Modular Performance Analysis

Lothar Thiele

Simon Perathoner, Ernesto Wandeler

ETH Zurich, Switzerland

Embedded Systems

Computation/Communication ⊕
Resource Interaction

Models of Computation

How can we classify and compare them?

stepwise refinement

concurrency hierarchy incremental design

beauty modular simple safe

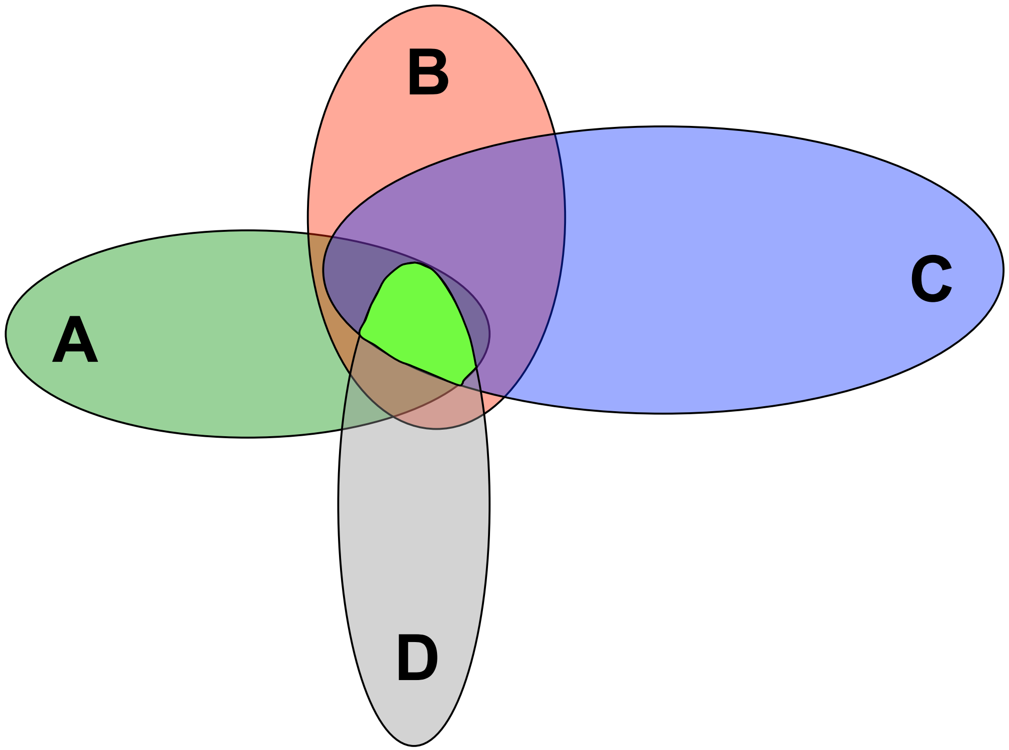
accuracy expressive tools formal

compositional easy to use efficient

executable scope scalable implementation

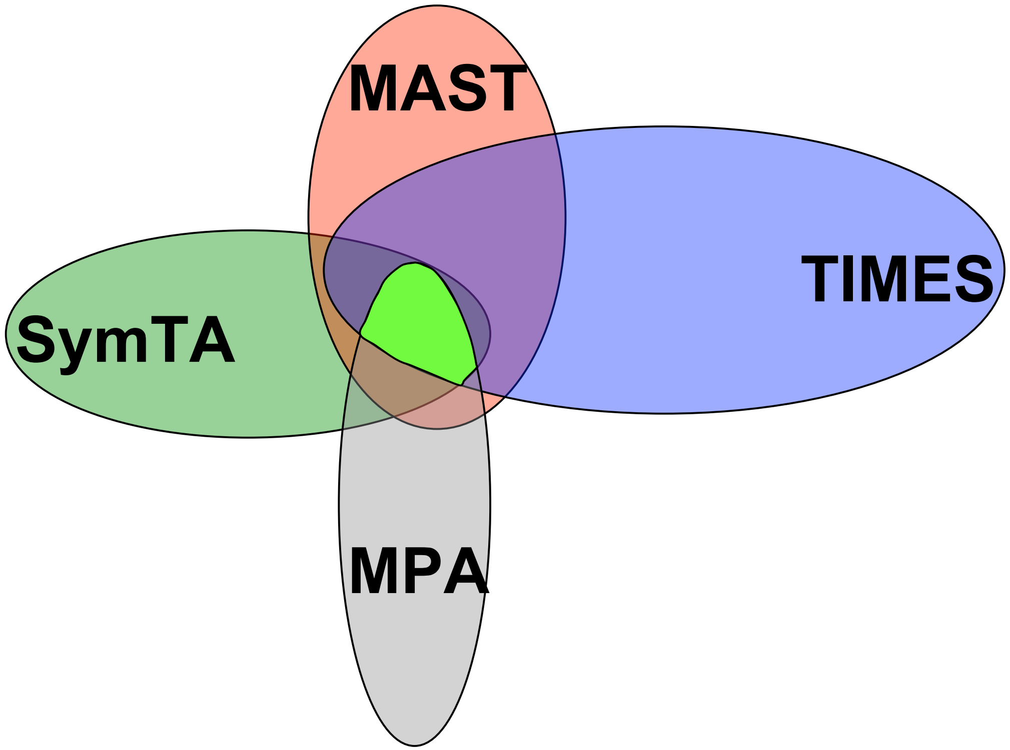
Why is it difficult ?

- Many aspects can not be quantified.
- Models cover different scenarios:



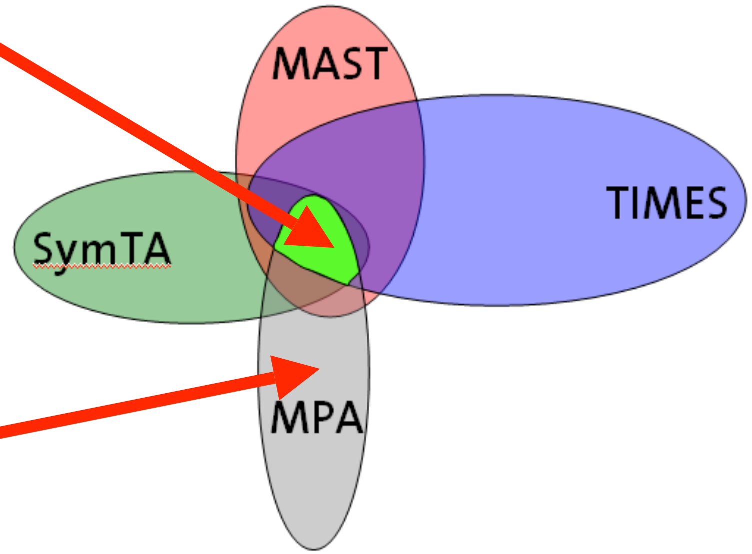
Intention

- Compare models and methods that analyze the timing properties of distributed systems.



Approach

- Define a set of benchmark examples that cover common area (***obligatory***)



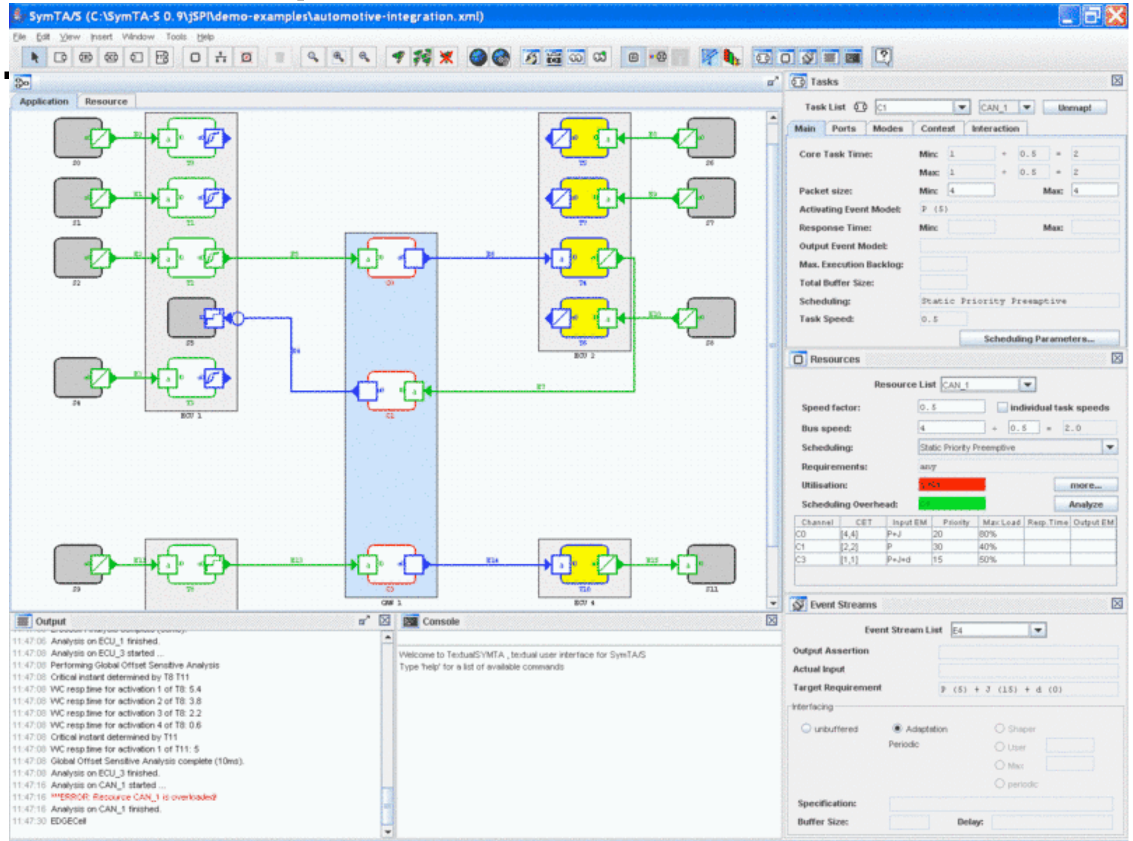
- Define benchmark examples that show the power of each method (***free style***)
- Leiden Workshop on Distributed Embedded Systems: http://www.tik.ee.ethz.ch/~leiden05/

MAST

Wait another 20 minutes ...

SymTA/S

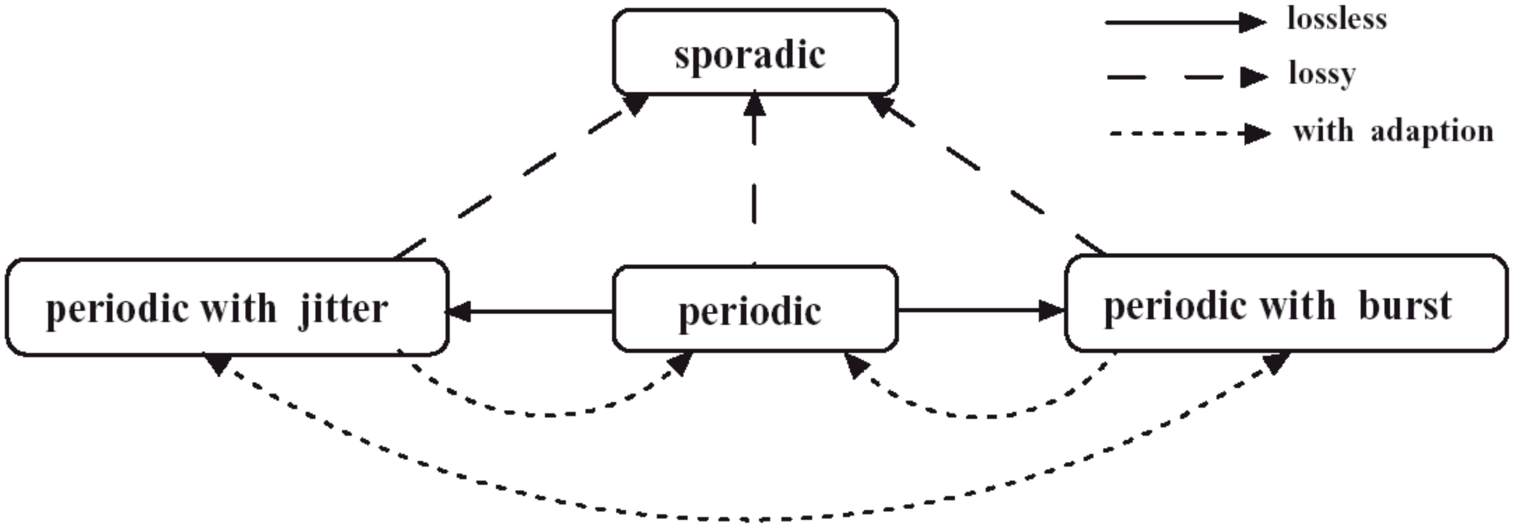
- Based on classical RT analysis (periodic, jitter).
- Simplified relations and adaptors in order to achieve modularity.
- Computation and Communication



SYMTA VISION

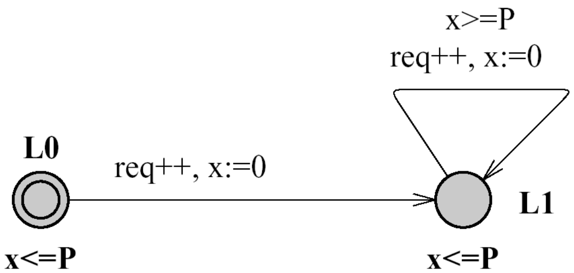
SymTA/S

- Based on classical RT analysis (periodic, jitter).
- Simplified relations and adaptors in order to achieve modularity.



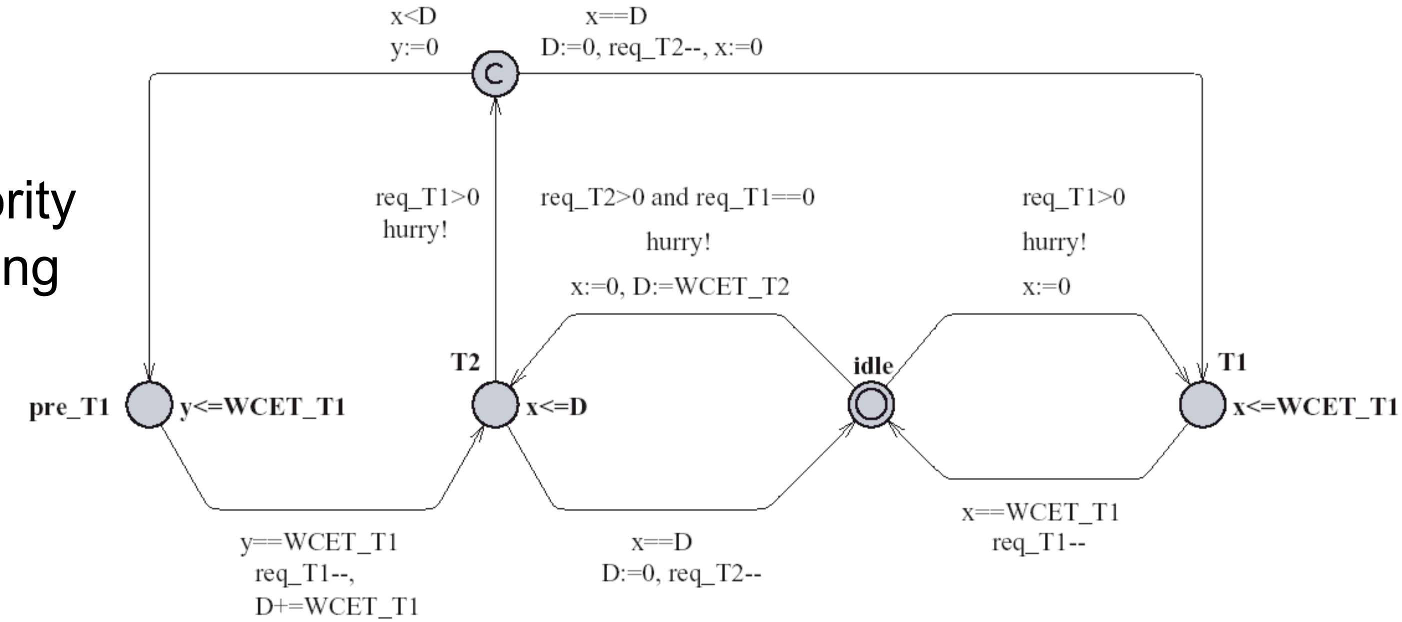
TIMES/UPPAAL

- Models are based on Timed Automata.

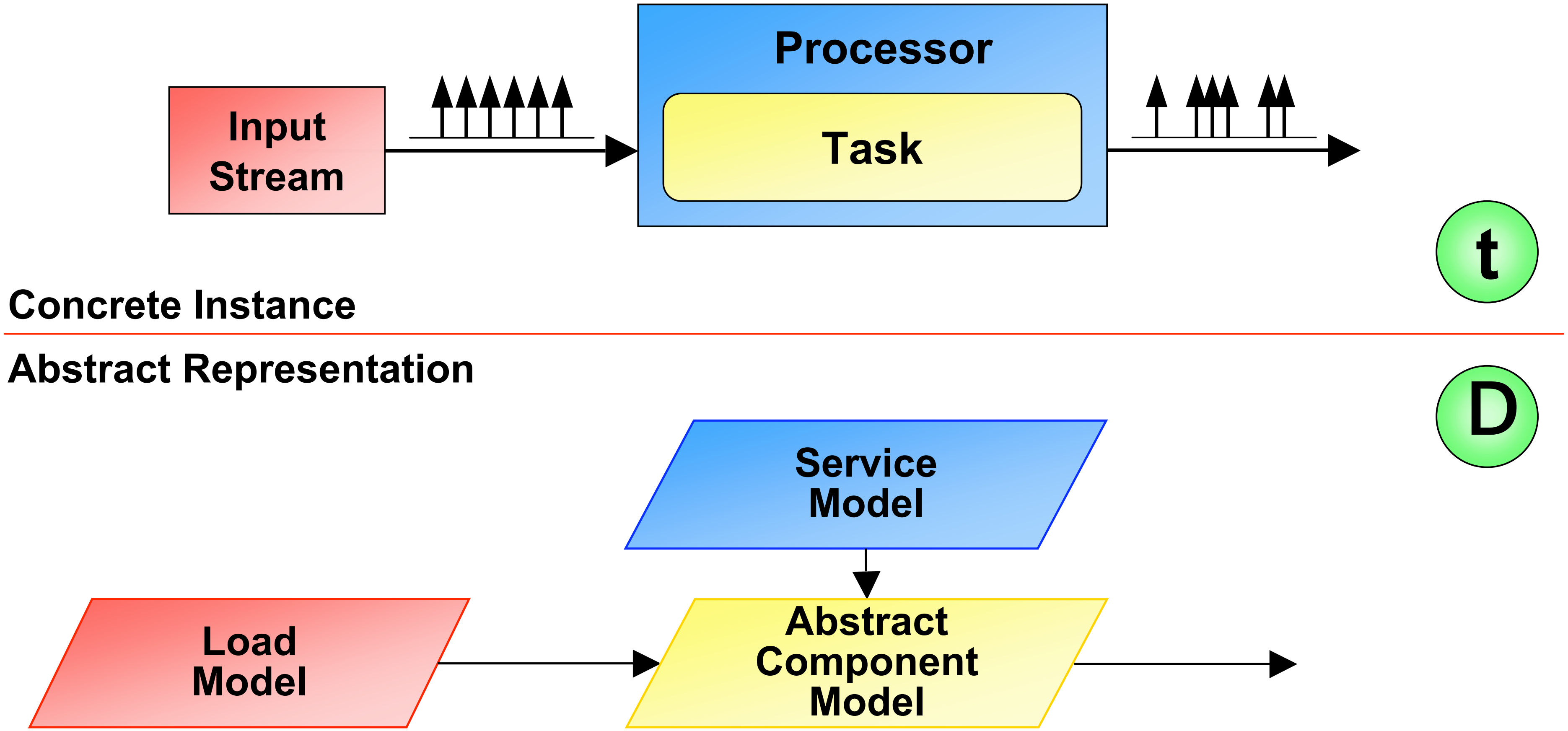


periodic stream

fixed priority scheduling



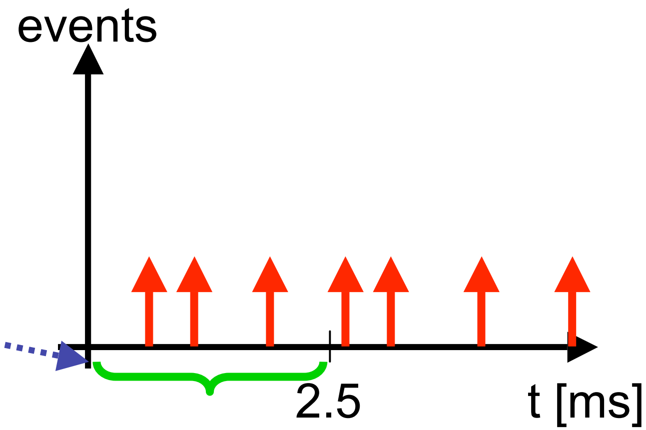
Modular Performance Analysis (MPA)



Abstract Stream Model

Event Stream

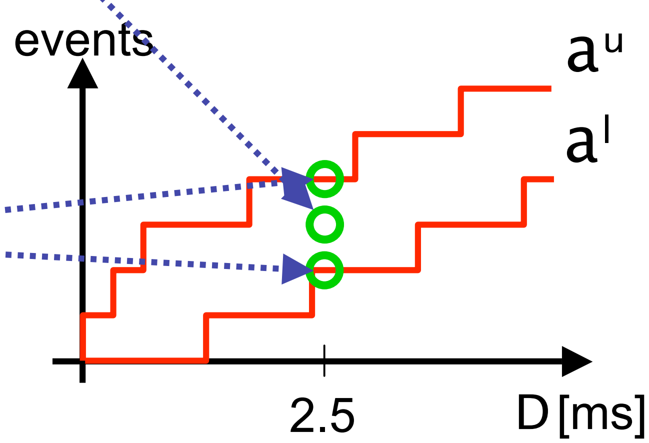
number of events in
in t=[0 .. 2.5] ms



t

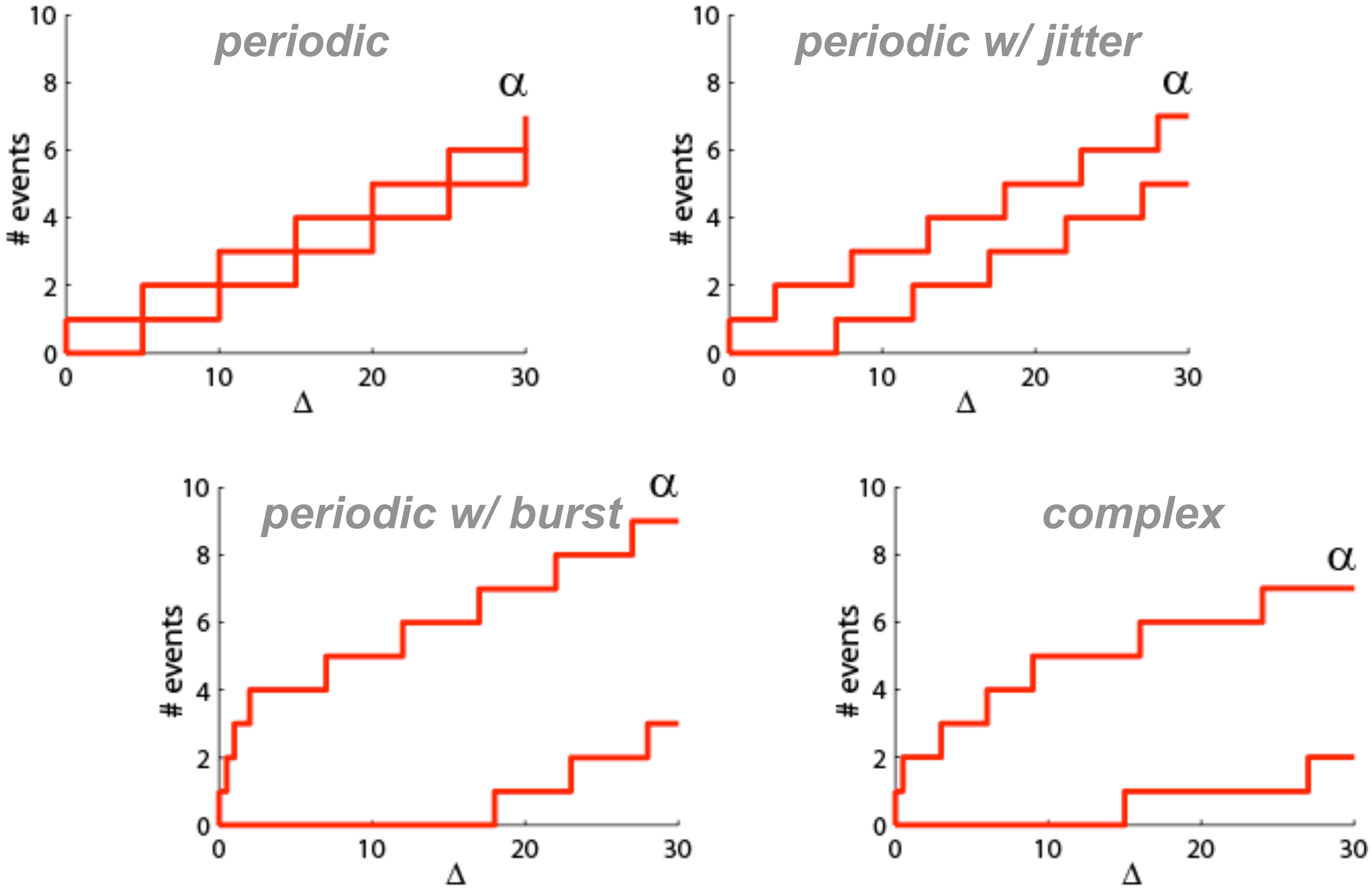
Arrival Curves

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



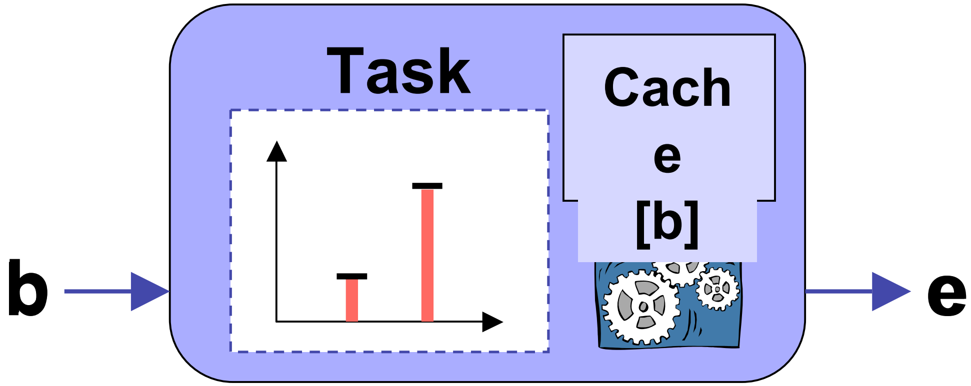
D

Load Model - Examples

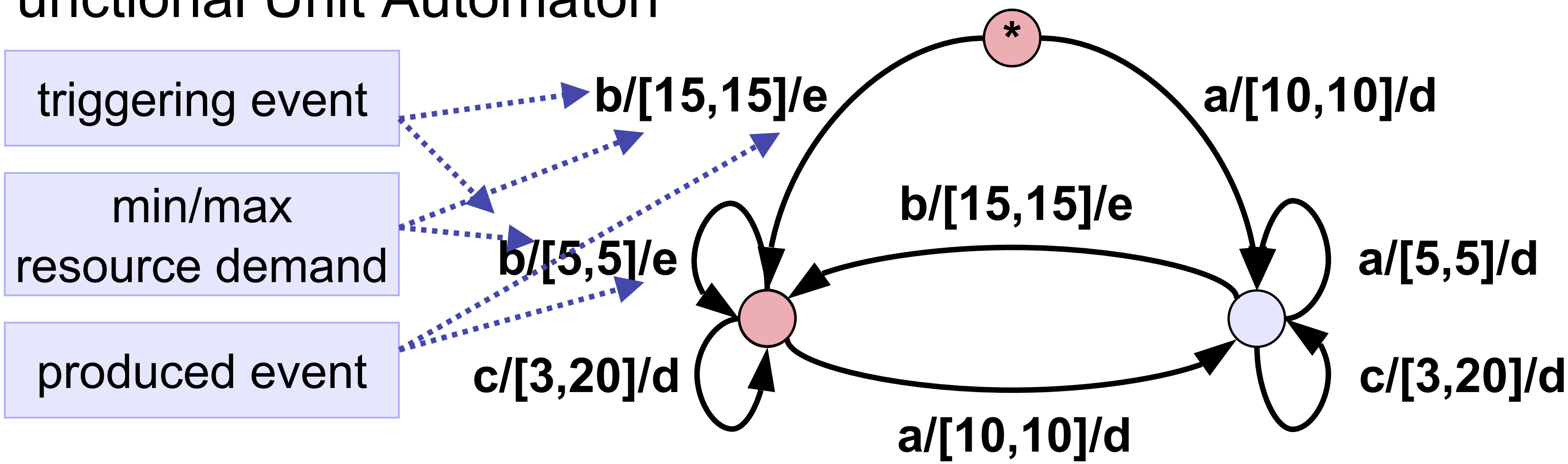


Process Abstraction

Formal Specification
Program Analysis
Data Sheets
...



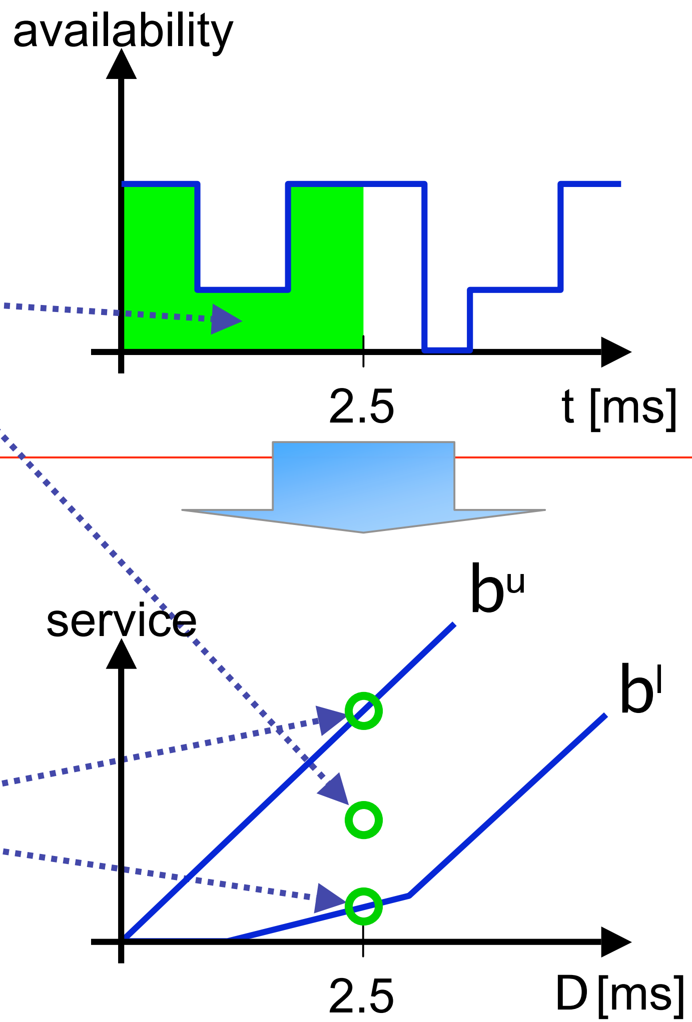
Functional Unit Automaton



Service Model (Resources)

Resource Availability

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



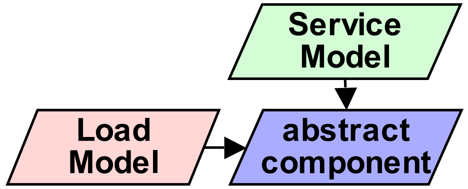
t

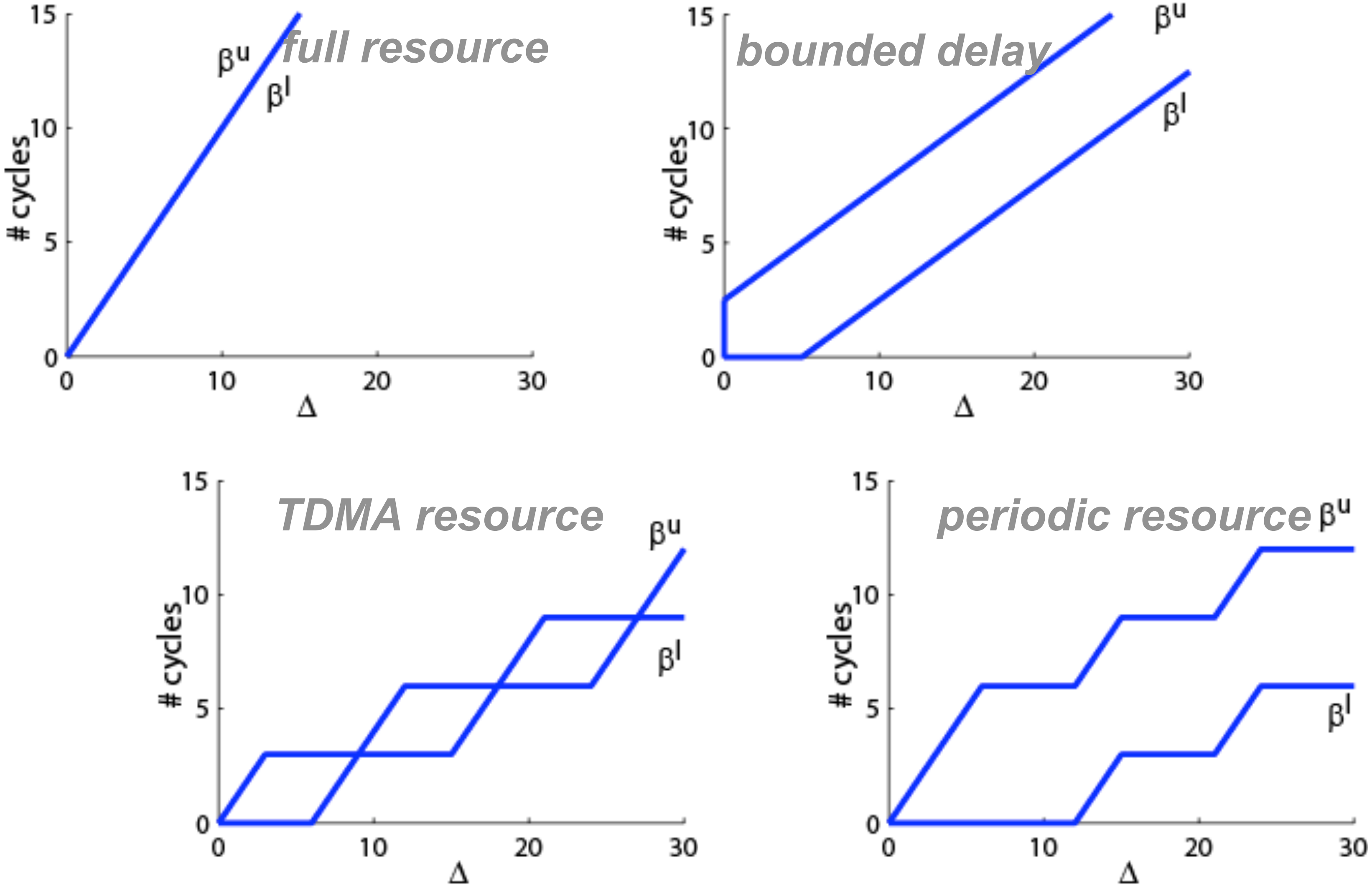
Service Curves

D

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

Service Model - Examples





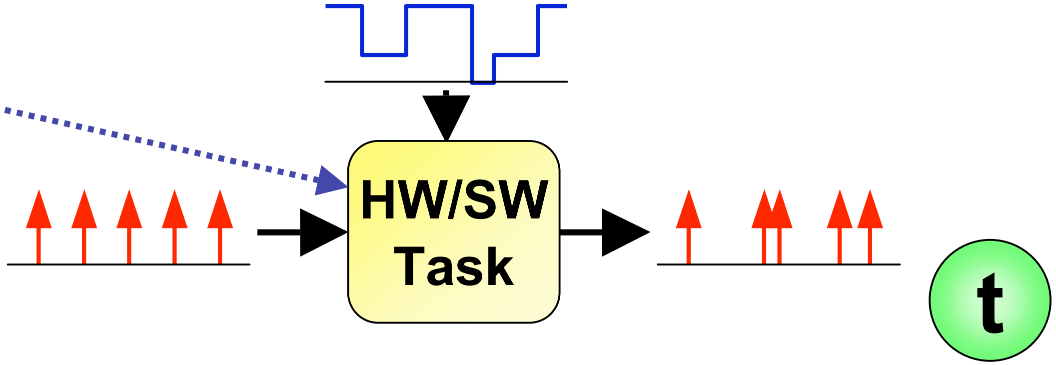
What kind of resources can be modeled?

- Memory (buffer space)
- Delay (end-to-end delay / processing and waiting)
- Computation
- Communication
- Energy

Processing Model (HW/SW)

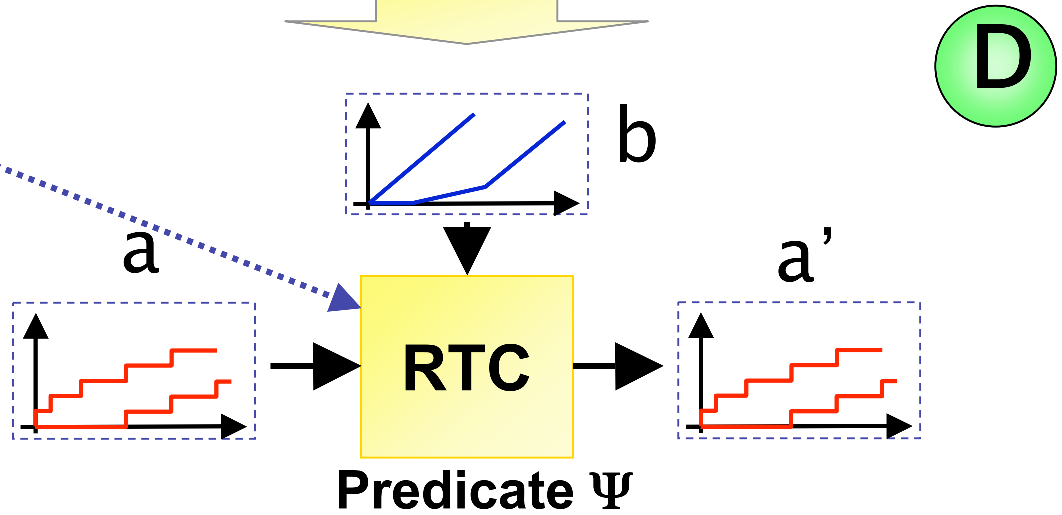
HW/SW Components

Processing semantics and functionality of HW/SW tasks

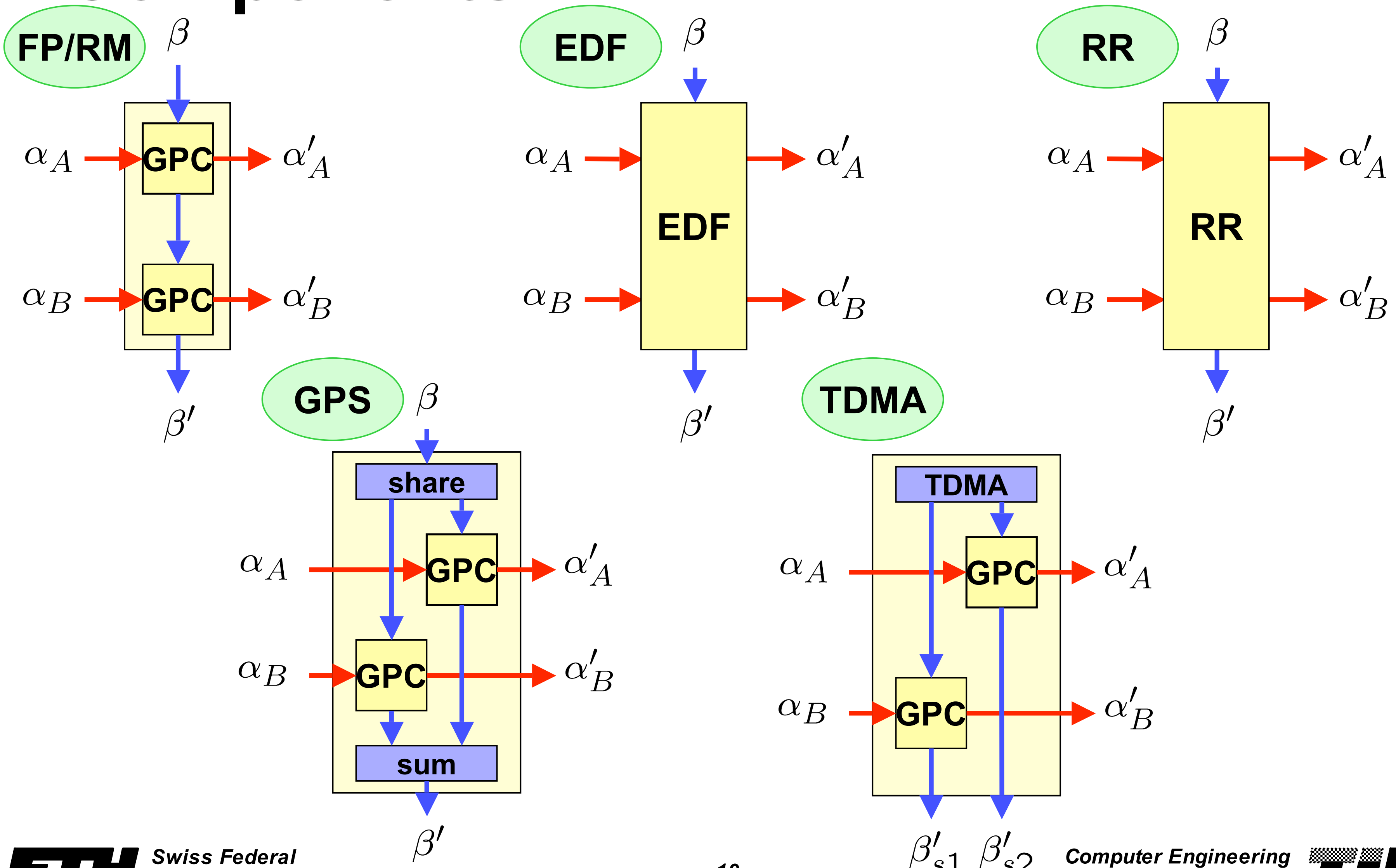


Abstract Components

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



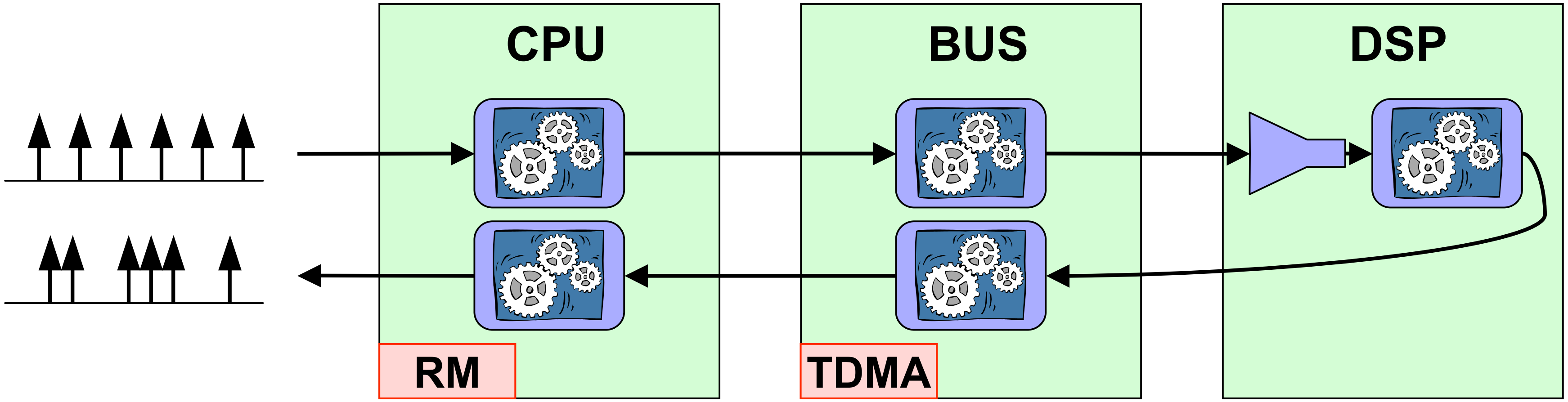
Scheduling and Arbitration Components

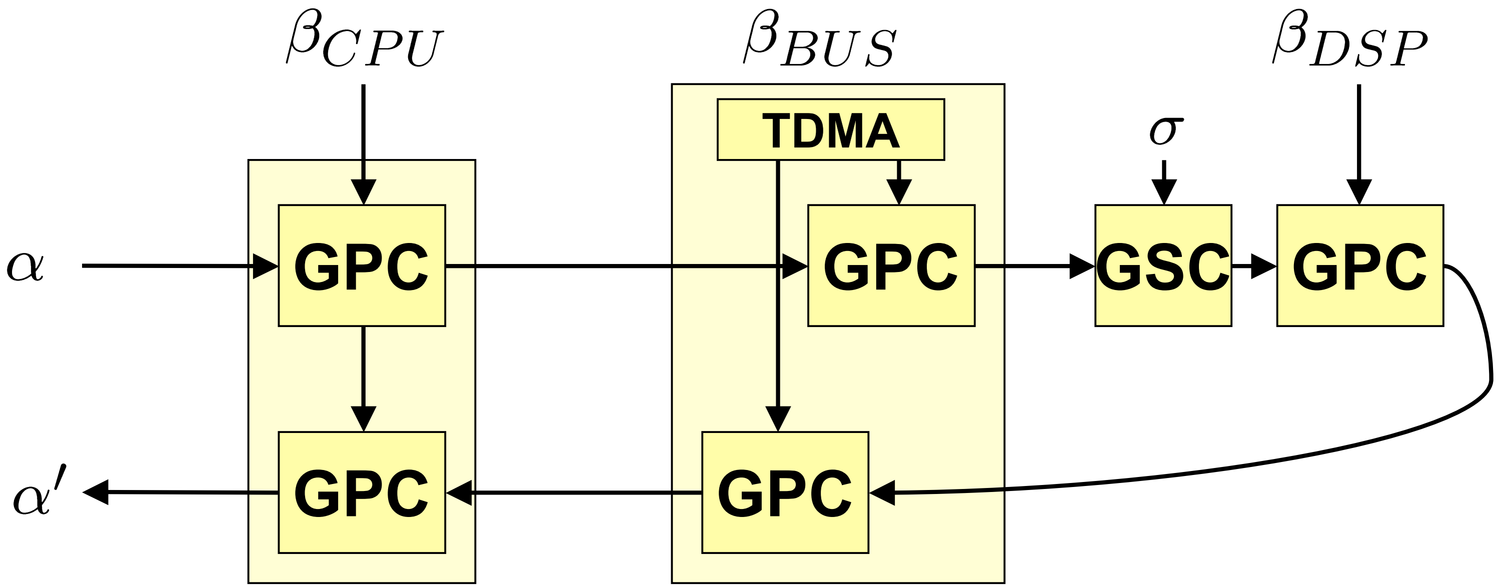


What kind of resource usage can be modeled?

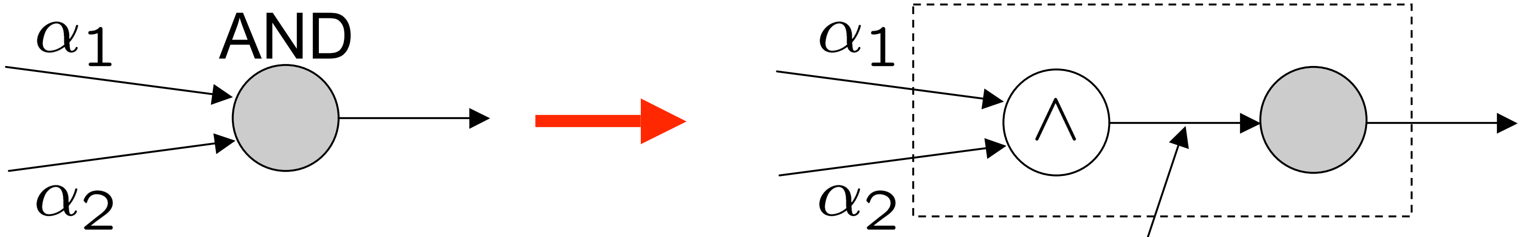
- Different resource sharing strategies
  - EDF
  - TDMA
  - Fixed Priority
  - GPS
- Different processing semantics
  - Greedy Processing
  - Greedy Shaper
  - Blocking

Complete System Composition

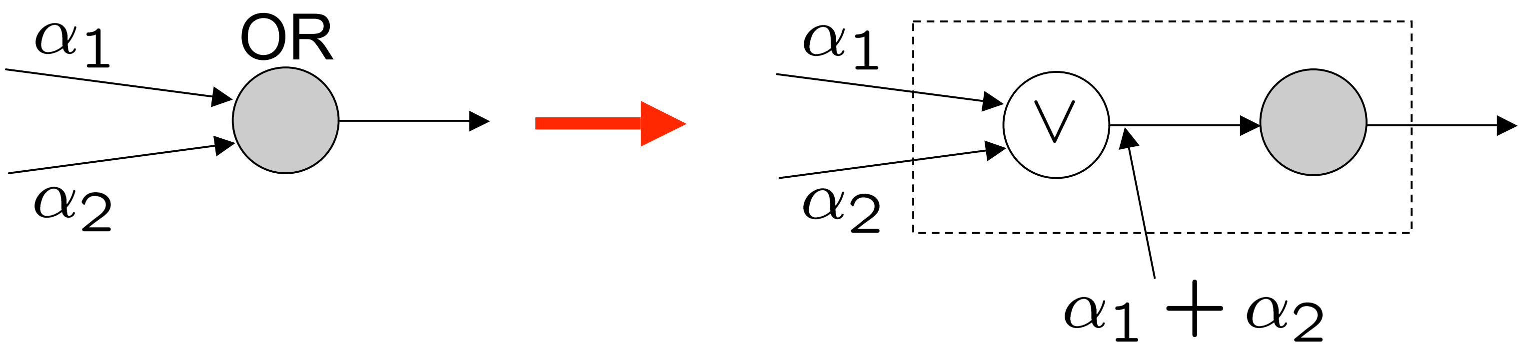




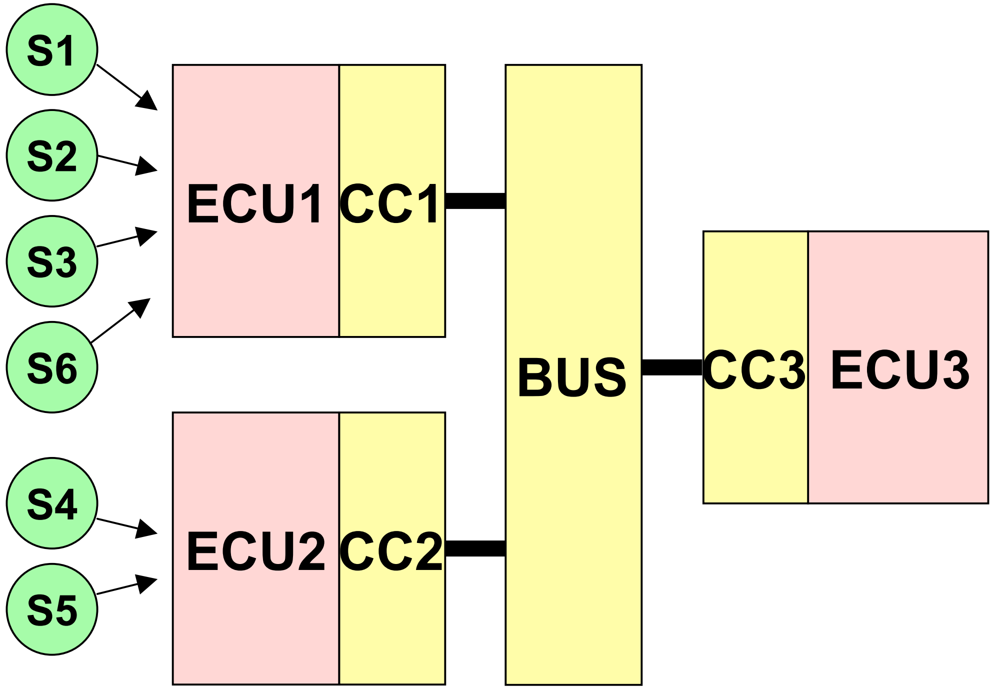
Basic Concepts for Describing Activations



$$\max\left\{\min\left\{\alpha_1^u \oslash \alpha_2^l + B_1^0 - B_2^0, \alpha_2^u\right\},\right.$$
$$\left.\min\left\{\alpha_2^u \oslash \alpha_1^l + B_2^0 - B_1^0, \alpha_1^u\right\}\right\}$$



Free Style



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

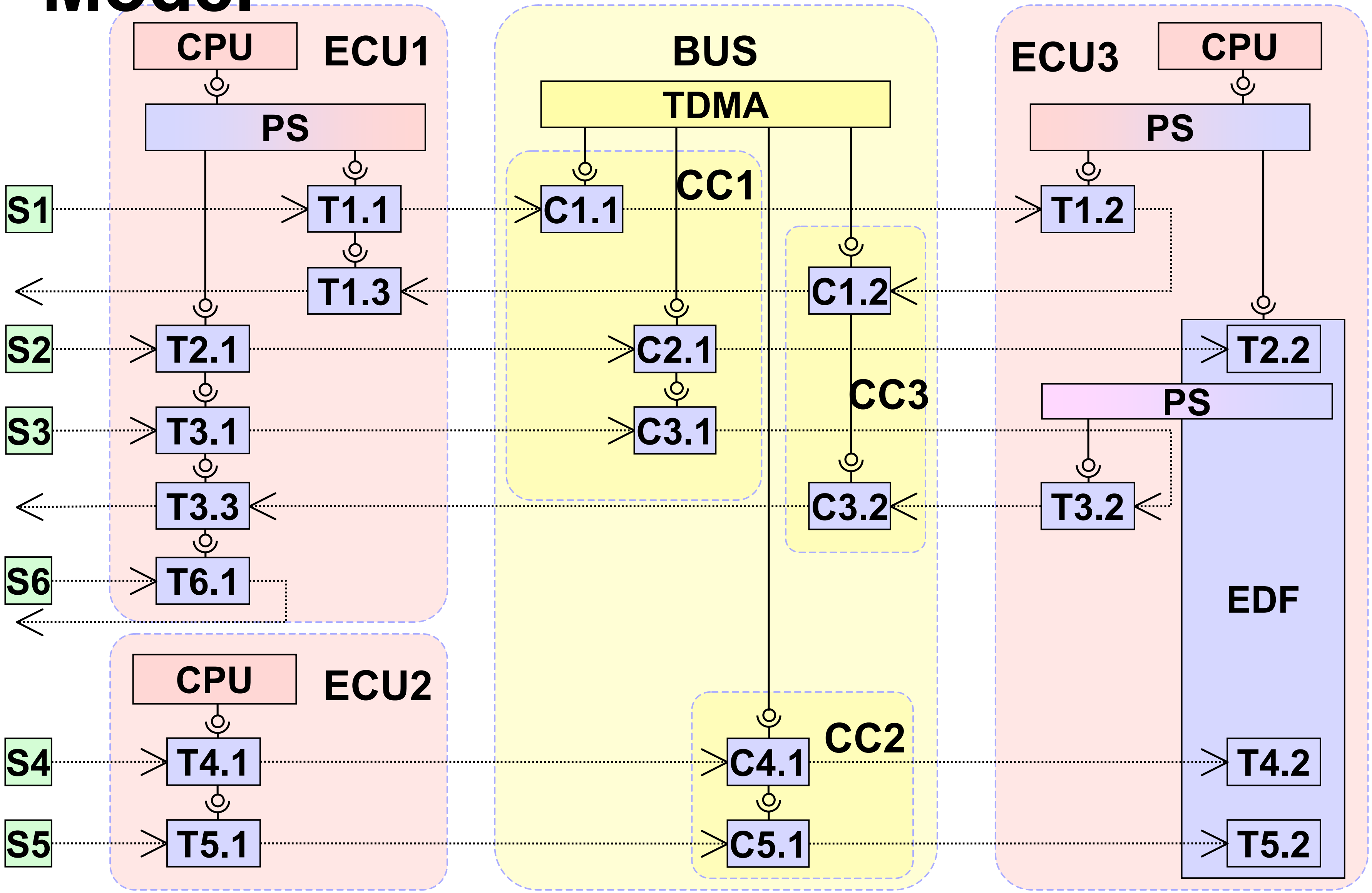
13 Tasks & 7 Messages
- with different WCED

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

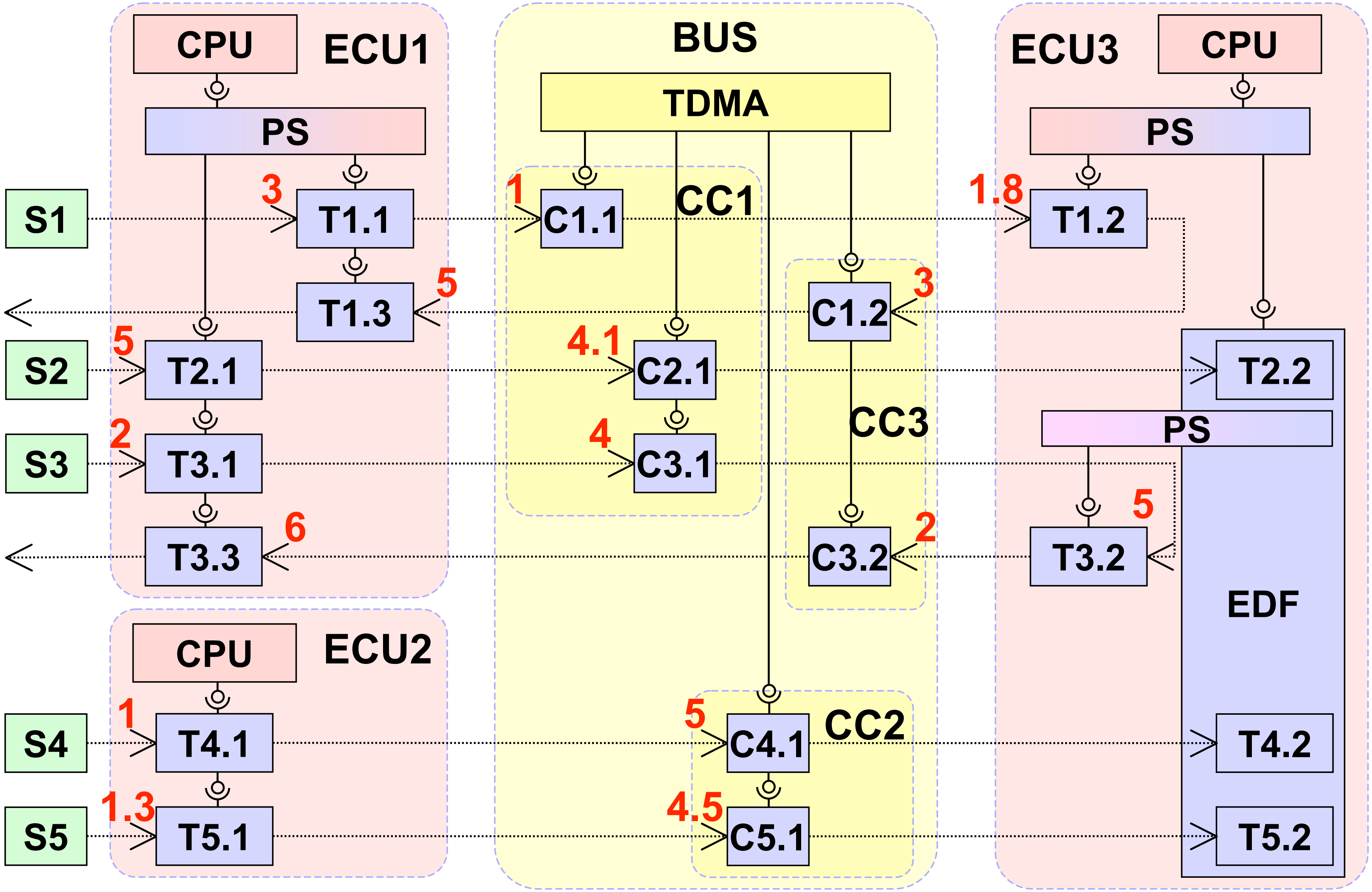
Hierarchical Scheduling
- Static & Dynamic Polling Servers

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

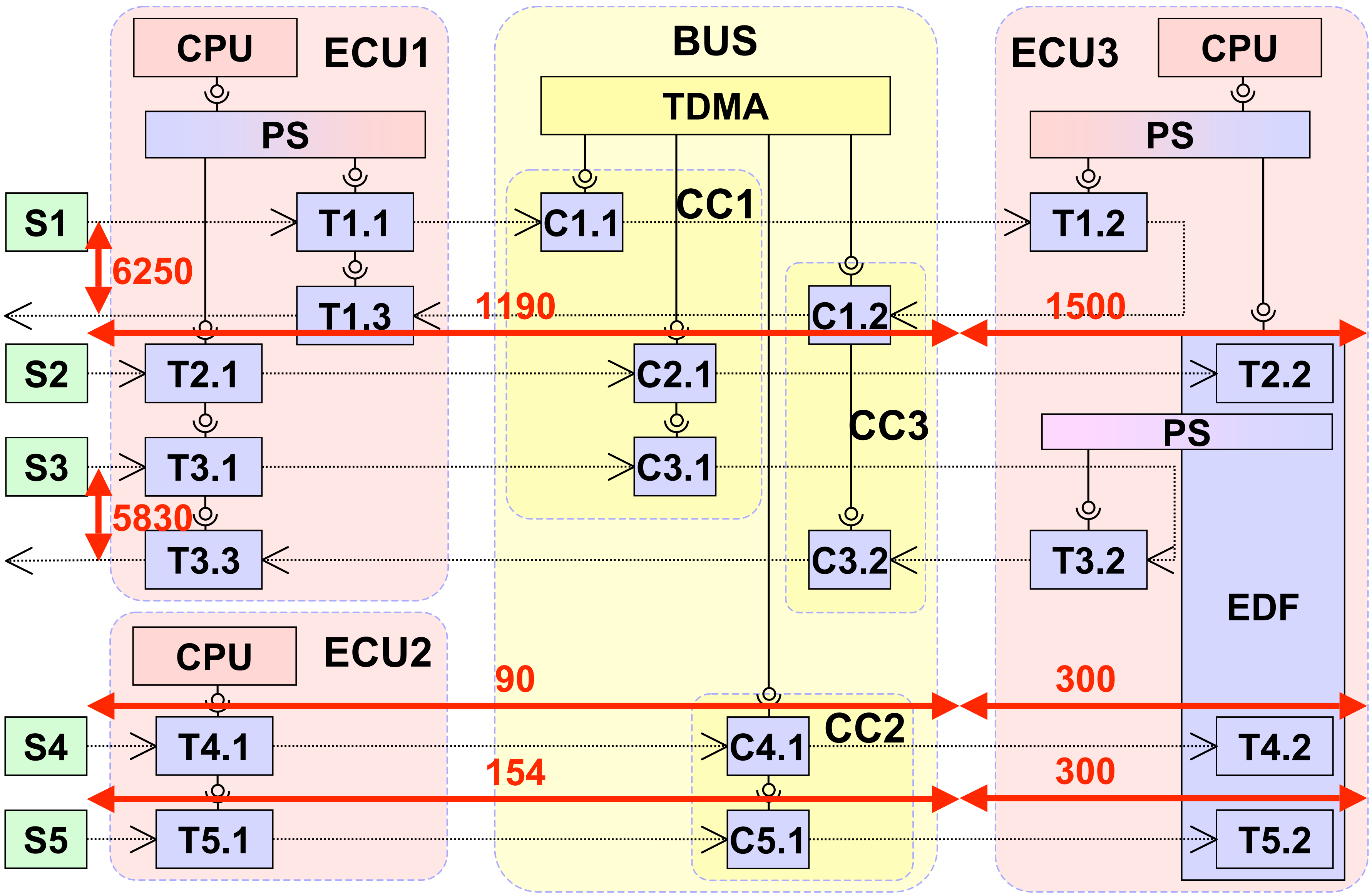
... and its Abstract Component Model



Buffer Requirements

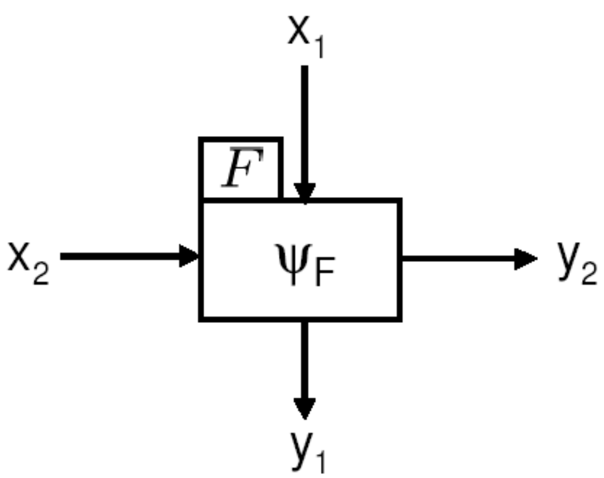


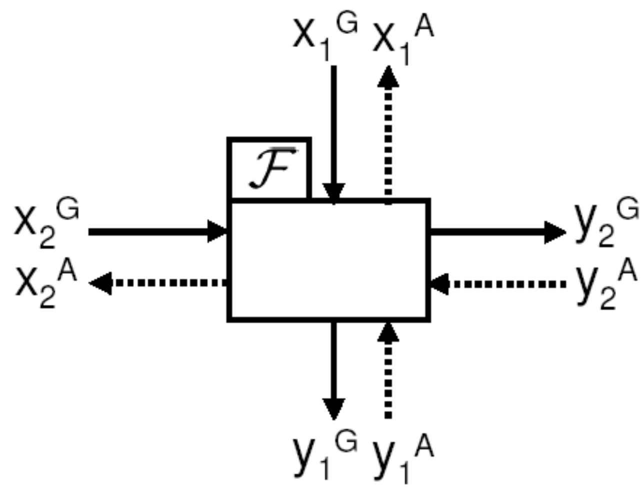
Delay Guarantees



Interface-Based Design

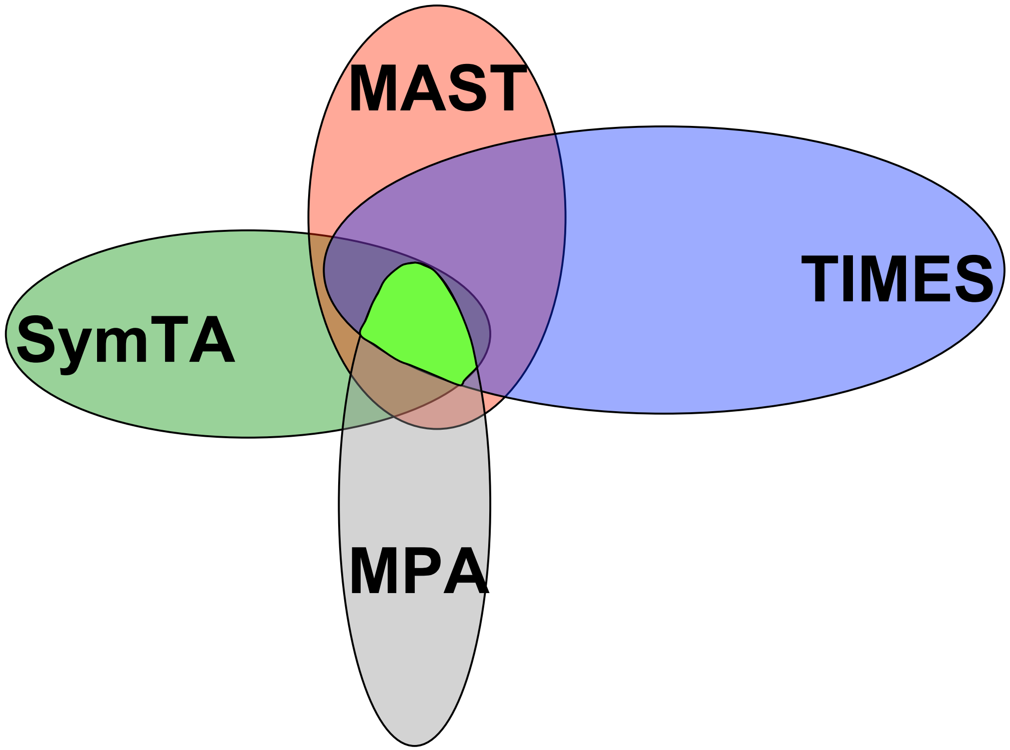
- MPA is suited for interface-based design
  - Stepwise refinement
  - Inverse relations because of min-+ algebra
  - Assume/Guarantee by means of partial order





Intention

- Compare models and methods that analyze the timing properties of distributed systems.

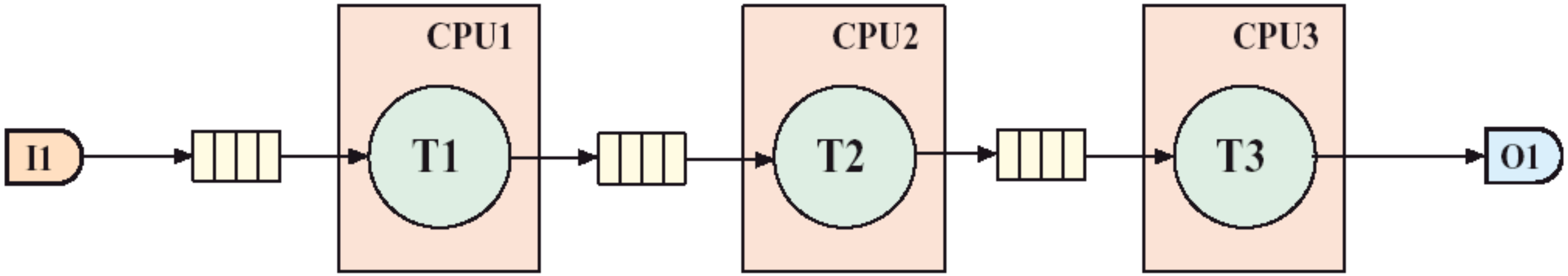


Benchmarks

- **Pay Burst Only Once**
- **Cyclic Dependencies**
- **Variable Feedback**
- AND/OR task activation
- Intra-context information
- Workload Correlation
- Data Dependencies

Benchmark 1

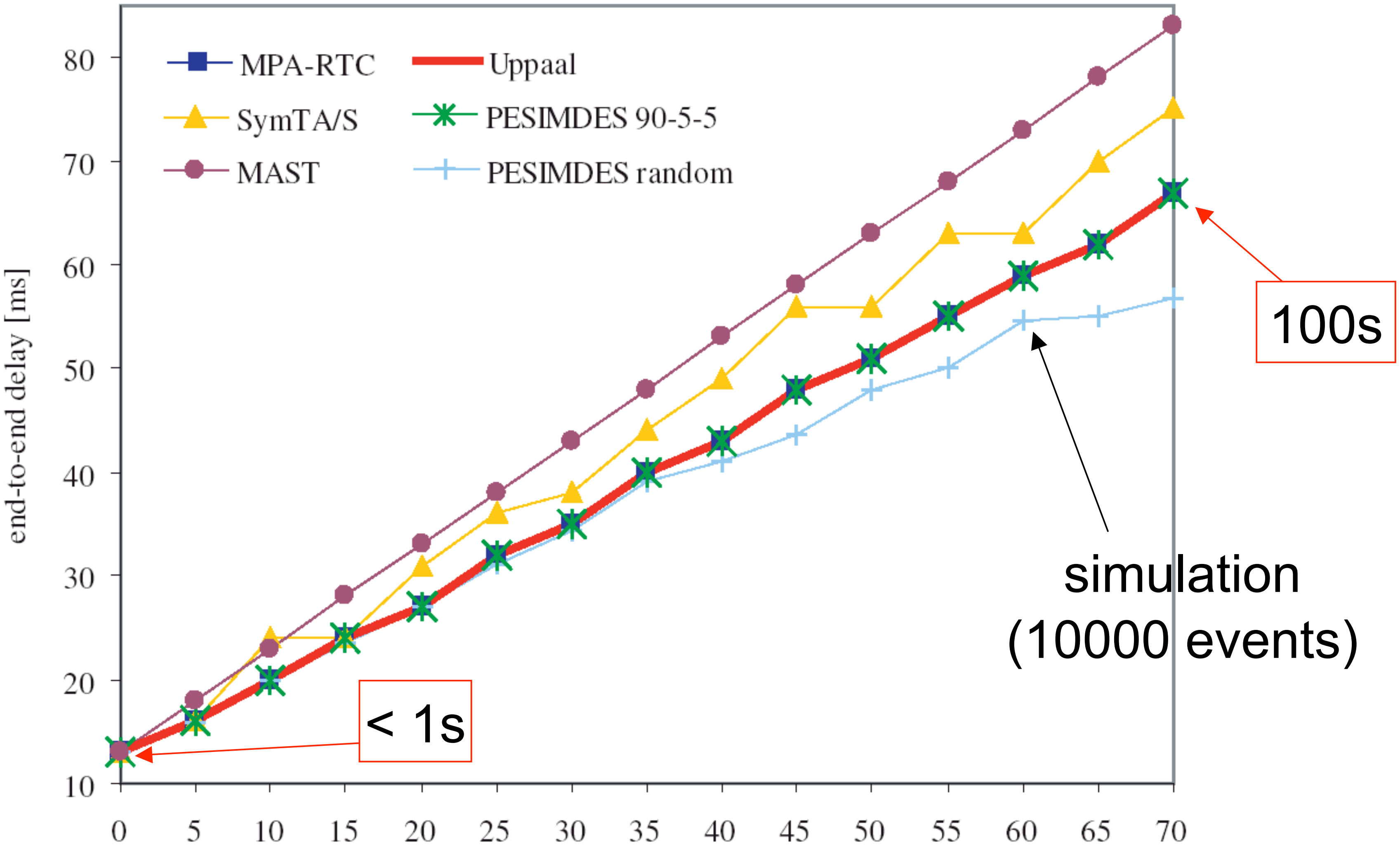
- Pay Bursts Only Once



Input stream I1	periodic with burst (P=10ms, J=50ms, d=1ms)
Task WCETs	T1: 1ms, T2: 4ms, T3: 8ms

$0 \leq J \leq 70ms$

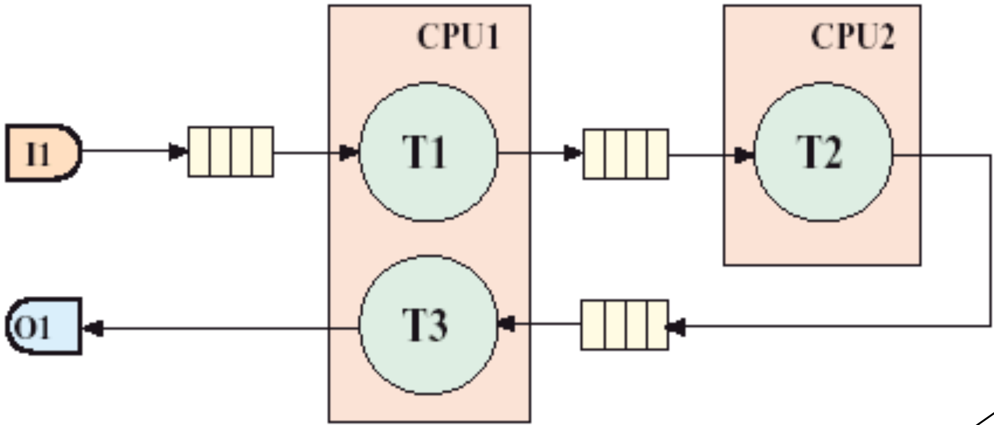
Benchmark 1



MPA-RTC
Uppaal
SymTA/S
PESIMDES 90-5-5
MAST
PESIMDES random
end-to-end delay [ms]
80
70
60
50
40
30
20
10
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
100s
simulation
(10000 events)
< 1s

Benchmark 2

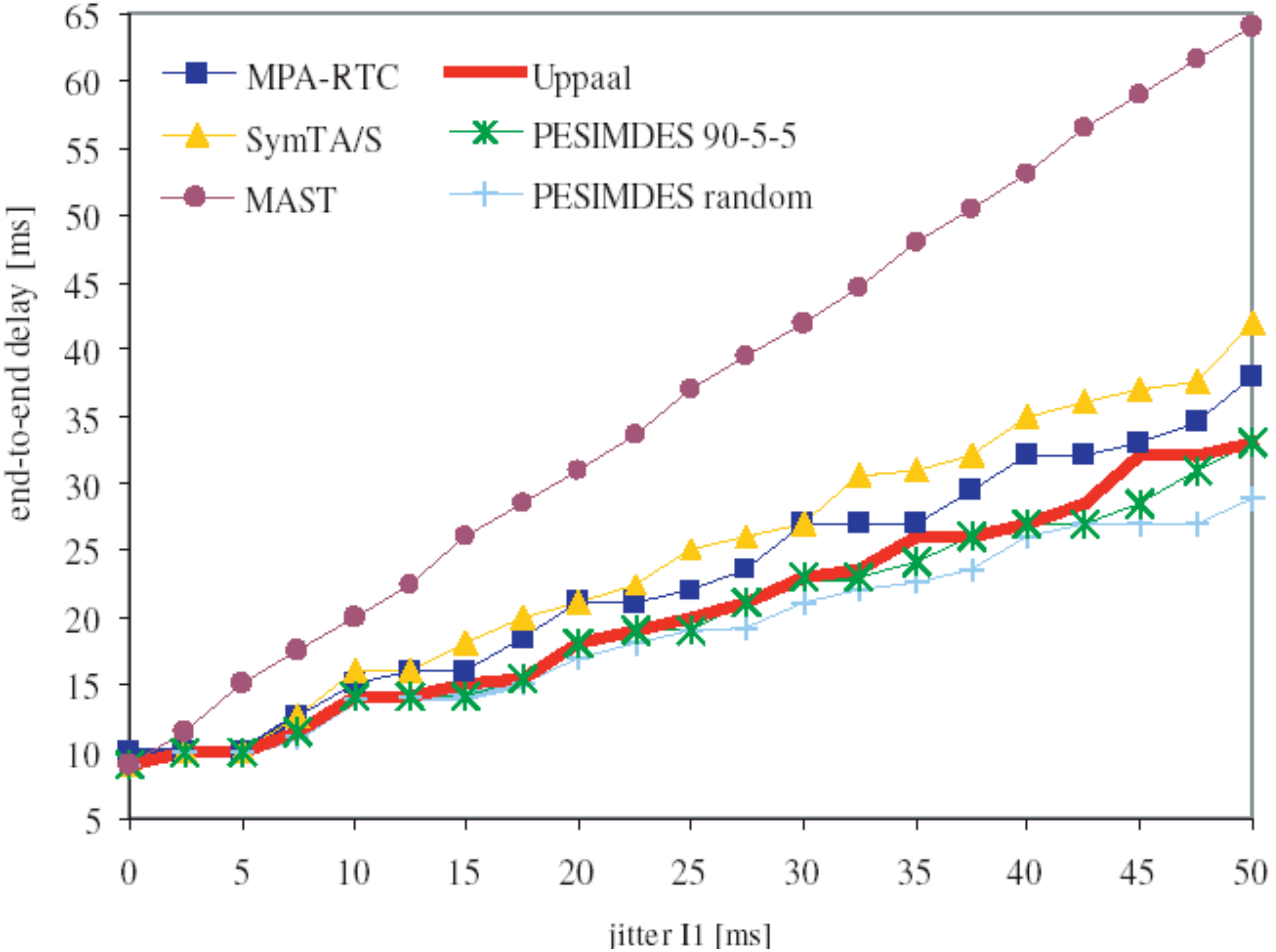
- Cyclic Dependencies

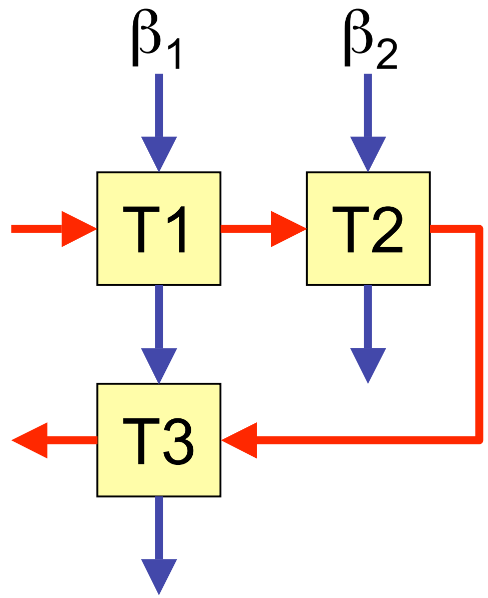


0 ≤ J ≤ 50ms

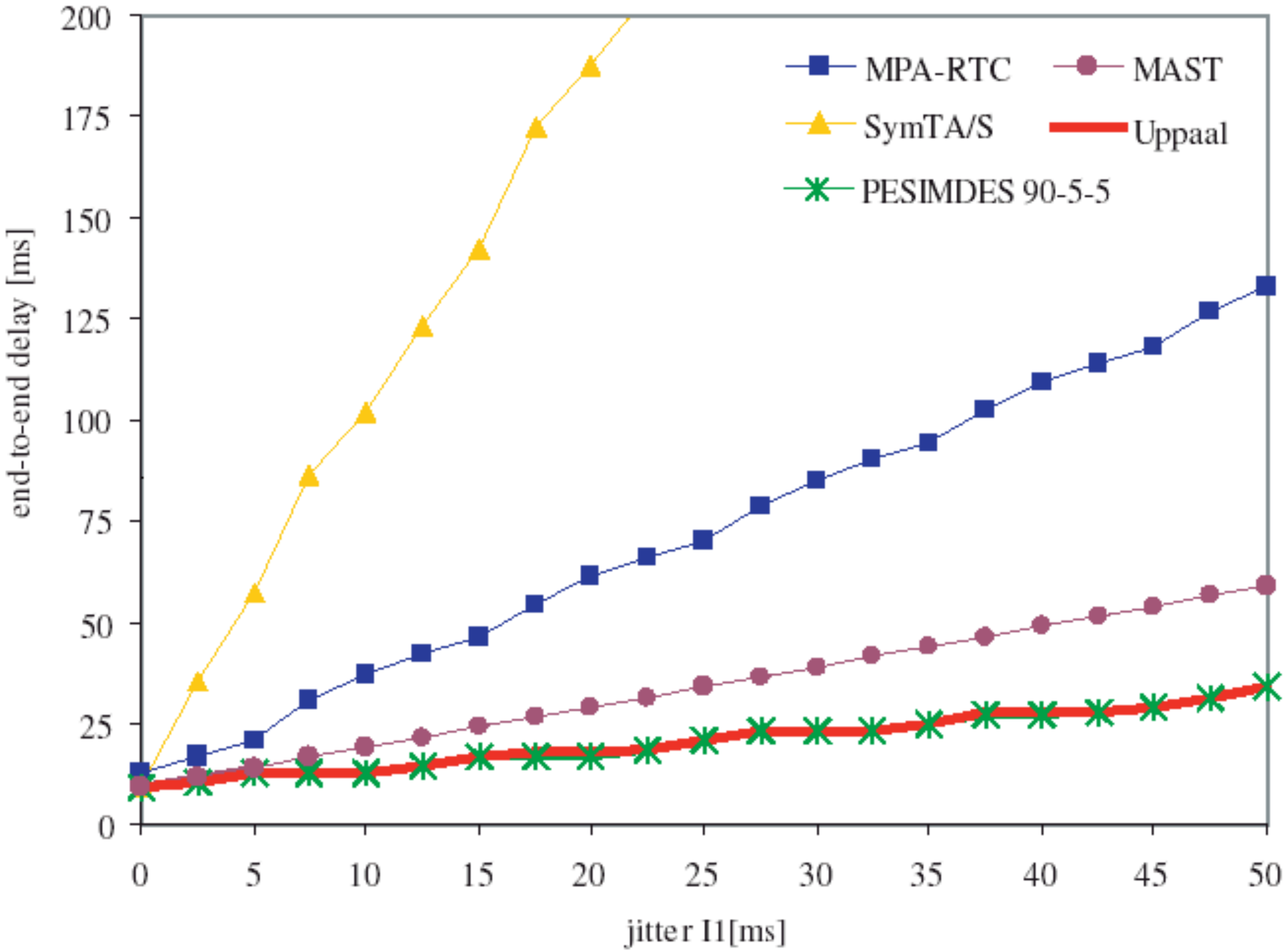
Input stream I1	periodic with burst (P=10ms, J=50ms, d=0ms)
Resource sharing	CPU1: FP preemptive
Task WCETs	T1: 1ms, T2: 4ms, T3: 4ms
Scheduling param.	1) priority T1: high, priority T3: low 2) priority T1: low, priority T3: high

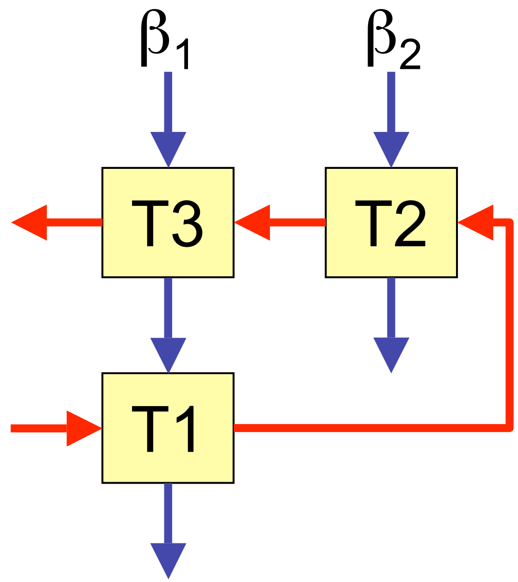
Benchmark 2-1 : T1 high





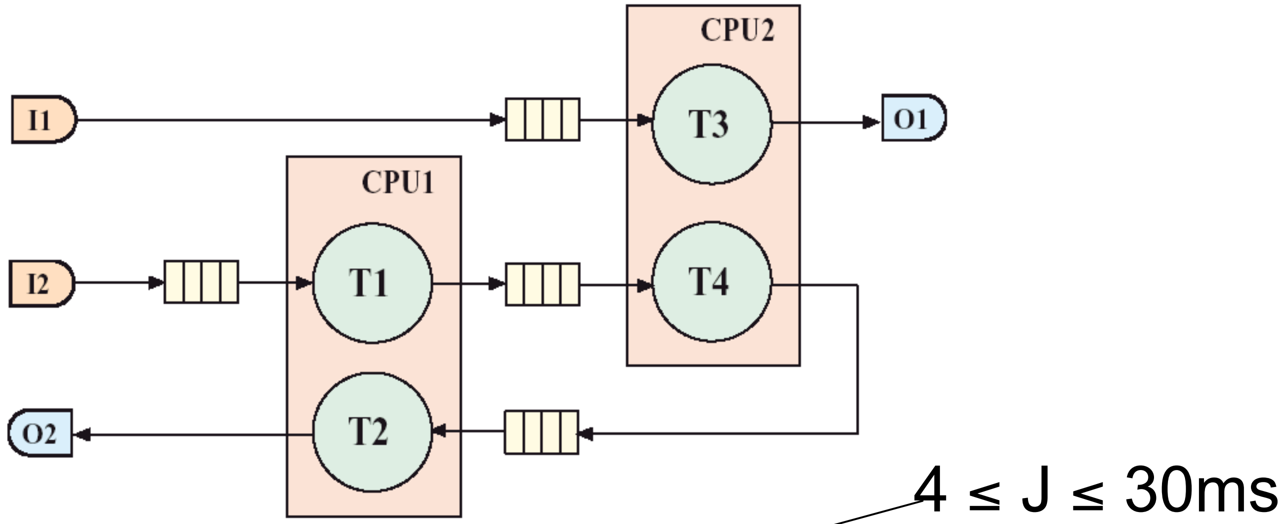
Benchmark 2-2 : T3 high





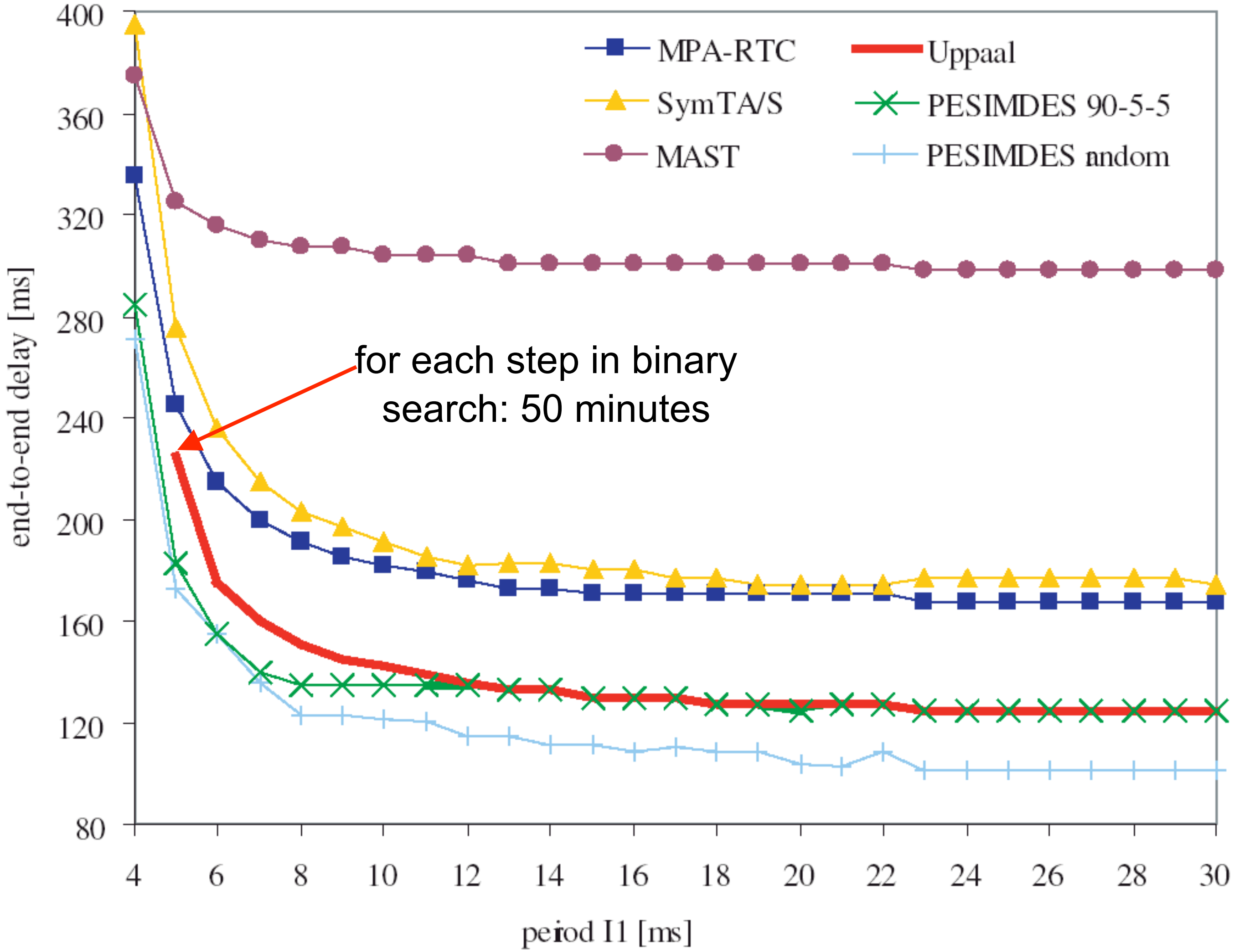
Benchmark 3

- Variable Feedback



Input streams	I1: periodic (P=4ms) I2: periodic with burst (P=100ms, J=200ms, d=0ms)
Resource sharing	CPU1: FP preemptive, CPU2: FP preemptive
Task WCETs	T1: 20ms, T2: 15ms, T3: 3ms, T4: 20ms
Scheduling param.	priority T1: high, priority T2: low priority T3: high, priority T4: low

Benchmark 3 : T1 high



(Expected) Results

- Understand the modeling power of different models and the relation between models and analysis accuracy.
- Improve methods by combining ideas and abstractions.
- Not: competition ... .

In models for timed systems abstraction matters

Knowledge about MoCCs that (also) talk about resource usage are far less understood