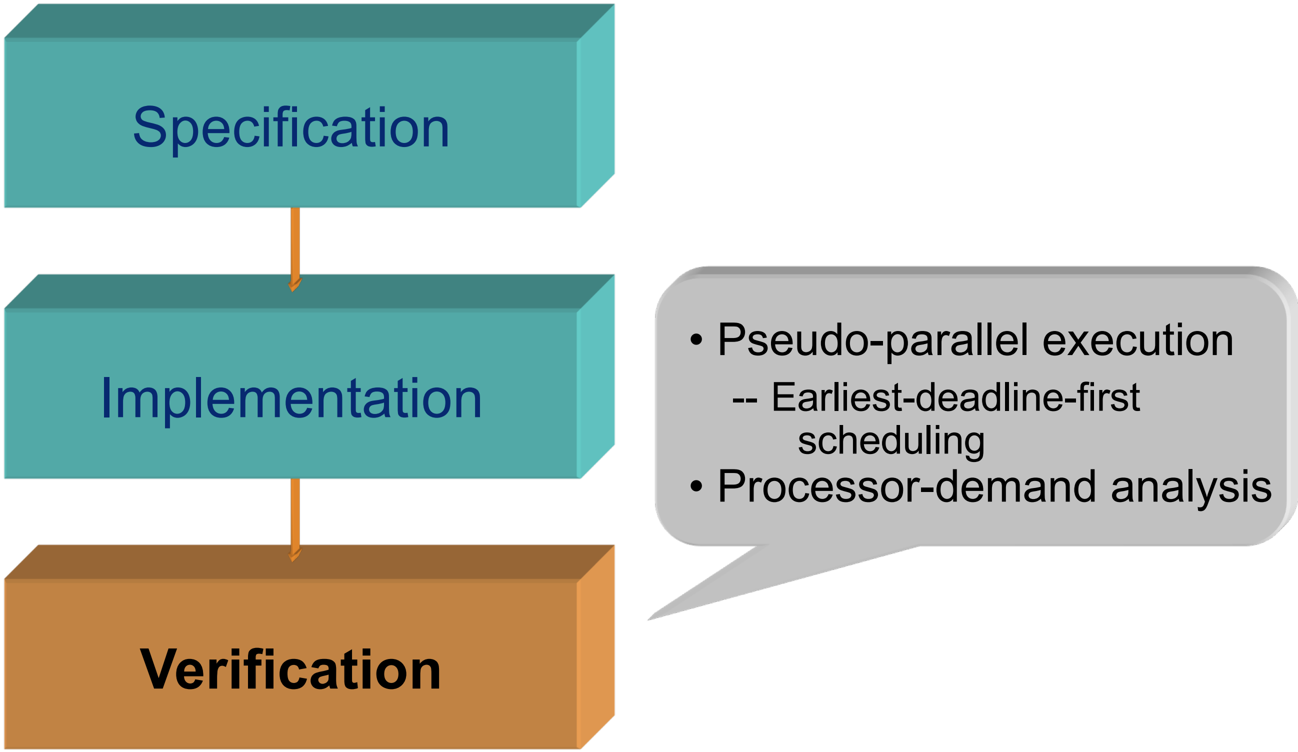
Real-Time Systems

Lecture #13

Professor Jan Jonsson

Department of Computer Science and Engineering
Chalmers University of Technology

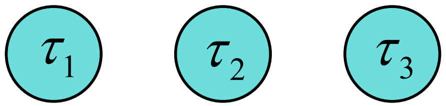
Real-Time Systems



Example: scheduling using EDF

Problem: Assume a system with tasks according to the figure below. The timing properties of the tasks are given in the table. All tasks arrive the first time at time 0.

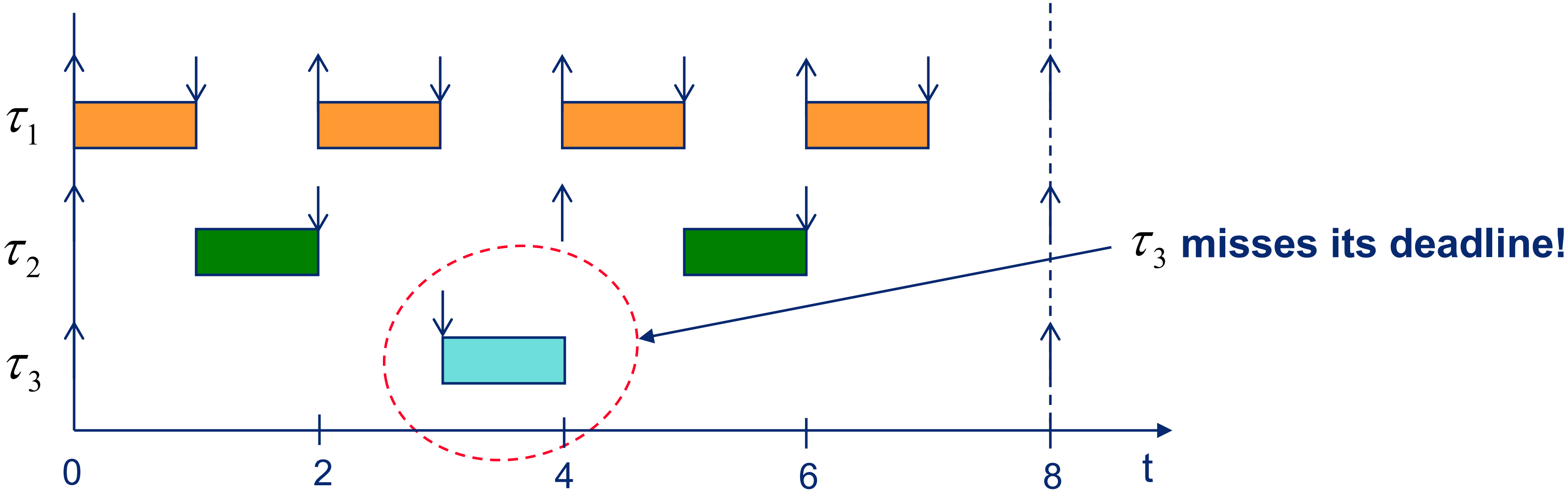
Investigate the schedulability of the tasks when EDF is used. (Note that $D_i < T_i$ for all tasks)



Task	C_i	D_i	T_i
τ_1	1	1	2
τ_2	1	2	4
τ_3	1	3	8

Example: scheduling using EDF

Simulate an execution of the tasks:



The tasks are not schedulable even though

$$U = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{7}{8} = 0.875 < 1$$

Feasibility analysis for EDF

What analysis methods are suitable for general EDF:

- Utilization-based analysis?
  Not suitable! Not general enough or exact enough
  - Does not work well for the case of $D_i < T_i$
- Response-time analysis?
  Not suitable! Analysis much more complex than for DM/RM
  - The critical instant of a task does not necessarily occur when the task arrives at the same time as some other tasks.
  - Instead, the response time of the task may be maximized at some other (asynchronous) arrival pattern.
  - Consequently, the critical instant for each task can in general only be identified by observing the actual schedule.

Feasibility tests

What types of feasibility tests exist?

- Hyper period analysis (for any type of scheduler)
  - In an existing schedule no task execution may miss its deadline
- Processor utilization analysis (static/dynamic priority scheduling)
  - The fraction of processor time that is used for executing the task set must not exceed a given bound
- Response time analysis (static priority scheduling)
  - The worst-case response time for each task must not exceed the deadline of the task
- Processor demand analysis (dynamic priority scheduling)
  - The accumulated computation demand for the task set under a given time interval must not exceed the length of the interval

Processor-demand analysis

Processor demand:

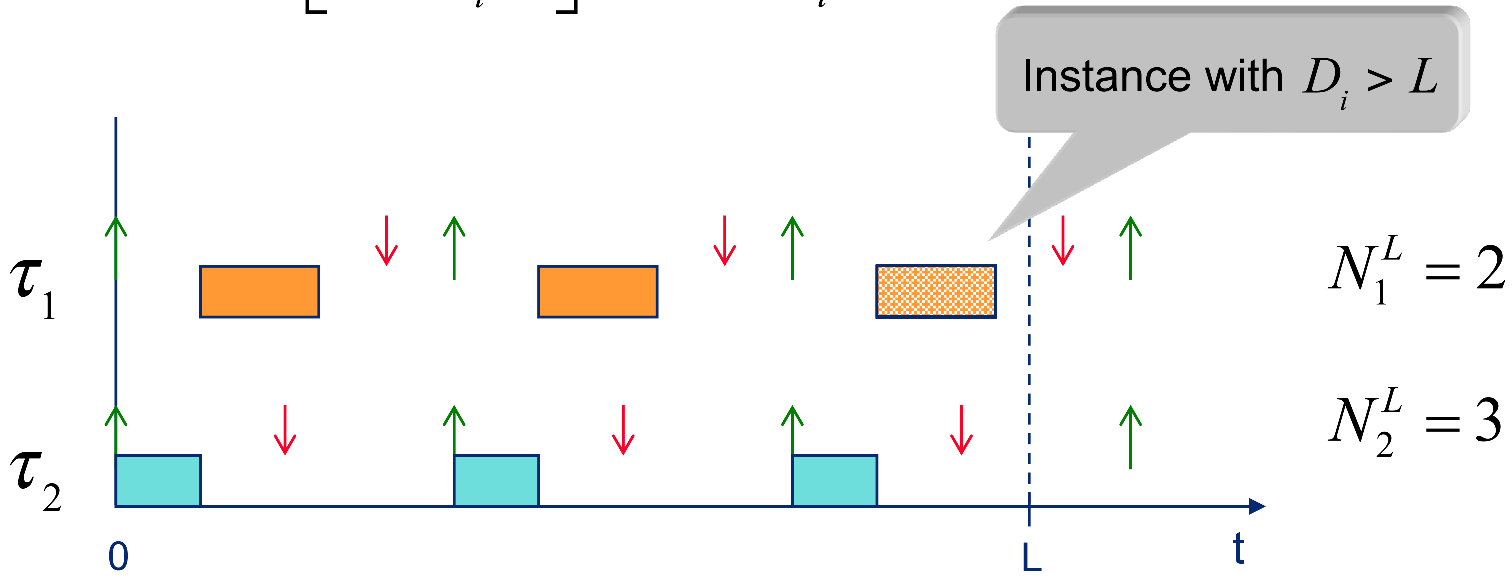
- The processor demand for a task τ_i in a given time interval $[0, L]$ is the amount of processor time that the task needs in the interval in order to meet the deadlines that fall within the interval.
- Let N_i^L represent the number of instances of τ_i that must complete execution before L .
- The total processor demand up to L is

$$\boxed{C_P(0, L) = \sum_{i=1}^{n} N_i^L C_i}$$

Processor-demand analysis

Processor demand:

- We can calculate N_i^L by counting how many times task τ_i has arrived during the interval $\left[0, L - D_i\right]$.
- We can ignore instances of the task that arrived during the interval $\left[L - D_i, L\right]$ since $D_i > L$ for these instances.



Processor-demand analysis

Processor-demand analysis:

- We can express N_i^L as

$$\boxed{N_i^L = \left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1}$$

- The total processor demand is thus

$$\boxed{C_P(0,L) = \sum_{i=1}^{n} \left(\left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1 \right) C_i}$$

Exact feasibility test for EDF

(Sufficient and necessary condition)

A sufficient and necessary condition for EDF scheduling of synchronous task sets, for which $D_i \leq T_i$, is

$$\forall L : C_P(0, L) \leq L$$

where $C_P(0, L)$ is the total processor demand in $[0, L]$.

In other words: *for the task set to be schedulable with EDF there must not exist an interval of length L in the schedule where the processor demand in that interval exceeds the length L.*

The processor-demand analysis and associated feasibility test was presented by S. Baruah, L. Rosier and R. Howell in 1990.

Exact feasibility test for EDF

(Sufficient and necessary condition)

The test is valid under the following assumptions:

1. All tasks are independent
   – There must not exist dependencies due to precedence or mutual exclusion
2. All tasks are periodic or sporadic
3. All tasks have identical offsets (= synchronous task set)
4. Task deadline does not exceed the period
   (= constrained-deadline tasks)
5. Task preemptions are allowed

Processor-demand analysis

(Performance aspects)

How many intervals must be examined?

- Only intervals coinciding with the absolute deadlines of tasks need to be examined

For synchronous task sets the feasibility test can consequently be rewritten as follows:

$$\forall L \in K : C_P(0, L) \leq L$$

$$K = \left\{ D_i^k \mid D_i^k = kT_i + D_i,\, D_i^k \leq L_{\max},\, 1 \leq i \leq n,\, k \geq 0 \right\}$$

Processor-demand analysis

(Performance aspects)

What is the largest interval that must be examined?

- For synchronous task sets in general the largest required interval will always be bounded by the hyper period.
  ⇒ the analysis may have exponential time complexity

$$L_{\mathrm{LCM}} = \mathrm{LCM}\{T_1,\ldots,T_n\}$$

- For most synchronous task sets the largest required interval can be shorter than the hyper period.
  ⇒ the analysis can have pseudo-polynomial time complexity
  (by using a special upper bound) [see Appendix]

Processor-demand analysis

(Performance aspects)

What is the largest interval that must be examined?

- For synchronous task sets with a utilization U < 1 we can use the bound by Baruah, Rosier and Howell:

$$L_{\mathrm{BRH}} = \max\left\{ D_1, \ldots, D_n, \frac{\sum_{i=1}^{n} (T_i - D_i) U_i}{1 - U} \right\}$$

- Properties:
  - ✓ For most task sets $L_{\mathrm{BRH}} < L_{\mathrm{LCM}}$
  - ✓ However, there exist task sets for which $L_{\mathrm{BRH}} > L_{\mathrm{LCM}}$ (typically: task sets with harmonic periods)

Processor-demand analysis

(Performance aspects)

Recommendations for largest required interval:

- For synchronous task sets in general:

$$L_{\max} = L_{\text{LCM}}$$

- For synchronous task sets with utilization U < 1 the following least upper bound can be used:

$$L_{\max} = \min(L_{\text{BRH}}, L_{\text{LCM}})$$

Processor-demand analysis

(Performance aspects)

Examples:

Task	C_i	D_i	T_i
τ_1	1	1	2
τ_2	1	2	4
τ_3	1	3	8

$L_{\text{BRH}} = 13 \quad L_{\text{LCM}} = 8$

Note: $L_{\text{BRH}} > L_{\text{LCM}}$

Task	C_i	D_i	T_i
τ_1	3	10	20
τ_2	10	27	30
τ_3	25	54	60

$L_{\text{BRH}} = 54 \quad L_{\text{LCM}} = 60$

Note: $L_{\text{BRH}} < L_{\text{LCM}}$

Task	C_i	D_i	T_i
τ_1	1	4	4
τ_2	3	10	15
τ_3	8	14	17

$L_{\text{BRH}} = 30 \quad L_{\text{LCM}} = 1020$

Note: $L_{\text{BRH}} \ll L_{\text{LCM}}$

Feasibility tests

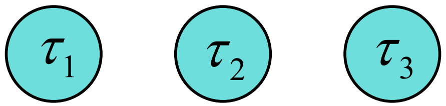
Summary for single-processor scheduling

	Implicit-deadline tasks $D_i = T_i$	Constrained-deadline tasks $D_i \leq T_i$
Static priority (RM/DM)	$U \leq n(2^{1/n} - 1)$	$\forall i : R_i = C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{R_i}{T_j} \right\rceil C_j \leq D_i$
Dynamic priority (EDF)	$U \leq 1$	$\forall L : \sum_{i=1}^{n} \left(\left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1 \right) C_i \leq L$

Example: scheduling using EDF

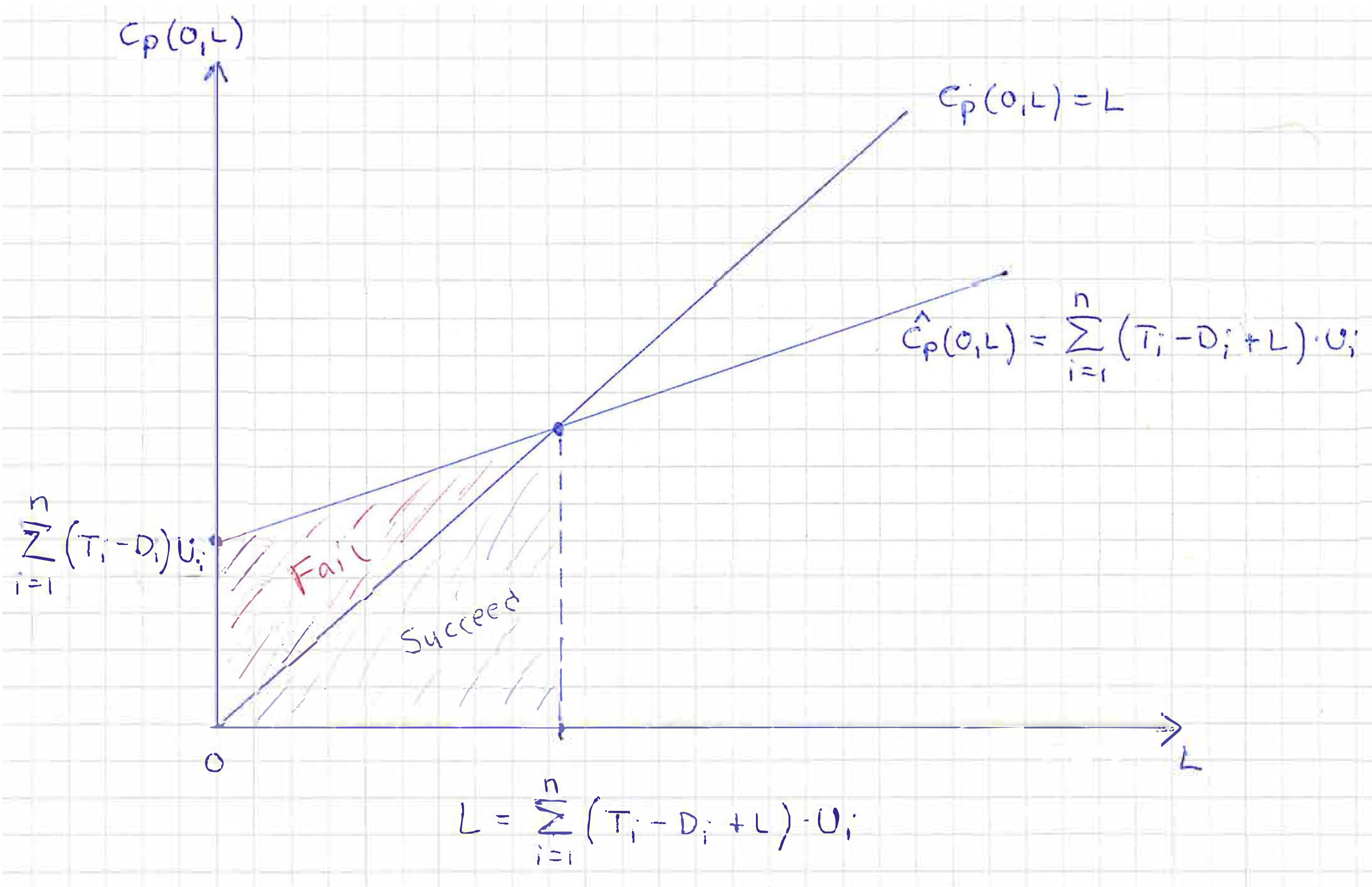
Problem: We once again assume the system with tasks given in the beginning of this lecture.

Show, by using processor-demand analysis, that the tasks are not schedulable using EDF.



Task	C_i	D_i	T_i
τ_1	1	1	2
τ_2	1	2	4
τ_3	1	3	8

Appendix: deriving the L_{BRH} bound



Appendix: deriving the L_{BRH} bound

Upper bound of total processor demand in $[0, L]$:

$$C_P(0,L) \le \hat{C}_P(0,L) = \sum_{i=1}^{n}\left(\frac{L-D_i}{T_i}+1\right)C_i = \sum_{i=1}^{n}\left(\frac{L-D_i}{T_i}+\frac{T_i}{T_i}\right)C_i =$$

$$= \sum_{i=1}^{n}\left(T_i - D_i + L\right)\cdot\frac{C_i}{T_i} = \sum_{i=1}^{n}\left(T_i - D_i + L\right)\cdot U_i$$

Intersection between the two lines in diagram:

$$L = \sum_{i=1}^{n}\left(T_i - D_i + L\right)U_i = \sum_{i=1}^{n}\left(T_i - D_i\right)\cdot U_i + \sum_{i=1}^{n} L\cdot U_i =$$

$$= \sum_{i=1}^{n}\left(T_i - D_i\right)\cdot U_i + L\cdot U \Rightarrow$$

$$L(1-U) = \sum_{i=1}^{n}\left(T_i - D_i\right)\cdot U_i \Rightarrow L = \frac{\sum_{i=1}^{n}\left(T_i - D_i\right)\cdot U_i}{1-U}$$