

LECTURE 2

SIMULATION AND MODELING

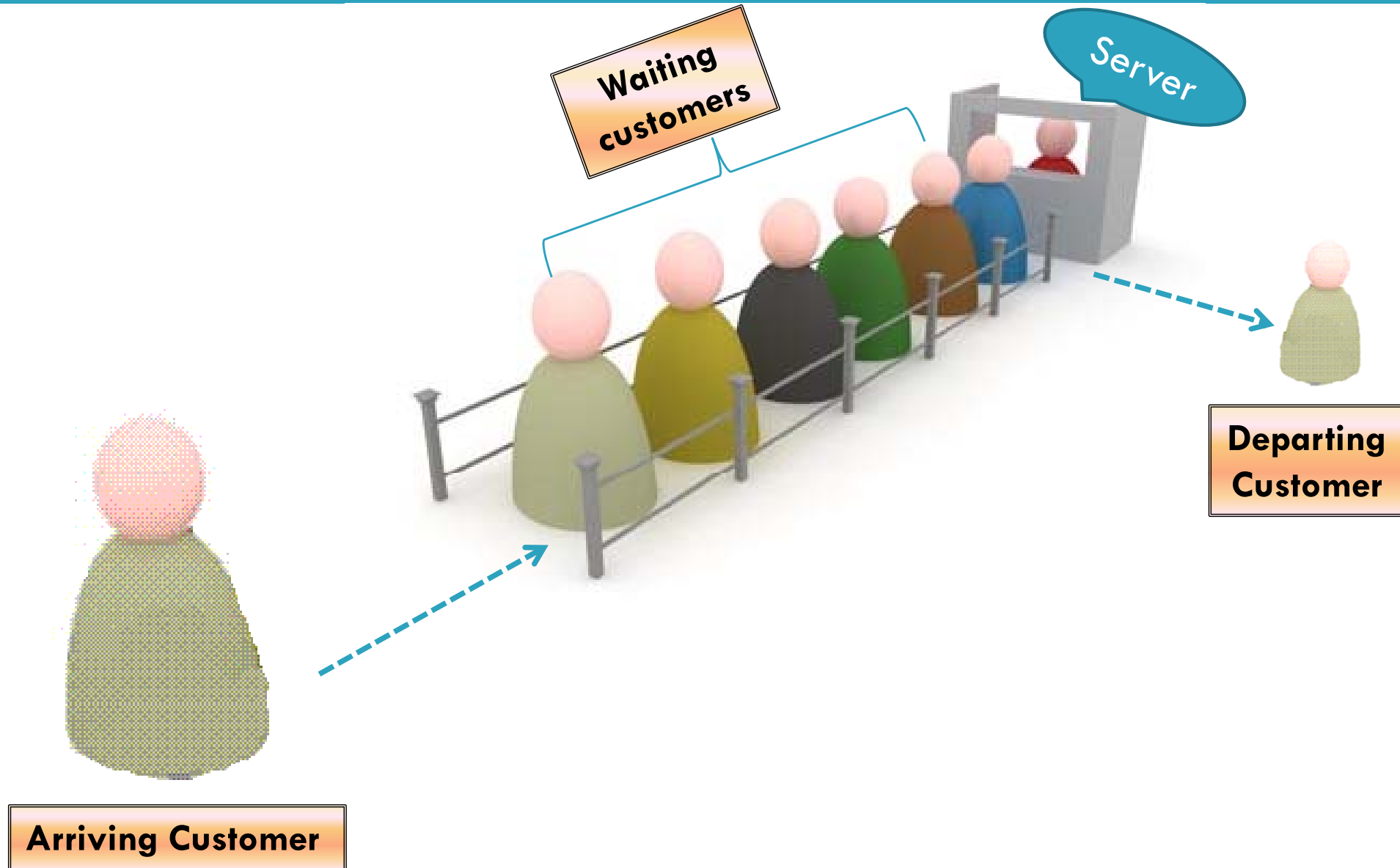
CSE 411

Md. Tanvir Al Amin, Lecturer, Dept. of CSE, BUET

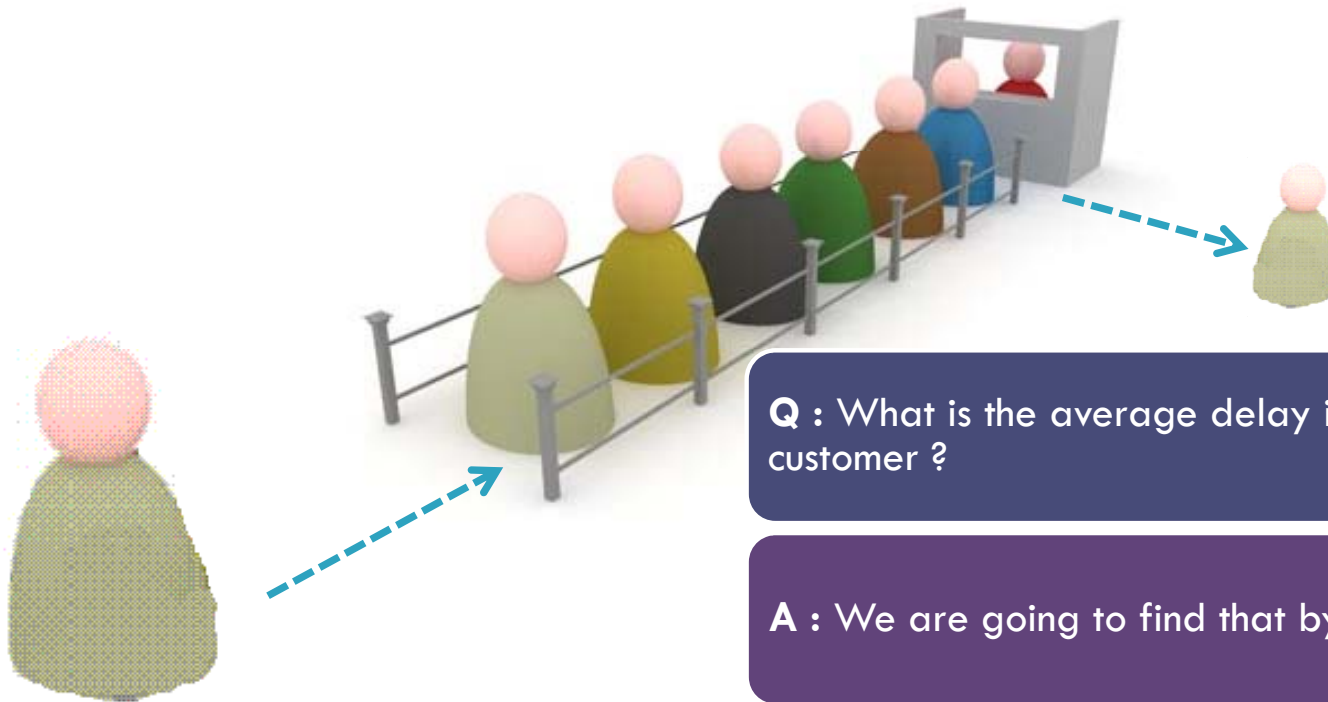
Discrete Event Simulation

- ...is a **Simulation** where, State variables change, only at a **discrete** set of **points** in time.
 - ▣ Those **points** in time are the ones, at which an **Event** occurs.
 - ▣ Hence called **Discrete Event Simulation**
- Ques : What is **Event** ?
 - ▣ Ans : An instantaneous occurrence, that may change the state of the system.

Example : Single Server Queue



Example : Single Server Queue



Q : What is the average delay in queue for an arriving customer ?

A : We are going to find that by simulation

Q : But isn't there a simple queuing theory formula which finds average delay ? Then why simulation ??

A : Yes, there are analytic solutions for this type of simple queues, but what about multiple queues, or queues with several conditions ? Queuing systems can get arbitrarily complex !!!

Single Server Queue : State variables

- For the objective of estimating the average delay, **state variables** for a **discrete event simulation** of the queue would be :
 - ▣ **Server Status** --- IDLE / BUSY
 - ...to determine upon a customer's arrival, whether the customer can be served immediately or must join in the queue.
 - ▣ **Number of customers waiting** --- if any
 - ...to determine whether the server will become idle or begin serving the first customer in the queue.
 - ▣ **Time of arrival** of each person in queue
 - ...to compute his delay in queue, which is service starting time minus arrival time.

Single Server Queue : Events

□ Arrival of a customer

- Arrival is an event, since it causes the state variable “**Server Status**” to change from **idle** to **busy**
- OR, the state variable “**Number of customers waiting**” to increase by 1.

□ Completion of service for a customer (Departure)

- Departure is an event, since it causes the “**Server Status**” to change from **busy** to **idle**.
- OR “**Number of customers waiting**” to decrease by 1.

Other events

- Both events described here, change the state of the state variables.
- But some more events may be used which do not actually change any state
 - ▣ Example : Schedule the end of simulation run at a particular time.

Simulation Clock

- DES models are dynamic.
- Needed to keep the track of current value of **simulated time**.
- We keep a clock named, **simulation_clock**
- There is generally no relationship between simulated time and the time needed to run a simulation on the computer

Time Advance Mechanisms

- Next Event time advance
 - ▣ Mainly used by simulators
 - ▣ Explained in next slides.
- Fixed Increment time advance
 - ▣ Simulation clock advanced in increments of exactly Δt time units.
 - ▣ In one or more events were scheduled to have occurred during the interval, they are considered to have occurred at the end of the interval and the system state are updated accordingly.

Next Event time advance

- Simulation clock initialized to zero
- Time of occurrence of future events (which are known at time = 0) are determined.
- Simulation clock is then advanced to the time of occurrence of most imminent of these future events and system states are updated accordingly.
 - In the process of this update, new events will be scheduled
- Simulation clock is again advanced to the time of next imminent event....again update system states.
- Continued, until some prespecified stopping condition is satisfied.

Next Event time advance

- Periods of inactivity are skipped over by jumping the clock from event time to event time.
- Fixed increment time advance does not skip over these inactive periods, which can eat up a lot of computer time
- We can use a priority queue, or a sorted linked list to implement it.
- Can you remember nachos ... `scheduler.cc` or `interrupt.cc` ??

Example : Single Server Queue

- Here we use next event time advance approach to manage the events of the single server queue.
- We use the following notation :

t_i = time of arrival of the i - th customer

$A_i = t_i - t_{i-1}$ = interarrival time between $(i - 1)$ st and i - th arrivals of customers

S_i = Time that server actually spends serving the i -th customer (exclusive of queue delay)

D_i = delay in queue of the i -th customer

$c_i = t_i + D_i + S_i$ = time that i - th customer completes service and departs

e_i = time of occurrence of the i -th event of any type (i - th value the simulation clock takes on)

Example : Single Server Queue

- Each of the defined quantities are generally random variables.
- Assume we already know the probability distributions of the inter arrival times A_1, A_2, \dots and the service time S_1, S_2, \dots . Suppose the cumulative distribution functions are F_A and F_S

Example : Single Server Queue

- At time $e_0=0$ status of server is idle, time t_1 of first arrival is determined by generating A_1 from FA and adding to it 0.
- Simulation clock advanced from e_0 to the time of the next event, $e_1 = t_1$.
- Customer arriving at t_1 , finds server idle, so se immediately enters service and has a delay in queue $D_1 = 0$.
- Status of server changed from idle to busy.

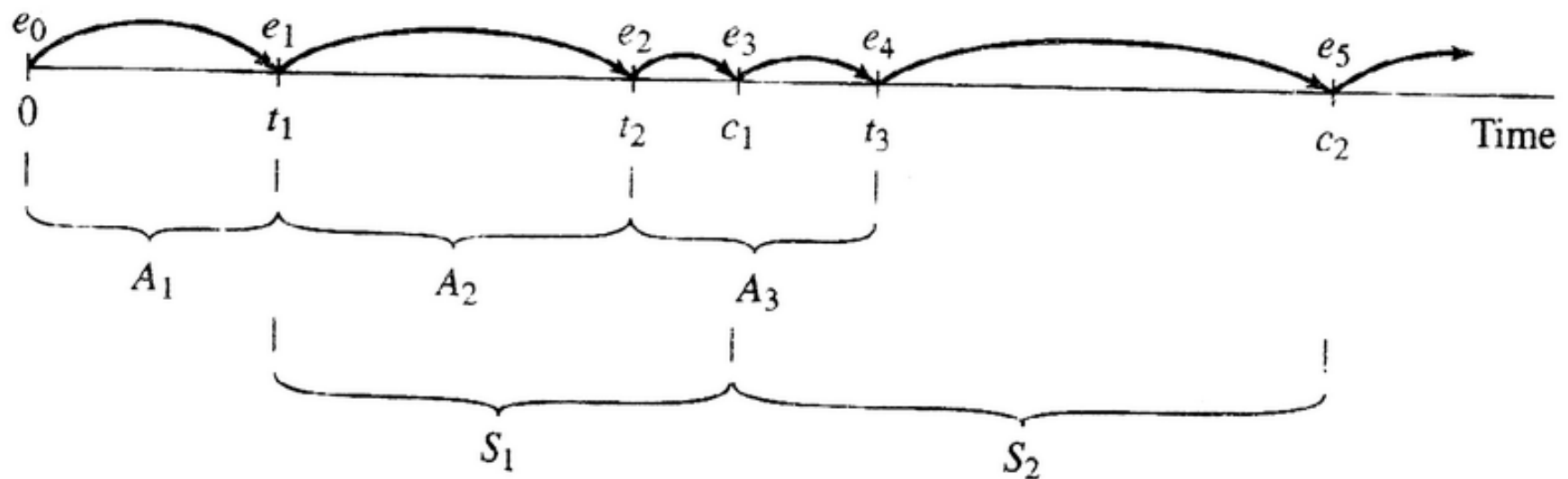
Example : Single Server Queue

- C1, when arriving customer will complete, is computed by generating $S1$ from F_s and adding it to $t1$.
- Finally, second arrival time is computed as $t2 = t1 + A2$, where $A2$ is generated from F_A .
- Simulation clock advanced from $e1$ to time of next event, $e2$.
- Customer arriving at $t2$, finds server busy,
 - ▣ Number of customers in queue becomes 0 to 1
 - ▣ Time of arrival of this customer is recorded.
 - ▣ His service time $S2$ is **not** generated at this time.
 - ▣ Time of third arrival $t3$ is computed as $t3 = t2 + A3$.

Example : Single Server Queue

- If $t_3 > c_1$,
 - ▣ simulation clock advanced from e_2 to $e_3=c_1$
 - ▣ Customer completing service departs
 - ▣ Customer in the queue begins service, and his delay in queue and service completion time are computed as $D_2=c_1-t_2$ and $c_2=c_1+S_2$ (S_2 generated from F_3)
 - ▣ Number of customers in queue, decreased from 1 to 0.
- If $t_3 < c_2$
 - ▣ Simulation clock advanced from e_3 to $e_4=t_3$
 - ▣ ...

Example : Single Server Queue



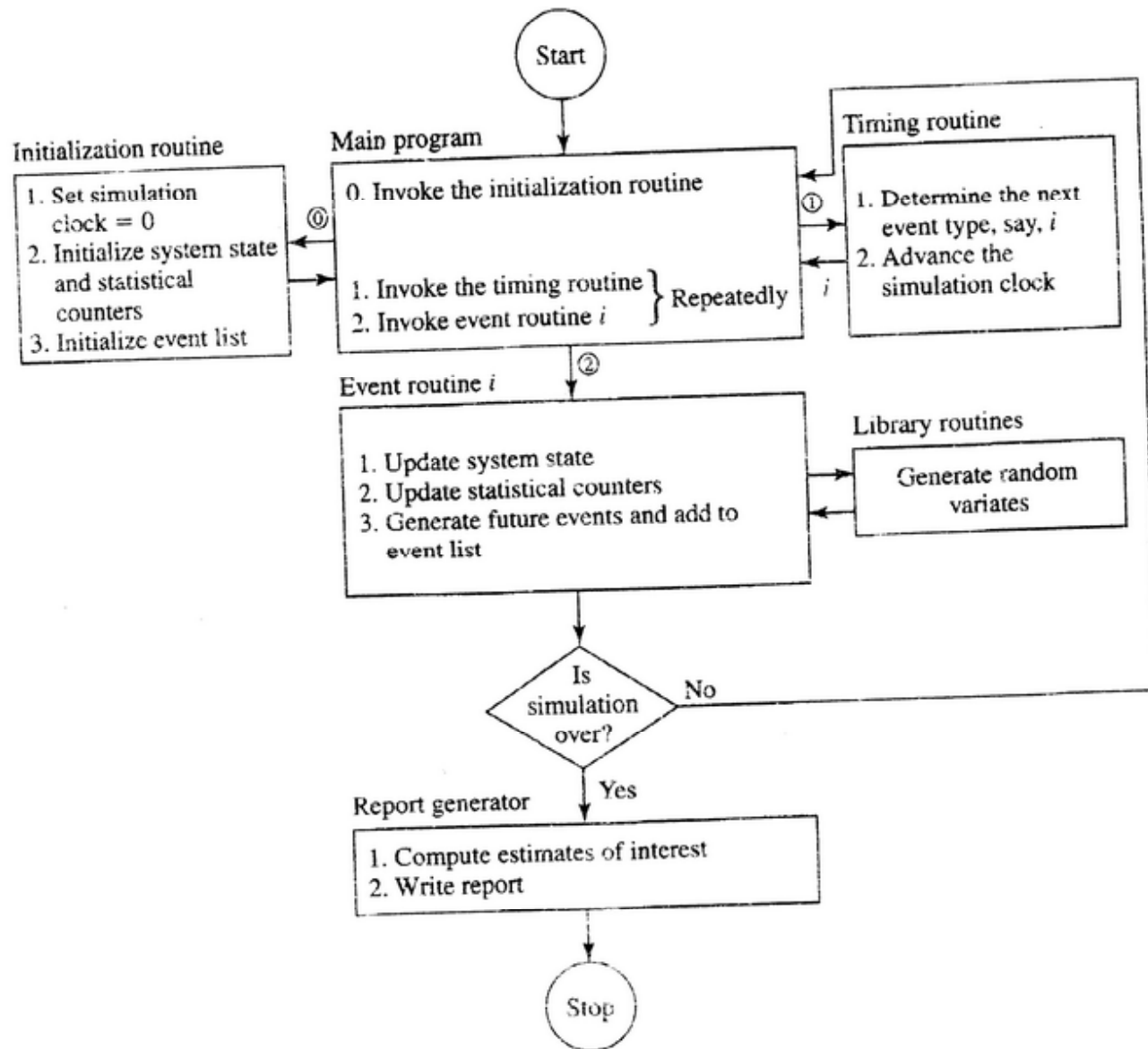
Data structures of DES

- System State
 - Collection of state variables necessary to describe the system at a particular time
- Simulation Clock
 - Current value of simulated time
- Event List
 - List containing the next times of the future events.
- Statistical Counters
 - Variables used for storing statistical information about System performance

Routines for DES

- Initialization Routine
 - A subprogram to initialize the simulation model at time = 0
- Timing Routine
 - Determines next even from event list
 - Advances the simulation clock to that time
- Event Routine
 - Updates the system state when a particular type of event occurs
- Library Routines
 - Set of subprograms used to generate random observations from probability distributions.
- Report Generators
 - Computes estimations of performance parameters from statistical counters
- Main Program
 - Invokes the timing routine to determine next event and transfers control to the corresponding event routine to update system state. Main program may also check for termination or invoke report generator when simulation is over.

Flow of control for DES



Reference in the text

- In today's lecture, we covered :
 - ▣ Kelton 1.3, 1.3.1, 1.3.2, Appendix 1A,
- In Lecture 1, we covered :
 - ▣ Kelton 1.1, 1.2