Relaxing Mixed Criticality Scheduling Strictness For Task Sets Scheduled With Fixed Priority

François Santy[1], Laurent George[2], Philippe Thierry[2], Joël Goossens[1]

[1]Université Libre de Bruxelles
Bruxelles, Belgium

[2]ECE / LACSC
Paris, France

What are mixed-criticality systems?

A task's tolerance to a deadline miss is represented by a criticality level:

- High criticality tasks: tolerate no deadline miss
- Low criticality tasks: tolerate occasional deadline misses

Mixed-criticality systems are systems composed of tasks having heterogeneous criticality levels.

Mixed-criticality in the avionics

Level	Failure Condition	Effects
A	Catastrophic	Failure may cause a crash. Error or loss of critical function required to safely fly and land aircraft.
B	Hazardous	Failure has a large negative impact on safety or performance, or reduces the ability of the crew to operate the aircraft due to physical distress or a higher workload, or causes serious or fatal injuries among the passengers.
C	Major	Failure is significant, but has a lesser impact than a Hazardous failure (for example, leads to passenger discomfort rather than injuries) or significantly increases crew workload.
D	Minor	Failure is noticeable, but has a lesser impact than a Major failure (for example, causing passenger inconvenience or a routine flight plan change).
E	No effect	Failure has no impact on safety, aircraft operation, or crew workload.

RTCA-DO178B

Work hypotheses

- Uni-processor
- Sporadic tasks
- Preemptive tasks
- CAPA scheduler

Mixed criticality systems are subject to certification

- $\tau_1 = (D_1, T_1, X_1=2, C_1)$
- $\tau_2 = (D_2, T_2, X_2=1, C_2)$

$CertificationAuthority_1$

(Reasonable degree of assurance)

- $\tau_1 = (D_1, T_1, X_1=2, 1)$
- $\tau_2 = (D_2, T_2, X_2=1, 2)$

$CertificationAuthority_2$

(High degree of assurance)

- $\tau_1 = (D_1, T_1, X_1=2, 3)$
- $\tau_2 = (D_2, T_2, X_2=1, 4)$

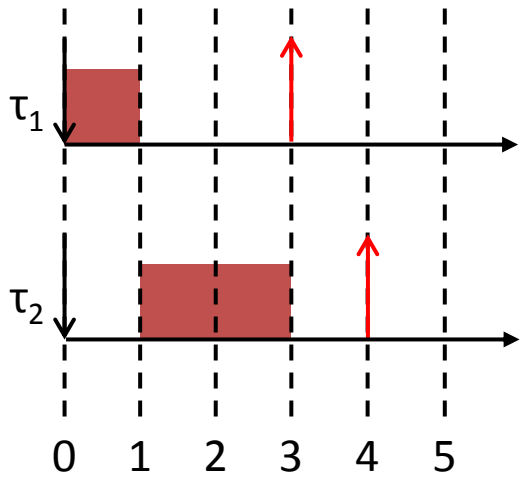
D_i = Deadline
T_i = Period
X_i = criticality level
C_i = WCET

Each certification authority might only be interested in a subset of the tasks

CertificationAuthority$_1$ will certify the system if the tasks:

τ_1, τ_2

always meet their deadline:

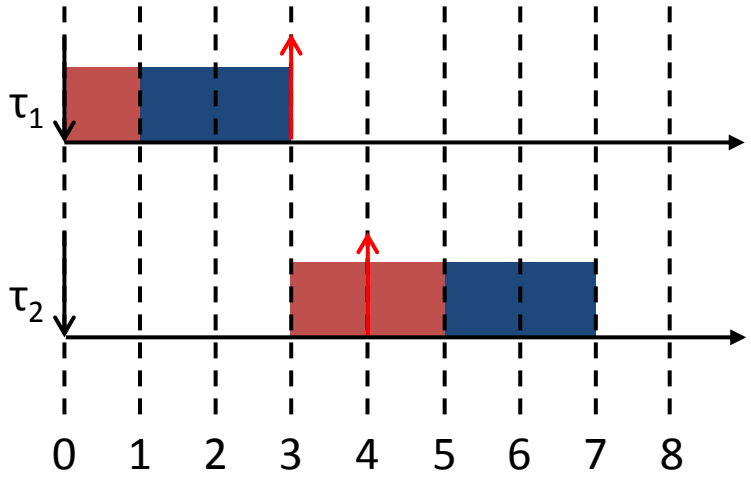


- $\tau_1 = (D_1, T_1, X_1=2, 1)$
- $\tau_2 = (D_2, T_2, X_2=1, 2)$

CertificationAuthority$_2$ will certify the system if the task:

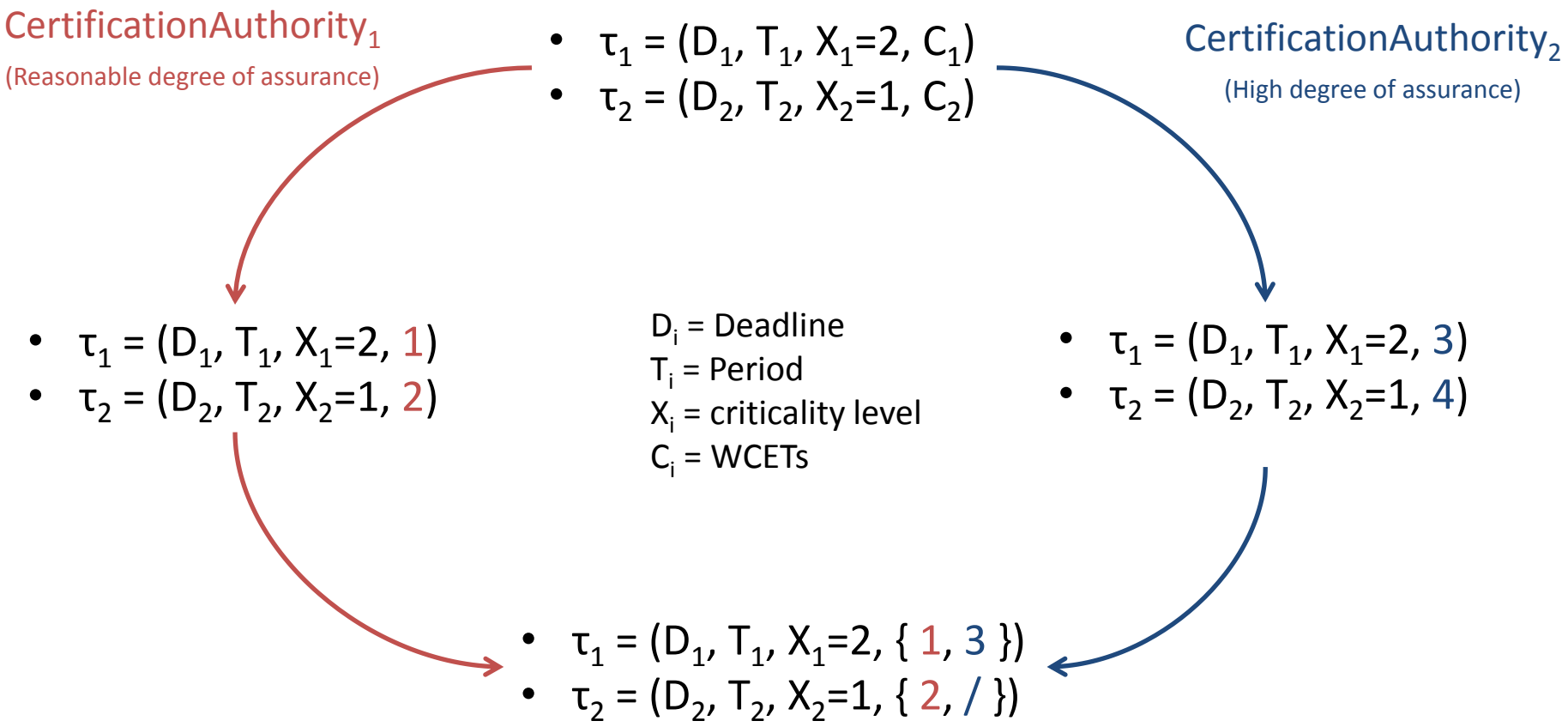
τ_1

always meet its deadline:



- $\tau_1 = (D_1, T_1, X_1=2, 3)$
- $\tau_2 = (D_2, T_2, X_2=1, 4)$

Mixed criticality systems are subject to certification

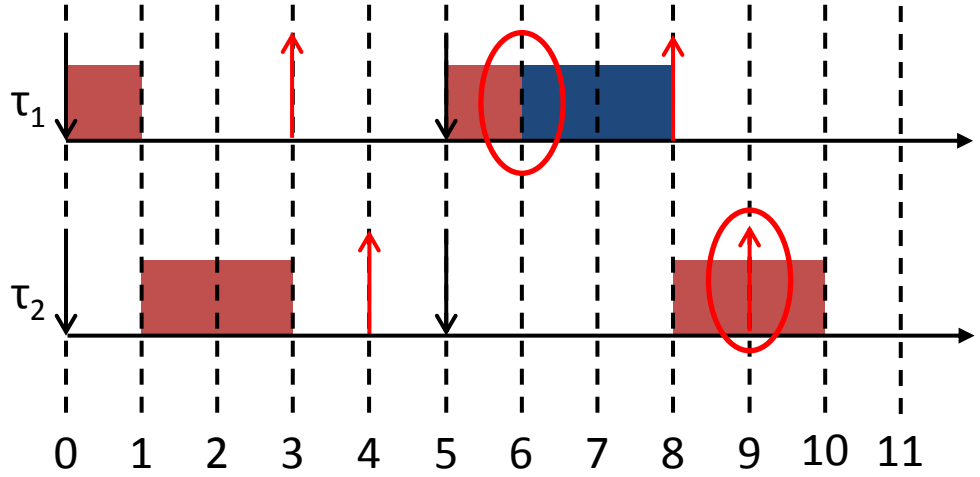


How to deal with mixed criticality tasks?

- Prioritize the deadline of high criticality tasks
- Possibly at the expense of lower criticality tasks
- Task suspension may occur during the scheduling of the system

Task suspension only relies on the certification hypotheses

- $\tau_1 = (D_1, T_1, X_1=2, \{ 1, 3 \})$
- $\tau_2 = (D_2, T_2, X_2=1, \{ 2, / \})$



- $\tau_1 = (D_1, T_1, X_1=2, 1)$
- $\tau_2 = (D_2, T_2, X_2=2, 2)$

Initial task set

- $\tau_1 = (D_1, T_1, X_1=2, 3)$

Resulting task set

Task suspension occurs as soon as the certification hypotheses are not met anymore.

Task suspension is undesirable

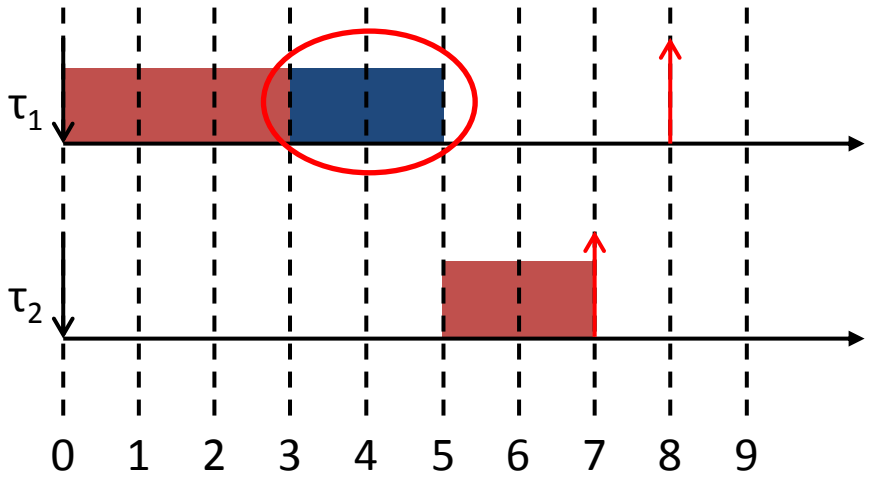
- It is a reasonable agreement

- Nevertheless:
  - It should be avoided when it is not necessary
  - It should be restrained as much as possible in time when required

Task suspension is not always carried out when strictly necessary!

- $\tau_1 = (D_1 = T_1 = 8, X_1 = 2, C_1 = \{ 3, 7 \})$
- $\tau_2 = (D_2 = T_2 = 7, X_2 = 1, C_2 = \{ 2, / \})$

τ_1's allowance

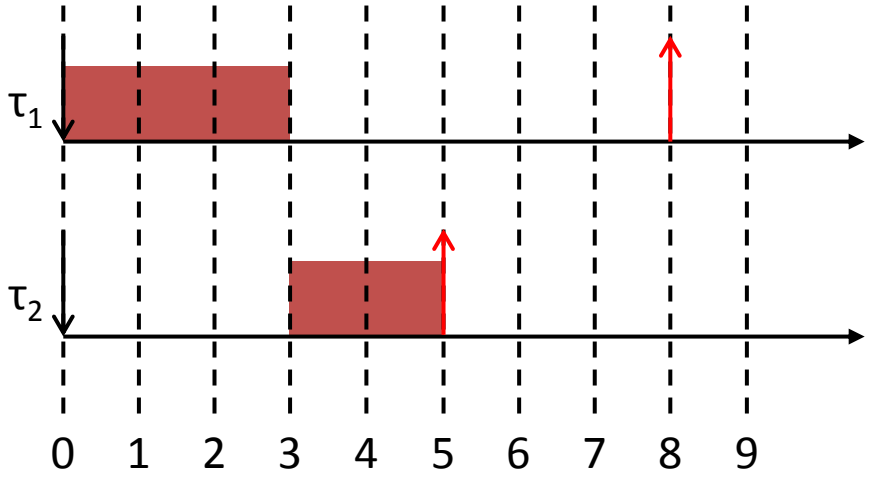


Computation of the allowance: a fair distribution

- Advantage: straightforward
- Drawback: maximum value is restrained by the least flexible task

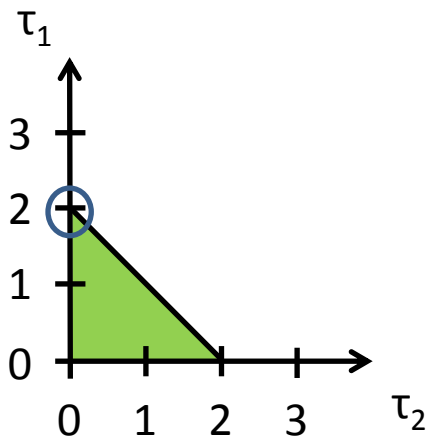
- $\tau_1 = (D_1 = T_1 = 8, X_1 = 2, C_1 = \{ 3, 7 \})$
- $\tau_2 = (D_2 = T_2 = 5, X_2 = 1, C_2 = \{ 2, / \})$

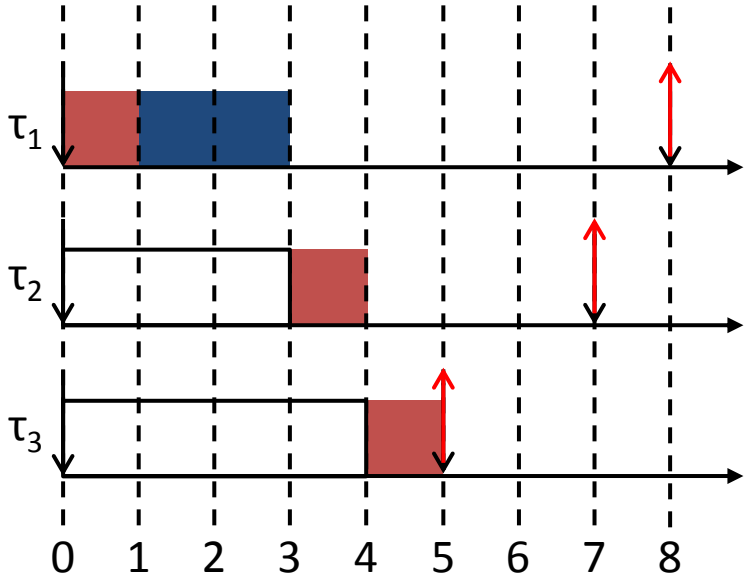
τ_1 can be granted no allowance because of τ_2



Computation of the allowance: any distribution

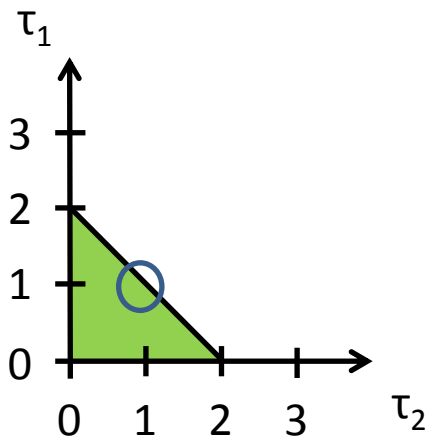
- Much more flexible
- Requires the computation of the allowance domain
  - $\tau_1 = (D_1 = T_1 = 8, X_1 = 3, C_1 = \{ 1, 3, 7 \})$
  - $\tau_2 = (D_2 = T_2 = 7, X_2 = 2, C_2 = \{ 1, 3, / \})$
  - $\tau_3 = (D_3 = T_3 = 5, X_2 = 1, C_3 = \{ 1, /, / \})$

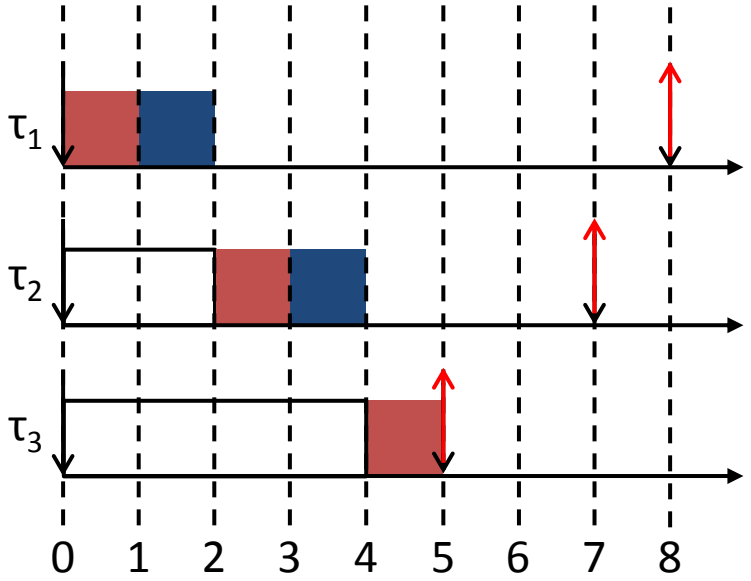




Computation of the allowance: any distribution

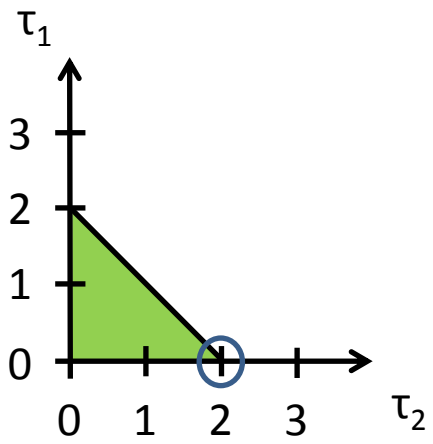
- Much more flexible
- Requires the computation of the allowance domain
  - $\tau_1 = (D_1 = T_1 = 8, X_1 = 3, C_1 = \{ 1, 3, 7 \})$
  - $\tau_2 = (D_2 = T_2 = 7, X_2 = 2, C_2 = \{ 1, 3, / \})$
  - $\tau_3 = (D_3 = T_3 = 5, X_2 = 1, C_3 = \{ 1, /, / \})$

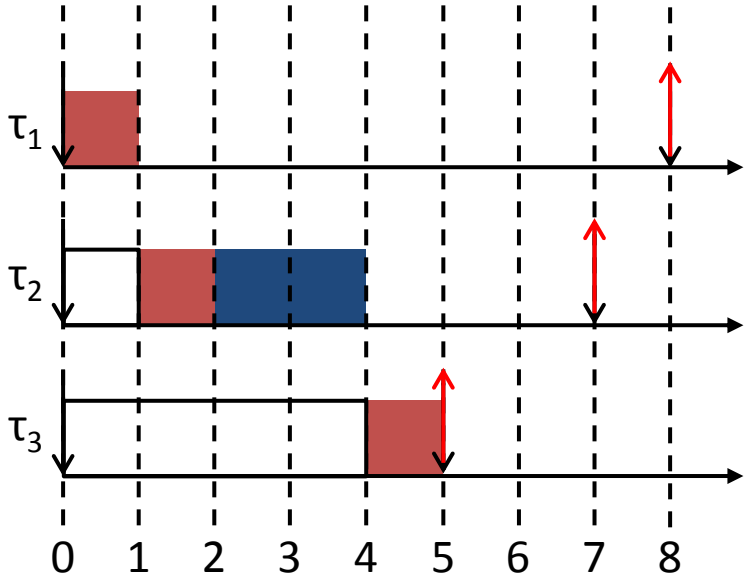




Computation of the allowance: any distribution

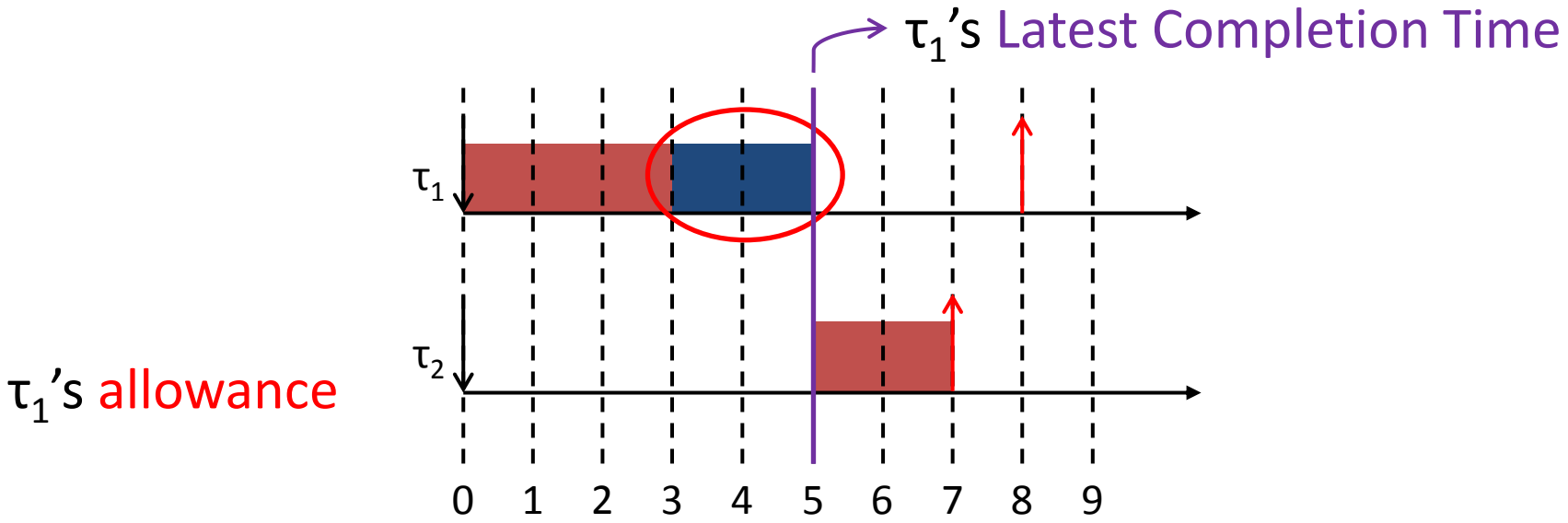
- Much more flexible
- Requires the computation of the allowance domain
  - $\tau_1 = (D_1 = T_1 = 8, X_1 = 3, C_1 = \{ 1, 3, 7 \})$
  - $\tau_2 = (D_2 = T_2 = 7, X_2 = 2, C_2 = \{ 1, 3, / \})$
  - $\tau_3 = (D_3 = T_3 = 5, X_2 = 1, C_3 = \{ 1, /, / \})$





Implementation of the allowance

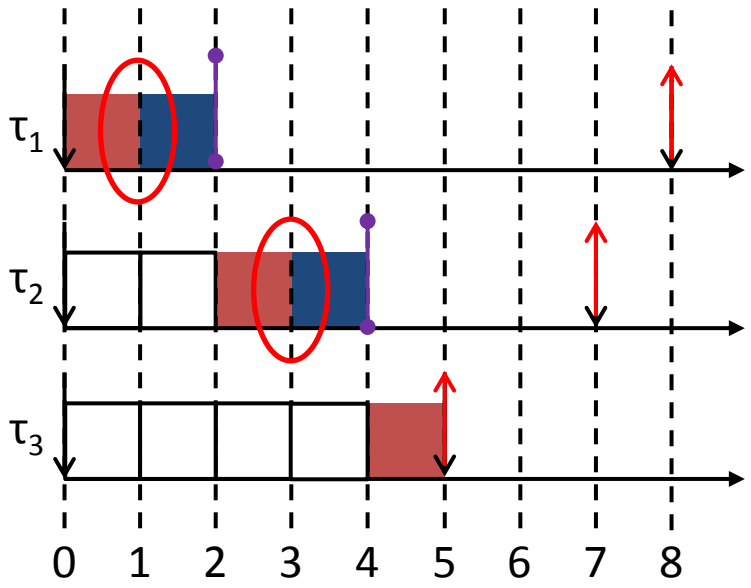
- We need to detect absolute time instants
- Online management of the allowance: Latest Completion Time (LCT)



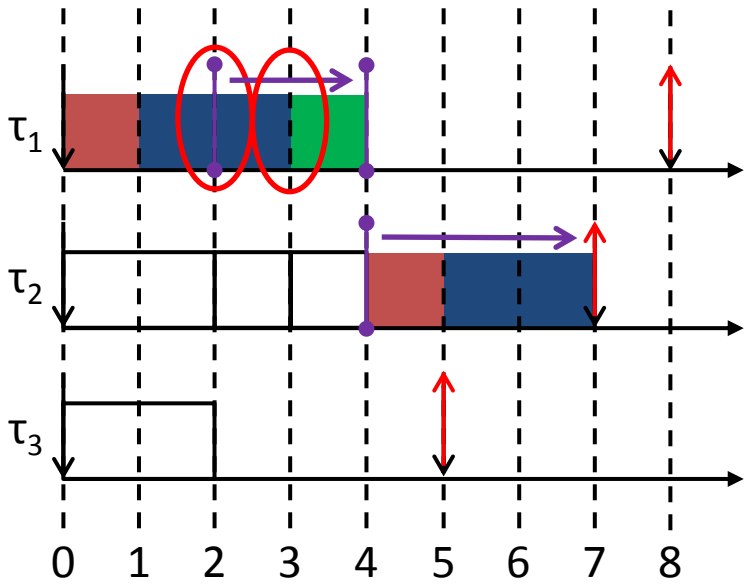
Implementation of the allowance

- $\tau_1 = (D_1 = T_1 = 8, X_1 = 3, C_1 = \{ 1, 3, 7 \})$
- $\tau_2 = (D_2 = T_2 = 7, X_2 = 2, C_2 = \{ 1, 3, / \})$
- $\tau_3 = (D_3 = T_3 = 5, X_2 = 1, C_3 = \{ 1, /, / \})$

Criticality level 1

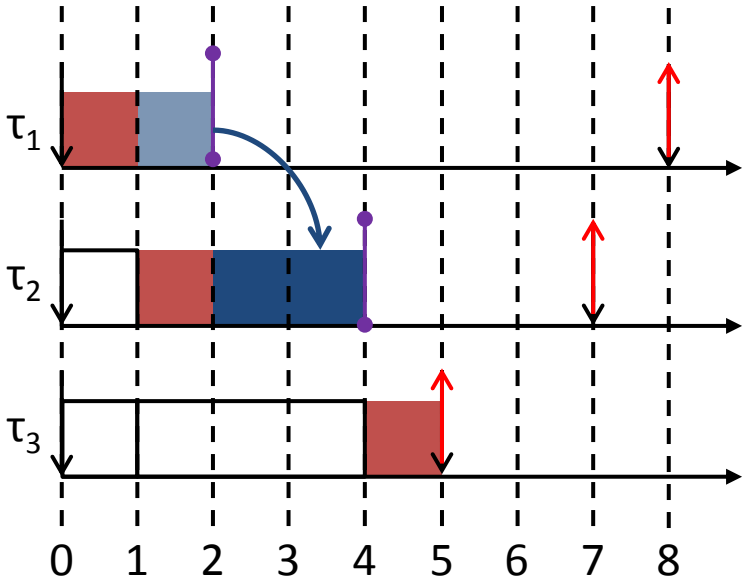


Criticality level 2



Allowance recovery based on the LCT

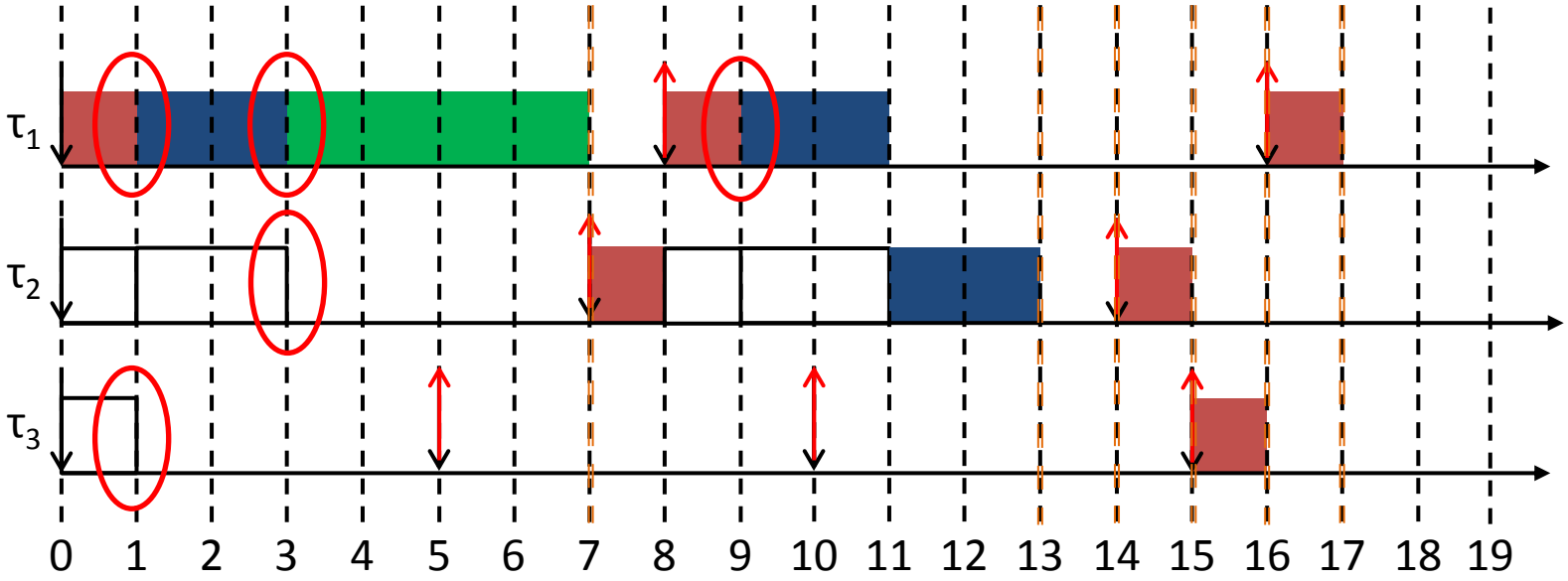
- Unused allowance can be recovered by tasks having a lower priority:
  - $\tau_1 = (D_1 = T_1 = 8, X_1 = 3, C_1 = \{ 1, 3, 7 \})$
  - $\tau_2 = (D_2 = T_2 = 7, X_2 = 2, C_2 = \{ 1, 3, / \})$
  - $\tau_3 = (D_3 = T_3 = 5, X_2 = 1, C_3 = \{ 1, /, / \})$



Recovered allowance

Task suspension can be restrained in time

- $\tau_1 = (D_1 = T_1 = 8, X_1 = 3, C_1 = \{ 1, 3, 7 \})$
- $\tau_2 = (D_2 = T_2 = 7, X_2 = 2, C_2 = \{ 1, 3, / \})$
- $\tau_3 = (D_3 = T_3 = 5, X_2 = 1, C_3 = \{ 1, /, / \})$



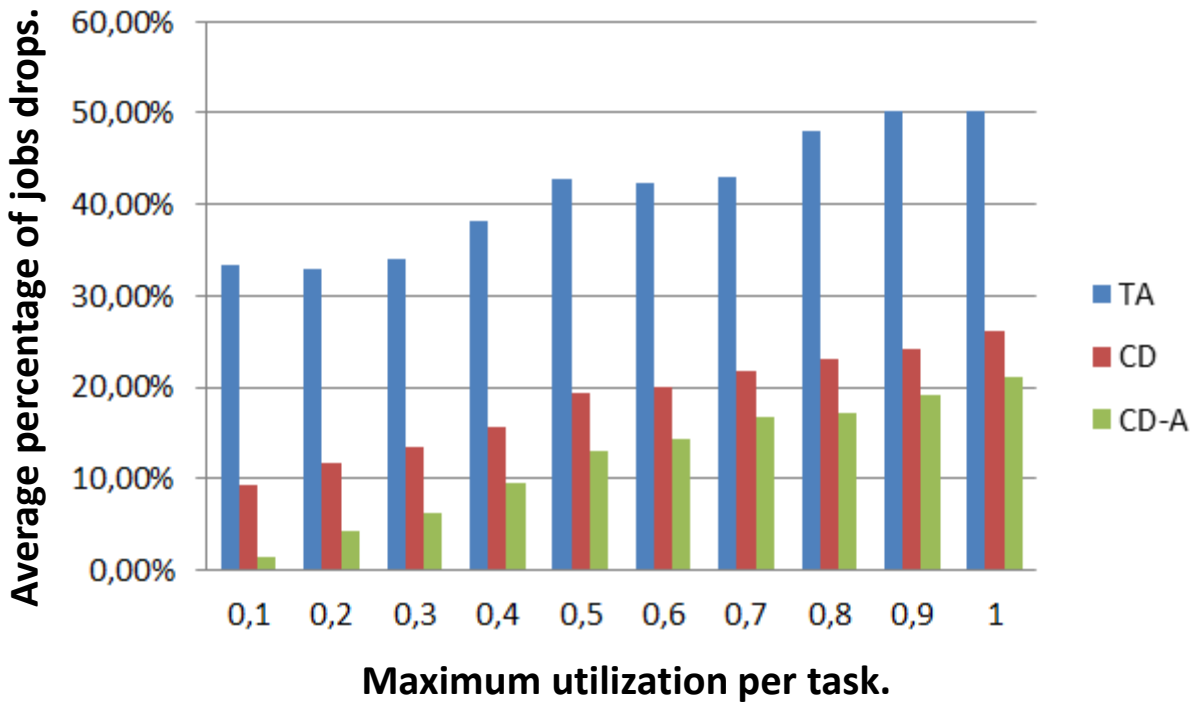
Each time an idle time is met, the criticality of the system can be reset.

Quantifying the benefits of our improvements

We compared three different approaches:

- the traditional approach (TA)
- the task reversion mechanism (CD)
- the task reversion mechanism as well as the ability to consume allowance (CD-A)

Quantifying the benefits of our improvements



We notice an average decrease of 30% of the jobs drops

Conclusion

We were facing two problems:

1. Unnecessary task suspension
2. Everlasting task suspension

We solved those problems by:

1. The computation of the allowance and LCT online mechanism
2. Restraining task suspensions within finite time intervals

The combination of 1 and 2 allows an average decrease of 30% of jobs drops