Problem 1: State Dependent Service Rates

System Specification:

- Arrival Rate: λ
- Service Rate Logic:
  - If 1 customer is in the system: rate = μ
  - If 2 customers are in the system: rate = 2μ
- System Capacity: 2 (3 possible states: 0, 1, 2)

State Definitions:

- State 0: Customers = 0 (System Empty). Service Rate = N/A.
- State 1: Customers = 1 (1 in service). Service Rate = μ.
- State 2: Customers = 2 (1 in service, 1 in queue). Service Rate = 2μ.

Balance Equations:

$$\begin{aligned}
\text{State 0:} \quad & \lambda P_0 = \mu P_1 \\
\text{State 1:} \quad & (\lambda + \mu) P_1 = \lambda P_0 + 2\mu P_2 \\
\text{State 2:} \quad & 2\mu P_2 = \lambda P_1
\end{aligned}$$

Constraint: $P_0 + P_1 + P_2 = 1$.

Solving for Probabilities: From State 0:

$$P_1 = \frac{\lambda}{\mu} P_0$$

From State 2:

$$2\mu P_2 = \lambda P_1 \implies P_2 = \frac{\lambda}{2\mu} P_1$$

Substitute P_1:

$$P_2 = \frac{\lambda}{2\mu} \left(\frac{\lambda}{\mu} P_0 \right) = \frac{\lambda^2}{2\mu^2} P_0$$

Using the normalization constraint:

$$\begin{gathered}
P_0 + P_1 + P_2 = 1 \\
P_0 + \frac{\lambda}{\mu} P_0 + \frac{\lambda^2}{2\mu^2} P_0 = 1 \\
P_0 \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2\mu^2} \right) = 1 \\
P_0 \left(\frac{2\mu^2 + 2\lambda\mu + \lambda^2}{2\mu^2} \right) = 1
\end{gathered}$$

Final Result:

$$P_0 = \frac{2\mu^2}{2\mu^2 + 2\lambda\mu + \lambda^2}$$

Problem 2: Finite Population Systems

Part a) M/M/1/2/2 System

Setup:

- 2 TAs (Teaching Assistants), 1 Workstation.
- State represents number of TAs at the workstation (using or waiting).

State Transitions:

- State 0 $\rightarrow$ 1: 2 TAs outside, rate 2λ.
- State 1 $\rightarrow$ 2: 1 TA outside, rate λ.
- Service Rate: μ (1 workstation).

Balance Equations:

$$\begin{aligned} &\text{State 0:} \quad 2\lambda P_0 = \mu P_1 \\ &\text{State 1:} \quad (\mu + \lambda) P_1 = 2\lambda P_0 + \mu P_2 \\ &\text{State 2:} \quad \mu P_2 = \lambda P_1 \end{aligned}$$

Solving:

$$P_1 = \frac{2\lambda}{\mu} P_0$$

$$P_2 = \frac{\lambda}{\mu} P_1 = \frac{\lambda}{\mu}\left(\frac{2\lambda}{\mu} P_0\right) = \frac{2\lambda^2}{\mu^2} P_0$$

Normalization:

$$P_0 + P_1 + P_2 = 1$$

$$P_0 \left(1 + \frac{2\lambda}{\mu} + \frac{2\lambda^2}{\mu^2}\right) = 1$$

$$P_0 = \frac{1}{1 + \frac{2\lambda}{\mu} + \frac{2\lambda^2}{\mu^2}}$$

Probability of Idle: Since the workstation is only idle at State 0:

$$P(\text{idle}) = P_0 = \frac{1}{1 + \frac{2\lambda}{\mu} + \frac{2\lambda^2}{\mu^2}}$$

Part b) M/M/2/4/4 System

Setup:

- Population $N = 4$ TAs.
- Servers $c = 2$ (2 Workstations).
- Service rates: μ if 1 user, 2μ if ≥ 2 users.

Rates:

- Arrivals: $4\lambda \to 3\lambda \to 2\lambda \to \lambda$.
- Service: μ (from state 1), 2μ (from states 2, 3, 4).

Balance Equations:

$$
\begin{aligned}
4\lambda P_0 &= \mu P_1 \\
(3\lambda + \mu)P_1 &= 4\lambda P_0 + 2\mu P_2 \\
(2\lambda + 2\mu)P_2 &= 3\lambda P_1 + 2\mu P_3 \\
(\lambda + 2\mu)P_3 &= 2\lambda P_2 + 2\mu P_4 \\
2\mu P_4 &= \lambda P_3
\end{aligned}
$$

Recursive Solution: 1. $P_1 = \frac{4\lambda}{\mu} P_0$
2. Solving for P_2:

$$3\lambda P_1 - 2\mu P_2 = 4\lambda P_0 - \mu P_1$$

(Using cut equations simplifies this to Rate In = Rate Out across boundaries):

$$2\mu P_2 = 3\lambda P_1 \implies P_2 = \frac{3\lambda}{2\mu} P_1 = \frac{3\lambda}{2\mu}\left(\frac{4\lambda}{\mu} P_0\right) = \frac{6\lambda^2}{\mu^2} P_0$$

3. Solving for P_3:

$$2\mu P_3 = 2\lambda P_2 \implies P_3 = \frac{\lambda}{\mu} P_2 = \frac{\lambda}{\mu}\left(\frac{6\lambda^2}{\mu^2} P_0\right) = \frac{6\lambda^3}{\mu^3} P_0$$

4. Solving for P_4:

$$2\mu P_4 = \lambda P_3 \implies P_4 = \frac{\lambda}{2\mu} P_3 = \frac{\lambda}{2\mu}\left(\frac{6\lambda^3}{\mu^3} P_0\right) = \frac{3\lambda^4}{\mu^4} P_0$$

Normalization:

$$P_0 \left(1 + \frac{4\lambda}{\mu} + \frac{6\lambda^2}{\mu^2} + \frac{6\lambda^3}{\mu^3} + \frac{3\lambda^4}{\mu^4}\right) = 1$$

Workstation Idle Probability: State 0: Both workstations idle (P_0). State 1: Only 1 TA, so 1 workstation idle (P_1). Total Idle Fraction (Probability a specific server is idle):

$$P(\text{idle}) = P_0 + \frac{1}{2} P_1$$

Substituting $P_1 = \frac{4\lambda}{\mu} P_0$:

$$P(\text{idle}) = P_0 + \frac{1}{2}\left(\frac{4\lambda}{\mu} P_0\right) = P_0 \left(1 + \frac{2\lambda}{\mu}\right)$$

Part c) Single Fast Server (2μ)

Setup:

- 1 Workstation, but speed is 2μ.
- Population $N = 4$.

Rates:

- Arrivals: $4\lambda, 3\lambda, 2\lambda, \lambda$.
- Service: 2μ for all states $1 \to 4$ (unlike Part b where state 1 was μ).

Balance Equations (Cut Method):

$$
\begin{aligned}
P_1 &= \frac{4\lambda}{2\mu} P_0 = \frac{2\lambda}{\mu} P_0 \\
P_2 &= \frac{3\lambda}{2\mu} P_1 = \frac{3\lambda}{2\mu}\left(\frac{2\lambda}{\mu} P_0\right) = \frac{3\lambda^2}{\mu^2} P_0 \\
P_3 &= \frac{2\lambda}{2\mu} P_2 = \frac{\lambda}{\mu}\left(\frac{3\lambda^2}{\mu^2} P_0\right) = \frac{3\lambda^3}{\mu^3} P_0 \\
P_4 &= \frac{\lambda}{2\mu} P_3 = \frac{\lambda}{2\mu}\left(\frac{3\lambda^3}{\mu^3} P_0\right) = \frac{3\lambda^4}{2\mu^4} P_0
\end{aligned}
$$

Normalization:

$$
P_0 = \frac{1}{1 + \frac{2\lambda}{\mu} + \frac{3\lambda^2}{\mu^2} + \frac{3\lambda^3}{\mu^3} + \frac{3\lambda^4}{2\mu^4}}
$$

Probability of Idle: As there is only 1 workstation, the system is idle only at State 0.

$$
P(\text{idle}) = P_0
$$

Problem 3: Mean Value Analysis (MVA) Simulation

a. Simulation Code

The following Java implementation performs the Mean Value Analysis for $N = 5$ customers across $K = 7$ devices.

```java
public class mva {
    private static final int K = 7;
    private static final int N = 5;
    private static final int[] S = {20, 5, 15, 10, 10, 15, 20};
    private static final double[] V = {1, 1, 0.6, 0.4, 0.4, 0.3, 0.3};

    public static void main(String[] args) {
        MVAMetrics metrics = performMVA(N);
        printResults(metrics);
```

```java
    }

    private static MVAMetrics performMVA(int maxCustomers) {
        double[][] responseTimes = new double[K][maxCustomers + 1];
        double[] throughput = new double[maxCustomers + 1];
        double[][] queueLengths = new double[K][maxCustomers + 1];

        for (int n = 1; n <= maxCustomers; n++) {
            double totalResponse = 0.0;
            for (int i = 0; i < K; i++) {
                responseTimes[i][n] = S[i] * (1 + queueLengths[i][n - 1]);
                totalResponse += V[i] * responseTimes[i][n];
            }
            throughput[n] = (double) n / totalResponse;
            for (int i = 0; i < K; i++) {
                queueLengths[i][n] = throughput[n] * V[i] * responseTimes[i][n];
            }
        }
        return new MVAMetrics(throughput, queueLengths);
    }

    private static void printResults(MVAMetrics metrics) {
        System.out.printf("System Throughput for N=%d: %.4f customers/sec%n",  N, metrics.throughput()[N]);
        for (int i = 0; i < K; i++) {
            System.out.printf("Device %d Queue Length for N=%d: %.4f customers%n", i, N, metrics.queueLengths()[i][N]);
        }
        System.out.println("\nThroughput per device:");
        for (int i = 0; i < K; i++) {
            double deviceThroughput = V[i] * metrics.throughput()[N];
            System.out.printf("Device %d Throughput: %.4f customers/sec%n", i, deviceThroughput);
        }
    }

    private record MVAMetrics(double[] throughput, double[][] queueLengths) {}
}
```

b. Simulation Results

The code above generates the following performance metrics for the system:

```
System Throughput for N=5: 0.0467 customers/sec
Device 0 Queue Length for N=5: 2.9526 customers
Device 1 Queue Length for N=5: 0.2966 customers
Device 2 Queue Length for N=5: 0.6644 customers
Device 3 Queue Length for N=5: 0.2255 customers
Device 4 Queue Length for N=5: 0.2255 customers
Device 5 Queue Length for N=5: 0.2602 customers
Device 6 Queue Length for N=5: 0.3751 customers

Throughput per device:
Device 0 Throughput: 0.0467 customers/sec
Device 1 Throughput: 0.0467 customers/sec
Device 2 Throughput: 0.0280 customers/sec
```

```
Device 3 Throughput: 0.0187 customers/sec
Device 4 Throughput: 0.0187 customers/sec
Device 5 Throughput: 0.0140 customers/sec
Device 6 Throughput: 0.0140 customers/sec
```