

# Introduction to Simulation

---

Hongwei Zhang

<http://www.cs.wayne.edu/~hzhang>



The best advice to those about to embark on a very large simulation is often the same as Punch's advice to those about to marry: Don't!

--- Bratley, Fox, and Schrage (1986)

# Outline

---

- Common mistakes in simulation
- Terminology
- Discrete event simulation

# Outline

---

- Common mistakes in simulation
- Terminology
- Discrete event simulation

# Common mistakes in simulation

---

- Inappropriate level of detail: may think more detail is always better, but
  - More detail => More time, *Bugs*, & CPU time
  - More parameters  $\neq$  More accurate (if unable to precisely model a parameter)
- Unverified simulation models: e.g., due to software bugs
- Invalid models (e.g., for parameters): model vs. reality

# Common mistakes (contd.)

---

- Improperly Handled Initial Conditions
  - Initial part of a simulation is usually not the same as steady state, and thus should be treated separately
- Too Short Simulations
  - Length of simulation should depend on the required accuracy and the variance of observed quantities
- Poor Random Number Generators: Safer to use a well-known generator
- Improper Selection of Seeds
  - E.g., zero seeds or same seeds for all streams => correlated random streams

# Outline

---

- Common mistakes in simulation
- Terminology
- Discrete event simulation

# Terminology

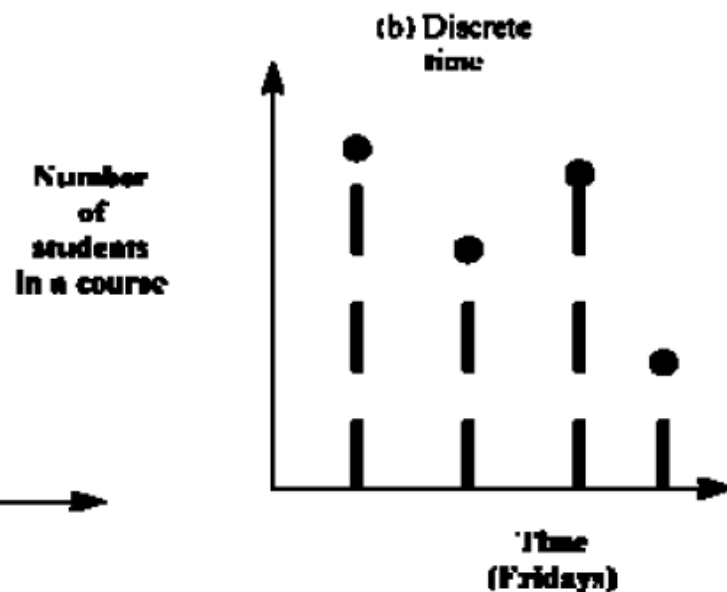
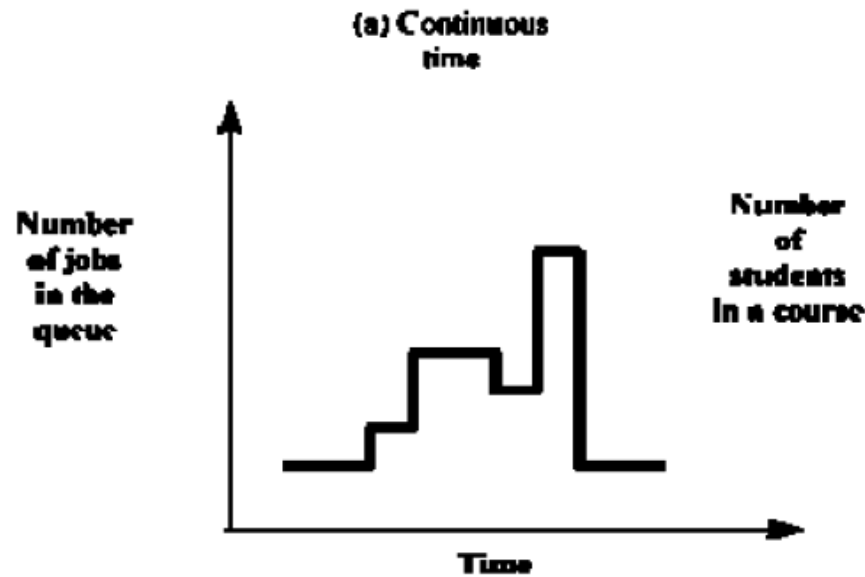
---

- **State Variables:** Define the state of a system
  - Can restart simulation from a certain stored system state, e.g., lengths of the packet queues
- **Event:** Change in the system state
  - E.g., arrival, processing, and departure of packets

# Type of models

---

- ❑ **Continuous Time Model:** State is defined at all times
- ❑ **Discrete Time Models:** State is defined only at some instants

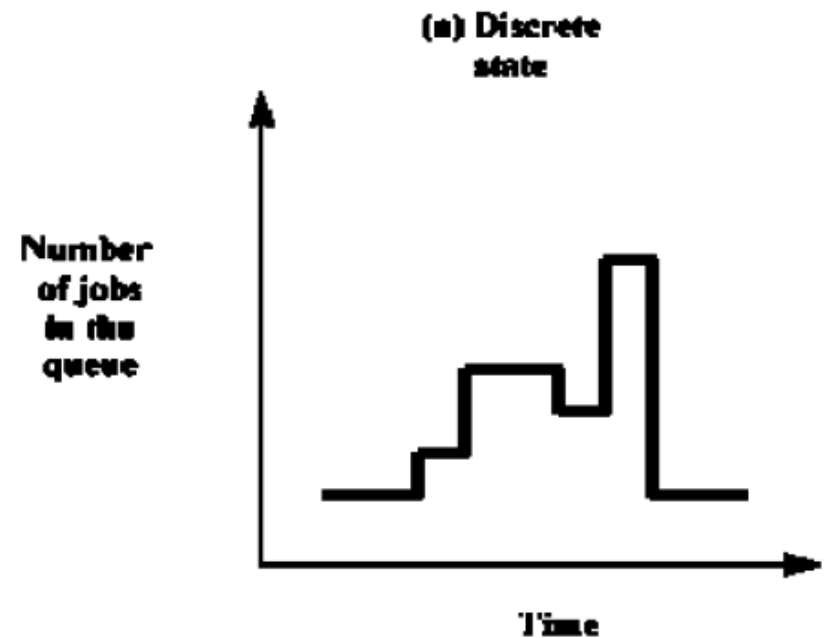
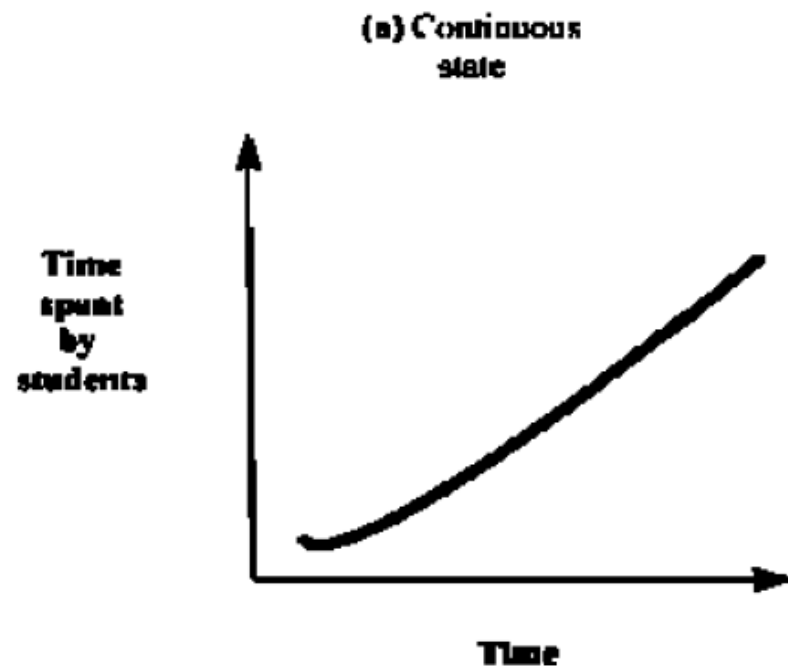




# Type of models (contd.)

---

- ❑ **Continuous State Model:** State variables are continuous
- ❑ **Discrete State Models:** State variables are discrete



## Type of models (contd.)

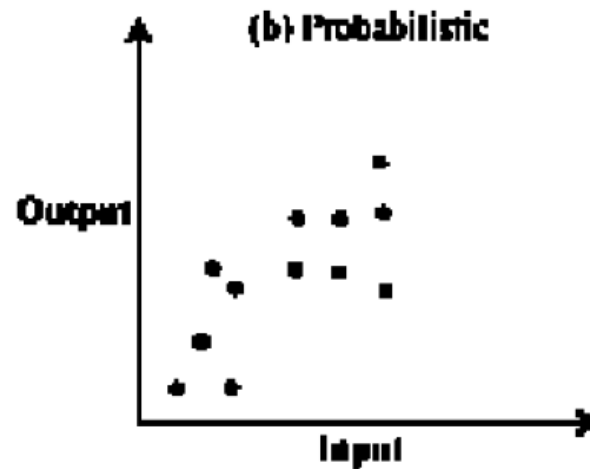
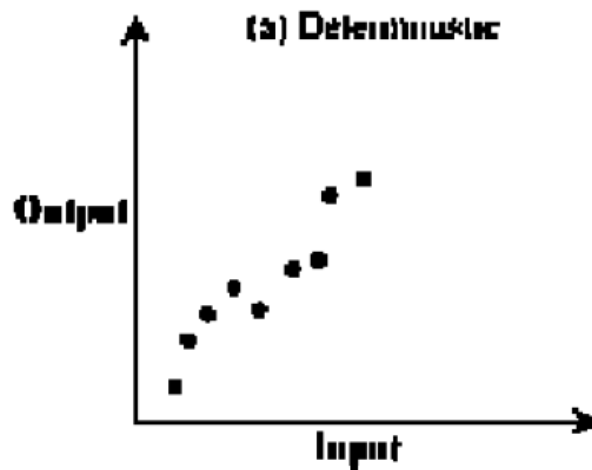
---

- ❑ Continuity of time  $\neq$  Continuity of state
  
- ❑ Four possible combinations:
  1. discrete state/discrete time
  2. discrete state/continuous time
  3. continuous state/discrete time
  4. continuous state/continuous time models

# Type of models (contd.)

---

- Deterministic and probabilistic models

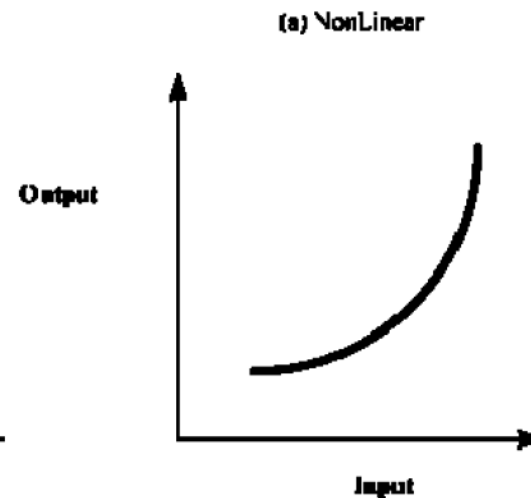
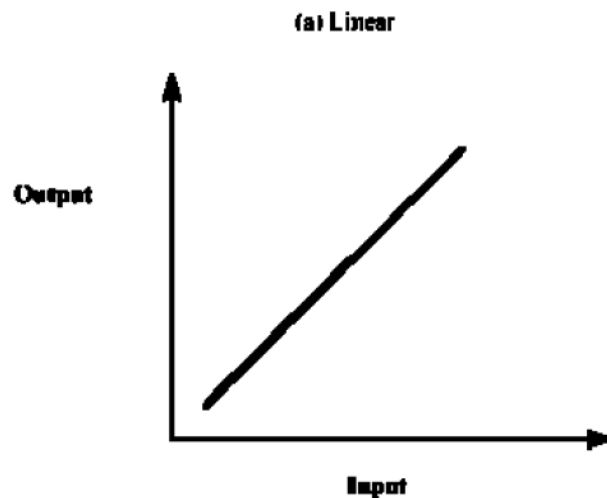


# Type of models (contd.)

---

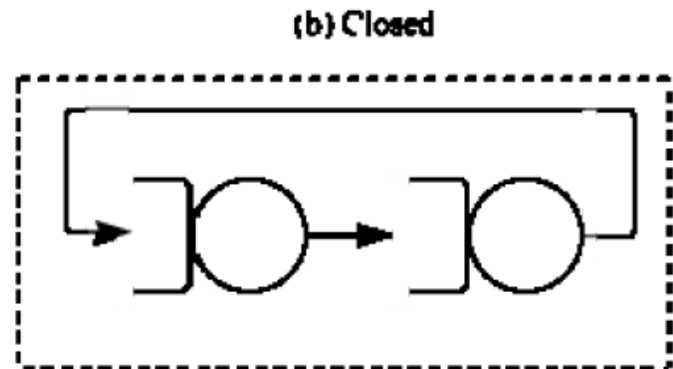
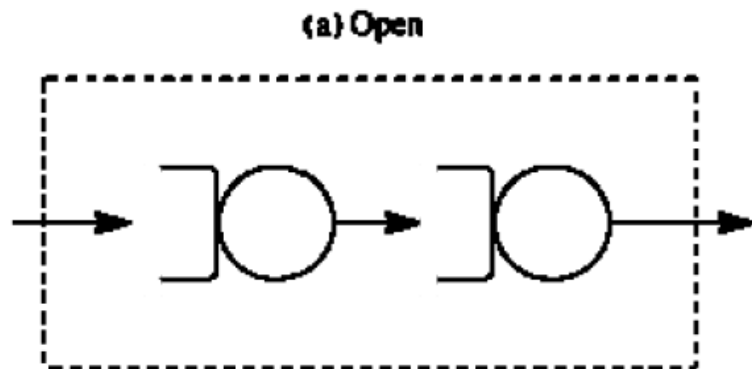
- **Static and dynamic models**: whether time is a variable in the model: no => static, yes => dynamic
- **Linear and nonlinear models**

$$\text{Output} = \text{fn}(\text{Input})$$



## Type of models (contd.)

- Open and closed models: external input => open



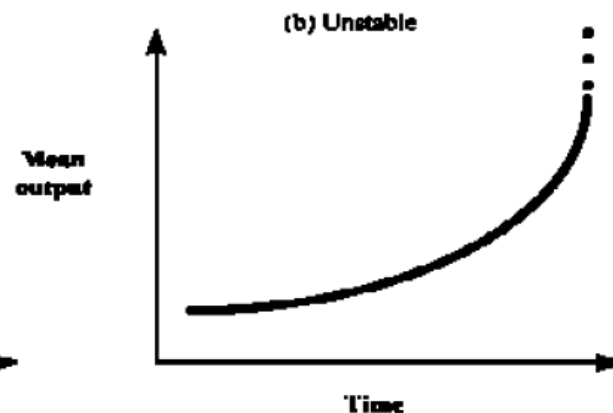
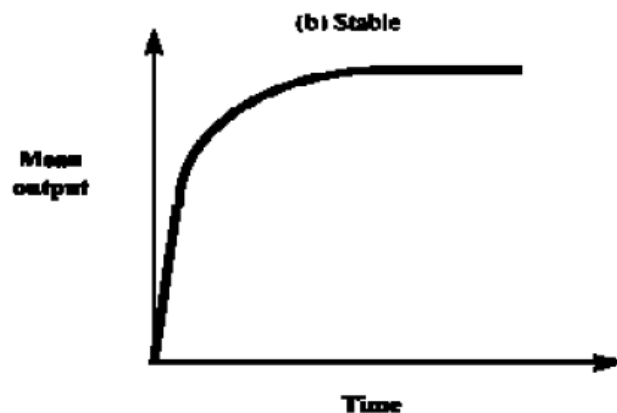
# Type of models (contd.)

---

- Stable and unstable models

Stable  $\Rightarrow$  Settles to steady state

Unstable  $\Rightarrow$  Continuously changing.



# Outline

---

- Common mistakes in simulation
- Terminology
- Discrete event simulation

# Discrete event simulations

---

- Commonly used in computer and network simulation
  - Number of queued packets => discrete event simulation
  - *In contrast*, concentration of a chemical substance => continuous event simulations
- Discrete state/event  $\neq$  discrete time



# Components of discrete event simulators

---

- Event Scheduler: controls events in a simulation. E.g.,
  - Schedule event X at time T
  - Cancel a previously scheduled event X
  - Hold event X for a time interval  $d_t$
  - Hold event X indefinitely
  - Schedule an indefinitely held event
- (Simulation Clock &) Time Advancing Mechanism
  - Unit-time approach: check events to be activated after every unit time
  - Event-driven approach: directly increments time to the time of the next earliest occurring event; commonly used in computing simulation

# Components (contd.)

---

- System State Variables: e.g.,
  - Global = Number of packets
  - Local = CPU time required for processing a packet
- Event Routines: One per event
  - E.g., job arrivals, job scheduling, and job departure
- Input Routines: Get model parameters from users
- Initialization Routines: Set the initial state and seeds
- Trace Routines: print out intermediate variables; on/off feature is usually desirable

## Components (contd.)

---

- Report Generator
- Dynamic Memory Management: Garbage collection
- Main Program: brings all the routines together (e.g., calls input routines, initializes the simulation, executes various iterations, and finally calls the output routines)

# Commonly used network simulators

---

- Sensor networks
  - TOSSIM: TinyOS networks
  - EmStar: general sensor networks based on TinyOS, Linux, etc.
- Generic network simulators
  - ns-3
  - qualnet/glomosim
  - opnet

# Summary

---

- Common mistakes in simulation
- Terminology
- Discrete event simulation