ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Computer Engineering and
Networks Laboratory

Analytic Real-Time Analysis and Timed Automata: A Hybrid Method for Analyzing Embedded Real-Time Systems

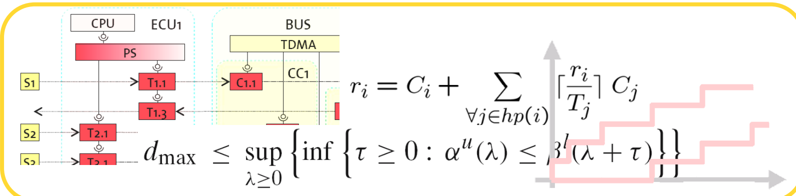
Kai Lampka, Simon Perathoner, Lothar Thiele

Artist-Design Cluster Meeting (Hardware Platforms)

Haus der Wissenschaft, Braunschweig, Germany, 26 June 2009

Performance Analysis of Embedded Real-Time Systems

Analytic Real-Time Analysis

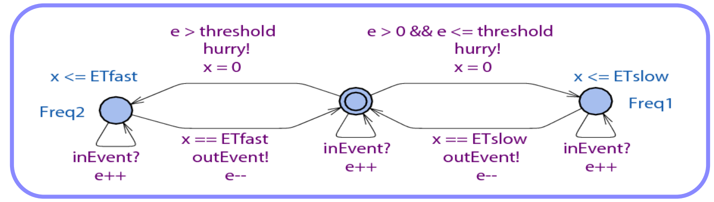


Solution of closed form expressions

Examples: RTC, SymTA/S, MAST, ...

- \+ Good scalability
- \+ Fast analysis
- – Limited to few specific measures (e.g. delays, buffer sizes)
- – Systems restricted to specific models
- – Overly conservative results

State-based Real-Time Analysis



Model checking of properties

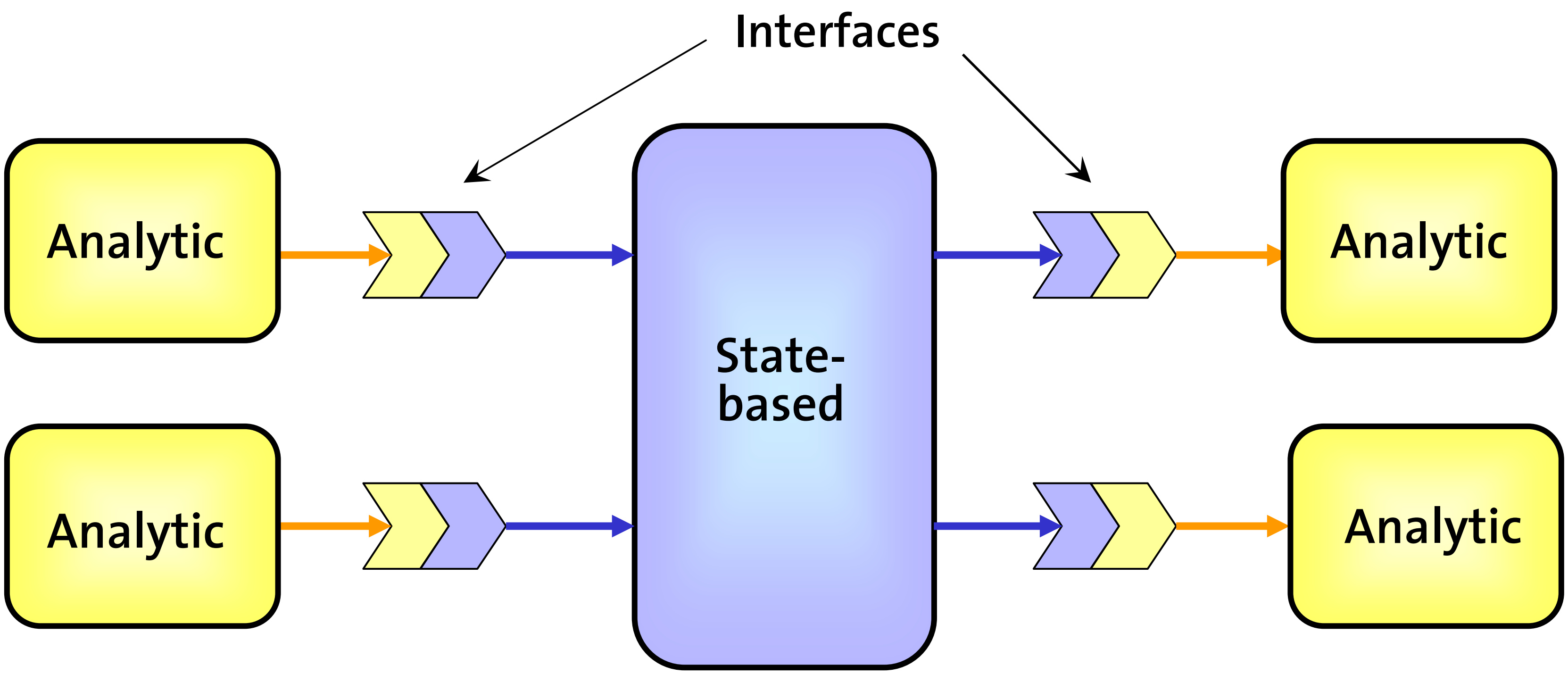
Examples: Timed Automata (TA), FSM, ...

- – Poor scalability
- – Slow verification

} State space explosion

- \+ Verification of functional and non-functional properties
- \+ Modeling power
- \+ Exact results

New Compositional Framework for Hybrid Analysis



Motivation for Hybrid Approach

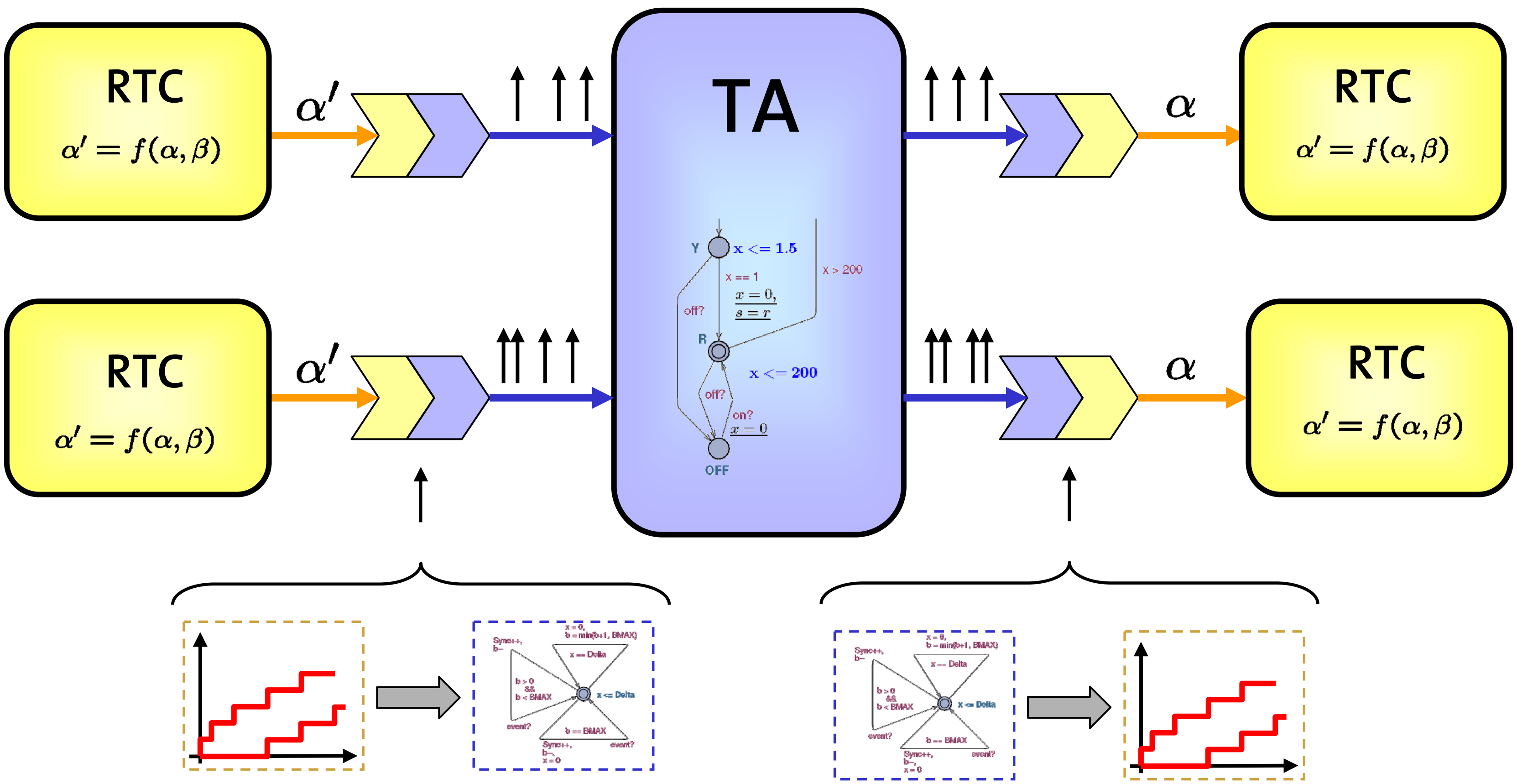
1. The obtained performance metrics are not destructively over-approximated

   (Tighter analysis results compared to purely analytical abstraction)

2. The problem of state space explosion is limited to the level of isolated components

   (Faster verification compared to purely state-based models)

Interfacing Real-Time Calculus and Timed Automata



Contributions

- Pattern for conversion of abstract event stream models (such as PJD or arrival curves) to a network of cooperating TA
- Proof of correctness and completeness
- Pattern for derivation of abstract event stream models from a TA-based system model
- Implementation and Case Study

Related work

- **Event Count Automata**

  L. T. X. Phan, S. Chakraborty, P. S. Thiagarajan, and L. Thiele. ***Composing functional and state-based performance models for analyzing heterogeneous real-time systems***. In Proc. of the 28th IEEE Real-Time Systems Symposium (RTSS 2007), pages 343–352. IEEE Computer Society, 2007.

- **CATS Tool**

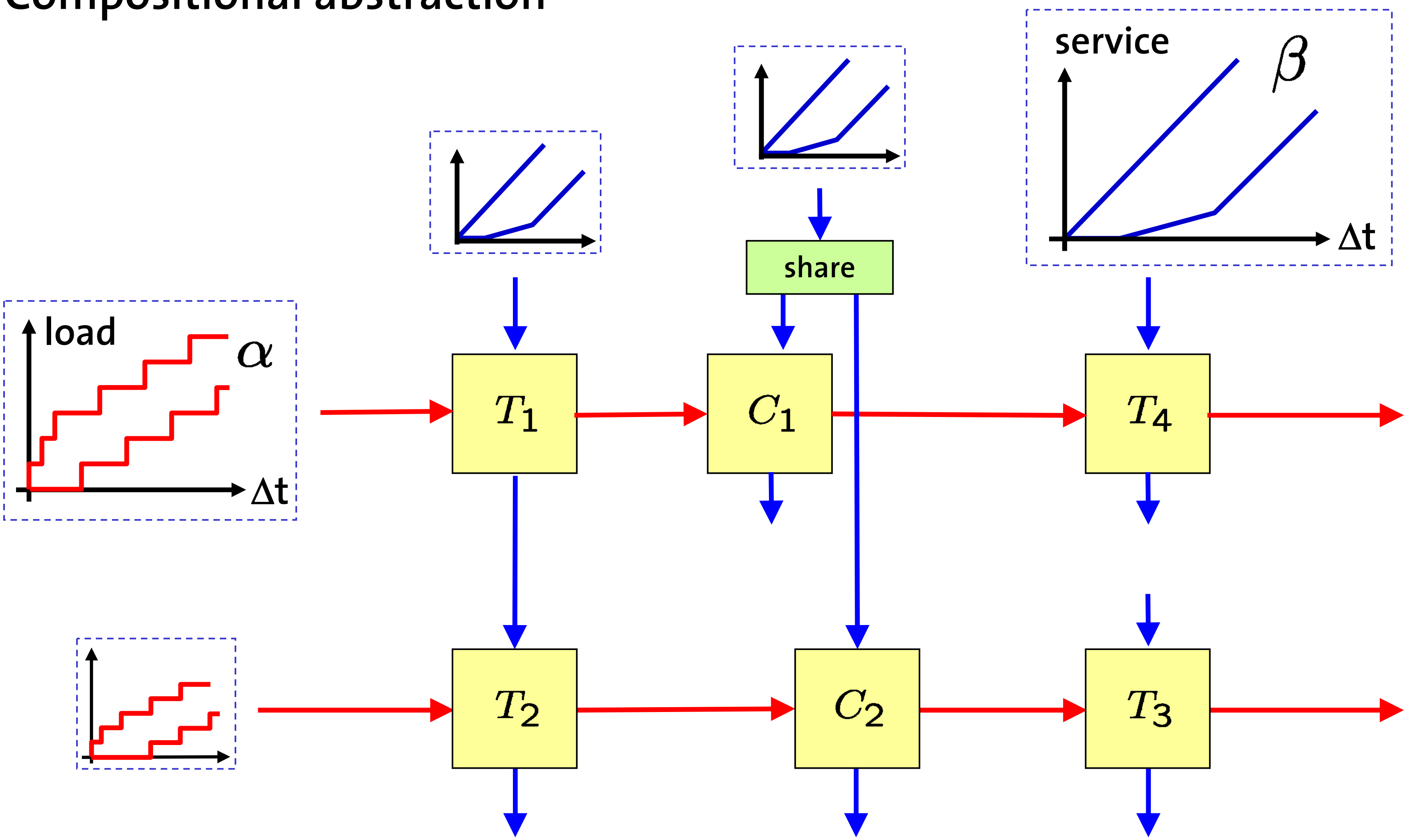
  P. Krcal, L. Mokrushin, and W. Yi. ***A tool for compositional analysis of timed systems by abstraction*** (extended abstract). In Proc. of NWPT07, the 19th Nordic Workshop on Programming Theory, October 2007.

- **Efficient Model-Checking for Real-Time Task Networks**

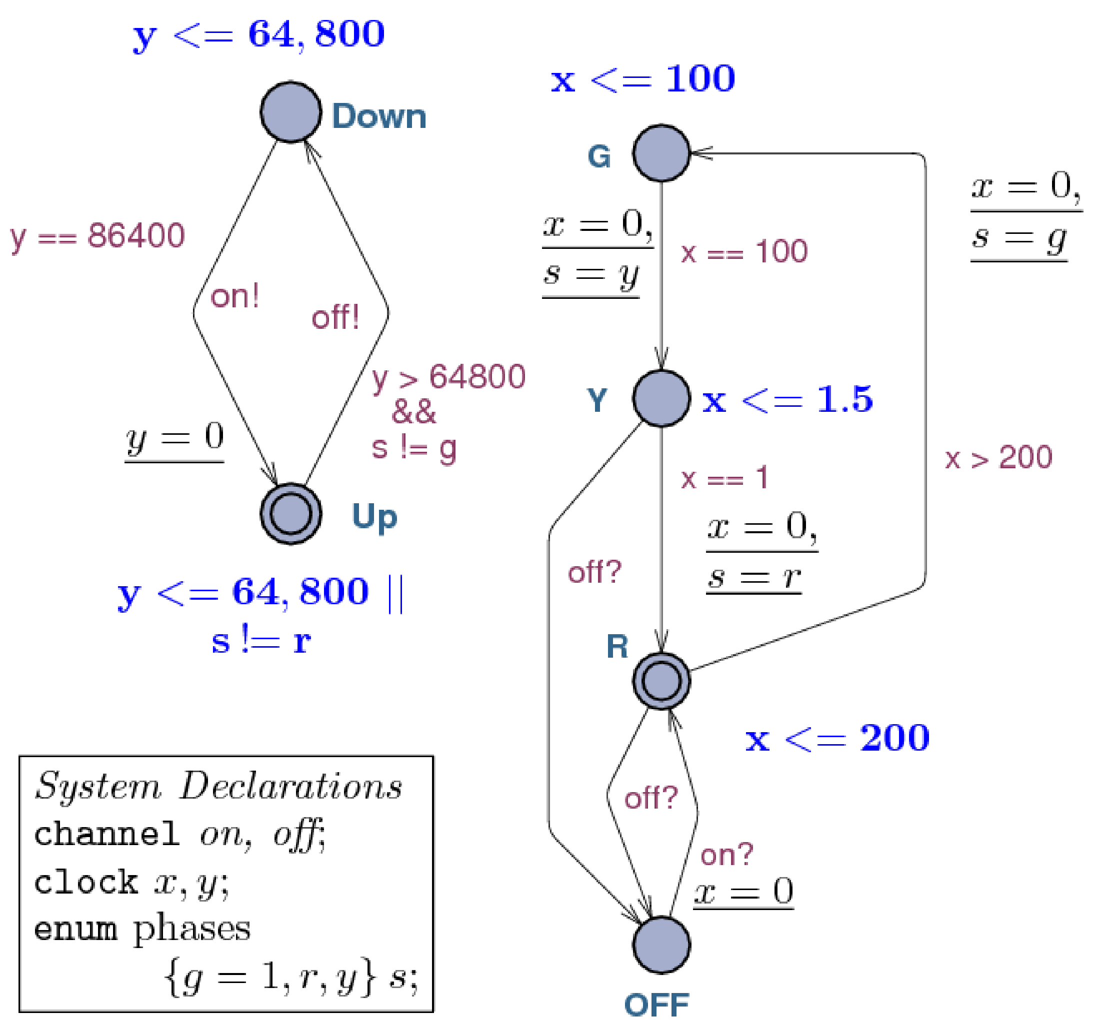
  H. Dierks, A. Metzner, and I. Stierand. ***Efficient Model-Checking for Real-Time Task Networks***. In Int. Conf. on Embedded Software and Systems 2009. Accepted for publication.

Real-Time Calculus (RTC)

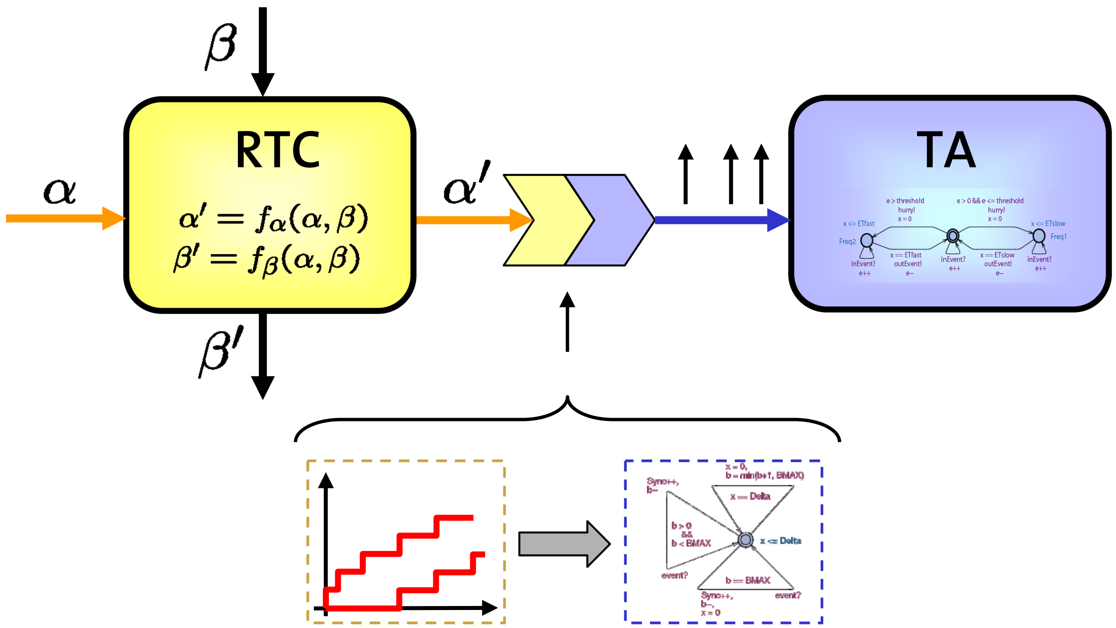
Compositional abstraction



Timed Automata (TA)



Interface RTC → TA



How to represent arrival curves as TA?

Principle

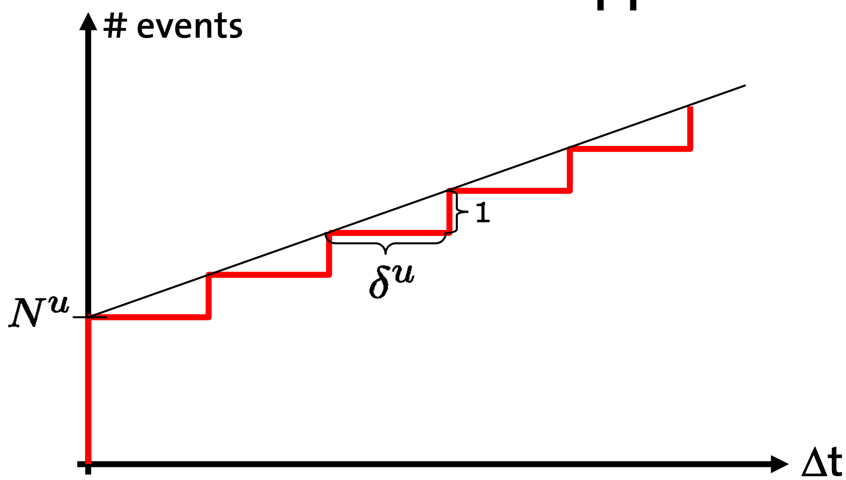
1. Decompose arrival curves to set of simpler curve components
   → Set of linear staircase functions

2. Represent each curve component as TA (Leaky Bucket pattern)
   → Set of simple TA

3. Synchronize all TA such to obtain same event stream model as described by arrival curve
   → Network of synchronized TA

Linear arrival curves

Upper arrival curve

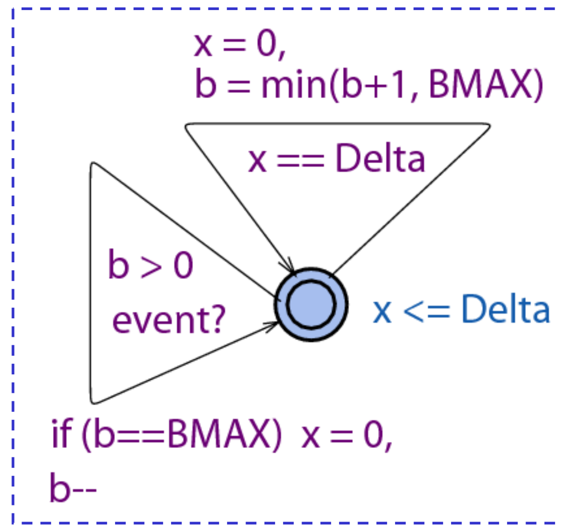


$$\alpha^u(\Delta) = N^u + \left\lfloor \frac{\Delta}{\delta^u} \right\rfloor$$

Max fill level: N^u

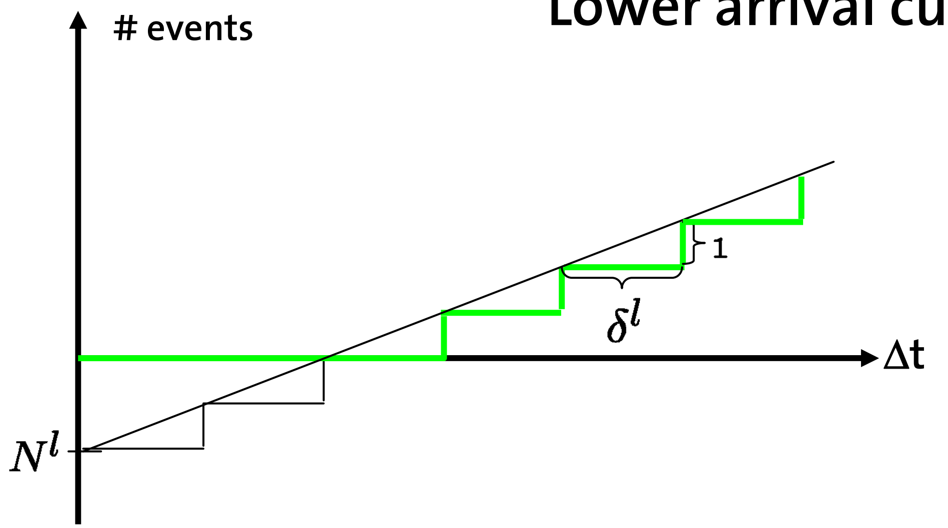
Fill rate: $1/\delta^u$

Event emission allowed if fill level > 0



Automaton for linear upper arrival curve (UTA)

Linear arrival curves



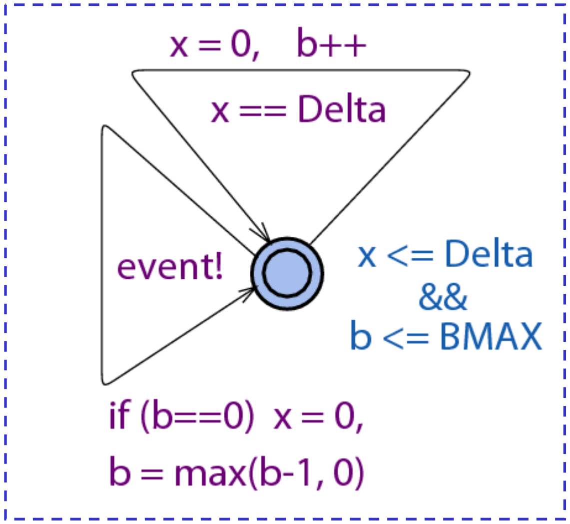
Lower arrival curve

$$\alpha^l(\Delta) = \max\left\{0, N^l + \left\lfloor \frac{\Delta}{\delta^l} \right\rfloor\right\}$$

Max fill level: $|N^l|$

Fill rate: $1/\delta^l$

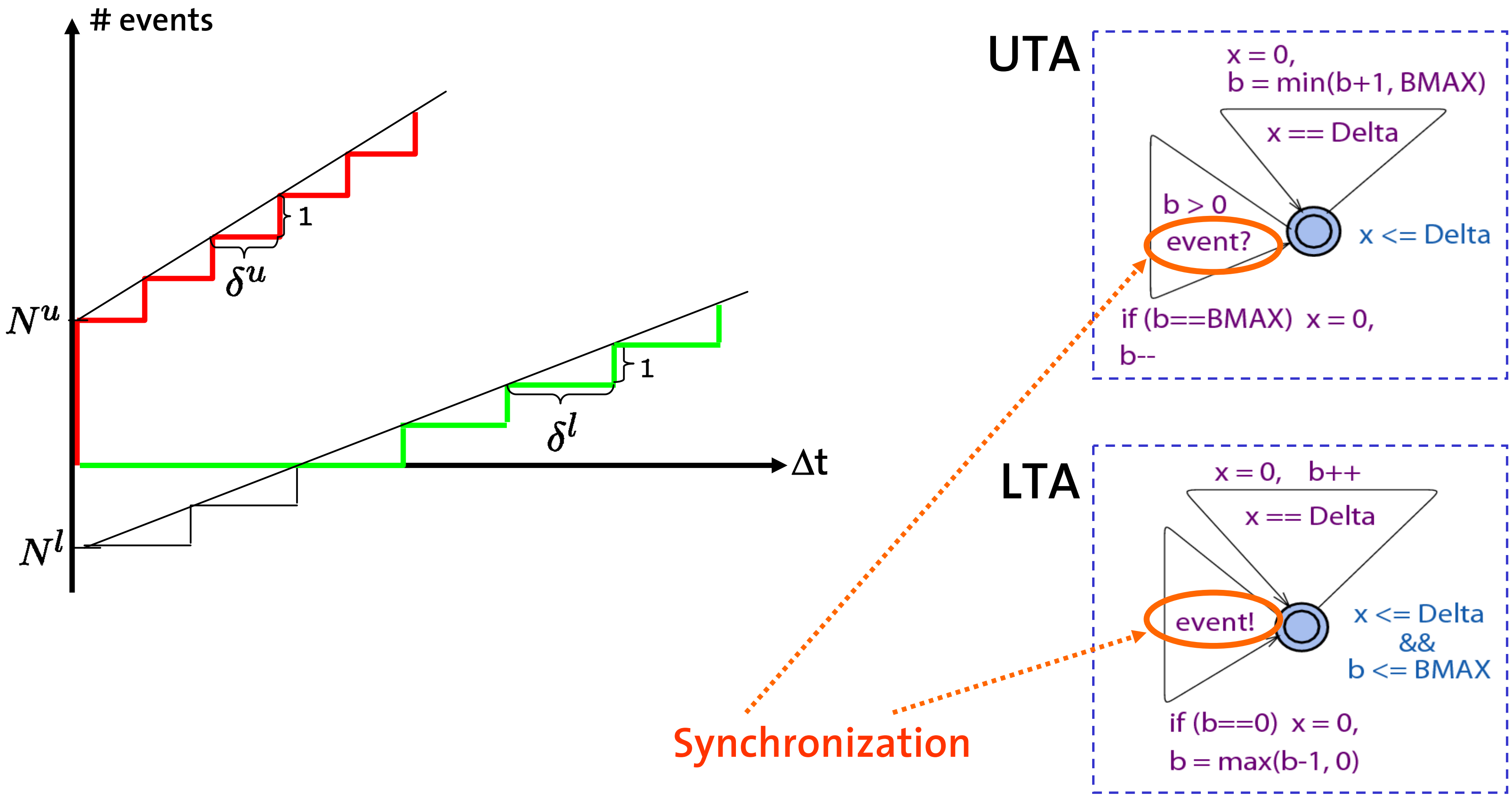
Event emission enforced if maximum fill level reached



Automaton for linear lower arrival curve (LTA)

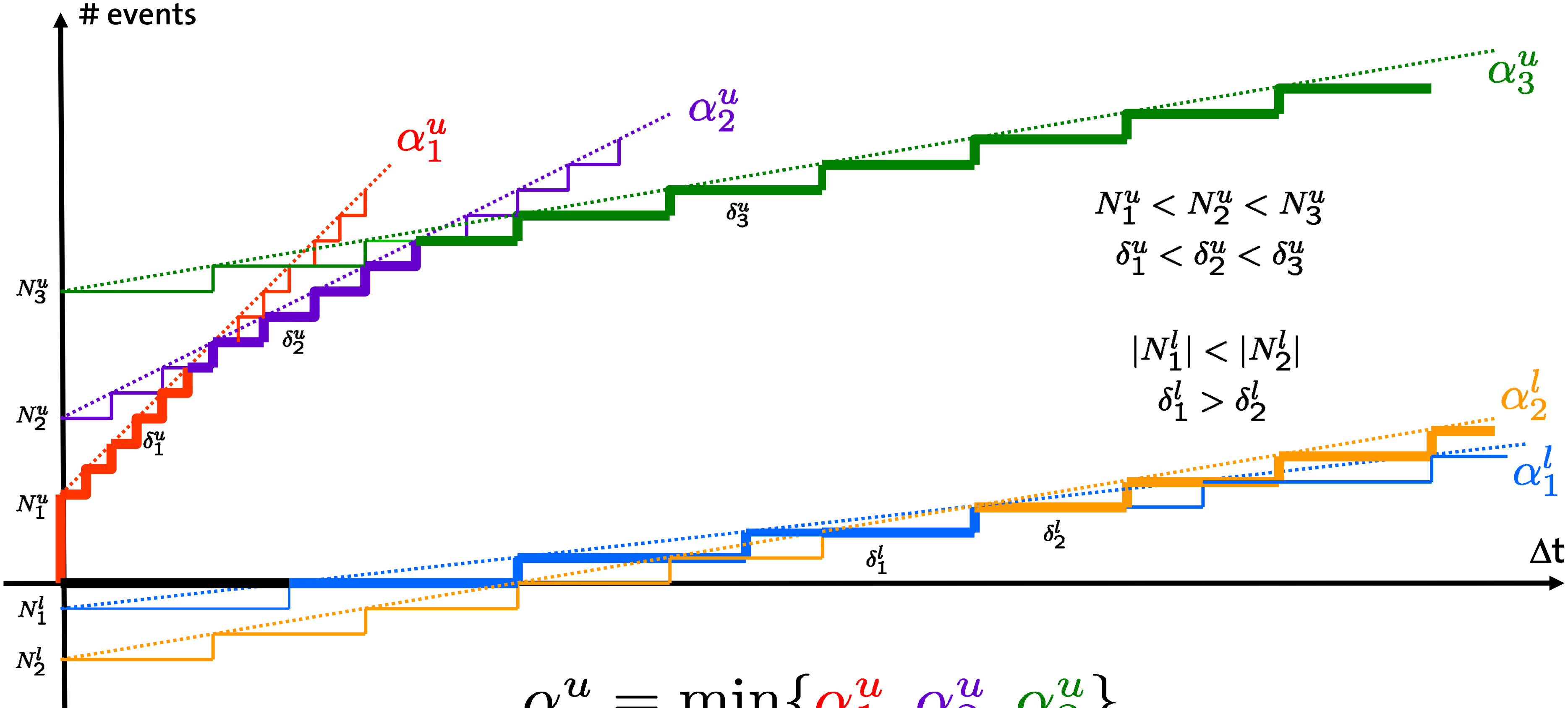
Linear arrival curves

Combination of lower and upper arrival curves



Convex and concave patterns

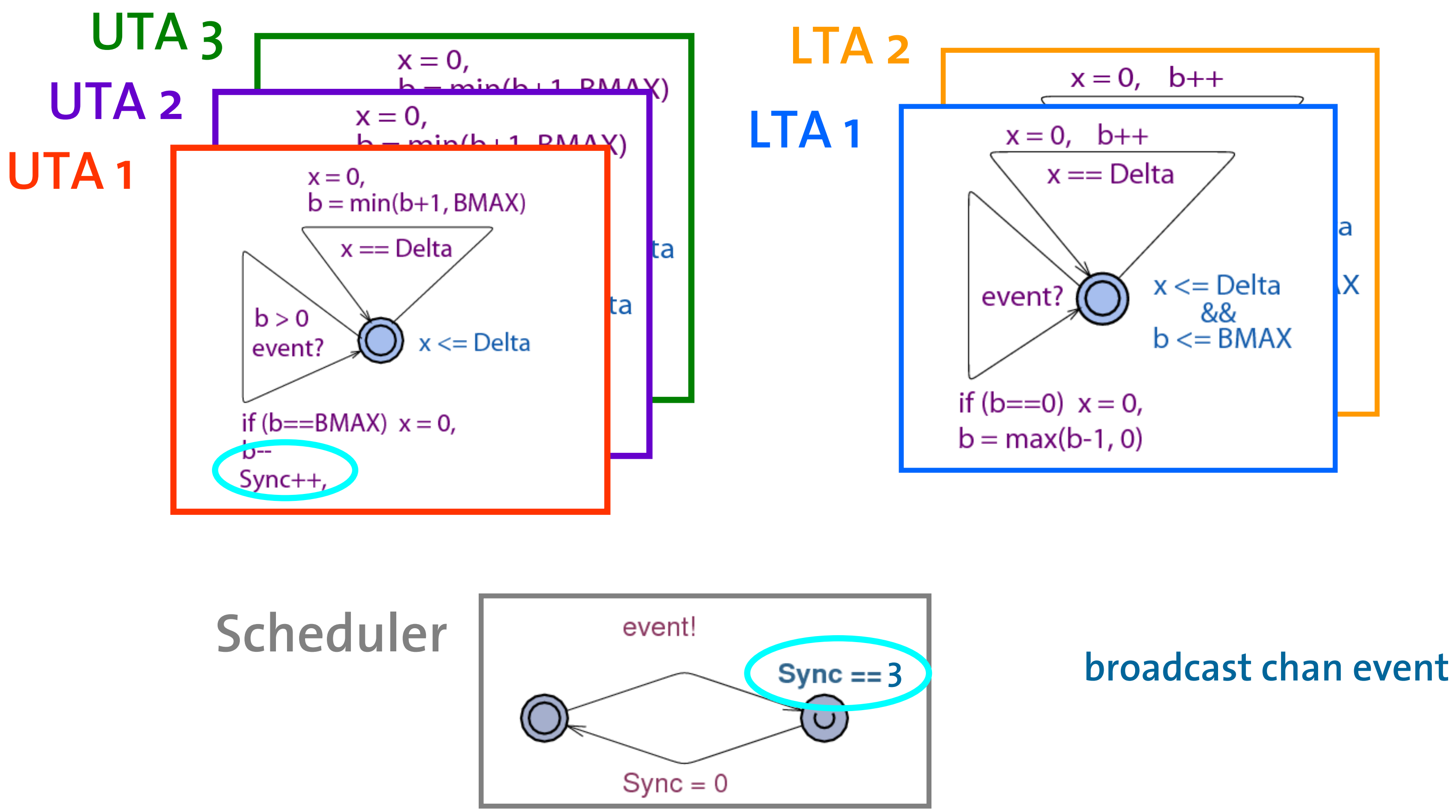
Composition of linear staircase functions



$$\alpha^u = \min\{\alpha_1^u, \alpha_2^u, \alpha_3^u\}$$

$$\alpha^l = \max\{0, \alpha_1^l, \alpha_2^l\}$$

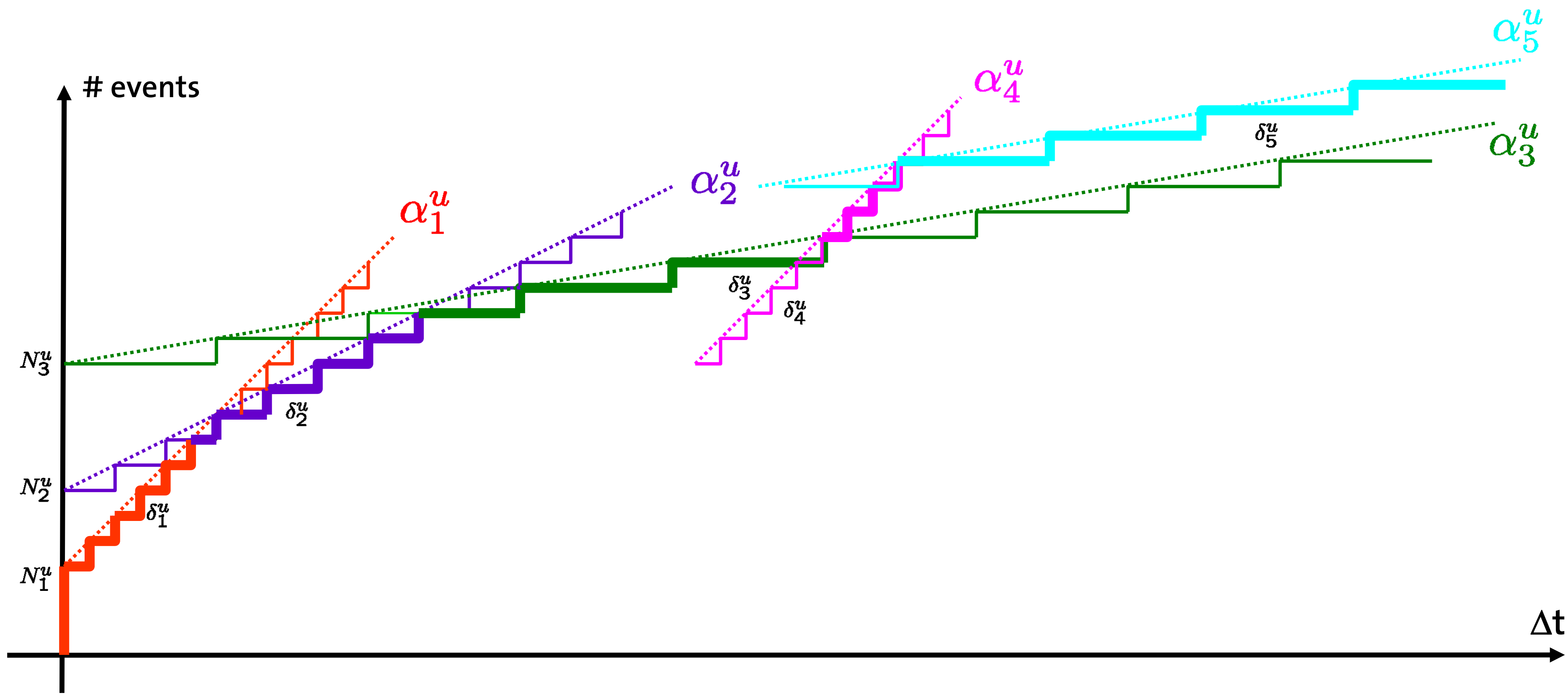
Convex and concave patterns



- Event generation only if <u>all</u> UTA permit it (AND composition)
- <u>Single</u> LTA can enforce event generation (OR composition)

General arrival curves

How to represent non-convex/concave patterns?



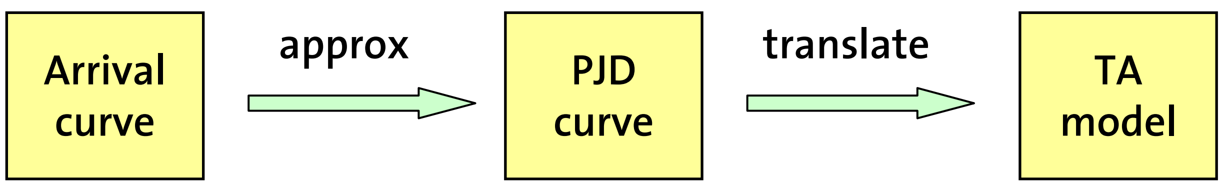
Use min/max operators locally on subsets of UTA/LTA

Complexity

Run-time of verification increases exponentially with number of clocks

→ Approximate arrival curves with few staircase functions

e.g.

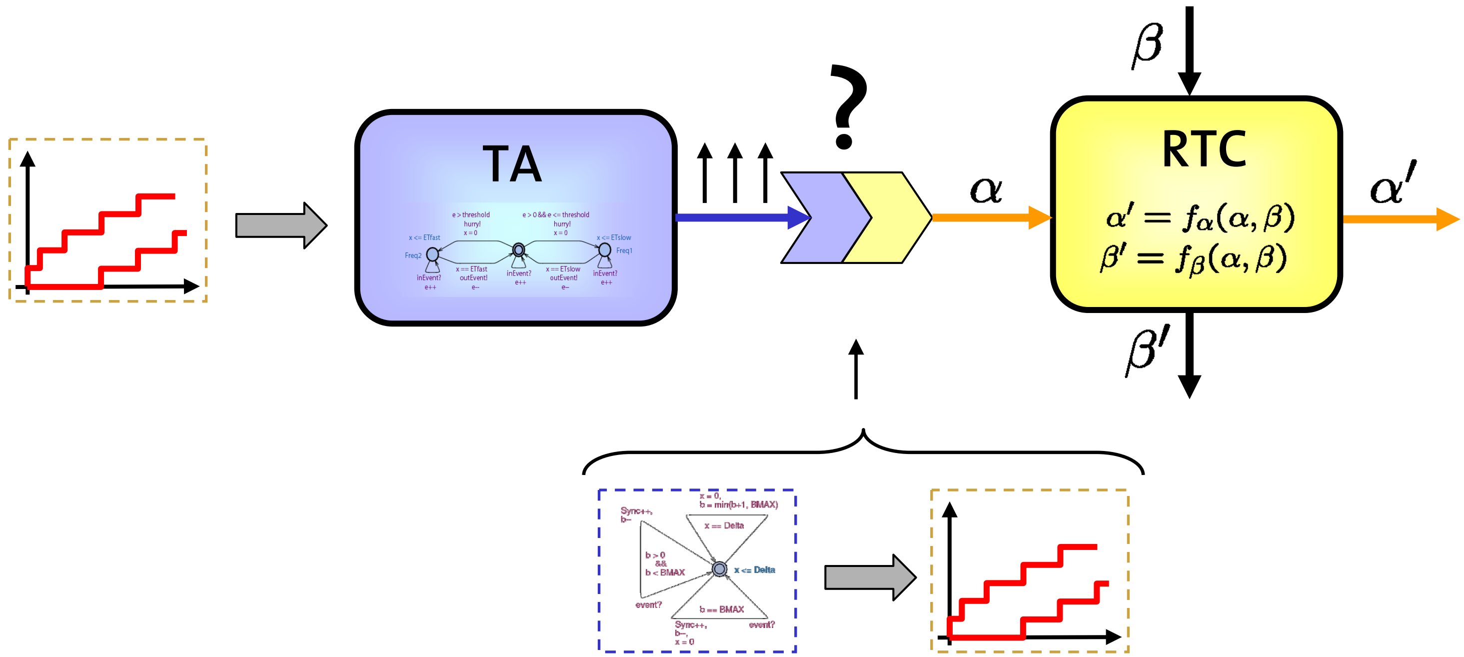


$$\#UTA \leq 2$$
$$\#LTA = 1$$

$$d = 0 \ \vee \ d \leq p - j : \quad N^u = \left\lceil \frac{j}{p} \right\rceil + 1; \ \ N^l = \left\lceil \frac{j}{p} \right\rceil; \ \ \delta^u = \delta^l = p$$

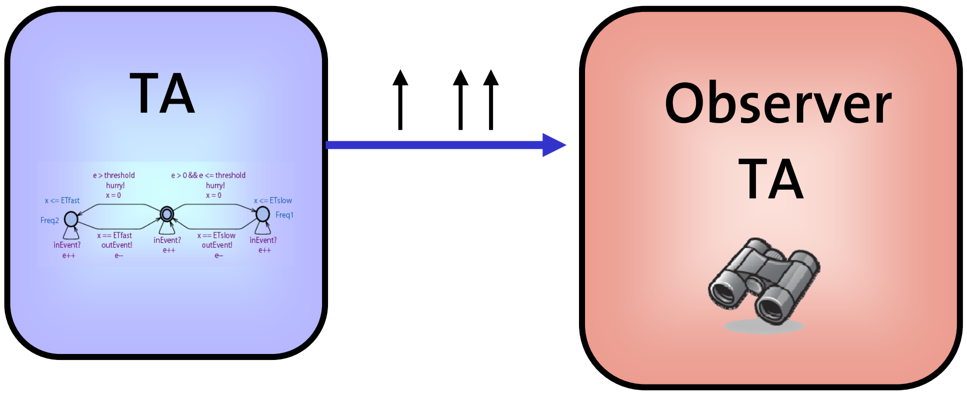
$$d > 0 \ \wedge \ d > p - j : \quad N_1^u = 1; \ \ \delta_1^u = d; \ \ N_2^u = \left\lceil \frac{j}{p} \right\rceil + 1$$
$$N^l = \left\lceil \frac{j}{p} \right\rceil; \ \ \delta_2^u = \delta^l = p$$

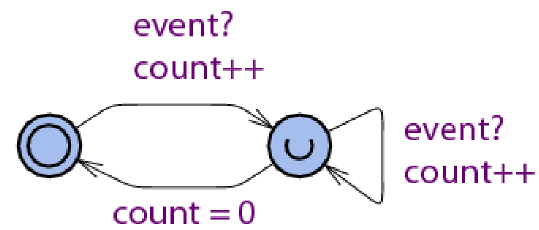
Interface TA → RTC



How to derive output arrival curves from a TA sub-system model?

Interface TA → RTC



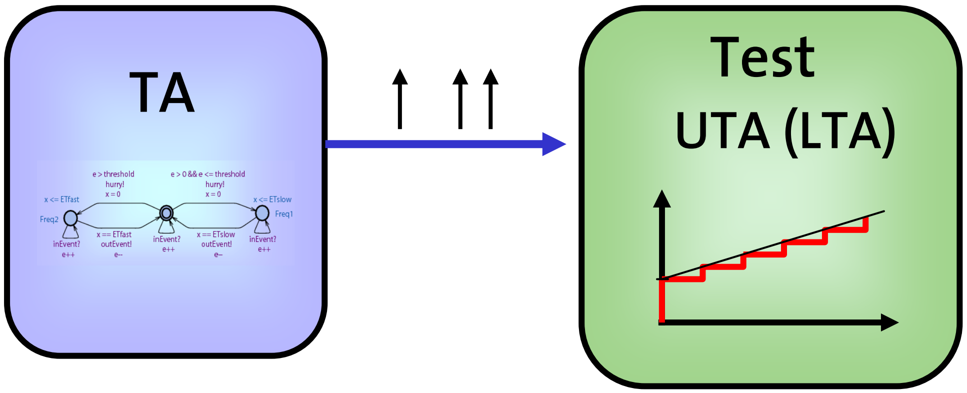


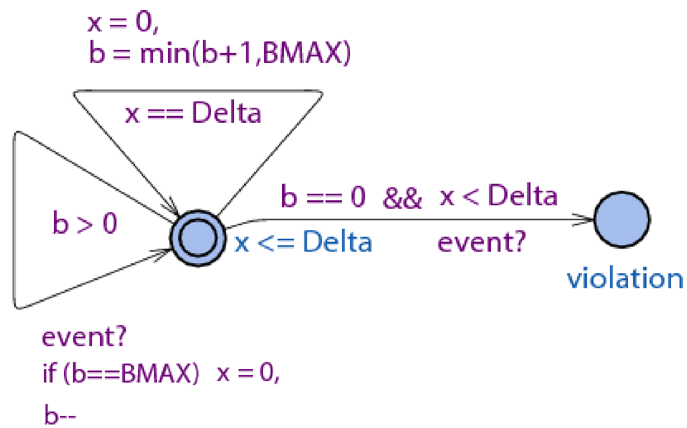
Verify

`A[] (count<=estimate)`

Key parameters of curve (e.g. max burst) are determined by appropriate observer TA and binary search

Interface TA → RTC





Verify

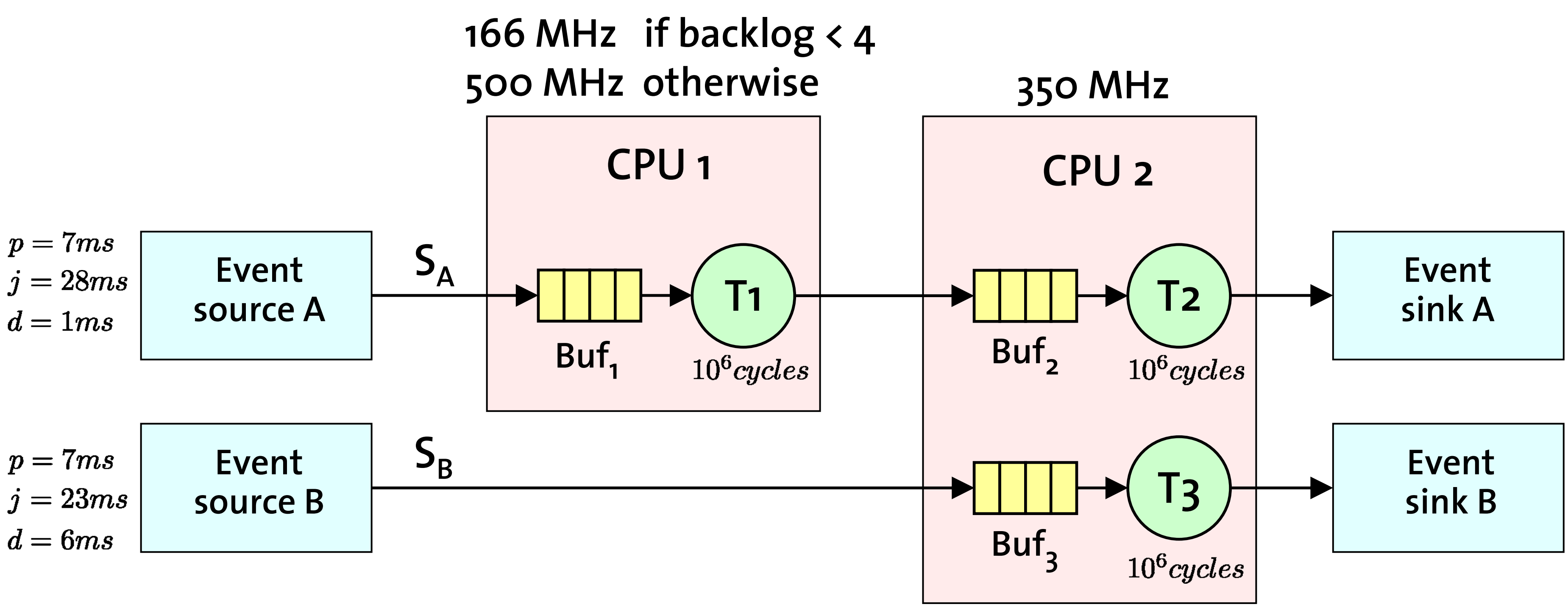
`(A[] (not violation))`

- Verify compliance of system output with a number of UTA (N_i,δ_i) and LTA (N_i,δ_i) (Search strategy: Fix one parameter and modify the other by binary search)
- Combine obtained linear staircase functions by min and max operators

→ Yields convex/concave approximation of system output

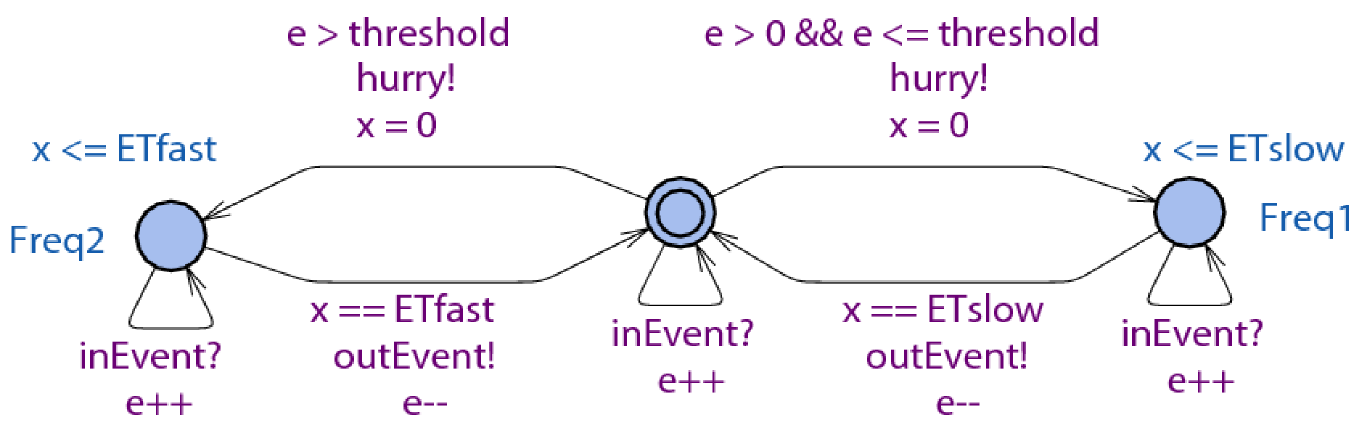
Case Study

CPU1: Load-dependent frequency adaptation



- Characterize output of T1
- Determine delays and required buffer sizes

Case Study

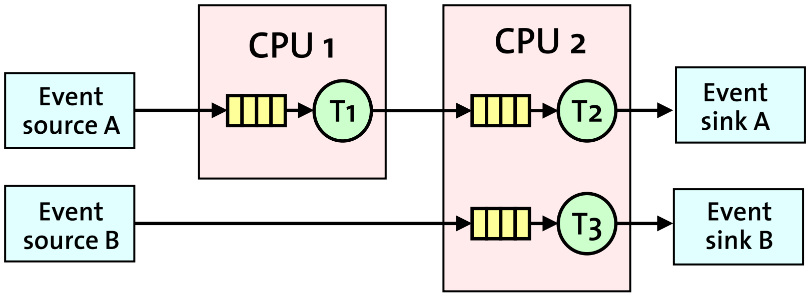


TA model for CPU1

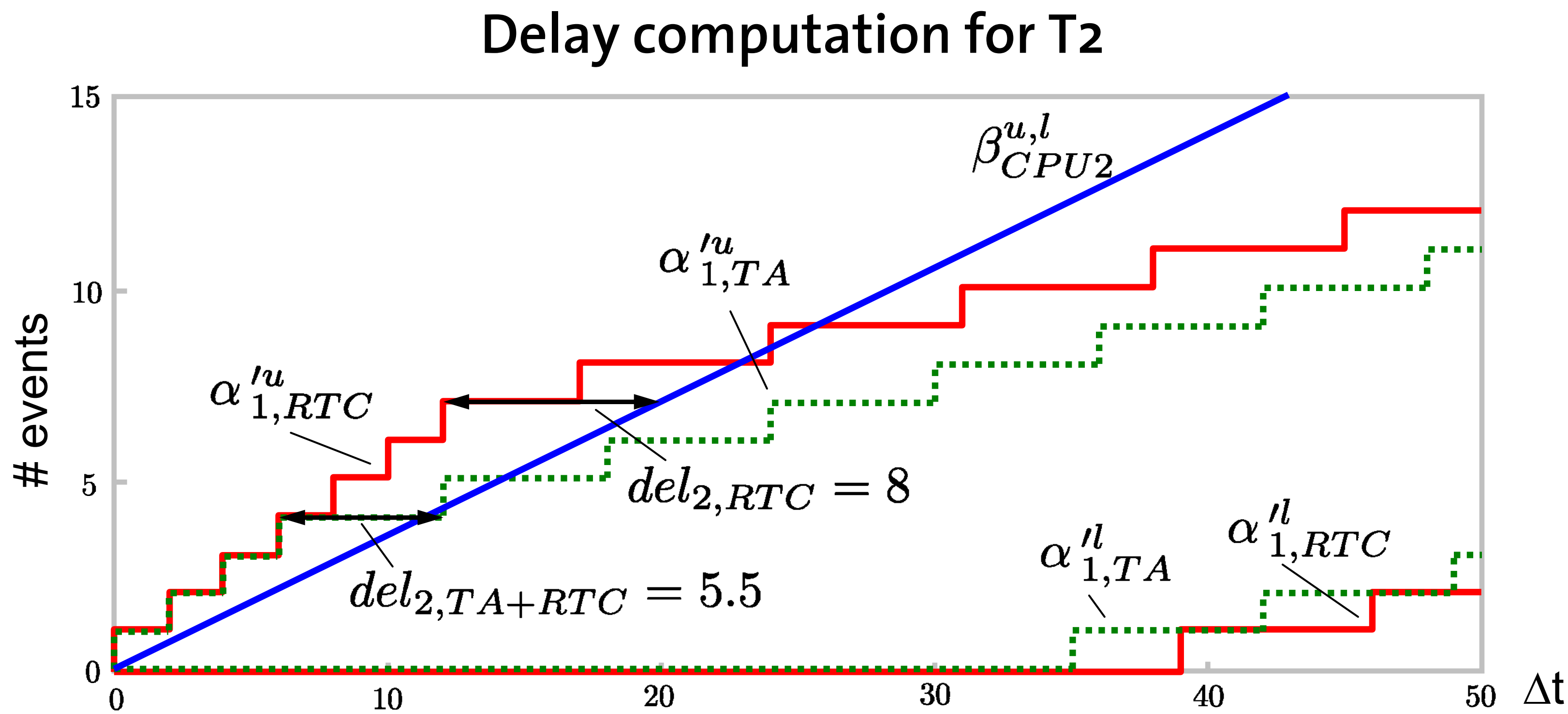
Case Study

Results of performance analysis

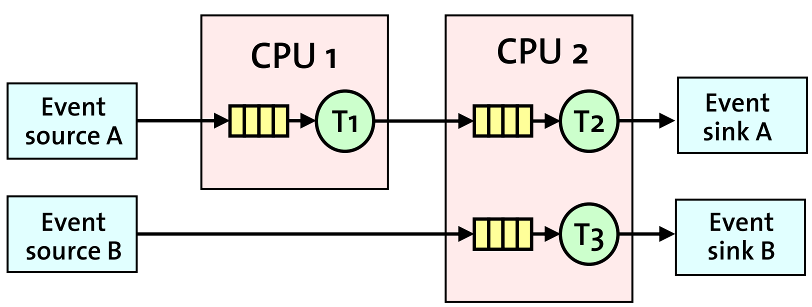
	Max delay [ms]				Max buffer [events]		
	T_1	T_2	T_3	EE_A	T_1	T_2	T_3
RTC	29	8	28.6	31.9	5	3	5
TA + RTC	25	5.5	17.2	30.5	5	2	3
TA	25	4.6	14.3	27.9	5	2	3



Case Study



Delay computation for T2
events
Δt
$\beta^{u,l}_{CPU2}$
$\alpha'^{u}_{1,TA}$
$\alpha'^{u}_{1,RTC}$
$del_{2,RTC} = 8$
$del_{2,TA+RTC} = 5.5$
$\alpha'^{l}_{1,TA}$
$\alpha'^{l}_{1,RTC}$
0
5
10
15
0
10
20
30
40
50



CPU 1
CPU 2
Event source A
Event source B
T1
T2
T3
Event sink A
Event sink B

Case Study

Results of performance analysis

	Max delay [ms]				Max buffer [events]		
	T_1	T_2	T_3	EE_A	T_1	T_2	T_3
RTC	29	8	28.6	31.9	5	3	5
TA + RTC	25	5.5	17.2	30.5	5	2	3
TA	25	4.6	14.3	27.9	5	2	3

Run-times

	RTC	TA + RTC	TA
Total run-time	< 1s	11min	1h

Conclusions

- Hybrid and compositional analysis method that couples analytical approach (RTC) with state-based approach (TA)
- Permits to trade off analysis accuracy against verification time
- Key principle: Represent arrival curves by min/max of linear staircase functions