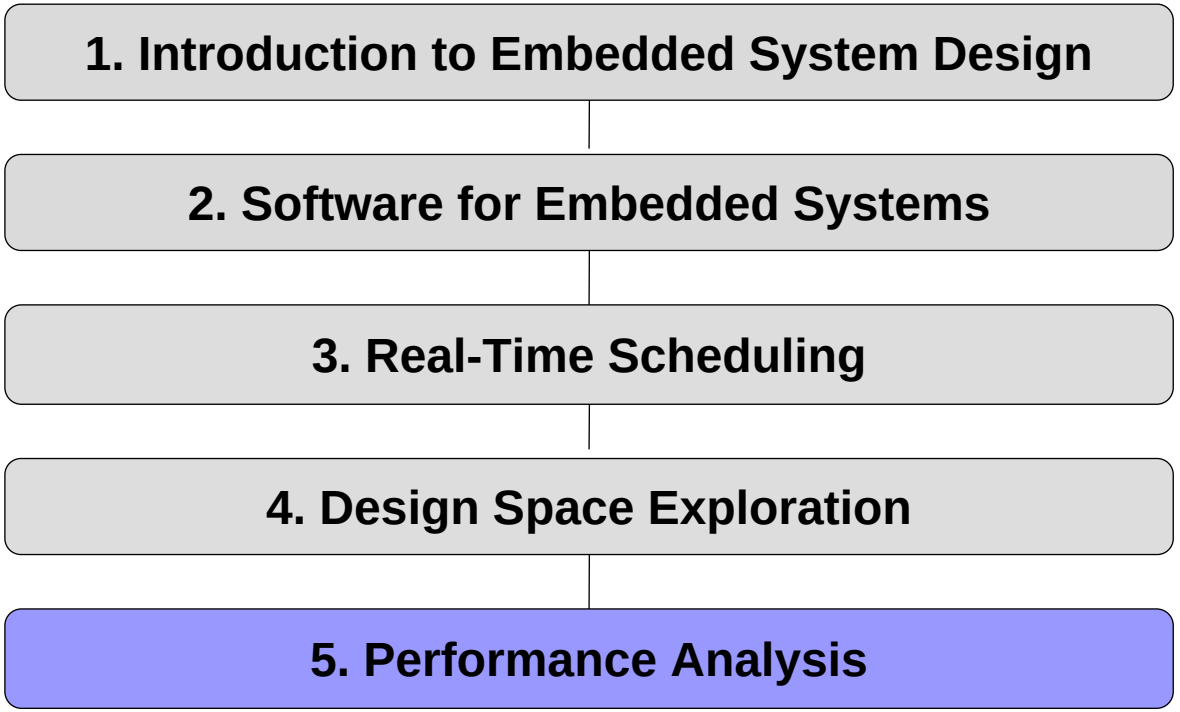
5. Performance Analysis of Distributed Embedded Systems

© Lothar Thiele

Ernesto Wandeler

ETH Zurich, Switzerland

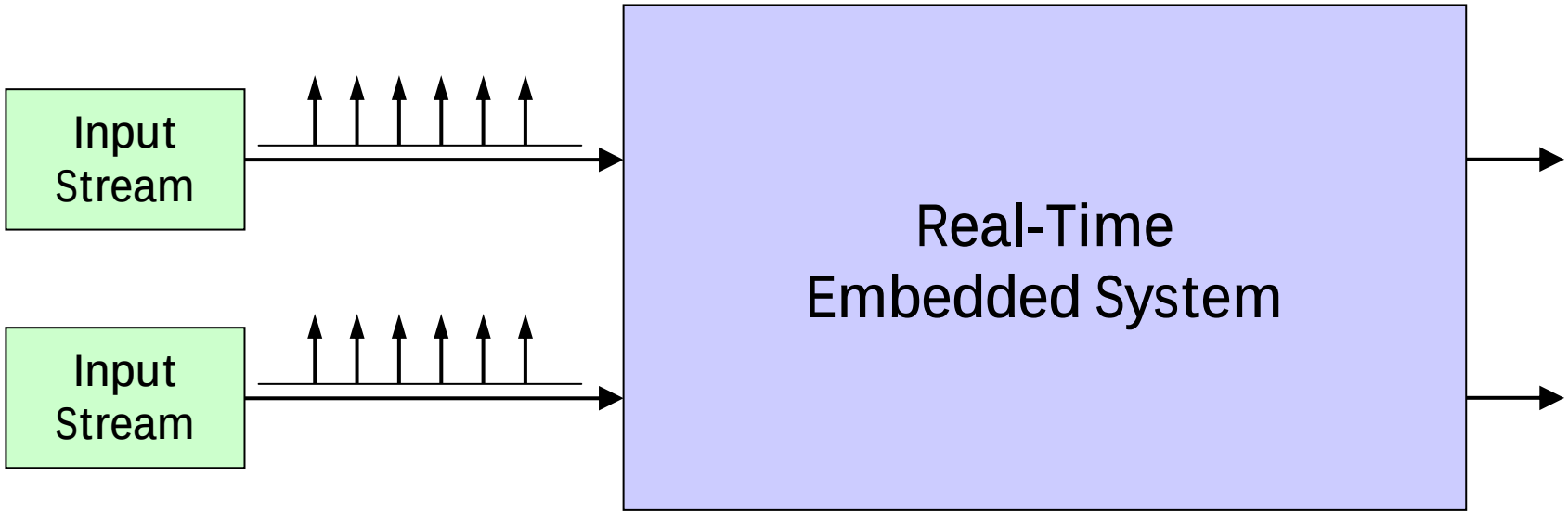
Contents of Lectures (Lothar Thiele)



Outline

- Motivation / Problem Statement
- Modular Performance Analysis
- MPA Case Study
- Real-Time Interfaces / Interface-Based Design
- IBD Case Study
- Efficient Computation and Tool Support

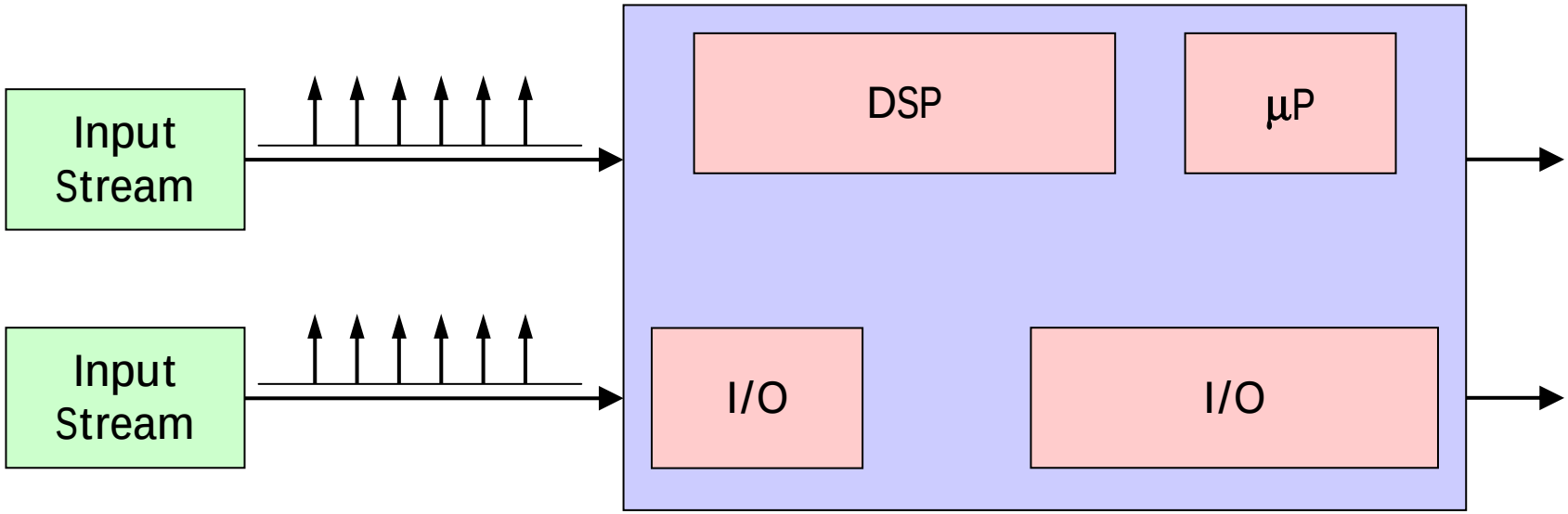
Real-Time Embedded System



Real-Time Embedded System…

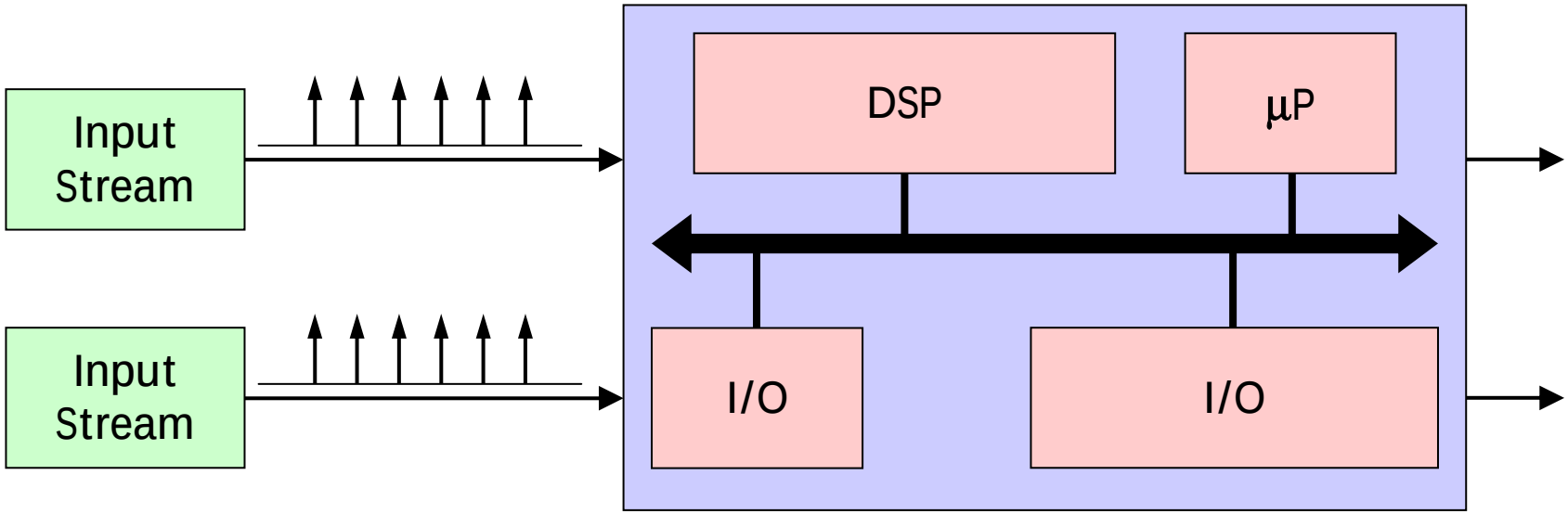
…System Level Performance Analysis

Real-Time Embedded System



Computational Resources ...

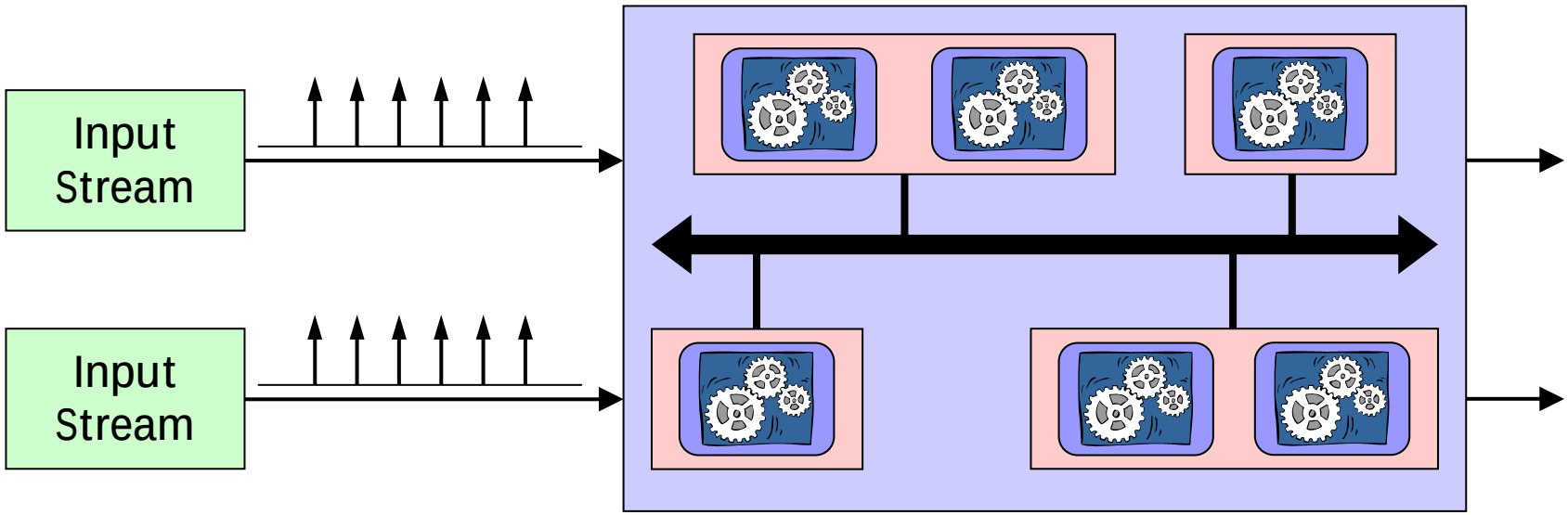
Real-Time Embedded System



Computational Resources ...

... Communication Resources ...

Real-Time Embedded System

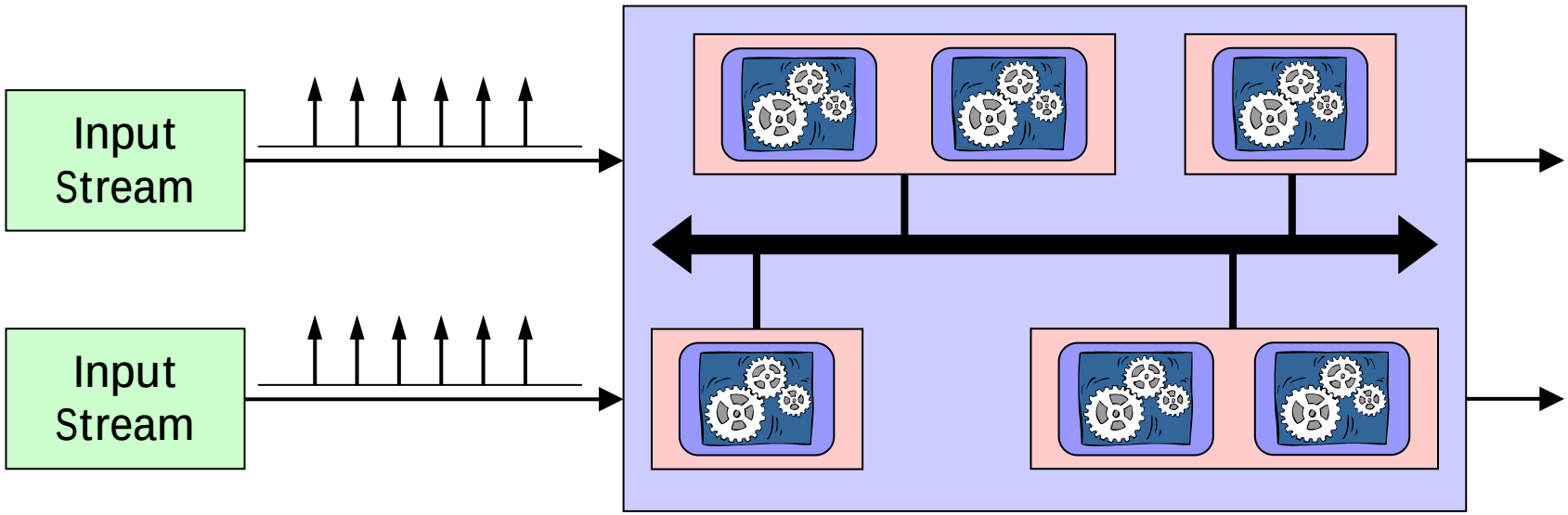


Computational Resources ...

... Communication Resources ...

... Tasks (HW/SW Components)

System-Level Performance Analysis



Memory Requirements?

Timing Properties?

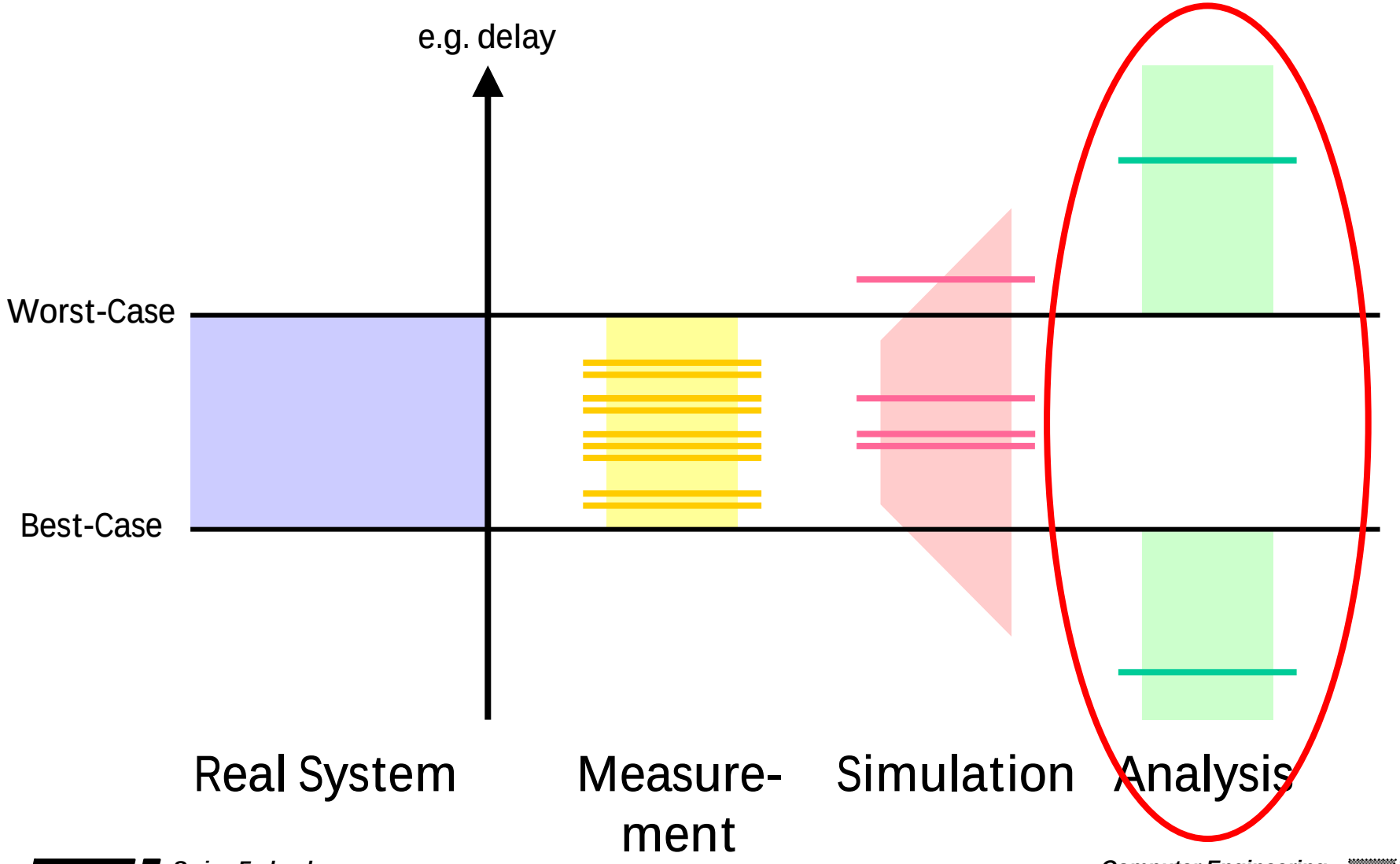
Analytical

Bottleneck?

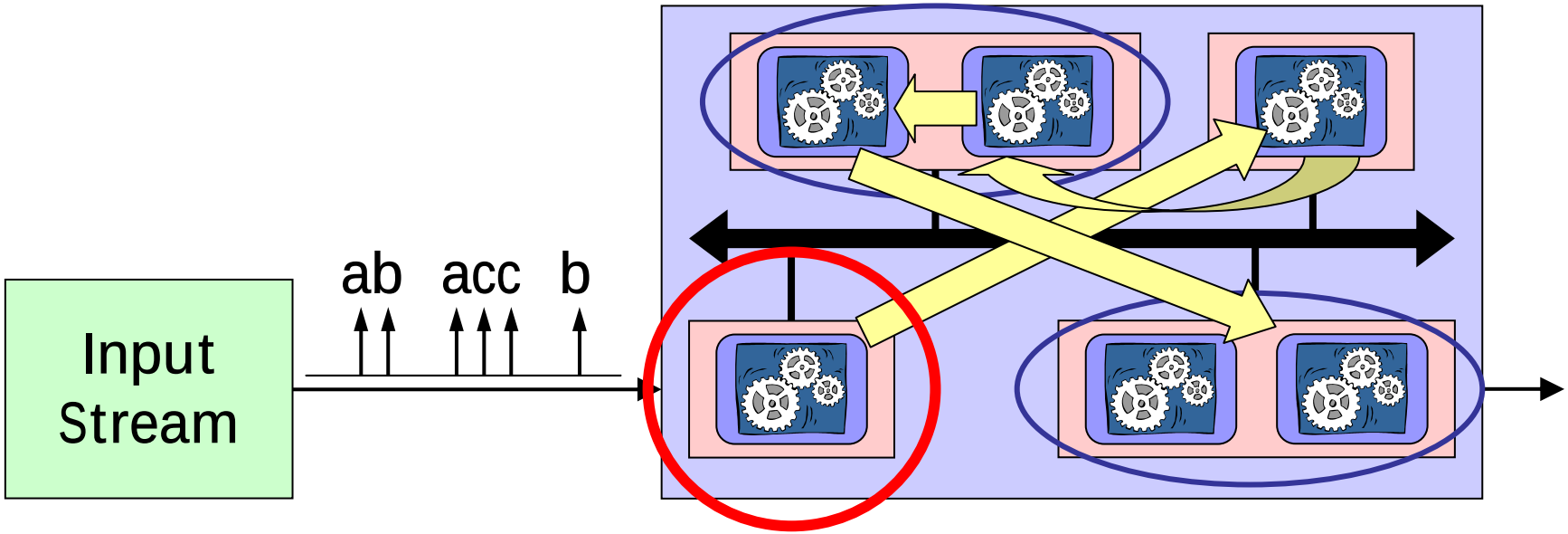
Processor Speeds?

Bus Utilization?

System-Level Performance Methods



Difficulties



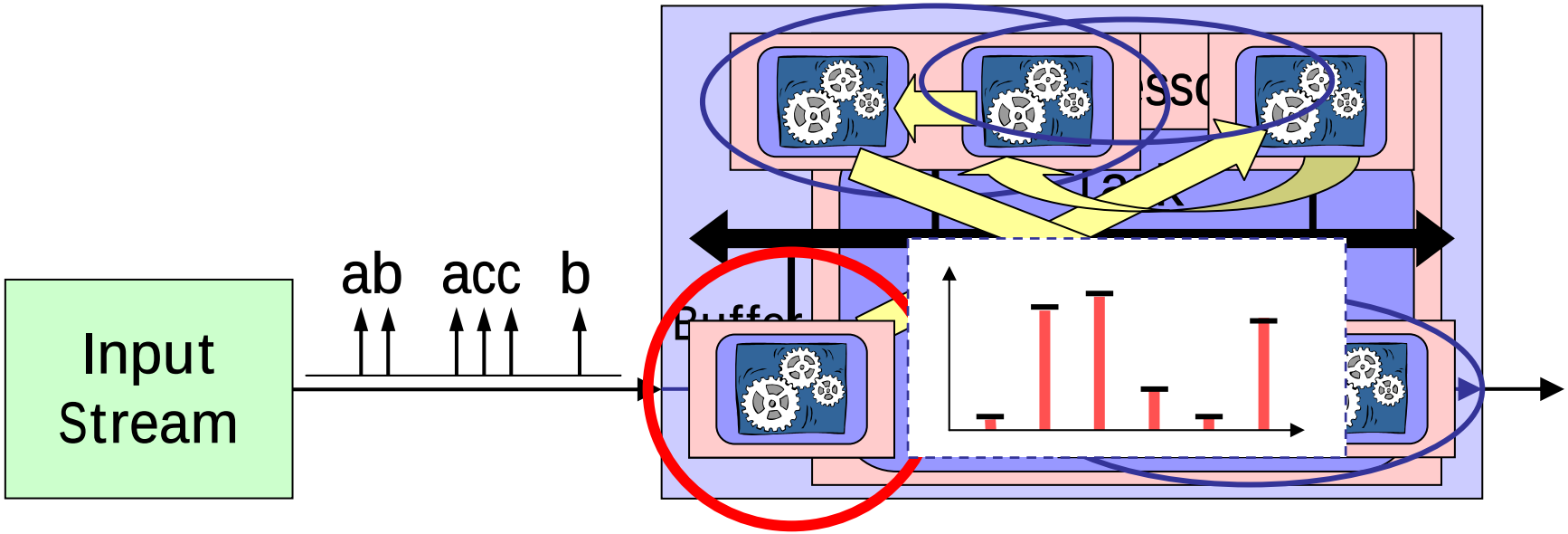
Task Communication

Task Scheduling

Complex Input:

- Timing (jitter, bursts, ...)
- Different Event Types

Difficulties



Task Communication

Task Scheduling

Complex Input:

- Timing (jitter, bursts, ...)
- Different Event Types

Variable Resource Availability

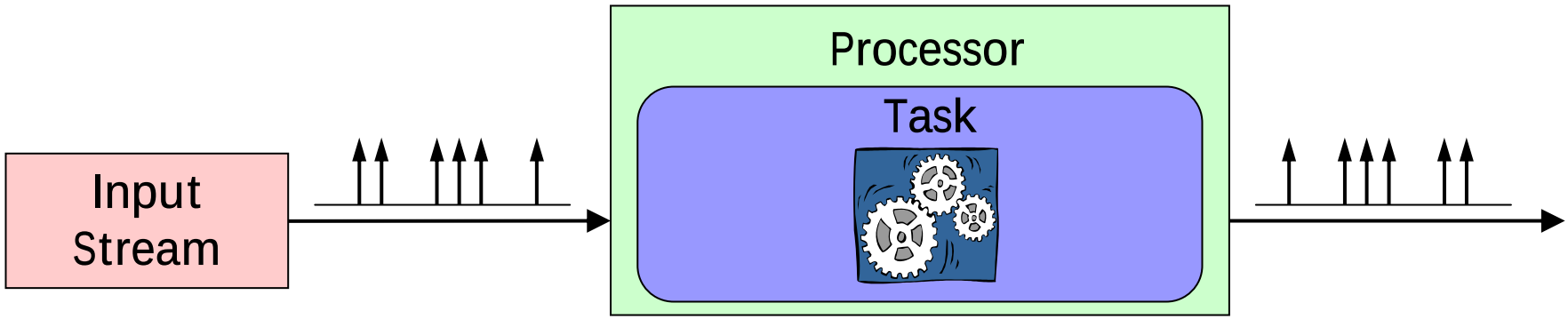
Variable Execution Demand

- Input (different event types)
- Internal State (Program, Cache, ...)

Outline

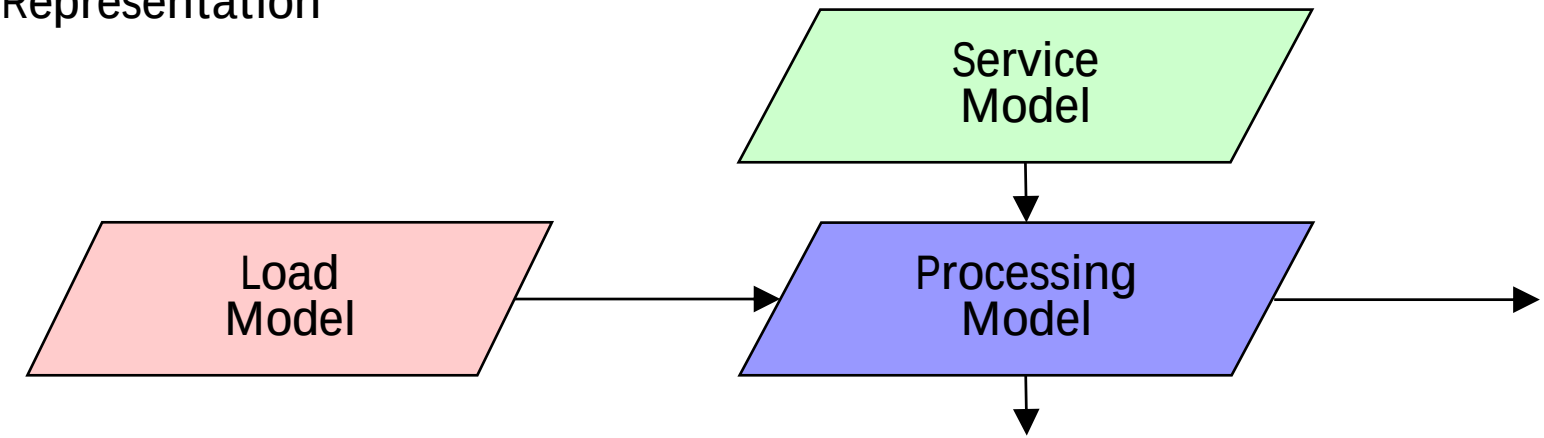
- Motivation / Problem Statement
- Modular Performance Analysis
- MPA Case Study
- Real-Time Interfaces / Interface-Based Design
- IBD Case Study
- Efficient Computation and Tool Support

Abstract Models for Performance Analysis

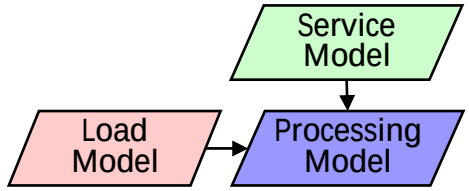


Concrete
Instance

Abstract
Representation

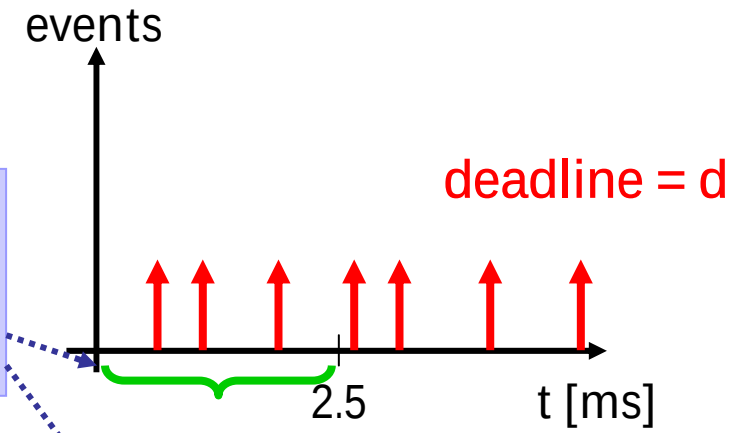


Load Model (Environment)



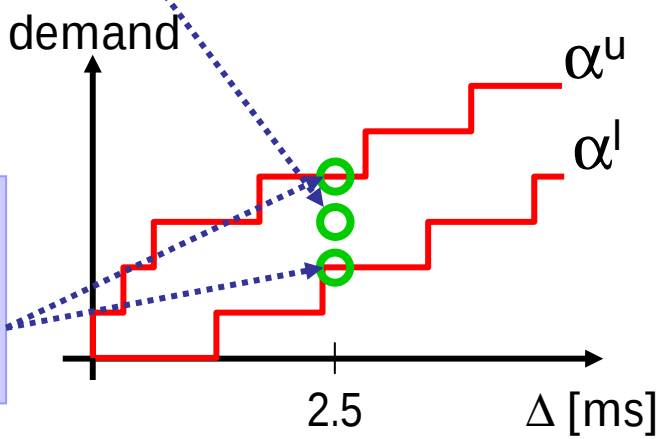
Event Stream

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

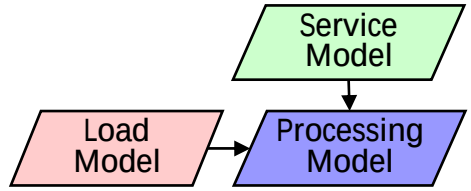


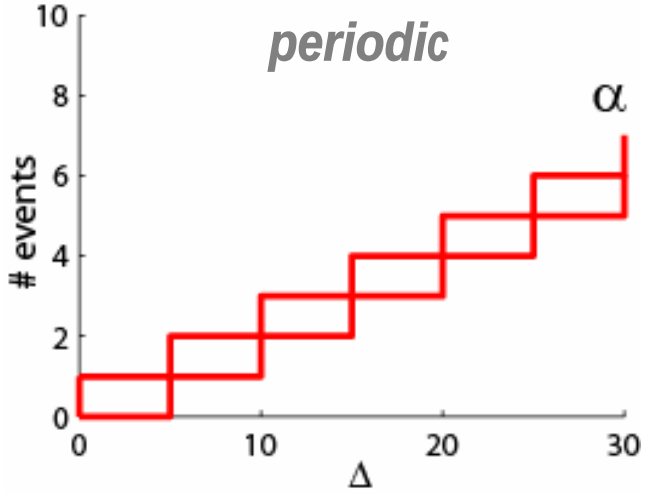
Arrival Curve α & Delay d

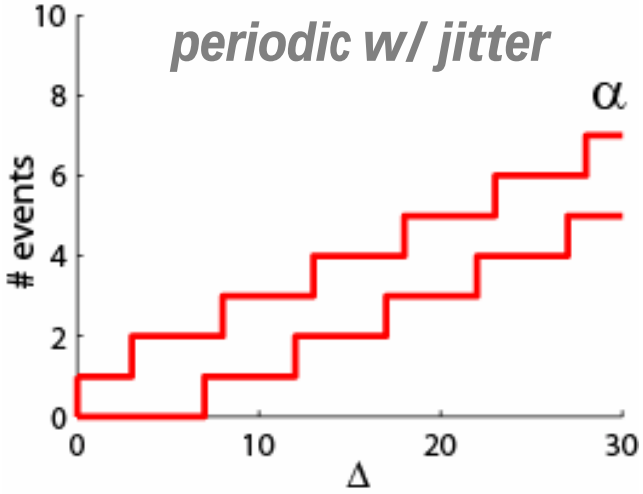
maximum / minimum arriving demand in *any interval* of length 2.5 ms

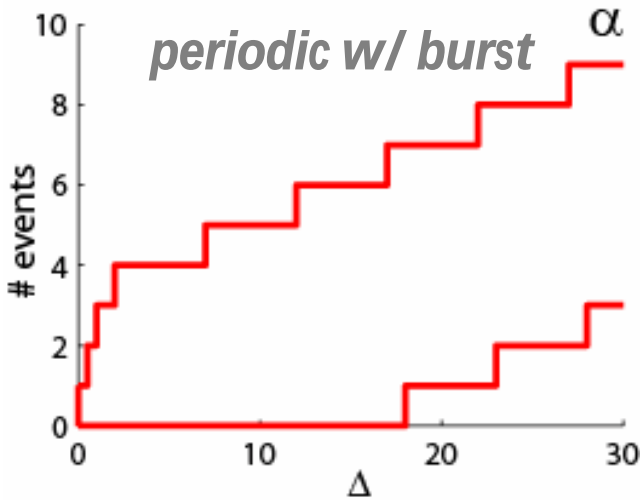


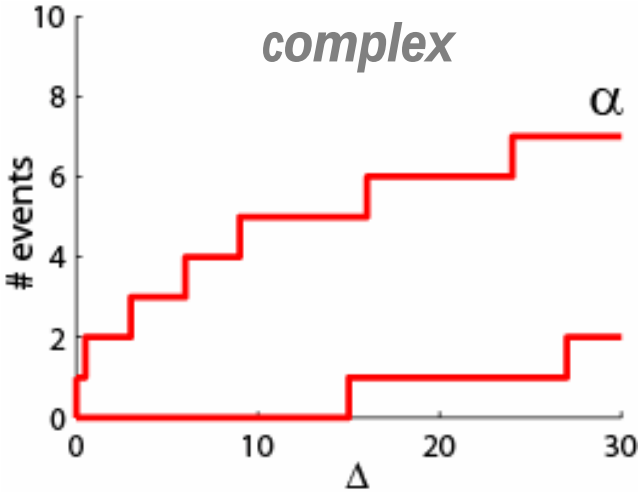
Load Model - Examples



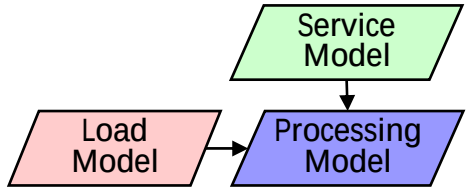






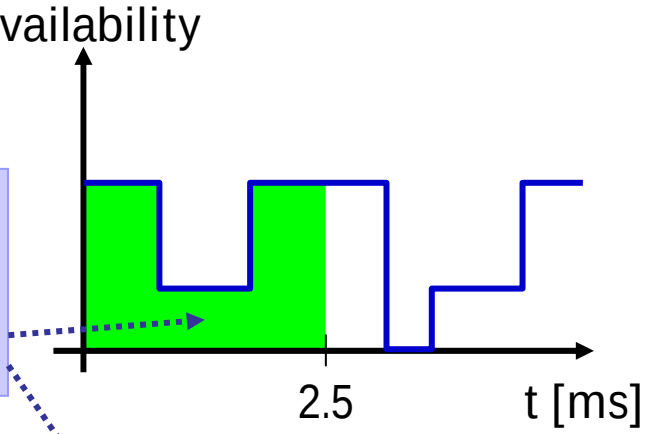


Service Model (Resources)



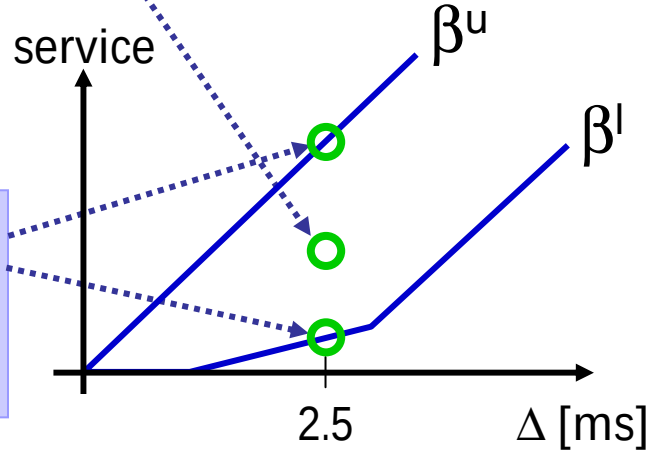
Resource Availability

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

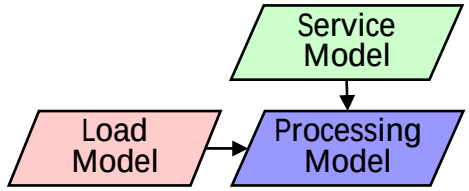


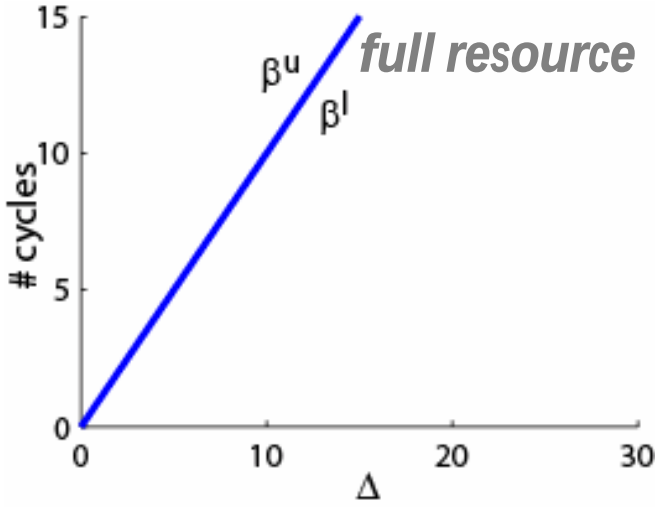
Service Curves [β^l, β^u]

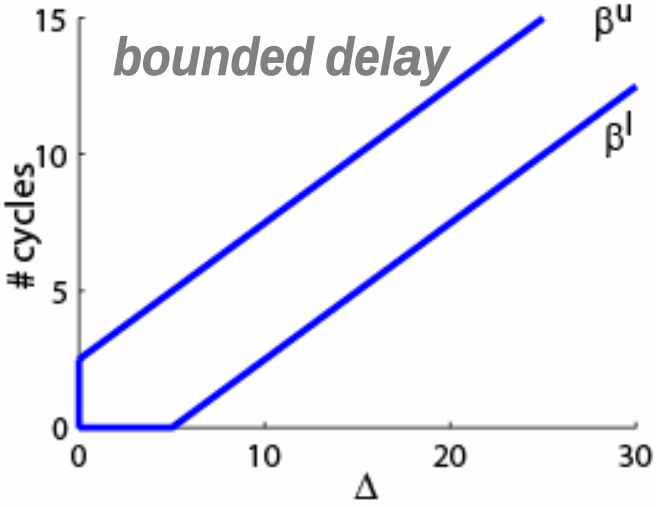
maximum/minimum available service in *any interval* of length 2.5 ms

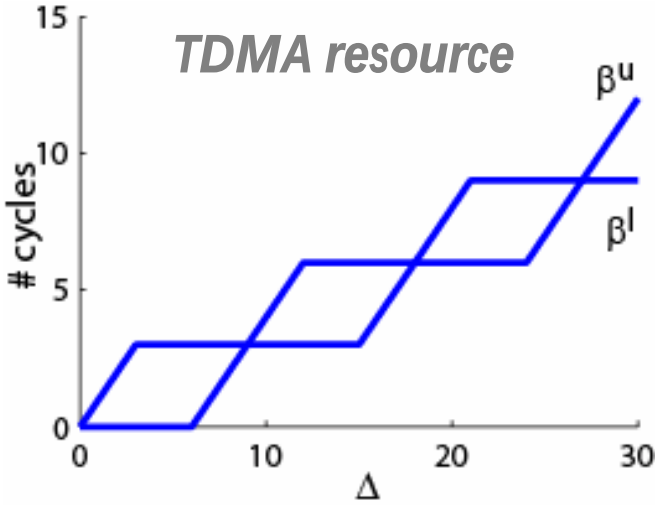


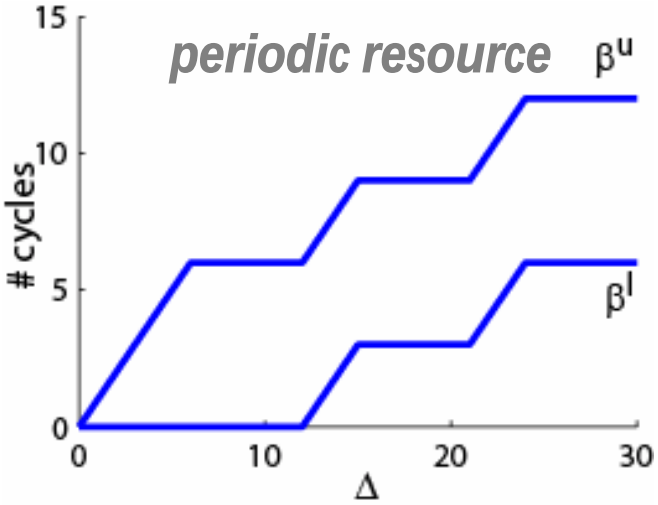
Service Model - Examples



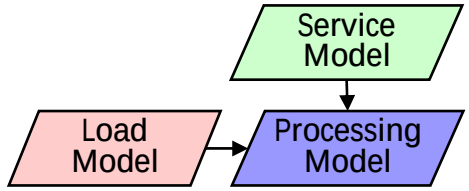






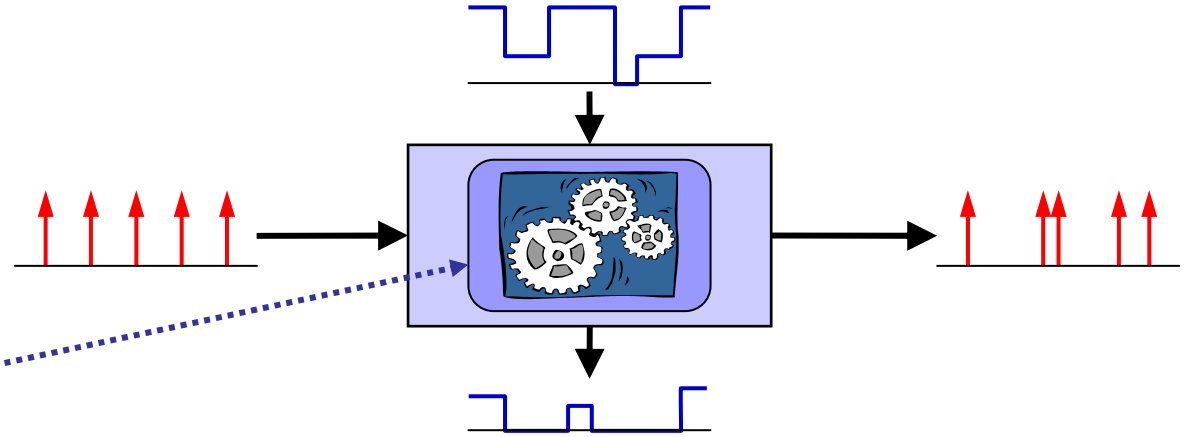


Processing Model (HW/SW)



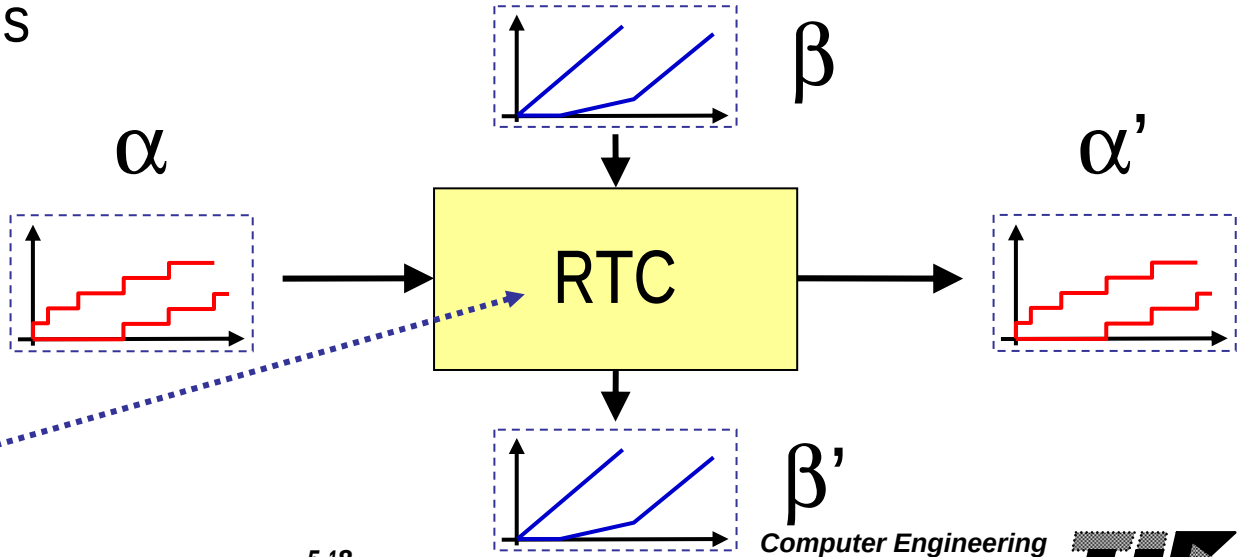
HW/SW Components

Processing semantics and functionality of hardware or software tasks

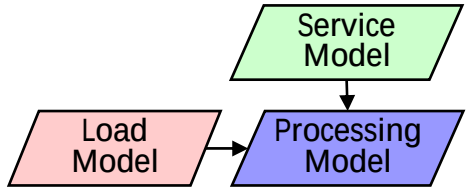


Abstract Components

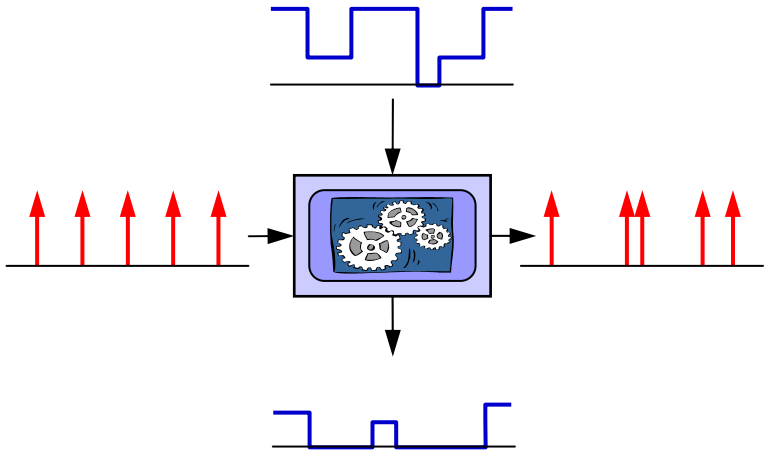
$$\alpha'(\Delta) = f_\alpha(\alpha, \beta)$$
$$\beta'(\Delta) = f_\beta(\alpha, \beta)$$



Processing Model – Examples



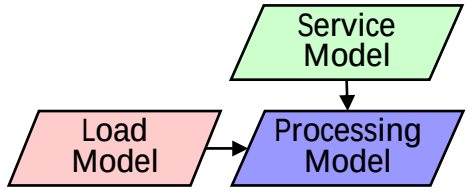
Greedy Processing Component



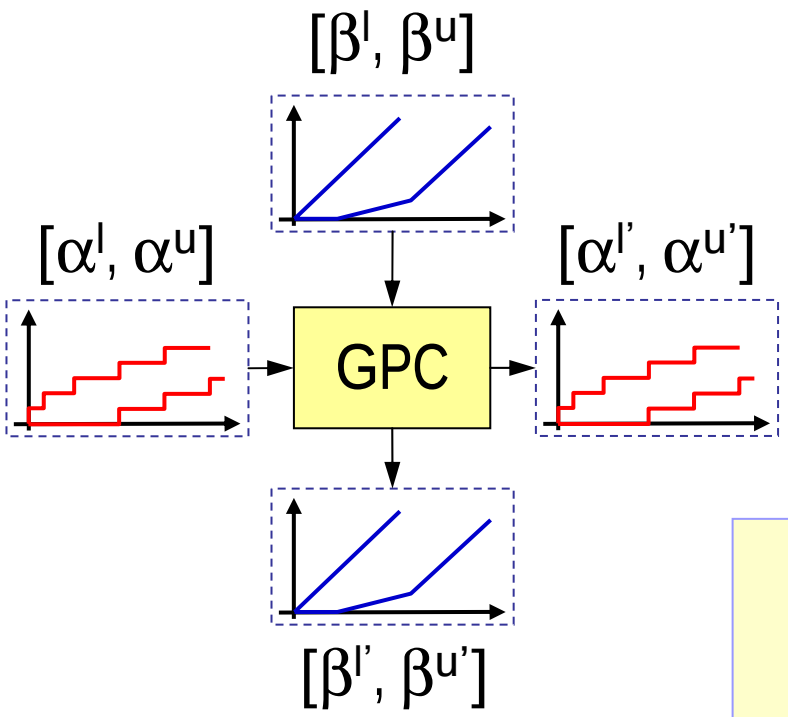
Behavioral Description

- 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.

Processing Model – Examples



Greedy Processing Component

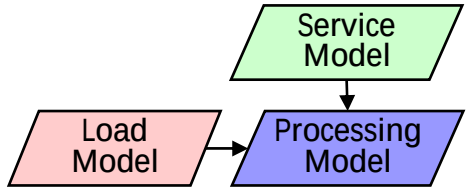


Real-Time Calculus

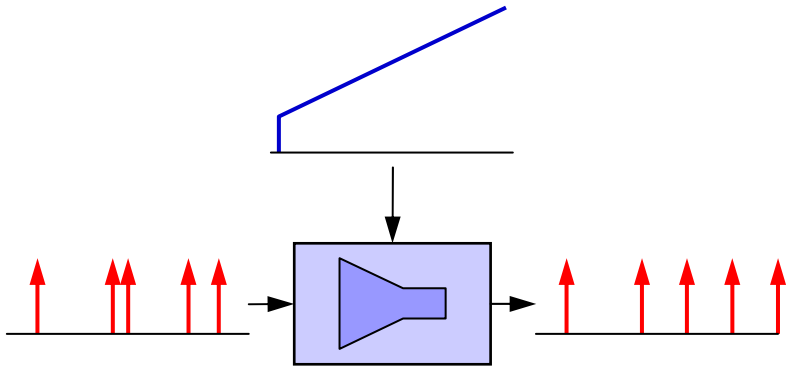
$$
\begin{aligned}
\alpha'^{u} &= \min\{(\alpha^{u} \otimes \beta^{u}) \oslash \beta^{l}, \beta^{u}\} \\
\alpha'^{l} &= \min\{(\alpha^{l} \oslash \beta^{u}) \otimes \beta^{l}, \beta^{l}\} \\
\beta'^{u} &= (\beta^{u} - \alpha^{l}) \overline{\oslash}\, 0 \\
\beta'^{l} &= (\beta^{l} - \alpha^{u}) \overline{\otimes}\, 0
\end{aligned}
$$

$$
\begin{aligned}
(f \otimes g)(\Delta) &= \inf_{0 \le \lambda \le \Delta} \{f(\Delta - \lambda) + g(\lambda)\} \\
(f \oslash g)(\Delta) &= \sup_{\lambda \ge 0} \{f(\Delta + \lambda) - g(\lambda)\} \\
(f \overline{\otimes} g)(\Delta) &= \sup_{0 \le \lambda \le \Delta} \{f(\Delta - \lambda) + g(\lambda)\} \\
(f \overline{\oslash} g)(\Delta) &= \inf_{\lambda \ge 0} \{f(\Delta + \lambda) - g(\lambda)\}
\end{aligned}
$$

Processing Model – Examples



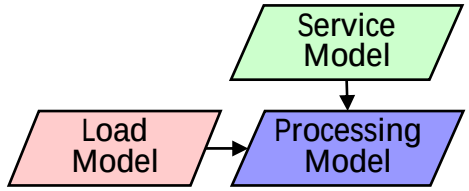
Greedy Shaper Component



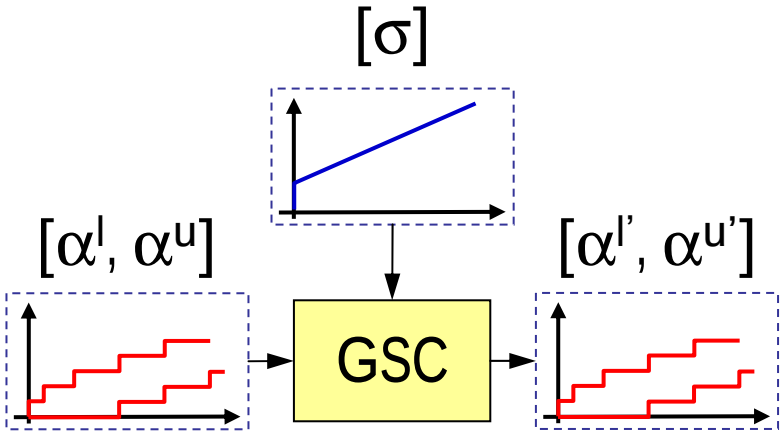
Behavioral Description

- Delays incoming events such that the output conforms to a given traffic specification.
- Guarantees that no events get delayed any longer than necessary.
- Works also with bursty traffic specifications.

Processing Model – Examples



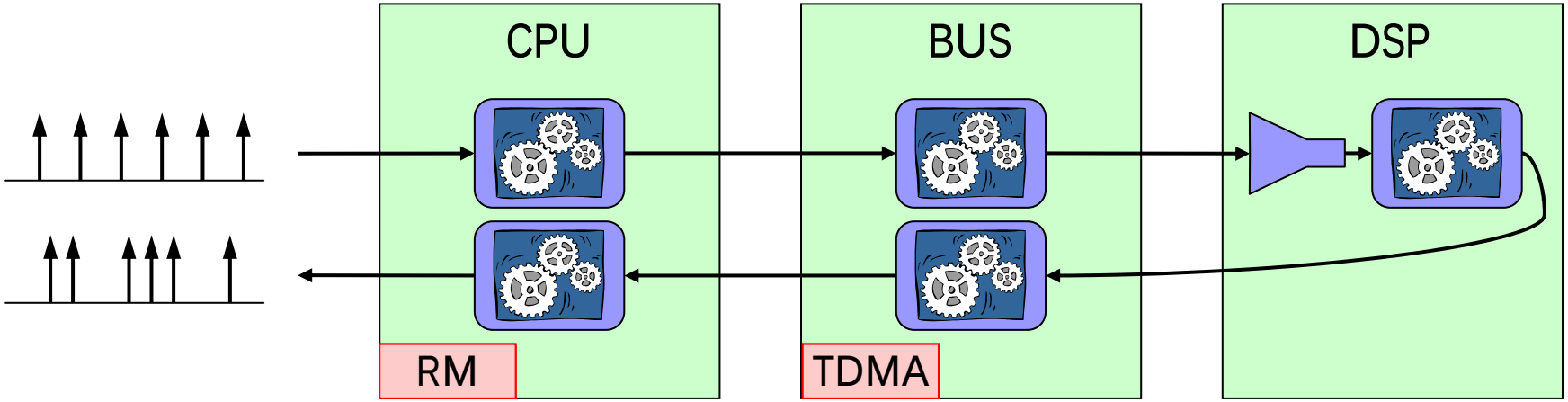
Greedy Shaper Component



Real-Time Calculus

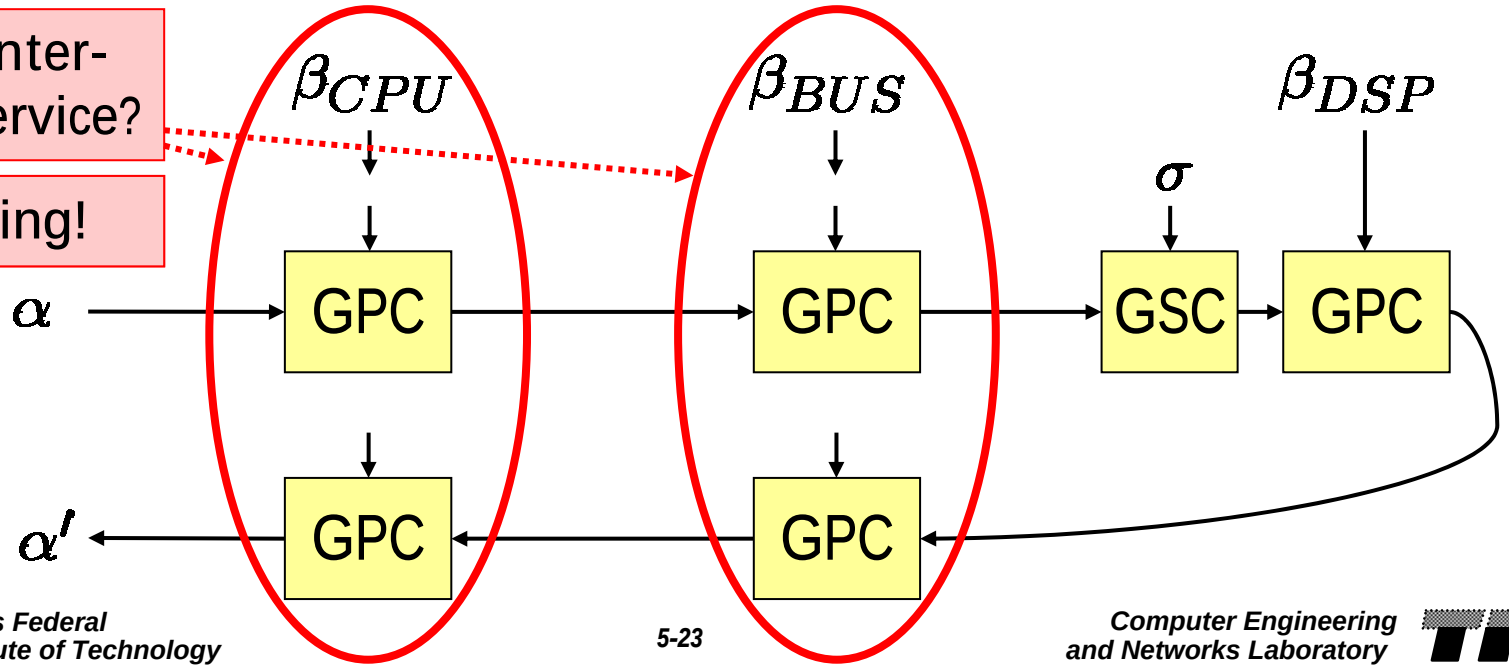
$$\alpha'^{u} = \alpha^{u} \otimes \sigma$$
$$\alpha'^{u} = \alpha^{l} \otimes (\sigma \overline{\oslash} \sigma)$$

System Composition

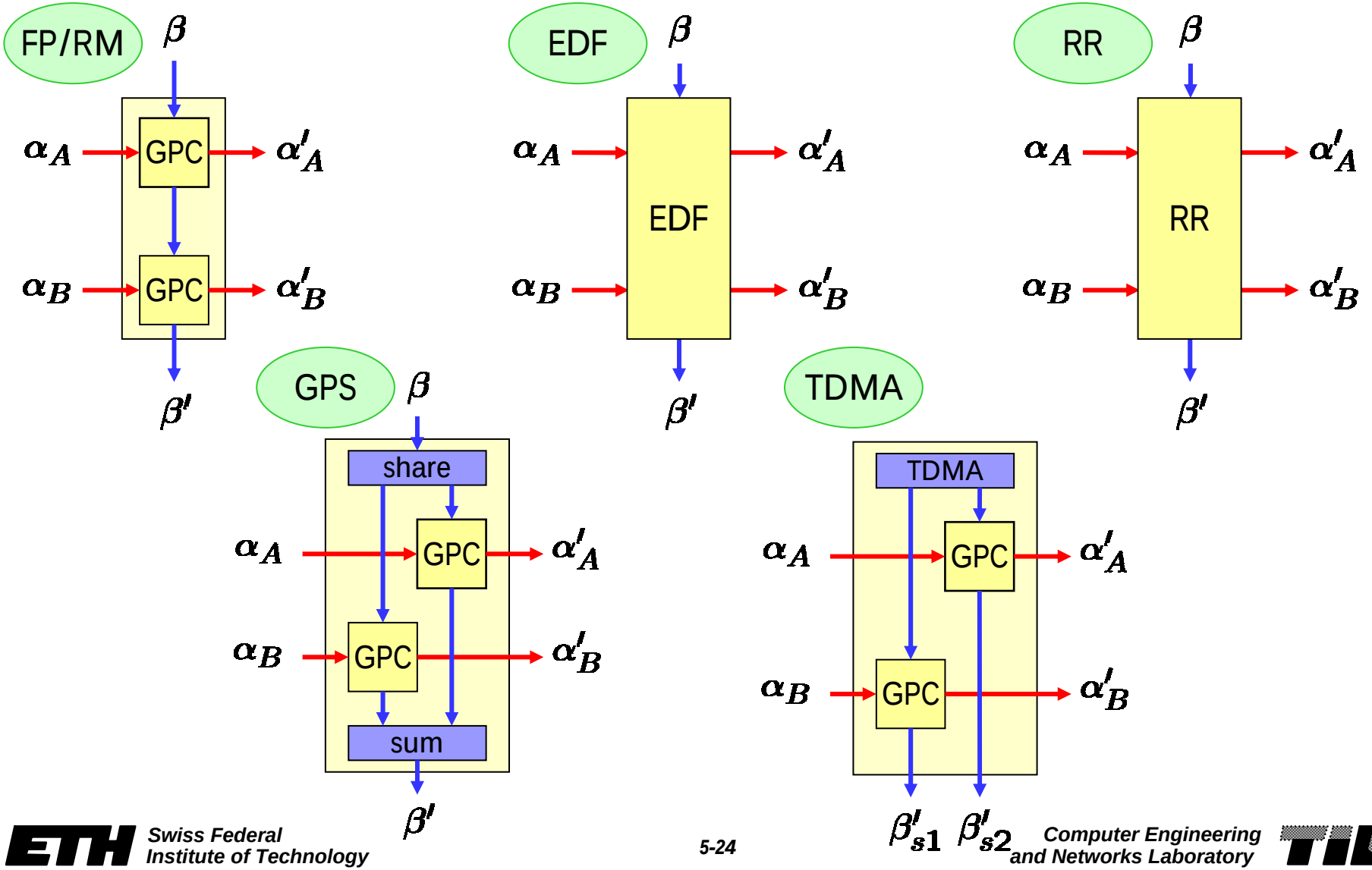


How to inter-connect service?

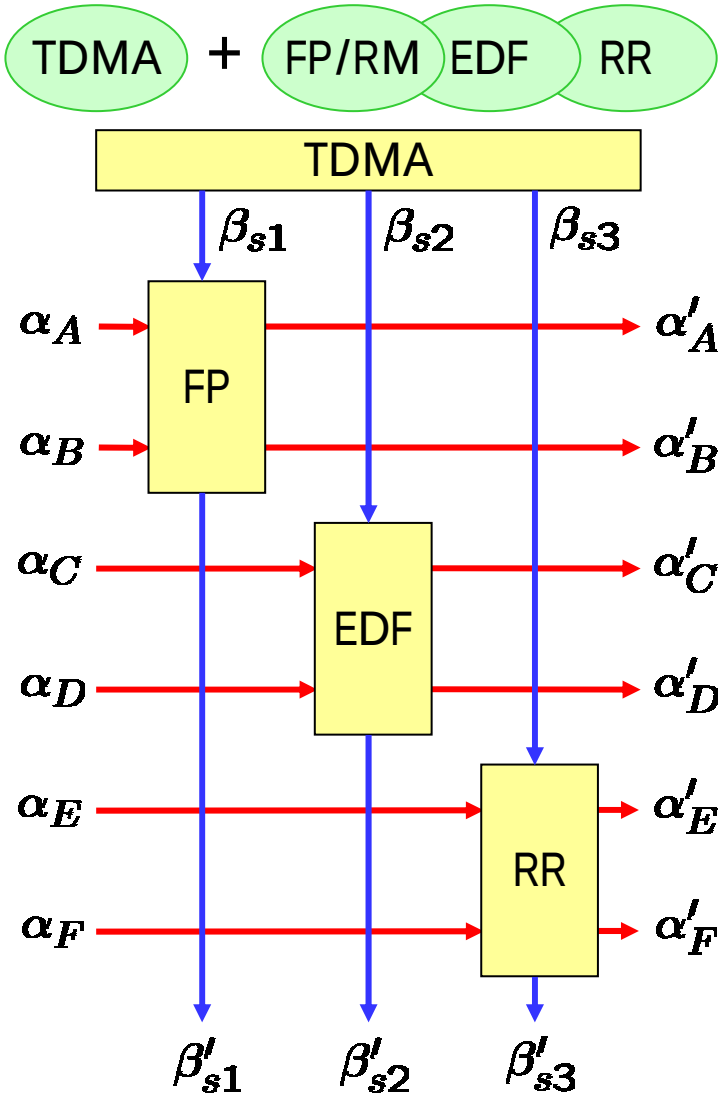
Scheduling!

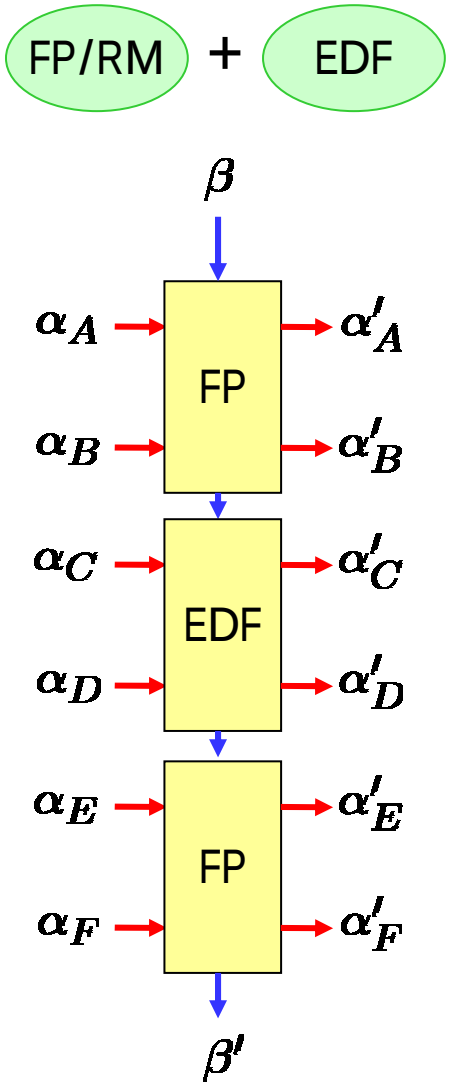


Scheduling and Arbitration



Mixed Hierarchical Scheduling





...and many other combinations:

RR + EDF

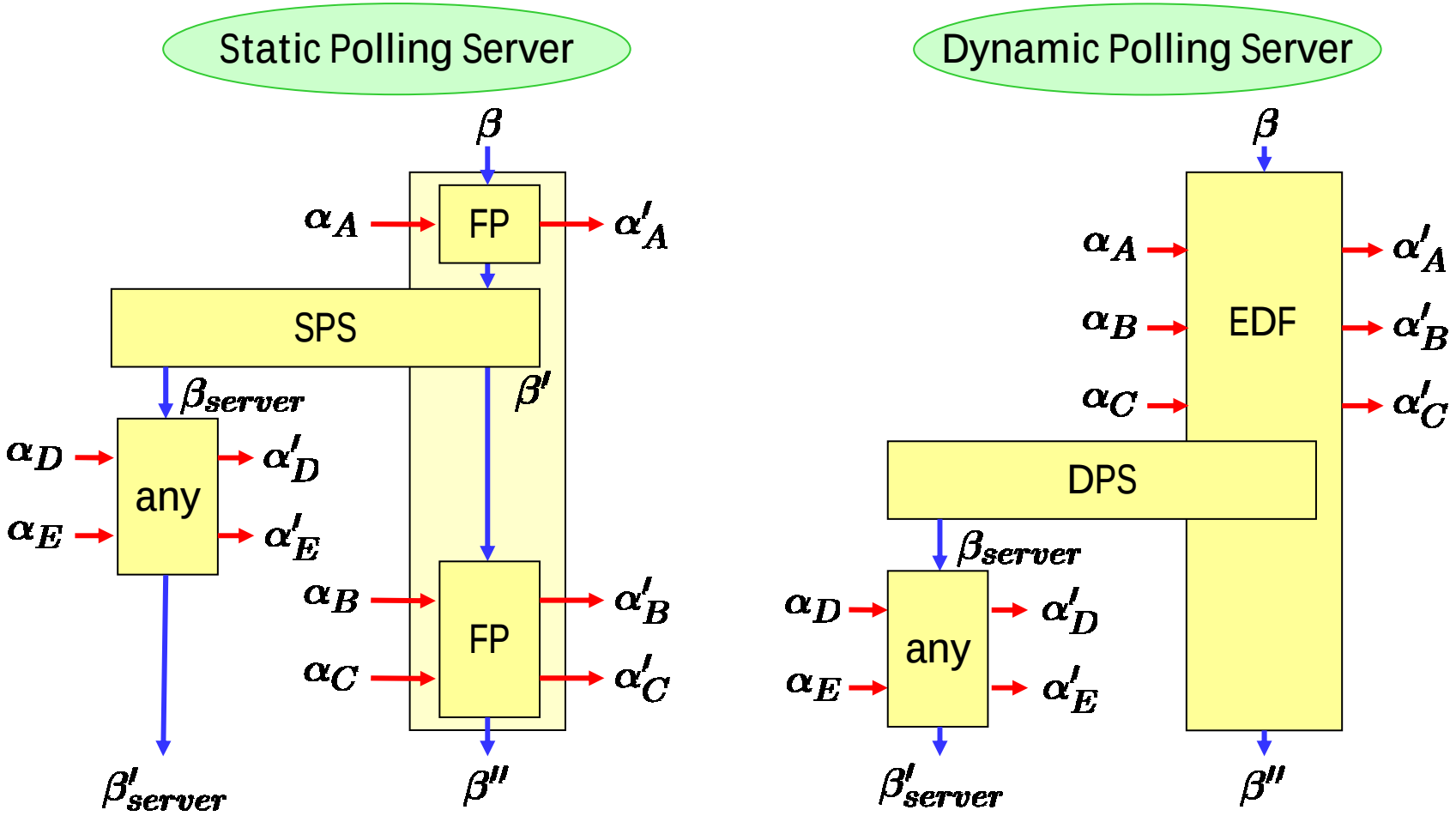
FP/RM + RR

FP/RM + GPS

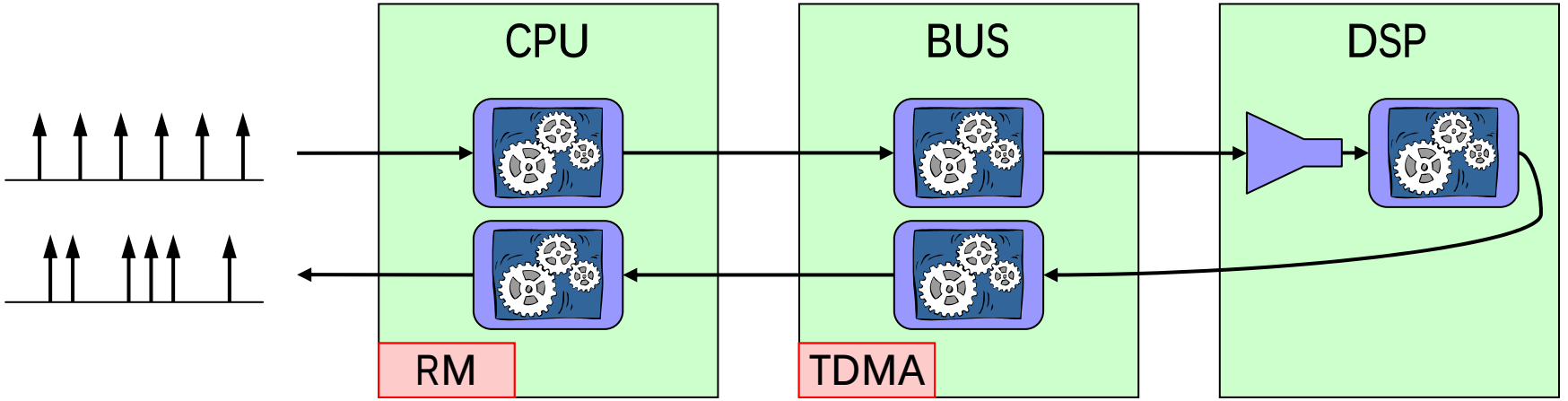
GPS + EDF

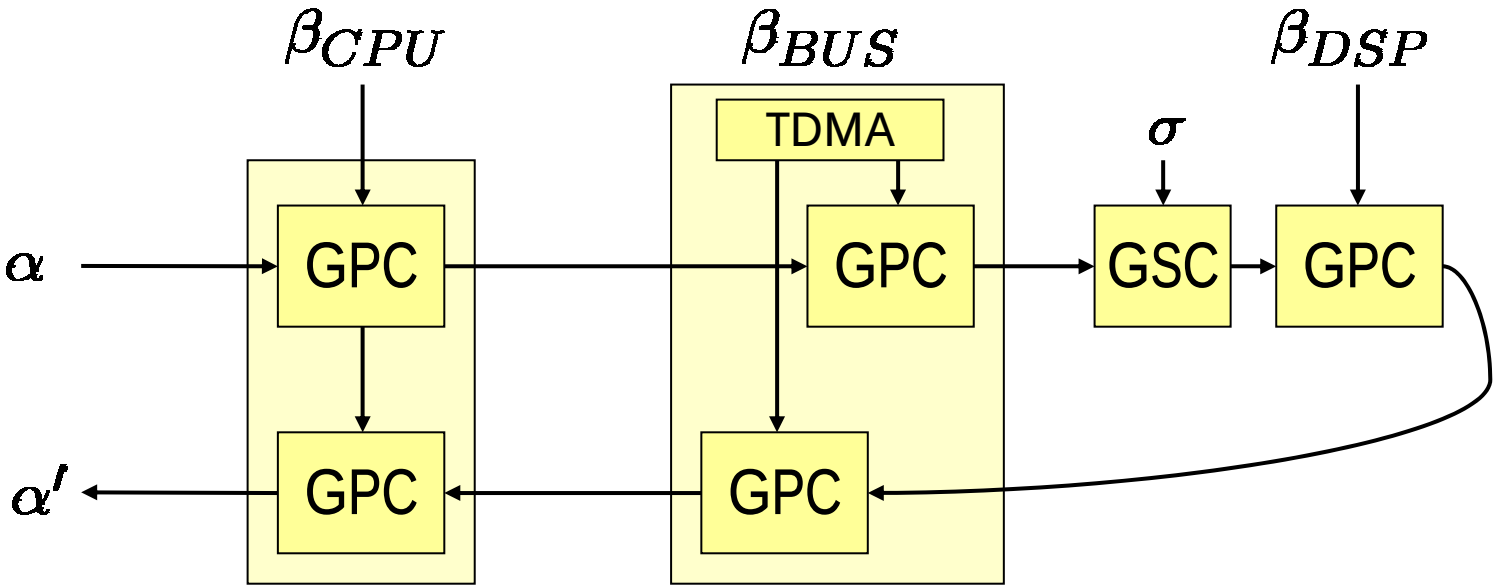
. . .

Hierarchical Scheduling with Servers

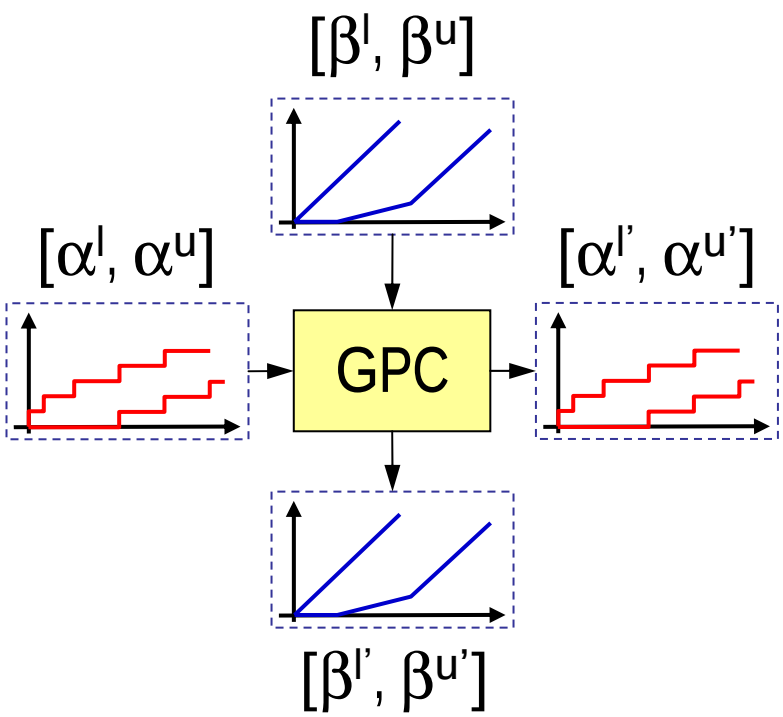


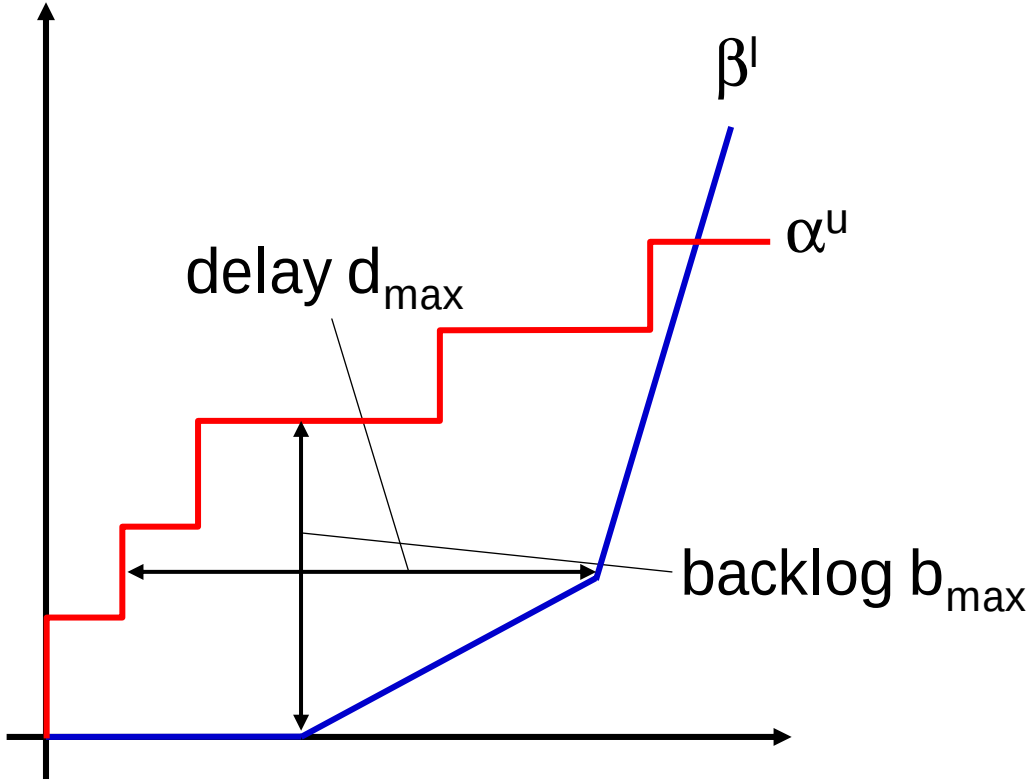
Complete System Composition





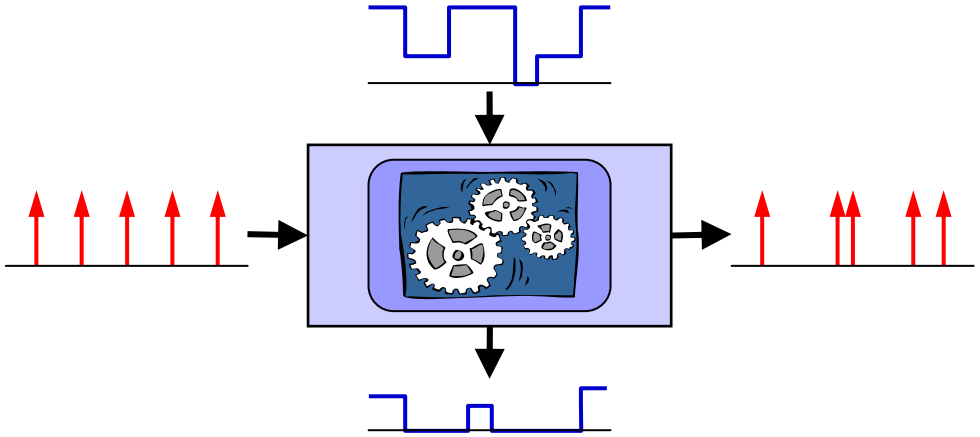
Analysis: Delay and Backlog





Extending the Framework

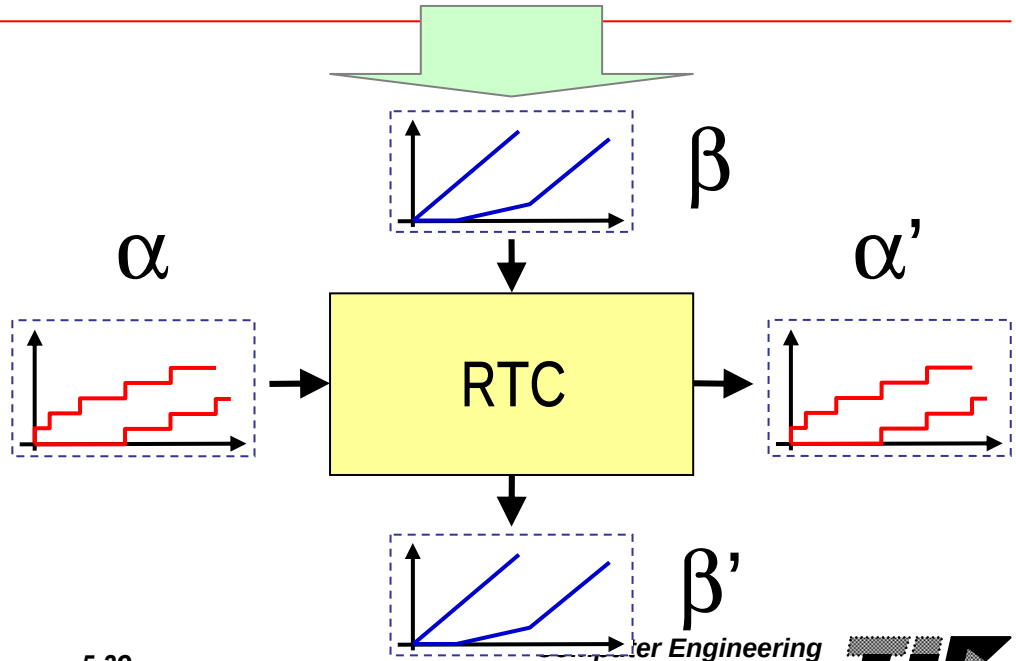
- New HW behavior
- New SW behavior
- New scheduling scheme
- ...



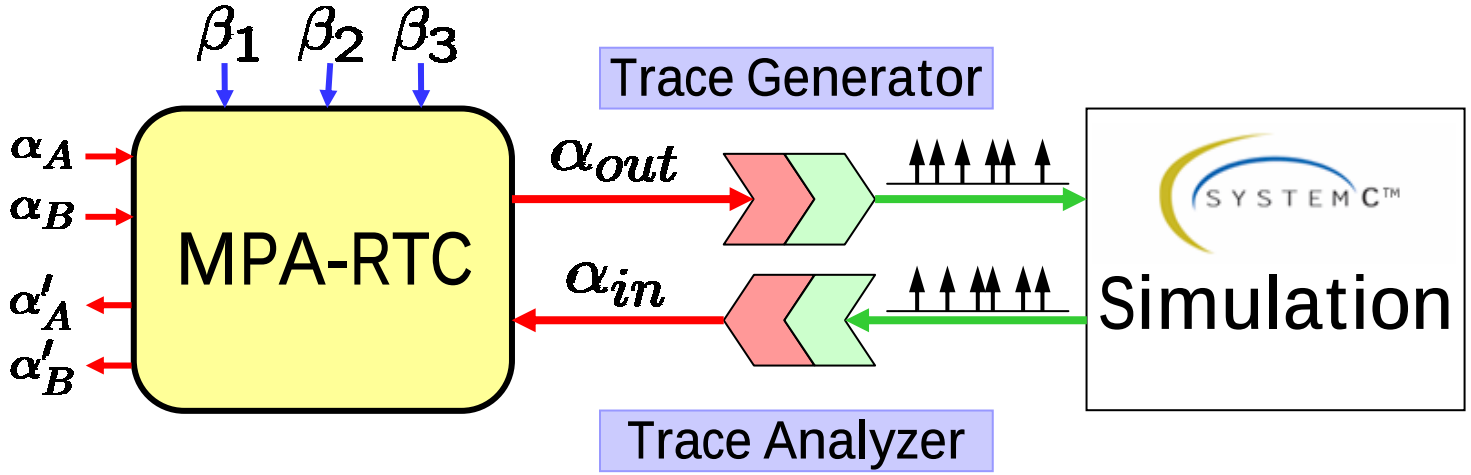
- Find new relations:

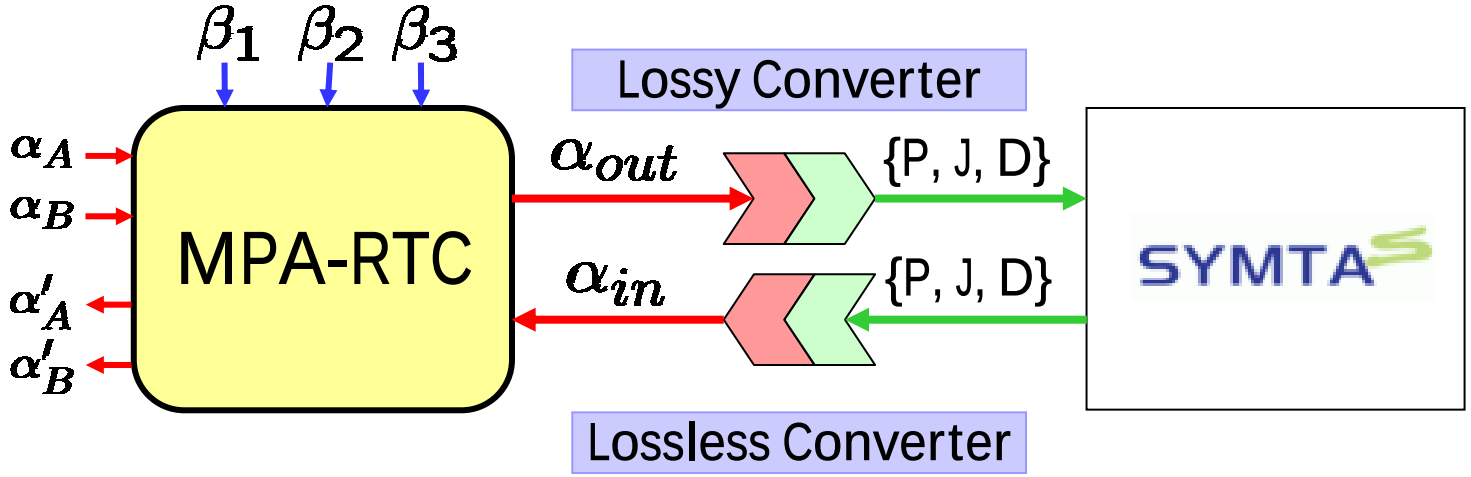
$$\alpha'(\Delta) = f_\alpha(\alpha, \beta)$$
$$\beta'(\Delta) = f_\beta(\alpha, \beta)$$

This is the hard part...!



Embedding with other Frameworks

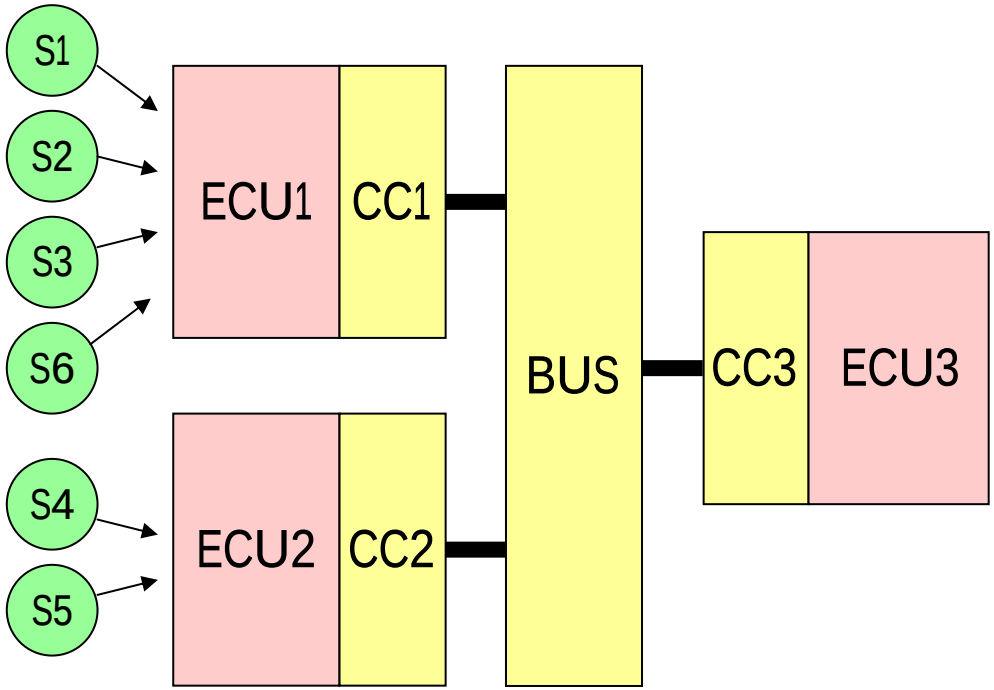




Outline

- Motivation / Problem Statement
- Modular Performance Analysis
- MPA Case Study
- Real-Time Interfaces / Interface-Based Design
- IBD Case Study
- Efficient Computation and Tool Support

Case Study



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

3 ECU's with own CC's

13 Tasks & 7 Messages
- with different WCED

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)

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

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

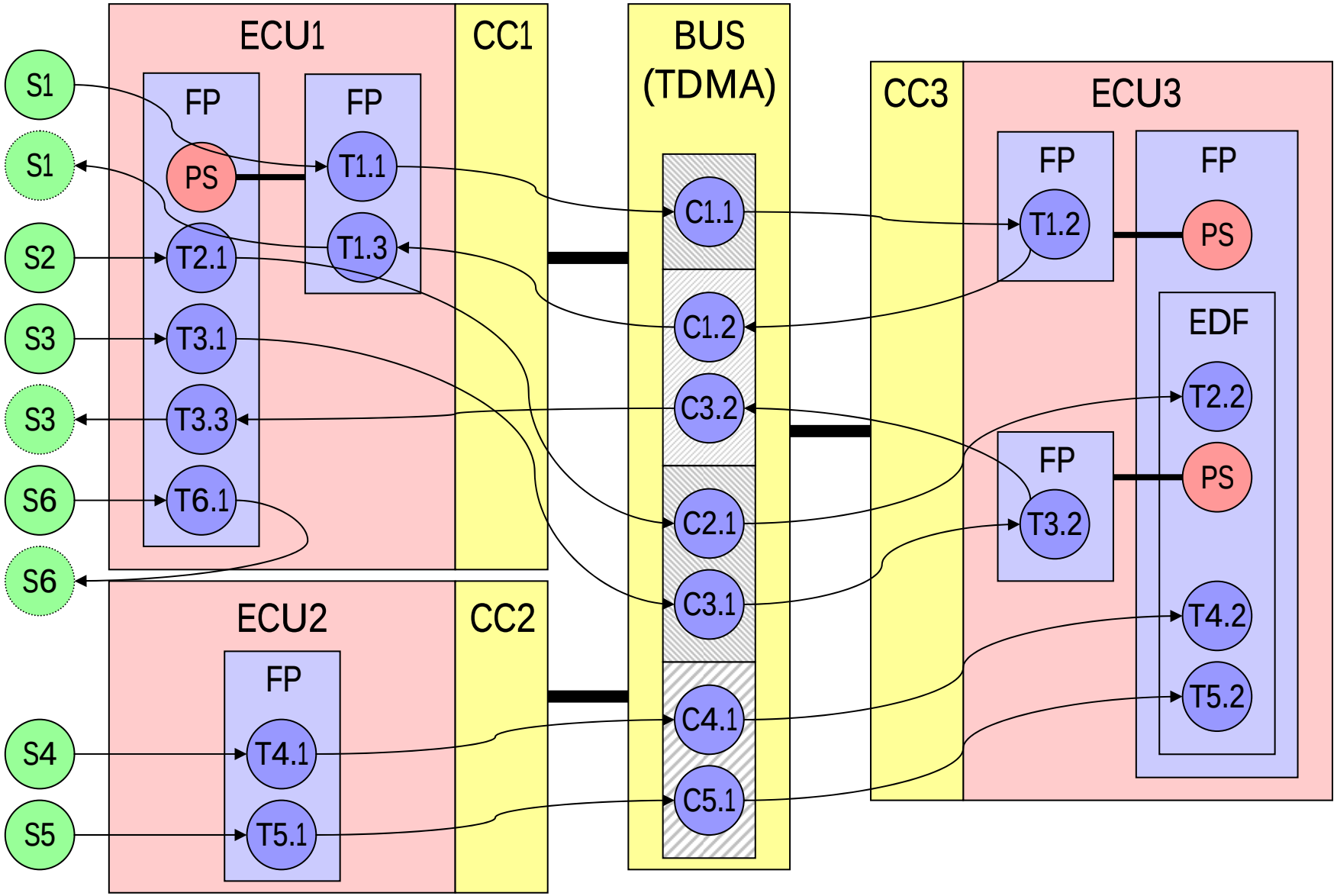
Task	e
T1.1	200
T1.2	300
T1.3	30
T2.1	75
T2.2	25
T3.1	60
T3.2	60
T3.3	40
T4.1	12
T4.2	2
T5.1	8
T5.2	3
T6.1	100

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

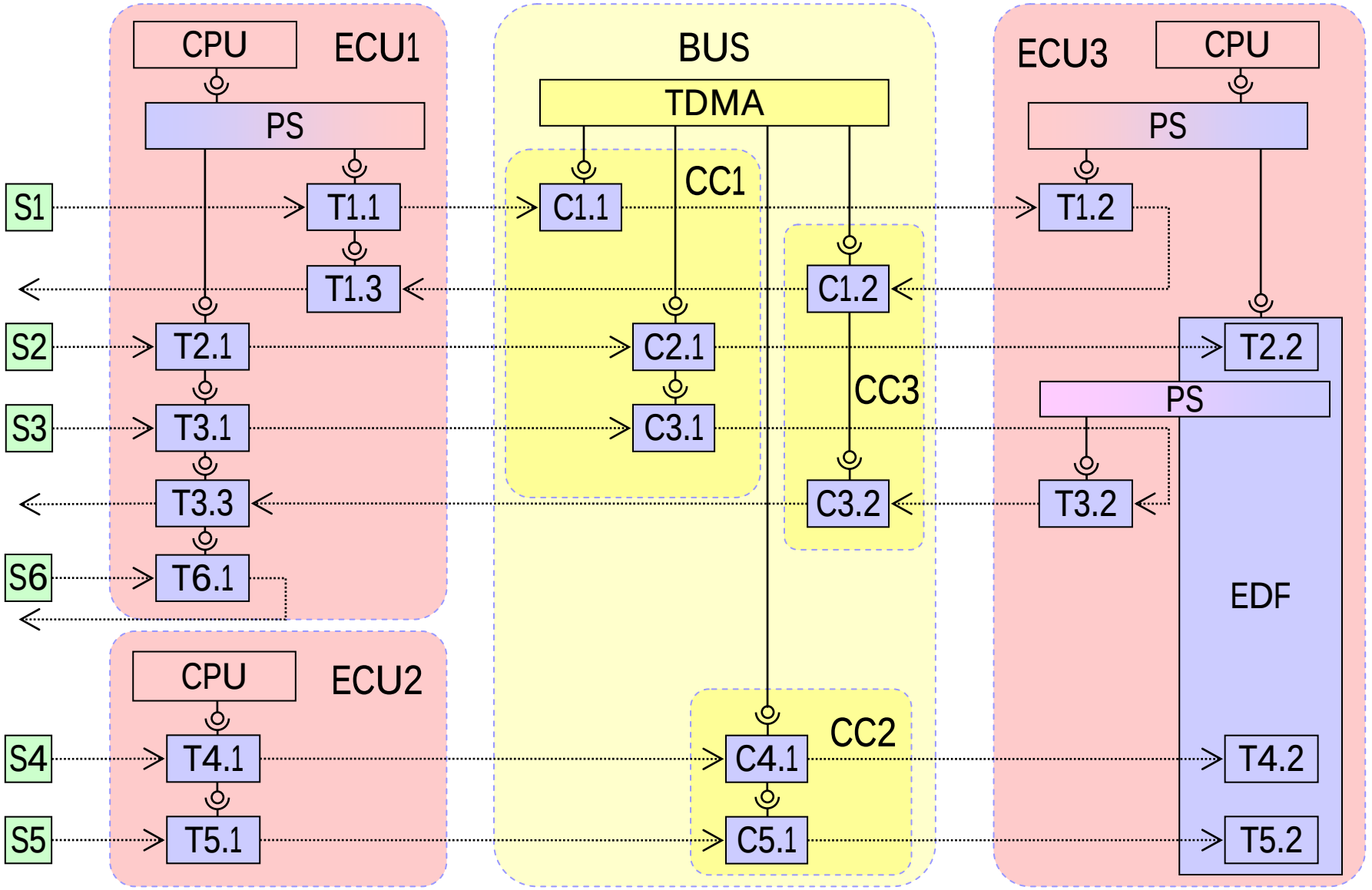
Perdiodic Server	p	e
SPS_{ECU1}	500	200
SPS_{ECU3}	500	250
DPS_{ECU3}	600	120

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

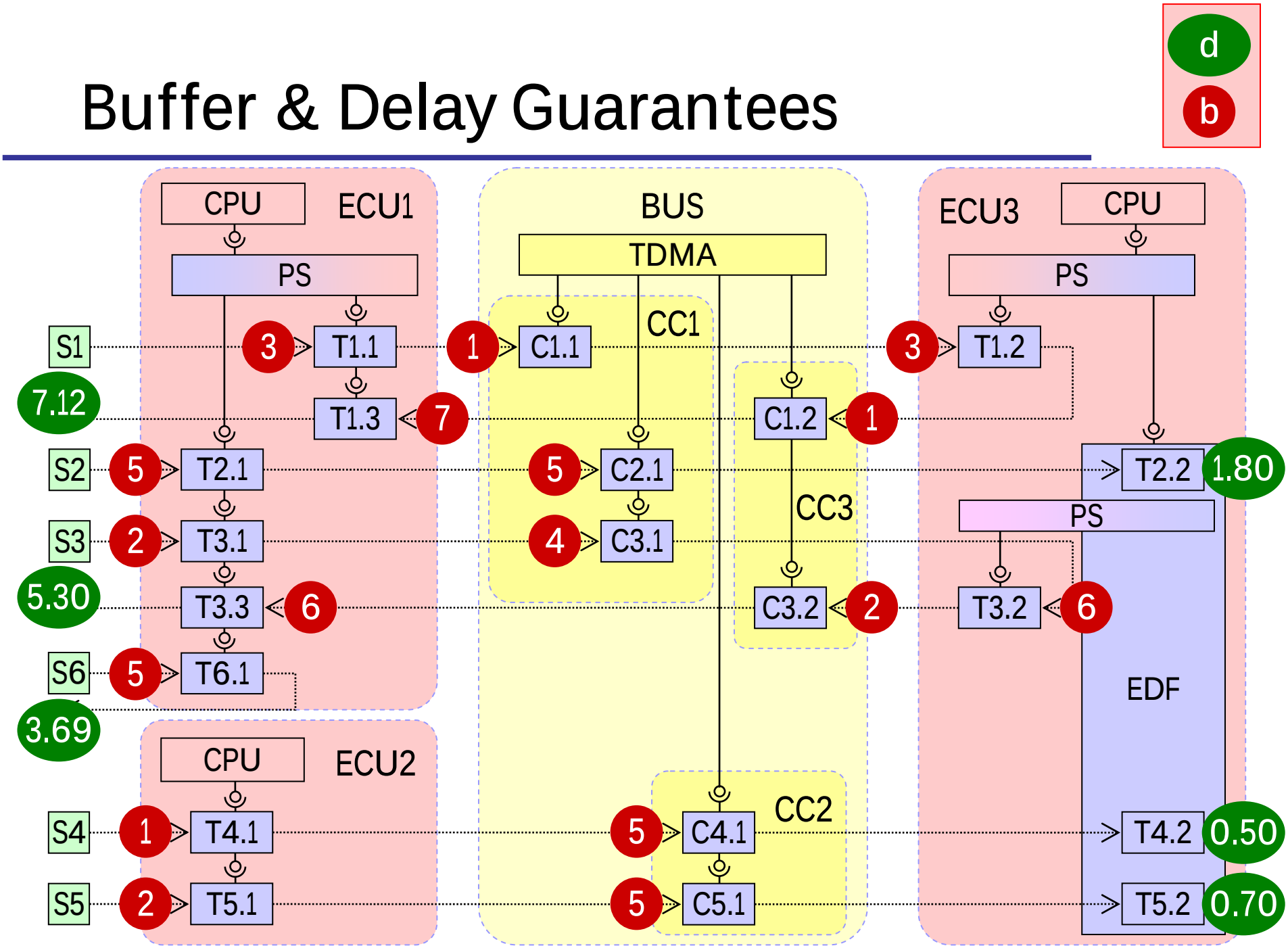
The Distributed Embedded System...



... and its MPA Model

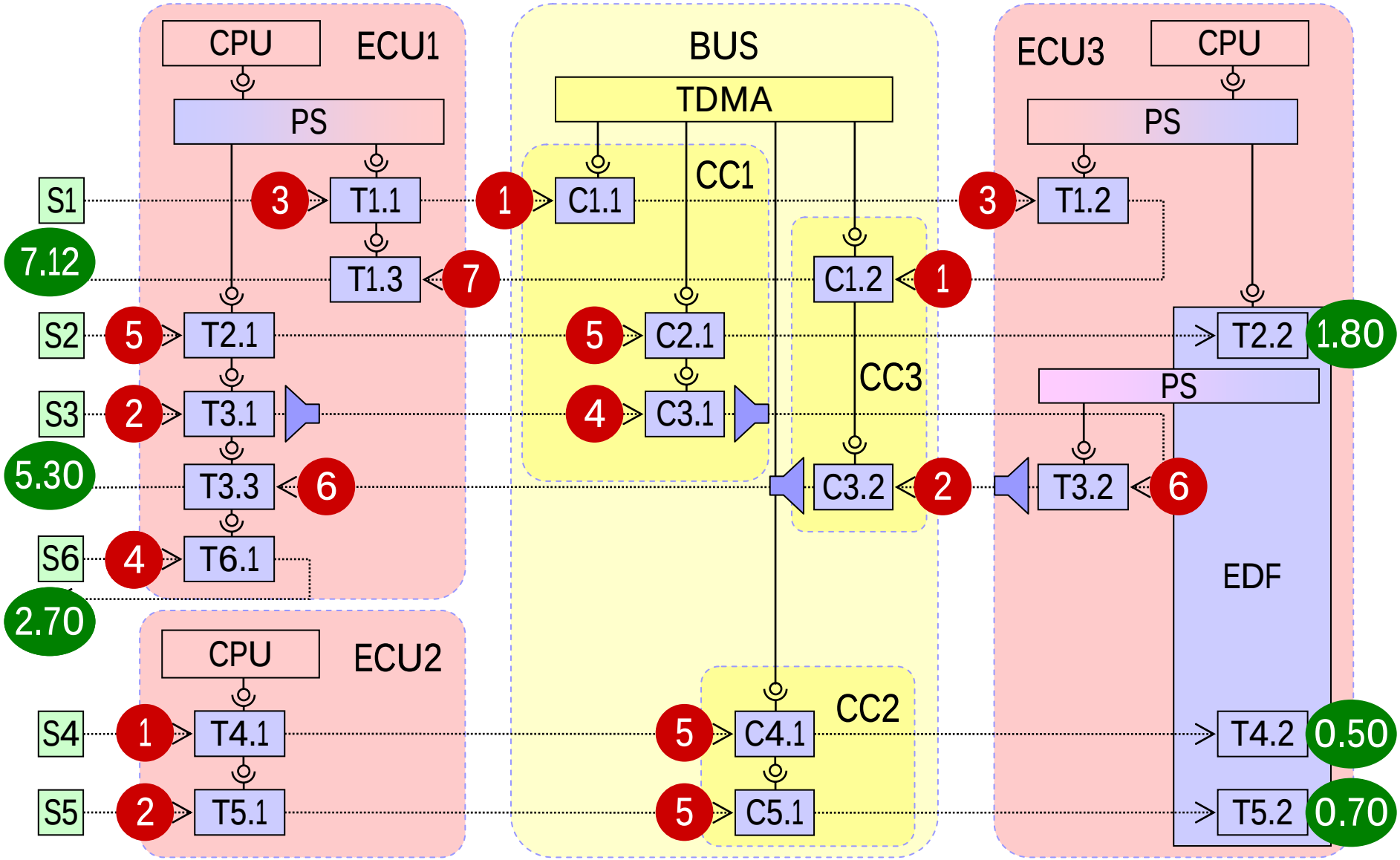


Buffer & Delay Guarantees

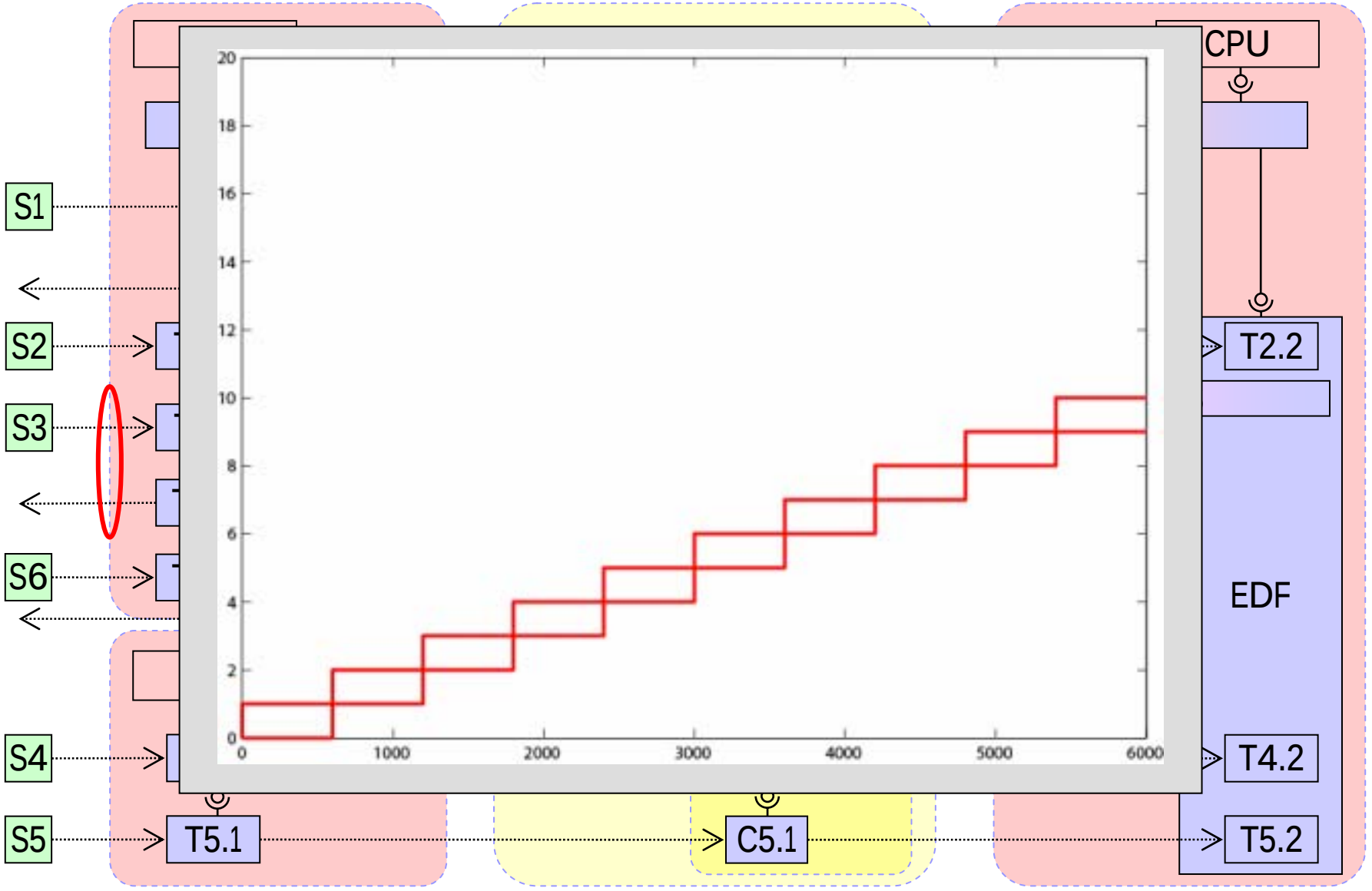


Adding Greedy Shapers

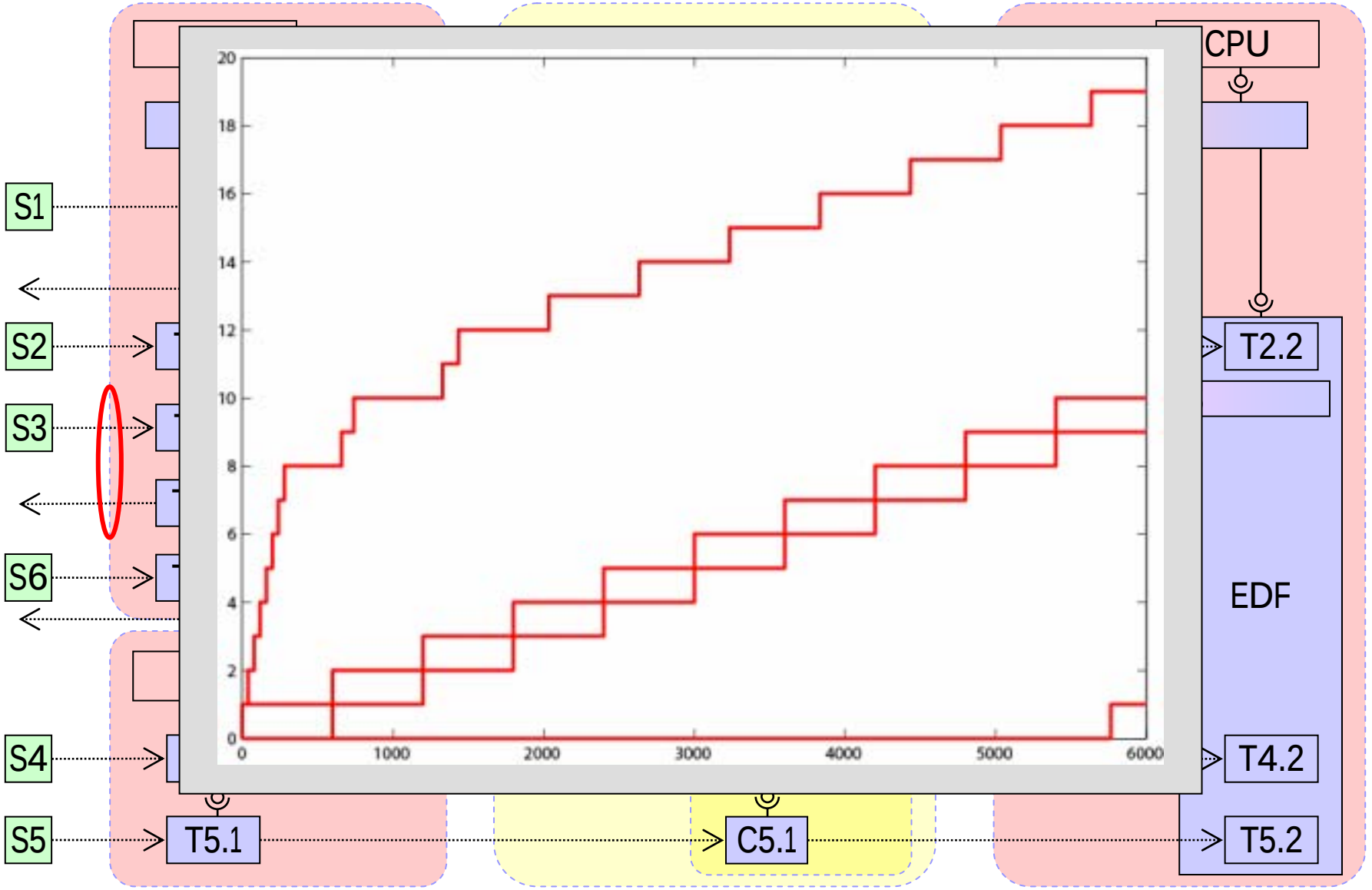
Delay D_{S6}: - 27%
Buffer B_{S6}: - 20%



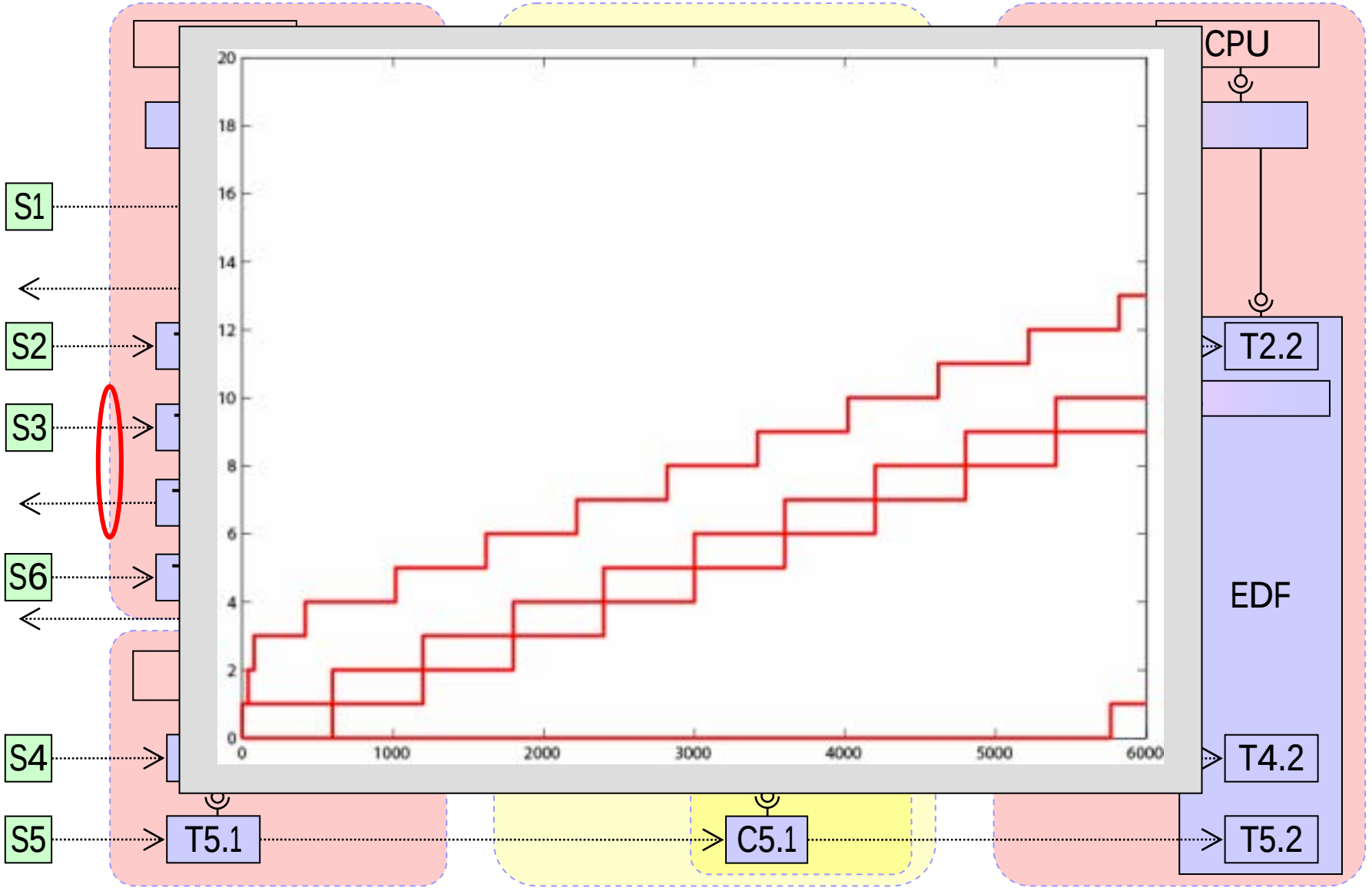
Input of Stream 3



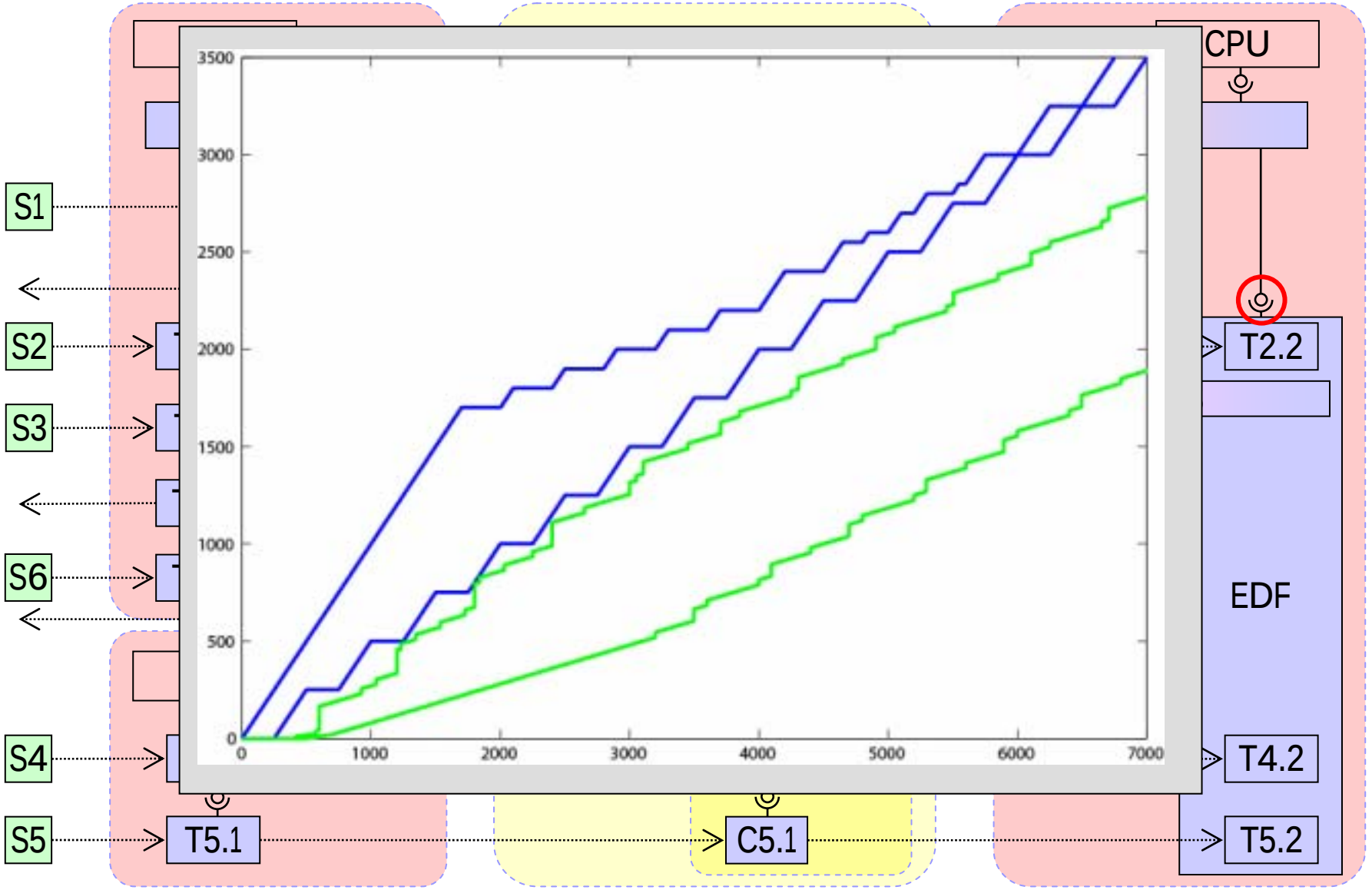
Output of Stream 3



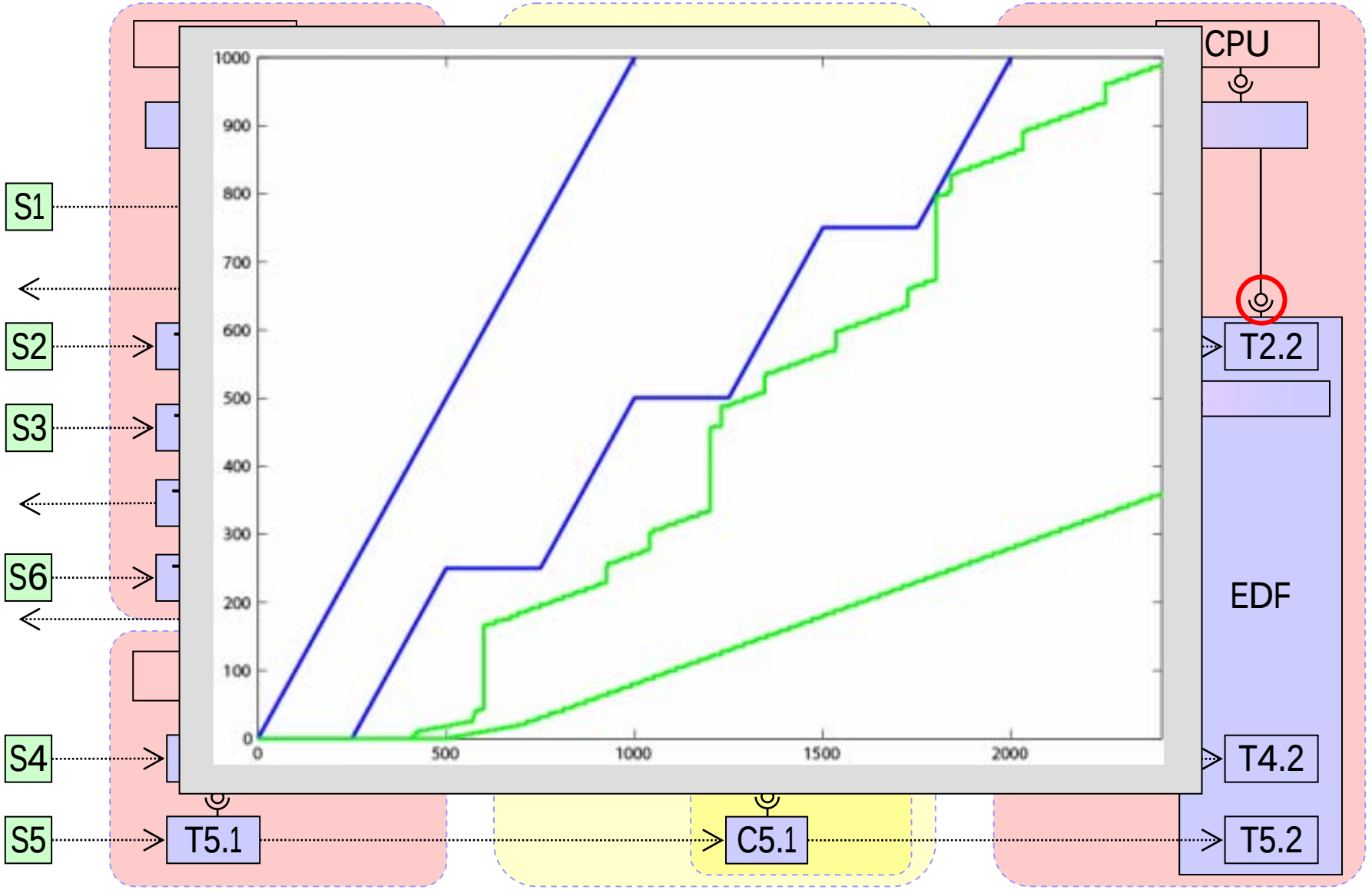
Output of Stream 3 with Greedy Shapers



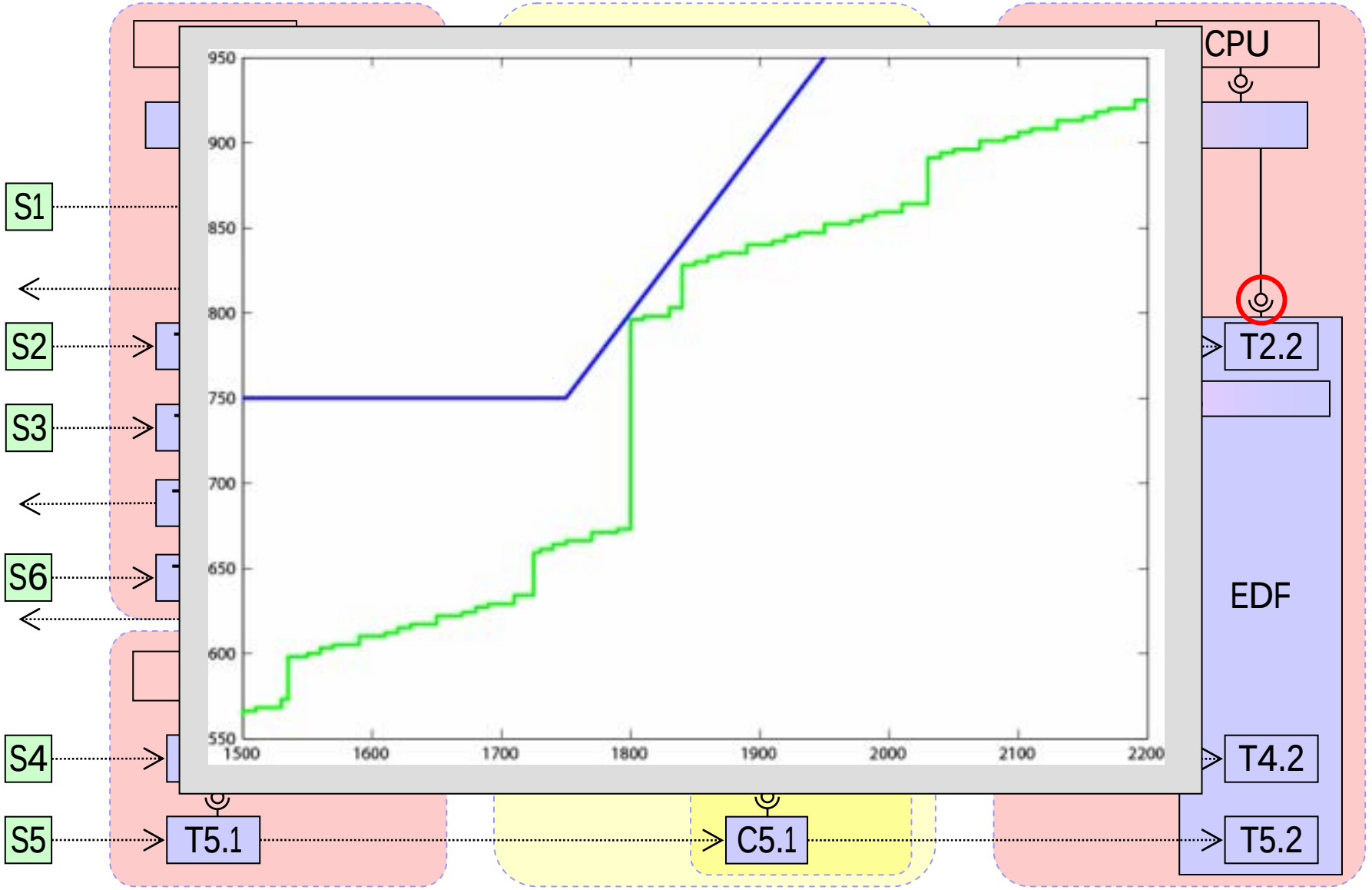
Service Demand & Supply for EDF Block



Service Demand & Supply for EDF Block



Service Demand & Supply for EDF Block



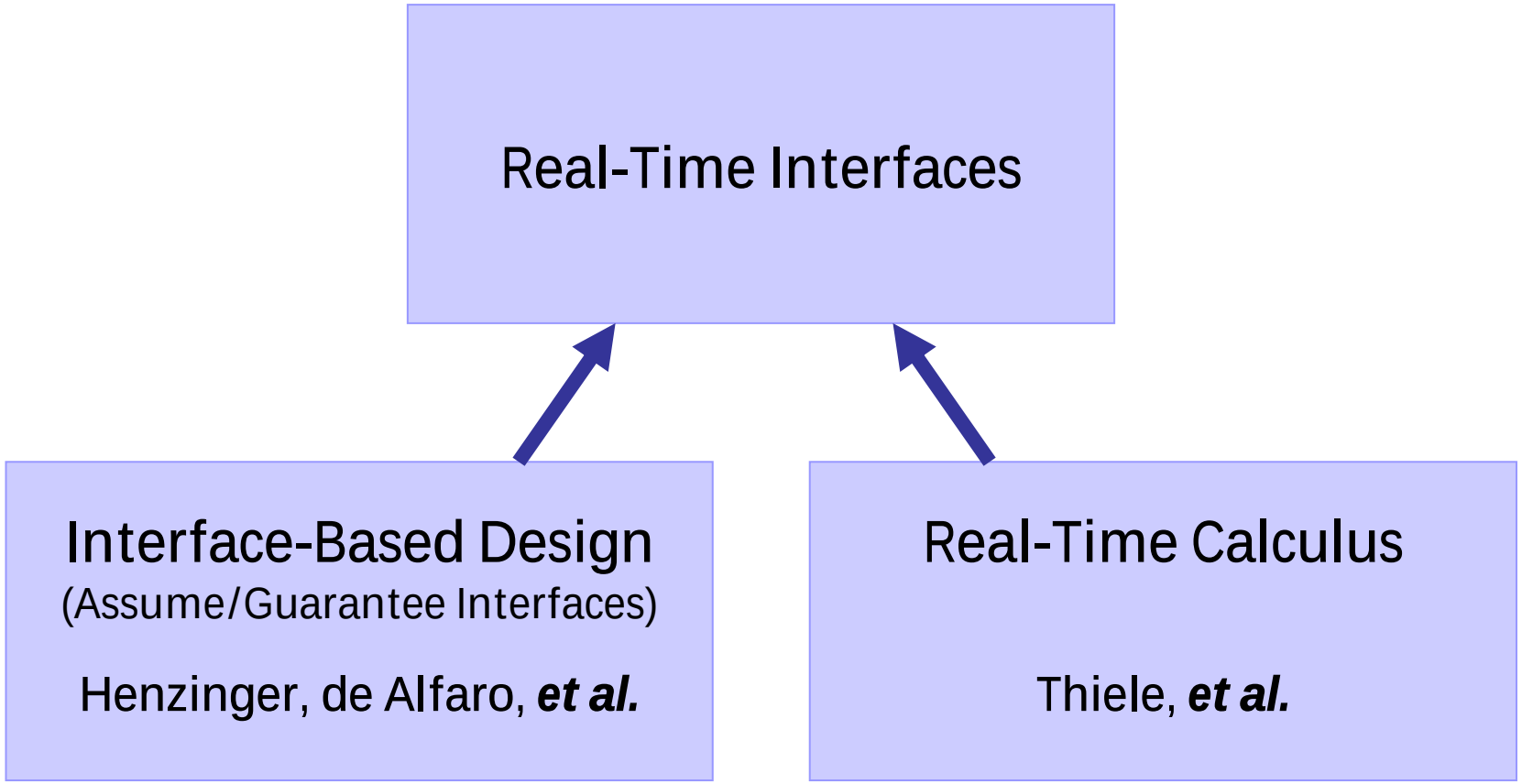
System Analysis Time

- 10 seconds
  - Pentium Mobile 1.6 GHz
  - Matlab 7 SP2
  - RTC Toolbox

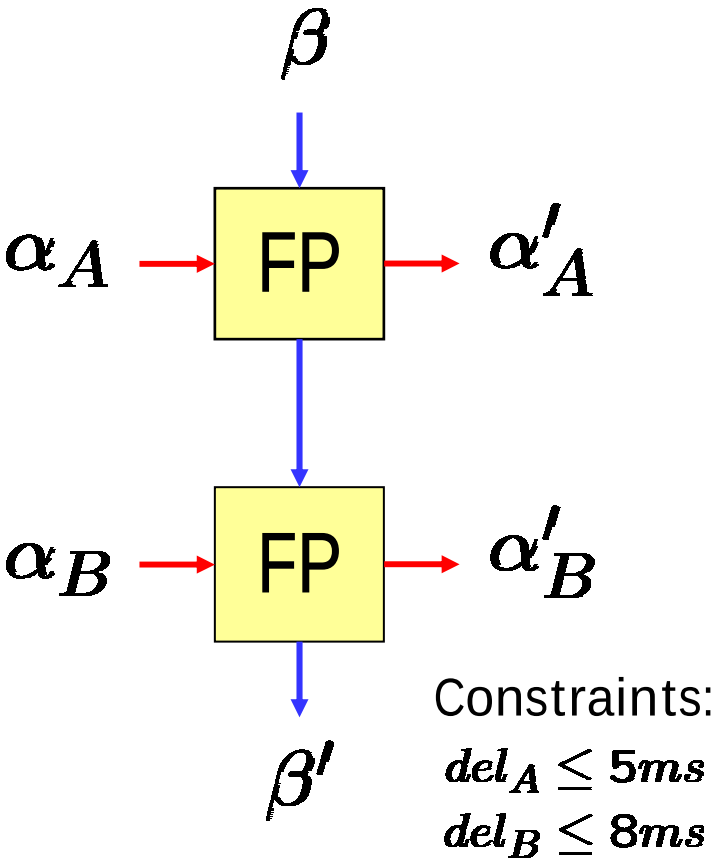
Outline

- Motivation / Problem Statement
- Modular Performance Analysis
- MPA Case Study
- Real-Time Interfaces / Interface-Based Design
- IBD Case Study
- Efficient Computation and Tool Support

Real-Time Interfaces



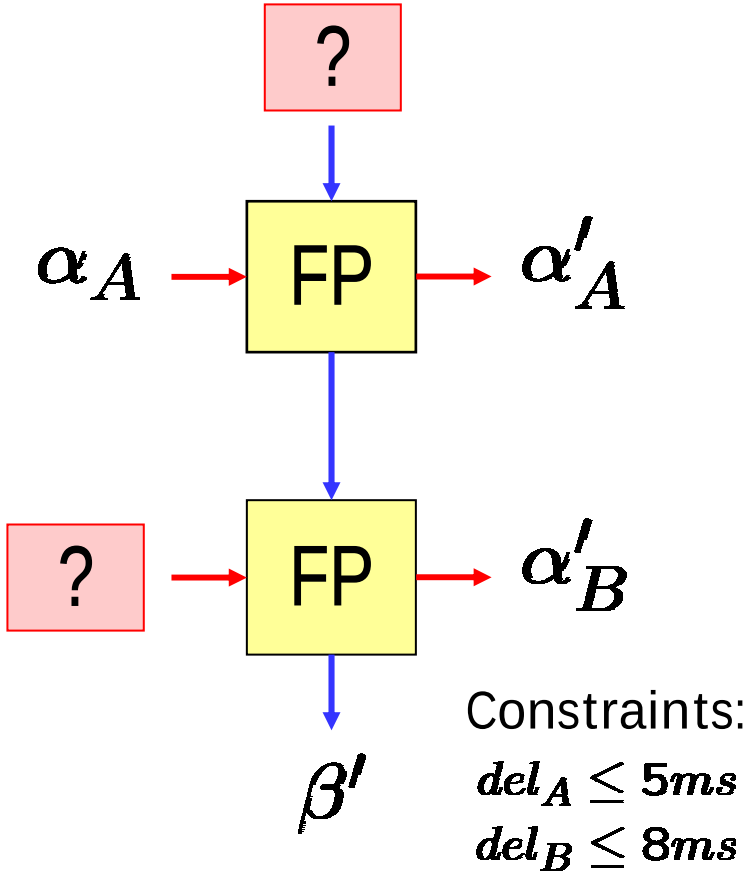
Component-Based Design



Schedulable?

1. ***Design***
2. ***Analysis***
   - Given: ***all*** components, their interconnections structure and all inputs from environment
   - Question: do the components ***work together properly***?

Component-Based Design



1. *Design and Composition*

- Given: ***some*** components, their interconnection structure and some inputs from environment
- Questions: Is there the chance that the components ***work together properly***? What are the ***assumptions*** towards the environment? How can I ***change the environment*** such that the components still work together?

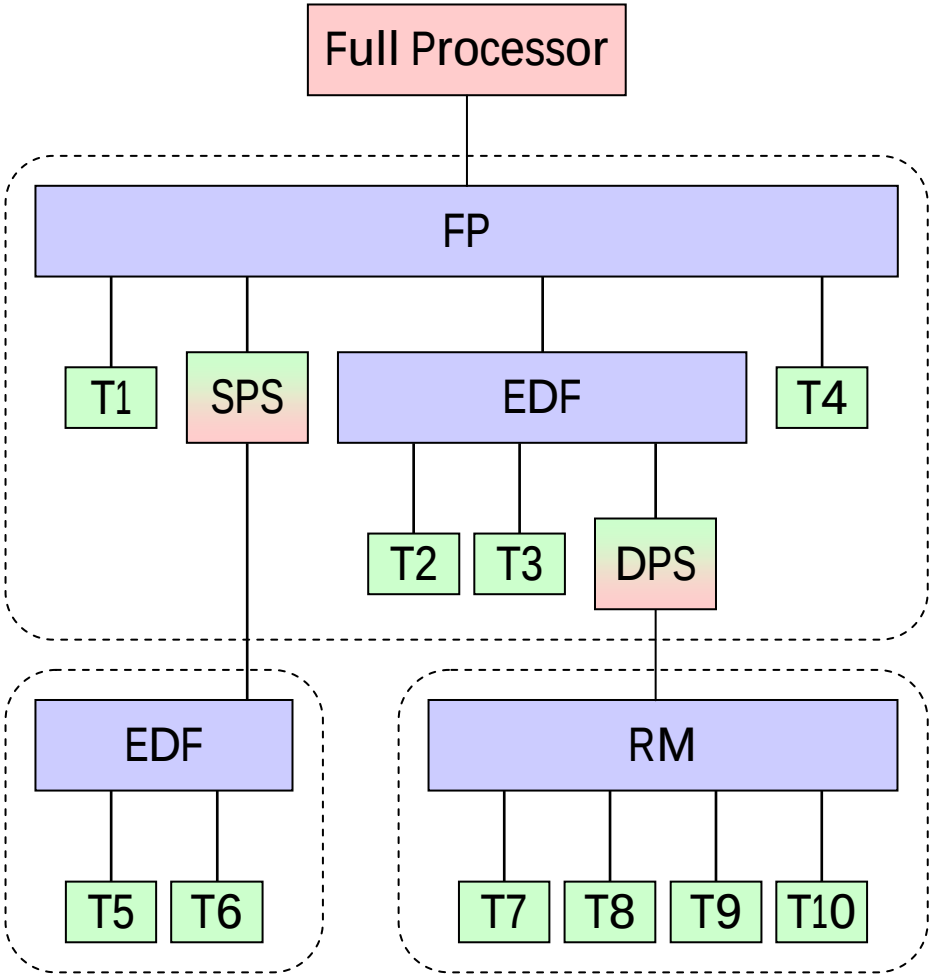
Applications of Real-Time Interfaces

- Interface-Based Design:
  - Find minimum processor speed for complex systems with mixed hierarchical scheduling.
  - Find optimal TDMA slot and cycle length allocations.
  - Specify maximum allowable input stream rates.
  - …
- Interface-Based Schedulability Analysis
- On-Line Service & Load Adaption
- On-Line Admission Tests

Outline

- Motivation / Problem Statement
- Modular Performance Analysis
- MPA Case Study
- Real-Time Interfaces / Interface-Based Design
- IBD Case Study
- Efficient Computation and Tool Support

A System with Complex Scheduling...



10 Tasks
- with jitter
- with bursts
- deadline = period
- deadline < period
- deadline > period

3 Scheduling Policies
- Rate Monotonic
- Earliest Deadline First
- Fixed Priority

Hierarchical Scheduling
Static & Dynamic Polling Servers

Total Utilization: 98.5%

Real-Time Load Specification

Load	$\hat{\alpha}^G(P, J, D)$	$\hat{d}^G$	e	øU
T1	(5, 0, 0)	2	0.2	0.04
T2	(10, 5, 0)	10	1	0.1
T3	(25, 0, 0)	15	1.5	0.06
T4	(100, 2, 0)	150	40	0.4
T5	(25, 80, 5)	50	2	0.08
T6	(20, 0, 0)	30	2	0.1
T7	(12, 0, 0)	12	0.5	0.042
T8	(16, 0, 0)	16	0.75	0.047
T9	(20, 0, 0)	20	1	0.05
T10	(30, 0, 0)	30	2	0.067
Total				0.985

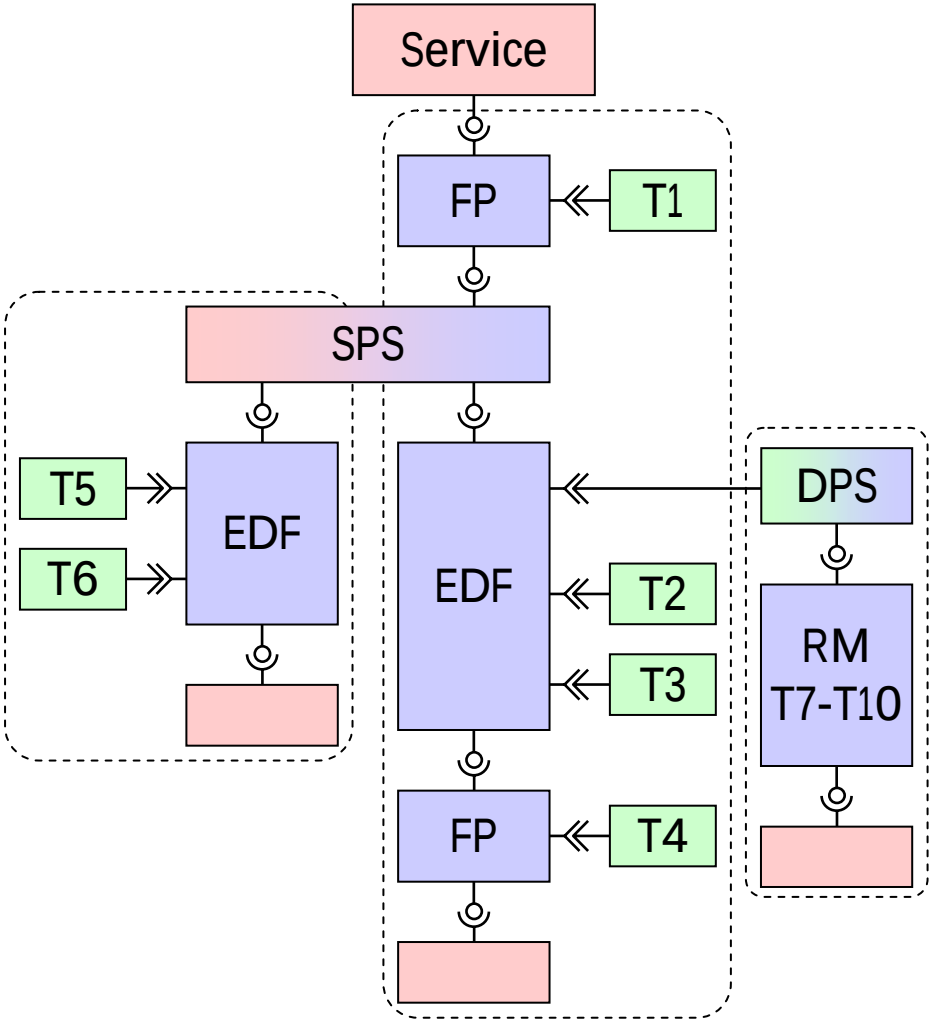
bursty

d > P

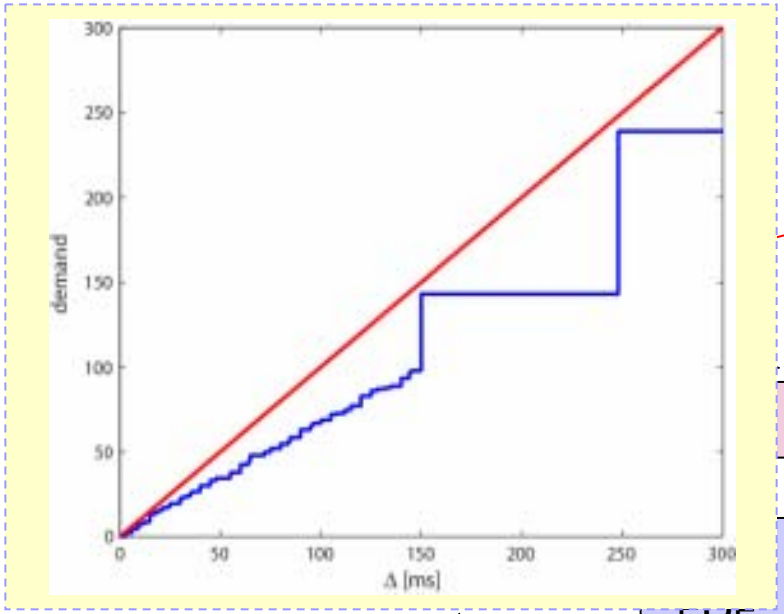
... and its Real-Time Interface Model

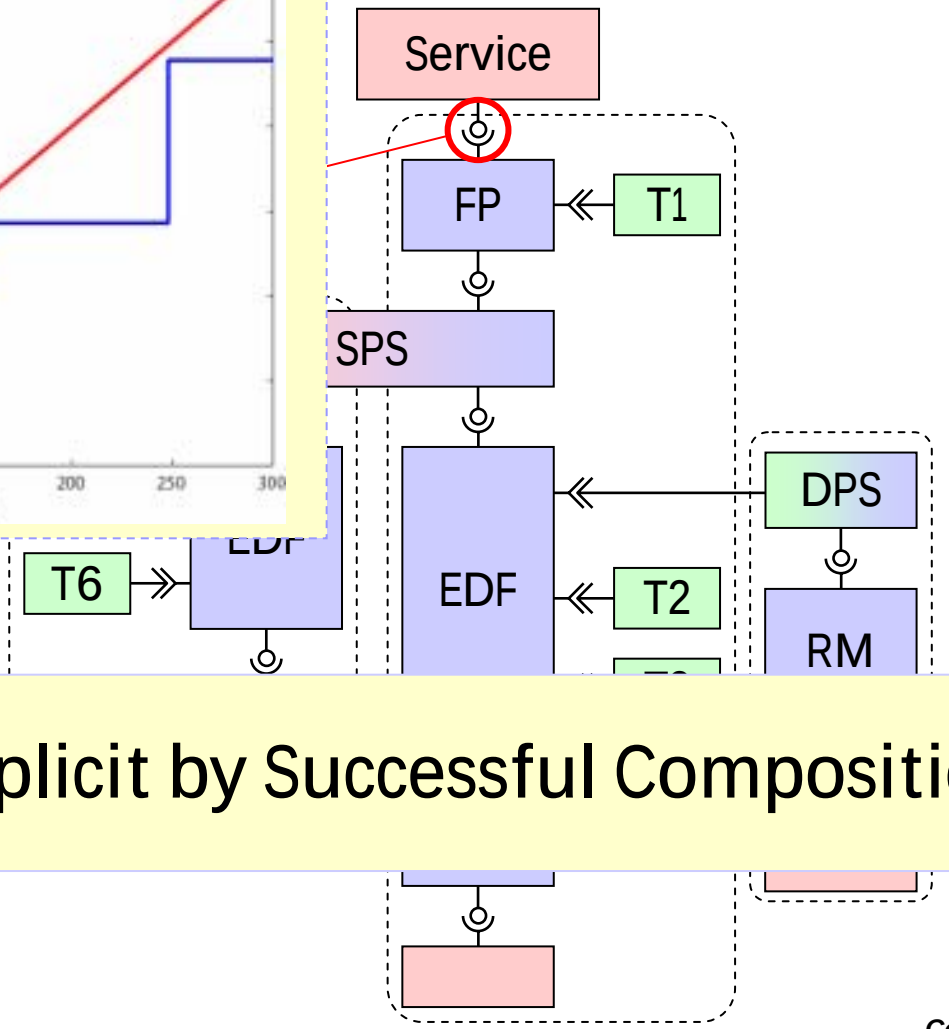
Combining RTC with Assume/Guarantee Interfaces

Constraint propagation!



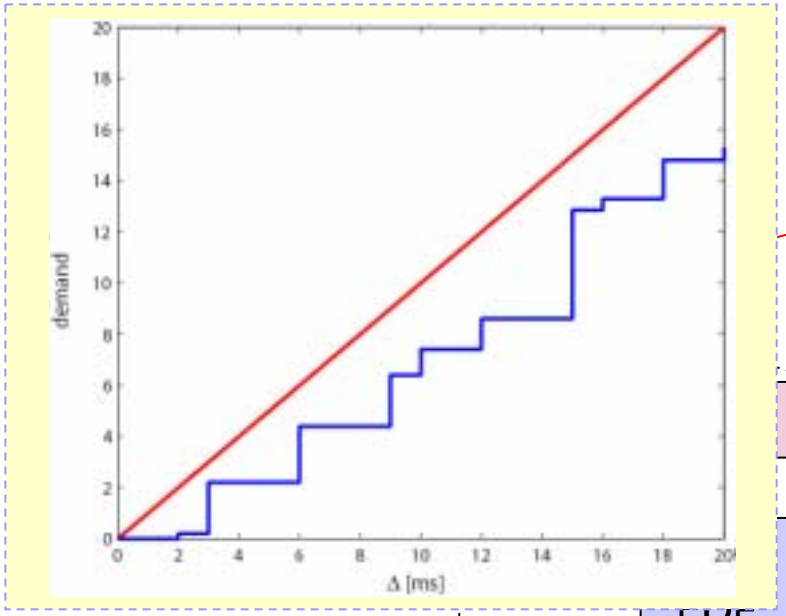
Schedulability Analysis

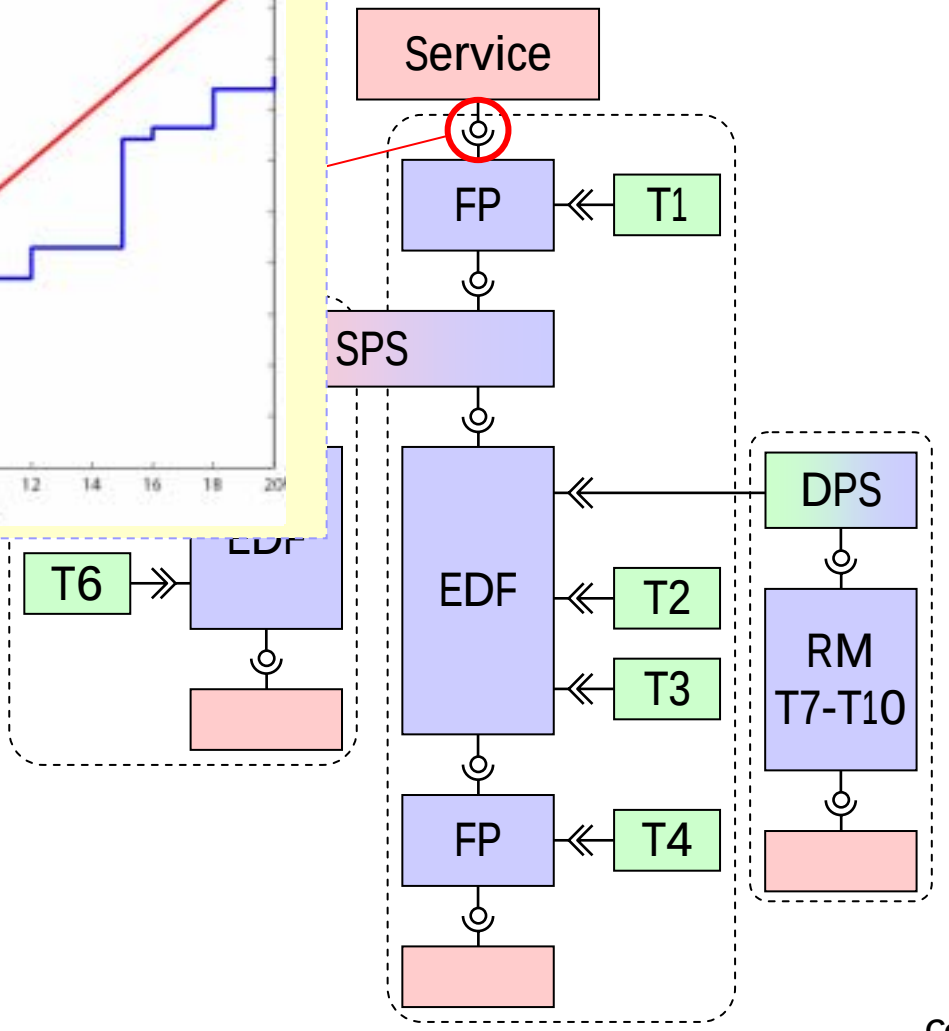




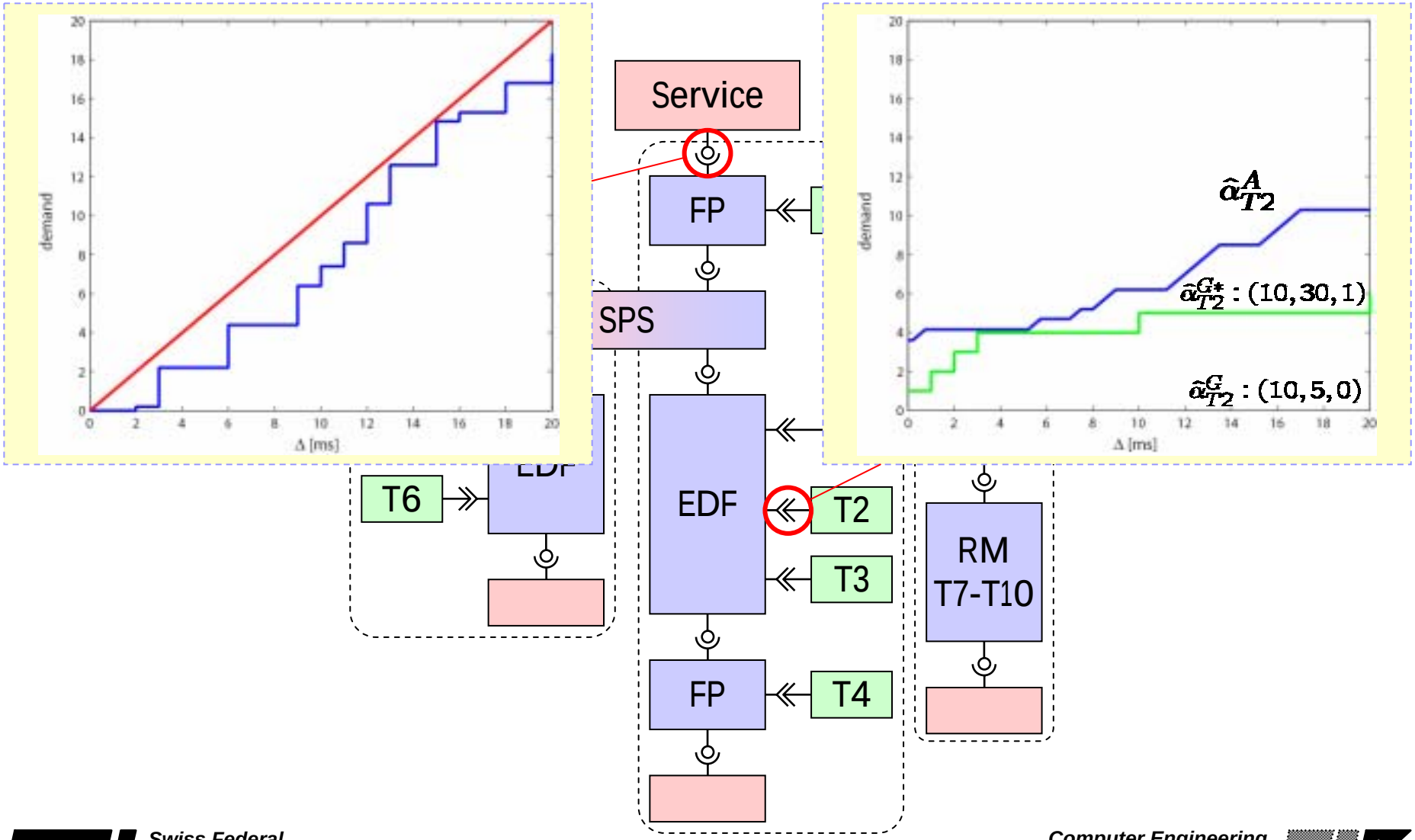
Implicit by Successful Composition!

Schedulability Analysis

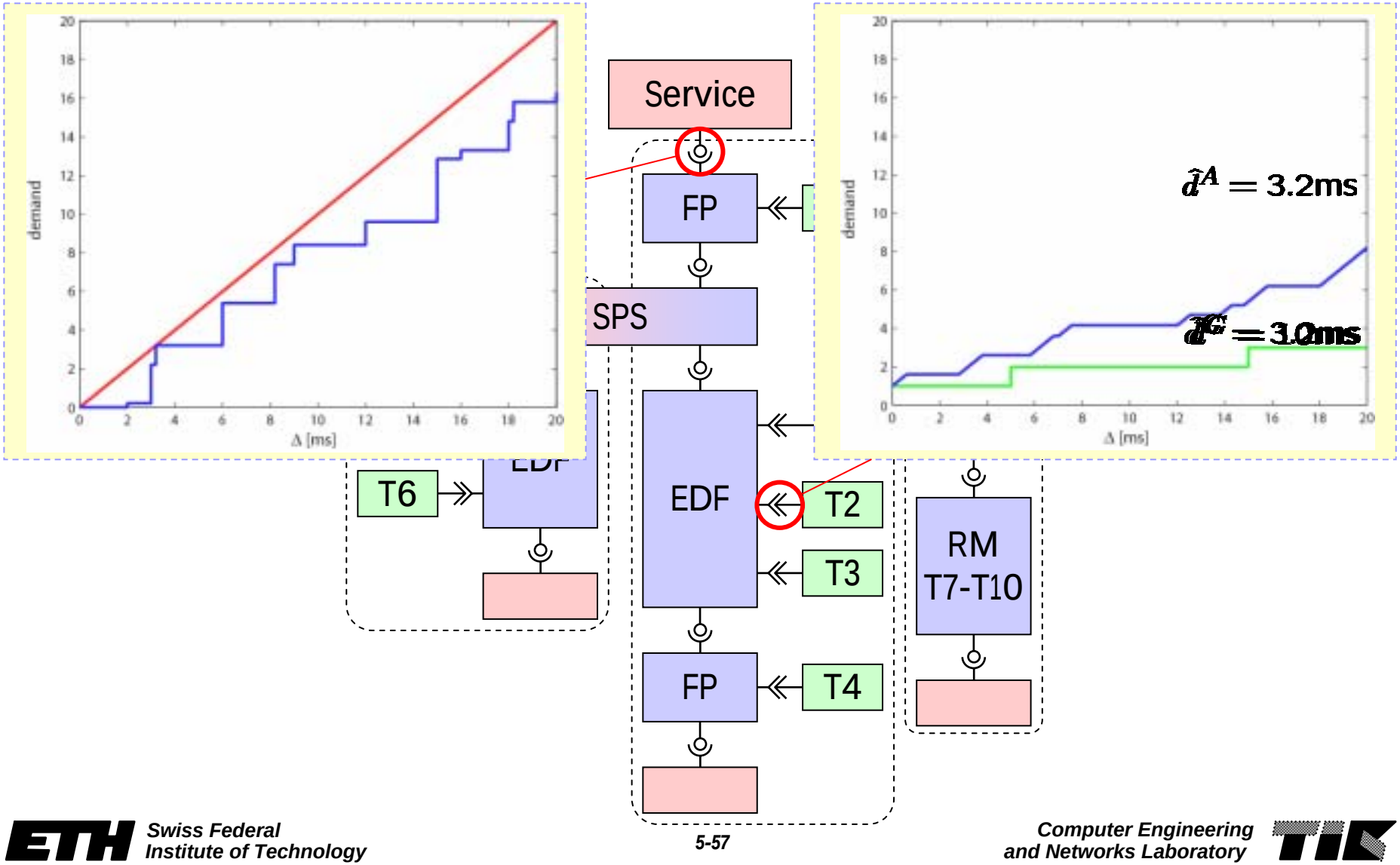




System Adaption I: Burstiness of T2



System Adaption II: Deadline of T2



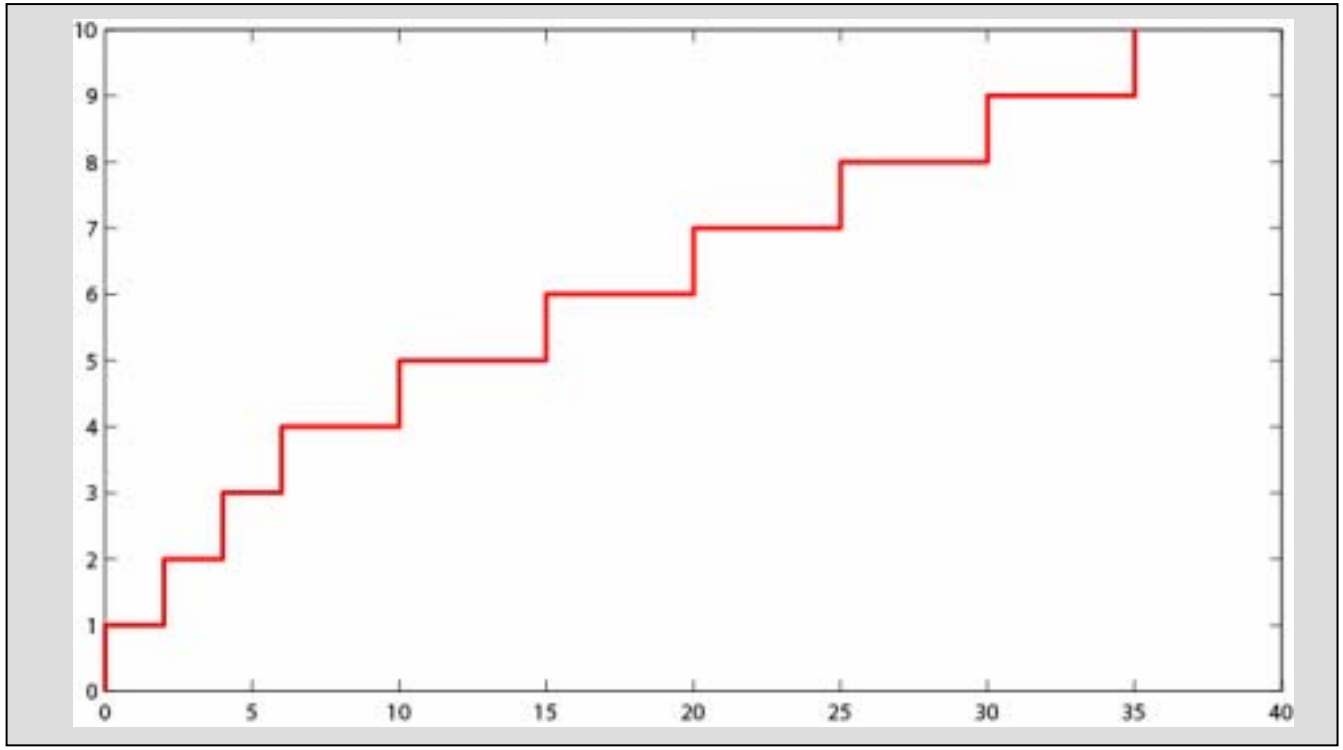
System Analysis Time

- < 1 second
  - Pentium Mobile 1.6 GHz
  - Matlab 7 SP2
  - RTC Toolbox

Outline

- Motivation / Problem Statement
- Modular Performance Analysis
- MPA Case Study
- Real-Time Interfaces / Interface-Based Design
- IBD Case Study
- Efficient Computation and Tool Support

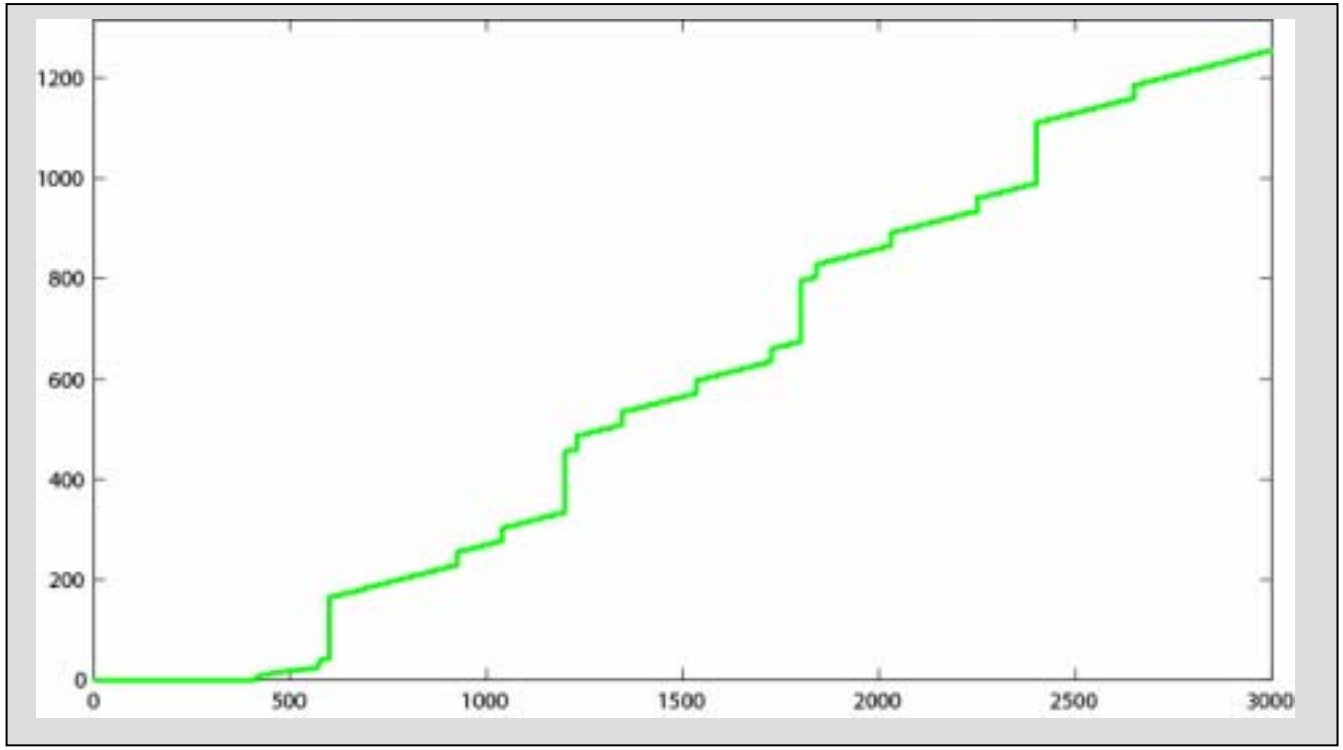
An Efficient Curve Representation



A curve is defined by:

- a ***start segment*** of arbitrary length & form
- a ***periodically repeated segment*** of arbitrary length & form

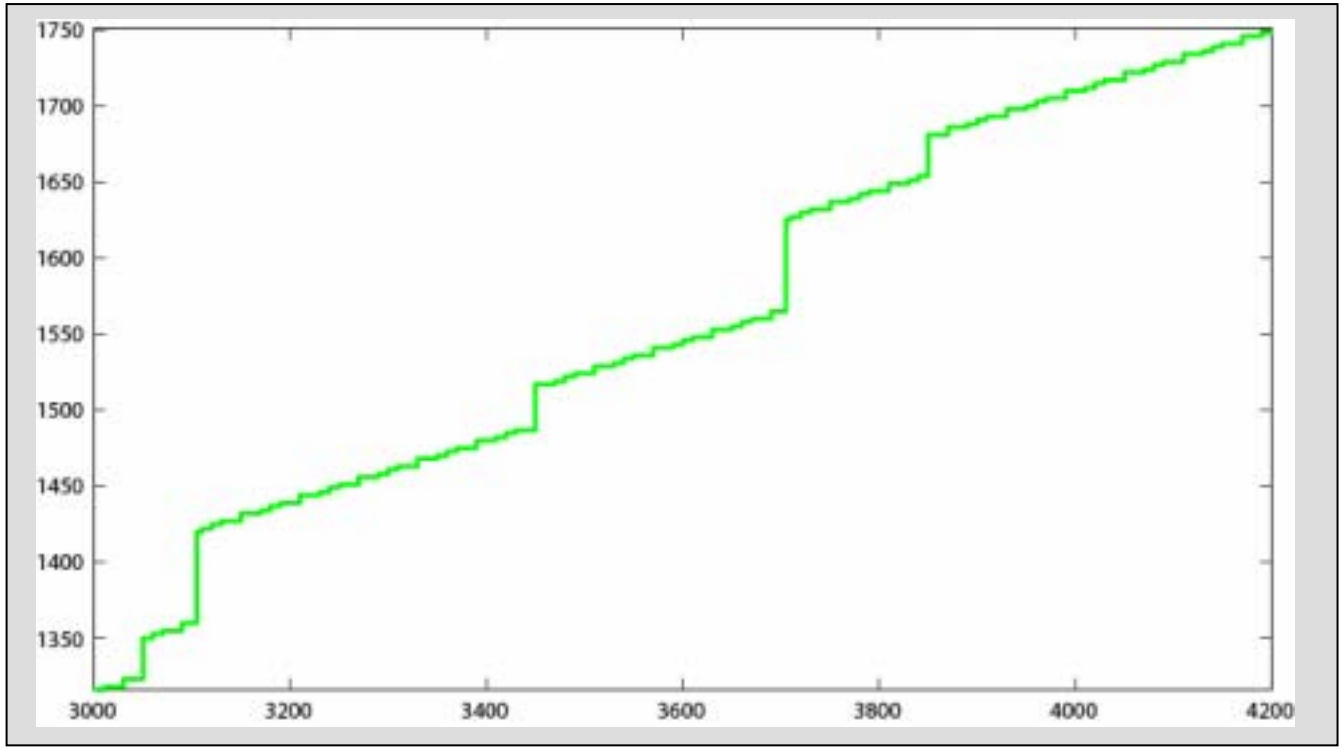
Complex Start Segment …



A curve is defined by:

- a ***start segment*** of arbitrary length & form
- a ***periodically repeated segment*** of arbitrary length & form

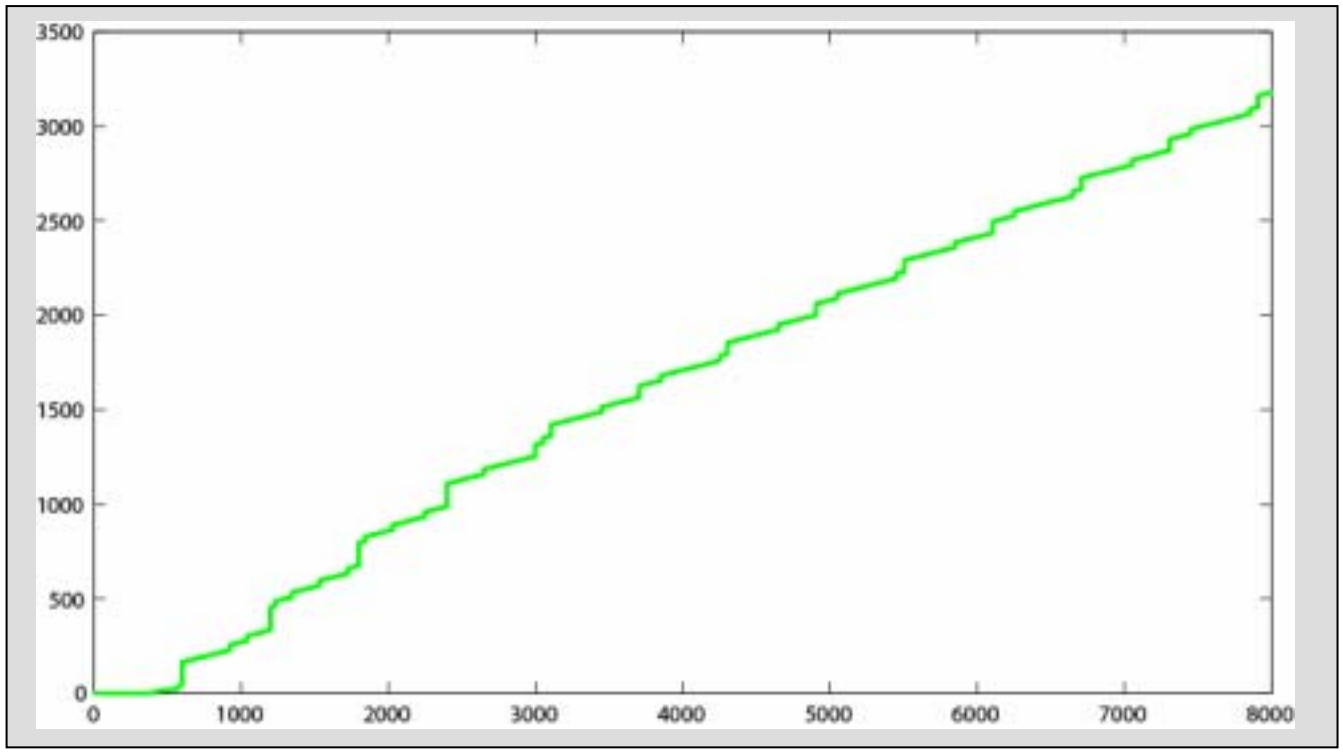
… + Complex Periodic Segment …



A curve is defined by:

- a ***start segment*** of arbitrary length & form
- a ***periodically repeated segment*** of arbitrary length & form

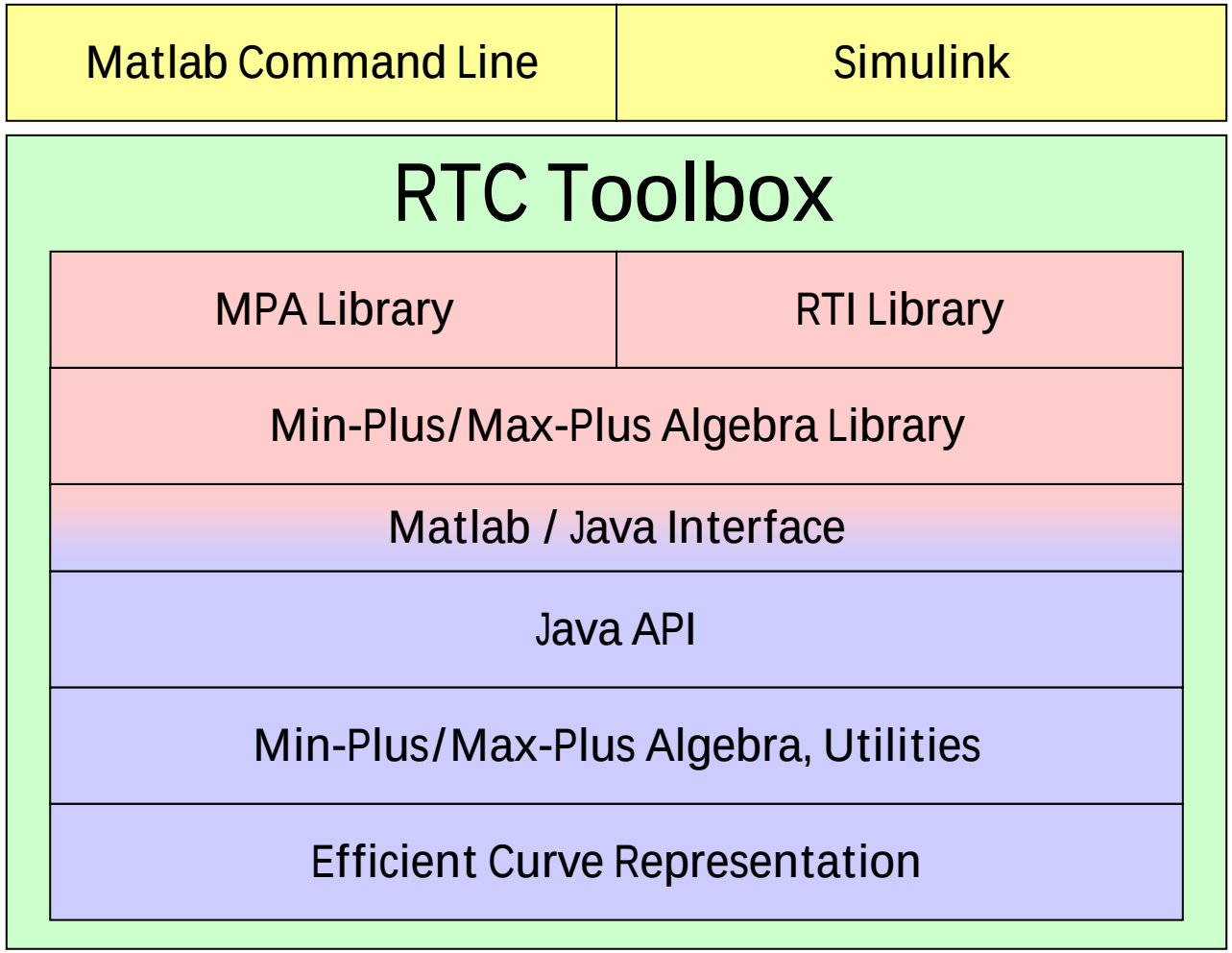
... = Complex Curve (EDF Service Demand of the IBD Case Study)

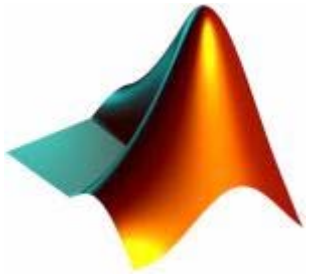


A curve is defined by:

- a ***start segment*** of arbitrary length & form
- a ***periodically repeated segment*** of arbitrary length & form

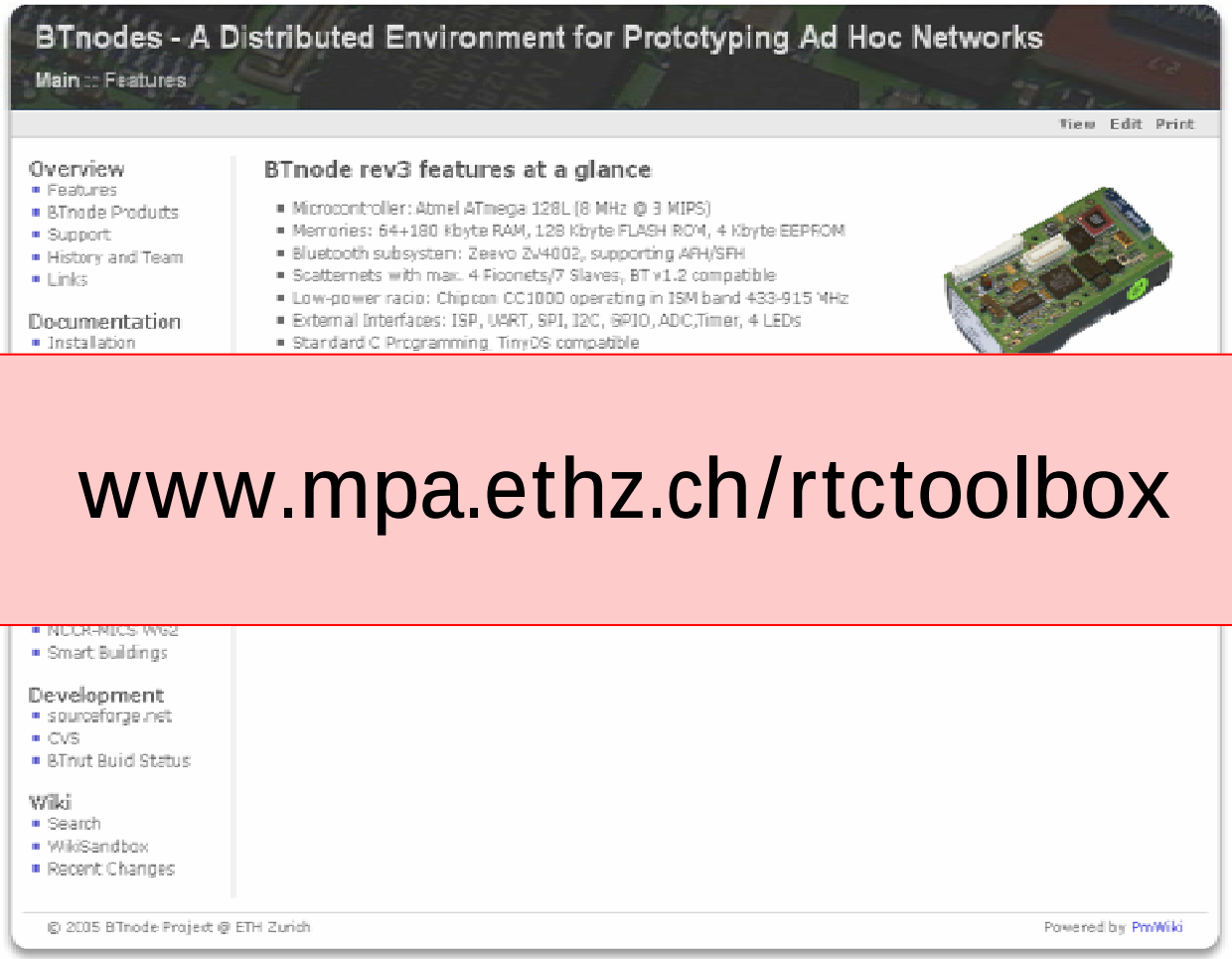
RTC Toolbox


Matlab Command Line
Simulink
RTC Toolbox
MPA Library
RTI Library
Min-Plus/Max-Plus Algebra Library
Matlab / Java Interface
Java API
Min-Plus/Max-Plus Algebra, Utilities
Efficient Curve Representation



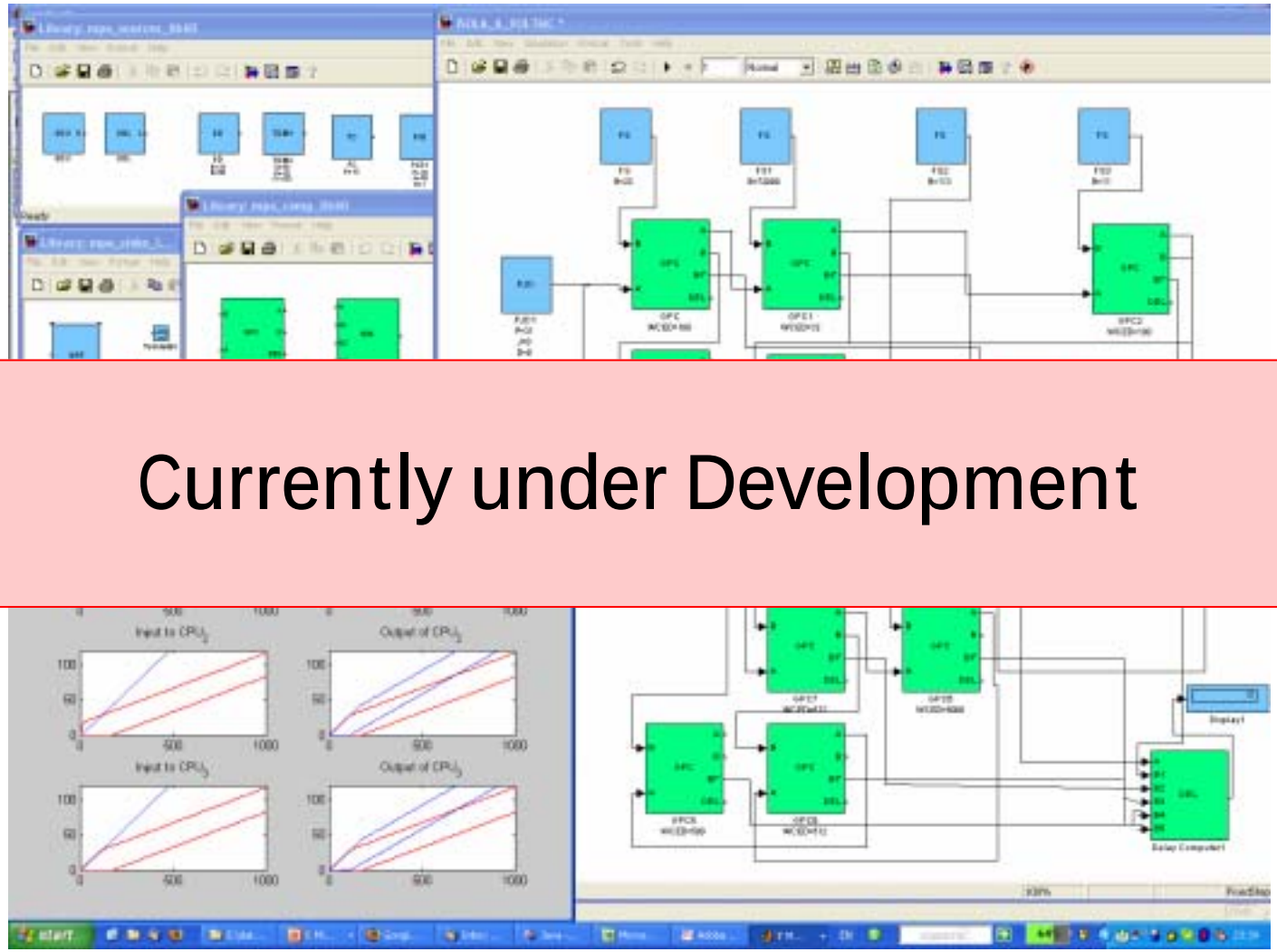

Java

RTC Toolbox: Version 0.9 Released



www.mpa.ethz.ch/rtctoolbox

RTC Toolbox: Simulink Frontend



Currently under Development

Acknowledgement

- Collaborators:
  - Ernesto Wandeler
  - Samarjit Chakraborty
  - Simon Künzli
  - Alexander Maxiaguine
- Funding:
  - SNF, KTI, MEDEA+/SPEAC, ARTIST2 NoE