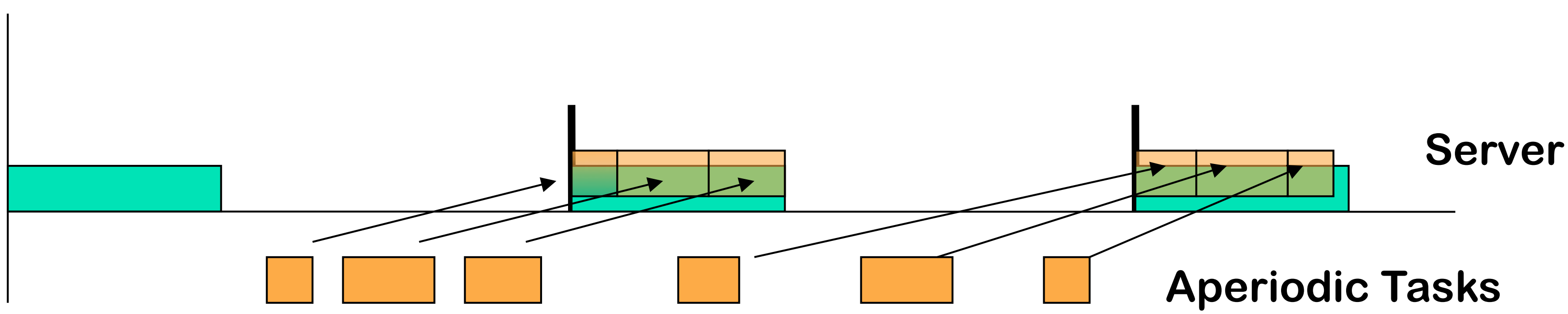
Servers for aperiodic tasks

Principles for server design
Explaining servers through example

Server-based systems

- Periodically invoke a service task ("server") to execute aperiodic tasks
- The server is modeled as a periodic task and can be included in schedulability analysis
- Allocate the server a computation budget C_s and a period P_s
- The server can serve aperiodic tasks until the budget expires; the budget can be replenished every period
- Many choices: Servers have different flavours depending on the details of when they are invoked, what priority they have, and when budgets are replenished



Principles of server design

- It is simple enough to represent the servers as periodic tasks
- So, why so many rules?
  - We want to reduce the response times for aperiodic tasks
  - Avoid the problems with the polling server: retain unused budget
  - If we want to retain the budget
    - When does it expire?
      - If a server has budget 2 and deadline 5, it cannot have a budget of 2 when t=4; there is only one unit of computation remaining but a budget of 2
      - We can not make the operating system do too much work. It only schedules by priority or deadline and does not verify if the deadline has expired or not.

Principles of server design

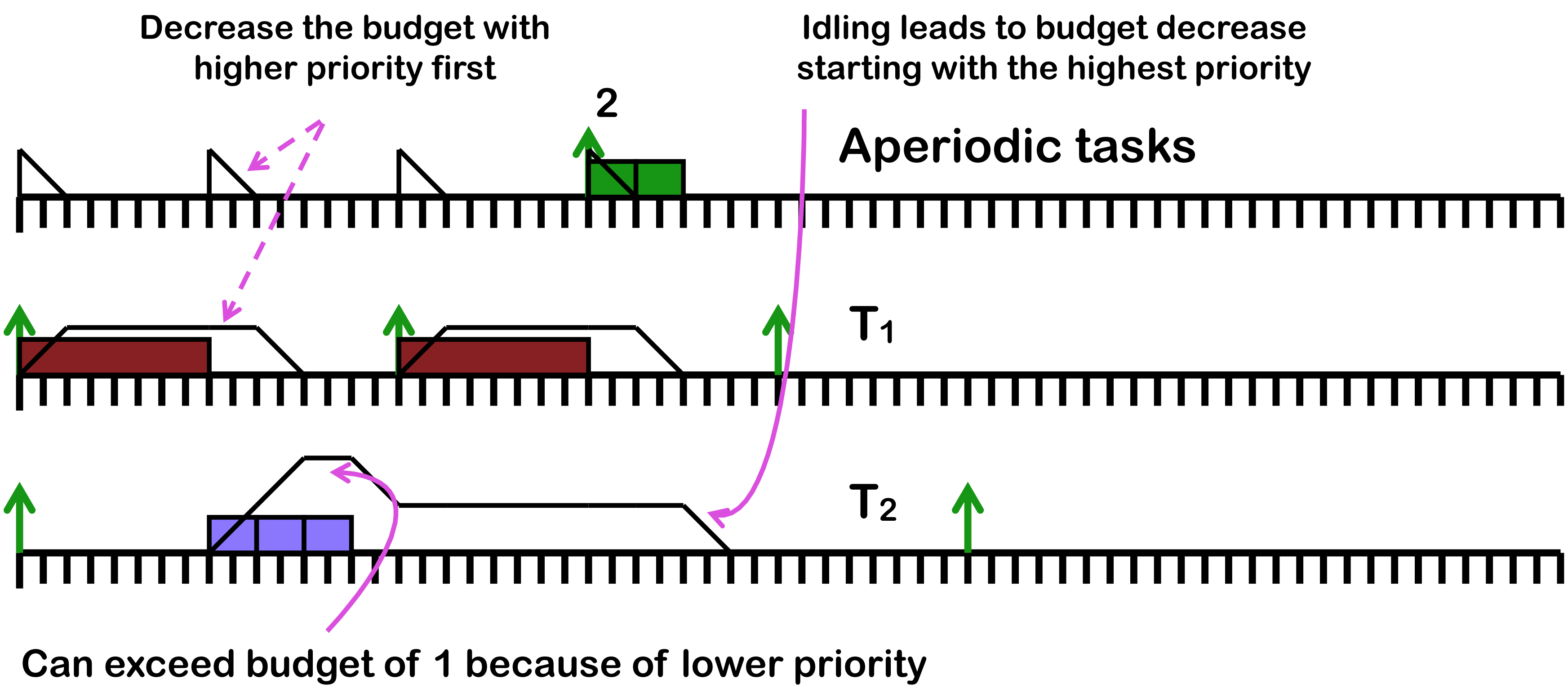
- It is simple enough to represent the servers as periodic tasks
- So, why so many rules?
  - We want to reduce the response times for aperiodic tasks
  - Avoid the problems with the polling server: retain unused budget
  - If we want to retain the budget
    - When does it expire?
    - When does it increase?
      - If we consume a portion of the budget, when do we restore it?
    - We cannot allow the server to use more than the allotted fraction of the processor: if the server has a utilization of 0.4, it can not use more than 2 units of time every 5 units (or 4 in every 10, 8 in every 20, ...)
    - How can we implement these easily? [The polling server is easy to implement.]

Priority Exchange Server

- Like the deferrable server, this server retains its budget until the end of the server period
- Unlike the deferrable server, this server's priority slips over time: when not used, the priority is exchanged for that of the executing periodic task
- Note that if a server has utilization 0.25, it can
  - Use 1 time unit every 4 units
  - Use 2 units every 8 units
  - …
  - *As priority slips, the budget may increase*

Priority Exchange Server: Example

T_1: (P_1=8,C_1=4)
T_2: (P_2=20,C_2=3)
Priority exchange server: (P_s=4,C_s=1)



Sporadic Server

- Server is said to be active if it is in the running or ready queue, otherwise it is idle.
- When an aperiodic task arrives and the budget is not zero, the server becomes active
- Every time the server becomes active, say at t_A, it sets replenishment time one period into the future, $t_A + P_s$ (but does not decide on replenishment amount).
- When the server becomes idle, say at t_I , set replenishment amount to capacity consumed in $[t_A, t_I]$

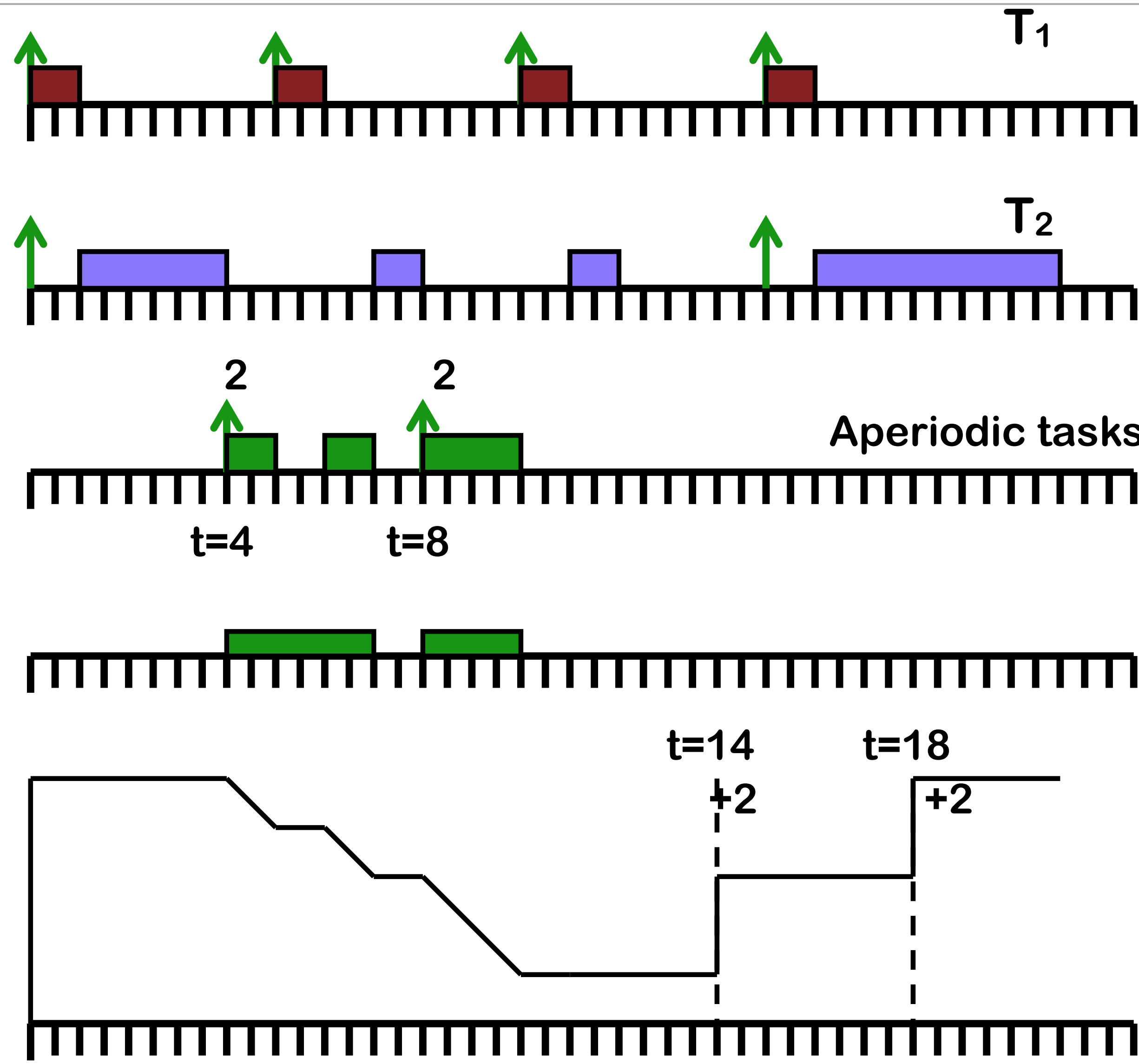
Sporadic Server: Example

Two periodic tasks
T_1: (P_1=5, C_1=1)
T_2: (P_2=15, C_2=5)

Sporadic Server
P_s=10, C_s=5

Sporadic server activity

Sporadic server budget



Dynamic priority aperiodic task servers

Dynamic priority exchange server

- Server has period P_s and budget C_s
- At the beginning of each period, a server capacity $C_S^d = C_s$ is allotted with deadline d.
- A capacity $C_{Si}^{di} = 0$ is initially allotted at the priority of each of the periodic tasks
- If the highest-priority task is an aperiodic capacity, C:
  - If aperiodic requests are pending, execute them and decrement the capacity.
  - If no aperiodic requests are pending, execute the top priority periodic task, say i. Subtract the execution interval E_i from C and add E_i to C_{Si}^{di}.
  - If no periodic tasks exist either, subtract the execution interval E_i from C.
- Schedulable if $U_p + U_s < 1$

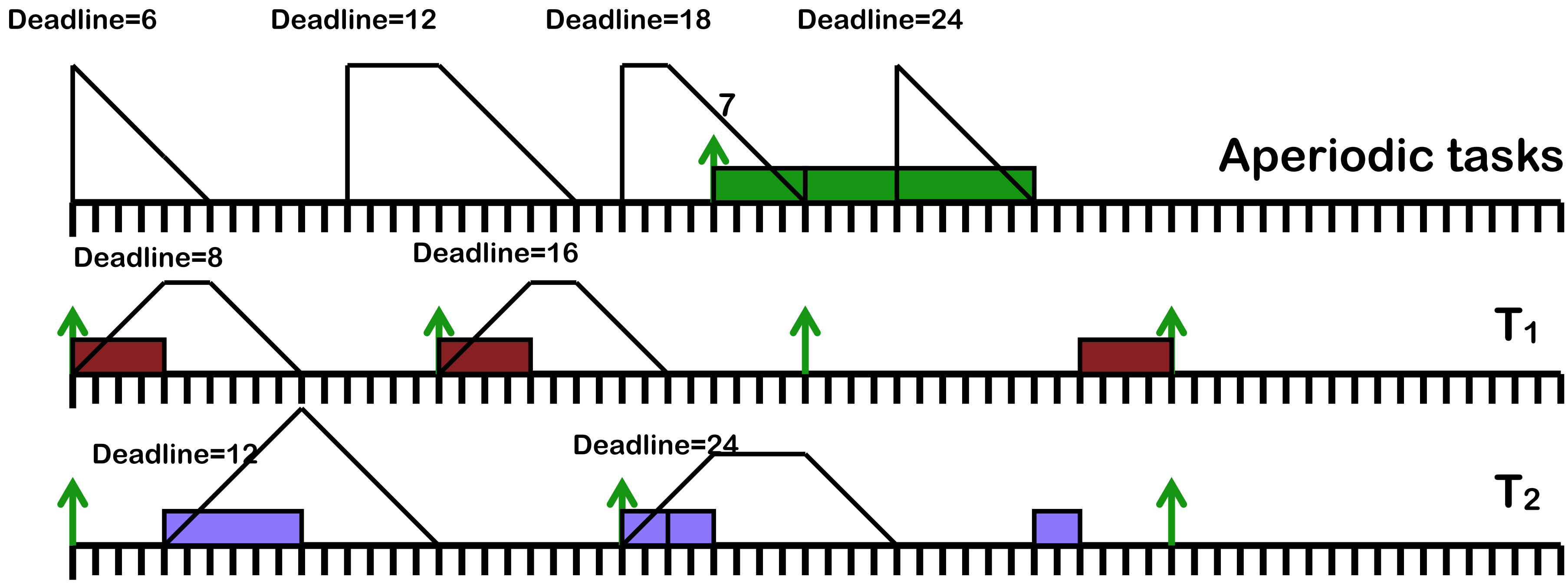
Example of DPE server

Two periodic tasks

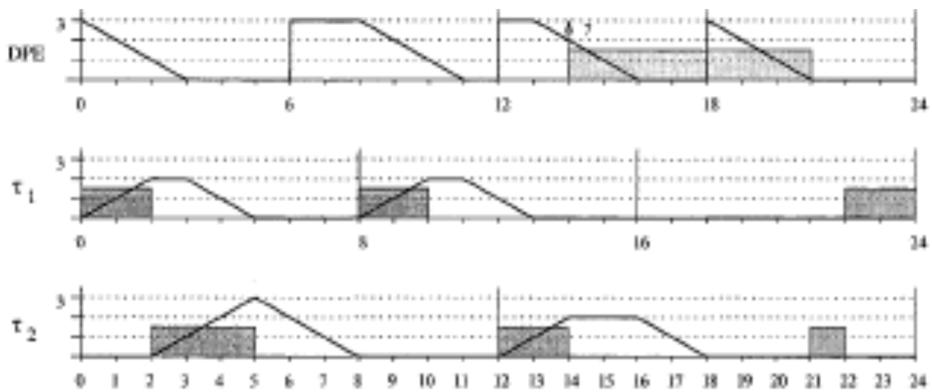
T_1: (P_1=08, C_1=2)

T_2: (P_2=12, C_2=3)

DPE server: *P_S=6; C_S=3*



Example of DPE server



Two periodic tasks

T_1: (P_1=08, C_1=2)

T_2: (P_2=12, C_2=3)

DPE server

P_S=6

C_S=3

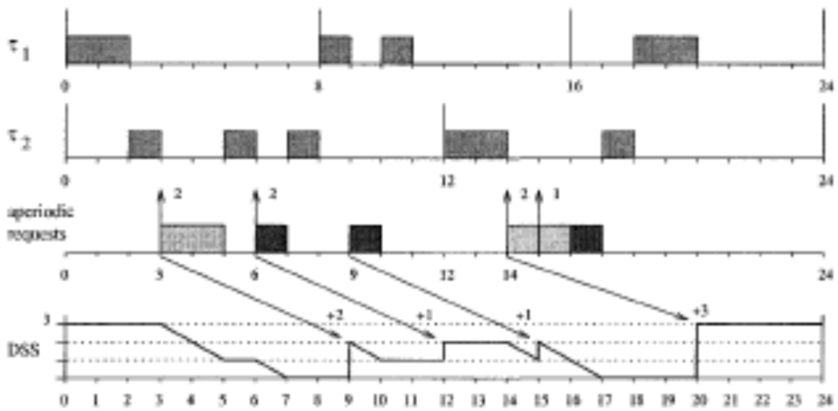
If the highest-priority task is an aperiodic capacity, *C*:

- **If aperiodic requests are pending, execute them and charge the capacity.**
- **If no aperiodic requests are pending, execute the top priority periodic task, say *i*. Subtract the execution interval E_i from *C* and add E_i to C_{Si}^{di}.**
- **If no periodic tasks exist either, subtract the execution interval E_i from *C*.**

Dynamic sporadic server

- When the server is created its capacity C_s is initialized.
- When there is an aperiodic task and $C_s > 0$, server becomes “active”
  - Set a replenishment time one period into the future (deadline)
- When the server becomes inactive set the replenishment amount as the capacity consumed

Example for dynamic sporadic server



Two periodic tasks

T_1: (P_1=08, C_1=2)

T_2: (P_2=12, C_2=3)

Dynamic sporadic server

P_S=6

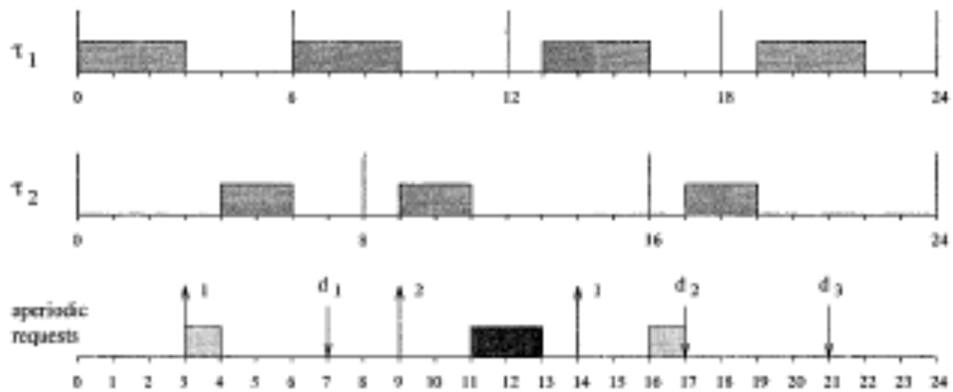
C_S=3

- When the server is created its capacity C_s is initialized.
- When there is an aperiodic task and $C_s > 0$, server becomes "active"
  - Set a replenishment time one period into the future (deadline)
- When the server becomes inactive set the replenishment amount as the capacity consumed

Total bandwidth server

- When the k^{th} aperiodic request arrives at time $t=r_k$, give it a deadline:
  - $D_k = \max(r_k, d_{k-1}) + C_k/U_s$
  - where U_s is the utilization alloted to the aperiodic task server
- There is no need to specify a budget and a period for the server
- Note that $U_P+U_S <= 1$ ensures schedulability
  - U_P is the utilization of the periodic tasks

Example for TBS

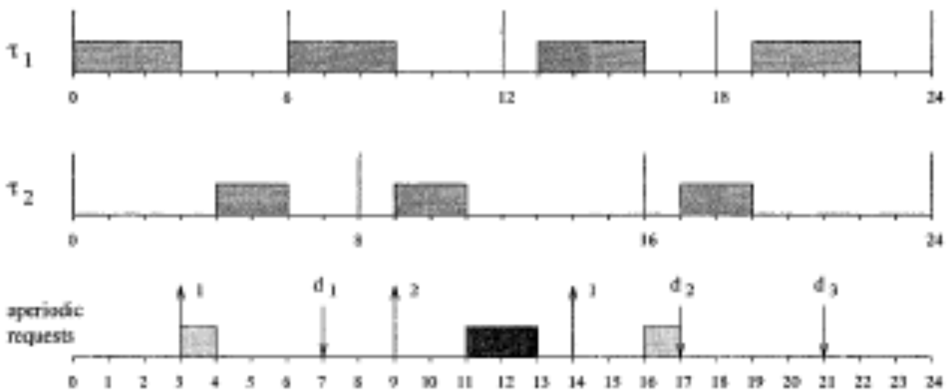


Two periodic tasks
T_1: (P_1=6, C_1=3)
T_2: (P_2=8, C_2=2)

Total bandwidth server
U_s = 0.25

- When the k^{th} aperiodic request arrives at time $t=r_k$, give it a deadline:
- $d_k = \max(r_k, d_{k-1}) + C_k/U_s$
- where U_s is the utilization alloted to the aperiodic task server

Example for TBS

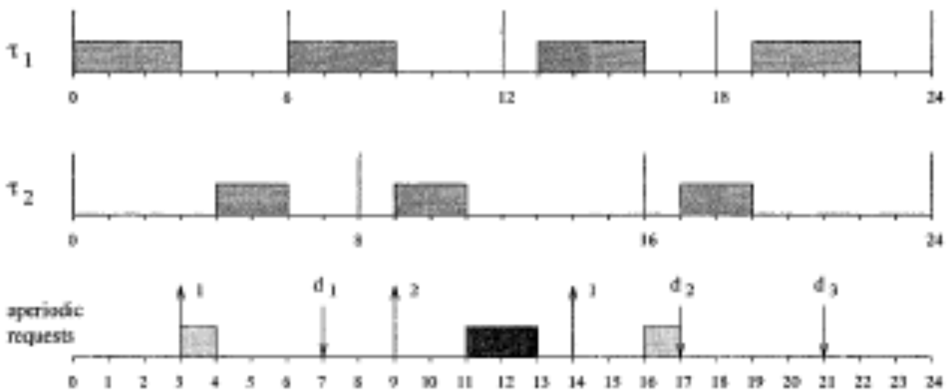


Two periodic tasks
T_1: (P_1=6, C_1=3)
T_2: (P_2=8, C_2=2)

Total bandwidth server
$U_s = 0.25$

- $d_0 = 0$
- The first aperiodic job arrives at t=3 and requires 1 unit of computation
- $d_1 = \max(r_1, d_0) + 1/0.25 = max(3,0)+4=7$

Example for TBS



Two periodic tasks

T_1: (P_1=6, C_1=3)

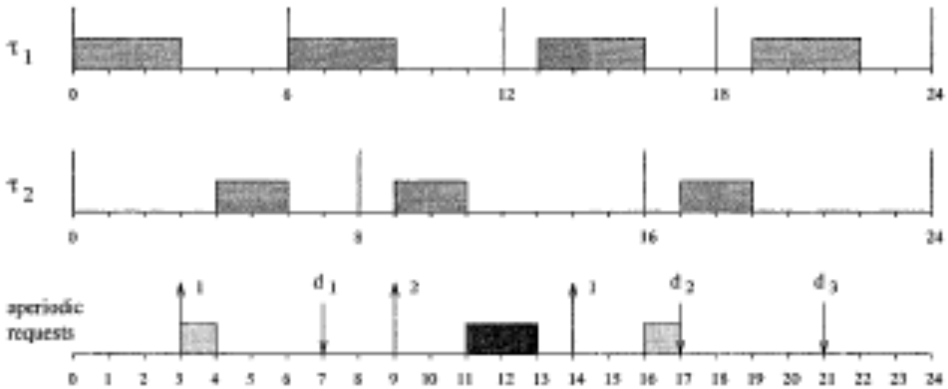
T_2: (P_2=8, C_2=2)

Total bandwidth server

$U_s = 0.25$

- $d_1 = 7$
- The second aperiodic job arrives at t=9 and requires 2 units of computation
- $d_2 = \max(r_2, d_1) + 2/0.25 = \max(9,7)+8=17$

Example for TBS



Two periodic tasks
T_1: (P_1=6, C_1=3)
T_2: (P_2=8, C_2=2)

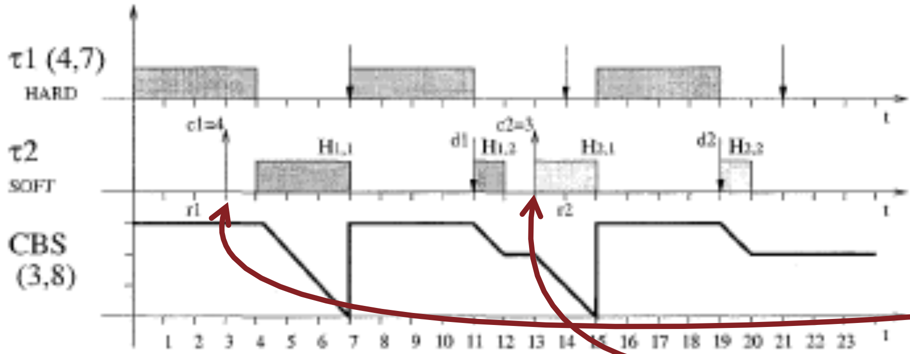
Total bandwidth server
$U_s = 0.25$

- $d_2 = 17$
- The third aperiodic job arrives at t=14 and requires 1 unit of computation
- $d_3 = \max(r_3, d_2) + 1/0.25 = \max(14,17)+4=21$

The constant bandwidth server

- Server has a maximum budget Q_s and a period P_s
- The server is said to be active if jobs are pending, otherwise it is idle
- When an aperiodic job arrives it inherits the server deadline, d_s
- When an aperiodic job executes the server budget is decreased by the same amount
- When the budget is zero it is recharged to Q_s and deadline d_s is increased by P_s
- When a job arrives at time t and the server is idle:
  - If remaining budget (C_s) > $(d_s - t)\, U_s$, the deadline is advanced to $t + P_s$
- The main advantage of the CBS is that it can deal with overruns -- when jobs exceed their estimated computation times

Example for CBS



One hard periodic task
T_1: $(P_1=7,\ C_1=4)$

One soft task served by a CBS
$P_S=8,\ Q_S=3$

Maximum budget

$J_{2,1}$ has computation time *4* and arrives at $t=3$
$J_{2,2}$ has computation time *3* and arrives at $t=13$

Current budget

The first instance of Task 2 ($J_{2,1}$) arrives at $t=3$. At $t=3$, $d_s=8$ and $C_s=3$. $C_s = 3 > (d_s\text{-}t)U_s = 15/8$. Therefore the server budget is recharged to *3* and the deadline is set to $3+8=11$.

At $t=7$, the budget is exhausted so the new deadline is set to $11+8=19$ and the budget replenished. At $t=12$, $J_{2,1}$ is complete.

At $t=13$, $J_{2,2}$ is released. $C_s = 2 < (d_s\text{-}t)U_s = 9/4$. $J_{2,2}$ starts executing with deadline *19*.

At $t=15$, the budget is exhausted. The new deadline of $19+8=27$ is assigned to the server and the budget is reset to 3. $J_{2,2}$ completes at $t=20$.

Summarizing aperiodic servers

- Quite a few aperiodic server mechanisms
  - Dynamic priority exchange server
  - Dynamic sporadic server
  - Total bandwidth server
  - Constant bandwidth server
- The difference between these schemes concerns performance and complexity (implementation, memory etc.)
- CBS is used most often: reasonable performance and easy implementation

Lecture summary

- Aperiodic task servers
  - Static priorities
    - Priority Exchange Server
    - Sporadic Server
  - Dynamic priorities
    - Dynamic Priority Exchange Server
    - Dynamic Sporadic Server
    - Total Bandwidth Server
    - Constant Bandwidth Server