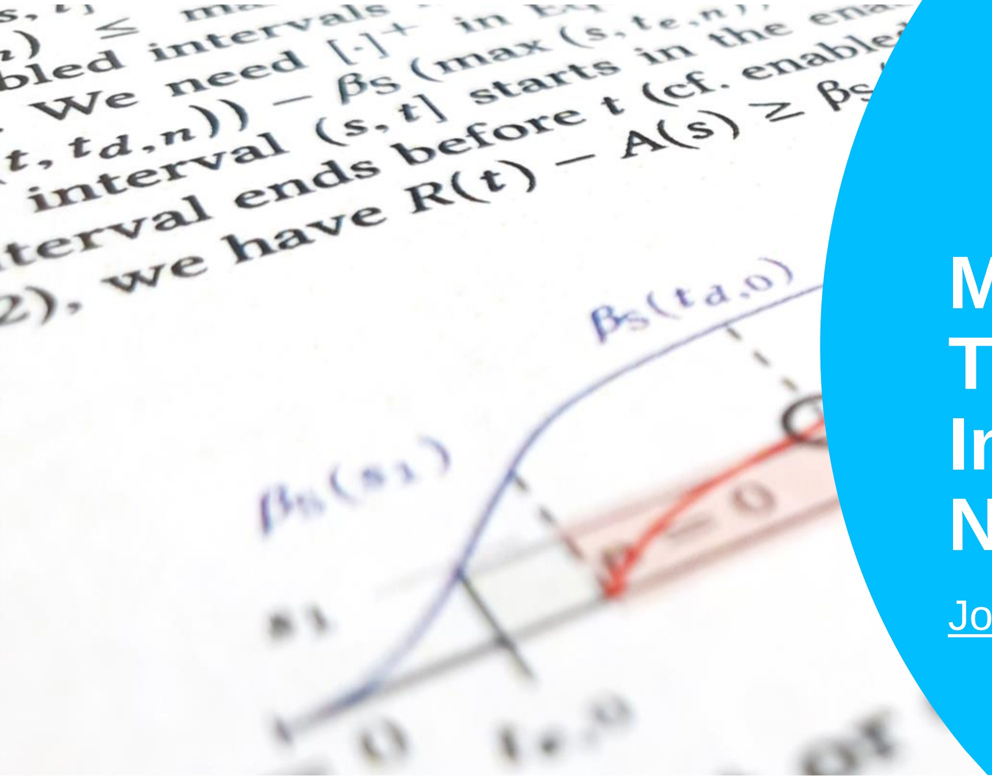
University of Stuttgart
Institute of Parallel and Distributed Systems -
Distributed Systems Group



Modeling Time-Triggered Service Intermittence In Network Calculus

Jonathan Falk, Frank Dürr, Kurt Rothermel

RTNS 2019

IPVS

The age of the cyber-physical machine.

More distributed systems interfacing with the physical world

- “Smart” {city, factory, home}
- Autonomous Driving
- IEEE Time-sensitive Networking (TSN) Workgroup

- Increasingly complex network setups and network behavior
- Often non-trivial analysis

The age of the cyber-physical machine.

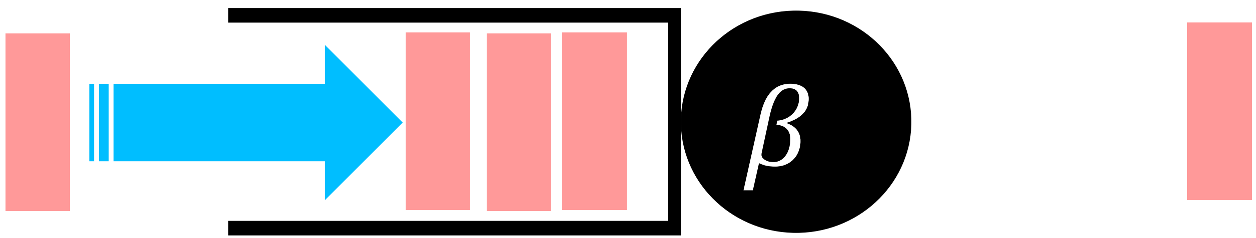
More distributed systems interfacing with the physical world

- “Smart” {city, factory, home}
- Autonomous Driving
- IEEE Time-sensitive Networking (TSN) Workgroup

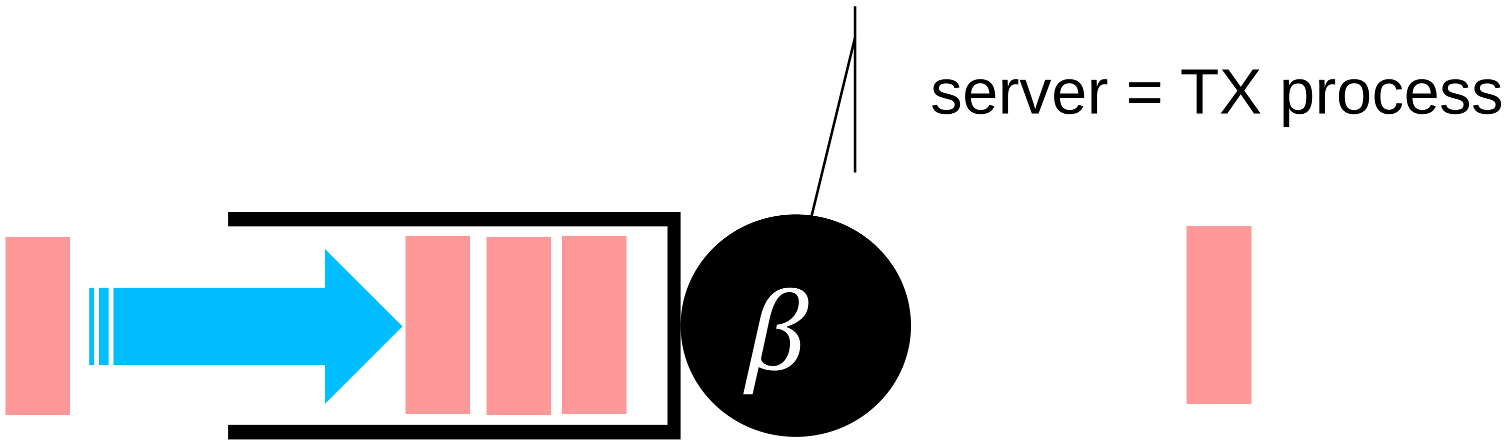
- Increasingly complex network setups and network behavior
- Often non-trivial analysis

Formal frameworks

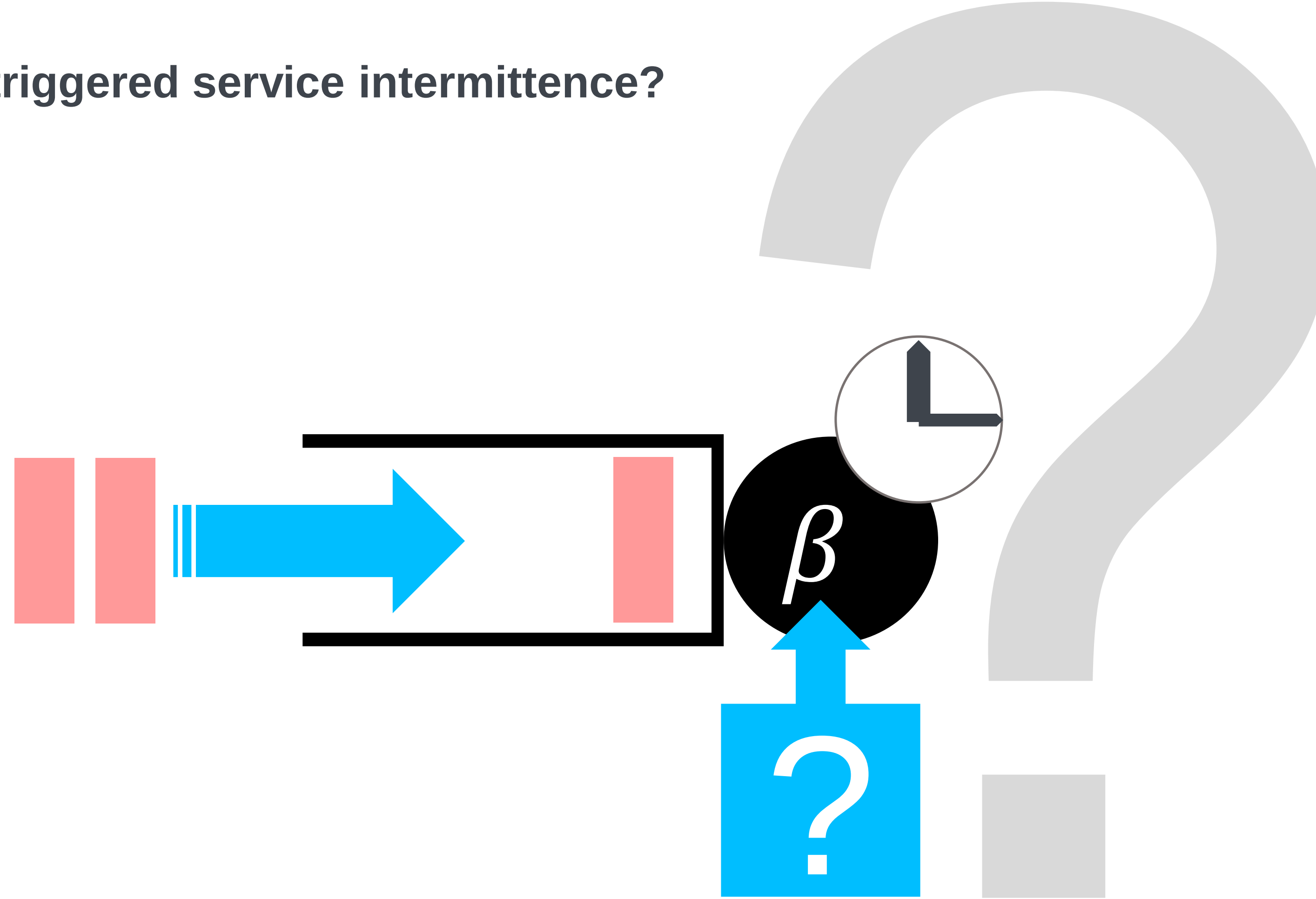
(Deterministic) network calculus to compute bounds in queuing systems



(Deterministic) network calculus to compute bounds in queuing systems



Time-triggered service intermittence?

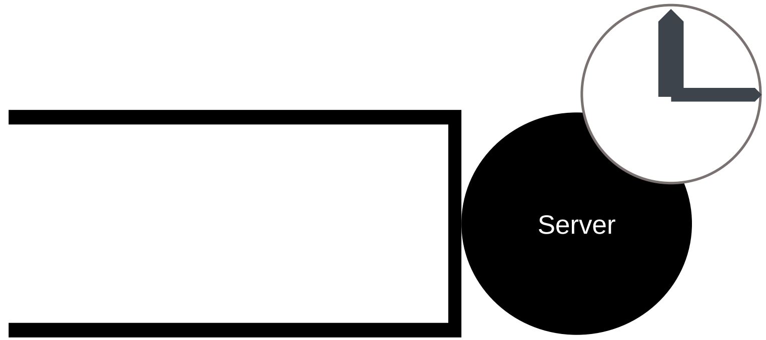


Contributions

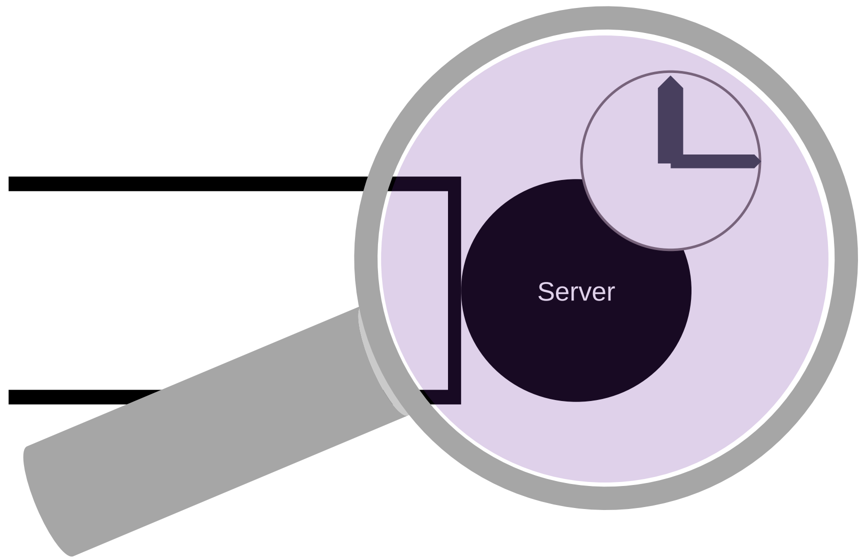
- Modeling 2 classes of systems with time-triggered service intermittence
  - Time-variant
  - Time-invariant
  - Implications?

- Highly generic model
  - Complex schedules for service intermittence
  - Beyond constant-rate service

Systems with intermittent service



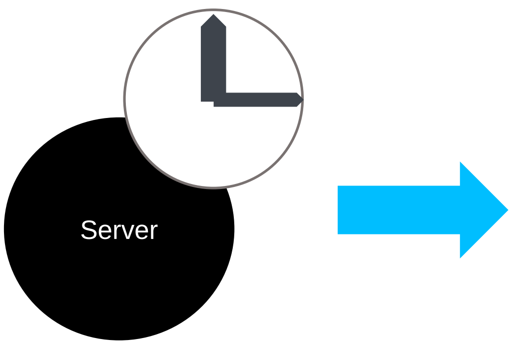
Systems with intermittent service

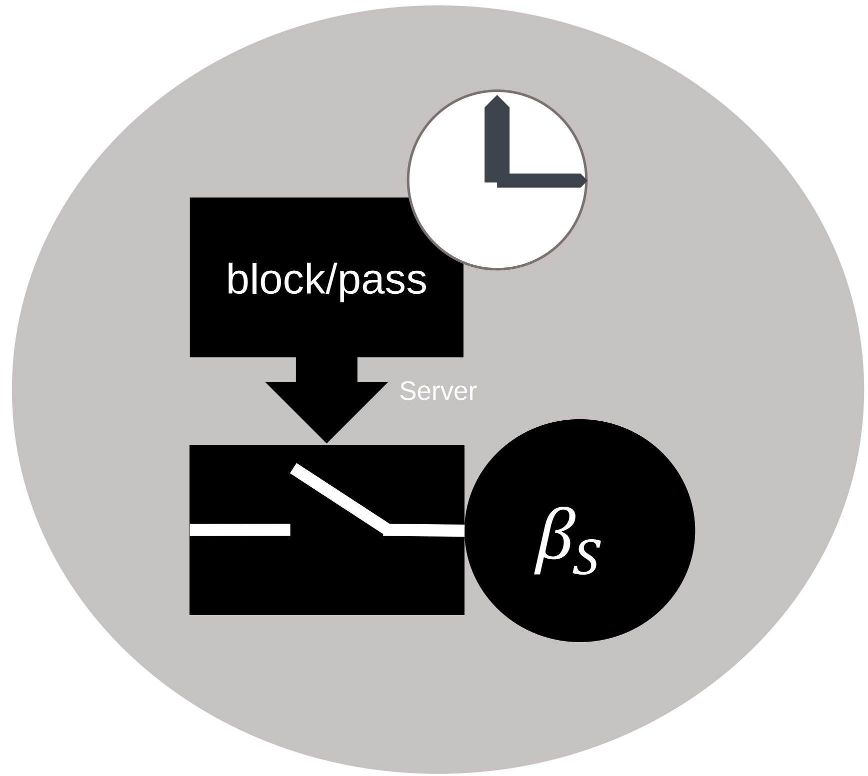


Time-triggered blocking

Motivating example:

Time-aware shaping

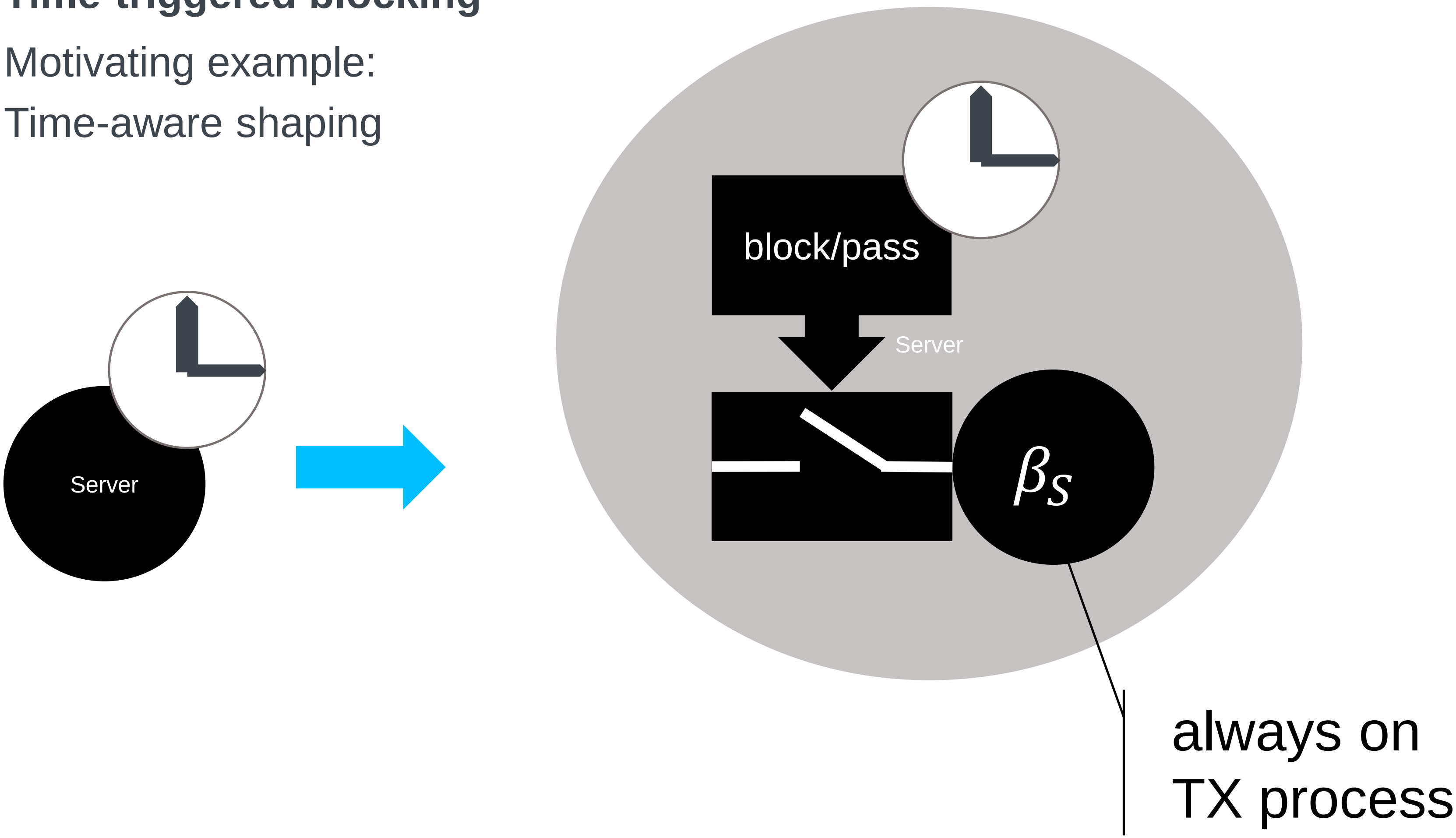




Time-triggered blocking

Motivating example:

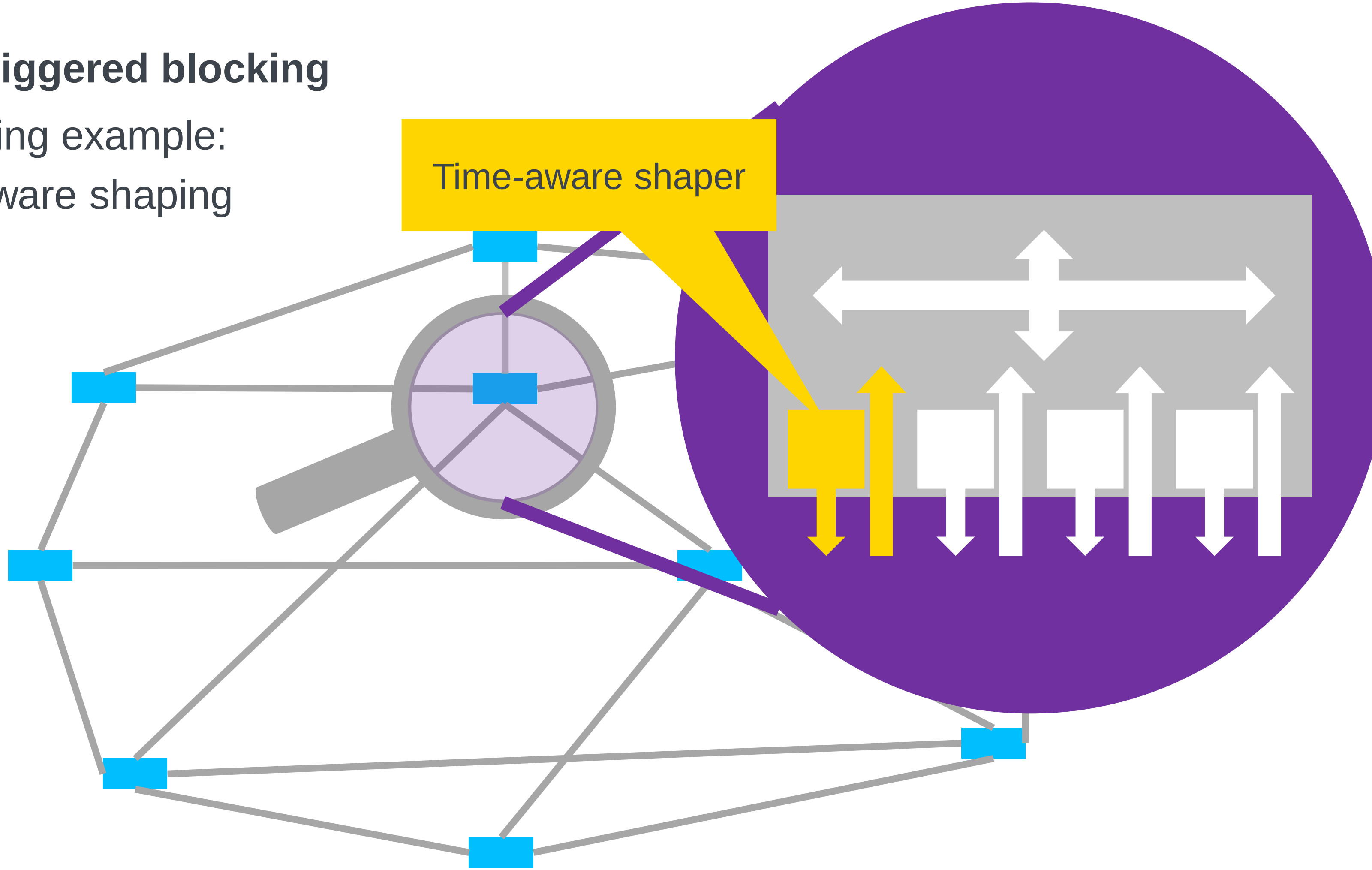
Time-aware shaping



Time-triggered blocking

Motivating example:

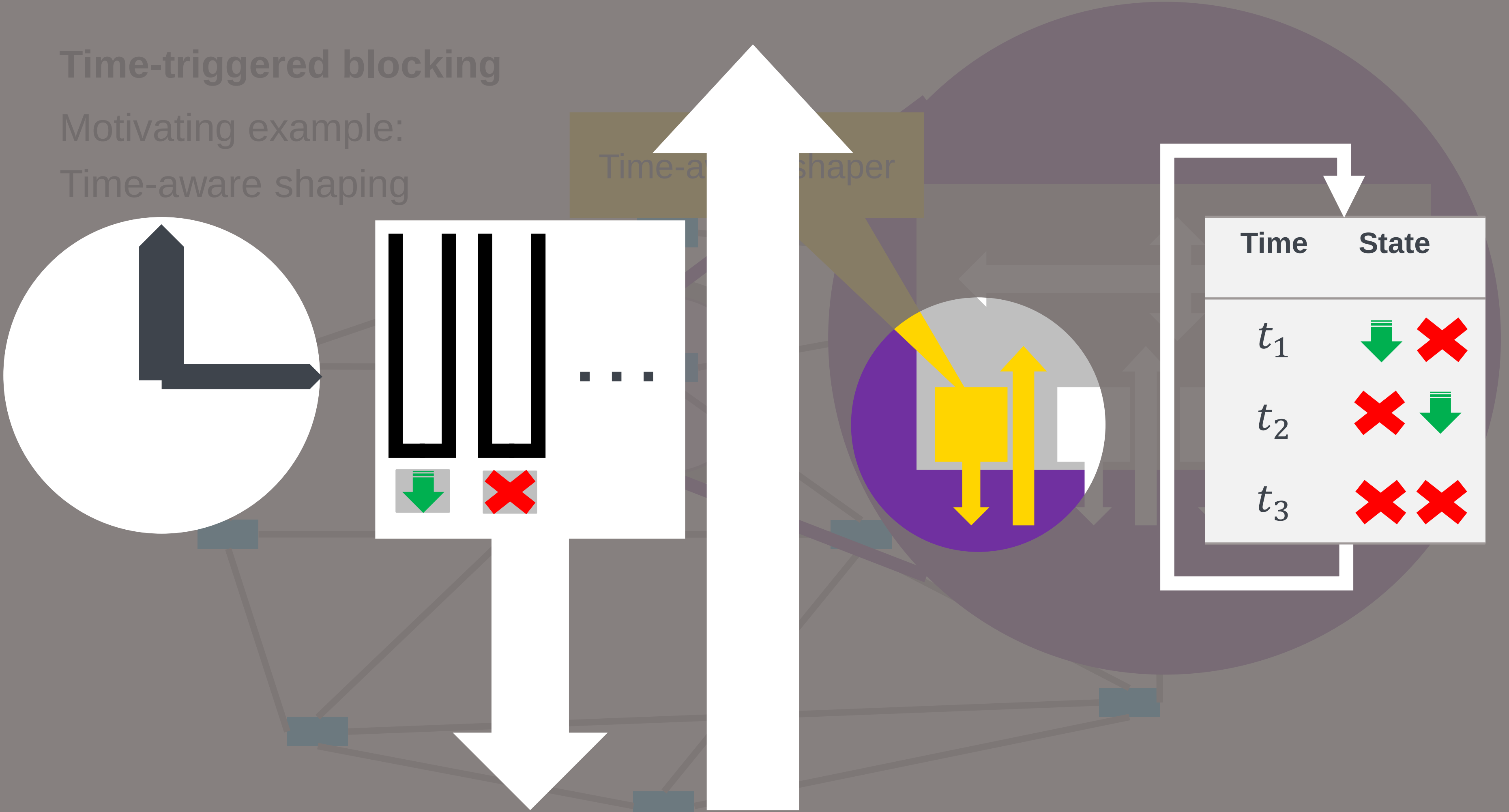
Time-aware shaping



Time-triggered blocking

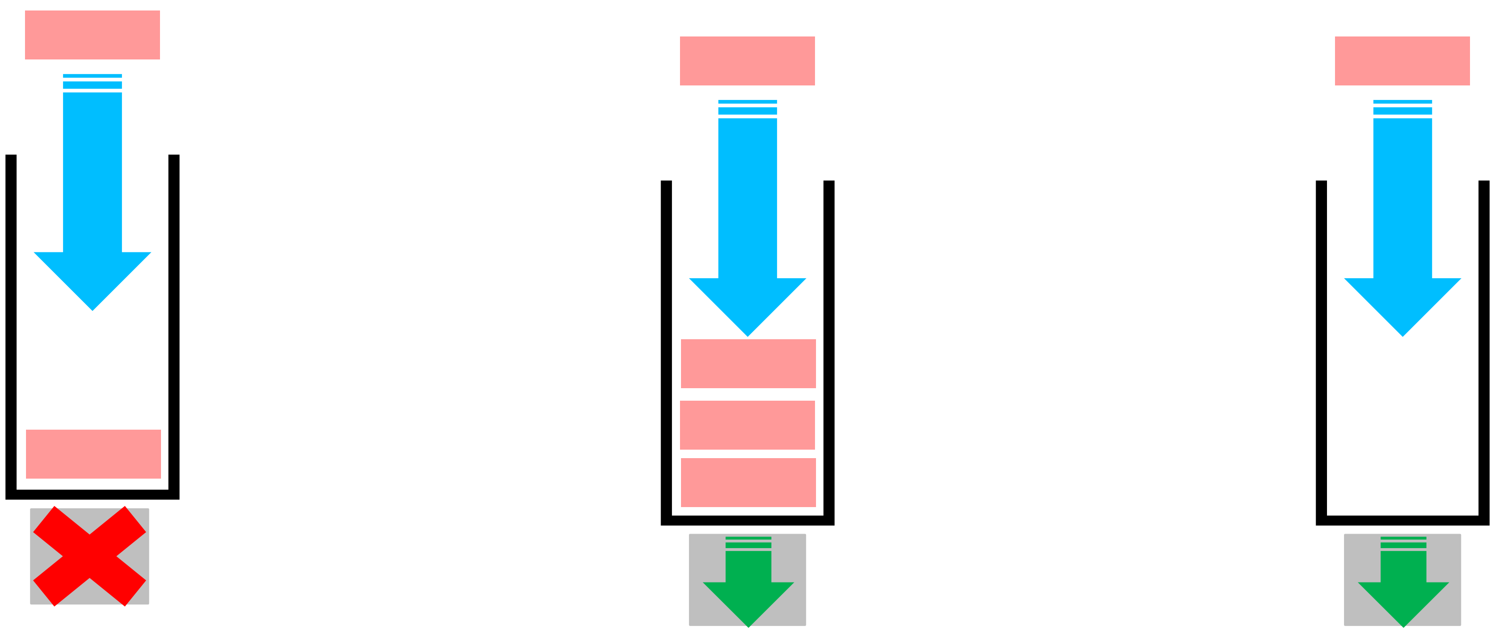
Motivating example:

Time-aware shaping



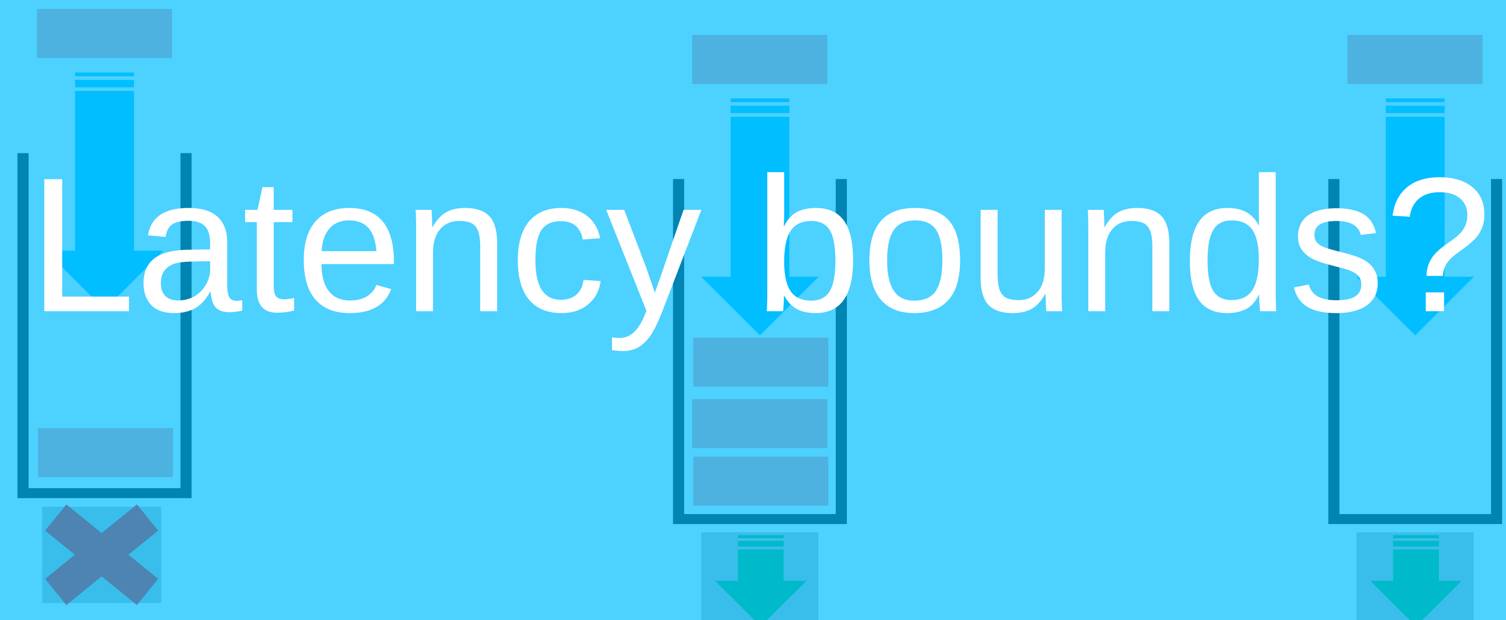
Time-triggered blocking

Data transmission and network elements are **not synchronized.**



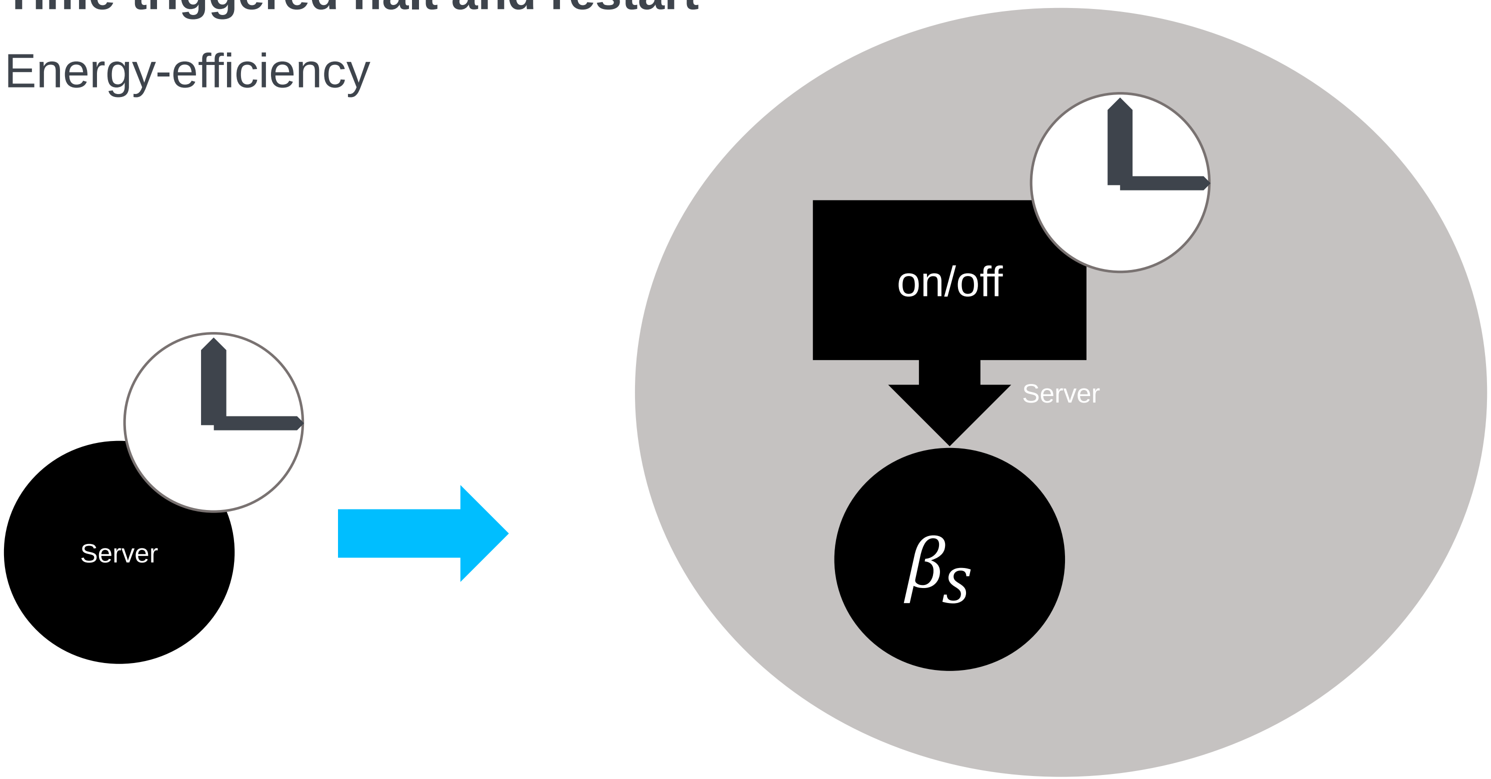
Time-triggered blocking

Data transmission and network elements are **not synchronized.**



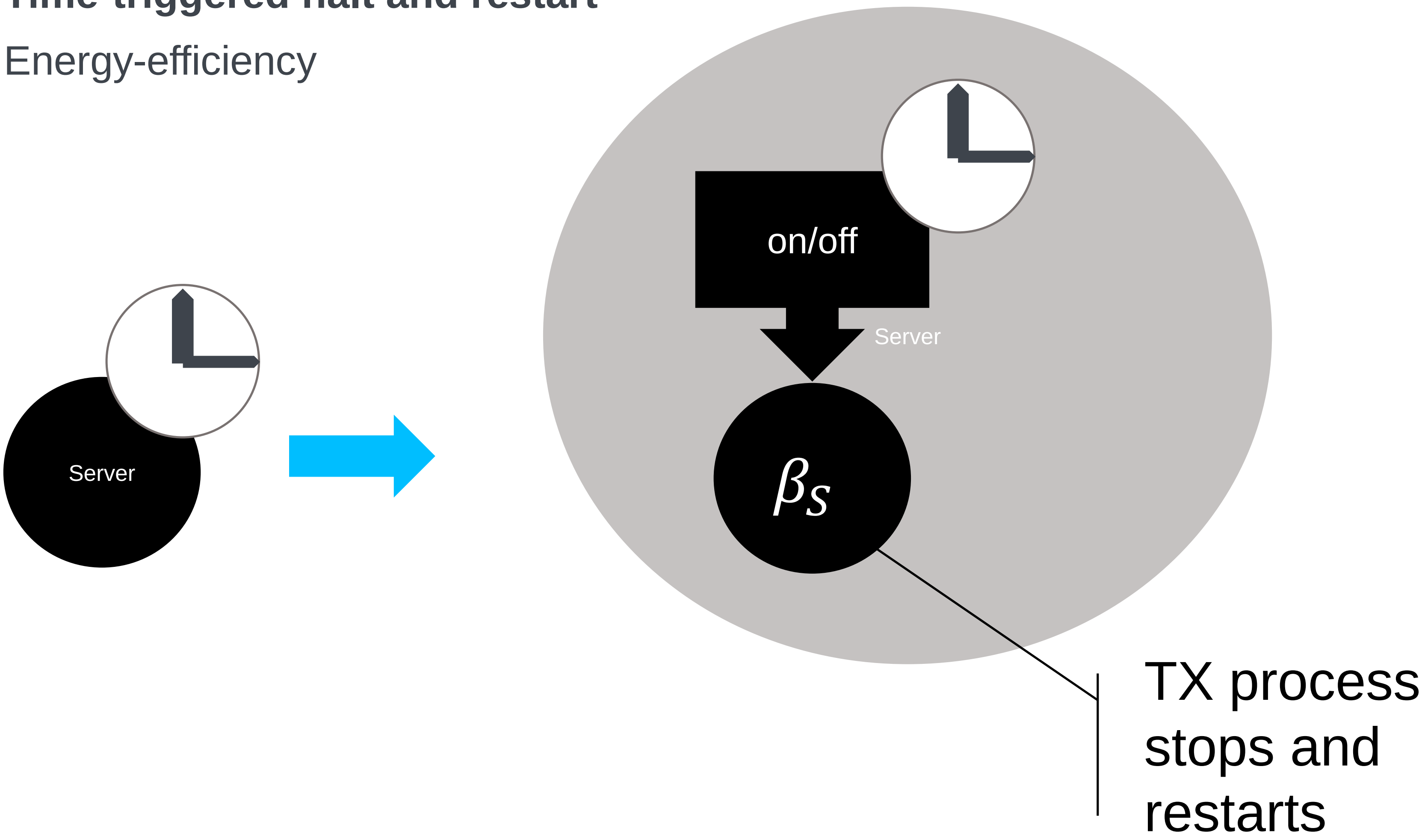
Time-triggered halt and restart

Energy-efficiency



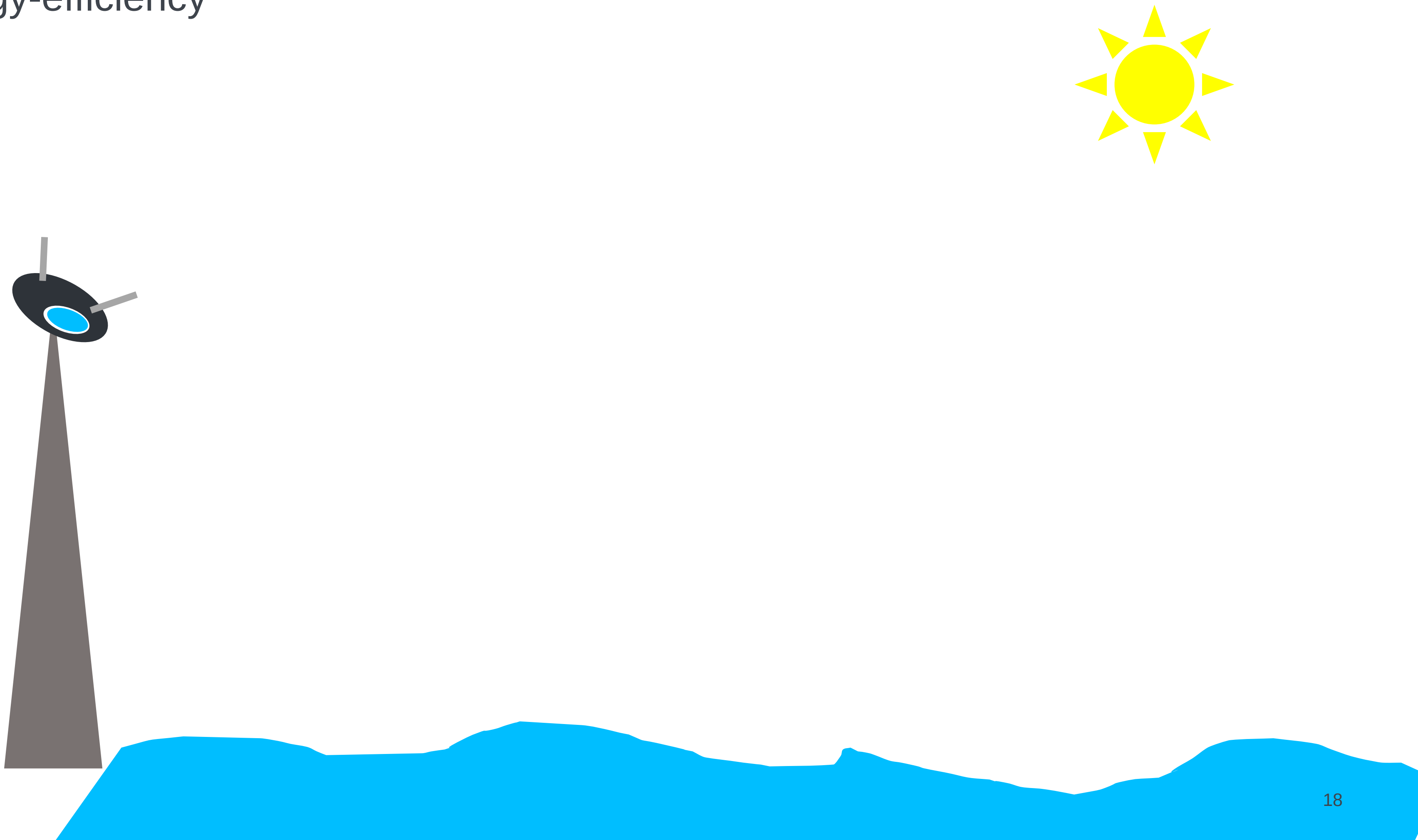
Time-triggered halt and restart

Energy-efficiency



Time-triggered halt and restart

Energy-efficiency

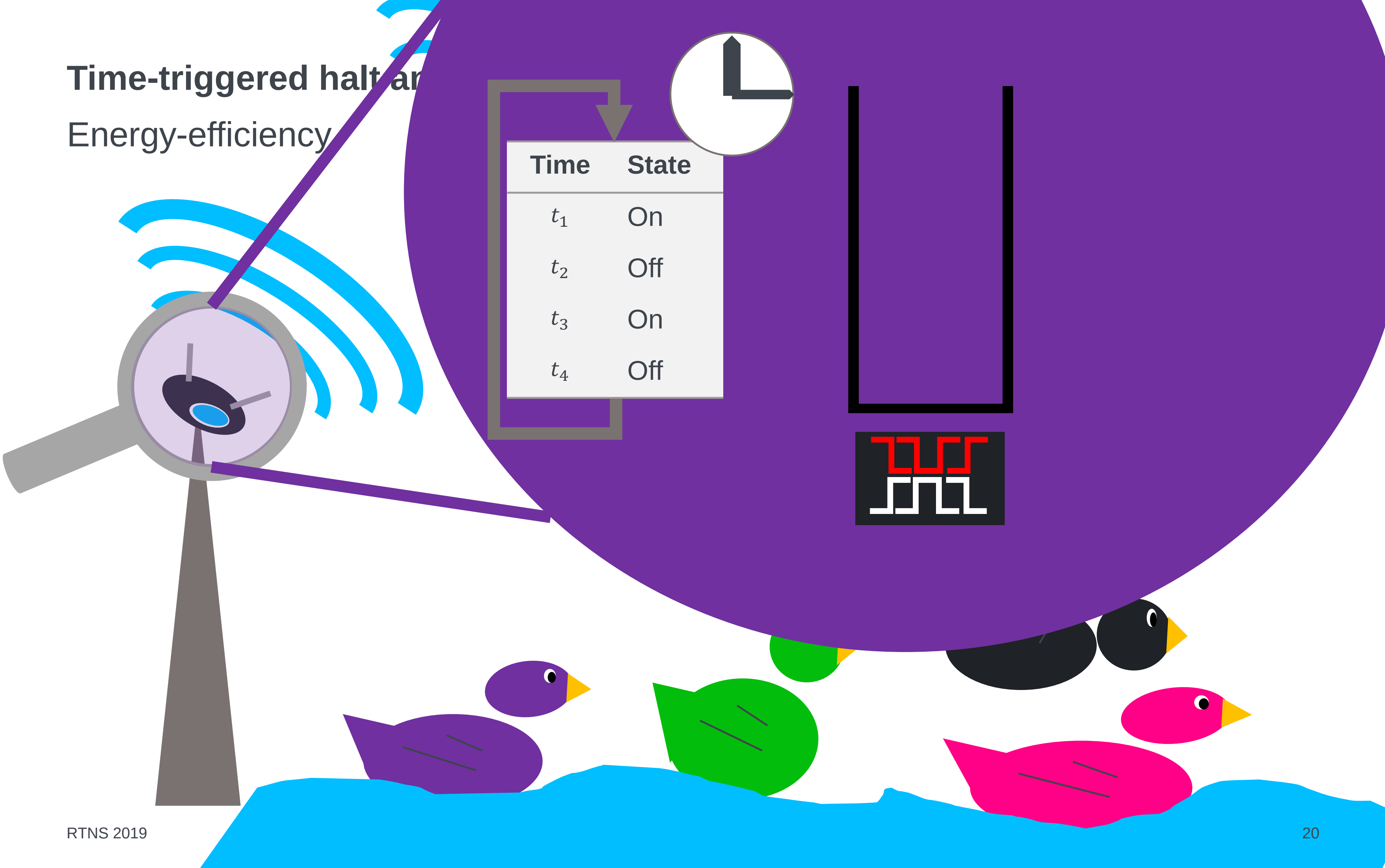


Time-triggered halt and restart

Energy-efficiency

Time-triggered halt an

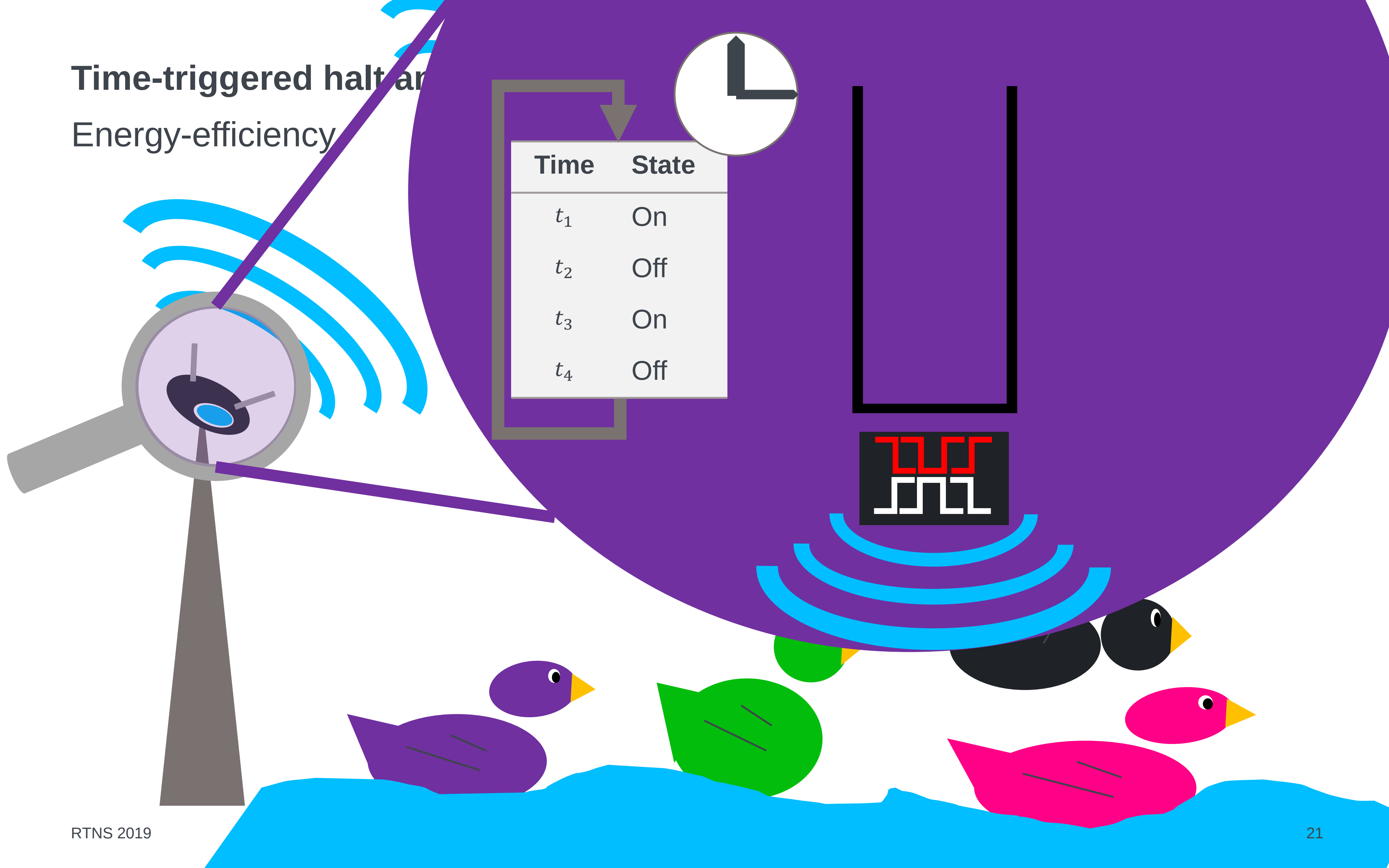
Energy-efficiency



Time	State
t_1	On
t_2	Off
t_3	On
t_4	Off

Time-triggered halt a

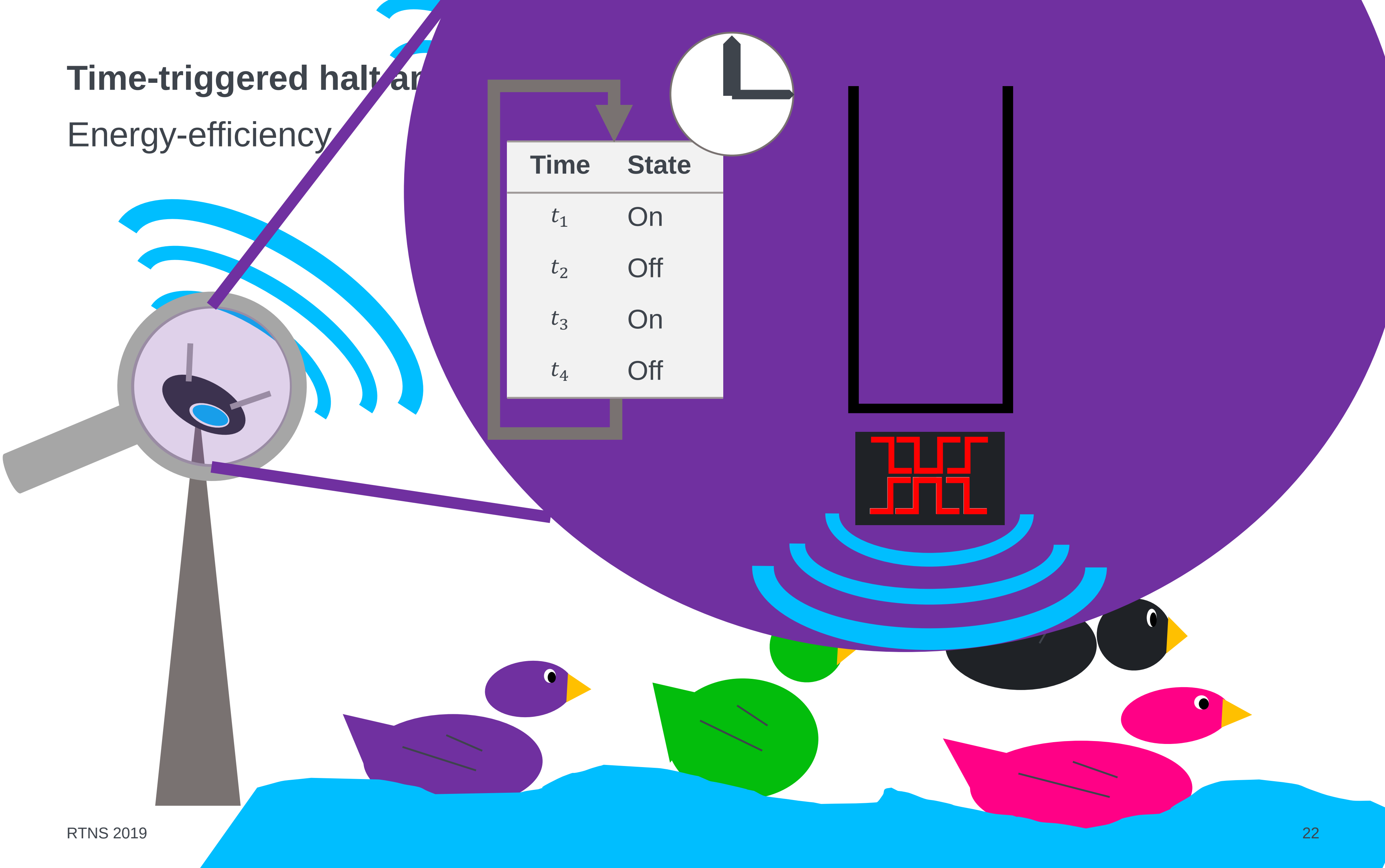
Energy-efficiency



Time	State
t_1	On
t_2	Off
t_3	On
t_4	Off

Time-triggered halt ar

Energy-efficiency

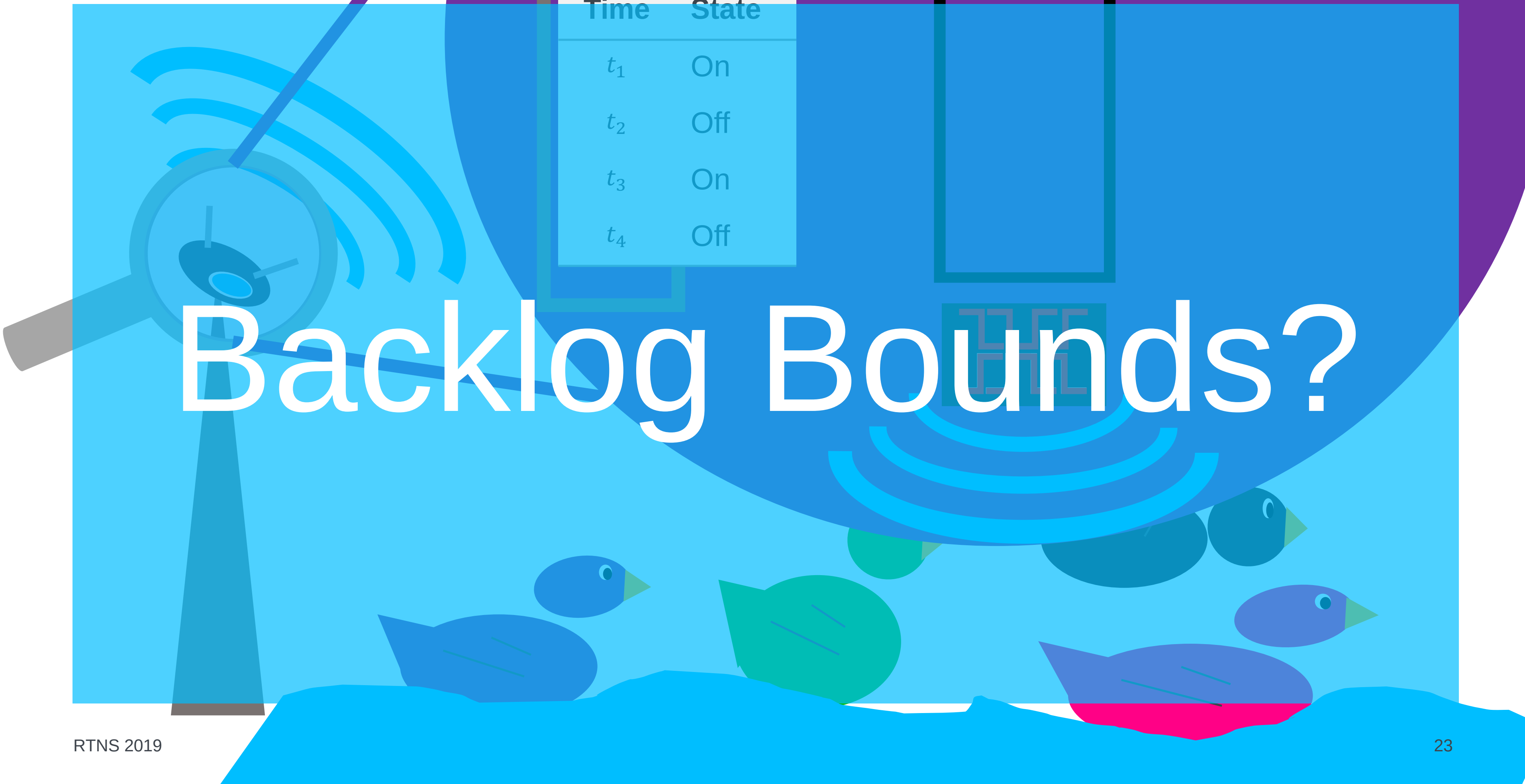


Time	State
t_1	On
t_2	Off
t_3	On
t_4	Off

Time-triggered halt an

Energy-efficiency

Time	State
t_1	On
t_2	Off
t_3	On
t_4	Off



Backlog Bounds?

Related work

- TDMA
  - Gollan, N., and J. Schmitt. “Energy-Efficent TDMA Design Under Real-Time Constraints in Wireless Sensor Networks.” In Proceedings of the 2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 80–87. MASCOTS ’07. Washington, DC, USA.
  - Dang Dinh Khanh, and Ahlem Mifdaoui. “Timing Analysis of TDMA-Based Networks Using Network Calculus and Integer Linear Programming.” In 2014 IEEE 22nd International Symposium on Modelling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 21–30. Paris, France.
- Ethernet
  - Zhao, Luxi, Paul Pop, Qiao Li, Junyan Chen, and Huagang Xiong. “Timing Analysis of Rate-Constrained Traffic in TTEthernet Using Network Calculus.” Real-Time Systems 53, no. 2 (March 1, 2017): 254–87.
  - Zhao, L., P. Pop, and S. S. Craciunas. “Worst-Case Latency Analysis for IEEE 802.1Qbv Time Sensitive Networks Using Network Calculus.” IEEE Access 6 (2018): 41803–15.
  - Zhao, L., P. Pop, Z. Zheng, and Q. Li. “Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus.” In 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 25–36, 2018.

Related work

simple schedules

- TDMA
  - Gollan, N., and J. Schmitt. “Energy-Efficent TDMA De
    2007 15th International Symposium on Modeling, Ana
    MASCOTS ’07. Washington, DC, USA.
  - Dang Dinh Khanh, and Ahlem Mifdaoui. “Timing Analy
    Programming.” In 2014 IEEE 22nd International Symp
    Systems (MASCOTS), 21–30. Paris, France.
- Ethernet
  - Zhao, Luxi, Paul Pop, Qiao Li, Junyan Chen, and Huagang Xiong. “Timing Analysis of Rate-Constrained Traffic in TTEthernet Using Network Calculus.” Real-Time Systems 53, no. 2 (March 1, 2017): 254–87.
  - Zhao, L., P. Pop, and S. S. Craciunas. “Worst-Case Latency Analysis for IEEE 802.1Qbv Time Sensitive Networks Using Network Calculus.” IEEE Access 6 (2018): 41803–15.
  - Zhao, L., P. Pop, Z. Zheng, and Q. Li. “Timing Analysis of AVB Traffic in TSN Networks Using Network Calculus.” In 2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), 25–36, 2018.

Related work

- TDMA
  - Gollan, N., and J. Schmitt. “Energy-Efficent TDMA De
    2007 15th International Symposium on Modeling, Ana
    MASCOTS ’07. Washington, DC, USA.
  - Dang Dinh Khanh, and Ahlem Mifdaoui. “Timing Analy
    Programming.” In 2014 IEEE 22nd International Symp
    Systems (MASCOTS), 21–30. Paris, France.
- Ethernet

simple schedules

Analysis of Rate-Constrained Traffic in TTEthernet Using Network

EE 802.1Qbv Time Sensitive Networks Using Network Calculus.”

N Networks Using Network Calculus.” In 2018 IEEE Real-Time
018.

limited scope

Related work

- TDMA
  - Gollan, N., and J. Schmitt. “Energy-Efficent TDMA De
    2007 15th International Symposium on Modeling, Ana
    MASCOTS ’07. Washington, DC, USA.
  - Dang Dinh Khanh, and Ahlem Mifdaoui. “Timing Analy
    Programming.” In 2014 IEEE 22nd International Symp
    Systems (MASCOTS), 21–30. Paris, France.
- Ethernet

simple schedules

time-invariant

limited scope

Related work

simple schedules

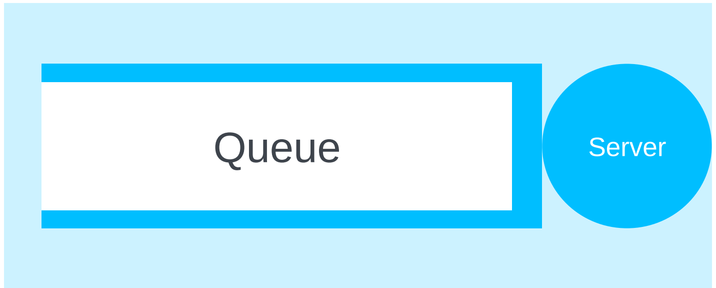
ultimately constant-rate service curves*

time-invariant

limited scope

Network calculus

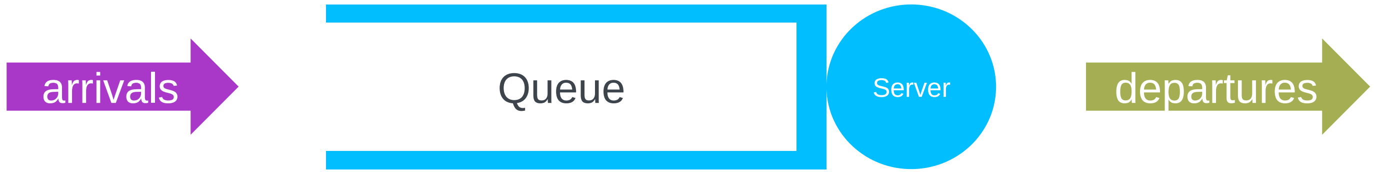
- “System theory” for queuing systems
  - Modularity (cf. convolution operation)
  - Deterministic (guaranteed) bounds

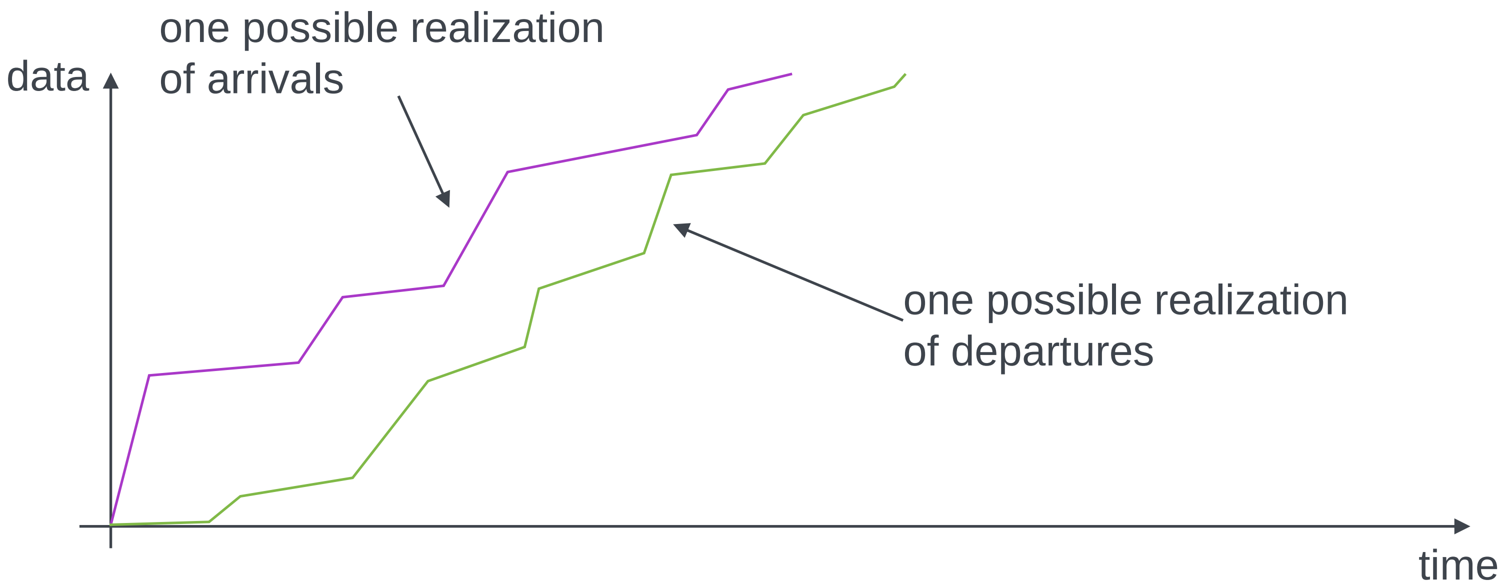


- NC allows to compute **deterministic bounds**:
  - (virtual) delay bound
  - backlog bounds
  - departure bound

Network calculus

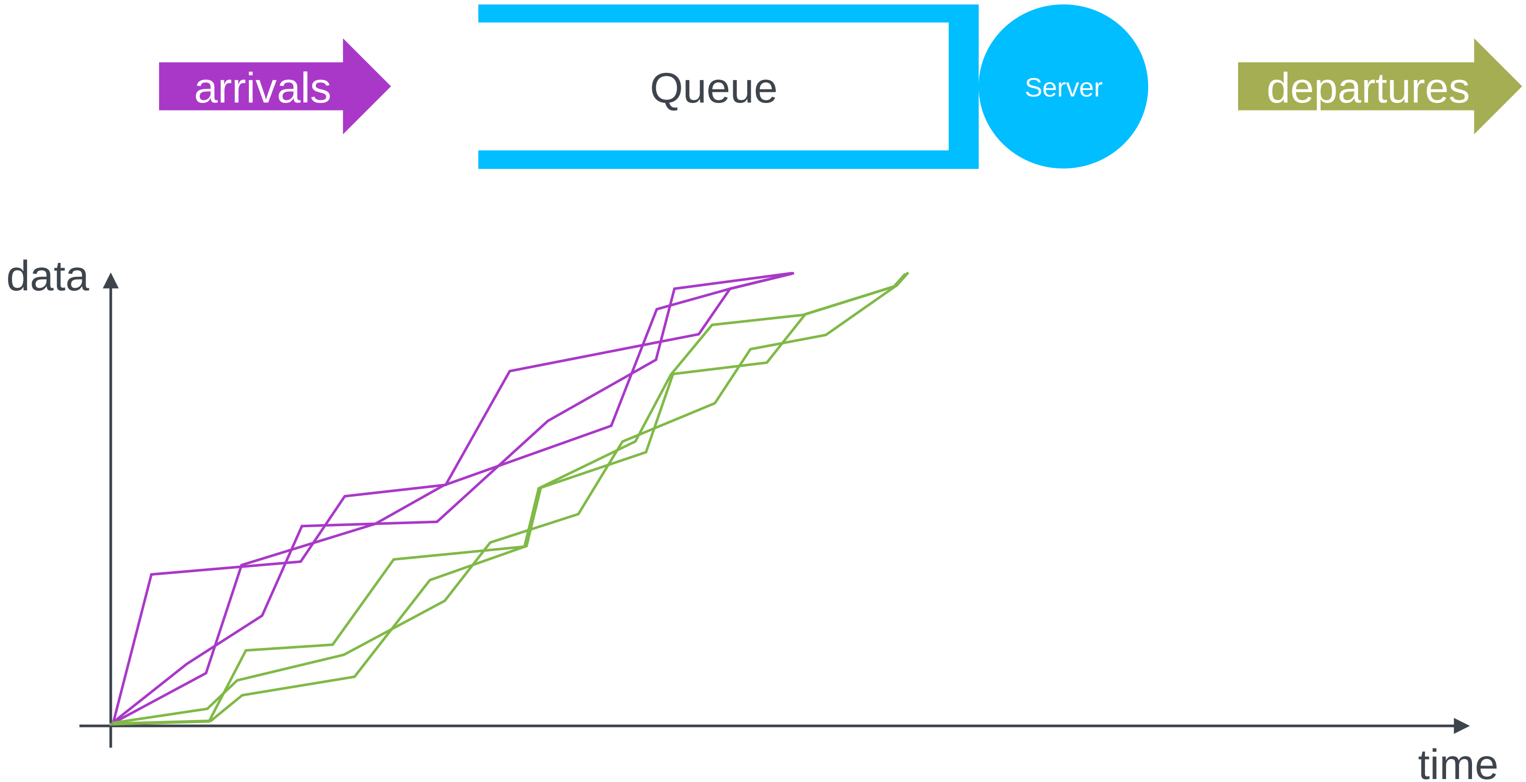
Operating on cumulative curves.





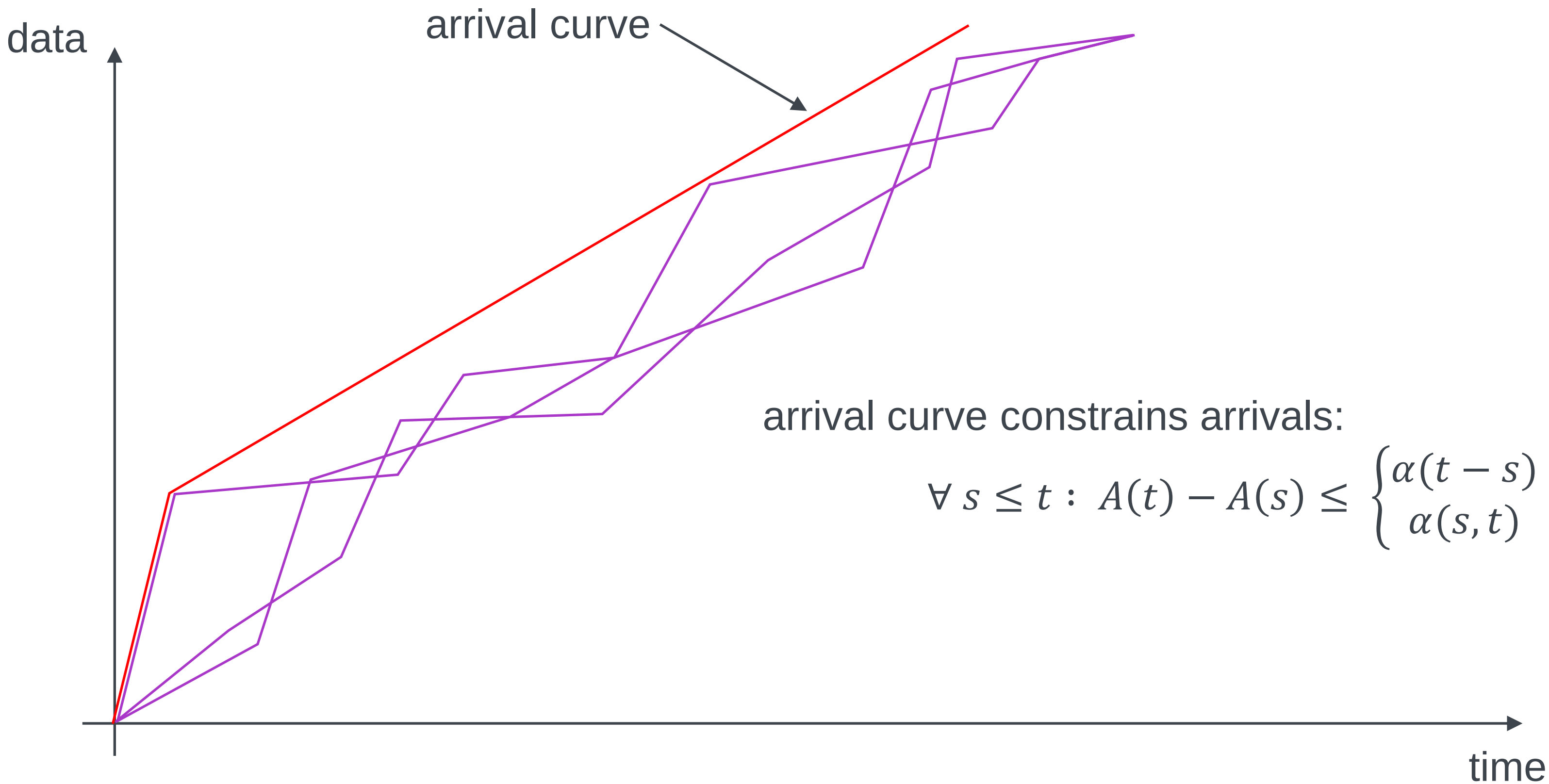
Network calculus

Consider the worst-case (deterministic network calculus).



Network calculus

Arrival curve (models data entering the system).

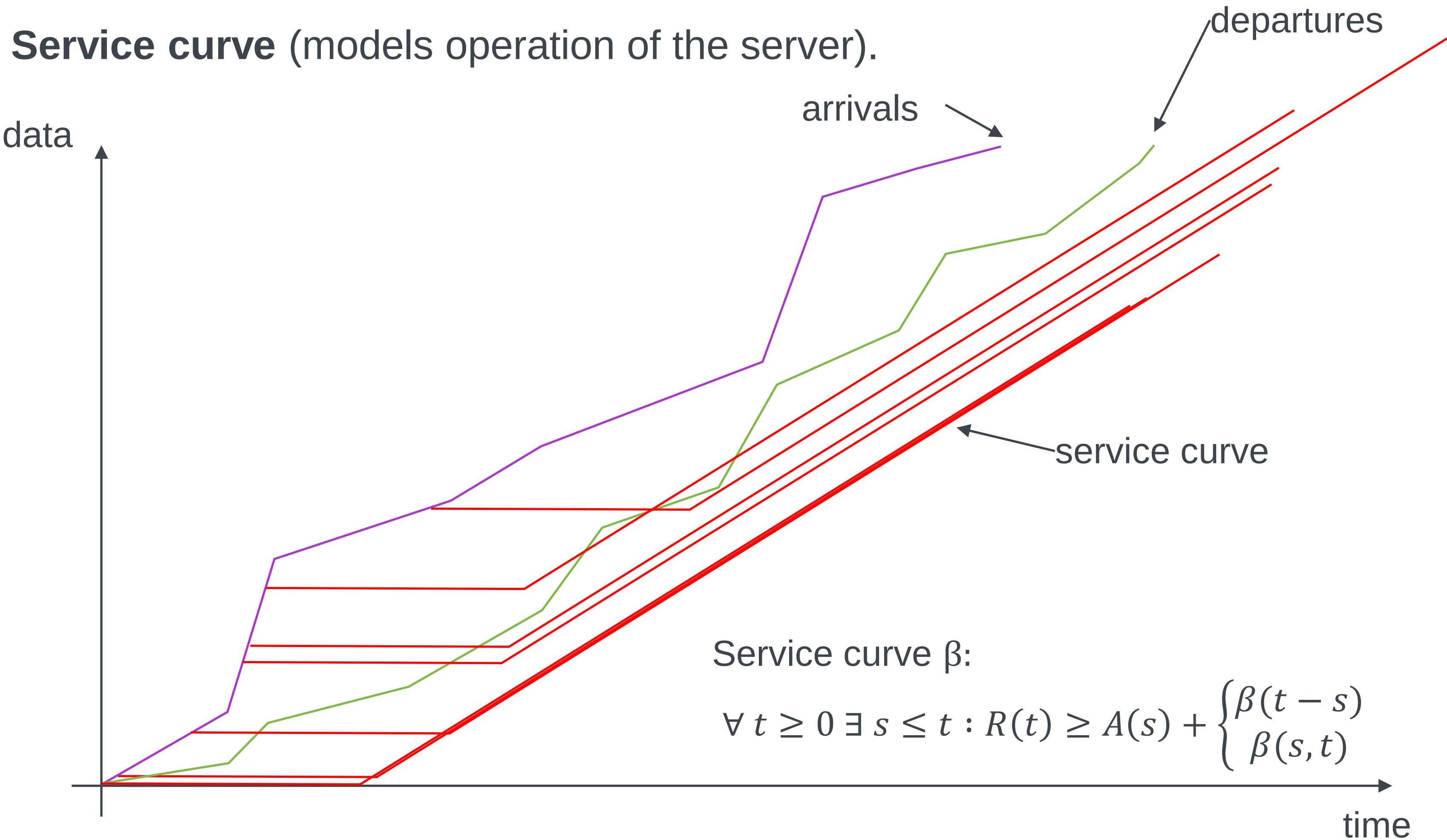


arrival curve constrains arrivals:

$$\forall\, s \leq t:\; A(t) - A(s) \leq \begin{cases} \alpha(t-s) \\ \alpha(s,t) \end{cases}$$

Network calculus

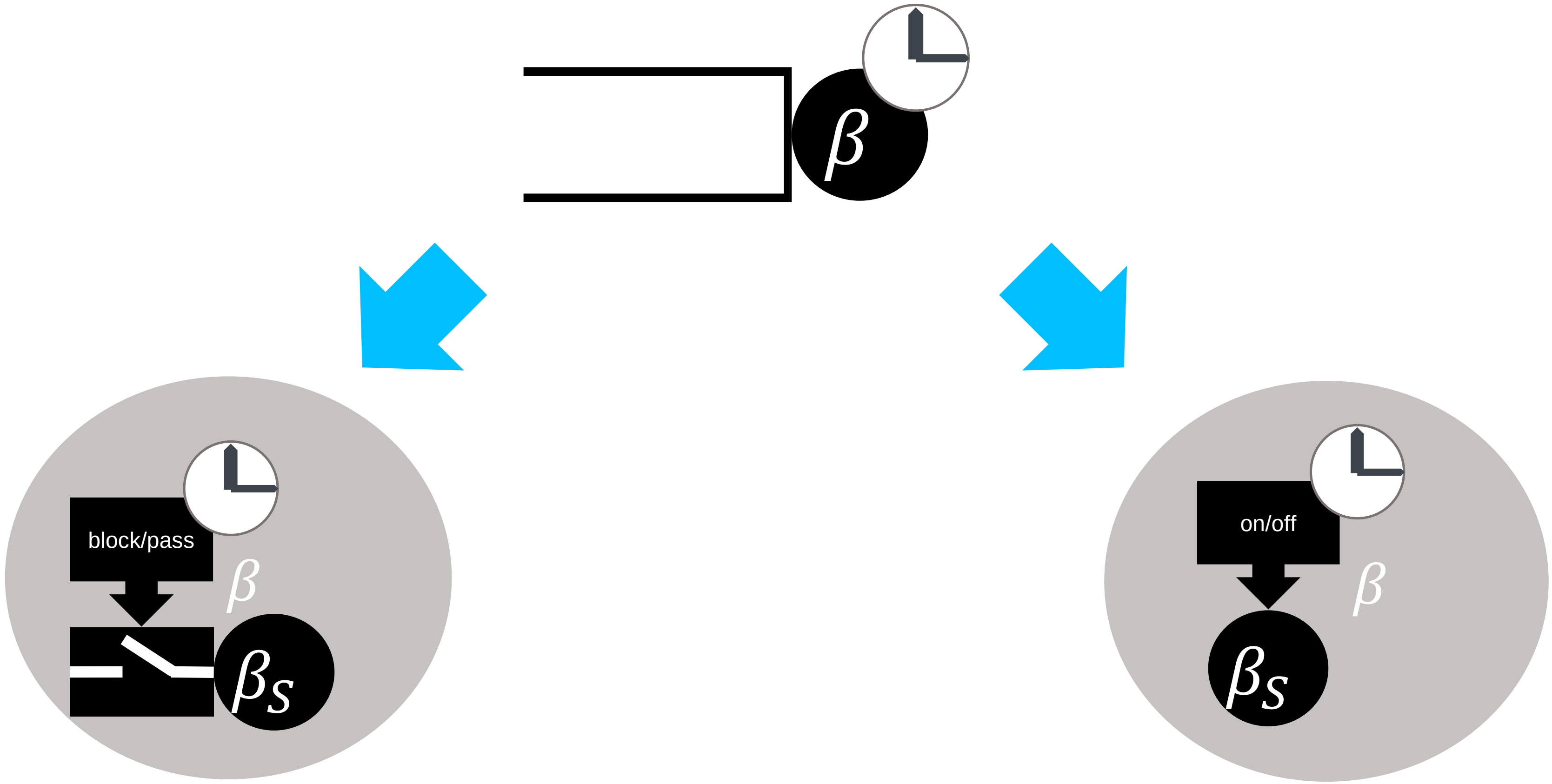
Service curve (models operation of the server).



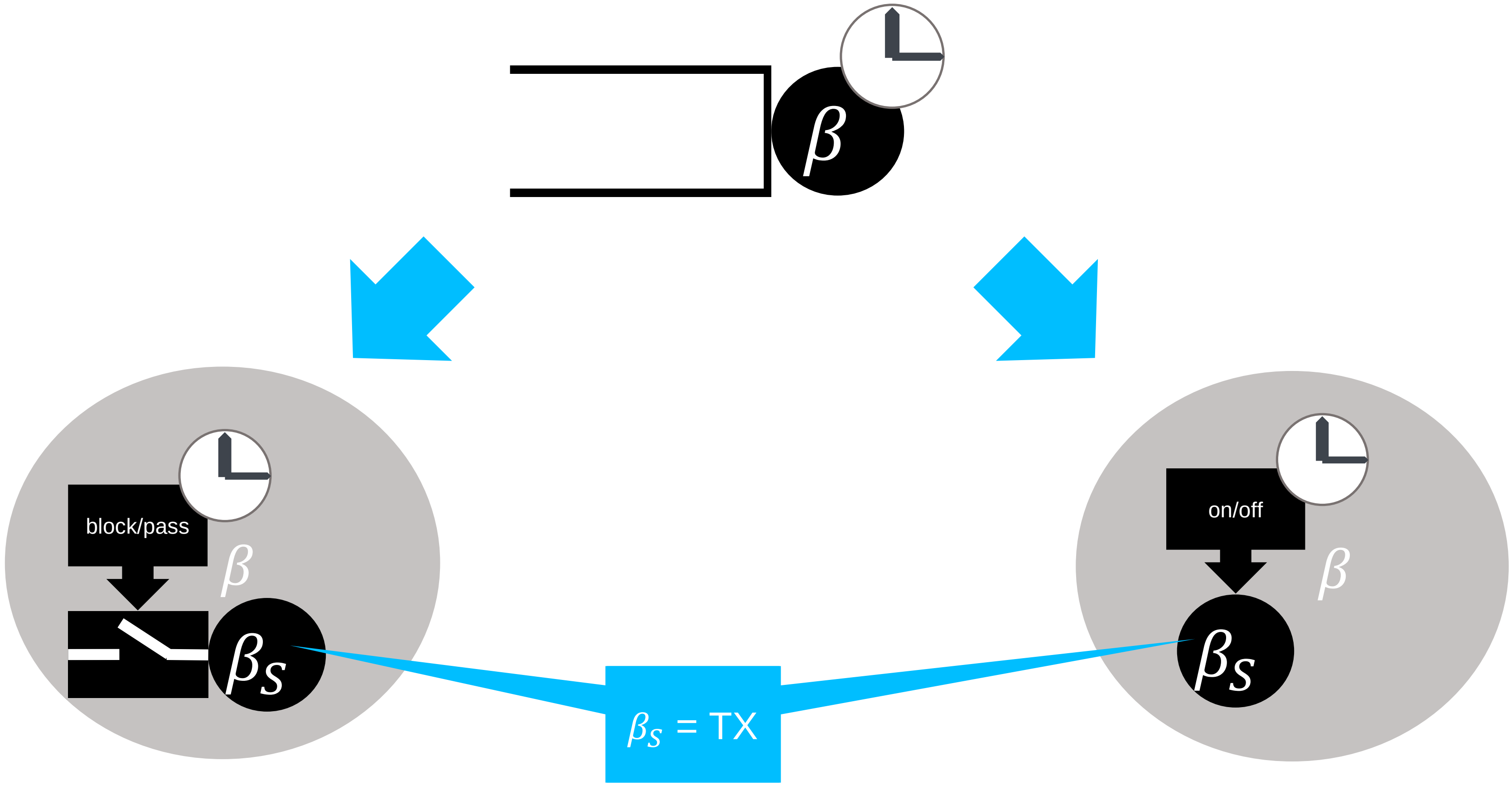
Service curve β:

$$\forall\, t \geq 0\ \exists\, s \leq t : R(t) \geq A(s) + \begin{cases} \beta(t-s) \\ \beta(s,t) \end{cases}$$

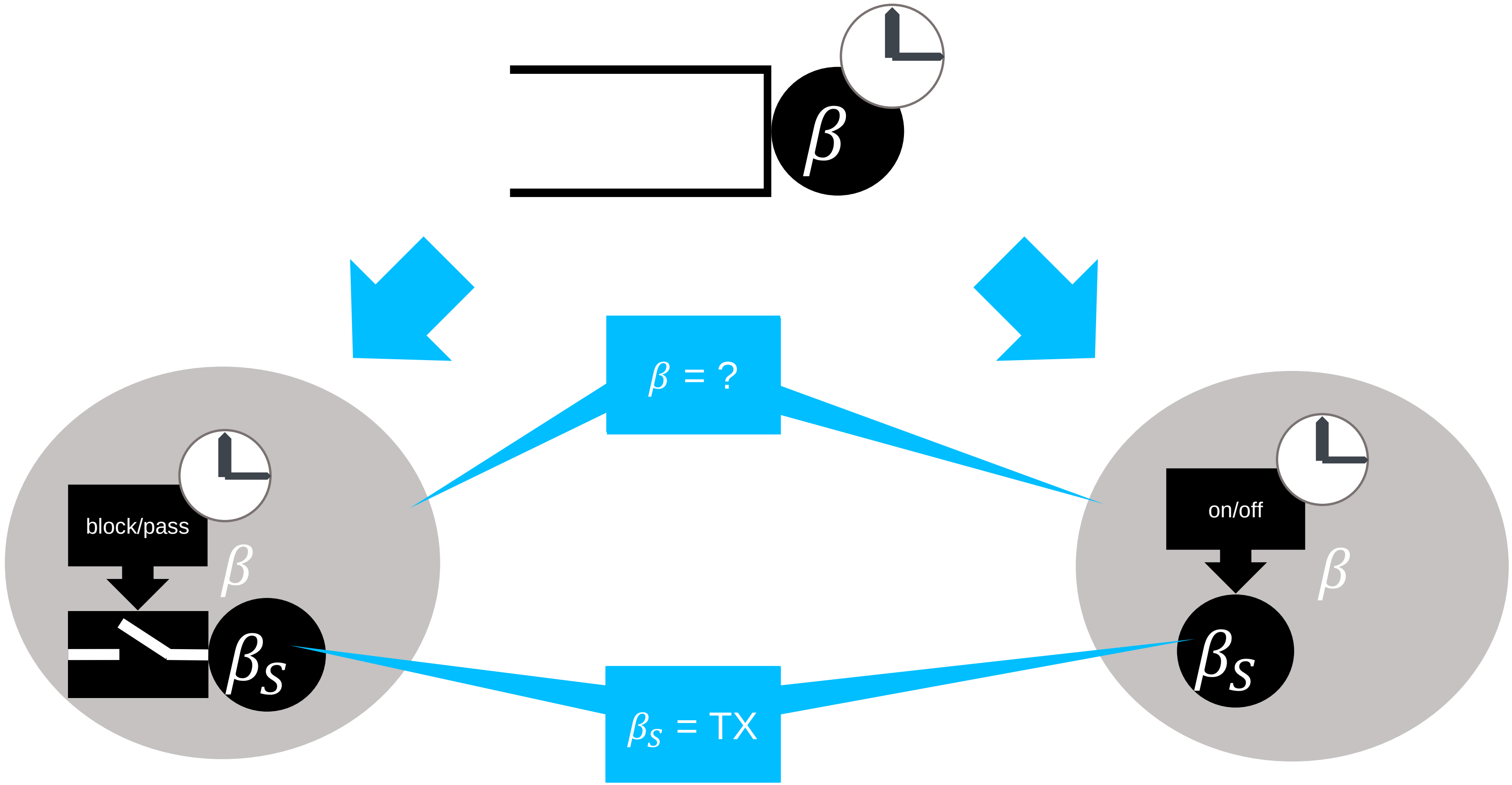
How to model services curves?



How to model services curves?



How to model services curves?



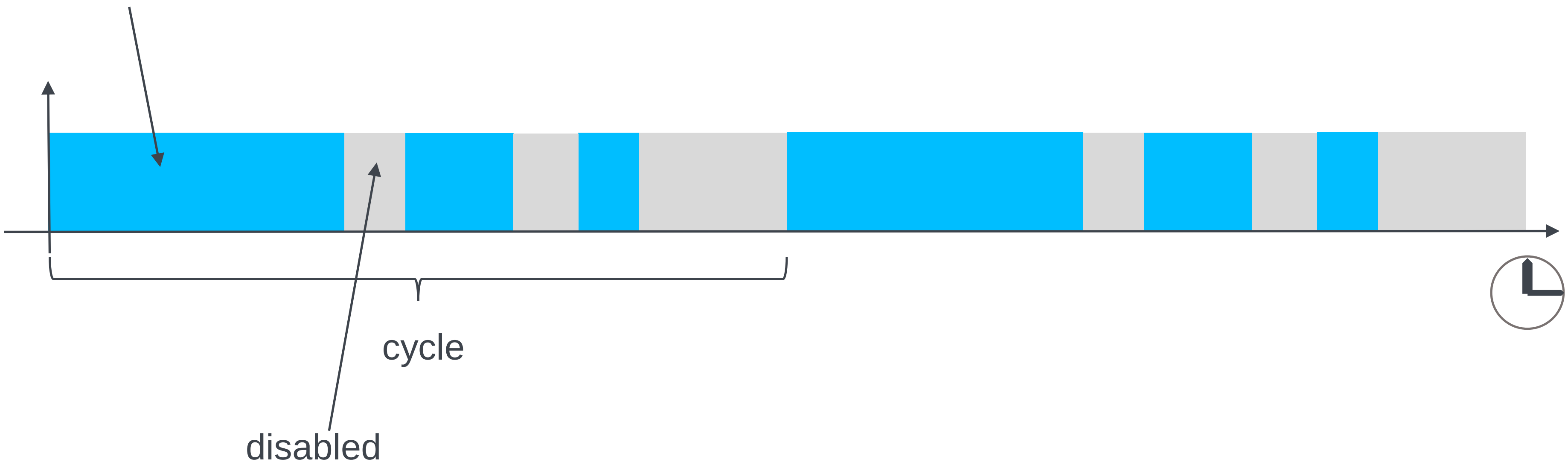
Time-variant service curve

Observation: “enabled” intervals and “disabled” intervals

enabled:

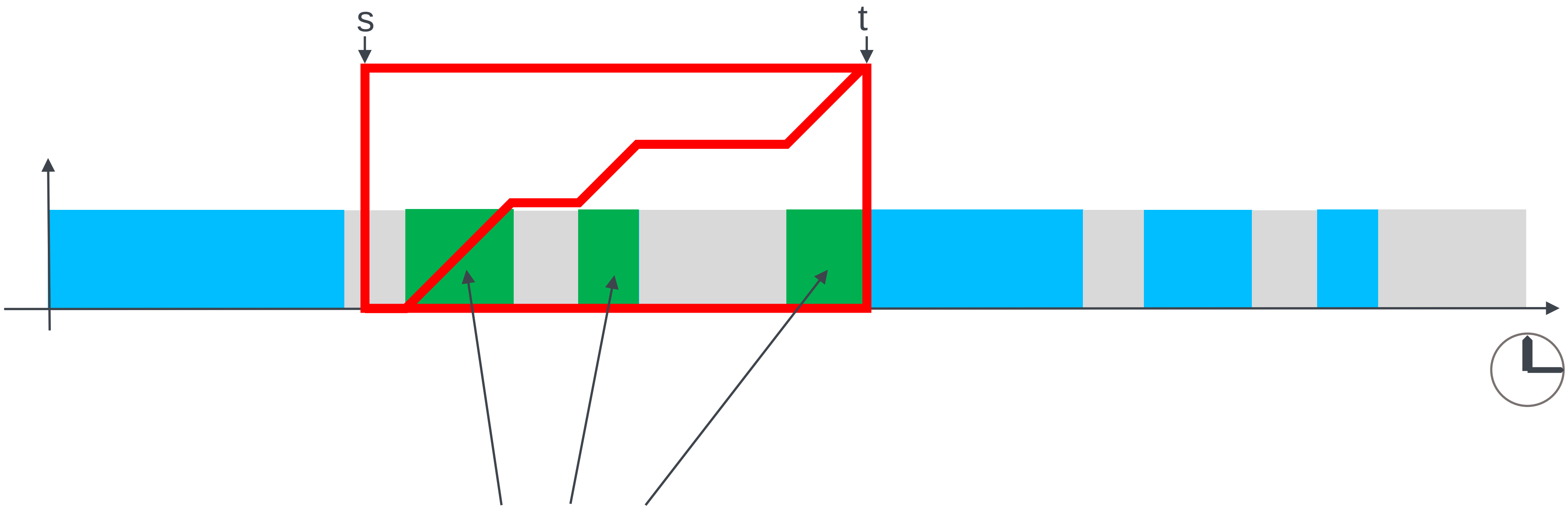
Time-triggered blocking: data is being passed through

Time-triggered halt and restart: server is on



Time-variant service curve

Service in interval (s,t]?



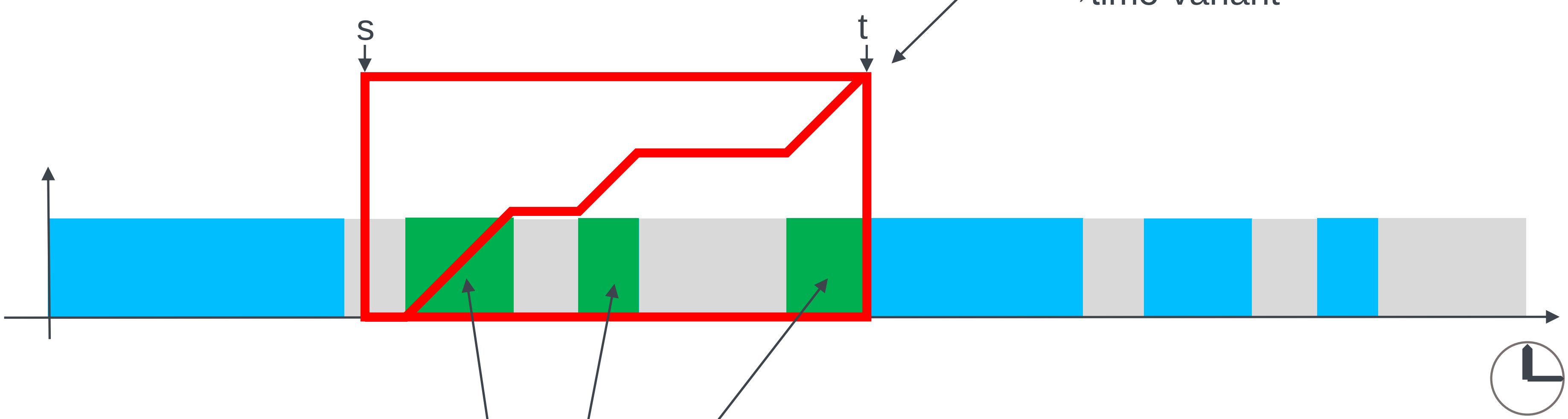
contribute to the total amount of
service offered in (s,t]

Time-variant service curve

Service in interval (s,t]?

Service offered in interval depends on “position” in the schedule

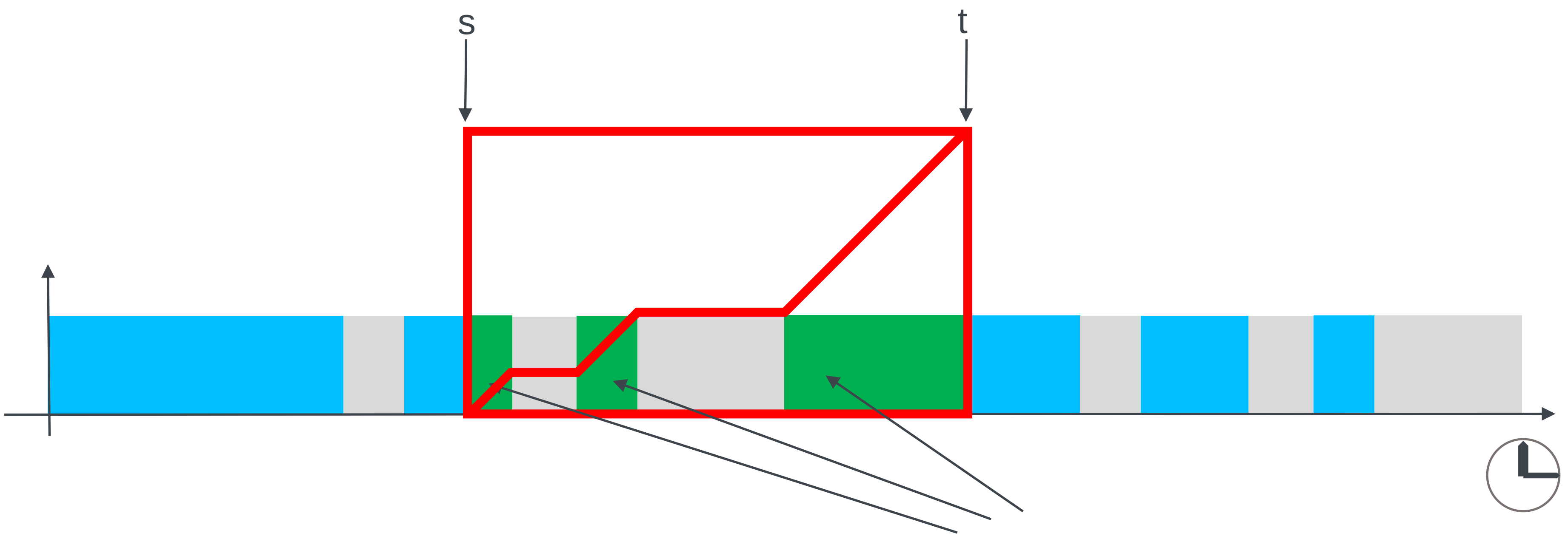
→time-variant



contribute to the total amount of service offered in (s,t]

Service curve formulation: idea

Service in interval (s,t]?



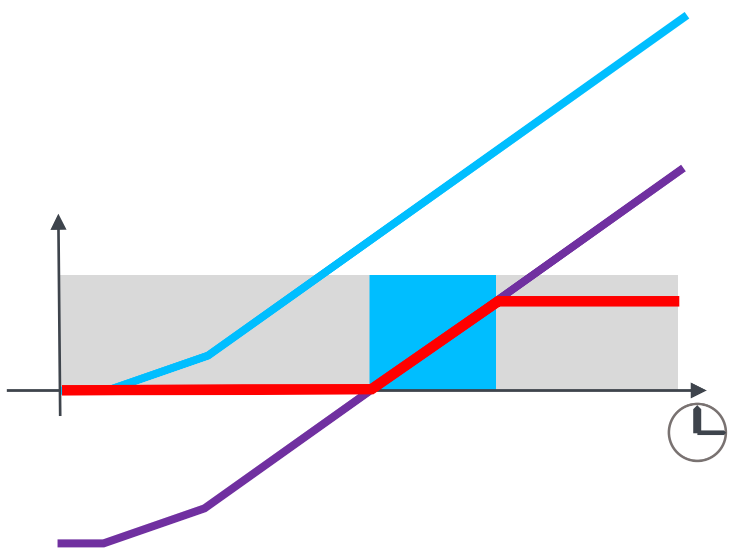
different cases for enabled intervals:

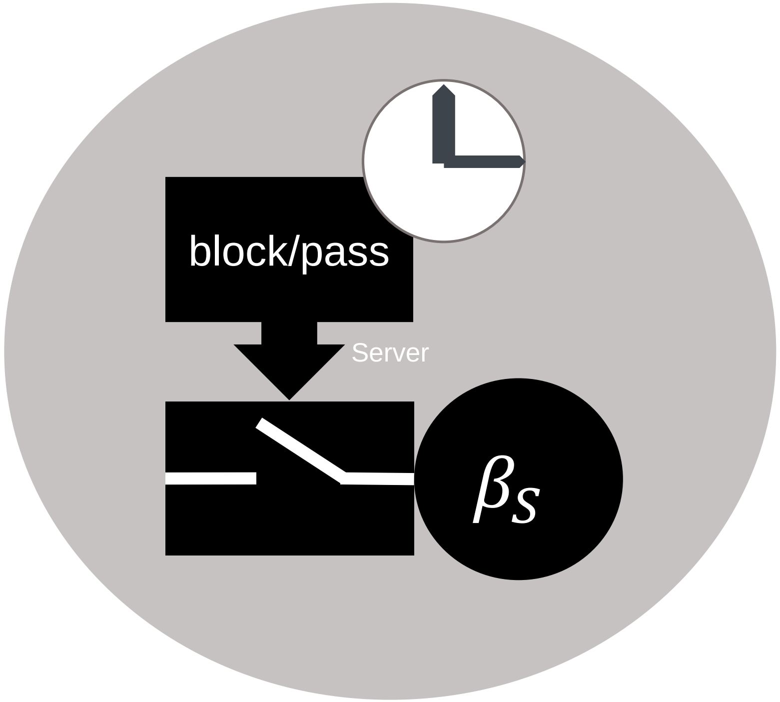
- entirely included / excluded
- partially covered

Time-variant service curve

Time-triggered blocking

$$\beta(s,t) = \sum_{n=0}^{\infty} \left[\beta_S(\min(t, t_{d,n})) - \beta_S(\max(s, t_{e,n})) \right]^{+}$$



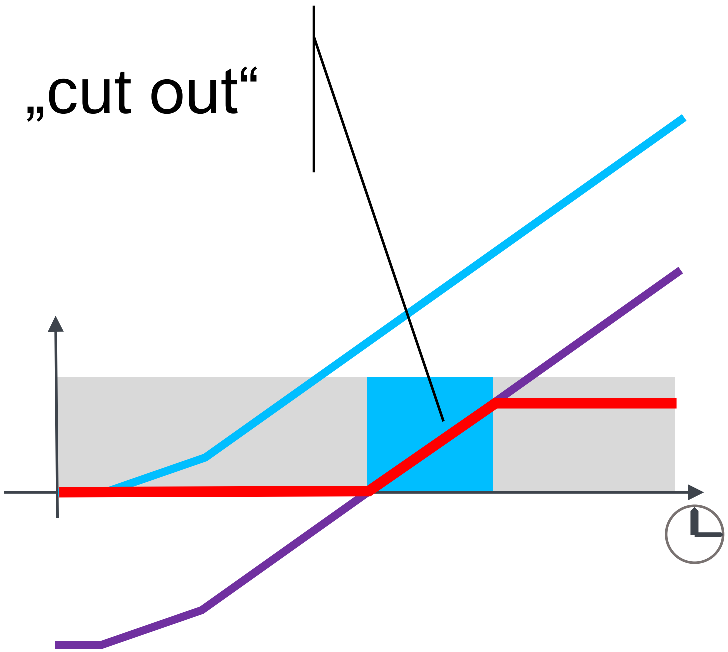


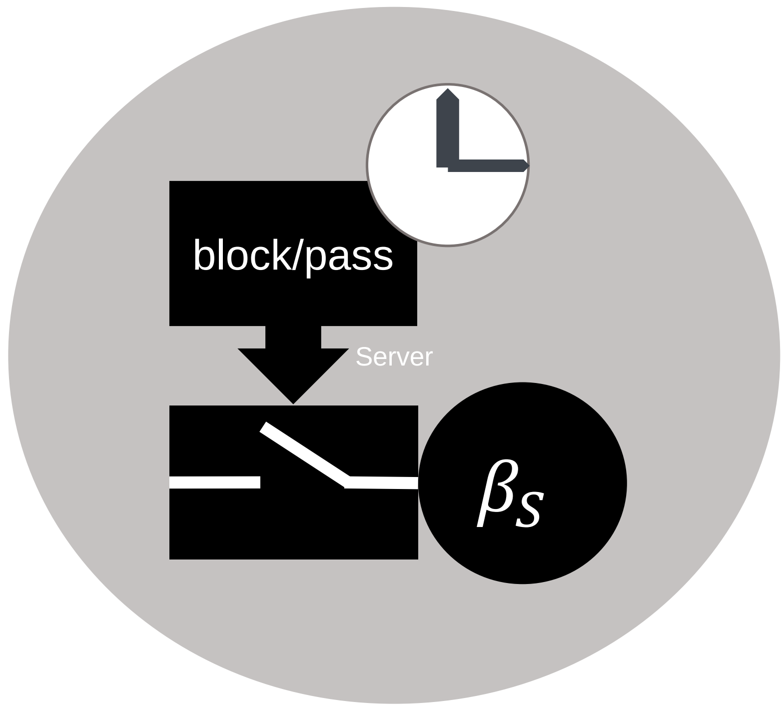
Time-variant service curve

Time-triggered blocking

$$\beta(s,t) = \sum_{n=0}^{\infty} \left[\beta_S\left(\min(t, t_{d,n})\right) - \beta_S\left(\max(s, t_{e,n})\right)\right]^+$$

„cut out“

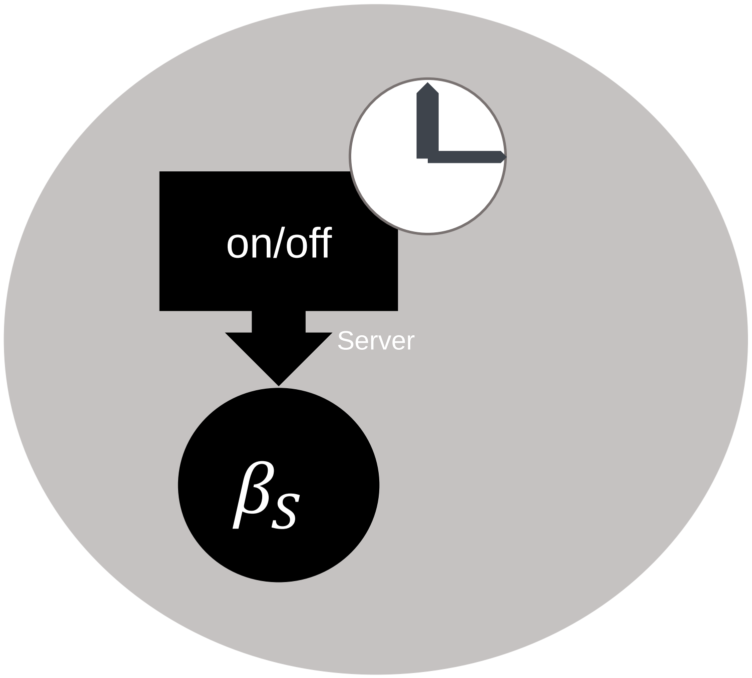


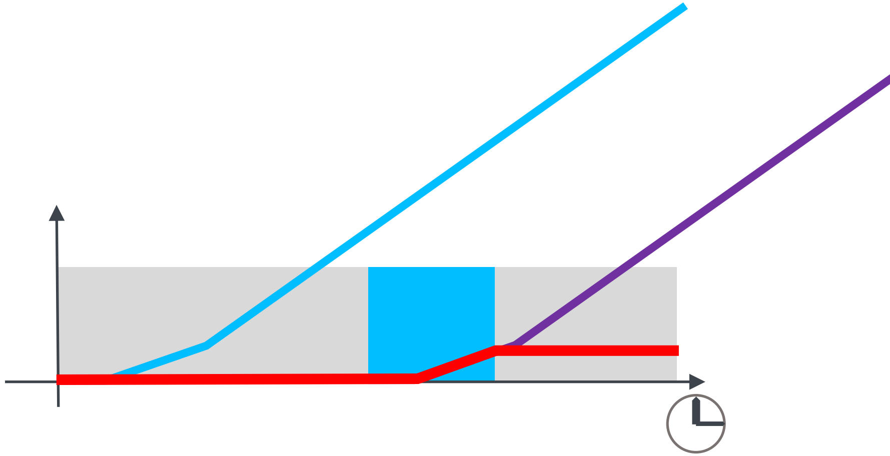


Time-variant service curve

Time-triggered halt and restart

$$\beta(s,t) = \sum_{n=0}^{\infty} \left[\beta_S\left(\min(t, t_{d,n}) - \max(s, t_{e,n})\right)\right]^+$$

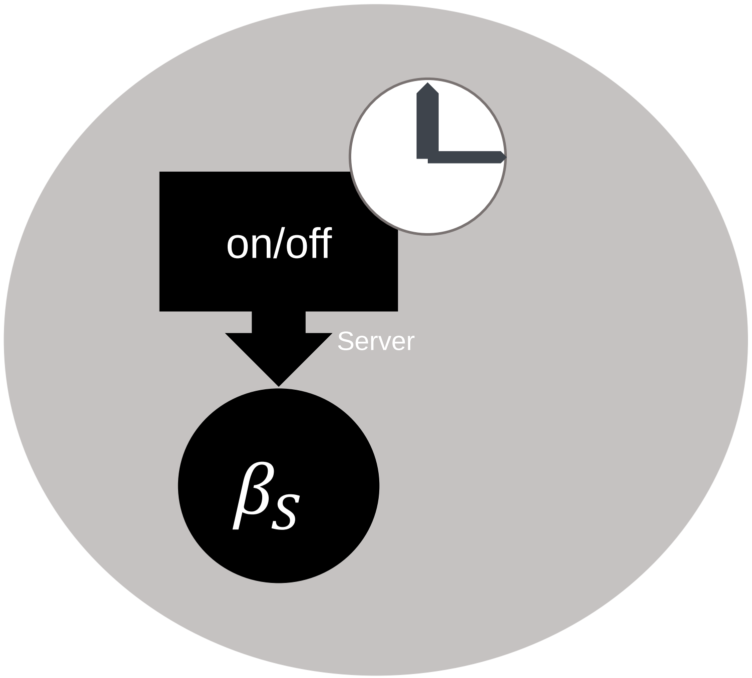


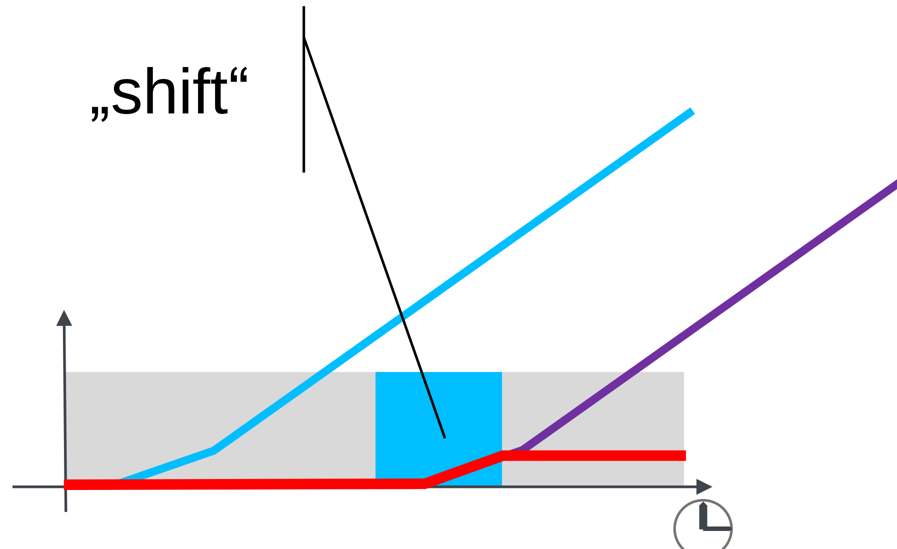


Time-variant service curve

Time-triggered halt and restart

$$\beta(s,t) = \sum_{n=0}^{\infty}\left[\beta_S\left(\min(t, t_{d,n}) - \max(s, t_{e,n})\right)\right]^+$$





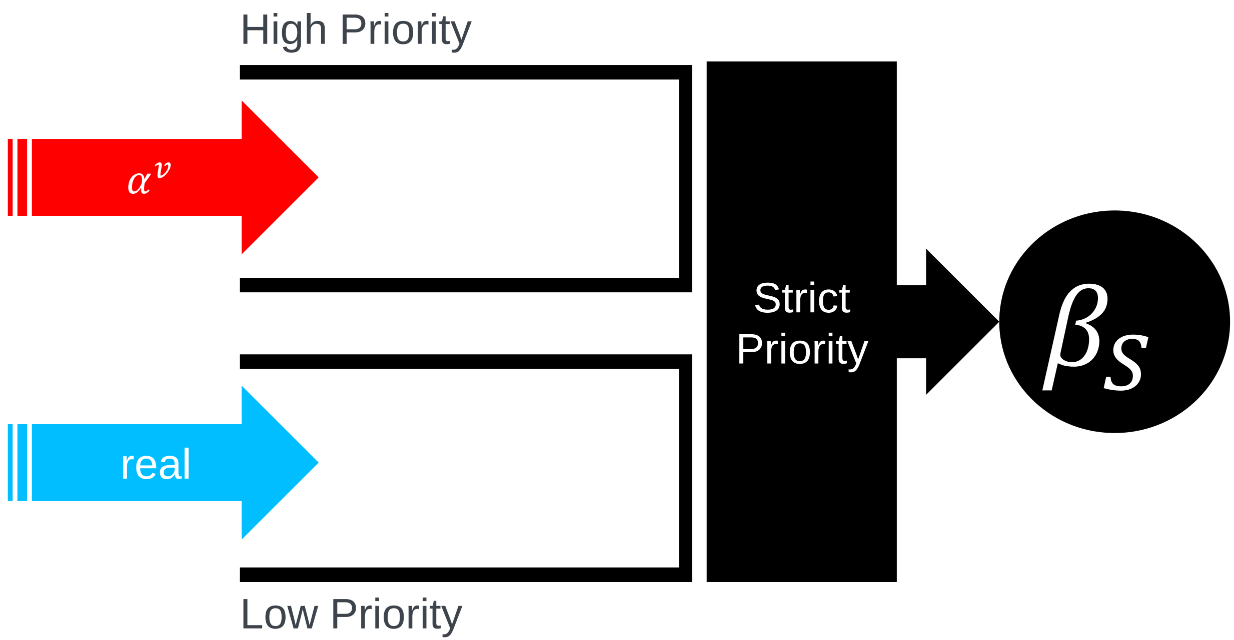
Time-invariant service curve

- Derived from time-variant service curve

- DNC for time-invariant functions
  - “less complicated”
  - computational support available (to some degree)

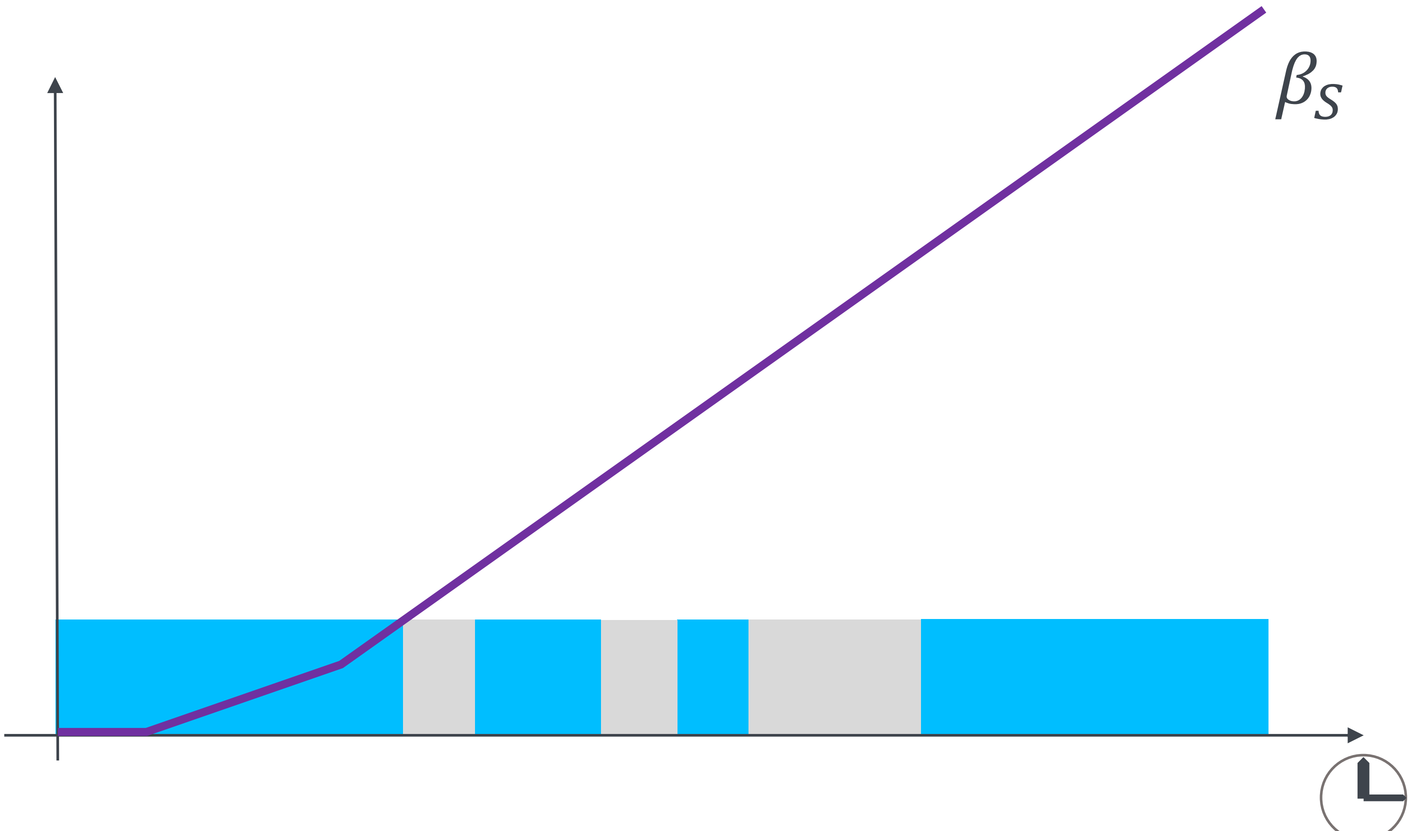
Time-invariant service curve

Leftover service curve: “virtual arrivals”



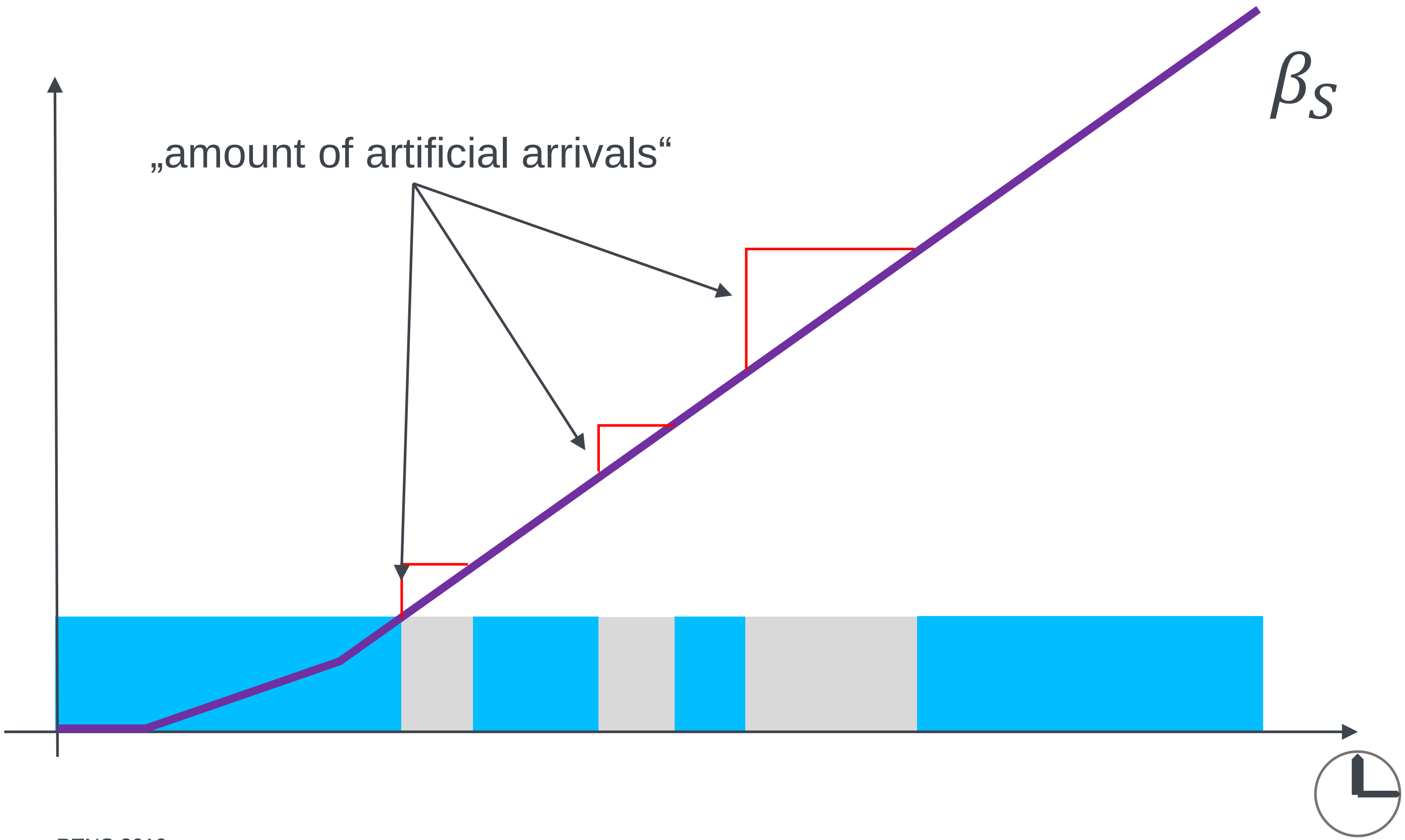
Time-invariant service curve

Leftover service curve: subtract virtual arrivals: $\beta(t) = \left[\sup_{s \le t}(\beta_S(s) - \alpha^v(s))\right]^+$



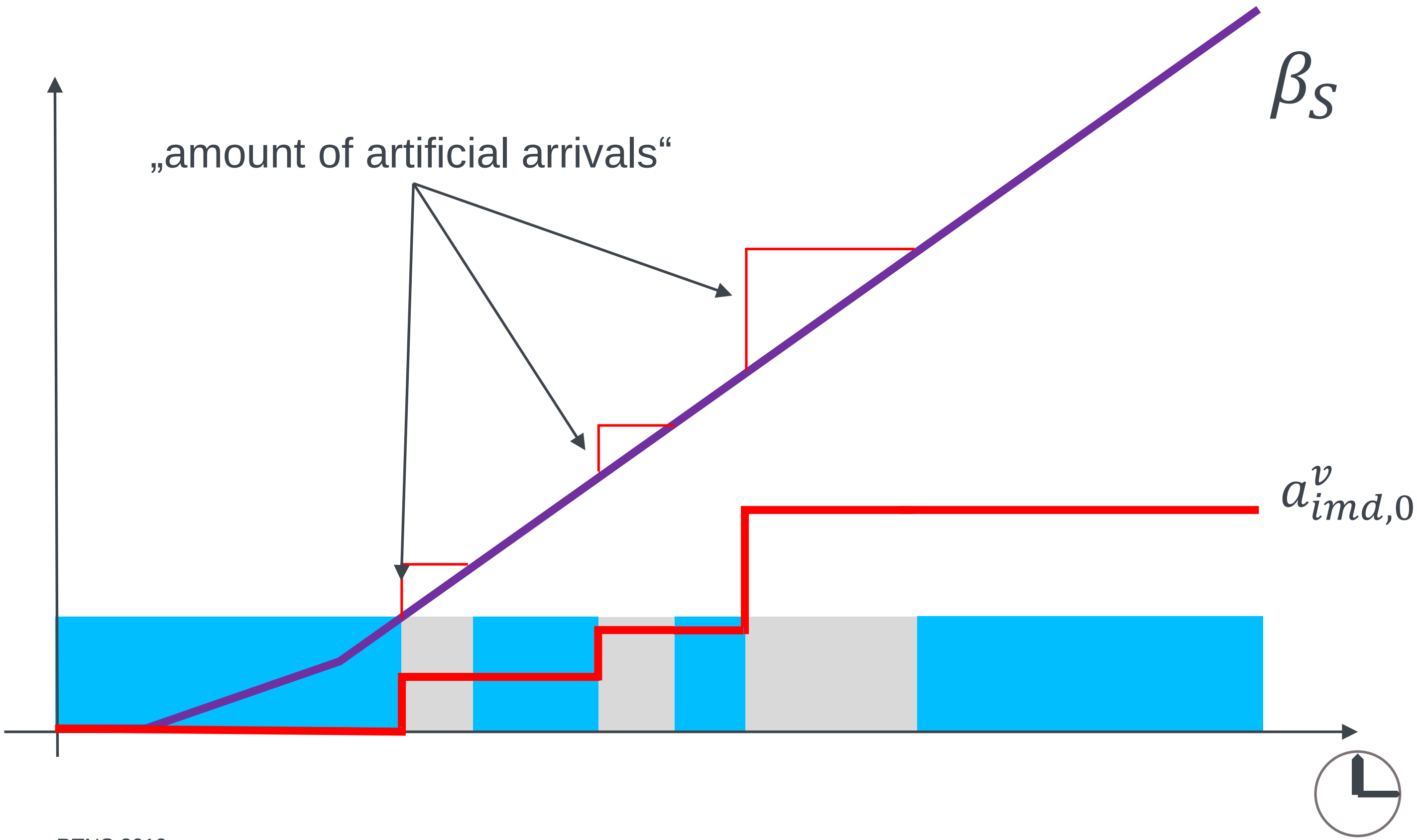
Time-invariant service curve

Leftover service curve: subtract virtual arrivals: $\beta(t) = \left[\sup_{s \leq t}(\beta_S(s) - \alpha^v(s))\right]^+$



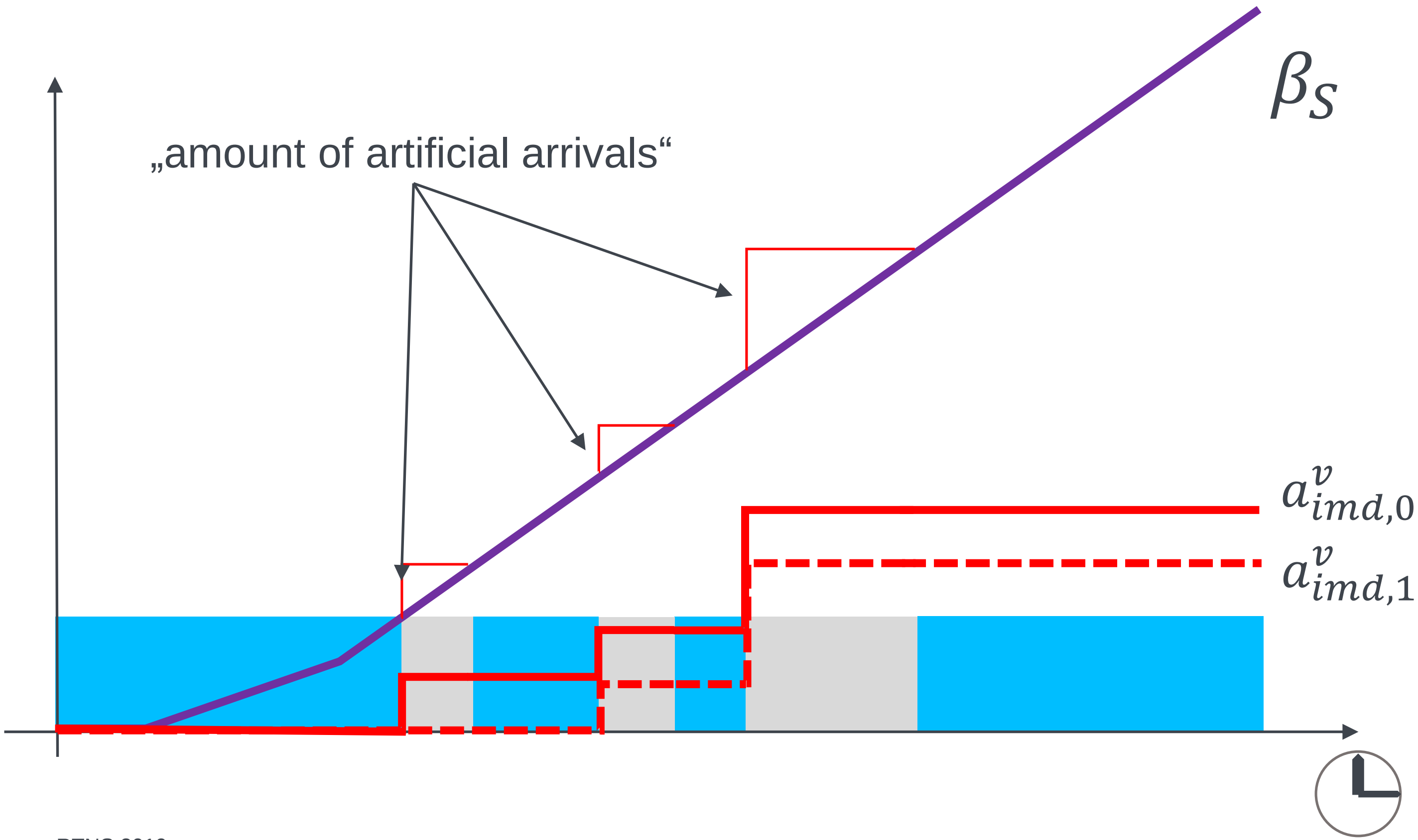
Time-invariant service curve

Leftover service curve: subtract virtual arrivals: $\beta(t) = \left[\sup_{s \le t}(\beta_S(s) - \alpha^v(s))\right]^+$



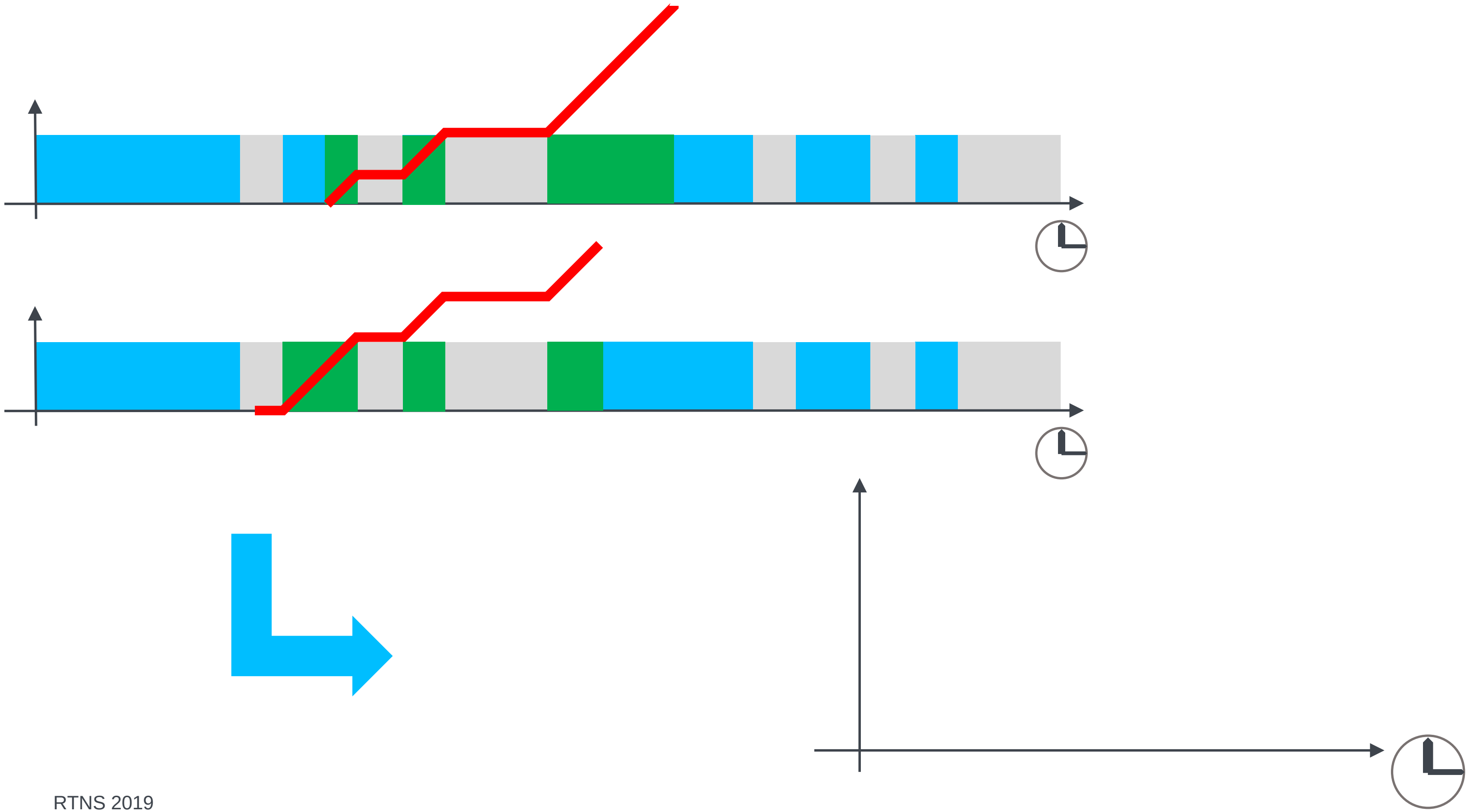
Time-invariant service curve

Leftover service curve: subtract virtual arrivals: $\beta(t) = \left[\sup_{s \le t}(\beta_S(s) - \alpha^v(s))\right]^+$



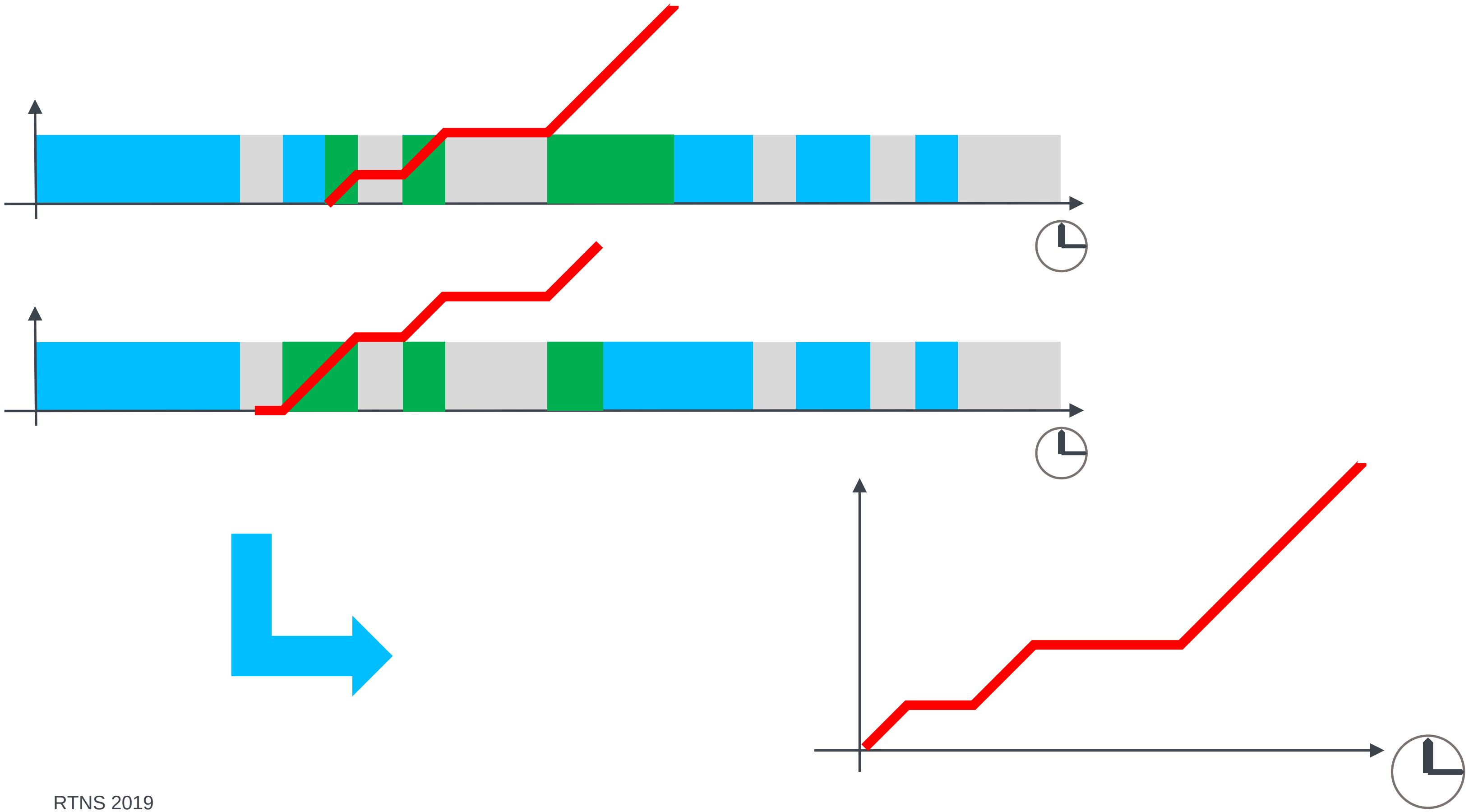
Time-invariant service curve

Direct service curve: find the worst case: $\beta(t) = \inf_{0<s}(\beta(s, s+t))$



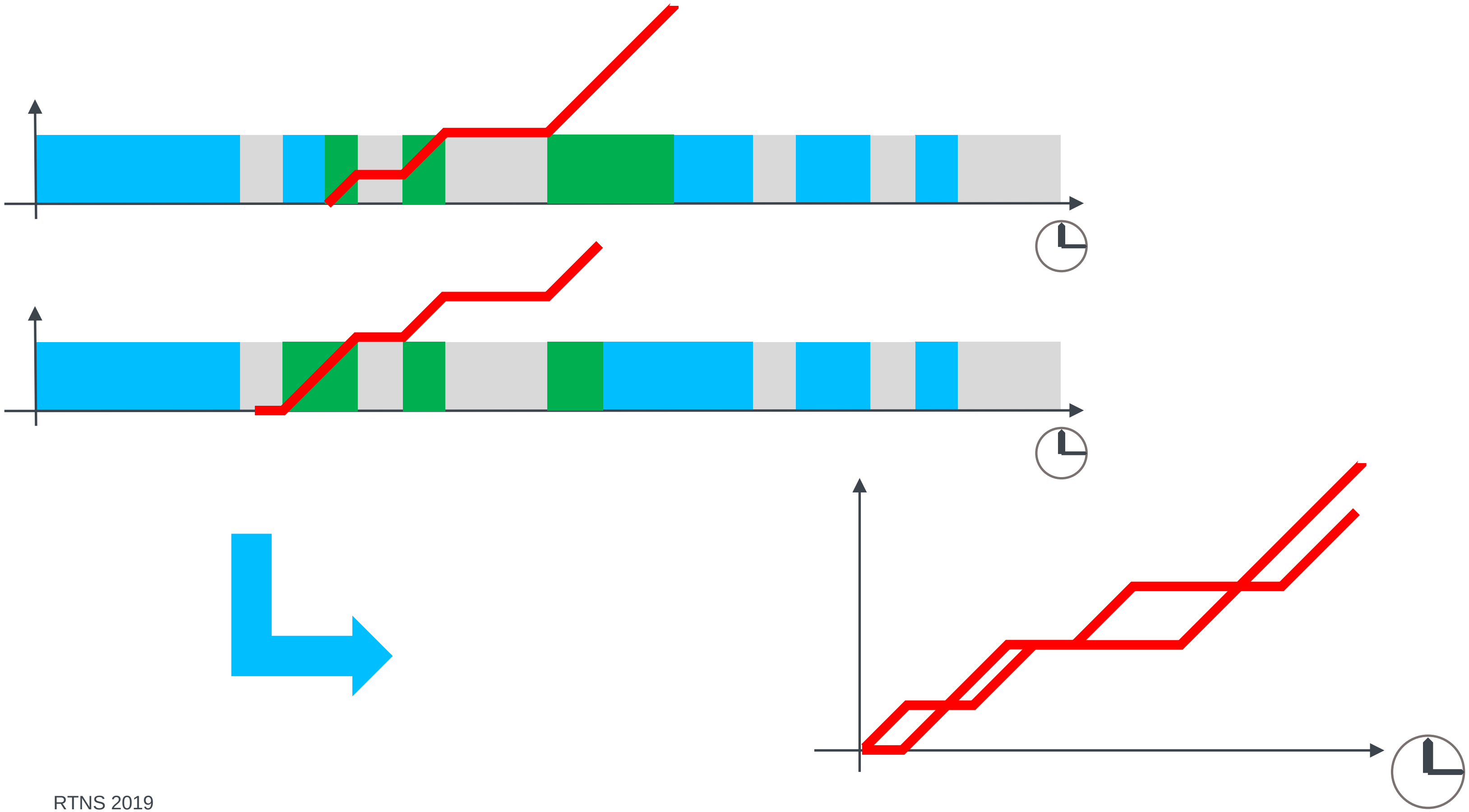
Time-invariant service curve

Direct service curve: find the worst case: $\beta(t) = \inf_{0<s}(\beta(s, s+t))$



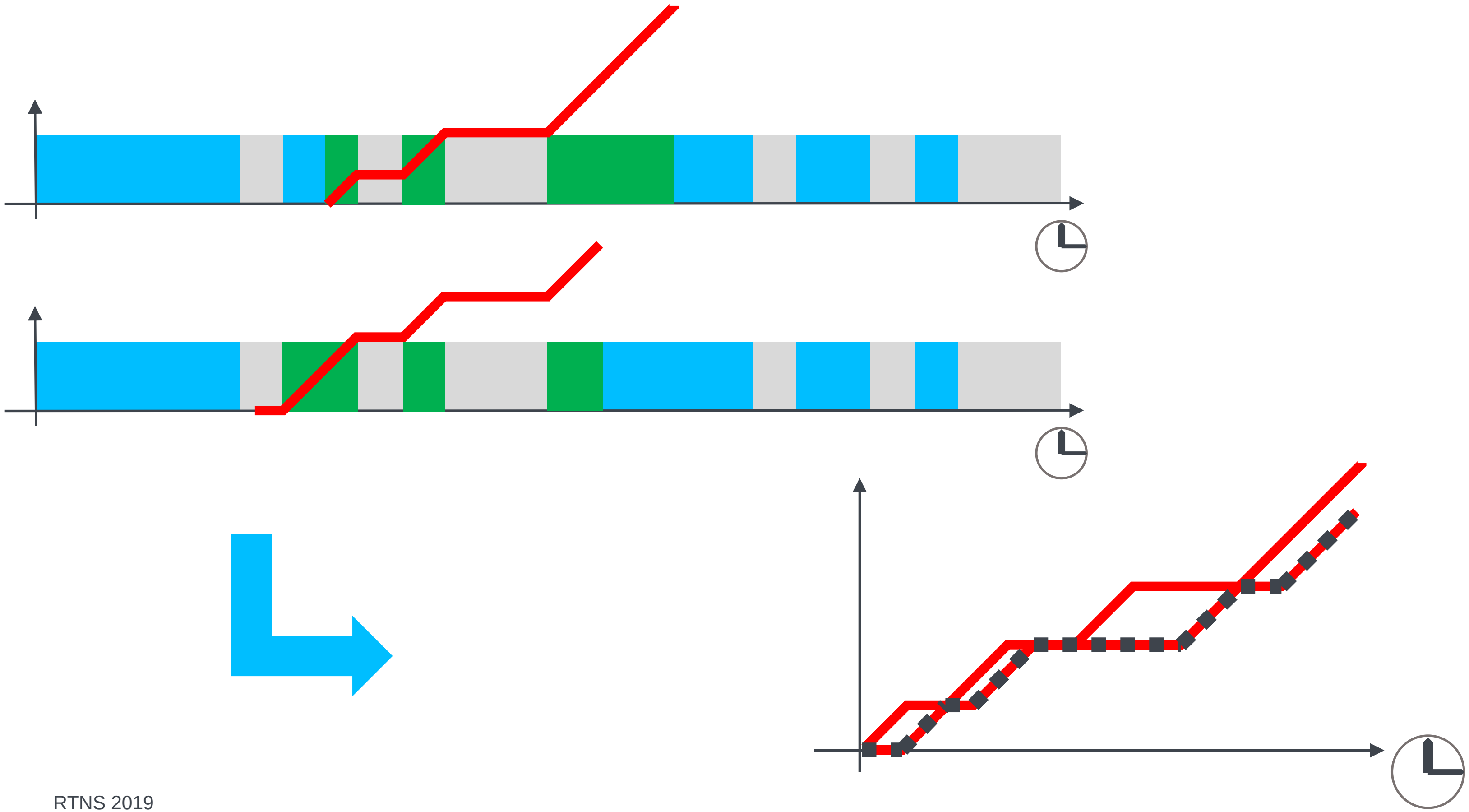
Time-invariant service curve

Direct service curve: find the worst case: $\beta(t) = \inf_{0<s}(\beta(s, s+t))$



Time-invariant service curve

Direct service curve: find the worst case: $\beta(t) = \inf_{0<s}(\beta(s, s+t))$



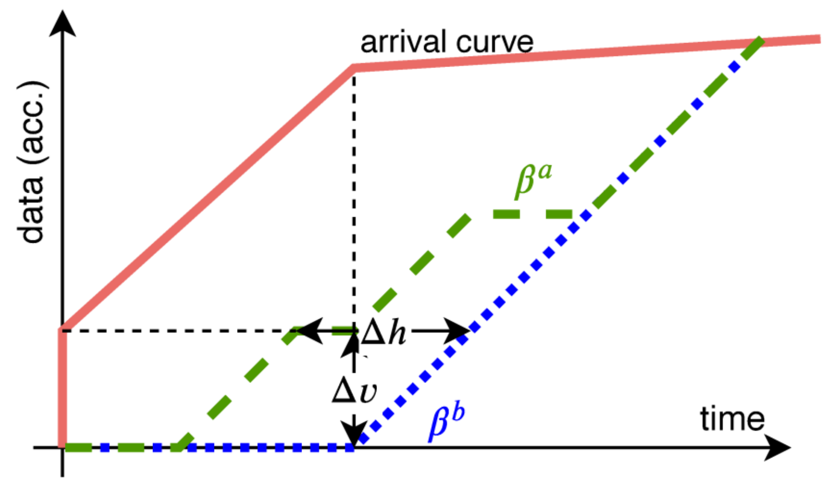
Evaluation

How much pessimism?

- Randomly generated schedules

- Numerical evaluation of “worst-case” overestimation

- Arrival curve independent metric
  - $\Delta v = \text{backlog}(\beta^a, \beta^b)$
  - $\Delta h = \text{virtual delay}(\beta^a, \beta^b)$



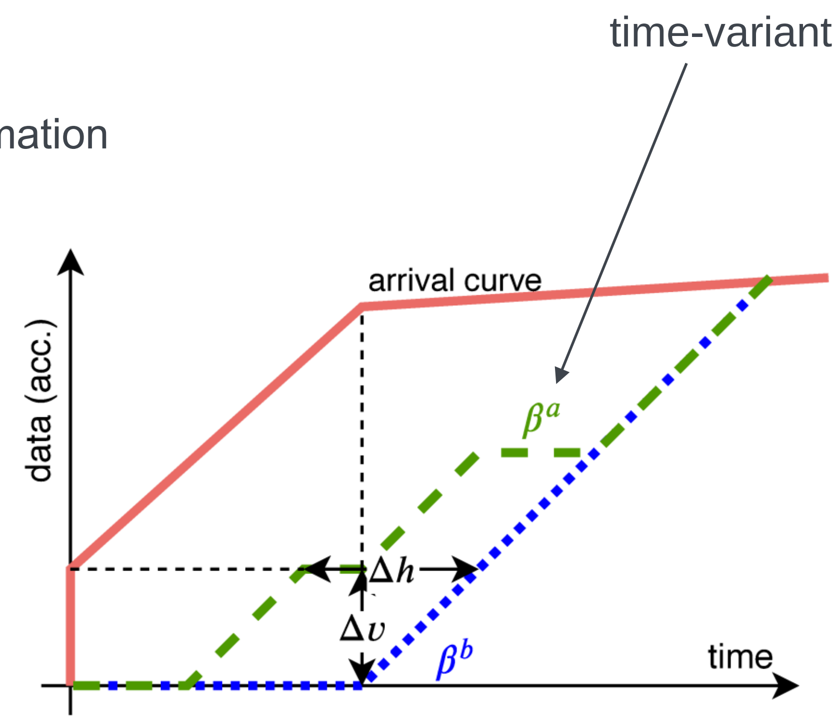
Evaluation

How much pessimism?

- Randomly generated schedules

- Numerical evaluation of “worst-case” overestimation

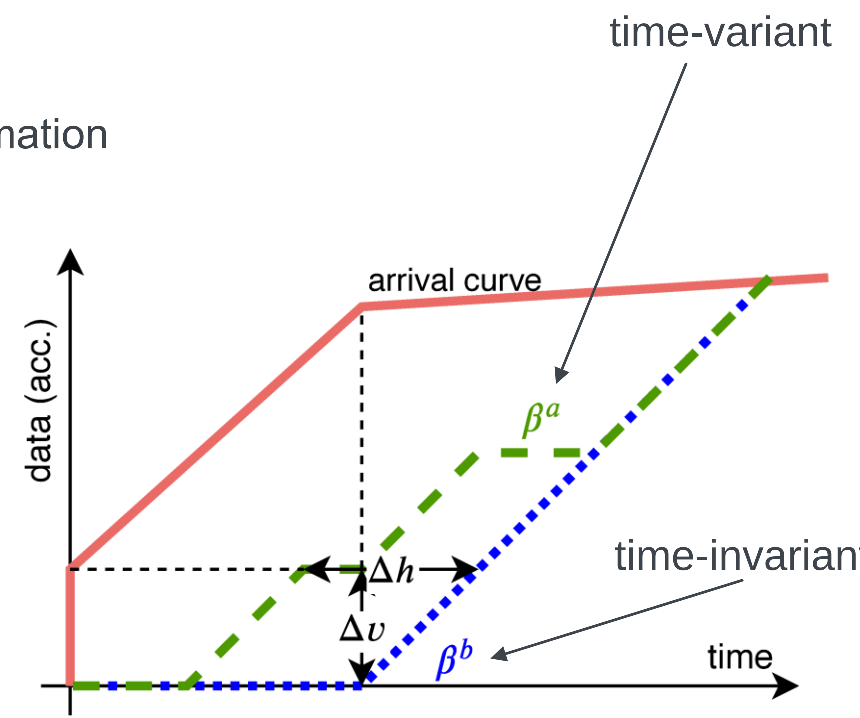
- Arrival curve independent metric
  - $\Delta v = \text{backlog}(\beta^a, \beta^b)$
  - $\Delta h = \text{virtual delay}(\beta^a, \beta^b)$



Evaluation

How much pessimism?

- Randomly generated schedules
- Numerical evaluation of “worst-case” overestimation
- Arrival curve independent metric
  - $\Delta v = \text{backlog}(\beta^a, \beta^b)$
  - $\Delta h = \text{virtual delay}(\beta^a, \beta^b)$



Evaluation

- Proof-of-concept numeric Python implementation

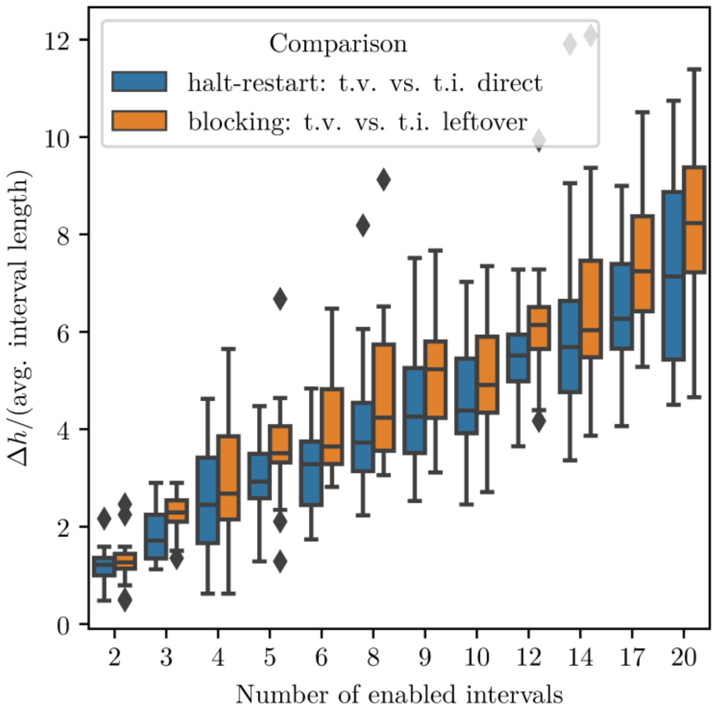
- $\beta_S = t$

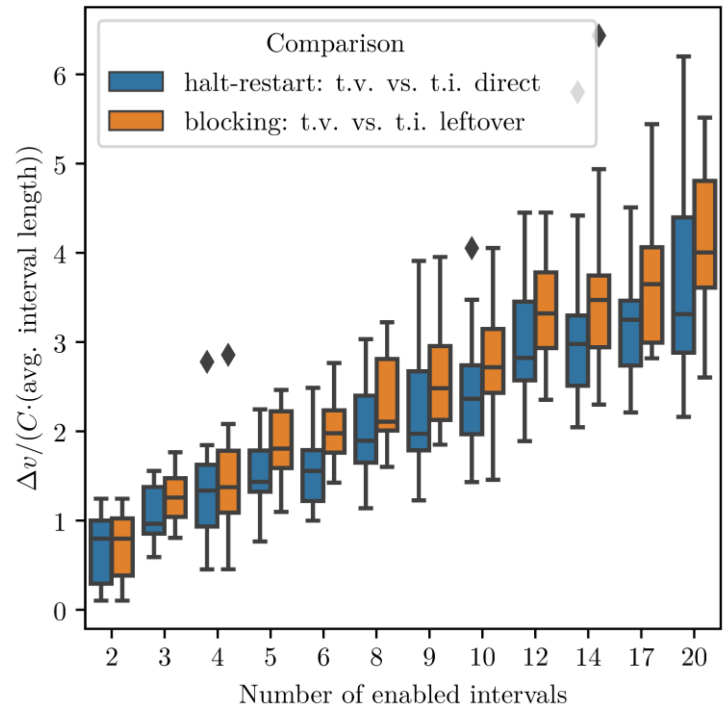
- Comparison of time-variant vs. time-invariant for:
  - blocking: t.v. vs. t.i. leftover*
  - halt-restart: t.v. vs. t.i. direct

- Extend t.i. curves with $\beta(s, t) = \beta_{t.i.}(t - s)$

Evaluation results

Random interval length [1,100] time units; 20 schedules per number of enabled intervals



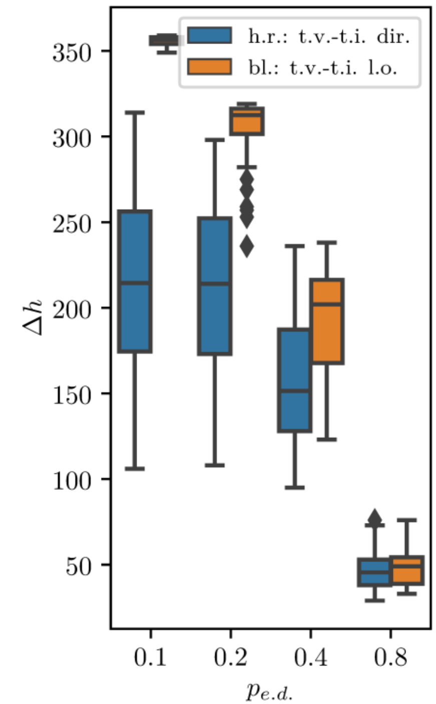


Evaluation results

Varying ratio of enabled/disabled intervals per cycle:

- 10 enabled intervals per cycle
- cycle length: 400 time units

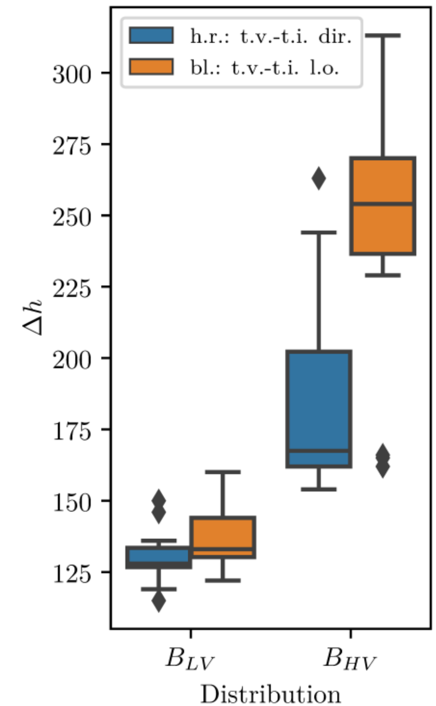
Time-invariant service curve is potentially more pessimistic for long disabled intervals with short interspersed enabled intervals.



Evaluation results

Equal mean interval length, different interval length variance:

- 20 enabled intervals per cycle
- Binomial distribution B_{HV}:
  - Mean interval length: 100
  - Variance: 90
- Binomial distribution B_{LV}:
  - Mean interval length: 100
  - Variance: 20



Discussion

Practical systems have some favorable properties.

- Usually β_S is quite constant
  - Time-invariant service curves can be evaluated more easily

- In converged networks, it is not unlikely to have long enabled intervals with short disabled intervals (from the perspective of the traffic to be analyzed)

Concluding remarks

- Closer look at the fundamental properties of systems with intermittent service
- Similar but Distinct: Blocking vs. Halt-Restart
- Open problems
  - computational support for NC with complicated functions
    - Schedule for service intermittence results in complicated service curves
    - Computational algorithms for time-variant network calculus
  - Multiplexing (i.e., multiple streams sharing one queue)

La fin.