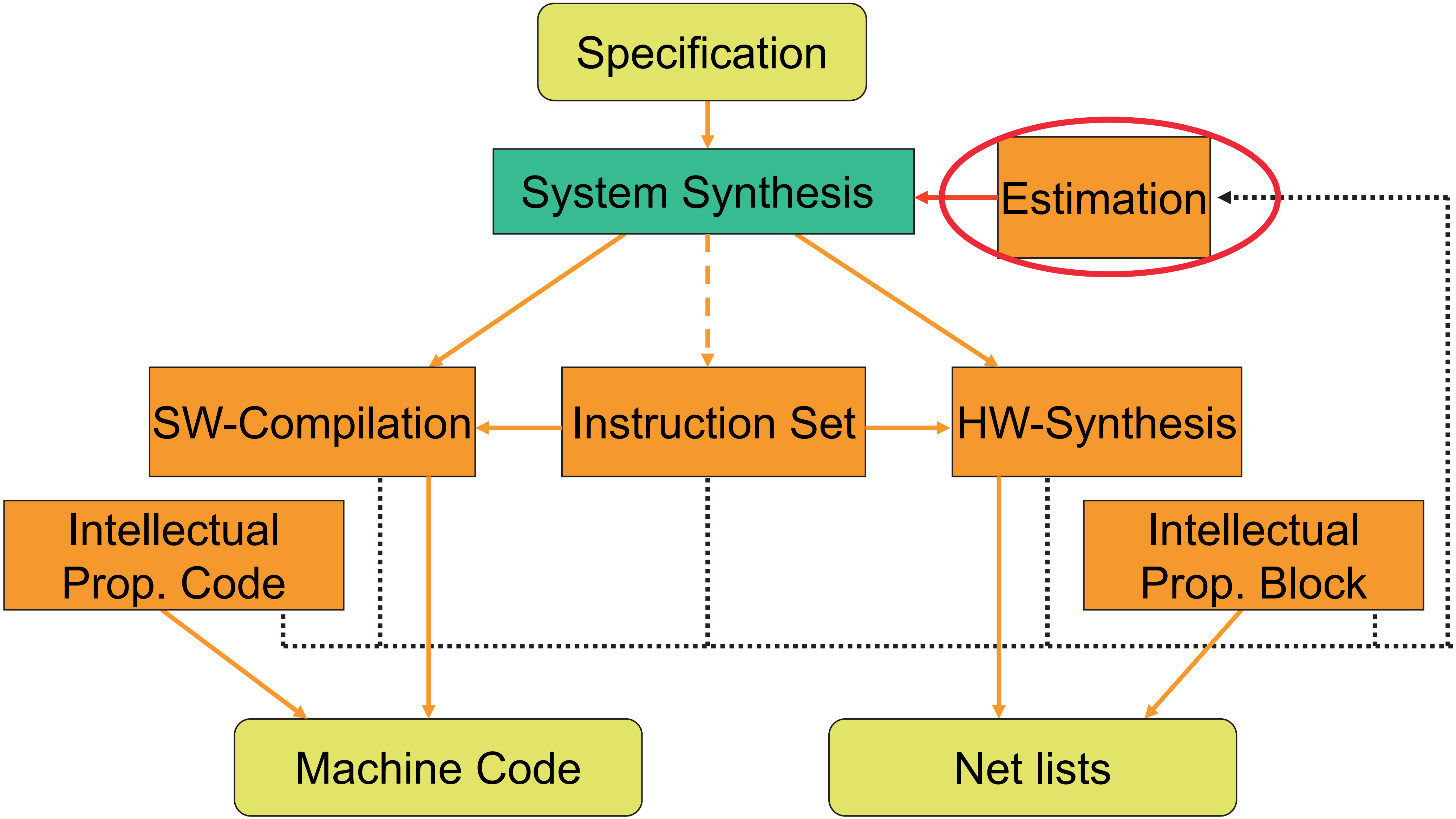
Hardware-Software Codesign

10. Performance Analysis of Distributed Embedded Systems

Lothar Thiele

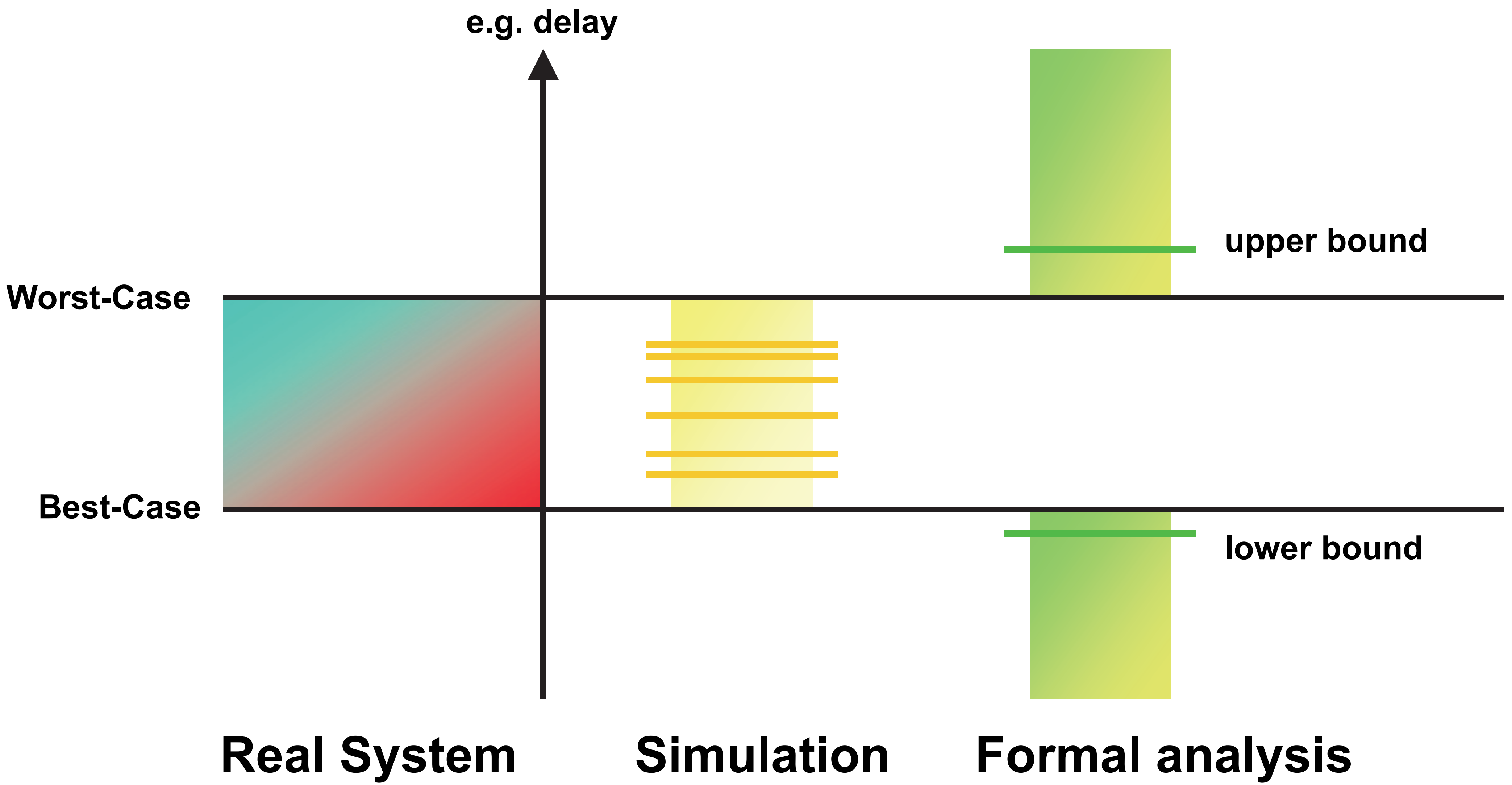
System Design



Contents

- *Overview*
- Real-Time Calculus
- Modular Performance Analysis
- Examples

Formal Analysis vs. Simulation



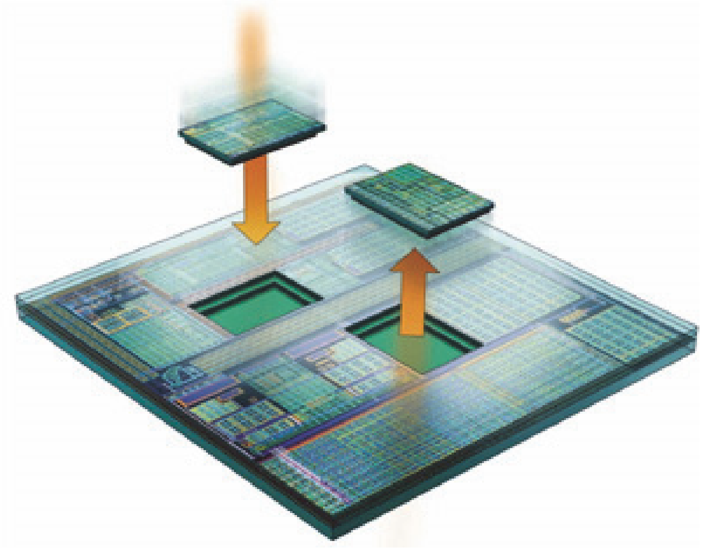
Analysis and Design

Embedded System =

Computation + Communication + Resource Interaction

Analysis:

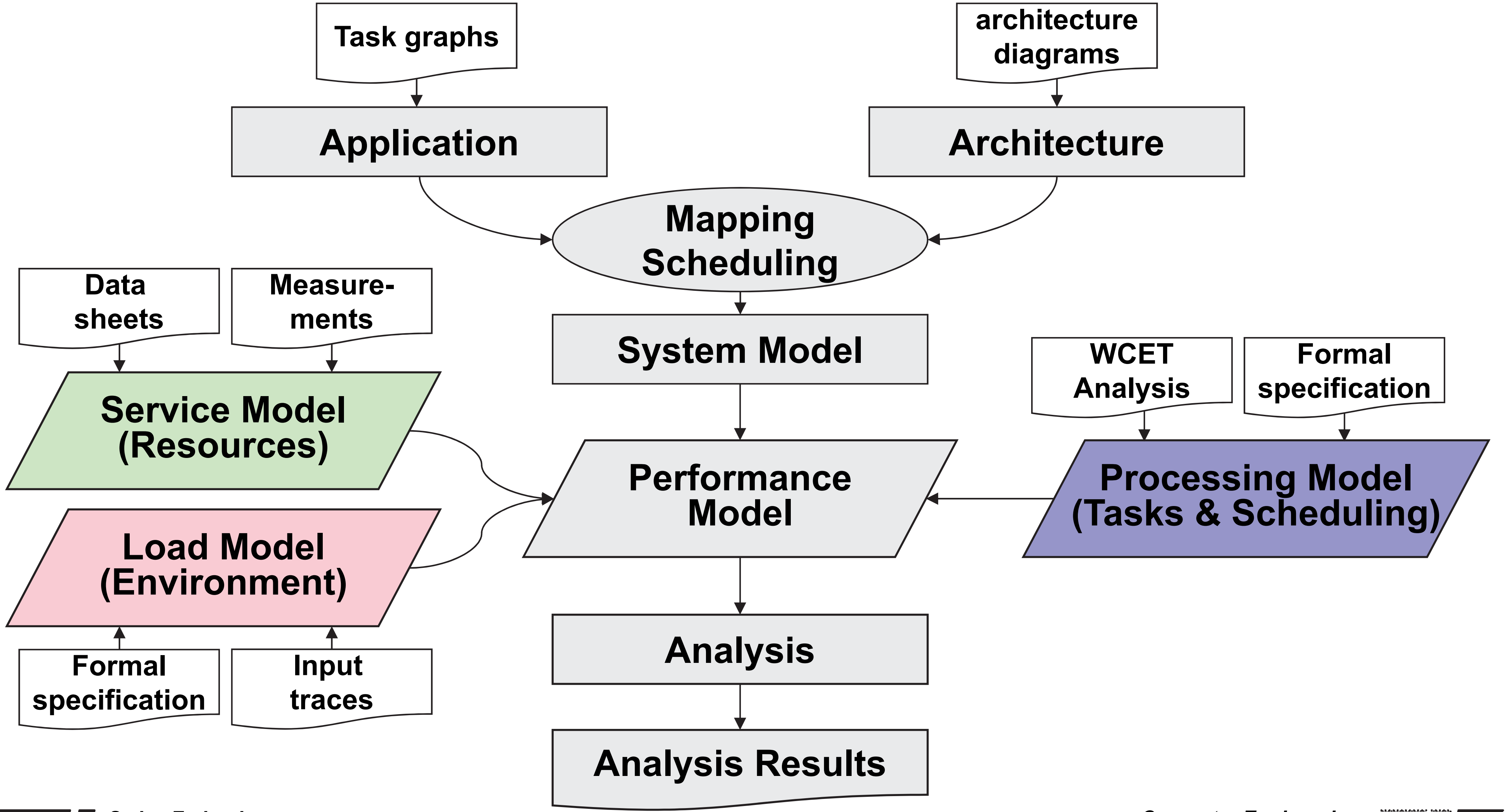
Infer system properties from subsystem properties.



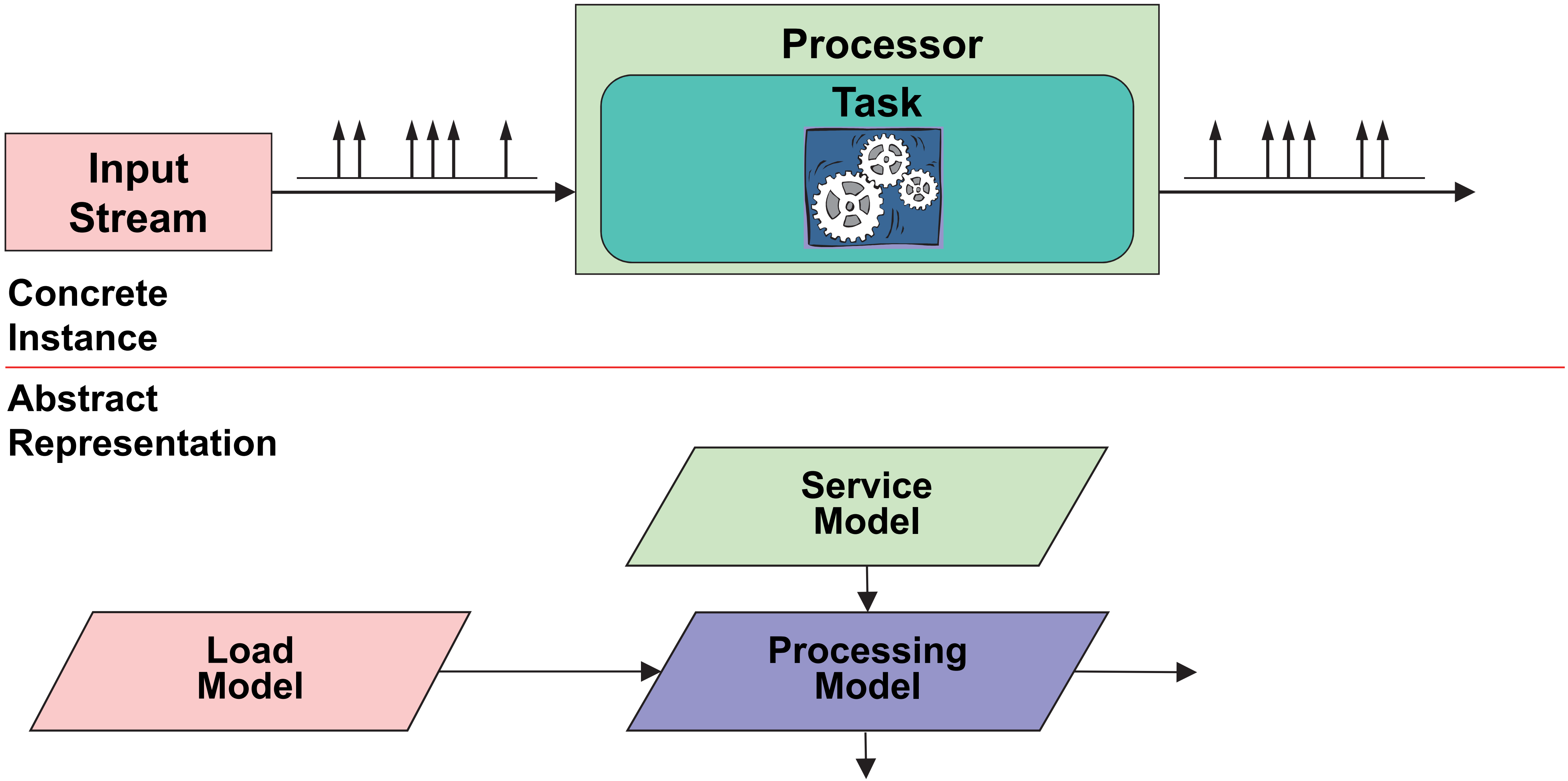
Design:

Build a system from subsystems while meeting requirements.

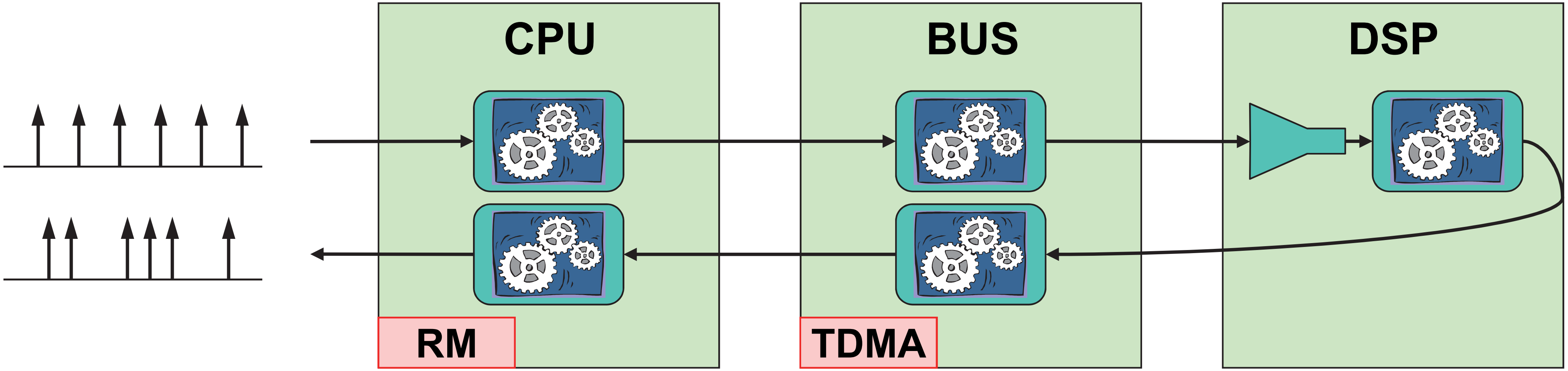
Modular Performance Analysis

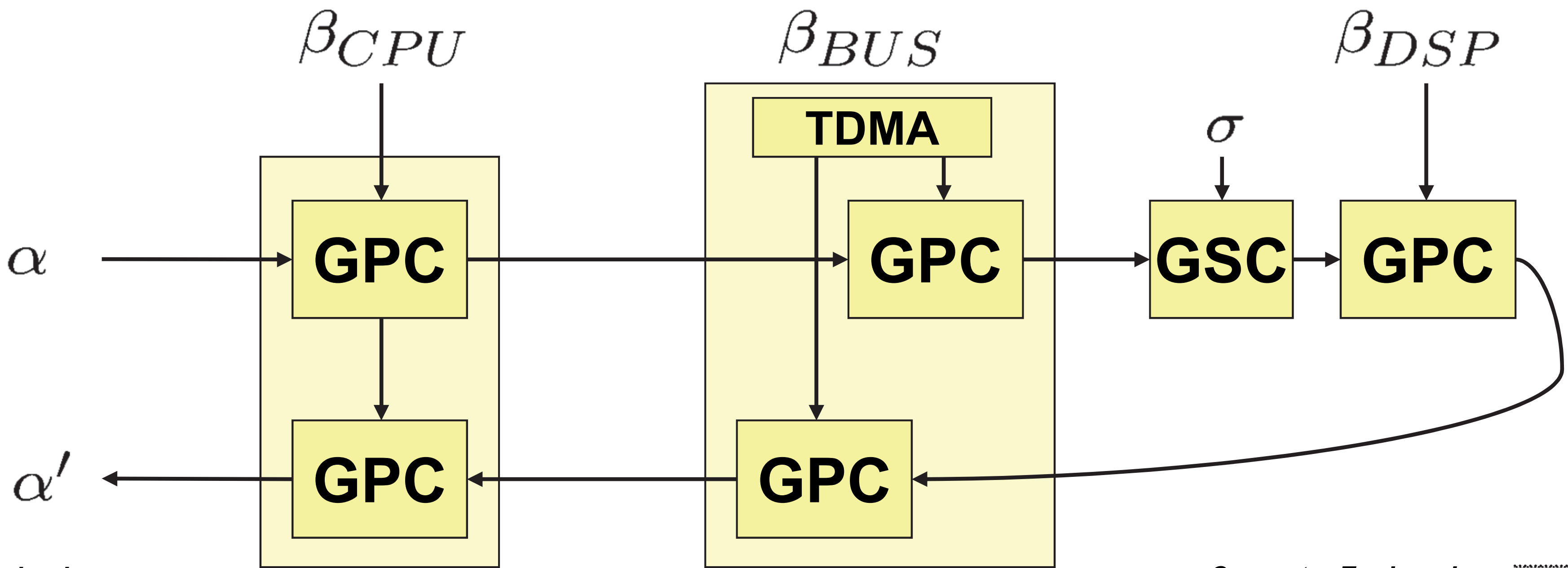


Abstract Models for Performance Analysis

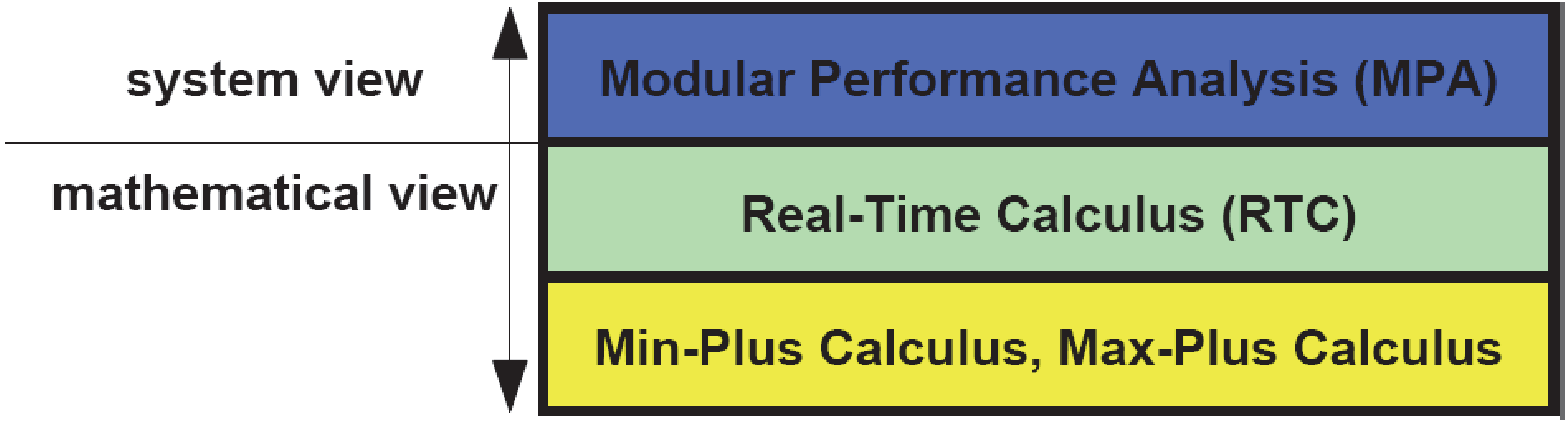


Modular System Composition





Overview


system view
mathematical view
Modular Performance Analysis (MPA)
Real-Time Calculus (RTC)
Min-Plus Calculus, Max-Plus Calculus

Contents

- Overview
- ***Real-Time Calculus***
- Modular Performance Analysis
- Examples

Foundation

- Real-Time Calculus can be regarded as a ***worst-case/best-case variant of classical queuing theory***. It is a formal method for the analysis of distributed real-time embedded systems.

- ***Related Work***:
  - *Min-Plus Algebra*: F. Baccelli, G. Cohen, G. J. Olster, and J. P. Quadrat, Synchronization and Linearity --- An Algebra for Discrete Event Systems, Wiley, New York, 1992.
  - *Network Calculus*: J.-Y. Le Boudec and P. Thiran, Network Calculus - A Theory of Deterministic Queuing Systems for the Internet, Lecture Notes in Computer Science, vol. 2050, Springer Verlag, 2001.

Comparison of Algebraic Structures

- ***Algebraic structure***
  - set of elements S
  - one or more operators defined on elements of this set
- Algebraic structures ***with two operators*** $\boxplus$, $\boxdot$
  - plus-times: $(S, \boxplus, \boxdot) = (\mathbf{R}, +, \times)$
  - min-plus: $(S, \boxplus, \boxdot) = (\mathbf{R} \cup \{+\infty\}, \inf, +)$
- ***Infimum***:
  - The infimum of a subset of some set is the greatest element, not necessarily in the subset, that is less than or equal to all other elements of the subset.
  - $\inf\{[3,4]\} = 3, \quad \inf\{(3,4]\} = 3$
    $\min\{[3,4]\} = 3, \quad \min\{(3,4]\}$ not defined

Comparison of Algebraic Structures

- ***Joint properties*** : $\boxdot$

  Closure of $\boxdot$: $a \boxdot b \in \mathcal{S}$
  Associativity of $\boxdot$: $a \boxdot (b \boxdot c) = (a \boxdot b) \boxdot c$
  Commutativity of $\boxdot$: $a \boxdot b = b \boxdot a$
  Existence of identity element for $\boxdot$: $\exists \nu : a \boxdot \nu = a$
  Existence of negative element for $\boxdot$: $\exists a^{-1} : a \boxdot a^{-1} = \nu$
  Identity element of $\boxplus$ absorbing for $\boxdot$: $a \boxdot \varepsilon = \varepsilon$
  Distributivity of $\boxdot$ w.r.t. $\boxplus$: $a \boxdot (b \boxplus c) = (a \boxdot b) \boxplus (a \boxdot c)$

- ***Example***:
  - plus-times: $a \times (b + c) = a \times b + a \times c$
  - min-plus: $a + \inf\{b, c\} = \inf\{a + b, a + c\}$

Comparison of Algebraic Structures

- ***Joint properties*** : $\boxplus$

  Closure of $\boxplus$: $a \boxplus b \in \mathcal{S}$

  Associativity of $\boxplus$: : $a \boxplus (b \boxplus c) = (a \boxplus b) \boxplus c$

  Commutativity of $\boxplus$: $a \boxplus b = b \boxplus a$

  Existence of identity element for $\boxplus$: $\exists \varepsilon : a \boxplus \varepsilon = a$

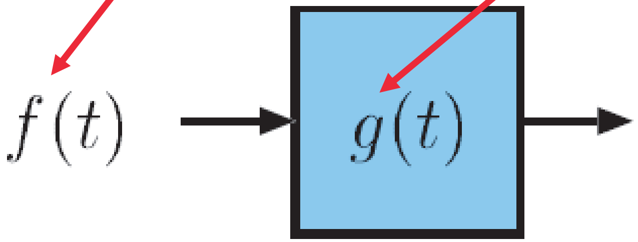
- ***Differences*** $\boxplus$:
  - *plus-times*: Existence of a negative element for $\boxplus$:

    $$\exists(-a) : a \boxplus (-a) = \varepsilon$$

  - *min-plus*: Idempotency of $\boxplus$: $a \boxplus a = a$

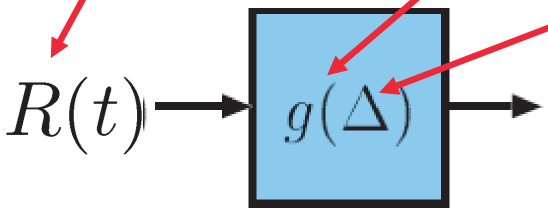
Comparison of System Theories

- ***Plus-times system theory***
  - signals, impulse response, convolution, time-domain



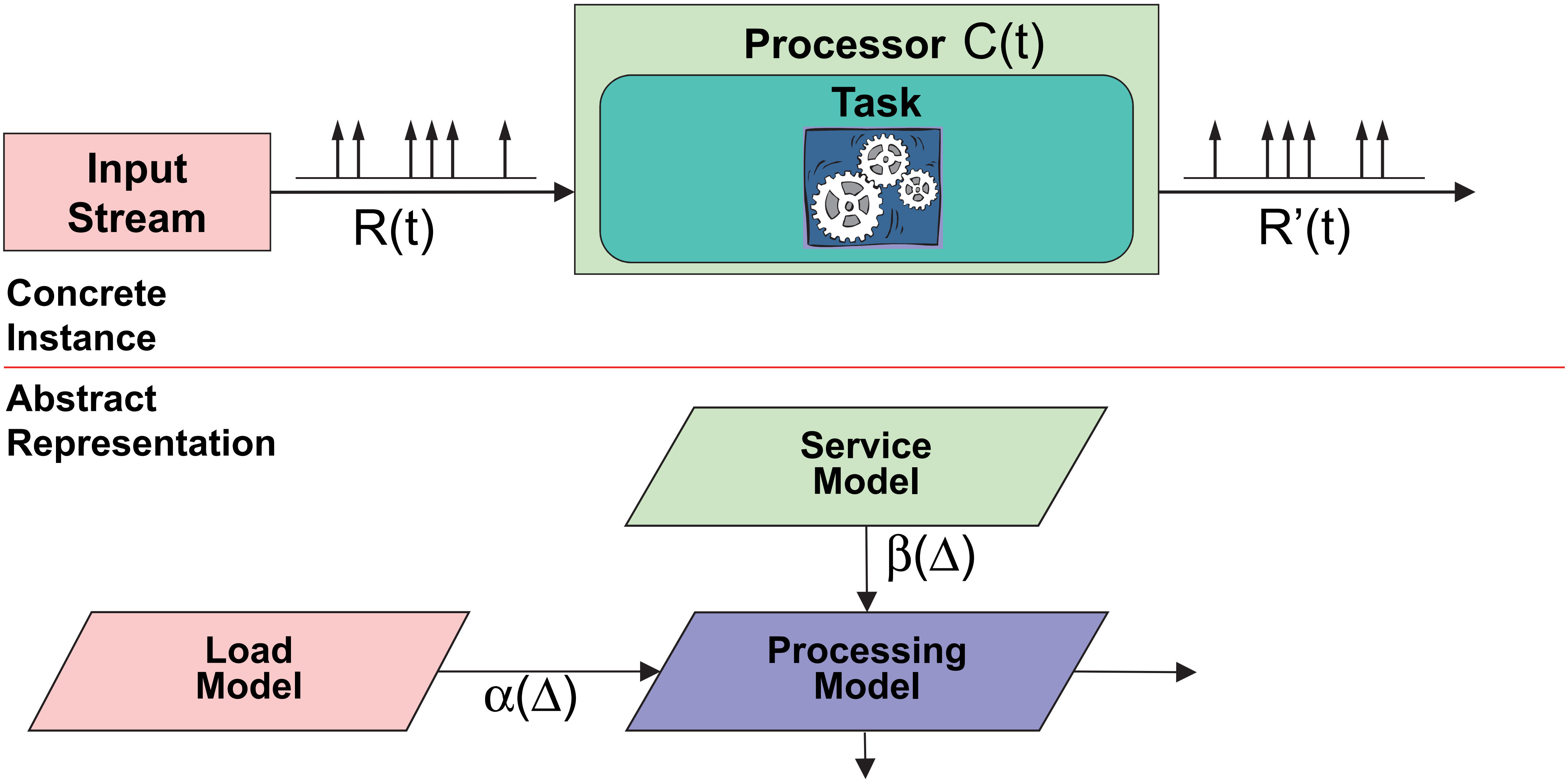
$$h(t) = (f * g)(t) = \int_0^t f(t-s) \cdot g(s)\, ds$$

- ***Min-plus system theory***
  - streams, variability curves, time-interval domain, convolution



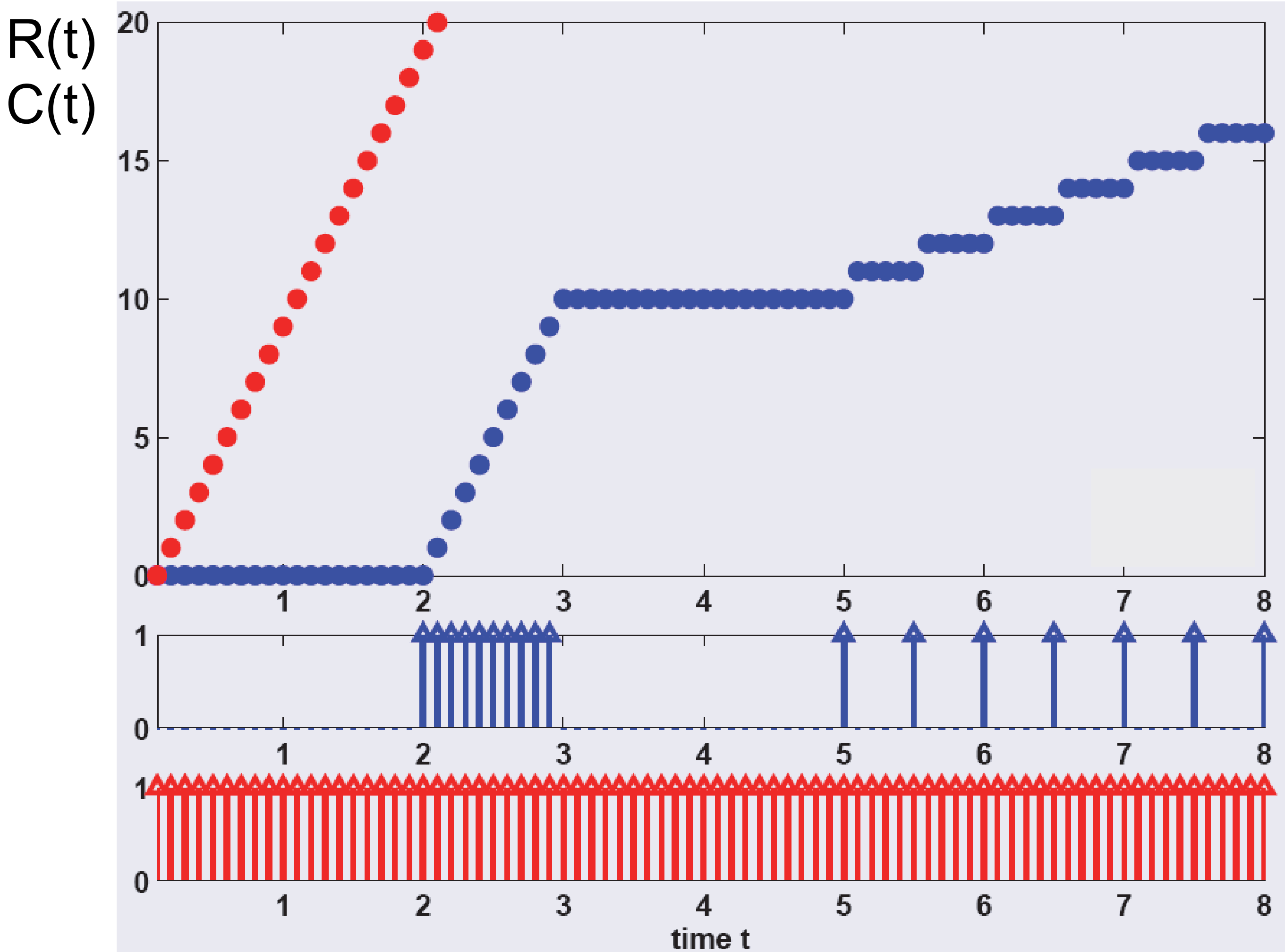
$$R'(t) \geq (R \otimes g)(t) = \inf_{0 \leq \lambda \leq t} \{R(t-\lambda) + g(\lambda)\}$$

Abstract Models for Performance Analysis



From Streams to Cumulative Functions

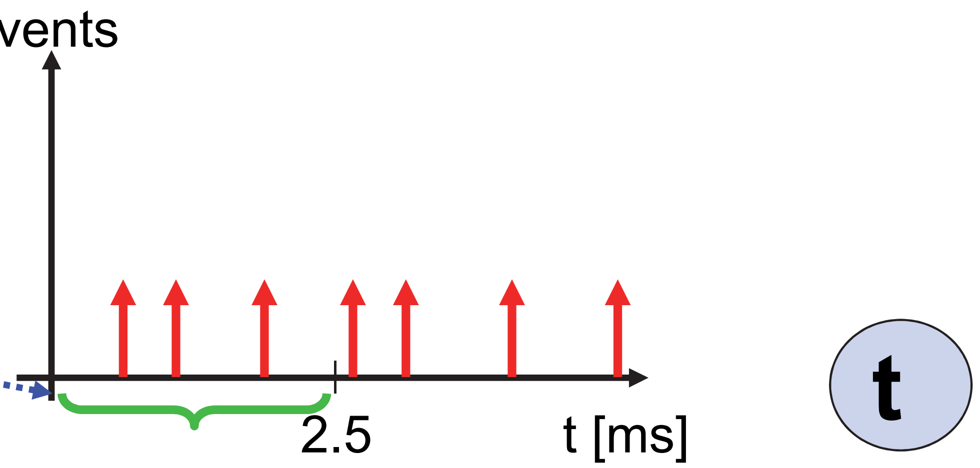
- *Data streams*: R(t) = number of events in [0, t)
- *Resource stream*: C(t) = available resource in [0, t)



From Event Streams to Arrival Curves

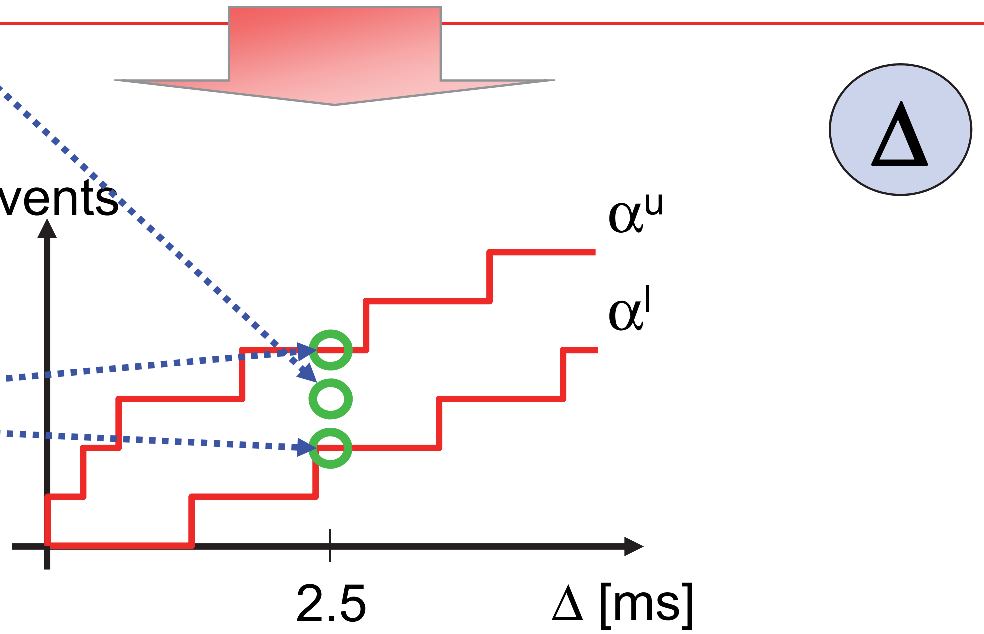
Event Stream

number of events in
in t=[0 .. 2.5] ms



Arrival Curves $\alpha = [\alpha^l, \alpha^u]$

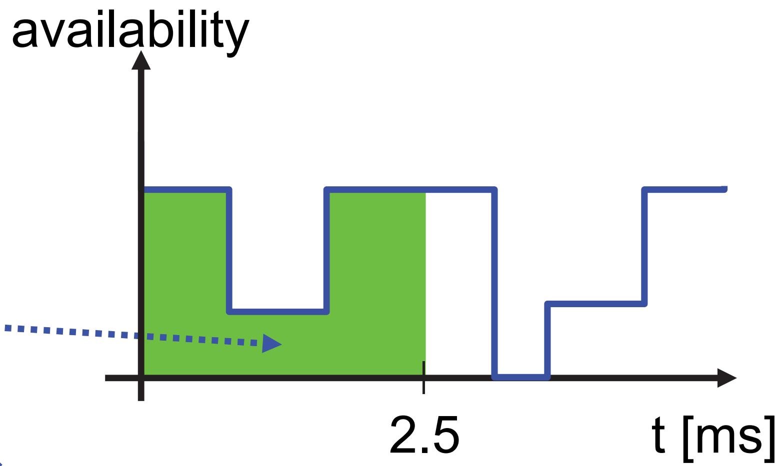
maximum / minimum
arriving events in *any interval* of length 2.5 ms



From Resources to Service Curves

Resource Availability

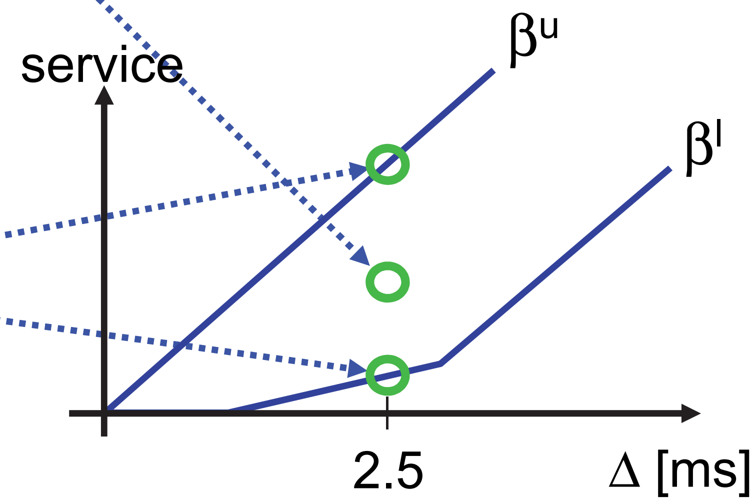
available service
in t=[0 .. 2.5] ms



t

Service Curves $\beta = [\beta^l, \beta^u]$

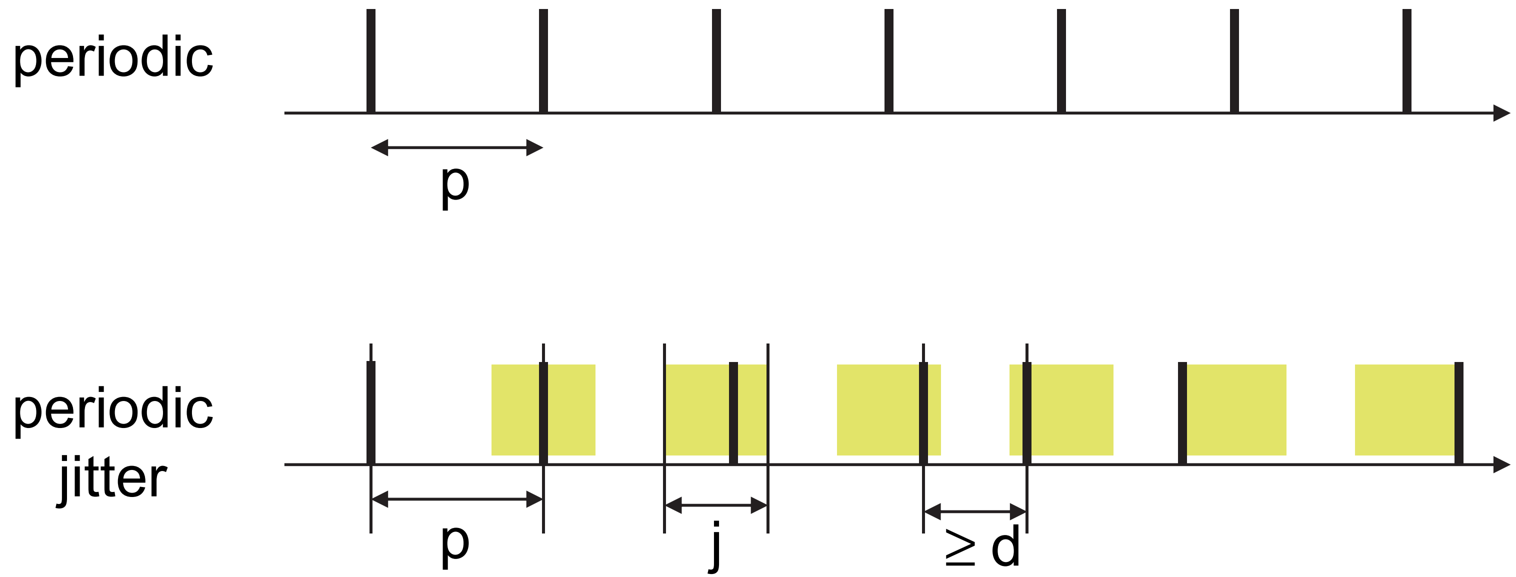
maximum/minimum
available service in *any interval* of length 2.5 ms



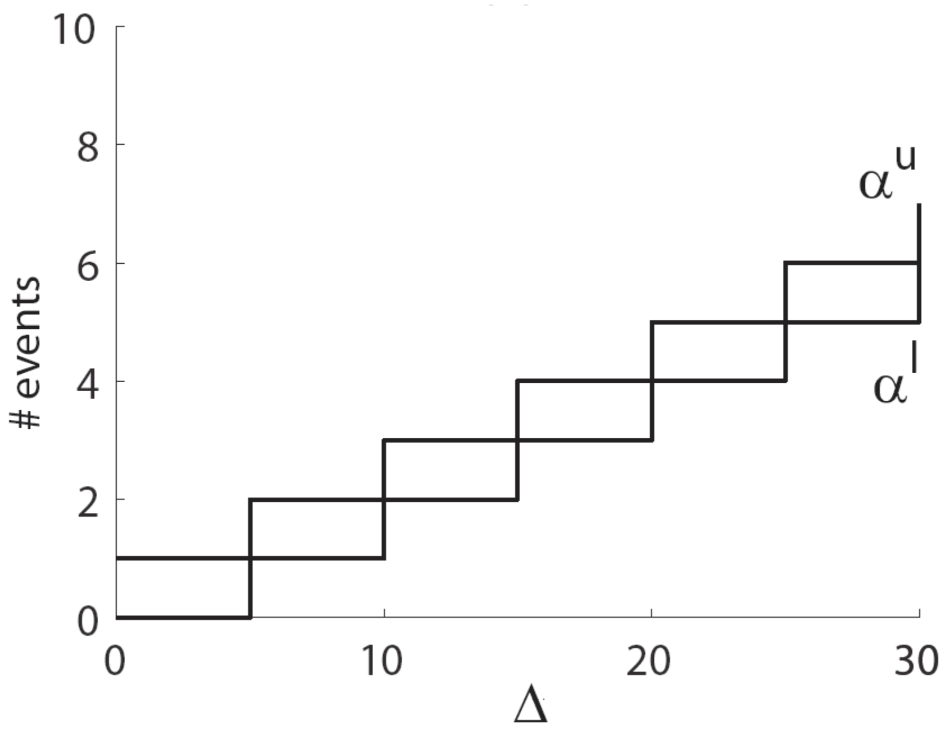
Δ

Example 1: Periodic with Jitter

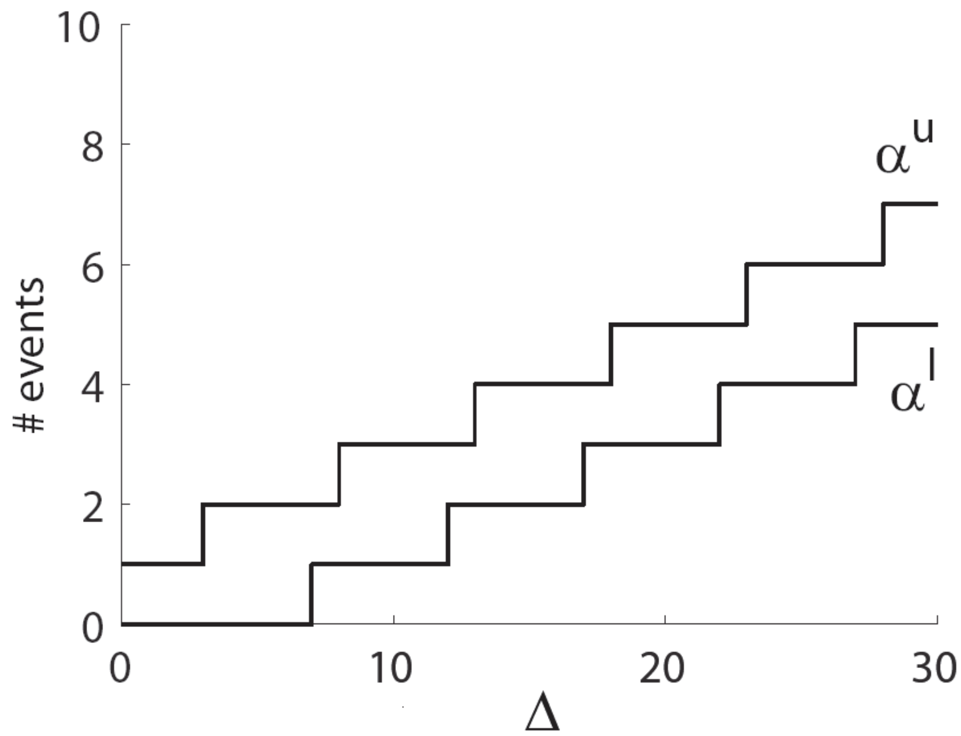
- A ***common event pattern*** that is used in literature can be specified by the parameter triple (*p, j, d*), where *p* denotes the period, *j* the jitter, and *d* the minimum inter-arrival distance of events in the modeled stream.



Example 1: Periodic with Jitter



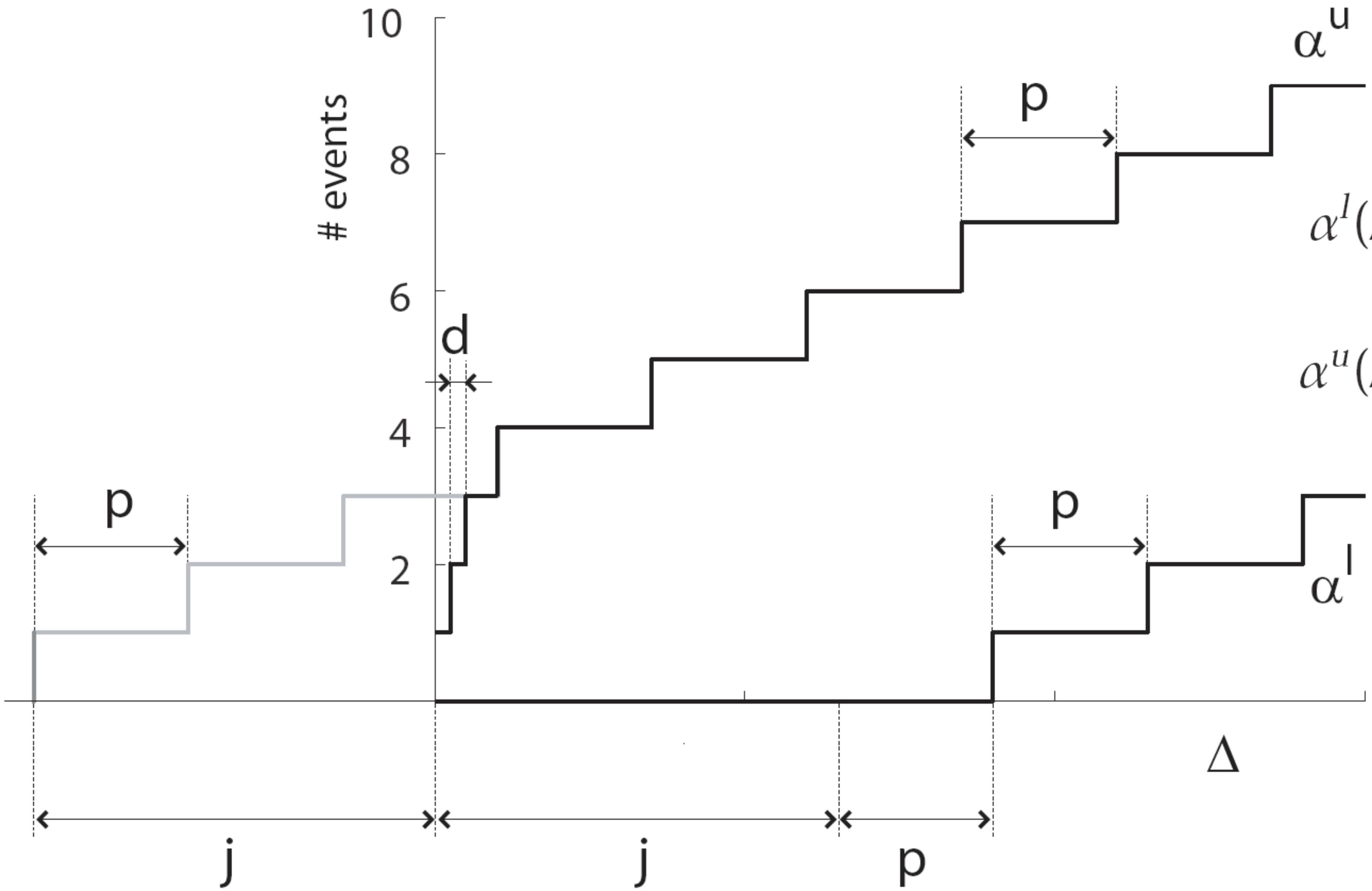
periodic



periodic with jitter

Example 1: Periodic with Jitter

- *Arrival curves*:

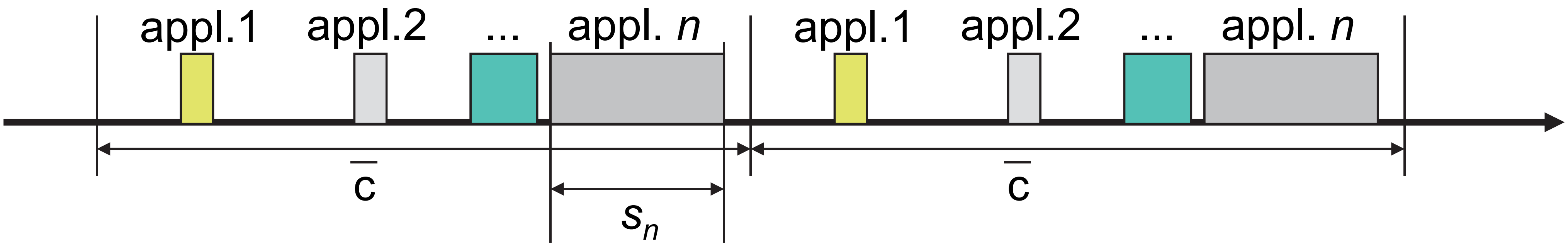


$$\alpha^l(\Delta) = \left\lfloor \frac{\Delta - j}{p} \right\rfloor$$

$$\alpha^u(\Delta) = \min\left\{ \left\lceil \frac{\Delta + j}{p} \right\rceil, \left\lceil \frac{\Delta}{d} \right\rceil \right\}$$

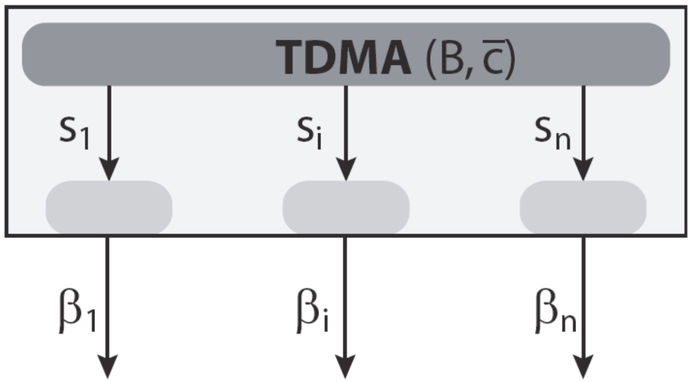
Example 2: TDMA Resource

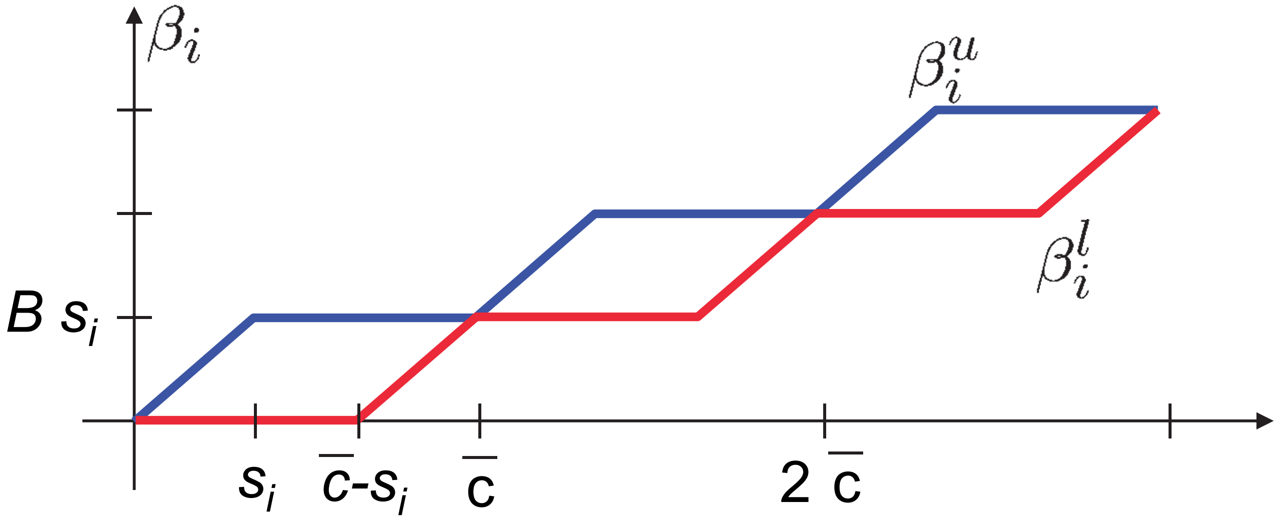
- Consider a real-time system consisting of n ***applications*** that are executed on a resource with bandwidth B that controls resource access using a ***TDMA policy***.
- Analogously, we could consider a distributed system with n ***communicating nodes***, that communicate via a shared bus with bandwidth B, with a bus arbitrator that implements a TDMA policy.
- ***TDMA policy***: In every TDMA cycle of length $\bar{c}$, one single resource slot of length s_i is assigned to application i.



Example 2: TDMA Resource

- ***Service curves*** available to the applications / node *i:*

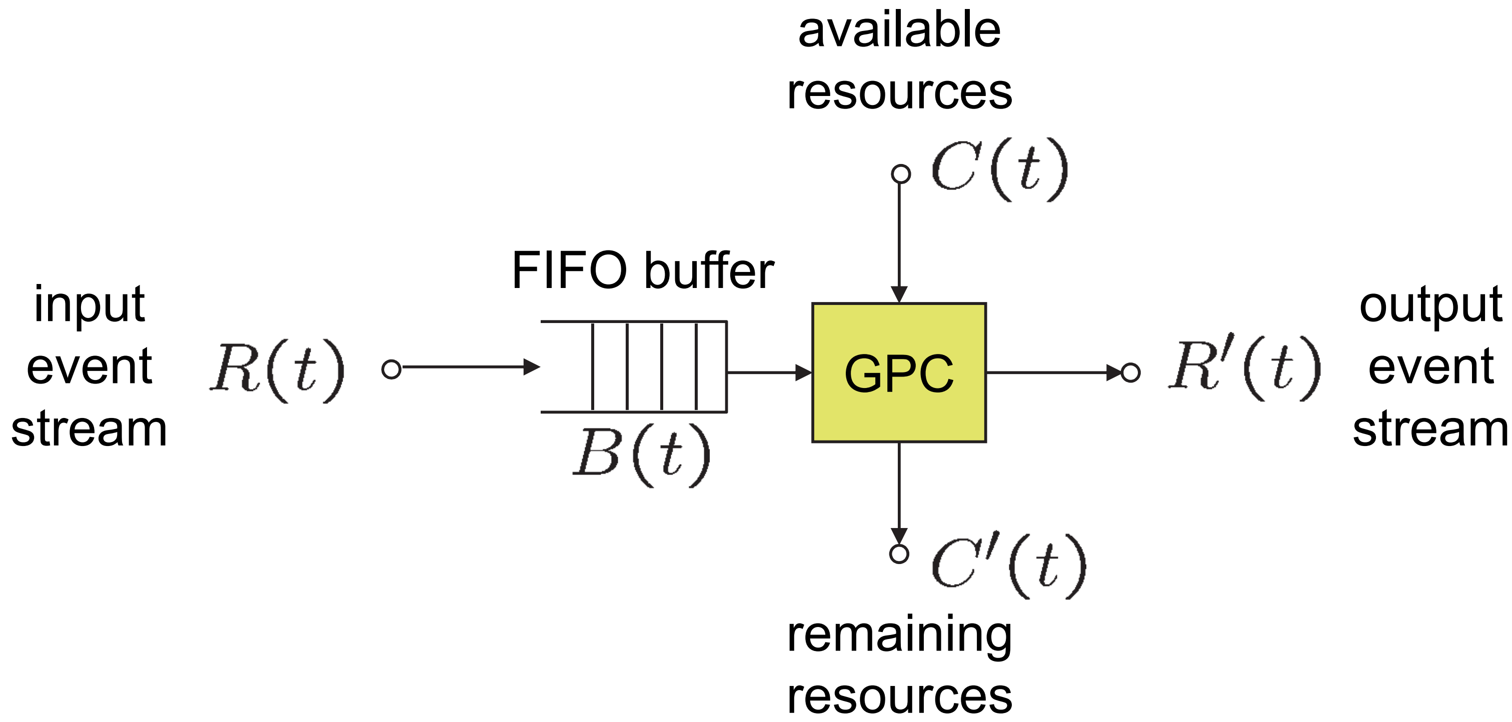




$$\beta_i^l(\Delta) = B\max\{\left\lfloor\frac{\Delta}{\bar{c}}\right\rfloor s_i, \Delta - \left\lceil\frac{\Delta}{\bar{c}}\right\rceil(\bar{c}-s_i)\}$$

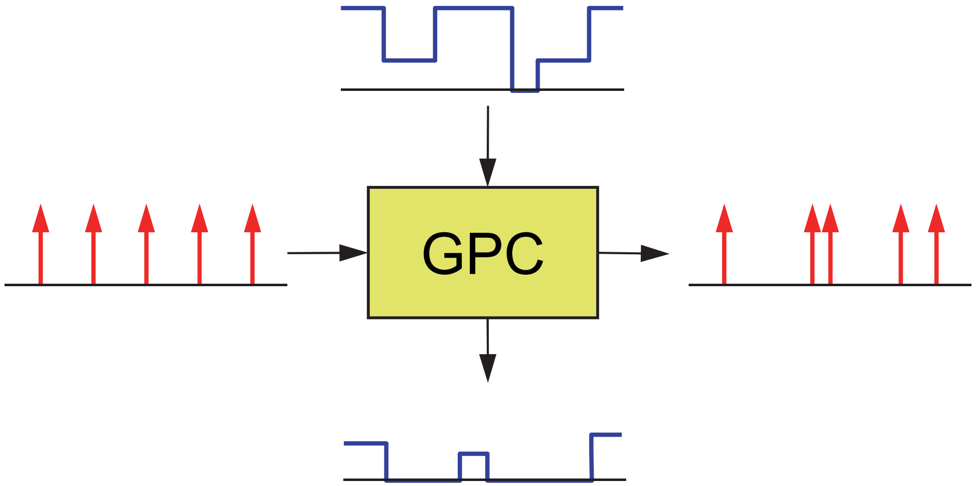
$$\beta_i^u(\Delta) = B\min\{\left\lceil\frac{\Delta}{\bar{c}}\right\rceil s_i, \Delta - \left\lfloor\frac{\Delta}{\bar{c}}\right\rfloor(\bar{c}-s_i)\}$$

Greedy Processing Component (GPC)



- ***Examples:***
  - computation (event – task instance, resource – computing resource [tasks/second])
  - communication (event – data packet, resource – bandwidth [packets/second])

Greedy Processing Component



Behavioral Description

- Component is triggered by incoming events.
- A fully preemptable task is instantiated at every event arrival to process the incoming event.
- Active tasks are processed in a greedy fashion in FIFO order.
- Processing is restricted by the availability of resources.

Greedy Processing Component (GPC)

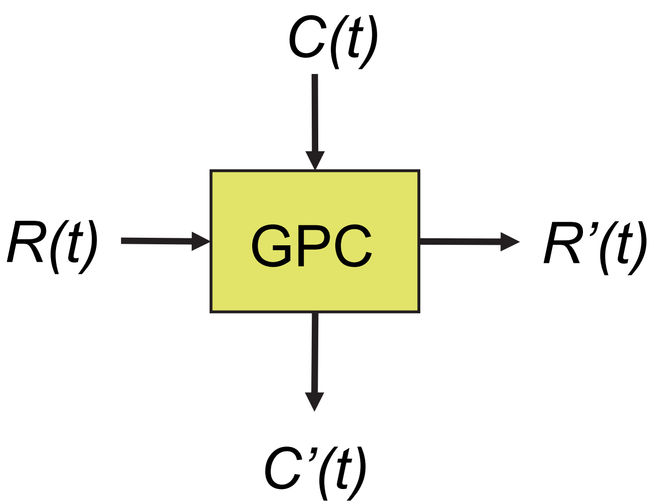
If the resource and event streams describe available and requested units of processing or communication, then

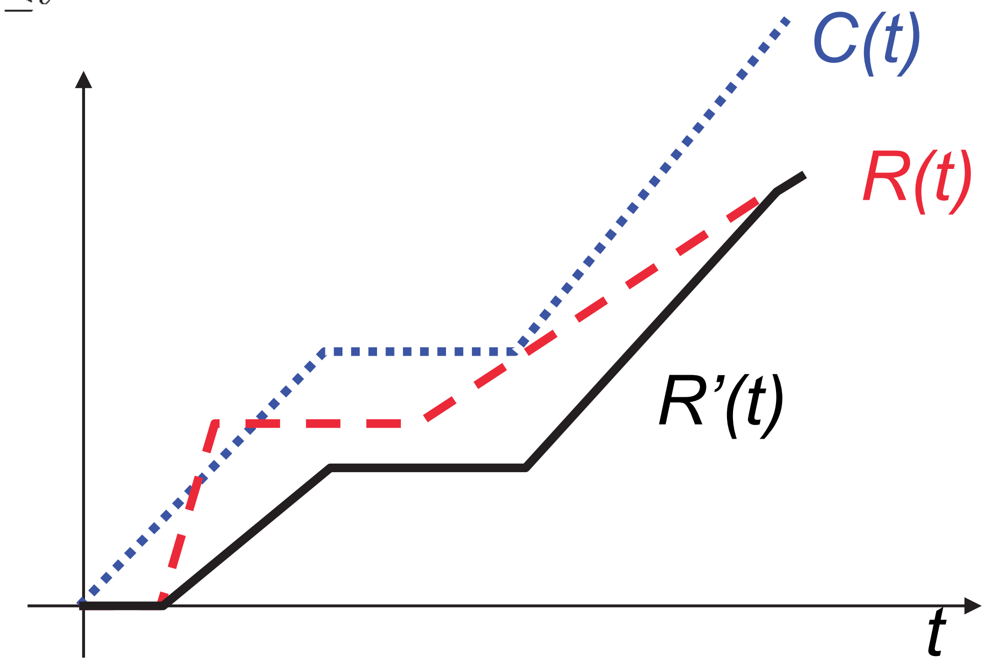
$$C(t) = C'(t) + R'(t)$$

$$B(t) = R(t) - R'(t)$$

} Conservation Laws

$$R'(t) = \inf_{0 \le u \le t} \{R(u) + C(t) - C(u)\}$$

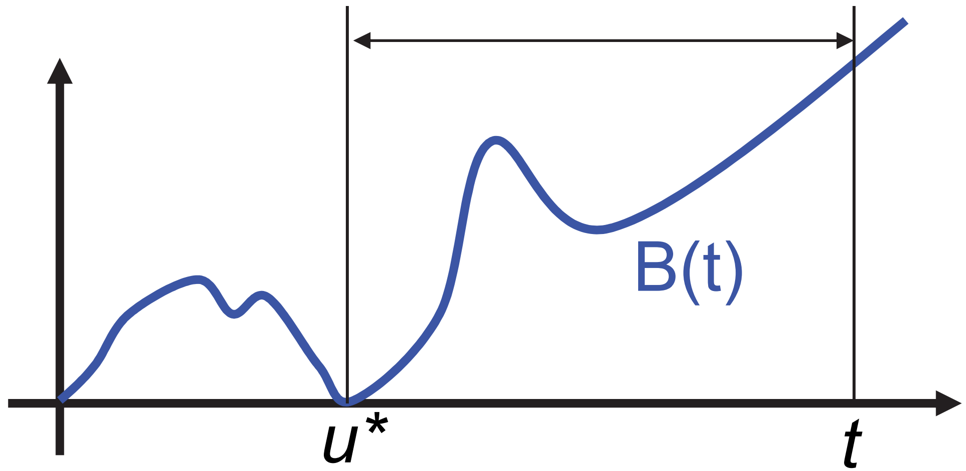




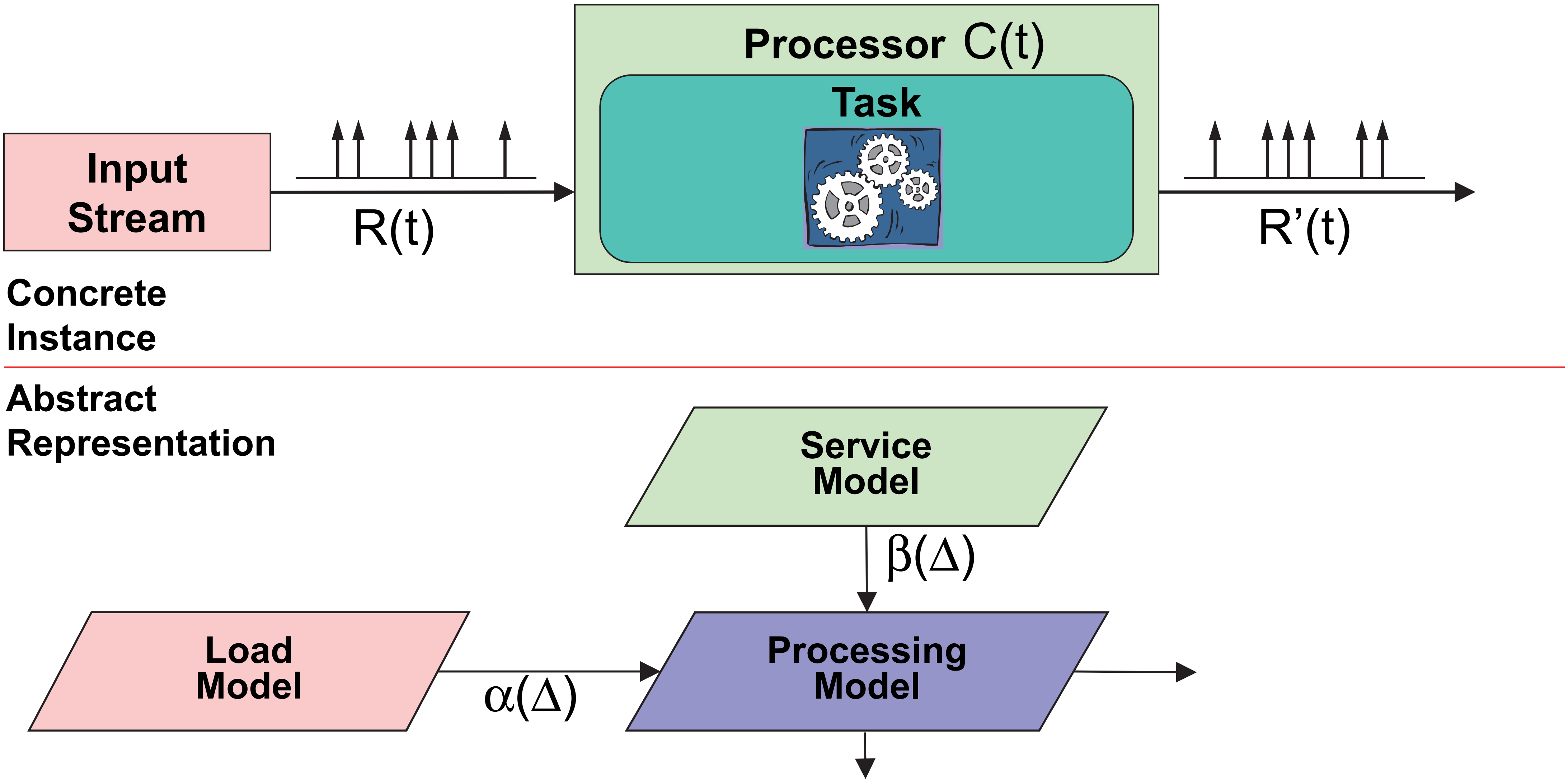
Greedy Processing

- For all times $u \leq t$ we have $R'(u) \leq R(u)$ (conservation law).
- We also have $R'(t) \leq R'(u)+C(t)-C(u)$ as the output can not be larger than the available resources.
- Combining both statements yields $R'(t) \leq R(u) + C(t) - C(u)$.
- Let us suppose that u^* is the last time before t with an empty buffer. We have $R(u^*) = R'(u^*)$ at u^* and also $R'(t) = R'(u^*) + C(t) - C(u^*)$ as all available resources are used to produce output. Therefore, $R'(t) = R(u^*) + C(t) - C(u^*)$.
- As a result, we obtain

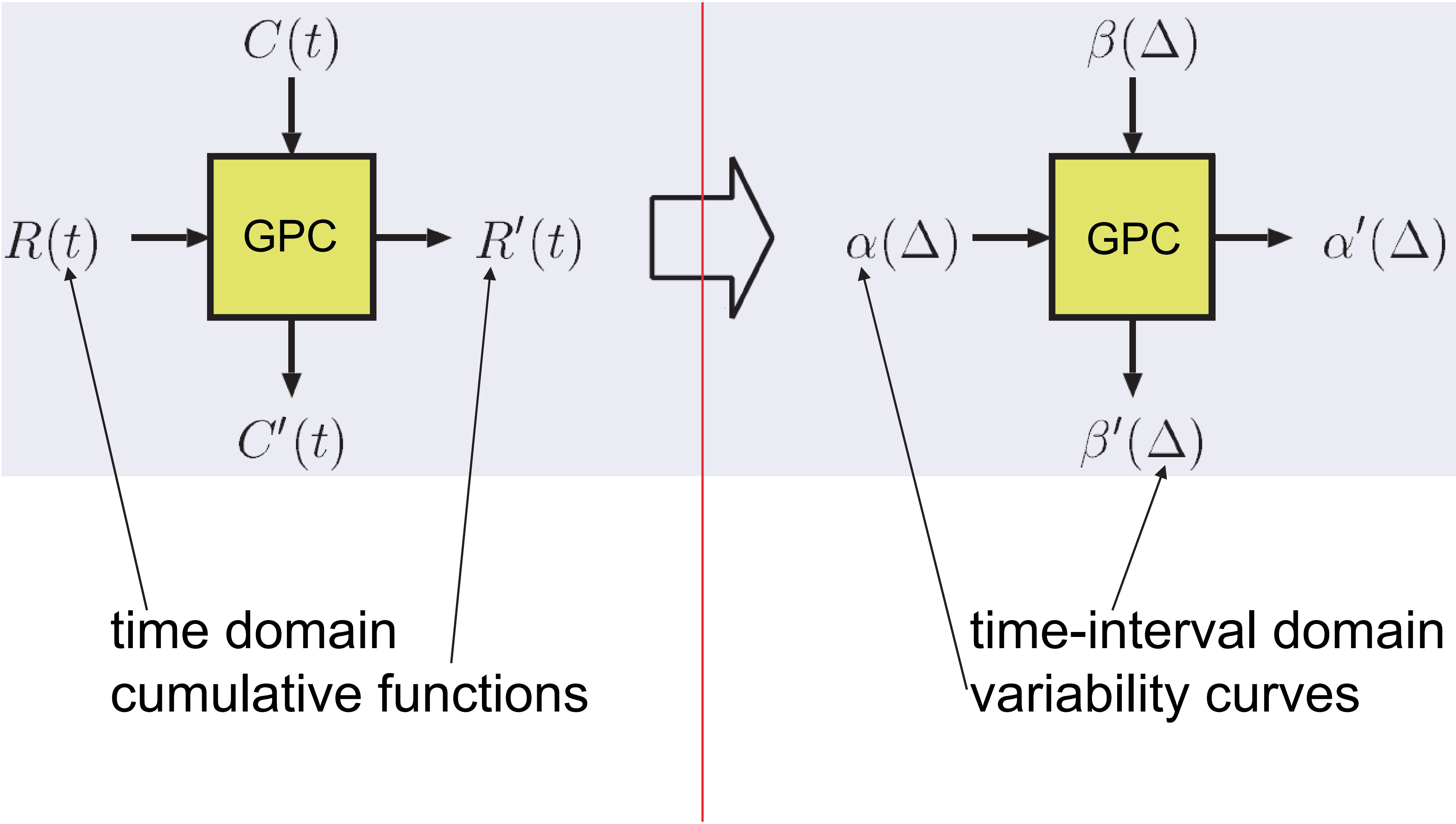
$$R'(t) = \inf_{0 \leq u \leq t} \{R(u) + C(t) - C(u)\}$$



Abstract Models for Performance Analysis



Abstraction



Some Definitions and Relations

- $f \otimes g$ is called ***min-plus convolution***

$$(f \otimes g)(t) = \inf_{0 \le u \le t} \{f(t-u) + g(u)\}$$

- $f \oslash g$ is called ***min-plus de-convolution***

$$(f \oslash g)(t) = \sup_{u \ge 0} \{f(t+u) - g(u)\}$$

- For ***max-plus convolution and de-convolution***:

$$(f \overline{\otimes} g)(t) = \sup_{0 \le u \le t} \{f(t-u) + g(u)\}$$

$$(f \overline{\oslash} g)(t) = \inf_{u \ge 0} \{f(t+u) - g(u)\}$$

- Relation between ***convolution and deconvolution***

$$f \le g \otimes h \Leftrightarrow f \oslash h \le g$$

Arrival and Service Curve

- The arrival and service curves provide bounds on event and resource functions as follows:

$$\alpha^l(t-s) \le R(t) - R(s) \le \alpha^u(t-s) \quad \forall s \le t$$

$$\beta^l(t-s) \le C(t) - C(s) \le \beta^u(t-s) \quad \forall s \le t$$

- We can determine valid variability curves from cumulative functions as follows:

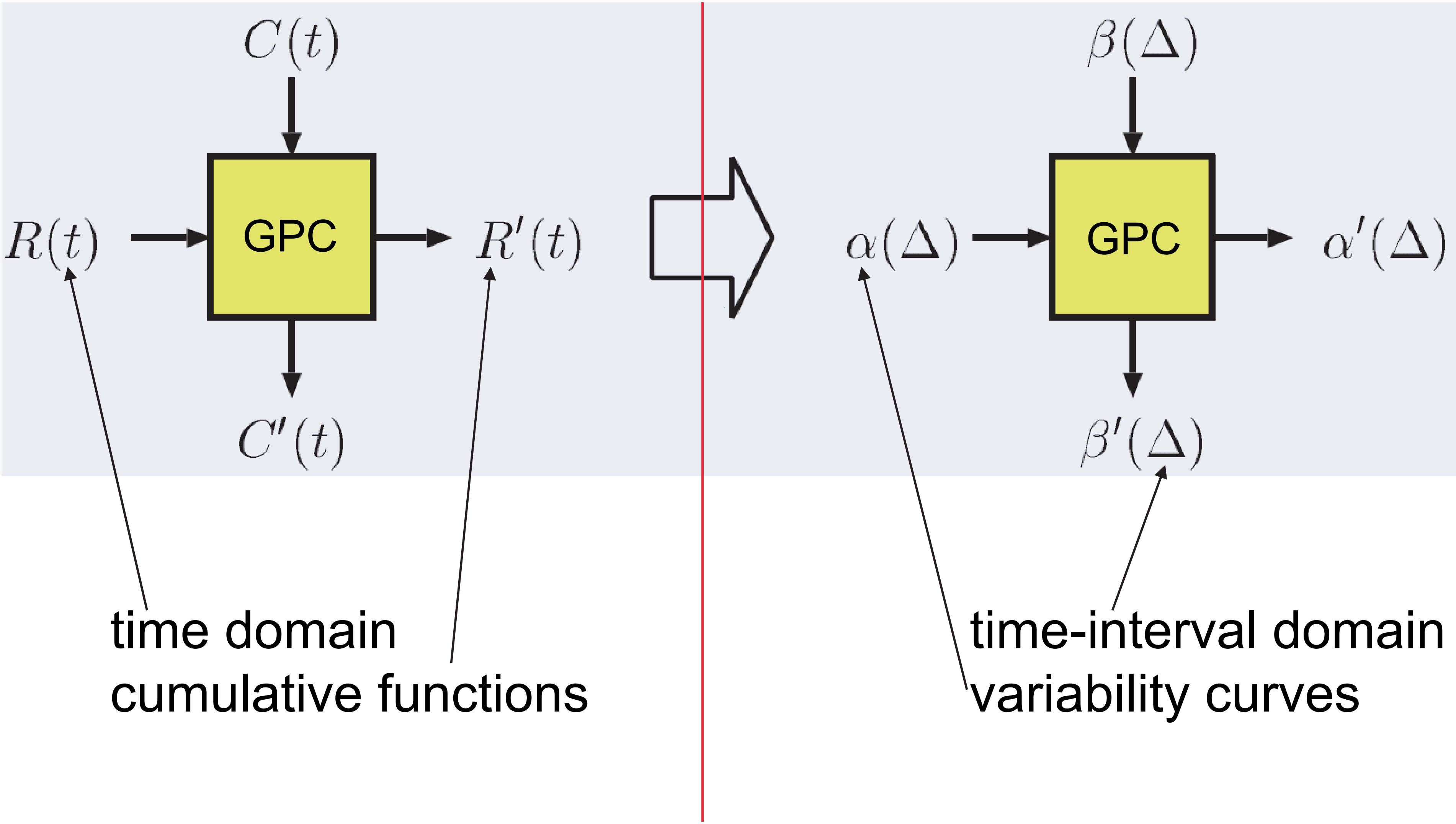
$$\alpha^u = R \oslash R; \quad \alpha^l = R \,\overline{\oslash}\, R; \quad \beta^u = C \oslash C; \quad \beta^l = C \,\overline{\oslash}\, C$$

- One proof:

$$\alpha^u = R \oslash R \Rightarrow \alpha^u(\Delta) = \sup_{u \ge 0} \{R(\Delta + u) - R(u)\} \Rightarrow$$

$$\alpha^u(\Delta) = \sup_{s \ge 0} \{R(\Delta + s) - R(s)\} \Rightarrow \alpha^u(t-s) \ge R(t) - R(s) \ \ \forall t \ge s$$

Abstraction



The Most Simple Relations

- The *output stream* of a component satisfies:

$$R'(t) \geq (R \otimes \beta^l)(t)$$

- The *output upper arrival curve* of a component satisfies:

$$\alpha^{u\prime} = (\alpha^u \oslash \beta^l)$$

- The *remaining lower service curve* of a component satisfies:

$$\beta^{l\prime}(\Delta) = \sup_{0 \leq \lambda \leq \Delta} (\beta^l(\lambda) - \alpha^u(\lambda))$$

Two Sample Proofs

$R'(t) \geq (R \otimes \beta^l)(t)$

$$\begin{aligned} R'(t) &= \inf_{0 \leq u \leq t} \{R(u) + C(t) - C(u)\} \\ &\geq \inf_{0 \leq u \leq t} \{R(u) + \beta^l(t-u)\} \\ &= (R \otimes \beta^l)(t) \end{aligned}$$

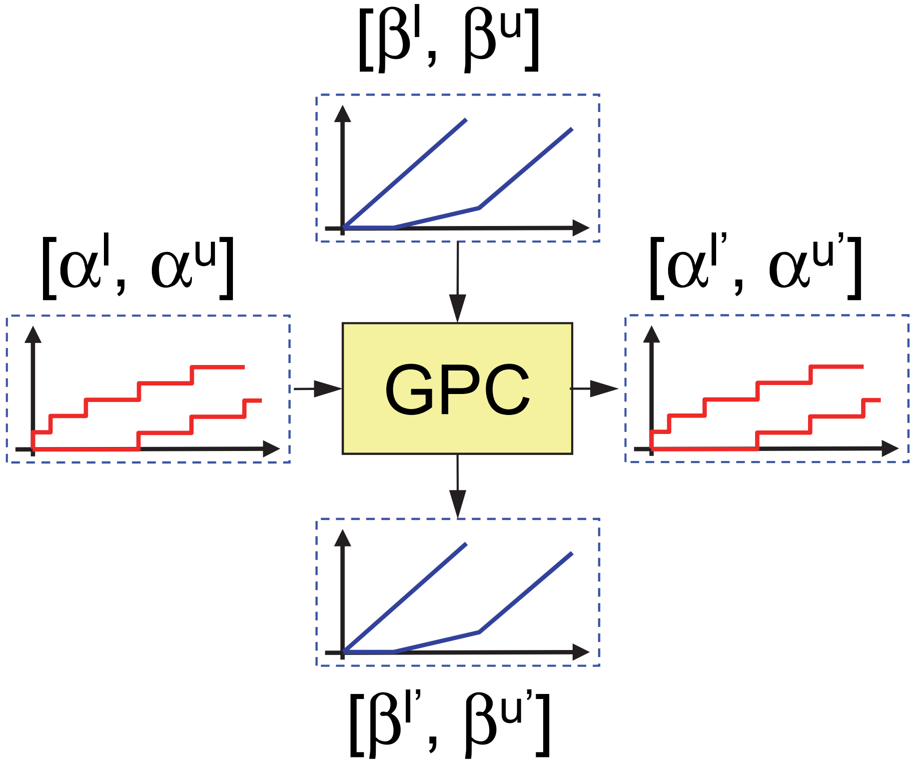
$\beta^{l'}(\Delta) = \sup_{0 \leq \lambda \leq \Delta} (\beta^l(\lambda) - \alpha^u(\lambda))$

$$\begin{aligned} C'(t) - C'(s) &= \sup_{0 \leq a \leq t} \{C(a) - R(a)\} - \sup_{0 \leq b \leq s} \{C(b) - R(b)\} = \\ &= \inf_{0 \leq b \leq s} \{ \sup_{0 \leq a \leq t} \{(C(a) - C(b)) - (R(a) - R(b))\}\} \\ &= \inf_{0 \leq b \leq s} \{ \sup_{0 \leq a-b \leq t-b} \{(C(a) - C(b)) - (R(a) - R(b))\}\} \\ &\geq \inf_{0 \leq b \leq s} \{ \sup_{0 \leq \lambda \leq t-b} \{\beta^l(\lambda) - \alpha^u(\lambda)\}\} \geq \sup_{0 \leq \lambda \leq t-s} \{\beta^l(\lambda) - \alpha^u(\lambda)\} \end{aligned}$$

Tighter Bounds

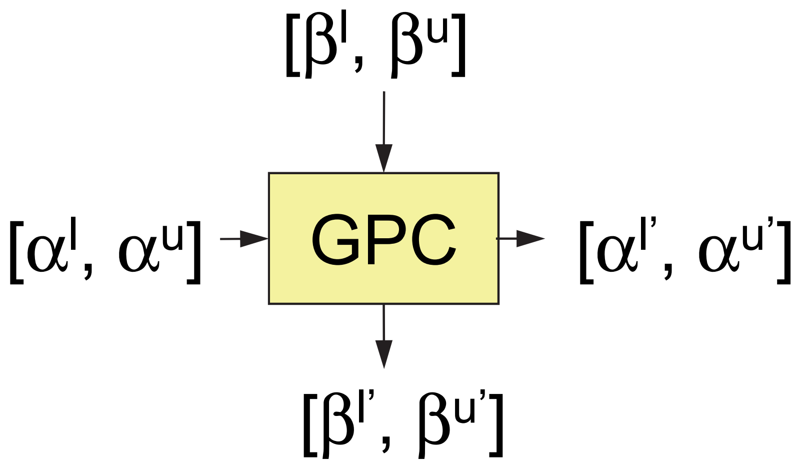
The greedy processing component transforms the variability curves as follows:

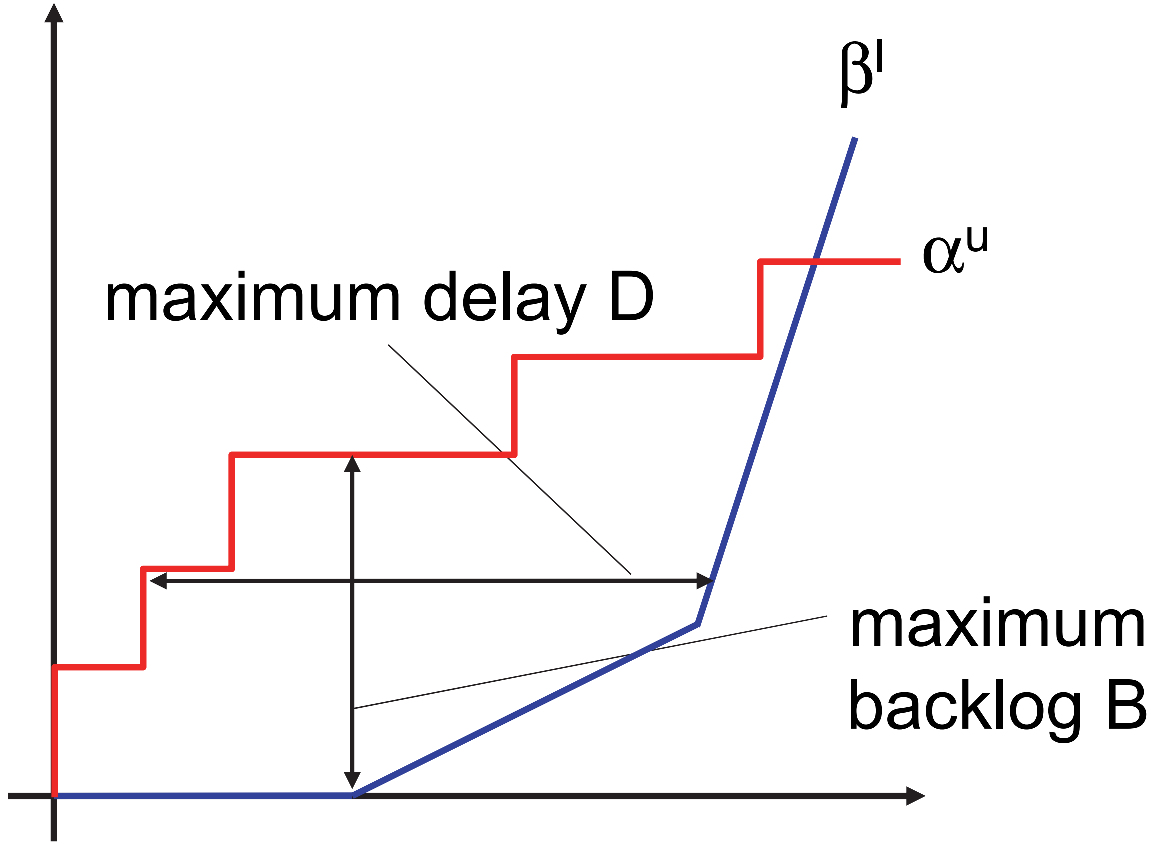
$$\alpha^{u\prime} = [(\alpha^u \otimes \beta^u) \oslash \beta^l] \wedge \beta^u$$
$$\alpha^{l\prime} = [(\alpha^l \oslash \beta^u) \otimes \beta^l] \wedge \beta^l$$
$$\beta^{u\prime} = (\beta^u - \alpha^l)\overline{\oslash}0$$
$$\beta^{l\prime} = (\beta^l - \alpha^u)\overline{\otimes}0$$



Without proof ... .

Delay and Backlog



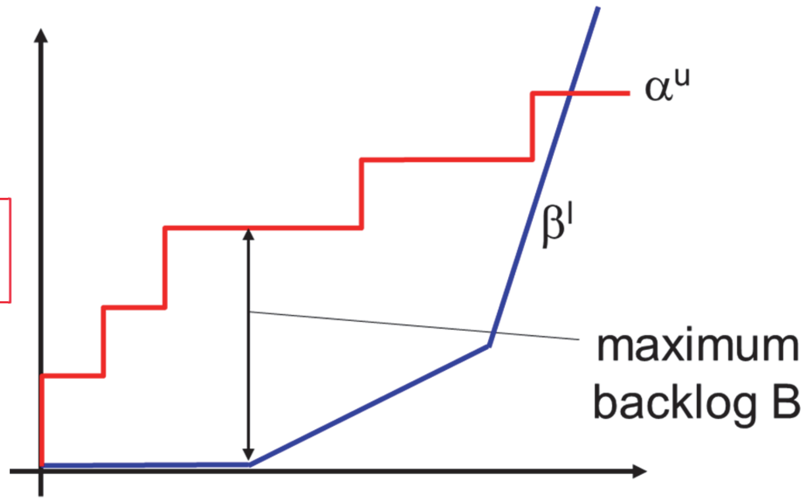


$$B = \sup_{t \geq 0} \{R(t) - R'(t)\} \leq \sup_{\lambda \geq 0} \left\{\alpha^u(\lambda) - \beta^l(\lambda)\right\}$$

$$\begin{aligned} D &= \sup_{t \geq 0} \{\inf \{\tau \geq 0 : R(t) \leq R'(t+\tau)\}\} \\ &= \sup_{\Delta \geq 0} \left\{\inf \left\{\tau \geq 0 : \alpha^u(\Delta) \leq \beta^l(\Delta + \tau)\right\}\right\} \end{aligned}$$

Proof of Backlog Bound

$$B = \sup_{t \geq 0} \{R(t) - R'(t)\} \leq \sup_{\lambda \geq 0} \left\{\alpha^u(\lambda) - \beta^l(\lambda)\right\}$$



$$B(t) = R(t) - R'(t) = R(t) - \inf_{0 \leq u \leq t} \{R(u) + C(t) - C(u)\}$$

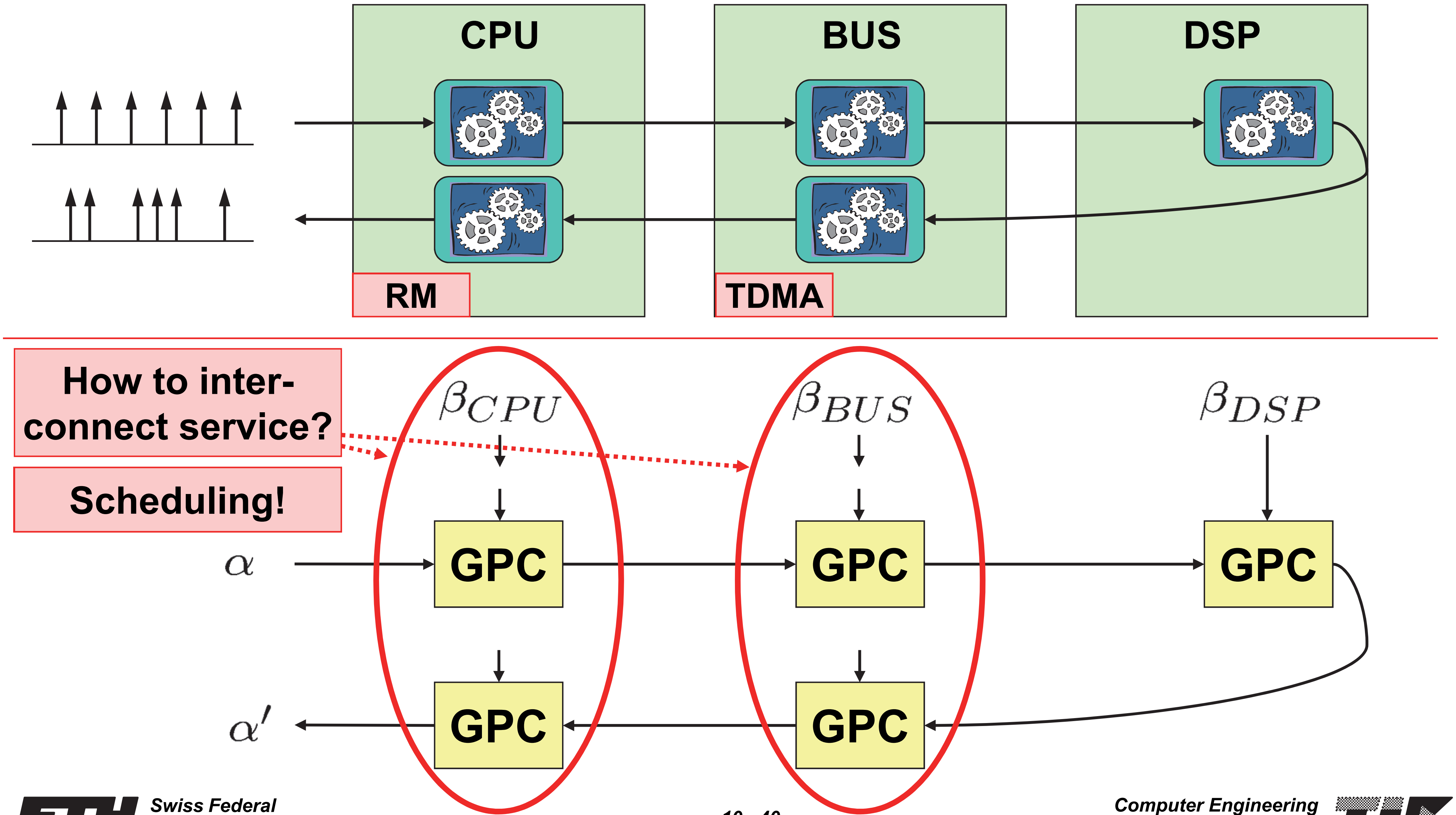
$$= \sup_{0 \leq u \leq t} \{(R(t) - R(u)) - (C(t) - C(u))\}$$

$$\leq \sup_{0 \leq u \leq t} \{\alpha^u(t-u) - \beta^l(t-u)\}$$

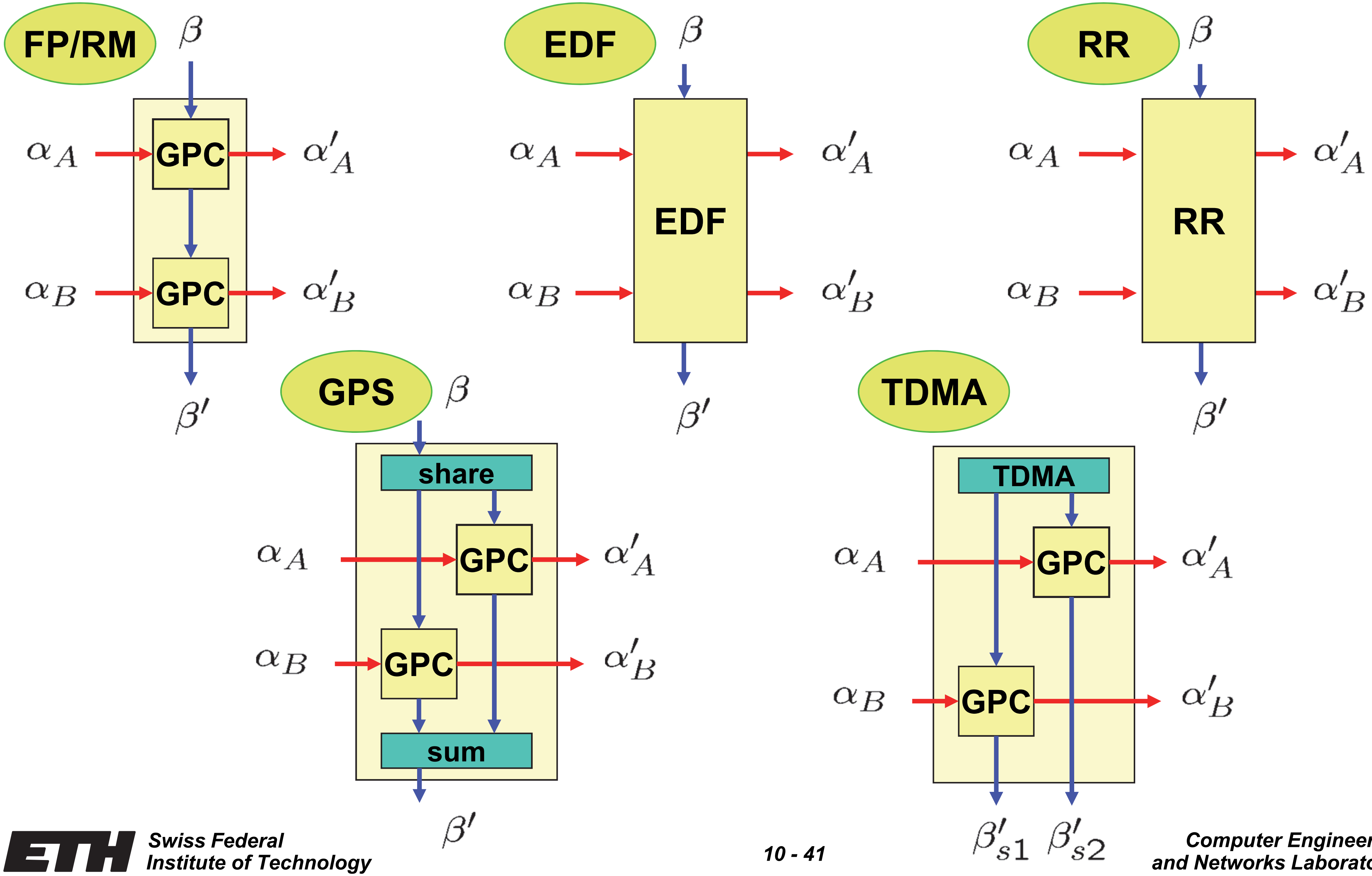
$$\leq \sup_{0 \leq \lambda} \{\alpha^u(\lambda) - \beta^l(\lambda)\}$$

Contents

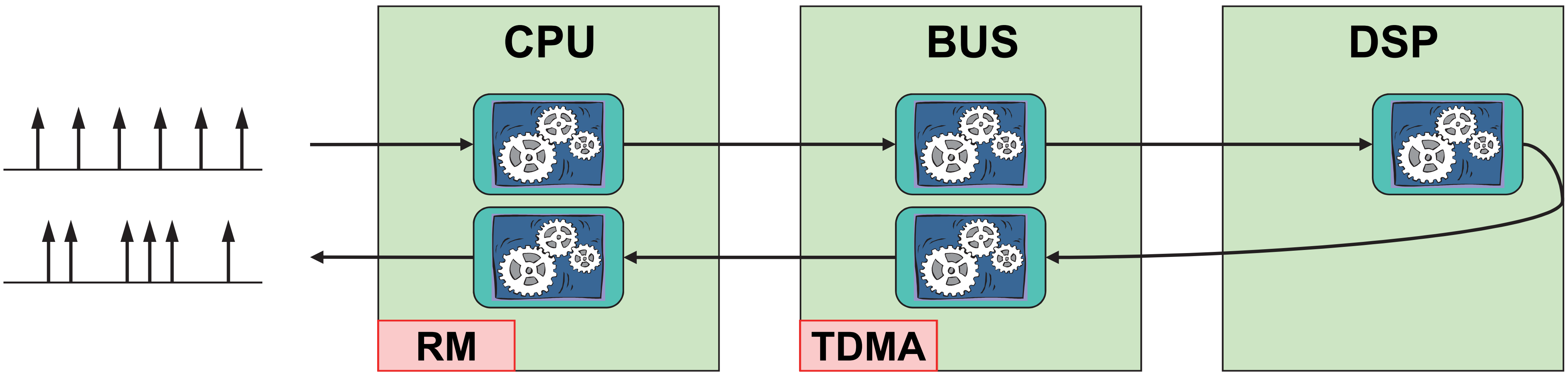
System Composition

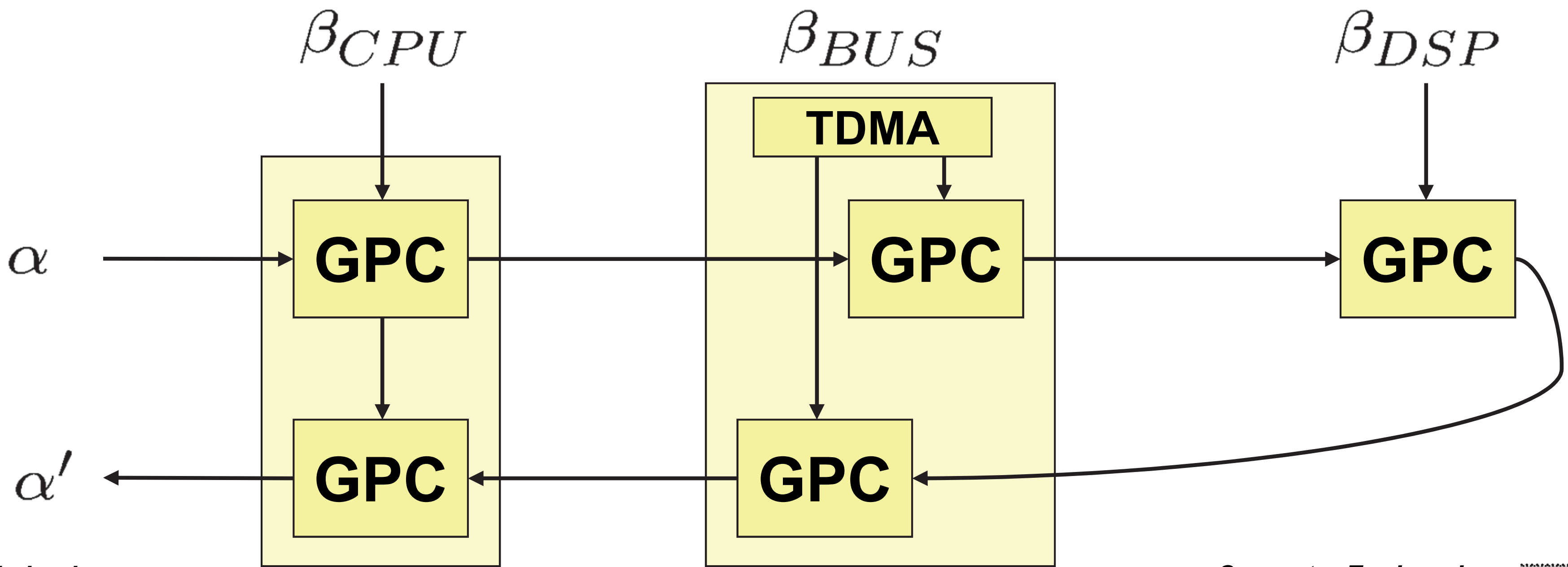


Scheduling and Arbitration



Complete System Composition





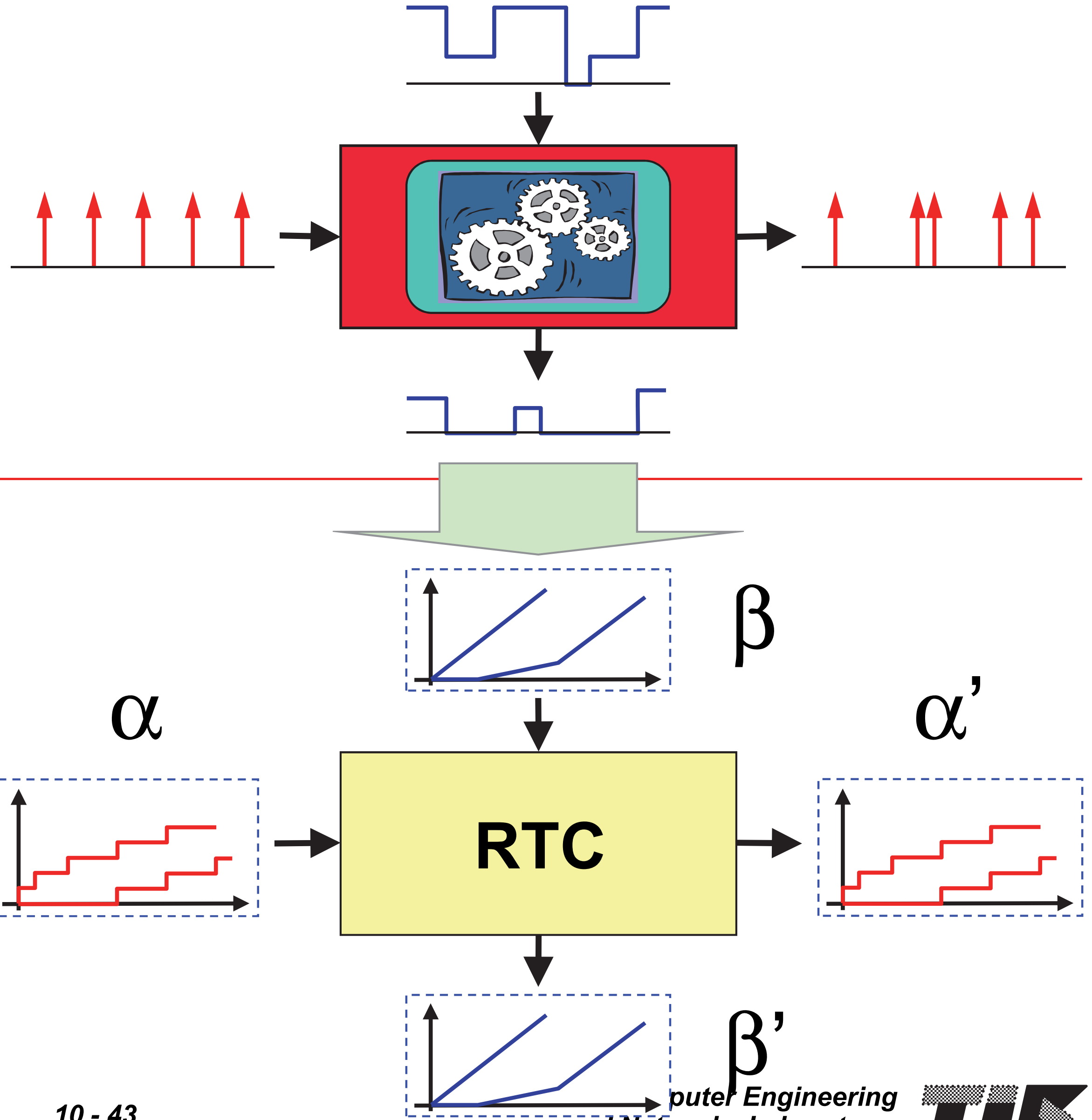
Extending the Framework

- New HW behavior
- New SW behavior
- New scheduling scheme
- ...

- Find new relations:

$$\alpha'(\Delta) = f_\alpha(\alpha, \beta)$$
$$\beta'(\Delta) = f_\beta(\alpha, \beta)$$

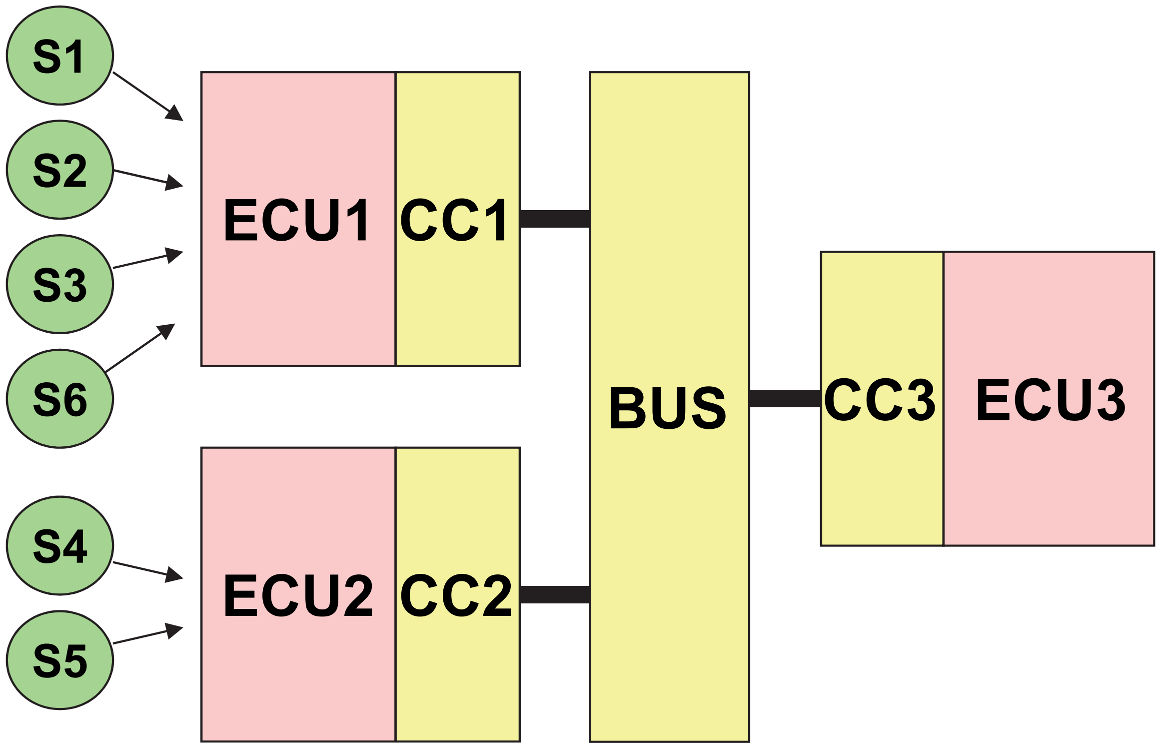
This is the hard part…!



Contents

- Overview
- Real-Time Calculus
- Modular Performance Analysis
- ***Examples***

Case Study



Total Utilization:
- ECU1 59 %
- ECU2 87 %
- ECU3 67 %
- BUS 56 %

6 Real-Time Input Streams
- with jitter
- with bursts
- deadline > period

3 ECU's with own CC's

13 Tasks & 7 Messages
- with different WCET

2 Scheduling Policies
- Earliest Deadline First (ECU's)
- Fixed Priority (ECU's & CC's)

Hierarchical Scheduling
- Static & Dynamic Polling Servers

Bus with TDMA
- 4 time slots with different lengths
  (#1,#3 for CC1, #2 for CC3, #4 for CC3)

Specification Data

Stream	(p,j,d) [ms]	D [s]	Task Chain
S1	(1000, 2000, 25)	8.0	T1.1 → C1.1 → T1.2 → C1.2 → T1.3
S2	(400, 1500, 50)	1.8	T2.1 → C2.1 → T2.2
S3	(600, 0, -)	6.0	T3.1 → C3.1 → T3.2 → C3.2 → T3.3
S4	(20, 5, -)	0.5	T4.1 → C4.1 → T4.2
S5	(30, 0, -)	0.7	T4.1 → C4.1 → T4.2
S6	(1500, 4000, 100)	3.0	T6.1

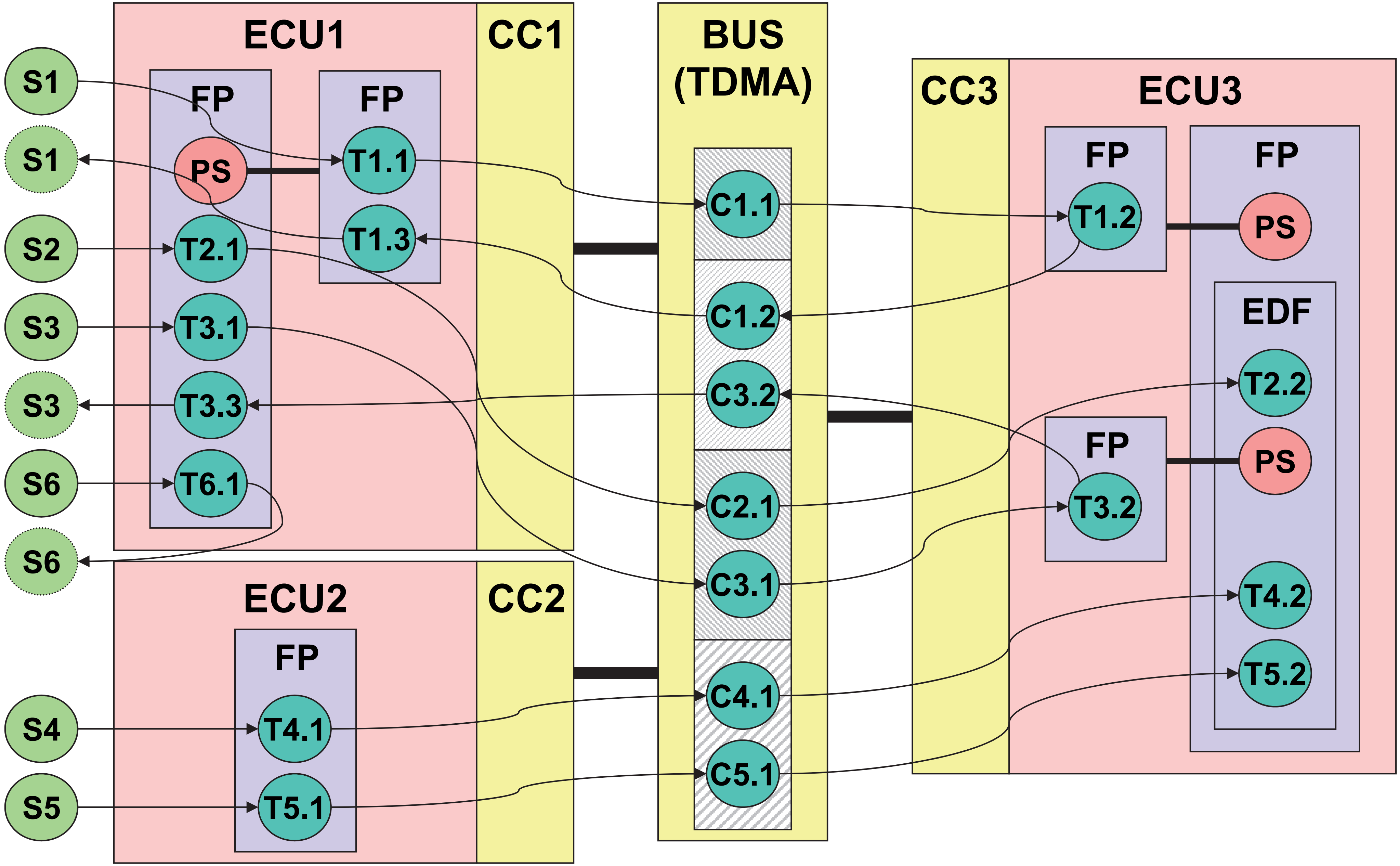
Task	e
T1.1	200
T1.2	300
T1.3	30
T2.1	75
T2.2	25
T3.1	60
T3.2	60
T3.3	40
T4.1	12
T4.2	2
T5.1	8
T5.2	3
T6.1	100

Message	e
C1.1	100
C1.2	80
C2.1	40
C3.1	25
C3.2	10
C4.1	3
C5.1	2

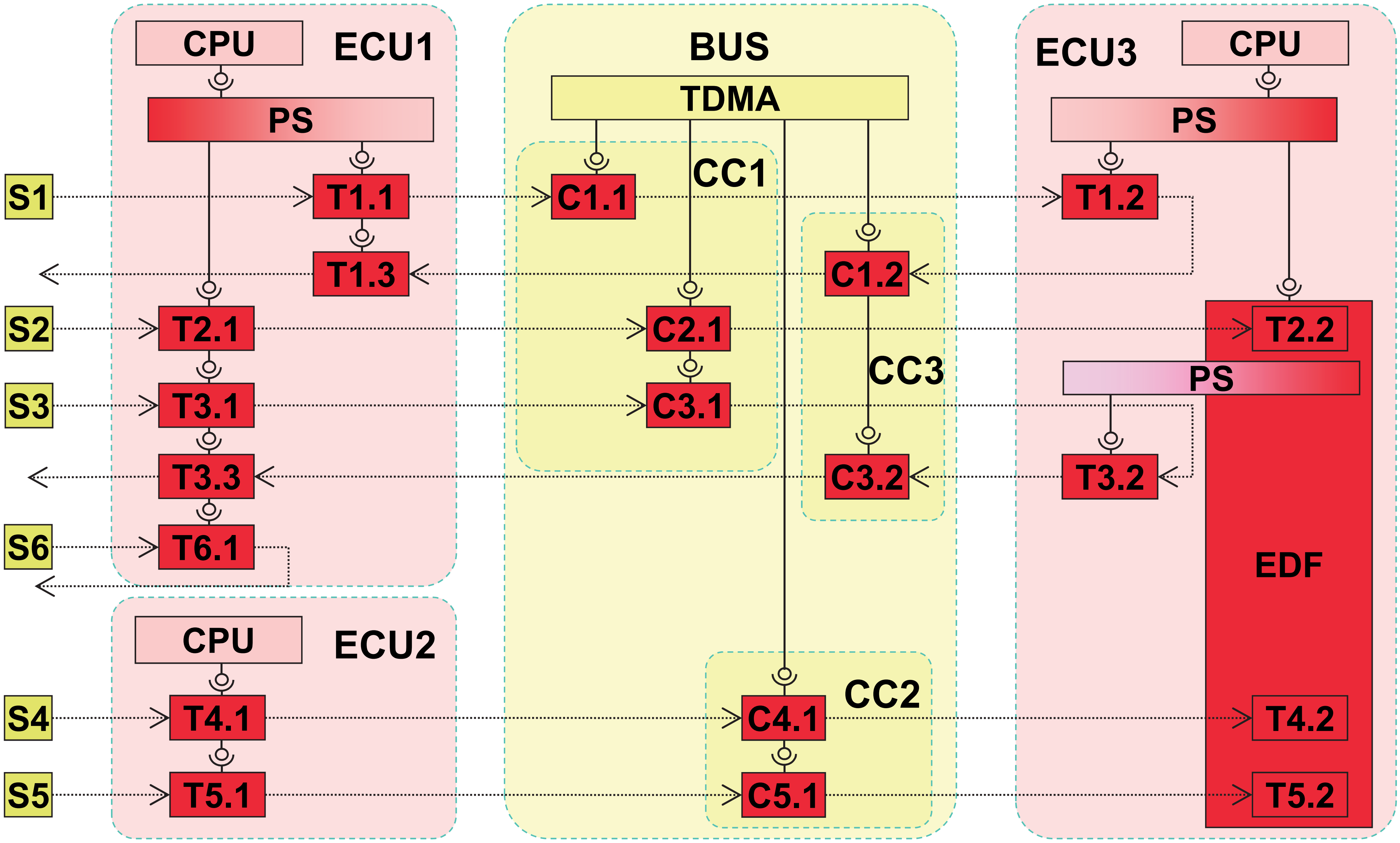
Perdiodic Server	p	e
SPS_{ECU1}	500	200
SPS_{ECU3}	500	250
DPS_{ECU3}	600	120

TDMA	t
Cycle	100
$Slot_{CC1a}$	20
$Slot_{CC1b}$	25
$Slot_{CC2}$	25
$Slot_{CC3}$	30

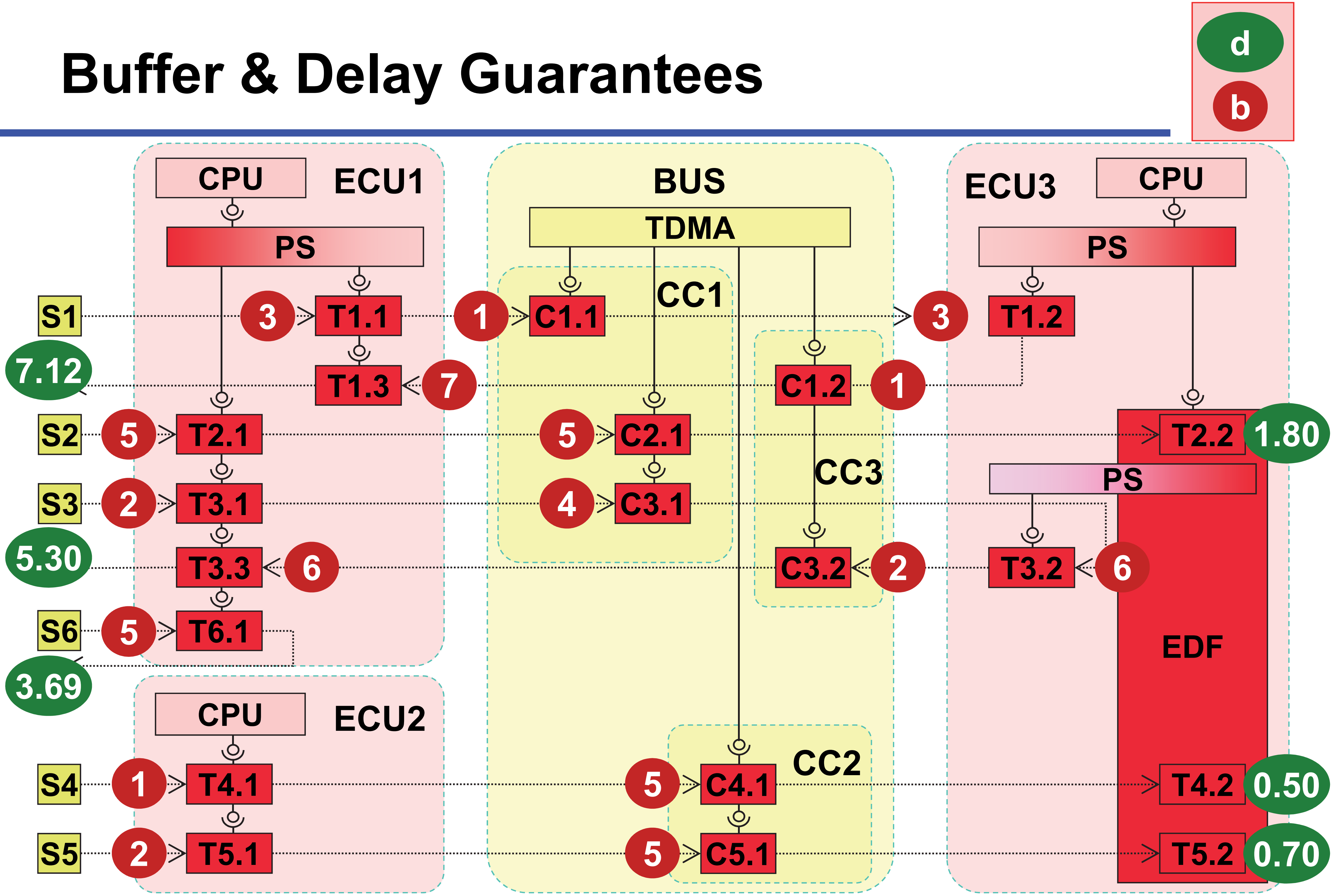
The Distributed Embedded System...



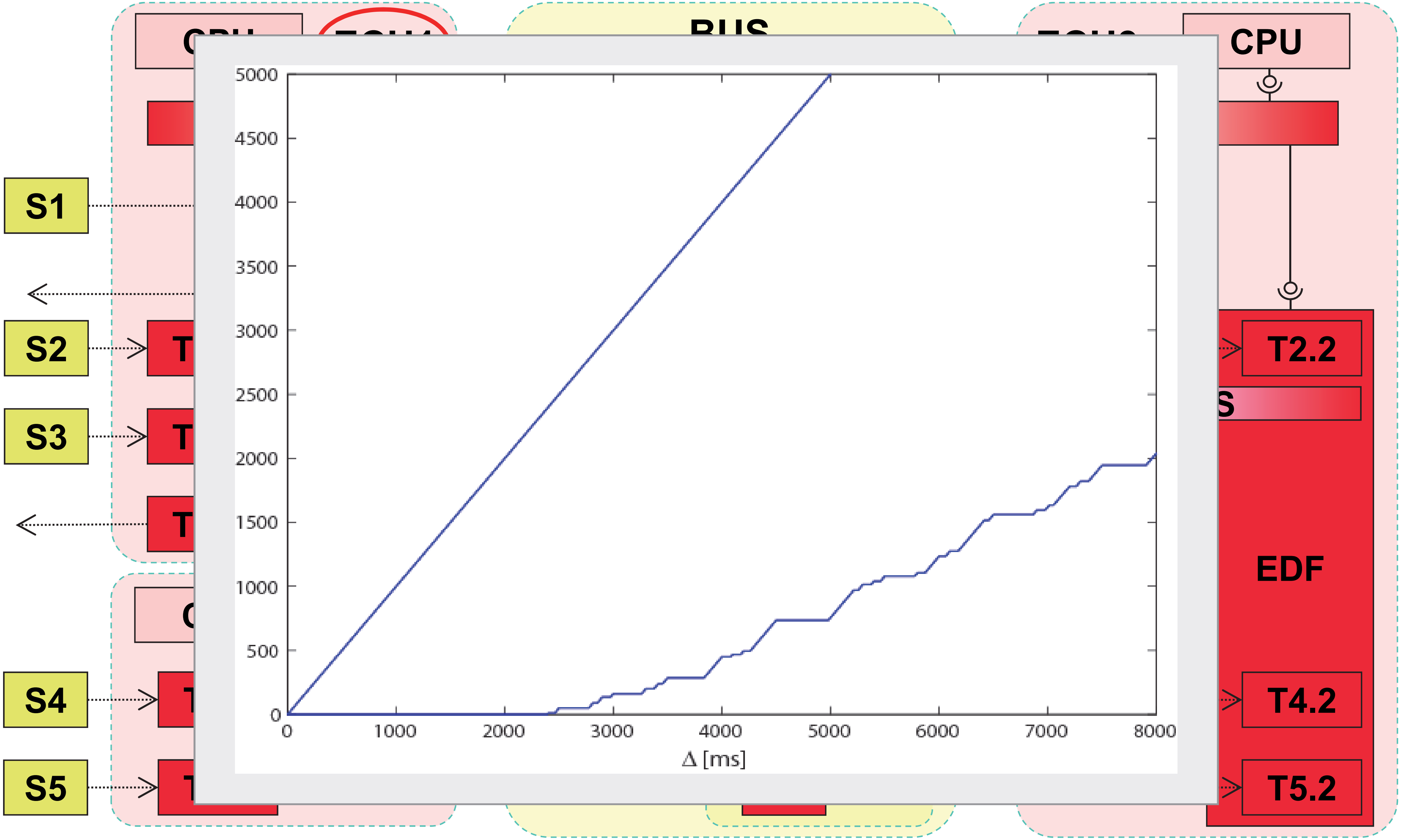
... and its MPA Model



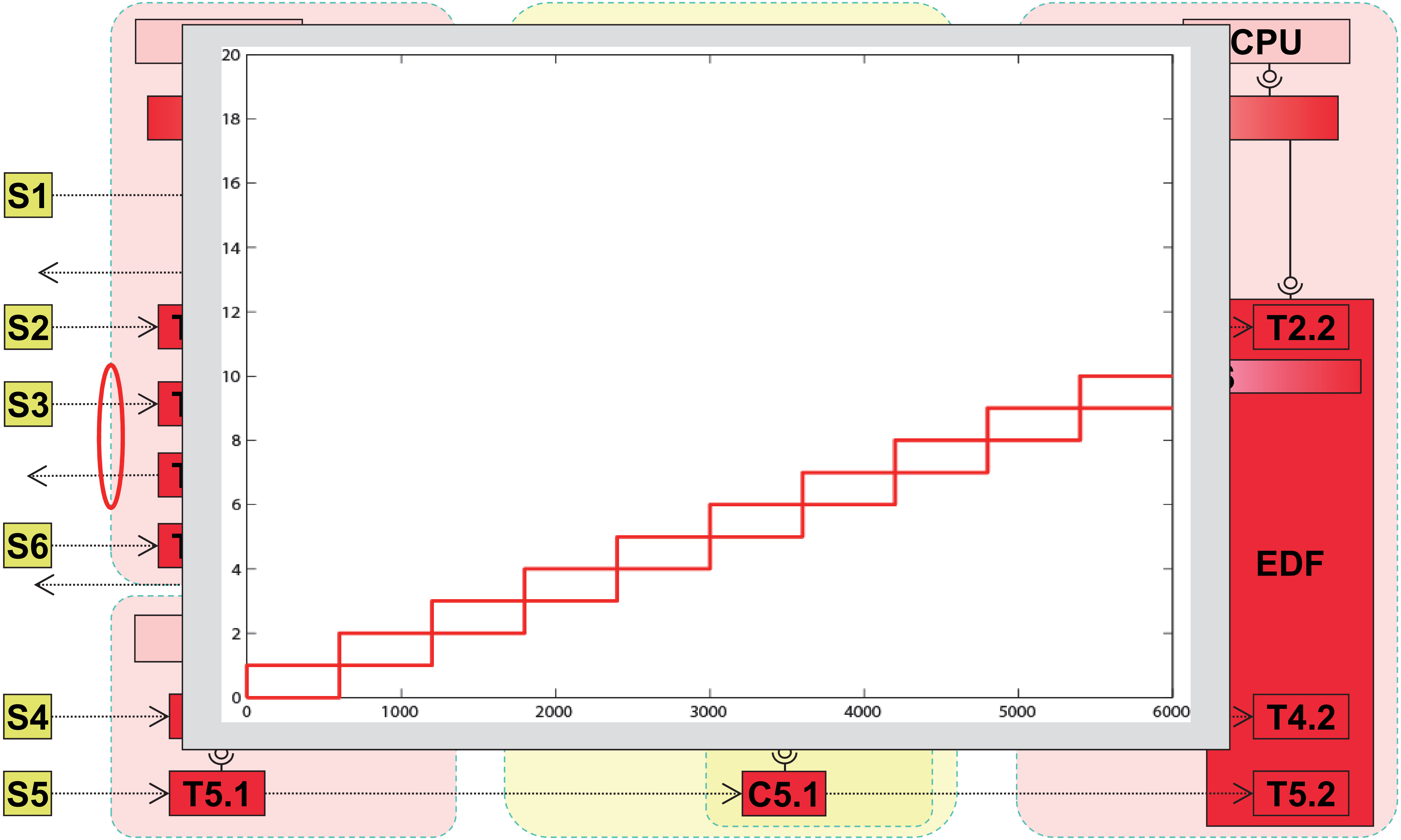
Buffer & Delay Guarantees



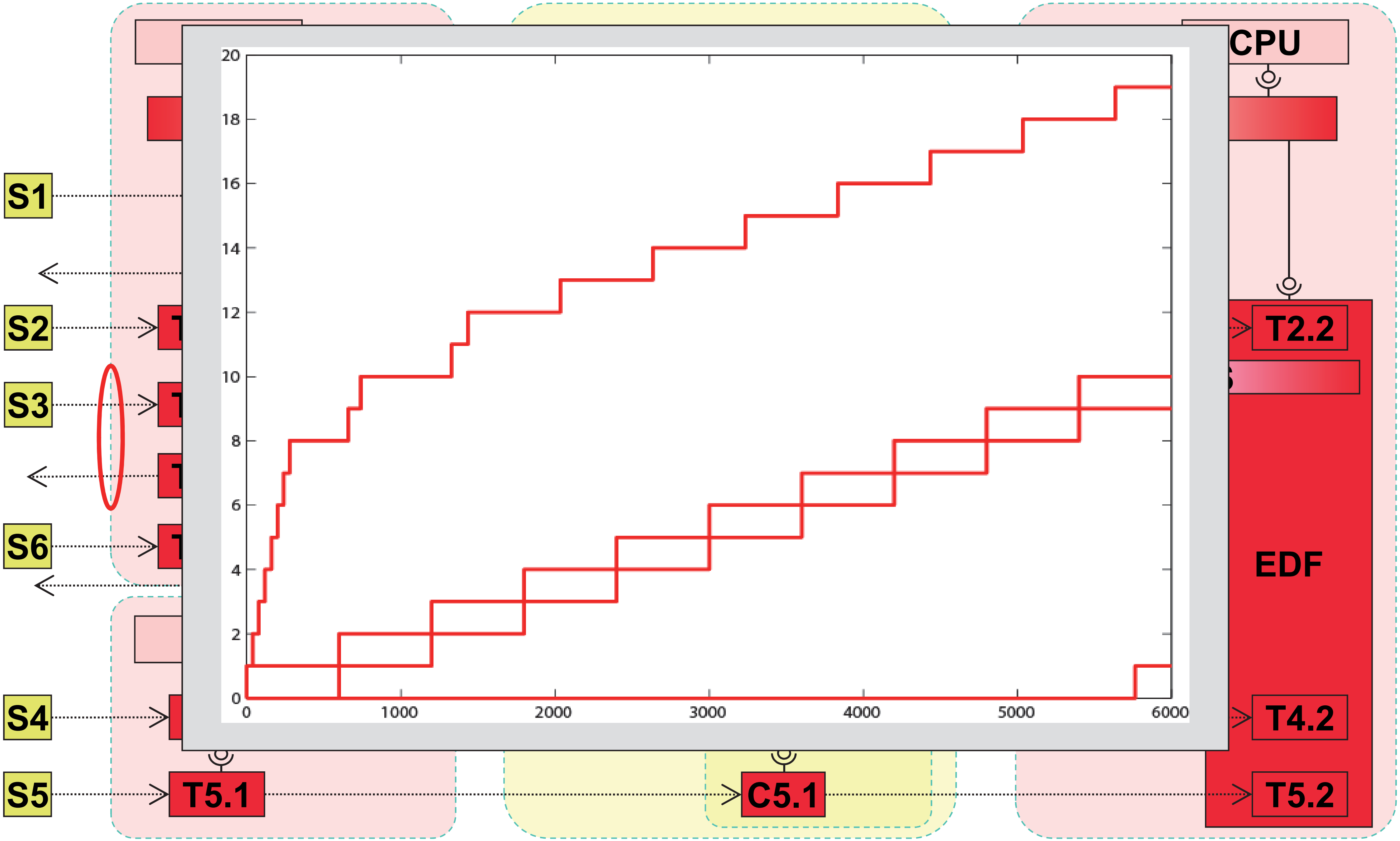
Available & Remaining Service of ECU1



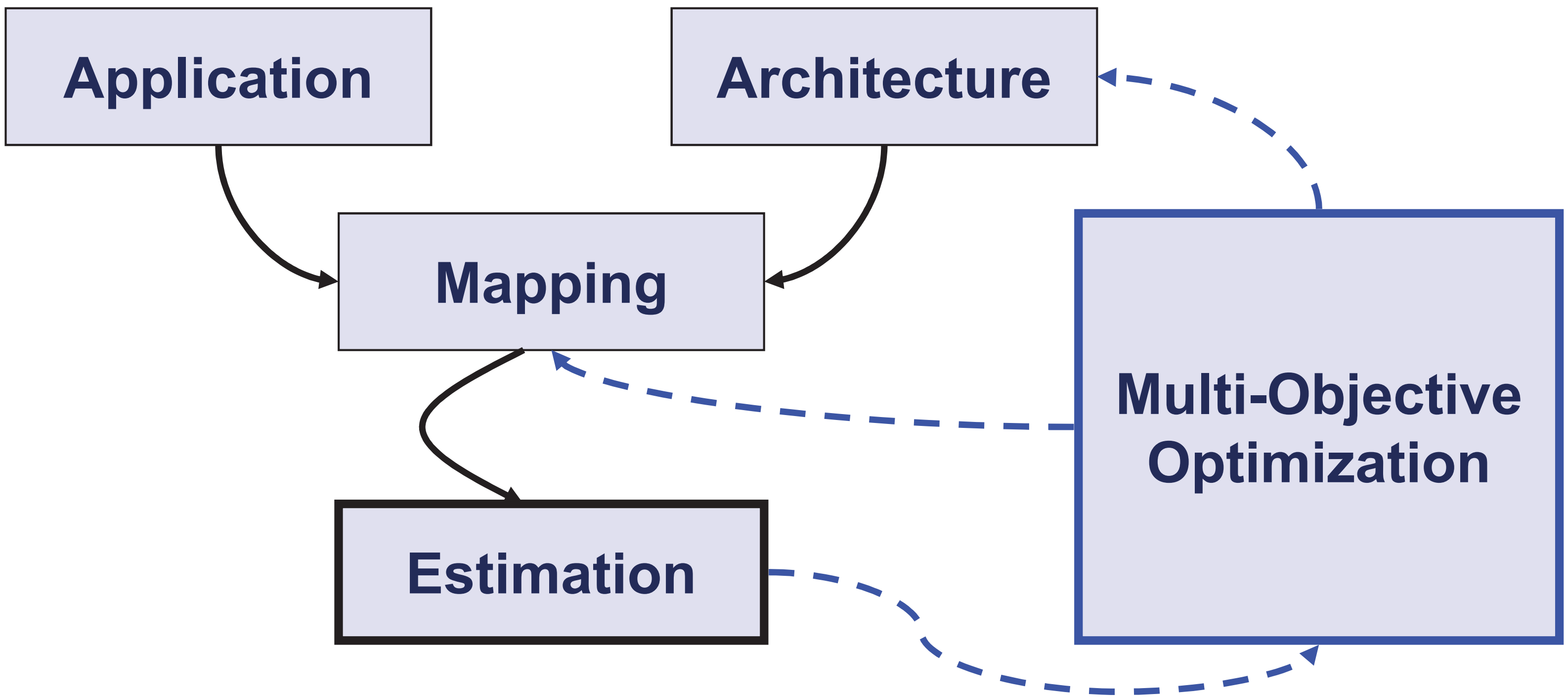
Input of Stream 3



Output of Stream 3

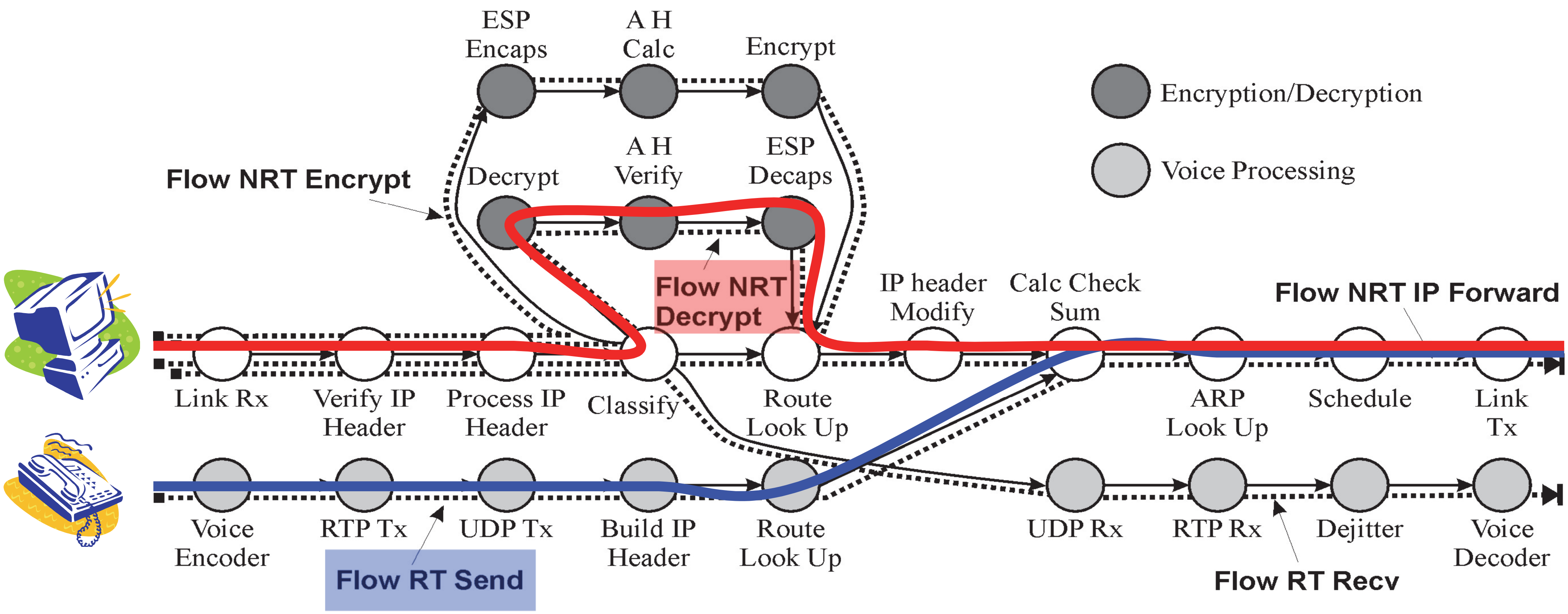


Automated Design Space Exploration

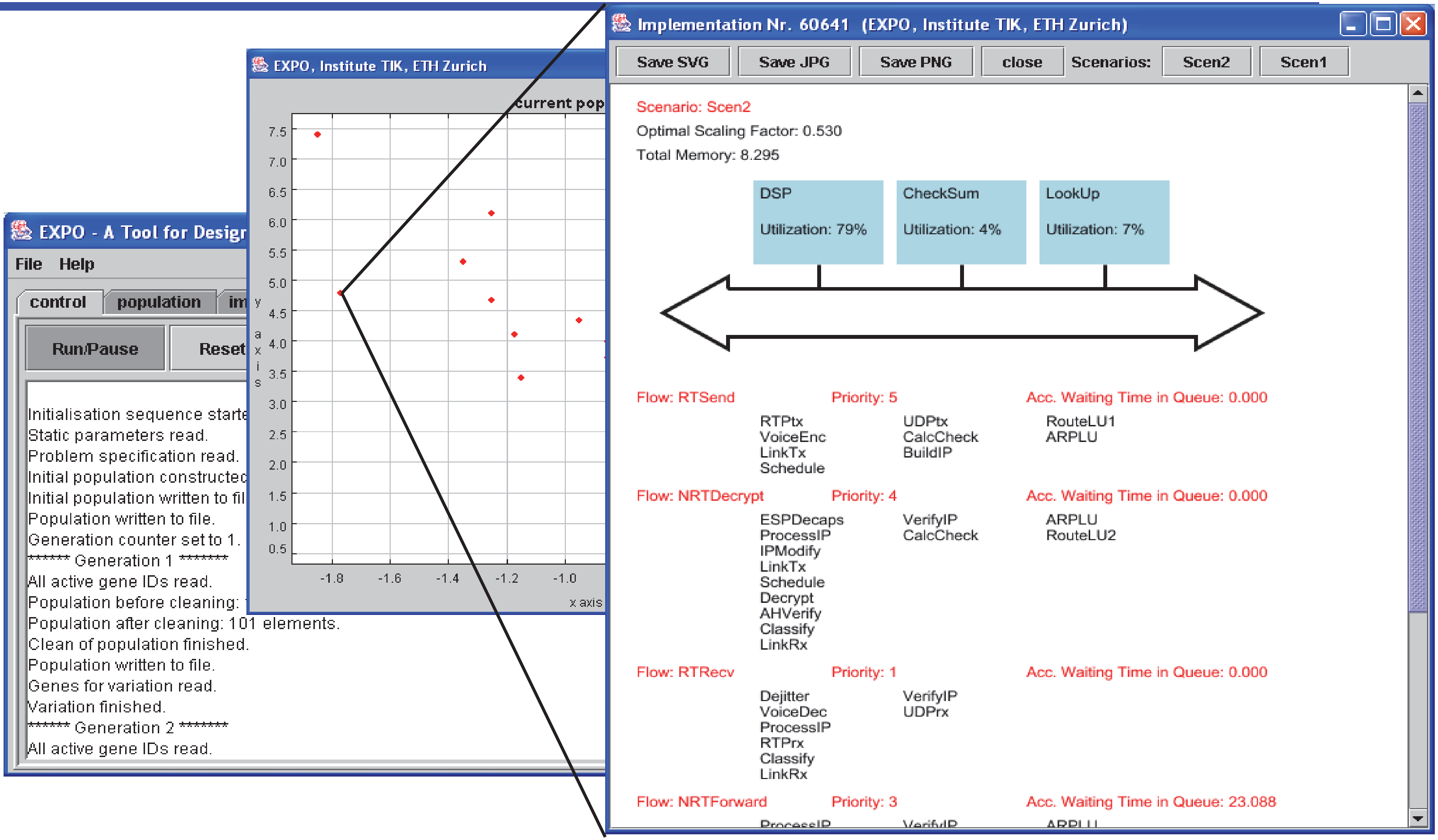


We use evolutionary algorithms for multi-objective optimization!

Network Processor Task Model

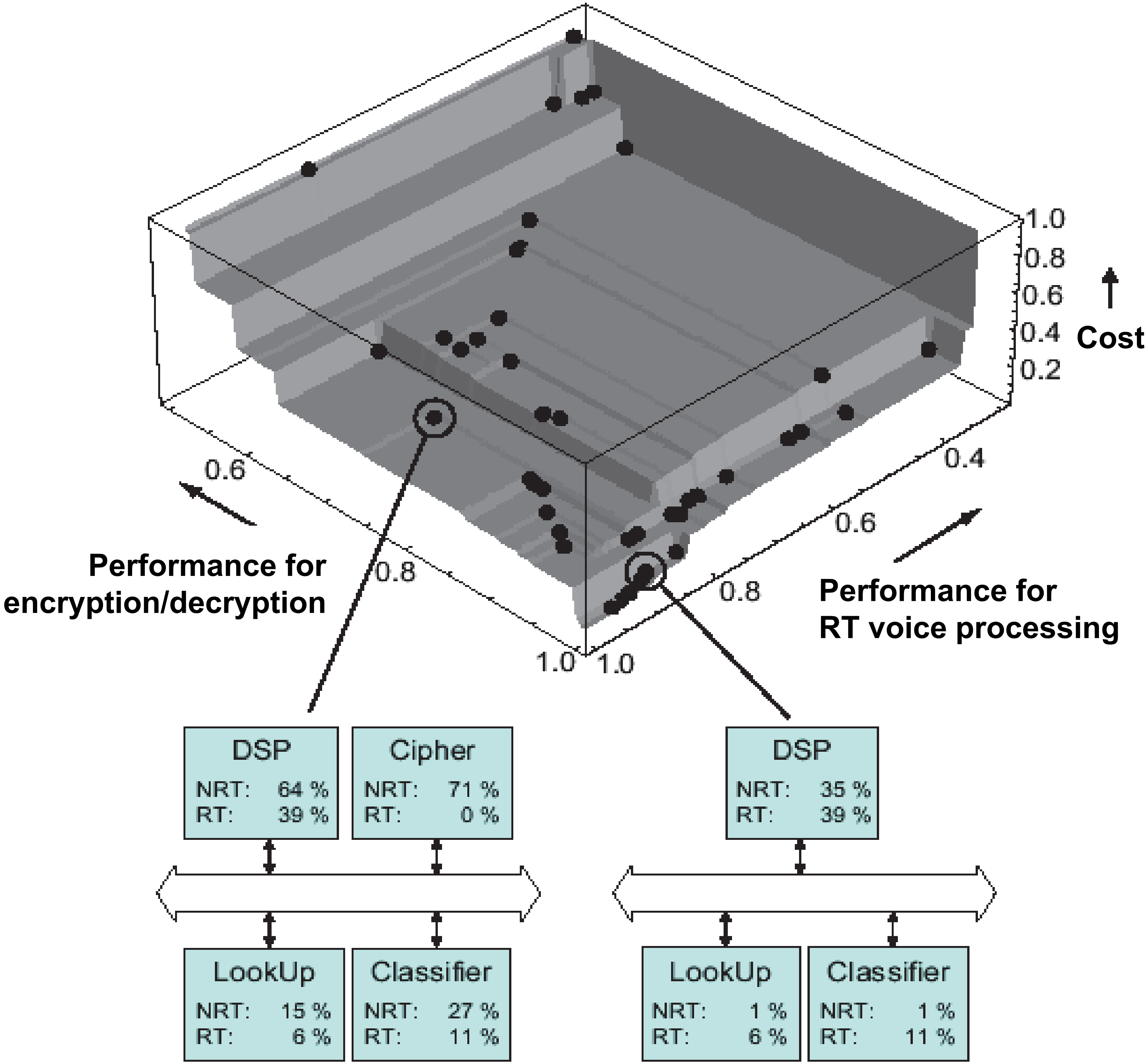


EXPO

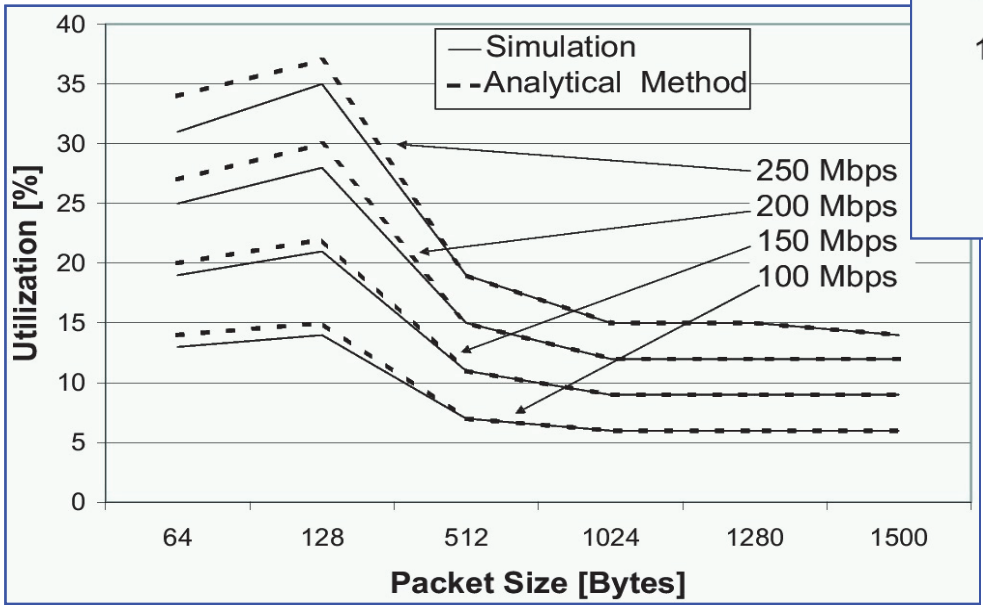


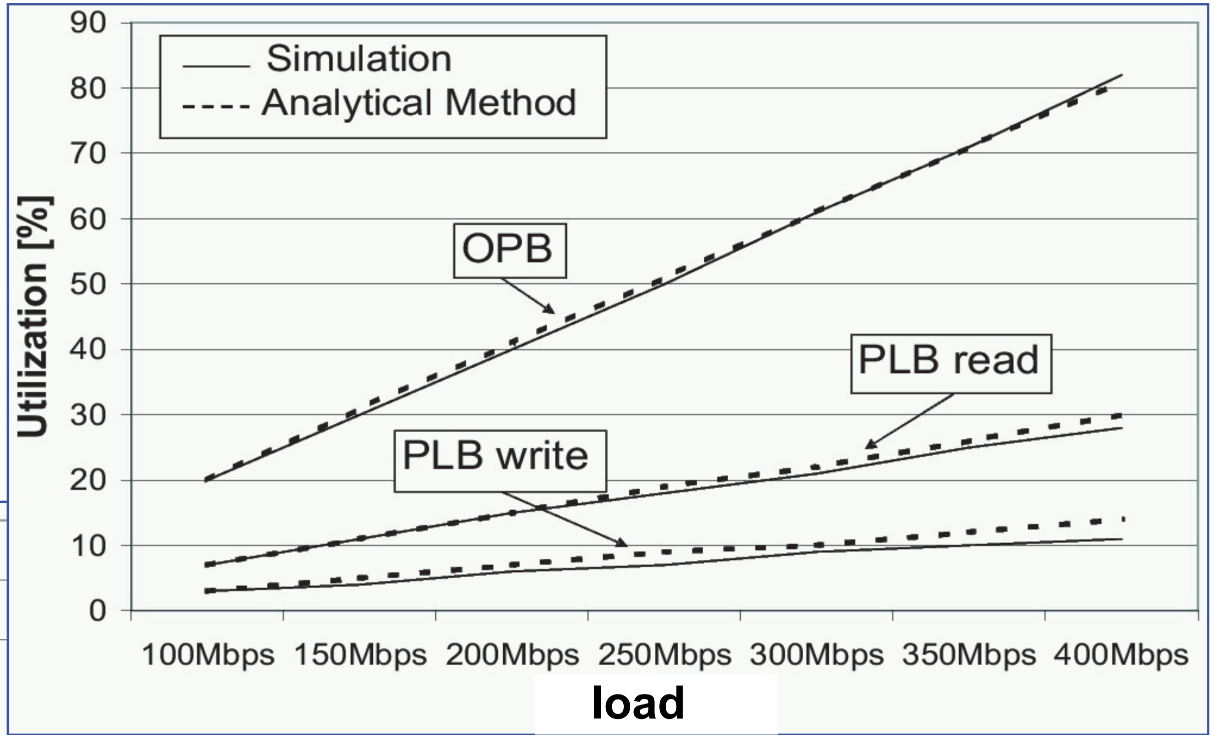
EXPO - A Tool for Design
File Help
control
population
Run/Pause
Reset
Initialisation sequence started
Static parameters read.
Problem specification read.
Initial population constructed
Initial population written to fil
Population written to file.
Generation counter set to 1.
****** Generation 1 *******
All active gene IDs read.
Population before cleaning:
Population after cleaning: 101 elements.
Clean of population finished.
Population written to file.
Genes for variation read.
Variation finished.
****** Generation 2 *******
All active gene IDs read.
EXPO, Institute TIK, ETH Zurich
current pop
y axis
x axis
7.5
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
-1.8
-1.6
-1.4
-1.2
-1.0
Implementation Nr. 60641 (EXPO, Institute TIK, ETH Zurich)
Save SVG
Save JPG
Save PNG
close
Scenarios:
Scen2
Scen1
Scenario: Scen2
Optimal Scaling Factor: 0.530
Total Memory: 8.295
DSP
Utilization: 79%
CheckSum
Utilization: 4%
LookUp
Utilization: 7%
Flow: RTSend
Priority: 5
Acc. Waiting Time in Queue: 0.000
RTPtx
VoiceEnc
LinkTx
Schedule
UDPtx
CalcCheck
BuildIP
RouteLU1
ARPLU
Flow: NRTDecrypt
Priority: 4
Acc. Waiting Time in Queue: 0.000
ESPDecaps
ProcessIP
IPModify
LinkTx
Schedule
Decrypt
AHVerify
Classify
LinkRx
VerifyIP
CalcCheck
ARPLU
RouteLU2
Flow: RTRecv
Priority: 1
Acc. Waiting Time in Queue: 0.000
Dejitter
VoiceDec
ProcessIP
RTPrx
Classify
LinkRx
VerifyIP
UDPrx
Flow: NRTForward
Priority: 3
Acc. Waiting Time in Queue: 23.088

Results


1.0
0.8
0.6
0.4
0.2
Cost
0.6
0.8
1.0
Performance for
encryption/decryption
1.0
0.8
0.6
0.4
Performance for
RT voice processing
DSP
NRT: 64 %
RT: 39 %
Cipher
NRT: 71 %
RT: 0 %
LookUp
NRT: 15 %
RT: 6 %
Classifier
NRT: 27 %
RT: 11 %
DSP
NRT: 35 %
RT: 39 %
LookUp
NRT: 1 %
RT: 6 %
Classifier
NRT: 1 %
RT: 11 %

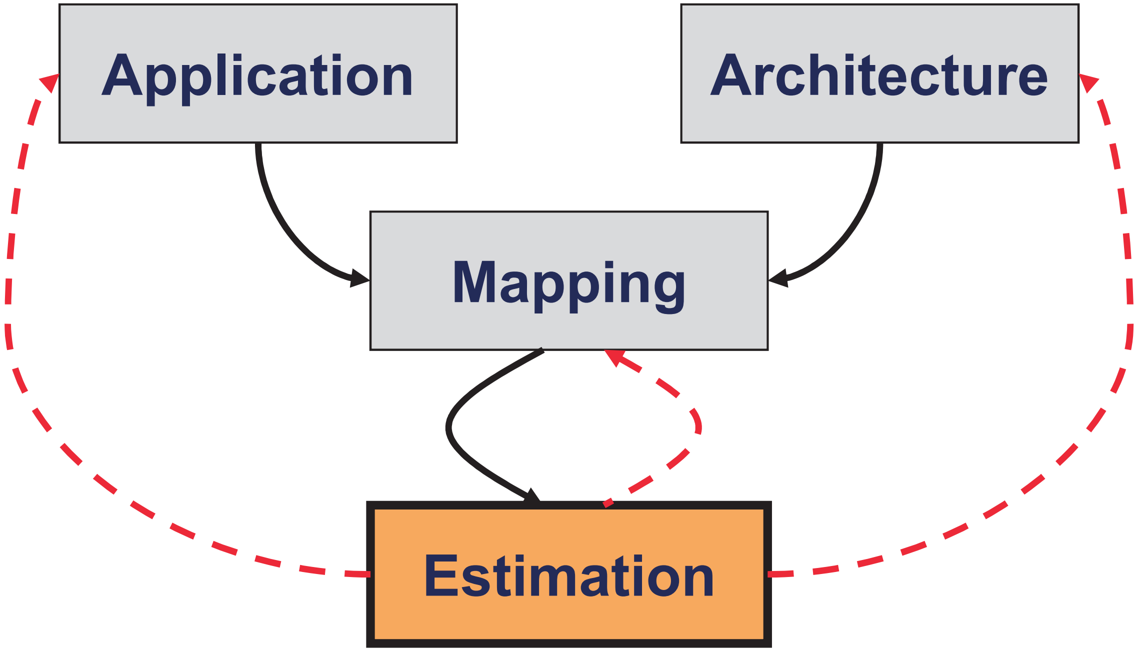
Analysis vs. Simulation



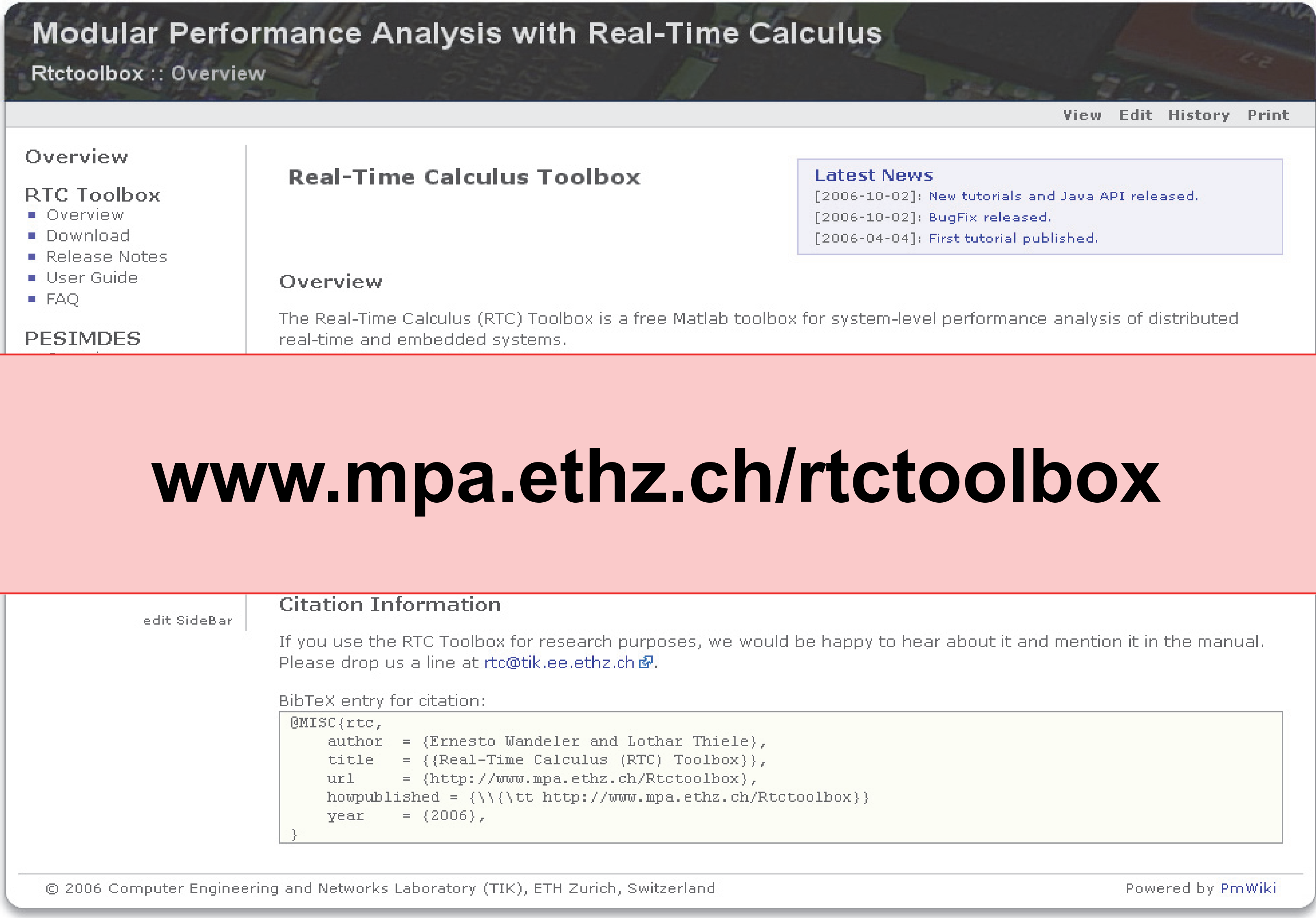


Design Space Exploration

- Determine mapping
- Determine performance network
- Solve system of equations
- Determine important paramerters (end-to-end delay, throughput, buffer space output jitter, ...)
- Give feedback to optimization



RTC Toolbox



Modular Performance Analysis with Real-Time Calculus

Rtctoolbox :: Overview

View Edit History Print

Overview

RTC Toolbox

- Overview
- Download
- Release Notes
- User Guide
- FAQ

PESIMDES

edit SideBar

Real-Time Calculus Toolbox

Latest News

[2006-10-02]: New tutorials and Java API released.

[2006-10-02]: BugFix released.

[2006-04-04]: First tutorial published.

Overview

The Real-Time Calculus (RTC) Toolbox is a free Matlab toolbox for system-level performance analysis of distributed real-time and embedded systems.

www.mpa.ethz.ch/rtctoolbox

Citation Information

If you use the RTC Toolbox for research purposes, we would be happy to hear about it and mention it in the manual. Please drop us a line at rtc@tik.ee.ethz.ch.

BibTeX entry for citation:

```
@MISC{rtc,
    author  = {Ernesto Wandeler and Lothar Thiele},
    title   = {{Real-Time Calculus (RTC) Toolbox}},
    url     = {http://www.mpa.ethz.ch/Rtctoolbox},
    howpublished = {\\{\tt http://www.mpa.ethz.ch/Rtctoolbox}}
    year    = {2006},
}
```


Powered by PmWiki