Aperiodic Task Scheduling - EDF

Real-Time Systems

Gerhard Fohler

Dynamic Priority Servers

- EDF based
- **Dynamic priority exchange server**
  - Spuri, Buttazzo 1994, 1996
  - like rate monotonic priority exchange, but for EDF
- **Dynamic sporadic server**
  - Spuri, Buttazzo 1994, 1996
- **Earliest deadline late server**
  - Chetto, Chetto 1989

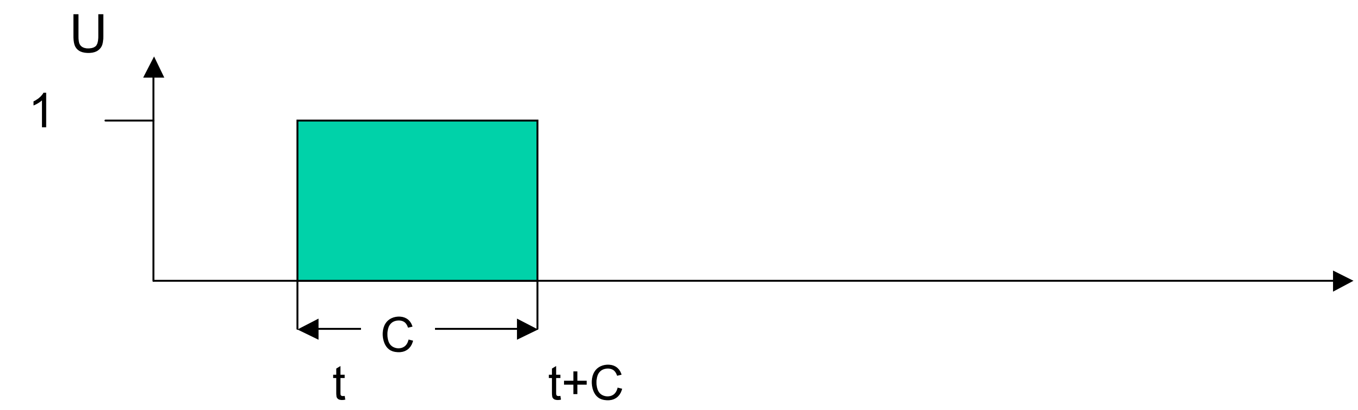
Total bandwidth server

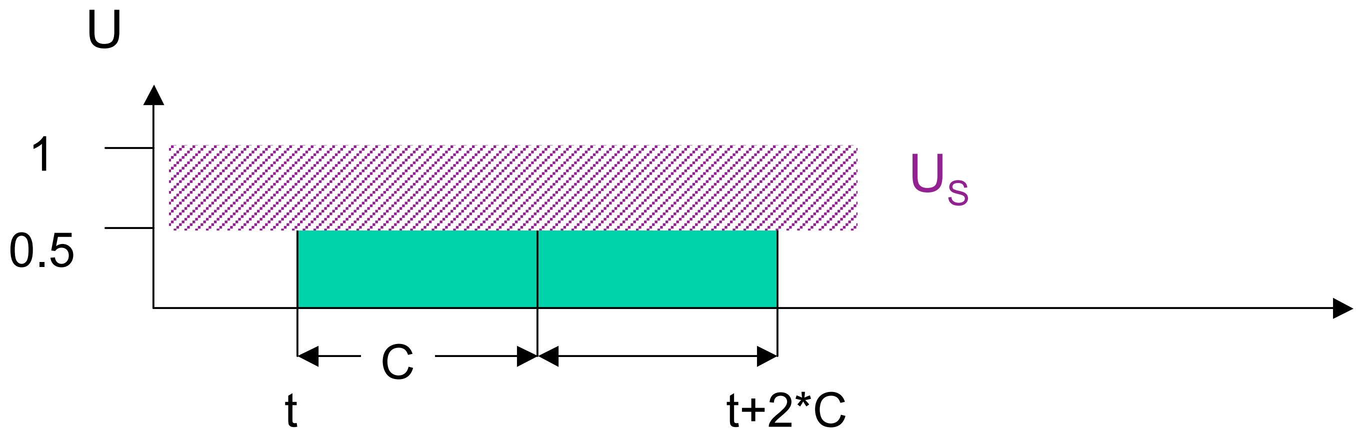
- Spuri, Buttazzo 1994, 1996
- response time dependent on server period:
  - shorter periods have shorter response times
  - but higher overhead
- how else shorter response times?
  - change the deadline of the aperiodic to earlier time
    (its EDF here, so it will get serviced earlier)
  - but make sure that total load of aperiodics does not exceed maximum value (bandwidth) U_s

How can we calculate minimum deadline for U_S?

assume we have all CPU for us:

$dl = C$





k^{th} aperiodic request

- arrival time r_k
- computation time c_k
- deadline d_k
- server utilization U_S

$d_k = \max(r_k, d_{k-1}) + C_k/U_s$
$d_0 = 0$

- uses all bandwidth of server
- very simple run-time mechanism
- no extra server task

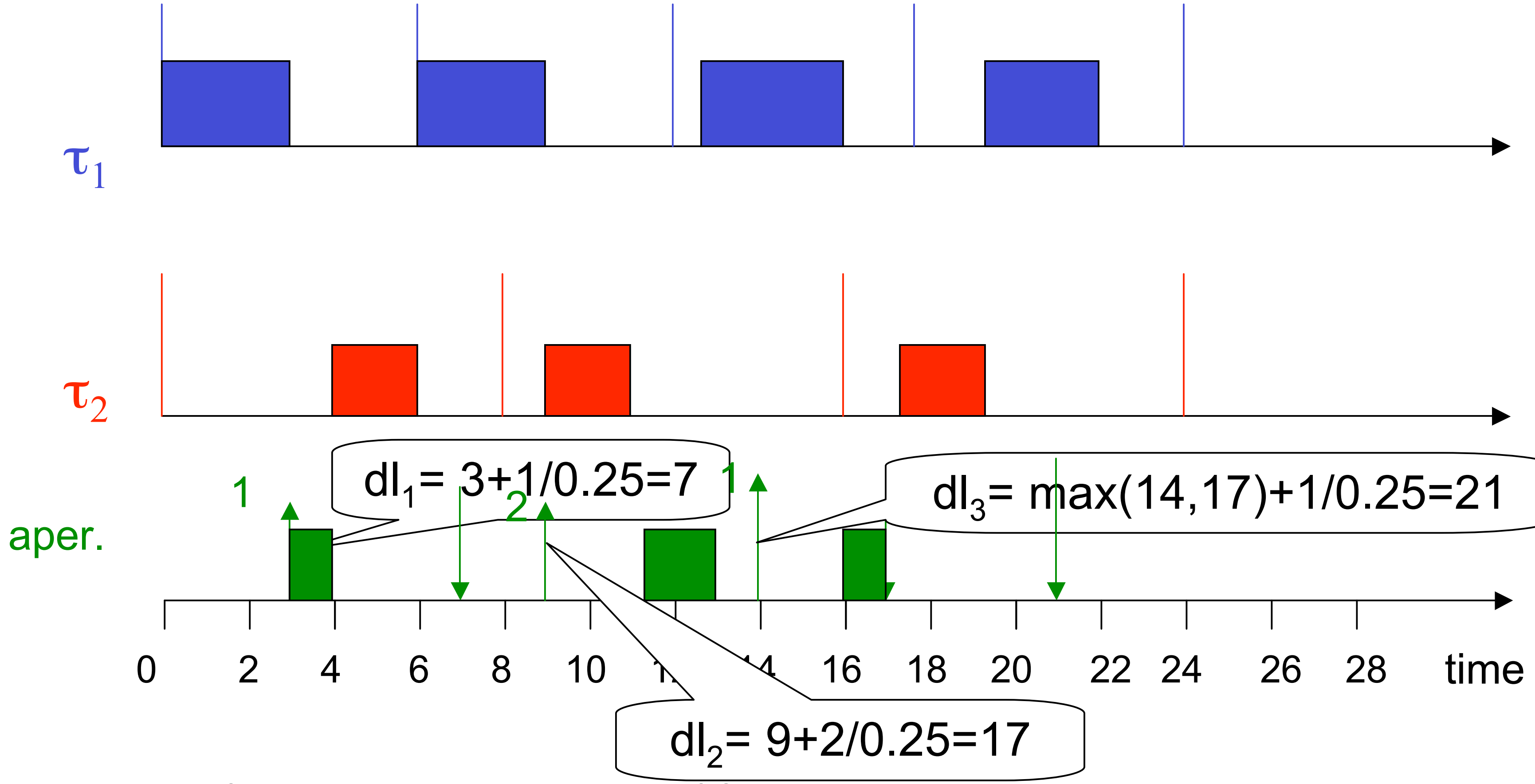
schedulability

$U_p + U_s <= 1$

Sum of periodic load and bandwidth of server less or equal 1.

Example Total Bandwidth Server

- periodic tasks τ_1 (3,6), τ_2 (2,8)
- TBS $U_s = 1 - U_p = 0.25$

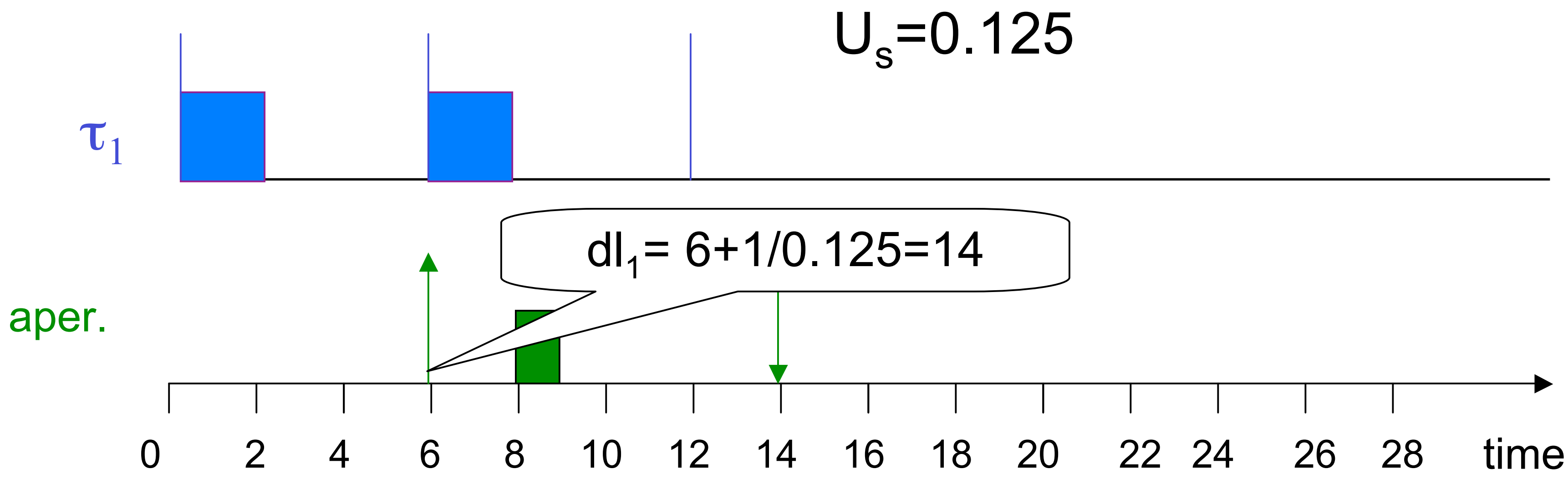


Total Bandwith Server - Comments

- based on
  - U_s not actual periodic load
  - worst case c

Total Bandwith Server - Comments

- TBS assigns deadlines based on maximum U_s (not actual load)
  $d_k = \max(r_k, d_{k-1}) + C_k/\underline{U_s}, d_0 = 0$



TB*

- Buttazzo, Sensini - 1997
- assigns deadlines d_k first according to TBS
- then shortens, as much as periodics allow
  - new $d'_k = f_k$ …finishing time according to EDF schedule, including periodics
  - apply recursively
  - maintains schedulability, since order maintained
- complexity, many steps

Constant Bandwidth Server

- Abeni and Buttazzo, 1998
- designed for multimedia applications
  - sporadic (hard) tasks
  - soft tasks: mean execution, interarrival times, not fixed
  - periodic tasks
- assign maximum bandwidth of CPU to each soft task
- handles overload of aperiodics
  - limited by assigned bandwidth
  - might slow down, but not impair effect other tasks
- EDF based

CBS Definitions

- task τ_i
  - sequence of jobs $J_{i,j}$
  - $r_{i,j}$ … request, arrival time of the j^{th} job of task τ_l
- hard task
  - (C_i,T_i)
    - C_i worst case execution time
    - T_i minimum interarrival time
    - deadline equal to next period: $d_{i,j} = r_{i,j} + T_i$
- soft task
  - (C_i,T_i)
    - C_i *mean* execution time
    - T_i *desired* interarrival time
    - soft deadline equal to next period: $d_{i,j} = r_{i,j} + T_i$

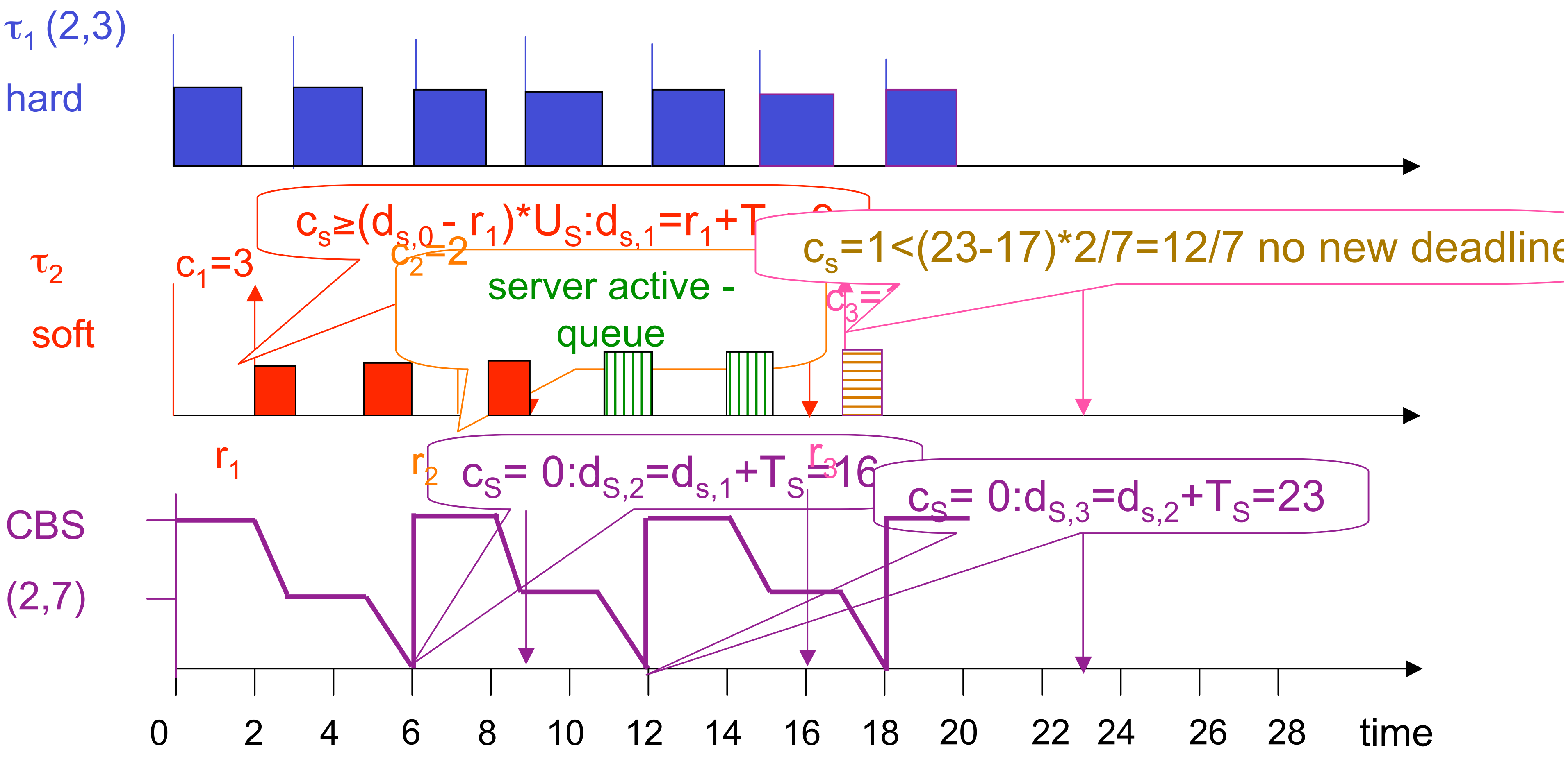
- c_S … <u>budget</u>
- (Q_S, T_S)
  - Q_S … maximum budget
  - T_S … period of server
- $U_S = Q_S / T_S$ … *<u>server bandwidth</u>*
- $d_{S,k}$ … deadline associated to server
  - initial $d_{S,0} = 0$

- job $J_{i,j}$ <u>comes in</u>, is served, <u>assigned dynamic deadline $d_{i,j}$</u> equal to current server deadline $d_{S,k}$
  - job executes, server budget c_s decreased

- $c_S=0$:
  - <u>budget recharged</u> to maximum Q_S
  - new server deadline: $d_{s,k+1}=d_{s,k}+T_S$

- $J_{i,j}$ arrives, CBS active (jobs pending): put in queue
- $J_{i,j}$ arrives, CBS idle:
  - <u>$c_S \geq (d_{s,k} - r_{i,j})*U_S$</u>:
    - <u>new deadline</u> $d_{s,k+1} = r_{i,j} + T_S$
    - c_S recharged to Q_S
  - else
    - job served with last server deadline $d_{s,k}$
- job finishes: next job in queue
- at any time, job assigned last deadline generated by server

Example CBS



τ1 (2,3)
hard
τ2
soft
c1=3
c2=2
c3=
server active - queue
cs=1<(23-17)*2/7=12/7 no new deadline
cs= 0:dS,2=ds,1+TS=16
cs= 0:dS,3=ds,2+TS=23
r1
r2
r3
CBS
(2,7)
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
time

- limits impact “harm” by ill behaved aperiodics, e.g., exec time overrun
- various improvements
  - several servers
  - capacity exchange
  - feedback control
  - …..

Articles

- TBS:
  Spuri, Buttazzo
  “Efficient Aperiodic Service under Earliest Deadline Scheduling”
  Proceedings of the 15th IEEE Real-Time System Symposium
  (RTSS 94), Portorico, pp. 2-21, December 1994
- CBS:
  L. Abeni and G. Buttazzo, "Integrating Multimedia Applications in Hard Real-Time Systems", Proceedings of the IEEE Real-Time Systems Symposium, Madrid, Spain, pp. 4-13, December 1998.

Schedulability Analysis

First show that aperiodic load executed not exceeds U_S of server

Lemma: *In each interval of time $[t_1,t_2]$, if C_{ape} is the total execution time demanded by aperiodic requests arrived at t_1 or later and served with deadlines less than or equal to t_2, then*

$$C_{ape} \leq (t_2 - t_1)\, U_S$$

Proof: by definition:

$$C_{ape} = \sum_{t_1 \leq r_k, d_k \leq t_2} C_k$$

- TB* uses periodic interference…can now calculate it

- (formulae for completeness only)
  $I_f(t, d_k^s) =$

$$\sum_{i=1}^{n} \max\left(0, \left\lceil \frac{d_k^s - next\_r_i(t)}{T_i} \right\rceil - 1\right) C_i$$

next_r$_i$(t)…time at which next instance of I after t starts

TBS assigns deadlines in increasing order,

therefore there must exist two aperiodic requests with indeces k_1 and k_2 such that

$$\sum_{t_1 \le r_k, d_k \le t_2} C_k = \sum_{k=k_1}^{k_2} C_k$$

$$C_{ape} = \sum_{k=k_1}^{k_2} C_k = \sum_{k=k_1}^{k_2} [d_k - \max(r_k, d_{k-1})] * U_S$$

$$\le [d_{k_2} - \max(r_{k_1}, d_{k_1 -1})] * U_S$$

□

$$\le (t_2 - t_1) * U_S$$

Proof main result:

Theorem: Given a set of n periodic tasks with processor utilization U_p and a TBS with processor utilization of U_S, the whole set is schedulable by EDF if and only if

$U_p + U_S \leq 1$

Proof: If:

- assume $U_p + U_S \leq 1$ plus overflow at time t
- overflow preceded by continuous utilization
- from a point t' on ($t' < t$), only instances of tasks ready at t' or later *and* having deadlines less than or equal to t are run
- C total execution time demanded by these instances
- since there is overflow at t: $t - t' < C$

- we also know that

$$C \leq \sum_{i=1}^{n} \left\lfloor \frac{t - t^{'}}{T_i} \right\rfloor * C_i + C_{ape}$$

$$\leq \sum_{i=1}^{n} \frac{t - t^{'}}{T_i} * C_i + (t - t^{'}) * U_S$$

$$\leq (t - t^{'}) * (U_p + U_S)$$

it follows: $U_p + U_S > 1$ …# contradiction

- **only if:**
- assume aperiodic request enters periodically with period T_S and execution time $C_S=T_SU_S$, then server behaves like periodic task
- total utilization of processor is then U_p+U_S
- if task set schedulable: $U_P+U_S \leq 1$

□