieldbus application layer on top needed (e.g. diagnosis, parameterization, system redundancy...)
 - Existing fieldbus protocols migrate their proprietary solutions to TSN (e.g. PROFINET)

Automotive will use TSN in the car

- Reducing cabling in the car
- Audio and brakes on the same network

How to reach low latency

Firefighters want to get to the fire as fast as possible, i.e. with a minimum latency (= real time)

However they may be blocked due to “low priority” traffic



Special rules (e.g. “Rettungsgasse”) allow prioritizing the firefighters

If every traffic participant sticks to the rules, minimum latency is guaranteed



Mapping the principle to Ethernet

At flight altitude, switched Ethernet is a road system with

- Intersections = **bridges**,
- Roads = **links**,
- Cars = **frames**

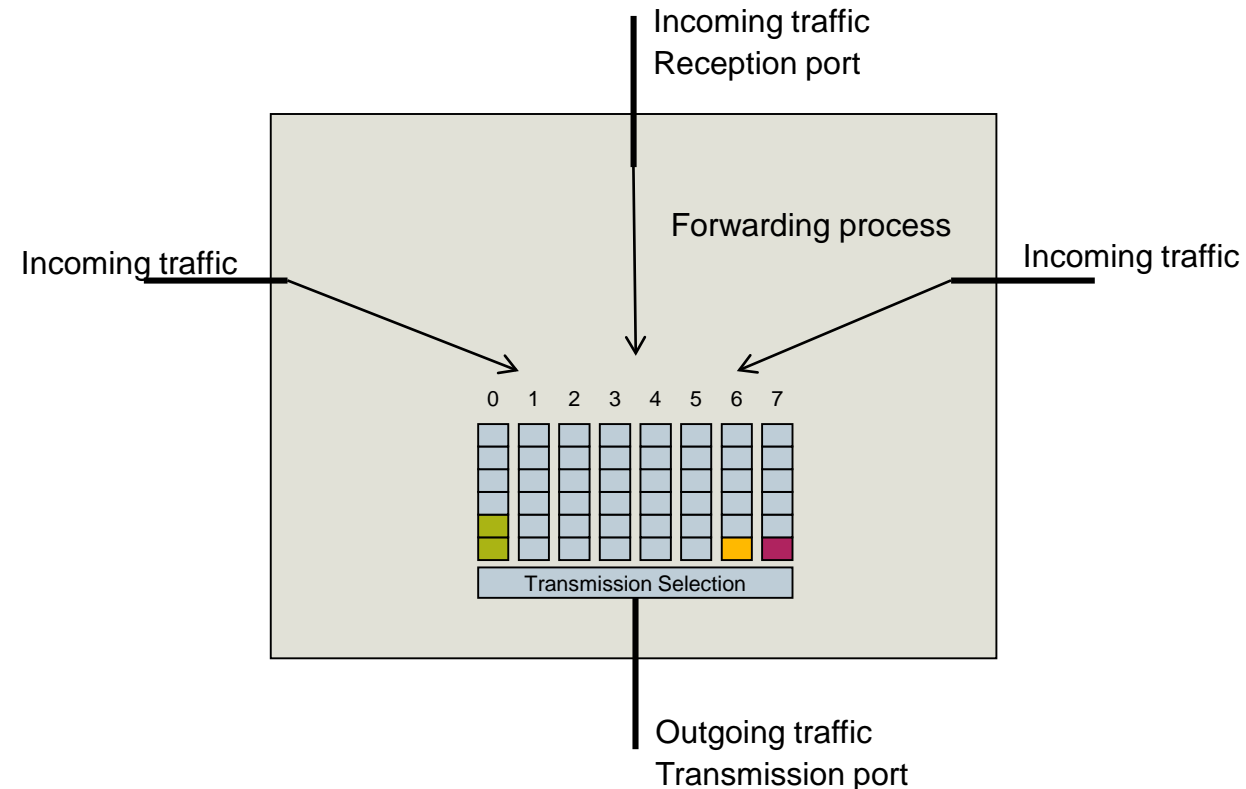
Overtaking on links is forbidden

Overtaking in bridges allowed



Frames (PDUs on the physical layer) carry a priority as part of the VLAN tag (TCI.PCP, 0 to 7)

Frames are assigned to separate queues according to their priority.



Time spend on links / bridges

Link delay: 5.48 ns/m (independent of link speed)

Time needed to transmit an octet:

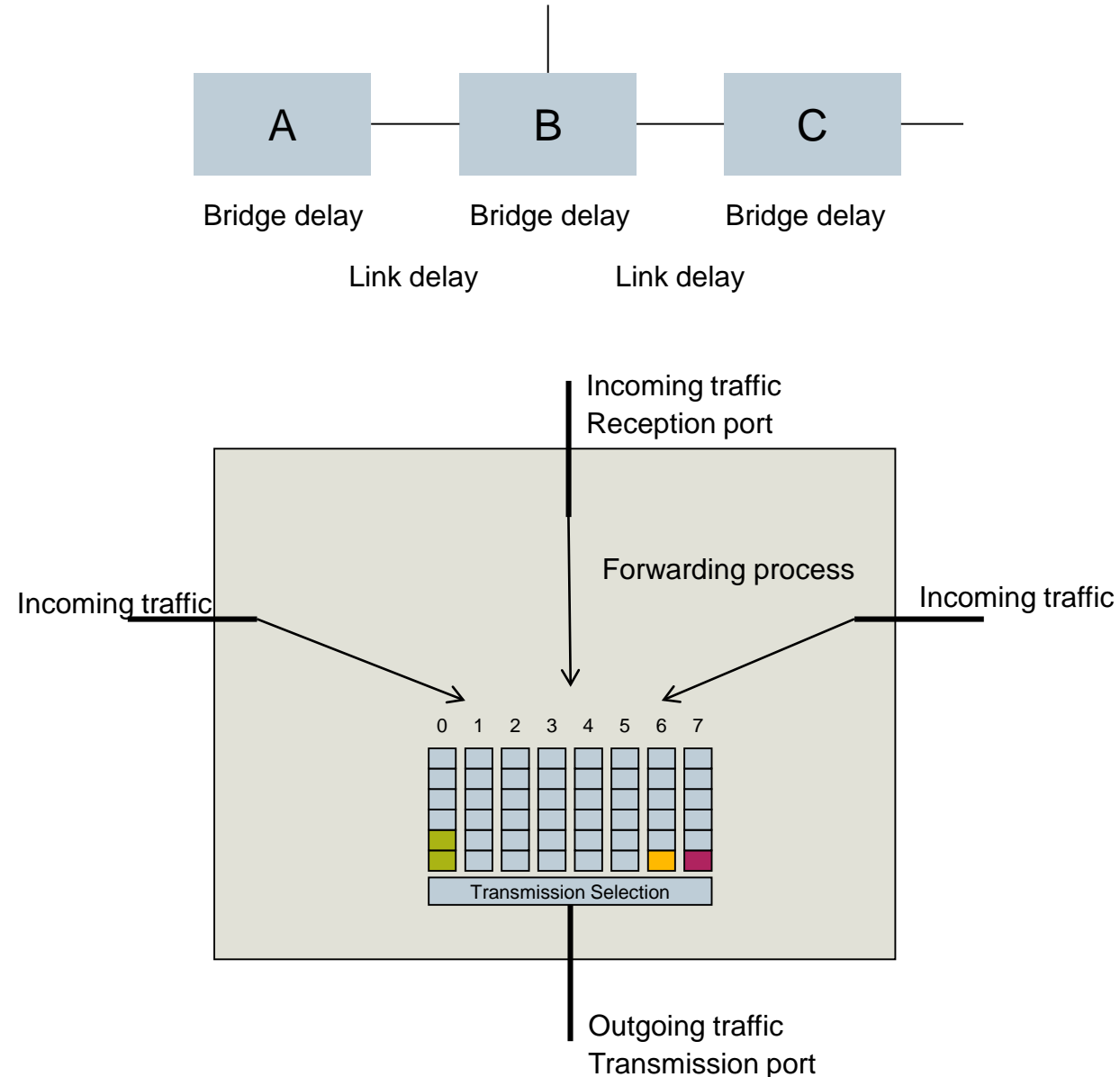
- 100 Mbit/s 80 ns
- 1 Gbit/s 8 ns
- 10 Gbit/s 0.8 ns
- ...
- Min.frame: 84 octets (= 64 + 12 IFG + 8 preamble)

Bridge delay

- Vendor specific
- For automation: 100 Mbit/s ~ 3 μ s, 1 Gbit/s ~ 1 μ s
- For low latency: delayed cut through needed

Notes:

- a blink of an eye lasts around 200 ms (= 200 000 000 ns)
- automation components have typically two ports
- automation ends up in long lines (up to 64 hops)



Traffic shaping: Preemption

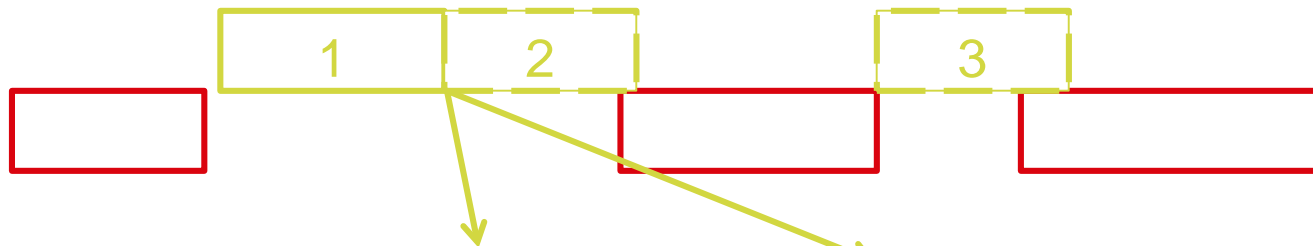
Express frames suspend the transmission of preemptable frames

- Decrease delay variation for express, increase bandwidth for preemptable
- It is link local per hop, i.e., it is not IP fragmentation

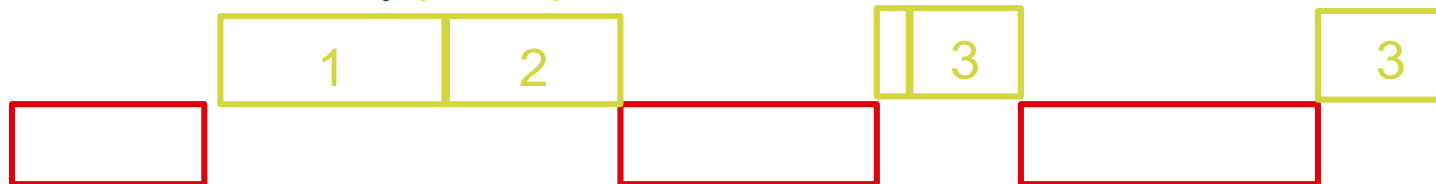
Scheduled **rocks of real-time packets** in network cycle:



Conflict excessively with **non-real-time packet rocks**:



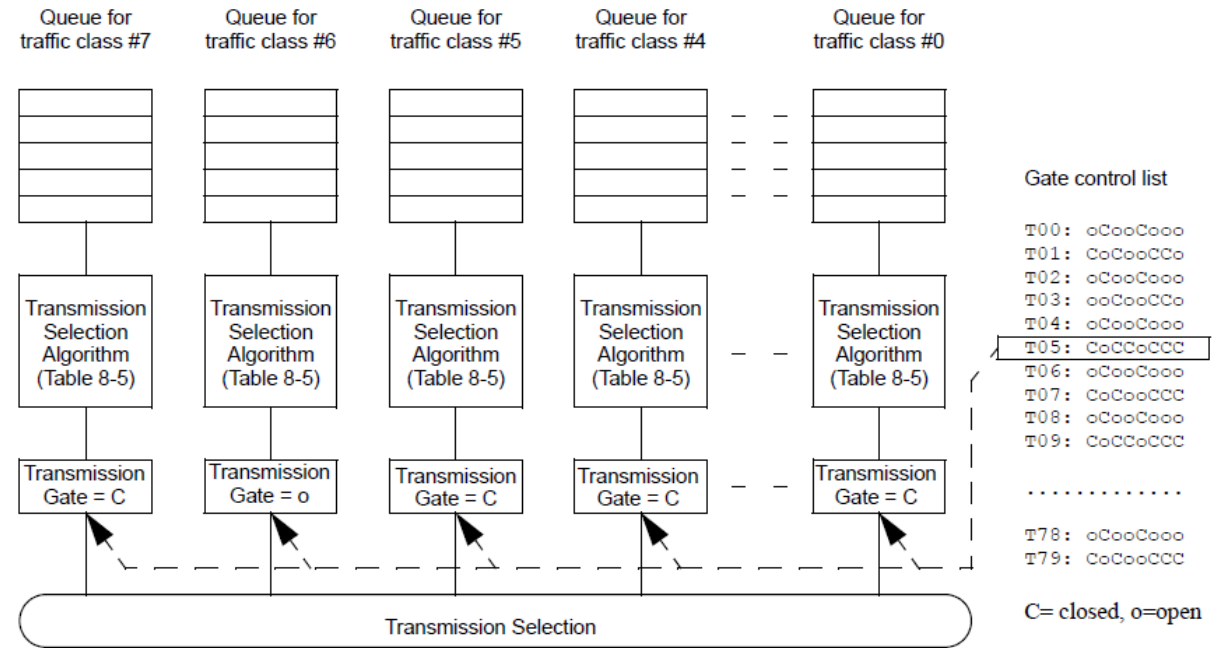
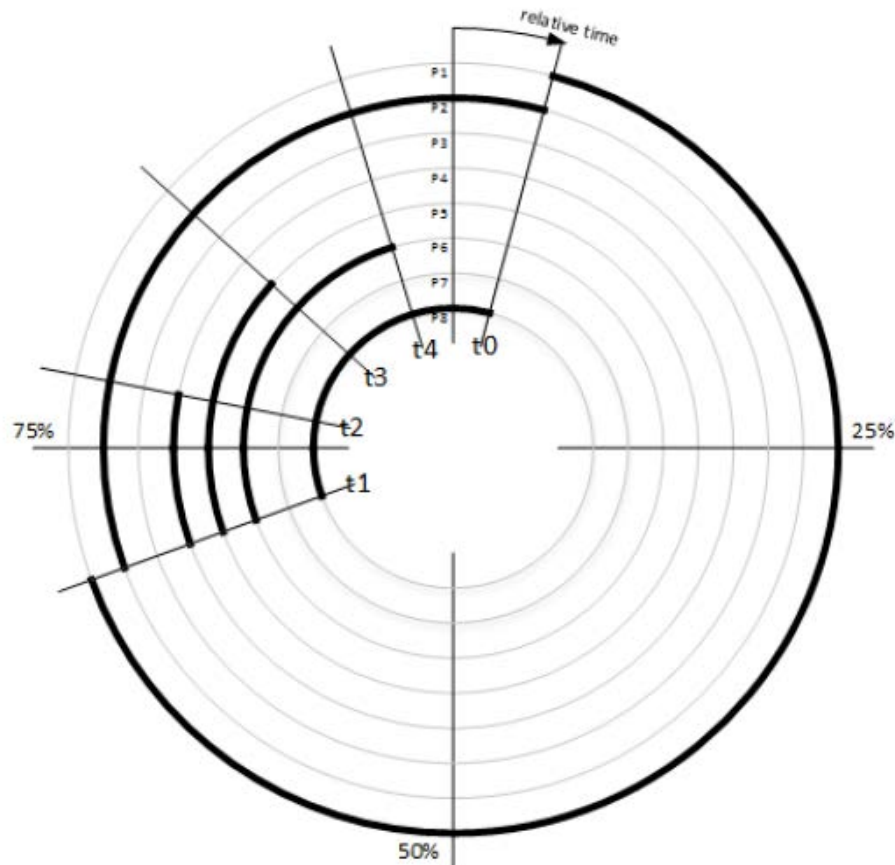
Problem solved by **preemptable sand** between the **rocks**:



Traffic shaping: Time aware shaper

TAS is the “most famous” TSN element

- Transmission gates open and close cyclically (e.g. 1000 μ s network cycle)
- Offers exclusive time periods for traffic classes



Synchronization as a base

Time sync for global time (wall clock) and working clock

- Two separate domains

Reachable accuracy ~ 100 ns network wide

Global time for

- time stamping (e.g. time stamped inputs, log entries)
- time related tasks (e.g. lunch break)

Working clock for

- coordination of bridges and end-stations, when TAS is used
- coordination of applications (e.g. isochronous mode)

Time Sync bases on the following

- Link delay measure delay for a link
- Clock tree defined the sync domain
- Synchronization distributes the master clock to the slaves



Clock tree

The clock tree is a loop free tree with sync capable bridges / end stations; root is the grand master

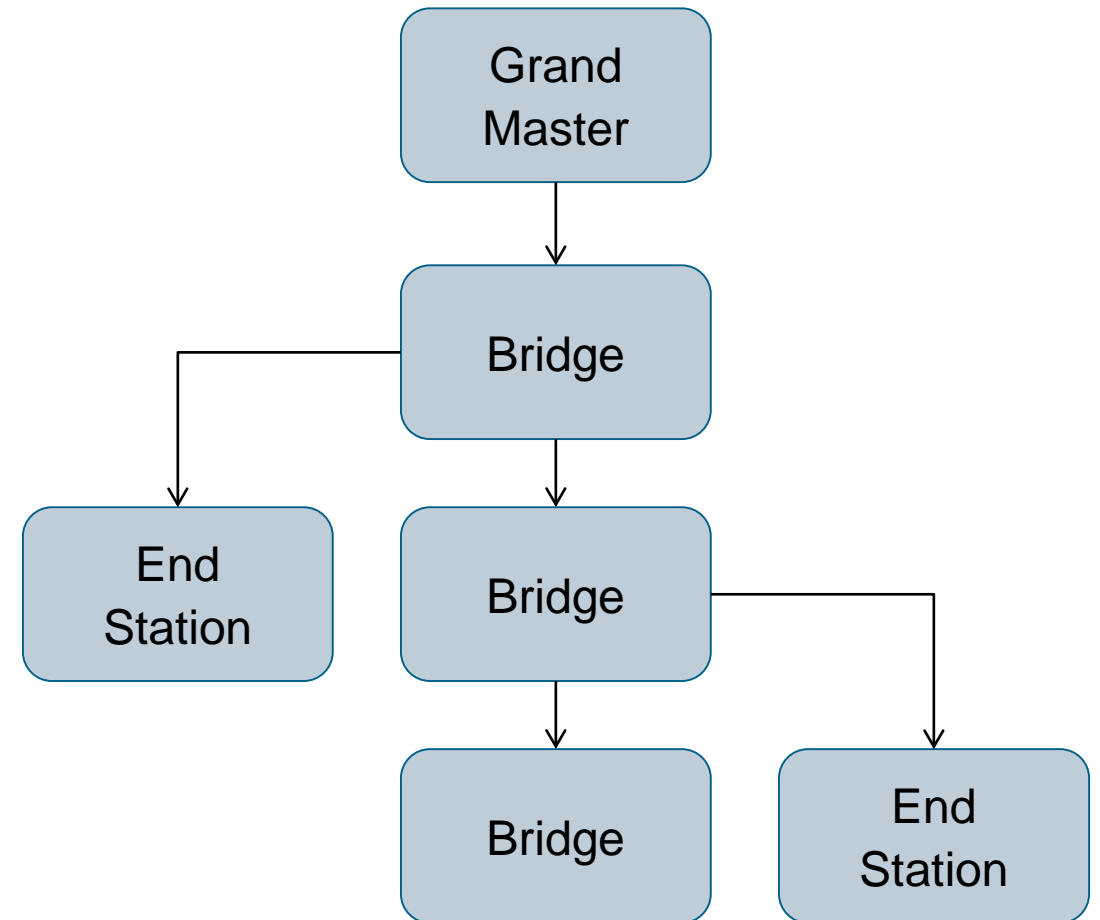
- Best clock wins (default)
- Administered

Sync boundaries (implicit and explicit) allow separation of sync domains

Sync messages are propagated along the clock tree
Link delay measurement is independent of clock tree

Additional features (not covered here) for reliability

- Redundancy of grand masters supported
- Redundancy of paths



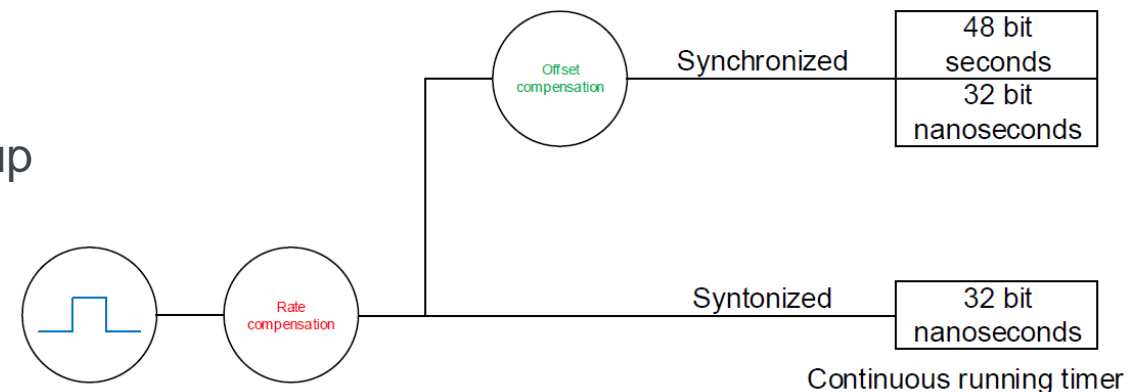
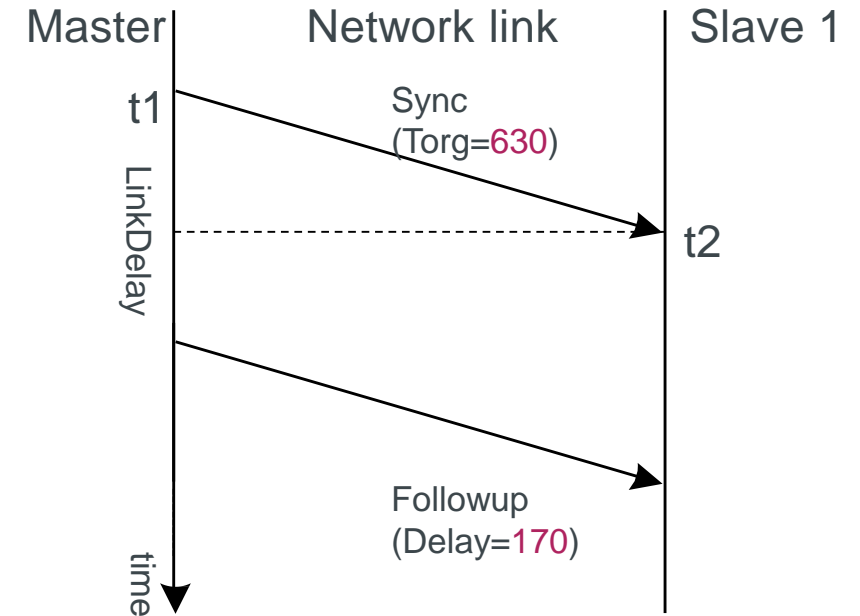
Synchronization

Sync messages are propagated peer to peer.

- Master send periodically sync with Torg
- As sync passes out the PHY, t1 is captured
- t2 is captured when it arrives
- Followup contains the Delay from master to slave (for the master this is t1-Torg)

Slave knows that $t2 + \text{offset}$ should be equal to $\text{Torg} + \text{accumulated Delay} + \text{LinkDelay}$; he adjusts his clock rate to follow the master's clock

Slave adds his delay for propagating the sync message ($= \text{LinkDelay} + \text{internal delay}$) to the Delay field in his followup message (= transparent clock).



Link delay measurement

Each Device has a local timer, which sets a time stamp, when frames are transmitted or received.

The followup frame contains the timestamps t_2 and t_2' .

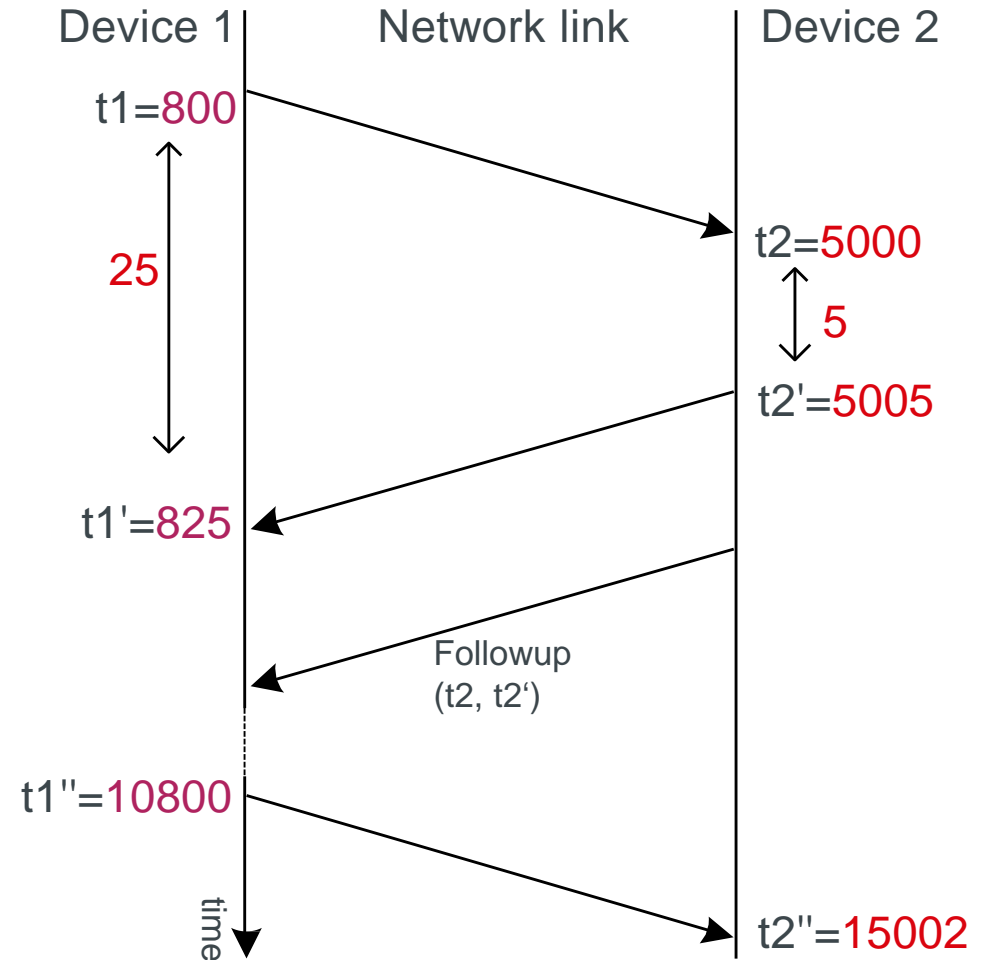
Device 1 is able to compute the LinkDelay as follows:

$$\begin{aligned} \text{LinkDelay} &= \frac{(t_1' - t_1) - (t_2' - t_2)}{2} \\ &= \frac{(825 - 800) - (5005 - 5000)}{2} = 10 \end{aligned}$$

LinkDelay is measured cyclically; by building the median over all measurements, inaccuracies are such smoothed out.

Rate compensation improves additionally

$$\text{neighborRateRatio} = \frac{(t_1'' - t_1)}{(t_2'' - t_2)} = \frac{10000}{10002} = 0.998$$

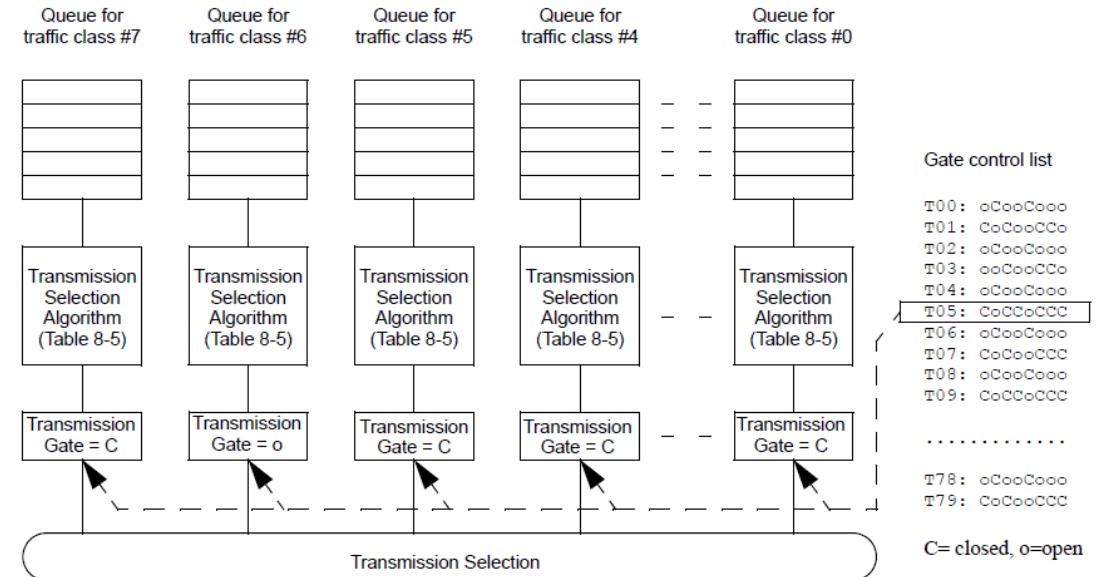


Extremely low loss: Dedicated resources by scheduling

Bridges hold resources per traffic class (= queue)

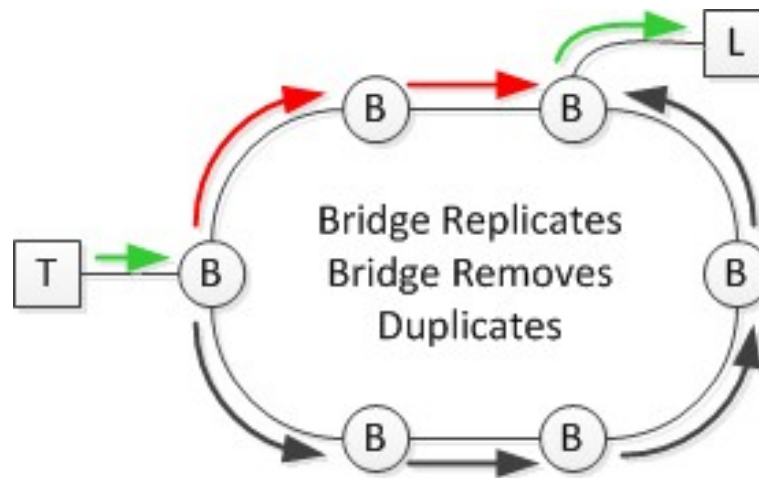
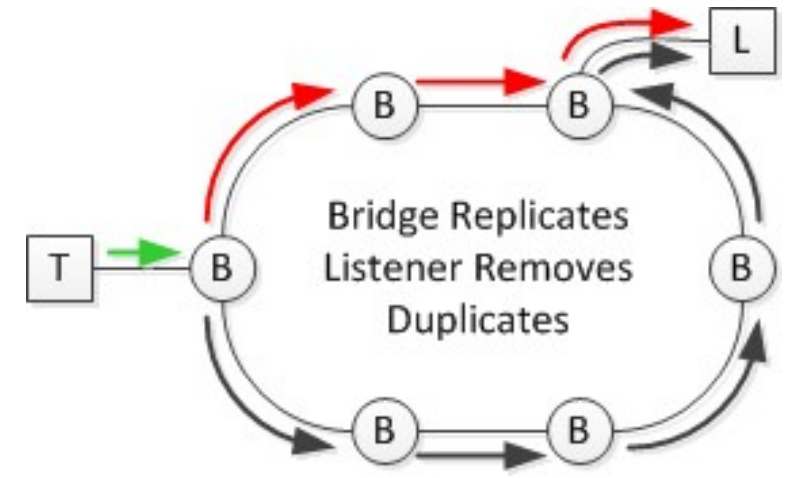
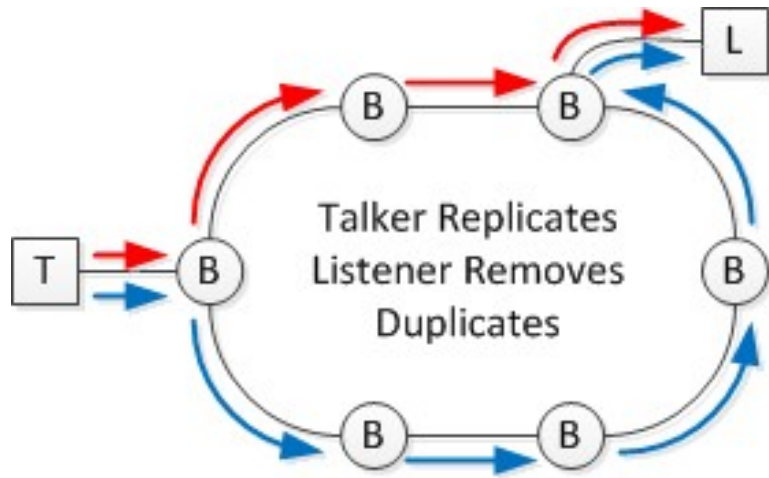
Real time communication is modeled as a stream

- One talker, 1 to n listeners
- TrafficSpecification describes talker's requirements (interval, maximum amount of data, max latency...)
- Stream's data plane uses a fixed tree
- Stream granted only, if
 - dedicated resources along the tree are available
 - AND
 - latency requirements are fulfilled



Extremely low loss: Seamless redundancy

IEEE 802.1CB



Is TSN the superhero?

TSN provides

- a significant enhancement to standard Ethernet
- shaping and scheduling mechanisms for bounded low latency and jitter
- due to being IEEE
 - backward compatibility
 - vendor-neutrality

Benefits for customers

- Multiple independent applications use real time in the same network and even over the same links
- Plug and play

Benefits for vendors

- Scalable from 10 Mbit/s up to x Gbit/s
- Standard Ethernet ASICs offer real time
 - Specific ASICs are no longer needed (e.g. PROFINET IRT, Ethercat, Sercos...)
- Cheap hardware expected (“TSN inside”)
 - Automotive uses TSN → mass production