

Aperiodic task scheduling

Scheduling aperiodic tasks with fixed priorities

Aperiodic tasks in conjunction with periodic tasks



Aperiodic tasks

- How do we deal with aperiodic tasks?
- Why?
 - Critical, but occasional, operations that require immediate attention
 - Occasional events that need to be completed soon, but periodic tasks are more important and need to meet their deadlines; aperiodic tasks do not have hard deadlines
 - Examples: system mode changes, activity logs, garbage collection

Mixing periodic and aperiodic tasks

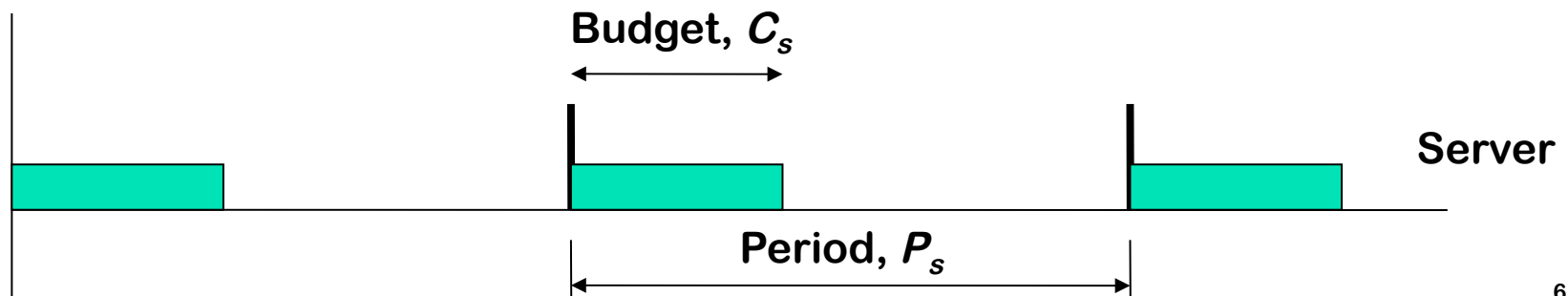
- Question: how to execute aperiodic tasks without violating schedulability guarantees given to periodic tasks?
- And in a static-priority environment

Mixing periodic and aperiodic tasks

- Question: how to execute aperiodic tasks without violating schedulability guarantees given to periodic tasks?
- And in a static-priority environment
- Easy approach: schedule aperiodic tasks at the lowest priority level
 - Problem: Extremely poor performance for aperiodic tasks; periodic tasks can be delayed as long as they do not miss their deadlines

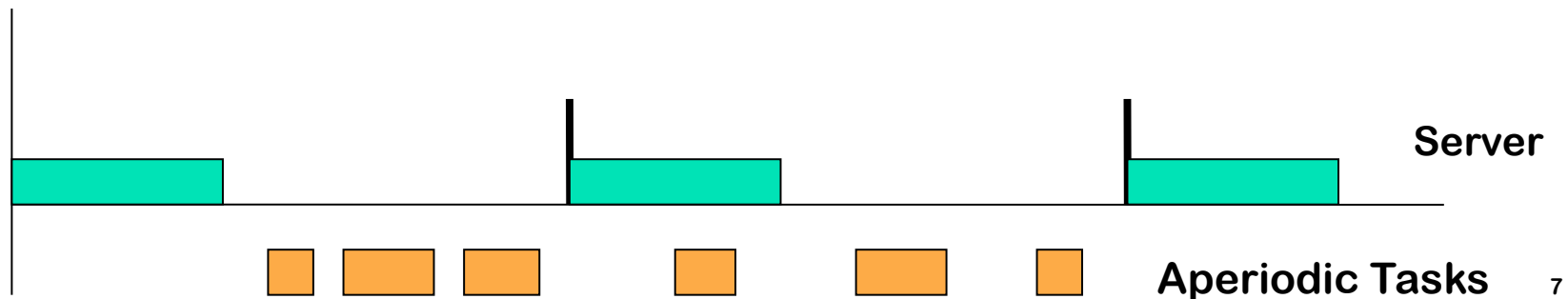
Server-based systems

- Periodically invoke a service task (“server”) to execute aperiodic tasks
- The server is modeled as a periodic task and can be included in schedulability analysis
- Allocate the server a computation budget C_s and a period P_s
- The server can serve aperiodic tasks until the budget expires; the budget can be replenished every period
- Many choices: Servers have different flavours depending on the details of when they are invoked, what priority they have, and when budgets are replenished



Server-based systems

- Periodically invoke a service task (“server”) to execute aperiodic tasks
- The server is modeled as a periodic task and can be included in schedulability analysis
- Allocate the server a computation budget C_s and a period P_s
- The server can serve aperiodic tasks until the budget expires; the budget can be replenished every period
- Many choices: Servers have different flavours depending on the details of when they are invoked, what priority they have, and when budgets are replenished



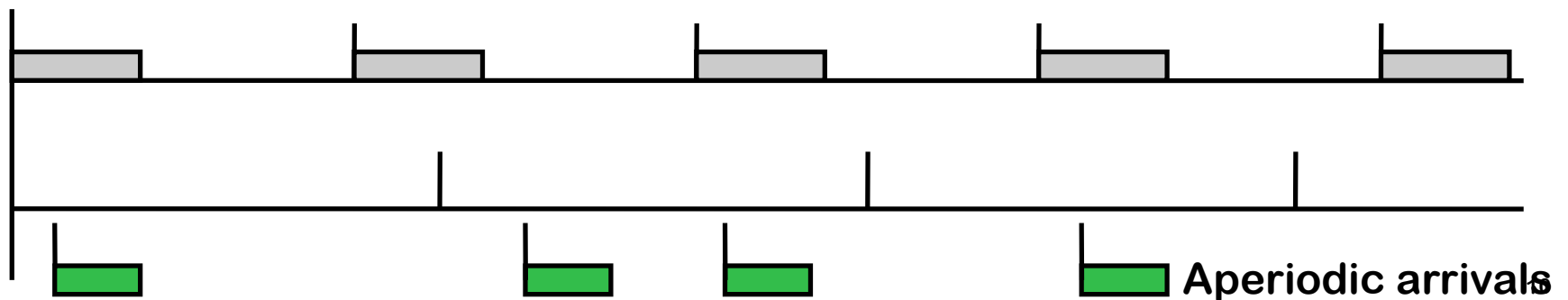


Polling Server

- Runs as a periodic task (priority set according to RM)
- Aperiodic arrivals are queued until the server task is invoked
- When the server is invoked it serves the queue until it is empty or until the budget expires then suspends itself
 - If the queue is empty when the server is invoked it suspends itself immediately
- Server is treated as a regular periodic task in schedulability analysis

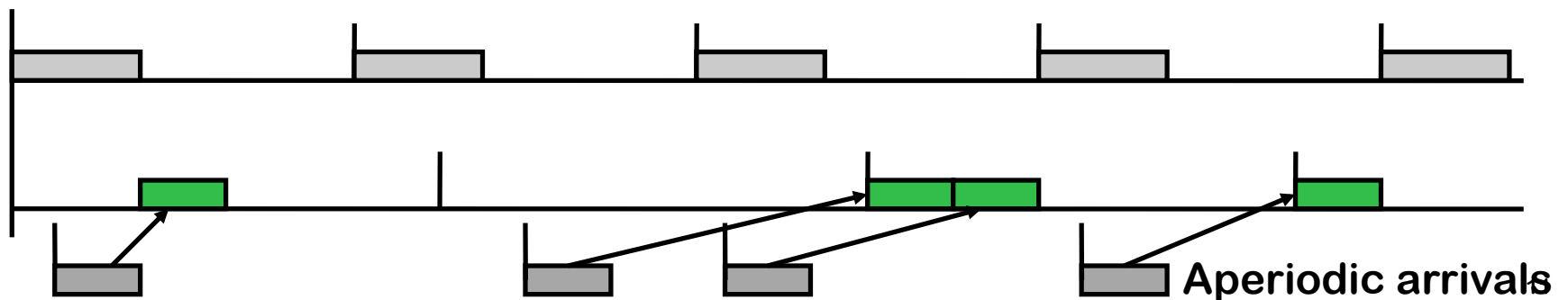
Example of a Polling Server

- Polling server:
 - Period $P_s = 5$
 - Budget $C_s = 2$
- Periodic task
 - $P = 4$
 - $C = 1.5$
- All aperiodic arrivals have $C=1$



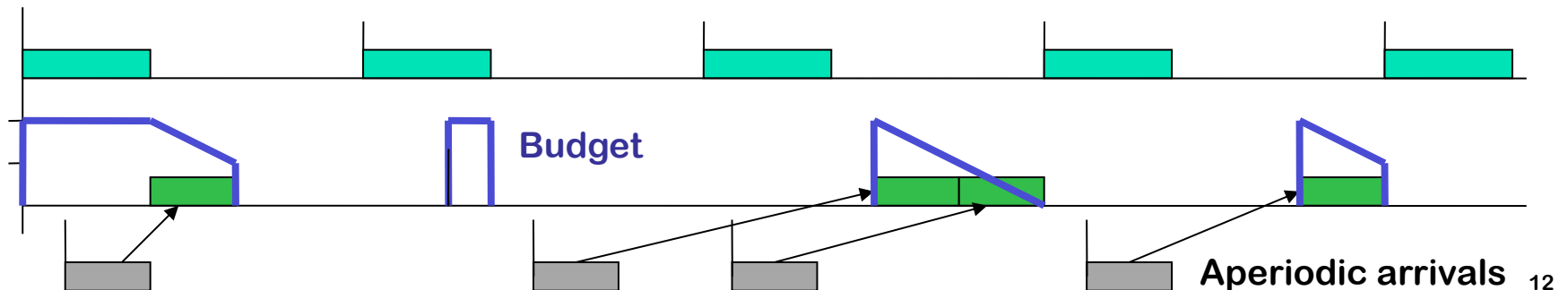
Example of a Polling Server

- Polling server:
 - Period $P_s = 5$
 - Budget $C_s = 2$
- Periodic task
 - $P = 4$
 - $C = 1.5$
- All aperiodic arrivals have $C=1$



Example of a Polling Server

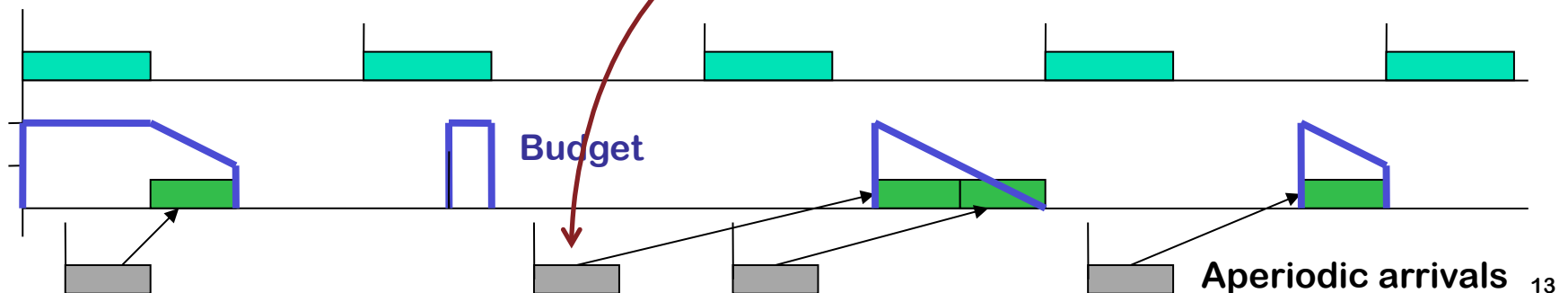
- Polling server:
 - Period $P_s = 5$
 - Budget $C_s = 2$
- Periodic task
 - $P = 4$
 - $C = 1.5$
- All aperiodic arrivals have $C=1$



Example of a Polling Server

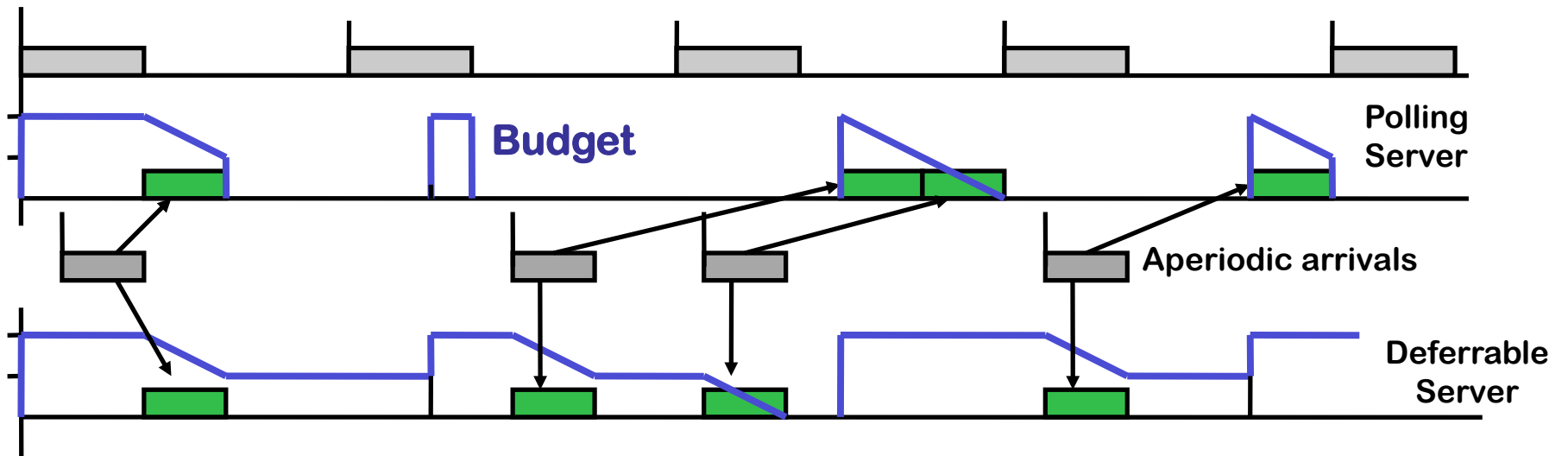
- Polling server:
 - Period $P_s = 5$
 - Budget $C_s = 2$
- Periodic task
 - $P = 4$
 - $C = 1.5$
- All aperiodic arrivals have $C=1$

Why not execute immediately?



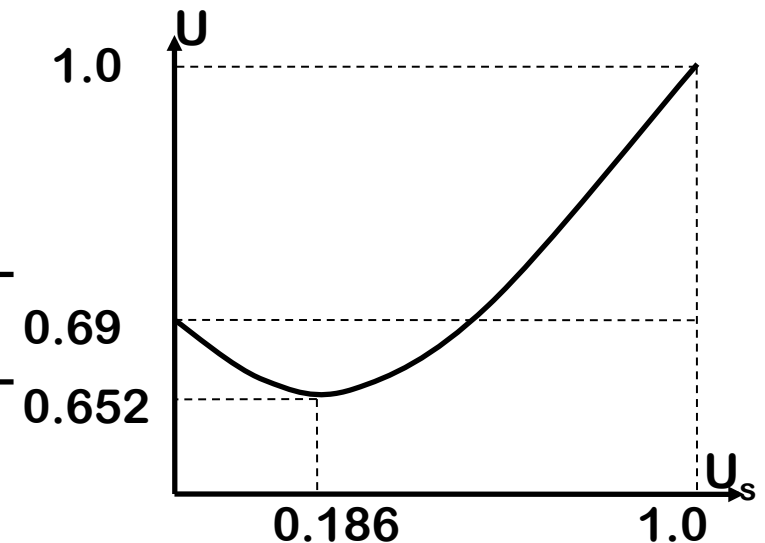
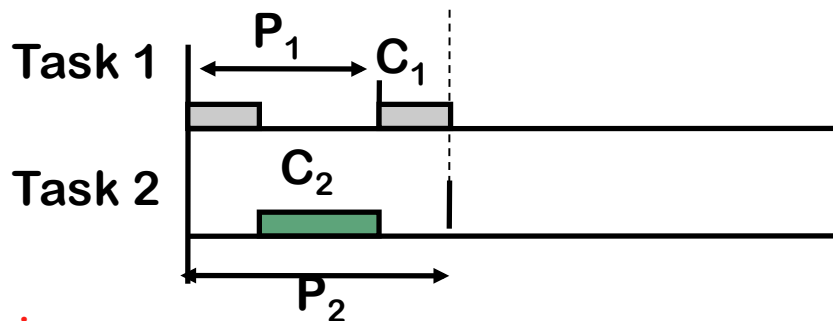
Deferrable Server

- Keeps the balance of the budget until the end of the period
- Example (continued)

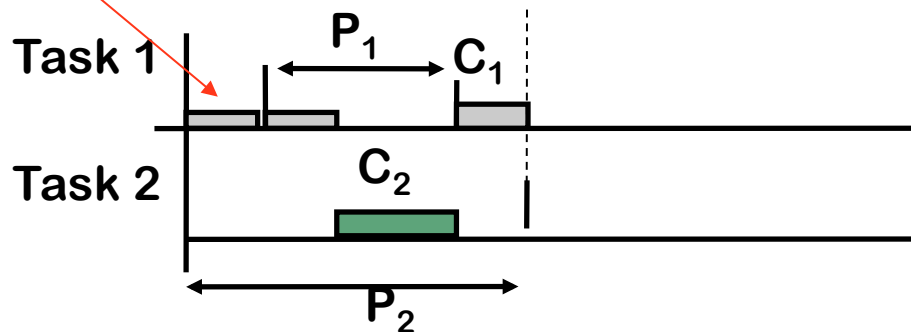


Utilization bound with a Deferrable Server

Worst-case scenario



Deferred previous invocation



$$U_p \leq \ln \left(\frac{U_s + 1}{2U_s + 1} \right)$$

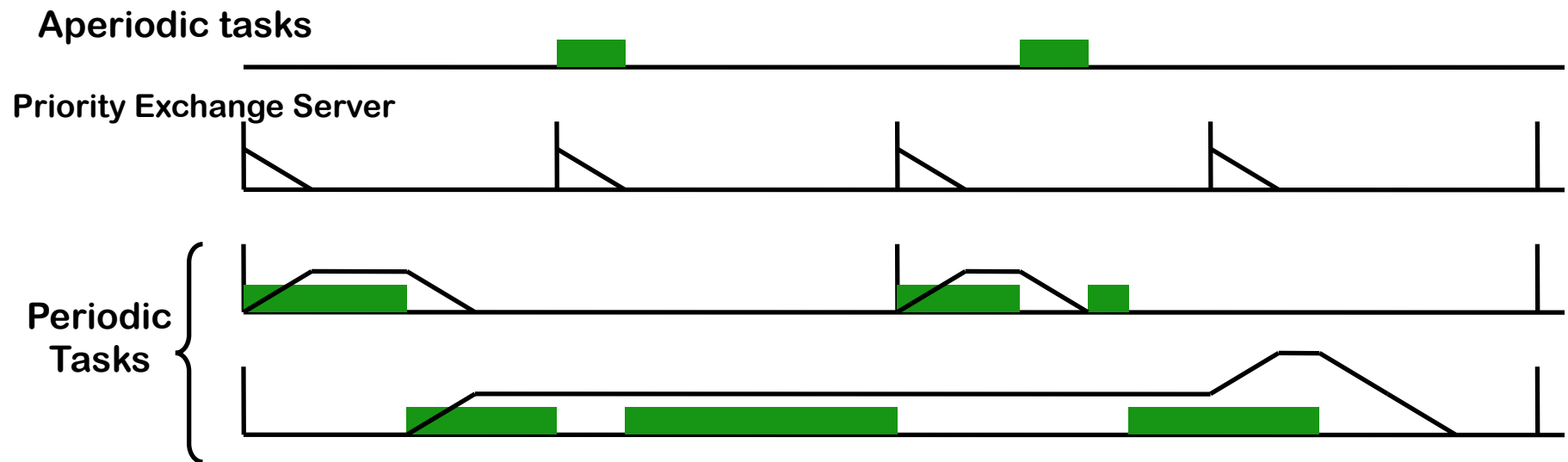
Exercise: Derive the utilization bound for a deferrable server plus one periodic task

Priority Exchange Server

- Like the deferrable server, this server retains its budget until the end of the server period
- Unlike the deferrable server, this server's priority slips over time: when not used, the priority is exchanged for that of the executing periodic task

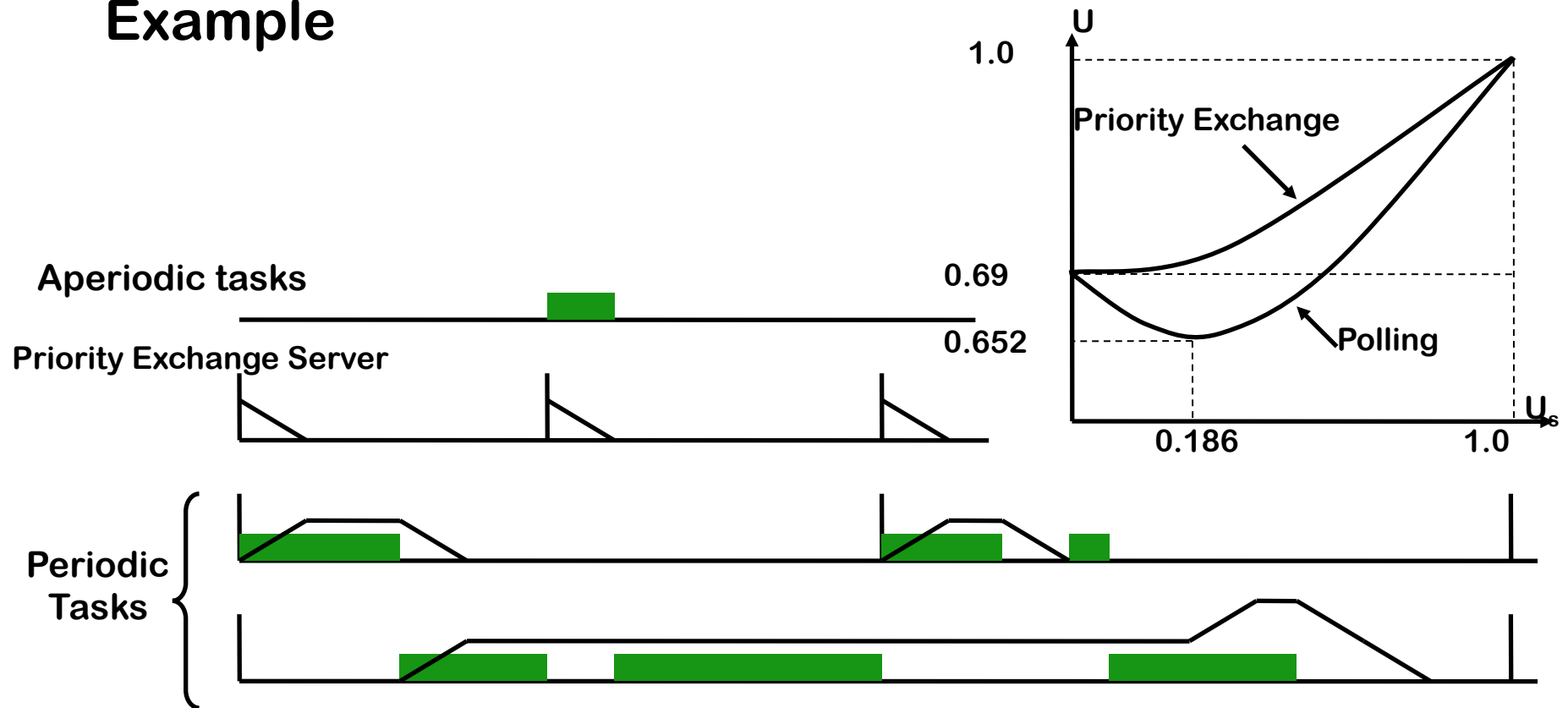
Priority Exchange Server

Example



Priority Exchange Server

Example



$$U_p \leq \ln \left(\frac{2}{U_s + 1} \right)$$

Sporadic Server

- Server is said to be active if it is in the running or ready queue, otherwise it is idle.
- When an aperiodic task comes and the budget is not zero, the server becomes active
- Every time the server becomes active, say at t_A , it sets replenishment time one period into the future, $t_A + P_s$ (but does not decide on replenishment amount).
- When the server becomes idle, say at t_I , set replenishment amount to capacity consumed in $[t_A, t_I]$

Example for Sporadic Server

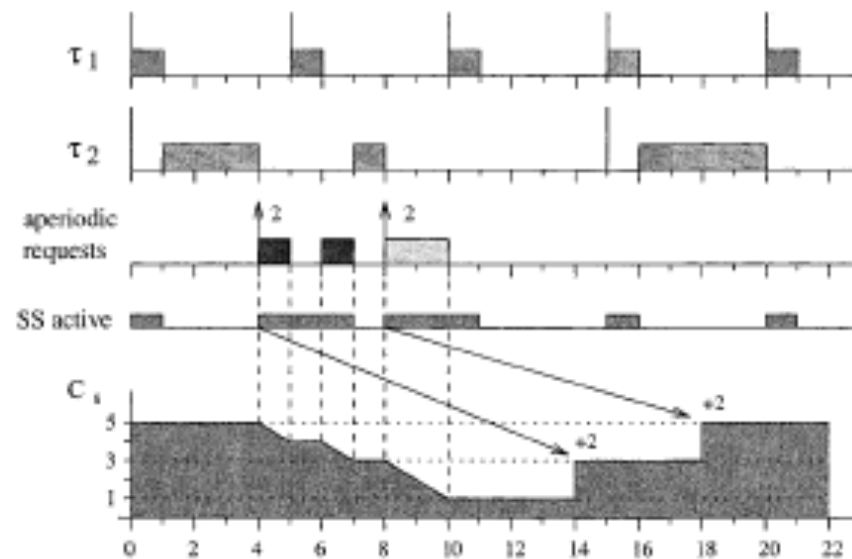
Two periodic tasks

$T_1: (P_1=5, C_1=1)$

$T_2: (P_2=15, C_2=5)$

Sporadic Server

$P_s=10, C_s=5$



Lecture summary

- Aperiodic task scheduling
 - Why schedule aperiodic tasks?
 - Servers
 - Polling Server
 - Deferrable Server
 - Priority Exchange Server
 - Sporadic Server
 - Note: you do not need to memorize the utilization bounds for the servers.