A "Simple" Fixed-Priority Sporadic Server

- Consider a system T of N independent preemptable periodic tasks, plus a single sporadic server task with parameters (p_s, e_s)
  - Tasks are scheduled using a fixed-priority algorithm; system schedulable if we assume (p_s, e_s) behaves as a standard periodic task

- Definitions:
  - T_H is the subset of periodic tasks with higher priorities than the server
    - That subset may be *idle* when no job in T_H is ready for execution, or *busy*
  - Define t_r as the last time the server budget replenished
  - Define t_f as the first instant after t_r at which the server begins to execute
  - At any time t define:
    - *BEGIN* as the start of the earliest busy interval in the most recent contiguous sequence of busy intervals of T_H starting before t
      - Busy intervals are contiguous if the later one starts immediately the earlier one ends
    - *END* as the end of the latest busy interval in this sequence if this interval ends before t; define $END = \infty$ if the interval ends after t

A "Simple" Fixed-Priority Sporadic Server

- Consumption rule:
  - At any time t after t_r, if the server has budget and if either of the following two conditions is true, the server's budget is consumed at the rate of 1 per unit time:

    C1: The server is executing

    C2: The server has executed since t_r and $END < t$
  - When they are not true, the server holds its budget

- That is:
  - The server executes for no more time than it has execution budget
  - The server retains its budget if:
    - A higher-priority job is executing, or
    - It has not executed since t_r
  - Otherwise, the budget decreases when the server executes, or if it idles while it has budget

A "Simple" Fixed-Priority Sporadic Server

- Replenishment rules

  R1: When system begins executing, and each time budget is replenished, set the budget to e_S and t_r = the current time.

  R2: When server begins to execute (defined as time t_f)
  if $END = t_f$ then
  $t_e = \max(t_r, BEGIN)$
  else if $END < t_f$ then
  $t_e = t_f$
  The next replenishment time is set to $t_e + p_S$.

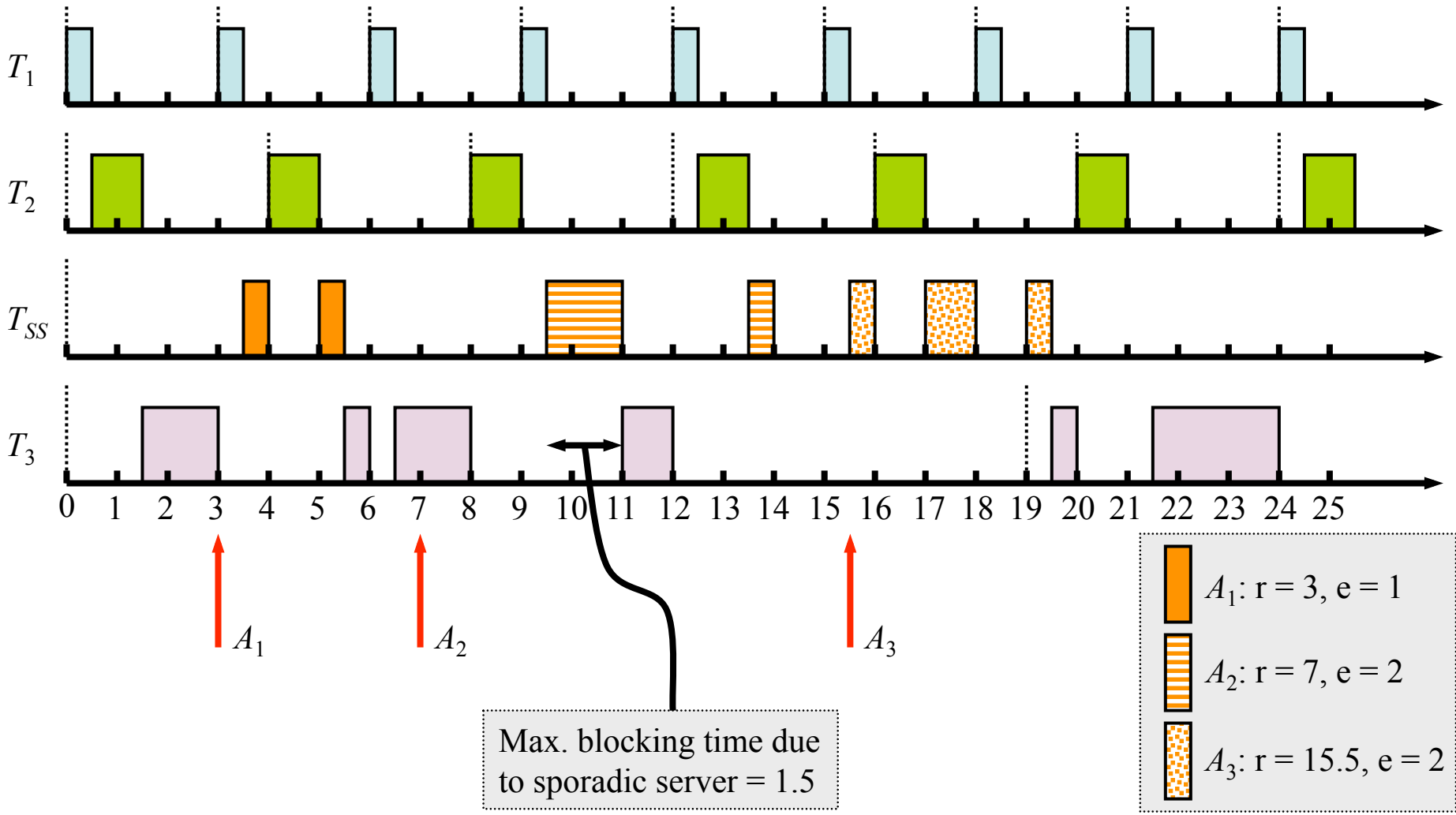
  t_e = the *effective replenishment time*

  R3: 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

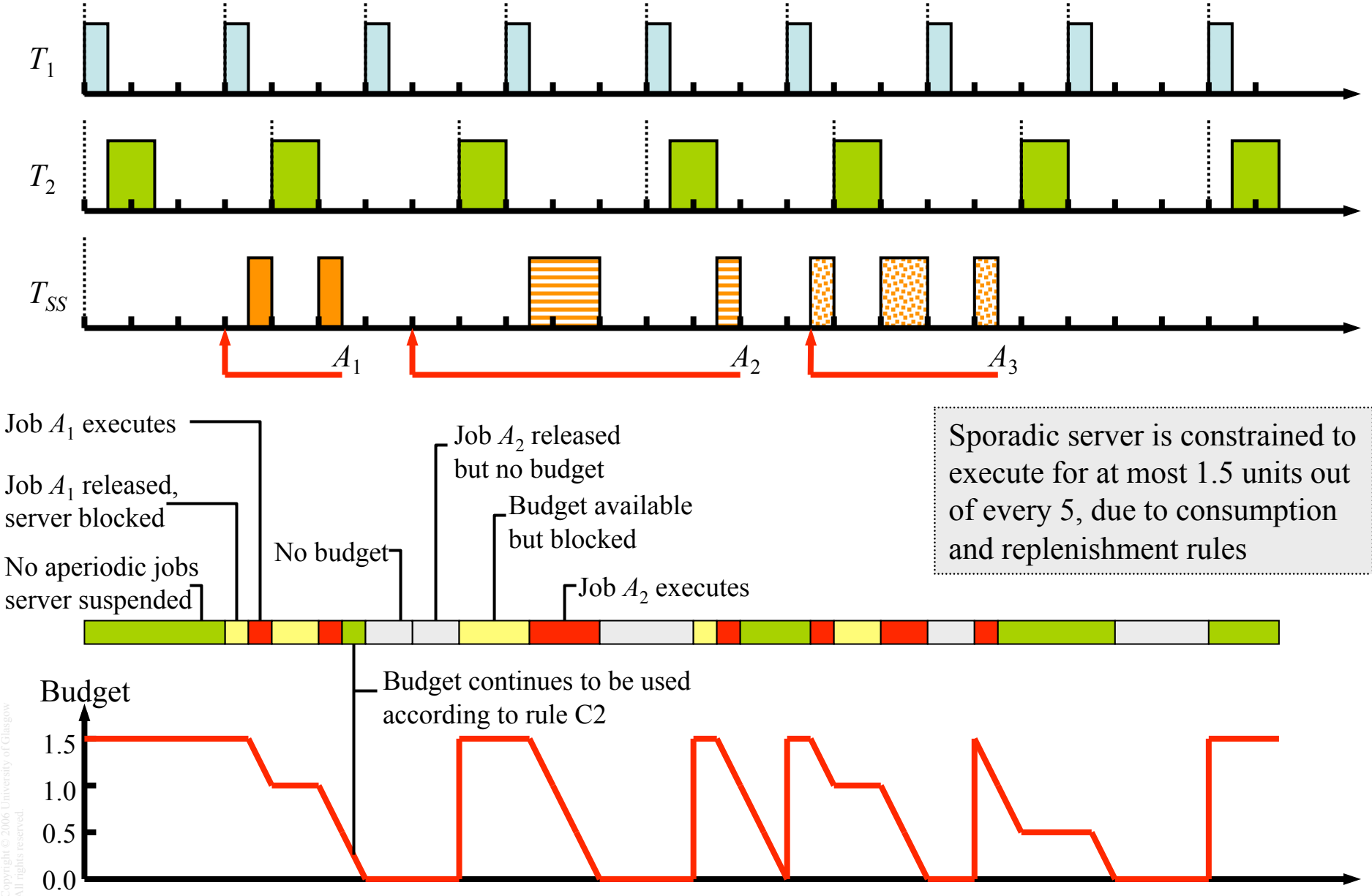
  (b) If T becomes idle before $t_e + p_S$, and becomes busy again at t_b, the budget is replenished at $\min(t_b, t_e + p_S)$

Example: Fixed-Priority Sporadic Server



T_1=(3, 0.5), T_2=(4, 1.0), T_3=(19, 4.5), T_{ss}=(5, 1.5)
Rate monotonic schedule; simple sporadic server

Example: Fixed-Priority Sporadic Server



Sporadic server is constrained to execute for at most 1.5 units out of every 5, due to consumption and replenishment rules