05

Real-Time End-to-End Scheduling

做下一件事要等其他事一起完成

Embedded System Software Design

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

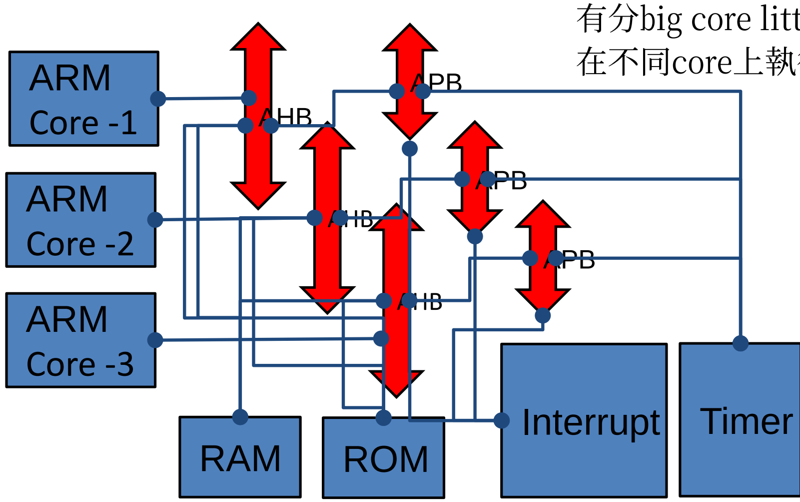
End-to-End Scheduling

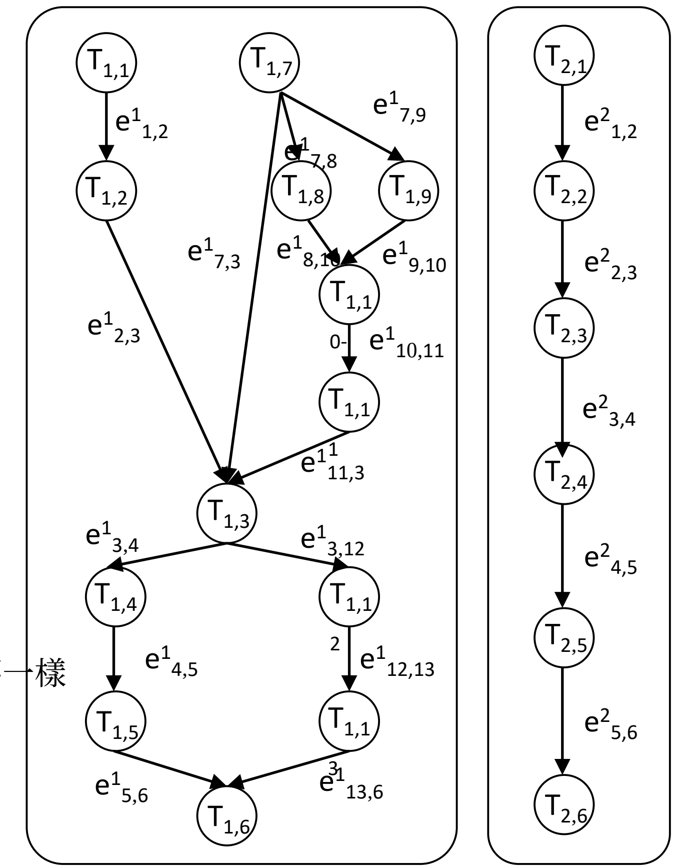
- Task model
  - Each task needs to execute on a set of processors in a certain order
  - Each task may require a different order
- Problems in End-to-End scheduling
  - Priority assignment 優先分配
    - Assign fixed priorities to tasks so that the system is schedulable
  - Synchronization of tasks 任務同步
    - Control the releases of subtask instances (non-first subtasks)
  - Schedulability analysis 可調度性分析
    - 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): $c^m_{i,j}$

有分big core little core
在不同core上執行時間可能不一樣





T_1, p_1, d_1 T_2, p_2, d_2

只要找到最長的路徑
能排就能排

Priority Assignment

- To find feasible priority assignments off-line if all tasks executed are known in prior

同一條路優先權怎麼設
不一定 這有很多作法

- NP-hard problem

沒有定論 哪一個表現特別好

- Algorithms
  - Branch and bound
  - Search algorithm
    - Simulated annealing
    - Generic algorithm
  - Heuristic
    - Deadline assignment

Deadline Assignment

- Ultimate deadline
  - $UD_{i,k} = D_i$ 期限
- Effective deadline
  - $ED_{i,k} = D_i - \sum_{l=k+1}^{n(i)} e_{i,l}$ 期限 - 同T比他低的執行時間
- Proportional deadline
  - $PD_{i,k} = D_i e_{i,k} / e_i$ 期限 * 執行時間 / 總執行時間
- Normalized Proportional deadline
  - $NPD_{i,k} = D_i \frac{e_{i,k} U(V_{i,k})}{\sum_{l=1}^{n(i)} e_{i,l} U(V_{i,l})}$ (執行時間 * U) / (總執行時間 * U)

    整個T的執行時間*U 總和
  - $U(V_{i,l})$ is the total utilization of the all the subtasks that execute on the processor $V_{i,l}$

U = e / d

Example

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

UD = D
UD11=15
UD13=15
UD21=20

ED11=15-(2+2)=11
ED12=15-(2)=13
ED13=15
ED21=20
ED31=2
ED41=20

PD11=(15*1)/(1+2+2)=3
PD13=(15*2)/(1+2+2)=6
PD12=(15*2)/(1+2+2)=6
PD21=(20*4)/(4)=20
PD31=(2*1)/(1)=2
PD41=(20*5)/(5)=20

U=E/D
Up1=1/15+2/15+4/20=0.4
Up2=1/2+2/15+5/20=0.88

NPD11=15*(1*0.4)/(1*0.4+2*0.88+2*0.4)=2
NPD13=15*(2*0.4)/(1*0.4+2*0.88+2*0.4)=4.1
NPD21=20*(4*0.4)/(4*0.4)=20
NPD31=2*(1*0.88)/(1*0.88)=2
NPD12=15*(2*0.88)/(1*0.4+2*0.88+2*0.4)=8.9
NPD41=20*(5*0.88)/(5*0.88)=20

The Synchronization Problem

同步問題

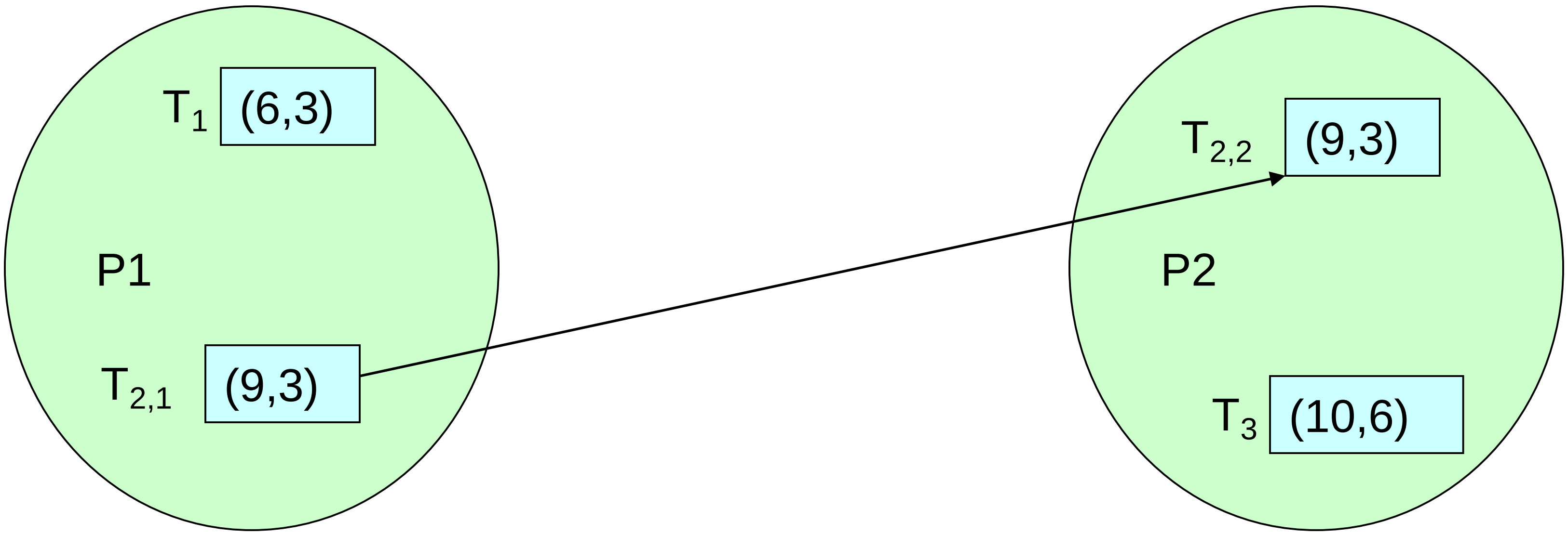
- Given that
  - Priorities are assigned to subtasks in a task chain using some fixed priority assignment algorithm
- How do we coordinate the release of subtasks in a task chain so that
  - Precedence constraints among subtasks are satisfied
  - Subtask deadlines are met
  - End-to-end deadlines are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

Synchronization Protocols

同步協議

- Direct Synchronization (DS) Protocol 做完呼叫 繼續做
  - Simple and straightforward 簡單直覺
- Phase Modification (PM) Protocol 相位修改
  - Used by flow-shop tasks
  - Extension called Modified Phase Modification (MPM) Protocol
- Release Guard Protocol 釋放
  - Reclaim the idle time 回收 idle 時

Example



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

Task T3 releases at 6

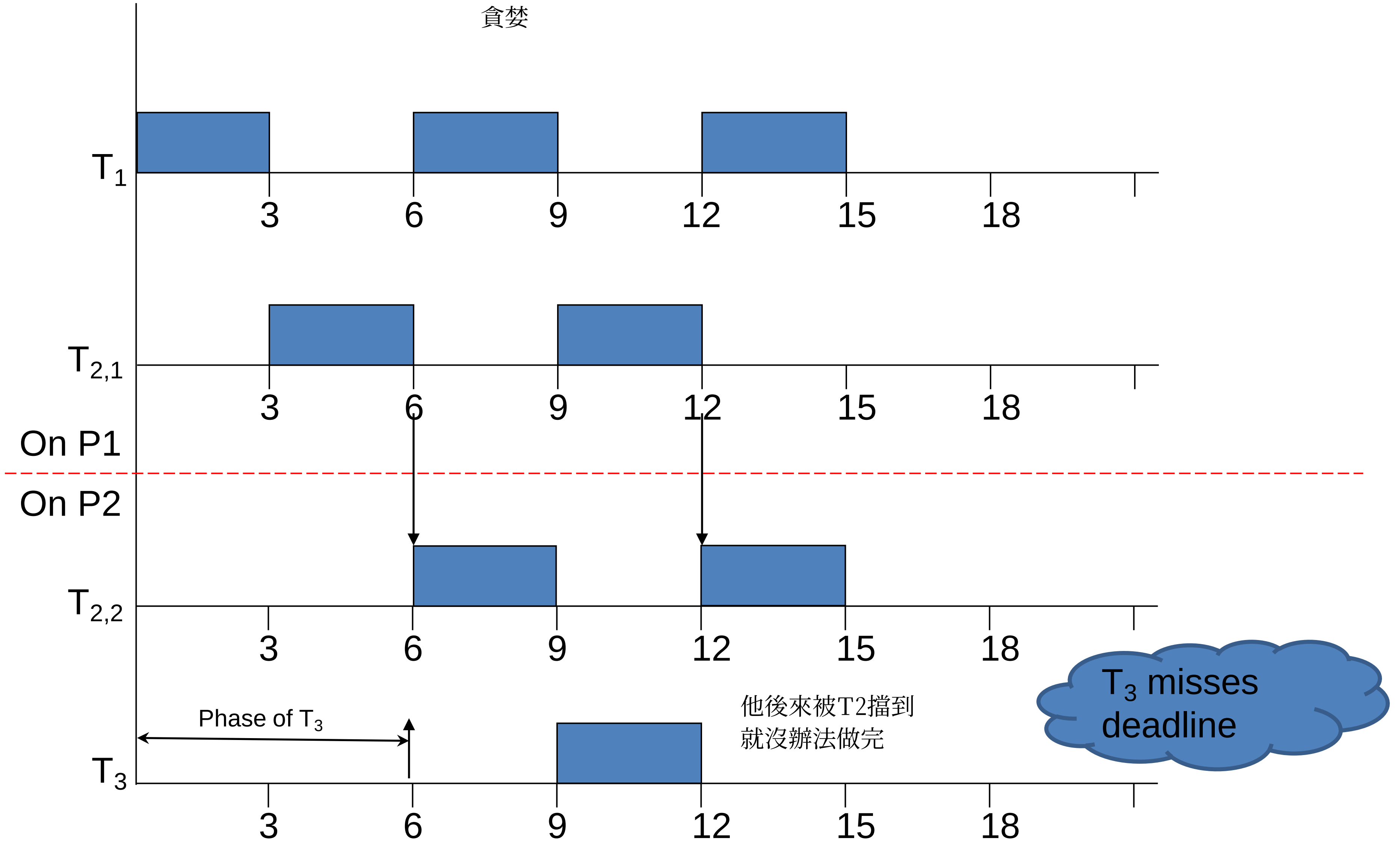
(period,execution time)

Period = relative deadline of parent task

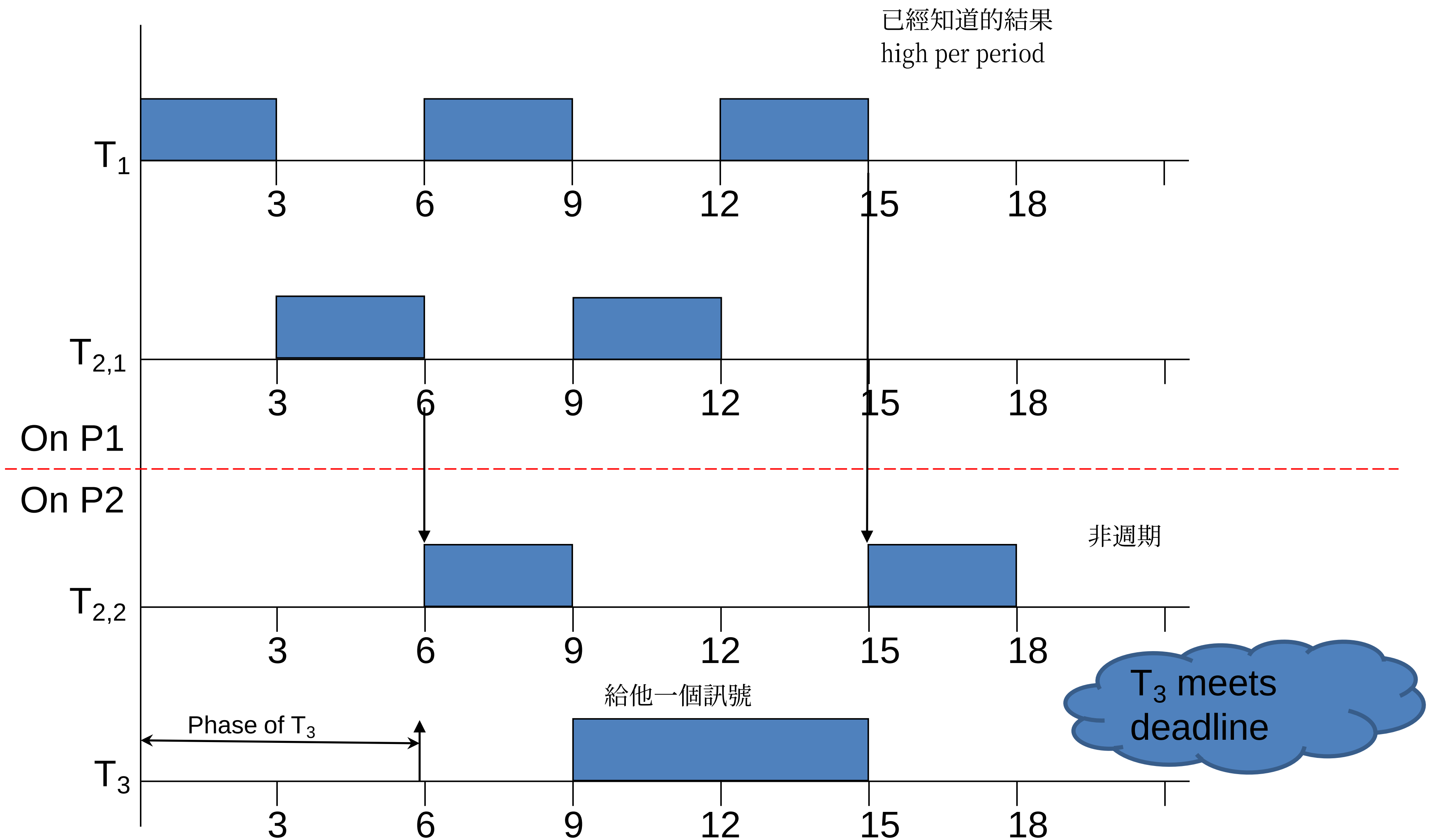
Direct Synchronization Protocol

- 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



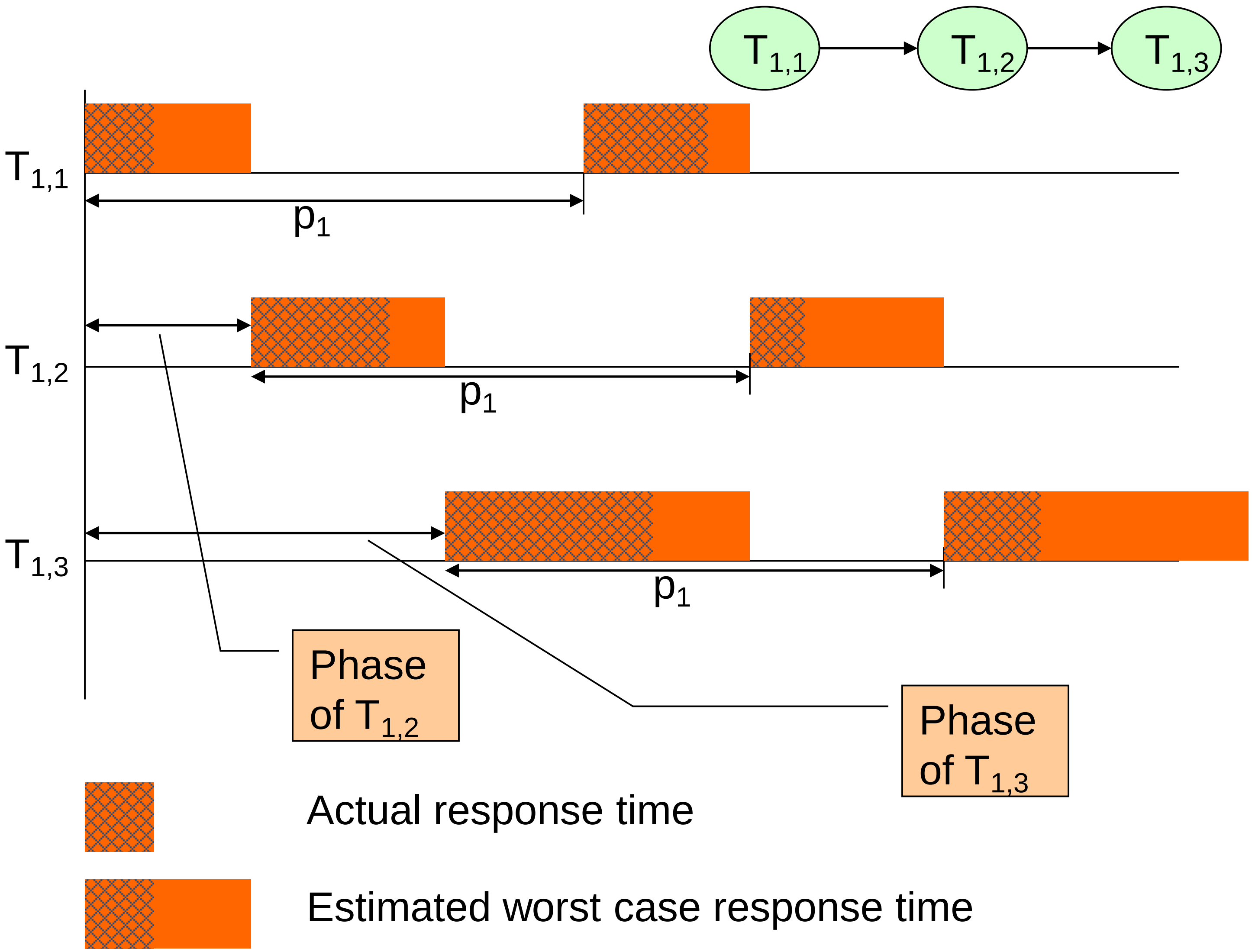
Non-greedy Example



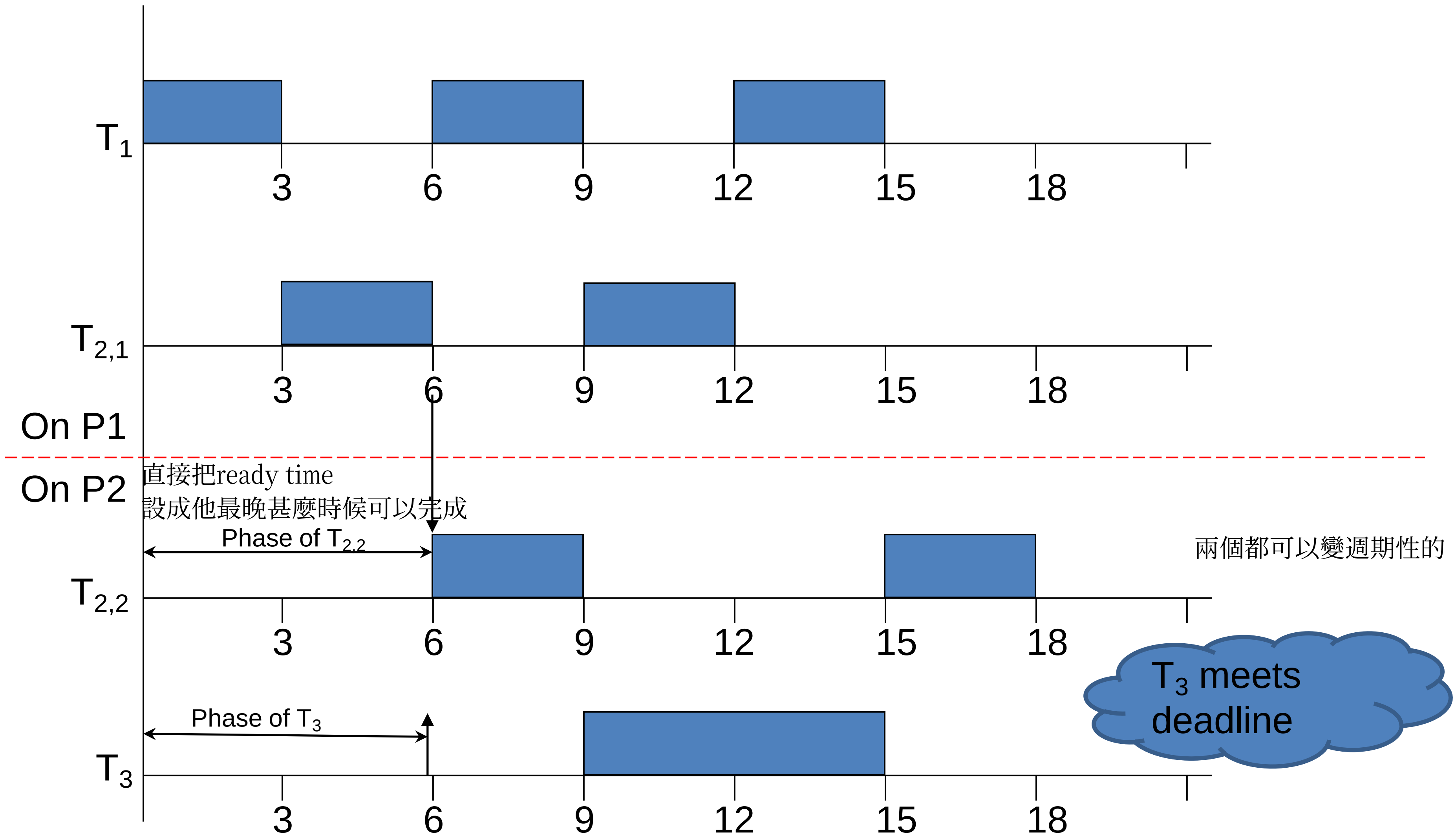
Phase Modification Protocol

- Release subtasks periodically 定期釋放子任務
  - According to the periods of their parent tasks 根據其上級任務的期限
- Each subtask given its own phase 每個子任務都有自己的階段
- Phase determined by subtask precedence constraints 階段由子任務優先級確定約束條件

Phase Modification Protocol (1/2)



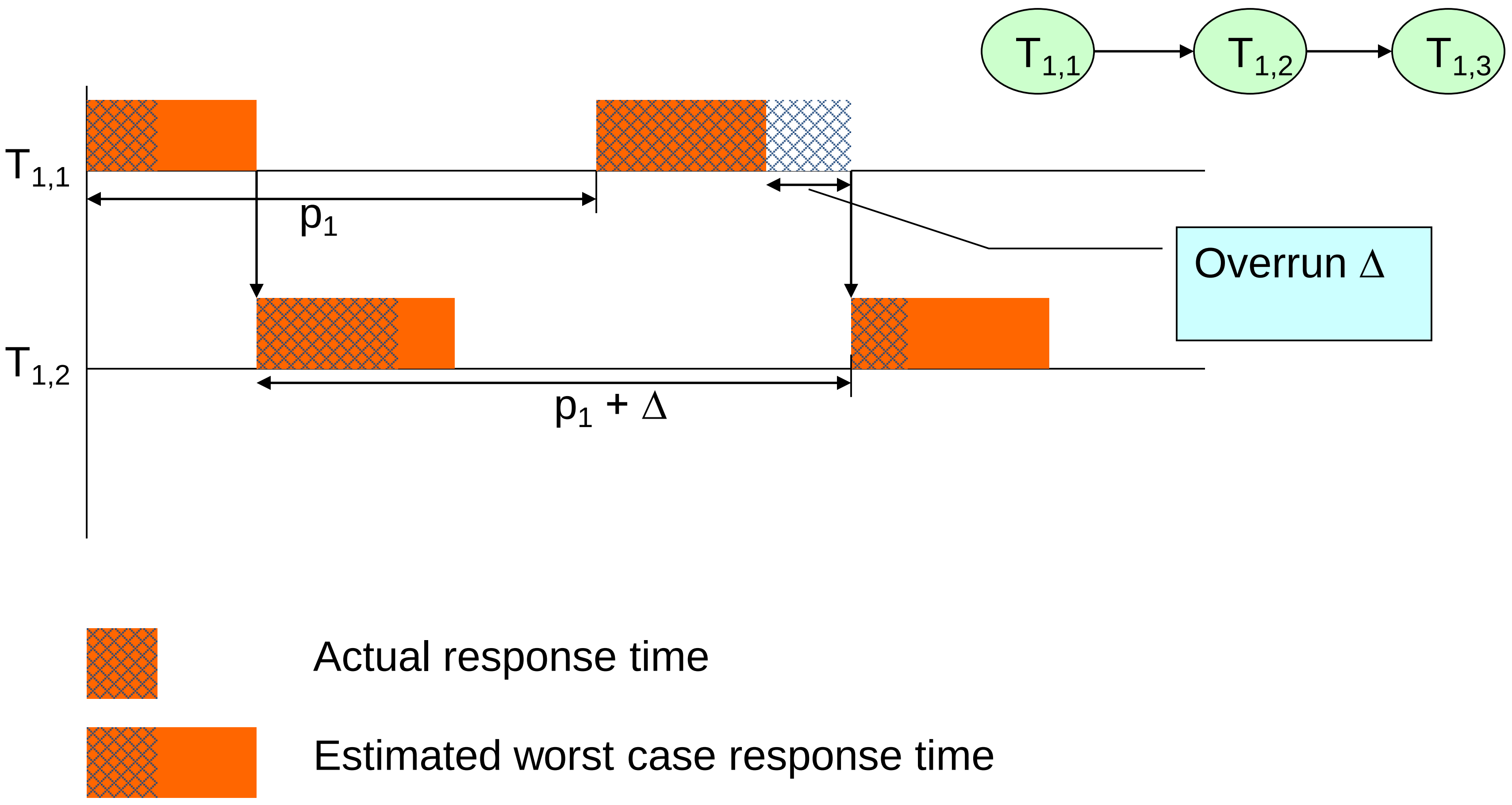
Phase Modification Protocol (2/2)



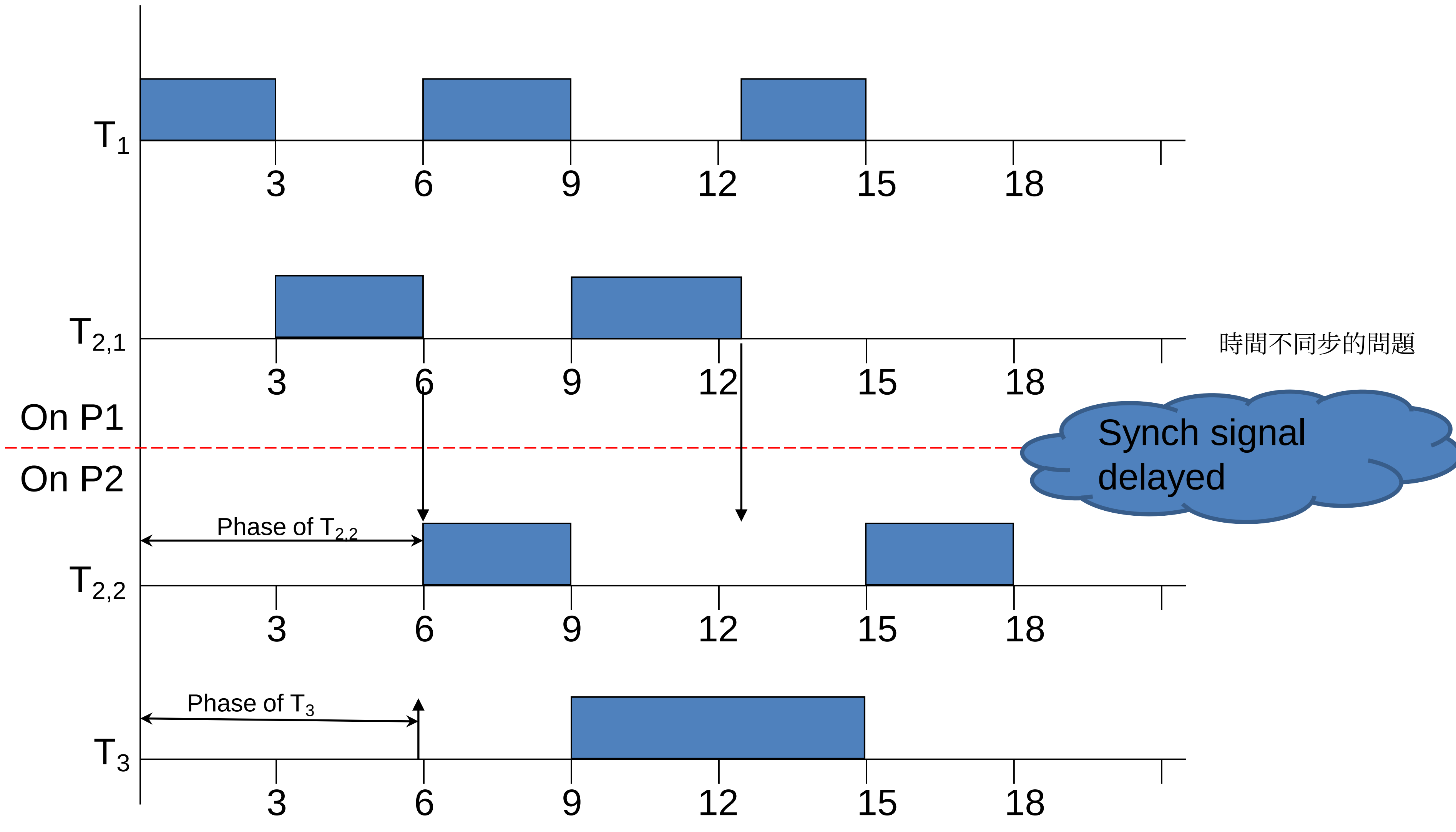
Phase Modification Protocol - Analysis

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

Modified PM Protocol (1/2)



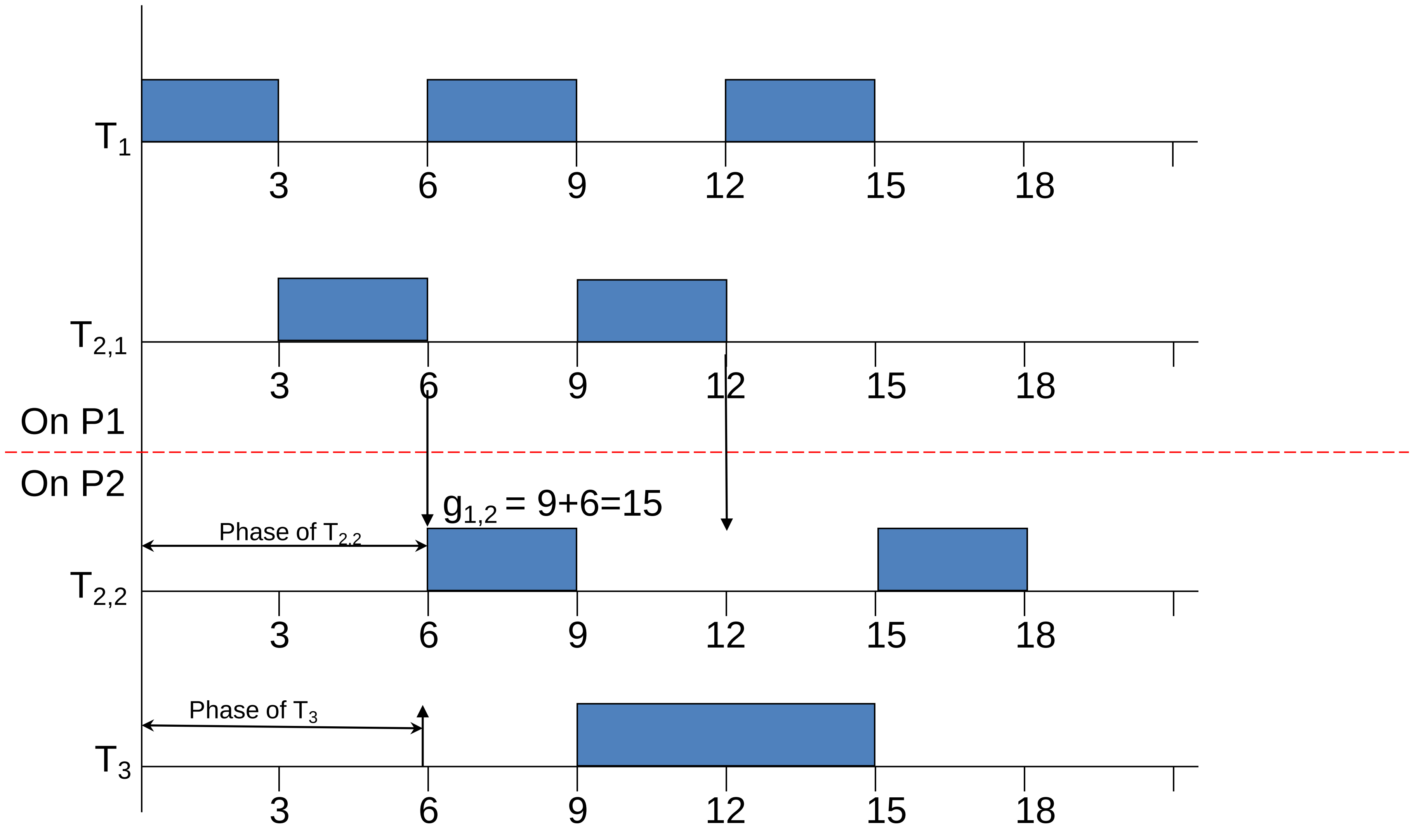
Modified PM Protocol (2/2)



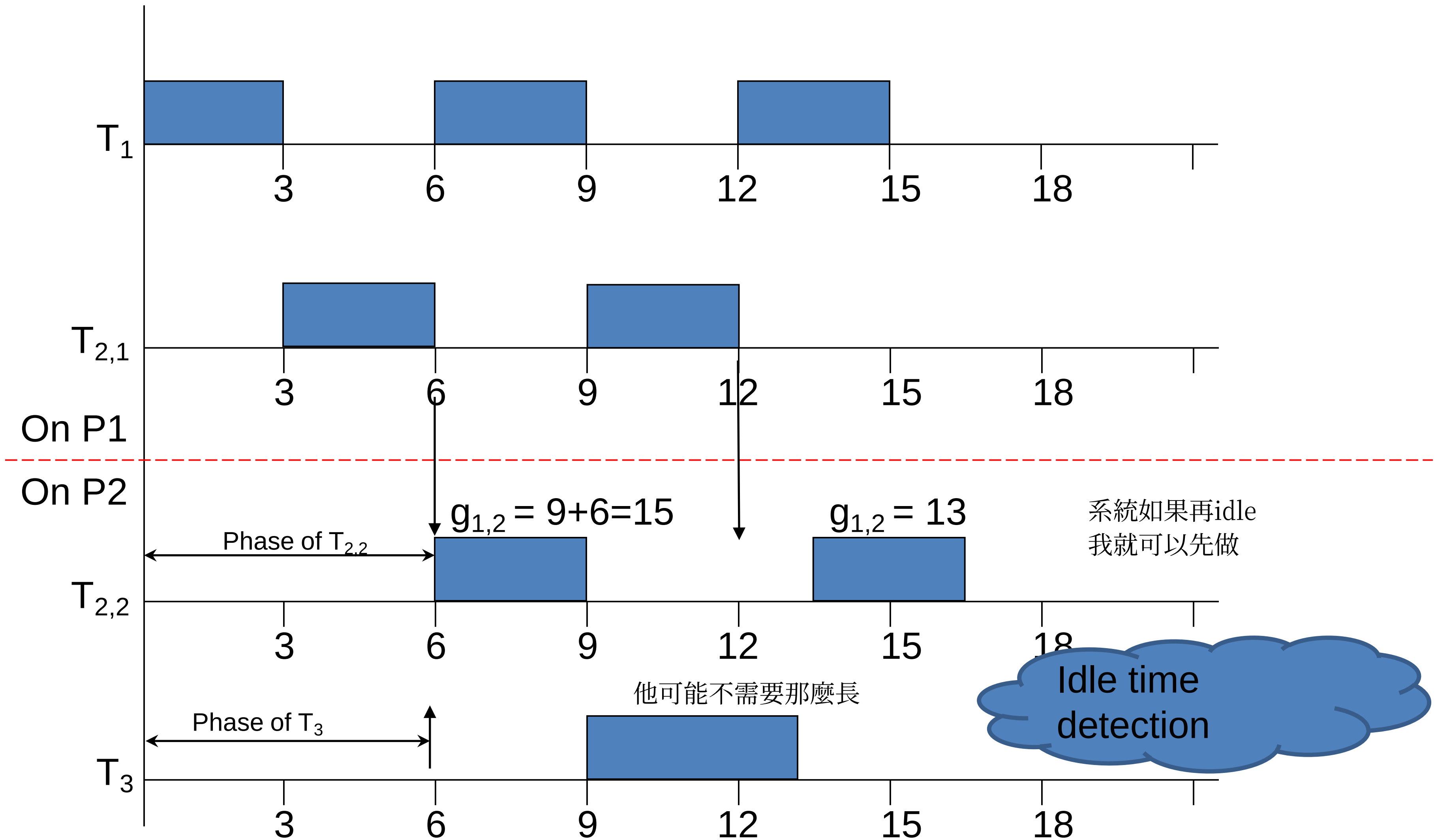
Release Guard Protocol

- 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

可排程分析

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

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

比我低block我的影響

自己執行時間　　　優先比我>=對我的影響

and

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

同一個processer上

1-優先比我高的人的U
通通不用比我高的資源

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.

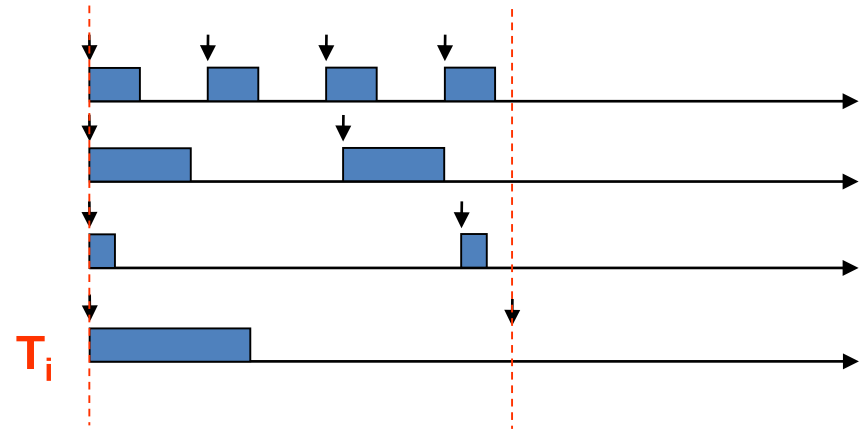
比較悲觀

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

W12=(1+0+2)/(1-0)=3
W13=(2+1+1)/(1-0)=4
W21=(4+0+(1+2))/(1-1/15-2/15)=8.75

W31=(1+0+0)/(1-0)=1
W12=(2+1+1)/(1-1/2)=8
W41=(5+0+(1+2))/(1-1/2-2/15)=21.8

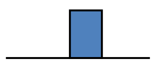
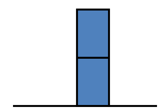
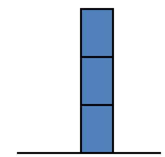
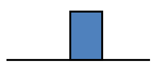
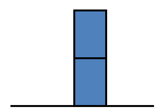
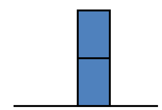
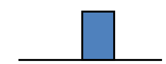
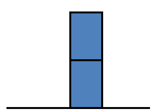
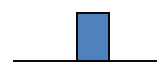
W12 R0=1+2=3
W12 R1=1*ceil(3/2)+2*ceil(3/15) = 4

W12b R0=1+2+1=4
W12b R1=1*ceil(4/2)+3*ceil(4/15) = 5
W12b R2=1*ceil(5/2) + 3*ceil(5/15) = 6

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

W41 R0=1+2+5=8
W41 R1=1*ceil(8/2)+2*ceil(8/15)+5*ceil(8/20) = 11
W41 R2=1*ceil(11/2)+2*ceil(11/15)+5*ceil(11/20)=13
W41 R3=1*ceil(13/2)+2*ceil(13/15)+5*ceil(13/20)=14
W41 R4=1*ceil(14/2)+2*ceil(14/15)+5*ceil(14/20)=14.3

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