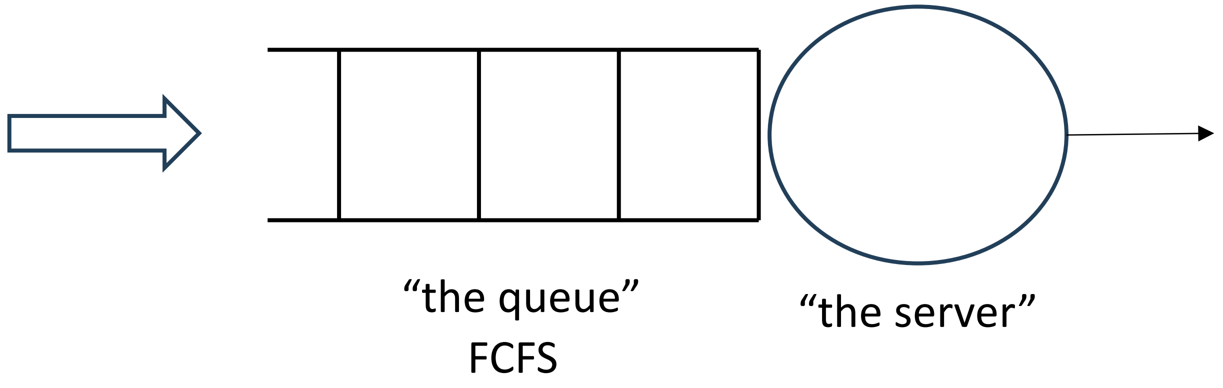
Chapter 14
Event-Driven Simulation

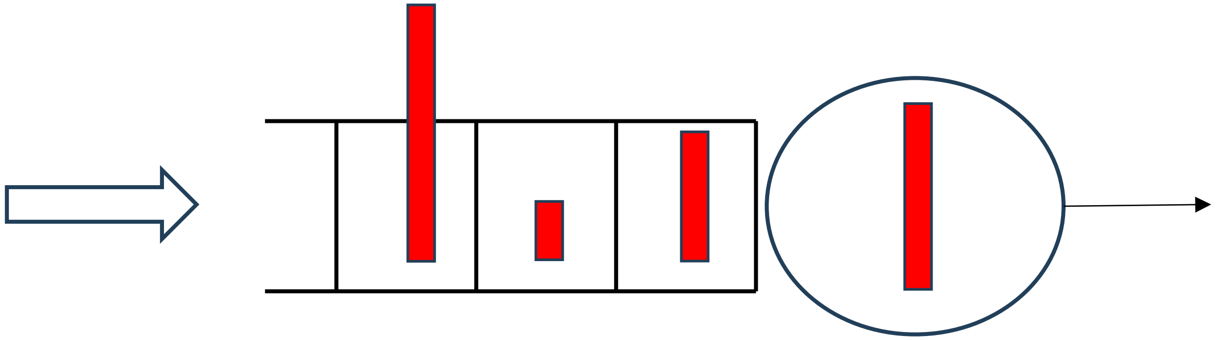
Queueing Theory Terminology: Simplest Model



- The server is the CPU
- A **job** is a red rectangle
- The **size** of a job is the height of the rectangle.
  - $Size = S =$ # seconds of CPU needed by the job
- Only one job is served (run) at a time.
- Jobs are served in FCFS order.

- Jobs arrive over time. The **interarrival time** is the time between subsequent arrivals.
- The **average arrival rate** (λ) is the average number of arrivals per sec:
  - $A_t =$ number of arrivals by time t
  - $\lambda = \lim_{t\to\infty} \frac{A_t}{t}$

Stochastic Setting vs. Trace-driven Simulation



Stochastic Setting

- S : r. v. for size of job.
  - Typically assume i.i.d. instances of S.
- I : r. v. for interarrival time.
  - Typically assume i.i.d. instances of I.

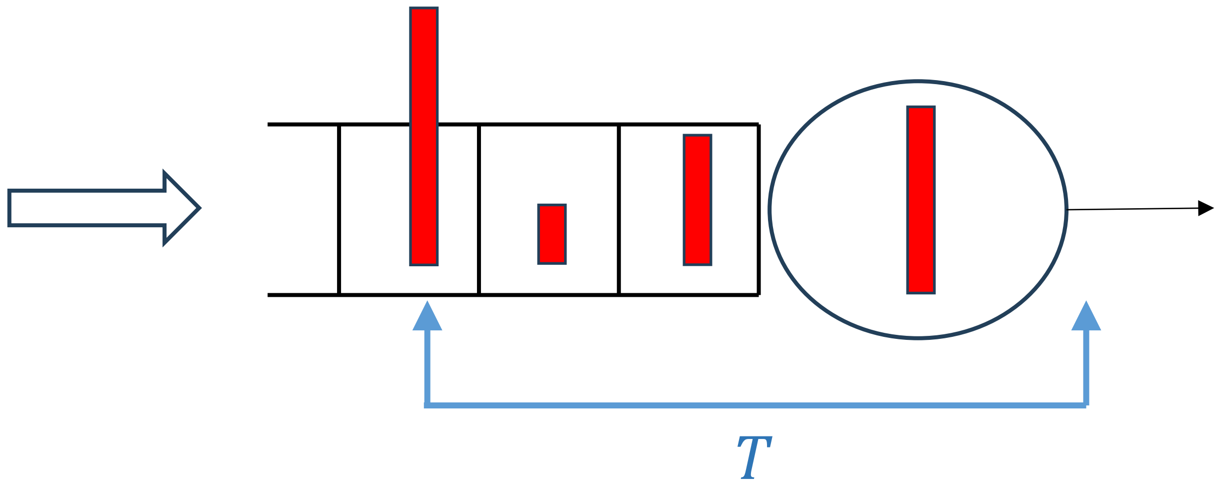
Given a Poisson Process w/ rate λ, how are λ and $\boldsymbol{E}[I]$ related?

$$\lambda = \frac{1}{\boldsymbol{E}[I]}$$

Trace-driven Simulation

- S and I instances are given by a trace.
  - At time 1.5, job arrives of size 7.
  - At time 1.7, job arrives of size 3.
  - At time 13, job arrives of size 1.2.

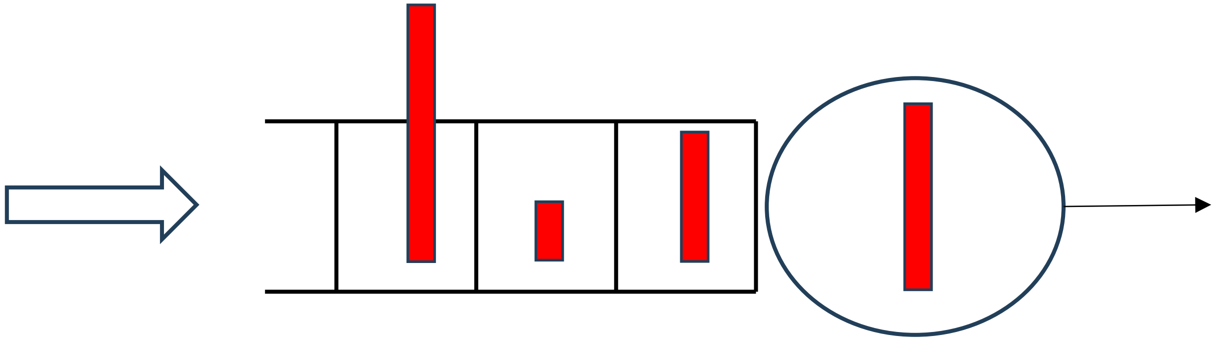
Queueing Metrics



- ❑ Response time of job, T
- ❑ Mean Response time, $\boldsymbol{E}[T]$
- ❑ Number of jobs in system, N
- ❑ Mean number of jobs, $\boldsymbol{E}[N]$

$$\boldsymbol{E}[T] = \lim_{n \to \infty} \frac{\mathrm{T}_1 + \mathrm{T}_2 + \cdots + T_n}{n}$$

Queueing Metrics



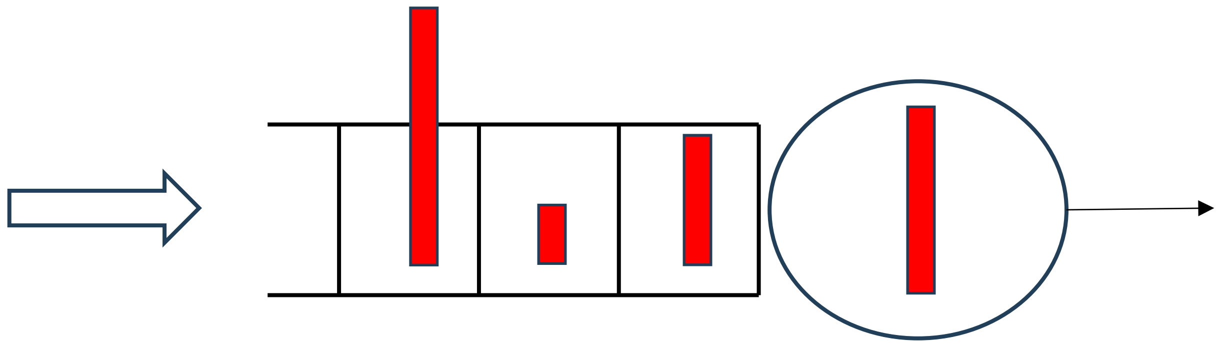
- Server utilization (a.k.a., load), ρ
  - ρ is the long-run fraction of time that the server is busy

$B(t) =$ total time server is busy by time t

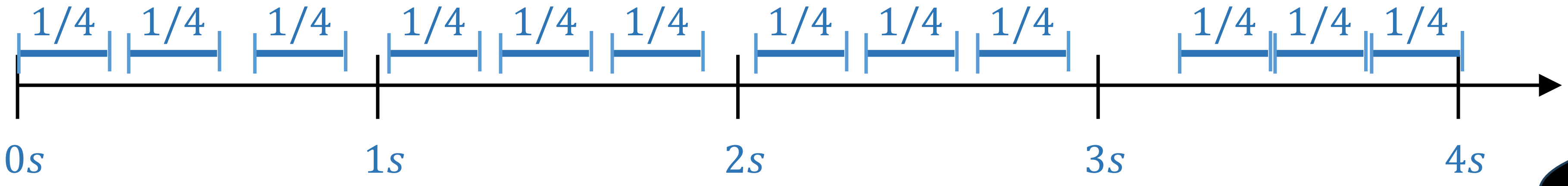
Express ρ in terms of $B(t)$

$$\rho = \lim_{t \to \infty} \frac{B(t)}{t}$$

Queueing Metrics



Q: Suppose $\lambda = 3$ jobs/sec and $\boldsymbol{E}[S] = \frac{1}{4}$ sec. What is ρ? Will there be queueing?

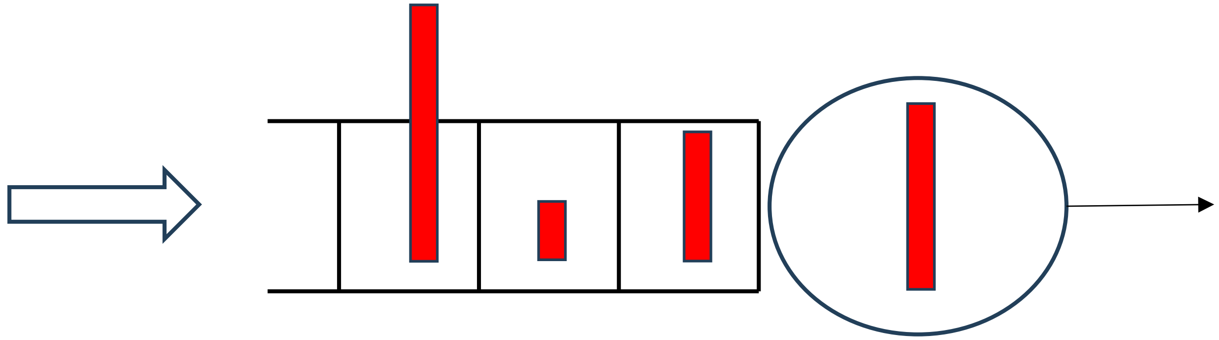


Not necessarily 3 arrivals/s

A: Seems like $\rho = \lambda E[S] = \frac{3}{4}$. We will prove this in Chapter 27.

- ❑ If I, S are Deterministic ➔ no queueing
- ❑ If I, S have high variability ➔ lots of queueing

Running a Simulation – Single Queue



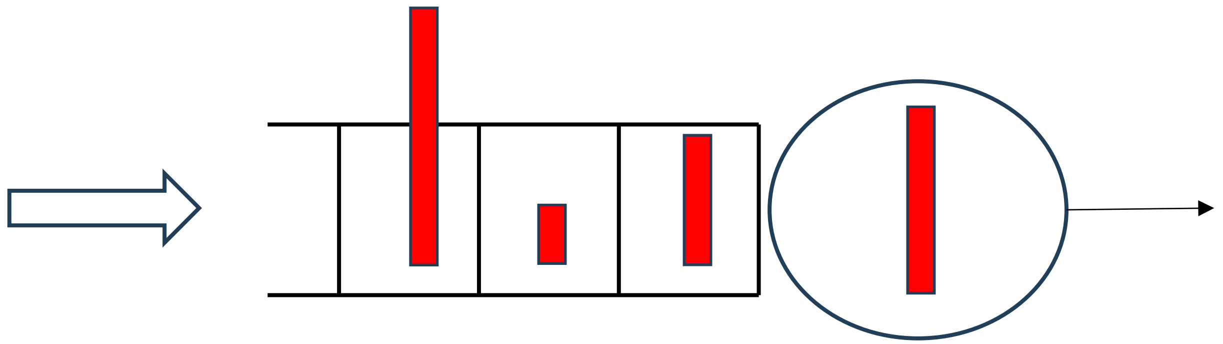
GOAL: Simulate this queue, where interarrival times $\sim I$ and service times $\sim S$

Determine $E[T]$ across 10^6 jobs

Do you start by generating 10^6 instances of I and S ?

No! Generate as needed

Running a Simulation – Single Queue



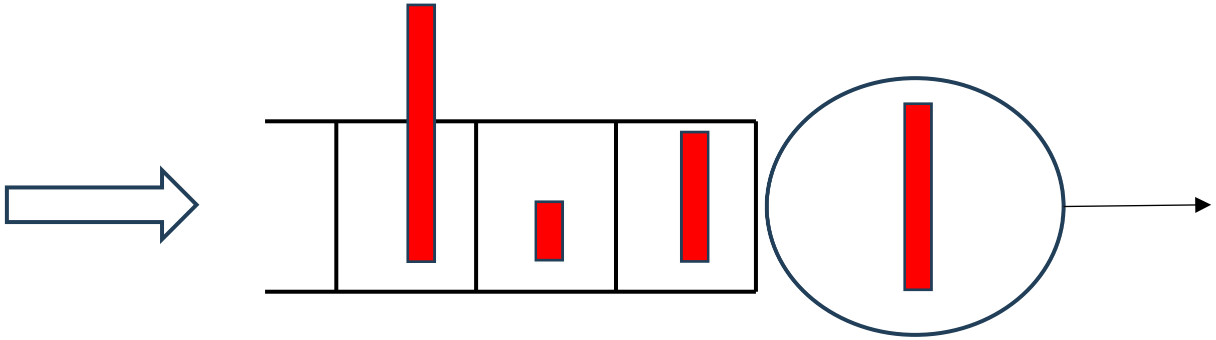
GOAL: Simulate this queue, where interarrival times $\sim I$ and service times $\sim S$

Determine $E[T]$ across 10^6 jobs

If a job takes $5s$, do we wait $5s$ on computer clock?

No! Simulate the clock!

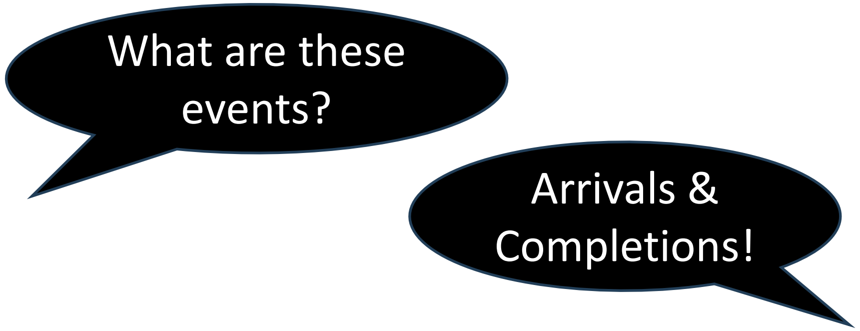
Event-driven Simulation



State = Number of jobs, N, currently in system

Track only events that **change** the state!

Generate instances of I and S as needed.



Event-driven Simulation Example

Suppose instances of I are: 5.3, 2, 9.5, …

Suppose instances of S are: 10, 1, 7, …

Iteration 1

CLOCK $= 0$

State $= N = 0$

Time-to-next-Compl $= \infty$

Time-to-next-Arrival $= Generate(I) = 5.3$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–A} = 5.3$

Event = Arrival @5.3

Iteration 2

CLOCK $= 5.3$

State $= N = 1$

Time-to-next-Compl $= Generate(S) = 10$

Time-to-next-Arrival $= Generate(I) = 2$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–A} = 2$

Event = Arrival @7.3

Event-driven Simulation Example

Suppose instances of I are: 5.3, 2, 9.5, …

Suppose instances of S are: 10, 1, 7, …

Iteration 2

CLOCK $= 5.3$

State $= N = 1$

Time-to-next-Compl $= Generate(S) = 10$

Time-to-next-Arrival $= Generate(I) = 2$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–A} = 2$

Event $=$ Arrival @7.3

Iteration 3

CLOCK $= 7.3$

State $= N = 2$

Time-to-next-Compl $= 10 - 2 = 8$

Time-to-next-Arrival $= Generate(I) = 9.5$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–C} = 8$

Event $=$ Completion @15.3

Event-driven Simulation Example

Suppose instances of I are: 5.3, 2, 9.5, …

Suppose instances of S are: 10, 1, 7, …

Iteration 3

CLOCK $= 7.3$

State $= N = 2$

Time-to-next-Compl $= 10 - 2 = 8$

Time-to-next-Arrival $= Generate(I) = 9.5$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–C} = 8$

Event $=$ Completion @15.3

Iteration 4

CLOCK $= 15.3$

State $= N = 1$

Time-to-next-Compl $= Generate(S) = 1$

Time-to-next-Arrival $= 9.5 - 8 = 1.5$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–C} = 1$

Event $=$ Completion @16.3

Event-driven Simulation Example

Suppose instances of I are: 5.3, 2, 9.5, …

Suppose instances of S are: 10, 1, 7, …

Iteration 4

CLOCK $= 15.3$

State $= N = 1$

Time-to-next-Compl $= Generate(S) = 1$

Time-to-next-Arrival $= 9.5 - 8 = 1.5$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–C} = 1$

Event = Completion @16.3

Iteration 5

CLOCK $= 16.3$

State $= N = 0$

Time-to-next-Compl $= \infty$

Time-to-next-Arrival $= 1.5 - 1 = 0.5$

Next-Event $= \min(\text{T–C}, \text{T–A}) = \text{T–A} = 0.5$

Event = Arrival @16.8

Event-driven Simulation Quiz

Q: In an event-driven simulation, what are the 4 variables you track?

1. Global Clock
2. State = Number jobs in system
3. Time-to-next-Arrival
4. Time-to-next-Completion

Q: When exactly do you generate a new instance of I?

1. Immediately after a job arrives
2. When drop to 0 jobs

Q: When exactly do you generate a new instance of S?

1. Immediately after a job completes, assuming job leaves behind ≥ 1 job.
2. When system moves from state 0 to state 1.

Getting $\boldsymbol{E}[T]$

$$\boldsymbol{E}[T] = \lim_{n \to \infty} \frac{\mathrm{T}_1 + \mathrm{T}_2 + \cdots + T_n}{n}$$

Q: How do we get T_i for our FCFS queue?

A: Log arrival times as they happen on this list:

$$\cancel{5.3} \to \cancel{7.3} \to 16.8$$

When completions happen:

- Subtract earliest arrival on list from current clock time.
- Delete earliest arrival from list

Example: Completion at 15.3 ➔ $T_1 = 15.3 - 5.3 = 10$

Completion at 16.3 ➔ $T_2 = 16.3 - 7.3 = 9$

Getting $\boldsymbol{E}[T]$

$$\boldsymbol{E}[T] = \lim_{n\to\infty} \frac{\mathrm{T}_1 + \mathrm{T}_2 + \cdots + T_n}{n}$$

Q: To get $\boldsymbol{E}[T]$ do I need to store all 10^6 $T_i s$?

A: No!

Let

$$A^{(n)} = \text{ average of first } n \; T_i s \;\; = \frac{1}{n}\sum_{i=1}^{n} T_i$$

$$A^{(n+1)} = \frac{1}{n+1}\left(n \cdot A^{(n)} + T_{n+1}\right)$$

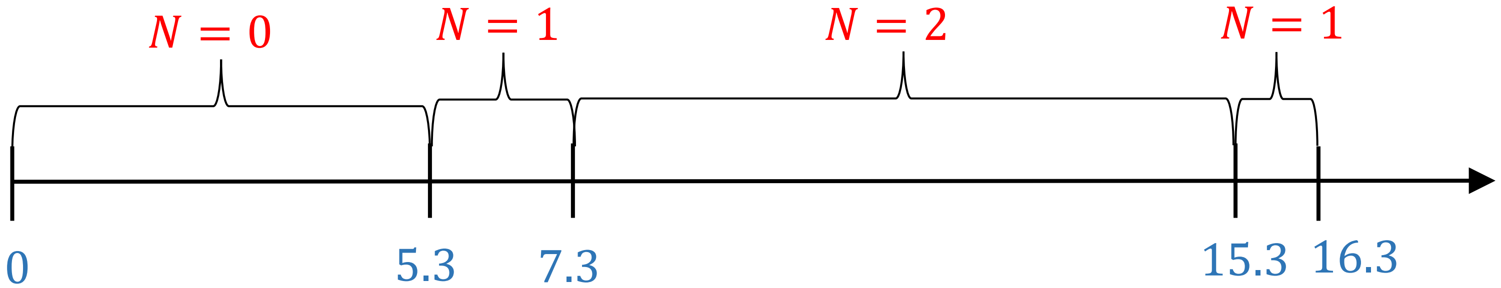
Getting $\boldsymbol{E}[N]$

Let $N(s)$ = Number of jobs in the system at time s

$$\boldsymbol{E}[N] = \lim_{t\to\infty} \frac{\int_0^t N(s)ds}{t}$$

Q: How to get $\boldsymbol{E}[N]$?

Idea 1: (Time Average)



Weight N by length of interval?

$$\boldsymbol{E}[N] = \frac{5.3(0) + 2(1) + 8(2) + 1(1)}{16.3}$$

Getting $\boldsymbol{E}[N]$

Let $N(s)$ = Number of jobs in the system at time s

$$\boldsymbol{E}[N] = \lim_{t\to\infty} \frac{\int_0^t N(s)ds}{t}$$

Q: How to get $\boldsymbol{E}[N]$?

Idea 2: (Ensemble Average)

- ❑ Whenever arrival happens, record how many jobs arrival sees in the system
- ❑ Take average over all these observations

Getting $\boldsymbol{E}[N]$

Let $N(s)$ = Number of jobs in the system at time s

$$\boldsymbol{E}[N] = \lim_{t\to\infty} \frac{\int_0^t N(s)ds}{t}$$

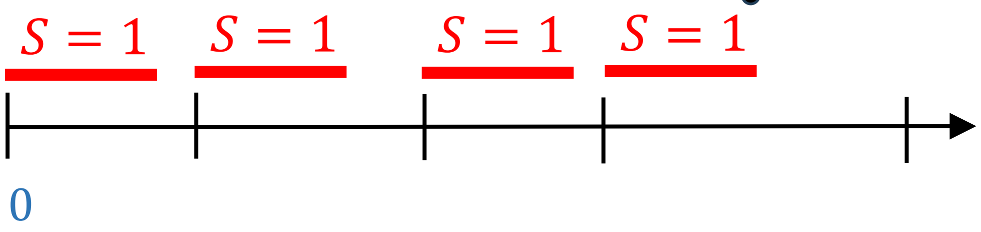
Q: Is $\boldsymbol{E}[N]^{TimeAvg} = \boldsymbol{E}[N]^{EnsembleAvg}$?

Suppose $I \sim Uniform(1,2)$ and $S = 1$. Are your answers the same?

Every arrival walks into empty system

$$\boldsymbol{E}[N]^{TimeAvg} = \frac{2}{3}$$

Correct $\boldsymbol{E}[N]$



$$\boldsymbol{E}[N]^{EnsembleAvg} = 0$$

Getting $\boldsymbol{E}[N]$

Why was $\boldsymbol{E}[N]^{TimeAvg} \neq \boldsymbol{E}[N]^{EnsembleAvg}$?

Arrival times were bad times to measure # jobs

Is it ever true that $\boldsymbol{E}[N]^{TimeAvg} = \boldsymbol{E}[N]^{EnsembleAvg}$?

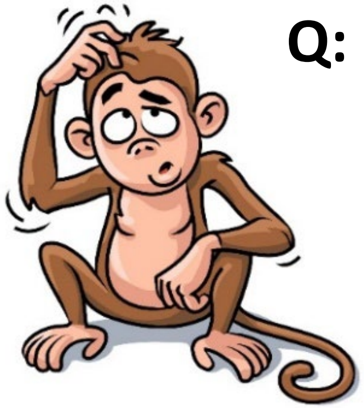
Need arrival times to be “random.”

Poisson Process!

PASTA

PASTA = **P**oisson **A**rrivals **S**ee **T**ime **A**verages

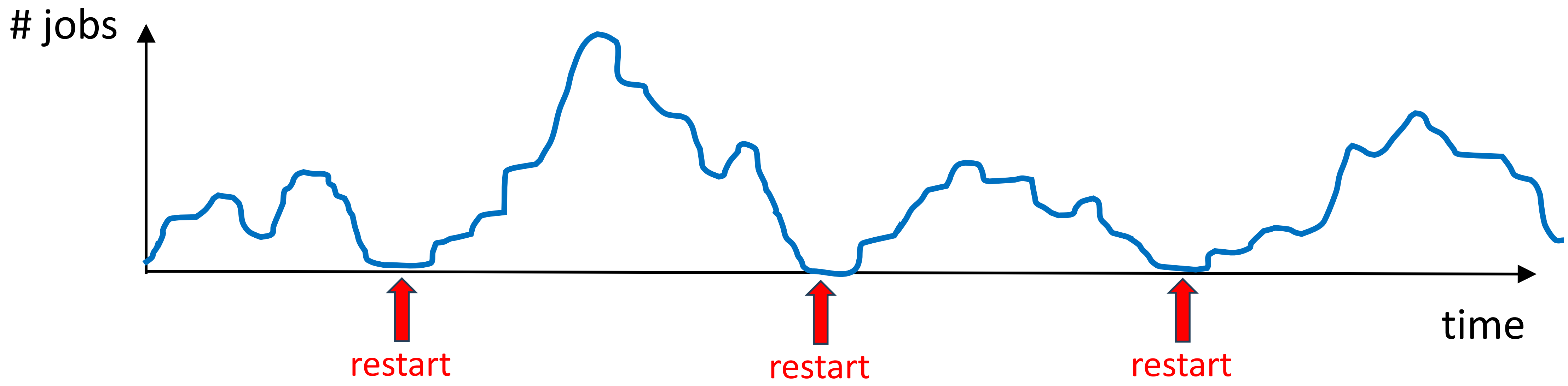
$$\boldsymbol{E}[N]^{TimeAvg} = \boldsymbol{E}[N]^{EnsembleAvg}$$



Q: But what if arrival process is not Poisson. Can we still average over what arrivals see?

A: No, but you can simulate a Poisson Process in the background, and record number of jobs at times of those events!

Running simulations: one long run?

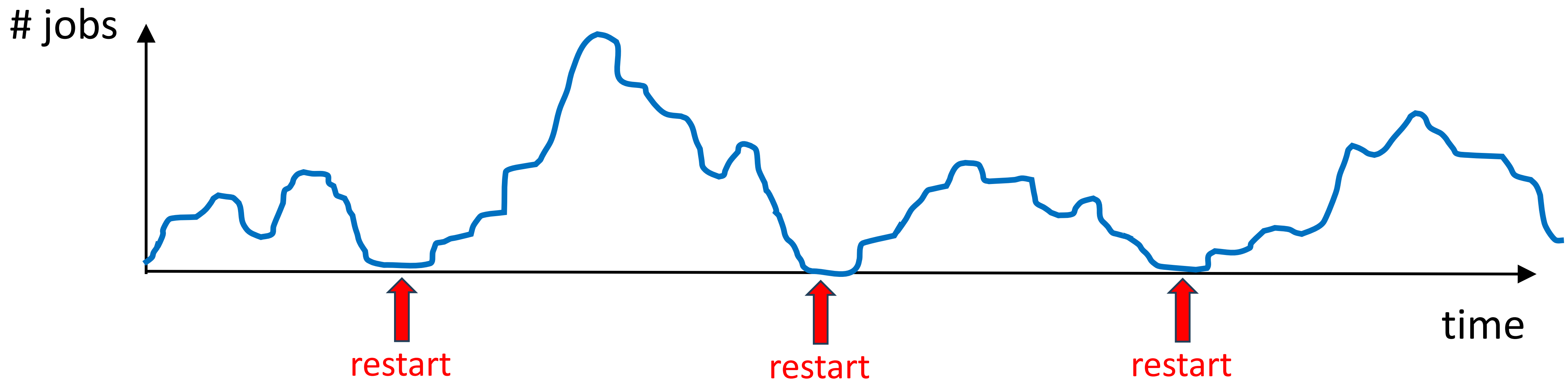


Q: When running simulations, is it better to consider time-average over one long run, or many short runs?

A: Turns out these are the same, provided simulation empties (restarts) infinitely often.

See Chpt 25 of your book!

Running simulations: convergence



Q: How long should we run our simulation? How many arrivals?

A: Run long enough to meet both these conditions:

1. Performance metric is no longer biased by initial state
2. Performance metric is no longer changing much (has converged)