reSilient coMputer archItectures and LIfE Sciences

Politecnico di Torino

Department of Control and Computer Engineering

SCHEDULING MIXED TASKS

STEFANO DI CARLO

PRIORITY SERVERS

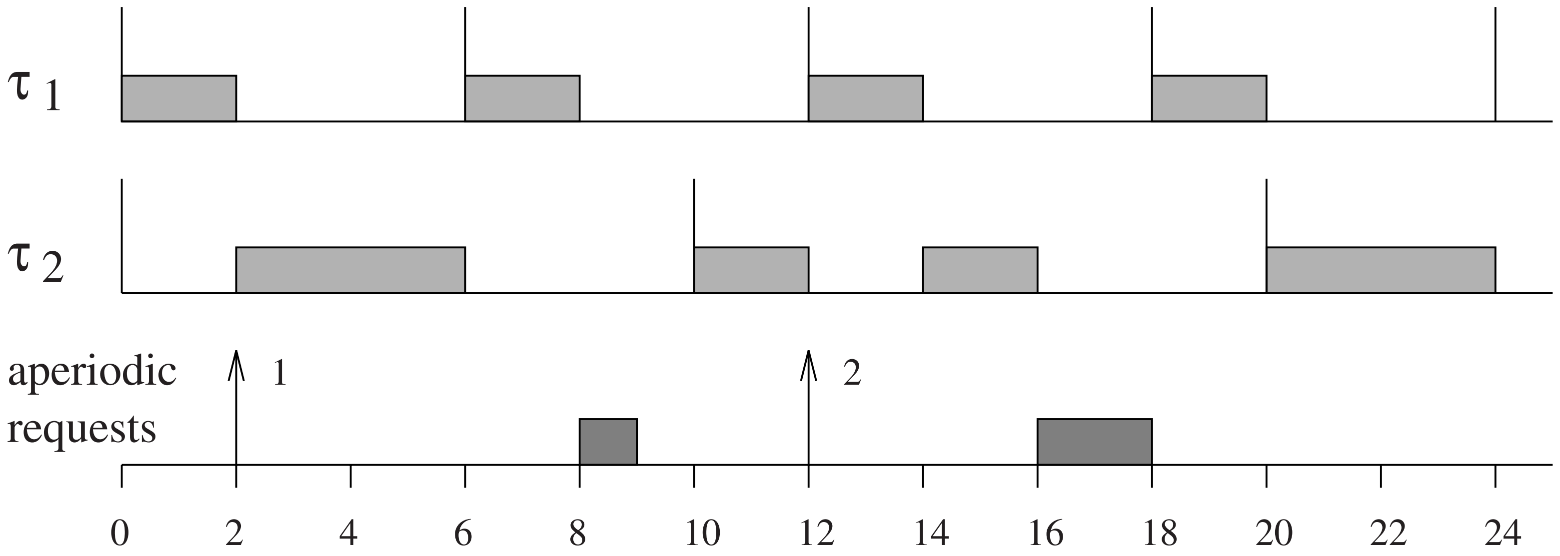
- So far, we considered only homogenous task sets
  - Either periodic or aperiodic tasks only
- In real application heterogeneous task sets are used (both periodic & aperiodic tasks)
  - Periodic tasks
    - Time-driven
    - Execute critical control activities with hard timing constraints aimed at guaranteeing regular activation rates
  - Aperiodic tasks
    - Event-driven
    - May have hard, soft, or non-real-time requirements depending on the specific application

PRIORITY SERVERS

- Specific scheduling algorithms are used
  - Background scheduling
  - Fixed priority servers
    - Periodic tasks are scheduled based on a fixed-priority assignment
    - All periodic tasks start simultaneously at time t = 0 and their relative deadlines are equal to their periods
    - Arrival times of aperiodic tasks are unknown
    - The minimum inter arrival time of a sporadic task is assumed to be equal to its deadline
    - All tasks are pre-emptible
  - Dynamic priority servers (not covered here)
    - Periodic tasks are scheduled based on a dynamic-priority assignment
    - Same as before

BACKGROUND SCHEDULING

- Periodic tasks are scheduled using RM (rate monotonic scheduling)
- Aperiodic tasks are scheduled in background, when no periodic instance is ready



BACKGROUND SCHEDULING

- Advantage
  - Simple
  - Two ready queues are needed
    - Periodic task queue: RM scheduling is used
    - Aperiodic task queue: first-come-first-serve scheduling is used

- Disadvantage
  - In case of high periodic workload, response time for aperiodic tasks can be very high
  - Suitable only when aperiodic tasks are soft real-time

POLLING SERVER

- A special periodic tasks, the server, is created to serve aperiodic tasks as soon as possible
- The server is characterized by:
  - Ts: period of the server
  - Cs: capacity/budget of the server
- Behavior of the server:
  - It is scheduled using RM as other periodic tasks
  - It consumes its budget either running the aperiodic task, or immediately if not aperiodic task is ready
  - Its budget is replenished at the beginning of each new period

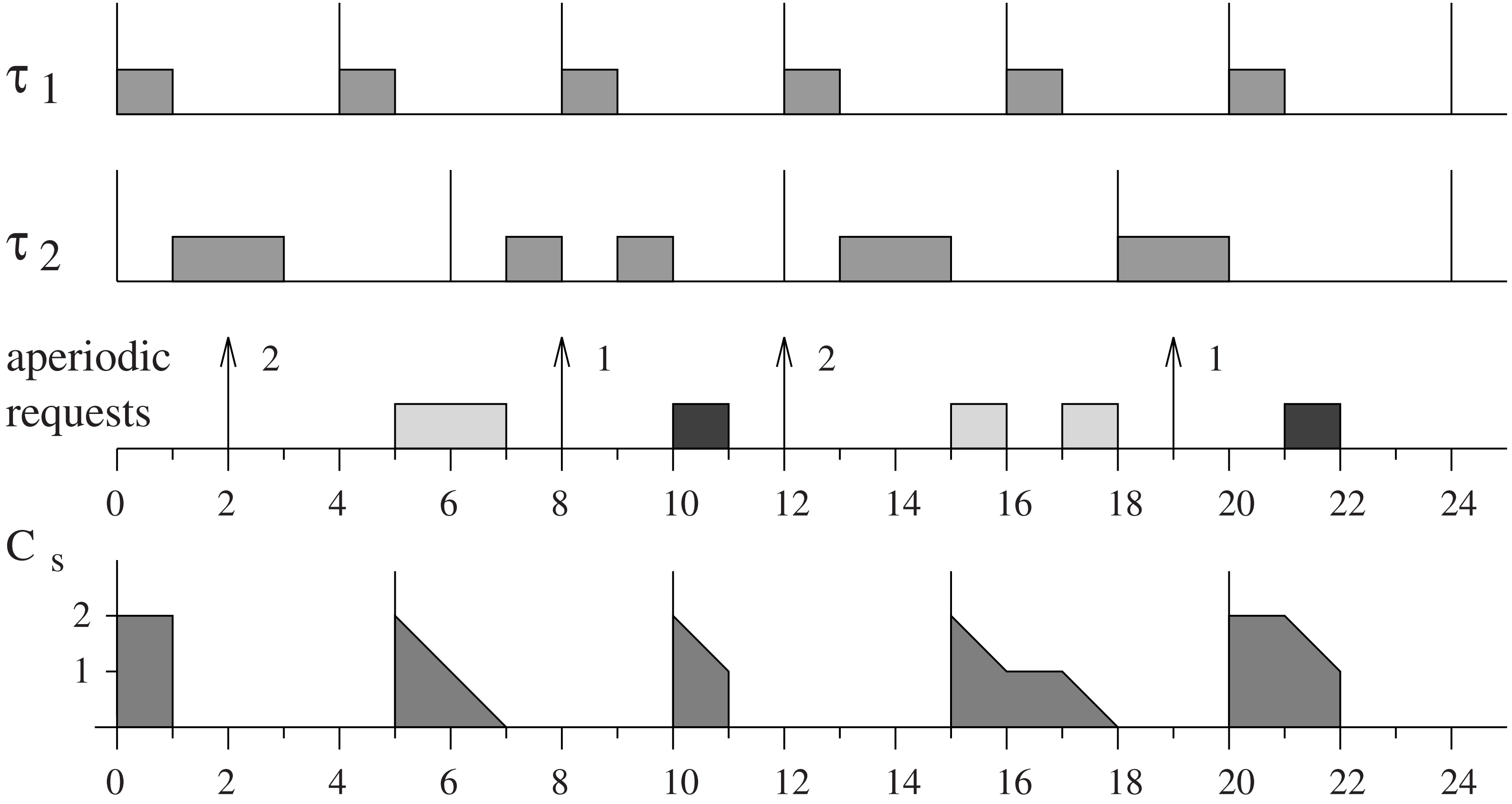
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



EXAMPLE

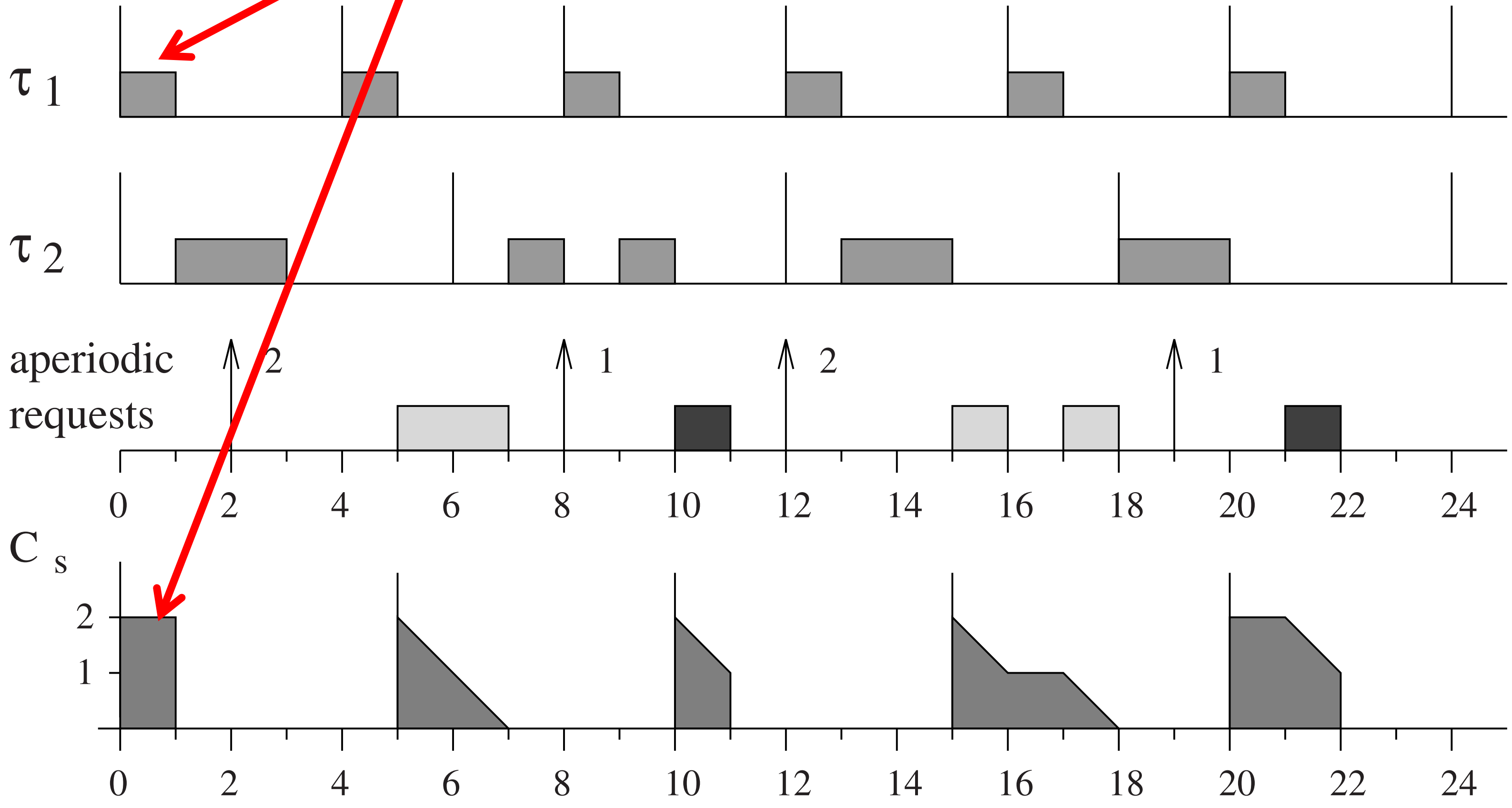
1 - CPU is assigned to t_1.
As no aperiodic task is ready, the server consumes all its budget.

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



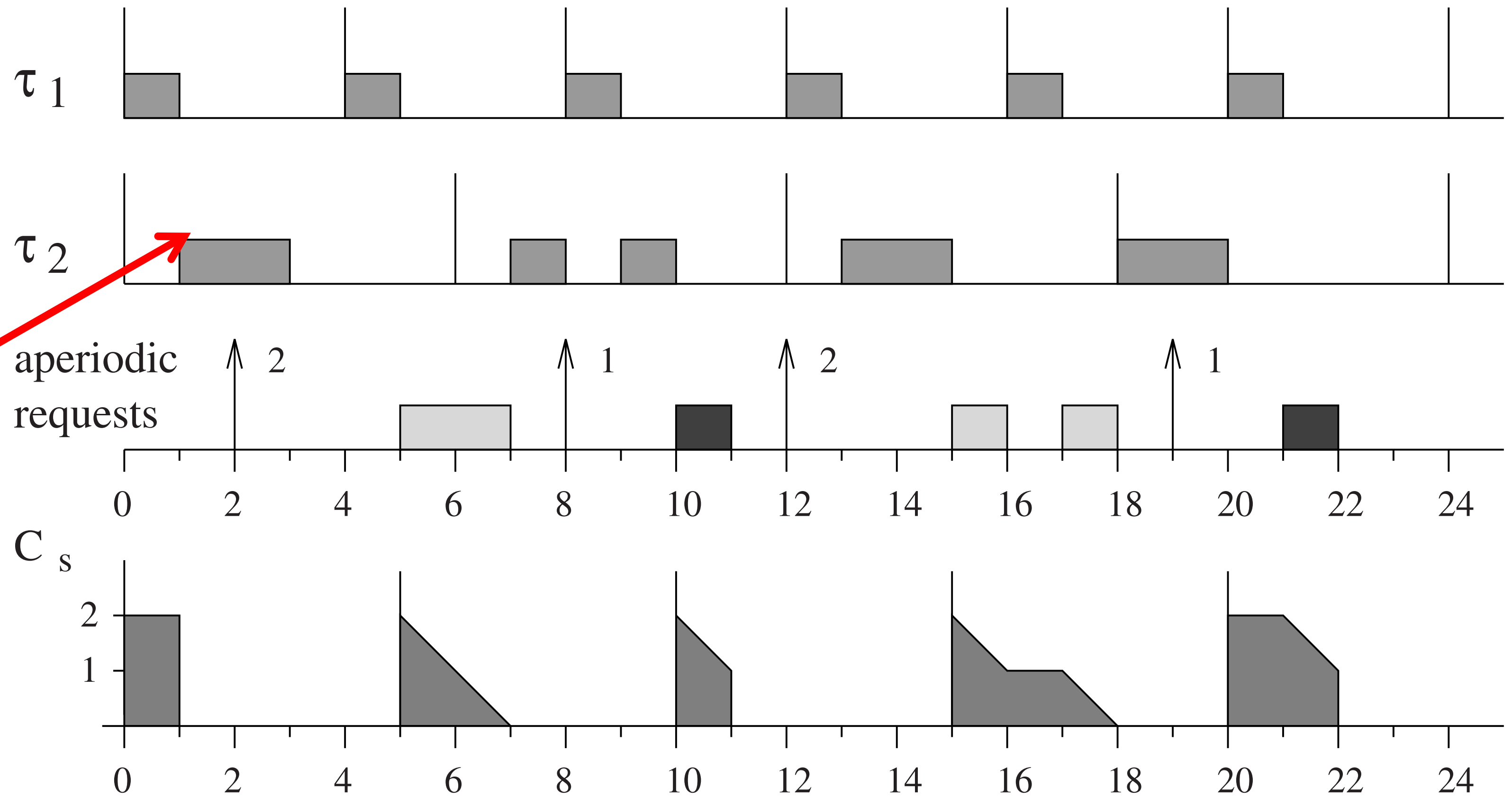
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



2 – The next task to run is Server but as no aperiodic task is ready, the CPU is given to t_2

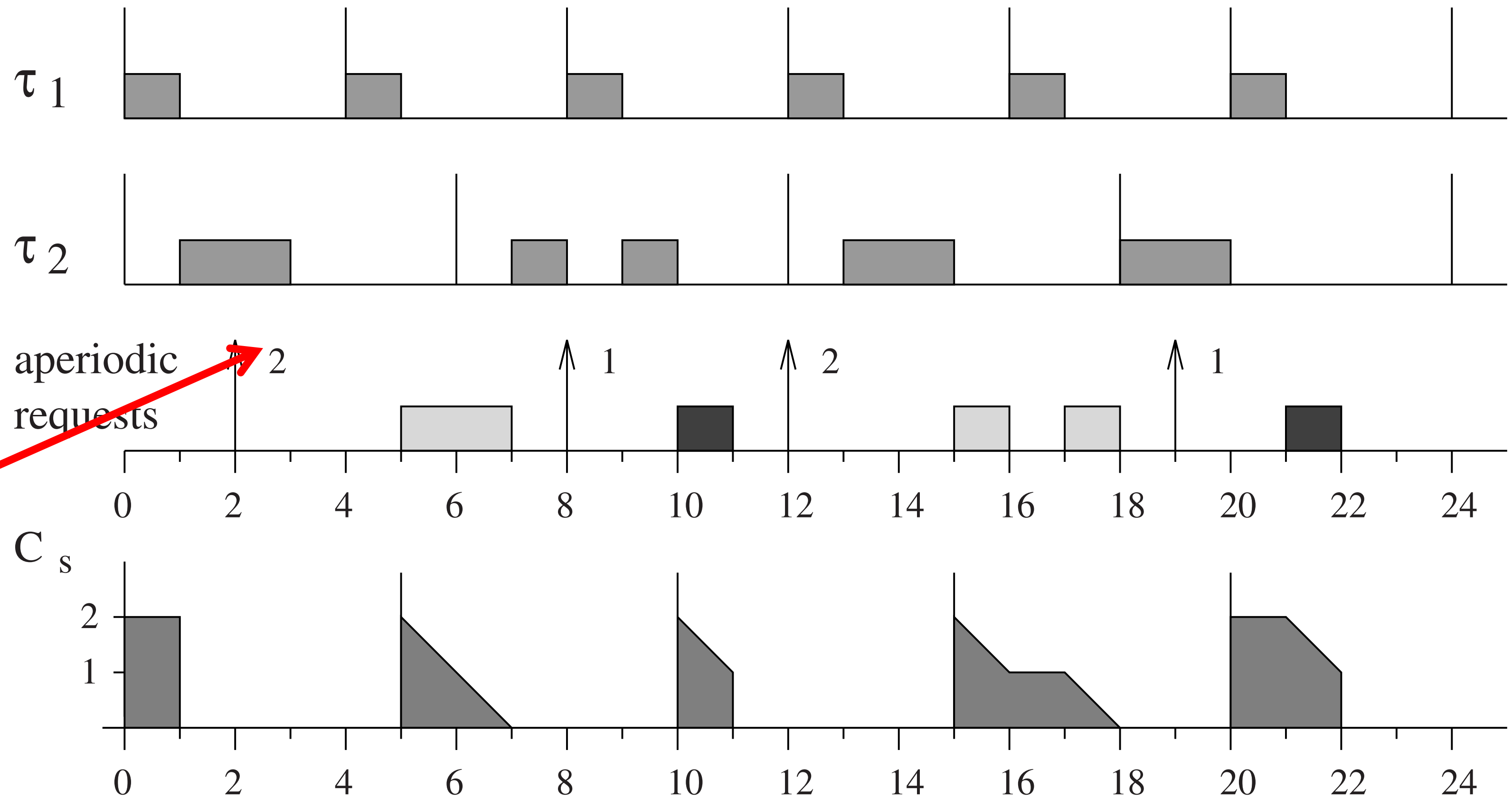
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



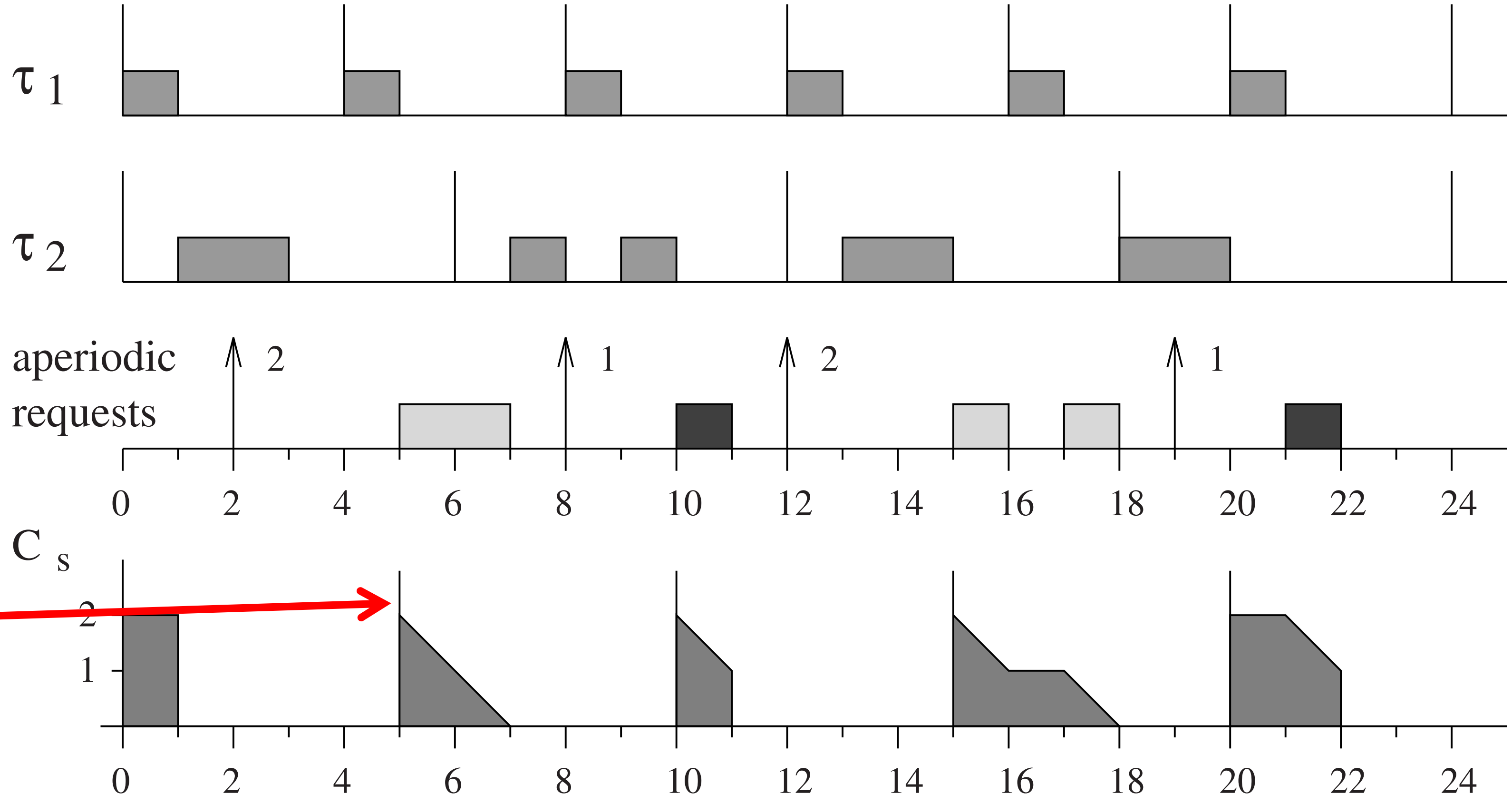
3 – A periodic task with C=2 becomes ready. As Sever will run on at time 5, the task remains ready

EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$
$T_s = 5$



4 – At time 5, the server becomes running, and consumes it budget to run the aperiodic task with C=2

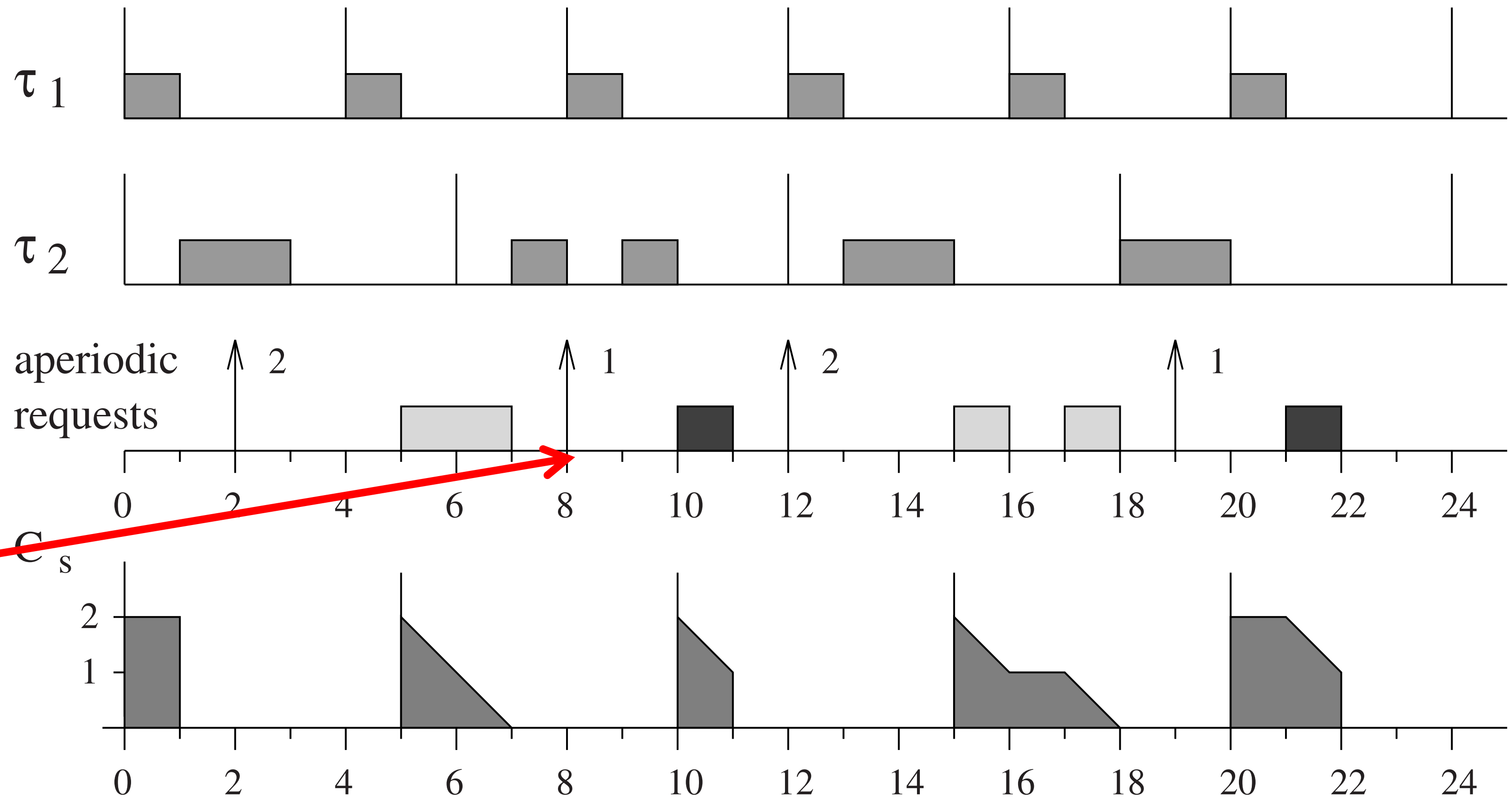
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



5 – At time 8, a new periodic task is ready with C=1. As the server will run only at 10, it has to wait.

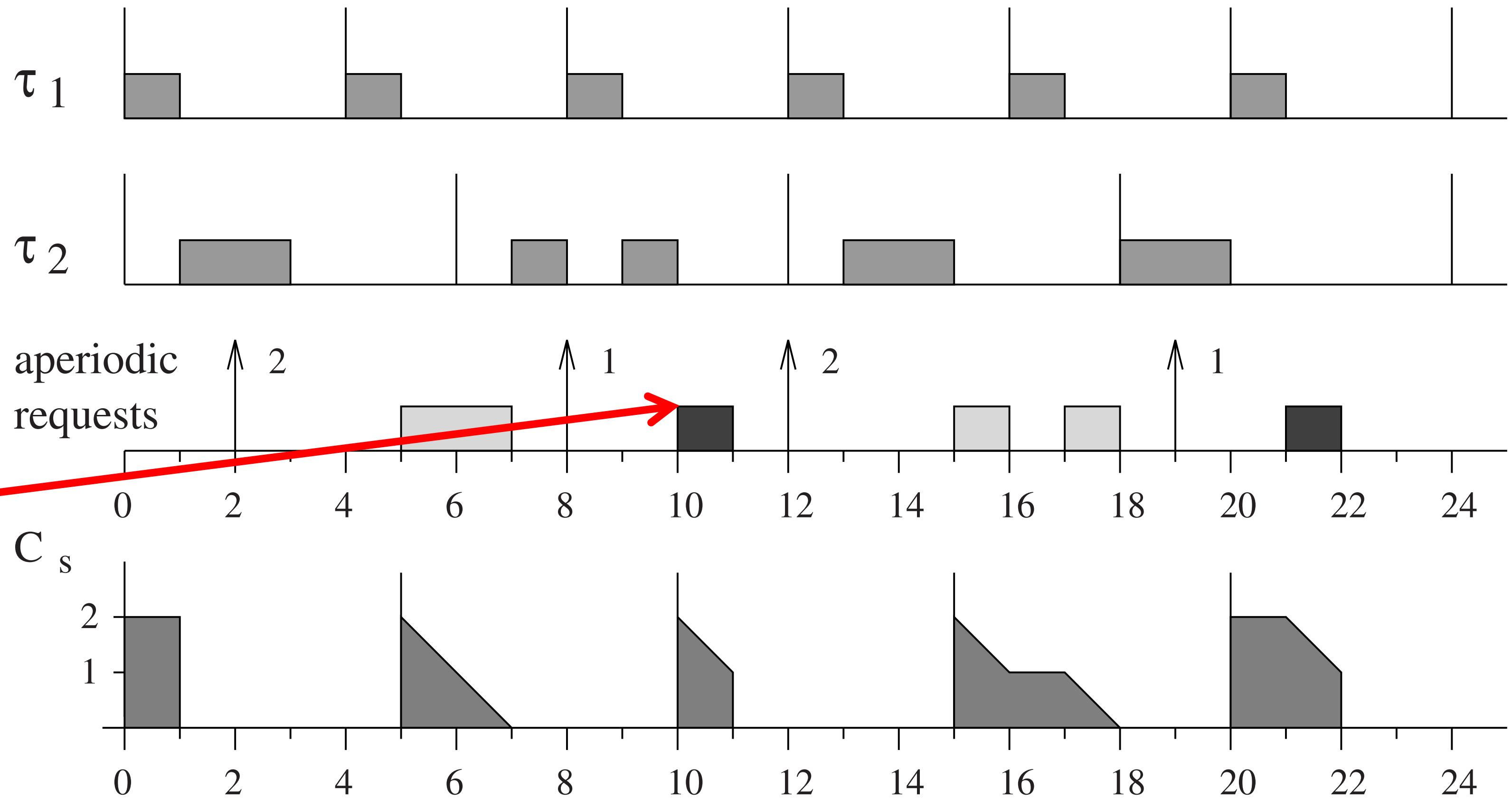
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



6 – At time 10, the server can run consuming one of its budget for running the aperiodic task with C=1.

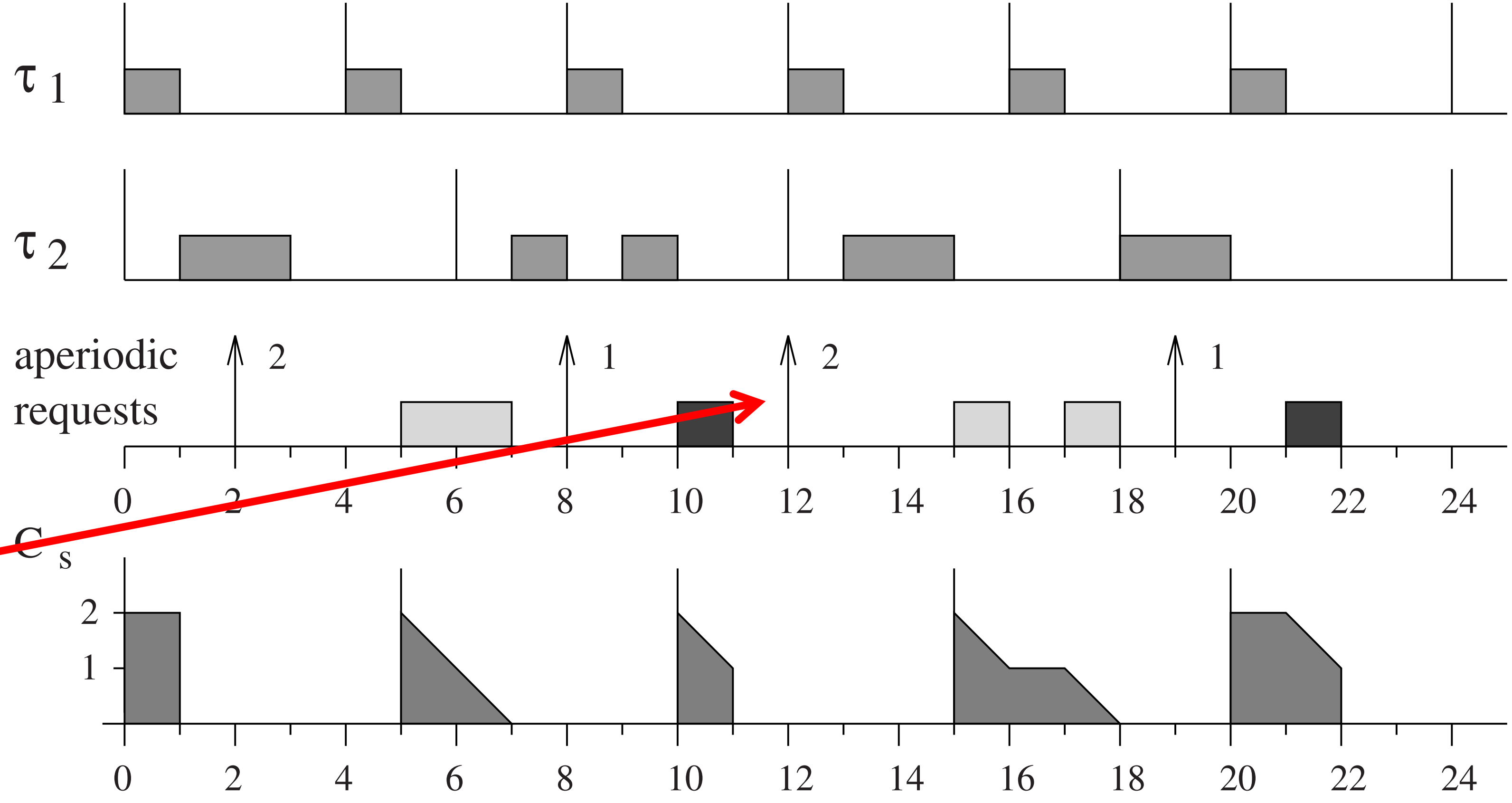
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



7 – At time 12, a aperiodic task is ready with C=2. It must wait the next release of the polling server

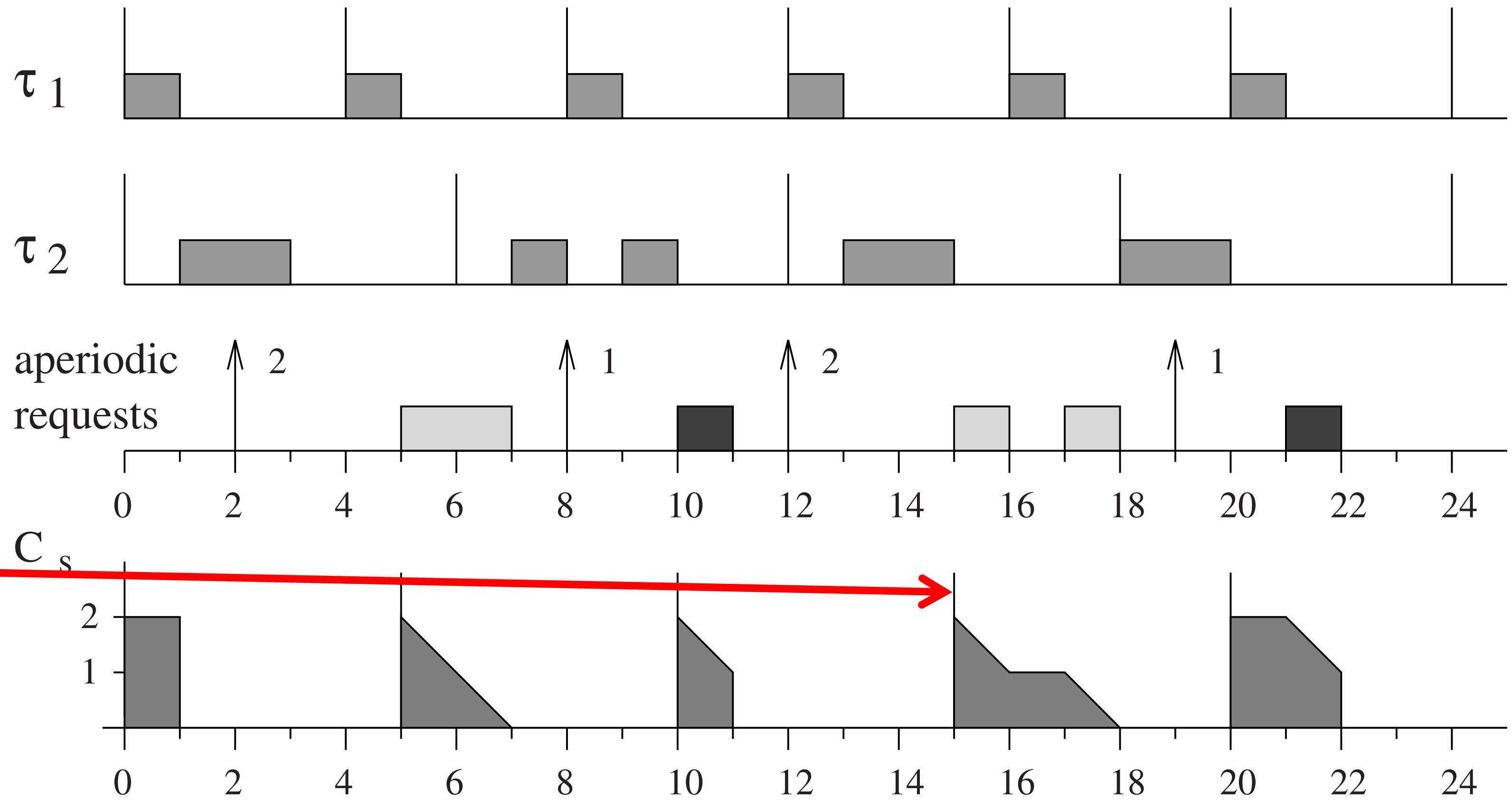
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



8 – At time 15, the server runs the aperiodic task with C=2.

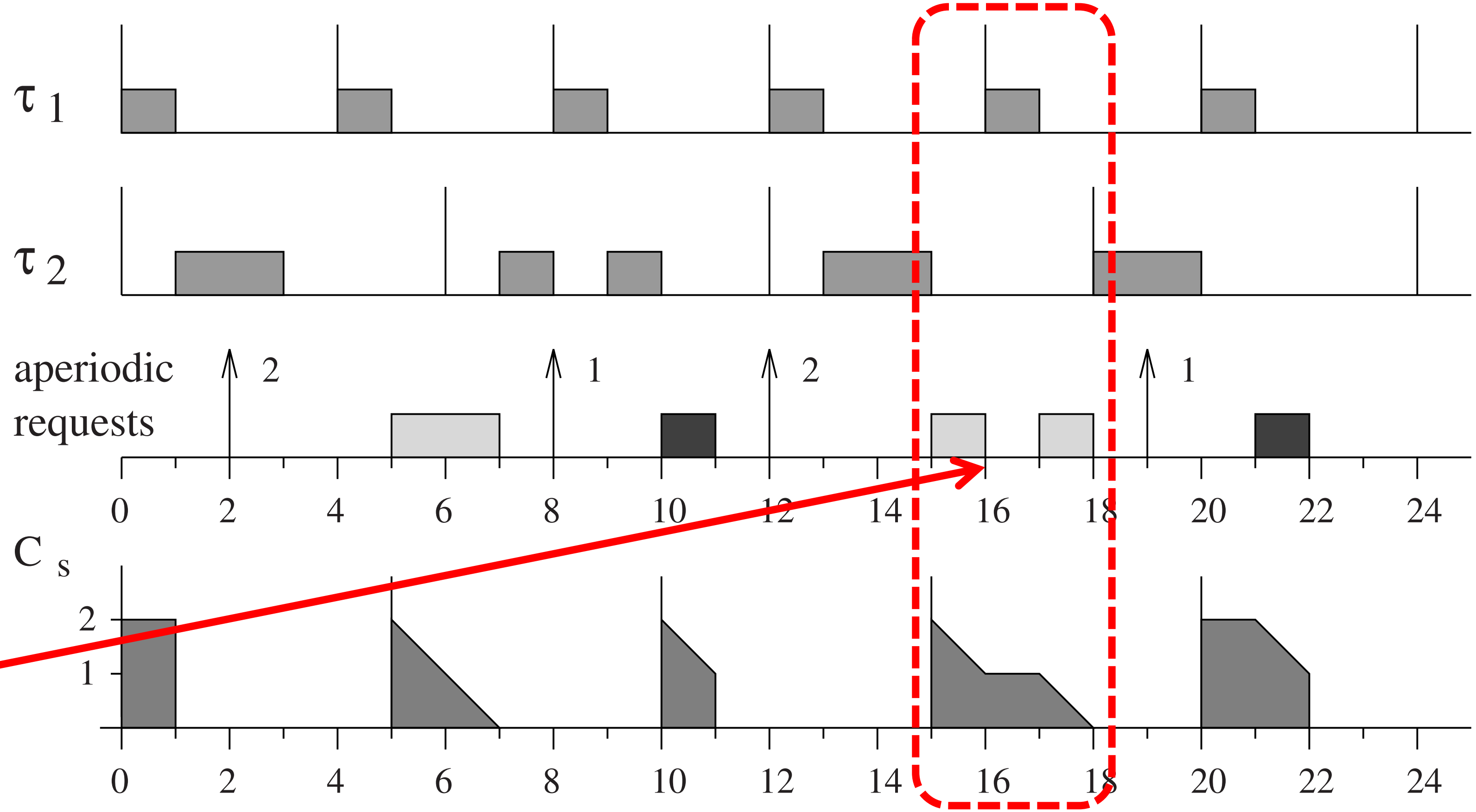
EXAMPLE

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



9 – At time 16, the server is preempted as t_1 becomes ready. The server consumed only 1 time slot of its budget.

FEASIBILITY OF POLLING SERVER

- Considering n period tasks each with utilization U_i and a polling server with utilization $U_s=C_s/T_s$, the task set is feasible with RM if

$$\prod_{i=1}^{n}(U_i + 1) \leq \frac{2}{U_s + 1}$$

DIMENSIONING THE POLLING SERVER

- How to set C_s and T_s so that the resulting scheduling is feasible? → we are looking for the polling server maximum utilization factor: U_s^{max}

$$P \stackrel{\text{def}}{=} \prod_{i=1}^{n} (U_i + 1)$$

$$U_s^{max} = \frac{2 - P}{P}$$

DIMENSIONING THE POLLING SERVER

- Given U_s^{max} the rule of thumb is the following:
  - Set U_s at most equal to U^{smax}
  - Set T_s as the period of the periodic task with the shortest period (the polling server becomes the highest priority task)
  - Set $C_s=U_sT_s$

DEFERRABLE SERVER

- As the Polling Server, the DS algorithm creates a periodic task for servicing aperiodic tasks
  - Usually, it has a high priority
- DS preserves its capacity if no requests are pending
- The capacity is maintained until the end of the period
- Aperiodic requests can be serviced at the same server's priority at anytime, as long as the capacity has not been exhausted
- At the beginning of any server period, the capacity is replenished at its full value

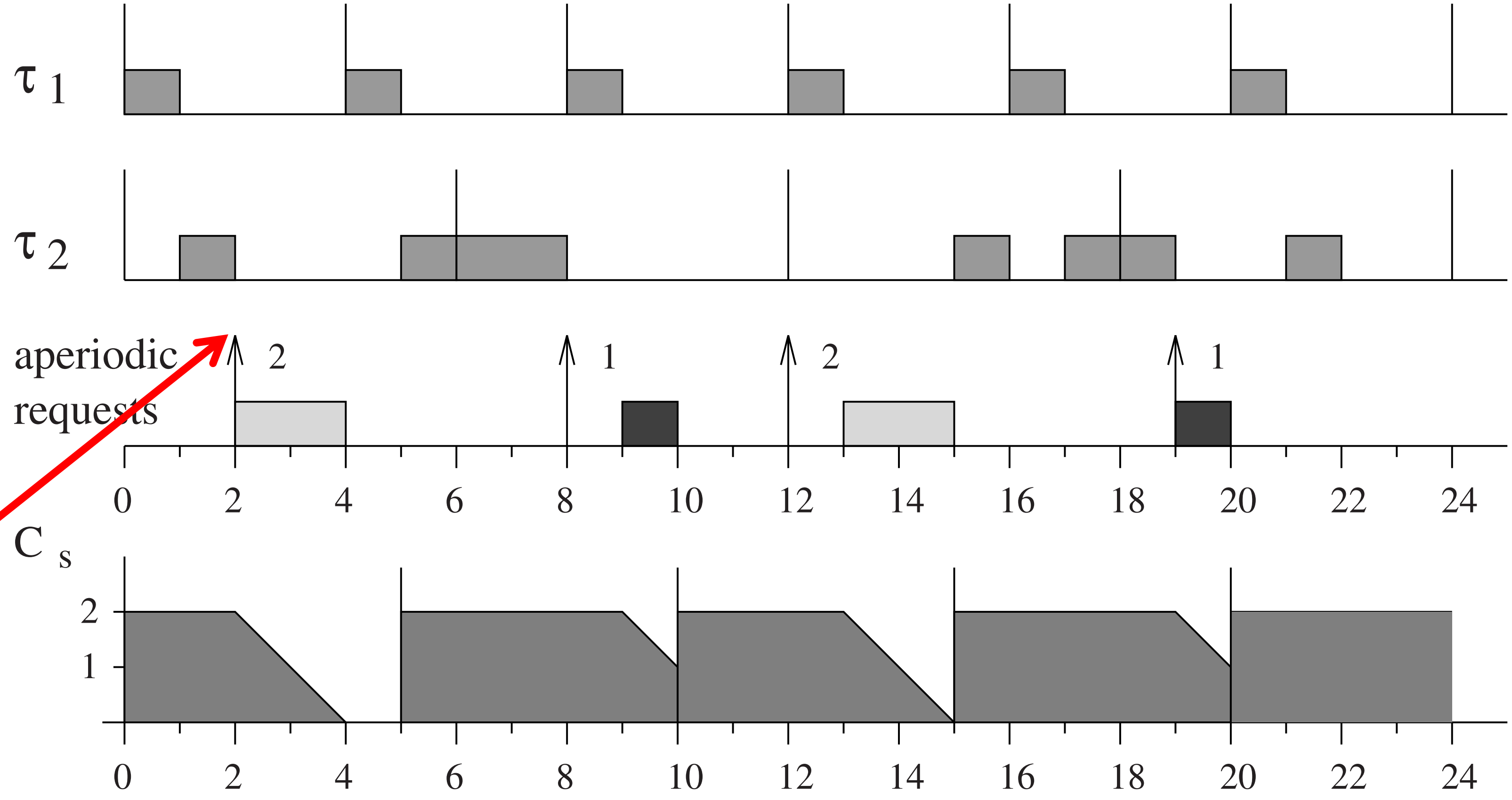
EXAMPLE 1

	C_i	T_i
τ_1	1	4
τ_2	2	6

Server

$C_s = 2$

$T_s = 5$



The server keeps its budget when no aperiodic task is ready. As soon as one is ready, if allowed, it can run.

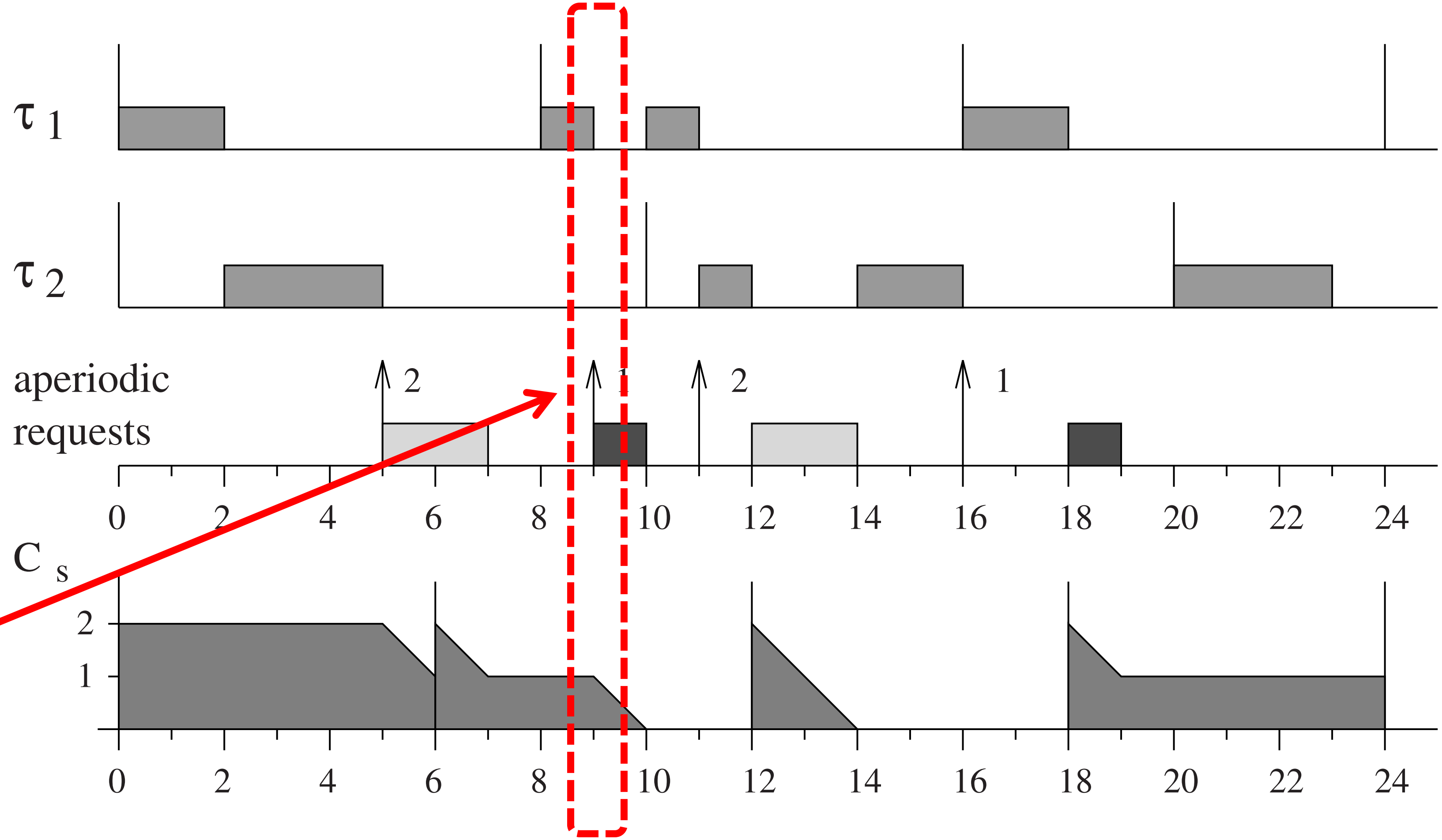
EXAMPLE 2

	C_i	T_i
τ_1	2	8
τ_2	3	10

Server

$C_s = 2$

$T_s = 6$



The server has the higher priority according to RM. It preempts the other tasks.

FEASIBILITY OF DEFERRABLE SERVER

- Considering n period tasks each with utilization U_i and a deferrable server with utilization $U_s=C_s/T_s$, the task set is feasible with RM if

$$\prod_{i=1}^{n}(U_i + 1) \leq \frac{U_s + 2}{2U_s + 1}$$

DIMENSIONING THE DEFERRABLE SERVER

- How to set C_s and T_s so that the resulting scheduling is feasible? → we are looking for the polling server maximum utilization factor: U_s^{max}

$$P \stackrel{\text{def}}{=} \prod_{i=1}^{n}(U_i + 1)$$

$$U_s^{max} = \frac{2 - P}{2P - 1}$$

SMILIES

reSilient coMputer archItectures and LIfE Sciences

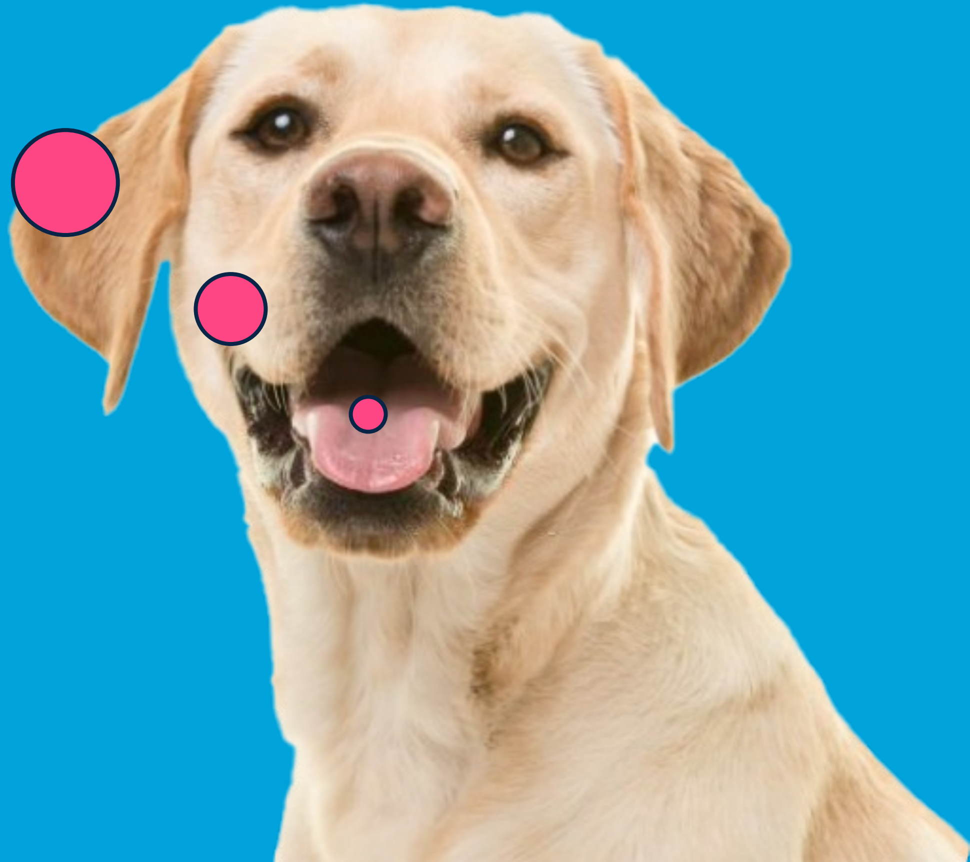
Politecnico di Torino

1859

Department of Control and Computer Engineering

DAUIN

QUESTIONS?



THANK YOU!