Inovace tohoto kurzu byla v roce 2011/12 podpořena projektem
CZ.2.17/3.1.00/33274 financovaným Evropským sociálním fondem
a Magistrátem hl. m. Prahy.

OPPA PRAHA PRAGUE PRAGA PRAG EVROPSKÁ UNIE

Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Embedded and Real-Time Systems

Dynamic-Priority Servers

Scheduling of Aperiodic Tasks

- When servers scheduled in EDF, we call them dynamic priority servers

- In this part we assume
  - Periodic tasks scheduled by EDF
  - All periodic tasks (τ_i) have hard deadlines
  - All aperiodic tasks (J_i) do not have deadlines
  - All periodic tasks start at t=0 and relative deadlines are equal to periods
  - Each aperiodic request has a known computation time but an unknown arrival time

Dynamic Priority Exchange Server

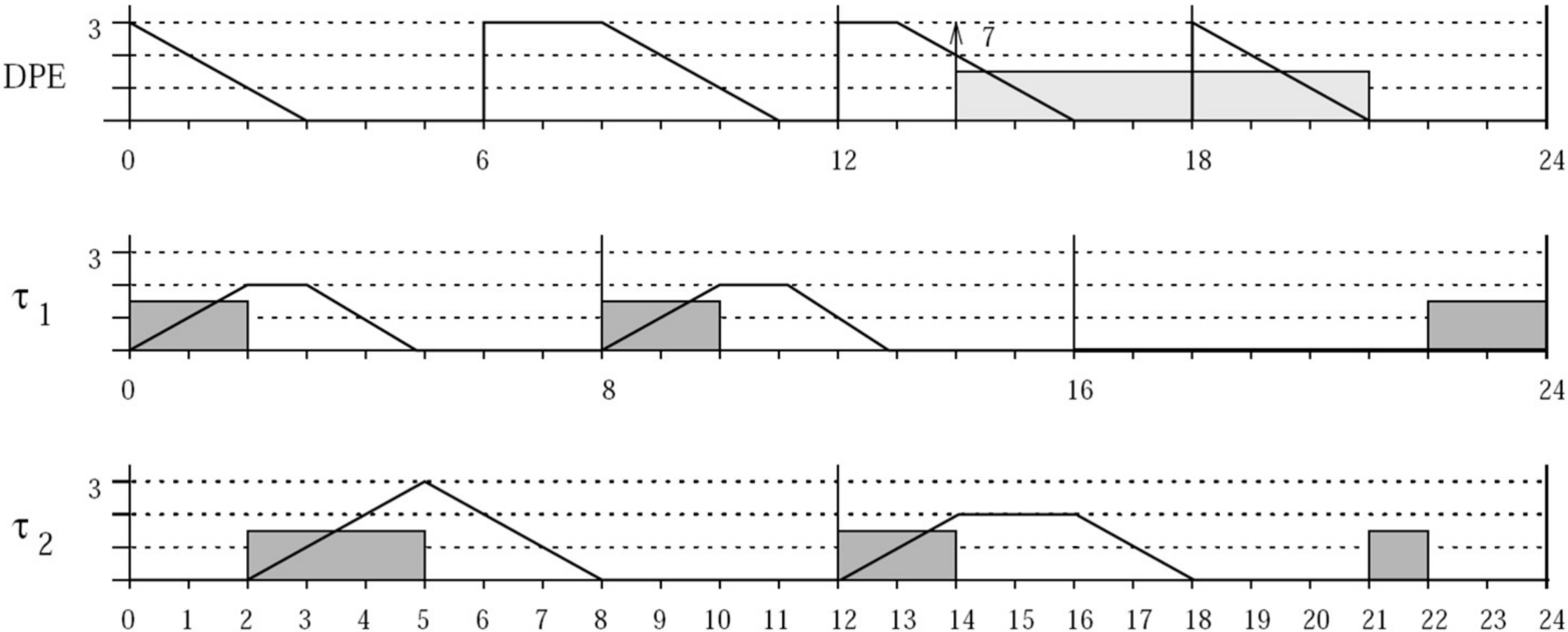
- Extension of PE to work under EDF

  DPE has a period T_s and a capacity C_s

  At the beginning of each period, the server's *aperiodic* capacity is set to C_S^d, where d is the deadline of the current server period

  Aperiodic capacities (those greater than 0) receive priorities according to their deadlines, like all the periodic task instances

DPE Example


DPE
3
7
0
6
12
18
24
τ 1
3
0
8
16
24
τ 2
3
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

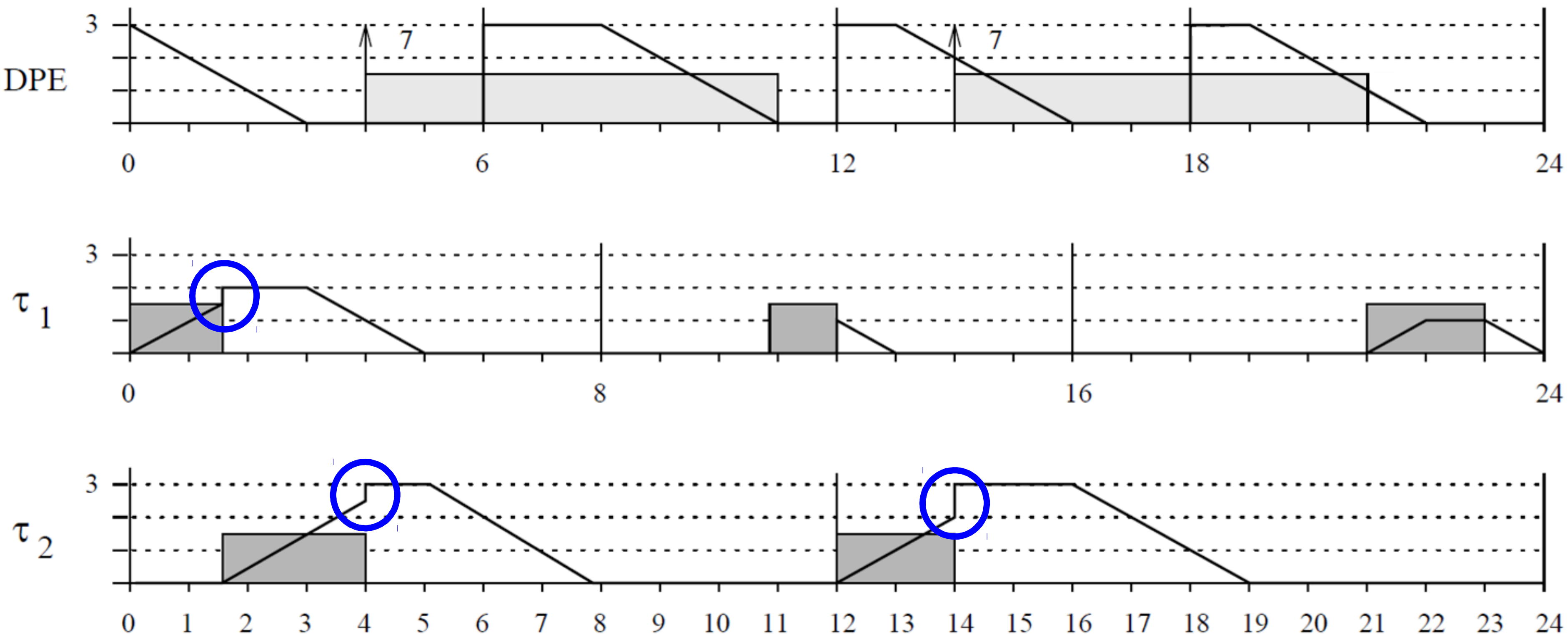
Schedulability Analysis

- Exchanging the priorities does not influence the schedulability.
- Thus the criterion is the same as under EDF

$$U_p + U_s \leq 1$$

Reclaiming Spare Time

- When task finishes before it's WCET, it is possible to add the remaining time to the aperiodic capacity



Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

Dynamic Sporadic Server

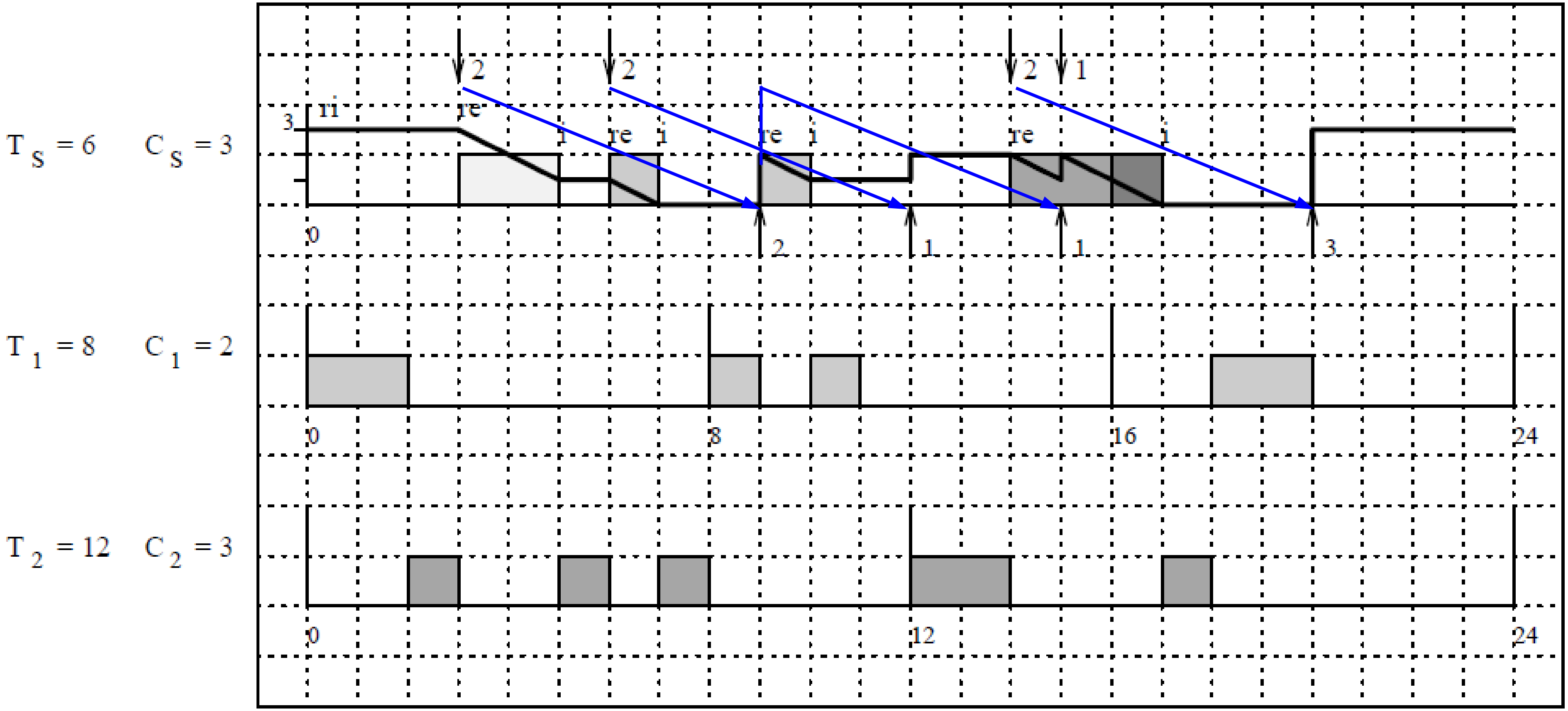
- Similar to sporadic server, only differs in how the priority (i.e. the deadline) is assigned

When the server is created, its capacity C_S is initialized at its maximum value.

The next replenishment time RT and current deadline d_S are set as soon as $C_S > 0$ and there is an aperiodic request pending. If t_A is such a time, then $RT = d_s = t_A + T_S$.

The replenishment amount RA to be done at time RT is computed when the last aperiodic request is completed or C_S has been exhausted. If t_I is such a time, then the value of RA is set equal to the capacity consumed within the interval $[t_A, t_I]$.

Dynamic Sporadic Server – Example



Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

Schedulability Analysis

- DSS behaves almost like a periodic task, thus the criterion is the same as under EDF

$$U_p + U_s \leq 1$$

Total Bandwidth Server

- When DSS has a long period, execution of aperiodic requests may take quite long.
  - One option is to shorten the period of the server
  - Another option is to assign an earlier deadline
- TBS assigns the earlier deadline
  - Does it in a way that the server never exceeds some pre-defined utilization

Total Bandwidth Server

- When kth aperiodic request arrives at time $t = r_k$, it receives a deadline

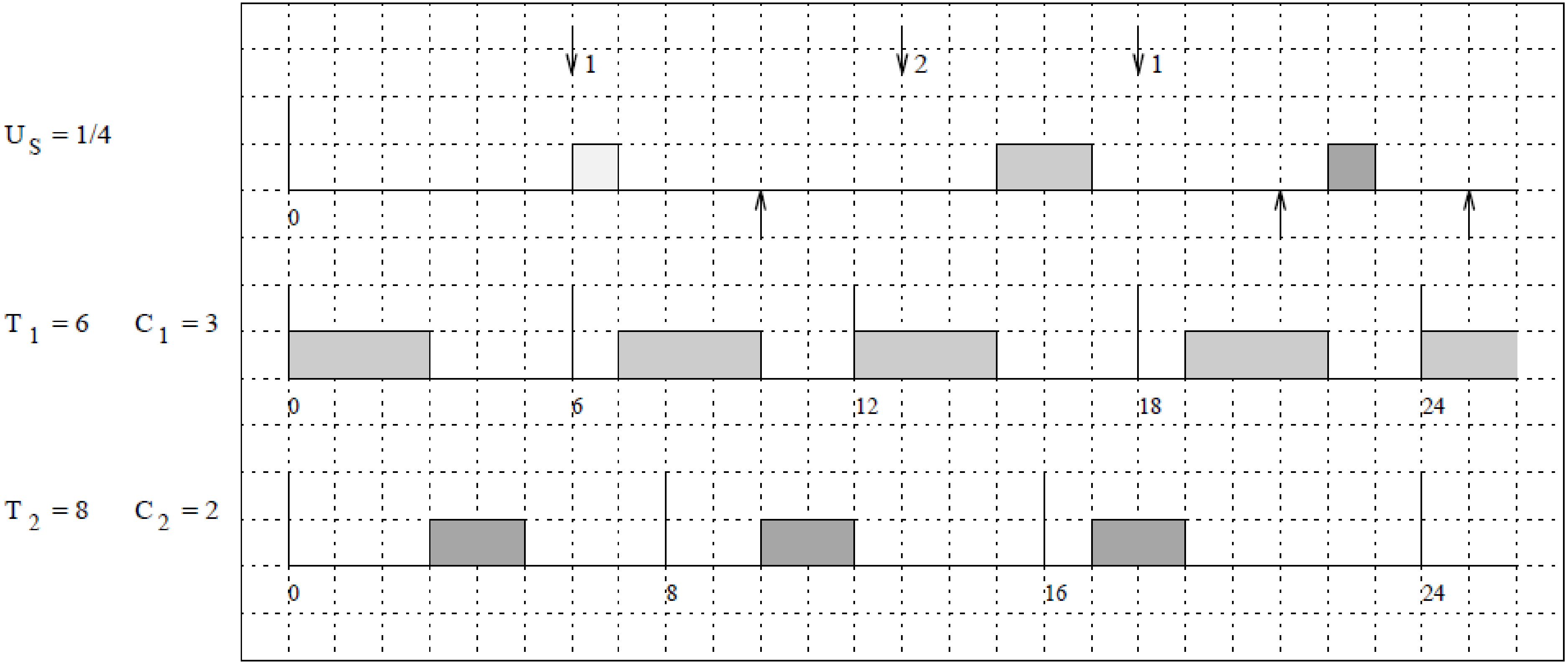
$$d_k = \max(r_k, d_{k-1}) + \frac{C_k}{U_S}$$

C_k is the execution time of the request,

U_S is the server utilization factor (bandwidth).

By definition $d_0 = 0$.

Total Bandwidth Server – Example



Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

Schedulability Analysis

- It again holds that a periodic task set together with a TBS is schedulable if and only if

$$U_p + U_s \leq 1$$

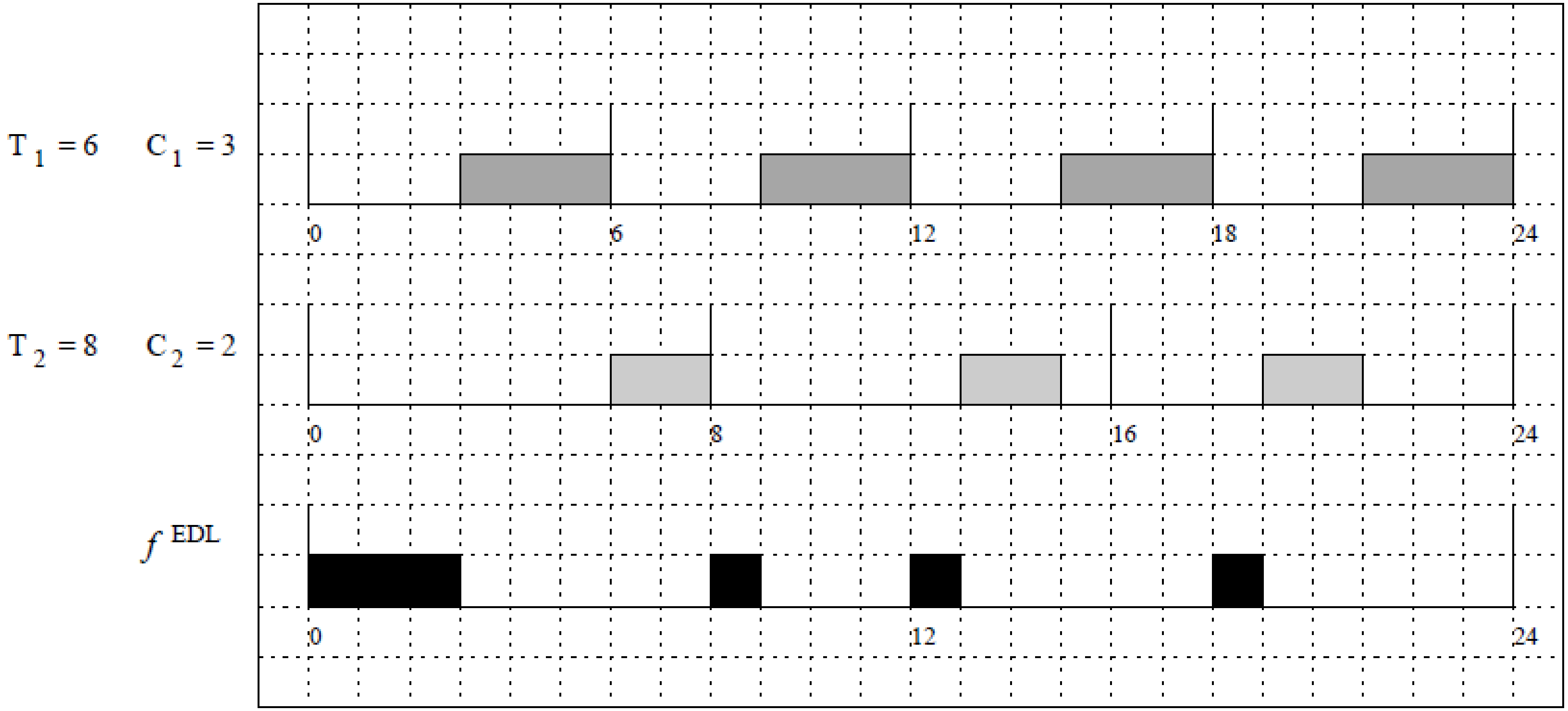
Improved Priority Exchange Server

- Improves the responsiveness of DPE by delaying periodic tasks as much as possible.
- If it has a capacity, it runs as the highest priority task regardless deadlines of the other tasks
- Replenishes capacity only at precomputed times with a precomputed value, otherwise is the same as DPE

- Has similar complexity of DPE, but can have bigger memory demand (to store the precomputed values)

IPE Slack Times

- In a regular schedule, it is possible to postpone periodic tasks such as it does not influence schedulability

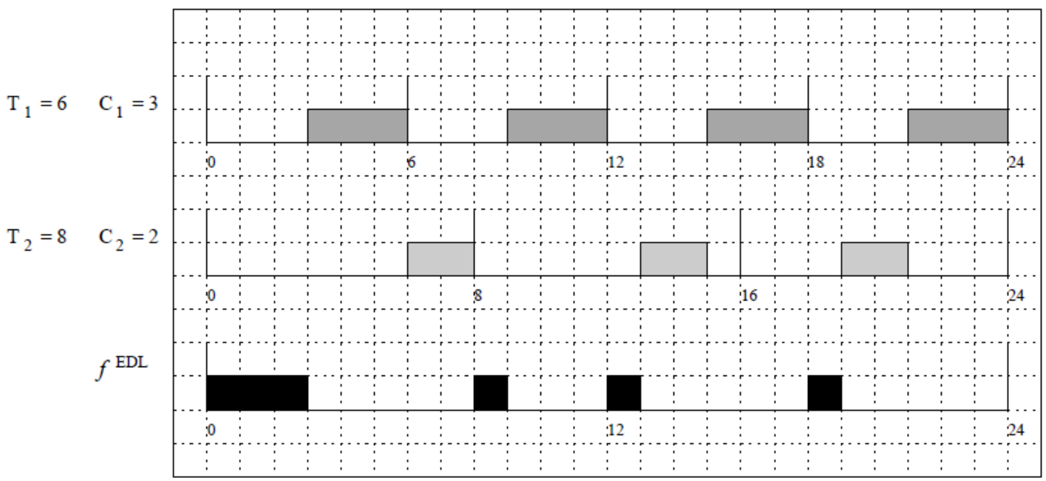


Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

IPE Slack Times

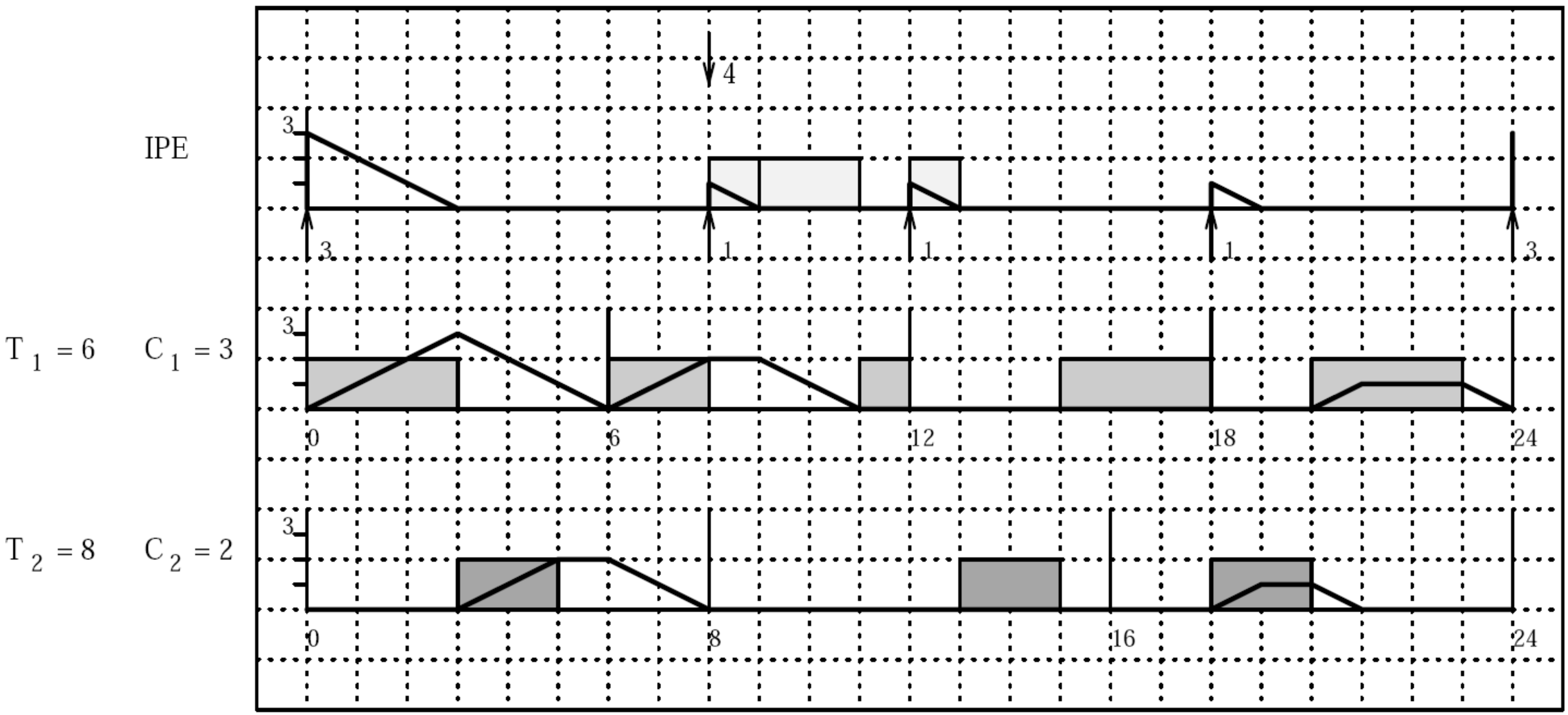
- Slack times appear only at the start of some period
- They can be represented in a table

i	0	1	2	3
e_i	0	8	12	18
Δ_i^*	3	1	1	1



Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

IPE – Example



Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

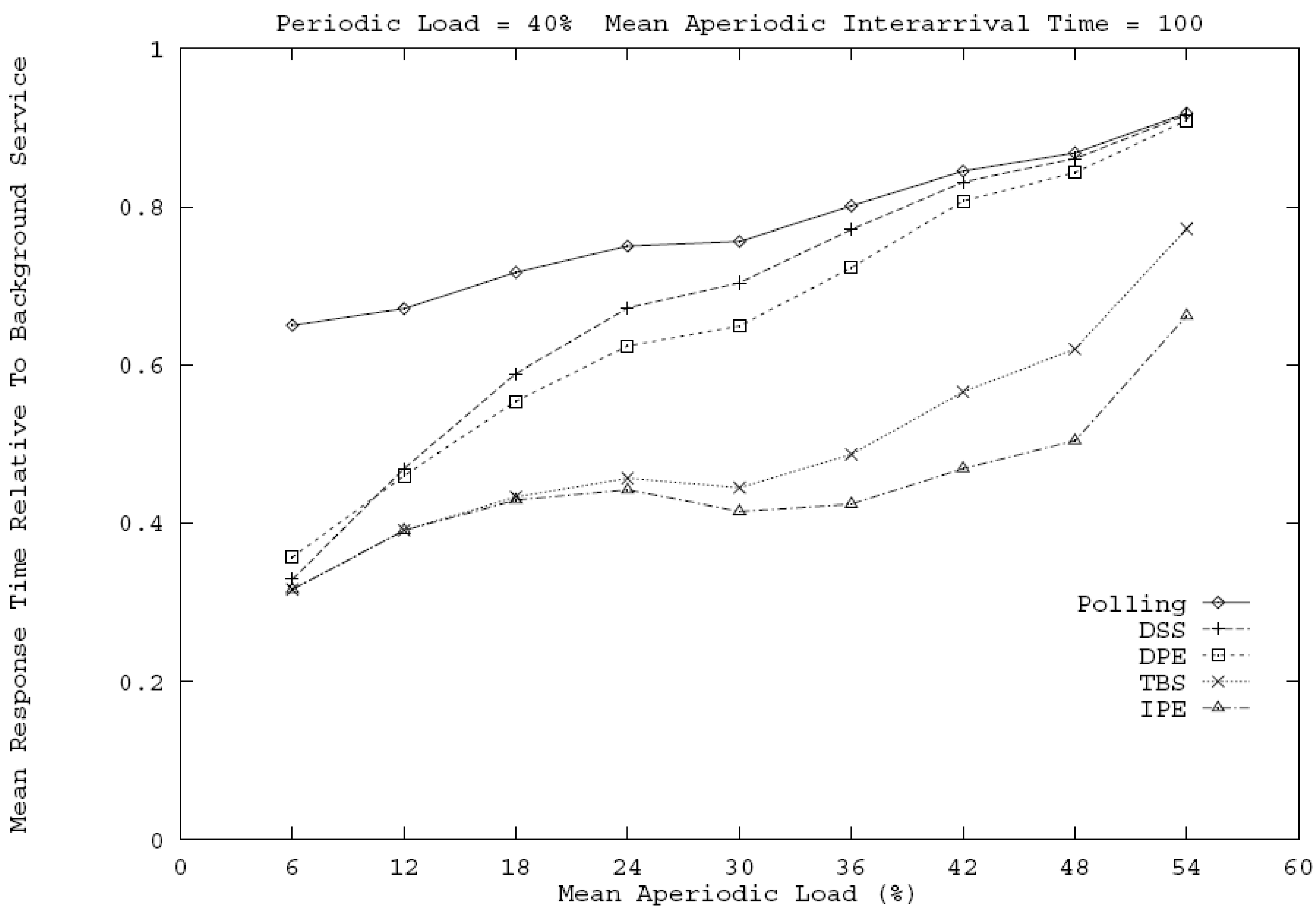
IPE – Schedulability

- The whole task set is schedulable if on only if

$$U_p \leq 1$$

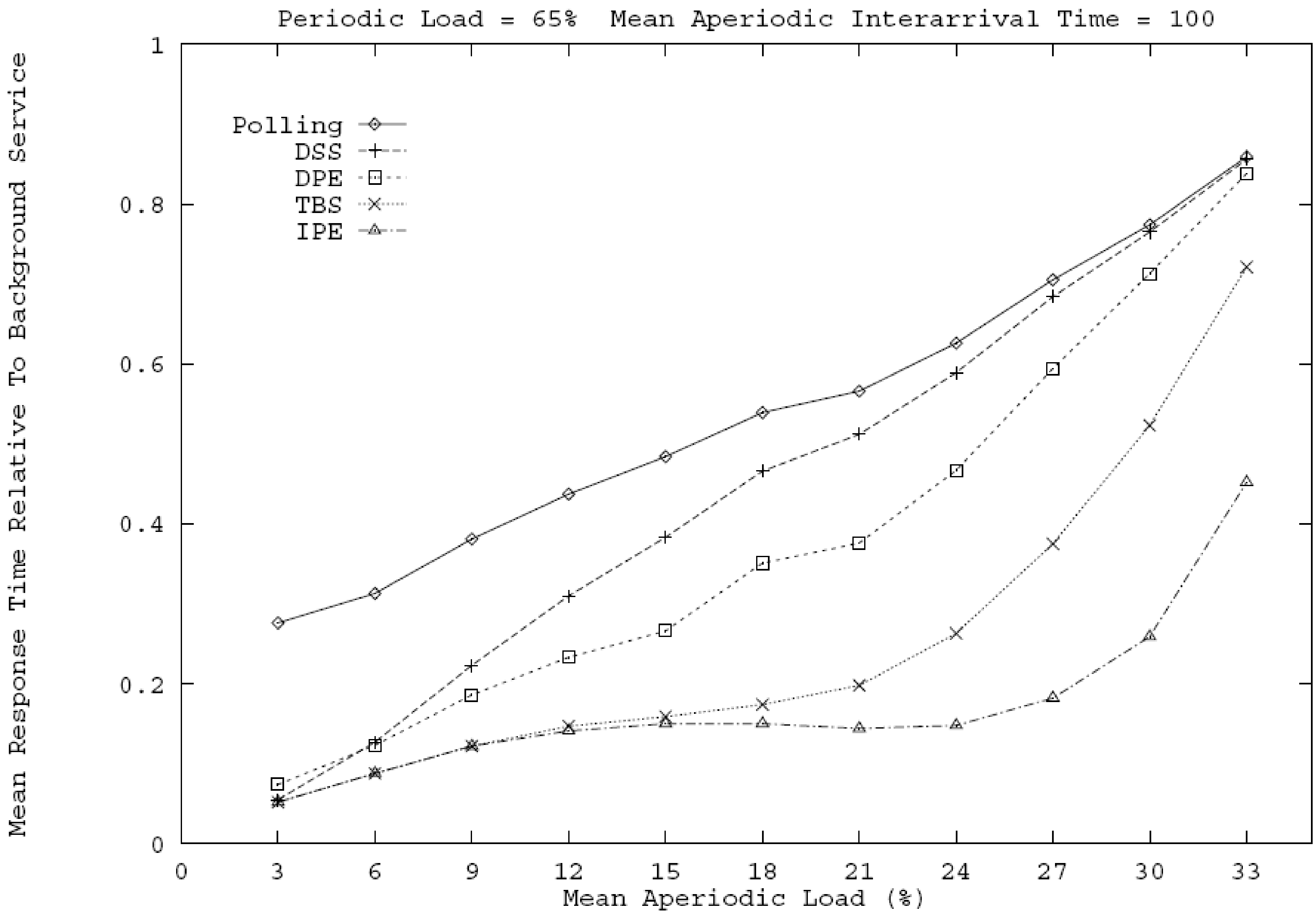
- The server automatically allocates the bandwidth $1 - U_p$ to aperiodic requests.

IPE Performance



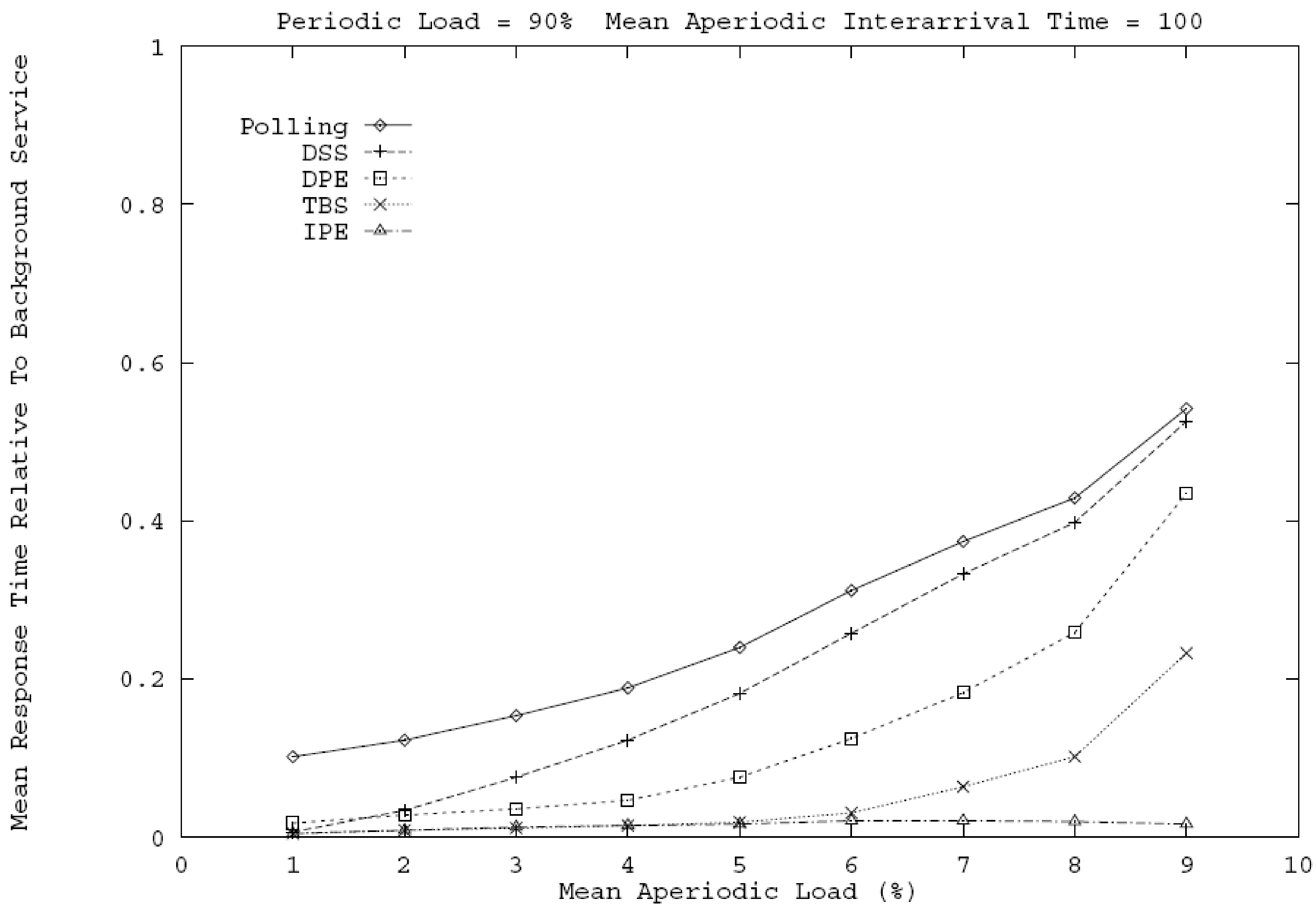
Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

IPE Performance



Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

IPE Performance



Figures taken from Spuri, M., Buttazzo, G.`:Scheduling Aperiodic Tasks in Dynamic Priority Systems

Constant Bandwidth Server

- Guarantees utilization (similar to TBS) and is almost as efficient as TBS.
- Can handle aperiodic task overruns (contrary to TBS)
  - This happens when WCET estimation of aperiodic task is wrong

- Basic idea
  - When a new job enters the system, it is assigned a suitable scheduling deadline and inserted in EDF ready queue
  - If the job tries to execute more than expected, its deadline is postponed (i.e. priority decreased)

Constant Bandwidth Server

- A CBS is characterized by a budget c_s and by an ordered pair (Q_s, T_s), where Q_s is the maximum budget and T_s is the period of the server. The ratio $U_s = Q_s/T_s$ is denoted as the server bandwidth. At each instant, a fixed deadline $d_{s,k}$ is associated with the server. At the beginning $d_{s,0} = 0$.
- Each served job $J_{i,j}$ is assigned dynamic deadline $d_{i,j}$ equal to current server deadline $d_{s,k}$.
- Whenever a served job executes, the budget c_s is decreased by the same amount.
- When $c_s = 0$, the server budget is recharged at the maximum value Q_s and a new server deadline is generated as $ds, k+1 = d_{s,k} + T_s$.

Constant Bandwidth Server

- A CBS is said to be active at time t if there are pending jobs; that is, if there exists a served job Ji, j such that $ri, j \leq t < f_{i,j}$. A CBS is said to be idle at time t if it is not active.
- When a job $J_{i,j}$ arrives and the server is idle, if $c_s \geq (d_{s,k} - r_{i,j})U_s$ the server generates a new deadline $d_{s,k+1} = r_{i,j} + T_s$ and c_s is recharged at the maximum value Q_s, otherwise the job is served with the last server deadline $d_{s,k}$ using the current budget.
- When a job finishes, the next pending job, if any, is served using the current duget and daedline. If there are no pending jobs, the server becomes idle.
- At any instant, a job is assigned the last deadline generated by the server.

CBS – Example

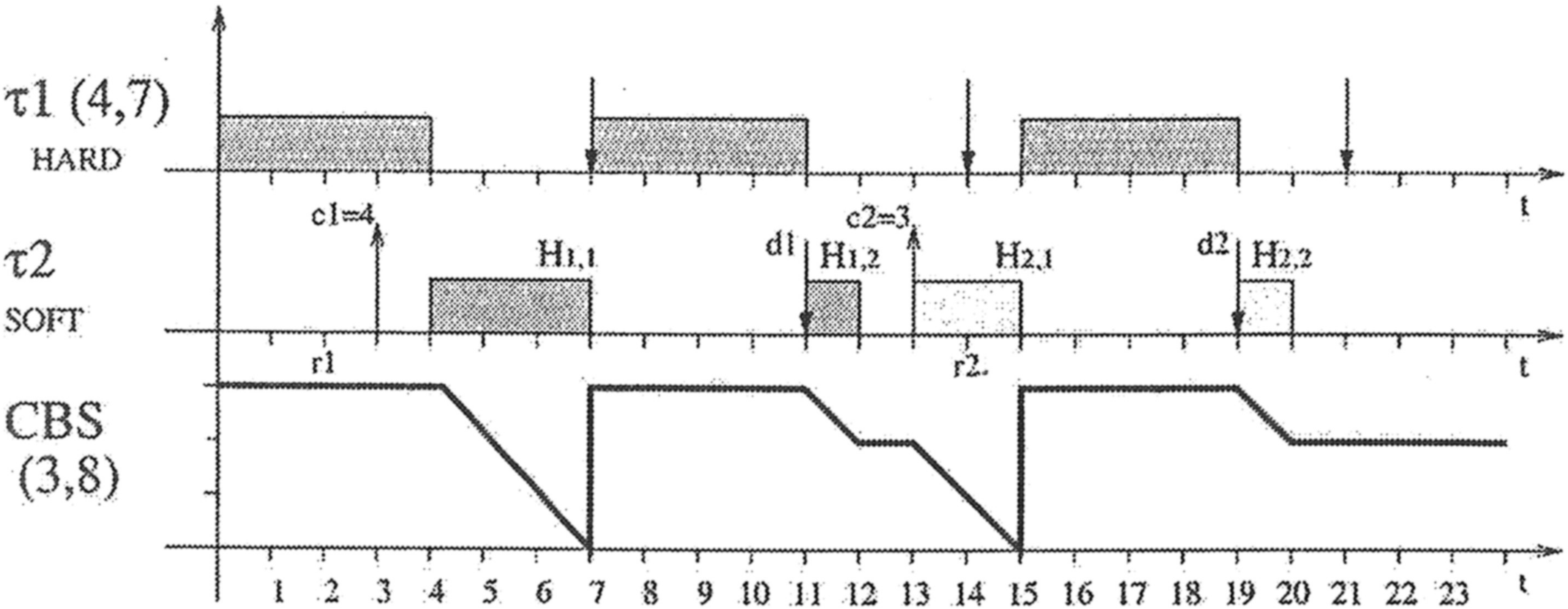


Figure taken from Buttazzo, G.:Hard Real-Time Computing Systems

CBS Properties

- The CPU utilization of a CBS S with parameters (Q_s, T_s) is $U_s = \frac{Q_s}{T_s}$, independently from the computation times and the arrival pattern of the served jobs.

- Given a set of n periodic hard tasks with processor utilization U_p and a set of m CBSs with processor utilization $U_s = \sum_{i=1}^{m} U_{s_i}$, the whole set is schedulable by EDF if and only if

$$U_p + U_s \leq 1$$

- The CBS automatically reclaims any spare time caused by early completions.

CBS Performance

DSS waits till capacity is replenished, thus cannot efficiently use the time between hard tasks

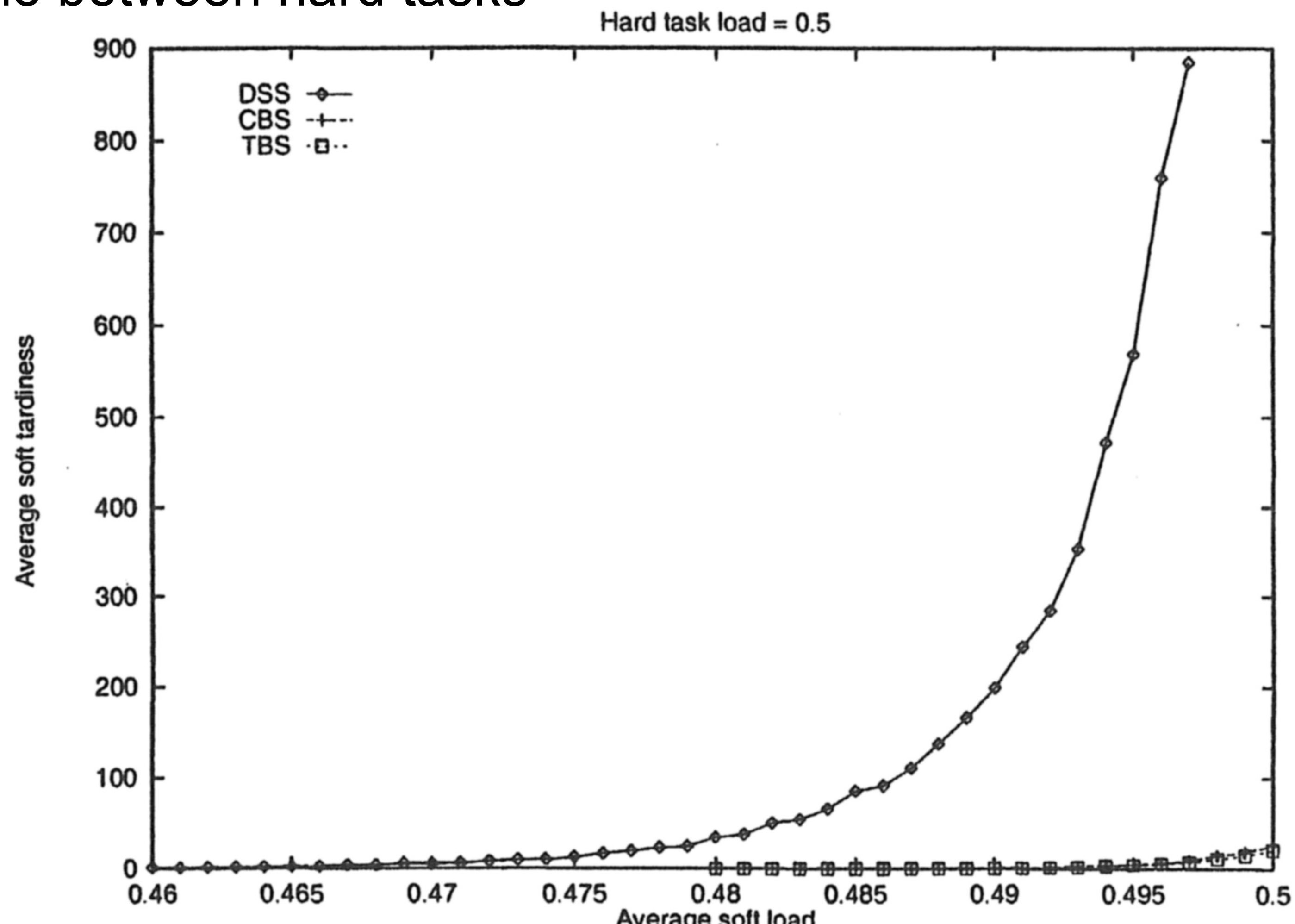


Figure taken from Buttazzo, G.:Hard Real-Time Computing Systems

CBS Performance

TBS outperforms CBS in average tardiness

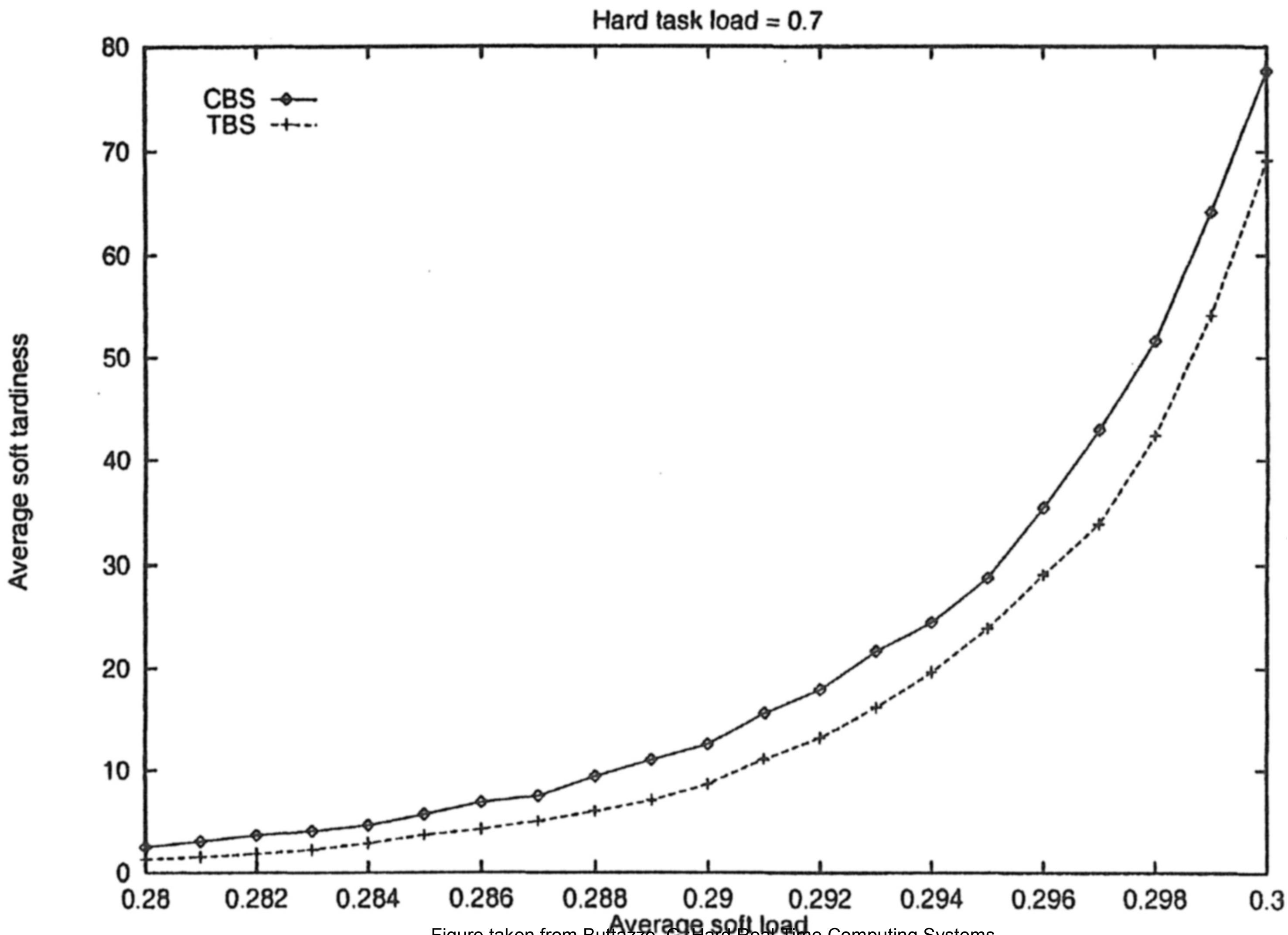


Figure taken from Buttazzo, G.: Hard Real-Time Computing Systems

CBS Performance

However, TBS adapts poorly to variance in computation times

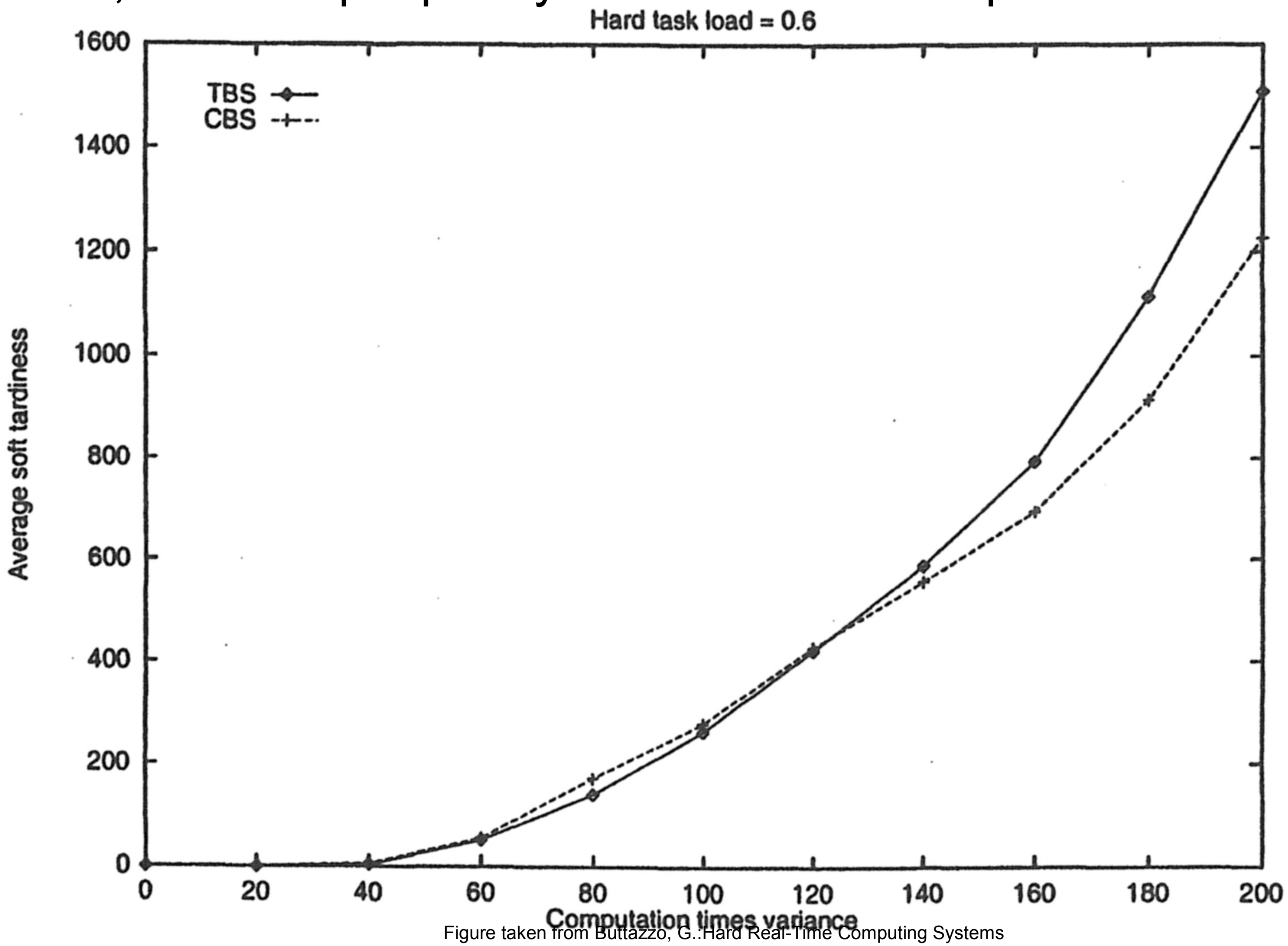


Figure taken from Buttazzo, G.:Hard Real-Time Computing Systems

Evaluation

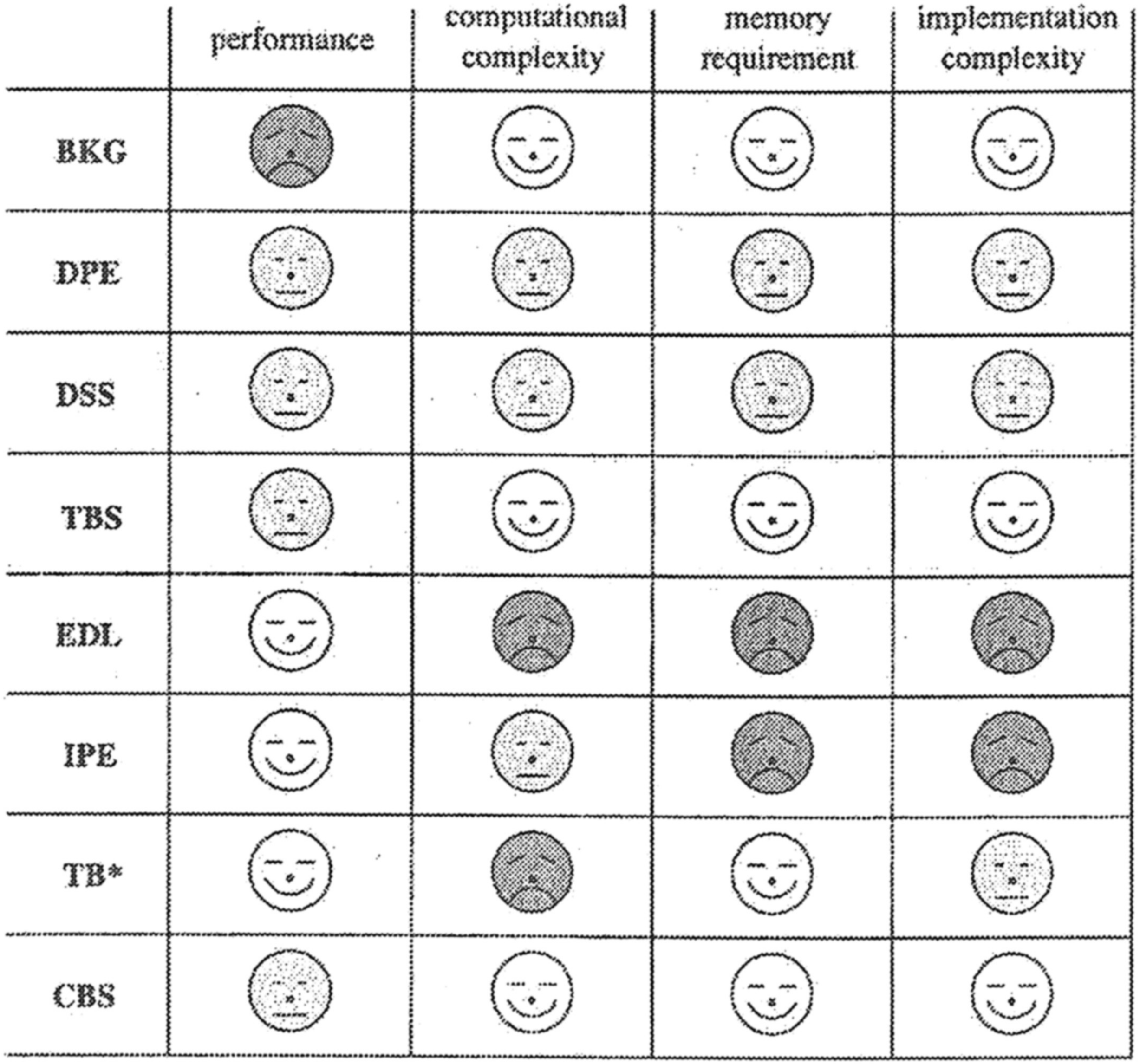


Figure taken from Buttazzo, G.:Hard Real-Time Computing Systems