Discrete Event Simulation

DES

DES

- Introduction:
  - System
  - Models
  - Discrete event simulation and
  - Continuous simulation
- Discrete Event Simulation:
  - Time-Advance Mechanisms
  - Event Modeling of discrete dynamic systems
  - Single-Server Single-Queue Model
  - Event graphics
  - Monte Carlo Simulation

Categories of Systems

↓

Discrete and Continuous Systems

Discrete Systems

The state of the system changes only at discrete points in time.
These points in time are the ones at which the event occurs/change in state occurs.

Firing of a gun on an enemy target.

Model of Bank:

Number of customers waiting in line being served.

Queuing, Inventory, Machine Shop Models

Continuous Systems

The state of the system changes continuously

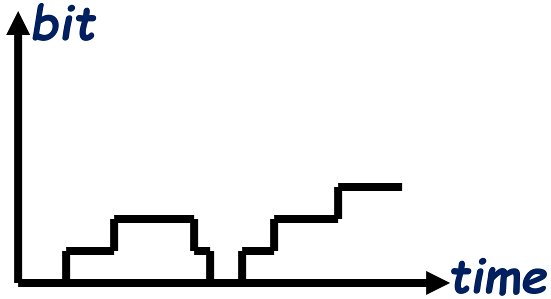
- Fluid flow in a pipe
- Chemical Process

Learn M & S : From B K Sharma

Discrete Vs. Continuous Models

Discrete Models

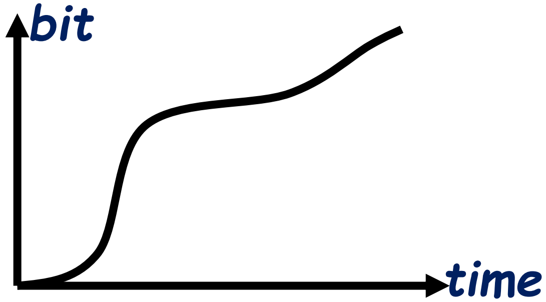
# of cars in a parking lot



Discrete Model

Continuous Models

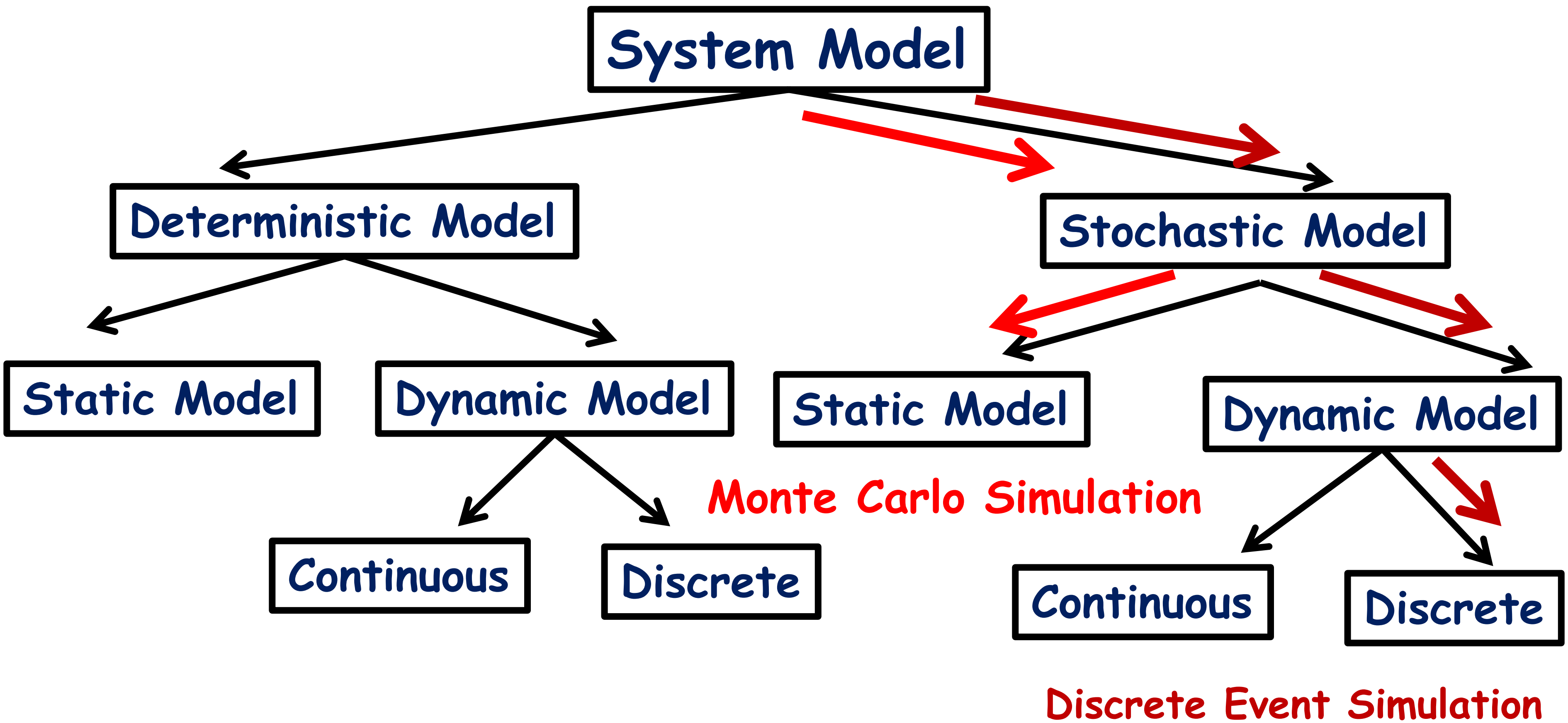
Bit Arrival in a Queue



Continuous Model

Learn M & S : From B K Sharma

Models Taxonomy



What is Discrete Event Simulation?

Computer model for a system

where

changes in the state of the system occurs (Dynamic)

at discrete points in simulation time (Discrete)

and

system state can not be predicted entirely (stochastic).

What is Discrete Event Simulation?

models a system

whose state may change

only at discrete point in time

and

randomly.

What is Discrete Event Simulation?

DES is

Stochastic: Probabilistic

at least some of the system state variables are random:

Inter-arrival times and service times are random variables

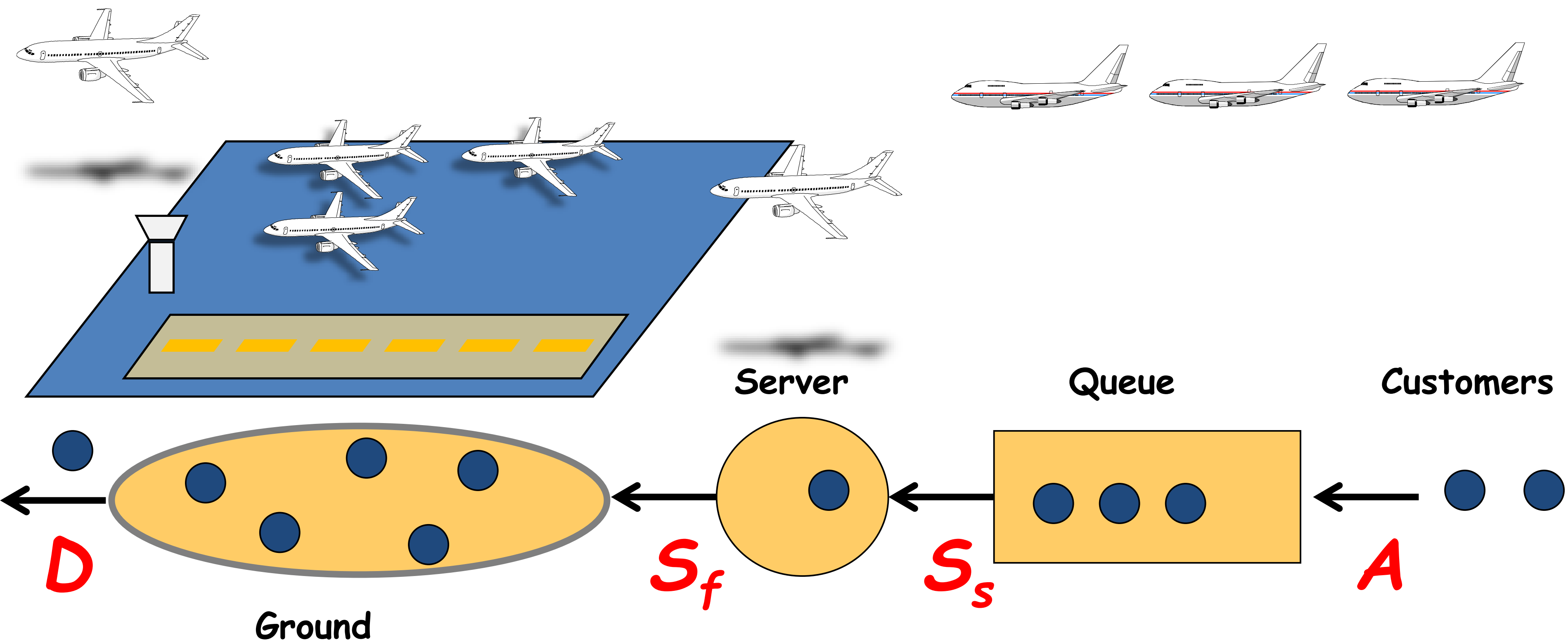
Dynamic: Changes over time and

Discrete Event : The state variables change instantaneously at separate points in time

The system can change at only a countable number of points in time.

These points in time are the ones at which an event occurs.

An Example: Airport System



DES :Key Concepts

System States

System state at Current Time is stored in a variable called State variable

Events

An instantaneous occurrence that changes the state of a system.

Simulation Time

When each event is scheduled / occurs.

Learn M & S : From B K Sharma

DES :Key Concepts

System States

State Variables

For Example: Airport System

- Q: # of aircrafts waiting for landing
- G: # of aircrafts on the ground
- B: y/n; y if the runway is busy

Events

- Arrival: denotes aircraft arriving in air space of airport
- Landed: denotes aircraft landing
- Departure: denotes aircraft leaving

Simulation Time

E1 is scheduled at 10:00 AM then it is called Simulation Time of Event E1

Learn M & S : From B K Sharma

DES :Key Concepts

Simulation Clock

A variable representing simulation time

Future Event List

A List of Events to occur in future time

Delay

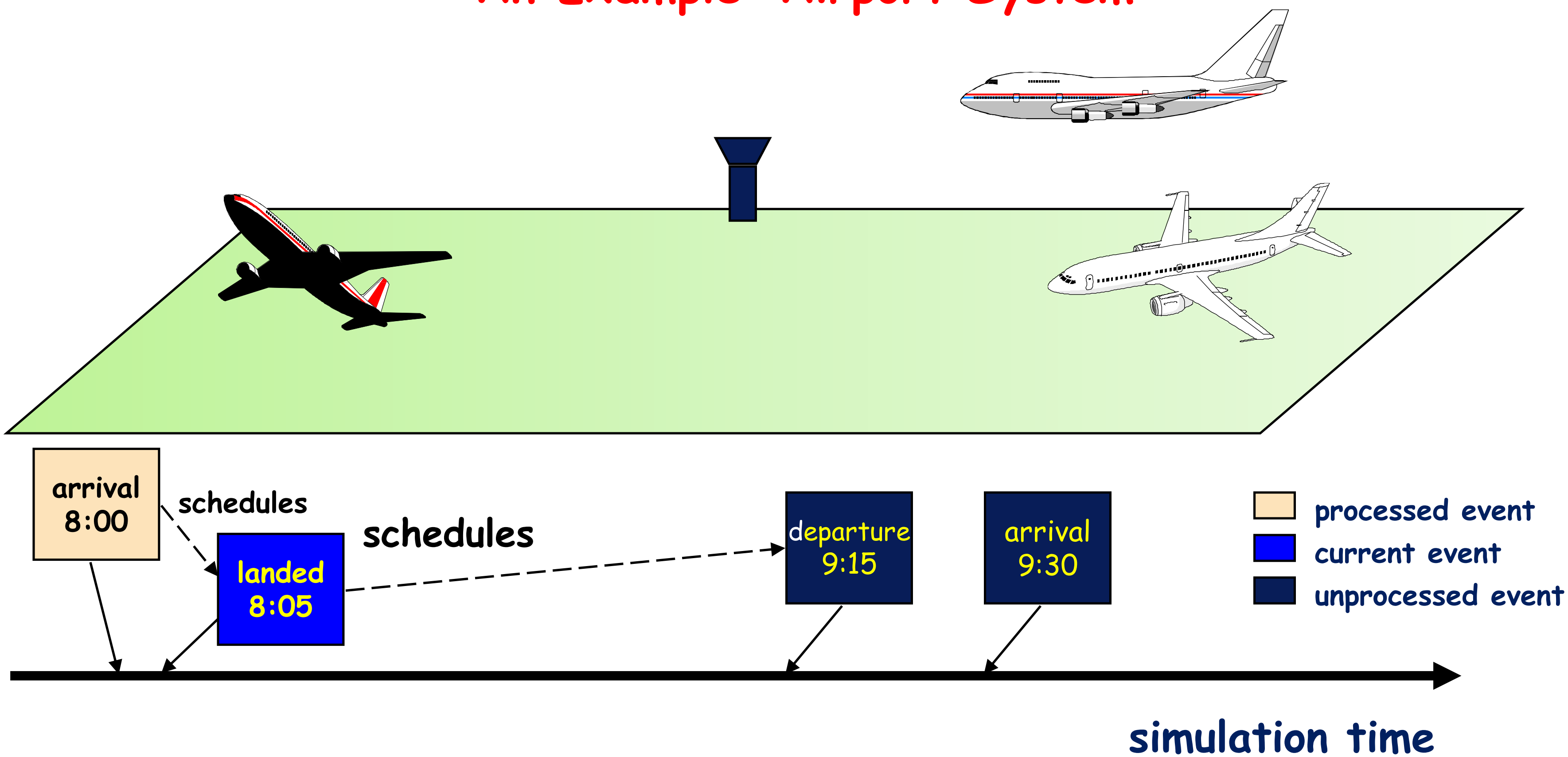
A duration of time of unspecified indefinite length, which is not known until it ends (customer delay in waiting line)

Learn M & S : From B K Sharma

DES: Key Concepts

Single Server Queue

An Example: Airport System



Single Server Queuing Model

The machine shop model

There is just one service technician.

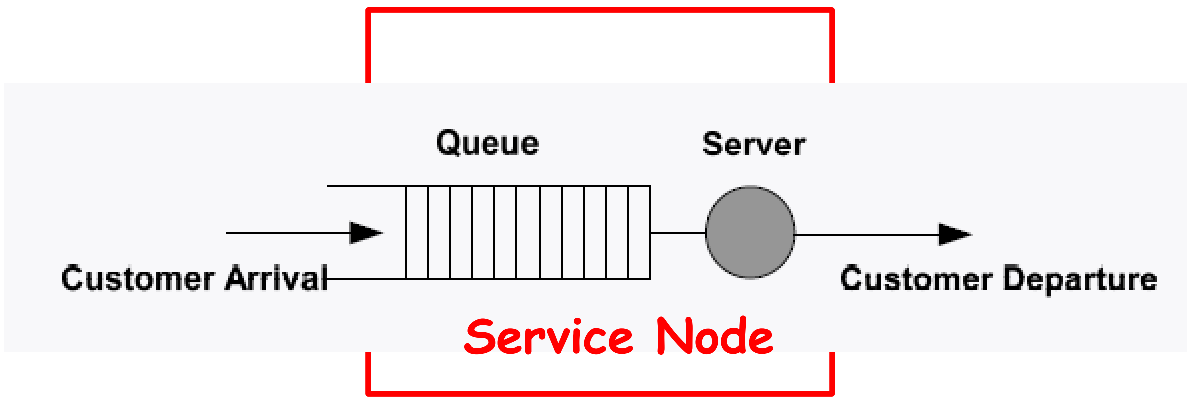
The “jobs” are the machines to be repaired and the “server" is the service technician.

Bank Model

If there is only one Cashier in the Bank.

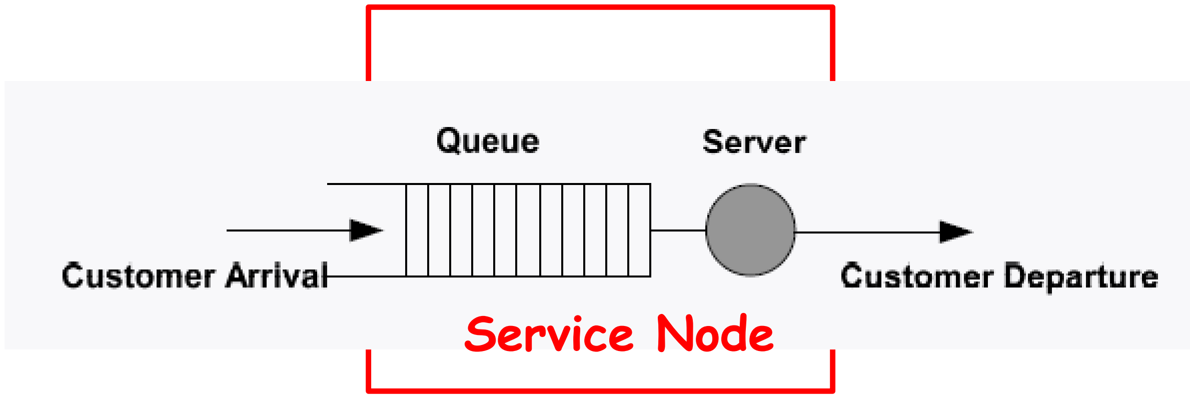
The “jobs” are Customers to be serviced and server is “Cashier”.

Example: A Single Server System



Service Node: Queue + Server

Example: A Single Server System



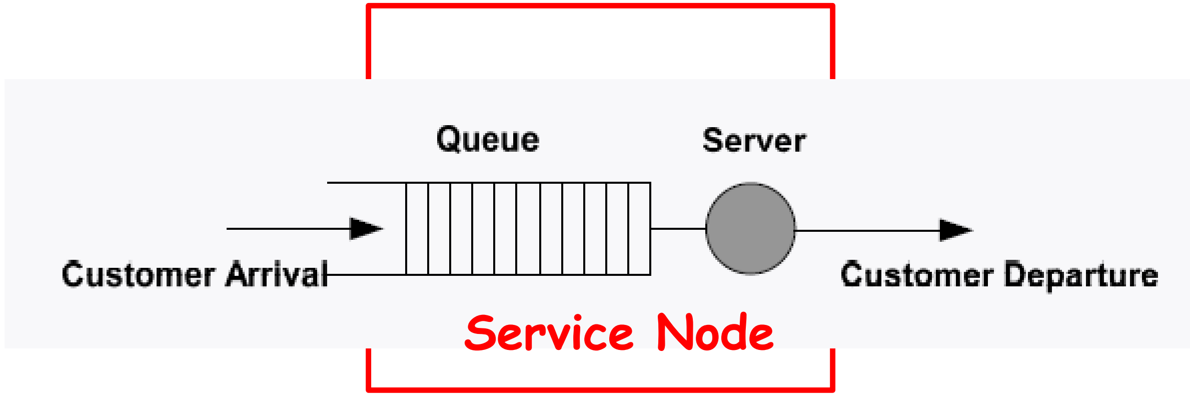
Service Node: Queue + Server

Jobs (customers) arrive at the service node at **random points in time** seeking service.

When service is provided, the **service time** involved is also **random.**

At the completion of service, jobs depart.

Example: A Single Server System



The service node operates as follows:

as each (new) job arrives:

if the server is busy then the job enters the queue

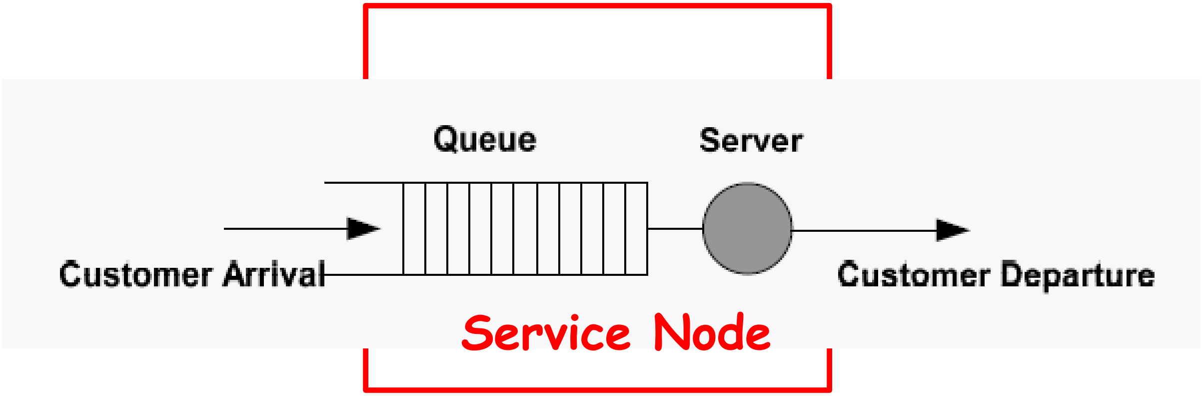
else the job immediately enters service

as each (old) job departs:

if the queue is empty then the server becomes idle,

else a job is selected from the queue to immediately enter service.

Example: A Single Server System



At any time, the state of the server will either be busy or idle and the state of the queue will be either empty or not empty.

If the server is idle, the queue must be empty

If the queue is not empty then the server must be busy.

Algorithm: A Single Server System

$a_1, a_2, \ldots\ldots a_n$ are arrival times
$s_1, s_2, \ldots\ldots s_n$ are service times
are known

c_n is the departure time of jobs Initially zero.

i_n denotes Job Numbers Initially zero.

The server is initially idle

Queue discipline is FIFO

Queue Capacity is infinite

The GetArrival() and GetService() procedures read the next arrival and service time respectively.

This algorithm computes the delays $d_1, d_2, \ldots.d_n$

Algorithm: A Single Server System

```
c_0 = 0.0; /* assumes that a_0 = 0.0 */
 i = 0;
while ( more jobs to process )
    {
        i++;
        a_i = GetArrival();
        if (a_i < c_{i-1})
           /* calculate delay for job i */
                 d_i = c_{i-1} - a_i;
        else
             /* job i has no delay */
                 d_i = 0;
        s_i = GetService();
        /* calculate departure time for job i */
        /* or c_i=a_i +w_i */
        c_i = a_i + d_i + s_i;
    }
n = i;
return d_1, d_2, ...... d_n;
```

i	1	2	3	4	5	6	7	8	9	10
a_i	15	47	71	111	123	152	166	226	310	320
s_i	43	36	34	30	38	40	31	29	36	30

Algorithm: A Single Server System

Example

Let $n=10$ $\quad c_0=0$ $\quad i=0$

Job	a_i	s_i		$c_i=a_i+s_i+d_i$	$d_i=c_{i-1}-a_i$
0				0	
1	15	43	Is $a_i<c_i-1$? No	58	0
2	47	36	Is $a_i<c_i-1$? Yes	94	58-47=11
3	71	34	Is $a_i<c_i-1$? Yes	128	94-71=23
4	111	30	Is $a_i<c_i-1$? Yes	158	128-111=17
5	123	38	Is $a_i<c_i-1$? Yes	196	158-123=35

Algorithm: A Single Server System

Example

Job	a_i	s_i	$c_i=a_i+s_i+d_i$	$d_i=c_{i-1}-a_i$
0			0	
1	15	43	58	0
2	47	36	94	58-47=11
3	71	34	128	94-71=23
4	111	30	158	128-111=17
5	123	38	196	158-123=35
6	152	40	236	196-152=44
7	166	31	267	236-166=70
8	226	29	296	267-226=41
9	310	36	346	0
10	320	30	376	346-320=26

347

Algorithm: A Single Server System

Example

From Algorithm

i	:	1	2	3	4	5	6	7	8	9	10
a_i	:	15	47	71	111	123	152	166	226	310	320
d_i	:	0	11	23	17	35	44	70	41	0	26
s_i	:	43	36	34	30	38	40	31	29	36	30

For future reference, note that the last job arrived at time $a_n = 320$ and departed at time $c_n = a_n + d_n + s_n = 320 + 26 + 30 = 376$.

Algorithm: A Single Server System

Job-Averaged Statistics

For the First n jobs,

The average inter-arrival time: $\bar{r} = \frac{1}{n}\sum_{i=1}^{n} r_i = \frac{a_n}{n}$

The average service time: $\bar{s} = \frac{1}{n}\sum_{i=1}^{n} s_i.$

The arrival Rate: $1/\bar{r}$

The Service Rate: $1/\breve{\bar{s}}$

Algorithm: A Single Server System

Job-Averaged Statistics

For n=10 jobs in our example:

$$\bar{r} = a_n/n = 320/10 = 32.0$$

And $\bar{s}$ =347/10 $\bar{s} = 34.7$

If time in this example is measured in seconds, then the average inter-arrival time is 32:0 seconds per job and the average service time is 34:7 seconds per job

The corresponding arrival rate is $1/\bar{r}$ = 1/32.0 ,approx =0.031 jobs per second

The service rate is $1/\bar{s}$ = 1/34.7, approx= 0.029 jobs per second.

Algorithm: A Single Server System

Job-Averaged Statistics

In this particular example, the server is not quite able to process jobs at the rate they arrive on average.

For the First n jobs,

the average delay in the queue: $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$

the average wait in the service node: $\bar{w} = \frac{1}{n}\sum_{i=1}^{n} w_i.$

$$= \frac{1}{n}\sum_{i=1}^{n}(d_i + s_i)$$

$$= \frac{1}{n}\sum_{i=1}^{n} d_i + \frac{1}{n}\sum_{i=1}^{n} s_i = \bar{d} + \bar{s}.$$

Algorithm: A Single Server System

Job-Averaged Statistics

Our Example $\bar{d}$ = 26.7 and $\bar{s}$ = 34:7.

Therefore, the average wait in the node is

$\bar{w}$ = 26:7 + 34:7 = 61:4.