Rate Monotonic Analysis

Introduction

Periodic tasks

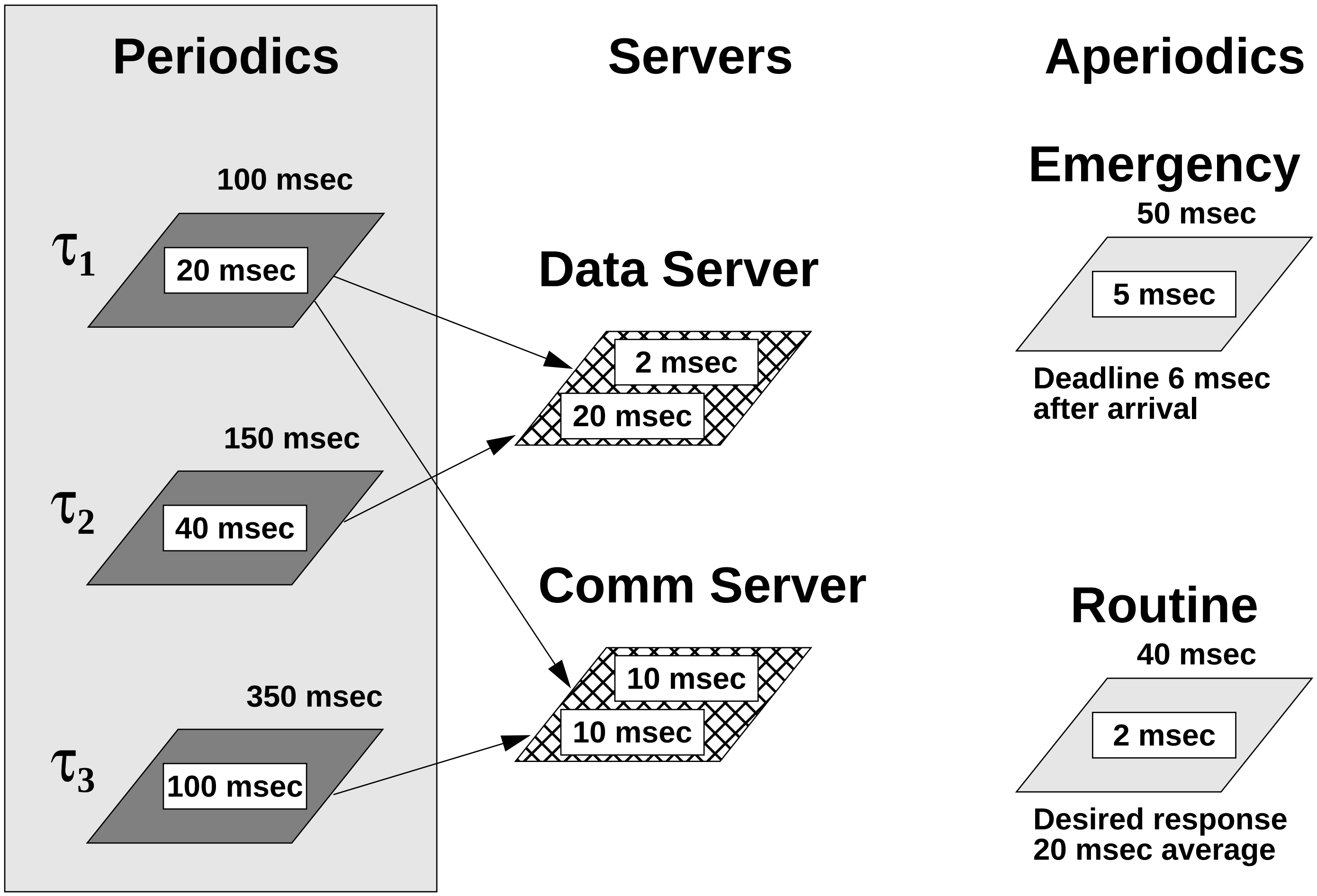
Extending basic theory

Synchronization and priority inversion

Aperiodic servers

Case Study: BSY-1 Trainer

A Sample Problem - Periodics



τ_2's deadline is 20 msec before the end of each period.

Concepts and Definitions - Periodics

Periodic task

- initiated at fixed intervals
- must finish before start of next cycle

Task's CPU utilization: $U_i = \frac{C_i}{T_i}$

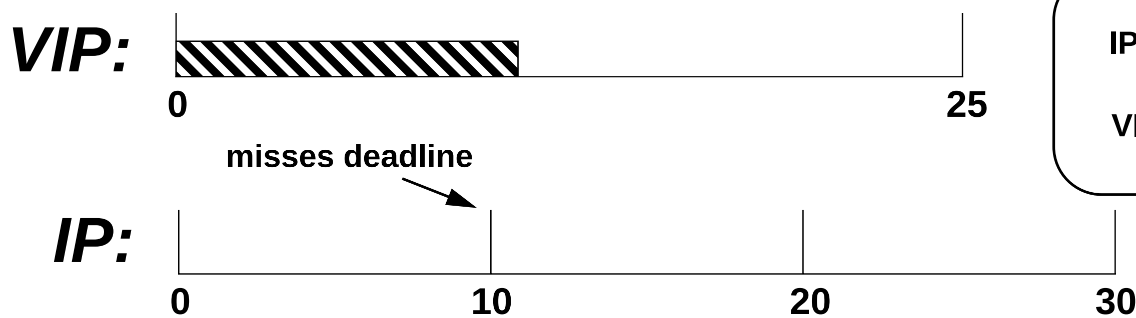
- C_i = compute time (execution time) for task τ_i
- T_i = period of task τ_i

CPU utilization for a set of tasks:

$$U = U_1 + U_2 + \ldots + U_n$$

Example of Priority Assignment

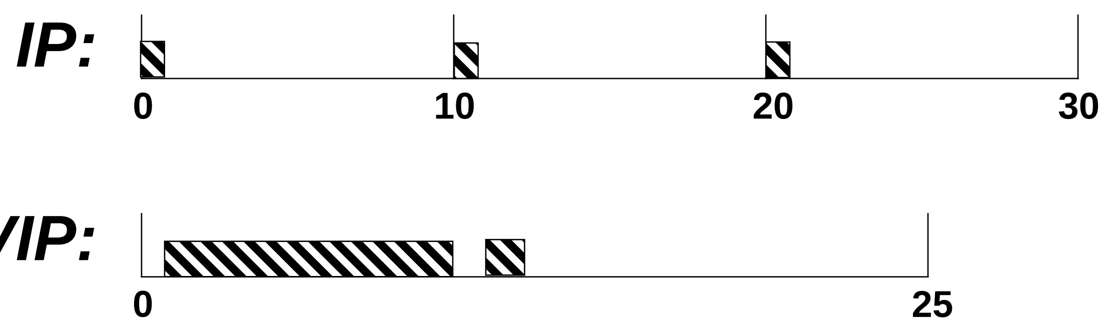
Semantic-Based Priority Assignment



IP: $U_{IP} = \frac{1}{10} = 0.10$

VIP: $U_{VIP} = \frac{11}{25} = 0.44$

Policy-Based Priority Assignment



Schedulability: UB Test

***Utilization bound(UB) test*: a set of *n* independent periodic tasks scheduled by the rate monotonic algorithm will always meet its deadlines, for all task phasings, if**

$$\frac{C_1}{T_1} + \dots + \frac{C_n}{T_n} \leq U(n) = n\,(2^{1/n} - 1)$$

U(1) = 1.0	***U(4) = 0.756***	***U(7) = 0.728***
U(2) = 0.828	***U(5) = 0.743***	***U(8) = 0.724***
U(3) = 0.779	***U(6) = 0.734***	***U(9) = 0.720***

For harmonic task sets, the utilization bound is *U(n)=1.00* for all *n*.

Note: UB test = Techniques 1 and 2 in handbook.

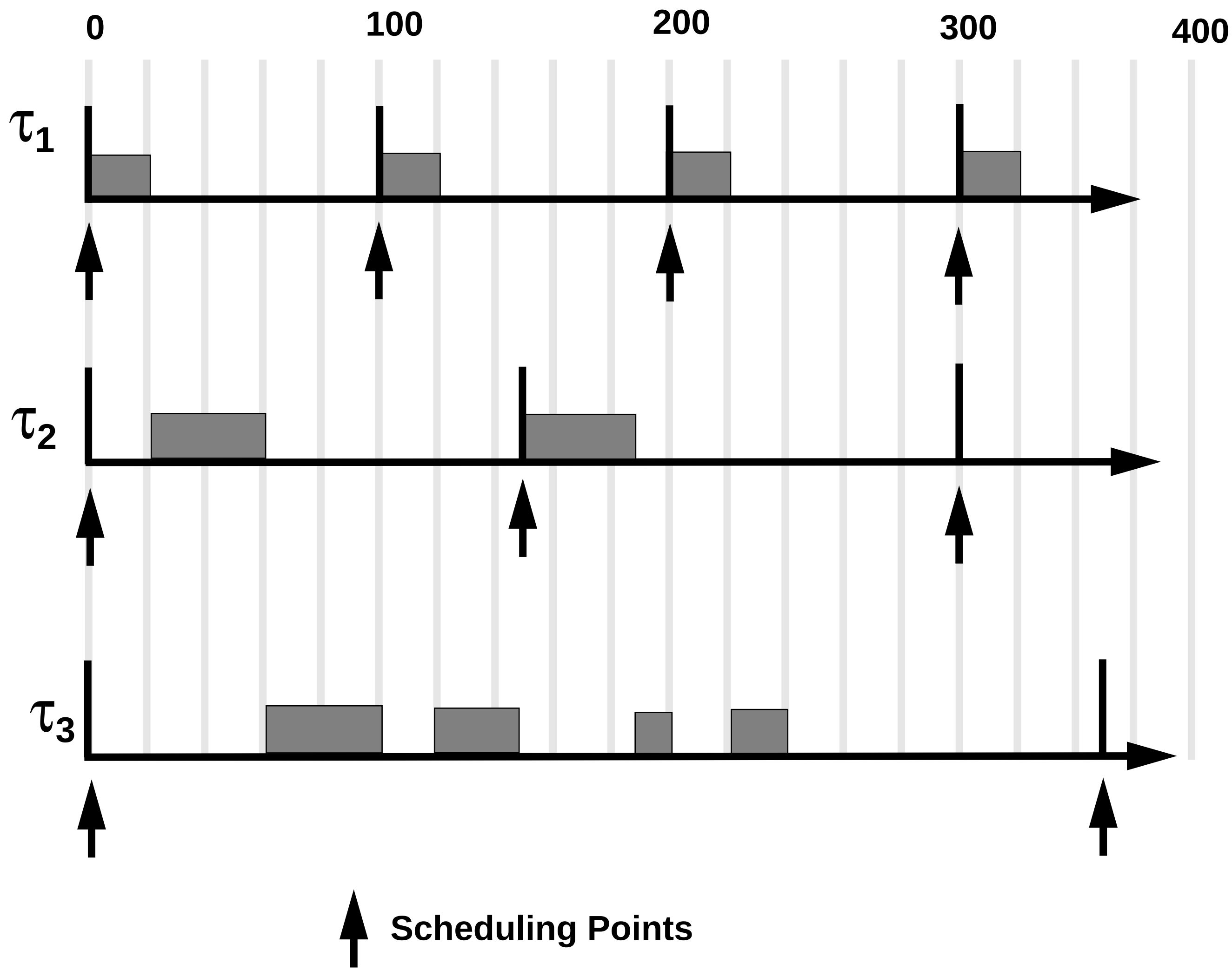
Sample Problem: Applying UB Test

	C	*T*	*U*
Task τ_1:	20	100	0.200
Task τ_2:	40	150	0.267
Task τ_3:	100	350	0.286

Total utilization is .200 + .267 + .286 = .753 < U(3) = .779

The periodic tasks in the sample problem are schedulable according to the UB test.

Timeline for Sample Problem



Exercise: Applying the UB Test

Given:

Task	*C*	*T*	*U*
τ_1	1	4	
τ_2	2	6	
τ_3	1	10	

a. What is total utilization?

b. Is the task set schedulable?

c. Draw the timeline.

d. What is the total utilization if $C_3 = 2$?

Toward a More Precise Test

UB test has three possible outcomes:

$$0 \le U \le U(n) \Rightarrow Success$$

$$U(n) < U \le 1.00 \Rightarrow Inconclusive$$

$$1.00 < U \Rightarrow Overload$$

UB test is conservative.

A more precise test can be applied.

Schedulability: RT Test

Theorem: for a set of independent, periodic tasks, if each task meets its first deadline, with worst-case task phasing, the deadline will always be met.

***Response time (RT) test*: let a_n = response time of task i. a_n may be computed by the following iterative formula:**

$$a_{n+1} = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{a_n}{T_j} \right\rceil C_j \qquad \text{where } a_0 = \sum_{j=1}^{i} C_j$$

Test terminates when $a_{n+1} = a_n$.

Task i is schedulable if its response time is before its deadline: $a_n \le T_i$

Example: Applying RT Test -1

Taking the sample problem, we increase the compute time of τ_1 from 20 to 40; is the task set still schedulable?

Utilization of first two tasks: 0.667 < U(2) = 0.828

- **first two tasks are schedulable by UB test**

Utilization of all three tasks: 0.953 > U(3) = 0.779

- **UB test is inconclusive**
- **need to apply RT test**

Example: Applying RT Test -2

Use RT test to determine if τ_3 meets its first deadline: i = 3

$$a_0 = \sum_{j=1}^{3} C_j = C_1 + C_2 + C_3 = 40 + 40 + 100 = 180$$

$${}_1 = \boldsymbol{C}_i + \sum_{j=1}^{i-1} \left\lceil \frac{a_0}{T_j} \right\rceil C_j = \boldsymbol{C}_3 + \sum_{j=1}^{2} \left\lceil \frac{a_0}{T_j} \right\rceil C_j$$

$$= 100 + \left\lceil \frac{180}{100} \right\rceil (40) + \left\lceil \frac{180}{150} \right\rceil (40) = 100 + 80 + 80 = 260$$

Example: Applying the RT Test -3

$$= \boldsymbol{C}_3 + \sum_{j=1}^{2} \left\lceil \frac{a_1}{T_j} \right\rceil C_j = 100 + \left\lceil \frac{260}{100} \right\rceil (40) + \left\lceil \frac{260}{150} \right\rceil (40) = 30$$

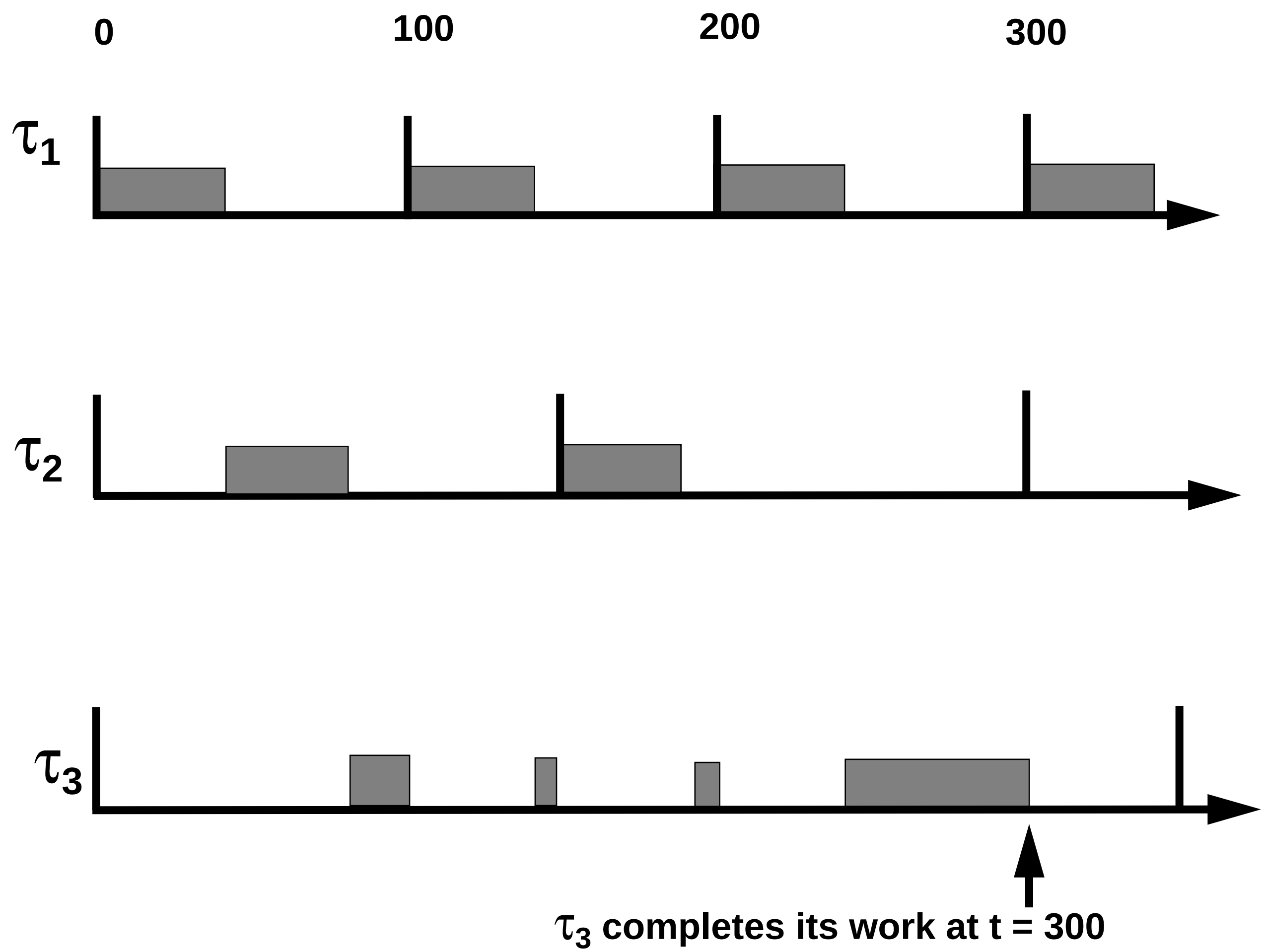
$$= \boldsymbol{C}_3 + \sum_{j=1}^{2} \left\lceil \frac{a_2}{T_j} \right\rceil C_j = 100 + \left\lceil \frac{300}{100} \right\rceil (40) + \left\lceil \frac{300}{150} \right\rceil (40) = 30$$

$$a_3 = a_2 = 300 \quad \text{Done!}$$

Task τ_3 is schedulable using RT test.

$$a_3 = 300 < T = 350$$

Timeline for Example



Exercise: Applying RT Test

Task τ_1: $C_1 = 1$ $T_1 = 4$

Task τ_2: $C_2 = 2$ $T_2 = 6$

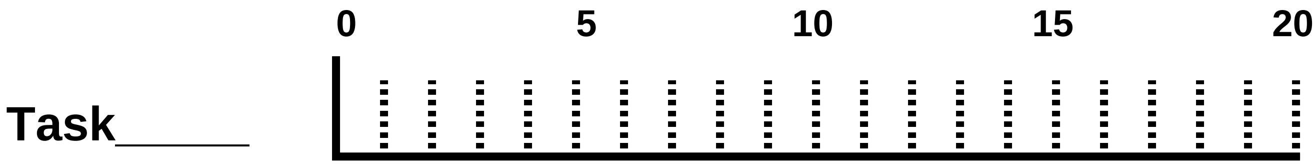
Task τ_3: $C_3 = 2$ $T_3 = 10$

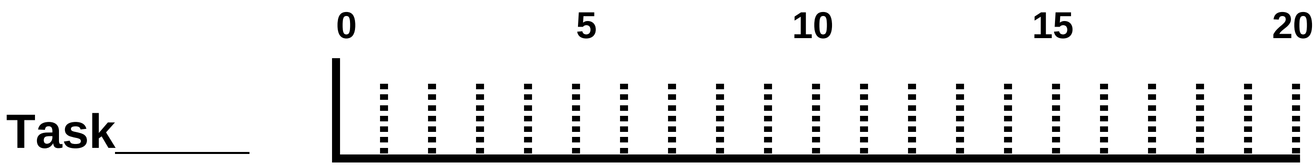
a) Apply UB test

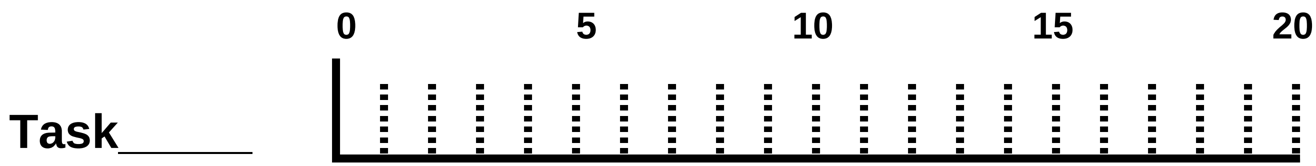
b) Draw timeline

c) Apply RT Test

Exercise: Worksheet







Summary

UB test is simple but conservative.

RT test is more exact but also more complicated.

To this point, UB and RT tests share the same limitations:

- **all tasks run on a single processor**
- **all tasks are periodic and noninteracting**
- **deadlines are always at the end of the period**
- **there are no interrupts**
- **rate monotonic priorities are assigned**
- **there is zero context switch overhead**
- **tasks do not suspend themselves**