2018 IEEE Real-Time and Embedded Technology
and Applications Symposium (RTAS), Porto, Portugal

IEEE 802.1Qbv Gate Control List Synthesis using Array Theory Encoding

Ramon Serna Oliver, Silviu S. Craciunas, Wilfried Steiner

{rse, scr, wst} @tttech.com

4/11/2018

Outline

01 Time Sensitive Networks
IEEE 802.1Qbv

02 Deterministic Ethernet
Network Model and Assumptions

03 Problem Formulation
Synthesis of Gate Control Lists

04 Formalization of Constraints
Examples

05 Evaluation Results

06 Conclusions
Future Directions

01

Time-Sensitive Networks

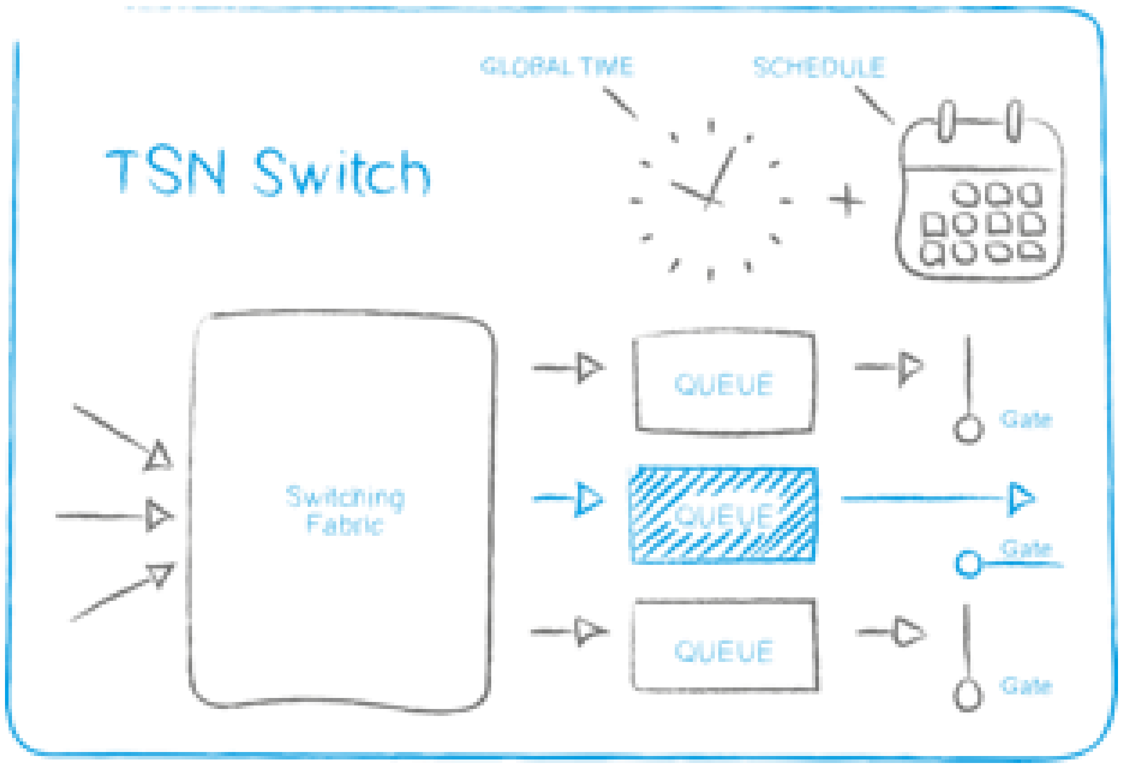
IEEE 802.1Qbv

IEEE TSN Standards

The Time-Sensitive Networking Task Group is part of the IEEE 802.1 Working Group.

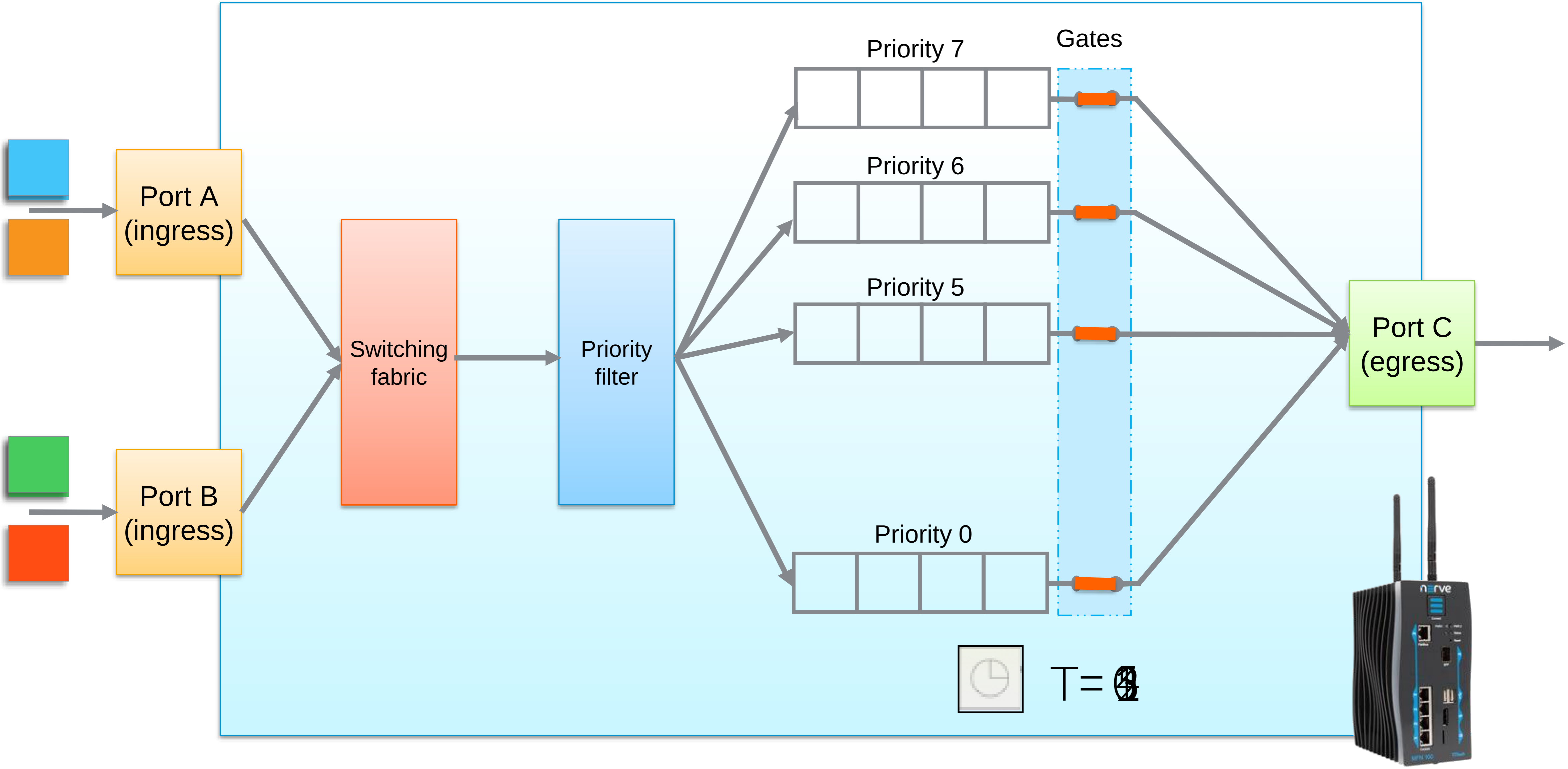
“…to provide deterministic services through IEEE 802 networks…” (*)

Standard	Description
802.1ASrev	Timing & Synchronization
802.1Qbv	Enhancements for Scheduled Traffic
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



(*) http://http://www.ieee802.org/1/pages/tsn.html

IEEE 802.1Qbv Functional Model



Gate Control List vs Time-Windows

- A GCL is nothing else than a cyclic timed list of gate status updates
- We can draw a symmetry with the GCL and a timeline of windows for each queue, where
  - A gate open event corresponds to a window start event for the queue
  - A gate close event corresponds to a window end event for the queue
  - A window spans along the timeline as long as the gate is open for that queue

Time	G_7	G_6	G_5	G_4	G_3	G_2	G_1	G_0
t_0	O	C	C	C	C	C	C	C
t_1	C	C	C	O	C	C	C	C
t_2	C	O	C	C	C	C	C	C
t_3	C	C	C	C	O	C	C	C
t_4	C	C	C	C	C	O	C	C
t_5	C	C	C	C	C	O	O	C
t_6	C	C	C	C	C	C	C	C

time

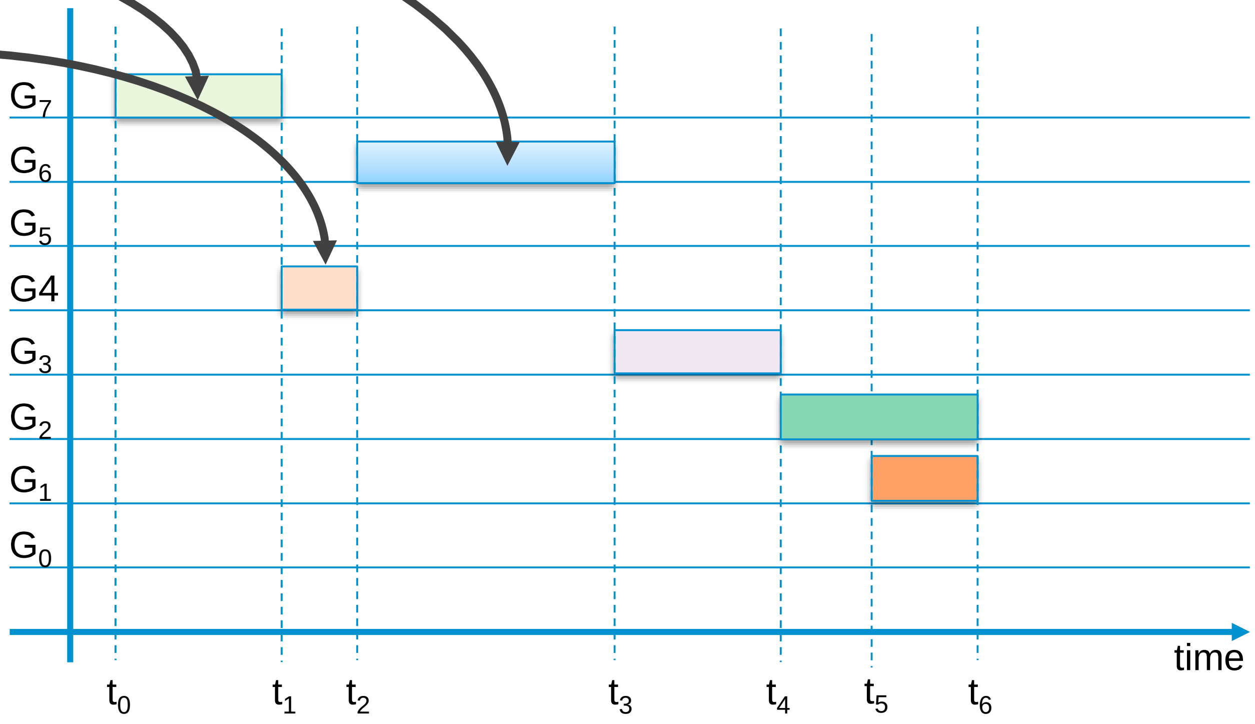
G_7
G_6
G_5
G4
G_3
G_2
G_1
G_0

Gate Control List vs Time-Windows

- A GCL is nothing else than a cyclic timed list of gate status updates
- We can draw a symmetry with the GCL and a timeline of windows for each queue, where
  - A gate open event corresponds to a window start event for the queue
  - A gate close event corresponds to a window end event for the queue
  - A window spans along the timeline as long as the gate is open for that queue

Time	G_7	G_6	G_5	G_4	G_3	G_2	G_1	G_0
t_0	O	C	C	C	C	C	C	C
t_1	C	C	C	O	C	C	C	C
t_2	C	O	C	C	C	C	C	C
t_3	C	C	C	C	O	C	C	C
t_4	C	C	C	C	C	O	C	C
t_5	C	C	C	C	C	O	O	C
t_6	C	C	C	C	C	C	C	C

time



02

Deterministic Ethernet

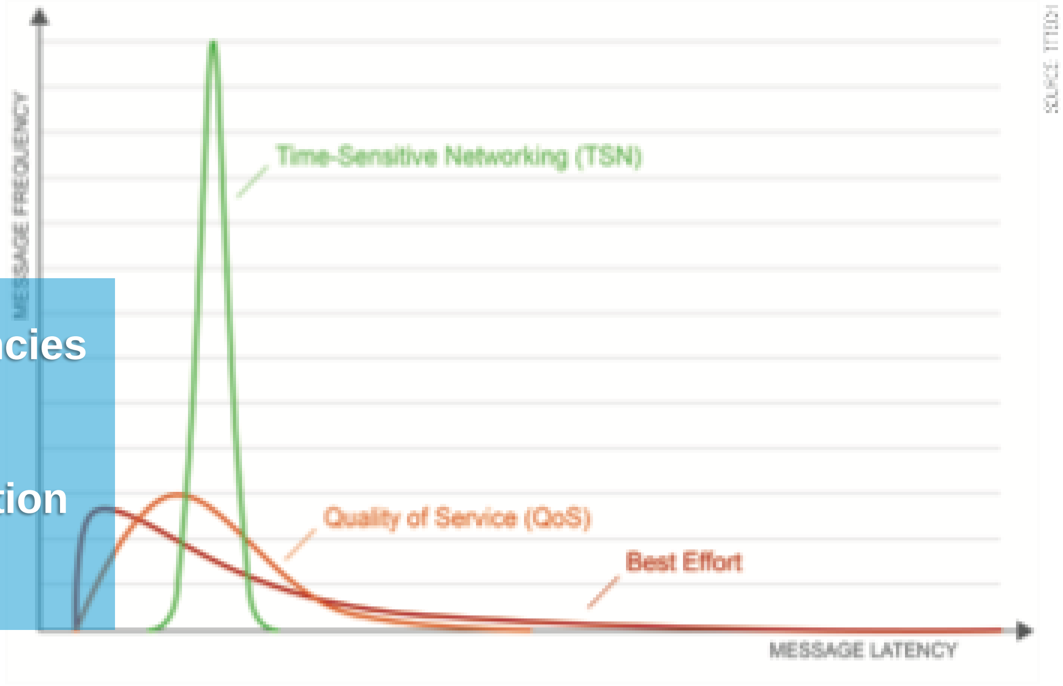
Network Model and Assumptions

Deterministic Ethernet

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

Devices (switches, end systems, etc.) have a common understanding of time.

- Calculable and guaranteed latencies
- Low and bounded jitter
- Control over bandwidth congestion
- 0 packet loss due to contention



SOURCE: TTTECH

Network and Traffic Model

TTTech

Topology and Network Model

- ✓ Multi-hop switched layer 2 full-duplex Ethernet network
- ✓ Time synchronization with bounded precision
  - ✓ E.g. 802.1AS-rev
- ✓ Known technology latencies
  - ✓ Wire delay, PHY, etc…

Communication Model

- ✓ Exchange of messages via periodic flows (~ stream)
  - ✓ Single sender to single receiver
    - ➢ Trivial extension: multiple receivers
  - ➢ Multiple Ethernet frames per message
- ➢ Routes are pre-computed and known

Device Capabilities

- ✓ Switches and end systems support 802.1Qbv
- ✓ Queues for scheduled traffic are not shared
- ✓ Switch capable to identify and assign frames to queues
  - ✓ E.g. based on PCP code in 802.1Q, or 802.1Qci

Message Constraints

- ✓ End-to-End latency
- ✓ Optional Maximum Jitter Requirements
- ✓ Optional Jitter Minimization

03

Problem Formulation

Synthesis of Gate Control Lists

Scheduling Problem

Find an assignment of frames to transmission windows at each egress port along the routed path, such that:

- The number of windows fits in the GCL of each port,
- The window is assigned to one of the port queues,
- The length of the window fits the length of all assigned frames,

Satisfying that:

- No two windows at the same port overlap in time,
- Frames of the same flow propagate with the right order and latency along the route,
- The required end-to-end latency of flows is fulfilled,

With the ultimate objective:

- Evaluate the suitability of Array Theory and the use of SMT/OMT solvers in solving the problem.

Array Theory Encoding

We use SMT/OMT solvers to leverage the computation of feasible solutions

Satisfiability Modulo Theories

- ✓ satisfiability of logical formulas in first-order formulation
- ✓ background theories
  - ✓ $\mathcal{LA}(\mathbb{Z})$ $\mathcal{BV}$ $\mathcal{T}_A$

Array Theory

○ Variables	$x_1, x_2, \ldots, x_n$
○ Arrays	$a[i]$ $a\langle i \leftarrow e \rangle$
○ Logical Symbols	$\vee, \wedge, \neg, (,)$
○ Non-logical Symbols	$+, =, \%, \leq$
○ ~~Quantifiers~~	~~$\exists, \forall$~~

Optimization Modulo Theories (OMT)

- ✓ Additional optimization of variables of simple expressions
  - ✓ optimization (OMT) [Bjørner@TACAS15]

Tool support

- ❖ Many solvers exist as well as a very active community
  - ❖ OpenSMT [Bruttomesso@TACAS10]
  - ❖ CVC4 [Barrett@CAV11]
  - ❖ Yices [Dutertre@CAV14]
  - ❖ Z3 [de Moura@TACAS08]

Why Array Theory?

Arrays theories allows building expressions upon uninterpreted ”*lists of elements*”

An array in $\mathcal{T}_A$ is an indexed list of elements, for which the index (an integer) may be an uninterpreted term itself.

- This allows building the expressions satisfying the communication constraints based on unassigned frame-to-window sets.

$$\forall a : array, \forall i, j : index, \forall x : elem$$
$$i = j \rightarrow a\langle i \leftarrow x\rangle[j] = x$$
$$i \neq j \rightarrow a\langle i \leftarrow x\rangle[j] = a[j]$$

We define three arrays for each egress port, denoting:

- Window start time
- Window end time
- Window queue
- And each frame on that port is given a variable denoting the respective index in all three arrays (frame-to-window assignment)
- The solution to this variable decides the **frame-to-window** assignment fulfilling the system of inequalities defining a feasible schedule

How does it compare to (our) previous work?

Similar problems with different solutions

Steiner @ RTSS 2010

- Based on TTEthernet
- Single frames

Craciunas et al. @ RTNS 2014

- TTEthernet + Tasks
- Single frames
- 0-jitter

@RTS Journal 2016

- Improved algorithms
- Scalability
- Optimization

Craciunas et al. @RTNS 2016

- TSN: Frame-based Qbv scheduler
- 0-jitter
- Issolation of frames or streams in one window
- Schedule not tailored to Gate Control Lists.

This work

- Based on TSN
- Multi-frame streams
- Window-based scheduling
- Direct mapping to Gate Control Lists
- Allows jitter between periodic instances
- … but can also enforce 0-jitter
- Better use of queues

04

Formalization of Constraints

Examples

System of Inequalities

We construct a system of inequalities formalizing the technology and user-constraints derived from the problem formulation, applied to our model:

$$\forall s_i \in \mathcal{S} : \forall (a,b) \in \mathcal{E} : \forall j \in \left[0, \frac{T_s}{T_i} - 2\right] : \\ \left(\phi^{(a,b)}[\omega_{i,j+1}^{(a,b)}] - \phi^{(a,b)}[\omega_{i,j}^{(a,b)}] = T_i\right).$$

$$\forall s_i \in \mathcal{S} : \forall (a,b) \in \mathcal{E} : \forall j \in \left[0, \frac{T_s}{T_i} - 1\right] : \\ \left(\tau^{(a,b)}[\omega_{i,j}^{(a,b)}] - \phi^{(a,b)}[\omega_{i,j}^{(a,b)}] = l_i^{(a,b)}\right).$$

$$\forall (a,b) \in \mathcal{E} : \\ \left(\phi^{(a,b)}[1] \geq 0\right) \wedge \left(\tau^{(a,b)}[\mathcal{W}^{(a,b)}] < T_s\right)$$

$$\forall s_i \in \mathcal{S} : \forall (a,b) \in \mathcal{E} : \forall j \in \left[0, \frac{T_s}{T_i} - 1\right] : \\ \left(\phi^{(a,b)}[\omega_{i,j}^{(a,b)}] \geq j \times T_i\right) \wedge \\ \left(\tau^{(a,b)}[\omega_{i,j}^{(a,b)}] < (j+1) \times T_i\right),$$

$$\forall (a,b) \in \mathcal{E} : \forall k \in \{1, \ldots, \mathcal{W}^{(a,b)}\} \\ \tau^{(a,b)}\langle k \leftarrow \phi^{(a,b)}[k] \rangle.$$

$$\forall (a,b) \in \mathcal{E} : \forall f_{i,j}^{(a,b)} \in \mathcal{F}^{(a,b)} : \\ \left(\omega_{i,j}^{(a,b)} \geq 1\right) \wedge \left(\omega_{i,j}^{(a,b)} \leq \mathcal{W}^{(a,b)}\right)$$

$$\forall k \in \left[0, \frac{T_s}{T_i} - 1\right] : \forall l \in \left[0, \frac{T_s}{T_j} - 1\right] : \\ \left(\left(\tau^{(a,b)}[\omega_{i,k}^{(a,b)}] + \delta \leq \phi^{(y,a)}[\omega_{j,l}^{(y,a)}]\right) \vee \\ \left(\tau^{(a,b)}[\omega_{j,l}^{(a,b)}] + \delta \leq \phi^{(x,a)}[\omega_{i,k}^{(x,a)}]\right)\right) \vee \\ \left(\kappa^{(a,b)}[\omega_{i,k}^{(a,b)}] \neq \kappa^{(a,b)}[\omega_{j,l}^{(a,b)}]\right) \vee \left(\omega_{i,k}^{(a,b)} = \omega_{j,l}^{(a,b)}\right)$$

$$\forall (a,b) \in \mathcal{E} : \forall i,j \in \{1, \ldots, \mathcal{W}^{(a,b)}\}, i \neq j : \\ \left(\tau^{(a,b)}[i] \leq \phi^{(a,b)}[j]\right) \vee \left(\tau^{(a,b)}[j] \leq \phi^{(a,b)}[i]\right)$$

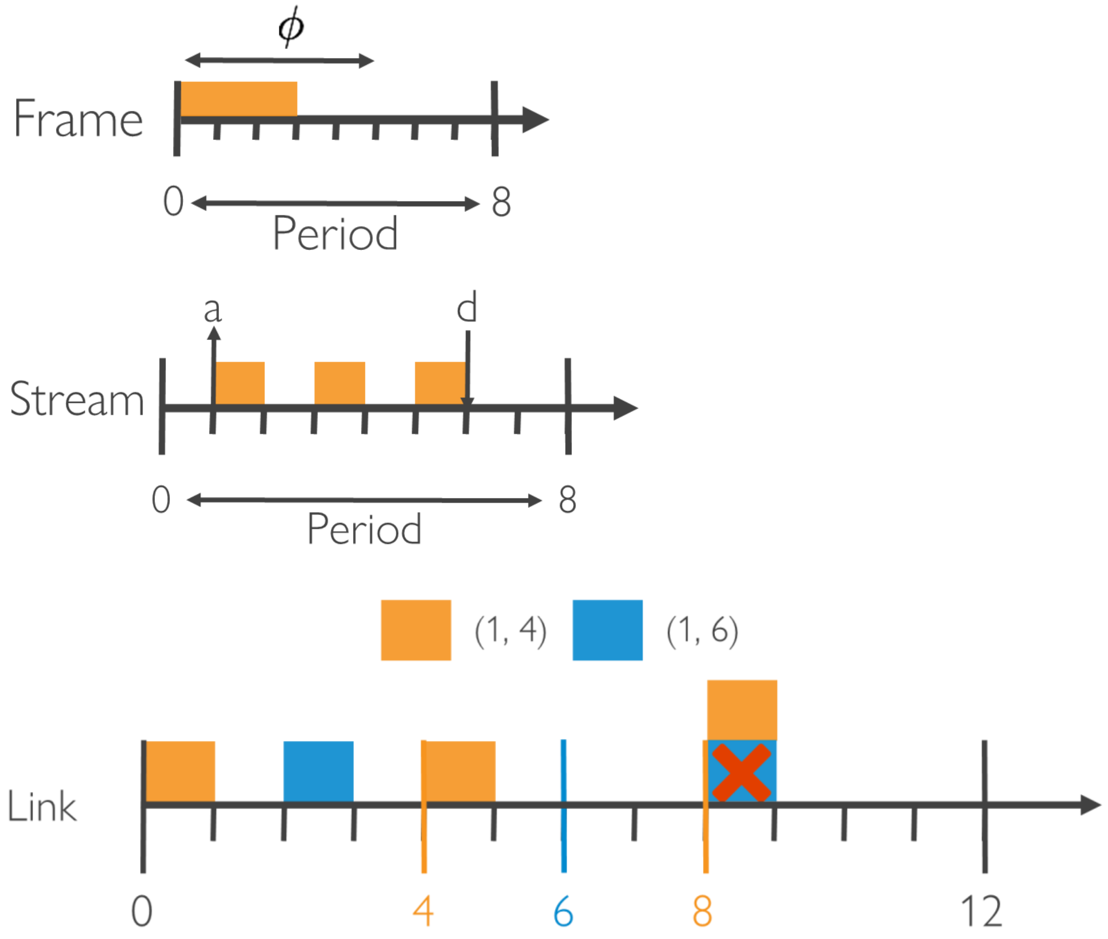
$$\forall (a,b) \in \mathcal{E} : \forall i \in \{1, \ldots, \mathcal{W}^{(a,b)} - 1\} : \\ \tau^{(a,b)}[i] \leq \phi^{(a,b)}[i+1],$$

$$\forall (a,b) \in \mathcal{E} : \forall k \in \{1, \ldots, \mathcal{W}^{(a,b)}\} : 0 \leq \kappa^{(a,b)}[k] < \aleph_{tt}$$

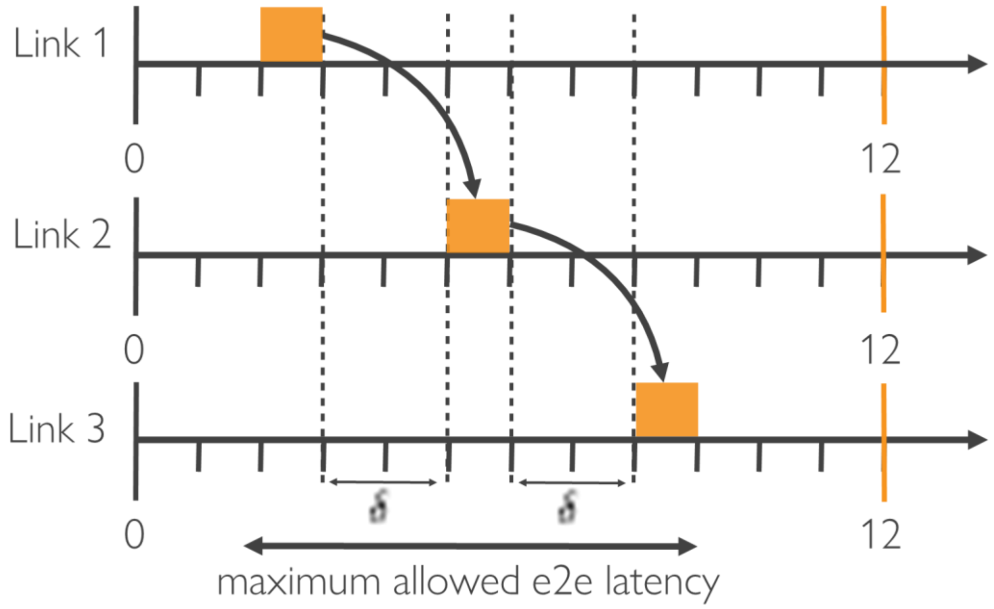
$$\forall (a,b) \in \mathcal{E} : \forall k \in \{1, \ldots, \mathcal{W}^{(a,b)}\} : \\ \left(\phi^{(a,b)}[k] \geq 0\right) \wedge \left(\tau^{(a,b)}[k] < T_s\right),$$

System of Inequalities

Technology constraints model the necessary communication requirements



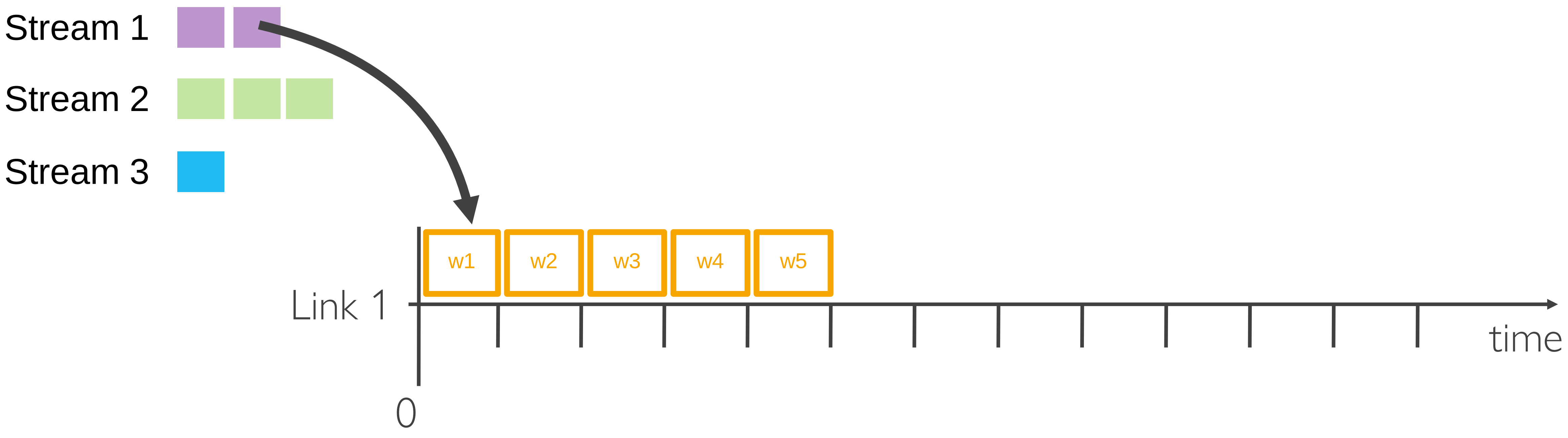
User constraints model optional communication requirements



optimization, …

Frame-to-Window Assignment

Each frame is assigned to a window in the respective egress port



System of Inequalities

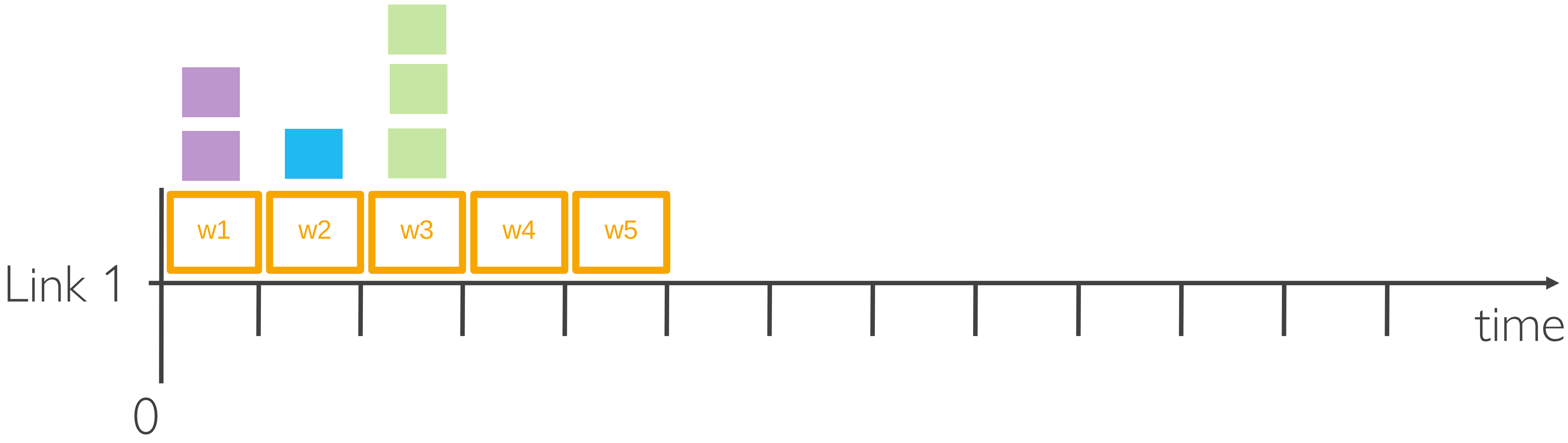
Frame-to-Window Assignment

Each frame is assigned to a window in the respective egress port

Stream 1

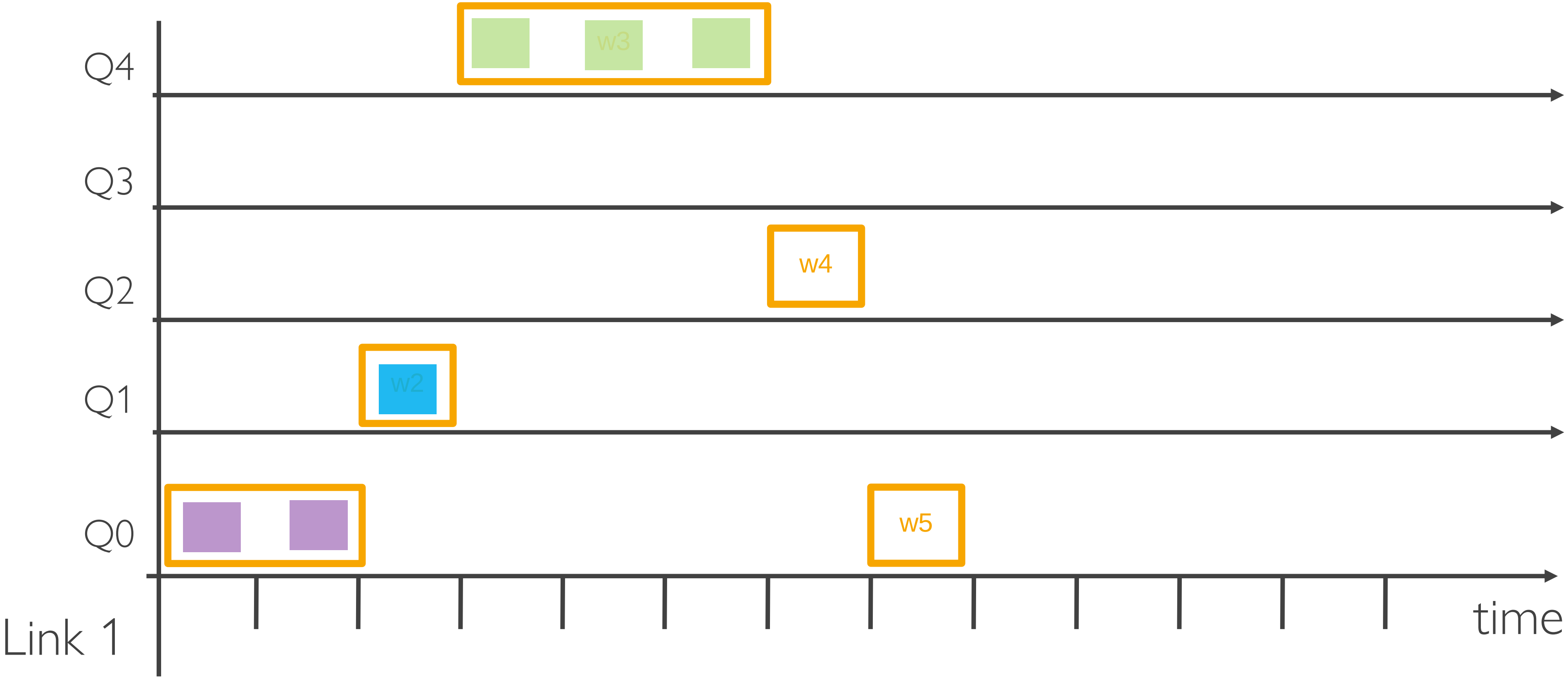
Stream 2

Stream 3



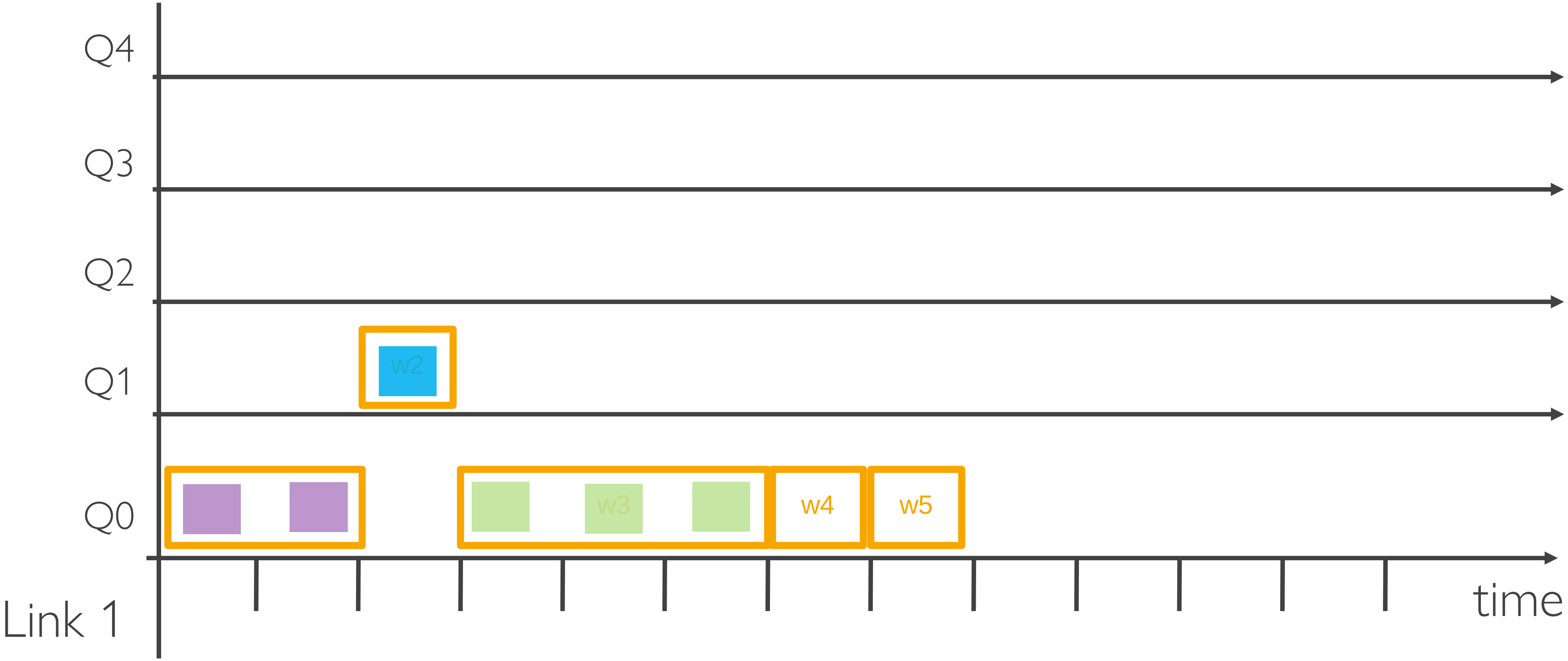
System of Inequalities

Values for window start, end, and queues are derived from the constraints…



System of Inequalities

Additional user-constraints as well as optimization shall be satisfied…

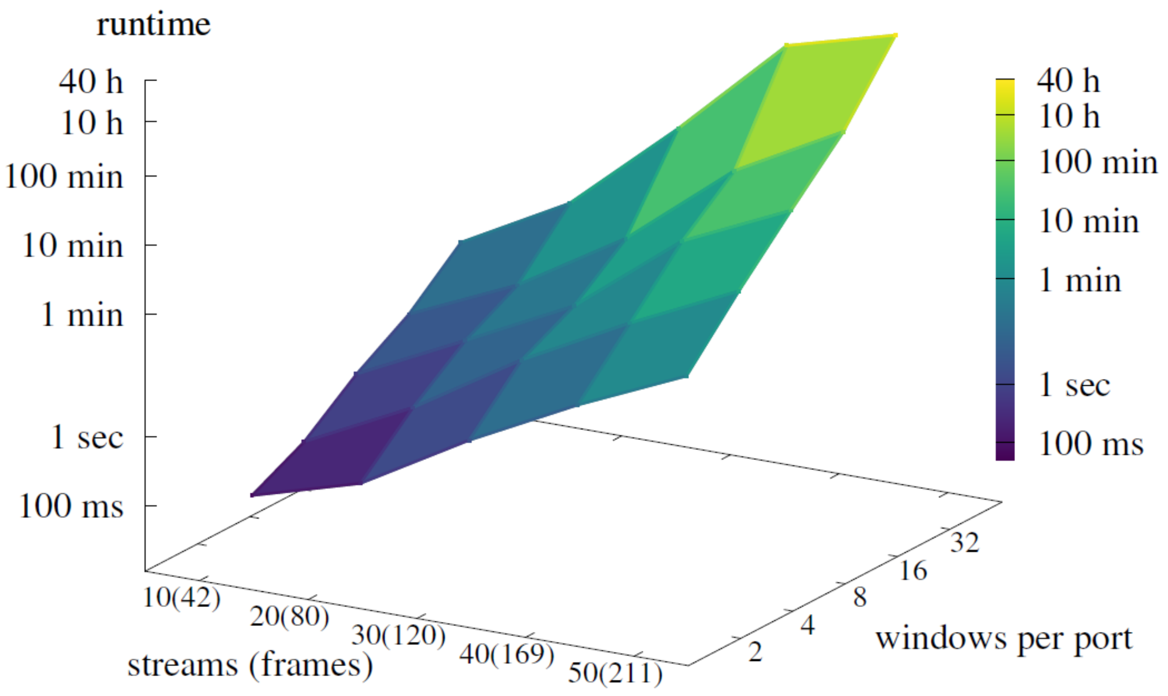


05

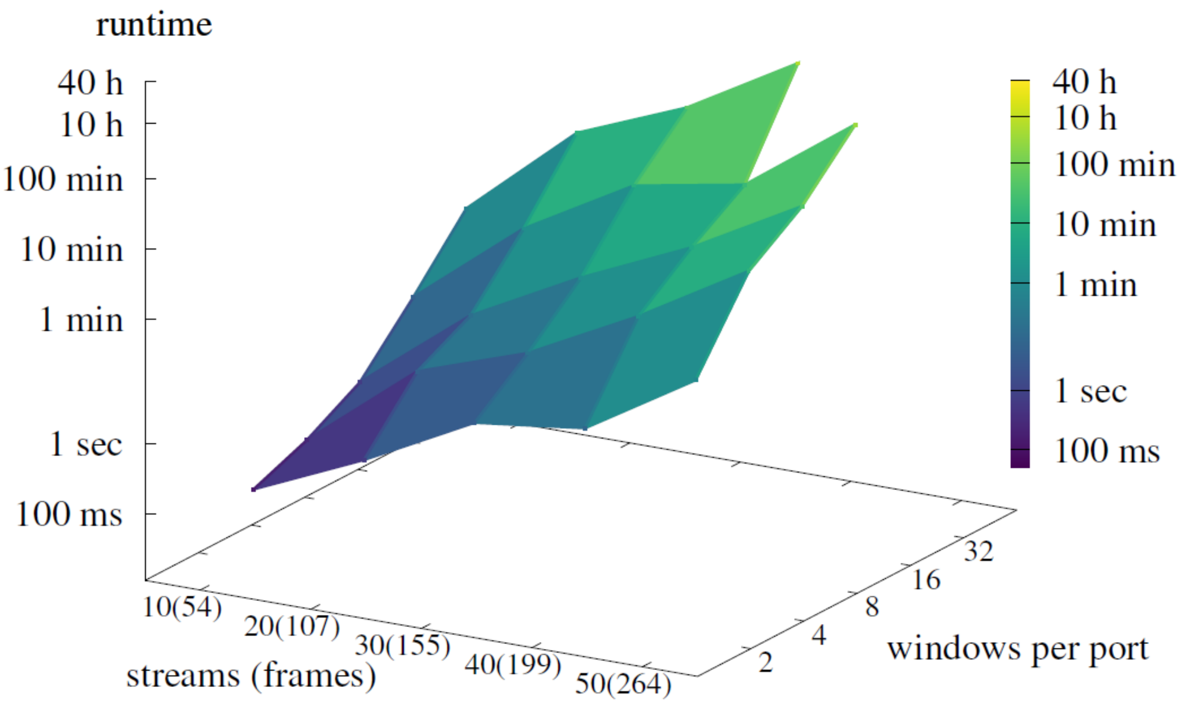
Evaluation Results

Synthesis Time with two topology sizes

Streams with periods {10,20}ms, 4 queues



5 switches, 15 end-systems

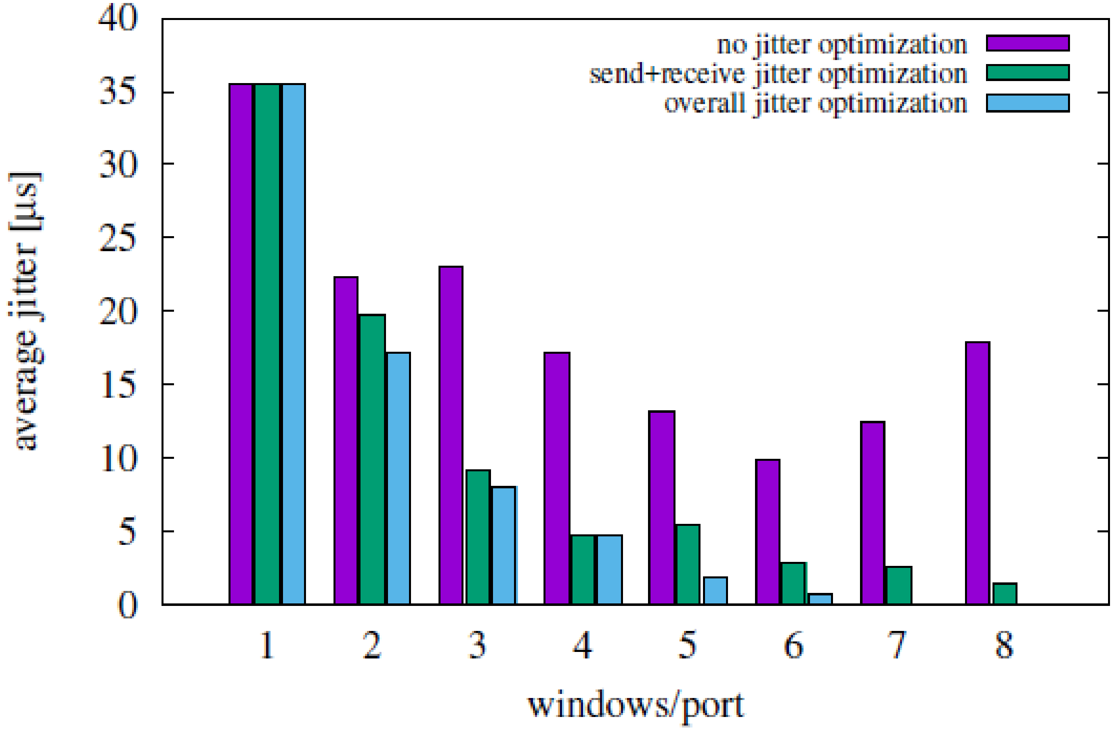


10 switches, 50 end-systems

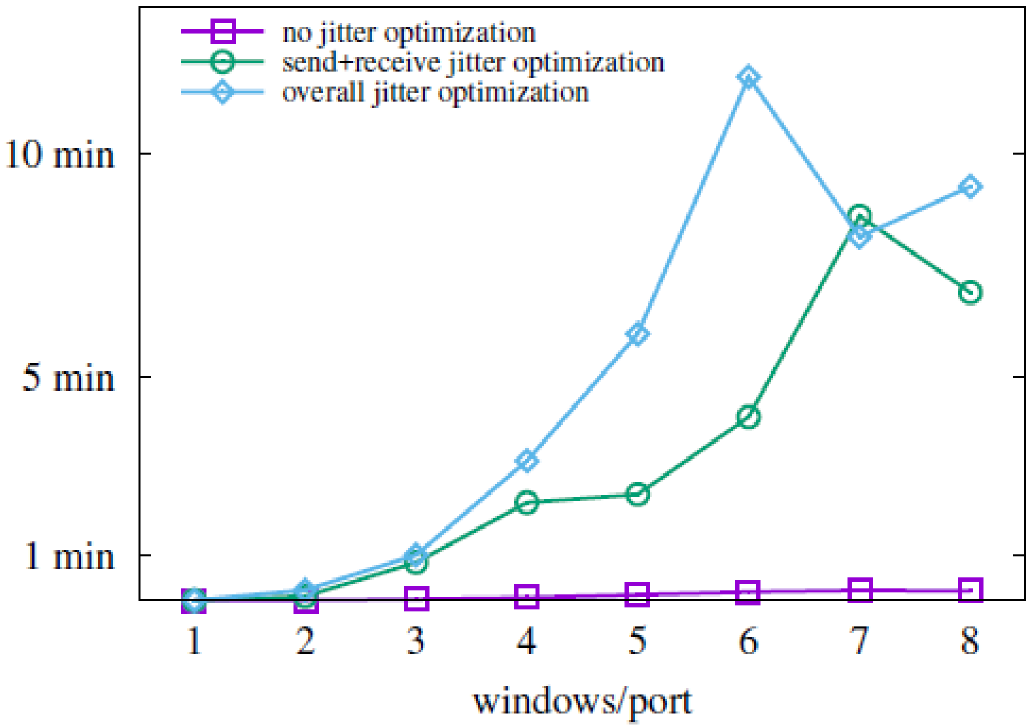
Evaluation Results

Jitter Optimization vs Synthesis Time

5 switches, 15 end-systems



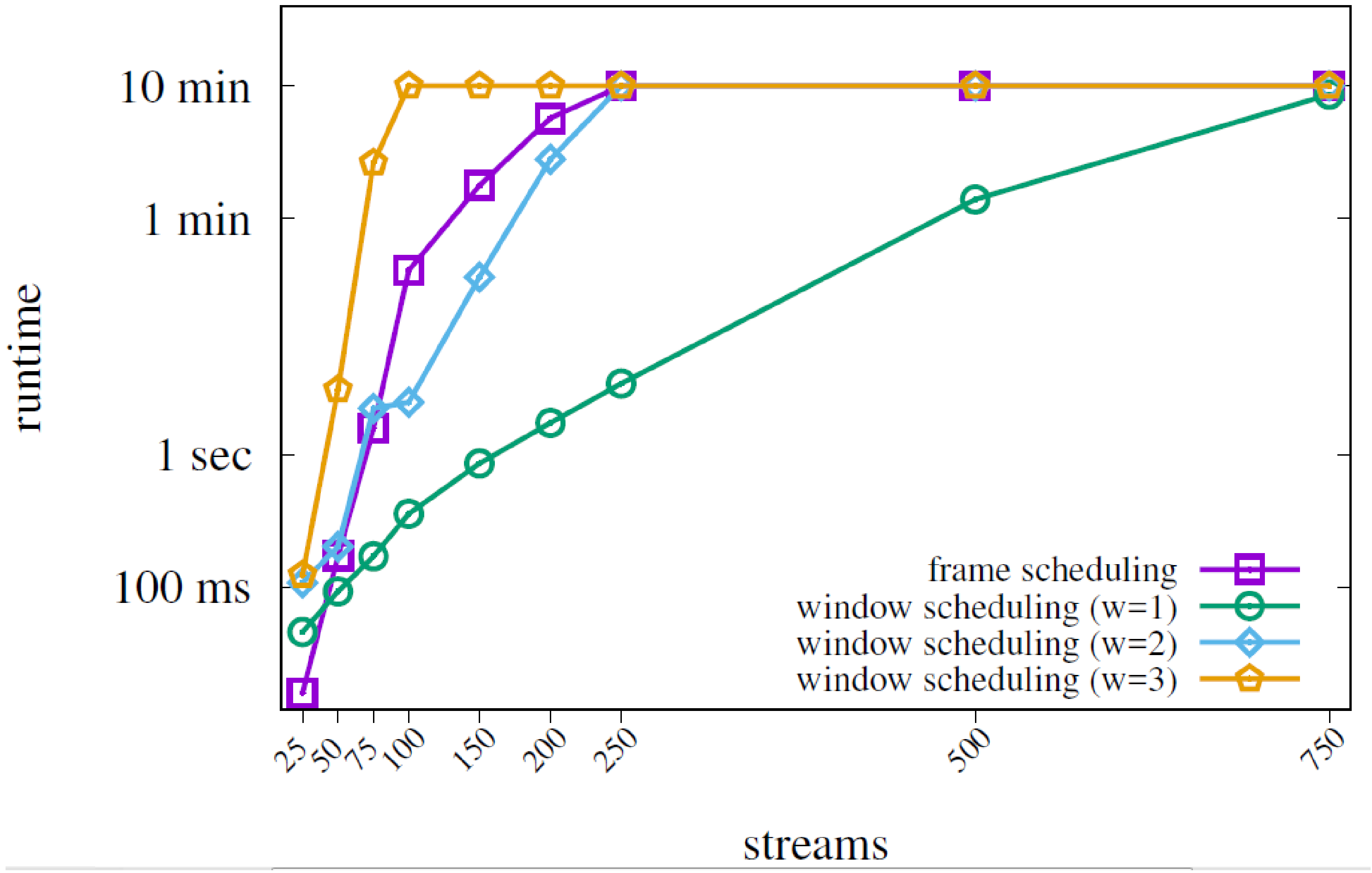
Average jitter



Synthesis Time

Evaluation Results

Scalability comparison against frame-based scheduler (Craciunas et al @RTNS 16)



06

Conclusions

Future Directions

Conclusions

Main conclusions of this work

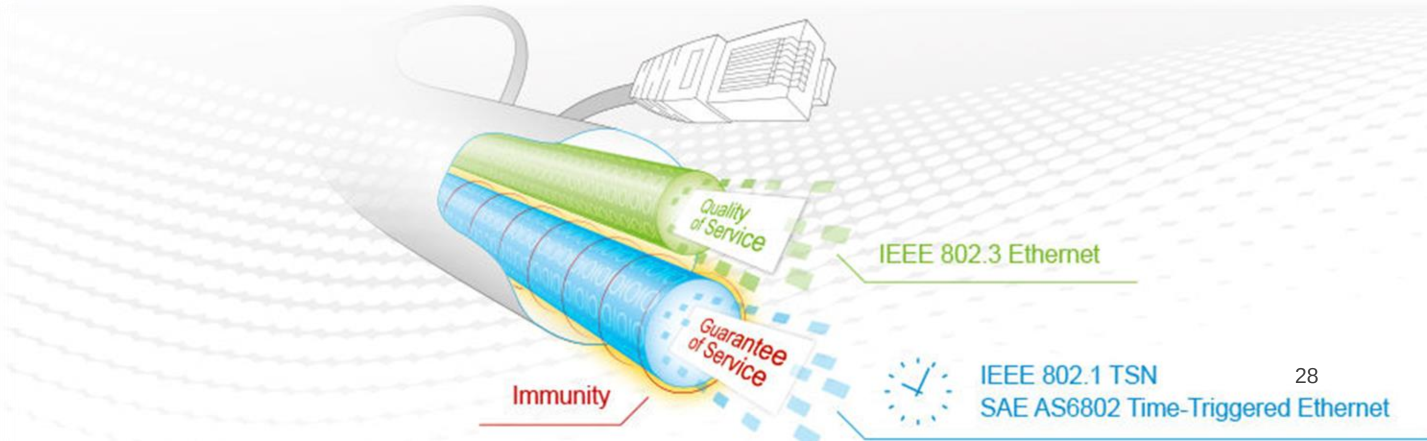
Window-based scheduling and synthesis of 802.1Qbv Gate Control Lists proves feasible.

Array Theory encoding suits the problem and allows leveraging the heavy work to SMT/OMT solvers.

Our new model relaxing jitter requirements maps better to TSN general concepts.

Yet 0-jitter requirements can still be met.

The methods exhibits potential to solve large networks with the aid of incremental algorithms.



Future Directions

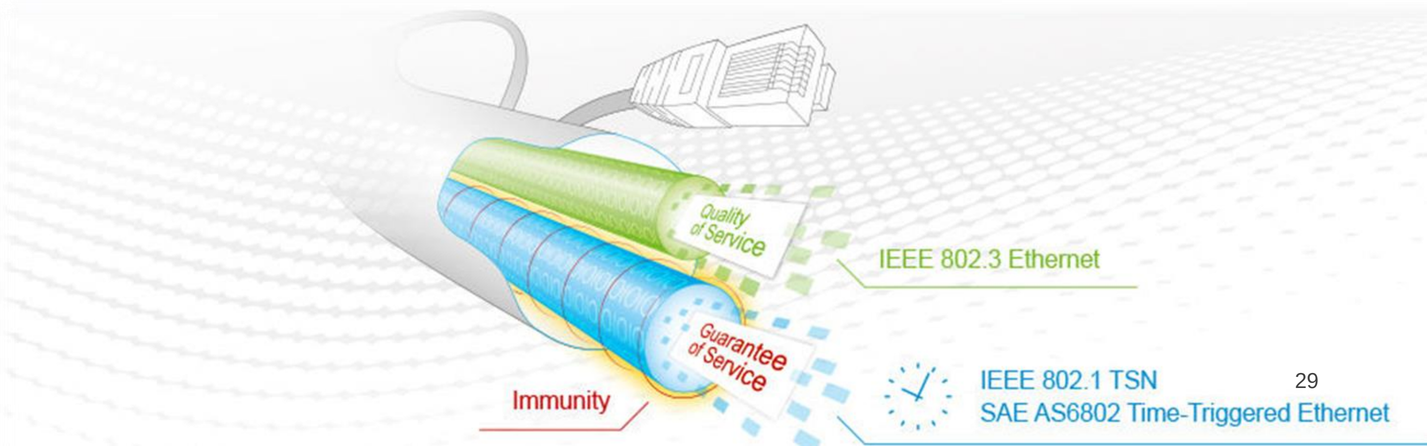
There‘s a lot left to be explored…

Incremental algorithms applied along multiple dimensions (streams, windows, queues, …)

For very large networks heuristic-guided algorithms seem best.

For multi-path topologies route decisions can be included to the problem.

Stream-redundancy, fault-tolerance, etc…



Thank You

Questions?

TTTech

Vienna, Austria (Headquarters)	USA	Japan	China
Phone +43 1 585 34 34-0 office@tttech.com	Phone +1 978 933 7979 usa@tttech.com	Phone +81 52 485 5898 office@tttech.jp	Phone +86 21 5015 2925-0 china@tttech.com

www.tttech.com