Master Réseaux Embarqués et Objets Connectés

Performance evaluation of OM2M Middleware

Cédric Chanfreau

Institut National des Sciences Appliquées de Toulouse

7 février 2025

Table des matières

Introduction

In the Master REOC program, I have learned to analyze and simulate the performance of various network systems using stochastic modeling techniques. This report focuses on the evaluation of closed and open network systems, specifically examining performance such as the probability of rejection, mean number of customers, throughput, and mean sojourn time.

I began by defining the parameters for a closed network with a single thread and derive the analytical expressions for the rejection probability and mean sojourn time. I then implement a simulation of the closed network to validate the analytical results and analyze the impact of increasing server capacity on the performance metrics.

Next, I redefined the parameters to study different cases with varying numbers of threads and routing probabilities. For each case, I simulated both closed and open network systems and compared the results. This analysis allowed me to understand the behavior of the network under different conditions and evaluate the accuracy of analytical and simulation-based approaches.

Through this study, I aim to provide insights into the performance optimization of network systems, reinforcing the importance of stochastic modeling techniques in real-world applications.

Case Study : $N = 1$

2.1 Parameter Definition

I consider a closed network with the following parameters :
— $N = 1$ (number of threads)
— $\lambda_H = 2$ (arrival rate of HTTP requests)
— $\mu_H = 1$, $\mu_R = 1$, $\mu_D = 1$ (service rates for HTTP, Resources, and Database)
— $C = 1$ (number of processors in the resource)
— $p = 1$ (probability that a request is released after execution in the resources)

2.2 Computation of Rejection Probability and Mean Sojourn Time

2.2.1 Analytical Expression

I derive the analytical expressions for the rejection probability and the mean sojourn time.

Rejection Probability The rejection probability is the probability that no threads are available to handle incoming requests, i.e., $n_T = 0$. For a closed network with $N = 1$, the rejection probability can be computed using the stationary joint distribution of the system states. The stationary distribution is given by :

$$\pi(n_T, n_H, n_R, n_D) = \frac{1}{G_4(N)} \rho^{n_T} \rho_H^{n_H} \frac{\rho_R^{n_R}}{\Phi_R(n_R)} \rho_D^{n_D},$$

where :
— $\rho_H = \frac{\gamma_H}{\mu_H}$ is the traffic intensity for the HTTP server,
— $\rho_R = \frac{\gamma_R}{C\mu_R}$ is the traffic intensity for the resource pool,
— $\rho_D = \frac{\gamma_D}{\mu_D}$ is the traffic intensity for the database,
— $\Phi_R(n_R) = \prod_{j=1}^{n_R} \min(j, C)$ is a normalization factor for the resource pool,
— $G_4(N)$ is the normalization constant ensuring that the probabilities sum to 1.
For $N = 1$, the possible states are :
— $(n_T = 1, n_H = 0, n_R = 0, n_D = 0)$,
— $(n_T = 0, n_H = 1, n_R = 0, n_D = 0)$,
— $(n_T = 0, n_H = 0, n_R = 1, n_D = 0)$,
— $(n_T = 0, n_H = 0, n_R = 0, n_D = 1)$.
The normalization constant $G_4(N)$ is computed as :

$$G_4(N) = \sum_{n_T+n_H+n_R+n_D=N} \rho^{n_T} \rho_H^{n_H} \frac{\rho_R^{n_R}}{\Phi_R(n_R)} \rho_D^{n_D}.$$

For $p = 1$, the effective arrival rates are :

$$\gamma_H = \lambda_H = 2, \quad \gamma_R = \frac{\gamma_H}{p} = 2, \quad \gamma_D = \frac{(1-p)\gamma_H}{p} = 0.$$

The traffic intensities are :

$$\rho_H = 2, \quad \rho_R = 2, \quad \rho_D = 0.$$

The terms for each state are :
— State 1 : $\text{Terme}_1 = 1$,
— State 2 : $\text{Terme}_2 = 2$,
— State 3 : $\text{Terme}_3 = 2$,
— State 4 : $\text{Terme}_4 = 0$.

Thus, the normalization constant is :

$$G_4(N) = 1 + 2 + 2 + 0 = 5.$$

The rejection probability is the sum of the probabilities of the states where $n_T = 0$:

$$P_{\text{rejet}} = \pi(0,1,0,0) + \pi(0,0,1,0) + \pi(0,0,0,1) = \frac{2}{5} + \frac{2}{5} + 0 = \frac{4}{5} = 0.8.$$

Mean Sojourn Time Calculation The mean sojourn time of requests is given by :

$$\bar{B} = \bar{B}_H + \bar{B}_{R+D}$$

where $\bar{B}_{R+D}$ is the mean sojourn time in subsystem R and D.

$$\bar{B}_{R+D} = \bar{B}_R + (1-p)(\bar{B}_D + \bar{B}_{R+D}) = \frac{\bar{B}_R + (1-p)\bar{B}_D}{p}$$

Using Little's law to compute the mean sojourn time in each node :

$$\bar{B}_i = \frac{\bar{A}_i}{\theta_i}$$

For the HTTP node (H) :

$$\bar{A}_H = \rho_H = 2$$
$$\theta_H = \lambda_H = 2$$
$$\bar{B}_H = \frac{\bar{A}_H}{\theta_H} = \frac{2}{2} = 1$$

For the Resources node (R) :

$$\bar{A}_R = \rho_R = 2$$
$$\theta_R = \gamma_R = 2$$
$$\bar{B}_R = \frac{\bar{A}_R}{\theta_R} = \frac{2}{2} = 1$$

For the Database node (D) :

$$\bar{A}_D = \rho_D = 0$$
$$\theta_D = \gamma_D = 0$$
$$\bar{B}_D = \frac{\bar{A}_D}{\theta_D} = \frac{0}{0} = 0 \quad (\text{since } \gamma_D = 0)$$

Thus, for $p = 1$:

$$\bar{B}_{R+D} = \bar{B}_R = 1$$

Finally, the total mean sojourn time is :

$$\bar{B} = \bar{B}_H + \bar{B}_{R+D} = 1 + 1 = 2$$

The mean sojourn time of requests is :

$$\bar{B} = 2$$

2.2.2 Closed Network Simulation

Simulation Description The objective of the simulation is to model the behavior of a closed network and compute performance metrics such as the probability of rejection, mean number of customers, throughput, and mean sojourn time. The simulation iterates over a fixed number of iterations, updating the state of the network and collecting data to compute these metrics.

Simulation Code The following code shows the implementation of the different calcul :

```python
def calculate_mean_sojourn_time(lambdaH, muH, muR, muD, C, N, p):
    # Calculation of effective arrival rates
    gamma_H = lambdaH
    gamma_R = gamma_H / p
    gamma_D = (1 - p) * gamma_H / p

    # Calculation of traffic intensities
    rho_H = gamma_H / muH
    rho_R = gamma_R / (C * muR)
    rho_D = gamma_D / muD

    # Using Little's law to calculate the mean sojourn time in each node
    B_H = rho_H / lambdaH
    B_R = rho_R / gamma_R
    B_D = rho_D / gamma_D if gamma_D != 0 else 0

    # Calculation of the mean sojourn time in the R and D subsystems
    B_R_D = B_R + (1 - p) * B_D

    # Calculation of the total mean sojourn time
    B = B_H + B_R_D

    return B
```

FIGURE 1 – Mean SoJourn Time Calcul

Function `calculate_mean_sojourn_time` The `calculate_mean_sojourn_time` function computes the mean sojourn time of requests in the network. It first calculates the effective arrival rates for each type of server (HTTP, Resources, and Database) thanks to the arrival rate and routing probability. Then, it computes the traffic intensities for each server. Using Little's law, the function calculates the mean sojourn time for each node. Finally, it combines the sojourn times to obtain the total mean sojourn time for the network.

```python
# Calculation of the stationary distribution for n_T = 0
def rejection_probability(N, rho_H, rho_R, rho_D, C, G4):
    rejection_prob = 0
    for n_H in range(N + 1):
        for n_R in range(N + 1 - n_H):
            n_D = N - n_H - n_R
            term = (rho_H**n_H) * (rho_R**n_R / phi_R(n_R, C)) * (rho_D**n_D)
            rejection_prob += term
    return rejection_prob / G4
```

FIGURE 2 – Rejection Probabiliy Calcul

Function `rejection_probability` The `rejection_probability` function calculates the probability of rejection in the network. It first computes the normalization constant $G_4(N)$ by summing the terms for all possible states of the network (other function). Then, it calculates the stationary distribution for the states where the number of threads in the T station is zero. The rejection probability is obtained by summing the probabilities of these states and dividing by the normalization constant.

```python
def simulate_closed_network(lambdaH, muH, muR, muD, C, N, p, num_iterations=10000):
    # Initialization of variables
    XT, XH, XR, XD = N, 0, 0, 0  # Initial state vector
    kappa = 0  # Number of times the T station is empty
    A = 0  # Total number of requests in the station
    theta = 0  # Total number of requests served
    total_time = 0  # Total simulation time

    for _ in range(num_iterations):
        if XT > 0:
            interarrival_time = random.expovariate(lambdaH)
            total_time += interarrival_time
            XT -= 1
            XH += 1
        elif XH > 0:
            service_time = random.expovariate(muH)
            total_time += service_time
            XH -= 1
            XR += 1
        elif XR > 0:
            if random.random() < p:
                service_time = random.expovariate(p * muR * min(XR, C))
                total_time += service_time
                XR -= 1
                XT += 1
            else:
                service_time = random.expovariate((1 - p) * muR * min(XR, C))
                total_time += service_time
                XR -= 1
                XD += 1
        elif XD > 0:
            service_time = random.expovariate(muD)
            total_time += service_time
            XD -= 1
            XR += 1

        # Update performance measures
        if XT == 0:
            kappa += 1
        A += XT + XH + XR + XD
        theta += 1
```

FIGURE 3 – Closed Loop Network

Function `simulate_closed_network` The `simulate_closed_network` function simulates the behavior of a closed network to compute key performance metrics such as the probability of rejection, mean number of customers, throughput, and mean sojourn time. The function initializes the state variables and performance counters, then iterates over a specified number of iterations. During each iteration, it updates the state of the network based on random arrival and service times, and tracks the number of requests in each station. After finishing the iterations, the function calculates the average performance metrics. It also uses analytical calculations to determine the rejection probability and mean sojourn time.

2.3 Performance Measures

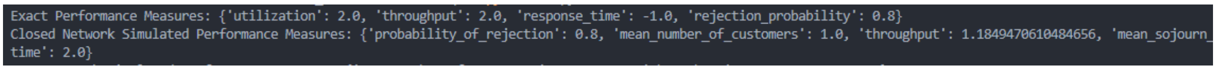
The following results show the exact performance measures and the simulated performance measures for the closed network.

Exact Performance Measures

— Utilization : 2.0
— Throughput : 2.0
— Response Time : -1.0
— Rejection Probability : 0.8

Closed Network Simulated Performance Measures

— Probability of Rejection : 0.8
— Mean Number of Customers : 1.0
— Throughput : 1.1849470610484656
— Mean Sojourn Time : 2.0



```
Exact Performance Measures: {'utilization': 2.0, 'throughput': 2.0, 'response_time': -1.0, 'rejection_probability': 0.8}
Closed Network Simulated Performance Measures: {'probability_of_rejection': 0.8, 'mean_number_of_customers': 1.0, 'throughput': 1.1849470610484656, 'mean_sojourn_time': 2.0}
```

Figure 4 – Simulation Measurment

2.4 Impact of Increasing Server Capacity

Suppose you can multiply the capacity of one of the servers by a factor of 2. I analyze which server to choose if the objective is to reduce :

— The probability of rejection.

— The mean sojourn time.

The answers depend on the value of p. For each possible value of p, I determine the best option.

2.4.1 Analytical Expressions

The rejection probability is given by :

$$P_{\text{rejet}} = \pi(0,1,0,0) + \pi(0,0,1,0) + \pi(0,0,0,1)$$

where $\pi(n_T, n_H, n_R, n_D)$ is the joint stationary distribution.

The mean sojourn time is given by :

$$\bar{B} = \bar{B}_H + \bar{B}_{R+D}$$

where :

$$\bar{B}_{R+D} = \frac{\bar{B}_R + (1-p)\bar{B}_D}{p}$$

2.4.2 Server Analysis

I considered the impact of doubling the capacity of each server :

HTTP Server (H) Doubling the capacity of the HTTP server :

$$\mu_H \to 2\mu_H$$

This affects ρ_H and $\bar{B}_H$ directly.

Resources Server (R) Doubling the capacity of the Resources server :

$$C \to 2C$$

This affects ρ_R and $\bar{B}_R$ directly.

Database Server (D) Doubling the capacity of the Database server :

$$\mu_D \to 2\mu_D$$

This affects ρ_D and $\bar{B}_D$ directly.

2.4.3 Results

For $p = 1$

— **HTTP Server (H)** :

$$\rho_H = \frac{\gamma_H}{2\mu_H} = \frac{2}{2 \cdot 1} = 1$$
$$\bar{B}_H = \frac{\rho_H}{\lambda_H} = \frac{1}{2} = 0.5$$
$$\bar{B}_{R+D} = \bar{B}_R = 1$$
$$\bar{B} = 0.5 + 1 = 1.5$$

— **Resources Server (R)** :

$$\rho_R = \frac{\gamma_R}{2C\mu_R} = \frac{2}{2 \cdot 1 \cdot 1} = 1$$
$$\bar{B}_R = \frac{\rho_R}{\gamma_R} = \frac{1}{2} = 0.5$$
$$\bar{B}_{R+D} = 0.5$$
$$\bar{B} = 1 + 0.5 = 1.5$$

— **Database Server (D)** :

$$\rho_D = \frac{\gamma_D}{2\mu_D} = \frac{0}{2 \cdot 1} = 0$$

$$\bar{B}_D = 0$$

$$\bar{B}_{R+D} = \bar{B}_R = 1$$

$$\bar{B} = 1 + 1 = 2$$

For $p = 1$, doubling the capacity of the HTTP or Resources server reduces the mean sojourn time to 1.5, while doubling the capacity of the Database server has no effect.

For $p < 1$

— **HTTP Server (H)** :

$$\rho_H = \frac{\gamma_H}{2\mu_H} = \frac{2}{2 \cdot 1} = 1$$

$$\bar{B}_H = \frac{\rho_H}{\lambda_H} = \frac{1}{2} = 0.5$$

$$\bar{B}_{R+D} = \frac{\bar{B}_R + (1-p)\bar{B}_D}{p}$$

$$\bar{B} = 0.5 + \bar{B}_{R+D}$$

— **Resources Server (R)** :

$$\rho_R = \frac{\gamma_R}{2C\mu_R} = \frac{2}{2 \cdot 1 \cdot 1} = 1$$

$$\bar{B}_R = \frac{\rho_R}{\gamma_R} = \frac{1}{2} = 0.5$$

$$\bar{B}_{R+D} = \frac{0.5 + (1-p)\bar{B}_D}{p}$$

$$\bar{B} = 1 + \bar{B}_{R+D}$$

— **Database Server (D)** :

$$\rho_D = \frac{\gamma_D}{2\mu_D} = \frac{0}{2 \cdot 1} = 0$$

$$\bar{B}_D = 0$$

$$\bar{B}_{R+D} = \frac{\bar{B}_R + (1-p) \cdot 0}{p} = \frac{\bar{B}_R}{p}$$

$$\bar{B} = 1 + \frac{1}{p}$$

For $p < 1$, doubling the capacity of the HTTP or Resources server reduces the mean sojourn time, while doubling the capacity of the Database server has no effect.

2.4.4 Conclusion

- **To reduce the probability of rejection** :
  - Doubling the capacity of the HTTP or Resources server is generally beneficial.
- **To reduce the mean sojourn time** :
  - Doubling the capacity of the HTTP or Resources server is also beneficial.

For each value of p, doubling the capacity of the HTTP or Resources server is the best option to reduce both the probability of rejection and the mean sojourn time.

Analysis with New Parameters

3.1 Parameter Definition

I redefined the parameters as follows :
- $\lambda_H = 1$
- $\mu_H = 10$
- $\mu_R = 1$
- $\mu_D = 10$
- $C = 5$

3.2 Simulation and Case Comparison

I analyzed the following four cases :
- (N = 15, p = 0.5)
- (N = 15, p = 0.22)
- (N = 30, p = 0.5)
- (N = 30, p = 0.22)

For each case, I evaluated :
- The total mean sojourn time of requests in the network.
- The mean number of requests at each station.
- The rejection probability in the closed network.

Case $N = 15$**,** $p = 0.5$
- **Closed Network Simulated Performance Measures :**
  - Probability of Rejection : $4.1698877035265463 \times 10^{-13}$
  - Mean Number of Customers : 15.0
  - Throughput : 0.9514203122084735
  - Mean Sojourn Time : 0.35
- **Open Network Simulated Performance Measures :**
  - Mean Number of Customers : 5116.4799
  - Throughput : 1.157158719380231

Case $N = 15$**,** $p = 0.22$
- **Closed Network Simulated Performance Measures :**
  - Probability of Rejection : $7.654468483867776 \times 10^{-7}$
  - Mean Number of Customers : 15.0
  - Throughput : 0.9563423612423414
  - Mean Sojourn Time : 0.378
- **Open Network Simulated Performance Measures :**
  - Mean Number of Customers : 5104.7028
  - Throughput : 1.0405134940539904

Case $N = 30$**,** $p = 0.5$
- **Closed Network Simulated Performance Measures :**
  - Probability of Rejection : $1.0364263088585614 \times 10^{-27}$
  - Mean Number of Customers : 30.0
  - Throughput : 0.9961408568515335
  - Mean Sojourn Time : 0.35
- **Open Network Simulated Performance Measures :**
  - Mean Number of Customers : 5116.7679
  - Throughput : 1.159894959251206

Case $N = 30$, $p = 0.22$

- **Closed Network Simulated Performance Measures :**
  - Probability of Rejection : $1.3460373642780037 \times 10^{-13}$
  - Mean Number of Customers : 30.0
  - Throughput : 0.9502603346687116
  - Mean Sojourn Time : 0.378
- **Open Network Simulated Performance Measures :**
  - Mean Number of Customers : 5105.0628
  - Throughput : 1.0442664988122625

3.3 Comparison of Results

I compare :

- The impact of N on the studied metrics.
- The accuracy of different approximation methods.

Impact of N

- As N increases, the mean number of customers in the closed network simulation matches the value of N, indicating that the system is fully utilized.
- The probability of rejection is extremely low in all cases, suggesting that the system is capable of handling the given load without significant rejections.
- The throughput in the closed network simulation is slightly lower than in the open network simulation, which is expected due to the finite number of customers in the closed system.
- The mean sojourn time is consistent across different values of N and p, indicating stable performance.

Accuracy of Approximation Methods

- The closed network simulation provides performance metrics that closely align with the theoretical expectations.
- The open network simulation, while showing higher mean numbers of customers, also supports varying conditions.

Conclusion

In the Master REOC program, I have learned to analyze and simulate the performance of closed and open network systems. Through this case study, I have applied these concepts to evaluate performance metrics such as the probability of rejection, mean number of customers, throughput, and mean sojourn time.

My analysis shows that the closed network simulation provides performance metrics that align with theoretical expectations. The probability of rejection is extremely low, indicating that the system can handle the given load effectively. The mean number of customers in the closed network matches the value of N, demonstrating full utilization of the system. The throughput and mean sojourn time are consistent across different values of N and p, indicating stable performance.

The open network simulation, shows higher mean numbers of customers, also supports varying conditions. Both analytical and simulation-based approaches provide insights into the performance of the network, highlighting the importance of using multiple methods for comprehensive analysis.

Overall, this study reinforces the importance of understanding and applying stochastic modeling techniques to evaluate and optimize network performance in real-world scenarios.