Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- More Definitions
  - A task that behaves more or less like a periodic task and is created for the purpose of executing aperiodic jobs is called a ***periodic server***.
  - A periodic server, $T_{PS} = (\varphi_{PS}, p_{PS}, e_{PS})$ never executes for more than e_{PS} units of time in any time interval of length p_{PS}.
  - The parameter e_{PS} is called the ***execution budget*** (or simply ***budget***) of the periodic server.
  - The ratio $u_{PS} = e_{PS} / p_{PS}$ is the ***size*** of the periodic server.
  - A poller is a kind of periodic server; at the beginning of each period, the budget of the poller is set to e_S – i.e. its budget is ***replenished*** by e_S units
  - A time instant when the server budget is replenished is called a ***replenishment time***.

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Yet More Definitions
  - A periodic server is ***backlogged*** whenever the aperiodic job queue is nonempty
  - The server is ***idle*** if the queue is empty
  - The server is ***eligible*** (ready) ***for execution*** ONLY WHEN IT IS BACKLOGGED AND HAS BUDGET
  - The server is scheduled just like any other periodic task based upon the priority scheme used by the scheduling algorithm
  - When the server is scheduled and executes aperiodic jobs, it ***consumes*** its budget at the rate of 1 per unit time
  - The server budget has been ***exhausted*** when the budget becomes 0.
  - Different kinds of periodic servers differ in how the server budget changes when the server still has budget but the server is idle.

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Bandwidth-preserving server algorithms
  - A deficiency of the polling server algorithm is that if the server is scheduled when it is not backlogged, it loses its execution budget until it is replenished when it is next released; an aperiodic job arriving just after the polling server has been scheduled and found the aperiodic job queue empty will have to wait until the next replenishment time
  - We would like to be able to ***preserve*** the execution budget of the server when it finds an empty queue, such that it can execute an aperiodic job that arrives later in the period if doing so will not affect the correctness of the schedule
  - Algorithms that improve the polling approach in this manner are called ***bandwidth-preserving server algorithms***

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Attributes of bandwidth-preserving (BP) algorithms
  - Bandwidth-preserving servers are periodic servers
  - Each algorithm is defined by a set of ***consumption*** and ***replenishment*** rules
    - Consumption – conditions under which the execution budget is preserved and consumed
    - Replenishment – how the budget is replenished
  - How do such servers work?
    - A backlogged BP server is ready for execution when it has budget.
    - The scheduler keeps track of the consumption of the server budget and suspends the server when the budget is exhausted or the server becomes idle.
    - The scheduler moves the server back to the ready queue once it replenishes the server budget if the server is backlogged at that time.
    - The server suspends itself whenever it finds the aperiodic job queue empty – i.e. when it becomes idle.
    - When the server becomes backlogged again upon the arrival of an aperiodic job, the scheduler puts the server back on the ready queue if the server has budget at that time.

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Deferrable server
  - The simplest of BP servers
  - Consumption rule – the execution budget of the server is consumed at the rate of one per unit time whenever the server executes
  - Replenishment rule – the execution budget of the server is set to e_S at time instants kp_S, for k = 0, 1, 2, …
  - Note that the server is not allowed to carry over budget from period to period

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Schedulability
  - To determine the schedulability of a fixed-priority system with a deferrable server, we need to determine the situation that corresponds to a critical instant
  - Often, the deferrable server is given the highest priority in a fixed-priority system
  - In a fixed-priority system T in which $D_i \leq p_i$ for all i, and there is a deferrable server (, p_S, e_S) with the highest priority among all tasks, a critical instant of every periodic tasks T_i occurs at a time t_0 when all of the following are true:
    - One of its jobs $J_{i,c}$ is released at t_0
    - A job in every higher-priority periodic task is released at t_0
    - The budget of the server is e_S at t_0, one or more aperiodic jobs are released at t_0, and they keep the server backlogged hereafter
    - The next replenishment time of the server is $t_0 + e_S$

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Time-demand analysis for fixed-priority systems
  - This definition of critical instant is identical to that for the periodic tasks without the deferrable server + the worst-case requirements for the server
  - Therefore, the expression for the time-demand function becomes $w_i(t) = e_i + [t/p_1]e_1 + [t/p_2]e_2 + \ldots + [t/p_{i-1}]e_{i-1} + e_S + [(t-e_S)/p_S]e_S$
  - To determine whether the task T_i is schedulable, we simply have to check whether $w_i(t) \le t$ for some $t \le D_i$.
  - If there are N tasks in the system, plus the deferrable server, we can prove the system schedulable if we can prove that $w_N(t) \le t$ for some $t \le D_N$.
  - Remember, this is a sufficient condition, not necessary – i.e. if this condition is not true, the system may not be schedulable

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Schedulability of deadline-driven systems with a deferrable server
  - The deadline of a deferrable server is its next replenishment time
  - A period task T_i in a system of N independent, preemptive, periodic tasks is schedulable with a deferrable server with period p_S, execution budget e_S, and utilization u_S, according to the EDF algorithm if

$$\sum_{k=1}^{N} \frac{e_k}{\min(D_k, p_k)} + u_s\left(1 + \frac{p_s - e_s}{D_i}\right) \le 1$$

Note that if the deferrable server was being treated just like any other periodic task, the second term on the left hand side would just be u_S.

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Sporadic servers
  - Limitation of deferrable servers – they may delay lower-priority tasks for more time than a periodic task with the same period and execution time
  - Sporadic server are designed to eliminate this limitation. Its consumption and replenishment rules ensure that a sporadic server with period p_S and budget e_S never demands more processor time than a periodic task with the same parameters

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Sporadic servers in fixed-priority systems
  - System **T** of N independent, preemptable, periodic tasks
  - One sporadic server, with arbitrary priority π_S; if this is the same priority as some periodic task, the tie is broken in favour of the server
  - $\mathbf{T}_H$ is the subset of periodic tasks with higher priorities than the server
  - $\mathbf{T}_H$ idles when no job in $\mathbf{T}_H$ is ready for execution; $\mathbf{T}_H$ is **busy** otherwise
  - A **server busy interval** is a time interval which begins when an aperiodic job arrives at an empty aperiodic job queue and ends when the queue becomes empty again
  - p_S and e_S have been chosen such that the system **T** plus the sporadic server is schedulable according to the fixed-priority algorithm use by the system

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Consumption and replenishment rules
  - Definitions
    - t_r denotes the latest (actual) replenishment time
    - t_f denotes the first instant after tr at which the server begins to execute
    - t_e denotes the latest effective replenishment time
    - At any time t, BEGIN is the beginning instant of the earliest busy interval among the latest contiguous sequence of busy intervals of T_H that started before t.
    - END is the end of the latest busy interval in this sequence if this interval ends before t and equal to ∞ if the interval ends after t
  - The scheduler sets t_r to the current time each time it replenishes the server's execution budget.
  - When the server first begins to execute after a replenishment, the scheduler determines the latest effective replenishment time t_e based on the history of the system and sets the next replenishment time to t_e+p_S

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Simple Sporadic Server
  - Consumption rules – at any time t after t_r, the server's budget is consumed at the rate of 1 per unit time until the budget is exhausted when either of the following two conditions is true. When they are not true, the server holds its budget
    1. The server is executing
    2. The server has executed since t_r and END < t

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Simple Sporadic Server
  - Replenishment rules
    1. Initially when the system begins execution and each time that the budget is replenished, set the execution budget to e_S and t_r = the current time.
    2. At time t_f, if END = t_f, $t_e = \max(t_r, \text{BEGIN})$. If END < t_f, $t_e = t_f$. The next replenishment time is set to $t_e + p_S$.
    3. The next replenishment occurs at the next replenishment time ($t_e + p_S$), except under the following conditions:
       a. If $t_e + p_S$ is earlier than t_f, the budget is replenished as soon as it is exhausted
       b. If **T** becomes idle before $t_e + p_S$, and becomes busy again at t_b, the budget is replenished at $\min(t_b, t_e + p_S)$

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- What do these rules mean in practice
  - Consumption rule 1 (C1) and Replenishment rule 1 (R1) are identical to those for deferrable server
  - C2 says that the server consumes its budget at any time t if it has executed since t_r, it is suspended at time t, and T_H is idle.
  - R2 says that the next replenishment time is p_S units after t_r (i.e. the effective replenishment time t_e is t_r) only if T_H has been busy through the interval (t_r, t_f); otherwise, t_e is later; it is the latest instant at which an equal or lower-priority task executes (or the system is idle) in (t_r, t_f)

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- More discussion
  - C1 ensures that each server job never executes for more time than its execution budget e_S.
  - C2 applies when the server becomes idle while it still has budget – the budget of an idle simple sporadic server continues to decrease with time as if the server were executing
  - These two rules guarantee that a server job never executes at times when the corresponding job of the periodic task T_S does not.
  - C2 also means that the server holds onto its budget at any time t after t_r in two situations:
    - Some higher-priority job is executing
    - The server has not executed since t_r

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- More discussion (continued)
  - R2 and R3a cause the next replenishment time to be at least p_S time units after the effective release time t_e; $t_e \geq t_r$; the next release time is never earlier than the next replenishment time – therefore, consecutive replenishments occur at least p_S units apart
  - R2 makes the effective replenishment time as soon as possible without making the server behave differently from a periodic task.
    - At t_f, t_e is set to t_r if higher-priority tasks have executed throughout the interval (t_r, t_f); this emulates a job in T_S released at t_r but has to wait for higher-priority tasks to become idle before it can execute
    - If lower-priority tasks executed in this interval, t_e is set to the latest time instant when a lower-priority task executes; this emulates a job in T_S that is released at t_e and waits for higher-priority tasks to become idle and then begins execution.

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- More discussion (continued)
  - R3a applies when the current server job has to wait for more than pS units of time before its execution begins; the budget is replenished as soon as it is exhausted

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Recall Simple Sporadic Server for fixed-priority
  - Release rules: the server is enabled whenever it is backlogged and has budget
  - Consumption rules: budget is consumed when
    1. the server is executing
    2. the server has executed since t_r and (lower-priority jobs are executing OR the system is idle)
  - Replenishment rules
    1. initially, and at each replenishment time, budget := e_s and set t_r := current time
    2. t_f is the time that the server is 1st scheduled after t_r; calculate t_e as:
       a. If t_f coincides with the end of a higher-priority job, t_e := max(t_r, BEGIN)
       b. If lower-priority job or idle system preceded t_f, t_e := t_f

       the next replenishment time is set for $t_e + p_s$
    3. Replenishment occurs at $t_e + p_s$ except …
       a. If $t_e + p_s < t_f$, budget is replenished as soon as exhausted
       b. If system idles before $t_e + p_s$ and becomes busy again at $t_b < t_e + p_s$, the budget is replenished at t_b

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- What do these rules really mean?
  - View each replenishment time as the nominal “release time” of a server job; actual release time is t_e
  - C1 implies each server job executes for no more time than its execution budget
  - C2 implies that the server retains its budget if
    - A higher-priority job is executing, or
    - It has not executed since t_r
    - This implies that if the server idles while it has budget, budget continues to decrease over time
  - R2 makes the effective replenishment time as soon as possible commensurate with the server acting like a periodic task (, p_s, e_s)
  - R3a assumes that $D_s > p_s$, and that this fact was taken into account in determining the schedulability of the system
  - The system is correct without rule R3b; the text discusses why it is still correct with R3b; in essence, it causes replenishment to happen sooner if the system becomes busy after an idle interval ➔ that the system will be able to react more quickly to the arrival of an aperiodic task

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Simple Sporadic Server for dynamic-priority (EDF)
  - Release rules: the server is enabled whenever it is backlogged, it has budget, and its deadline d is defined
  - Consumption rules: budget is consumed when
    1. the server is executing
    2. the deadline d is defined, the server is idle, and there are no jobs with a deadline before d ready for execution
  - Replenishment rules
    1. initially, and at each replenishment time, budget := e_s and set t_r := current time; initially, t_e and d are undefined
    2. whenever t_e is defined, d = $t_e + p_s$, and the next replenishment time is $t_e + p_s$; t_e defined as follows:
       a. At time t when an aperiod job arrives at an empty queue
          i. If only jobs with deadlines earlier than $t_r + p_s$ have executed in (t_r, t), $t_e := t_r$
          ii. If any jobs with deadlines after $t_r + p_s$ have executed in (t_r, t), t_e := t
       b. At replenishment time t_r
          i. If server is backlogged, $t_e := t_r$
          ii. If server is idle, t_e and d become undefined
    3. Replenishment occurs at $t_e + p_s$ except …
       a. If $t_e + p_s$ < t when server first becomes backlogged after t_r, budget is replenished as soon as exhausted
       b. budget is replenished at end of each idle interval of **T**

Scheduling Aperiodic and Sporadic Jobs in Priority-driven Systems

- Consider the following system:
  - T_1 = (, 3, 1)
  - T_2 = (, 4, 0.5)
  - T_s = (, 5, 0.5)
  - T_3 = (, 10, 2)
- Aperiodic tasks A_1, A_2, A_3, each requiring 0.75 units of time, arrive at 0.5, 12.25, 17.
- Show the schedules that result and calculate the response times for A_i for the following servers for both fixed- and dynamic-priority algorithms:
  - Polling
  - Deferrable
  - Simple sporadic

RM + Polling Server

Time	Aperiodic Q	Budget	Run Q	Execute
0.00	Ø	0.5	$J_{1,1}$[1]; $J_{2,1}$[0.5]; J_S; $J_{3,1}$[2]	$J_{1,1}$
0.50	A_1[0.75]	0.5	$J_{1,1}$[0.5]; $J_{2,1}$[0.5]; J_S; $J_{3,1}$[2]	$J_{1,1}$
1.00	A_1[0.75]	0.5	$J_{2,1}$[0.5]; J_S; $J_{3,1}$[2]	$J_{2,1}$
1.50	A_1[0.75]	0.5	J_S; $J_{3,1}$[2]	J_S
2.00	A_1[0.25]	0	$J_{3,1}$[2]	$J_{3,1}$
3.00	A_1[0.25]	0	$J_{1,2}$[1]; $J_{3,1}$[1]	$J_{1,2}$
4.00	A_1[0.25]	0	$J_{2,2}$[0.5]; $J_{3,1}$[1]	$J_{2,2}$
4.50	A_1[0.25]	0	$J_{3,1}$[1]	$J_{3,1}$
5.00	A_1[0.25]	0.5	J_S; $J_{3,1}$[0.5]	J_S
5.25	Ø	0.25	J_S; $J_{3,1}$[0.5]	J_S
5.25	Ø	0	$J_{3,1}$[0.5]	$J_{3,1}$
5.75	Ø	0	Ø	-
6.00	Ø	0	$J_{1,3}$[1]	$J_{1,3}$
7.00	Ø	0	Ø	-
8.00	Ø	0	$J_{2,3}$[0.5]	$J_{2,3}$
8.50	Ø	0	Ø	-
9.00	Ø	0	$J_{1,4}$[1]	$J_{1,4}$
10.00	Ø	0.5	J_S; $J_{3,2}$[2]	J_S
10.00	Ø	0	$J_{3,2}$[2]	$J_{3,2}$
12.00	Ø	0	$J_{1,5}$[1]; $J_{2,4}$[0.5]	$J_{1,5}$
12.25	A_2[0.75]	0	$J_{1,5}$[0.75]; $J_{2,4}$[0.5]	$J_{1,5}$
13.00	A_2[0.75]	0	$J_{2,4}$[0.5]	$J_{2,4}$
13.50	A_2[0.75]	0	Ø	-
15.00	A_2[0.75]	0.5	$J_{1,6}$[1]; J_S	$J_{1,6}$
16.00	A_2[0.75]	0.5	$J_{2,5}$[0.5]; J_S	$J_{2,5}$
16.50	A_2[0.75]	0.5	J_S	J_S
17.00	A_2[0.25]; A_3[0.75]	0	Ø	-
18.00	A_2[0.25]; A_3[0.75]	0	$J_{1,7}$[1]	$J_{1,7}$
19.00	A_2[0.25]; A_3[0.75]	0	Ø	-
20.00	A_2[0.25]; A_3[0.75]	0.5	$J_{2,6}$[0.5]; J_S; $J_{3,3}$[2]	$J_{2,6}$
20.50	A_2[0.25]; A_3[0.75]	0.5	J_S; $J_{3,3}$[2]	J_S
20.75	A_3[0.75]	0.25	J_S; $J_{3,3}$[2]	J_S
21.00	A_3[0.5]	0	$J_{1,8}$[1]; $J_{3,3}$[2]	$J_{1,8}$
22.00	A_3[0.5]	0	$J_{3,3}$[2]	$J_{3,3}$
24.00	A_3[0.5]	0	$J_{1,9}$[1]; $J_{2,7}$[0.5]	$J_{1,9}$
25.00	A_3[0.5]	0.5	$J_{2,7}$[0.5]; J_S	$J_{2,7}$
25.50	A_3[0.5]	0.5	J_S	J_S
26.00	Ø	0	Ø	-

Response time for A_1 = 5.25 – 0.5 = 4.75
Response time for A_2 = 20.75 – 12.25 = 8.50
Response time for A_3 = 26.00 – 17.00 = 9.00

RM + Deferrable Server

Time	Aperiodic Q	Budget	Run Q	Execute
0.00	Ø	0.5	$J_{1,1}$[1]; $J_{2,1}$[0.5]; $J_{3,1}$[2]	$J_{1,1}$
0.50	A_1[0.75]	0.5	$J_{1,1}$[0.5]; $J_{2,1}$[0.5]; J_S; $J_{3,1}$[2]	$J_{1,1}$
1.00	A_1[0.75]	0.5	$J_{2,1}$[0.5]; J_S; $J_{3,1}$[2]	$J_{2,1}$
1.50	A_1[0.75]	0.5	J_S; $J_{3,1}$[2]	J_S
2.00	A_1[0.25]	0	$J_{3,1}$[2]	$J_{3,1}$
3.00	A_1[0.25]	0	$J_{1,2}$[1]; $J_{3,1}$[1]	$J_{1,2}$
4.00	A_1[0.25]	0	$J_{2,2}$[0.5]; $J_{3,1}$[1]	$J_{2,2}$
4.50	A_1[0.25]	0	$J_{3,1}$[1]	$J_{3,1}$
5.00	A_1[0.25]	0.5	J_S; $J_{3,1}$[0.5]	J_S
5.25	Ø	0.25	$J_{3,1}$[0.5]	$J_{3,1}$
5.75	Ø	0.25	Ø	-
6.00	Ø	0.25	$J_{1,3}$[1]	$J_{1,3}$
7.00	Ø	0.25	Ø	-
8.00	Ø	0.25	$J_{2,3}$[0.5]	$J_{2,3}$
8.50	Ø	0.25	Ø	-
9.00	Ø	0.25	$J_{1,4}$[1]	$J_{1,4}$
10.00	Ø	0.5	$J_{3,2}$[2]	$J_{3,2}$
12.00	Ø	0.5	$J_{1,5}$[1]; $J_{2,4}$[0.5]	$J_{1,5}$
12.25	A_2[0.75]	0.5	$J_{1,5}$[0.75]; $J_{2,4}$[0.5]; J_S	$J_{1,5}$
13.00	A_2[0.75]	0.5	$J_{2,4}$[0.5]; J_S	$J_{2,4}$
13.50	A_2[0.75]	0.5	J_S	J_S
14.00	A_2[0.25]	0	Ø	-
15.00	A_2[0.25]	0.5	$J_{1,6}$[1]; J_S	$J_{1,6}$
16.00	A_2[0.25]	0.5	$J_{2,5}$[0.5]; J_S	$J_{2,5}$
16.50	A_2[0.25]	0.5	J_S	J_S
16.75	Ø	0.25	Ø	-
17.00	A_3[0.75]	0.25	J_S	J_S
17.25	A_3[0.50]	0	Ø	-
18.00	A_3[0.50]	0	$J_{1,7}$[1]	$J_{1,7}$
19.00	A_3[0.50]	0	Ø	-
20.00	A_3[0.50]	0.5	$J_{2,6}$[0.5]; J_S; $J_{3,3}$[2]	$J_{2,6}$
20.50	A_3[0.50]	0.5	J_S; $J_{3,3}$[2]	J_S
21.00	Ø	0	$J_{1,8}$[1]; $J_{3,3}$[2]	$J_{1,8}$
22.00	Ø	0	$J_{3,3}$[2]	$J_{3,3}$
24.00	Ø	0	$J_{1,9}$[1]; $J_{2,7}$[0.5]	$J_{1,9}$

Response time for A_1 = 5.25 – 0.5 = 4.75
Response time for A_2 = 16.75 – 12.25 = 4.50
Response time for A_3 = 21 – 17 = 4.00

RM + Sporadic Server

t	t_r	t_e	BEGIN	END	Aperiodic Q	Budget	Run Q	Execute
0.00	0	?	0	∞	Ø	0.5	$J_{1,1}$[1]; $J_{2,1}$[0.5]; $J_{3,1}$[2]	$J_{1,1}$
0.50	0	?	0	∞	A_1[0.75]	0.5	$J_{1,1}$[0.5]; $J_{2,1}$[0.5]; J_S; $J_{3,1}$[2]	$J_{1,1}$
1.00	0	?	0	∞	A_1[0.75]	0.5	$J_{2,1}$[0.5]; J_S; $J_{3,1}$[2]	$J_{2,1}$
1.50	0	0	0	1.5	A_1[0.75]	0.5	J_S; $J_{3,1}$[2]	J_S
2.00	0	0	0	1.5	A_1[0.25]	0	$J_{3,1}$[2]	$J_{3,1}$
3.00	0	0	3.0	∞	A_1[0.25]	0	$J_{1,2}$[1]; $J_{3,1}$[1]	$J_{1,2}$
4.00	0	0	3.0	∞	A_1[0.25]	0	$J_{2,2}$[0.5]; $J_{3,1}$[1]	$J_{2,2}$
4.50	0	0	3.0	4.5	A_1[0.25]	0	$J_{3,1}$[1]	$J_{3,1}$
5.00	5	5	3.0	4.5	A_1[0.25]	0.5	J_S; $J_{3,1}$[0.5]	J_S
5.25	5	5	3.0	4.5	Ø	0.25	$J_{3,1}$[0.5]	$J_{3,1}$
5.5	5	5	3.0	4.5	Ø	0	$J_{3,1}$[0.25]	$J_{3,1}$
5.75	5	5	3.0	4.5	Ø	0	Ø	-
6.00	6	?	6.0	∞	Ø	0.5	$J_{1,3}$[1]	$J_{1,3}$
7.00	6	?	6.0	7.0	Ø	0.5	Ø	-
8.00	8	?	8.0	∞	Ø	0.5	$J_{2,3}$[0.5]	$J_{2,3}$
8.50	8	?	8.0	8.5	Ø	0.5	Ø	-
9.00	9	?	9.0	∞	Ø	0.5	$J_{1,4}$[1]	$J_{1,4}$
10.00	9	?	9.0	10.0	Ø	0.5	$J_{3,2}$[2]	$J_{3,2}$
12.00	9	?	12.0	∞	Ø	0.5	$J_{1,5}$[1]; $J_{2,4}$[0.5]	$J_{1,5}$
12.25	9	?	12.0	∞	A_2[0.75]	0.5	$J_{1,5}$[0.75]; $J_{2,4}$[0.5]; J_S	$J_{1,5}$
13.00	9	?	12.0	∞	A_2[0.75]	0.5	$J_{2,4}$[0.5]; J_S	$J_{2,4}$
13.50	9	12.0	12.0	13.5	A_2[0.75]	0.5	J_S	J_S
14.00	9	12.0	12.0	13.5	A_2[0.25]	0	Ø	-
15.00	15	?	15.0	∞	A_2[0.25]	0.5	$J_{1,6}$[1]; J_S	$J_{1,6}$
16.00	15	?	15.0	∞	A_2[0.25]	0.5	$J_{2,5}$[0.5]; J_S	$J_{2,5}$
16.50	15	15.0	15.0	16.5	A_2[0.25]	0.5	J_S	J_S
16.75	15	15.0	15.0	16.5	Ø	0.25	Ø	-
17.00	15	15.0	15.0	16.5	A_3[0.75]	0	Ø	-
18.00	18	?	18.0	∞	A_3[0.75]	0.5	$J_{1,7}$[1]; J_S	$J_{1,7}$
19.00	18	18.0	18.0	19.0	A_3[0.75]	0.5	J_S	J_S
19.5	18	18.0	18.0	19.0	A_3[0.25]	0	Ø	-
20.00	20	?	20.0	∞	A_3[0.25]	0.5	$J_{2,6}$[0.5]; J_S; $J_{3,3}$[2]	$J_{2,6}$
20.50	20	20	20.0	20.5	A_3[0.25]	0.5	J_S; $J_{3,3}$[2]	J_S
20.75	20	20	20.0	20.5	Ø	0.25	$J_{3,3}$[2]	$J_{3,3}$
21.00	20	20	20.0	20.5	Ø	0	$J_{1,8}$[1]; $J_{3,3}$[1.75]	$J_{1,8}$
22.00					Ø		$J_{3,3}$[1.75]	$J_{3,3}$
23.75					Ø		Ø	-

Response time for A_1 = 5.25 – 0.5 = 4.75
Response time for A_2 = 16.75 – 12.25 = 4.50
Response time for A_3 = 20.75 – 17.0 = 3.75

EDF + Polling Server

Time	Aperiodic Q	Budget	Run Q	Execute
0.00	Ø	0.5	$J_{1,1}$[1,3]; $J_{2,1}$[0.5,4]; J_S[0.5,5]; $J_{3,1}$[2,10]	$J_{1,1}$
0.50	A_1[0.75]	0.5	$J_{1,1}$[0.5,3]; $J_{2,1}$[0.5,4]; J_S[0.5,5]; $J_{3,1}$[2,10]	$J_{1,1}$
1.00	A_1[0.75]	0.5	$J_{2,1}$[0.5,4]; J_S[0.5,5]; $J_{3,1}$[2,10]	$J_{2,1}$
1.50	A_1[0.75]	0.5	J_S[0.5,5]; $J_{3,1}$[2,10]	J_S
2.00	A_1[0.25]	0	$J_{3,1}$[2,10]	$J_{3,1}$
3.00	A_1[0.25]	0	$J_{1,2}$[1,6]; $J_{3,1}$[1,10]	$J_{1,2}$
4.00	A_1[0.25]	0	$J_{2,2}$[0.5,8]; $J_{3,1}$[1,10]	$J_{2,2}$
4.50	A_1[0.25]	0	$J_{3,1}$[1,10]	$J_{3,1}$
5.00	A_1[0.25]	0.5	J_S[0.5,10]; $J_{3,1}$[0.5,10]	J_S
5.25	Ø	0.25	J_S[0.25,10]; $J_{3,1}$[0.5,10]	J_S
5.25	Ø	0	$J_{3,1}$[0.5,10]	$J_{3,1}$
5.75	Ø	0	Ø	-
6.00	Ø	0	$J_{1,3}$[1,9]	$J_{1,3}$
7.00	Ø	0	Ø	-
8.00	Ø	0	$J_{2,3}$[0.5,12]	$J_{2,3}$
8.50	Ø	0	Ø	-
9.00	Ø	0	$J_{1,4}$[1,12]	$J_{1,4}$
10.00	Ø	0.5	J_S[0.5,15]; $J_{3,2}$[2,20]	J_S
10.00	Ø	0	$J_{3,2}$[2,20]	$J_{3,2}$
12.00	Ø	0	$J_{1,5}$[1,15]; $J_{2,4}$[0.5,16]	$J_{1,5}$
12.25	A_2[0.75]	0	$J_{1,5}$[0.75,15]; $J_{2,4}$[0.5,16]	$J_{1,5}$
13.00	A_2[0.75]	0	$J_{2,4}$[0.5,16]	$J_{2,4}$
13.50	A_2[0.75]	0	Ø	-
15.00	A_2[0.75]	0.5	$J_{1,6}$[1,18]; J_S[0.5,20]	$J_{1,6}$
16.00	A_2[0.75]	0.5	J_S[0.5,20]; $J_{2,5}$[0.5,20]	J_S
16.50	A_2[0.25]	0	$J_{2,5}$[0.5,20]	$J_{2,5}$
17.00	A_2[0.25]; A_3[0.75]	0	Ø	-
18.00	A_2[0.25]; A_3[0.75]	0	$J_{1,7}$[1,21]	$J_{1,7}$
19.00	A_2[0.25]; A_3[0.75]	0	Ø	-
20.00	A_2[0.25]; A_3[0.75]	0.5	$J_{2,6}$[0.5,24]; J_S[0.5,25]; $J_{3,3}$[2,30]	$J_{2,6}$
20.50	A_2[0.25]; A_3[0.75]	0.5	J_S[0.5,25]; $J_{3,3}$[2,30]	J_S
20.75	A_3[0.75]	0.25	J_S[0.25,25]; $J_{3,3}$[2,30]	J_S
21.00	A_3[0.5]	0	$J_{1,8}$[1,24]; $J_{3,3}$[2,30]	$J_{1,8}$
22.00	A_3[0.5]	0	$J_{3,3}$[2,30]	$J_{3,3}$
24.00	A_3[0.5]	0	$J_{1,9}$[1,27]; $J_{2,7}$[0.5,28]	$J_{1,9}$
25.00	A_3[0.5]	0.5	$J_{2,7}$[0.5,28]; J_S[0.5,30]	$J_{2,7}$
25.50	A_3[0.5]	0.5	J_S[0.5,30]	J_S
26.00	Ø	0	Ø	-

Response time for A_1 = 5.25 – 0.5 = 4.75
Response time for A_2 = 20.75 – 12.25 = 8.50
Response time for A_3 = 26.00 – 17.00 = 9.00

EDF + Deferrable Server

Time	Aperiodic Q	Budget	Run Q	Execute
0.00	Ø	0.5	$J_{1,1}$[1,3]; $J_{2,1}$[0.5,4]; $J_{3,1}$[2,10]	$J_{1,1}$
0.50	A_1[0.75]	0.5	$J_{1,1}$[0.5,3]; $J_{2,1}$[0.5,4]; J_S[0.5,5]; $J_{3,1}$[2,10]	$J_{1,1}$
1.00	A_1[0.75]	0.5	$J_{2,1}$[0.5,4]; J_S[0.5,5]; $J_{3,1}$[2,10]	$J_{2,1}$
1.50	A_1[0.75]	0.5	J_S[0.5,5]; $J_{3,1}$[2,10]	J_S
2.00	A_1[0.25]	0	$J_{3,1}$[2,10]	$J_{3,1}$
3.00	A_1[0.25]	0	$J_{1,2}$[1,6]; $J_{3,1}$[1,10]	$J_{1,2}$
4.00	A_1[0.25]	0	$J_{2,2}$[0.5,8]; $J_{3,1}$[1,10]	$J_{2,2}$
4.50	A_1[0.25]	0	$J_{3,1}$[1,10]	$J_{3,1}$
5.00	A_1[0.25]	0.5	J_S[0.5,10]; $J_{3,1}$[0.10]	J_S
5.25	Ø	0.25	$J_{3,1}$[0.5,10]	$J_{3,1}$
5.75	Ø	0.25	Ø	-
6.00	Ø	0.25	$J_{1,3}$[1,9]	$J_{1,3}$
7.00	Ø	0.25	Ø	-
8.00	Ø	0.25	$J_{2,3}$[0.5,12]	$J_{2,3}$
8.50	Ø	0.25	Ø	-
9.00	Ø	0.25	$J_{1,4}$[1,12]	$J_{1,4}$
10.00	Ø	0.5	$J_{3,2}$[2,20]	$J_{3,2}$
12.00	Ø	0.5	$J_{1,5}$[1,15]; $J_{2,4}$[0.5,16]	$J_{1,5}$
12.25	A_2[0.75]	0.5	$J_{1,5}$[0.75,15]; J_S[0.5,15]; $J_{2,4}$[0.5,16]	$J_{1,5}$
13.00	A_2[0.75]	0.5	J_S[0.5,15]; $J_{2,4}$[0.5,16]	J_S
13.50	A_2[0.75]	0	$J_{2,4}$[0.5,16]	$J_{2,4}$
14.00	A_2[0.25]	0	Ø	-
15.00	A_2[0.25]	0.5	$J_{1,6}$[1,18]; J_S[0.5,20]	$J_{1,6}$
16.00	A_2[0.25]	0.5	J_S[0.5,20]; $J_{2,5}$[0.5,20]	J_S
16.25	Ø	0.25	$J_{2,5}$[0.5,20]	$J_{2,5}$
16.75	Ø	0.25	Ø	-
17.00	A_3[0.75]	0.25	J_S[0.25,20]	J_S
17.25	A_3[0.50]	0	Ø	-
18.00	A_3[0.50]	0	$J_{1,7}$[1,21]	$J_{1,7}$
19.00	A_3[0.50]	0	Ø	-
20.00	A_3[0.50]	0.5	$J_{2,6}$[0.5,24]; J_S[0.5,25]; $J_{3,3}$[2,30]	$J_{2,6}$
20.50	A_3[0.50]	0.5	J_S[0.5,25]; $J_{3,3}$[2,30]	J_S
21.00	Ø	0	$J_{1,8}$[1,24]; $J_{3,3}$[2,30]	$J_{1,8}$
22.00	Ø	0	$J_{3,3}$[2,30]	$J_{3,3}$
24.00	Ø	0	$J_{1,9}$[1,27]; $J_{2,7}$[0.5,28]	$J_{1,9}$

Response time for $A_1 = 5.25 - 0.5 = 4.75$
Response time for $A_2 = 16.25 - 12.25 = 4.00$
Response time for $A_3 = 21.0 - 17.0 = 4.00$

EDF + Sporadic Server

t	t_r	t_e	d	Aperiodic Q	Budget	Run Q	Execute
0.00	0	?	?	Ø	0.5	$J_{1,1}$[1,3]; $J_{2,1}$[0.5,4]; $J_{3,1}$[2,10]	$J_{1,1}$
0.50	0	0	5	A_1[0.75]	0.5	$J_{1,1}$[0.5,3]; $J_{2,1}$[0.5,4]; J_S[0.5,5]; $J_{3,1}$[2,10]	$J_{1,1}$
1.00	0	0	5	A_1[0.75]	0.5	$J_{2,1}$[0.5,4]; J_S[0.5,5]; $J_{3,1}$[2,10]	$J_{2,1}$
1.50	0	0	5	A_1[0.75]	0.5	J_S[0.5,5]; $J_{3,1}$[2,10]	J_S
2.00	0	0	5	A_1[0.25]	0	$J_{3,1}$[2,10]	$J_{3,1}$
3.00	0	0	5	A_1[0.25]	0	$J_{1,2}$[1,6]; $J_{3,1}$[1,10]	$J_{1,2}$
4.00	0	0	5	A_1[0.25]	0	$J_{2,2}$[0.5,8]; $J_{3,1}$[1,10]	$J_{2,2}$
4.50	0	0	5	A_1[0.25]	0	$J_{3,1}$[1,10]	$J_{3,1}$
5.00	5	5	10	A_1[0.25]	0.5	J_S[0.5,10]; $J_{3,1}$[0.5,10]	J_S
5.25	5	5	10	Ø	0.25	$J_{3,1}$[0.5,10]	$J_{3,1}$
5.5	5	5	10	Ø	0	$J_{3,1}$[0.25,10]	$J_{3,1}$
5.75	5	5	10	Ø	0	Ø	-
6.00	6	?	?	Ø	0.5	$J_{1,3}$[1,9]	$J_{1,3}$
7.00	6	?	?	Ø	0.5	Ø	-
8.00	8	?	?	Ø	0.5	$J_{2,3}$[0.5,12]	$J_{2,3}$
8.50	8	?	?	Ø	0.5	Ø	-
9.00	8	?	?	Ø	0.5	$J_{1,4}$[1,12]	$J_{1,4}$
10.00	8	?	?	Ø	0.5	$J_{3,2}$[2,20]	$J_{3,2}$
12.00	8	?	?	Ø	0.5	$J_{1,5}$[1,15]; $J_{2,4}$[0.5,16]	$J_{1,5}$
12.25	8	12.25	17.25	A_2[0.75]	0.5	$J_{1,5}$[0.75,15]; $J_{2,4}$[0.5,16]; J_S[0.5,17.25]	$J_{1,5}$
13.00	8	12.25	17.25	A_2[0.75]	0.5	$J_{2,4}$[0.5,16]; J_S[0.5,17.25]	$J_{2,4}$
13.50	8	12.25	17.25	A_2[0.75]	0.5	J_S[0.5,17.25]	J_S
14.00	8	12.25	17.25	A_2[0.25]	0	Ø	-
15.00	15	15	20	A_2[0.25]	0.5	$J_{1,6}$[1,18]; J_S[0.5,20]	$J_{1,6}$
16.00	15	15	20	A_2[0.25]	0.5	J_S[0.5,20]; $J_{2,5}$[0.5,20]	J_S
16.25	15	15	20	Ø	0.25	$J_{2,5}$[0.5,20]	$J_{2,5}$
16.50	15	15	20	Ø	0	$J_{2,5}$[0.25,20]	$J_{2,5}$
16.75	15	15.0	20	Ø	0	Ø	-
17.00	17	17.0	22	A_3[0.75]	0.5	J_S[0.5,22]	J_S
18.00	18	17.0	22	A_3[0.25]	0	$J_{1,7}$[1,21]	$J_{1,7}$
19.00	18	17.0	22	A_3[0.25]	0	Ø	-
20.00	20	20.0	25	A_3[0.25]	0.5	$J_{2,6}$[0.5,24]; J_S[0.5,25]; $J_{3,3}$[2,30]	$J_{2,6}$
20.50	20	20.0	25	A_3[0.25]	0.5	J_S[0.5,24]; $J_{3,3}$[2,30]	J_S
20.75	20	20.0	25	Ø	0.25	$J_{3,3}$[2,30]	$J_{3,3}$
21.00	20	20.0	25	Ø	0	$J_{1,8}$[1,24]; $J_{3,3}$[1.75,30]	$J_{1,8}$
22.00	20	20.0	25	Ø		$J_{3,3}$[1.75,30]	$J_{3,3}$
23.75	20	20.0	25	Ø		Ø	-

Response time for A_1 = 5.25 – 0.5 = 4.75
Response time for A_2 = 16.25 – 12.25 = 4.00
Response time for A_3 = 20.75 – 17.0 = 3.75