Real-Time End-to-End Scheduling

Embedded System Software Design

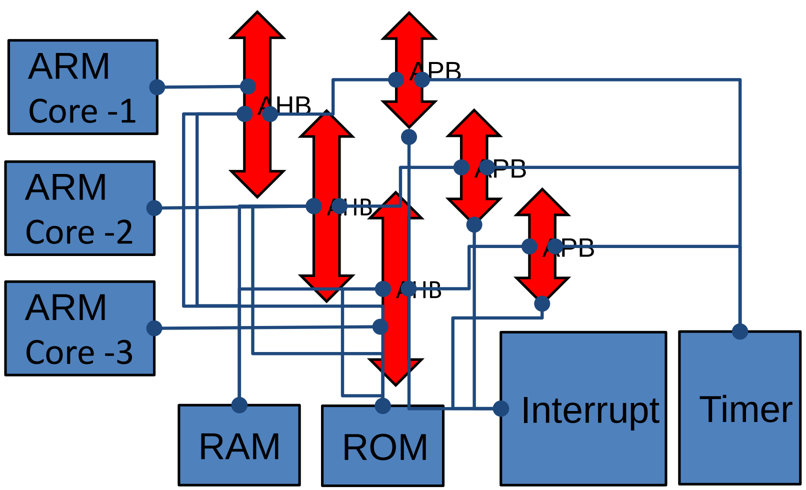
Prof. Ya-Shu Chen
National Taiwan University of
Science and Technology

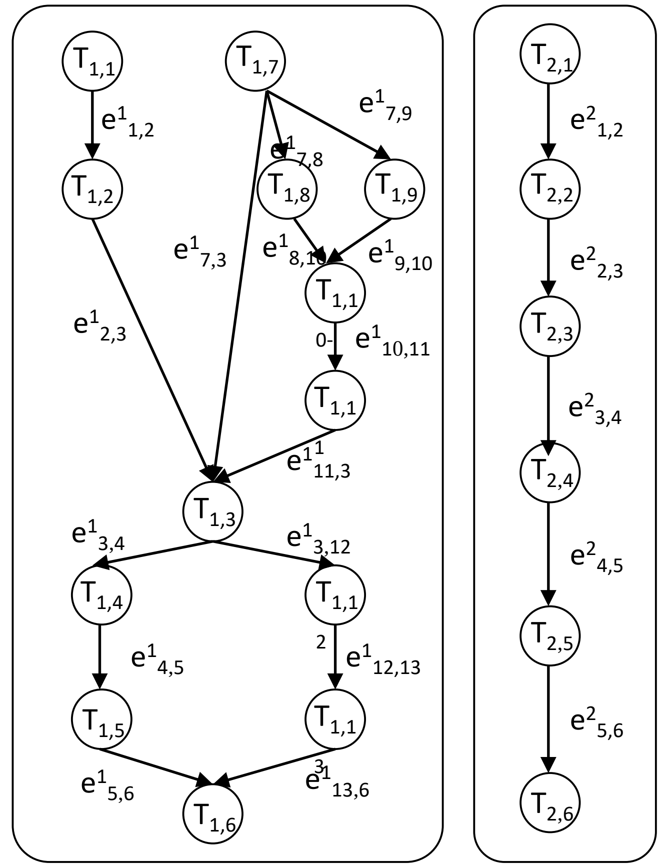
End-to-End Scheduling

- Task model
  - Each task needs to execute on a set of processors in a certain order
  - Each task may require a different order
- Problems in End-to-End scheduling
  - Priority assignment
    - Assign fixed priorities to tasks so that the system is schedulable
  - Synchronization of tasks
    - Control the releases of subtask instances (non-first subtasks)
  - Schedulability analysis
    - For a given priority assignment and a given synchronization protocol, whether every instance of each task meets its deadline

System Model

- Platform: A set of processors
- Task graph, $G = \{T_1, T_2, \ldots, T_n\}$
  - Sink node
  - Deadline: d_i
  - Precedence edge: $e^i_{j,k}$
  - Predecessors and Sucessors
  - Period or Minimum separation time: p_i
- Characteristics of $T_{i,j}$:
  - Execution time (on processor m): $c^m_{i,j}$





T_1, p_1, d_1 $\qquad$ T_2, p_2, d_2

Priority Assignment

- To find feasible priority assignments off-line if all tasks executed are known in prior
- NP-hard problem
- Algorithms
  - Branch and bound
  - Search algorithm
    - Simulated annealing
    - Generic algorithm
  - Heuristic
    - Deadline assignment

Deadline Assignment

- Ultimate deadline
  - $UD_{i,k} = D_i$
- Effective deadline
  - $ED_{i,k} = D_i - \sum_{l=k+1}^{n(i)} e_{i,l}$
- Proportional deadline
  - $PD_{i,k} = D_i e_{i,k} / e_i$
- Normalized Proportional deadline
  - $NPD_{i,k} = D_i \frac{e_{i,k} U(V_{i,k})}{\sum_{l=1}^{n(i)} e_{i,l} U(V_{i,l})}$
  - $U(V_{i,l})$ is the total utilization of the all the subtasks that execute on the processor $V_{i,l}$

Example

$T_{i,k}$	$V_{i,k}$	p_i	$e_{i,k}$	$UD_{i,k}$	$ED_{i,k}$	$PD_{i,k}$	$NPD_{i,k}$
$T_{1,1}$	$\boldsymbol{P_1}$	15	1	15	11	3	2.0
$T_{1,3}$	$\boldsymbol{P_1}$	15	2	15	15	6	4.1
$T_{2,1}$	$\boldsymbol{P_1}$	20	4	20	20	20	20
$T_{3,1}$	$\boldsymbol{P_2}$	2	1	2	2	2	2
$T_{1,2}$	$\boldsymbol{P_2}$	15	2	15	13	6	8.9
$T_{4,1}$	$\boldsymbol{P_2}$	20	5	20	20	20	20

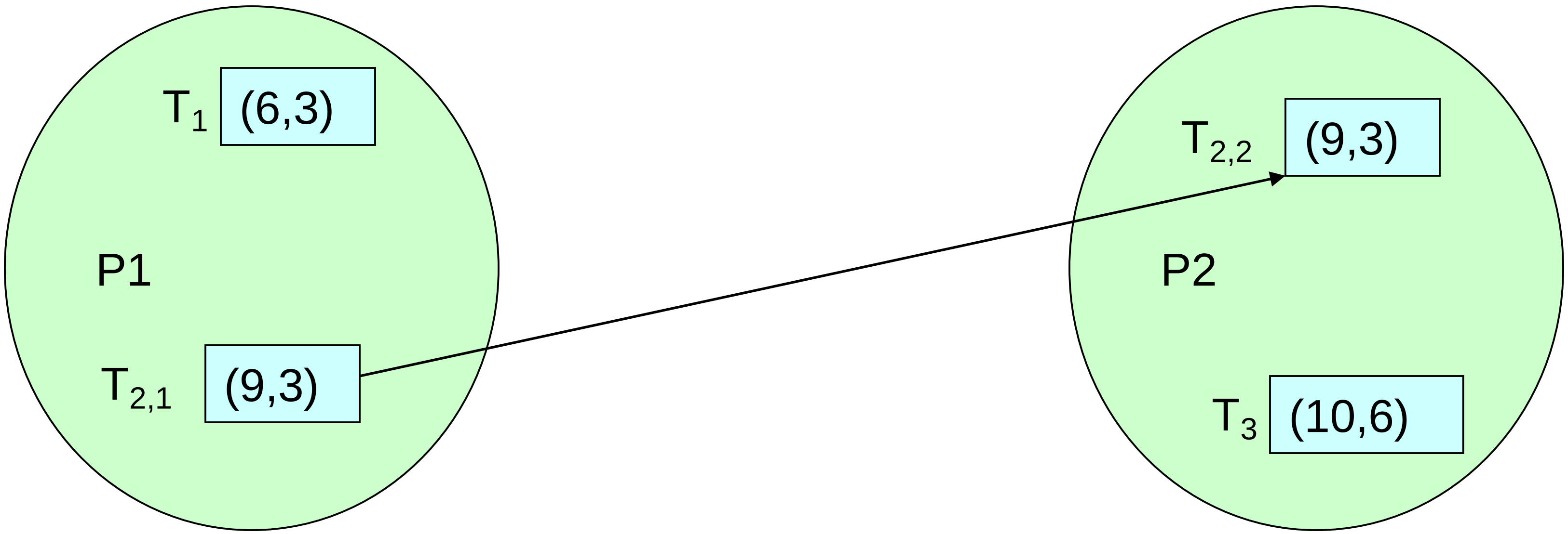
The Synchronization Problem

- Given that
  - Priorities are assigned to subtasks in a task chain using some fixed priority assignment algorithm
- How do we coordinate the release of subtasks in a task chain so that
  - Precedence constraints among subtasks are satisfied
  - Subtask deadlines are met
  - End-to-end deadlines are met

Synchronization Protocols

- Direct Synchronization (DS) Protocol
  - Simple and straightforward
- Phase Modification (PM) Protocol
  - Used by flow-shop tasks
  - Extension called Modified Phase Modification (MPM) Protocol
- Release Guard Protocol
  - Reclaim the idle time

Example



$T_{i,j}$ – j^{th} subtask of task T_i

Task T3 releases at 6

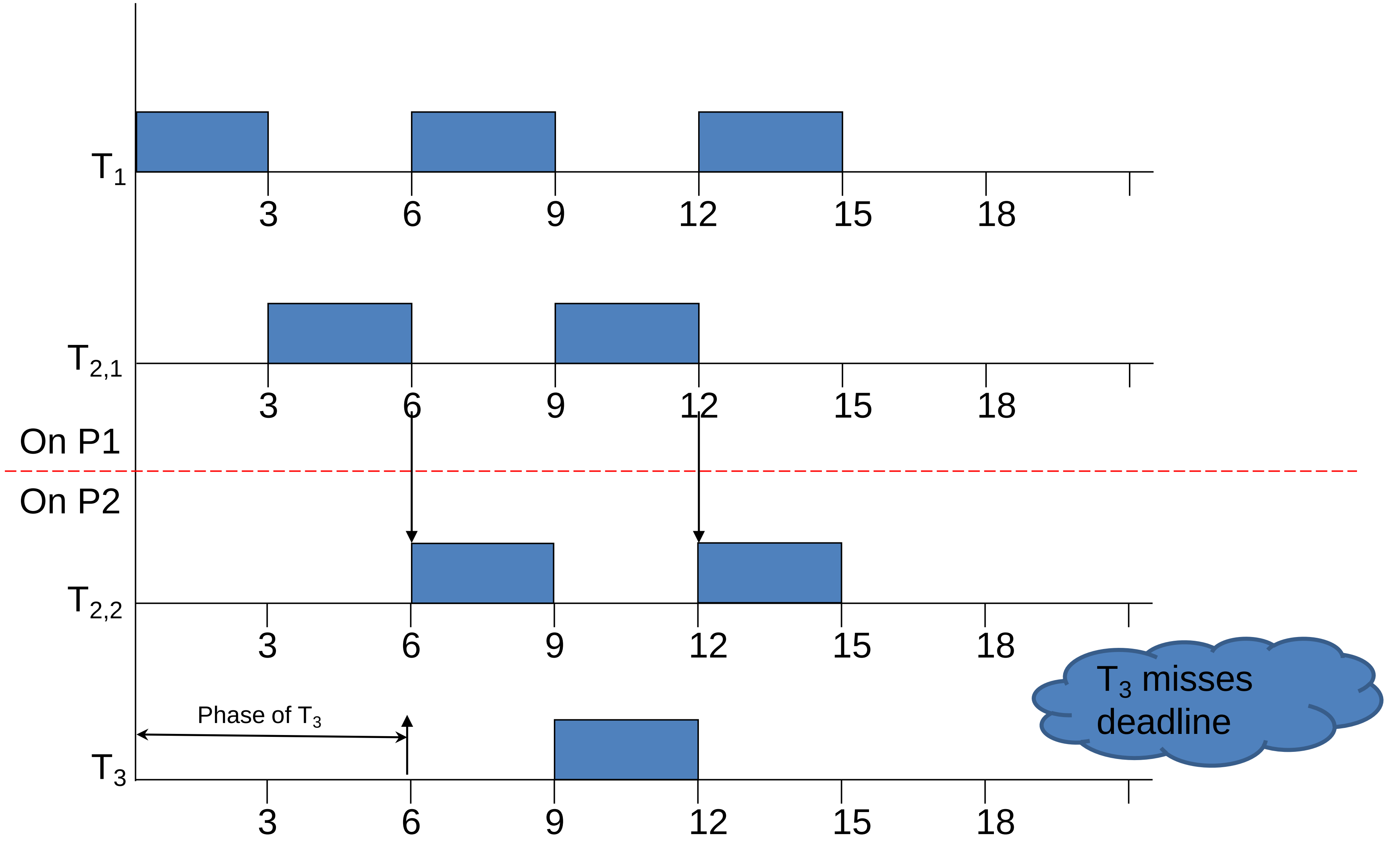
(period,execution time)

Period = relative deadline of parent task

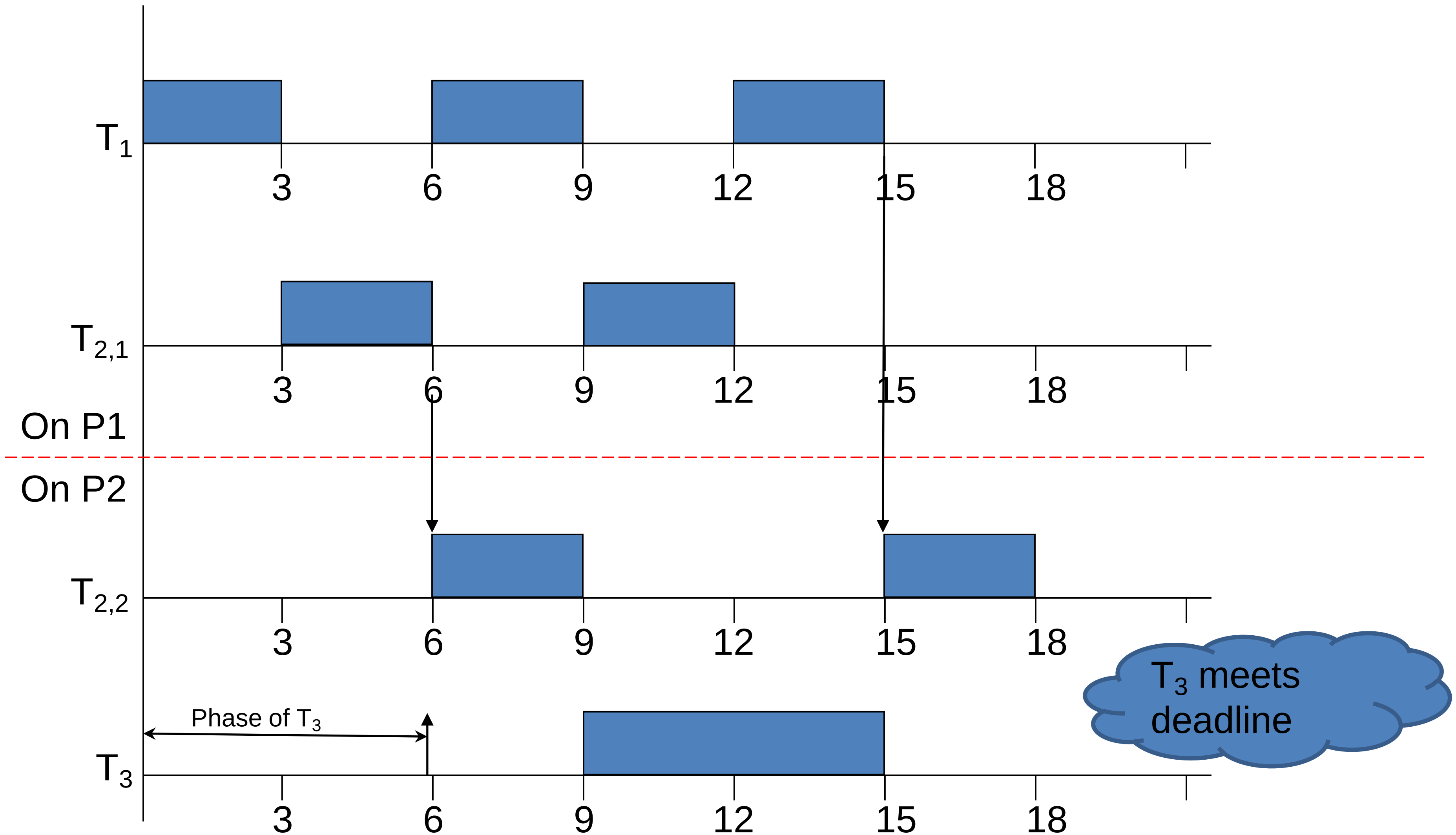
Direct Synchronization Protocol

- Greedy strategy
- On completion of subtask
  - A synchronization signal sent to the next processor
  - Successor subtask competes with other tasks/subtasks on the next processor

Greedy Example



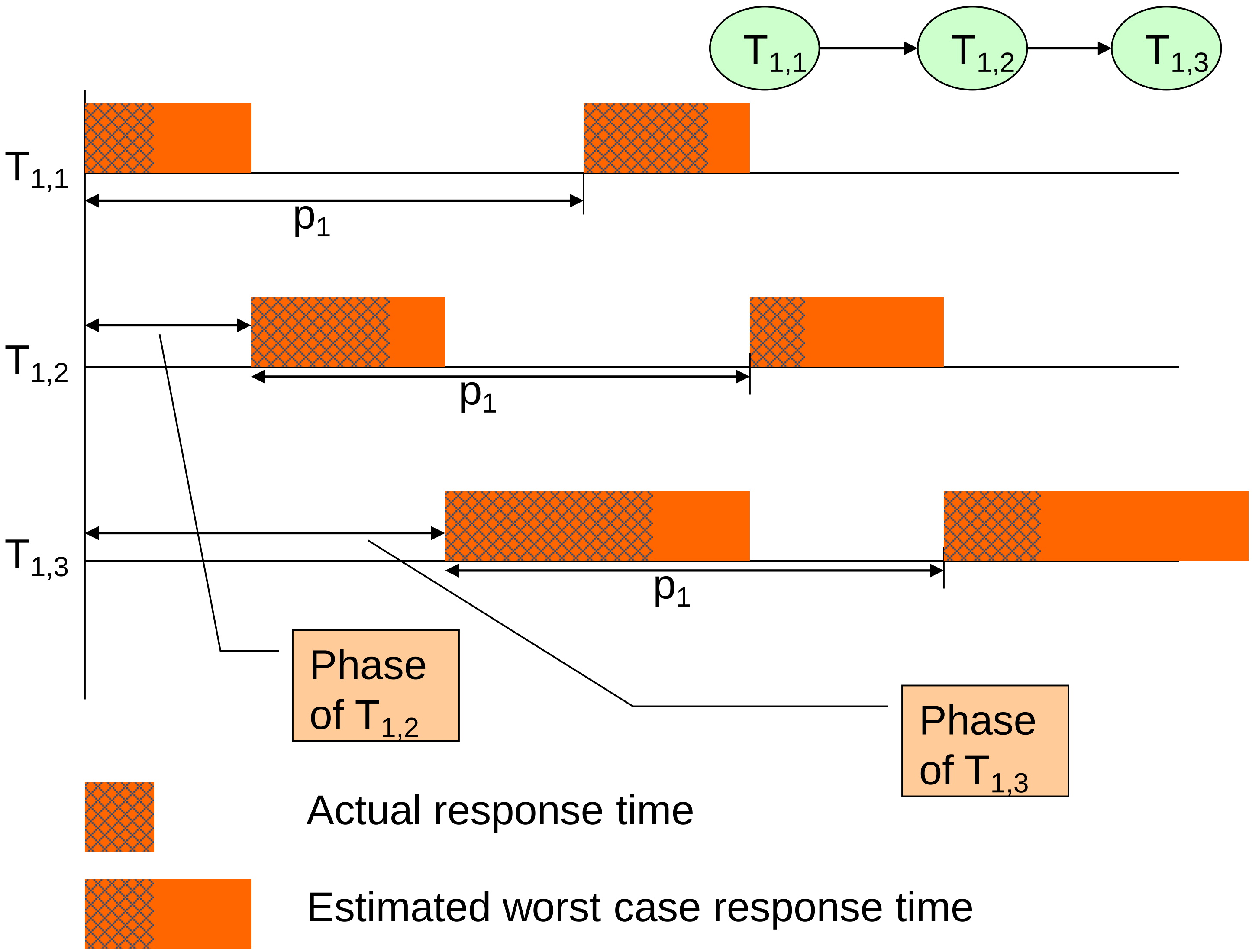
Non-greedy Example



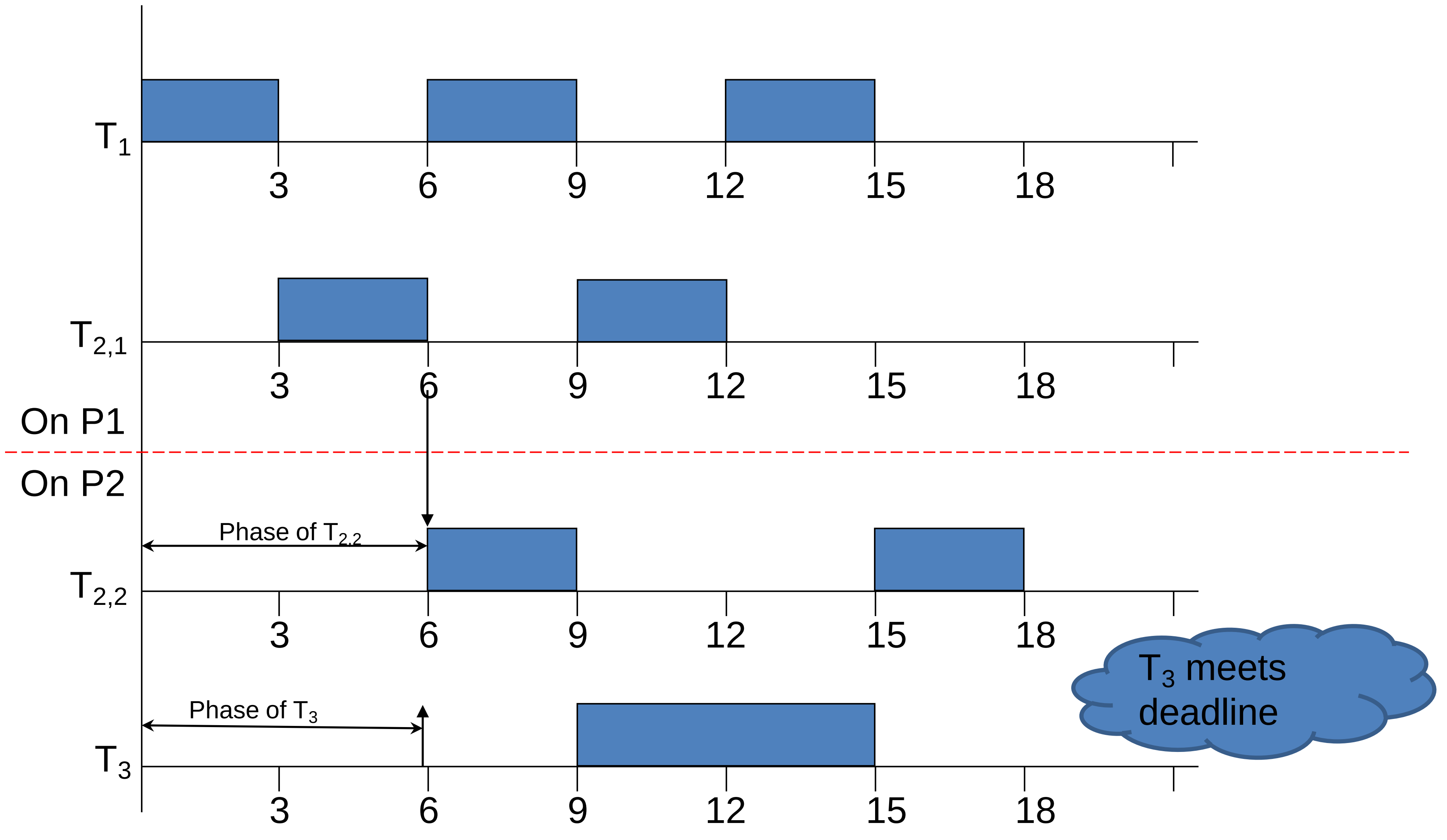
Phase Modification Protocol

- Release subtasks periodically
  - According to the periods of their parent tasks
- Each subtask given its own phase
- Phase determined by subtask precedence constraints

Phase Modification Protocol (1/2)



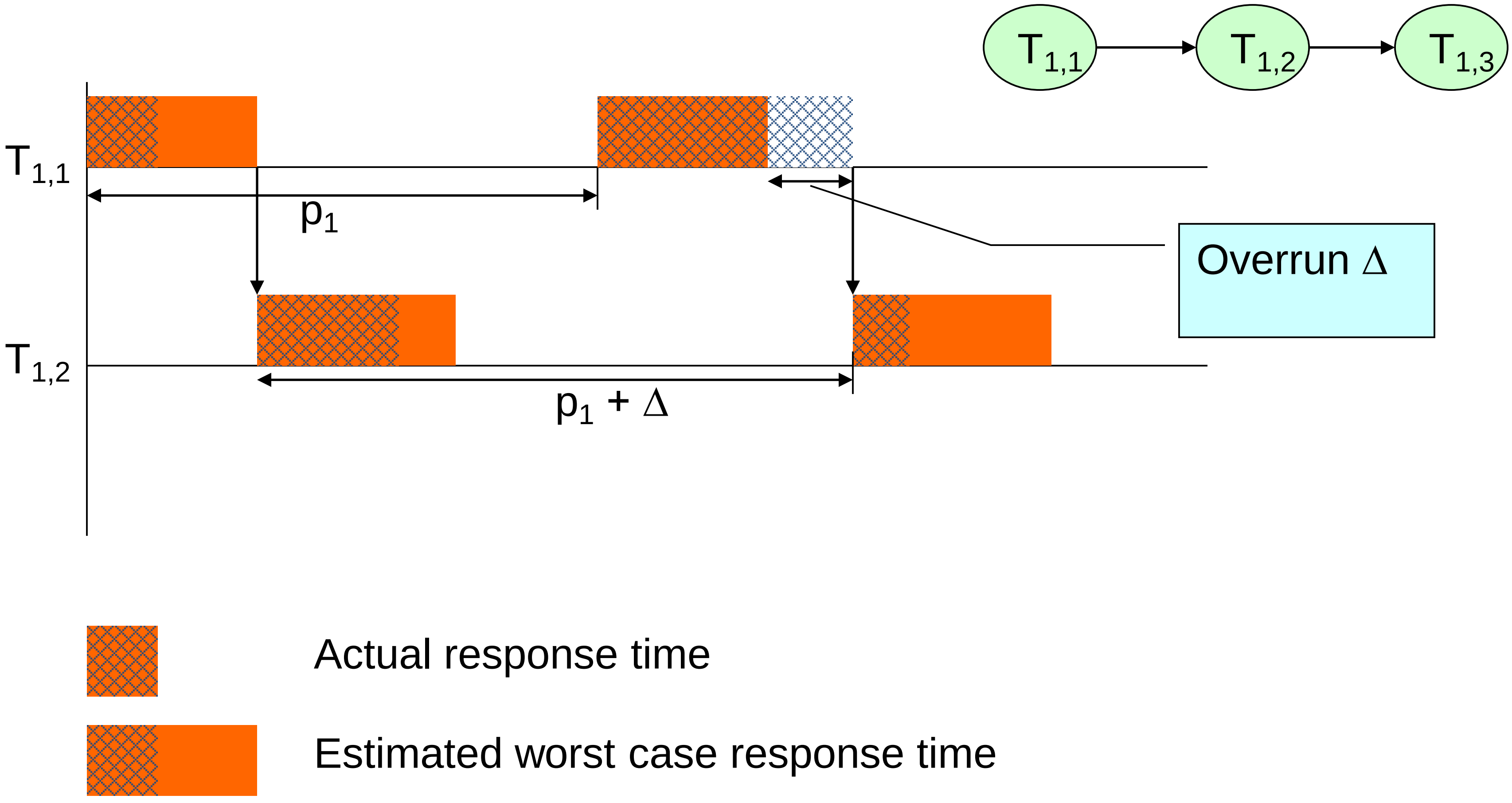
Phase Modification Protocol (2/2)



Phase Modification Protocol - Analysis

- Periodic timer interrupt to release subtasks
- Centralized clock or strict clock synchronization
- Task overruns could cause precedence constraint violations

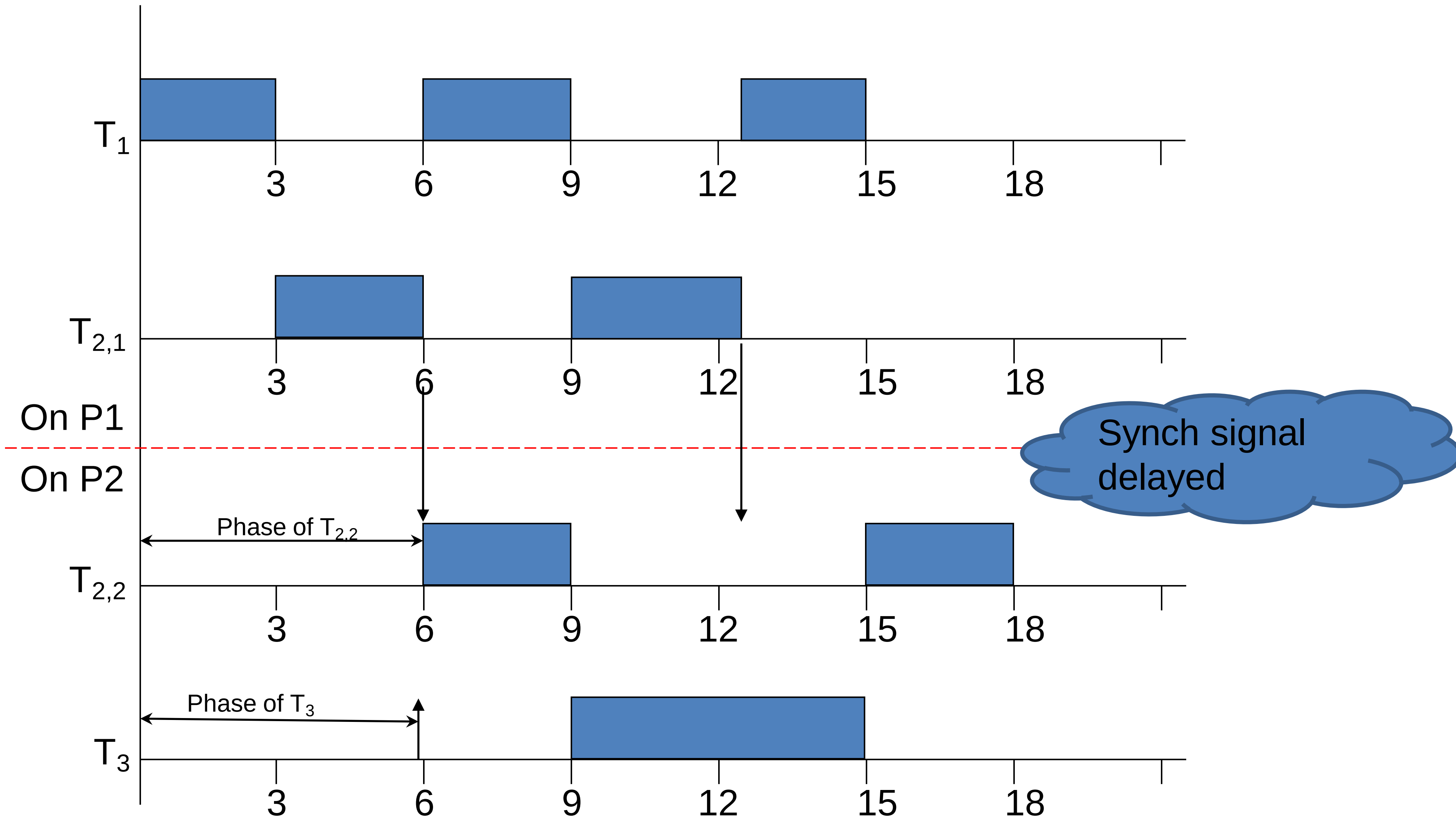
Modified PM Protocol (1/2)



Actual response time

Estimated worst case response time

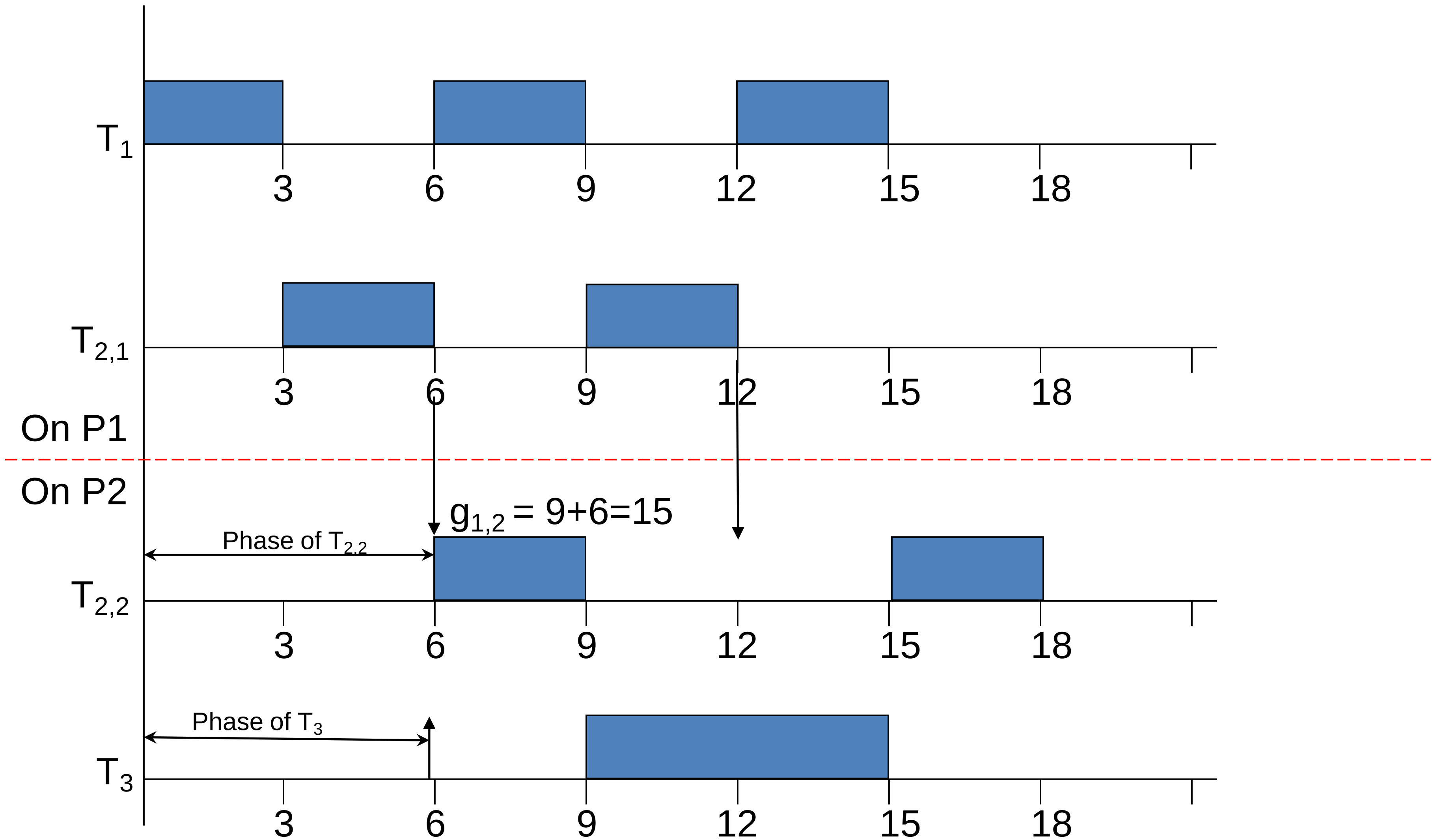
Modified PM Protocol (2/2)



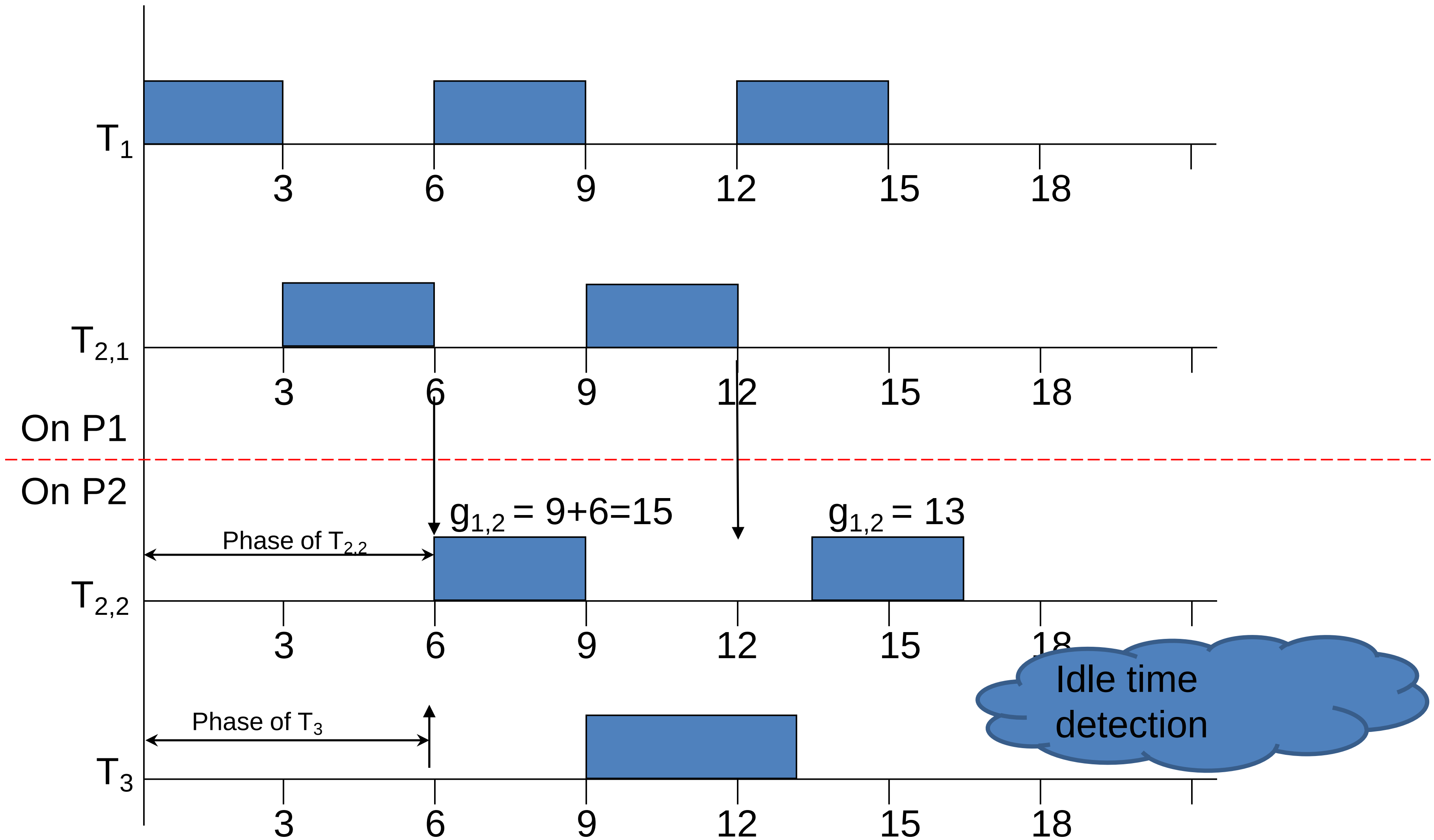
Release Guard Protocol

- A guard variable – *release guard* - associated with each subtask
- Release guard used to control release of each subtask
  - Contains next release time of subtask
- Synchronization signals as MPM
- Release guard updated
  - On getting synchronization signal
  - During idle time

Release Guard Protocol



Release Guard Protocol



Release Guard Protocol - Analysis

- Shares the same advantages as MPM
- Upper bound on EER still the same as MPM
  - Since upper bound on release time enforced by release guard
- Lower bound on EER less than that of MPM
  - If there are idle times
  - Results in lower average EER (end-to-end response time)

Schedulability Analysis

An upper bound W_i to the end-to-end response time of any periodic task T_i in a fixed-priority system synchronized according to the MPM protocol or the RG protocol is given by

$$W_i = \sum_{k=1}^{n(i)} W_{i,k}$$

and

$$W_{i,k} = \frac{e_{i,k} + b_{i,k} + \sum_{\phi_{j,l} \le \phi_{i,k} \text{ and } \tau_{j,l} \in V_{i,k}} e_{j,l}}{1 - \sum_{\phi_{j,l} < \phi_{i,k} \text{ and } \tau_{j,l} \in V_{i,k}} u_{j,l}}$$

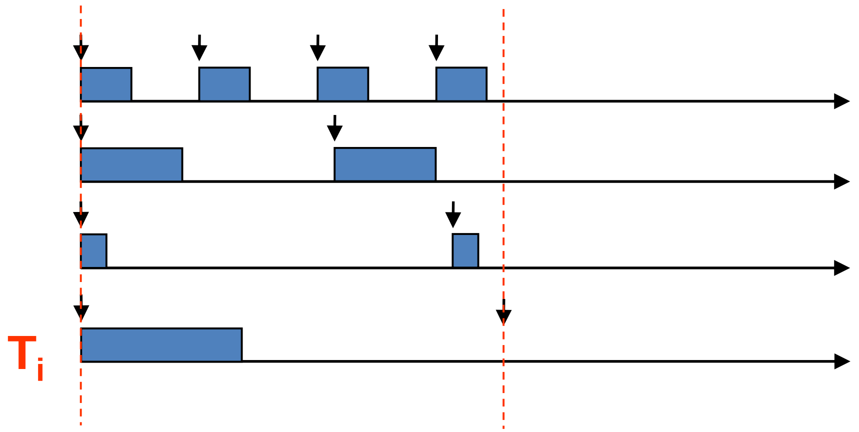
where $n(i)$ is the number of subtasks in T_i, $\phi_{i,k}$ is the priority of $\tau_{i,k}$, and the upper bound $W_{i,k}$ to the response time of every subtask $T_{i,k}$ is obtained by considering only subtasks on the same processor $V_{i,k}$, and by treating every such subtask $T_{j,l}$ as periodic task whose period is equal to the period p_j of the parent task T_j.

General Scheduling Test (GST)

- Response time analysis
  - The response time of the job of Ti at critical instant can be calculated by the following recursive function

$$r_0 = \sum_{\forall i} c_i$$

$$r_n = \sum_{\forall i} c_i \left\lceil \frac{r_{n-1}}{p_i} \right\rceil$$



  - Observation: the sequence of r_x, x>=0 may or may not converge

General Scheduling Test (GST)

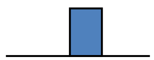
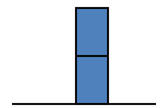
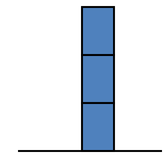
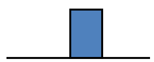
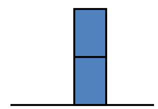
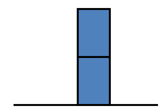
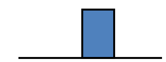
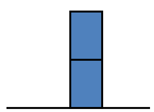
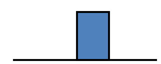
- Example: T1=(2,5), T2=(2,7), T3=(3,8)
  - T1:
    - $R_0=2 \le 5$ ok
  - T2:
    - $R_0=2+2=4 \le 7$
    - $R_1=2*\lceil 4/5 \rceil+2*\lceil 4/7 \rceil=4 \le 7$ ok
  - T3:
    - $R_0=2+2+3=7 \le 8$
    - $R_1= 2*\lceil 7/5 \rceil+2*\lceil 7/7 \rceil+3*\lceil 7/8 \rceil=9> 8$ failed

  - Note: each task succeeds→the task set succeeds

Example

$T_{i,k}$	$V_{i,k}$	p_i	$e_{i,k}$	$UD_{i,k}$	$b_{i,k}$	$W_{i,k}$	$W_{i,k}(GST)$
$T_{1,1}$	$\boldsymbol{P_1}$	15	1	15	0	3	3
$T_{1,3}$	$\boldsymbol{P_1}$	15	2	15	1	4	3(4)
$T_{2,1}$	$\boldsymbol{P_1}$	20	4	20	0	8.75	7
$T_{3,1}$	$\boldsymbol{P_2}$	2	1	2	0	1	1
$T_{1,2}$	$\boldsymbol{P_2}$	15	2	15	1	8	4(6)
$T_{4,1}$	$\boldsymbol{P_2}$	20	5	20	0	21.8	14

Comparison of Protocols

	DS	PM	MPM	RG
Implementation complexity	Synch interrupts	Timer interrupts clock synchronization	Synch & timer interrupts	Synch & timer interrupts
Run-time overhead				
Average EER				
Estimated worst case EER				
Inherently missed deadlines	Yes	No		

Reference

- Real-time Systems, Jane Liu
- Bettati,R.,``End-to-end scheduling to meet deadlines in distributed systems,” Ph.D. thesis, University of Illinois at Urbana-Champaign
- Sun, J., `` Fixed-Priority Scheduling of Periodic Tasks With End-to-End Deadlines,” Ph.D. thesis, University of Illinois at Urbana-Champaign