UNIVERSITY OF CAMBRIDGE
Department of Engineering

IfM MANAGEMENT TECHNOLOGY POLICY

Discrete Event Simulation Quick Guide

Dr. Yin-Chi Chan
Research Associate, Institute for Manufacturing
University of Cambridge

Intro to discrete-event simulation

- Generators create entities
- Entities map to processes
  - Conditional flow based on entity attributes & system state
- Processes consume resources
  - Seize/delay/release paradigm
  - If resource is depleted: wait or block depending on model
  - Can define arbitrary delay times
- Processes create events
  - Associated with a state change or other task to be performed in the future
  - A generator is a special process that creates ‘New Entity’ events (= arrivals) in a loop
- Simulation loop
  - Pop next event from queue
  - Jump to the associated process
  - Process: state changes/statistics logging/new event creation
  - Delay, wait for resource, or process end: yield control back to event scheduler

Intro to discrete-event simulation (2)

IfM | MANAGEMENT TECHNOLOGY POLICY

- Events are processed as quickly as possible, rather than in real-time
  - System clock will automatically jump to the time of the next pending event
- Simulation models are usually random
  - Multiple simulation runs are used to generate confidence intervals on system statistics
- Example software tools:

simmer

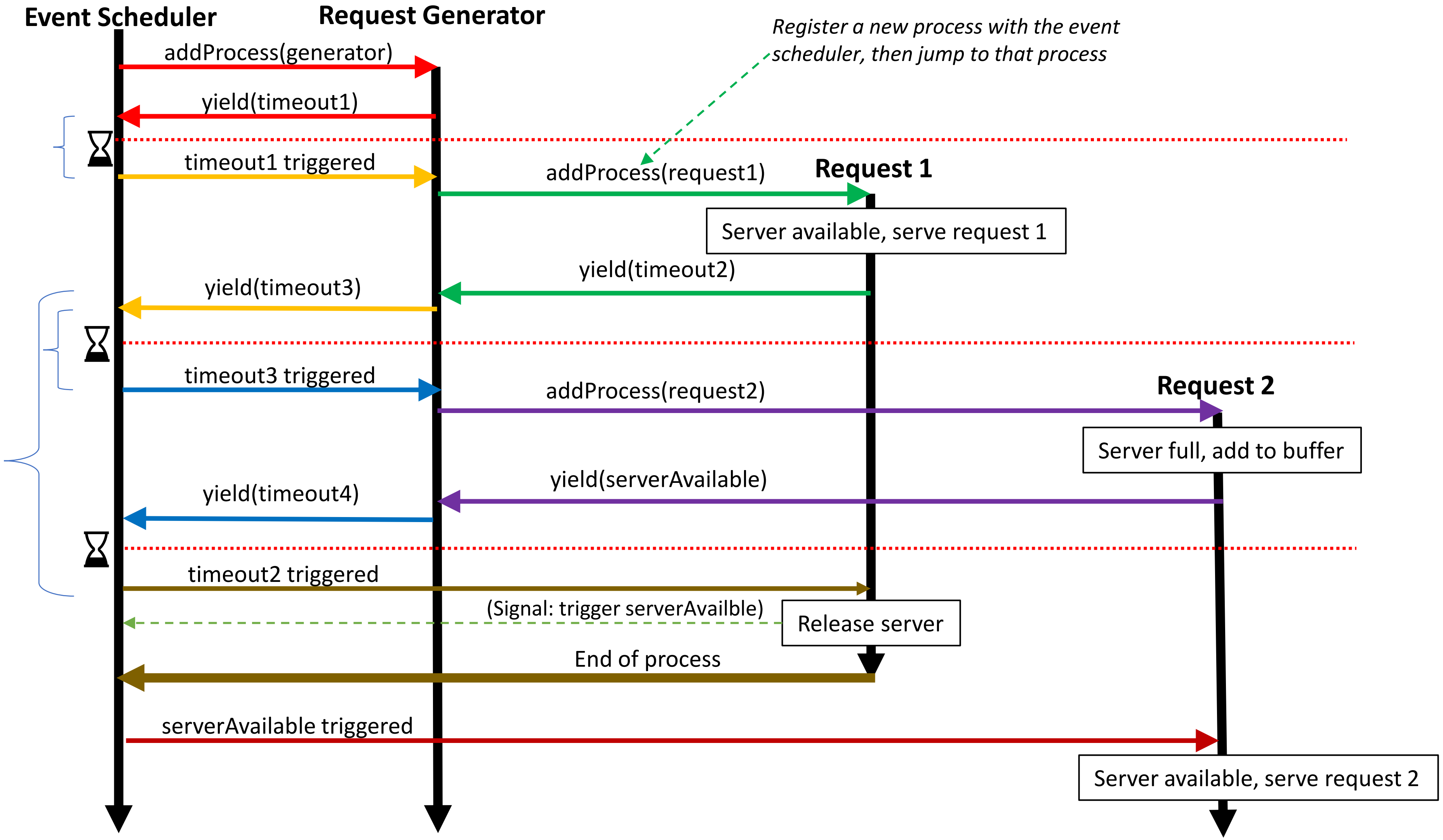
Arena®

SimPy

anylogic®

Simio
Forward Thinking

Single-server example

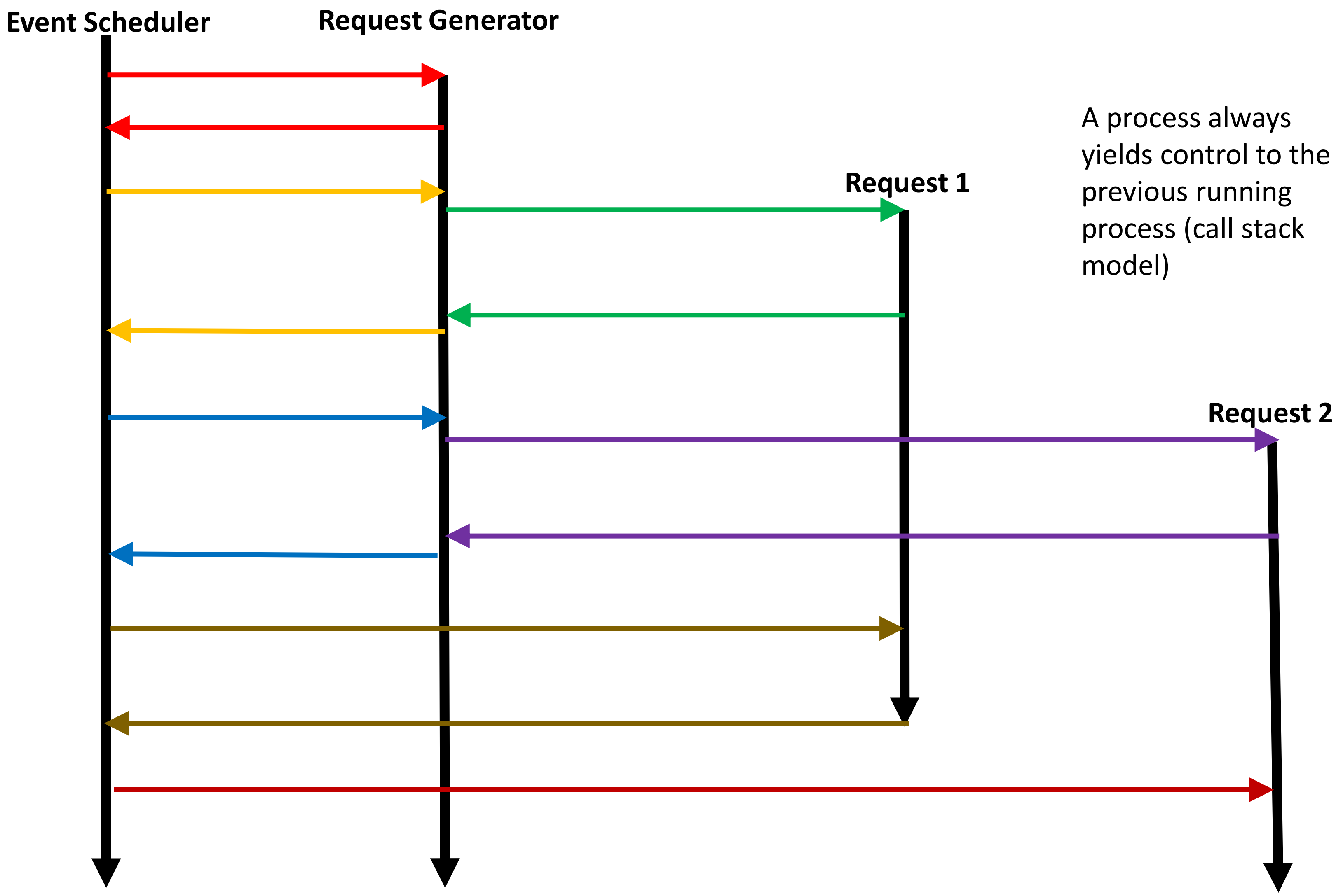


In this example, Request 2 arrives (`timeout3`) before Request 1 leaves (`timeout2`) and therefore needs to wait for the server to become available.

When the server becomes free, the event scheduler finds the first waiting process for the serverAvailable event and yields control to that process.

The symbol ⌛ represents the passage of time in the simulation model.

Single-server example (2)



A process always yields control to the previous running process (call stack model)