Sporadic Server Algorithm

Elena Ramon

Nicola Serlonghi

Introduction

- Tries to serve aperiodic requests as soon as they arrive
- Enhance the average response time of aperiodic tasks
- Preserve the schedulability of periodic tasks

How does it work?

P_s	Server priority
P_{exe}	Running task priority

- It creates a high-priority task for servicing aperiodic requests
- The server preserves its capacity until an aperiodic request occurs
- The server replenishes its capacity only when it has been used
- The server is active when $P_{exe} \geq P_s$
- The server is idle when $P_{exe} < P_s$

Replenishment rule

RT	Time at which the server capacity will be replenished	Let t_A be the time at which the server becomes active and $C_s > 0$: $RT = t_A + T_s$
RA	Amount that will be added to the server capacity	Let t_I be the time at which the server becomes idle or $C_s = 0$: RA = C_s consumed in the interval $[t_A, t_I]$

Example

	C_i	T_i
τ_1	1	5
τ_2	4	15

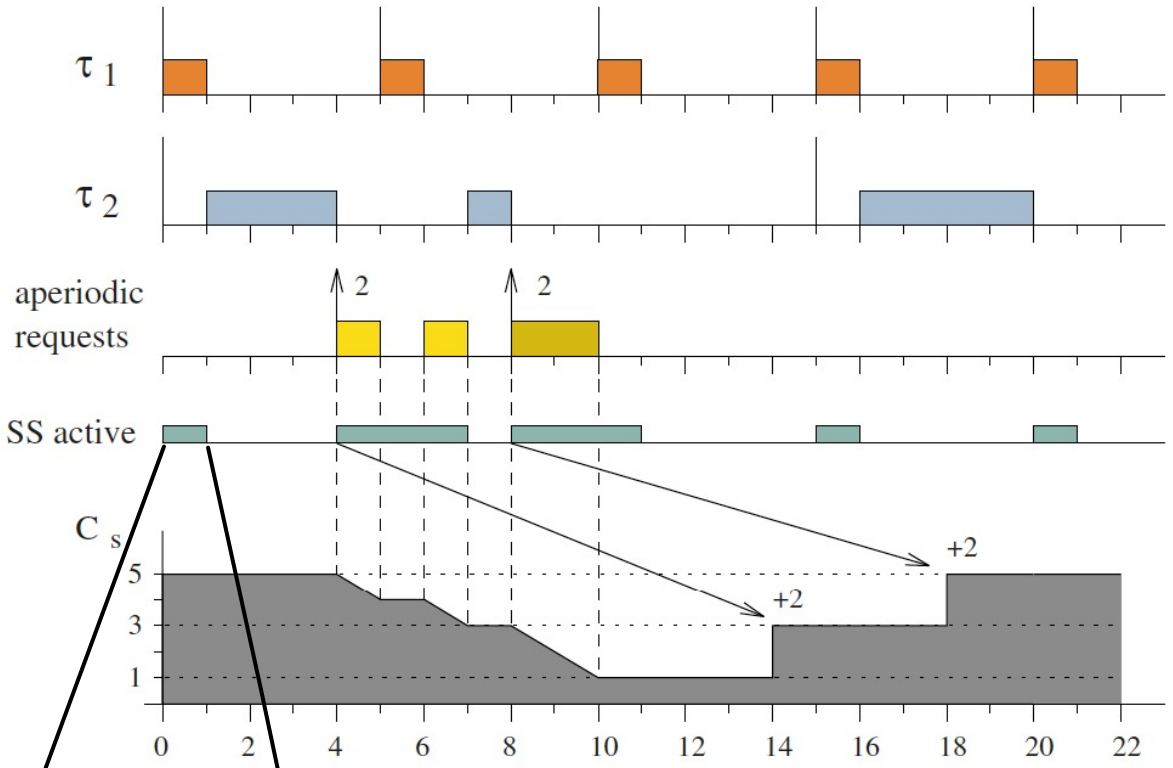
Server

$C_s = 5$

$T_s = 10$

	C_i	A_i
ape1	2	4
ape2	2	8

t_A	0	4	8
t_I	1	7	11
RT	10	14	18
RA	0	2	2



$P_{exe} \geq P_s$

$P_{exe} < P_s$

Schedulability analysis

Let $U_S = \frac{C_S}{T_S}$, the processor utilization factor of the server, a set of n periodic tasks with utilization factor U_P is schedulable if

$$U_P + U_S \leq U_S + n\left[\left(\frac{2}{U_S + 1}\right)^{1/n} - 1\right]$$

Apply to the previous example: $0{,}46 + 0{,}50 \leq 0{,}50 + 0{,}30$