

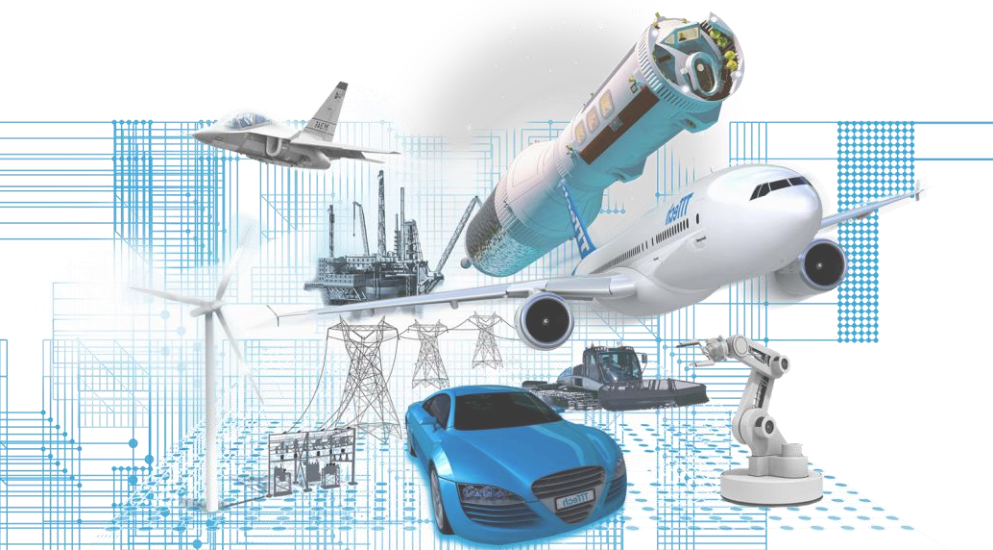
Optimization challenge

The TSN scheduling problem

Silviu S. Craciunas
Corporate scientist/Systems Architect

TTTech Computertechnik AG

September 25, 2019



What do these markets have in common?

Reliable networks and safety controls from TTTech

Boeing 787



NASA Orion



Vestas Wind Turbines



Audi Piloted Driving



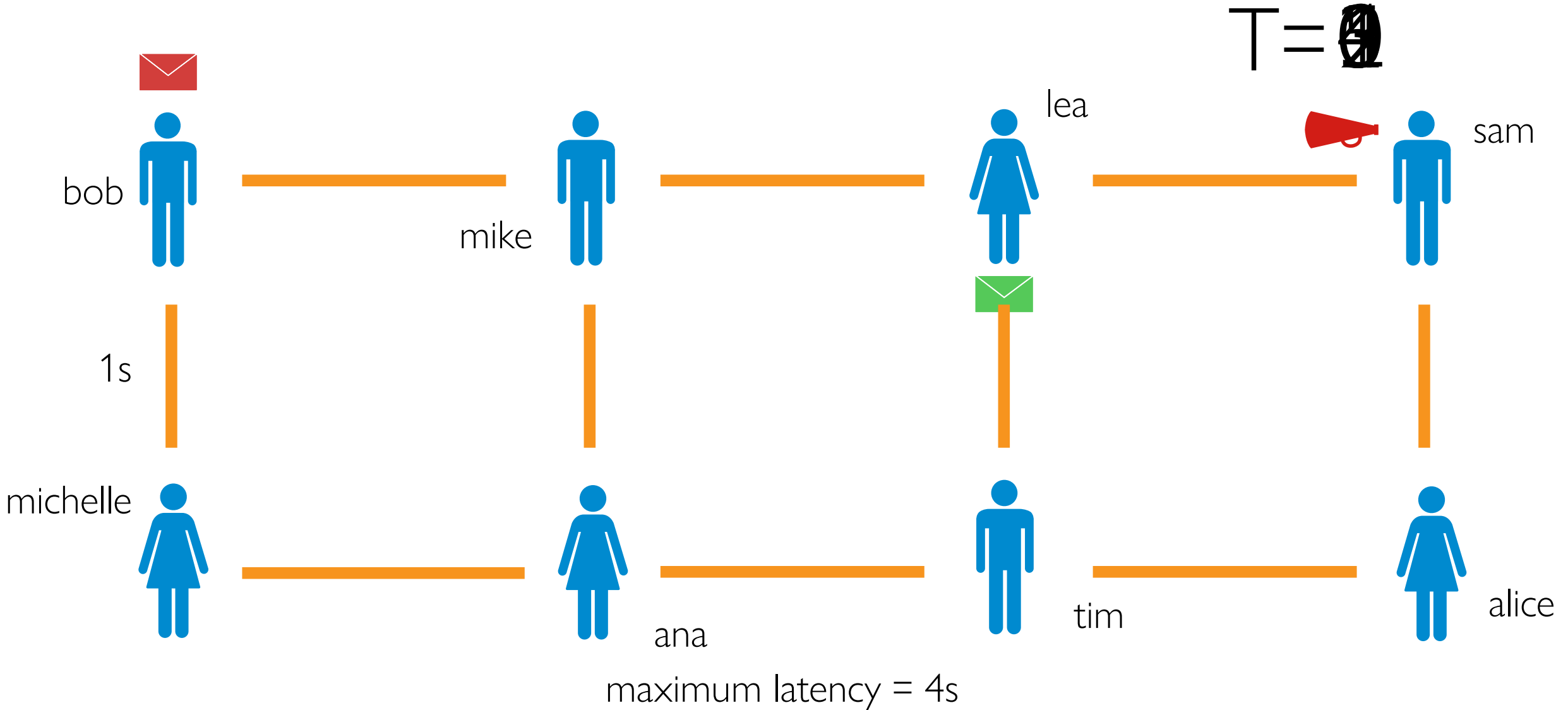
Prinoth Leitwolf



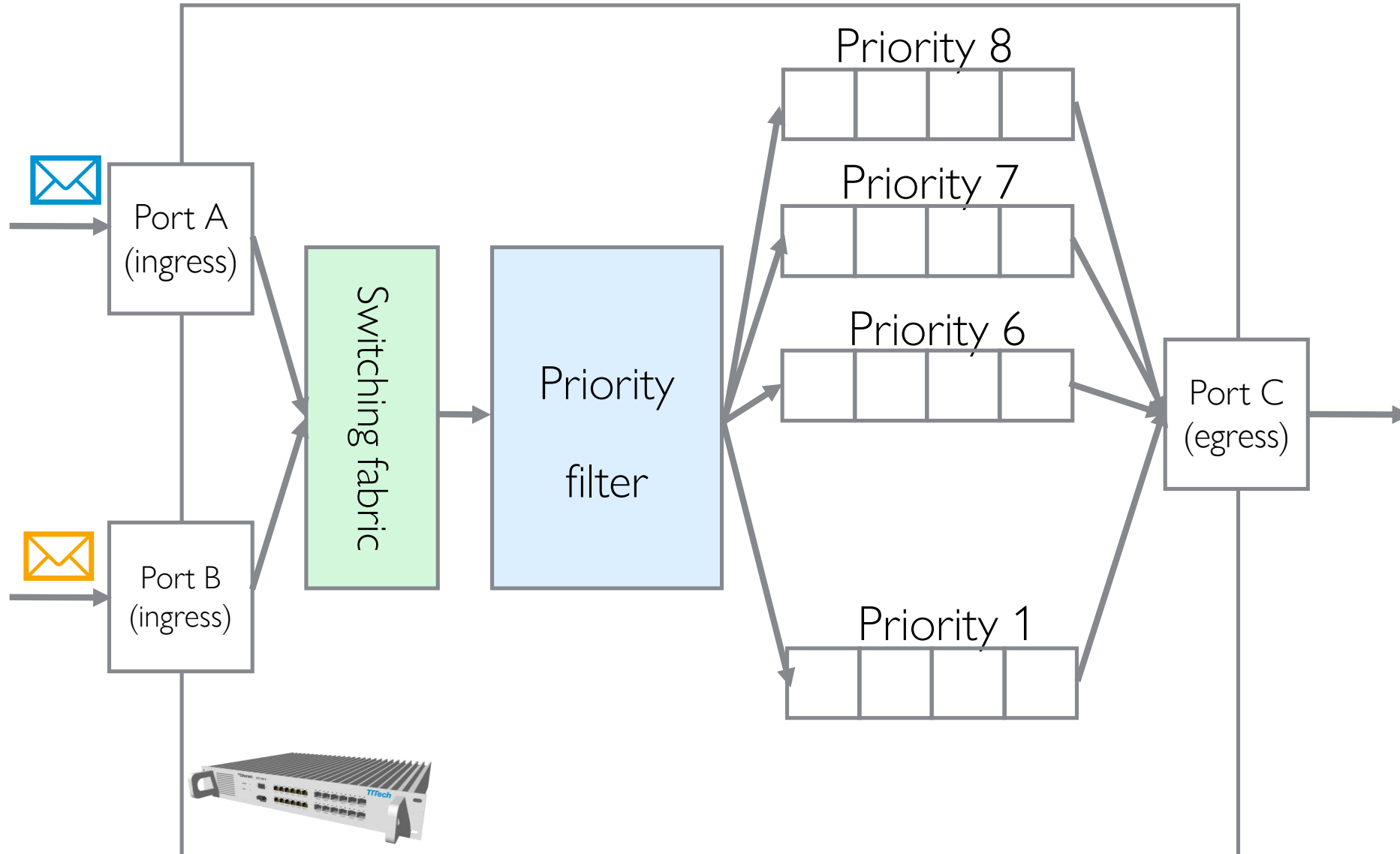
Thales Railway Signalling

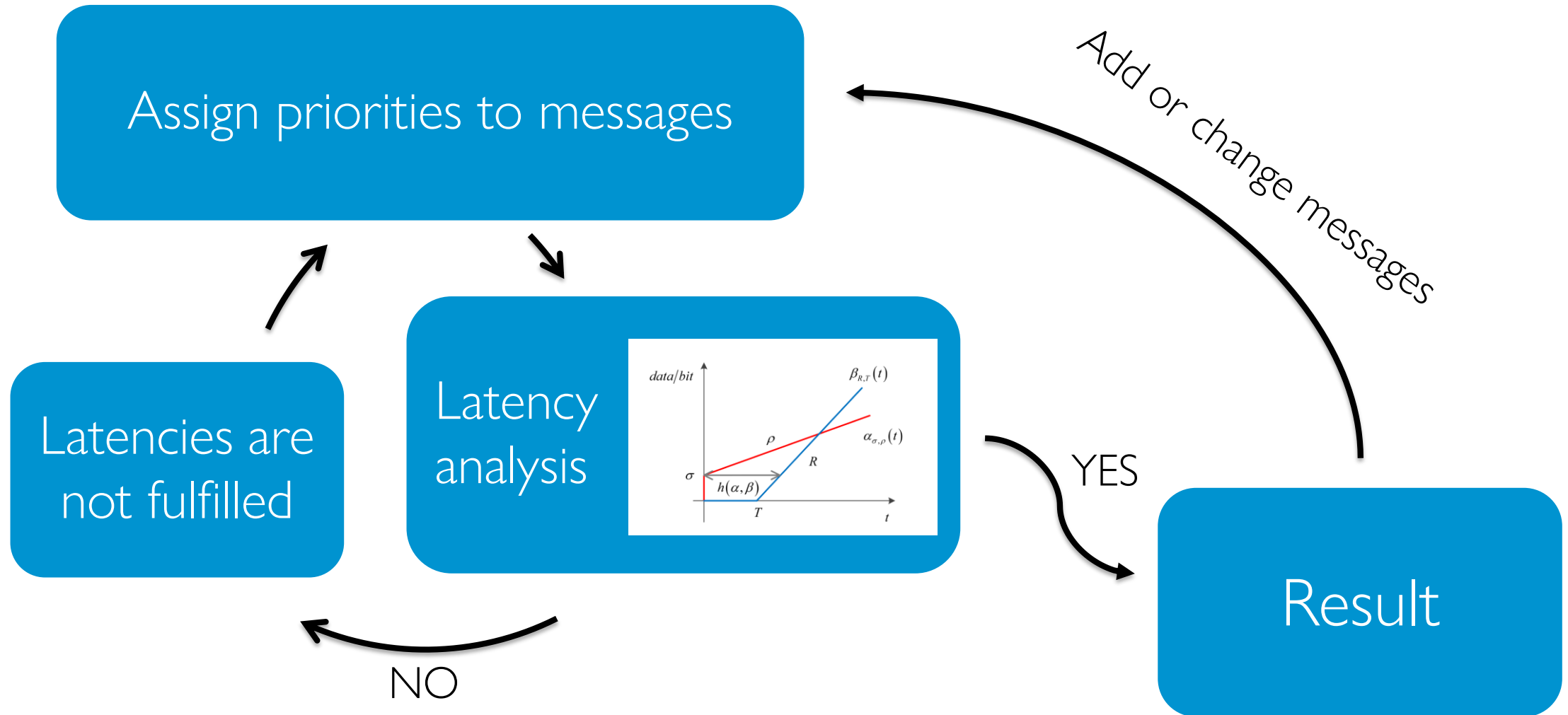


Experiment



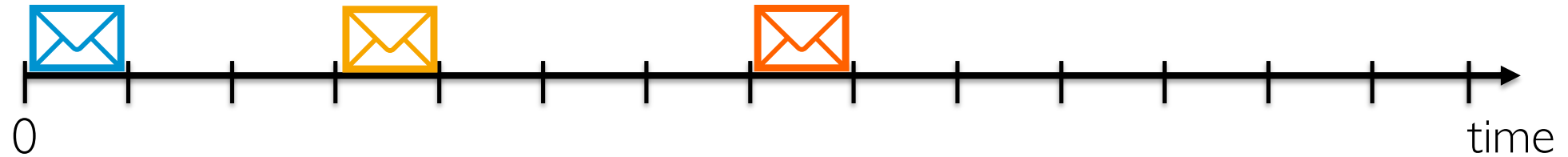
Priority switch







Sending and receiving of frames is done according to a global schedule.



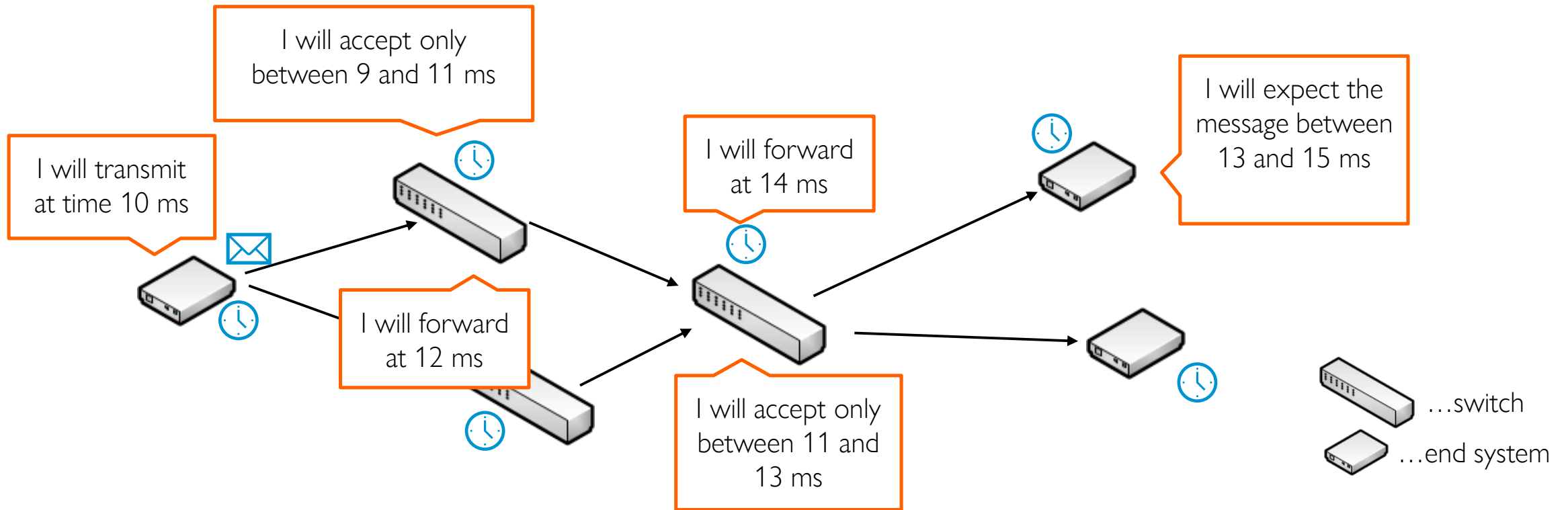
- Correct by design (schedule construction)
- Compositional (adding flows can be done incrementally)
- Minimal jitter and delay bounds

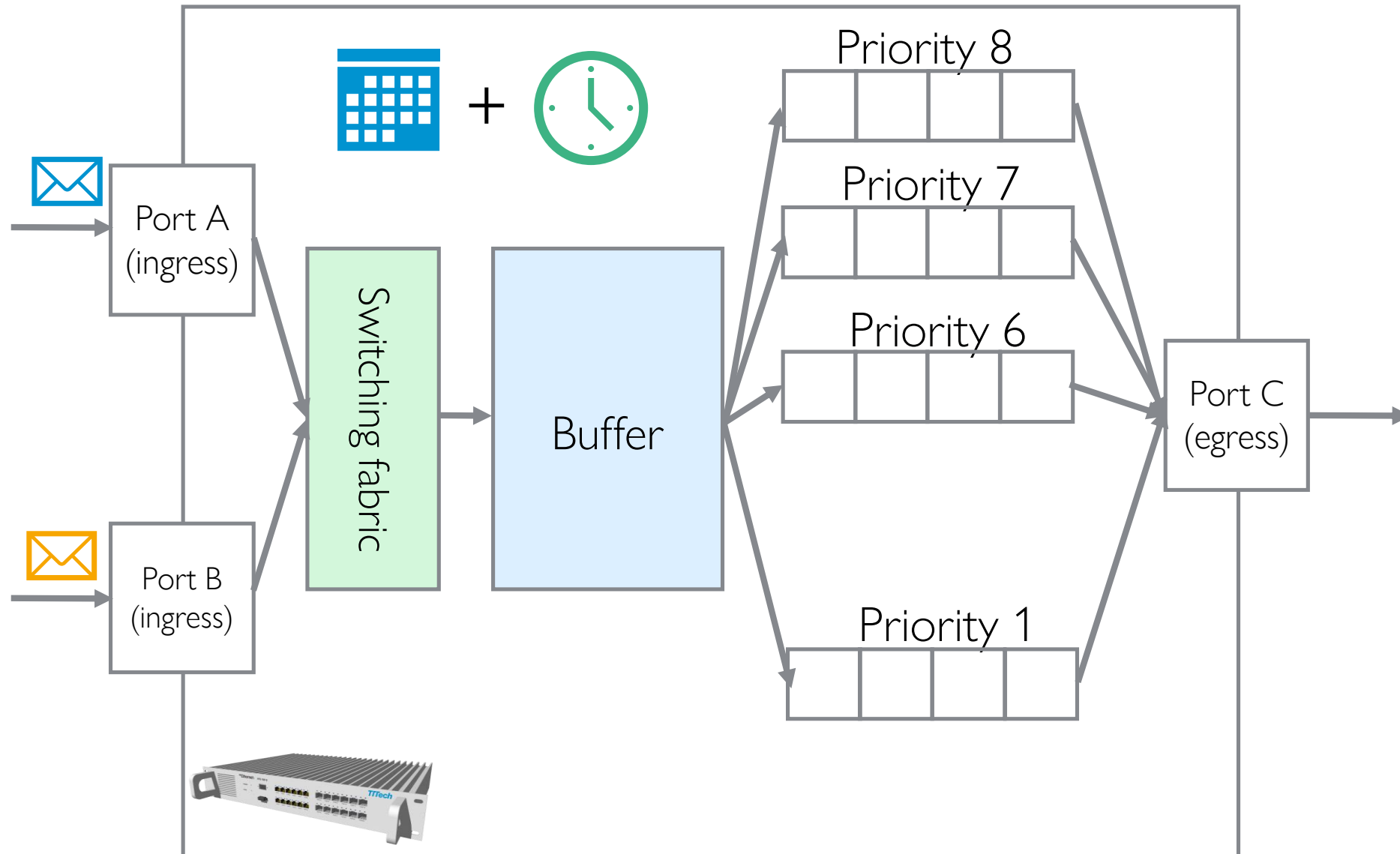


Devices must have a common understanding of time.

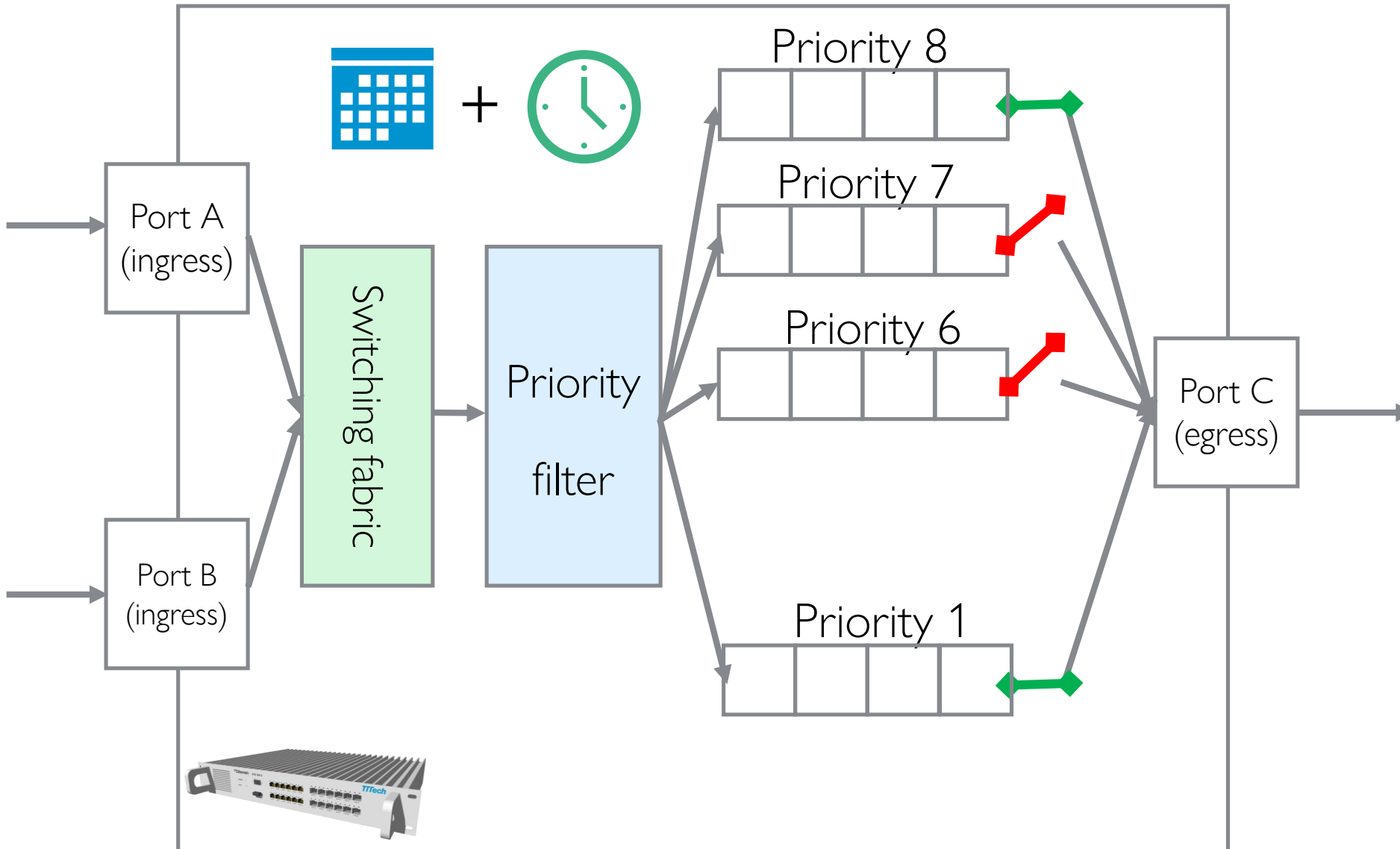


Time-triggered communication & timing checks in a network with forwarding:
global schedules & delays apply





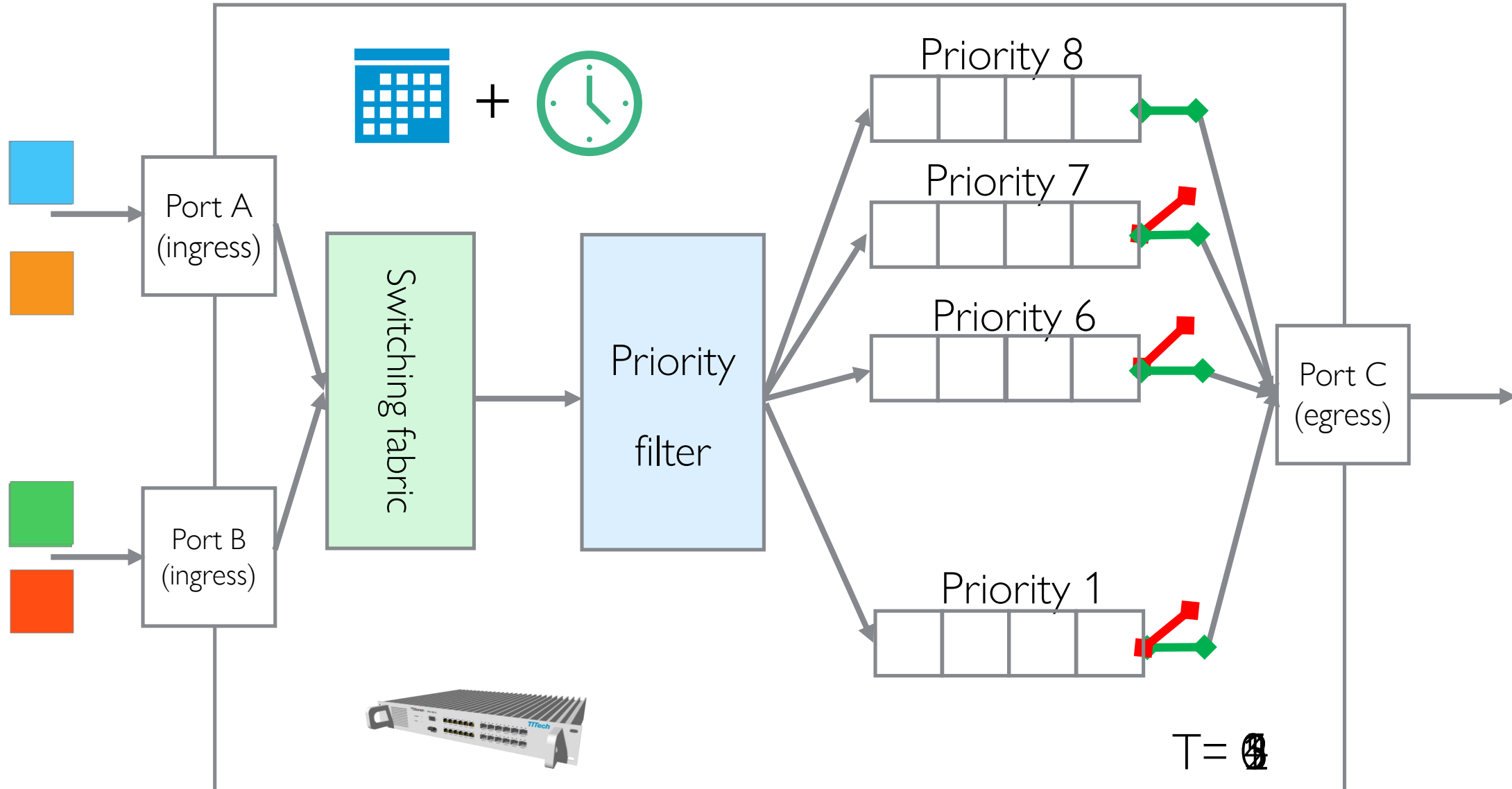
TSN (Qbv) switch



t=0 occcooc
t=1 oocooccc
t=5 ccccccco
t=7 ococcccc
.....

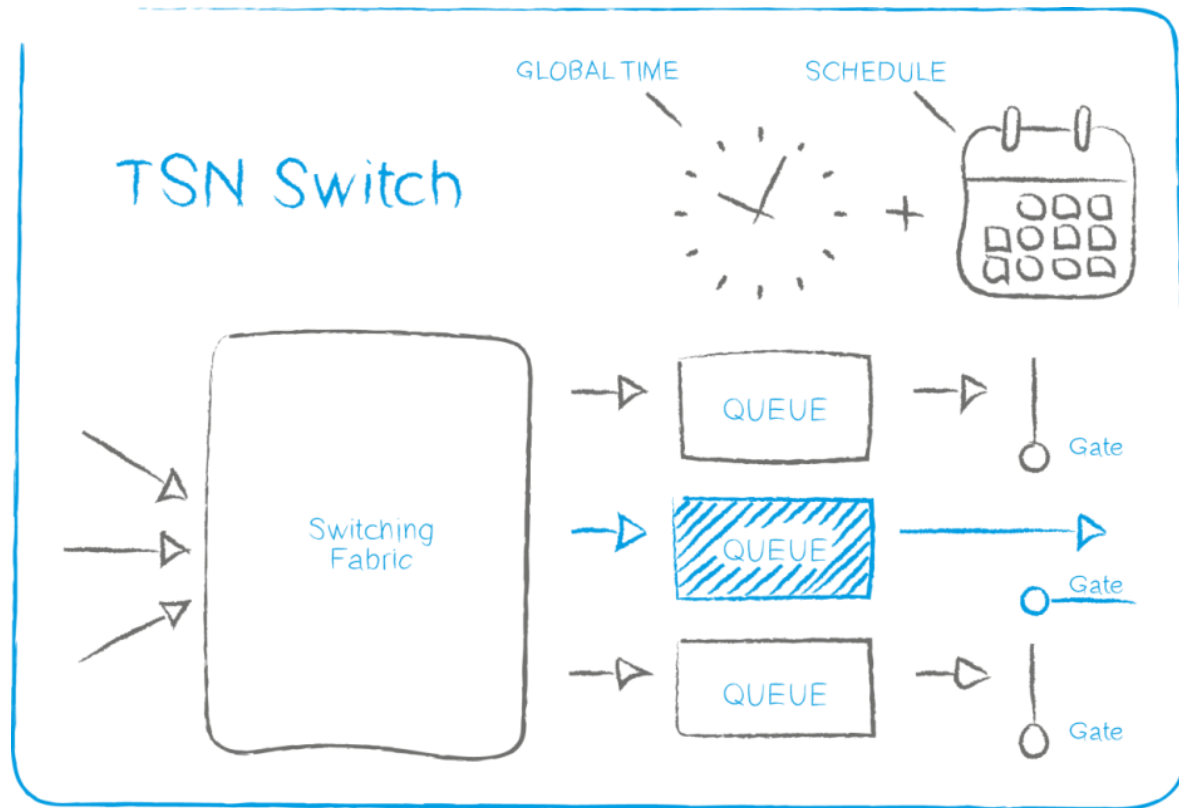
Gate Control List

TSN (Qbv) switch



What is a Time-Sensitive Network (TSN)

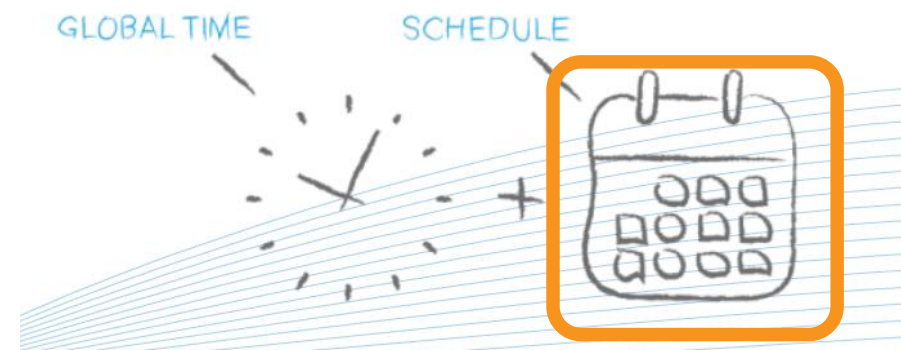
Time Sensitive Networking covers a set of Ethernet sub-standards currently defined in the IEEE 802.1 TSN task group



- At the heart of TSN are mechanisms that provide **time synchronization** for networked devices and **scheduled** forwarding of defined traffic flows through the network.
- Through time synchronization and scheduling, TSN delivers deterministic communication over standard Ethernet, thereby enabling the **convergence** of critical control traffic with data traffic over one infrastructure without the need for gateways or proprietary solutions.

IEEE TSN task group - collection of sub-standards that enhance 802 Ethernet with real-time capabilities

Standard	Description
802.1ASrev	Timing & Synchronization
802.1Qbv	Enhancements for Scheduled Traffic (Timed Gates for Egress Queues)
802.1Qbu	Frame Preemption
802.1Qca	Path Control and Reservation
802.1Qcc	Central Configuration Management
802.1Qci	Per-Stream Time-based Ingress Filtering and Policing
802.1CB	Redundancy, Frame Replication & Elimination



$$\langle G(E), G(Q) \rangle$$

Device capabilities

$$G(E)$$

$$V_e$$

Scheduled Es

$$V_s$$

Scheduled Sw

$$V_{e+s}$$

Scheduled Es+Sw

Queue configuration

$$G(Q) = \langle \mathcal{N}, \mathcal{N}_{tt}, \mathcal{N}_{prio} \rangle$$

$$\mathcal{N}_{tt} \geq 1$$

- Critical traffic assigned to the scheduled queues
- Non-critical traffic assigned to priority queues (post-analysis through network calculus [[Frances@ERTS06](#)])
- Isolation: non-critical streams may interfere with each other in priority queues, but not with critical streams (isolated in the scheduled queues)

$$\{V_{e+s}, \langle 1|1|0 \rangle\}$$

Only critical traffic (serialized similar to bus systems)

$$\{V_{e+s}, \langle n|1|n-1 \rangle\}$$

Legacy AVB systems that require a few additional high-criticality flows [Specht@ECRTS16]

$$\{V_{e+s}, \langle n|n|0 \rangle\}$$

Maximize solution space for critical traffic, non-critical traffic can be scheduled by inverting the cumulated schedule of scheduled queues

$$\{V_{e+s}, \langle n|m|n-m \rangle\}$$

High-criticality applications that feature both scheduled and non-scheduled traffic, trade-off between schedulability of critical traffic and timeliness properties and flexibility for non-scheduled traffic

$$\{V_{e+s}, \langle n|0|n \rangle\}$$

Standard AVB (IEEE 802.1BA) network in which flows are serviced according to the priority

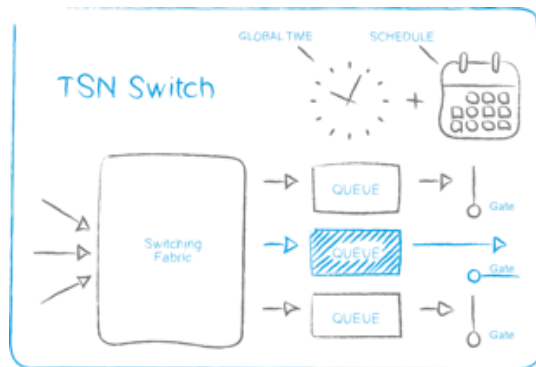
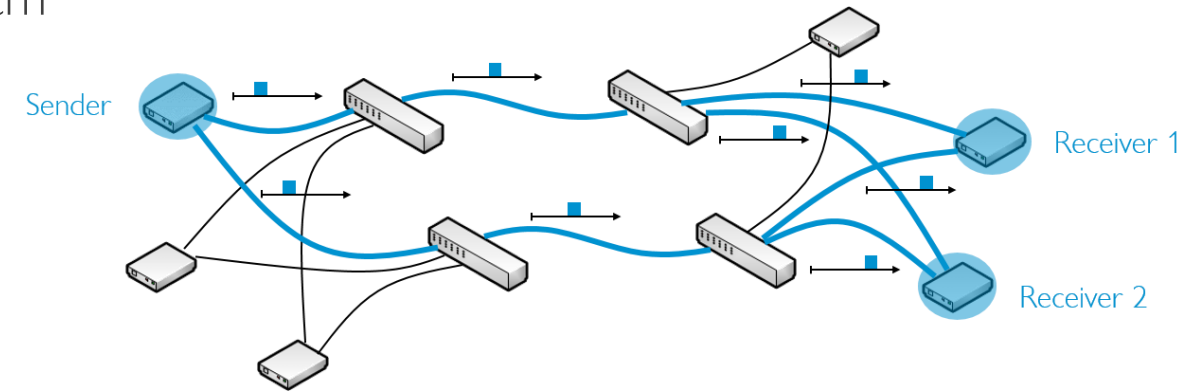
Sender

Receiver 1

Receiver 2

Network and traffic model

- multi-hop layer 2 switched network via full-duplex multi-speed links
- (multicast) TSN streams with multiple frames per stream
- communication is from one sender to one or multiple receivers
- synchronised time (<1 usec precision)
- wire and device delays

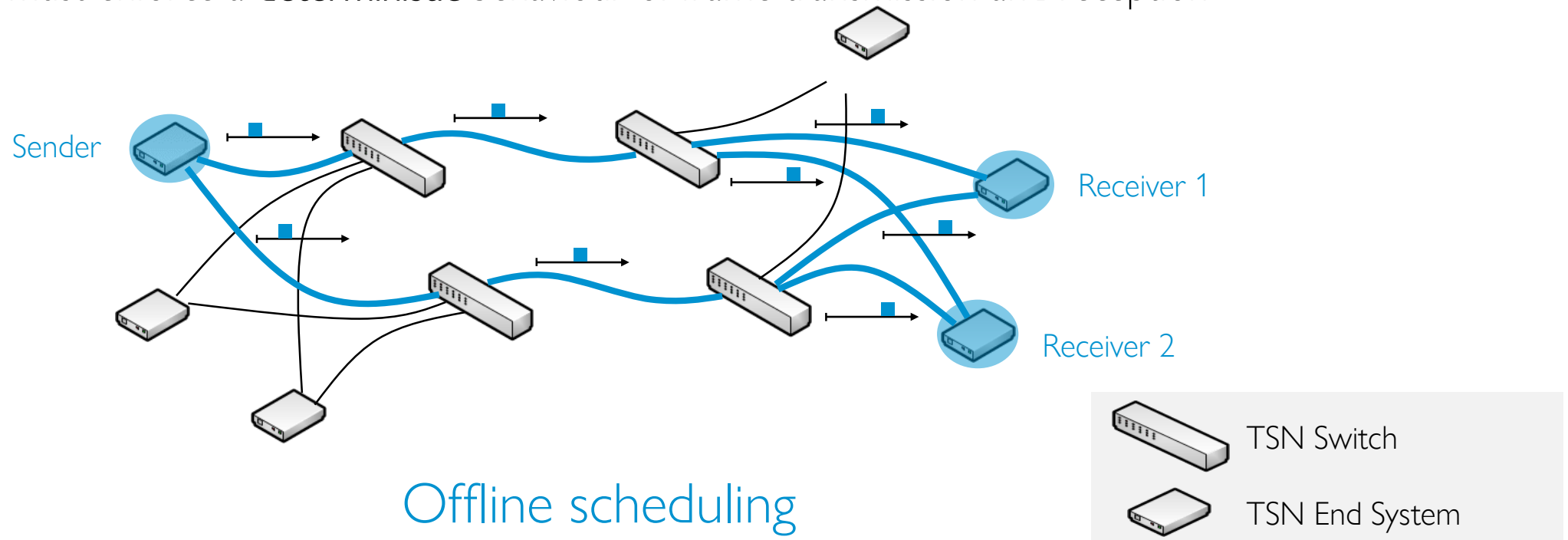


- Scheduled 802.1Qbv-compatible devices (Sw + Es)
- Scheduled (mutually exclusive) & priority queues
- Guaranteed delivery of critical traffic with known latency, small & bounded jitter

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 end-to-end latency, frame length, as well as constraints derived from resources and physical limitations.

The schedule must enforce a **deterministic** behaviour of frame transmission and reception





Scheduling for TSN therefore means:

- Find **transmit** (and implicitly receive) times and queue assignment for **all frames** on **all devices** such that
 - No collisions on ports occur
 - Frames are sent correctly in the time domain
 - End-to-end latencies are fulfilled
 - Other user-defined constraints are fulfilled
 - Etc.

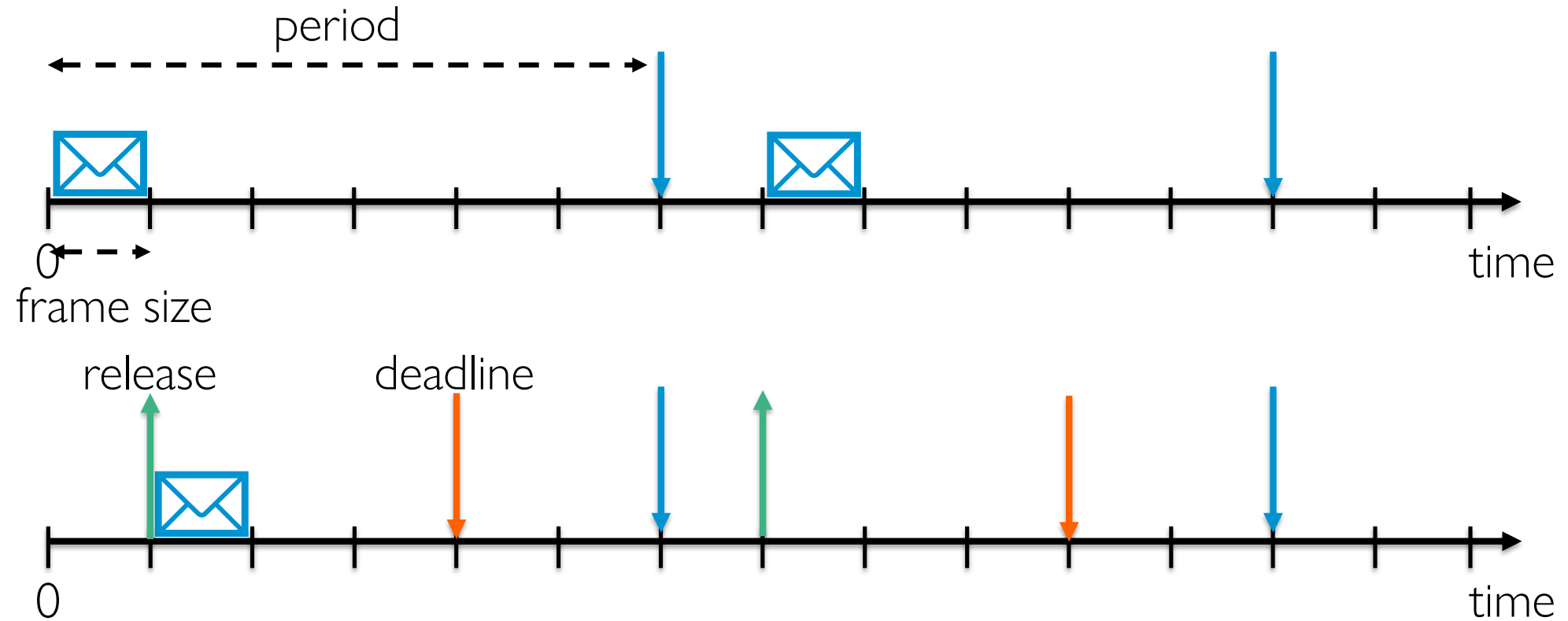
Schedule correctness conditions



- Technology
- Communication
- User-defined



Frame placement within period



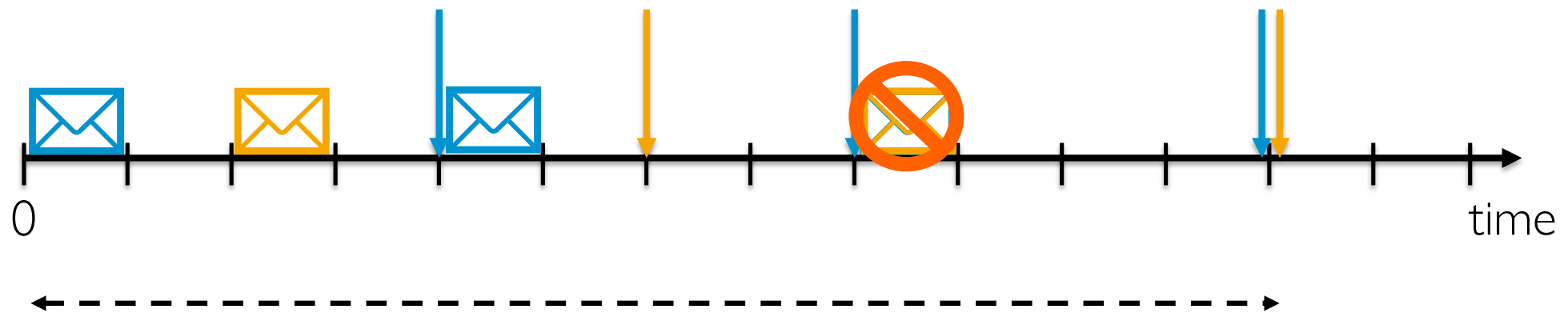
Correctness conditions



Frames are not allowed to overlap in the time domain

 period = 4

 period = 6



Hyperperiod = 12



Frames are not allowed to overlap in the time domain

 period = 4

 period = 6

- For all frames scheduled on the same link
- check overlapping in any period instance
 - until hyperperiod of the two considered periods

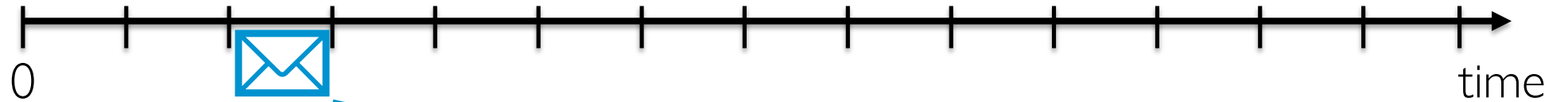


Hyperperiod = 12

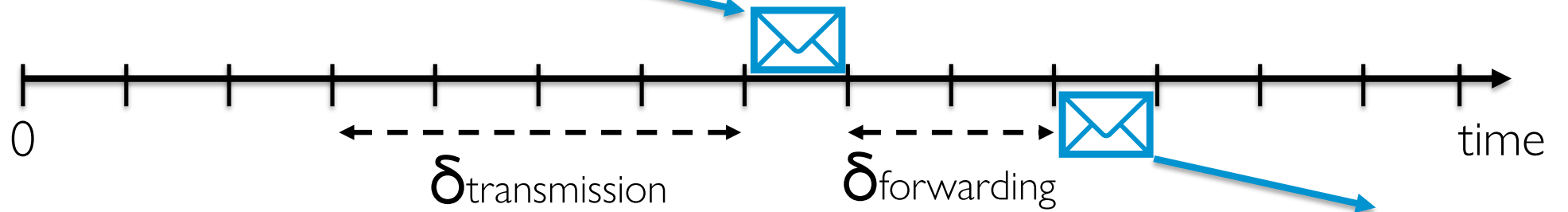


A frame must be received before it is forwarded

Device i



Device i+1

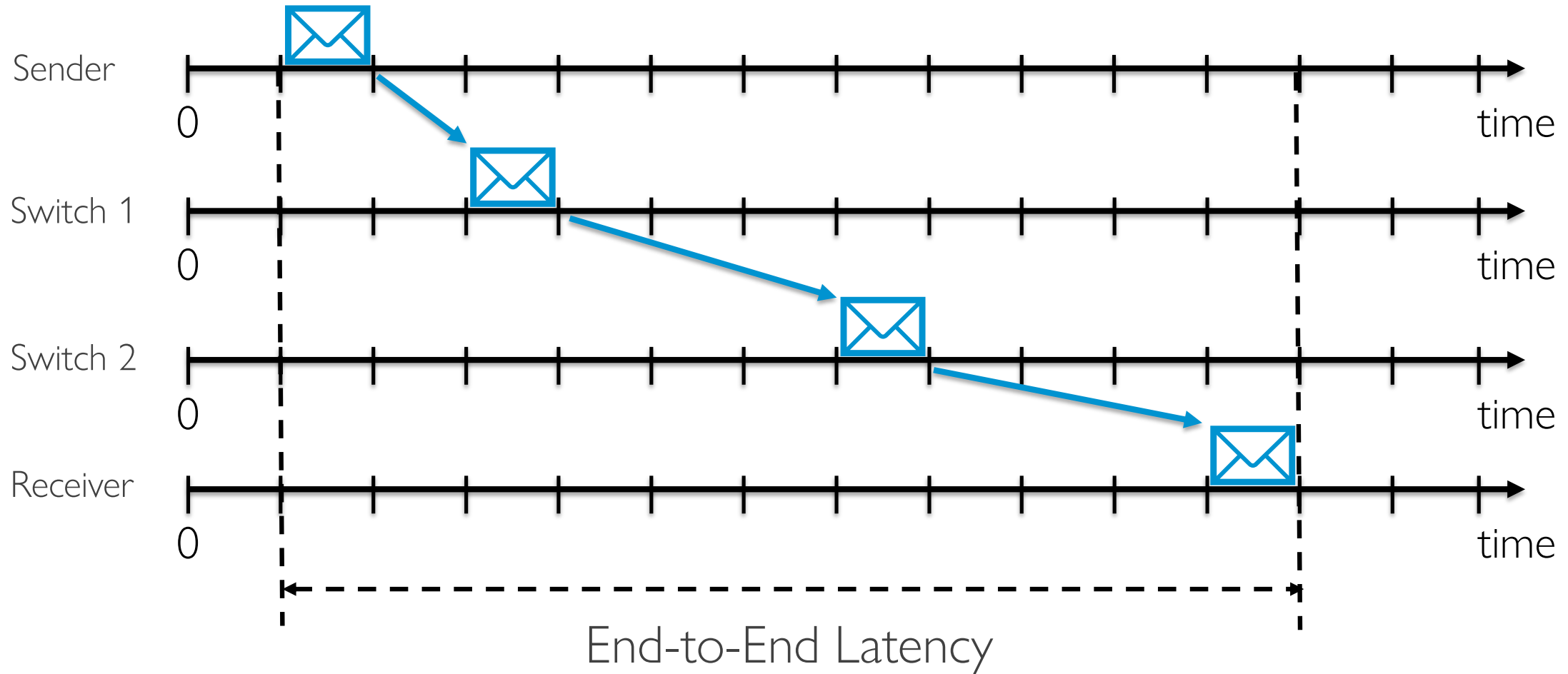


$$\delta_{\text{transmission}} = \text{Wire Delay} + \text{Port delays} + \text{Transmission delay} + \text{Precision}$$

$$\delta_{\text{forwarding}} = \text{Switch forwarding delay}$$



Maximum end-to-end latency constraint



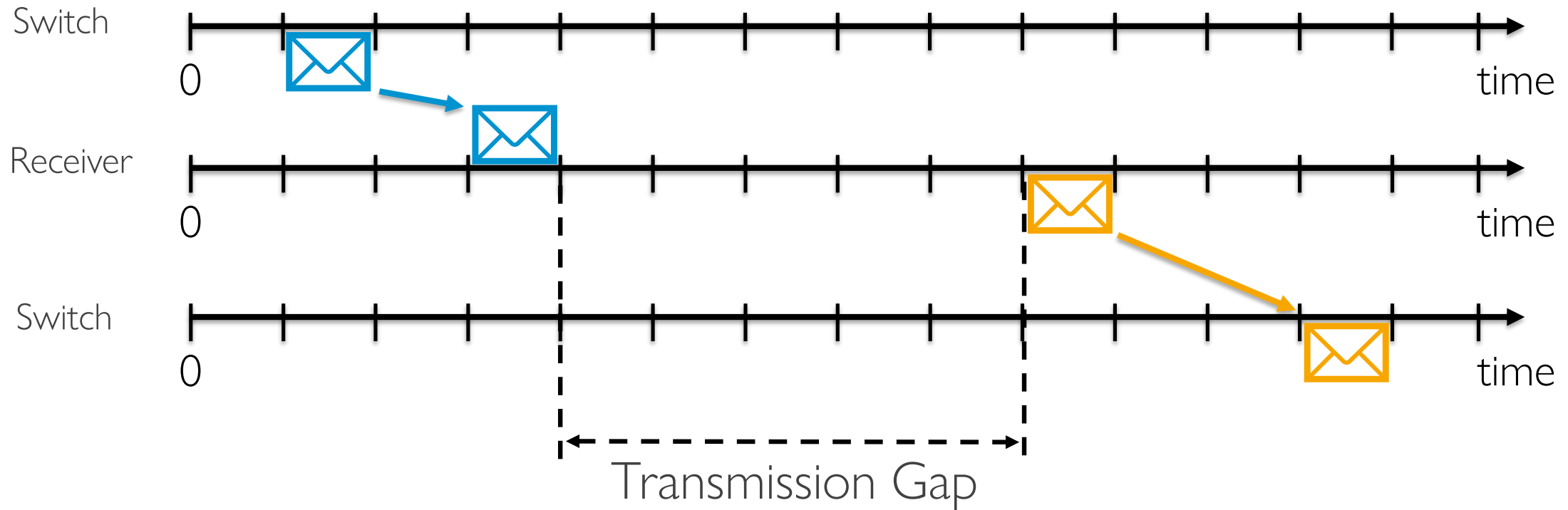
Scheduling constraints are additional conditions on the correctness of the schedule.

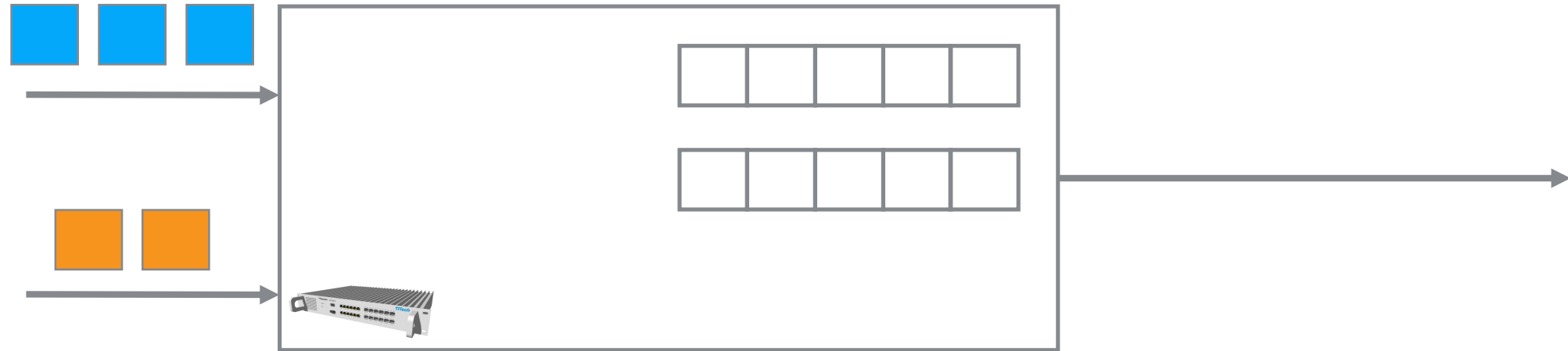
- Start time constraint
- End time constraint
- End-to-end latency constraint
- Transmission gap constraint
- Precedence constraint
- Floating window constraint
- Best-effort reservation constraint

Transmission Gap Constraint

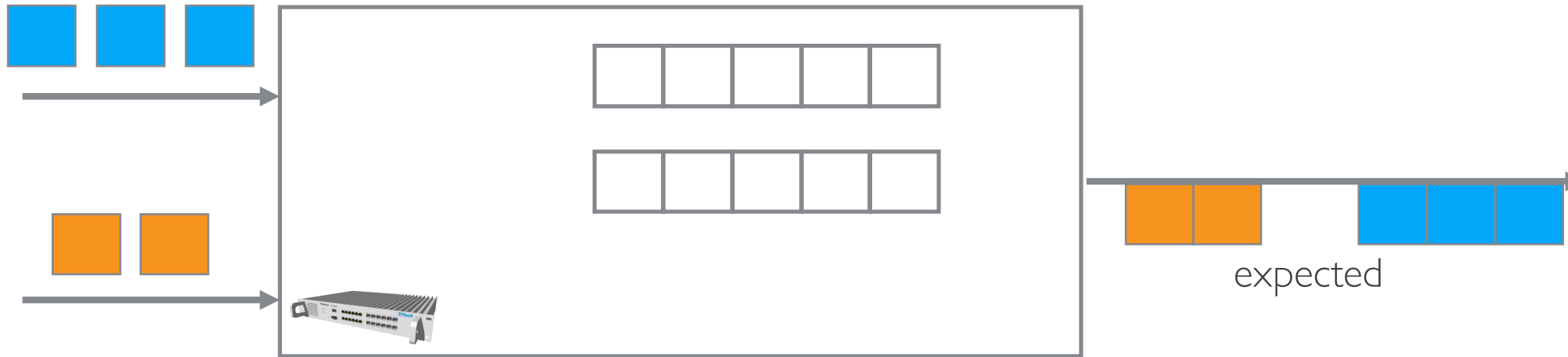


Enforce a gap between the reception of one flow and the transmission of another



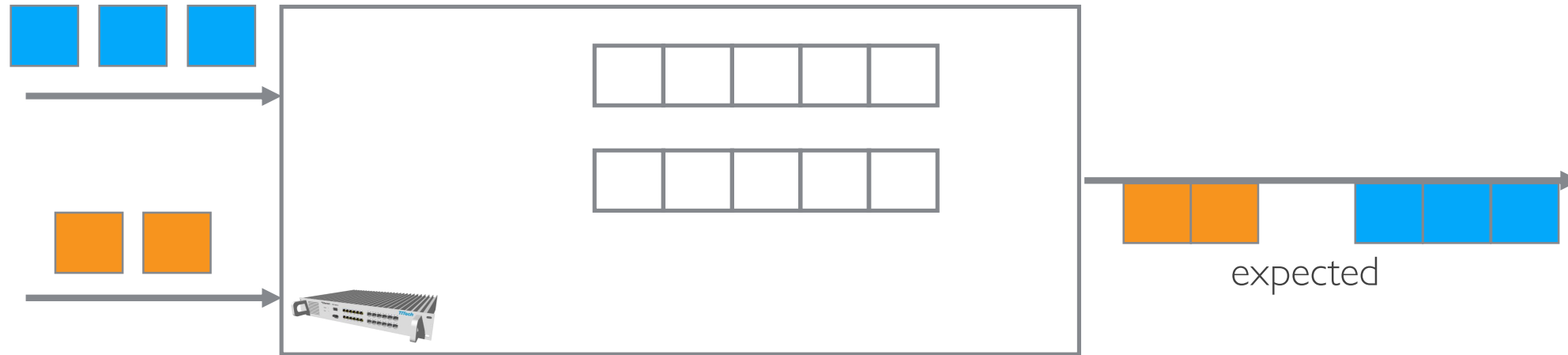


In order to maintain jitter and latency requirements we expect at each device a certain timely order of frames

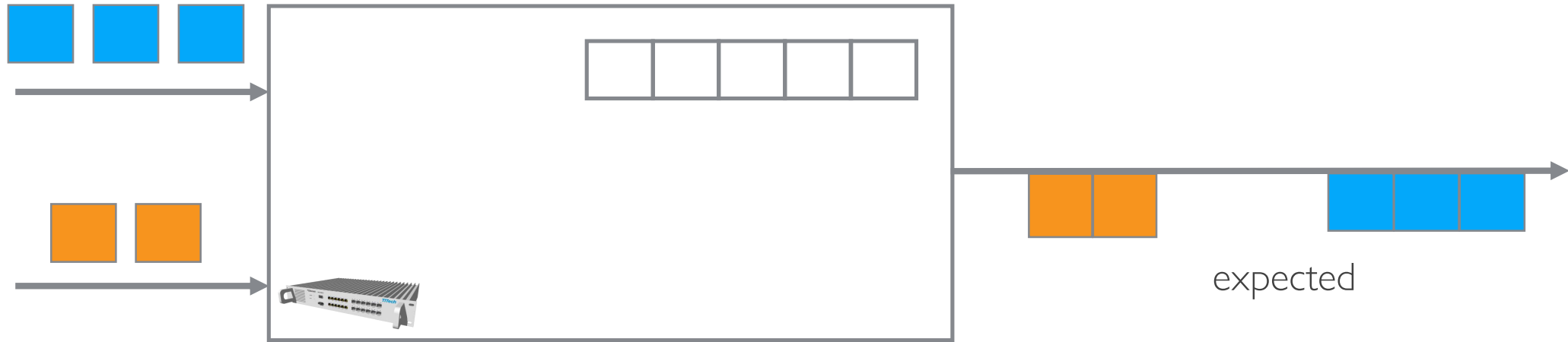


- synchronization errors, frame loss, time-based ingress policing (e.g. IEEE 802.1Qci) may lead to non-deterministic placement in queues during runtime
- timed gates control events on the egress port, not the order of frames in the queue
- placing of frames in the scheduled queues at runtime may be non-deterministic

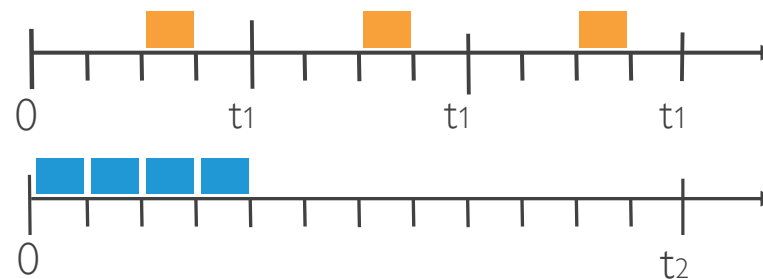
Timely behaviour of streams may oscillate, accumulating jitter for the overall end-to-end transmission

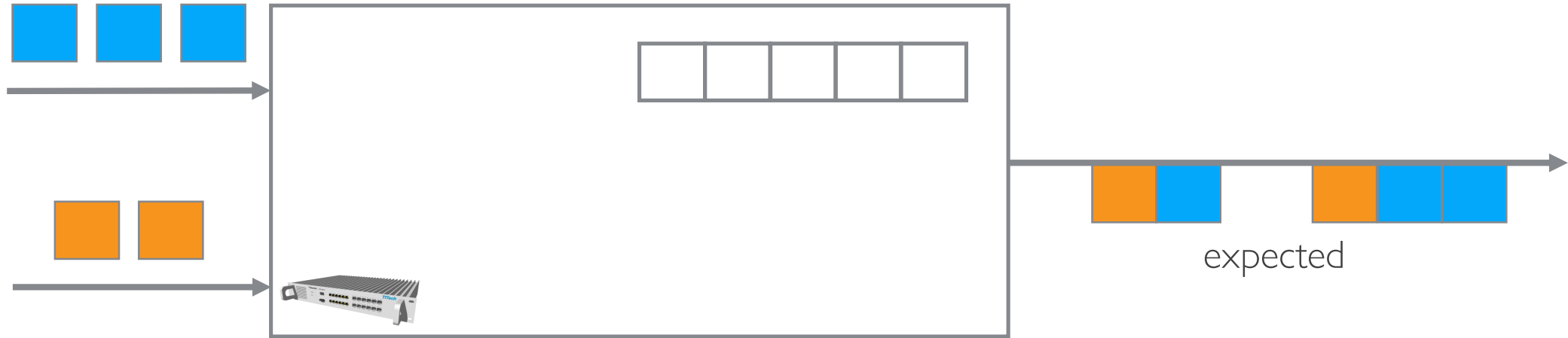


Solves the non-determinism problem but reduces the solution space



- Once a flow has arrived, no other flow can arrive in the same queue until the first flow has been completely sent
- Better than queue isolation but still restrictive





- Ensure that there are only frames of one flow in the queue at a time
- Frames from another flow may only enter the queue if the already queued frames of the initial flow have been serviced
- Less performant than stream isolation since the solver has to consider at all frame interleavings

The constraint for minimum jitter scheduling of critical traffic for 802.1Qbv networks is:

isolate frames/streams in the **time domain**
OR
isolate streams in **different queues**



find **routes, offsets** and **queue assignment** for individual frames of TSN streams from sender to receiver(s) that conforms to the constraints

- frame constraints
- link constraints
- stream constraints
- end-to-end latency constraints
- **stream or frame isolation**



802.1Qbv

NP-complete

$$\forall s_i \in \mathcal{S}, \forall [v_a, v_b] \in s_i, \forall f_{i,j}^{[v_a, v_b]} \in \mathcal{F}_i^{[v_a, v_b]} :$$

$$\left(f_{i,j}^{[v_a, v_b]} . \phi \geq 0 \right) \wedge \left(f_{i,j}^{[v_a, v_b]} . \phi \leq f_{i,j}^{[v_a, v_b]} . T - f_{i,j}^{[v_a, v_b]} . L \right).$$

$$\forall [v_a, v_b] \in \mathcal{L}, \forall s_i^{[v_a, v_b]}, s_j^{[v_a, v_b]} \in \mathcal{S},$$

$$\left(\Phi(s_i^{[v_a, v_b]}, s_j^{[v_a, v_b]}) \right) \vee \left(s_i^{[v_a, v_b]} . p \neq s_j^{[v_a, v_b]} . p \right)$$

$$\forall s_i \in \mathcal{S}, \forall [v_a, v_x], [v_x, v_b] \in s_i,$$

$$\forall f_{i,j}^{[v_a, v_x]} \in \mathcal{F}_i^{[v_a, v_x]}, \forall f_{i,j}^{[v_x, v_b]} \in \mathcal{F}_i^{[v_x, v_b]} :$$

$$f_{i,j}^{[v_x, v_b]} . \phi \times [v_x, v_b] . mt - [v_a, v_x] . d - \delta \geq$$

$$(f_{i,j}^{[v_a, v_x]} . \phi + f_{i,j}^{[v_a, v_x]} . L) \times [v_a, v_x] . mt.$$

$$\forall [v_a, v_b] \in \mathcal{L}, \forall s_i^{[v_a, v_b]}, s_j^{[v_a, v_b]} \in \mathcal{S}, i \neq j,$$

$$\forall \alpha \in [0, hp_i^j / s_i . T - 1], \forall \beta \in [0, hp_i^j / s_j . T - 1] :$$

$$\left(last(\mathcal{F}_i^{[v_a, v_b]}) . \phi \times [v_a, v_b] . mt + \alpha \times s_i . T + \delta \leq \right.$$

$$f_{j,1}^{[v_y, v_a]} . \phi \times [v_y, v_a] . mt + \beta \times s_j . T + [v_y, v_a] . d \left. \vee \right.$$

$$\left(last(\mathcal{F}_j^{[v_a, v_b]}) . \phi \times [v_a, v_b] . mt + \beta \times s_j . T + \delta \leq \right.$$

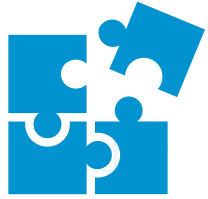
$$f_{i,1}^{[v_x, v_a]} . \phi \times [v_x, v_a] . mt + \alpha \times s_i . T + [v_x, v_a] . d \left. \right).$$

Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelik, and Wilfried Steiner

Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks

In Proc. 24th International Conference on Real-Time Networks and Systems (RTNS), 2016.

http://www.cs.uni-salzburg.at/~scraciunas/pdf/conferences/craciunas_rtms16.pdf



We have seen what constitutes a correct schedule.

The question is now, how do we generate such a correct schedule?

How complex is the schedule generation? Can we do it efficiently?



Satisfiability Modulo Theories

satisfiability of logical formulas in first-order formulation

background theories $\mathcal{LA}(\mathbb{Z})$ BV

variables x_1, x_2, \dots, x_n

logical symbols $\forall, \wedge, \neg, (,)$

non-logical symbols $+, =, \%, \leq$

quantifiers \exists, \forall

optimization criteria: Optimization Modulo Theories [[Björner@TACAS15](#)]

- Formalization of technology, communication and user constraints
- Optimal scheduler
- Feasible/Optimal schedule
- Easily extendable for new constraints
- A lot of solvers and a very active community

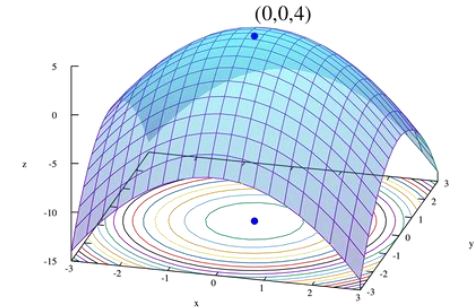
Yices [[Dutertre@CAV14](#)]

Z3 [[de Moura@TACAS08](#)]

CVC4 [[Barrett@CAV11](#)]

OpenSMT [[Bruttomesso@TACAS10](#)]

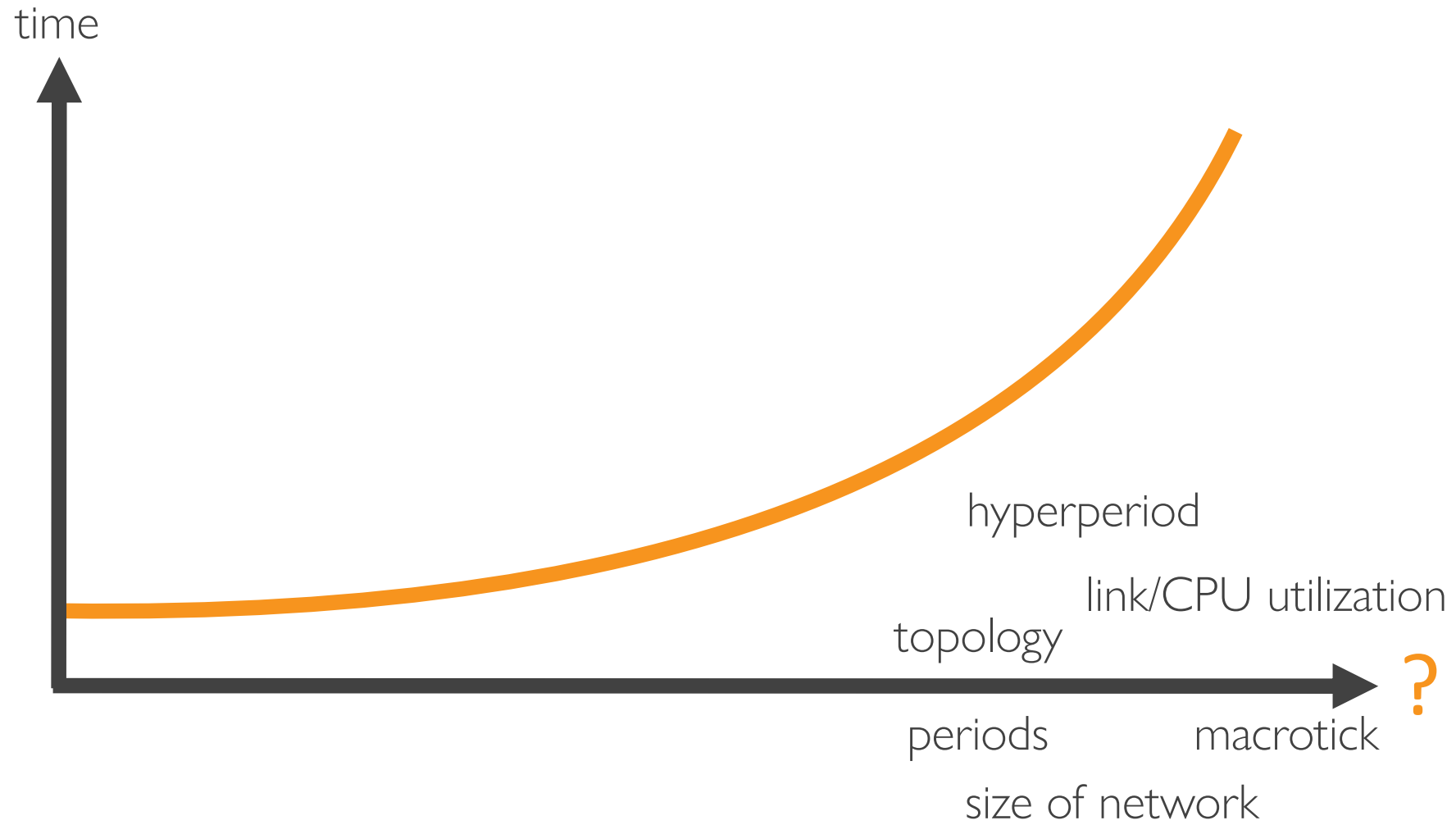
Optimize schedule with respect to certain properties of the system (e.g. minimize end-to-end latency of selected streams)



802.1Qbv-specific optimizations:

- QoS properties: minimize required scheduled queues in order to increase QoS properties of non-critical traffic
- Design space exploration in case of infeasible use-cases, i.e. find the minimal number of queues required for scheduled traffic such that a schedule is found

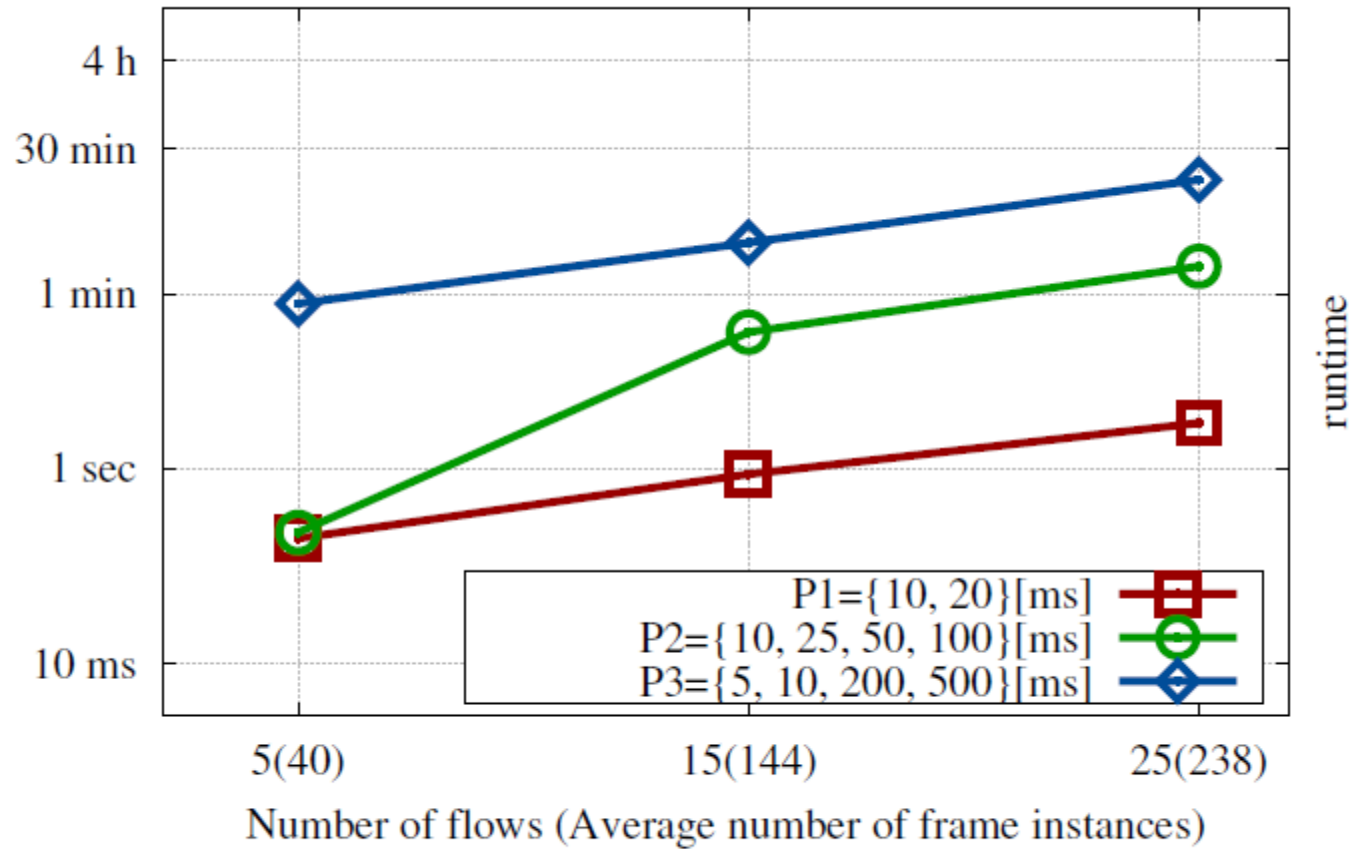
Many more optimization opportunities in combination with other TSN sub-standards (e.g. frame preemption)



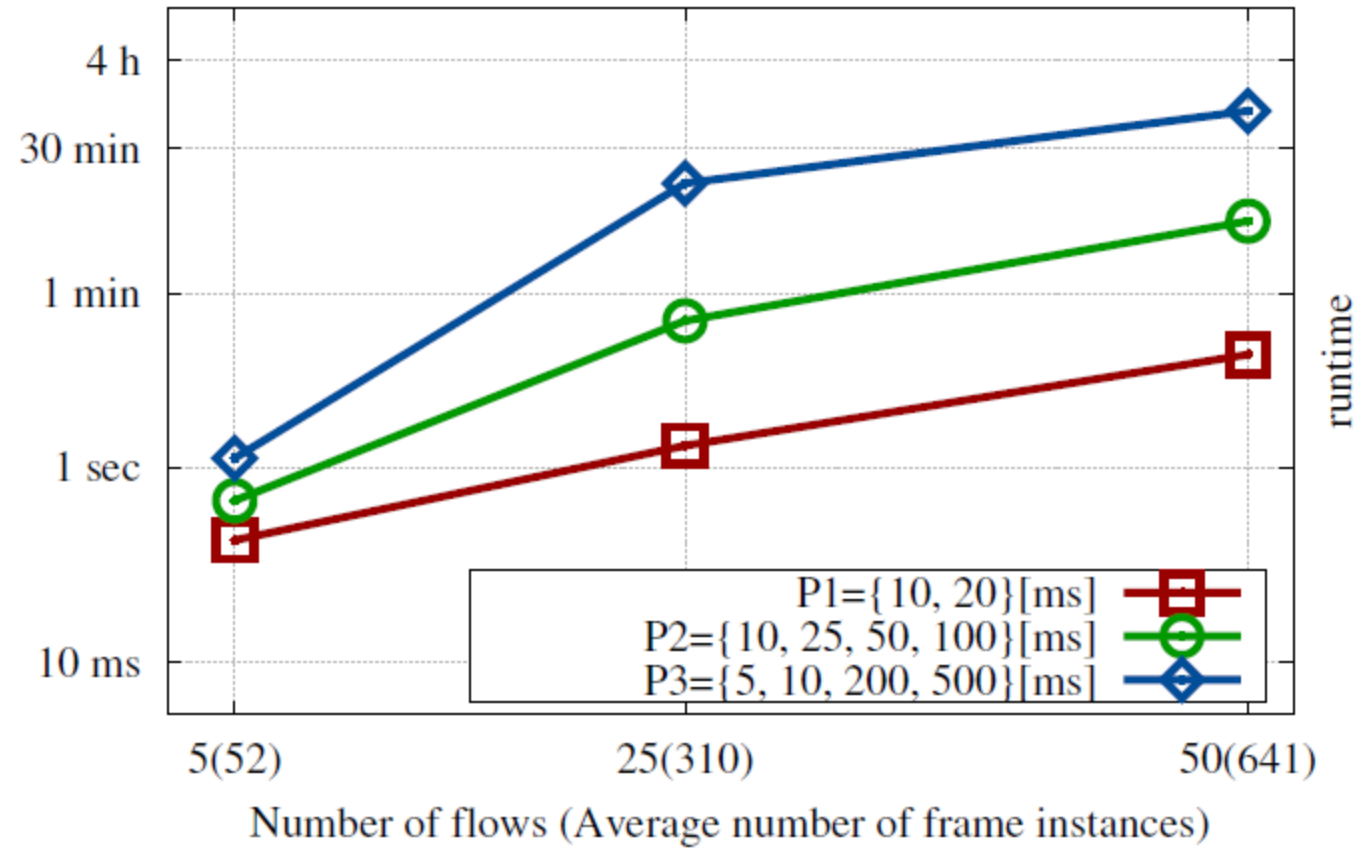
- Z3 v4.4.1 solver (64bit) (Yices v2.4.2 with quantifier-free linear integer arithmetic)
- 64bit 4-core 3.40GHz Intel Core-i7 PC with 4GB memory
- 3 predefined topologies ranging from 3 end-systems connected to one switch to 7 end-systems connected through 5 switches via 1Gbit/s links with a 1usec macrotick granularity (generate high utilization on the links)
- Time-out value for a run to 5 hours
- System configuration: $\{V_{e+s}, \langle 8, 8, 0 \rangle\}$

Scalability and schedulability experiments

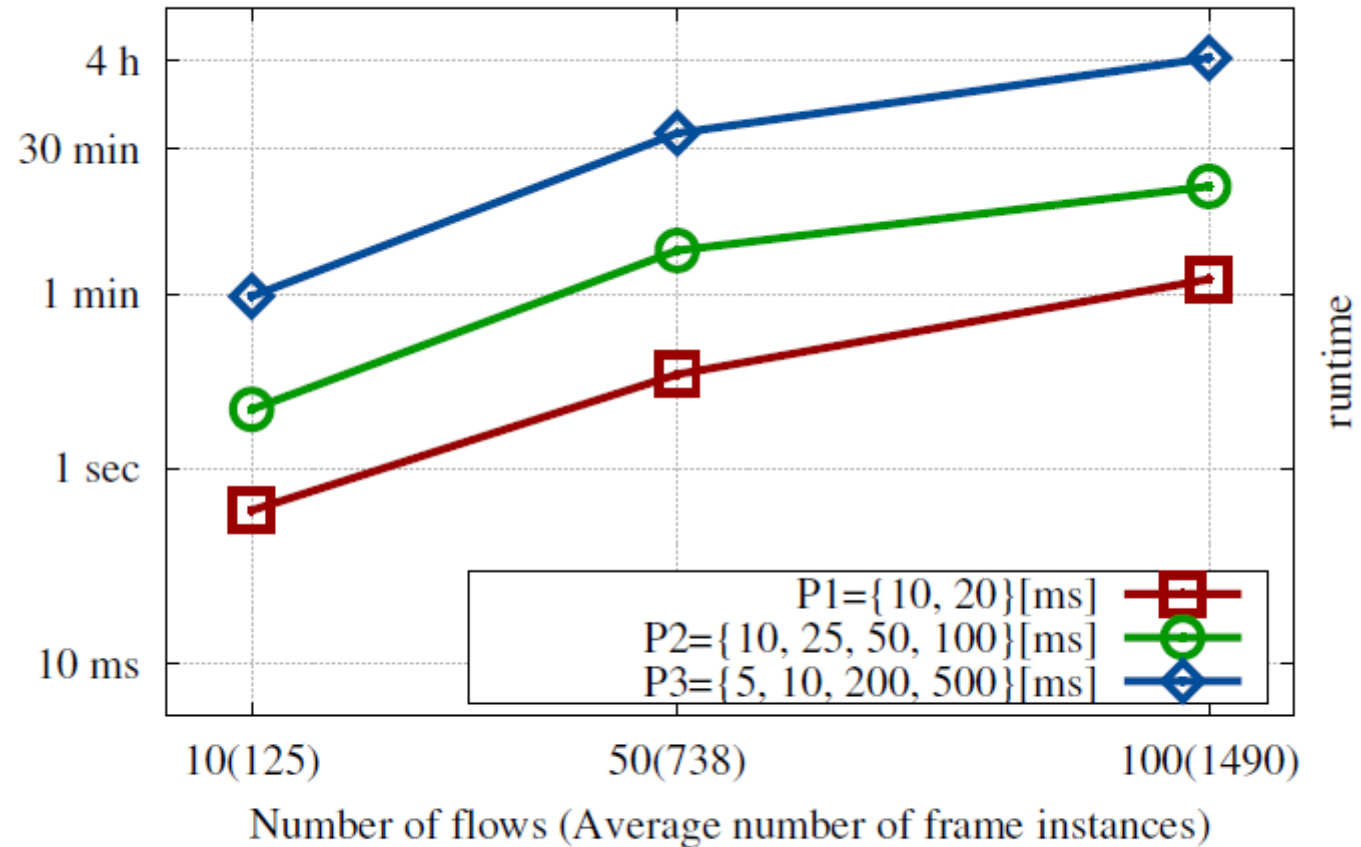
- Frame isolation method (using an incremental backtracking algorithm with step size of 1)
- Vary the problem set in 3 dimensions:
 1. topology size,
 2. number of flows,
 3. flow periods (chosen randomly from 3 sets of predefined periods)
- Data size uniformly between 2 and 8 MTU-sized frames
- Senders and receivers are chosen randomly



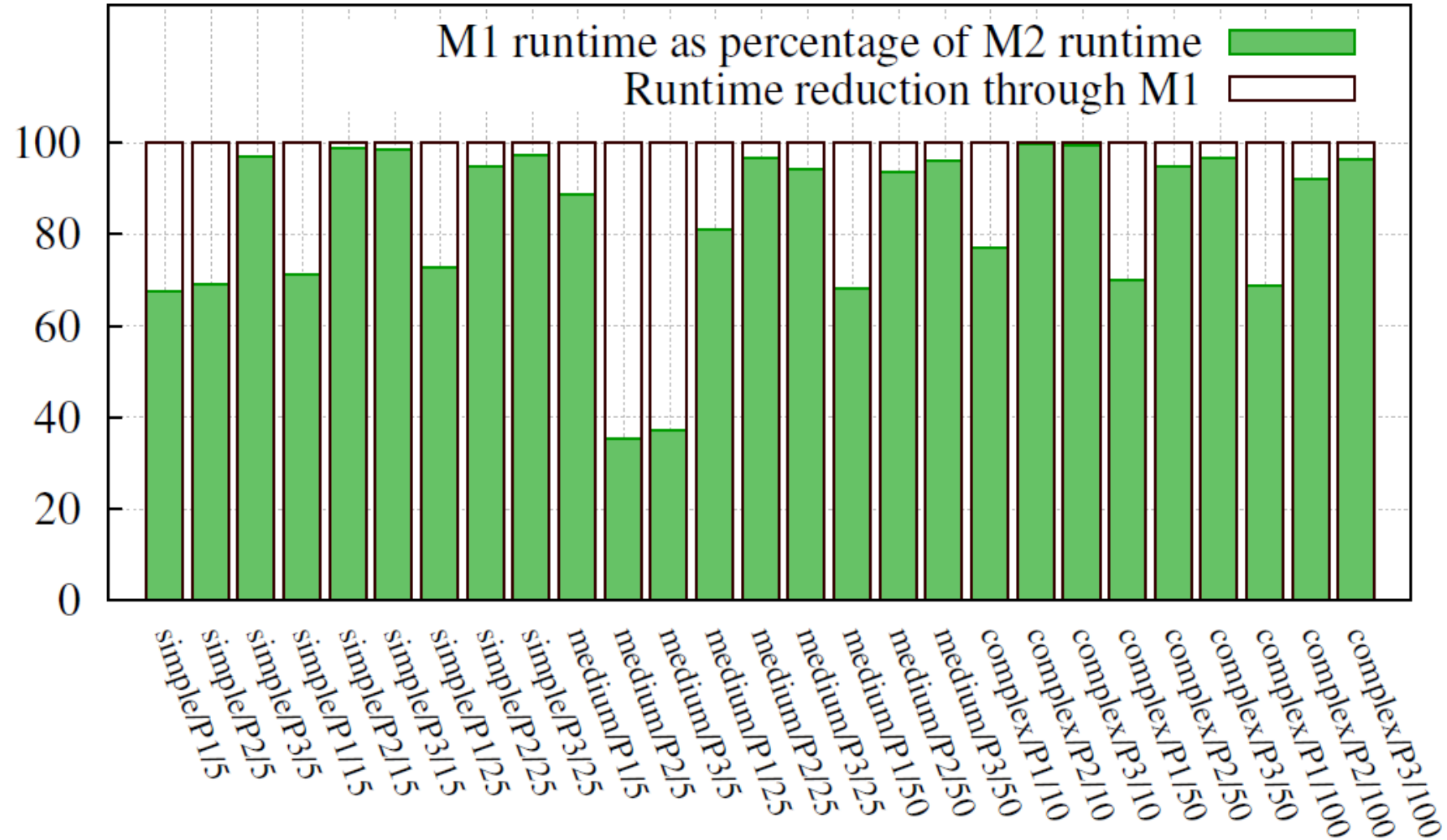
(a) Simple topology (3 ES, 1 SW)



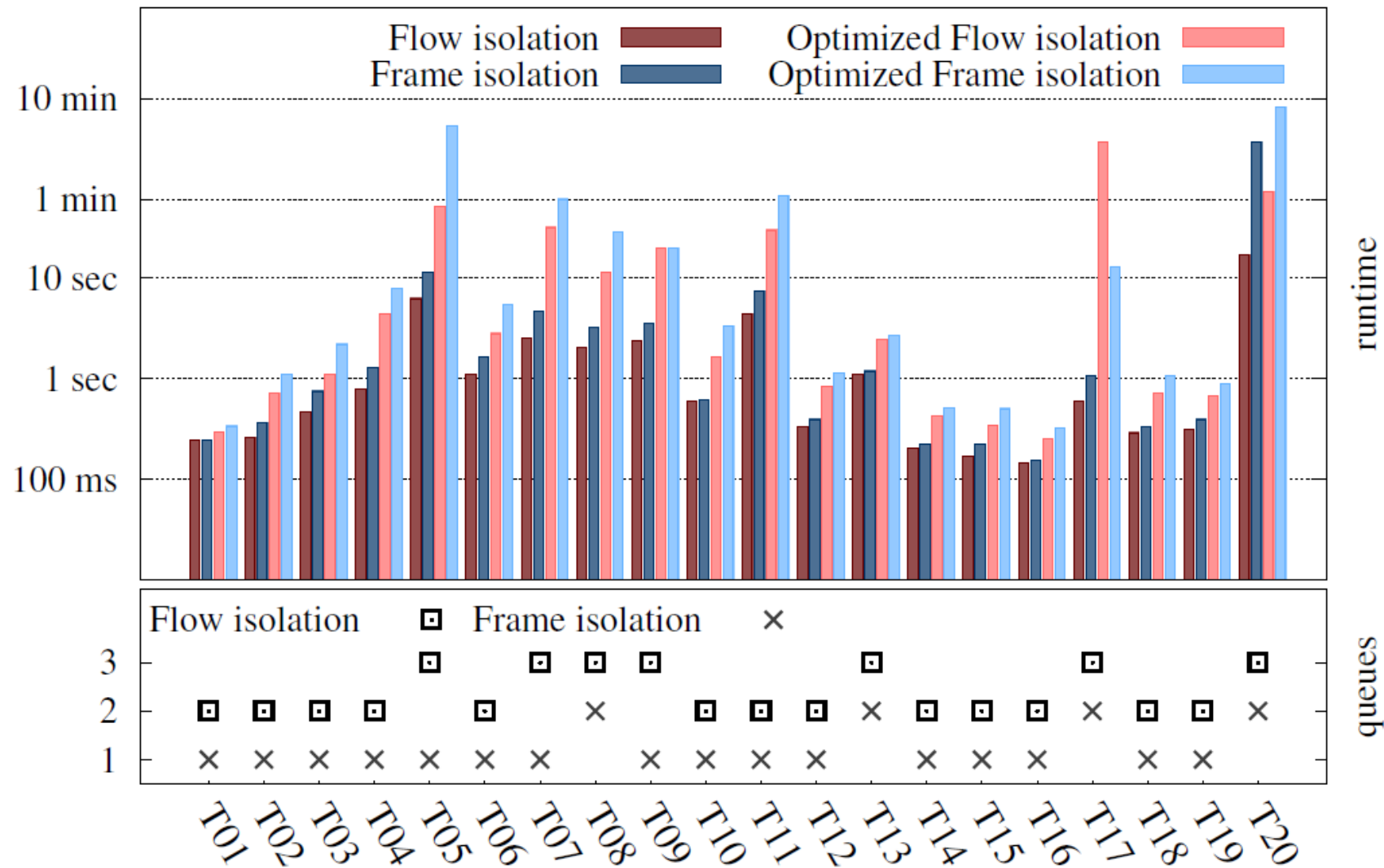
(b) Medium topology (5 ES, 2 SW)



(c) Complex topology (7 ES, 5 SW)



- Generated inputs that force streams to interleave if scheduled in the same egress queue
- Runs w/ and w/o optimization objectives using both stream and frame isolation methods
- Minimize accrued sum of the number of queues used per egress port
- No incremental steps for optimization runs





Vienna, Austria
(Headquarters)

Phone +43 1 585 34 34-0
office@tttech.com

USA

Phone +1 978 933 7979
usa@tttech.com

Japan

Phone +81 52 485 5898
office@tttech.jp

China

Phone +86 21 501529250
china@tttech.com

www.tttech.com