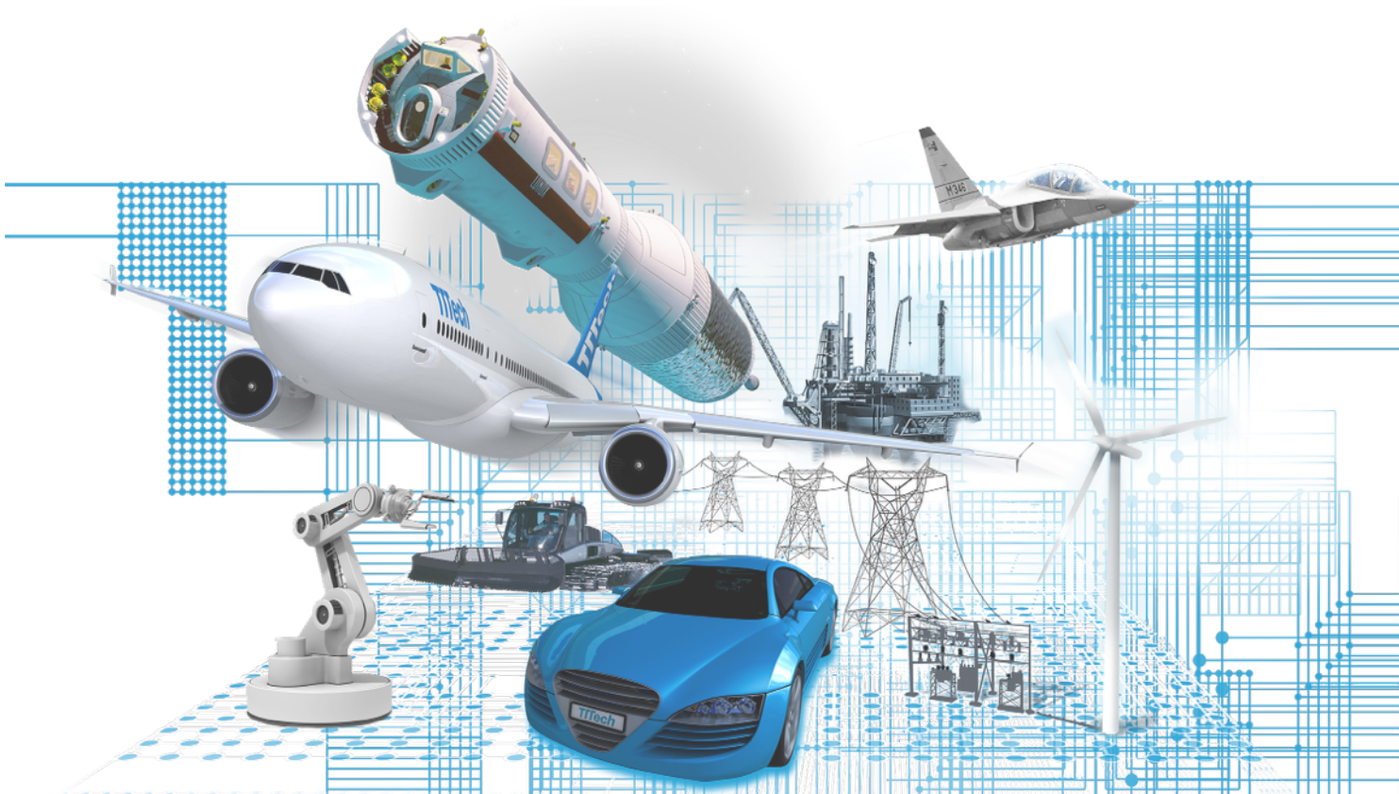
Ensuring Reliable Networks TTTech

SMT-based task- and network-level static schedule generation for time-triggered networked systems

Silviu S. Craciunas, Ramon Serna Oliver
TTTech Computertechnik AG
RTNS 2014, Versailles, France, October 5-8, 2014



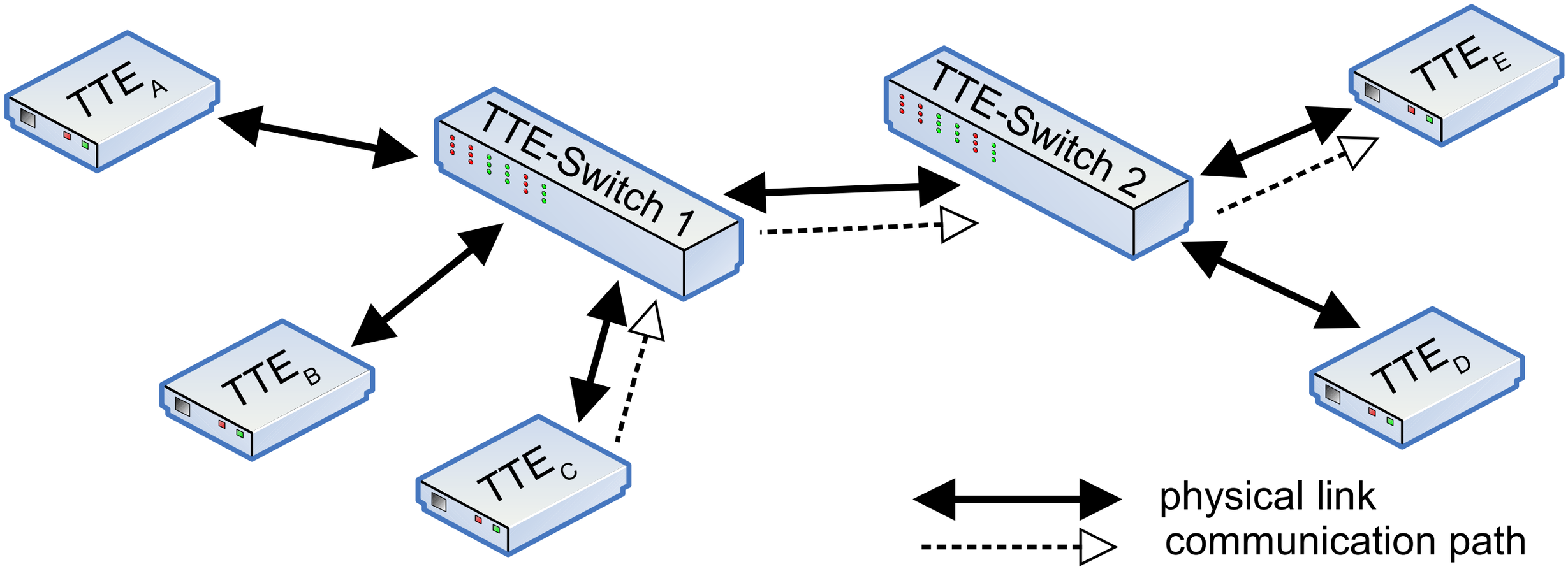
DREAMS

Time Triggered Networks

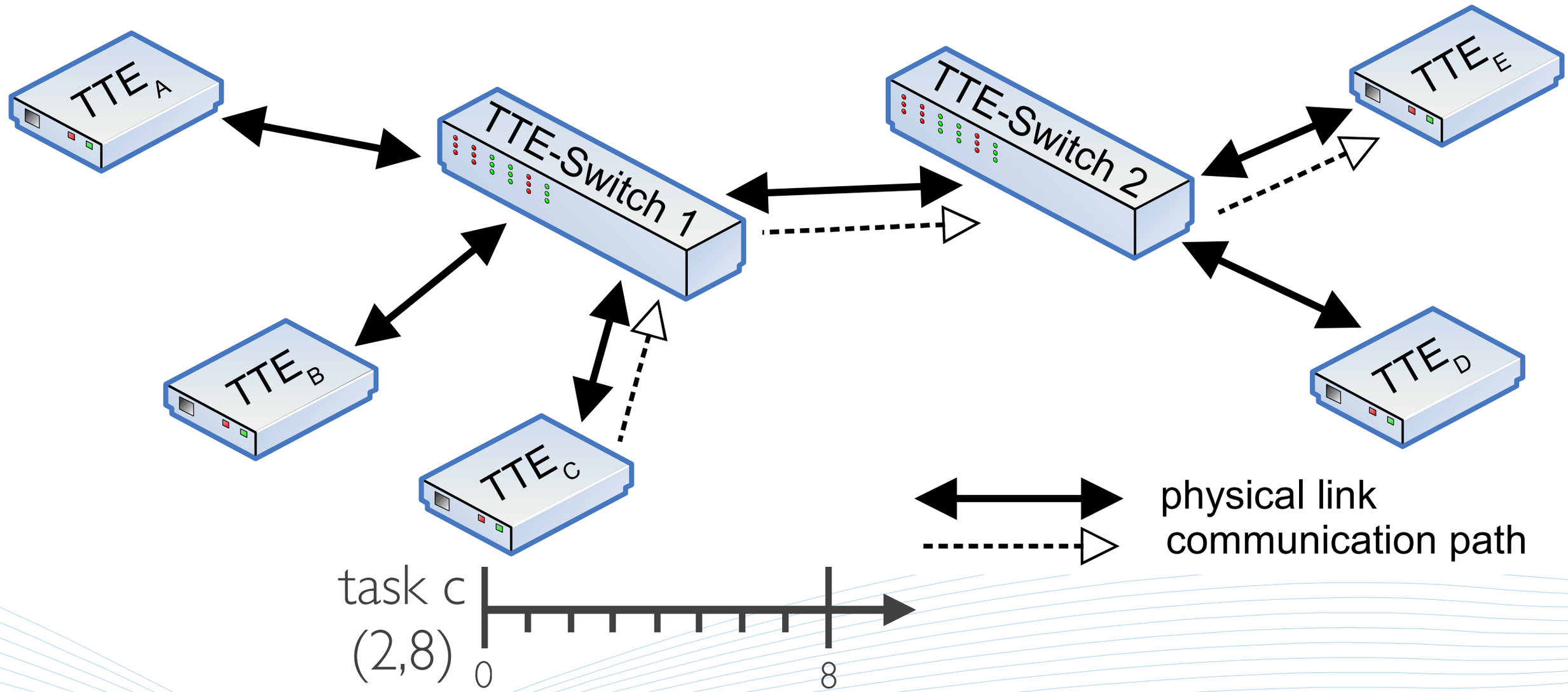
Ensuring Reliable Networks TTTech

Time triggered communication

Ensuring Reliable Networks TTTech

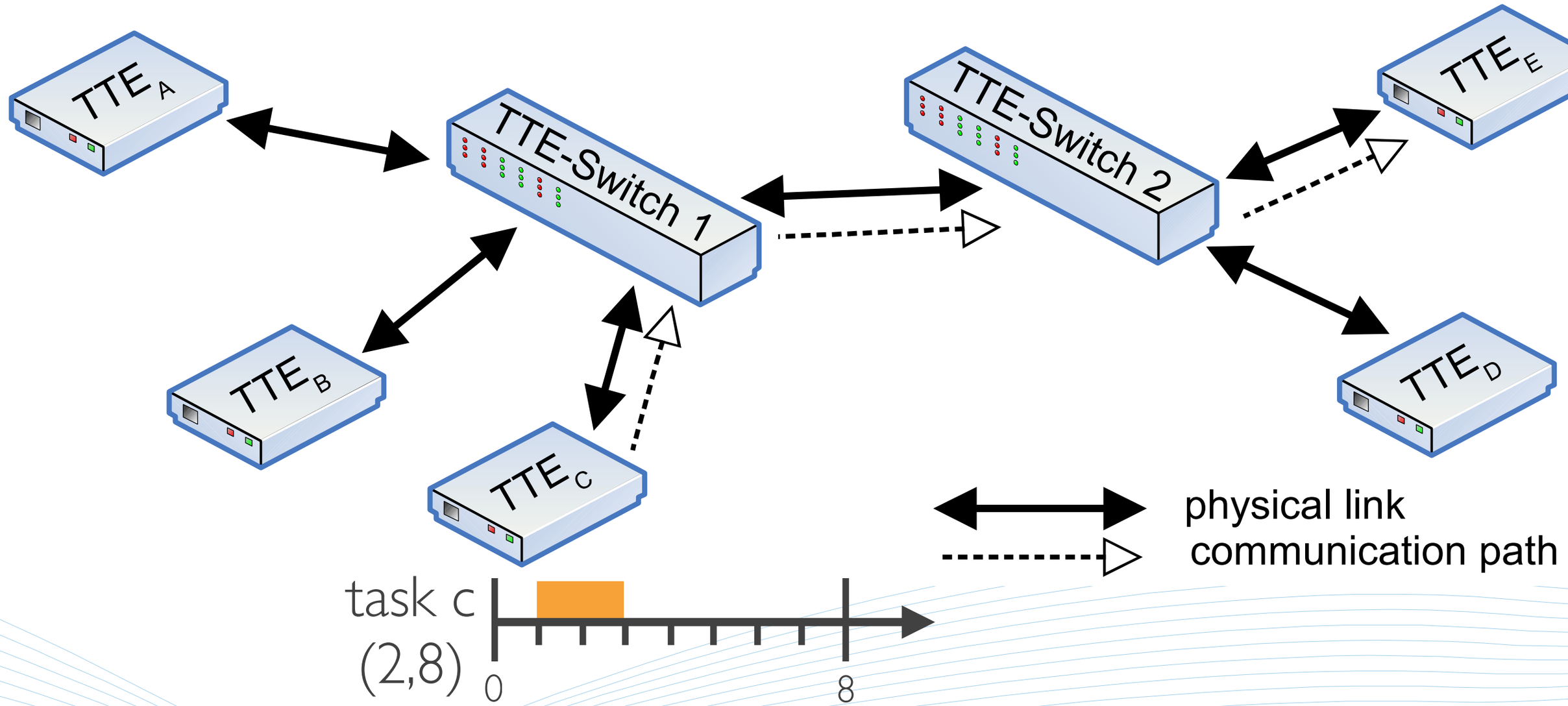


Time triggered communication

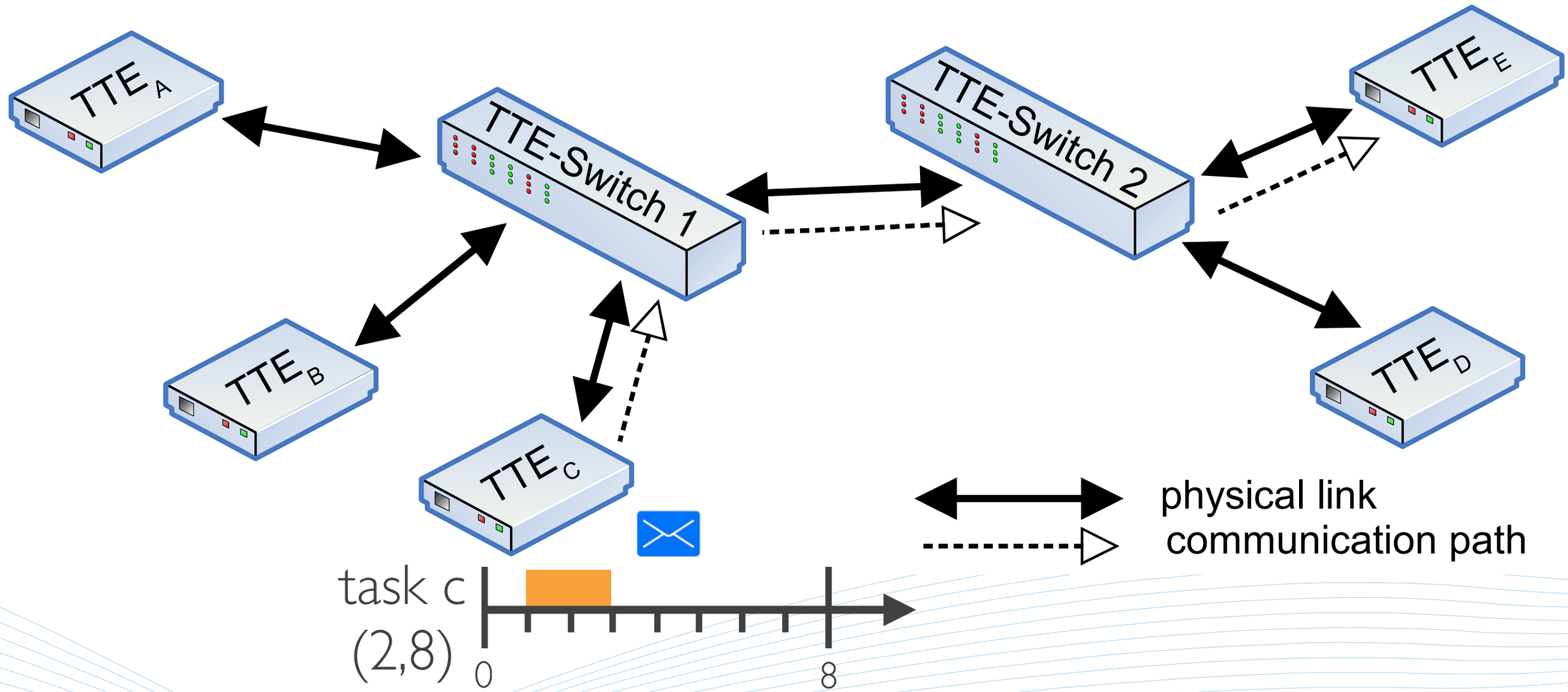


Time triggered communication

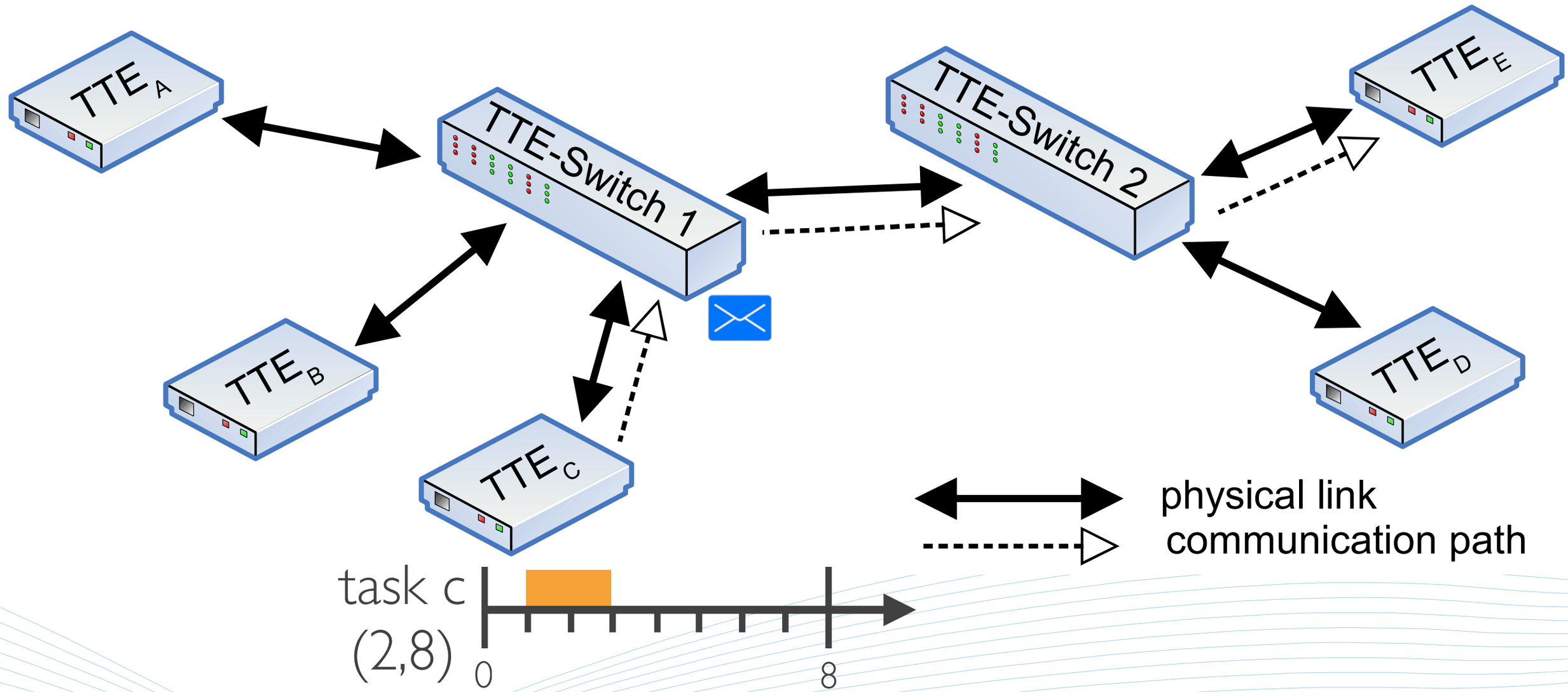
Ensuring Reliable Networks TTTech



Time triggered communication

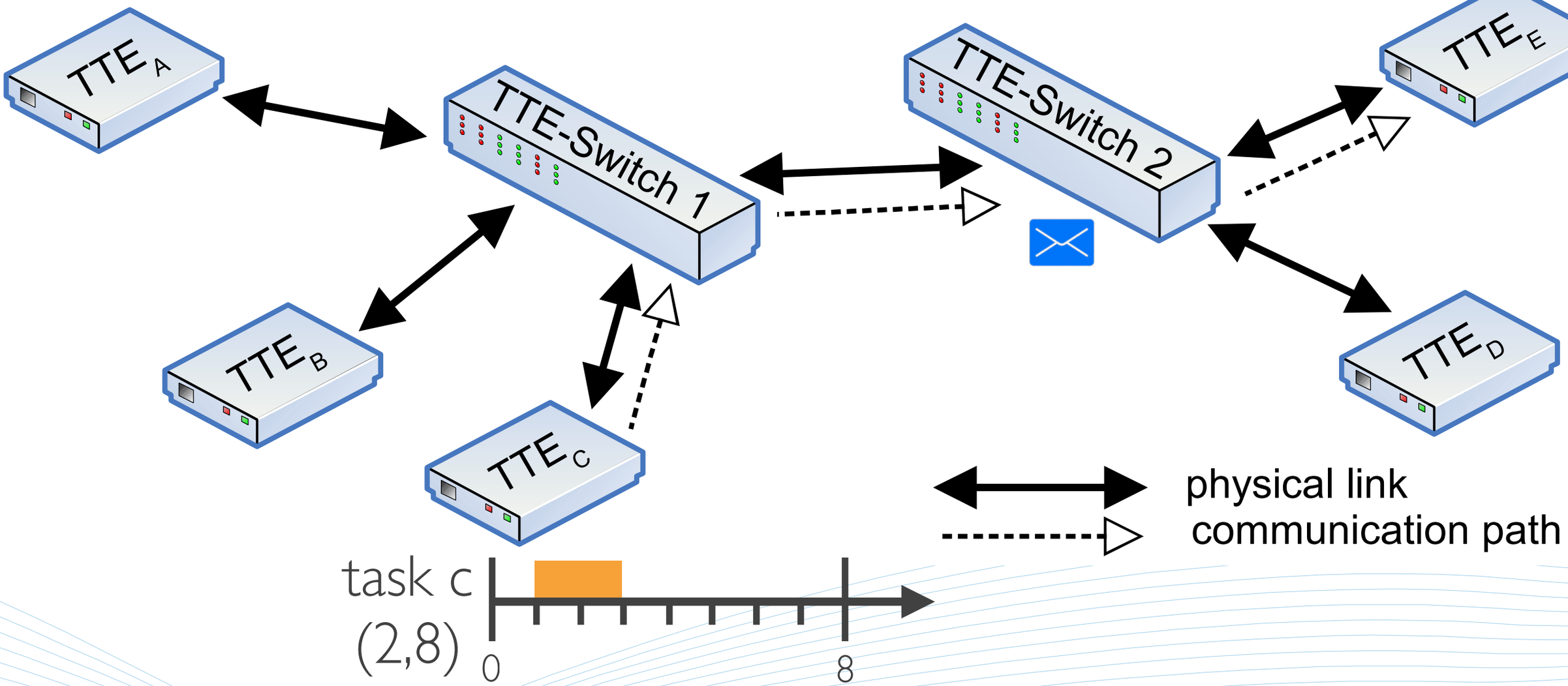


Time triggered communication



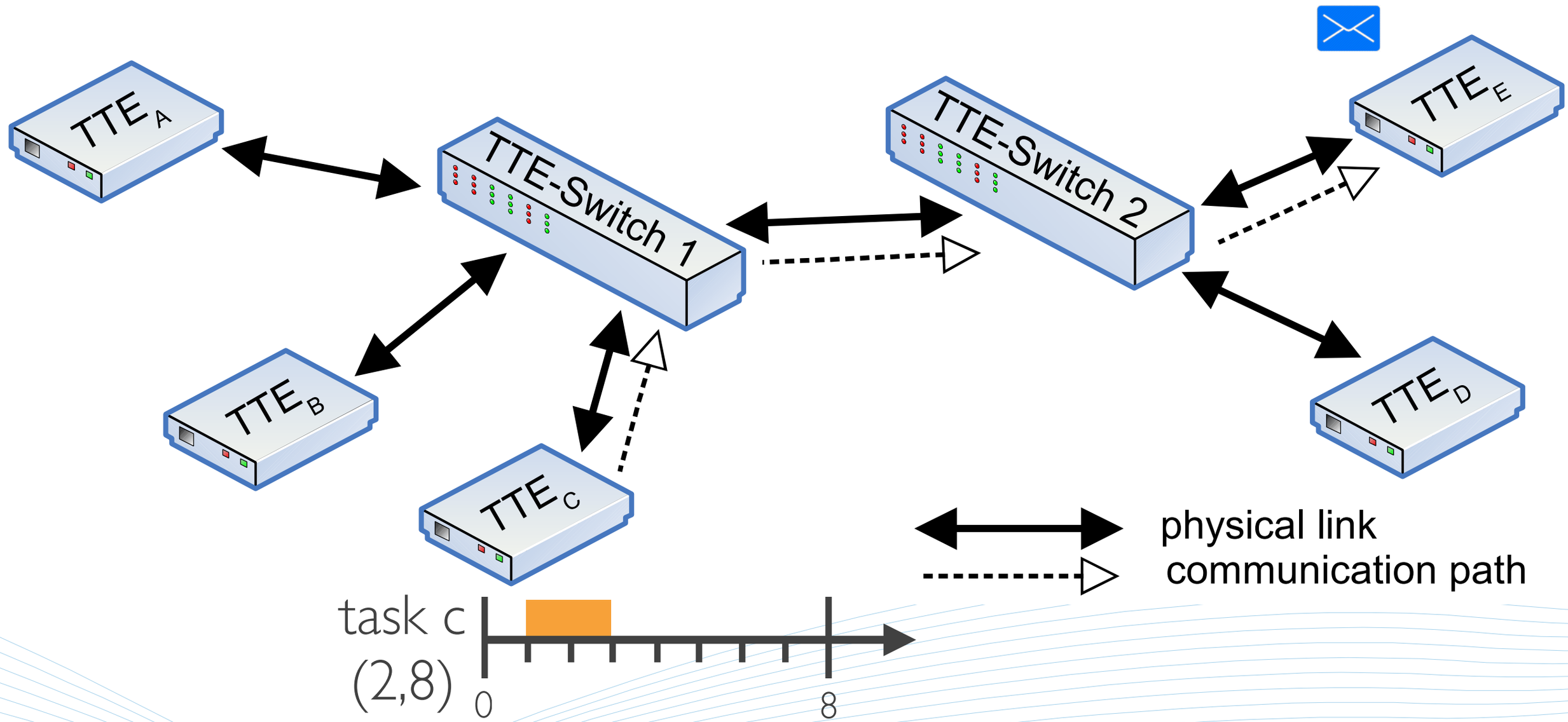
Time triggered communication

Ensuring Reliable Networks TTTech

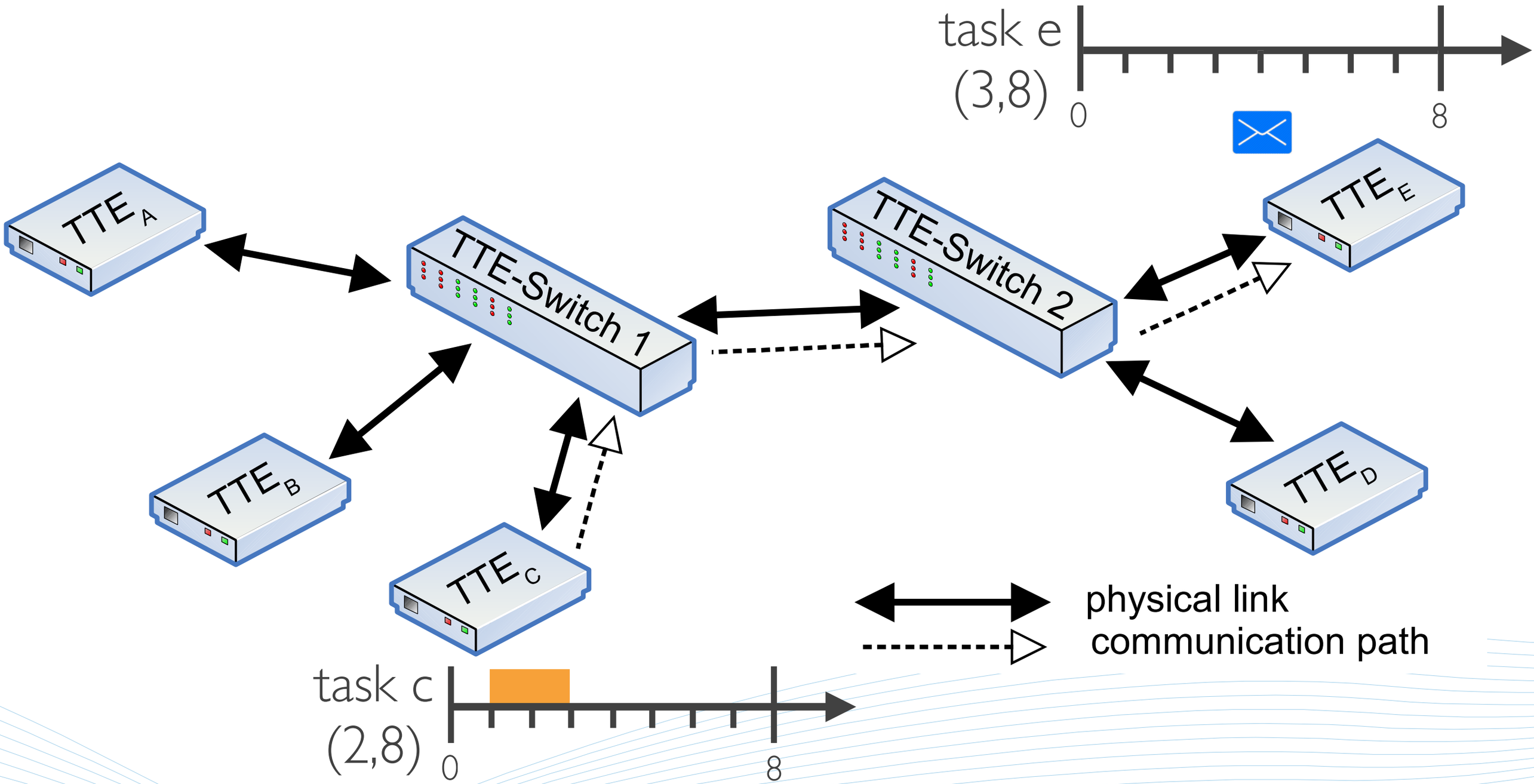


Time triggered communication

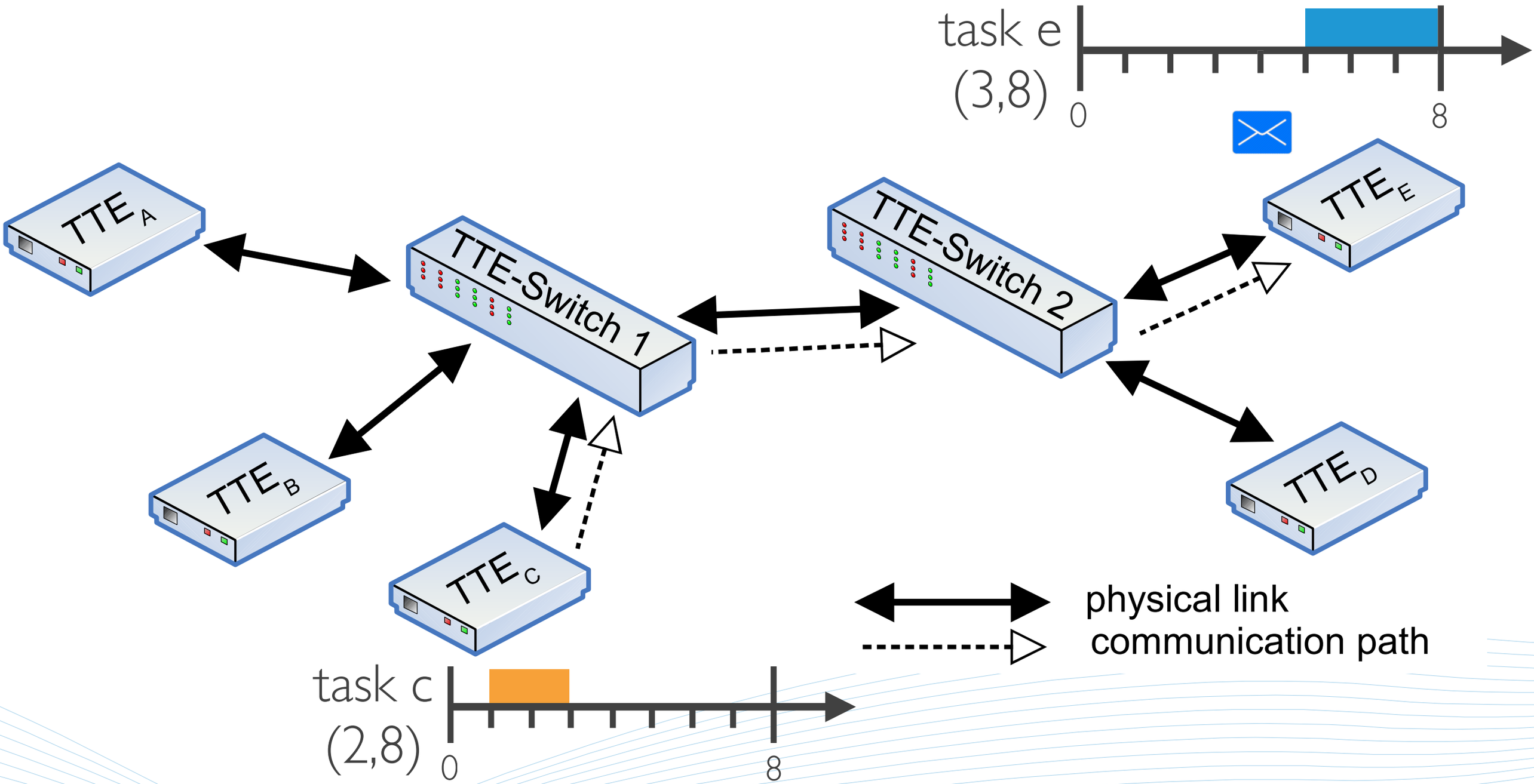
Ensuring Reliable Networks TTTech



Time triggered communication



Time triggered communication



Scheduling

Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

Network model

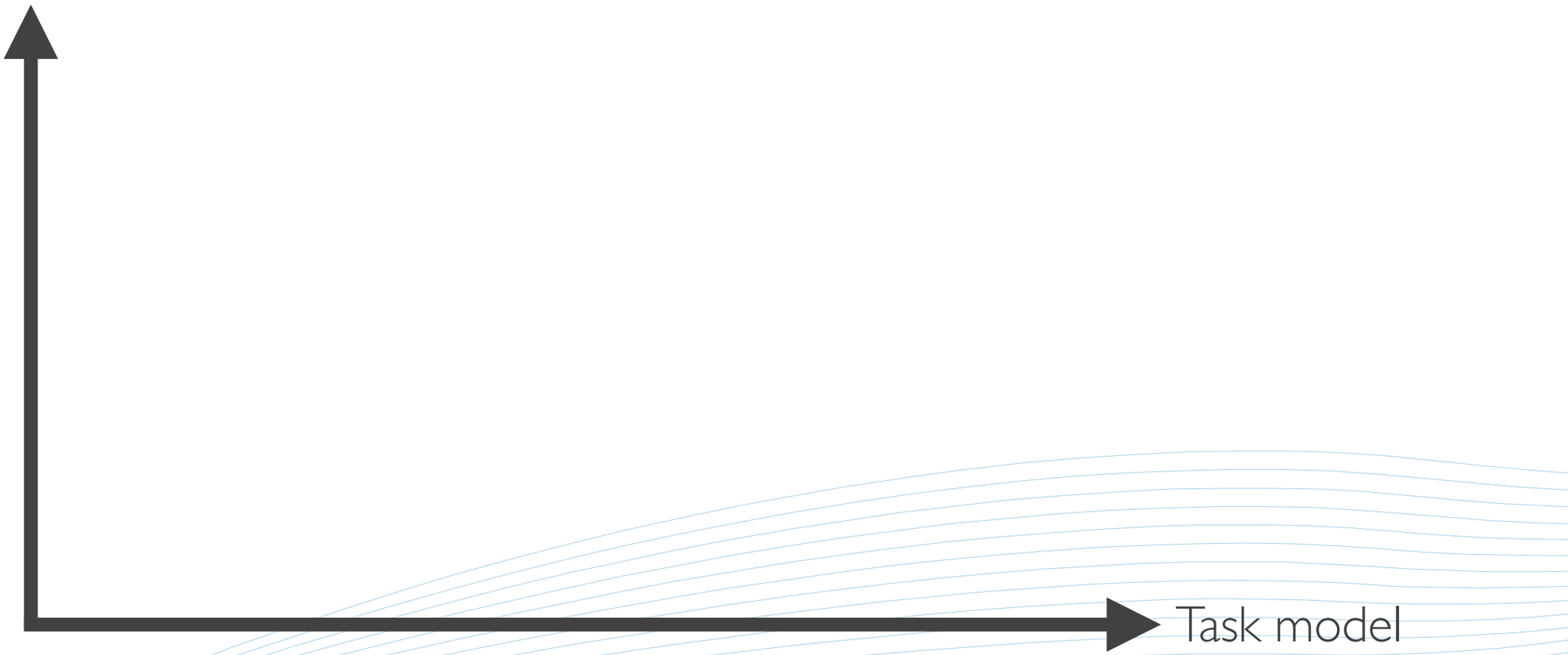
Scheduling

Ensuring Reliable Networks TTTech

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling



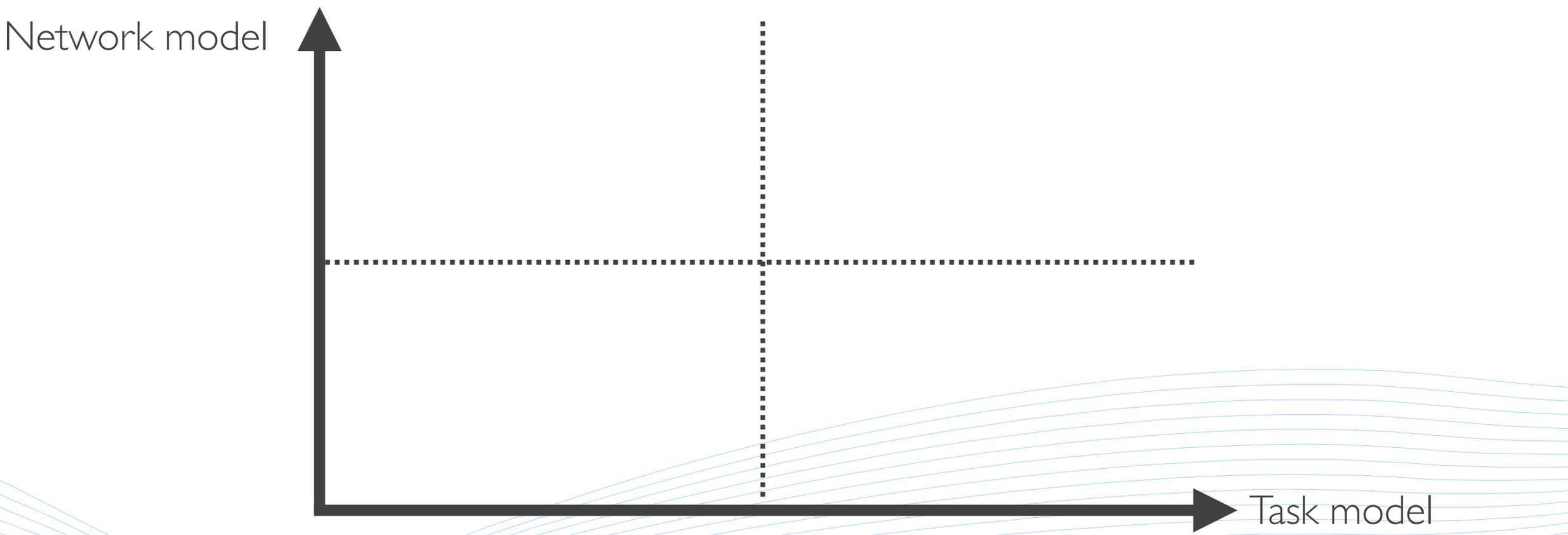
Scheduling

Ensuring Reliable Networks TTTech

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

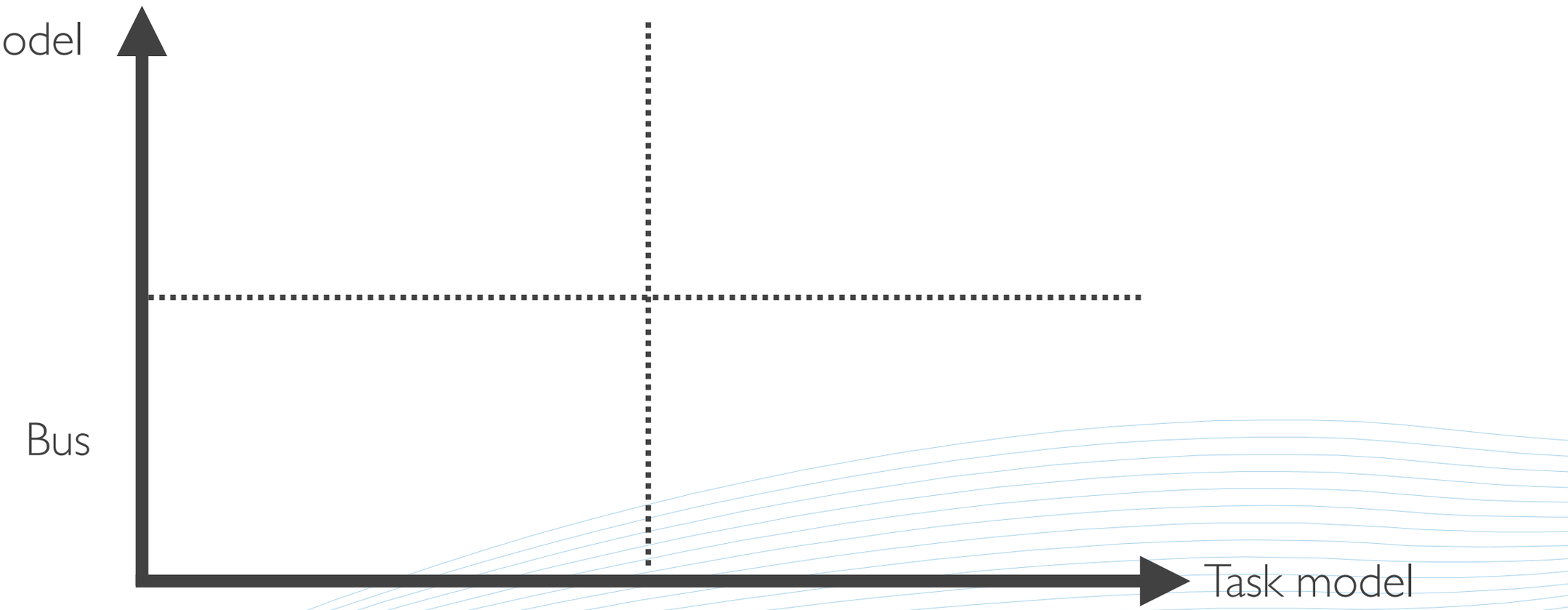


Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

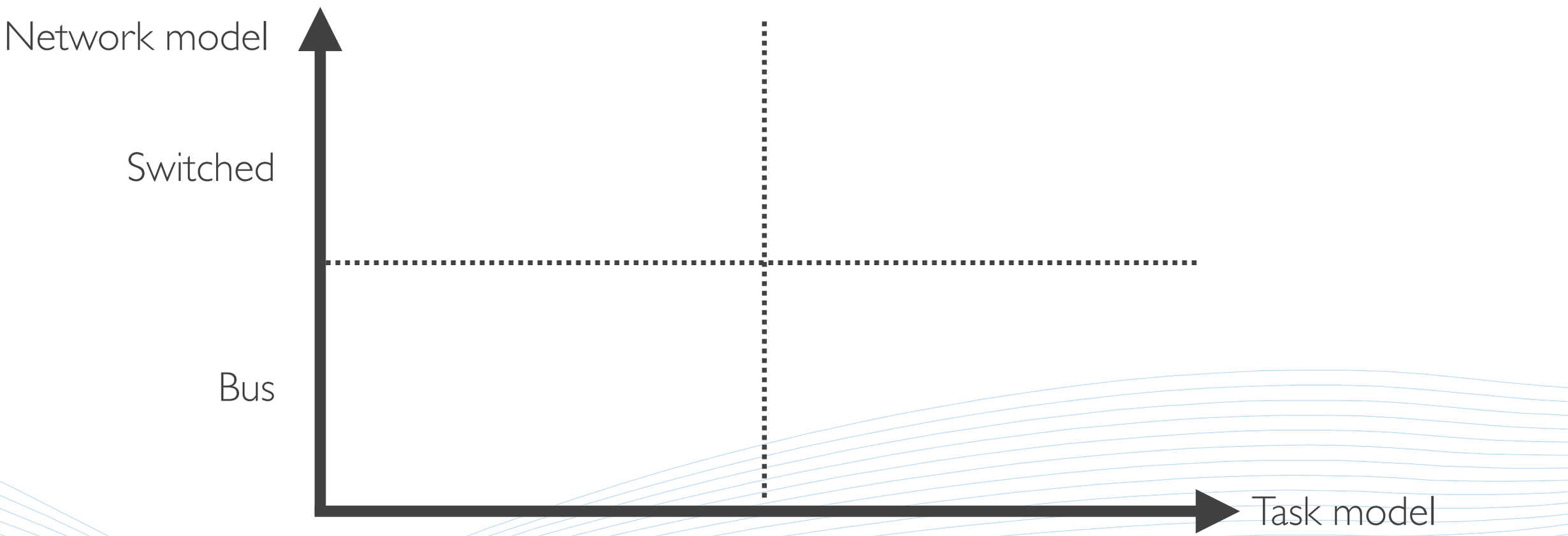


Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

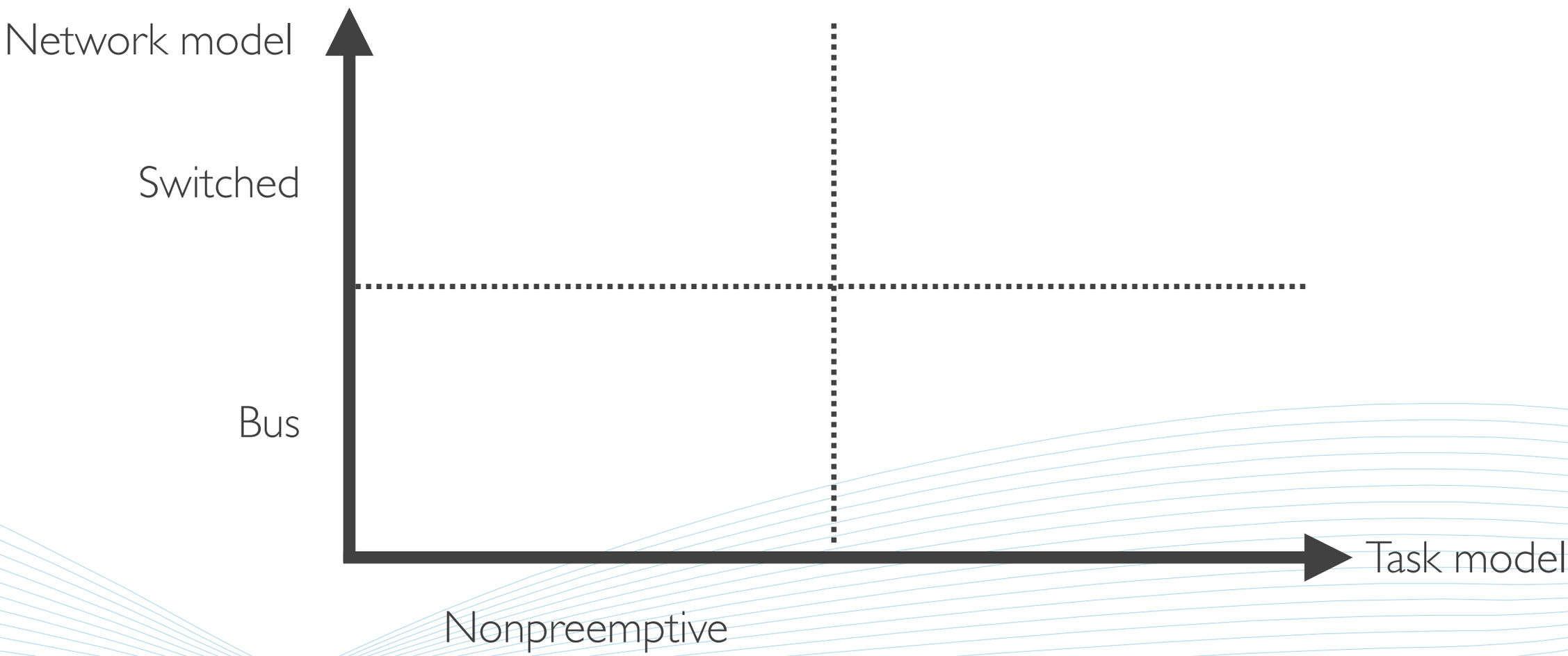


Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

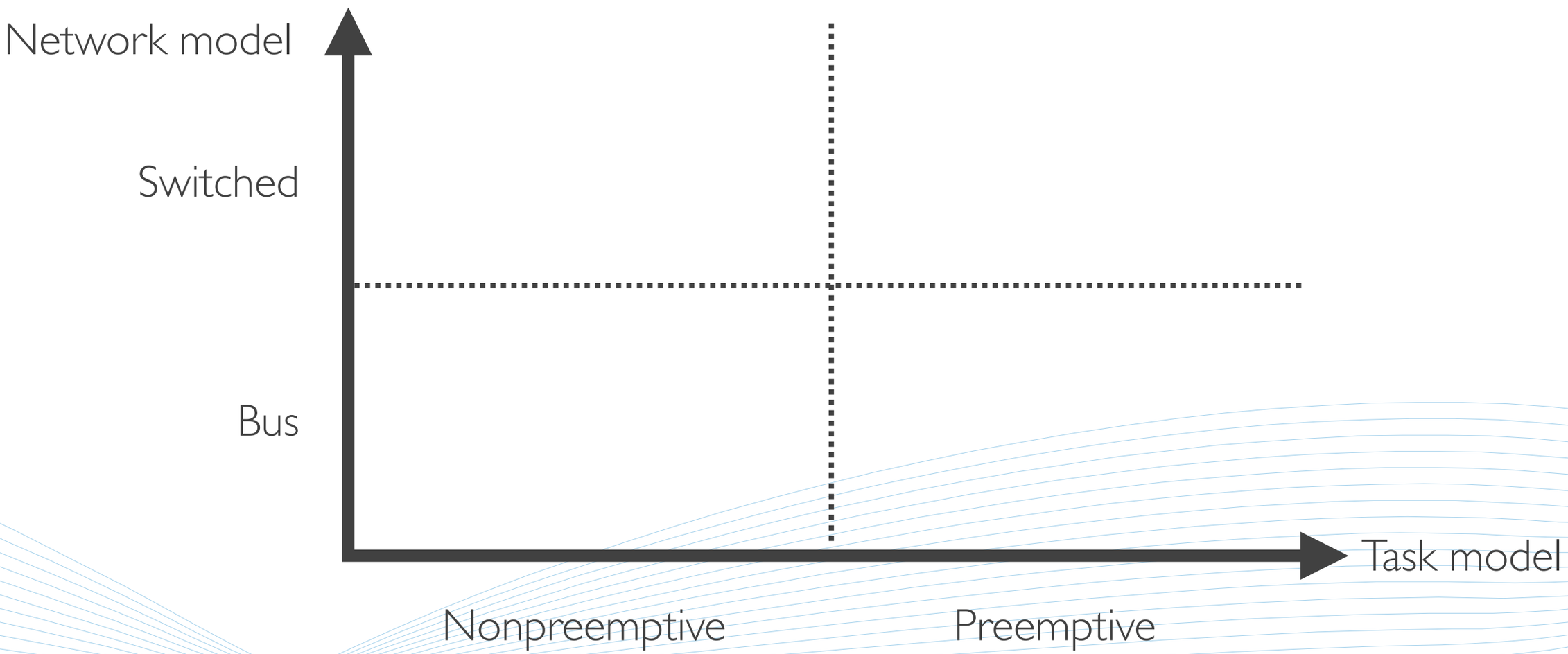


Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

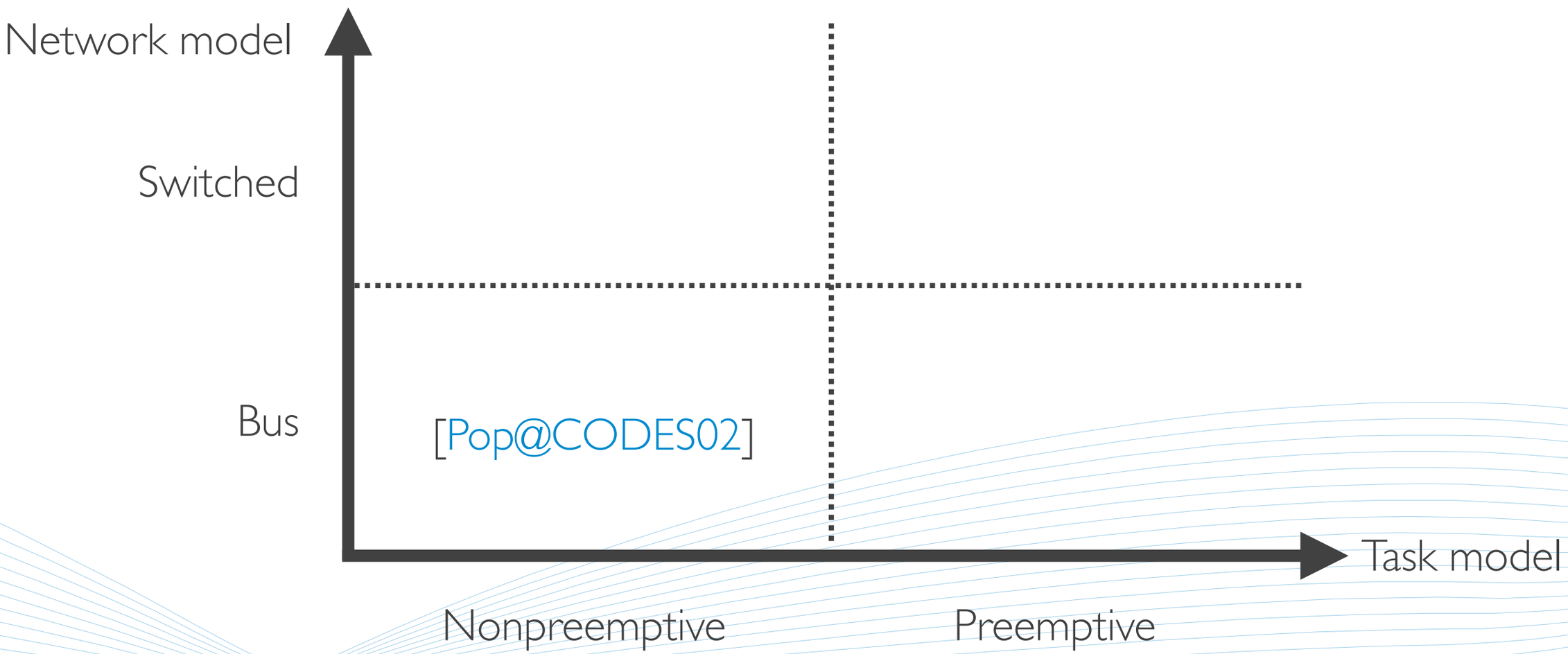


Scheduling

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling



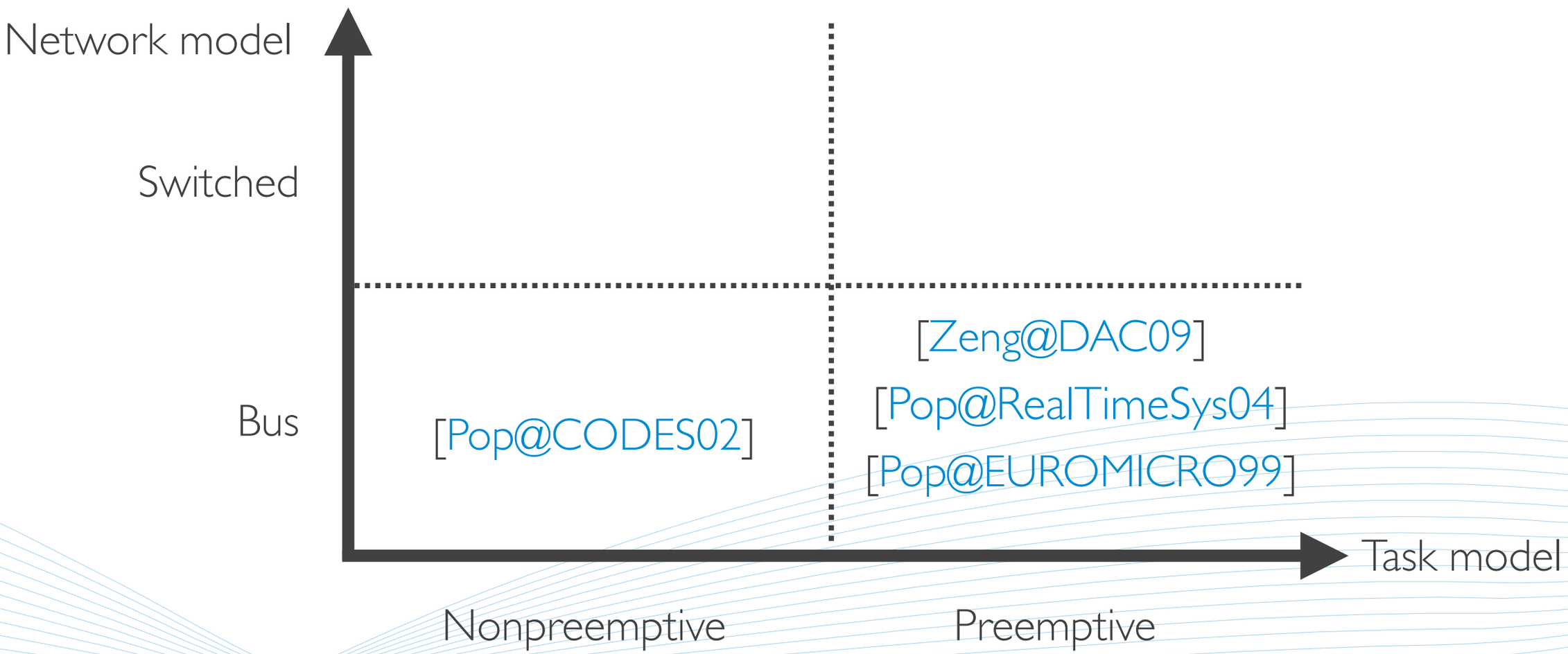
Scheduling

Ensuring Reliable Networks TTTech

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling



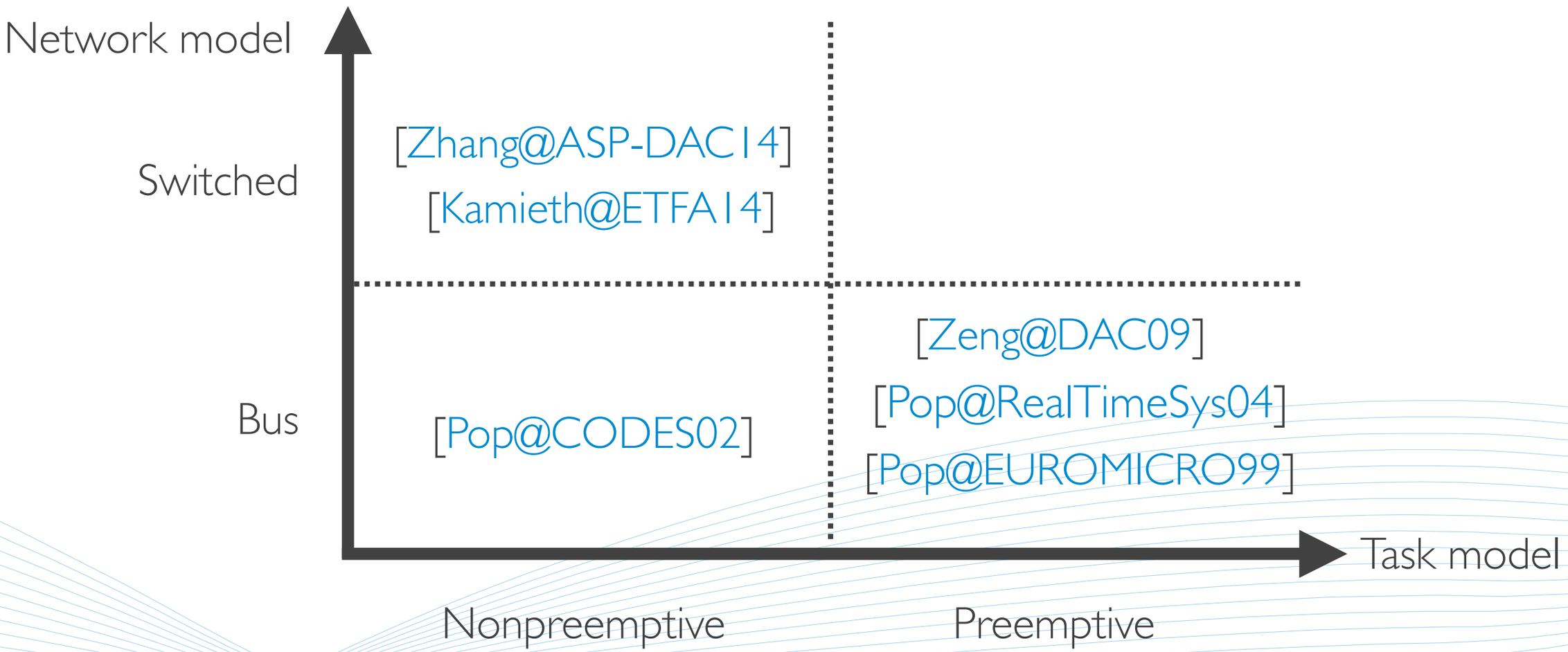
Scheduling

Ensuring Reliable Networks TTTech

Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling

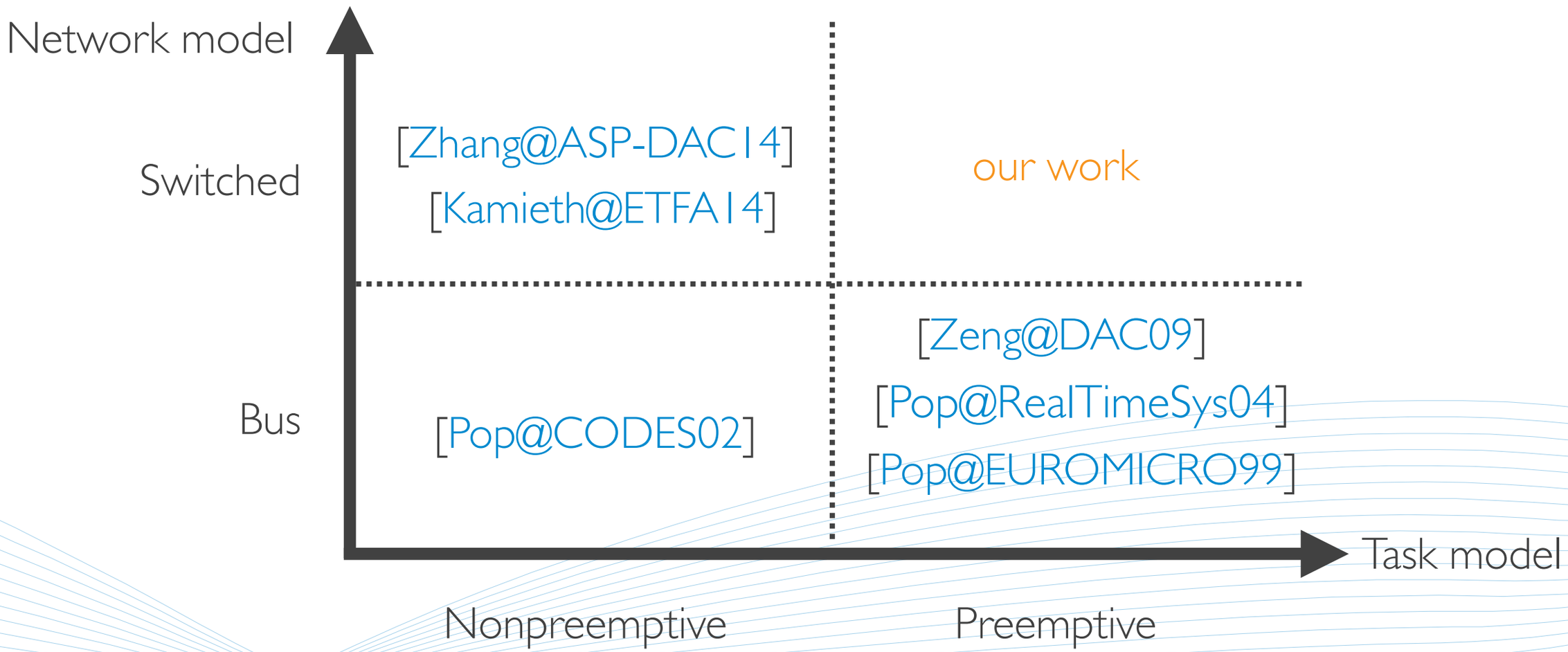


Scheduling

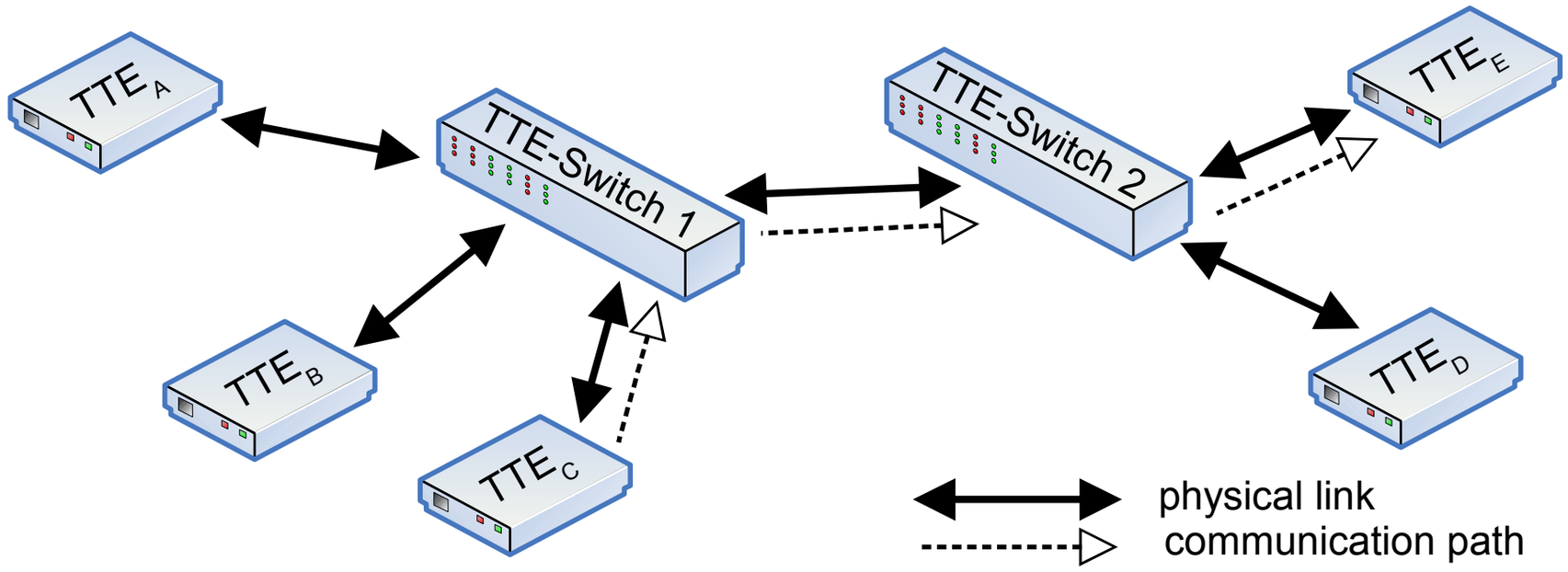
Sequential scheduling

- Network [Steiner@RTSS10] ▷ Tasks [Craciunas@ETFA14]
- Tasks ▷ Network [Hanzalek@ECRTS09]

Combined scheduling



Network model



- multi-hop layer 2 switched network via full-duplex multi-speed links
- virtual links (ARINC 664 P-7)
- TT-traffic class (RC [Tamas-Selicean@CODES+ISSS12], BE)
- synchronised time (< 1 used precision)
- link delay for each link
- memory buffers on switches

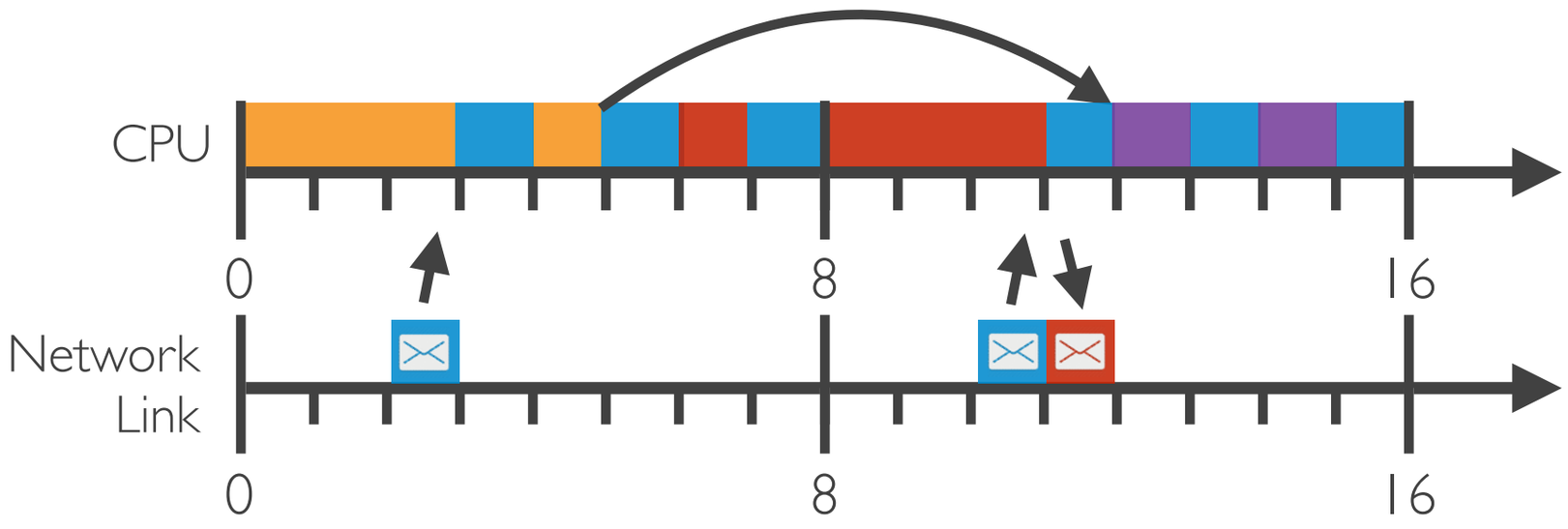
Task model

free (0, 4, 16, 16)

consumer (3, 3, 8, 6)

producer (5, 4, 16, 6)

free (12, 2, 16, 3)



- periodic asynchronous TT-tasks (offset ϕ, wcet C, period T, deadline D)
- static time-driven schedule with preemption
- 3 types of tasks (producer, consumer, free)
- macrotick on ES (usec - ms)
- communication at beginning/end of consumer/producer ([Derler@CIT10])
- end-to-end latency, dependencies between tasks

Networked system model

Ensuring Reliable Networks TTTech

Network $G(\mathcal{V}, \mathcal{L}) \quad \mathcal{L} \subseteq \mathcal{V} \times \mathcal{V}$

$$\forall [v_a, v_b] \in \mathcal{L} \Rightarrow [v_b, v_a] \in \mathcal{L}$$

Network links $[v_a, v_b]$

CPU self-links $[v_a, v_a]$

(speed, link delay, macrotick, memory buffer)

Virtual link - dataflow from one producer to one receiver

$$vl_i = [[v_a, v_a], [v_a, v_1], [v_1, v_2], \ldots, [v_{n-1}, v_n], [v_n, v_b], [v_b, v_b]].$$

Frames

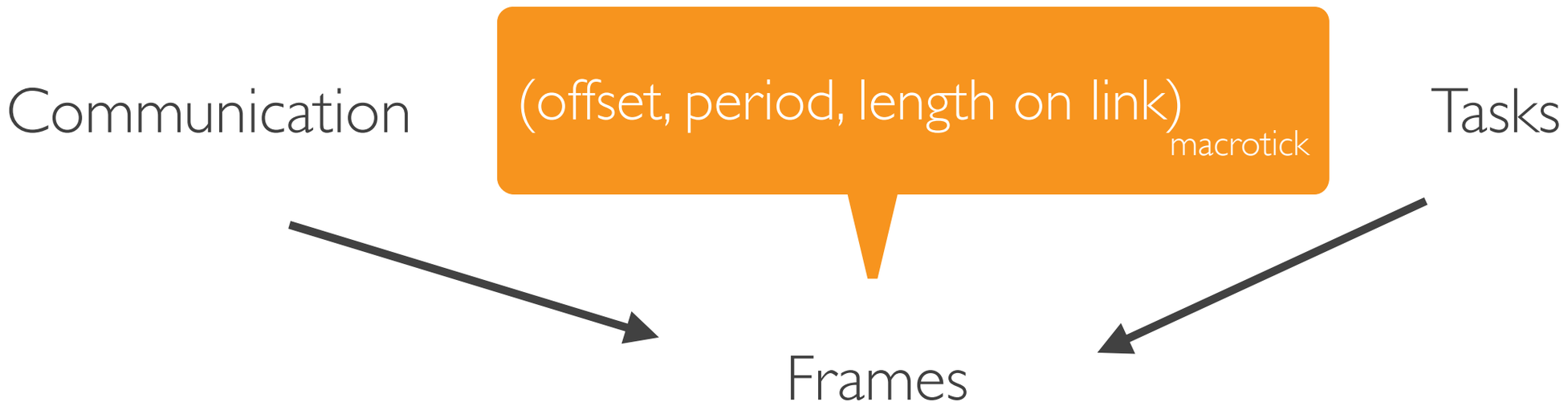
Frames

Ensuring Reliable Networks TTTech

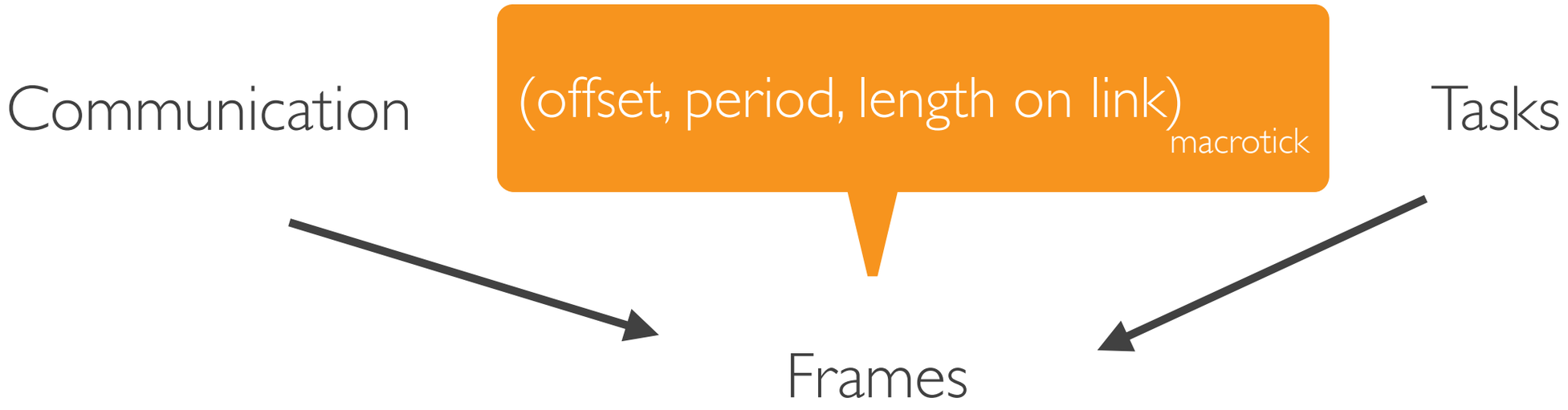
Communication

Tasks

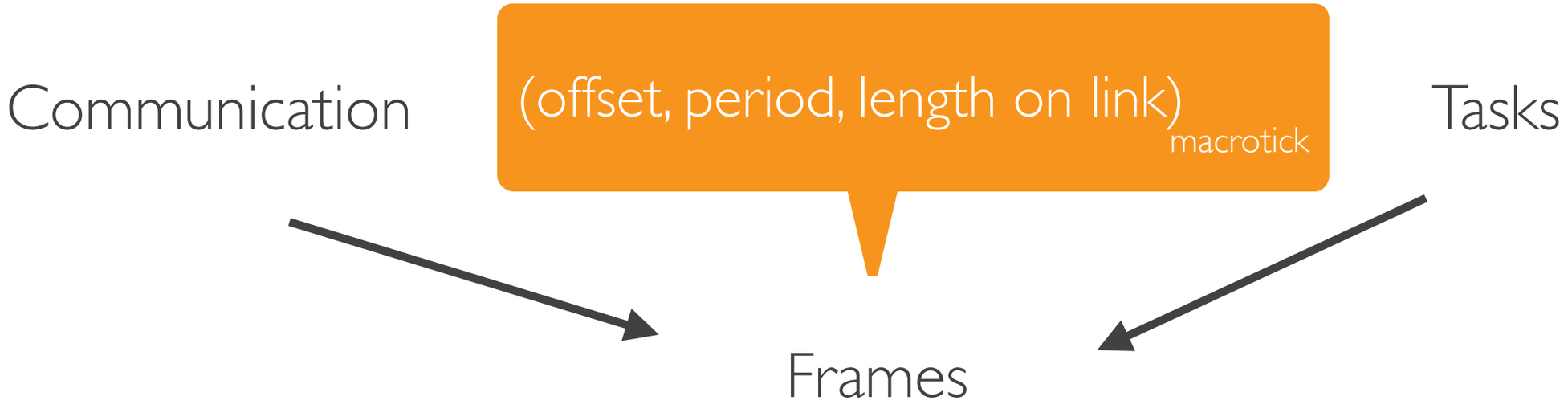
Frames

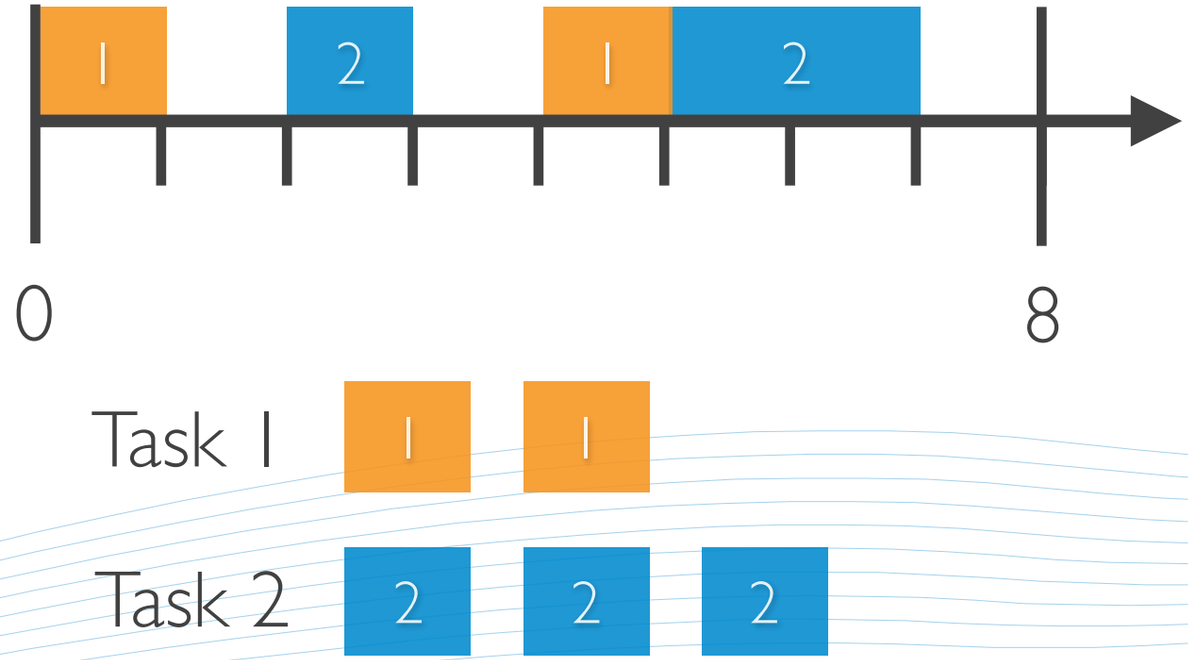


Frames



Frames





Scheduling problem

Scheduling problem

find offsets for the frames (on links and virtual task frames)

Scheduling problem

find offsets for the frames (on links and virtual task frames)

reduces to finding a solution for a set of constraints

- frame constraints
- link constraints
- virtual link constraints
- memory constraints
- end-to-end latency constraints
- precedence constraints

Scheduling problem

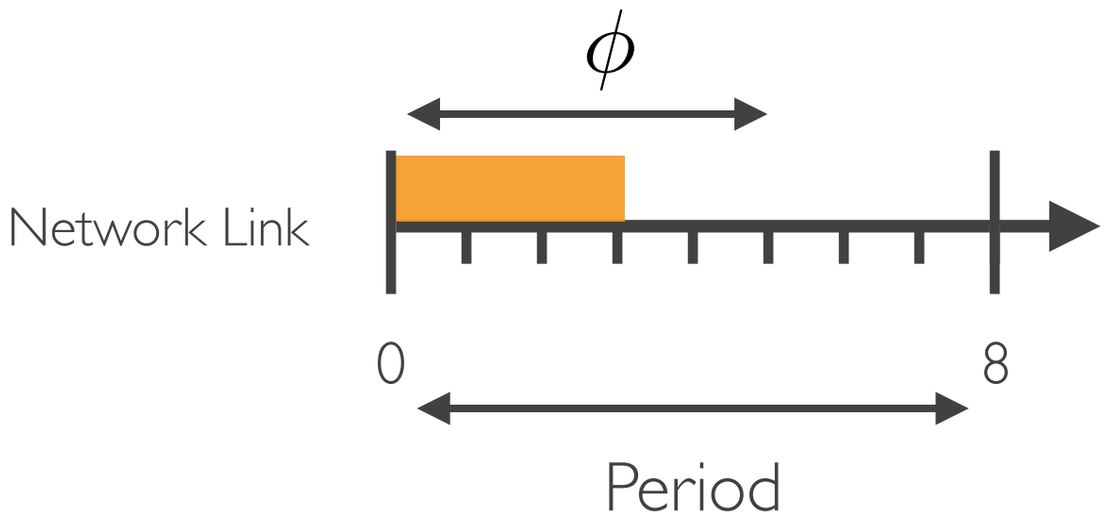
find offsets for the frames (on links and virtual task frames)

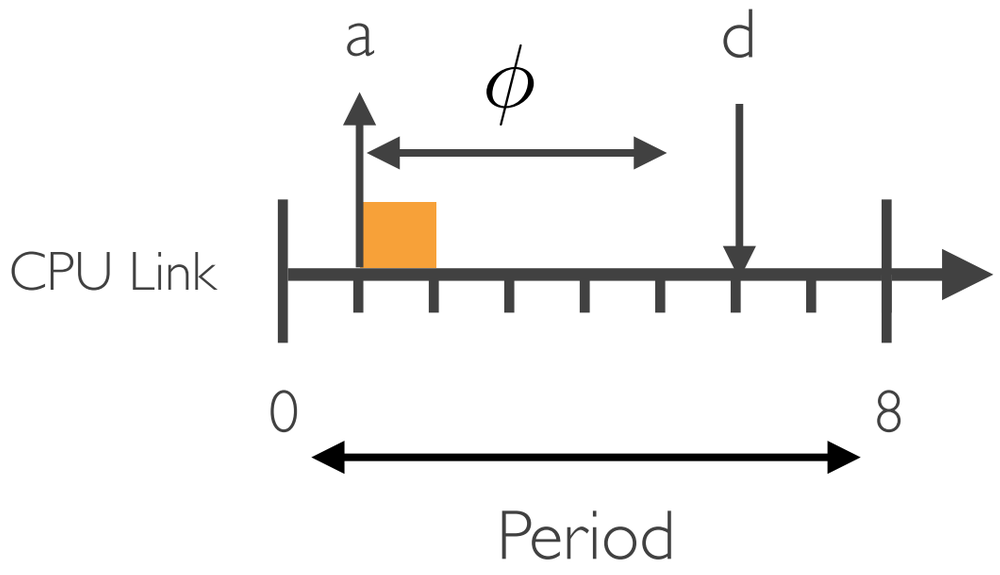
reduces to finding a solution for a set of constraints

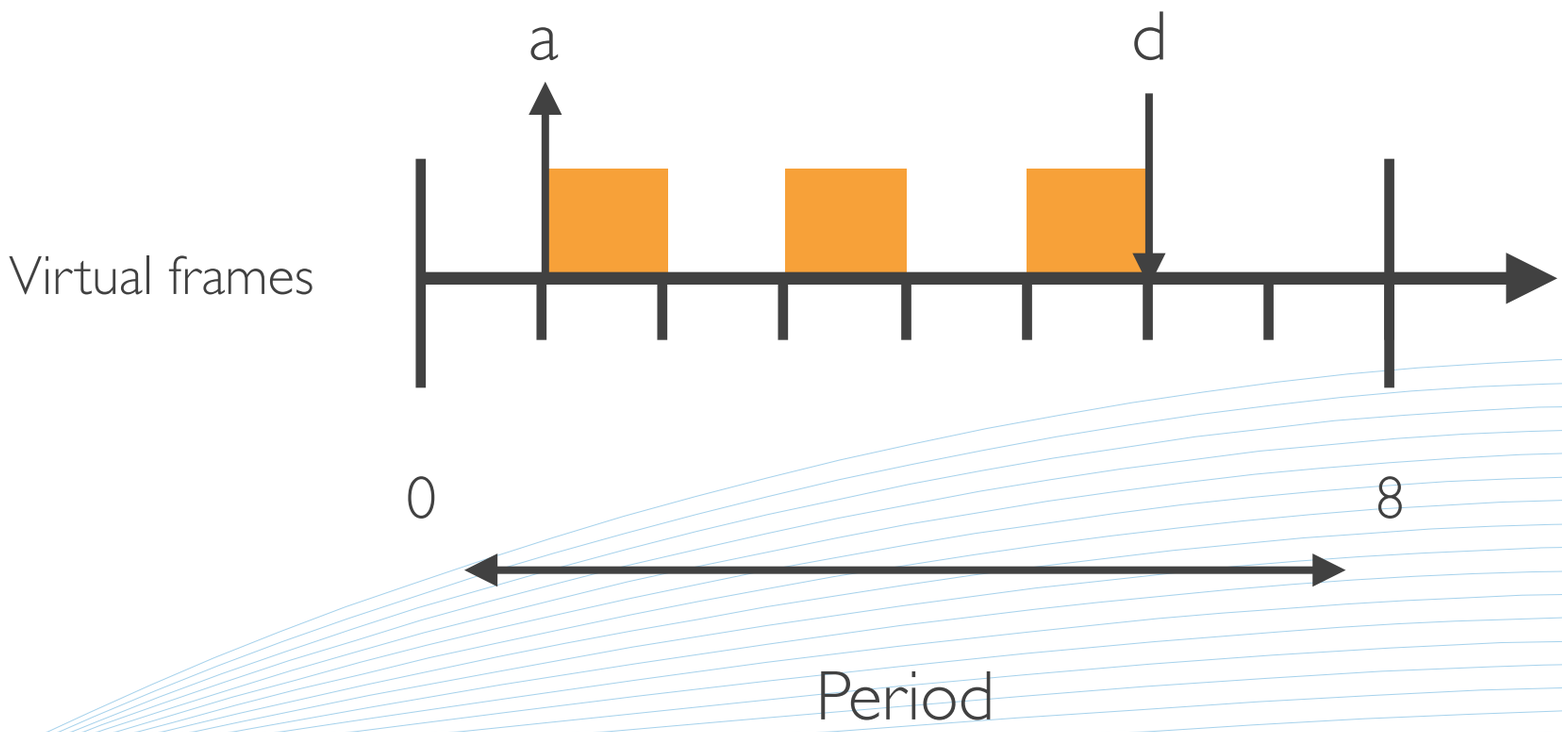
- frame constraints
- link constraints
- virtual link constraints
- memory constraints
- end-to-end latency constraints
- precedence constraints

NP-complete

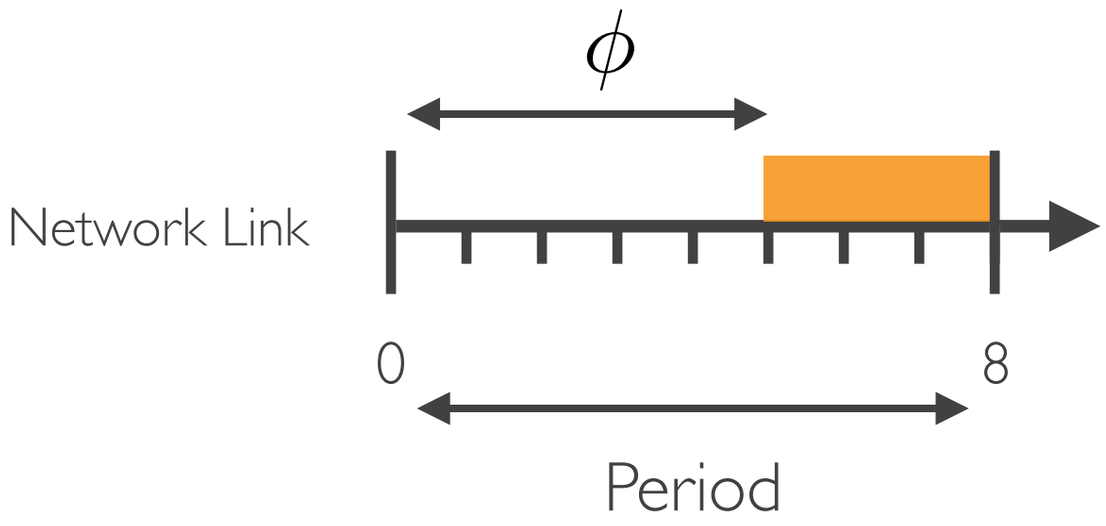
Frame constraints

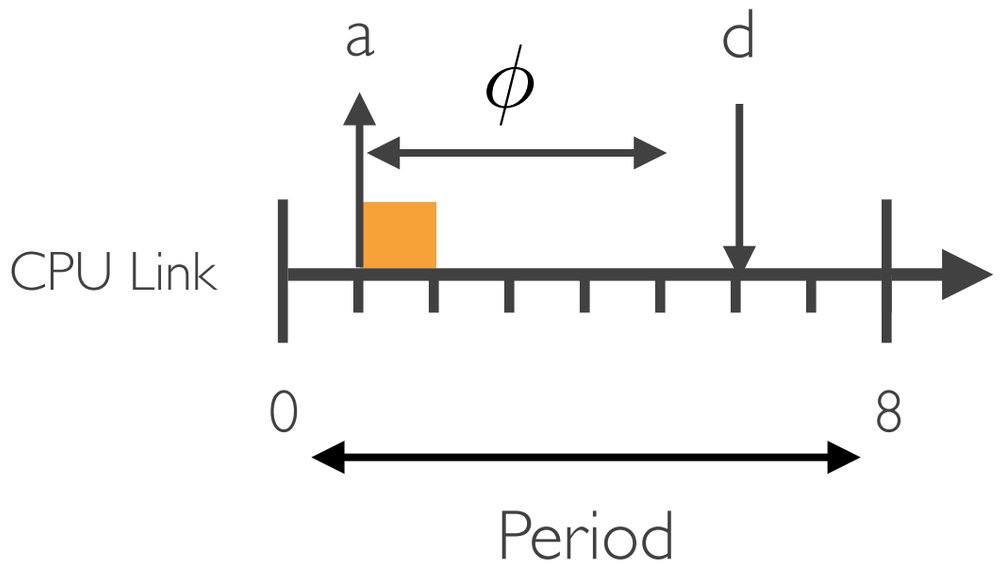


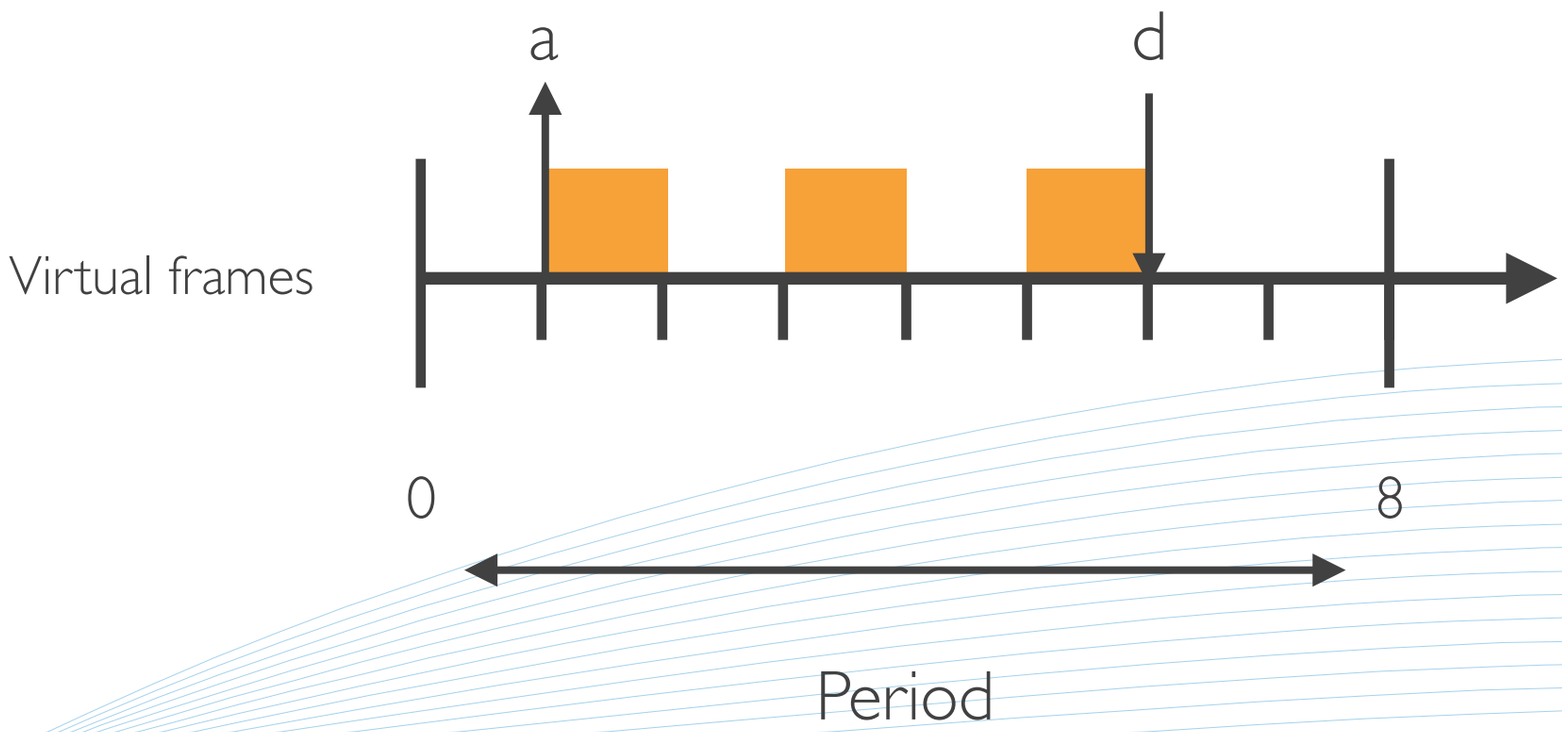




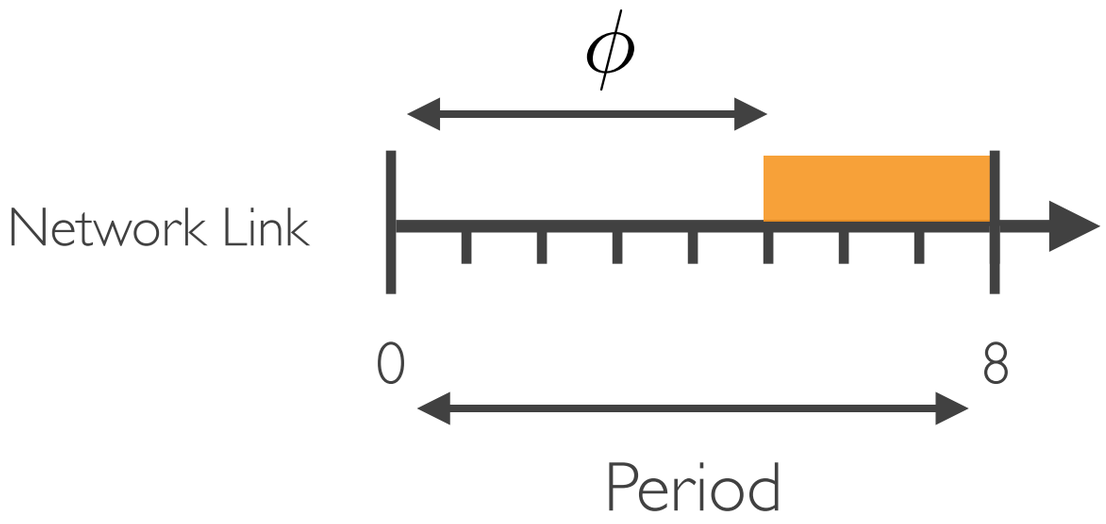
Frame constraints

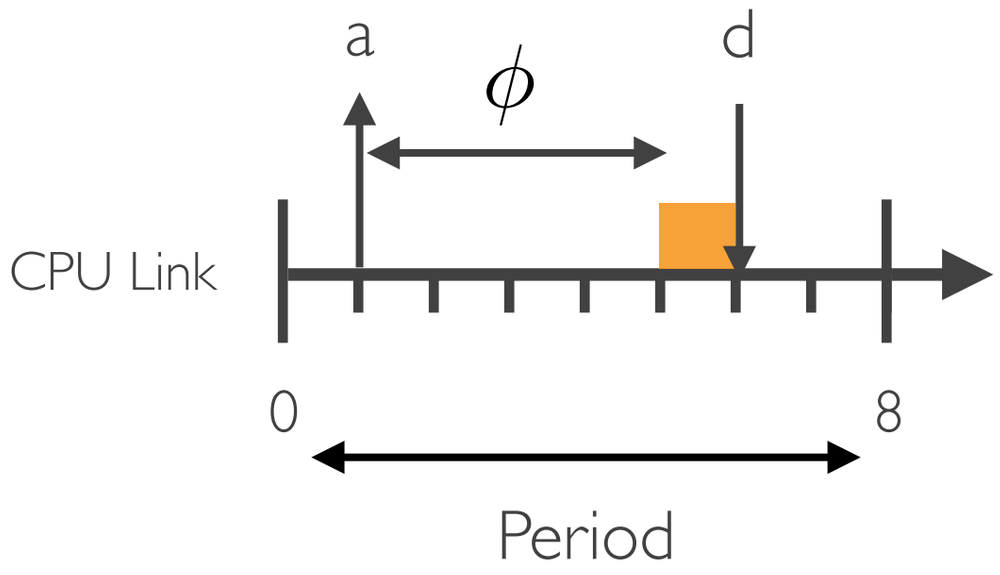


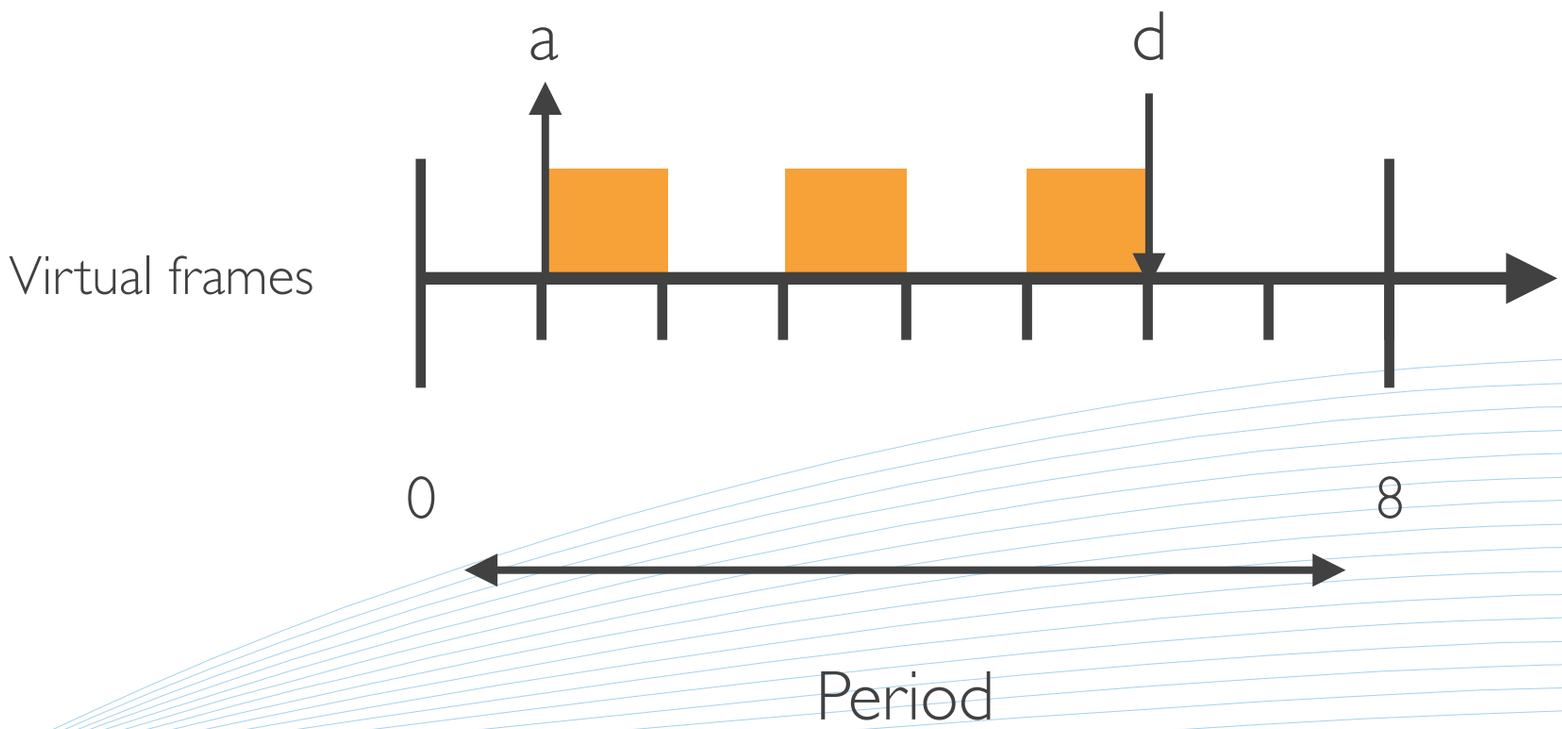




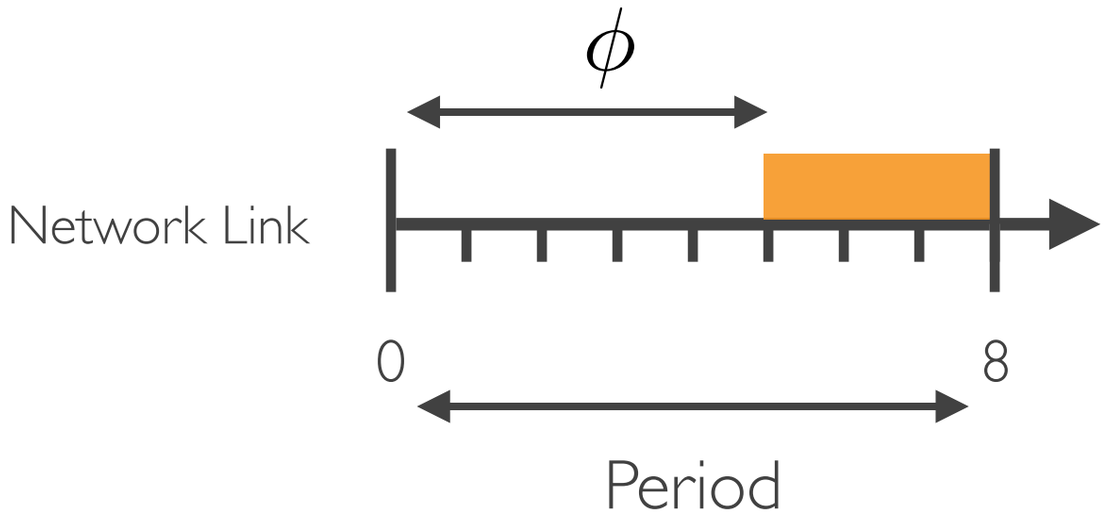
Frame constraints

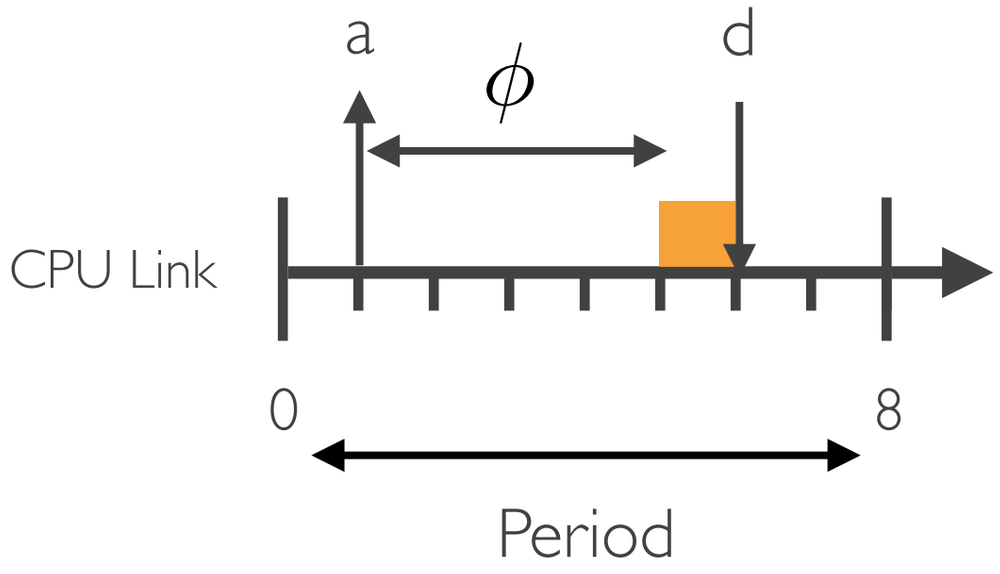


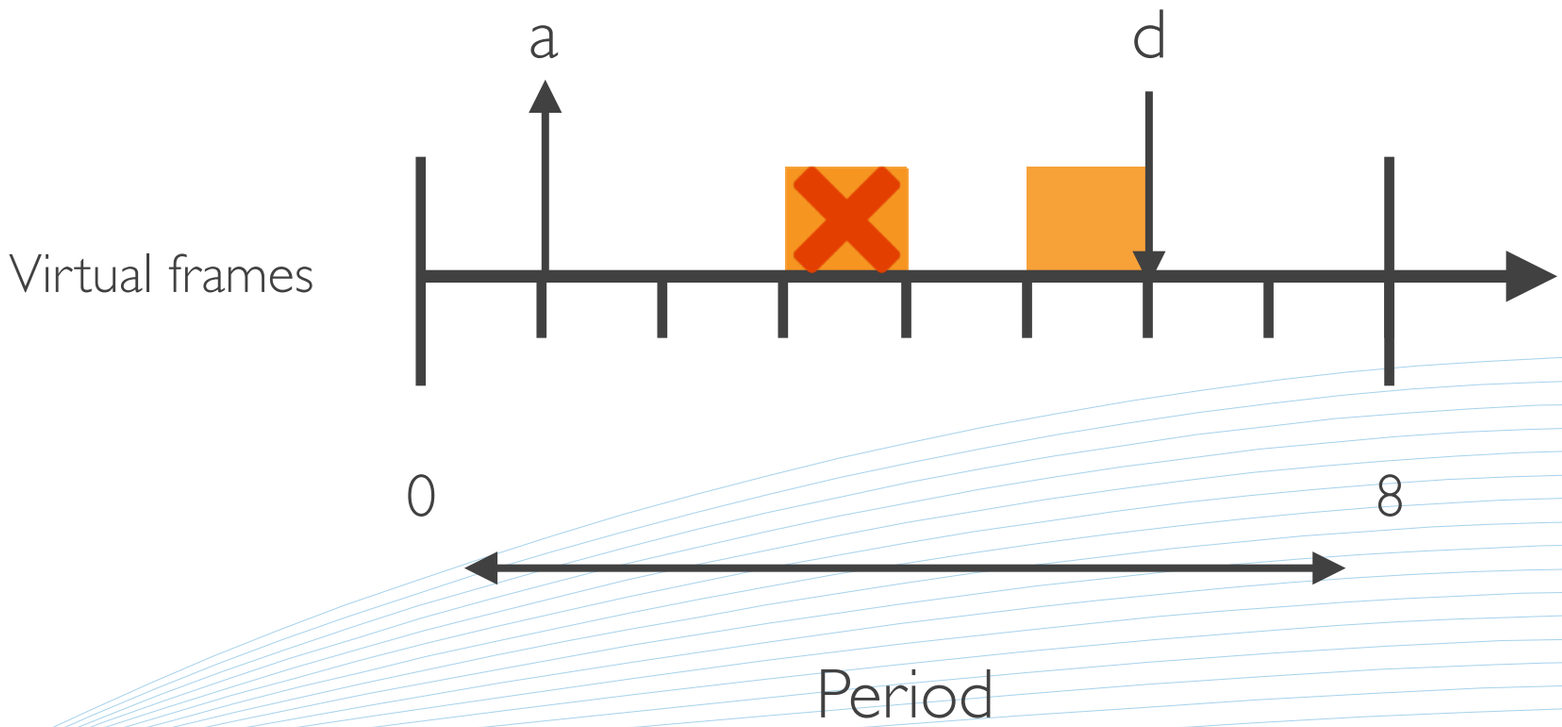




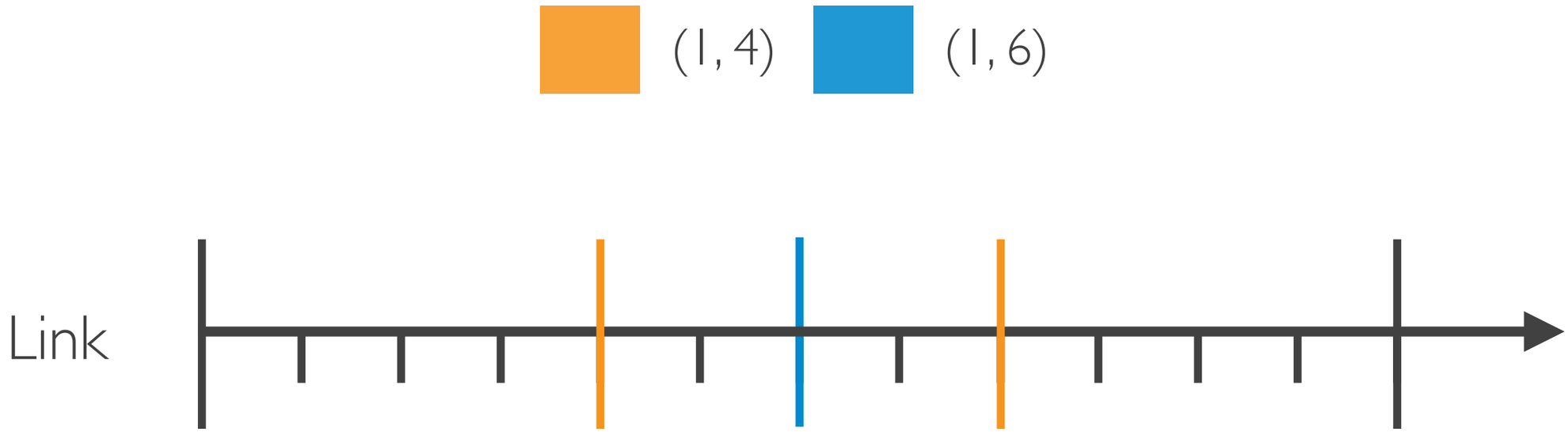
Frame constraints



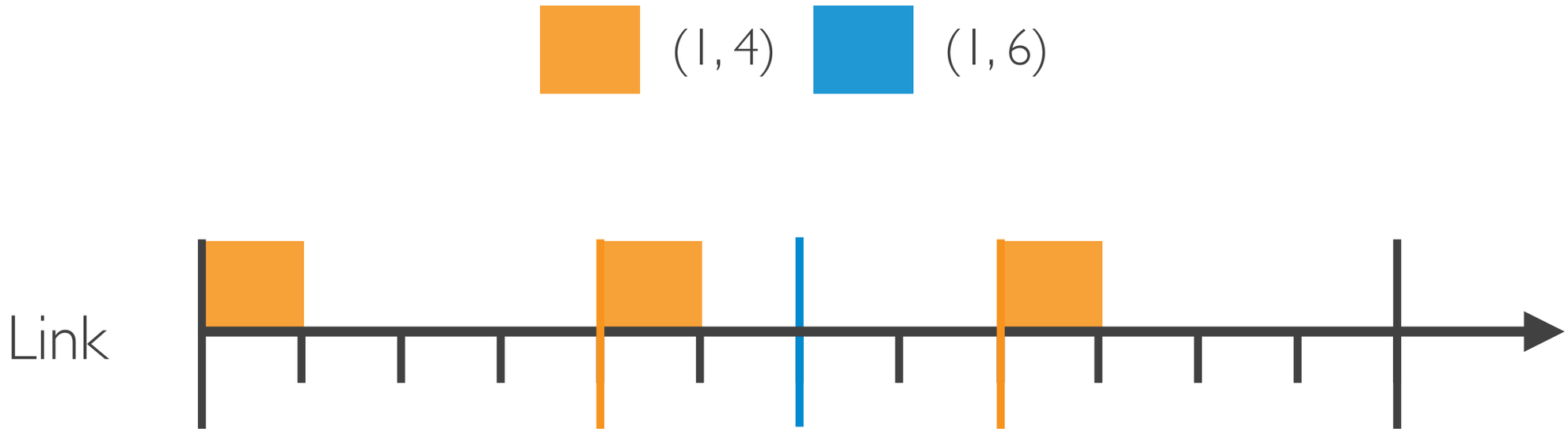




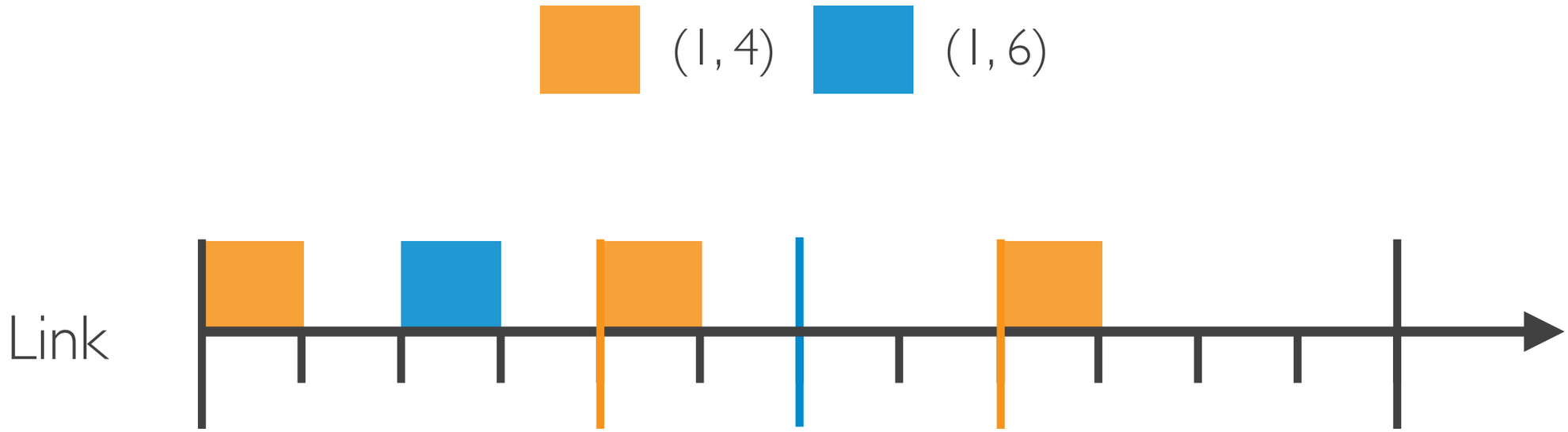
Link constraints



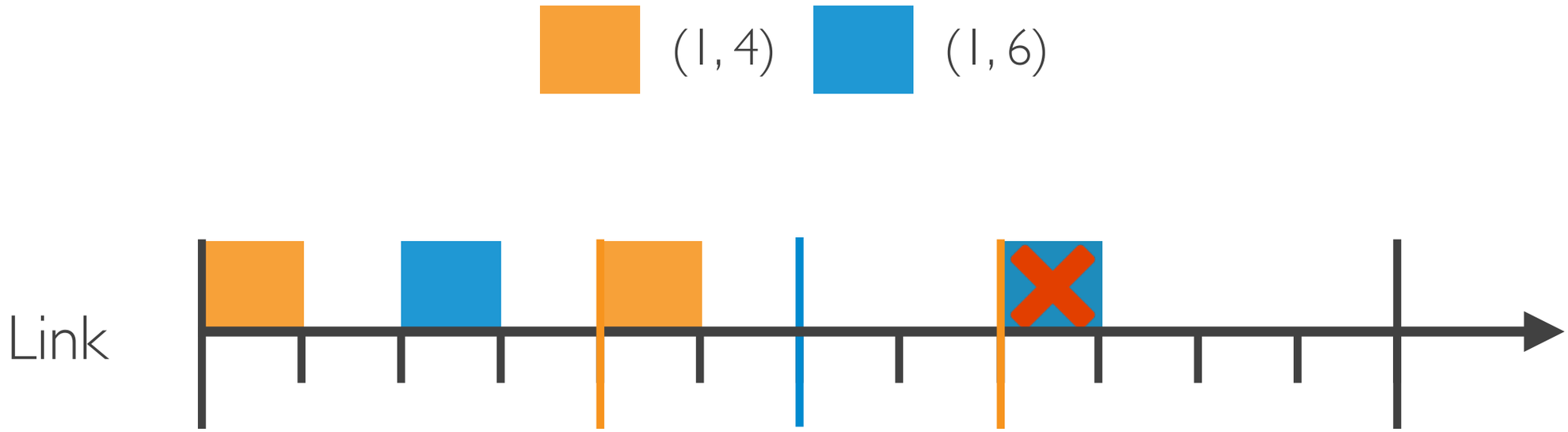
Link constraints



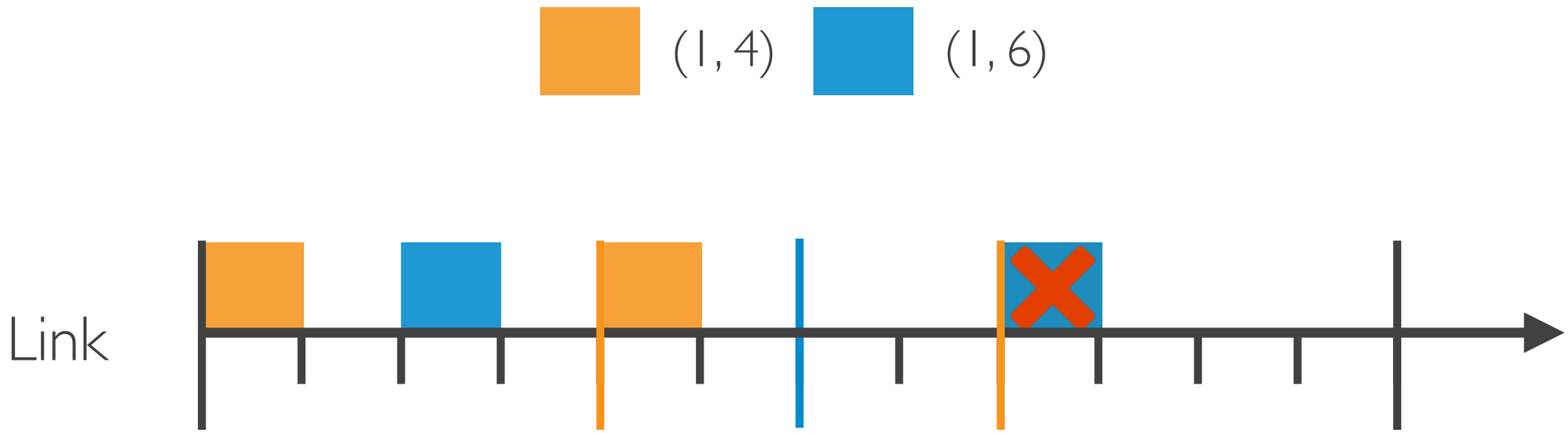
Link constraints



Link constraints



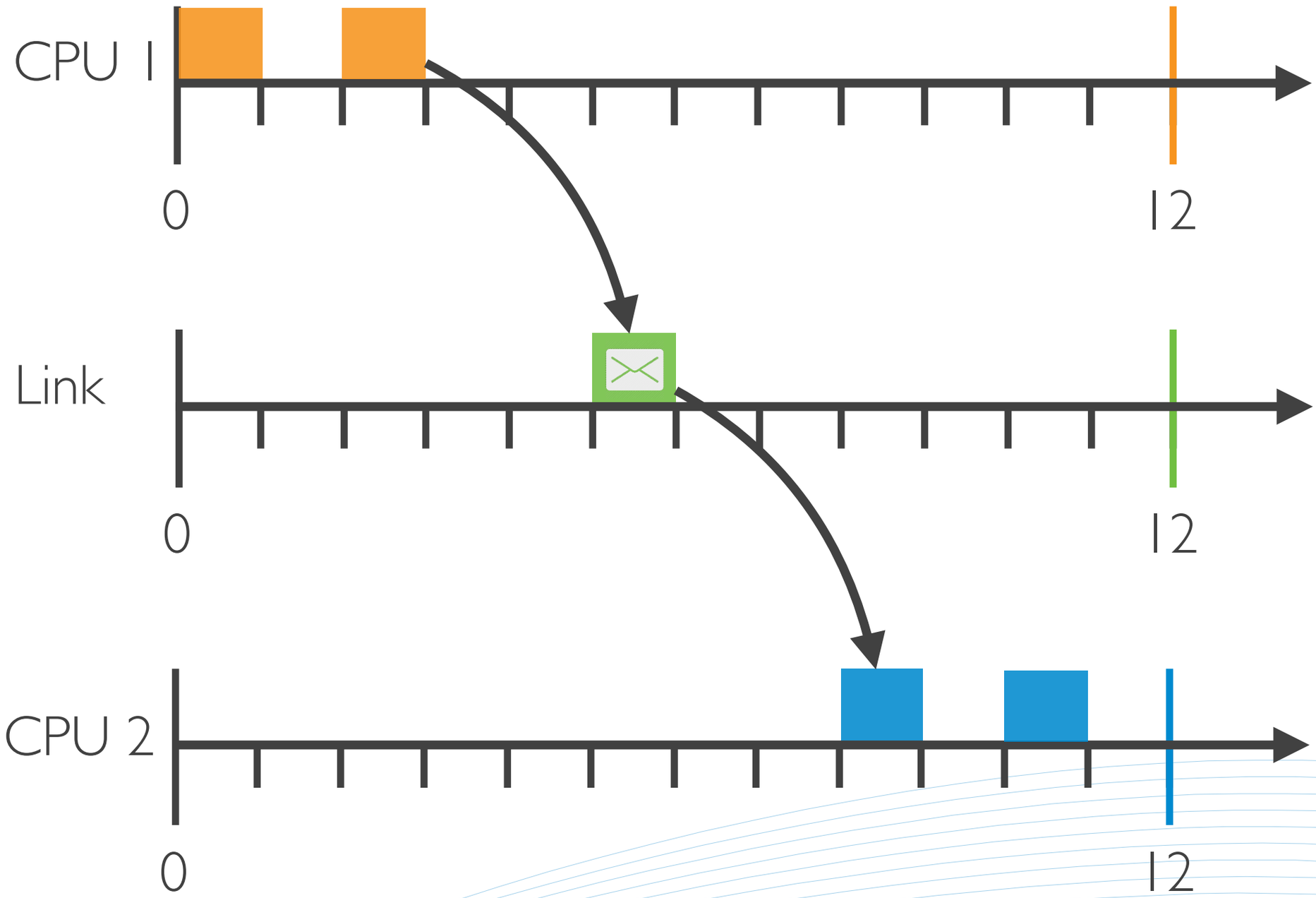
Link constraints



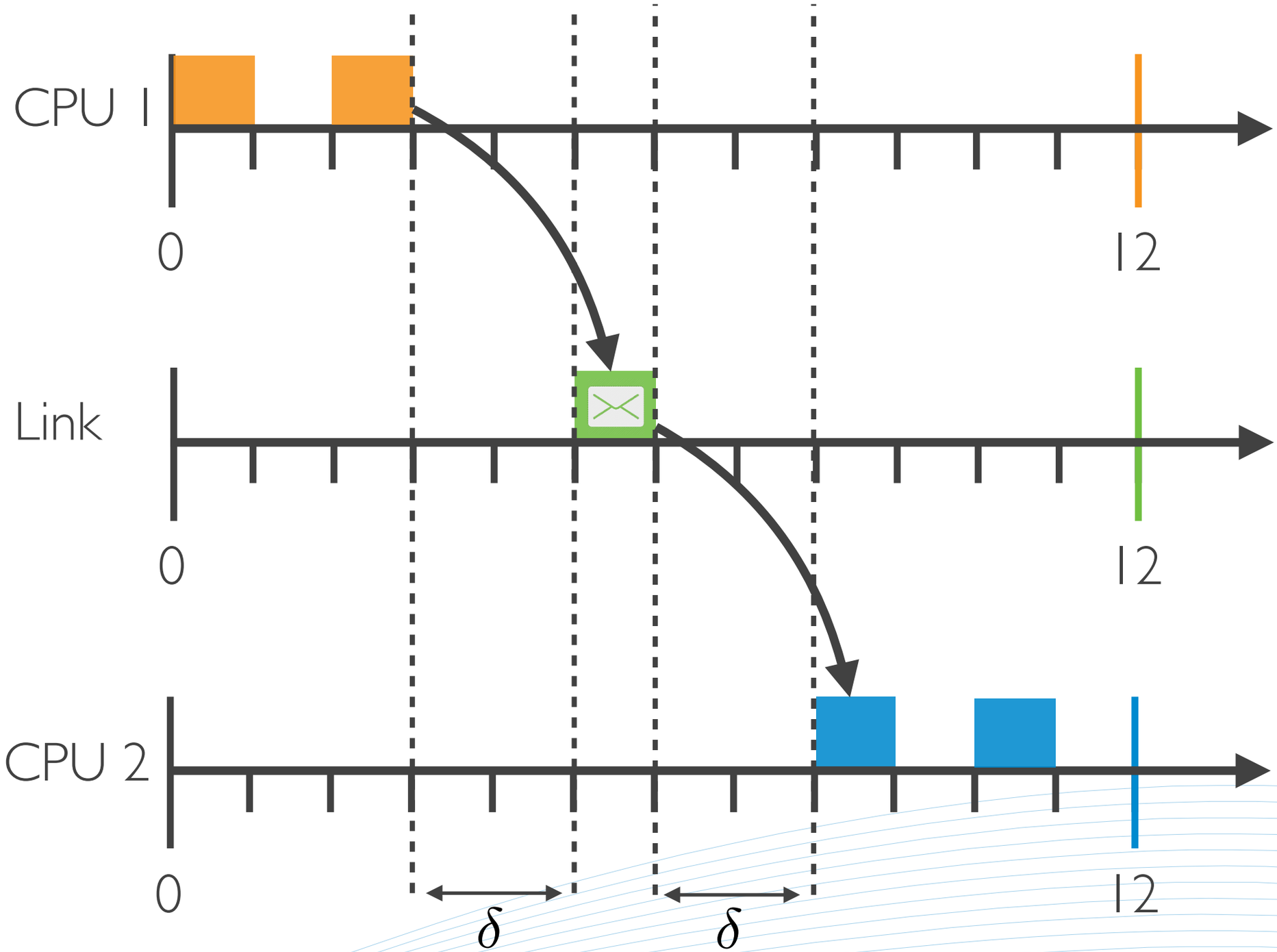
no two frames scheduled on the same link may overlap

Virtual link constraints

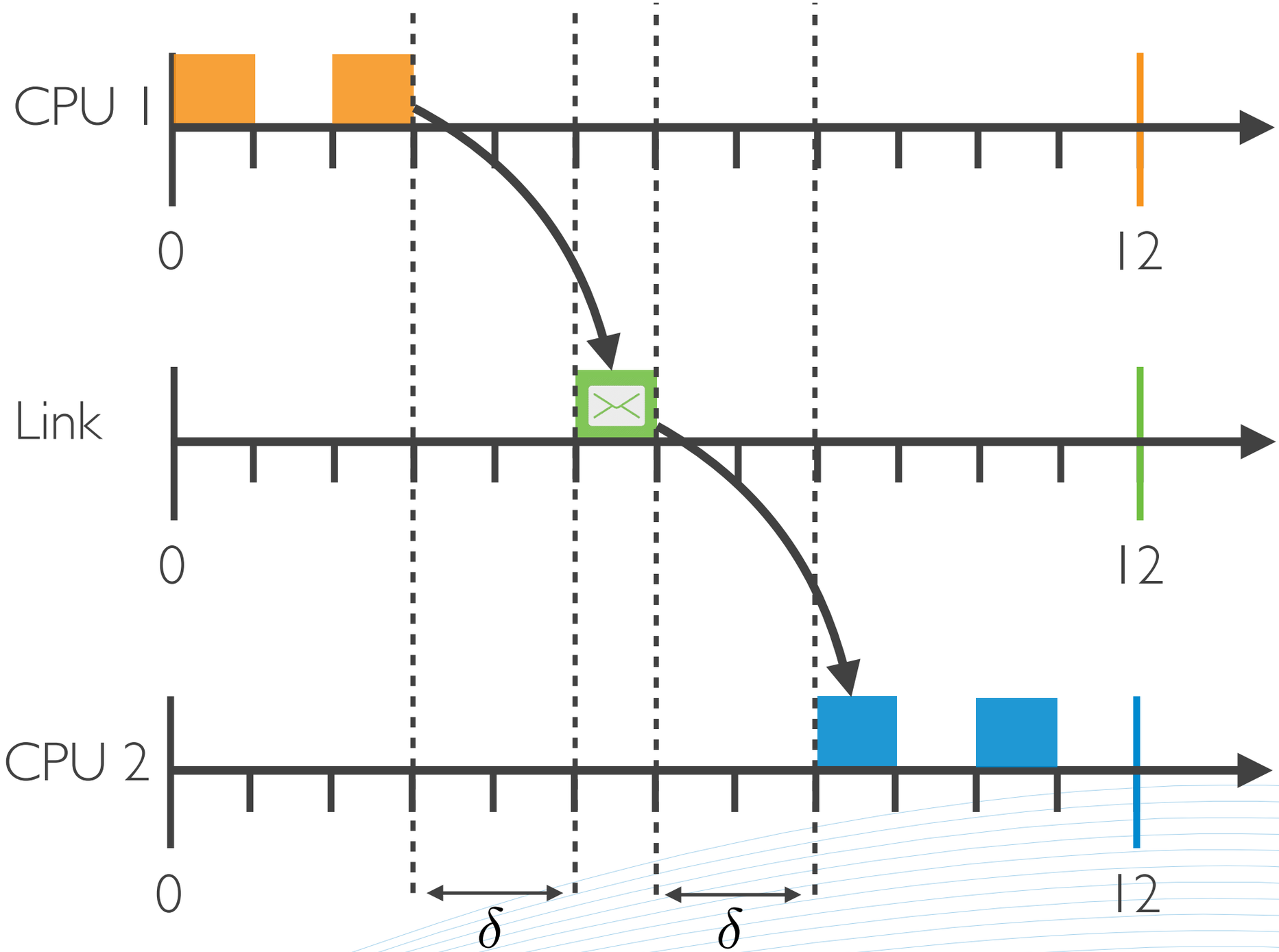
Ensuring Reliable Networks TTTech



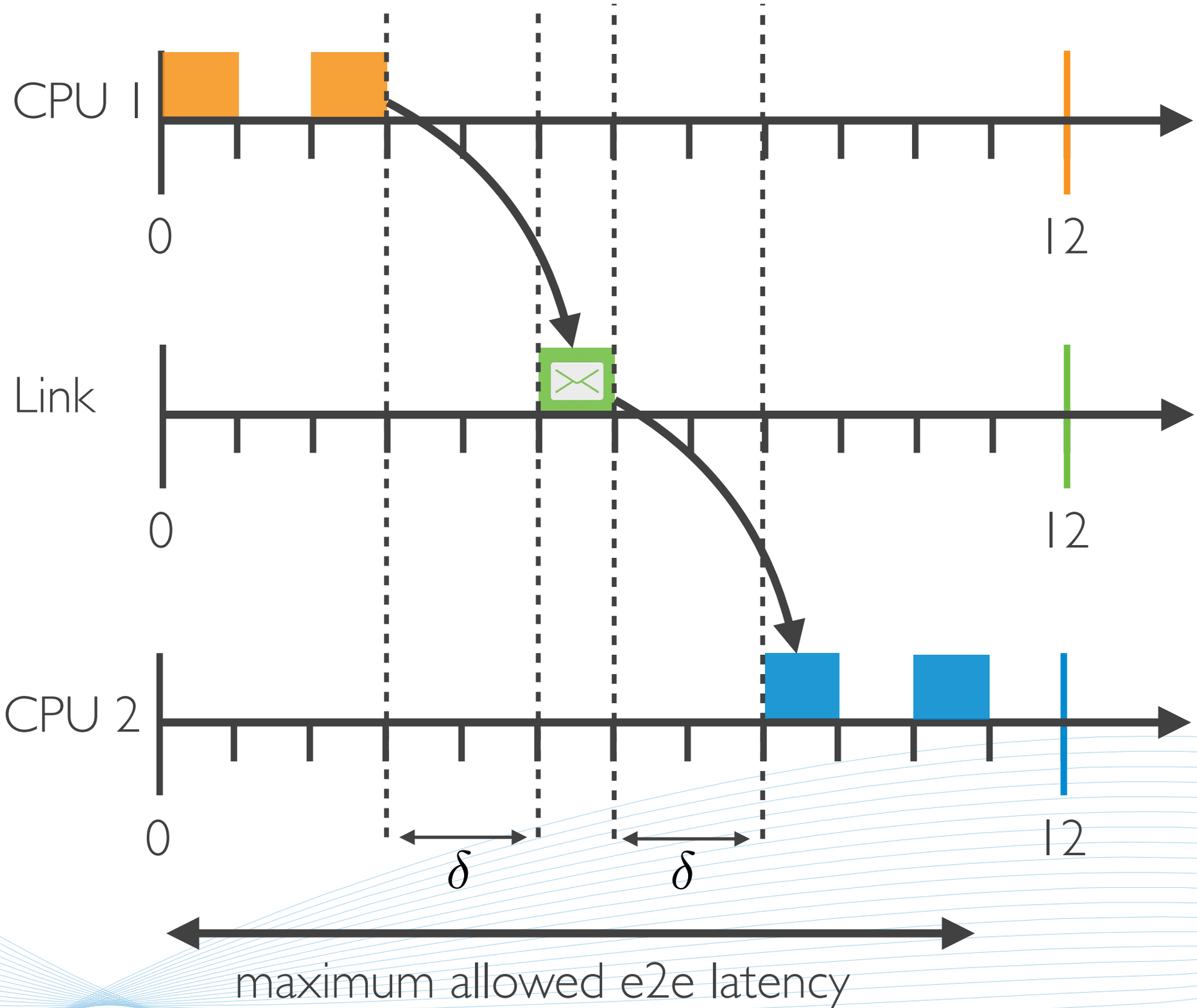
Virtual link constraints



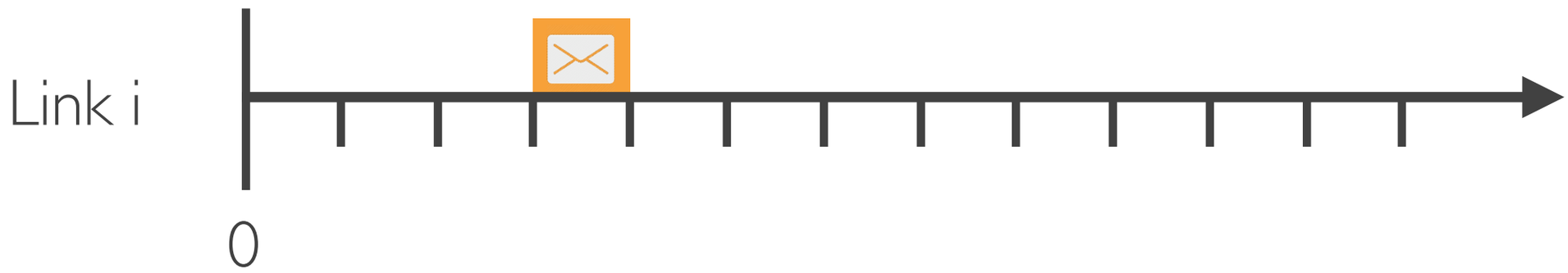
E2E latency constraints

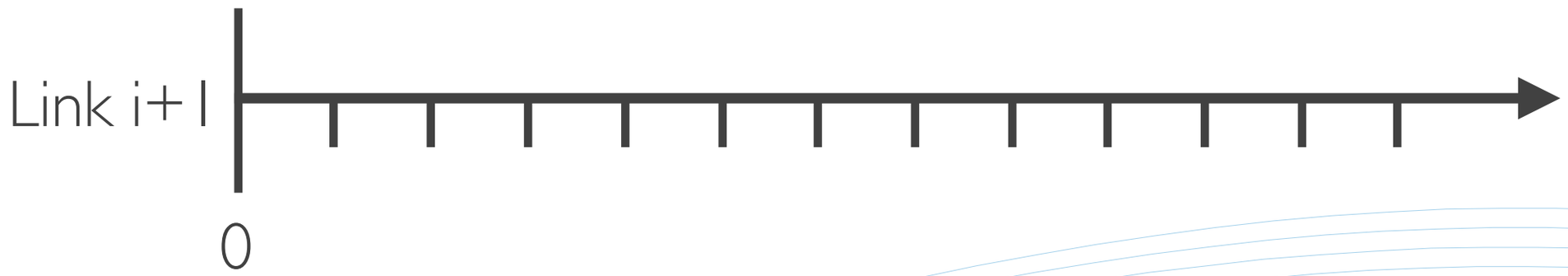


E2E latency constraints

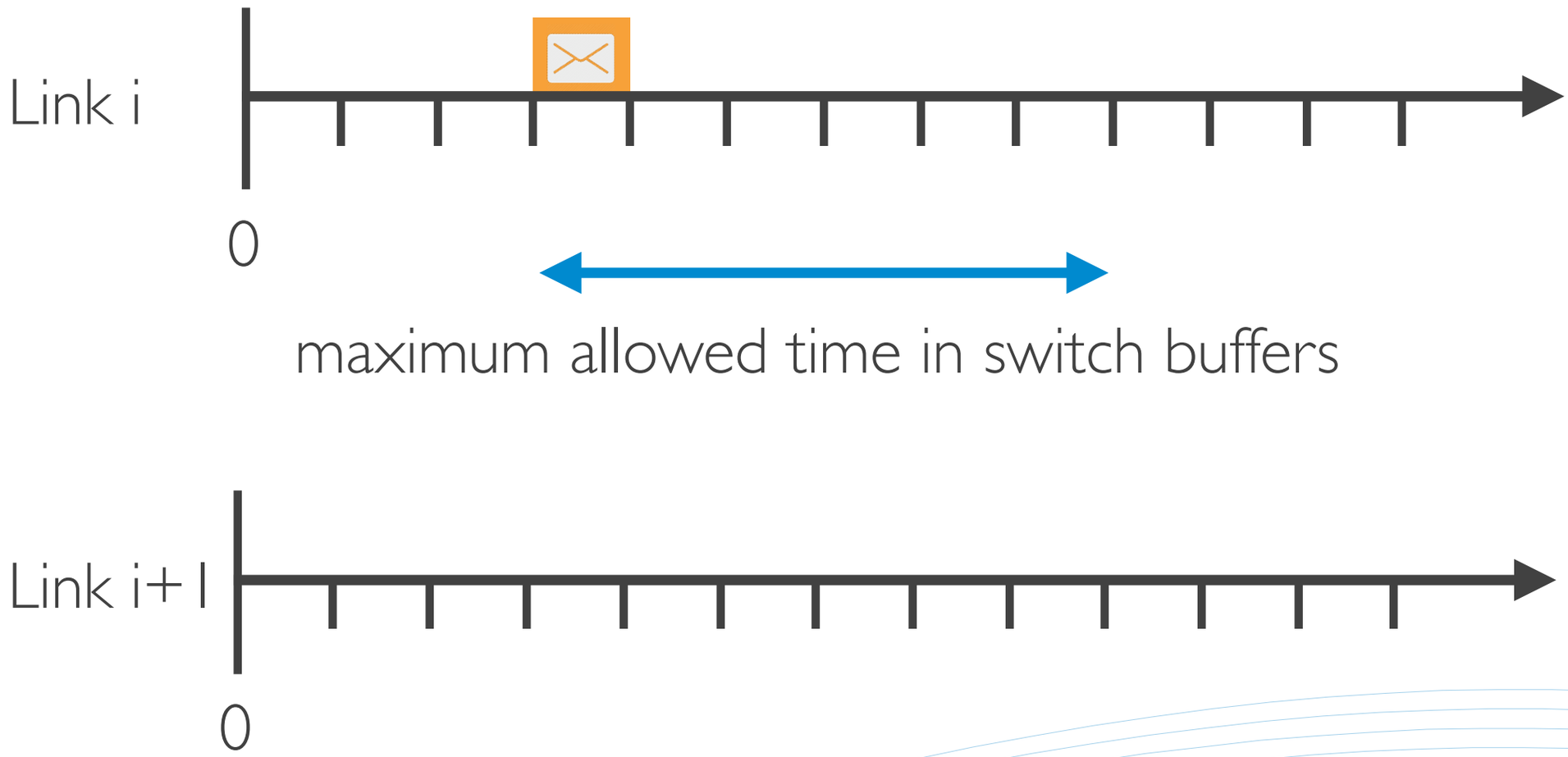


Memory constraints

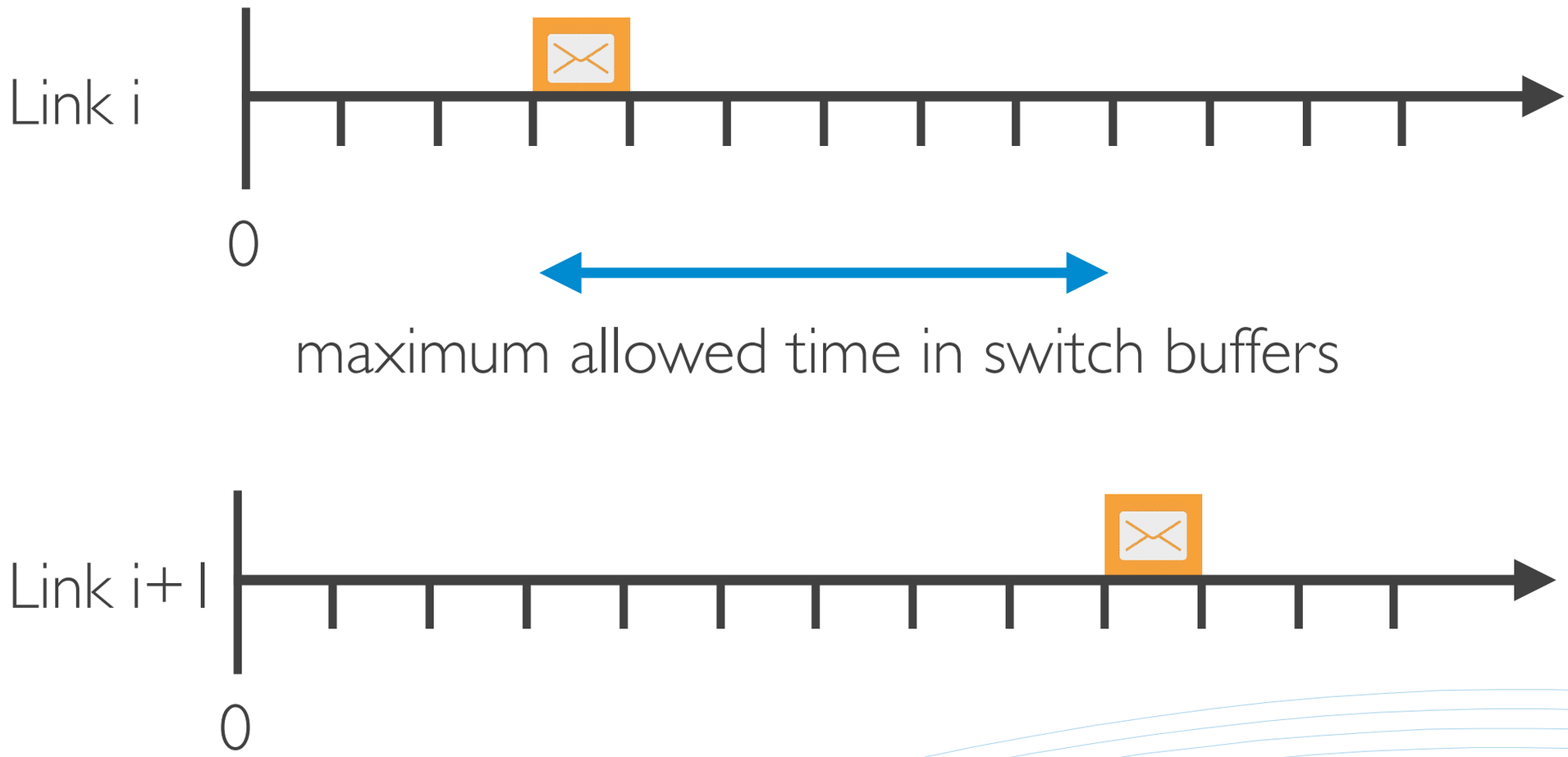




Memory constraints



Memory constraints



Satisfiability Modulo Theories

Ensuring Reliable Networks 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 $\exists, \forall$

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 $\exists, \forall$

A lot of solvers and a very active community

OpenSMT [Bruttomesso@TACAS10] Yices [Dutertre@CAV14]

CVC4 [Barrett@CAV11] Z3 [de Moura@TACAS08]

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 $\exists, \forall$

A lot of solvers and a very active community

OpenSMT [Bruttomesso@TACAS10] Yices [Dutertre@CAV14]

CVC4 [Barrett@CAV11] Z3 [de Moura@TACAS08]

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 ~~$\exists, \forall$~~

A lot of solvers and a very active community

OpenSMT [Bruttomesso@TACAS10] Yices [Dutertre@CAV14]

CVC4 [Barrett@CAV11] Z3 [de Moura@TACAS08]

Satisfiability Modulo Theories

Ensuring Reliable Networks TTTech

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 ~~$\exists, \forall$~~

A lot of solvers and a very active community

OpenSMT [Bruttomesso@TACAS10]

Yices [Dutertre@CAV14]

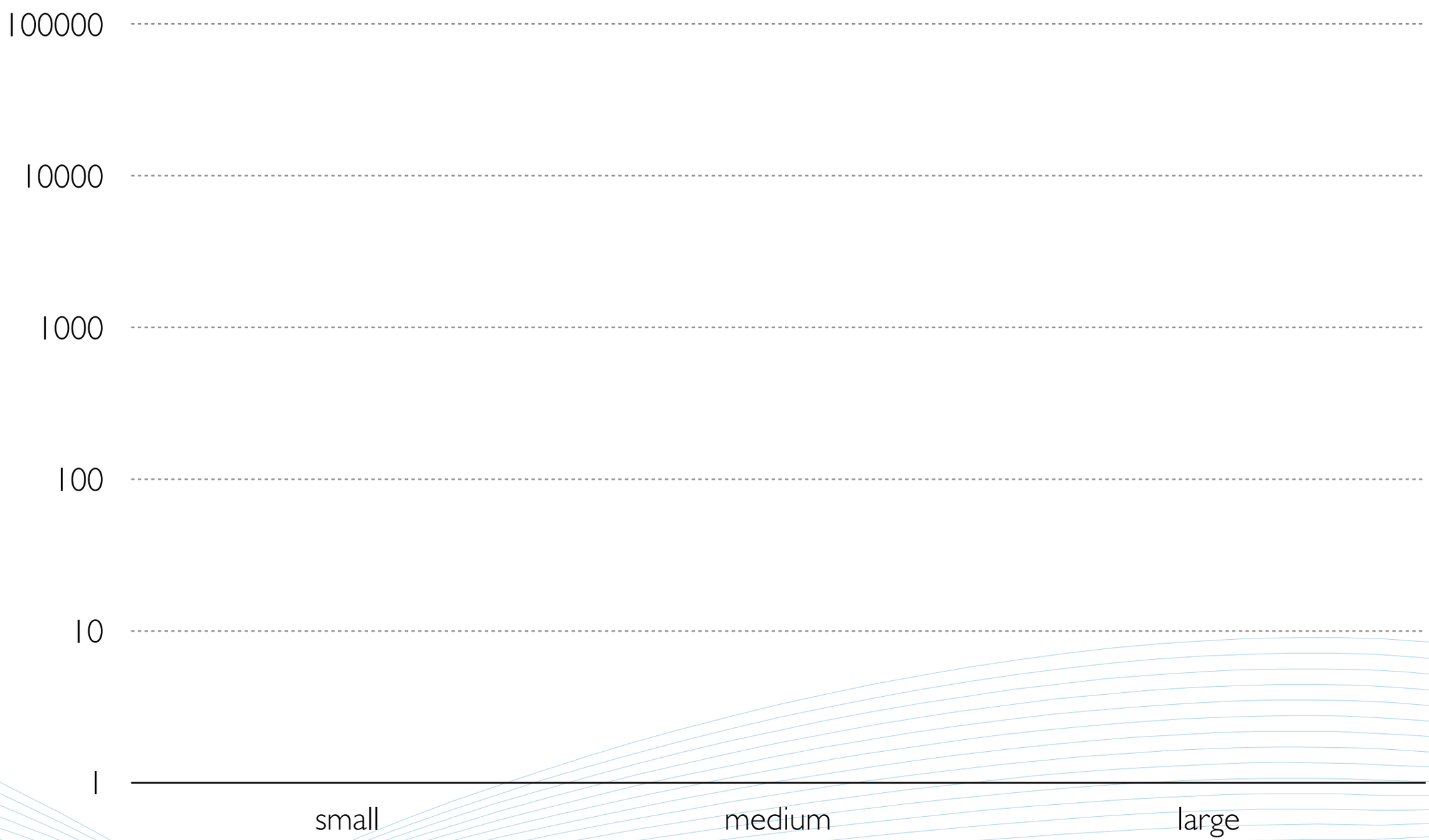
CVC4 [Barrett@CAV11]

Z3 [de Moura@TACAS08]

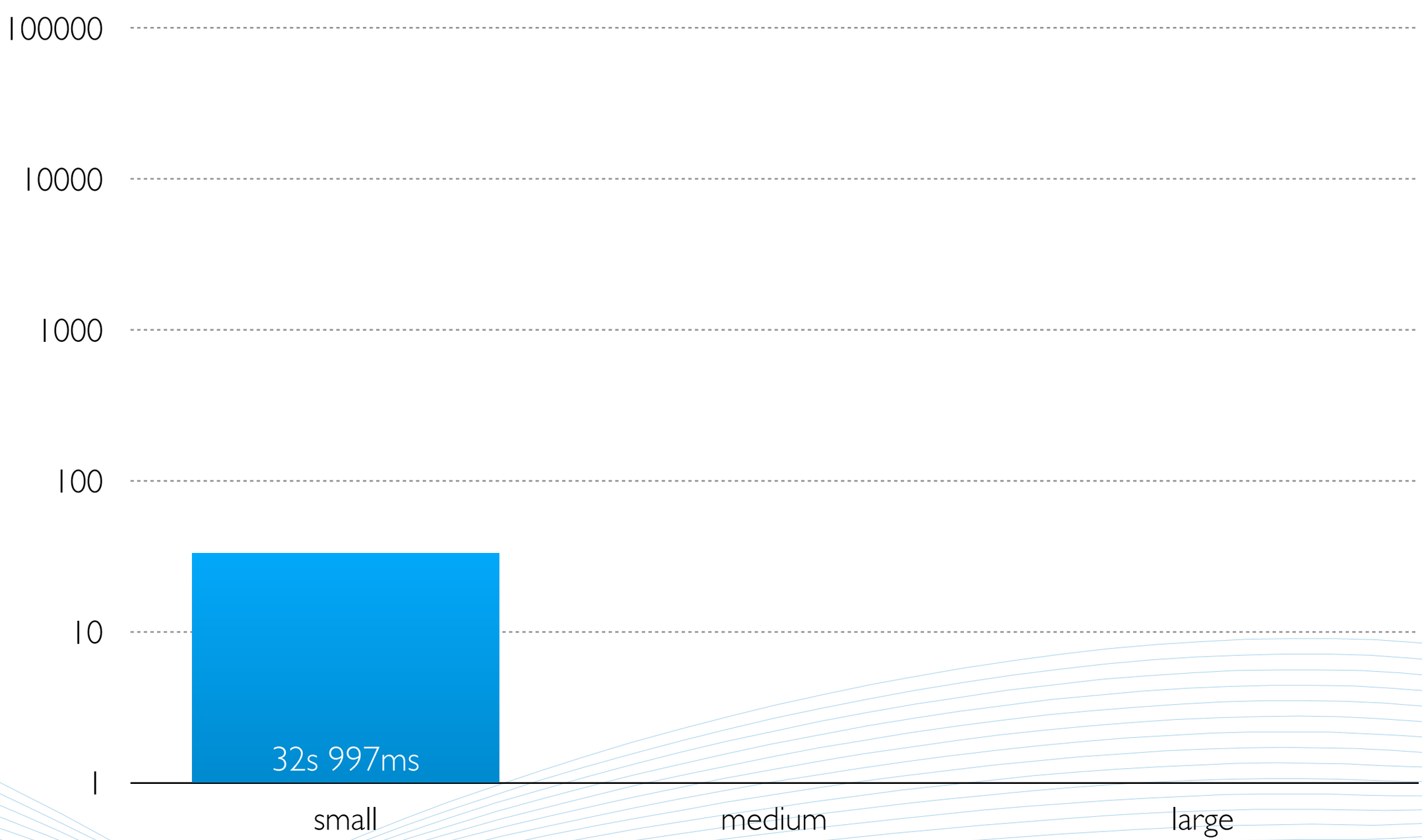
One-shot

Ensuring Reliable Networks TTTech

One-shot

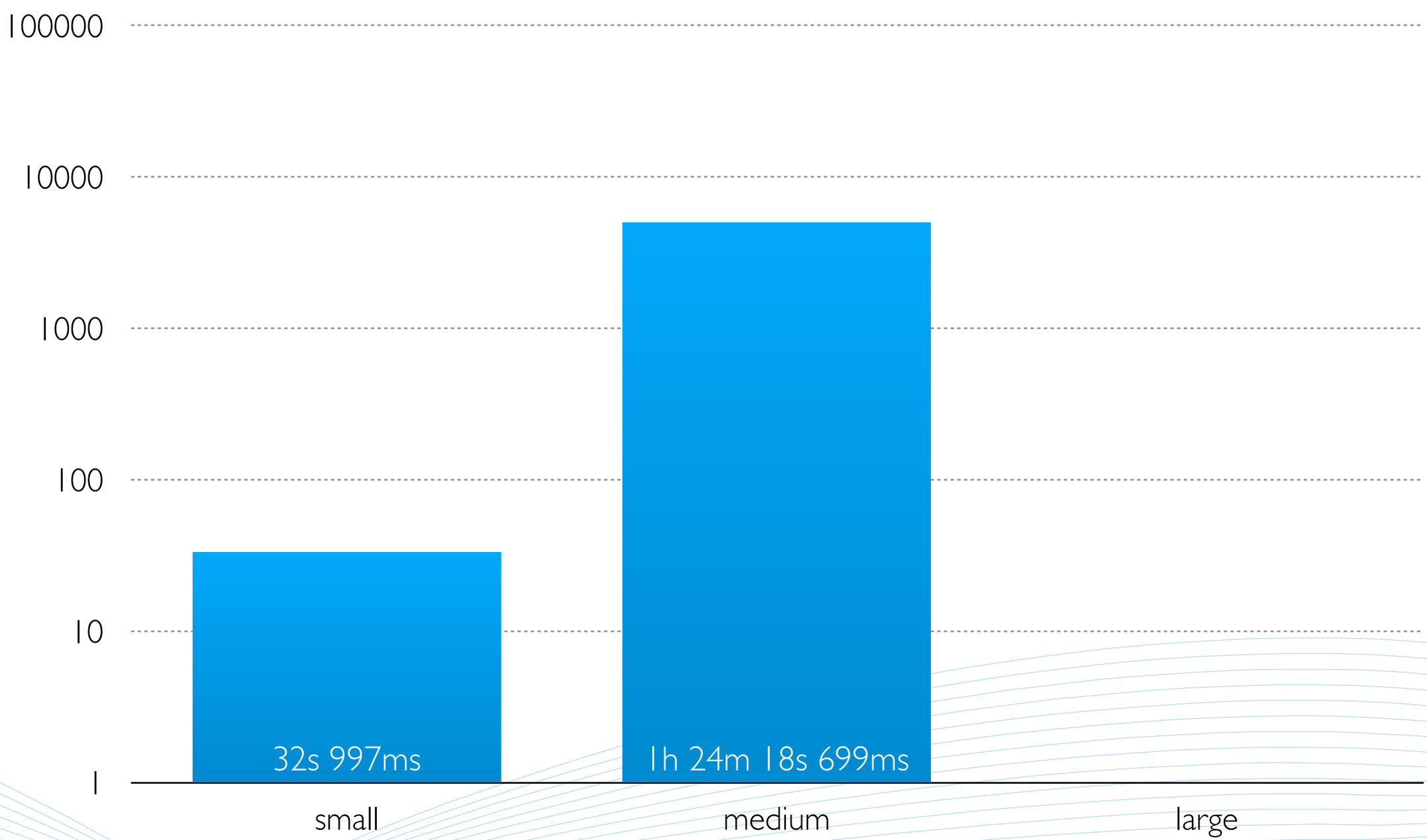


One-shot

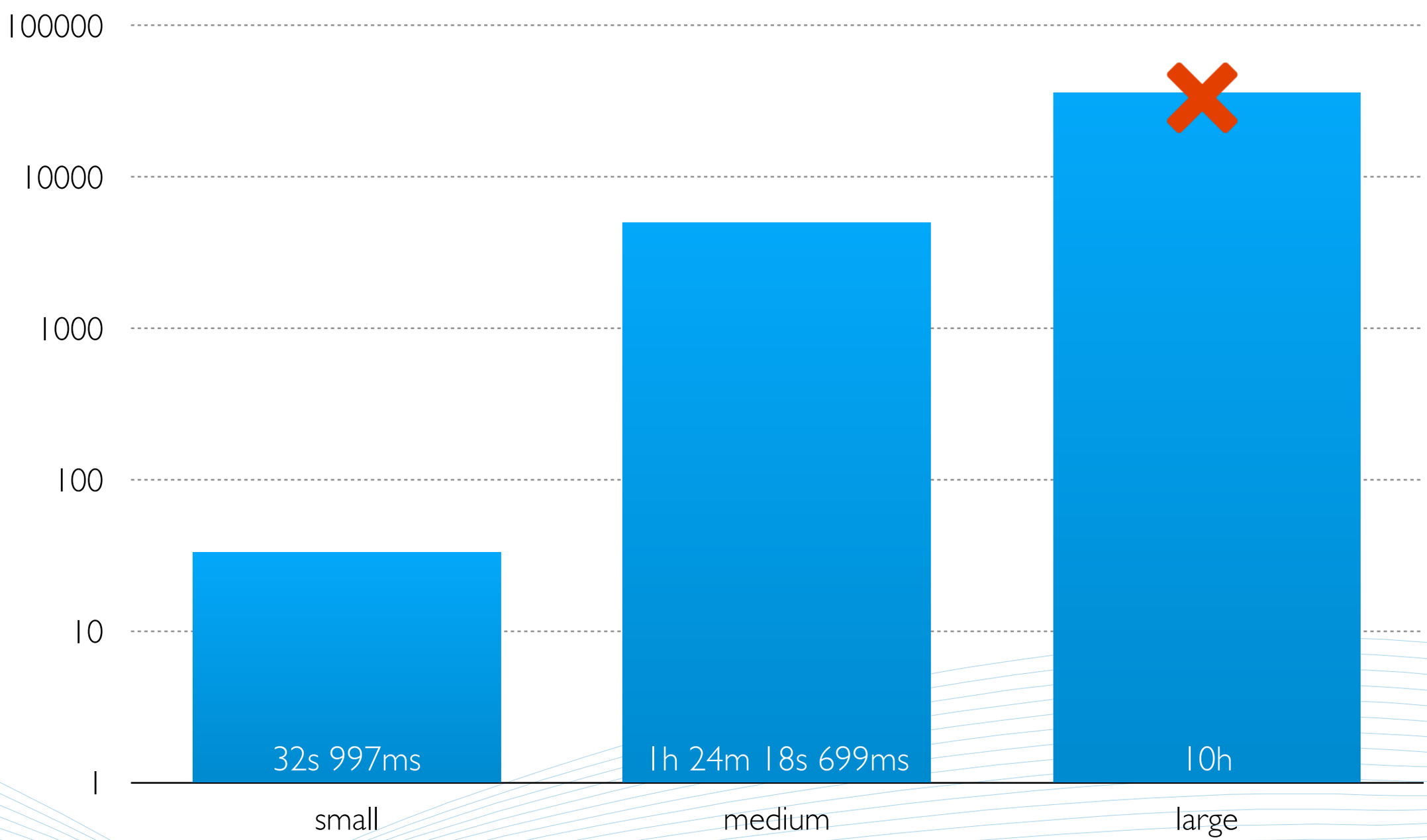


One-shot

Ensuring Reliable Networks TTTech



One-shot



One-shot

Ensuring Reliable Networks TTTech

One-shot

Ensuring Reliable Networks TTTech

where does the complexity come from?

One-shot

where does the complexity come from?

where do the frames come from?

One-shot

where does the complexity come from?

where do the frames come from?

consumer tasks
producer tasks
communication

One-shot

where does the complexity come from?

where do the frames come from?

consumer tasks
producer tasks
communication

free tasks

One-shot

where does the complexity come from?

where do the frames come from?

consumer tasks
producer tasks
communication

free tasks

One-shot

Ensuring Reliable Networks TTTech

where does the complexity come from?

where do the frames come from?

consumer tasks
producer tasks
communication

free tasks

let's treat them differently

Demand-based

Ensuring Reliable Networks TTTech

Demand-based

Ensuring Reliable Networks TTTech

consumer tasks
producer tasks
communication

free tasks

Demand-based

consumer tasks
producer tasks
communication

free tasks

SMT

Demand-based

free tasks

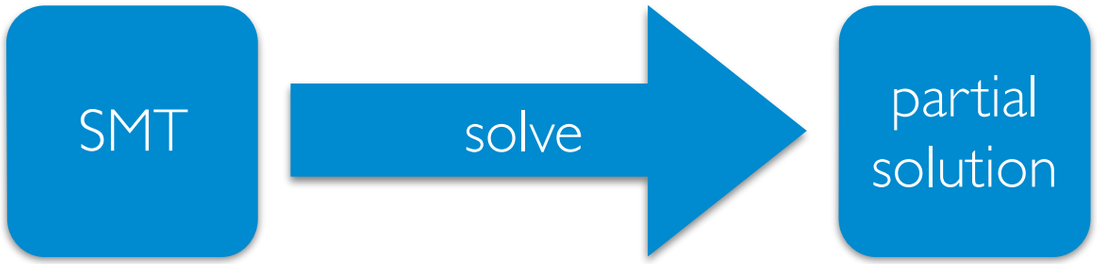
SMT

Demand-based

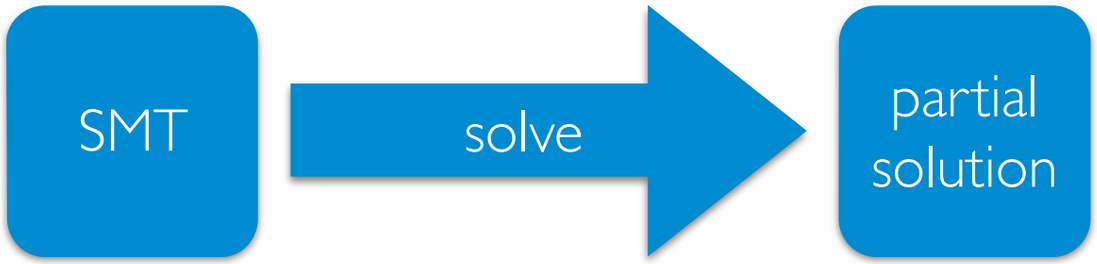
free tasks

Demand-based

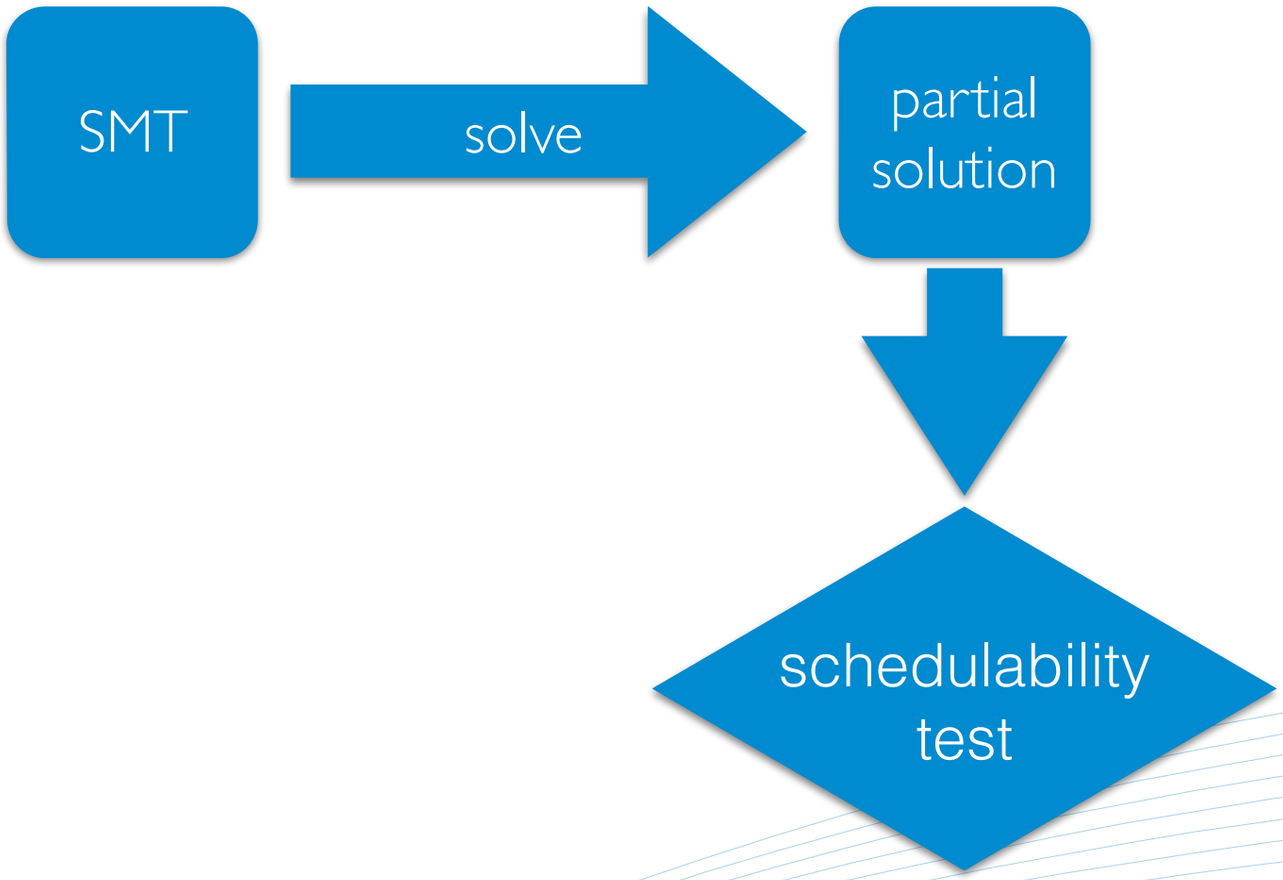
free tasks



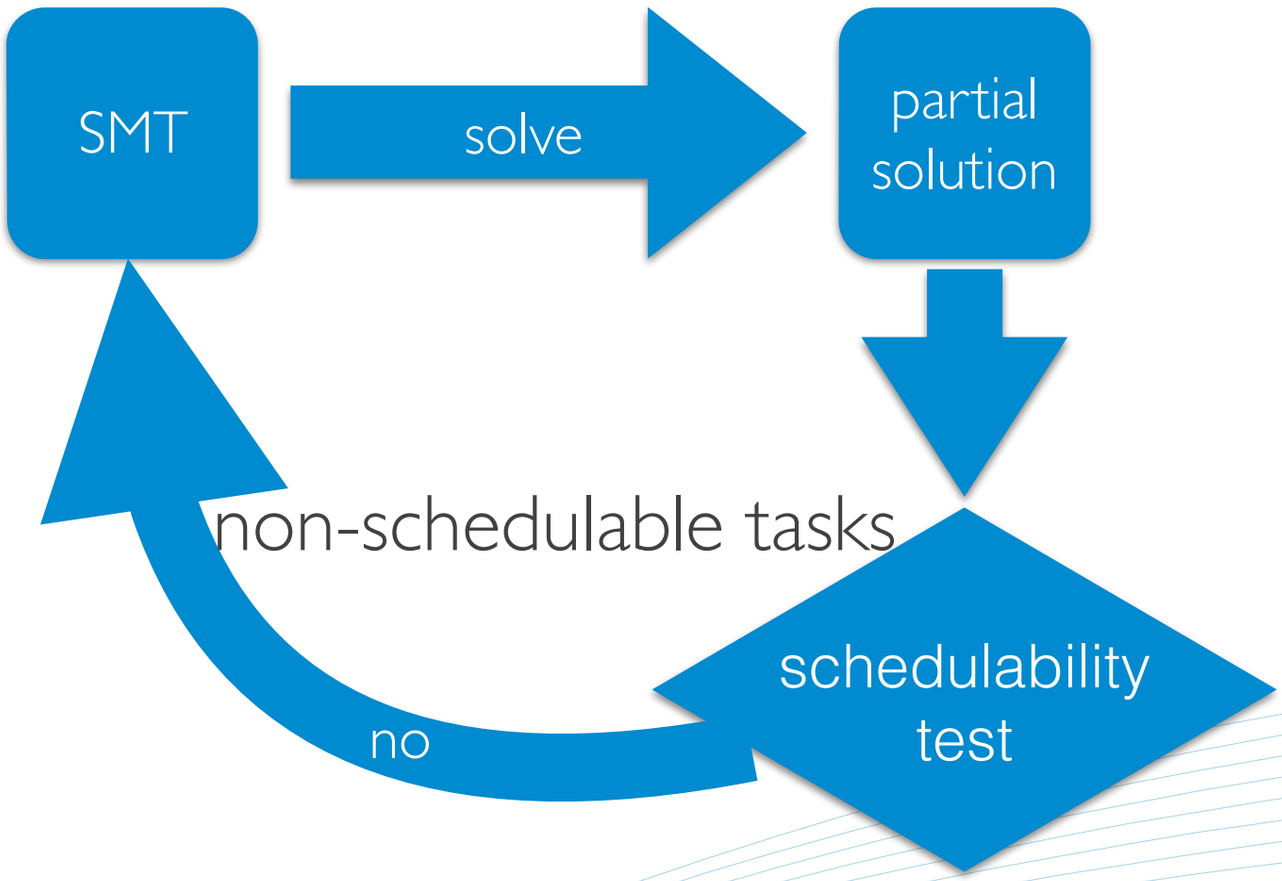
Demand-based



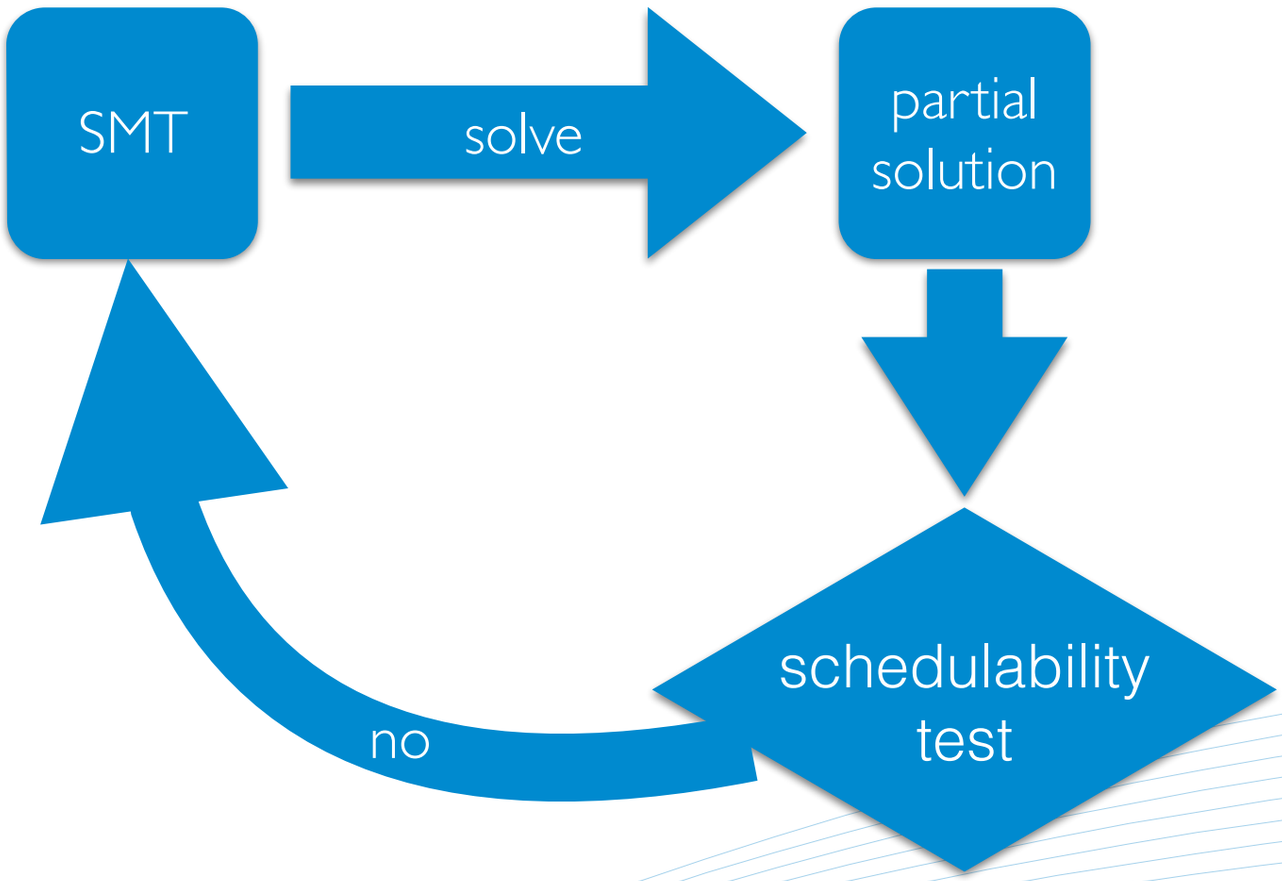
Demand-based



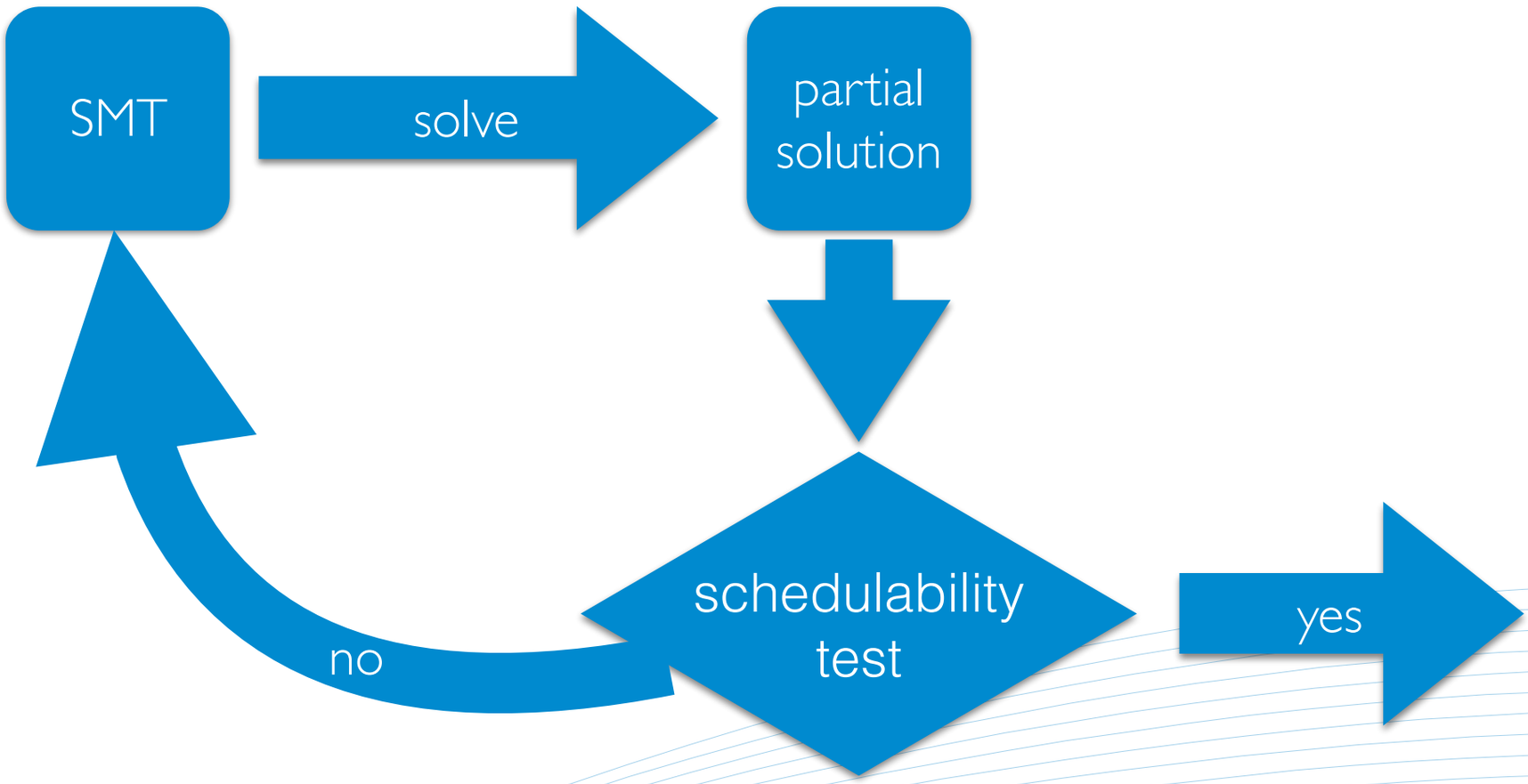
Demand-based



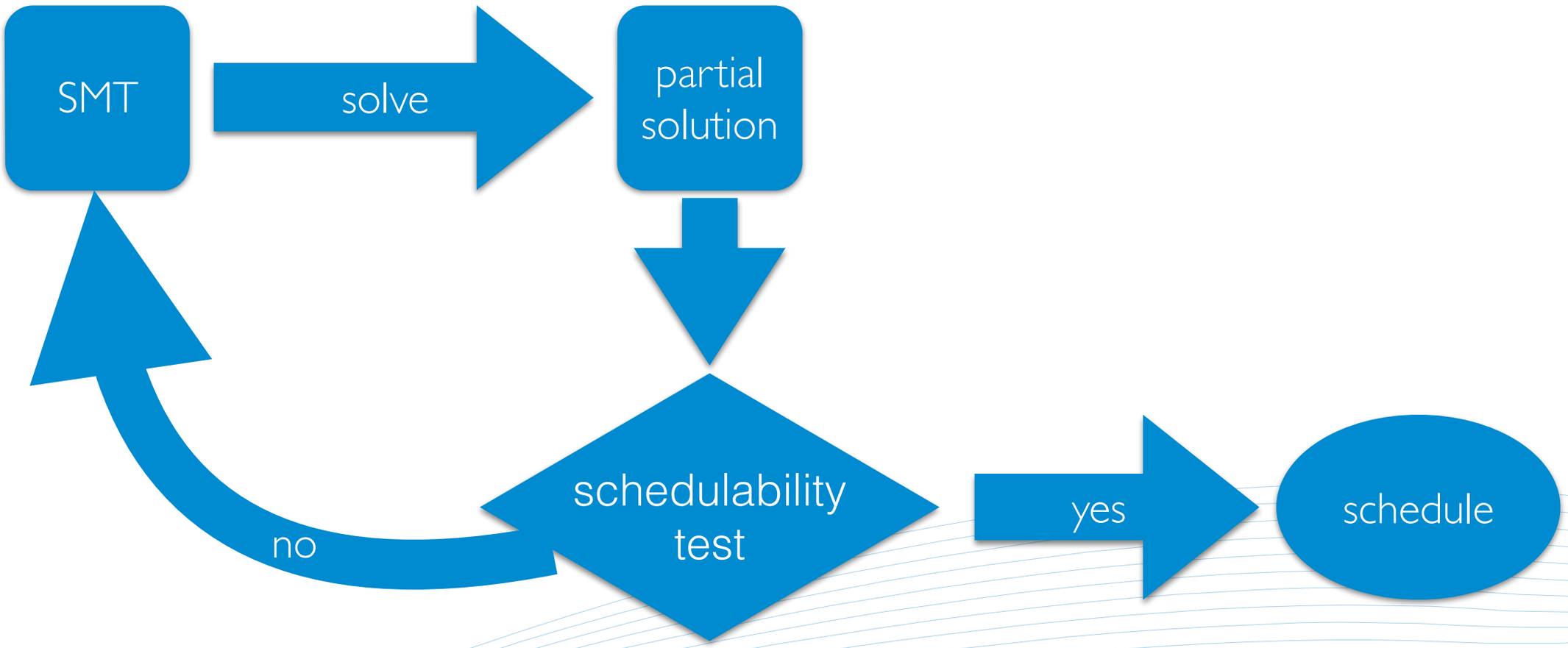
Demand-based



Demand-based



Demand-based

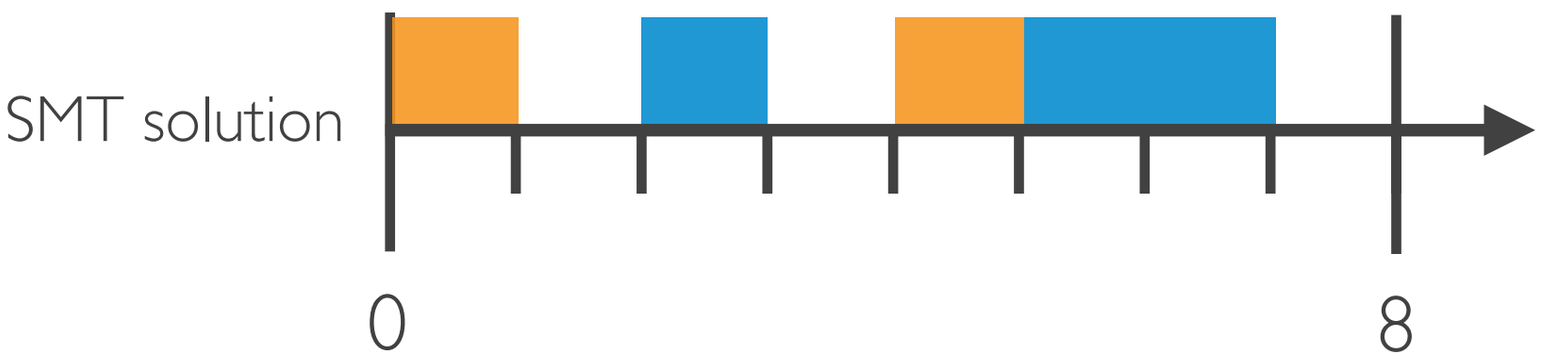


Demand-based

Ensuring Reliable Networks TTTech

Demand-based

Demand-based



EDF tasks

(0, 1, 8, 1)

(2, 1, 8, 1)

(4, 1, 8, 1)

(5, 2, 8, 2)

Demand-based

EDF tasks

(0, 1, 8, 1)

(2, 1, 8, 1)

(4, 1, 8, 1)

(5, 2, 8, 2)

⊕

free tasks

(0, 2, 8, 8)

(0, 1, 8, 8)

Demand-based

SMT solution

EDF ta

$$\forall v_a \in \mathcal{V}, \forall t_1 \in \Phi^{v_a}, \forall t_2 \in \Delta^{v_a}, t_1 < t_2 :$$

$$\sum_{\widetilde{\tau}_i{}^{v_a} \in \widetilde{\Gamma}^{v_a}} \widetilde{\tau}_i{}^{v_a}.C \times \left(\left\lfloor \frac{t_2 - \widetilde{\tau}_i{}^{v_a}.\phi - \widetilde{\tau}_i{}^{v_a}.D}{\widetilde{\tau}_i{}^{v_a}.T} \right\rfloor - \left\lceil \frac{t_1 - \widetilde{\tau}_i{}^{v_a}.\phi}{\widetilde{\tau}_i{}^{v_a}.T} \right\rceil + 1 \right)_0 \leq t_2 - t_1,$$

where

$$\Phi^{v_a} \stackrel{def}{=} \{a_{i,j}^{v_a} = \widetilde{\tau}_i{}^{v_a}.\phi + j \times \widetilde{\tau}_i{}^{v_a}.T | \widetilde{\tau}_i{}^{v_a} \in \widetilde{\Gamma}^{v_a}, j \geq 0, a_{i,j}^{v_a} \leq \lambda^{v_a}\},$$

$$\Delta^{v_a} \stackrel{def}{=} \{d_{i,j}^{v_a} = a_{i,j}^{v_a} + \widetilde{\tau}_i{}^{v_a}.D | \widetilde{\tau}_i{}^{v_a} \in \widetilde{\Gamma}^{v_a}, j \geq 0, d_{i,j}^{v_a} \leq \lambda^{v_a}\},$$

$$\lambda^{v_a} = max(\{\widetilde{\tau}_i{}^{v_a}.\phi | \widetilde{\tau}_i{}^{v_a} \in \widetilde{\Gamma}^{v_a}\}) + 2 \times lcm(\{\widetilde{\tau}_i{}^{v_a}.T | \widetilde{\tau}_i{}^{v_a} \in \widetilde{\Gamma}^{v_a}\}).$$

Demand-based

EDF tasks

(0, 1, 8, 1)

(2, 1, 8, 1)

(4, 1, 8, 1)

(5, 2, 8, 2)

$\oplus$

free tasks

(0, 2, 8, 8)

(0, 1, 8, 8)

$$\forall v_a \in \mathcal{V}, \forall t_1 \in \Phi^{v_a}, \forall t_2 \in \Delta^{v_a}, t_1 < t_2 :$$

$$\sum_{\widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}} \widetilde{\tau}_i^{v_a}.C \times \left(\left\lfloor \frac{t_2 - \widetilde{\tau}_i^{v_a}.\phi - \widetilde{\tau}_i^{v_a}.D}{\widetilde{\tau}_i^{v_a}.T} \right\rfloor - \left\lceil \frac{t_1 - \widetilde{\tau}_i^{v_a}.\phi}{\widetilde{\tau}_i^{v_a}.T} \right\rceil + 1 \right)_0 \leq t_2 - t_1,$$

where

$$\Phi^{v_a} \stackrel{def}{=} \{a_{i,j}^{v_a} = \widetilde{\tau}_i^{v_a}.\phi + j \times \widetilde{\tau}_i^{v_a}.T | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}, j \geq 0, a_{i,j}^{v_a} \leq \lambda^{v_a}\},$$

$$\Delta^{v_a} \stackrel{def}{=} \{d_{i,j}^{v_a} = a_{i,j}^{v_a} + \widetilde{\tau}_i^{v_a}.D | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}, j \geq 0, d_{i,j}^{v_a} \leq \lambda^{v_a}\},$$

$$\lambda^{v_a} = max(\{\widetilde{\tau}_i^{v_a}.\phi | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}\}) + 2 \times lcm(\{\widetilde{\tau}_i^{v_a}.T | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}\}).$$

Demand-based

Ensuring Reliable Networks TTTech

EDF tasks

(0, 1, 8, 1)

(2, 1, 8, 1)

(4, 1, 8, 1)

(5, 2, 8, 2)

⊕

free tasks

(0, 2, 8, 8)

(0, 1, 8, 8)

$$\forall v_a \in \mathcal{V}, \forall t_1 \in \Phi^{v_a}, \forall t_2 \in \Delta^{v_a}, t_1 < t_2 :$$

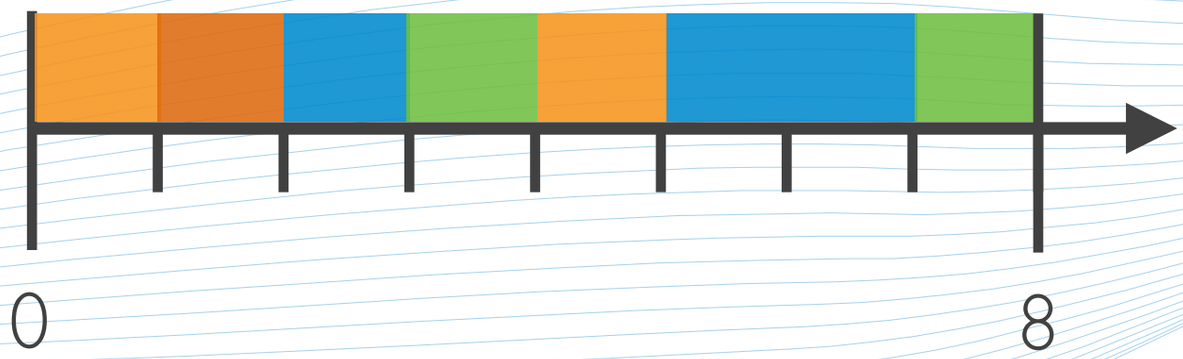
$$\sum_{\widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}} \widetilde{\tau}_i^{v_a}.C \times \left(\left\lfloor \frac{t_2 - \widetilde{\tau}_i^{v_a}.\phi - \widetilde{\tau}_i^{v_a}.D}{\widetilde{\tau}_i^{v_a}.T} \right\rfloor - \left\lceil \frac{t_1 - \widetilde{\tau}_i^{v_a}.\phi}{\widetilde{\tau}_i^{v_a}.T} \right\rceil + 1 \right)_0 \leq t_2 - t_1,$$

where

$$\Phi^{v_a} \stackrel{def}{=} \{a_{i,j}^{v_a} = \widetilde{\tau}_i^{v_a}.\phi + j \times \widetilde{\tau}_i^{v_a}.T | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}, j \geq 0, a_{i,j}^{v_a} \leq \lambda^{v_a}\},$$

$$\Delta^{v_a} \stackrel{def}{=} \{d_{i,j}^{v_a} = a_{i,j}^{v_a} + \widetilde{\tau}_i^{v_a}.D | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}, j \geq 0, d_{i,j}^{v_a} \leq \lambda^{v_a}\},$$

$$\lambda^{v_a} = max(\{\widetilde{\tau}_i^{v_a}.\phi | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}\}) + 2 \times lcm(\{\widetilde{\tau}_i^{v_a}.T | \widetilde{\tau}_i^{v_a} \in \widetilde{\Gamma}^{v_a}\}).$$



Demand-based

Ensuring Reliable Networks TTTech

Demand-based

- transform scheduled frames on CPUs into asynchronous periodic tasks

Demand-based

- transform scheduled frames on CPUs into asynchronous periodic tasks
- add free tasks and apply schedulability test [Pellizzoni@RealTimeSyst05]

Demand-based

- transform scheduled frames on CPUs into asynchronous periodic tasks
- add free tasks and apply schedulability test [Pellizzoni@RealTimeSyst05]
- if not schedulable, increment number of tasks that are solved with SMT

Demand-based

- transform scheduled frames on CPUs into asynchronous periodic tasks
- add free tasks and apply schedulability test [Pellizzoni@RealTimeSyst05]
- if not schedulable, increment number of tasks that are solved with SMT
- if schedulable, generate final schedule by simulating EDF until HP

Demand-based

- transform scheduled frames on CPUs into asynchronous periodic tasks
- add free tasks and apply schedulability test [Pellizzoni@RealTimeSyst05]
- if not schedulable, increment number of tasks that are solved with SMT
- if schedulable, generate final schedule by simulating EDF until HP
- incremental algorithm so we don't lose schedulability

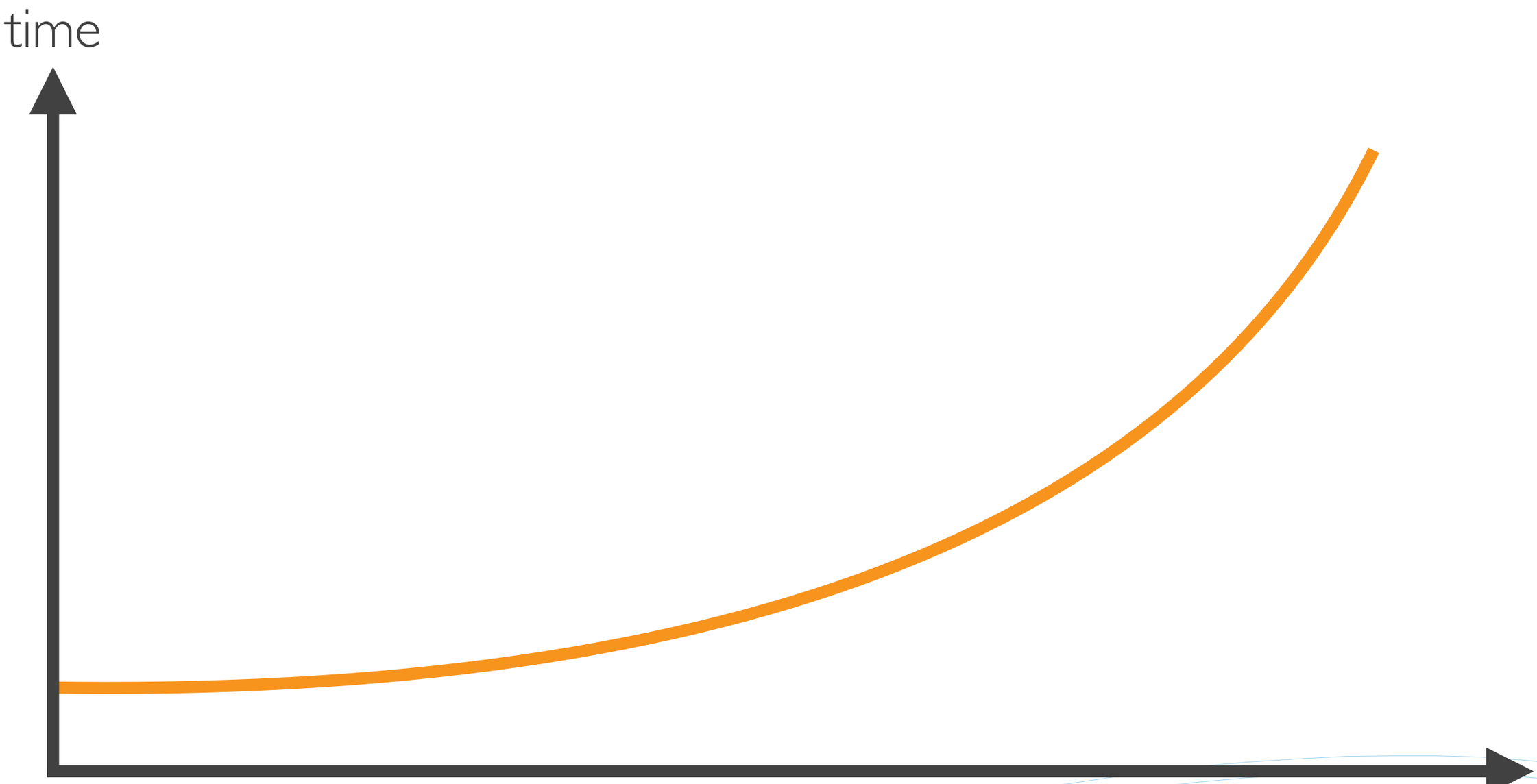
Demand-based

- transform scheduled frames on CPUs into asynchronous periodic tasks
- add free tasks and apply schedulability test [Pellizzoni@RealTimeSyst05]
- if not schedulable, increment number of tasks that are solved with SMT
- if schedulable, generate final schedule by simulating EDF until HP
- incremental algorithm so we don't lose schedulability
- we are still exponential but scale better for the average case

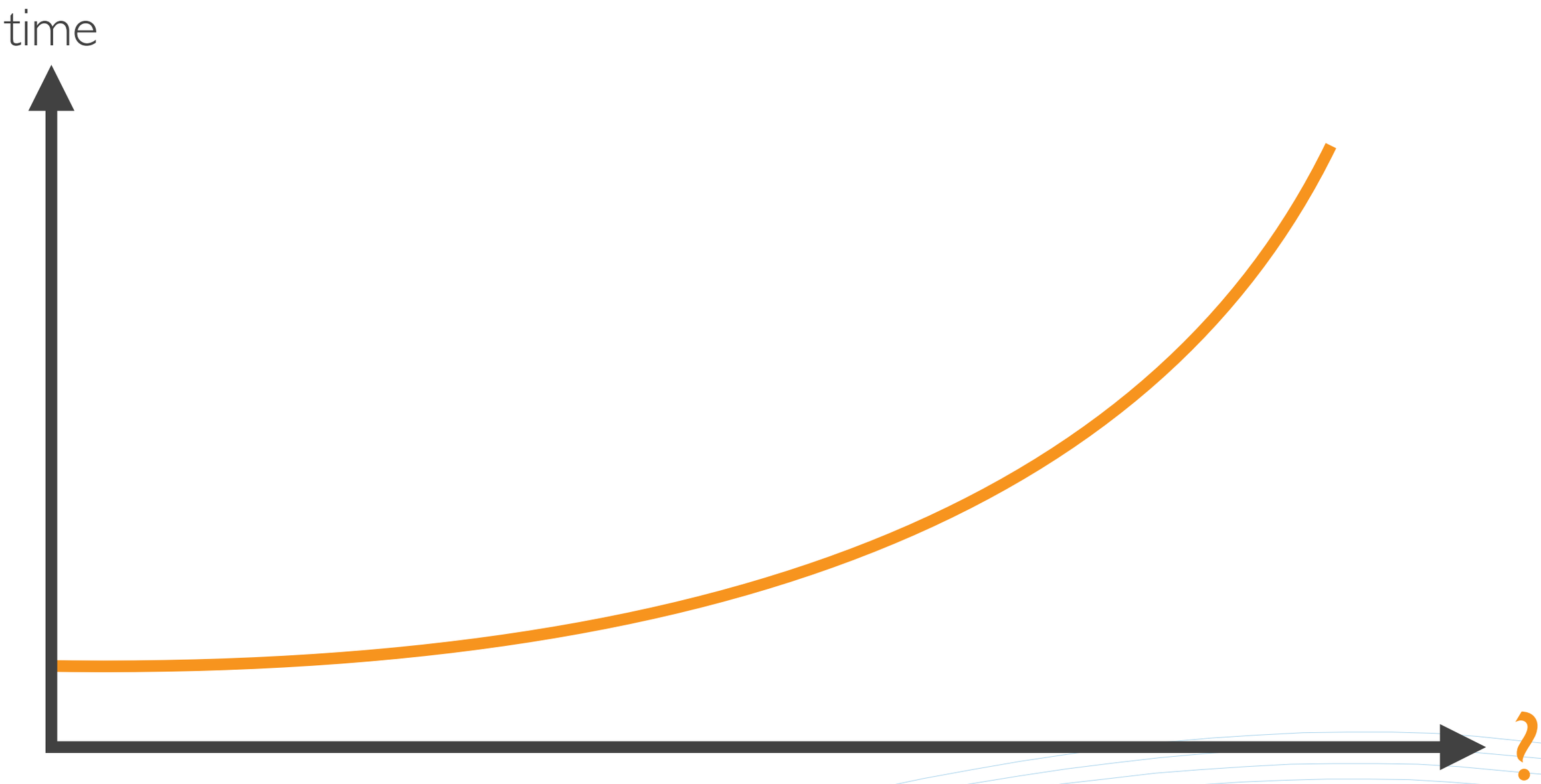
Evaluation

Evaluation

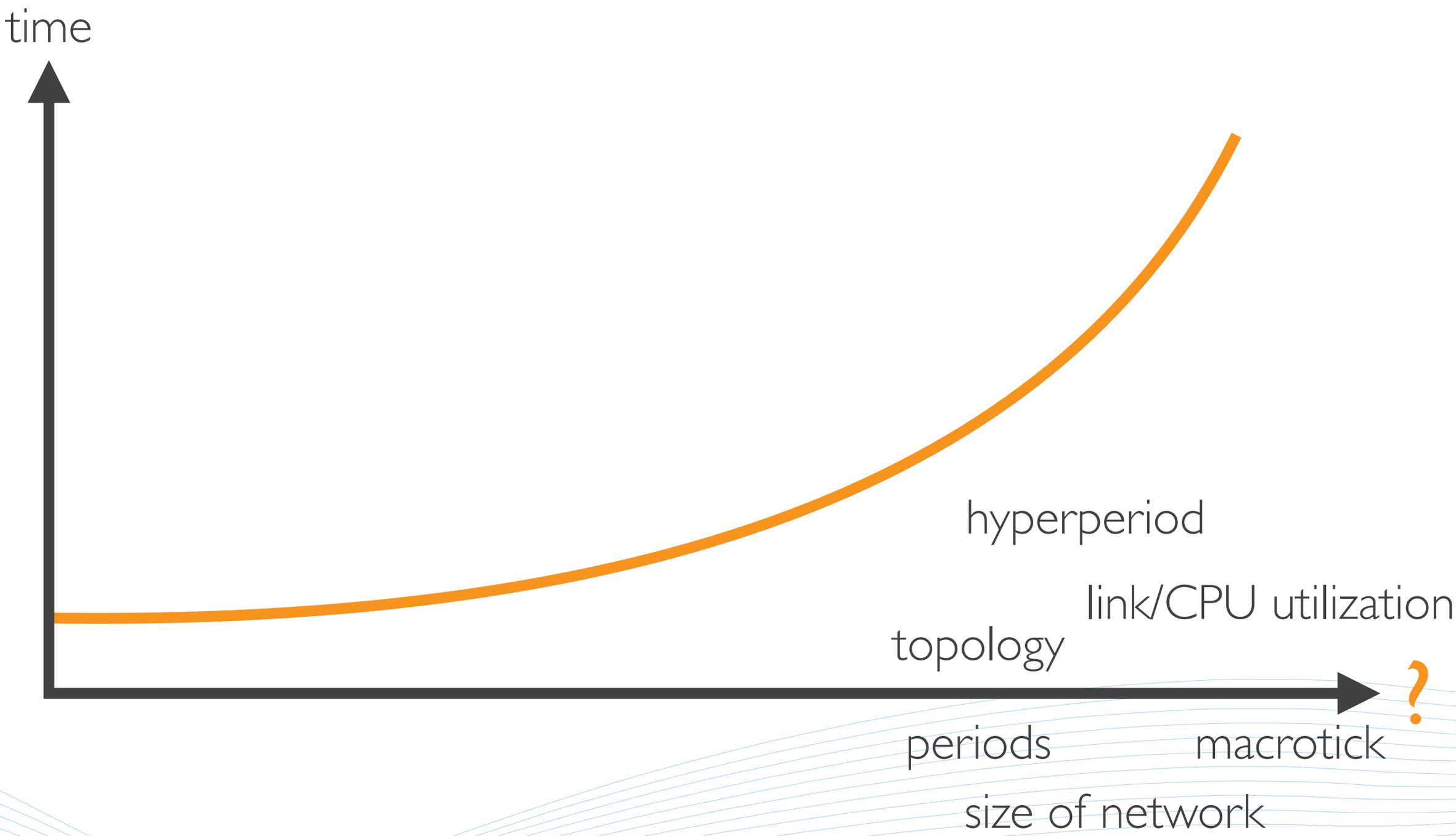
Evaluation



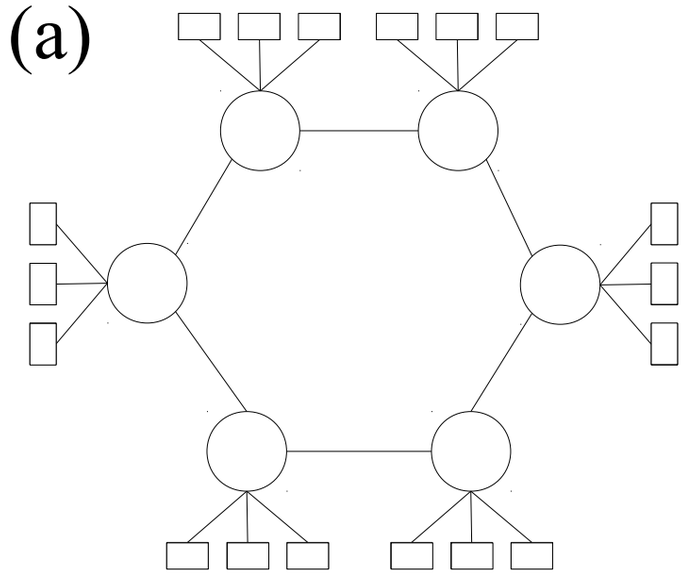
Evaluation

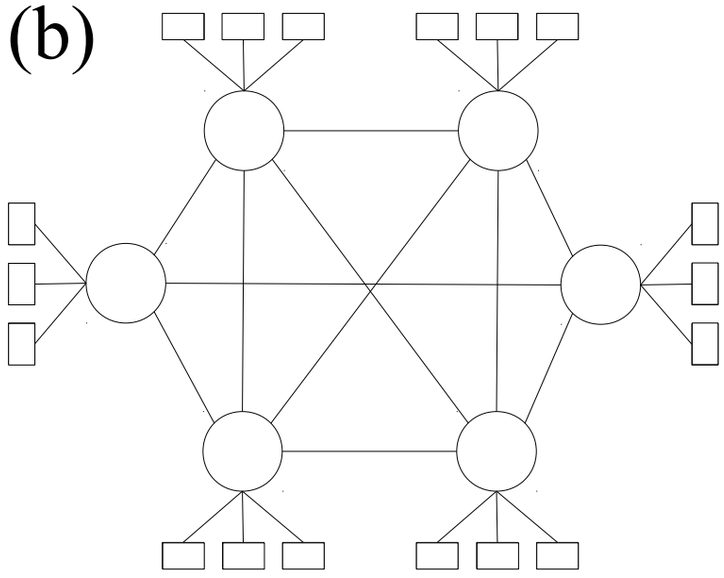


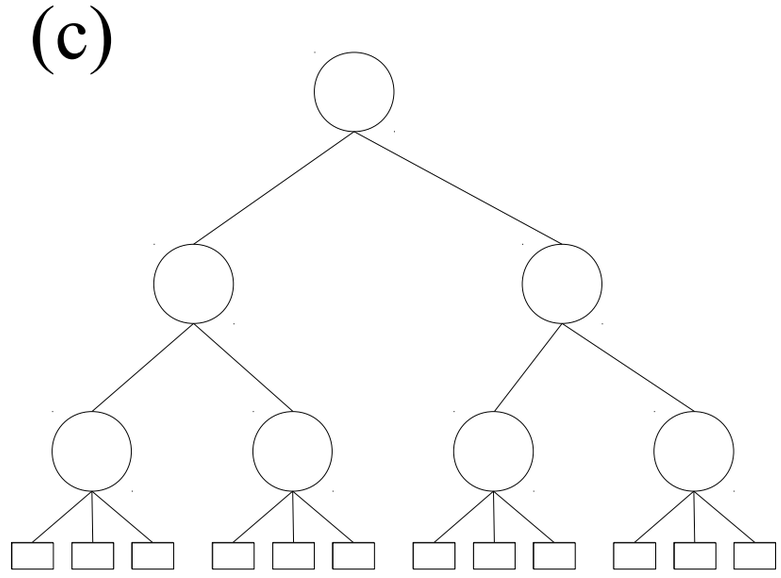
Evaluation



Topologies







Periods {10,20,25,50,100}, {10,30,100}, {50,75} ms

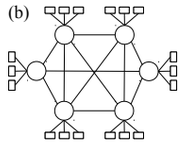
1usec network link granularity

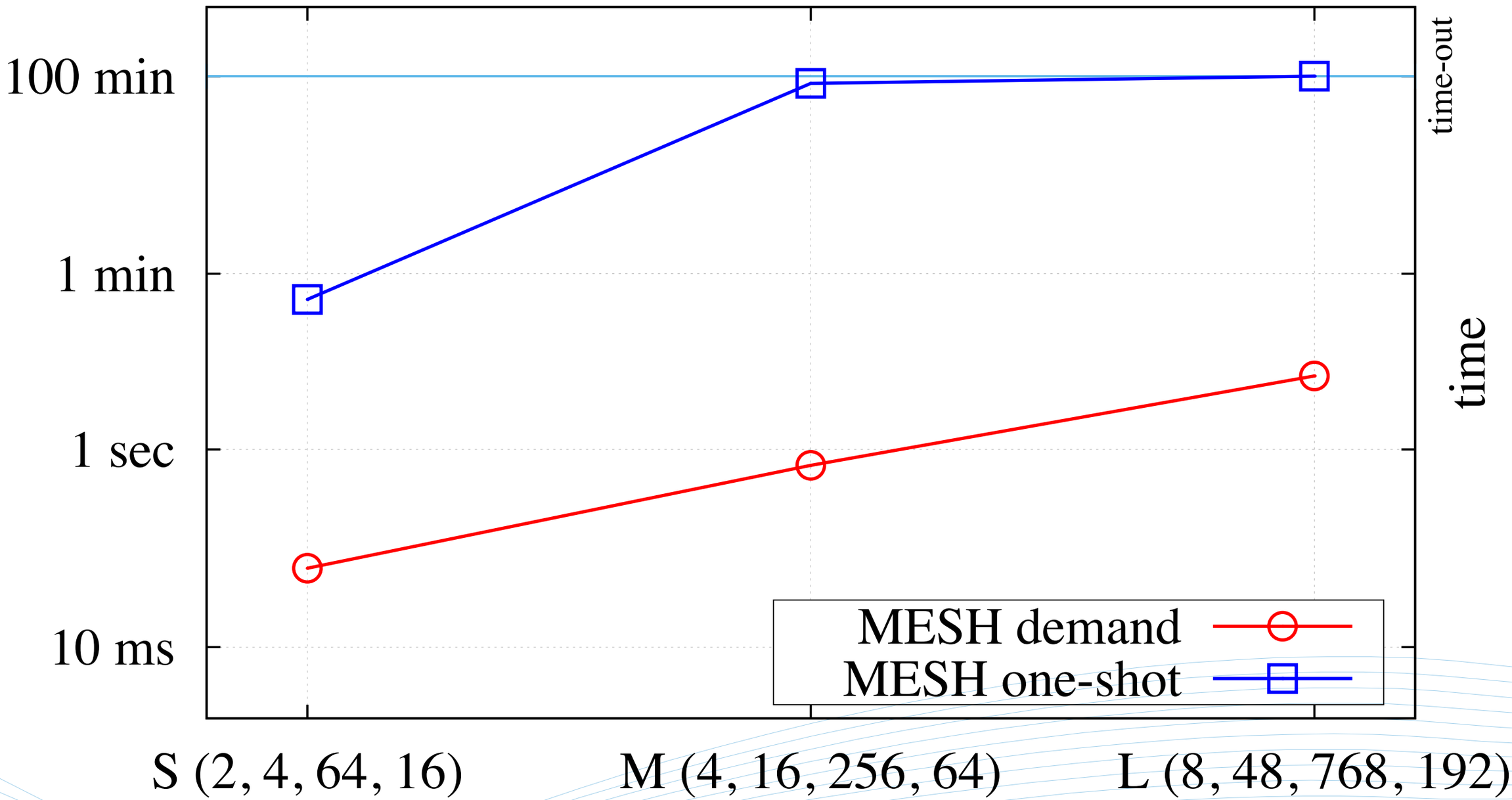
100Mbit/s and 1Gbit/s

random message size and virtual links

different macrotick and utilization configurations

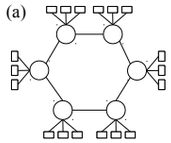
Mesh

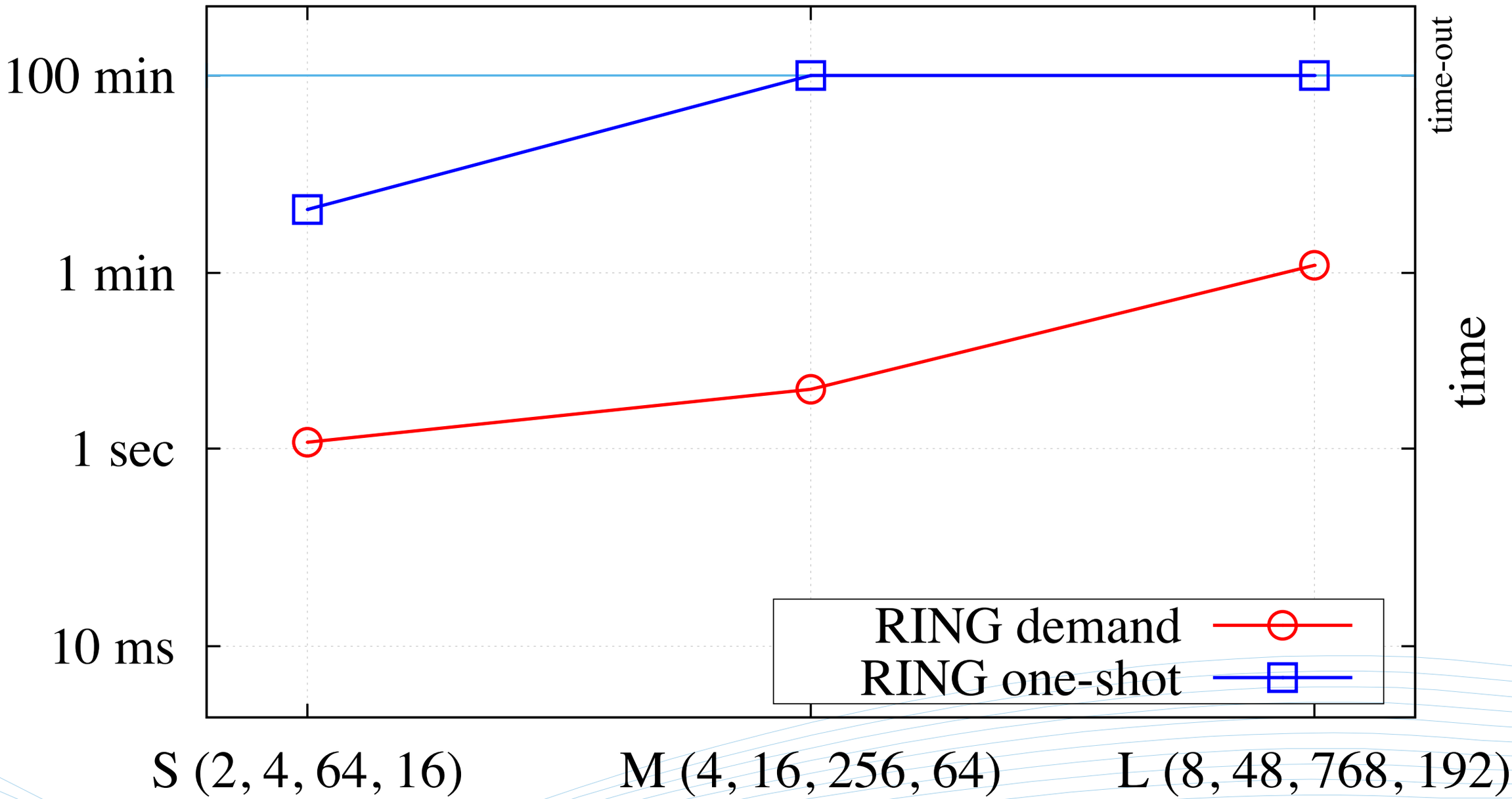




Periods {10,20,25,50,100} ms

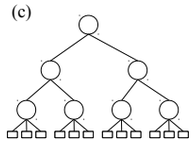
Ring

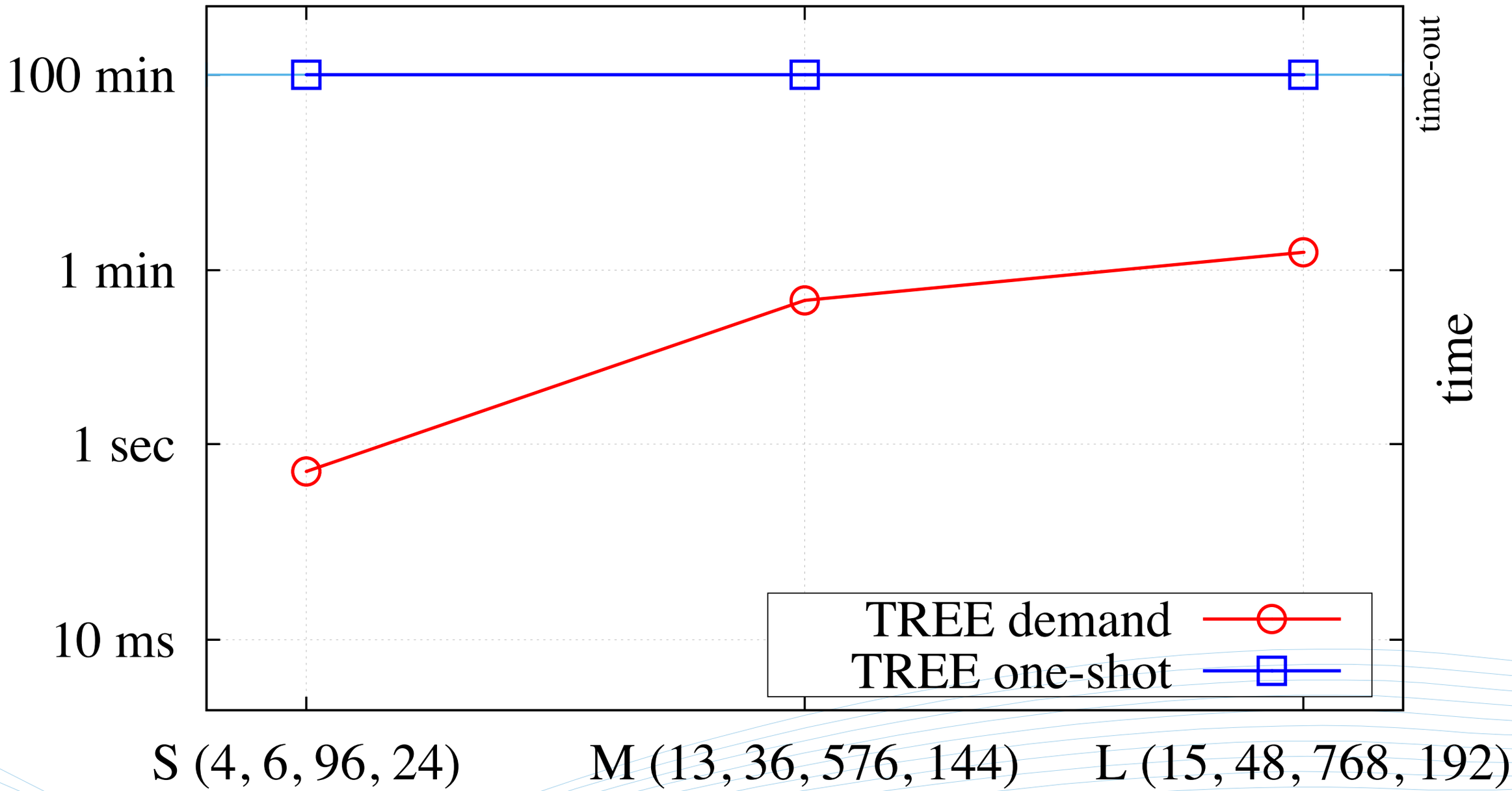




Periods {10,30,100} ms

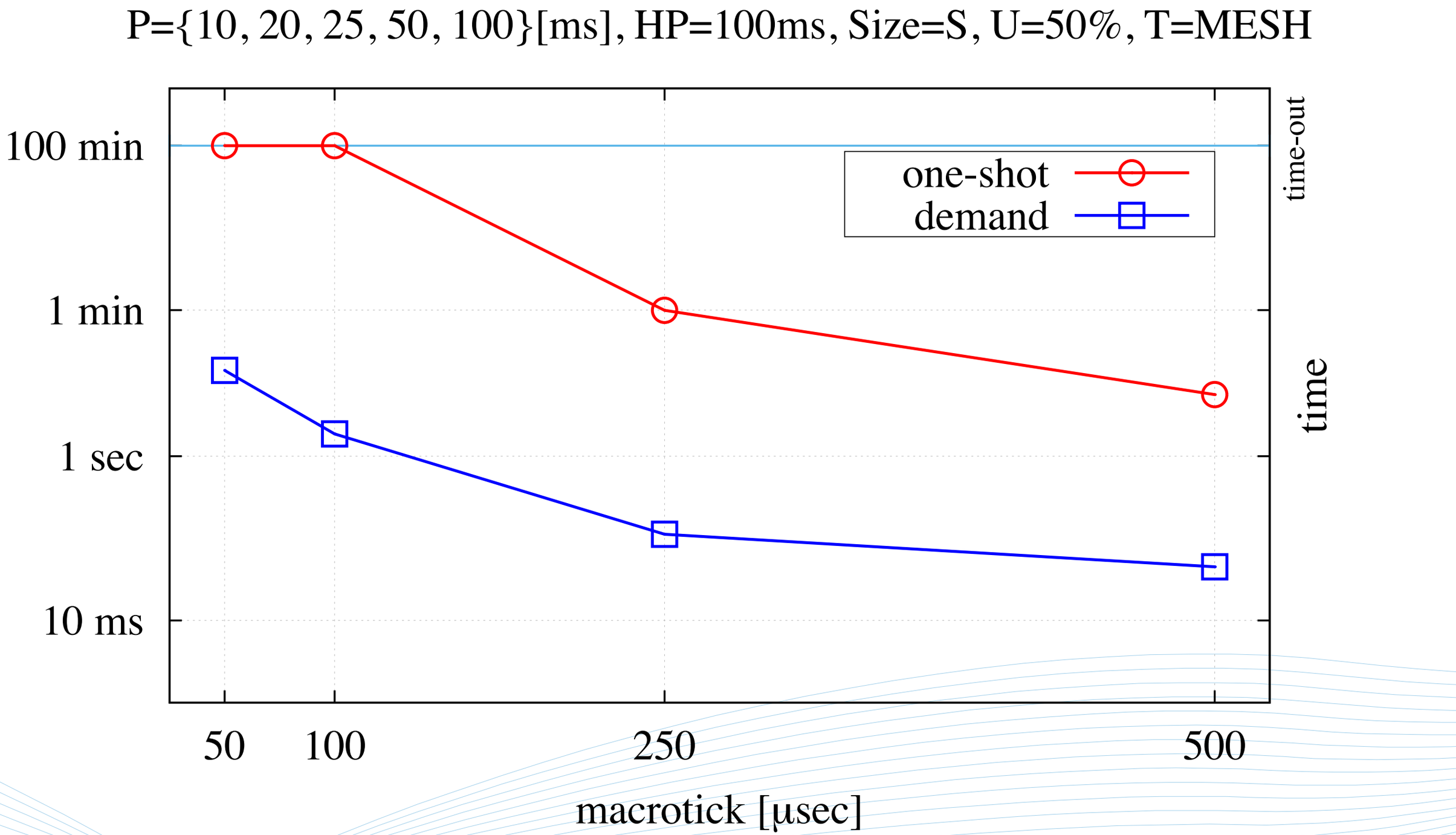
Tree



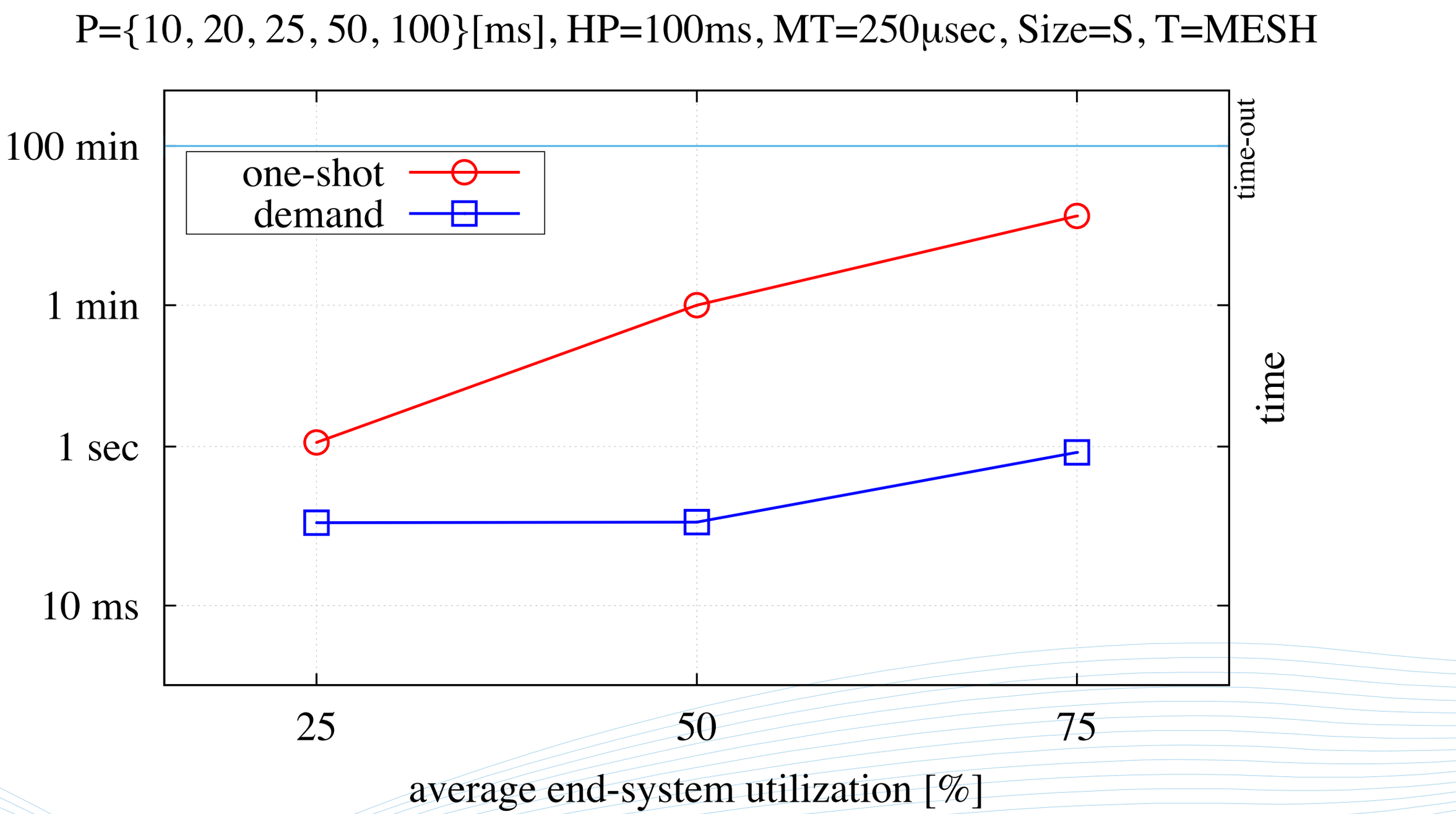


Periods {50,75} ms

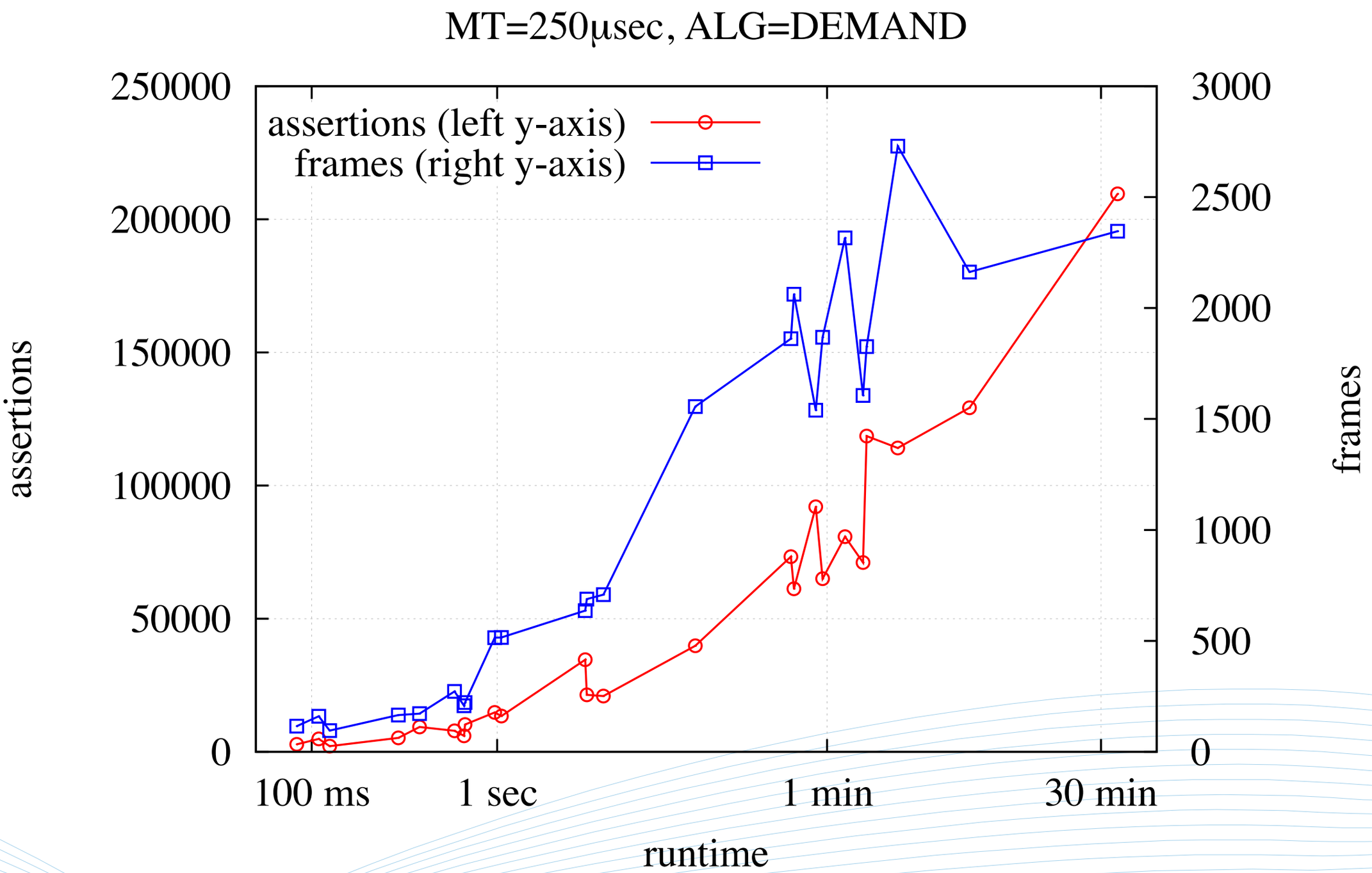
Macrotick



Utilization



SMT assertions



Conclusions

Ensuring Reliable Networks TTTech

Conclusions

Ensuring Reliable Networks TTTech

co-synthesis of task and message schedules

Conclusions

co-synthesis of task and message schedules

preemptive tasks

Conclusions

co-synthesis of task and message schedules

preemptive tasks

switched multi-speed TTEthernet networks

Conclusions

co-synthesis of task and message schedules

preemptive tasks

switched multi-speed TTEthernet networks

satisfiability modulo theories

Conclusions

co-synthesis of task and message schedules

preemptive tasks

switched multi-speed TTEthernet networks

satisfiability modulo theories

demand based approach

Conclusions

co-synthesis of task and message schedules

preemptive tasks

switched multi-speed TTEthernet networks

satisfiability modulo theories

demand based approach

scales for medium to large industrial systems

Thank you!

TTTech

Ensuring Reliable Networks

www.tttech.com

SMT-scheduled frames

Ensuring Reliable Networks TTTech