1.225J (ESD 205) Transportation Flow Systems

Lecture 9

Simulation Models

Prof. Ismail Chabini and Prof. Amedeo R. Odoni

Lecture 9 Outline

❑ About this lecture:
   - It is based on R16. Only material covered in class must be read
   - Download a spreadsheet model from the 1.225F01 website

❑ Introduction to simulation models and a simple example

❑ Random numbers

❑ Simulation models for M/M/1 queueing systems

❑ Random observations

❑ An event-driven simulation model for an *M/M/1* queueing system

❑ A spreadsheet implementation of an event-driven simulation model for *M/M/1*

❑ A discrete-time simulation model for an *M/M/1* queueing system

❑ Lecture summary

Introduction to Simulation Models

❑ Types of models covered in this subject:
- Analytical deterministic models (Lectures 1-7)
- Analytical probabilistic models (Lecture 8)
- Simulation models for stochastic systems (This lecture)

❑ Stochastic systems:
- Systems that evolve probabilistically over time
- Example: a queueing system

❑ We are interested in determining estimates of quantities such as true mean and variance for a given random variable of the **physical stochastic system**

❑ A computer simulation model is a computer representation that mimics the behavior of the physical system

❑ Simulation models can be used to obtain virtual statistical samples to estimate the performance of stochastic systems
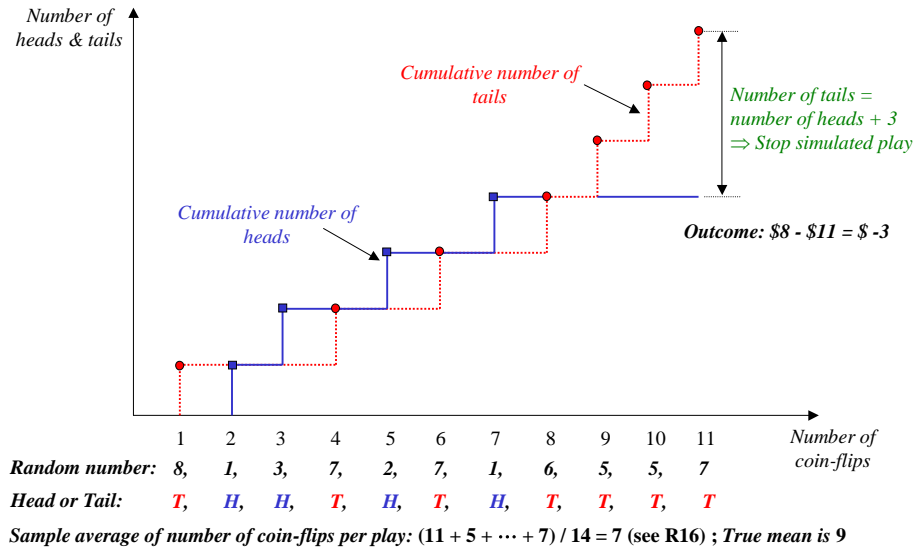
Examples Studied in this Lecture

❑ Coin-Flipping Game:
- Repeatedly flip a coin until the difference between the number of heads tossed and the number of tails tossed is ± 3
- You pay $1 per flip of the coin
- You receive $8 at the end of each play of the game
- You cannot quit when you start a play
  ⇒ *How would you decide whether you play the game or not ?*

❑ Queueing Models:
- *How to determine the steady-state variables of a queueing system with general interarrival and service time distributions ?*

❑ Simulation can answer questions such as the above two questions

Coin-Flipping Game: A Simulated Play

Number of heads & tails

Cumulative number of tails

Number of tails = number of heads + 3 ⇒ Stop simulated play

Cumulative number of heads

Outcome: $8 - $11 = $ -3

Number of coin-flips

	1	2	3	4	5	6	7	8	9	10	11
Random number:	*8,*	*1,*	*3,*	*7,*	*2,*	*7,*	*1,*	*6,*	*5,*	*5,*	*7*
Head or Tail:	*T,*	*H,*	*H,*	*T,*	*H,*	*T,*	*H,*	*T,*	*T,*	*T,*	*T*

Sample average of number of coin-flips per play: $(11 + 5 + \cdots + 7) / 14 = 7$ **(see R16)** ; *True mean is* **9**

Simulation Model for Coin-Flipping Game

❑ **Simulation 'clock'**: the number of (simulated) flips, t, that have occurred so far

❑ **State of the system**: $N(t)$ = (# of heads) – (# of tails)

❑ **Events**: flipping of a head or flipping of a tail

❑ **Event generation mechanism**:
- $0 \leq$ random digit $\leq 4 \Rightarrow$ head
- $5 \leq$ random digit $\leq 9 \Rightarrow$ tail

❑ **State transition mechanism**:

$$N(t) = \begin{cases} N(t-1)+1, & \text{if flip is a head} \\ N(t-1)-1, & \text{if flip is a tail} \end{cases}$$

❑ **Simulation end**: $N(t) = \pm 3$

❑ **Sampling observation** : $(8 - t)$ = amount won (or lost) for a simulated play of the game

Some Main Questions in Simulation Modeling

❑ Formulation and implementation of simulation models:
- • How to generate random numbers ?
- • How to construct a model ?
- • How to generate random observations from a probability distribution ?
- • How to prepare a simulation program ?
- • How to validate the model ?

❑ Experimental design and analysis:
- • When to end a simulation run ?
- • How many simulation runs are needed ?
- • How to perform a statistical analysis of results obtained from simulation runs?

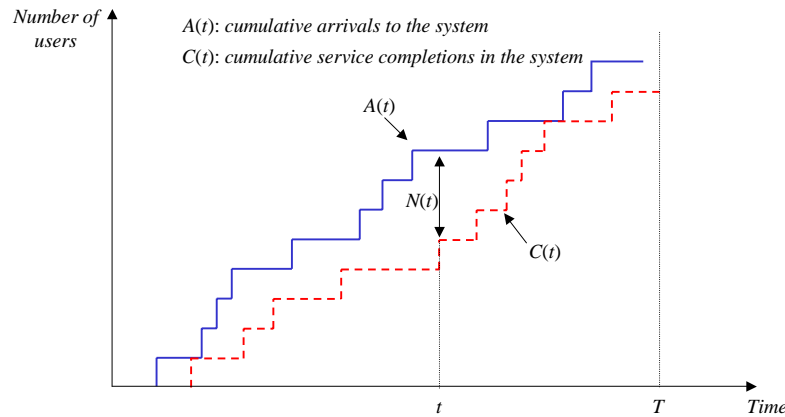
Random Numbers

❑ A random number is a random observation from the uniform distribution on interval (0, 1) (U(0,1))

❑ Theories related to random number generation methods are well established. There are examples in R16 if you are interested in details of such methods.

❑ Most computer systems have functions that allow for generating (artificial) random numbers

❑ In Excel, RAND() returns a random number that we denote r

Simulation Models for Queueing Systems

Number of users

A(t): cumulative arrivals to the system

C(t): cumulative service completions in the system

A(t)

N(t)

C(t)

t *T* *Time*

❑ Arrival and departure times are discrete events

❑ Two representations of time

- Discretize time in small time-intervals ⇒ discrete-time simulation
- Continuous-time ⇒ event-driven simulation

Simulation Model: *M/M/1* Queueing System

❑ **Simulation 'clock'**: amount of time that passed since the simulation started

❑ **State of the system**: $N(t)$ = number of customers in the queueing system at time t

❑ **Events**:

- arrival of a new customer
- service completion for the customer currently in service

❑ **Event generation mechanism**: depends on the type of the simulation

❑ **State transition mechanism for discrete-time simulation**:

$$N(t) = \begin{cases} N(t-1)+1, & \text{if an arrival occurs at } t \\ N(t-1)-1, & \text{if a service completion occurs at } t \\ N(t-1), & \text{otherwise} \end{cases}$$

❑ **Simulation end**: if the simulation clock exceeds a pre-specified (simulation) time

Events Generation for Event-Driven Simulation Models

❑ Time increment is a variable

❑ Next-event simulation-time incrementing:

- Advance time to the time of the next future event of any type, and
- Update system state and record information about performance of the system

❑ For *M/M/1* queueing system:

- future events are:
  - The next arrival
  - The next service completion
- how to obtain the time of each event?
  - Generate a random number, *r*
  - Use *r* to obtain a random observation from the probability distribution of the interarrival time or of the service time

Generation of Random Observations: The Problem

❑ *X* is a random variable

❑ Cumulative distribution function (c.d.f.) of X : $F(x) = \Pr\{X \le x\}$

❑ Problem: how to generate an x_i from X ?

❑ Random observation generation methods:

- **Inverse transformation method**
- Other methods such as Acceptance-Rejection method or the use of Central Limit Theorem

Inverse Transformation Method

❑ Method

  • Generate $r \in U(0,1)$

  • A random observation x is then the solution to $F(x) = r$

❑ $F(x)$ can be solved explicitly

❑ Example: Exponential distribution for interarrival time and service time in *M/M/1* queueing system

  • $F(x) = \Pr(t \leq x) = \int_0^x \lambda e^{-\lambda w} dw = 1 - e^{-\lambda x}$ (see Lecture 8)

  • $F(x) = r \Rightarrow 1 - e^{-\lambda x} = r$

$\Rightarrow e^{-\lambda x} = 1 - r$

$\Rightarrow -\lambda x = \ln(1-r)$

$\Rightarrow x = \dfrac{-\ln(1-r)}{\lambda}$

An Event-Driven Simulation Model for *M/M/1*

❑ A_n: arrival time of customer n

❑ D_n: departure time (service completion time) of customer n

❑ H_n: interarrival (arrival headway) time of customer n

❑ S_n: service time of customer n

❑ $A_1 = 0$ and $A_n = A_{n-1} + H_n$ for $n = 2, 3, \ldots$

❑ $D_1 = S_1$ and $D_n = S_n + \max\{A_n, D_{n-1}\}$ for $n = 2, 3, \ldots$

❑ H_n and S_n are generated from two exponential random variables

$$H_n = \frac{-\ln(1-r_A)}{\lambda} \qquad\qquad S_n = \frac{-\ln(1-r_D)}{\mu}$$

$$H_n = -LN(RAND())/\lambda \qquad\qquad S_n = -LN(RAND())/\mu$$

Length and Number of Simulation Runs

❑ As we are interested in steady state:

- We must ensure that the system has forgotten its initial state, and is not in a transient state
- Do not consider observations obtained before the system reaches steady state (This period is called a simulation warmup period)

❑ How to reach/ensure steady-state:

- Long enough simulation runs (using a large-enough number of customers)
- Statistical analysis of observations

❑ How many simulation runs:

- Typically 25 to 75, or
- A much longer simulation run, the results of which are then partition in 25 to 75 parts used to obtain 25 to 75 observations
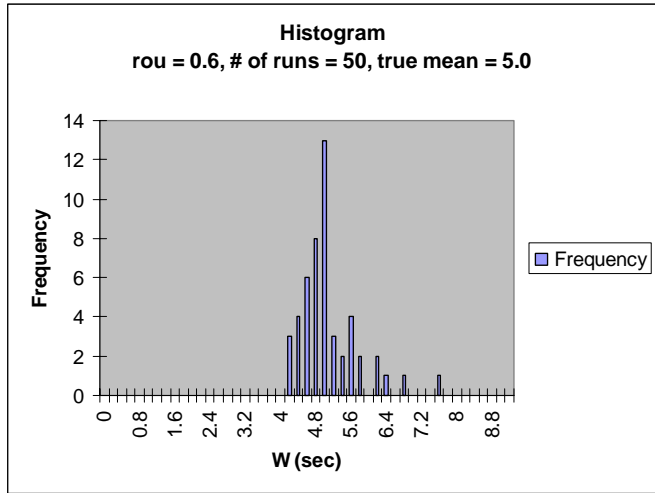
Steady State in Simulation

❑ In the simulation of queueing system:

- When the utilization ratio is low, we expect to reach steady state pretty soon, i.e. we need less number of customers

- When the utilization ratio is high, we expect to reach steady state long after the initial state, so that the system can "forget" the initial state, i.e. we need more number of customers
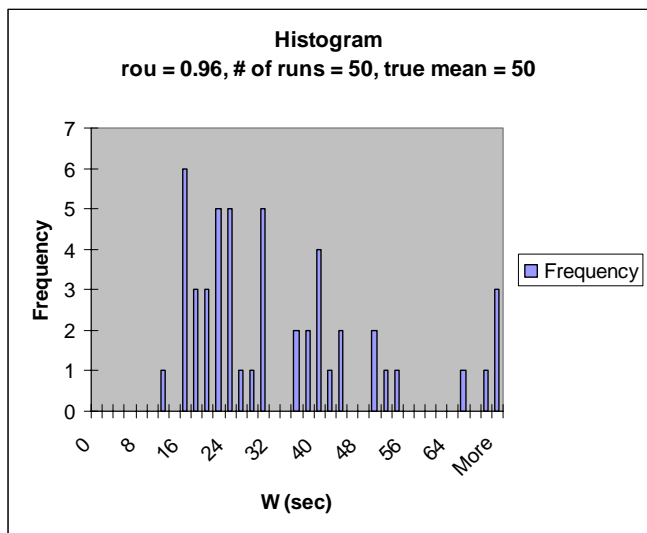
Example w/ Lower Utilization Factor

Histogram
rou = 0.6, # of runs = 50, true mean = 5.0

Example When Steady-State is Not Reached

Histogram
rou = 0.96, # of runs = 50, true mean = 50

9

Events Generation for Discrete-Time Simulation Models

❑ Discrete-time simulation model is also called fixed-time incrementing:

- Advance time by a small fixed amount, and
- Update the simulated system

❑ Two types of events during a fixed-time increment:

- One or more arrivals
- One or more service completions

❑ Note: if the fixed-time increment is small, the probability of multiple arrivals or multiple service completions is negligible (see Lecture 8)

❑ A random number is generated and used to determine whether there is an arrival (or a service completion) during the fixed-time interval

---

Discrete-Time Simulation Model for M/M/1

❑ *M/M/1* model with:

- arrival rate $\lambda = 3$ customers per hour
- service completion rate $\mu = 5$ customers per hour
- fixed-time interval $\Delta t = 6$ min

❑ Probability of one arrival during Δt:

$$P_A = \Pr(t \le \Delta t) = \int_0^{\Delta t} \lambda e^{-\lambda w} dw = 1 - e^{-\lambda \Delta t} = 1 - e^{-3 \cdot 6/60} = 0.259$$

❑ Probability of one service completion during Δt:

$$P_D = \Pr(t \le \Delta t) = \int_0^{\Delta t} \lambda e^{-\lambda w} dw = 1 - e^{-\lambda \Delta t} = 1 - e^{-5 \cdot 6/60} = 0.393$$

❑ **Event generation mechanism on fixed-time interval**, Δt:

- Generate $r_A \in (0,1)$; if $r_A < 0.259 \Rightarrow$ an arrival occurred
- Generate $r_D \in (0,1)$; if $r_D < 0.393 \Rightarrow$ a service completion occurred

Lecture 9 Summary

❑ Introduction to simulation models

❑ A simulation model for simple example (a coin-flipping game)

❑ Random numbers

❑ Simulation models for M/M/1 queueing systems

❑ Random observations

❑ An event-driven simulation model for an *M/M/1* queueing system

❑ A spreadsheet implementation of an event-driven simulation model for *M/M/1*

❑ A discrete-time simulation model for an *M/M/1* queueing system

11