MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Isae
Institut Supérieur de l'Aéronautique et de l'Espace
ENSMA

On the Existence of a Cyclic Schedule for Non-Preemptive Periodic Tasks with Release Offset

Mitra Nasri*   Emmanuel Grolleau

RTSOPS 2017

Motivations

Scheduling table is stored in memory

- Consumes energy and space
- Increases the total production cost

Arm Cortex MCU family

STM32 32-bit ARM Cortex MCUs | STM32F2 Series | STM32F3 Series | STM32F4 Series | STM32F7 Series

Total Parts: (752) for STM32 32-bit ARM Cortex MCUs | Matching Parts : (90)

Part Number	Package	Core	Operating Frequency (MHz) (Processor speed)	FLASH Size (kB) (Prog)	Internal RAM Size (kB)	I/Os (High Current)
STM32L011G4	UFQFPN 28 4x4 x0.55	ARM Cortex-M...	32	16	2	24
STM32L011K4	LQFP 32 7x7x1.4, ...	ARM Cortex-M...	32	16	2	28
STM32L021D4	TSSOP 14	ARM Cortex-M...	32	16	2	11
STM32L021F4	UFQFPN 20 3x3 x0.6	ARM Cortex-M...	32	16	2	16
STM32L021G4	UFQFPN 28 4x4 x0.55	ARM Cortex-M...	32	16	2	24
STM32L021K4	LQFP 32 7x7x1.4	ARM Cortex-M...	32	16	2	28
STM32L031F4	TSSOP 20	ARM Cortex-M...	32	16	8	15
STM32L071C8	LQFP 48 7x7x1.4	ARM Cortex-M...	32	64	20	37
STM32L071RZ	LQFP 64 10x10 x1.4, ...	ARM Cortex-M...	32	192	20	51
STM32L071VB	LQFP 100 14x14x1.4	ARM Cortex-M...	32	128	20	84

Finding an Observation Interval for Online Algorithms

System model and assumptions

- Uniprocessor
- **Non-preemptive**
- Periodic tasks
- Constrained deadline
- Independent tasks

We are interested in the approaches that build the table using an online algorithm

Work-conserving or Non-work-conserving?

- **What is the length of cyclic schedule?**
  - **Synchronous release**: one hyperperiod
  - **Asynchronous release**: depends on the scheduling algorithm

State-of-the-Art of Simulation Interval

Processor	Deadlines	Dependency	Scheduling algorithm	Simulation interval	Reference
1	$D_i \leq T_i$	Independent	Fixed-task priority	$[0, O^{\max} + 2H)$	Leung and Merrill (1980)
1	Arbitrary	Independent	Fixed-job priority	$[0, O^{\max} + 2H)$	Goossens and Devillers (1999)
1	$D_i \leq T_i$	Independent	Fixed-job priority	$[0, S_n + H)$	Goossens and Devillers (1997)
1	$D_i \leq T_i$	Mutual exclusion, simple precedence	Any work-conserving (with idle task)	$[0, \theta_c + H)$	Choquet-Geniet and Grolleau (2004), Bado et al. (2012)
Uniform	$D_i \leq T_i$	Independent	Global fixed-task priority	$[0, S_n + H)$	Cucu and Goossens (2006)
Unrelated	$D_i \leq T_i$	Independent	Global fixed-task priority	$[0, S_n + H)$	Cucu-Grosjean and Goossens (2011)
Identical	Arbitrary	Independent	Global fixed-task priority	$[0, \hat{S}_n + H)$	Cucu and Goossens (2007)
Identical	$D_i \leq T_i$	Independent	Any	$[0, O^{\max} + H\prod_{i=1}^{n}(C_i + 1))$	Baro et al. (2012), Nélis et al. (2013)
Identical	$D_i \leq T_i$	Simple precedence	Any	$[0, O^{\max} + H\prod_{i=1}^{n}(C_i + 1))$	Baro et al. (2012)
Identical	Arbitrary	Structural constraint	Any	$[0, H\prod_{i=1}^{n}((O_i + D_i - T_i)_0 + 1))$	Goossens et al. (2016)

Impractical!

$$\hat{s}_1 \doteq O_1$$

$$\hat{s}_i \doteq \max\left(O_i, O_i + \left\lceil \frac{\hat{s}_{i-1} - O_i}{T_i} \right\rceil T_i\right) + H_i$$

$$H_i \doteq \operatorname{lcm}_{j=1\ldots i}(T_j)$$

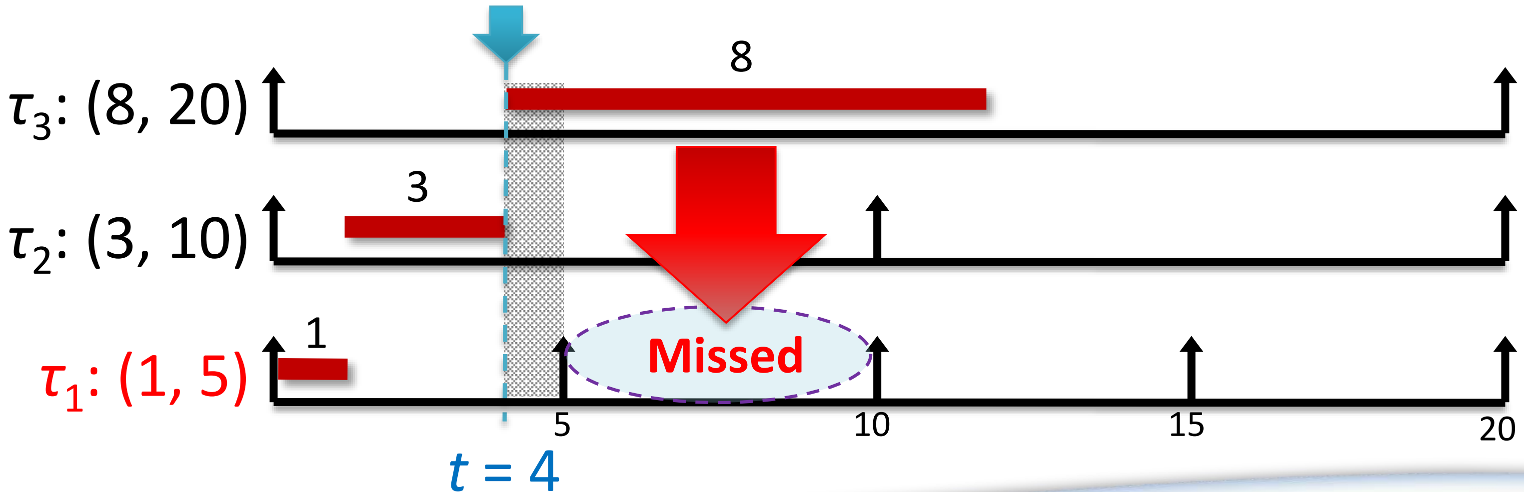
$$[0, H\prod_{i=1}^{n}((O_i + D_i - T_i)_0 + 1))$$

J. Goossens, E. Grolleau, L. Cucu-Grosjean, Periodicity of real-time schedules for dependent periodic tasks on identical multiprocessor platforms, Real-Time Systems, 2016.

Non-Work-Conserving Scheduling

- Leaves the processor idle even if there are tasks that are not yet scheduled

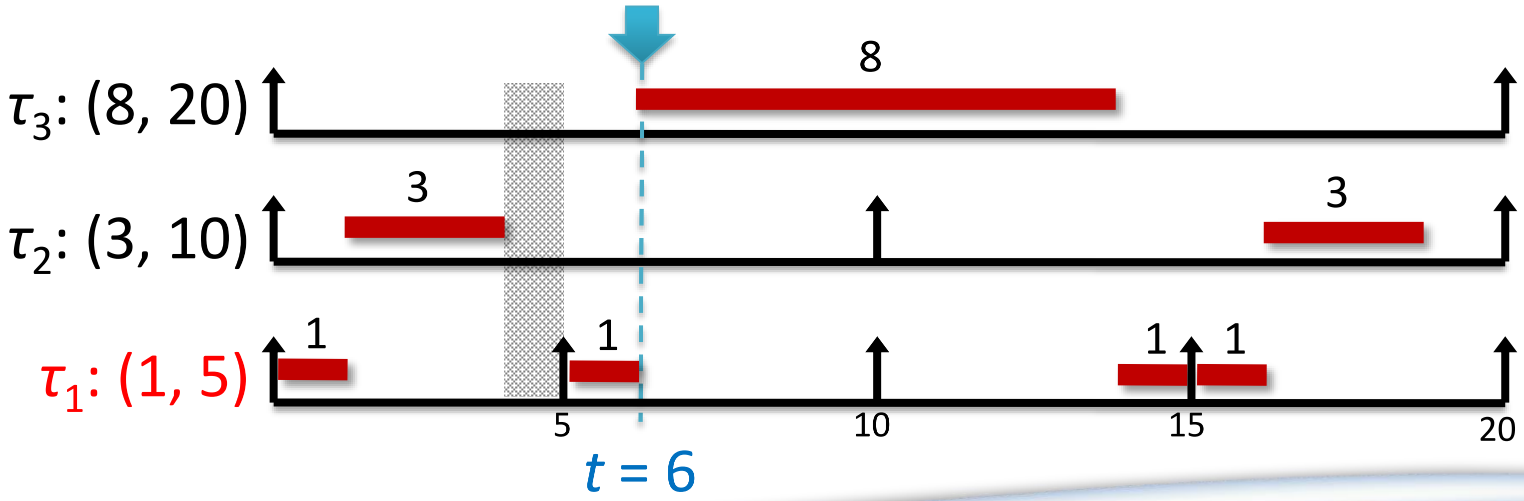
- Example: **Precautious-RM** [Nasri2014]
  - is an online algorithm
  - follows rate monotonic priorities
  - schedules the highest-priority task <u>only if</u> it will not cause a deadline miss for the next job of the task with the smallest period



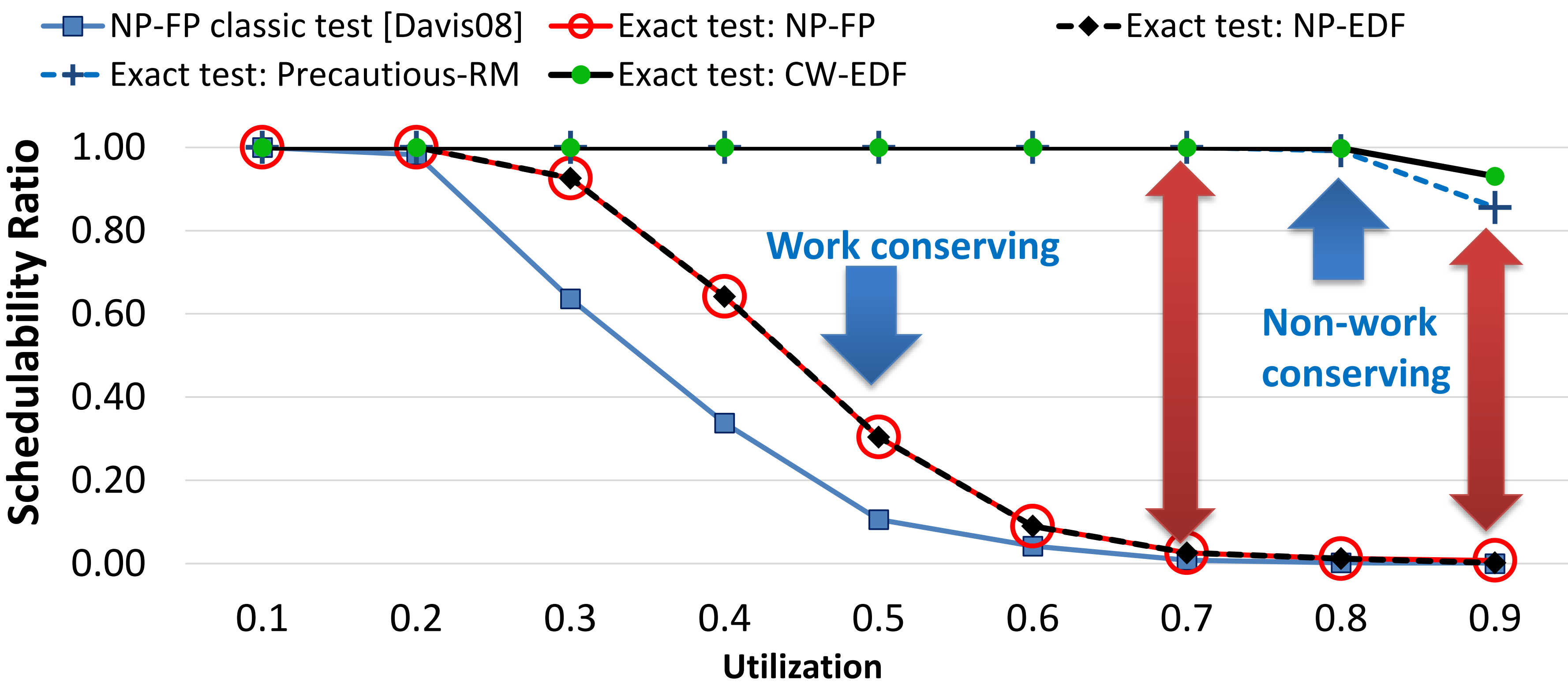
Non-Work-Conserving Scheduling

- Leaves the processor idle even if there are tasks that are not yet scheduled

- Example: **Precautious-RM** [Nasri2014]
  - is an online algorithm
  - follows rate monotonic priorities
  - schedules the highest-priority task only if it will not cause a deadline miss for the next job of the task with the smallest period



Why Non-Work-Conserving Algorithms are Needed?



- Autosar benchmark [Kramer2015] (only task set that satisfy the "necessary schedulability conditions").

- The exact test is from
  Nasri and Brandenburg, "An Exact and Sustainable Analysis of Non-Preemptive Scheduling", manuscript, available at https://people.mpi-sws.org/~bbb/ papers/pdf/preprint_np_exact_analysis.pdf

Problem 1

- Given:
  - Uniprocessor
  - Non-preemptive
  - Periodic tasks
  - Release offsets
  - Constrained (or arbitrary) deadline
  - Dependent (or independent)

- Scheduled by:
  - A **non-work-conserving** scheduling algorithm such as Precautious-RM

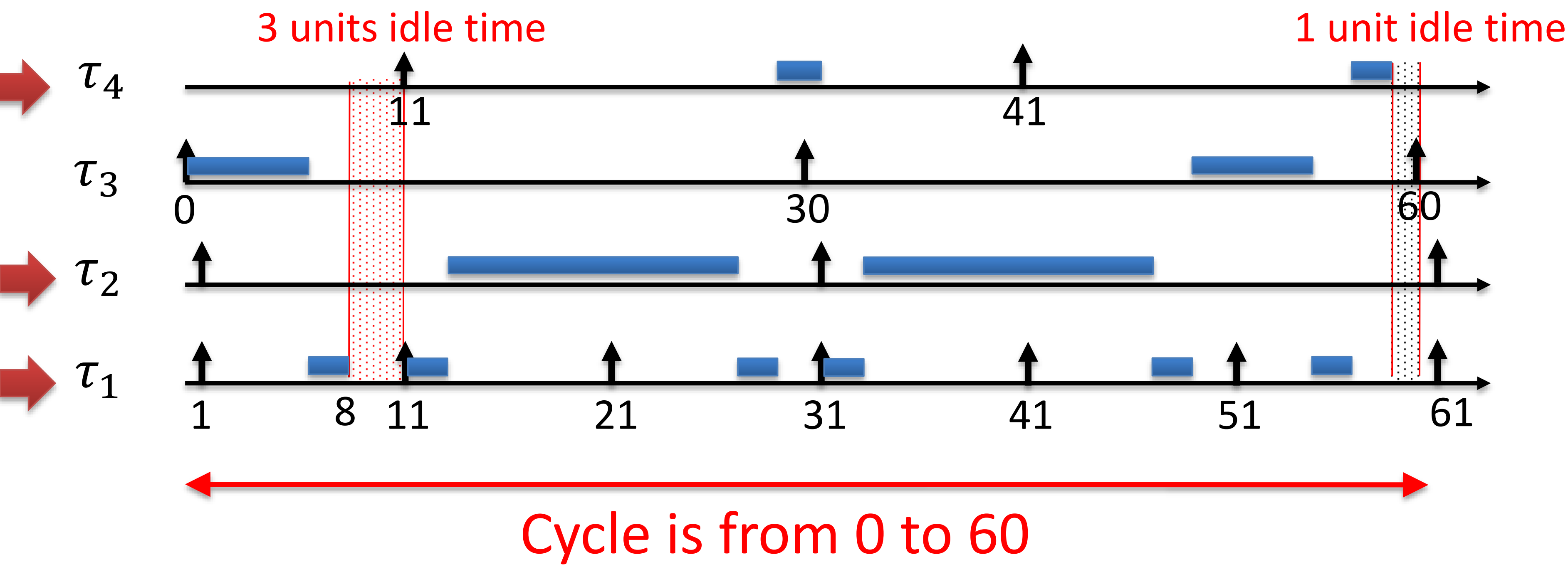
- **Problem:**
  What is the length of a cyclic schedule?
  When does the cyclic schedule start?

The Length of Cyclic Schedule

Hyperperiod is 30.

τ_i	C_i	D_i	T_i	O_i
τ_1	2	10	10	1
τ_2	14	30	30	1
τ_3	6	30	30	0
τ_4	2	30	30	11

Precautious-RM schedule:



Other Open Problems

- **Problem 3: Is there any feasible asynchronous task set τ that does not have <u>any</u> cyclic schedule with length H?**

Yes

No

Problem 4:

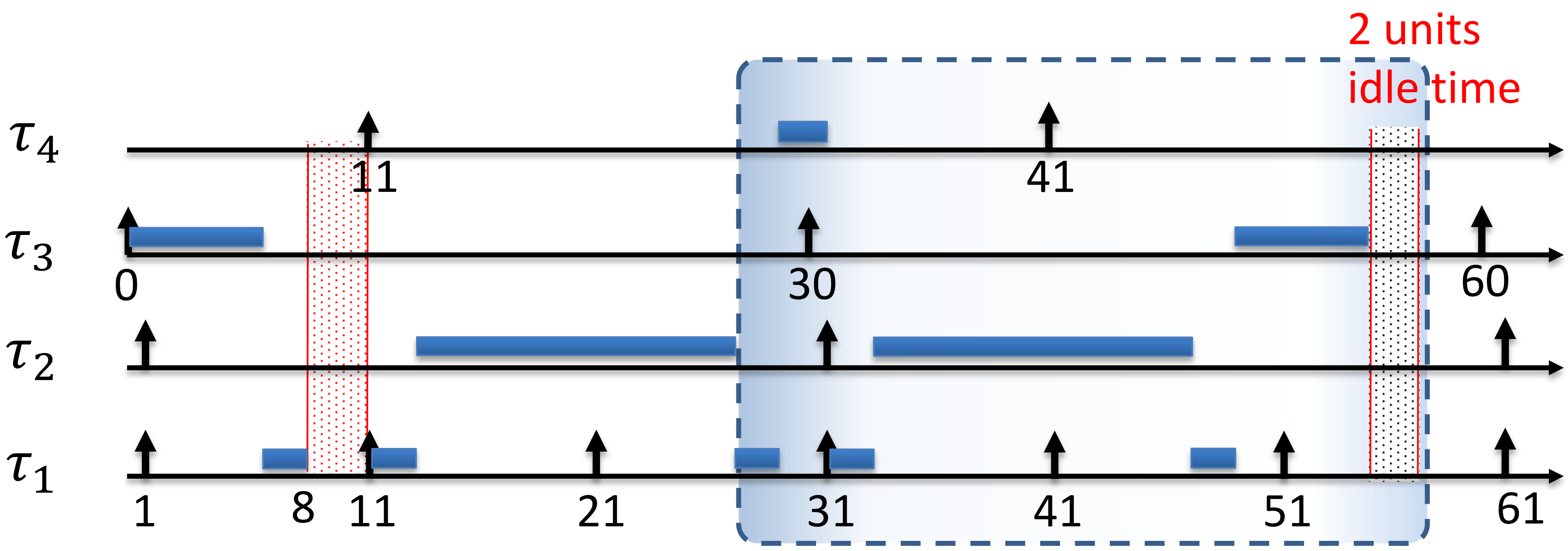
Given: a feasible non-preemptive task set
Find: **schedule *S* that has the smallest cycle length**

Problem 5:

Given: a feasible non-preemptive task set
Find: **How to find/build that schedule?**

H = hyperperiod
Assume: task set is periodic with constrained deadline and independent tasks

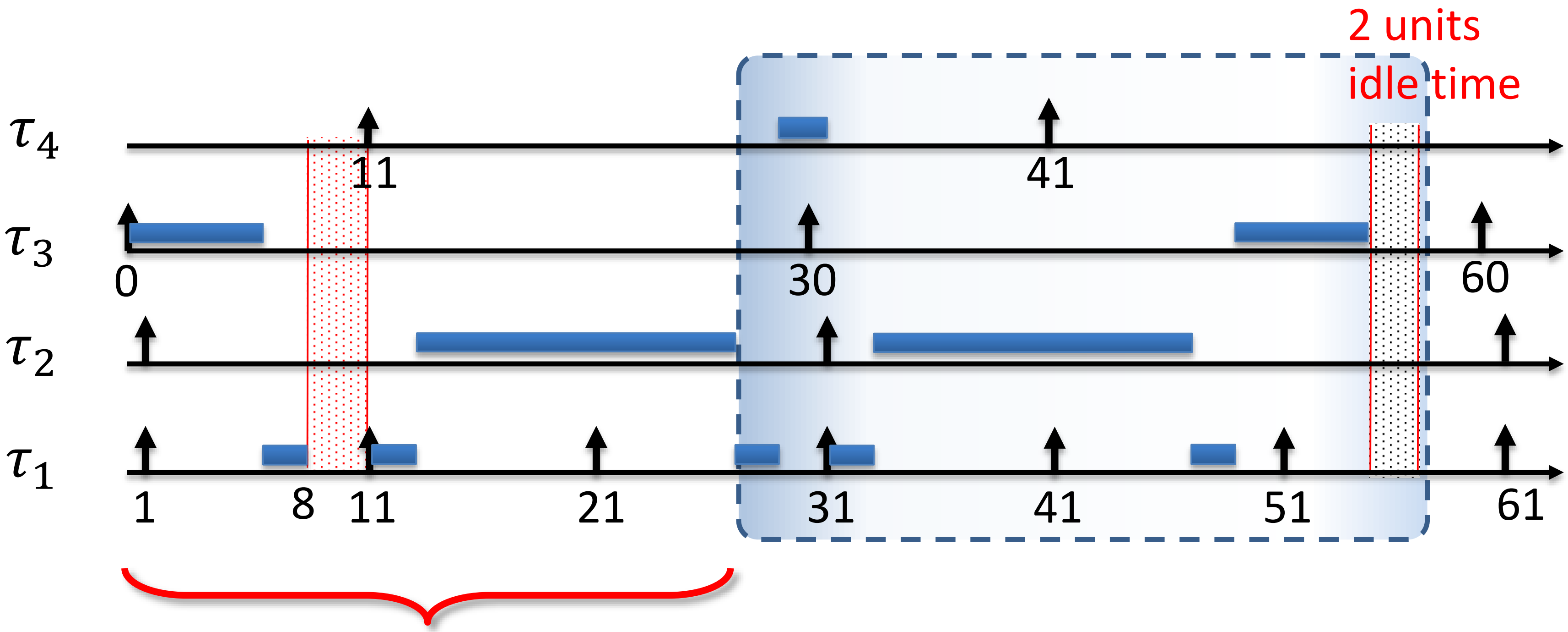
Find a Cyclic Schedule with Length H



A cyclic schedule with length H exists.

Hyperperiod = H = 30

Other Concerns



What about the length of non-cyclic schedule?
What if it is even larger than the cyclic schedule?

Hyperperiod = H = 30

Summary

Small offline tables

Motivations

- The need of creating small offline tables to save memory
- Using online scheduling algorithms is an efficient option

Performance and predictability

Non-work-conserving non-preemptive scheduling

- Promising performance
- High system predictability

The length and start time of a cyclic schedule

Open Problems

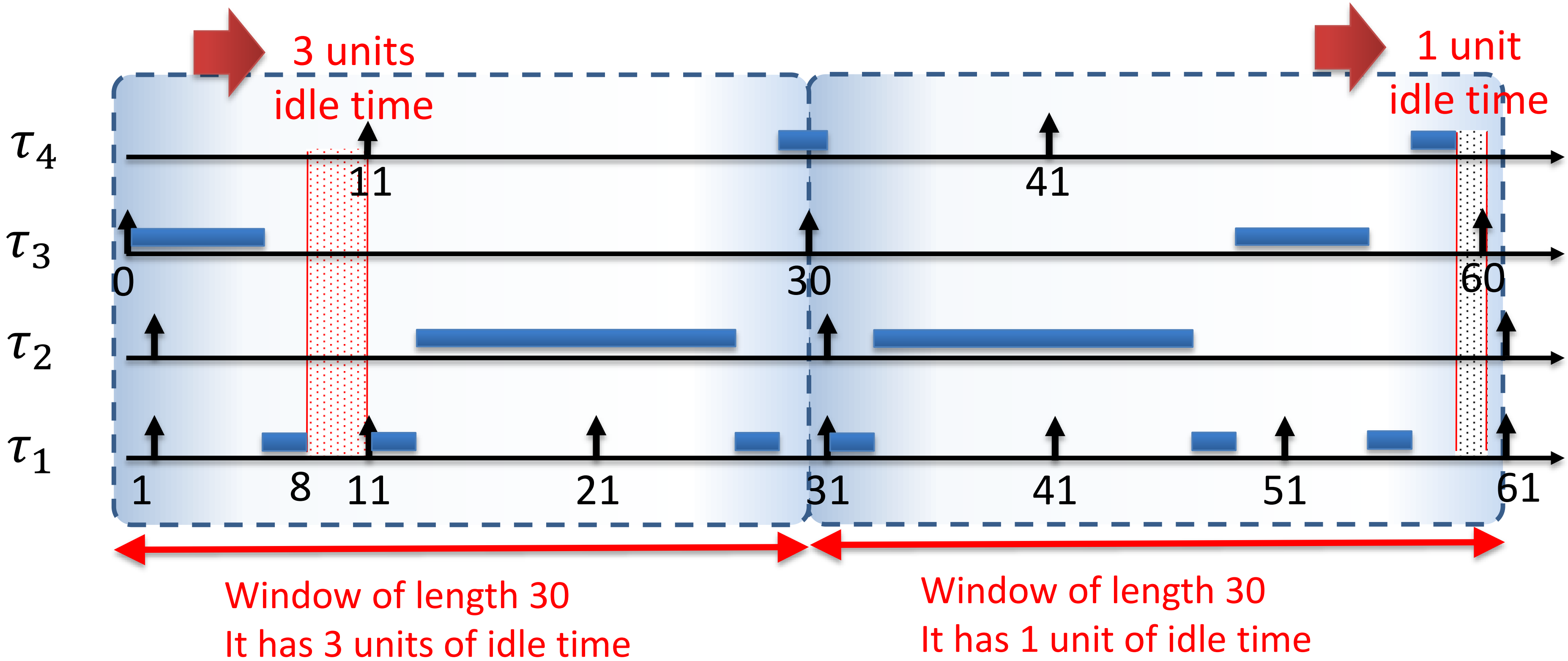
- No practical bound on the length of simulation interval
- We need methods to find the smallest cyclic schedule

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Isae
Institut Supérieur de l'Aéronautique et de l'Espace
ENSMA

Thank you

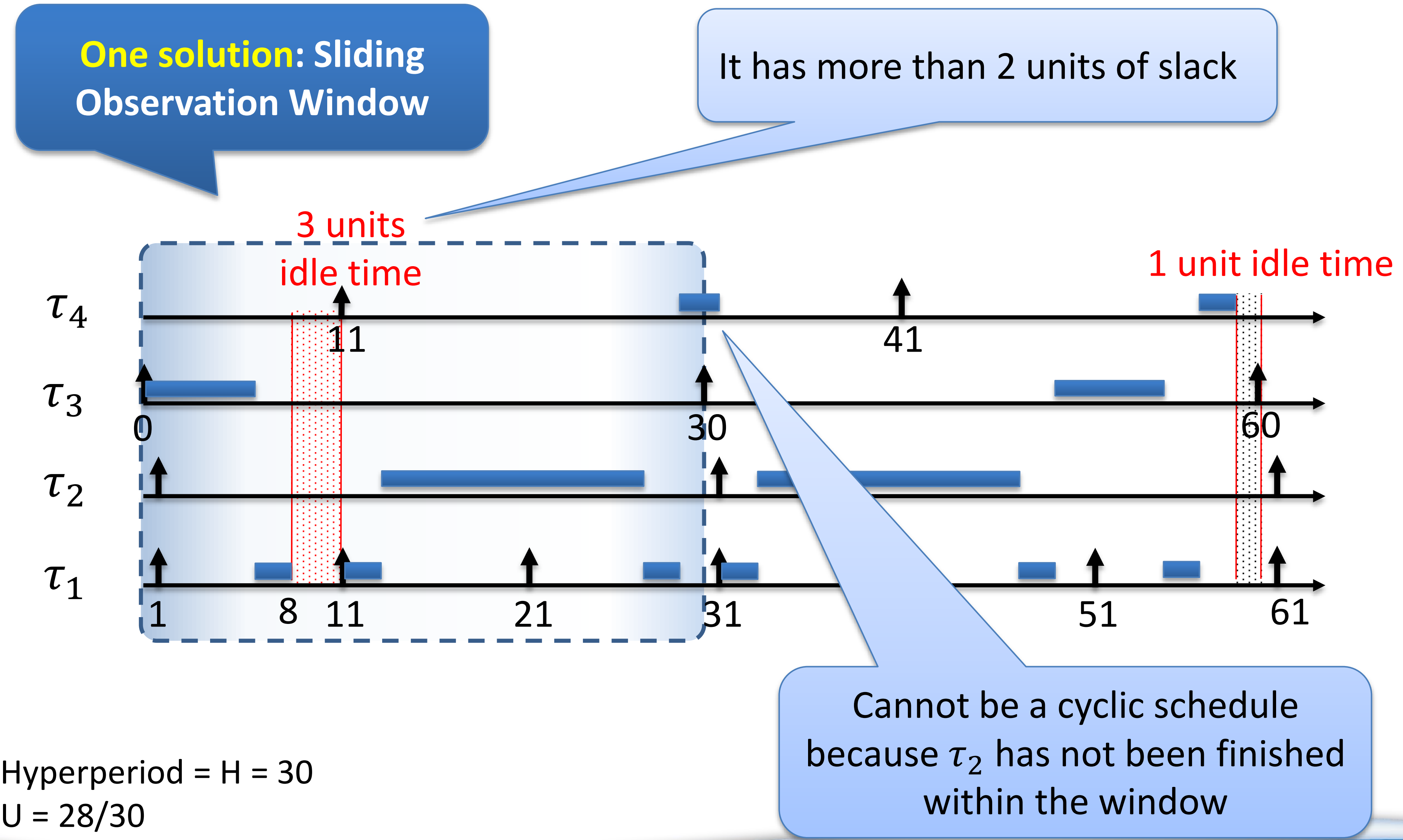
Looking Closer



- Hyperperiod = 30
- U = 28/30
- Slack in hyperperiod = (1-U)H = 2

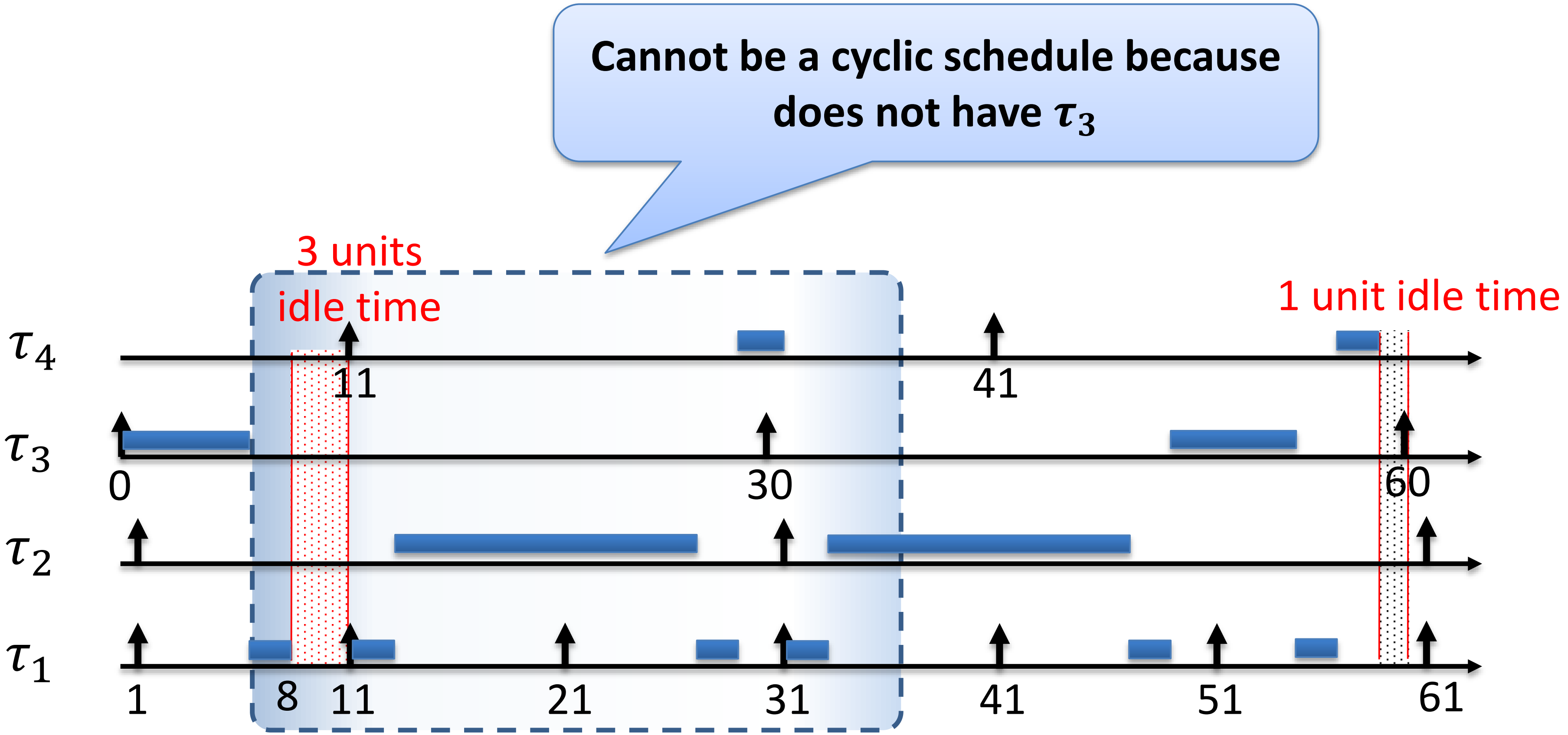
This window is borrowing slack from the next one

Find a Cyclic Schedule with Length H



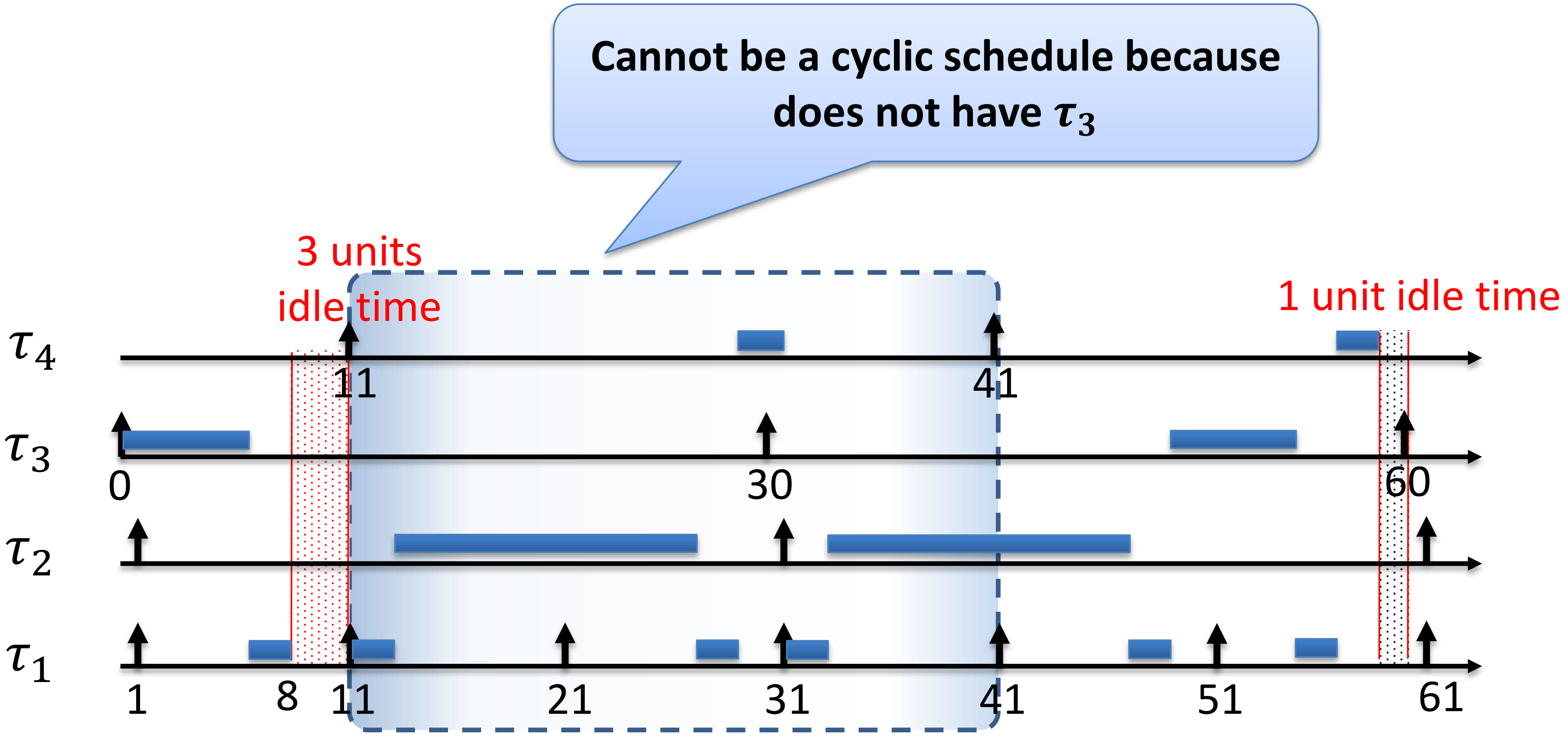
Hyperperiod = H = 30
U = 28/30

Find a Cyclic Schedule with Length H



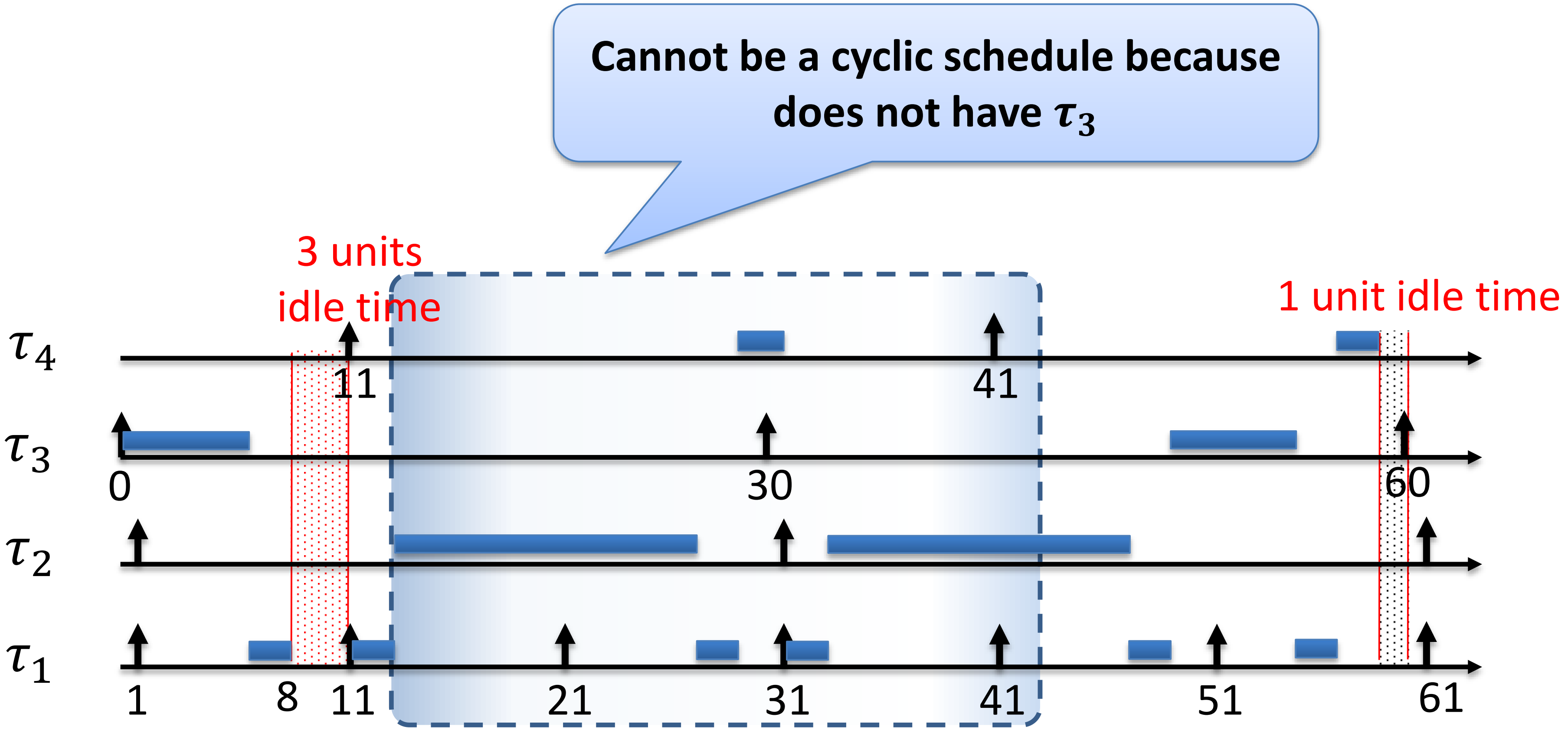
Hyperperiod = H = 30

Find a Cyclic Schedule with Length H



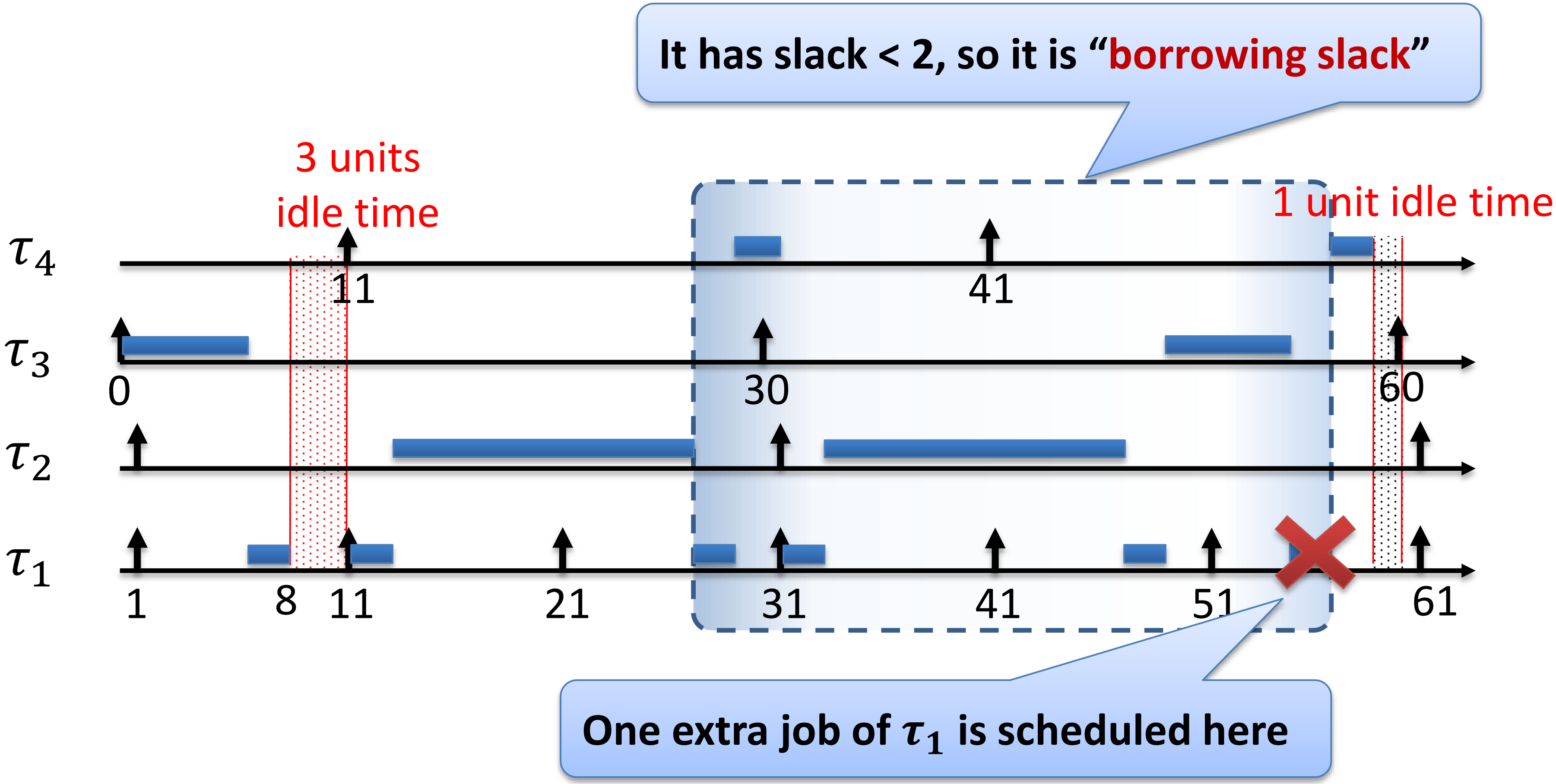
Hyperperiod = H = 30

Find a Cyclic Schedule with Length H



Hyperperiod = H = 30

Find a Cyclic Schedule with Length H



Hyperperiod = H = 30

Non-preemptive Scheduling is a Way to Increase Predictability

- It increases the predictability
  - Better estimation of the WCET
  - More accurate cache analysis
  - More accurate information about accesses to shared data

- It is inevitable in many systems
  - GPUs
  - CAN networks
  - small embedded systems

- It simplifies design and reduces overheads
  - Resource management becomes easier