PRACTICAL FILE

MODELING AND SIMULATION LAB

(CS 603)

BE CSE 6^{TH} SEM

(GROUP-4)

University Institute of Engineering and Technology (UIET), Panjab University, Chandigarh, India- 160014

Under the guidance of

Priyanka Mam

Department of Computer Science and Engineering

Submitted By

Ojas Arora

Roll No: UE223073

Practical 4

Aim

Simulate queueing system at ATM using Poisson Distribution.

Introduction to ATM Queueing System using Poisson Distribution

An Automated Teller Machine (ATM) queueing system can be analysed using **Queueing Theory** and **Poisson Distribution**, which helps in understanding how customers arrive at the ATM and how long they take to get served.

Implementation of Queuing Systems

The implementation of a queuing system involves several key steps:

1. Generating Random Arrival and Service Times:

- The inter-arrival times are generated using an exponential distribution based on the Poisson process.
- Service times are also generated following an exponential distribution.

2. Computing Customer Arrival Times:

- The cumulative sum of inter-arrival times determines when each customer arrives at the system.

3. Determining Service Start and Departure Times:

- The service start time for each customer is determined by checking the availability of the server.
- If the server is busy, the customer has to wait until the previous customer departs. o Departure time is computed as the sum of service start time and the respective service duration.

4. Measuring Server Utilization:

- The total time the server is busy is recorded.
- The proportion of busy time to total elapsed time is calculated to determine server efficiency.

Code for Implementation of Simulating queueing system at ATM using Poisson Distribution

```
clc; clear; close all;
lambda_arrival = 1/1.5;
lambda_service = 1/2.0;
num_customers = 10;
inter_arrival_times = exprnd(1/lambda_arrival, num_customers, 1);
service_times = exprnd(1/lambda_service, num_customers, 1);
arrival_times = zeros(num_customers, 1);
service_start_times = zeros(num_customers, 1);
departure_times = zeros(num_customers, 1);
for i = 1:num_customers
    if i == 1
        arrival_times(i) = inter_arrival_times(i);
        service_start_times(i) = arrival_times(i);
    else
        arrival_times(i) = arrival_times(i-1) + inter_arrival_times(i);
        service_start_times(i) = max(arrival_times(i), departure_times(i-1));
    end
    departure_times(i) = service_start_times(i) + service_times(i);
end
fprintf('\n--------------------------------------------------------------------------------\n');
fprintf('| %-10s | %-10s | %-10s | %-10s | %-15s | %-10s |\n', ...
    'Customer', 'IAT', 'AT', 'ST', 'Start Service', 'Departure');
fprintf('--------------------------------------------------------------------------------\n');

for i = 1:num_customers
    fprintf('| %-10d | %-10.2f | %-10.2f | %-10.2f | %-15.2f | %-10.2f |\n', ...
        i, inter_arrival_times(i), arrival_times(i), service_times(i), ...
        service_start_times(i), departure_times(i));
end
fprintf('--------------------------------------------------------------------------------\n');
figure;
stairs(arrival_times, 1:num_customers, 'b', 'LineWidth', 2); hold on;
stairs(departure_times, 1:num_customers, 'r', 'LineWidth', 2);
xlabel('Time');
ylabel('Number of Customers');
title('Poisson Process: Arrival and Departure Times');
legend('Arrivals', 'Departures');
grid on;
queue_length = zeros(num_customers, 1);
for i = 1:num_customers
    queue_length(i) = sum(service_start_times(1:i) > arrival_times(i));
end
figure;
stairs(arrival_times, queue_length, 'g', 'LineWidth', 2);
hold on;
plot(departure_times, queue_length, 'ro', 'MarkerFaceColor', 'r');
xlabel('Time');
ylabel('Queue Length');
title('Poisson Queue Length Over Time');
legend('Queue Length at Arrival', 'Queue Cleared at Departure');
grid on;
```

Output

Customer	IAT	AT	ST	Start Service	Departure
1	0.31	0.31	3.70	0.31	4.00
2	0.15	0.46	0.06	4.00	4.06
3	3.10	3.55	0.09	4.06	4.15
4	0.14	3.69	1.45	4.15	5.60
5	0.69	4.37	0.45	5.60	6.04
6	3.49	7.87	3.91	7.87	11.77
7	1.92	9.78	1.73	11.77	13.50
8	0.91	10.69	0.18	13.50	13.67
9	0.07	10.75	0.47	13.67	14.14
10	0.05	10.81	0.08	14.14	14.22