WFCS 2021

Out-of-sync Schedule Robustness for Time-sensitive Networks

Silviu S. Craciunas, Ramon Serna Oliver
TTTech Computertechnik AG
9-11 June 2021

TTTech

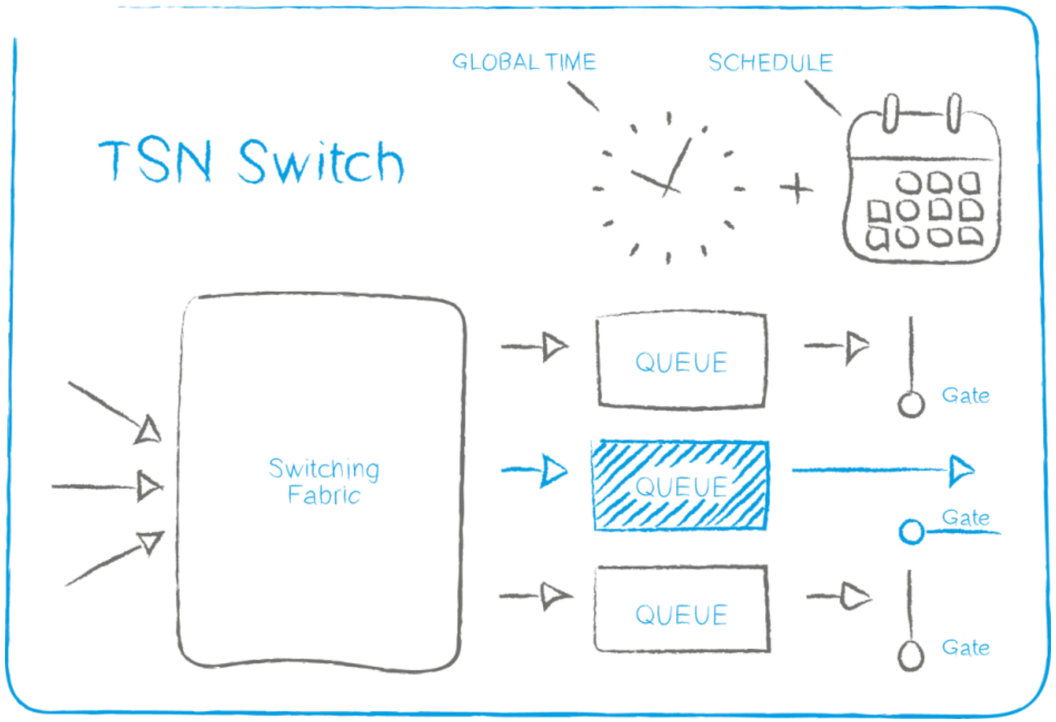
TTTech Group

- founded in 1998, headquartered in Vienna, Austria, with 19 offices in 14 countries worldwide
- develops safe networked computing platforms for a more connected, automated and sustainable world
- is based on 20+ years of expertise in scheduled data communication and fault-tolerance principles
- strong collaboration with academia
  36 ongoing research projects

2300+
employees
in TTTech Group

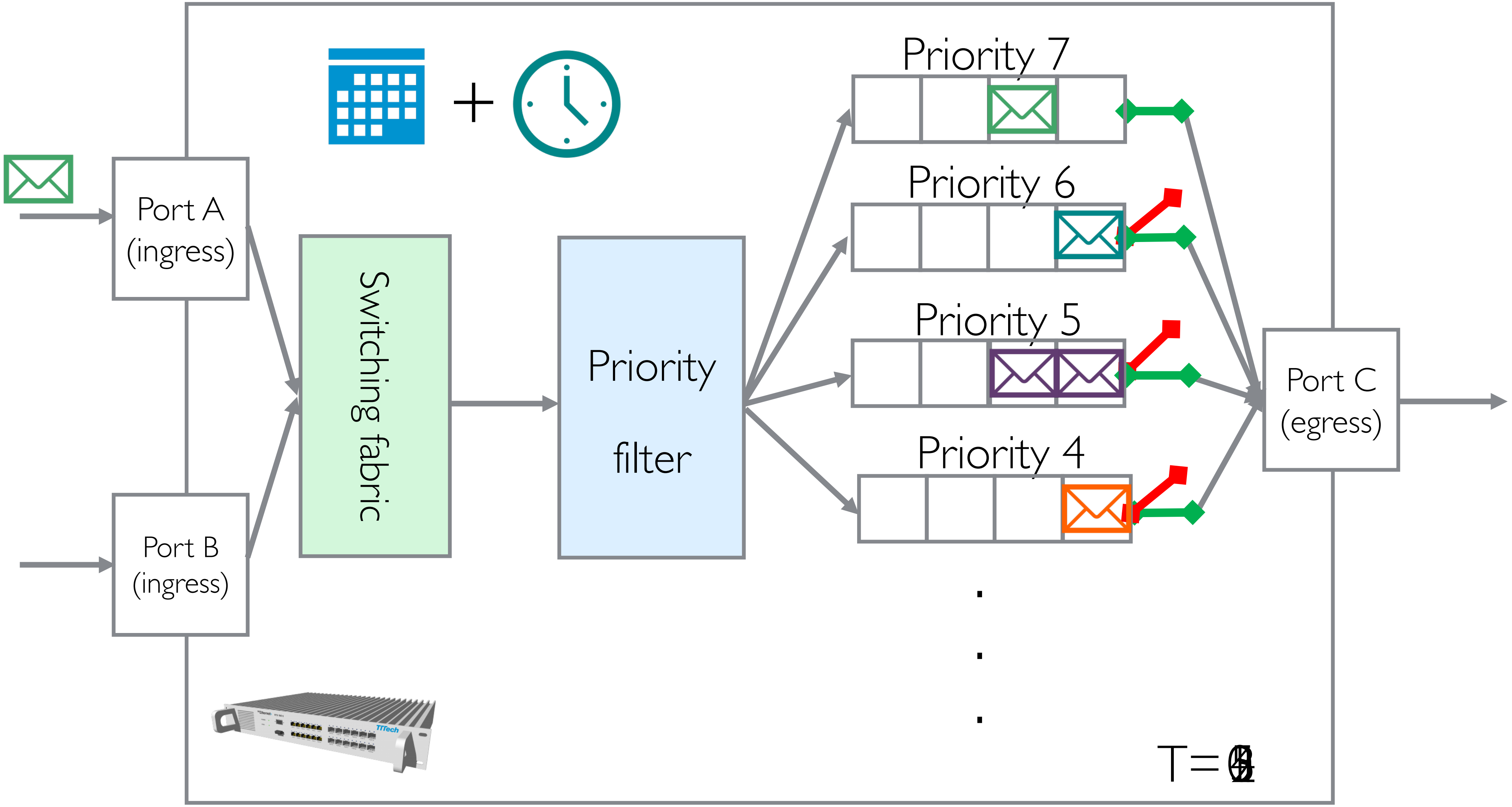
What is a Time-Sensitive Network (TSN)?

Time Sensitive Networking covers a set of Ethernet sub-standards and amendments currently defined in the IEEE 802.1 TSN task group



- Critical traffic guarantees through time synchronization and scheduled frame transmission
- TSN supports the coexistence of critical and non-critical traffic over the same communication backbone.

TSN (Qbv) switch

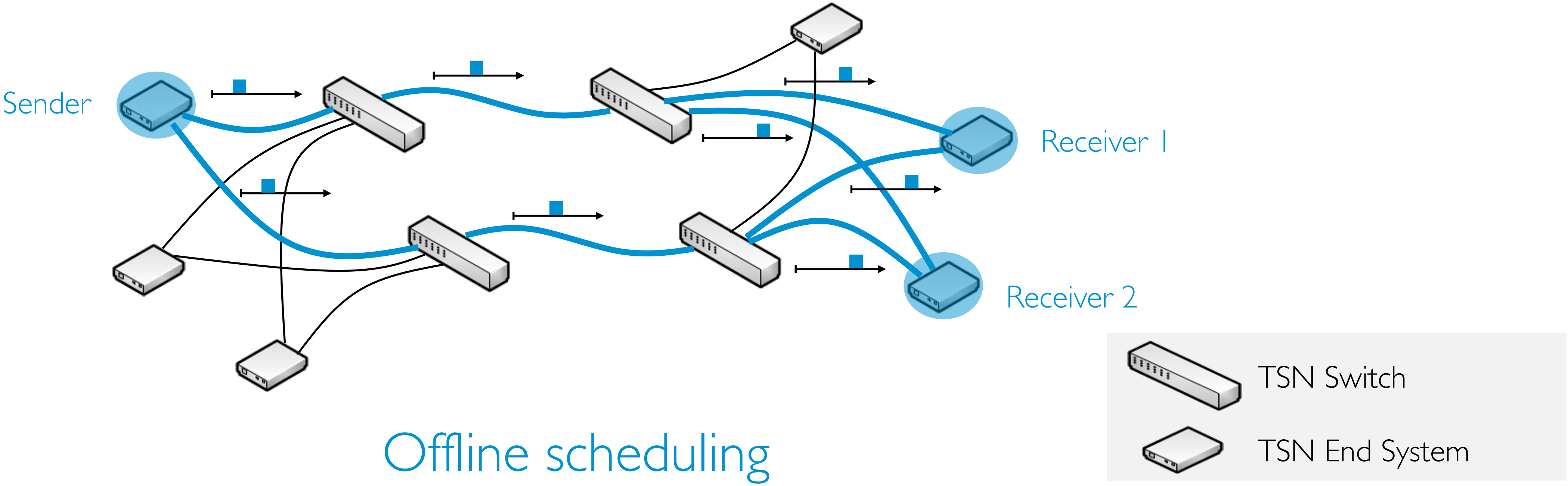


TSN schedule

The TSN (Qbv) schedule defines open and close events for the **Gate Control List (GCL)** in each output port of every TSN device in the network

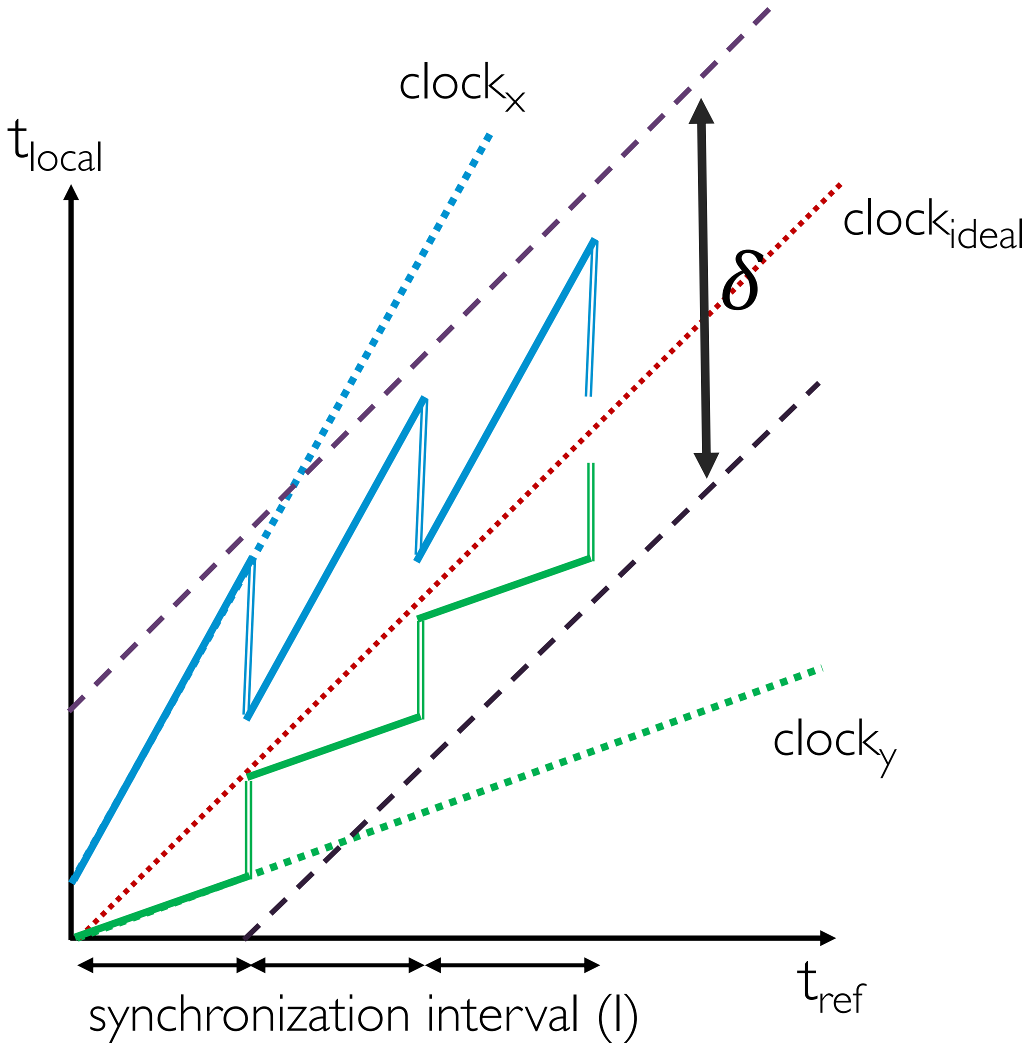
The schedule is build **off-line** taking into account the **maximum** possible **clock deviation** (precision) when all clocks are synchronized.

The schedule enforces a **deterministic behaviour** of frame transmission and reception.



Offline scheduling

Clock drift



- Each clock C_i has a drift rate ρ_i
- The maximum clock drift in the network is $\rho_{max} = max_i \{\rho_i\}$
- Typical values of ρ_{max} are 50 - 100ppm, i.e., between 50 and 100 µs/sec
- All clocks need to be synchronized with a certain rate - synchronization interval (I)
- The envelope, I, and ρ_{max} determine the value of the network precision δ
- The precision is a safe upper bound on the deviation between any two clocks in the network

Time synchronization in TSN

IEEE 802.1AS

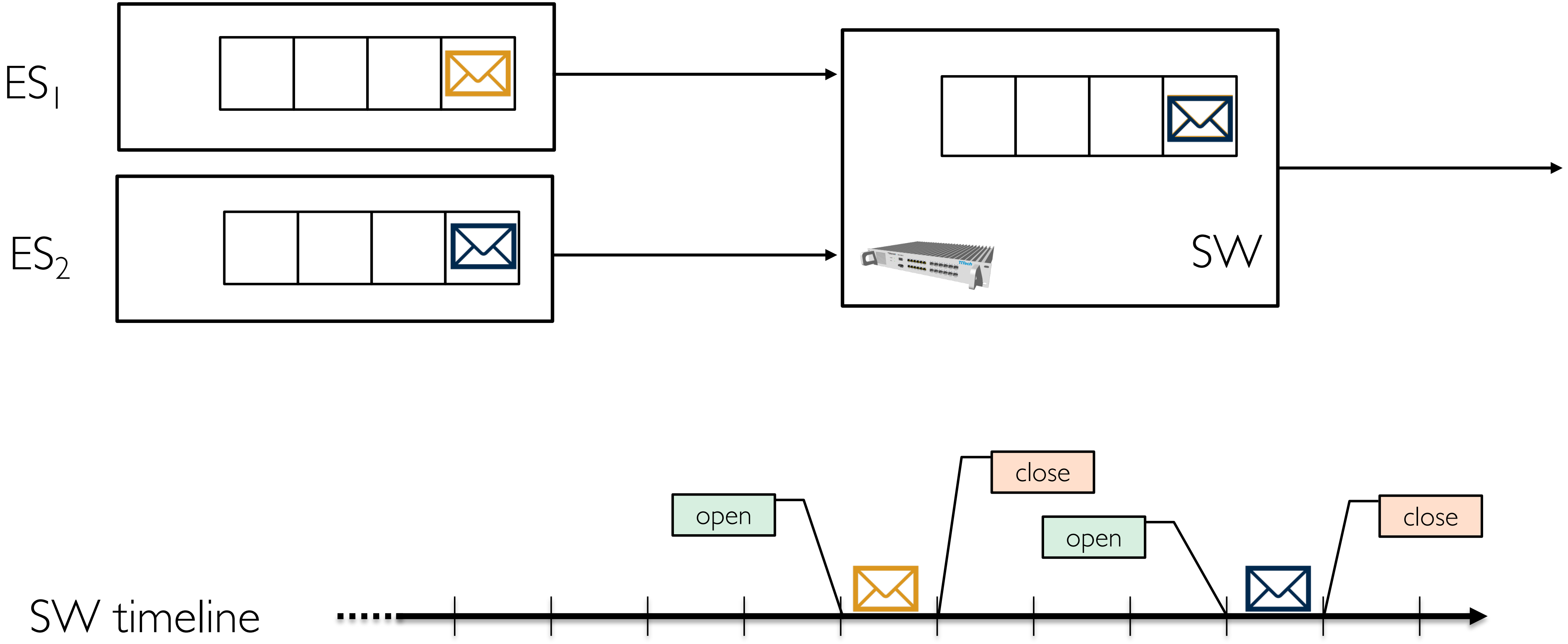
- clock synchronization protocol that provides a common clock reference for all network devices called GrandMaster (**GM**)
- each clock is at most δ (**precision**) away from the GM time
- Best Master Clock Algorithm (**BMCA**) constructs the synchronization spanning tree with the GM as root node
- time is propagated from the root to the leaves
- each bridge corrects the received time by adding the **propagation delay** and **residence time** in the bridge and forwards the corrected time to the next nodes in the tree
- if the GM node fails, a new GM has to be **elected**, and the spanning tree has to be recreated via the BCMA

IEEE 802.1AS-rev

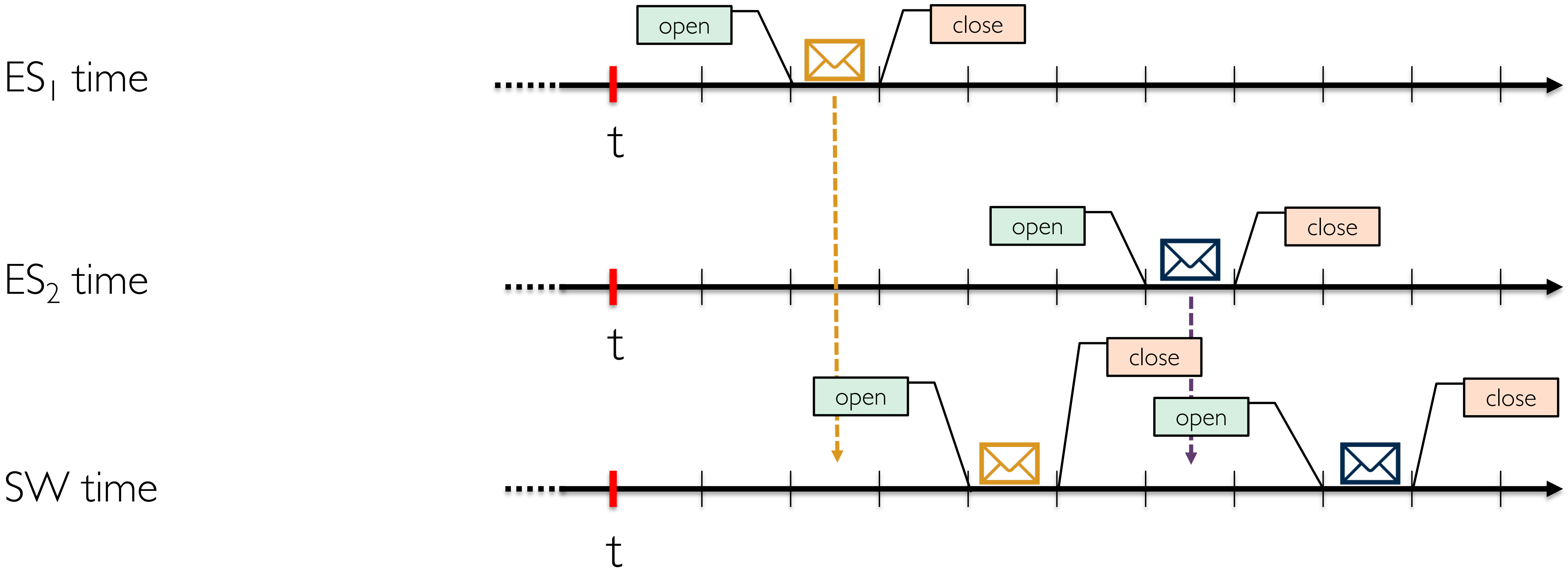
- is an update to 802.1AS
- introduces multiple domains:
  - domains are fully independent
  - separate BMCA
- introduces multiple time scales
- introduces **redundancy**: configure redundant paths and redundant GMs (hot standby)

Synchronized deterministic network

TTTech

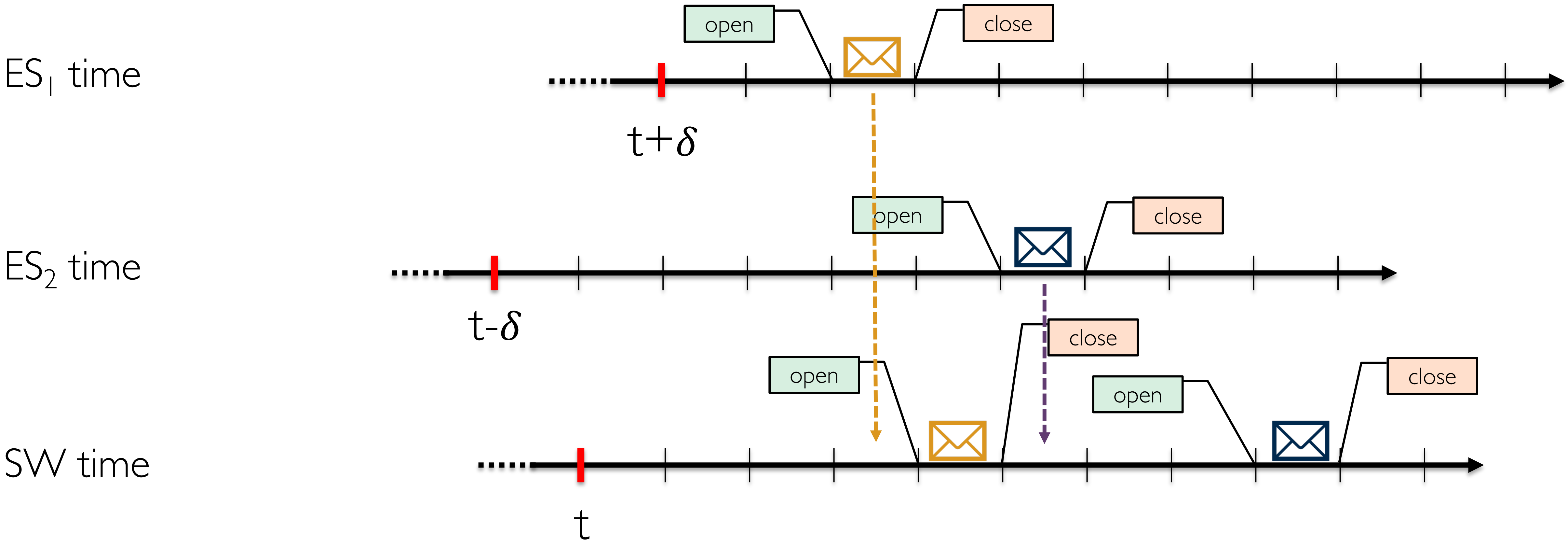


Perfectly synchronized network

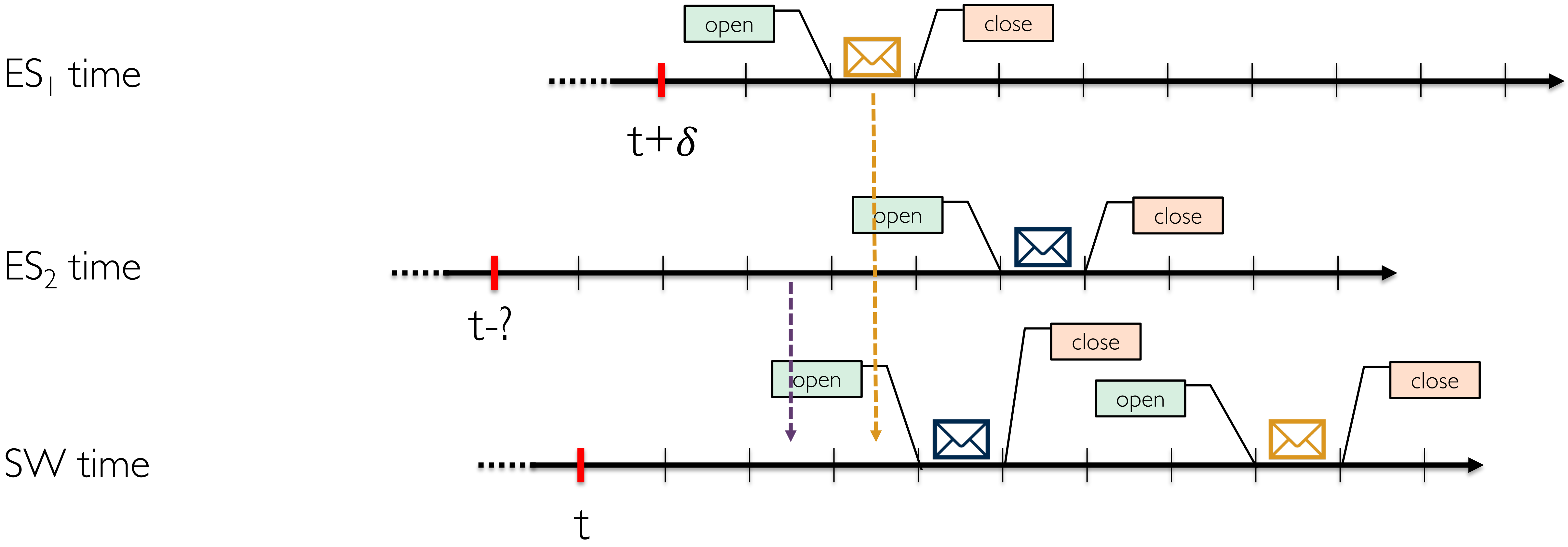


Synchronized network

TTTech



Synchronization loss



Out-of-sync interval

TTTech

Can we compute an upper bound on the time until the network is resynchronized in 802.1AS-rev?

$$t_{resync} = \delta_t + \delta_{hop} \times N_G$$

timeout bound for out-of-sync detection

configuration item in 802.1AS-rev: announceReceiptTimeout × announceInterval

upper bound for the BMCA to run and propagate information on each node

safe upper bound of 1sec per hop

number of hops for the longest path out of all the possible spanning trees

safe upper bound as a function of the topology and the subset of eligible GM candidate nodes

Out-of-sync drift

$$\Delta_s = 2 \times \rho_{max} \times t_{resync}$$

Example:

- N_G =3, δ_t =3s, δ_{hop} =1s, and ρ_{max} =100ppm
- the upper bound on the deviation between any two clocks following a GM failure at the point of re-synchronization is 1200µs

$$\boxed{\delta + \Delta_s}$$

If we can generate a schedule with an extended precision parameter, we can effectively maintain determinism even when sync is temporarily lost → *schedule robustness*

Schedule robustness

TTTech

It is trivial to extend the relevant constraints from our previous work✤ to include the robustness parameter added to the precision when generating the schedule tables.

$$\forall s_i \in \mathcal{S}, \forall (v_a, v_x), (v_x, v_b) \in R_i :$$
$$\phi_i^{(v_x,v_b)} - (\phi_i^{(v_a,v_x)} + l_i^{(v_a,v_x)}) \geq \delta + \Delta_s.$$

$$\forall \alpha \in \left[0, hp_i^j / T_i\right), \forall \beta \in \left[0, hp_i^j / T_j\right) :$$
$$(\phi_j^{(v_y,v_a)} + \beta \times T_j - \phi_i^{(v_a,v_b)} - \alpha \times T_i \geq \delta + \Delta_s) \vee$$
$$(\phi_i^{(v_x,v_a)} + \alpha \times T_i - \phi_j^{(v_a,v_b)} - \beta \times T_j \geq \delta + \Delta_s).$$

$$\forall s_i \in \mathcal{S} : \phi_i^{dest(s_i)} + l_i^{dest(s_i)} - \phi_i^{src(s_i)} \leq D_i - (\delta + \Delta_s).$$

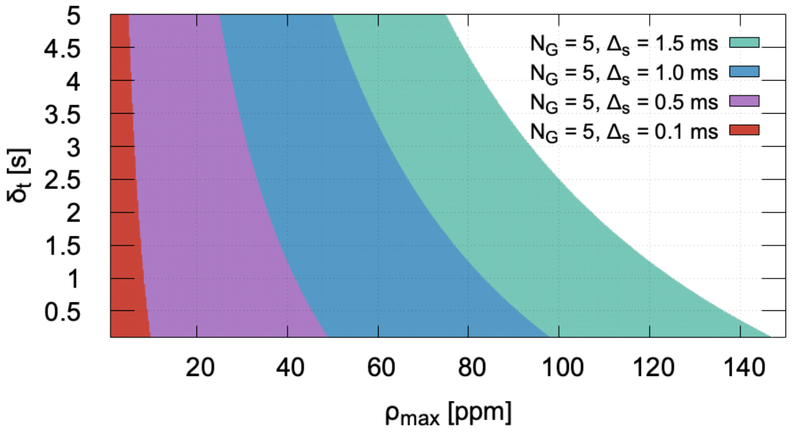
✤ our previous work:

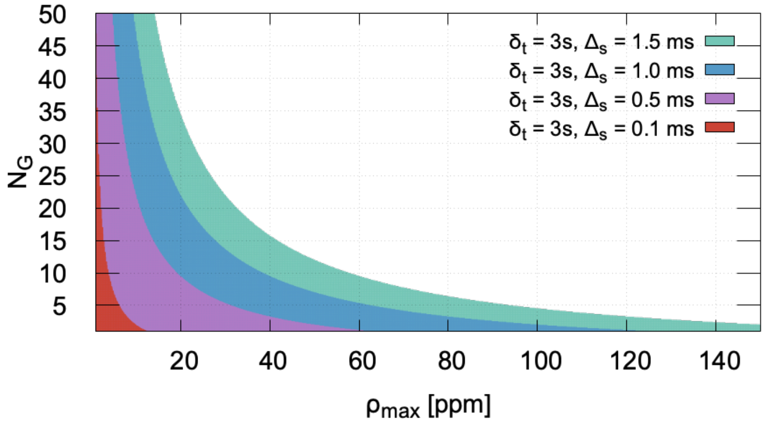
- S.S. Craciunas, R. Serna Oliver, M. Chmelik, and W. Steiner - Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks
  In Proc. 24th International Conference on Real-Time Networks and Systems (RTNS), pp. 183-192, ACM, 2016.
- R. Serna Oliver, S.S. Craciunas, and W. Steiner - IEEE 802.1Qbv Gate Control List Synthesis using Array Theory Encoding
  In Proc. 24th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pp. 13-24, IEEE, 2018.

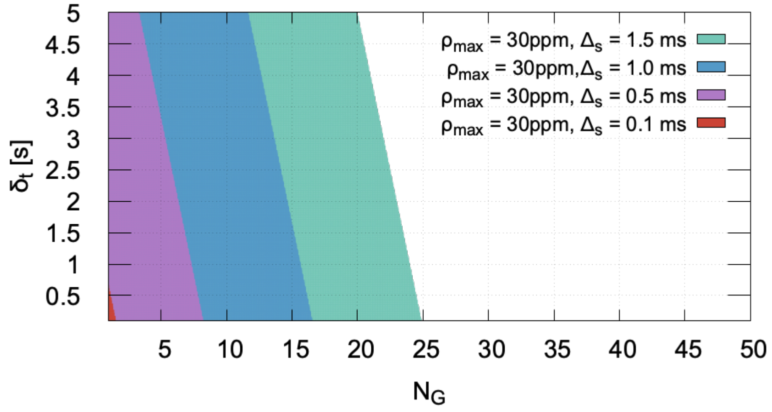
Design-space exploration

TTTech

- we transform the out-of-sync drift Δ_s to be a variable that is computed by the scheduler
- maximizing the out-of-sync drift Δ_s can help mitigate cascading failures
- selecting a value for one parameter will constrain the possible values for the other dimensions
- the easiest parameter to change is the out-of-sync detection bound δ_t
- we show the configuration space for different example networks below





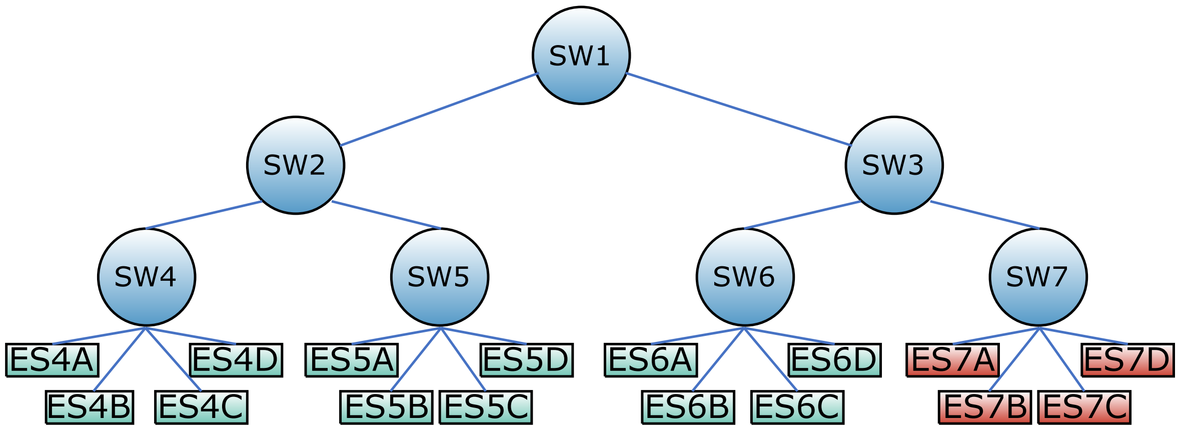


Schedule tool and experiment setup

TTTech

Satisfiability
Modulo
Theories

- satisfiability of logical formulas in first-order formulation
- background theories $\mathcal{LA}(\mathbb{Z})$ $\mathcal{BV}$
- variables $x_1, x_2, \ldots, x_n$
- logical symbols $\vee, \wedge, \neg, (,)$
- non-logical symbols $+, =, \%, \leq$
- quantifiers $\exists, \forall$
- optimization criteria: Optimization Modulo Theories [Bjørner@TACAS15]



- Z3 SMT/OMT solver v.4.8.10
- 2 dedicated queues for 802.1Qbv
- macrotick fixed at 1µs
- a constant link latency of 1µs
- homogeneous link speeds of 1Gbps
- Intel i7-8650U CPU @1.90GHz with16GB RAM

Experiments - schedulability

ρ_{max} [ppm]	100	100	50	50	5	5
$\delta_t[s]$	3	1	3	1	3	1
$\Delta_s[\mu s]$	1200	800	600	400	60	40
Max util. [%]	5.76	5.76	5.76	5.76	5.76	5.76
Runtime [ms]	437	359	343	390	389	422
Schedulability	true	true	true	true	true	true

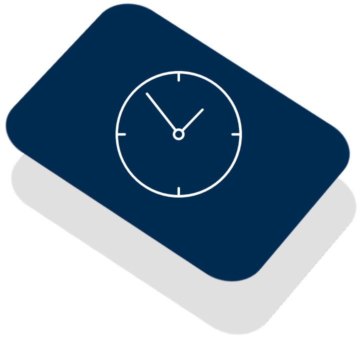
ρ_{max} [ppm]	100	100	50	50	5	5
$\delta_t[s]$	3	1	3	1	3	1
$\Delta_s[\mu s]$	1200	800	600	400	60	40
Max util. [%]	11.52	11.52	11.52	11.52	11.52	11.52
Runtime [ms]	219	312	391	407	406	422
Schedulability	false	false	true	true	true	true

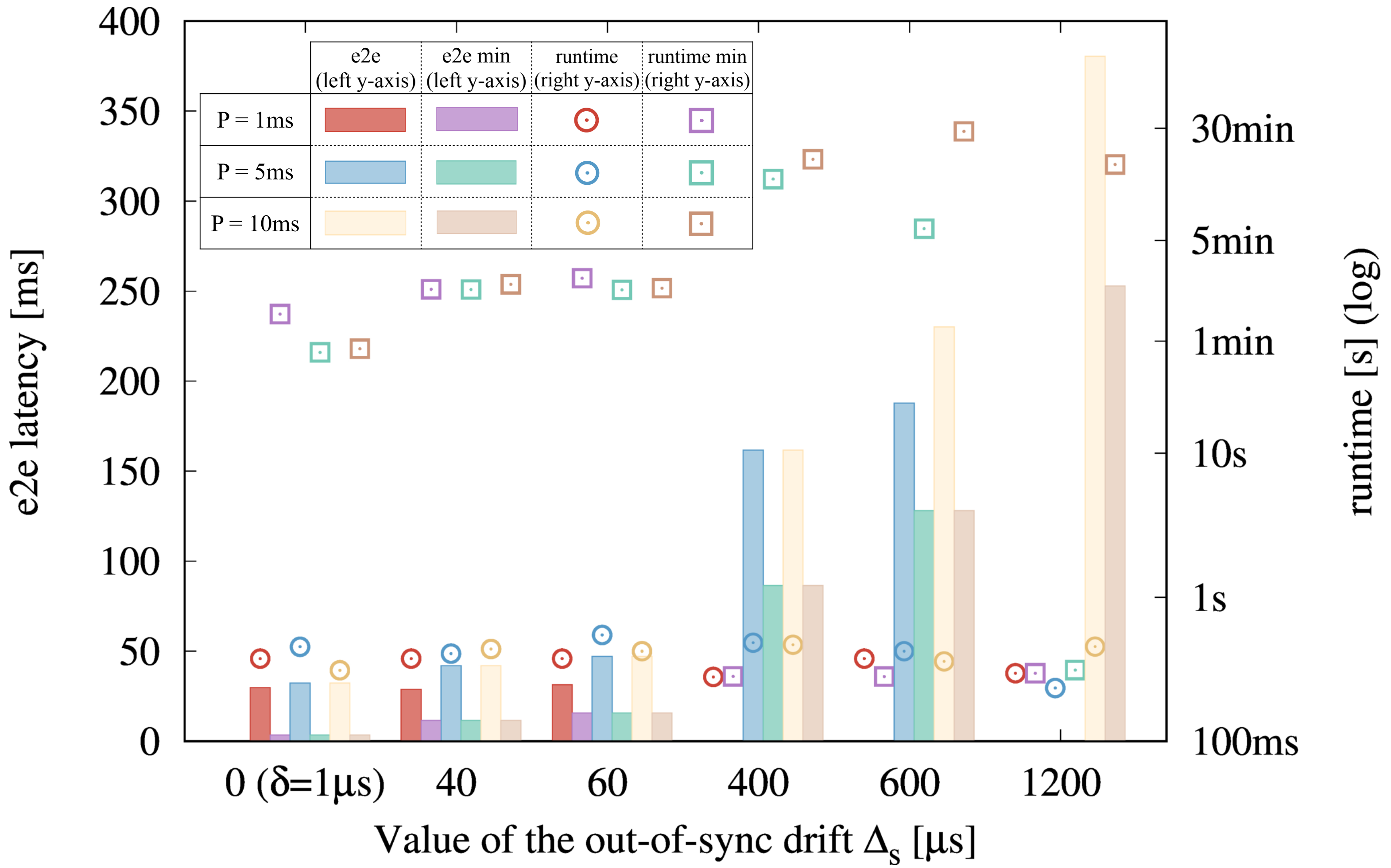
The higher the link utilization, the less robustness can be added

» Schedulability is reduced

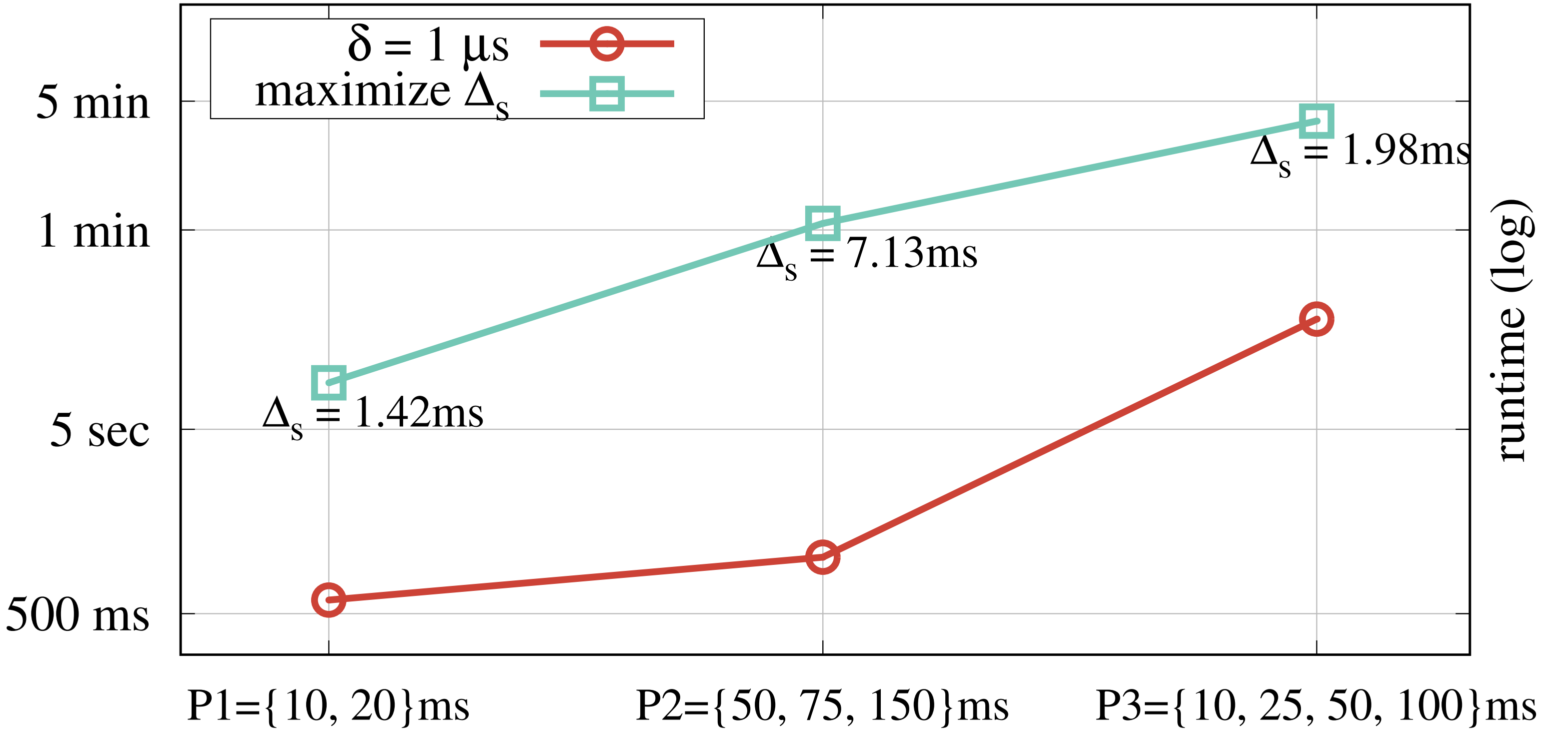
ρ_{max} [ppm]	100	100	50	50	5	5
$\delta_t[s]$	3	1	3	1	3	1
$\Delta_s[\mu s]$	1200	800	600	400	60	40
Max util. [%]	57.6	57.6	57.6	57.6	57.6	57.6
Runtime [ms]	218	249	203	234	343	390
Schedulability	false	false	false	false	true	true

Experiments – end-to-end latency





Experiments - schedule synthesis time



Thank you!

Email:

silviu.craciunas@tttech.com