M/M/1 Queue Theory – Simulations

1 System Description

The M/M/1 model is the simplest and most fundamental stochastic queuing model. Its name is derived from Kendall's notation, which describes the system's components:

Component	Interpretation
M (Markovian)	Exponentially distributed inter-arrival times (rate λ)
M (Markovian)	Exponentially distributed service times (rate μ)
1	A single server

Table 1: Kendall's Notation for M/M/1

2 Model Assumptions

The model relies on the following key assumptions:

1. **Arrivals:** Arrivals form a Poisson process with rate λ (arrivals per unit of time). Consequently, the mean time between arrivals is $1/\lambda$.
2. **Service Times:** Service times are independent and exponentially distributed with rate μ. The mean service time is $1/\mu$.
3. **Servers:** There is only one server.
4. **Service Discipline:** The order of service is First Come, First Served (FCFS).
5. **Queue Capacity:** The queue capacity is infinite (no customer is rejected/dropped).
6. **Stability:** The system reaches a steady state only if the arrival rate is less than the service rate ($\lambda < \mu$).

3 Traffic Intensity (Utilization Factor)

The traffic intensity, or utilization factor, is defined as:

$$\rho = \frac{\lambda}{\mu}$$

This expresses the percentage of time the server is busy.

- If $\rho < 1$, the system is stable.
- If $\rho \geq 1$, the number of customers increases indefinitely (the queue "explodes").

4 Probability Distribution

In a steady state, the probability P_n that there are exactly n customers in the system (including both those in the queue and the one being served) is given by:

$$P_n = (1-\rho)\rho^n, \quad \text{for } n = 0, 1, 2, \ldots$$

Specific cases:

- $P_0 = 1 - \rho$: The probability that the system is empty (idle).
- $P_1 = (1-\rho)\rho$: The probability that there is exactly 1 customer.

5 Average Performance Metrics

The following table summarizes the key performance metrics for an M/M/1 system:

Symbol	Interpretation	Formula
W	Mean time in system (waiting + service)	$W = \frac{1}{\mu-\lambda}$
W_q	Mean waiting time in queue	$W_q = \frac{\rho}{\mu(1-\rho)}$
L	Mean number of customers in system	$L = \frac{\rho}{1-\rho}$
L_q	Mean number of customers in queue	$L_q = \frac{\rho^2}{1-\rho}$

Table 2: Mean Performance Metrics

6 Little's Law

Little's Law is a fundamental relationship applicable to any stable queuing system, independent of the probability distribution (not just exponential):

$$L = \lambda W \quad \text{and} \quad L_q = \lambda W_q$$

In simple terms: The average number of customers in the system equals the arrival rate multiplied by the average time spent in the system.

Physical Interpretation

As the traffic intensity $\rho = \lambda/\mu$ increases, congestion grows exponentially. As $\rho \to 1$:

- The mean waiting time $W_q \to \infty$.
- The number of customers in the system increases drastically.

In simulation plots (e.g., W vs. ρ), this exponential growth is clearly visible as ρ approaches 1.

7 Simulation Details

7.1 Warm-up and Convergence

In simulations, we typically start with an empty system. Consequently, the first arrivals are served immediately, creating a **transient state** with artificially low values for W and L. To accurately estimate steady-state behavior, we employ a **warm-up period**: we ignore the statistics of the initial batch of customers (e.g., the first 5-10%).

7.2 Code Implementation Notes

Python (NumPy): The function `rng.exponential(scale)` expects the **scale** parameter, which corresponds to the mean value $(1/\lambda)$, not the rate. Additionally, `np.cumsum()` is used to calculate arrival timestamps from inter-arrival times.

C++ (Standard Library): The class `std::exponential_distribution(lambda)` accepts the **rate** (λ) directly, not the mean value.

7.3 Mapping Theory to Code

The following table maps theoretical symbols to the variable names used in the accompanying Python and C++ laboratory notebooks.

Metric	Symbol	Python Variable	C++ Variable
Arrival Rate	λ	`lam`	`lambda`
Service Rate	μ	`mu`	`mu`
Utilization	ρ	`rho`	`rho`
Simulation Results			
Mean Time in System	W	`W_hat, sim_W`	`W_sim`
Mean Time in Queue	W_q	`Wq_hat, sim_Wq`	`Wq_sim`
Mean Number in System	L	`L_hat, sim_L`	`L_sim`
Mean Number in Queue	L_q	`Lq_hat, sim_Lq`	`Lq_sim`
Theoretical Values			
Theoretical W	W	`th_W`	`W_th`
Theoretical W_q	W_q	`th_Wq`	`Wq_th`
Theoretical L	L	`th_L`	`L_th`
Theoretical L_q	L_q	`th_Lq`	`Lq_th`

Table 3: Variable Mapping

7.4 Connecting Theory and Simulation Logic

- **Calculating W:** In simulation, this is the average of `(depart[i] - arrivals[i])` for all customers.
- **Calculating W_q:** In simulation, this is the average of `(start_service[i] - arrivals[i])`.
- **Calculating L and L_q:** In the simulation code, these are often derived using Little's Law based on the simulated time averages: $L = \lambda \times W_{sim}$ and $L_q = \lambda \times Wq_{sim}$.