CENX570: Simulation and Modelling
Dr Nasser Eddine Rikli
December 2023

Project Phase 2
M/M/1 queue and variations

MOHAMMED SHAHZAD
444105788@STUDENT.KSU.EDU.SA

Contents of this presentation

- Introduction
  - M/M/1 Queues
  - Components
- Simple M/M/1 Queuing system
  - Algorithms, flowcharts, Analytical solutions and results discussion
- 3 Variations of M/M/1 queues implemented
  - Code snippets, flowcharts, Analytical solutions
  - Results comparison with M/M/1

Introduction

- M/M/1 queues
- In queueing theory, a discipline within the mathematical theory of probability.
- An M/M/1 queue represents a system having a single server, where arrivals are determined by a Poisson process and job service times have an exponential distribution.

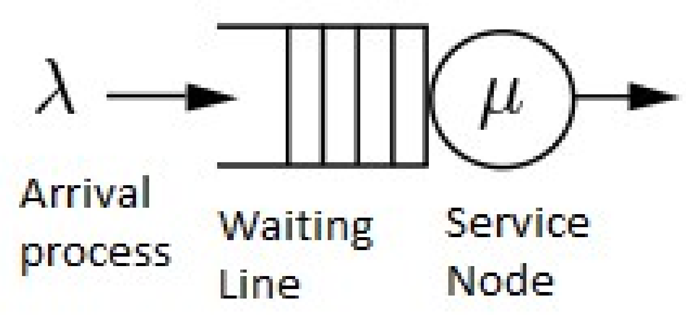


Figure: M/M/1 queuing system

Introduction

- The model's name is written in Kendall's notation.
- The model is the most elementary of queueing models
- An extension of this model with more than one server is the M/M/c queue.

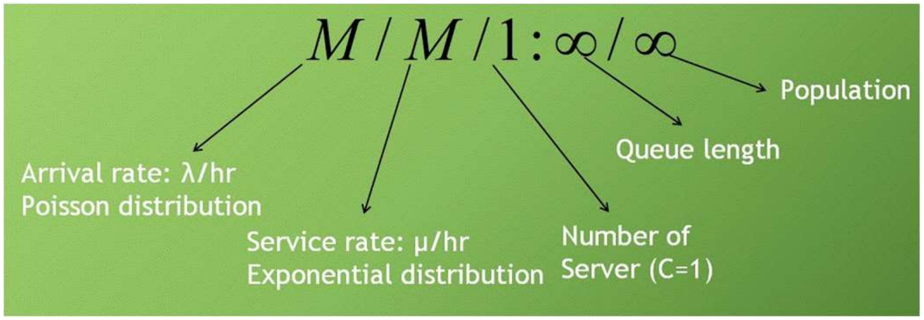


Figure: M/M/1 in Kendall notation

Introduction

- Components on M/M/1 queues
- If any customer is willing to get a service, he should check whether a server is idle or not.
- If the server is vacant, customer gets the service immediately.
- However, if at least one customer is waiting for the service in front of each of the servers, then the new arrival should line up.

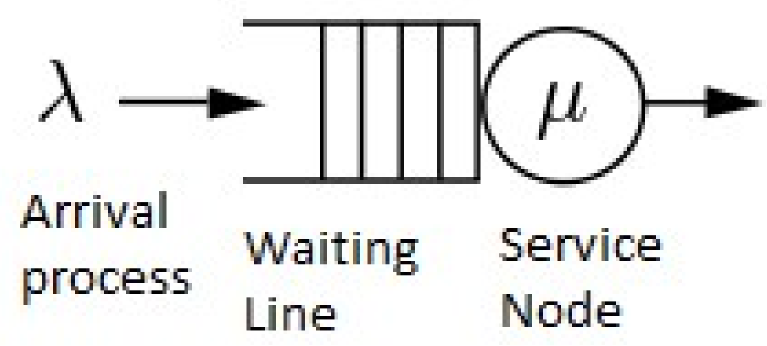
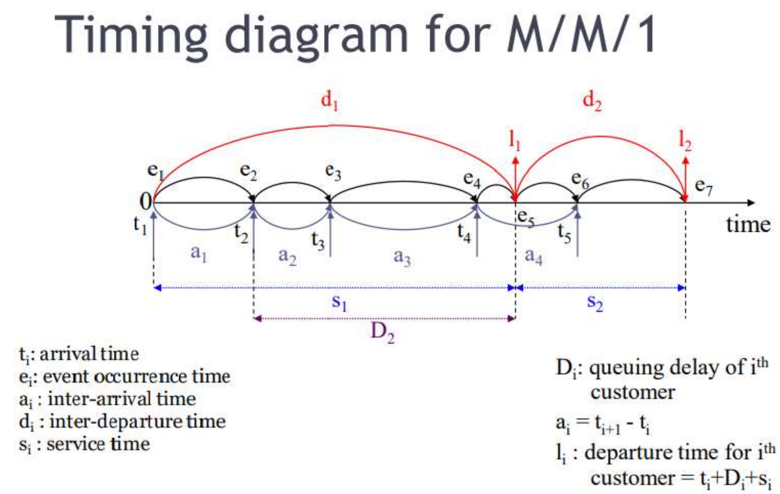


Figure: The basic queueing model where the procedure of a simple queueing system is shown.

Introduction

- Components on M/M/1 queues
- From the time someone starts standing in a queue until getting served, there are certain steps to follow.
- These steps are called the components of a queue which are characterized by the arrival process of customers, behavior of customers, service times, service discipline and service capacity



Simple M/M/1 queuing system

- m/m/1 example
- Algorithm

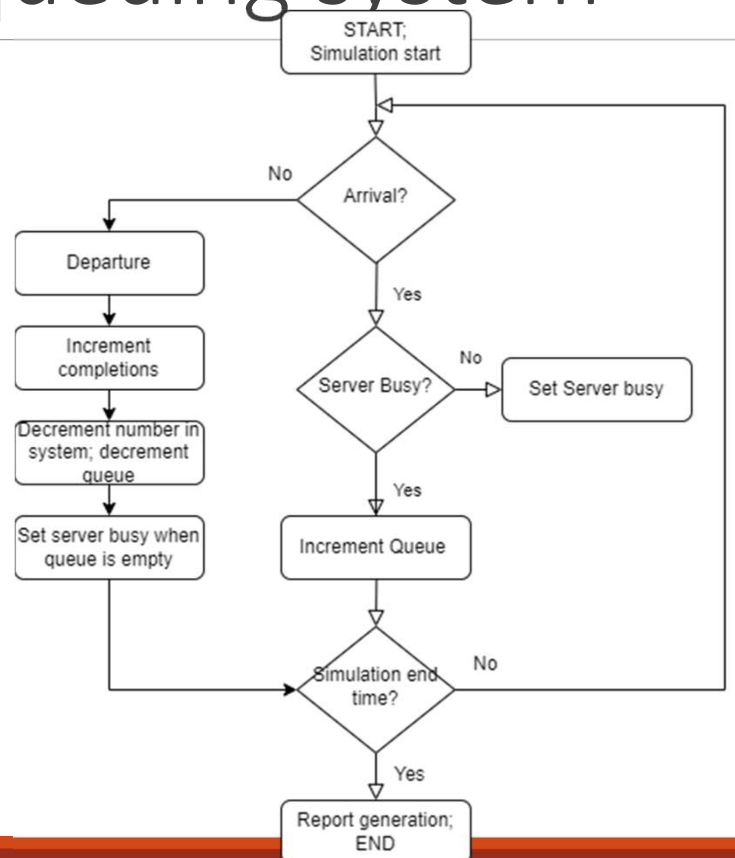
1. Input: Arrival time, service time, Queue max
2. Output: Server utilization, mean no of customers in queue, mean time in queue, etc
3. Start simulation clock
4. If arrival event? ⍰ Yes: Next step; No: Step 7
5. Check if number of customers in system is more than 1
6. Server is busy? Yes: Increment queue; No: Set server busy
7. Departure event
8. Schedule departure
9. Increment no of completions; Decrease queue
10. Simulation end time? No: loop back to Step 4, Yes: Step 11
11. Generate report
12. END

Simple M/M/1 queuing system

- m/m/1 example

Simple M/M/1 queuing system

M/M/1 queue Flowchart



Simple M/M/1 queuing system

- M/M/1 queue Analytical solution

- Server Utilization (ρ)
  - $\rho = \lambda/\mu$
- Mean number of customers in system (L)
  - $L = \rho /(1- \rho)$ | $L = \lambda W$
- Mean time delay in the system (W)
  - $W = \rho(1/ \mu) /(1- \rho)$ | $W = Wq + 1/\mu$
- Mean number in the queue (Lq)
  - $Lq = \rho2/(1- \rho)$
- Mean time in queue (Wq)
  - $Wq = Lq/\lambda$

Simple M/M/1 queuing system

- M/M/1 queue Analytical solution
- For: m = 1,2,3,4
- $\lambda = 1$; $\mu = 1$; ⍰ $\rho = 1.00$ or 100%; L = infinite; Lq = infinite; W = infinite ; Wq = infinite
- $\lambda = 1$; $\mu = 2$; ⍰ $\rho = 0.50$ or 50%; L = 1; Lq = 0.5; W = 1 ; Wq = 0.5
- $\lambda = 1$; $\mu = 3$; ⍰ $\rho = 0.33$ or 33%; L = 0.5; Lq = 0.1667; W = 0.5 ; Wq = 0.1667
- $\lambda = 1$; $\mu = 4$; ⍰ $\rho = 0.25$ or 25%; L = 0.3333; Lq = 0.0833; W = 0.3333 ; Wq = 0.0833
- $\lambda = 2$; $\mu = 2$; ⍰ $\rho = 1$ or 100%; L = infinite; Lq = infinite; W = infinite ; Wq = infinite
- $\lambda = 2$; $\mu = 4$; ⍰ $\rho = 0.5$ or 50%; L = 1; Lq = 0.5; W = 0.5 ; Wq = 0.25
- $\lambda = 2$; $\mu = 6$; ⍰ $\rho = 0.3333$ or 33.33%; L = 0.5; Lq = 0.1667; W = 0.25 ; Wq = 0.0833
- $\lambda = 2$; $\mu = 8$; ⍰ $\rho = 0.25$ or 25%; L = 0.3333; Lq = 0.0833; W = 0.1667 ; Wq = 0.0417

Simple M/M/1 queuing system

- M/M/1 queue Results

- We run the simulation for 1.0e9, so to remov transient conditions.
- As we observe, with the service rate increase the server utilization decreases. This is becau the server is idle when waiting for arrival of customers.
- The mean time in queue and mean customer in system also decrease with increase in servi rate.
- Our simulation results align with the expected analytical results which is a positive sign that our simulation shows positive correlation with expected real-world systems.

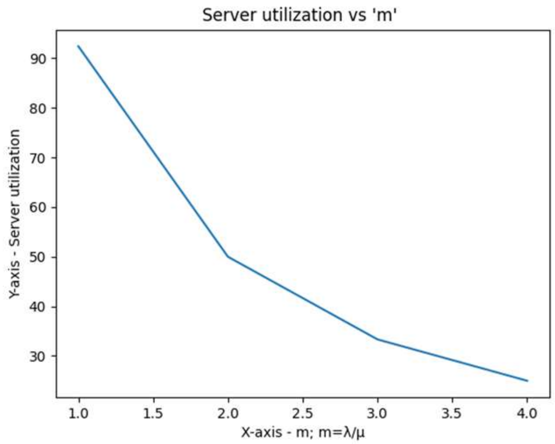
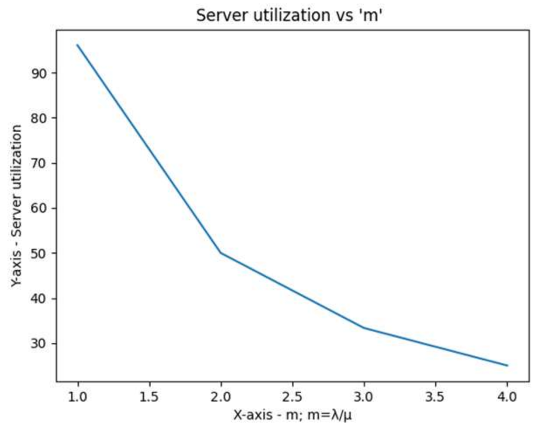


Figure: When arr and serv rate is one

Figure: When arr and serv rate is two



Variation of M/M/1 queues implemented

- 3 variations
  - 2 servers (M/M/k)
  - Max queue size
  - Max queue size in Kbytes

Variation of M/M/1 queues implemented

M/M/k 2 server

- M/M/k is a queue with 2 servers working in parallel

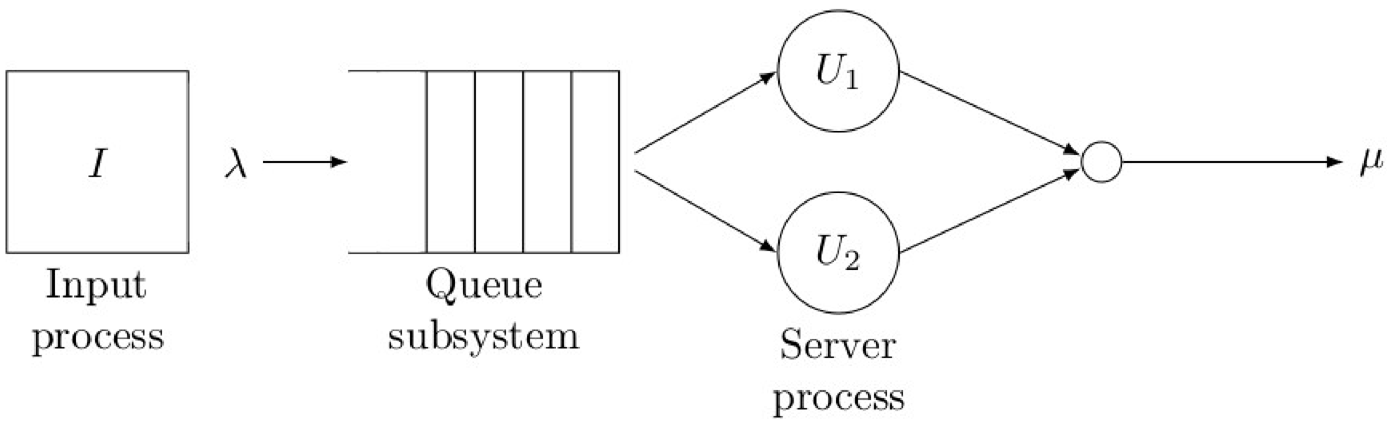
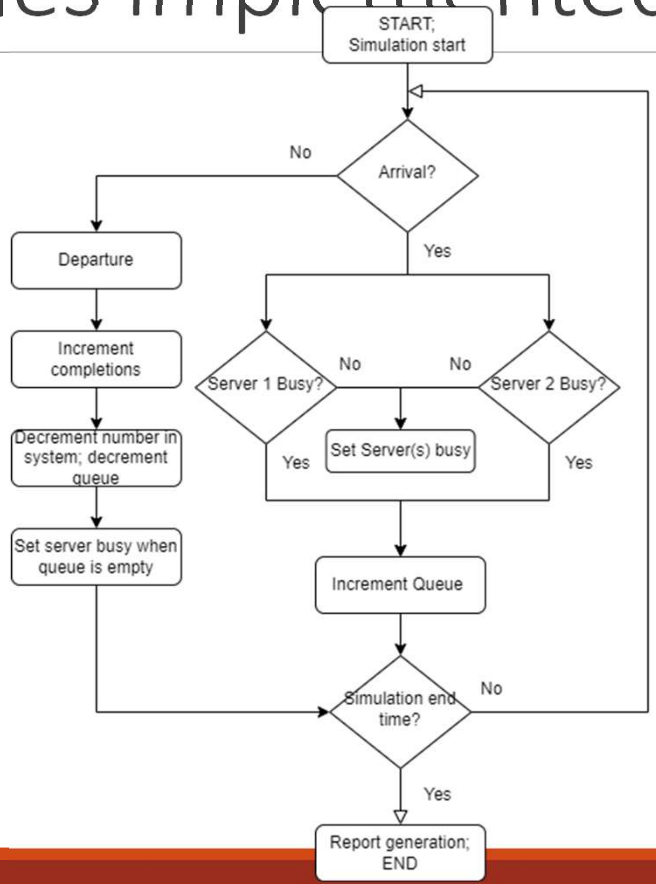


Figure: M/M/k queuing system

Variation of M/M/1 queues implemented

- M/M/k 2 server
  - M/M/k is a queue with 2 servers working in parallel

- Algorithm and flowchart
  1. Input: Arrival time, service time, Queue max
  2. Output: Server utilization, mean no of customers in queue, mean time in queue, etc
  3. Start simulation clock
  4. If arrival event? ⍰ Yes: Next step; No: Step 9
  5. Check if number of customers in system is more than 1
  6. If 2 Server is busy: ⍰ Yes: Next step; No: Set server(s) busy7. Increment queue
  8. Departure event
  9. Schedule departure
  10. Increment no of completions
  11. Simulation end time? No: loop back to Step 4, Yes: Step 13
  12. Generate report
  13. END

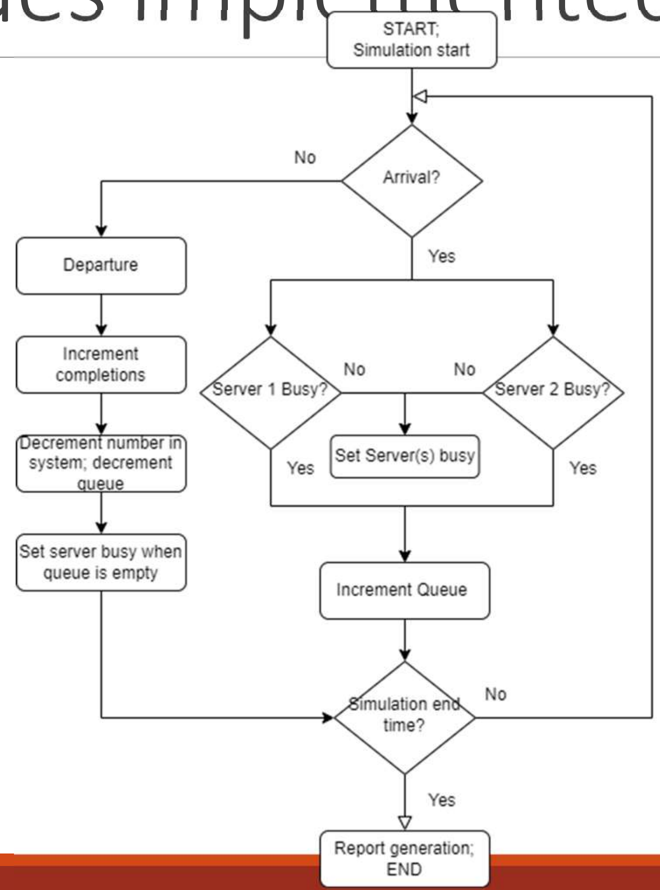


Variation of M/M/1 queues implemented

- M/M/k 2 server
  - M/M/k is a queue with 2 servers working in parallel

- Code snippets

```
if (n>2) // for 2 servers
 {
 qs = qs + ((n-2) * (time - lastEventTime));
 tq=tq+((n-2) * (time - lastEventTime));
 }
```



Variation of M/M/1 queues implemented

- M/M/k 2 server

- Analytical solutions
- Server Utilization (ρ)
  - $\rho = \lambda / c\mu$ where c is number of servers
- Mean number of customers in system (L) [3]
  - $L = \lambda 3/\mu(4\mu 2 – \lambda 2)$
- Mean time delay in the system (W) [3]
  - $W = \lambda 2/\mu(4\ \mu 2– \lambda 2)$

Variation of M/M/1 queues implemented

- M/M/k 2 server Analytical solutions
- λ = 1; μ = 1; ⍰ ρ = 0.5 or 50%; L = 1.3333; Lq = 0.3333; W = 1.3333; Wq = 0.3333
- λ = 1; μ = 2; ⍰ ρ = 0.25 or 25%; L = 0.5333 ; Lq = 0.0333; W = 0.5333 ; Wq = 0.0333
- λ = 1; μ = 3; ⍰ ρ = 0.1667 or 16.67%; L = 0.3429; Lq = 0.0095; W = 0.3429; Wq = 0.0095
- λ = 1; μ = 4; ⍰ ρ = 0.125 or 12.5%; L = 0.254; Lq = 0.004; W = 0.254; Wq = 0.004
- λ = 2; μ = 2; ⍰ ρ = 0.5 or 50%; L = 1.3333; Lq = 0.3333; W = 0.6667; Wq = 0.1667
- λ = 2; μ = 4; ⍰ ρ = 0.25 or 25%; L = 0.5333; Lq = 0.0333; W = 0.2667; Wq = 0.0167
- λ = 2; μ = 6; ⍰ ρ = 0.1667 or 16.67%; L = 0.3429; Lq = 0.0095; W = 0.1714; Wq = 0.0048
- λ = 2; μ = 8; ⍰ ρ = 0.125 or 12.5%; L = 0.254; Lq = 0.004; W = 0.127; Wq = 0.002

Variation of M/M/1 queues implemented

- M/M/k 2 server

- Results

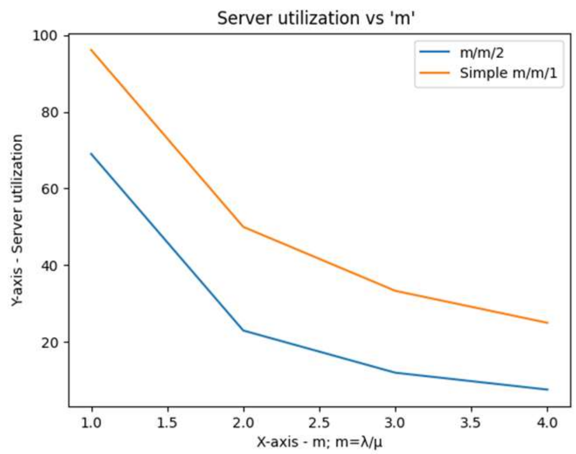


Figure: When arr and serv rate is one

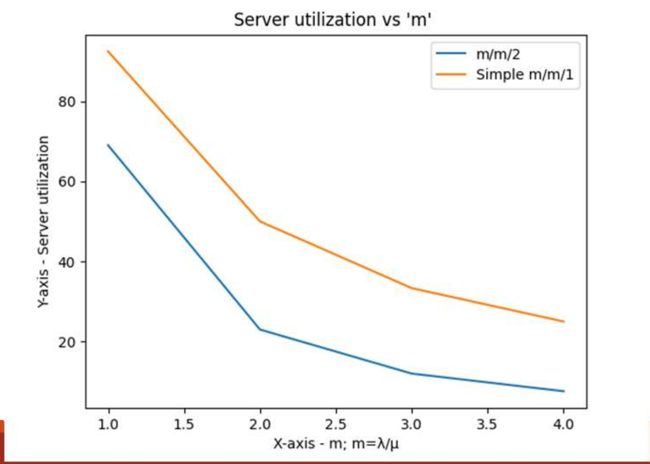


Figure: When arr and serv rate is one

Variation of M/M/1 queues implemented

- Comparison with m/m/1 queuing system
- As we observe, with the service rate increase, the server utilization decreases. This is because the server is idle when waiting for arrival of customers. The mean time in queue and mean customers in system also decrease with increase in service rate.
- When the server is idle, packets are serviced with less waiting time, therefore the mean time in queue also decreases.
- Our simulation results align with the expected analytical results which is a positive sign that our simulation shows positive correlation with expected real-world systems.
- Clearly, we see that due to 2 servers, we have better throughput and lesser mean time spent in queue, mean time in system is reduced when compared to simple m/m/1 queueing system.

Variation of M/M/1 queues implemented

- M/M/1 with Max queue size
  - M/M/1 with fixed, limited queue size
- Algorithm

Algorithm for mm1 with queue size

1. Input: Arrival time, service time, Queue max
2. Output: Server utilization, mean no of customers in queue, mean time in queue, etc
3. Start simulation clock
4. If arrival event? □ Yes: Next step; No: Step 9
5. Check if number of packets in system is more than 1
6. If Server is busy: □ Yes: Next step; No: Set server busy
7. If queue full: Yes: Drop packets to fit; No: Next step
8. Increment queue
9. Departure event
10. Schedule departure
11. Increment no of completions
12. Simulation end time? No: loop back to Step 4, Yes: Step
13. Generate report
14. END

Variation of M/M/1 queues implemented

- **M/M/1 with Max queue size**
  - M/M/1 with fixed, limited queue size
- Code snippets

if(n>1)

{

if(n <= QUEUE_MAX_USER+1) //total cust in system = cusomers in queue + 1 in service {

queue = (n-1); //number in queue = Number in system minus one in service

qs = qs + ((n-1) * (time - tn)); // Update area under "qs" curve

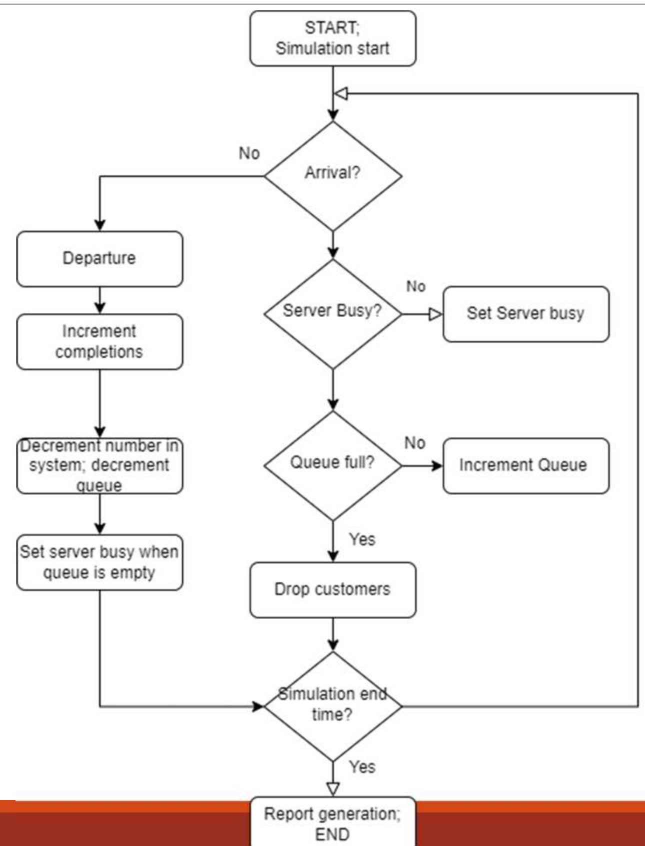
}

- else
- {
- drop = drop + (n - QUEUE_MAX_USER);
- n = QUEUE_MAX_USER+1; //drop excess customers
- }
- }

Variation of M/M/1 queues implemented

- M/M/1 with Max queue size
  - M/M/1 with fixed, limited queue size

- flowchart



Variation of M/M/1 queues implemented

- M/M/1 with Max queue size
- Results

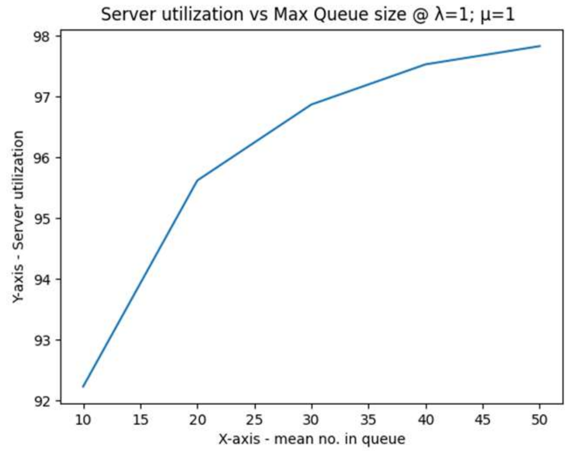


Figure: When arr and serv rate is one

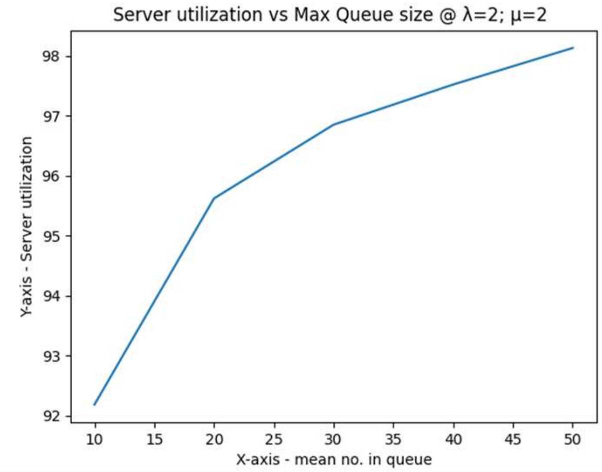


Figure: When arr and serv rate is one

Variation of M/M/1 queues implemented

- M/M/1 with Max queue size

- Results

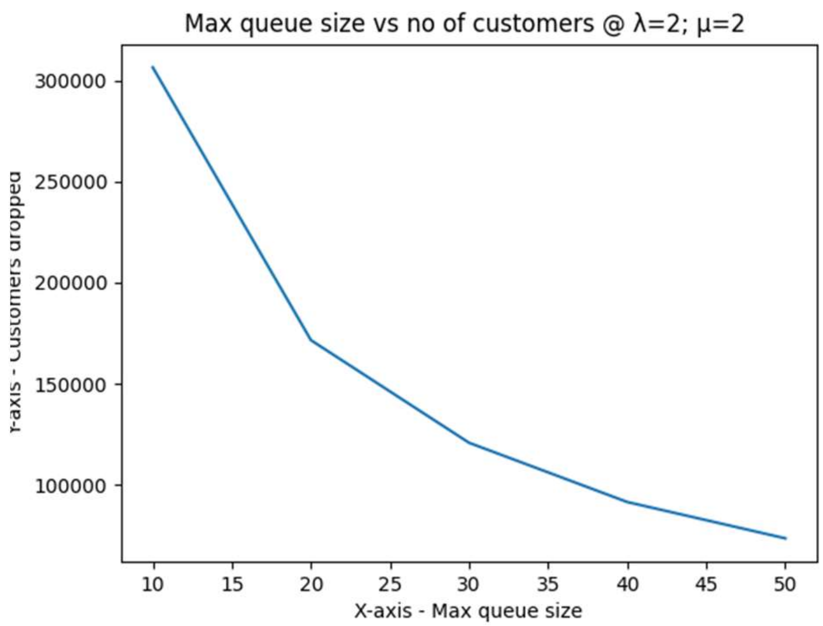


Figure: customers dropped when arr and serv rate is two

Variation of M/M/1 queues implemented

- Comparison with m/m/1 queuing system
- As service rate increases the server utilization decreases, because the server is idle when waiting for arrival.
- As queue size increase, eventually less packets are dropped. Packets dropped increases with increase in arrival rate, decrease with service rate.
- Server utilization increases with increase in queue size.
- Our simulation results align with the expected analytical results
- Even with restricted queue size, the performance and server utilization are slightly better than simple M/M/1 queueing system for identical service and arrival rates.

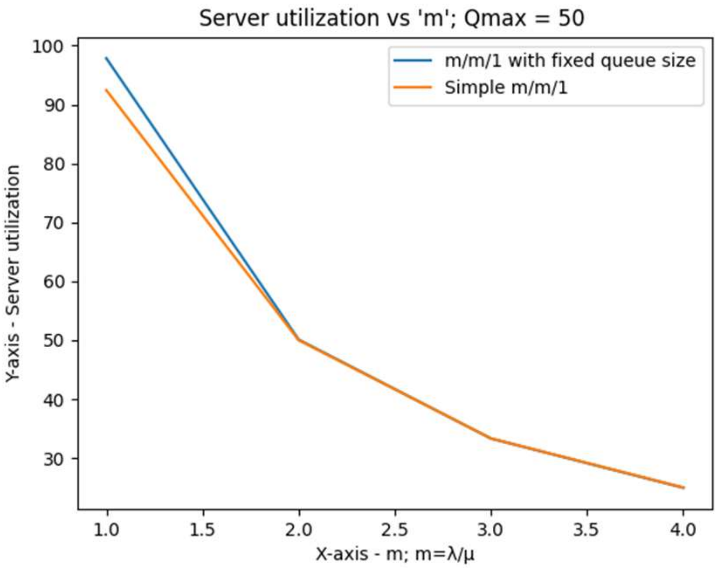


Figure: MM1 with fixed queue VS Simple MM1 queuing system

Variation of M/M/1 queues implemented

- **M/M/1 with Max queue size in Kbytes**
  - M/M/1 with fixed, limited queue size
- Algorithm

Algorithm

1. Input: Arrival time, service time, Queue max
2. Output: Server utilization, mean no of packets in queue, mean time in queue, etc
3. Start simulation clock
4. If arrival event? ⍰ Yes: Next step; No: Step 9
5. Check if number of packets in system is more than 1
6. If Server is busy: ⍰ Yes: Next step; No: Set server busy
7. If queue(in size Kb) full: Yes: Drop customers to fit; No: Next step 8. Increment queue
9. Departure event
10. Schedule departure
11. Increment no of completions
12. Simulation end time? No: loop back to Step 4, Yes: Step 13
13. Generate report
14. END

Variation of M/M/1 queues implemented

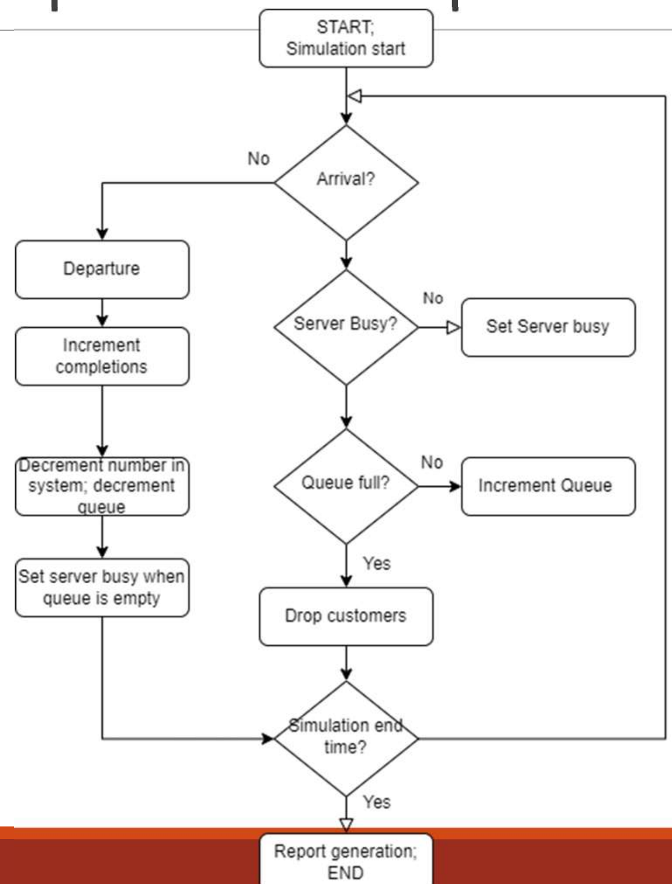
- M/M/1 with Max queue size in Kbytes
  - M/M/1 with fixed, limited queue size
- Code snippets

```
if(n > 1)
{
if(n <= (queue_size+1)) //total packets data in system = cusomers in queue + 1 in service
{qs = qs + (n-1) * (time - tn); // Update area under "qs" curve
} else
{ drop = drop + (n - queue_size); //data packets dropped
n = (queue_size+1); //drop excess packets
}
```

Variation of M/M/1 queues implemented

- M/M/1 with Max queue size in Kbytes
  - M/M/1 with fixed, limited queue size

- Flowchart



Variation of M/M/1 queues implemented

- Analytical solutions

Given data:

- ➔ Transmission rate R = 10kbps; Queue size [10,20,30,40,50] Kb
- ➔ Transmission rate R = 10 /8 Kbytes per sec = 1.25 Kbytes per sec

Lets say queue size [10,20,30,40,50] = 10 kb, and service rate = 1 packet/sec

➔ Size of 1 packet = service time * transmission rate R = 1 sec/pac * 1.25 kbytes /sec

➔ Size of 1 packet for given service time and transmission rate = 1.25 Kbytes/packet

When queue size is 10 Kb, and packet size is constant

Maximum queue size = queue size in kb / size of 1 packet

Max queue size = 10/1.25 = 8 packets

To validate, our analytical results we ran a M/M/1 simulation with fixed queue size in packets, where Queue size = 8 and compared our results with present case where queue size = 10kb. We received same results.

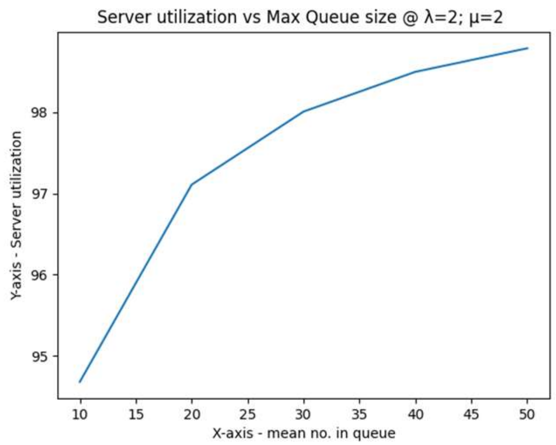
Variation of M/M/1 queues implemented

- M/M/1 with Max queue size in Kbytes

Figure: When arr and serv rate is one

- Results

Figure: When arr and serv rate is two



Variation of M/M/1 queues implemented

- M/M/1 with Max queue size in Kbytes
- Results

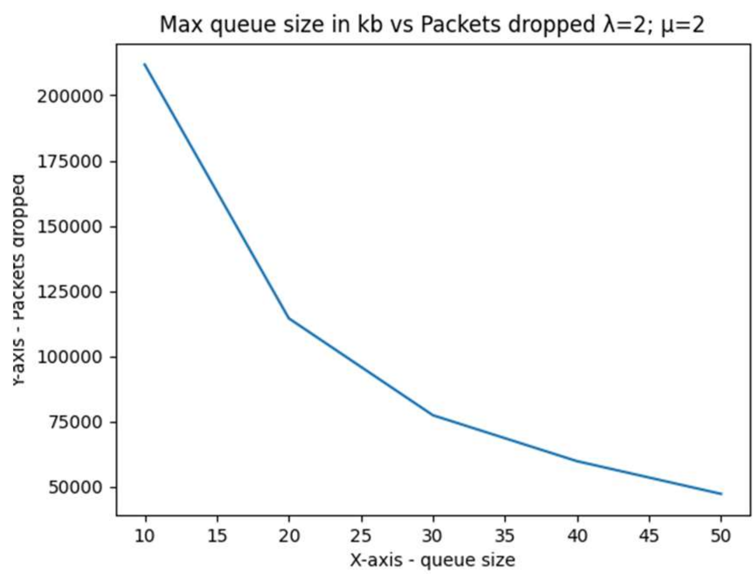


Figure: Packets dropped vs queue sizze

Variation of M/M/1 queues implemented

- Comparison with m/m/1 queuing system
- As service rate increases the server utilization decreases, because the server is idle when waiting for arrival.
- As queue size increase, eventually less packets are dropped. Packets dropped increases with increase in arrival rate, decrease with service rate.
- Server utilization increases with increase in queue size.
- When compared to simple M/M/1, Even with restricted queue size, the performance and server utilization are slightly better than simple M/M/1 queueing system for identical service and arrival rates.

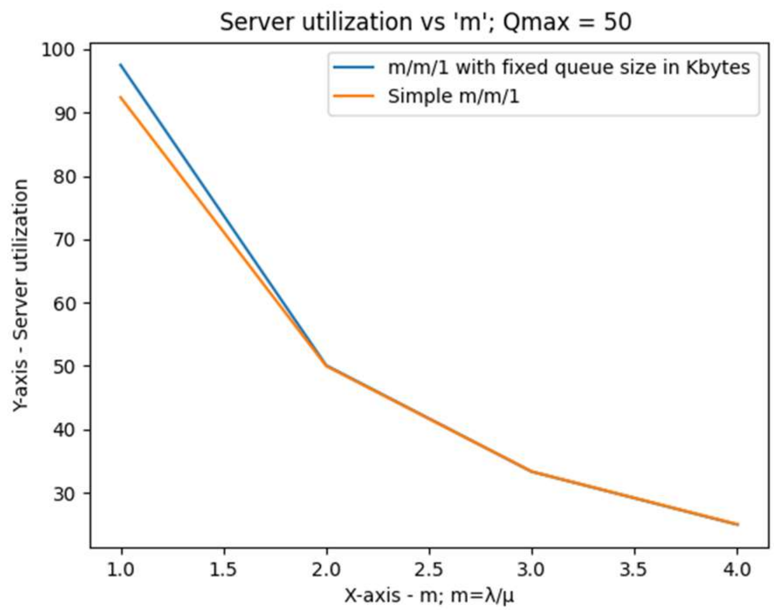


Figure: MM1 with fixed queue VS Simple MM1 queuing system

Thank You!