UNIVERSITÀ DI PISA

DEPARTMENT OF COMPUTER ENGINEERING

Edge Computing
Project Documentation

January 11, 2025

Team:

Cavedoni F.
Monaci M.
Pinna F.

ACADEMIC YEAR 2024/2025

Contents

Chapter 1

Introduction

In recent years, the rapid growth of mobile devices and wireless networks has led to a dramatic increase in the demand for efficient and low-latency computational services. Traditional cloud computing architectures, while powerful, often suffer from latency issues due to the physical distance between end-users and centralized data centers. To address these challenges, *edge computing* has emerged as a promising paradigm by decentralizing computational resources closer to the users, significantly reducing latency and improving service quality.

This report focuses on evaluating the performance of a cellular network enhanced with edge computing capabilities. The system under study consists of M base stations arranged in a 2D floorplan of size $L \times H$ according to a regular grid. Each base station is equipped with computational resources and processes user-generated tasks following a *First Come First Served* (FCFS) policy. Additionally, all base stations are interconnected via a mesh topology, allowing communication and task forwarding between them.

Users are distributed across the floorplan and generate computational tasks at regular intervals. These tasks are sent to the geographically closest base station, which can either process the task locally or forward it to a less-loaded neighboring base station. This introduces a trade-off between processing delays and task-forwarding latency, which depends on the system's load and user distribution.

1.1 Problem Description

We consider N users placed at random locations `(x, y)` within the same floorplan, where the coordinates `x` and `y` are random variables to be defined based on specific distributions. Each user generates a new computational task request every T seconds, and each task consists of I instructions to be executed. Both T and I are exponentially distributed random variables.

A user sends each new task request to its closest serving base station, which can process the task using one of the following methods:

- **(A)** Serve the request locally at the receiving base station.
- **(B)** Forward the request to the less-loaded base station in the network. If this option is chosen, an additional fixed latency of D milliseconds is added to the total processing time.

1.2 Objectives

This project aims to achieve the following objectives:

- Evaluate the time required to complete a computing task for various values of N, T and I, comparing the two methods:
  - **Method (A):** Local task execution at the receiving base station.
  - **Method (B):** Task forwarding to the less-loaded base station.
- Assess the performance of the system under the following user placement scenarios:
  - **Uniform distribution:** `x` and `y` are uniformly distributed random variables in the range:

$$x \in [0, \text{width}], \quad y \in [0, \text{height}]$$

  - **Lognormal distribution:** `x` and `y` follow a lognormal distribution with defined parameters.

1.3 Performance Metrics

To evaluate the system's performance, the following metrics are considered:

- **Response time:** The time taken for a task to leave the system after being processed.
- **Queue length:** The number of tasks waiting to be processed at a base station.
- **Packet Forwarded:** The number of tasks that are forwarded to an another basestation due to the option B.
- **Packet Dropped:** The number of tasks that are dropped due to insufficient resources.

The analysis presented in this report provides insights into the trade-offs and performance implications of local versus collaborative task processing in edge computing-enabled cellular networks. These findings contribute to the understanding and optimization of edge computing architectures for modern applications.

Chapter 2

Modeling

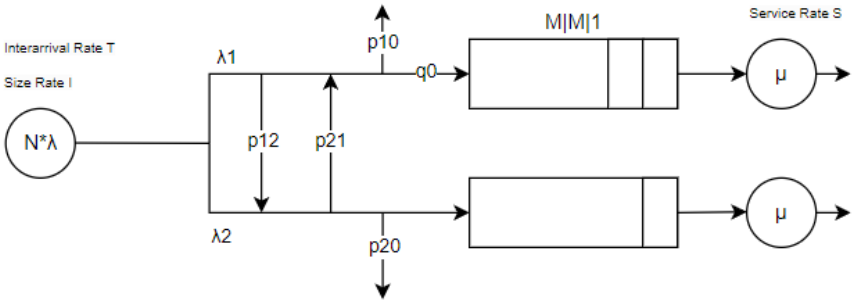

Figure 2.1: System modeling scheme.

The system under study is represented by the scheme shown in Figure 2.1. Each base station is modeled as an `M/M/1/K` queuing system, where λ_i represents the arrival rate of tasks at base station i, and μ is the service rate. Both λ_i and μ are exponentially distributed random variables, which aligns with the Markovian assumptions commonly used in queuing theory. The queues are of finite length, with a maximum capacity of K slots per base station. Any tasks arriving at a base station with a full queue is discarded.

The system consists of N users generating tasks at an interarrival rate λ. These tasks are routed to one of the available base stations according to their proximity and load. The system supports task forwarding between base stations, as indicated by the probabilities p_{ij}, which represent the likelihood of a task being forwarded from base station i to base station j.[1] Forwarding introduces an additional latency, which is assumed to be constant for all base stations.

[1] It is obvious that $p_{ij} = 0$ when evaluating method A

As illustrated in the scheme, each task is processed according to a *First Come First Served* (FCFS) policy. The finite queue length K ensures that the system reflects realistic constraints, where resources are limited, and overloading results in task loss. Additionally, tasks may originate from users distributed either uniformly or according to a lognormal distribution, allowing for the evaluation of the system's performance under diverse user distributions.

2.1 System Parameters

Below, we provide a detailed explanation of all the parameters used in the system model, with specific emphasis on their dependencies and characteristics:

- N**: Number of users**
  The total number of users in the system. These users are distributed across the 2D floorplan according to either a *uniform* or a *lognormal* distribution. Higher values of N result in increased load on the base stations.

- K**: Queue length**
  The maximum number of tasks a base station can hold in its queue at any given time. If the queue is full, additional incoming tasks are discarded. This parameter models resource constraints at the base station level.

- λ_i**: Interarrival rate for base station** i
  The task arrival rate for each base station i depends directly on the user distribution. In the case of a *uniform distribution*, users are evenly spread across the floorplan, leading to nearly balanced λ_i values across all base stations. In contrast, for a *lognormal distribution*, certain base stations may experience higher arrival rates due to user clustering in hotspots, resulting in uneven λ_i values.

- S**: Instruction rate (fixed)**
  The rate at which instructions are executed at each base station is fixed across the system and represents the computational capacity of the edge servers.

- μ**: Packet size distribution (exponential)**
  The size of each task (in instructions per packet) follows an *exponential distribution*. This variability in task size directly impacts the service rate for each base station, making it effectively exponential, as the time to serve a task depends on its size.

- p_{ij}**: Forwarding probability**
  The probability that a task is forwarded from base station i to base station j. This depends on the relative load between base stations and is used to balance the computational workload across the system.

- D**: Forwarding delay**
  The additional latency incurred when a task is forwarded to a less-loaded base station. This delay is assumed to be constant for all tasks and base stations and represents the propagation time in the mesh network.

- $\mu_{\mathbf{log}}$**: Mean of the lognormal distribution**
  The mean value of the lognormal distribution used to define the clustering of users in the floorplan. This parameter determines the overall central tendency of user placement in the floorplan.

- $\sigma_{\mathbf{log}}$**: Standard deviation of the lognormal distribution**
  The standard deviation of the lognormal distribution, controlling the spread of user placement. Higher values of $\sigma_{\log}$ result in more dispersed user locations, while lower values create tighter clusters.

Chapter 3

Implementation

The implemented system in OMNeT++ is organized into modular components to reflect the structure and behavior of the edge computing-enabled cellular network. Each module has a specific responsibility, ensuring a clear and maintainable architecture.

3.1 Defined Modules

The following modules have been defined for the simulation:

- **EdgeNetwork (Compound Module)**
  This is the top-level module representing the entire system. It hosts the following simple modules:
  - **BaseStation (Simple Module):** Represents a single base station in the network. It is responsible for receiving, processing, and, if necessary, forwarding packets generated by users.
  - **User (Simple Module):** Represents a user generating computational tasks. Each user sends packets (with lengths determined by a specified distribution) to its nearest base station.

3.2 Module Behavior

The behavior of each module is described in detail below:

3.2.1 EdgeNetwork

- Acts as the parent module, hosting all base stations and users within the simulation.
- Stores the parameters of all contained modules, allowing them to be retrieved during the simulation using parent pointers.

- Provides the spatial layout of base stations and users, ensuring that the correct associations (e.g., nearest base station) are maintained.

3.2.2 BaseStation

- Receives packets from users in the form of `cPacket` objects and processes them based on the specified scenario:
  - **Locally Managed:** If the queue has sufficient free slots, the base station enqueues the packet for local processing, ignoring the state of other base stations.
  - **Forwarding:** Upon receiving a packet, the base station evaluates the load of all other base stations and forwards the packet to the one with the lowest queue load. If no other base station has a lower load, the packet is processed locally.
- Records the following statistics:
  - **Dropped Packets:** The number of packets dropped due to full queues.
  - **Forwarded Packets:** The number of packets forwarded to other base stations.
  - **Queue Length:** The number of packets in the queue over time, used to compute averages.
  - **Response Time:** The time taken for packets to be processed, allowing for average response time computation.

3.2.3 User

- Generates packets with:
  - **Length:** Packet lengths follow an exponential distribution.
  - **Rate:** The packet generation rate also follows an exponential distribution.
- Sends each packet to the nearest `BaseStation`, determined by the Euclidean distance.[1]

[1] This ensures that users are associated with geographically closest base stations, a common assumption in cellular networks.

Chapter 4

Verification

This chapter describes the tests performed to verify the stability and behavior of the system under various scenarios, including extreme and degenerate cases. The results of these simulations help validate the robustness, consistency, and correctness of the implementation.

4.1 Degeneracy Test

This test evaluates the system's stability and behavior under extreme or edge-case conditions. The goal is to ensure that the system remains robust and operates without failures or unexpected behaviors when subjected to these scenarios. By testing the system in degenerate cases, we validate its ability to handle both minimal and excessive loads reliably.

4.1.1 Scenario 1 - No Users

In this degenerate scenario, no users are present in the system. As expected, the system remains stable because there are no packets to process or forward. This verifies the correctness of the system's behavior in edge cases with no load.

4.1.2 Scenario 2 - Very High Number of Users

This scenario tests the system under an extreme load with a very high number of users ($N = 500$) and four base stations ($M = 4$).

The system demonstrates robustness under these conditions, maintaining a stable *queue length* and consistent *response time*. However, the queue is always full, leading to a significant number of dropped packets, and the *response time* stabilizes due to the saturated queue. The results are shown in Figure 4.1 and Figure 4.2, confirming that the system can handle significant load without degenerating.

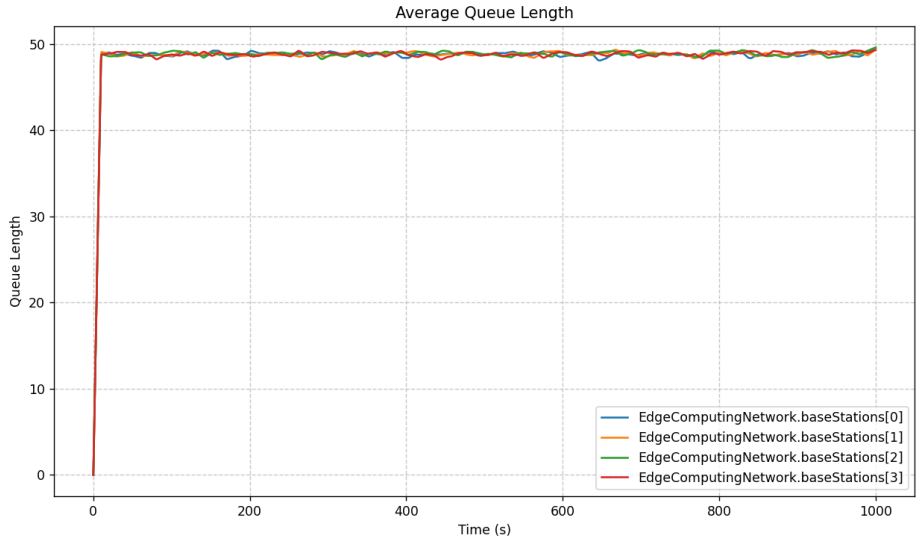

Figure 4.1: Average queue length under high load.

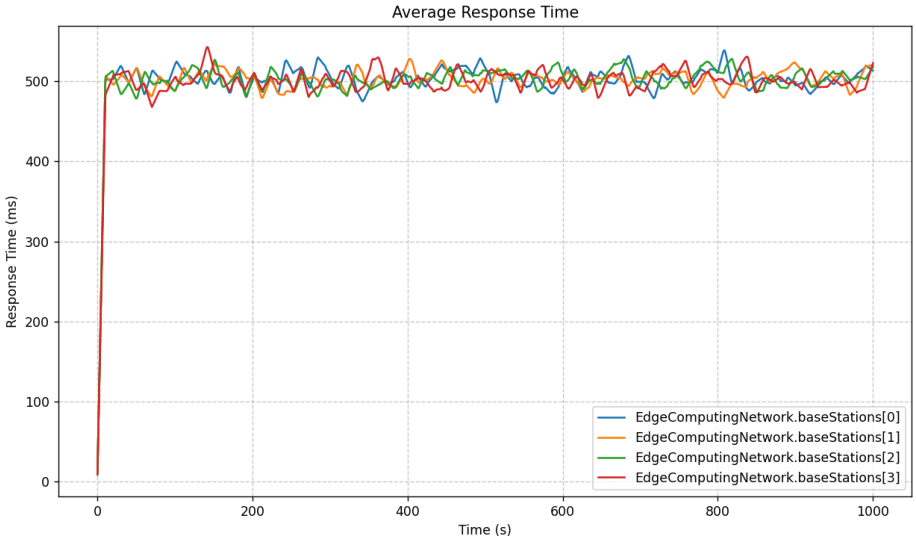

Figure 4.2: Average response time under high load.

4.1.3 Scenario 3 - Very Low Service Rate

This scenario evaluates the system when the service rate is significantly lower than the interarrival rate.

The simulations show that the queue length stabilizes very fast (Figure 4.3), while the response time exhibits a *linear growth* over time (Figure 4.4). This behavior is consistent with expectations under such extreme conditions.

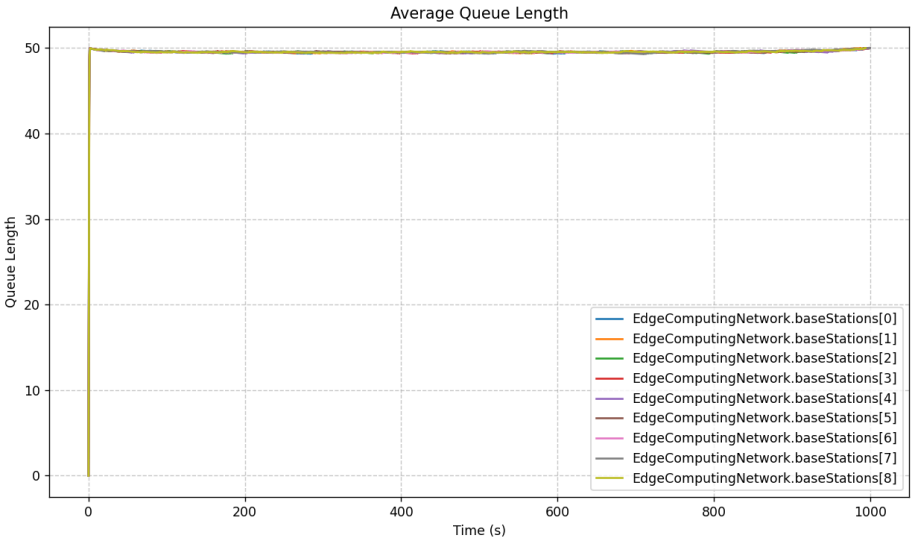

Figure 4.3: Queue length stabilizes under low service rate.

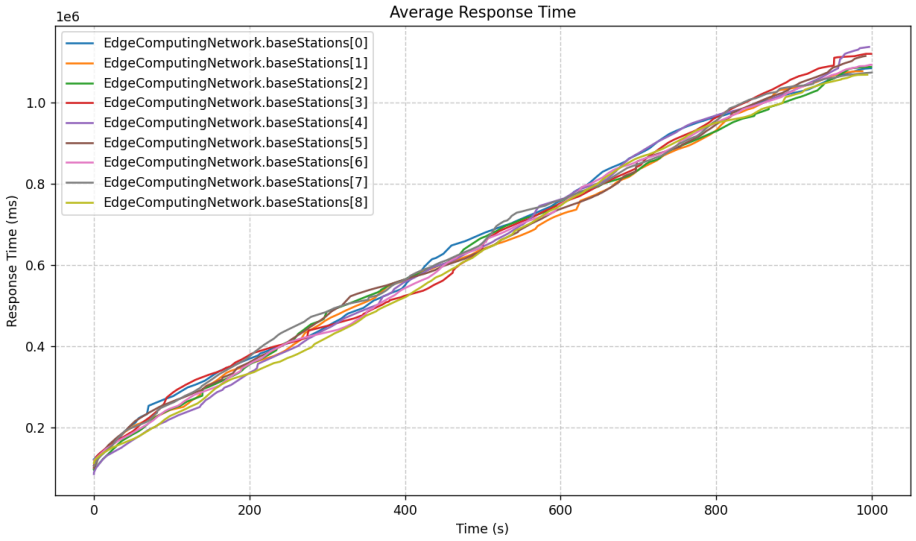

Figure 4.4: Linear growth in response time under low service rate.

4.2 Consistency Test

This test verifies that the system produces coherent results under both configurations: *Locally Managed* and *Forwarding*.

The results indicate similar mean response times, with the forwarding configuration showing a slight advantage. However, the forwarding configuration also exhibits lower variability, as seen in Figure 4.5 and Figure 4.6.

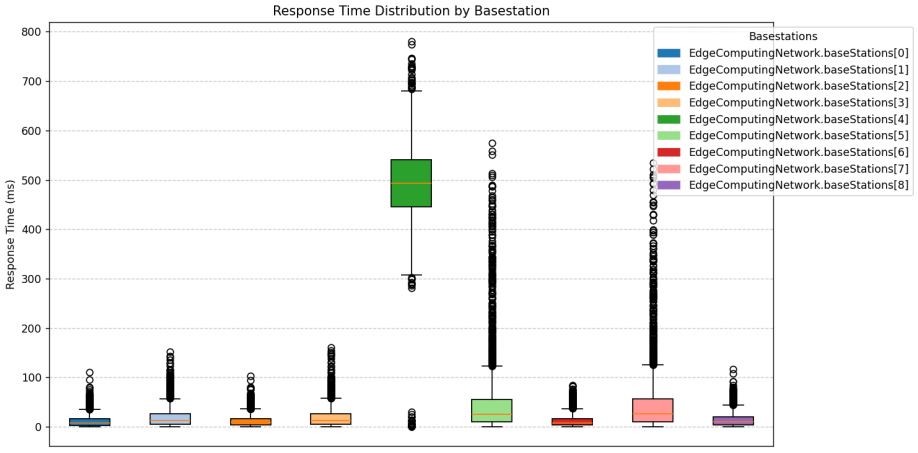

Figure 4.5: Response time: Locally managed.

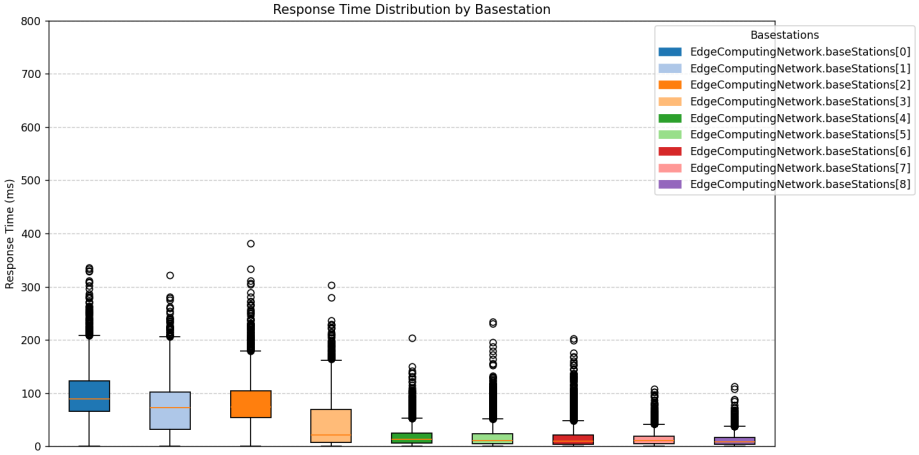

Figure 4.6: Response time: Forwarding.

4.3 Continuity Test

This test ensures that the system responds smoothly when parameters are varied gradually, without abrupt discontinuities in behavior.

4.3.1 Increment of Users ($N = 100, 250, 500$)

As expected, the mean response time increases with the rising number of users. It remains relatively stable between $N = 100$ and $N = 250$, but exhibits a sharp and significant increase as the number of users reaches $N = 500$.

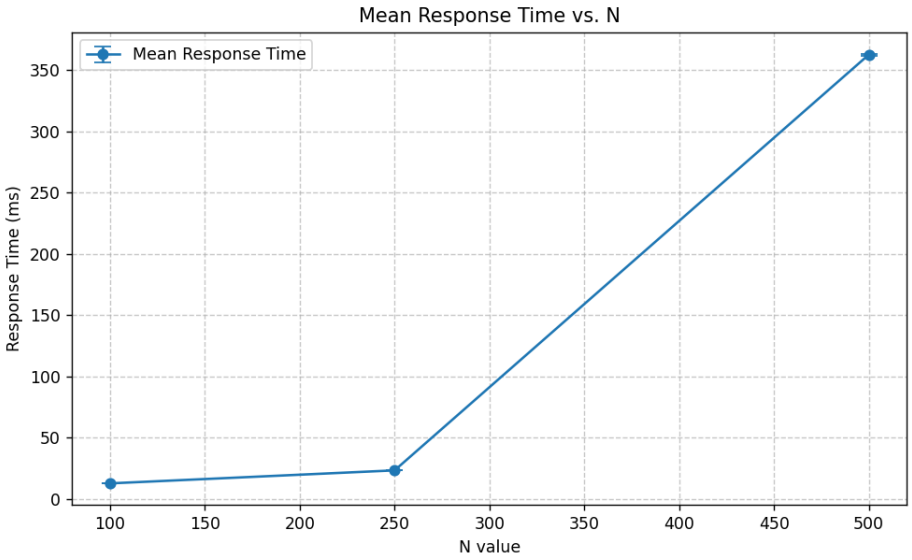

Figure 4.7: Response time for varying numbers of users.

4.3.2 Increment of Requests' Load ($\lambda = 1/0.1, 1/0.5, 1$)

At low values of λ, the mean response time remains relatively low and stable. However, as λ approaches $1/0.1$, the mean response time experiences a sharp increase, nearly an order of magnitude higher. This growth is also accompanied by increased variability in the response time.

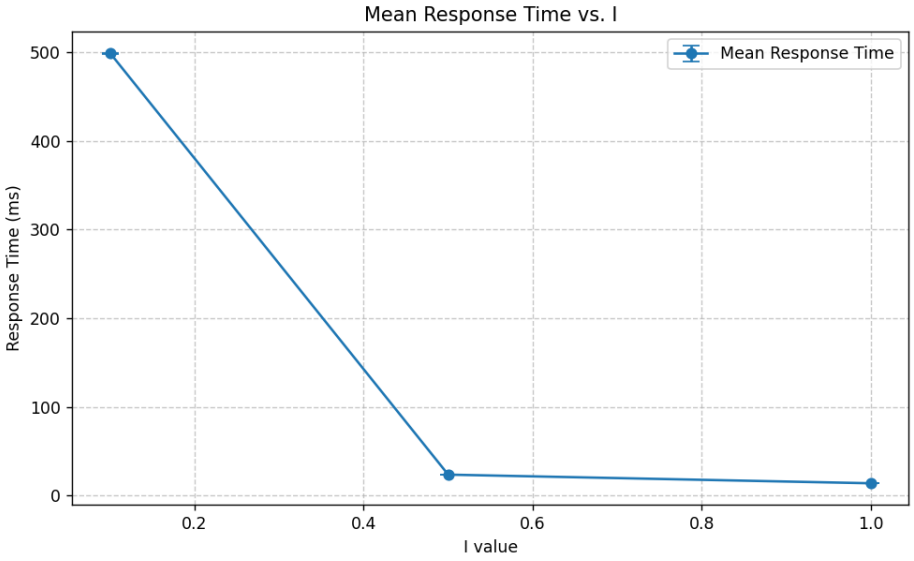

Figure 4.8: Response time for varying interarrival rates.

4.3.3 Increment of Size Rate ($\mu = 100, 1000, 10000$)

The response time increases significantly as the size rate grows. Notably, for $\mu = 10000$, the response time is two orders of magnitude higher compared to the lower size rates. This trend highlights the strong dependency of response time on the size rate of each packet, as larger packets require substantially more processing time.

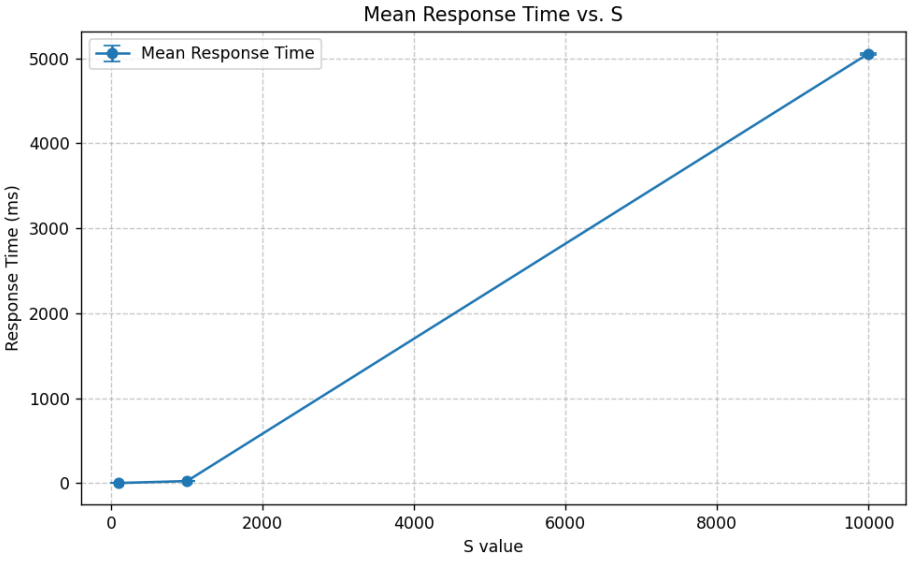

Figure 4.9: Response time for varying service rates.

Chapter 5

Calibration

In this chapter, we discuss how the factors and parameters have been chosen throughout the development of our system. We distinguish between *fixed factors* and *varying factors* (representing different scenarios).

5.1 Factors Calibration

The objective of this section is to specify the intervals of the key factors in order to correctly reproduce realistic conditions in our system.

5.1.1 Fixed Factors

The following factors remain constant throughout our experiments:

- **Number of base stations (M)**: 9
- **Dimensions of the area**: height = 1800 m, width = 1800 m
- **Service rate**: 10^5 (instructions/second)
- **Delay**: 50 ms
- **Queue size**: 50
- **Distribution parameters**:
  - Mean: $\log\left(\frac{1800}{2}\right)$
  - Standard Deviation: 0.4

The number of base stations ($M = 9$) aims to reflect a moderately sized area with multiple sites, balancing suburban and urban deployments. The dimensions of the area ($1800\,m \times 1800\,m$) were chosen to represent a typical environment where base stations are neither too sparse nor too dense. The service rate of 10^5 instructions/second, along with a 50 ms delay and a queue size of 50, is

used to simulate a realistic cellular-network-like environment where moderate buffering and short delays are expected. Finally, the log-normal distribution (characterized by the specified mean and standard deviation) models the spatial distribution of users. This distribution is particularly suitable for analyzing cases where user concentration is uneven across the area, such as in scenarios with hotspots and sparse regions.

5.1.2 Varying Factors

The following factors vary to represent different scenarios and test system performance under varying conditions:

- **Number of users (N)**:
  - $\boldsymbol{N = 100}$: Represents a lightly loaded scenario, simulating a low-density environment with fewer active users.
  - $\boldsymbol{N = 250}$: A moderately loaded scenario, reflecting a typical environment with a balanced user density.
  - $\boldsymbol{N = 500}$: A heavily loaded scenario, used to evaluate system performance under high user density and demand.
- **Interval rate**: This parameter determines how frequently users generate requests (i.e., the arrival rate). We consider two scenarios:
  - **Medium-case scenario**: $\mathbf{1/0.5}$, representing a moderate traffic load with steady user activity.
  - **Extreme-case scenario**: $\mathbf{1/0.1}$, emulating peak traffic conditions with frequent user requests.
- **Size rate**:
  - **Medium-case scenario**: $\mathbf{10^3}$: Corresponds to smaller data packet sizes, typical of lightweight applications or systems with low bandwidth requirements.
  - **Extreme-case scenario**: $\mathbf{10^4}$: Represents larger data packet sizes, increasing system load and reflecting heavier traffic conditions.

The combination of these parameters enables us to evaluate the system under a broad range of realistic conditions, from moderate loads with fewer users and smaller data sizes to extreme loads with higher user counts, larger data sizes, and higher arrival rates. These variations ensure that the analysis is comprehensive and captures critical performance metrics under both typical and challenging conditions.

5.2 Warmup Time Analysis

Warmup time refers to the period during which the system transitions from an initial transient state to a steady state. Properly setting the warmup time ensures that the system's performance metrics are evaluated only during the steady-state period, avoiding bias caused by the initial transient.

In our analysis, we observed the response time as a function of time. As shown in Figure 5.1, the response time stabilizes and reaches a steady state within 100 seconds. The transient period, characterized by larger variations in response time, does not exceed 100 seconds across all experiments.

Based on these observations, the warmup time is empirically set to 100 seconds. This ensures that any data collected during the transient state is excluded from the analysis, leading to more accurate and reliable results.

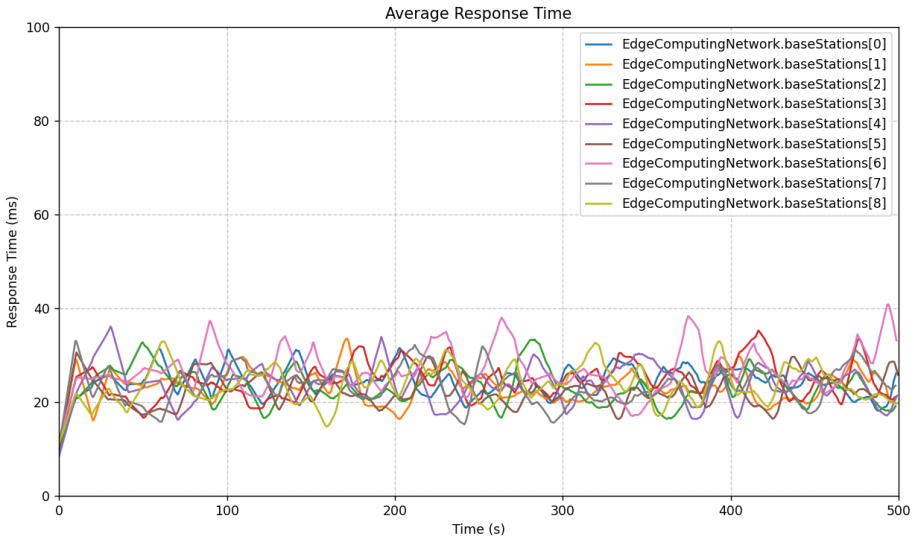

Figure 5.1: Response time as a function of time. The transient period does not exceed 100 seconds, indicating the warmup time.

Chapter 6

Simulation Experiments

6.1 Experiment design

We test our system in both cases A and B to see the effects of forwarding packets wether the packets are generated according to a uniform distribution or a lognormal distribution.

The tests also take into account 3 factors:

- **N**: (250, 500)
- **Interarrival rate**: $(\frac{1}{0.1}, \frac{1}{0.5})$
- **Size rate**: $(\frac{1}{10^3}, \frac{1}{10^4})$

Considering the average case as a reference:

- $N = 250$
- $\lambda = \frac{1}{0.5}$
- $\mu = \frac{1}{10^3}$

6.2 The effect of varying the number of users

In this section, we analyze the system performance when the user population N vary. As N grows, the overall load on each base station increases, affecting both response times and queue lengths.

Number of User variation - Method A

Low User Number: $N = 250$

When the user population is moderate ($N = 250$), both the uniform and lognormal distributions exhibit moderate queueing behavior. From Figure 6.1, the

mean response time $\big(E[R]\big)$ remains well below saturation for both distributions. The uniform distribution yields more predictable performance, while the lognormal distribution can generate short-lived spikes due to burstier arrivals. Overall, the uniform distribution demonstrates slightly better performance under medium load conditions.

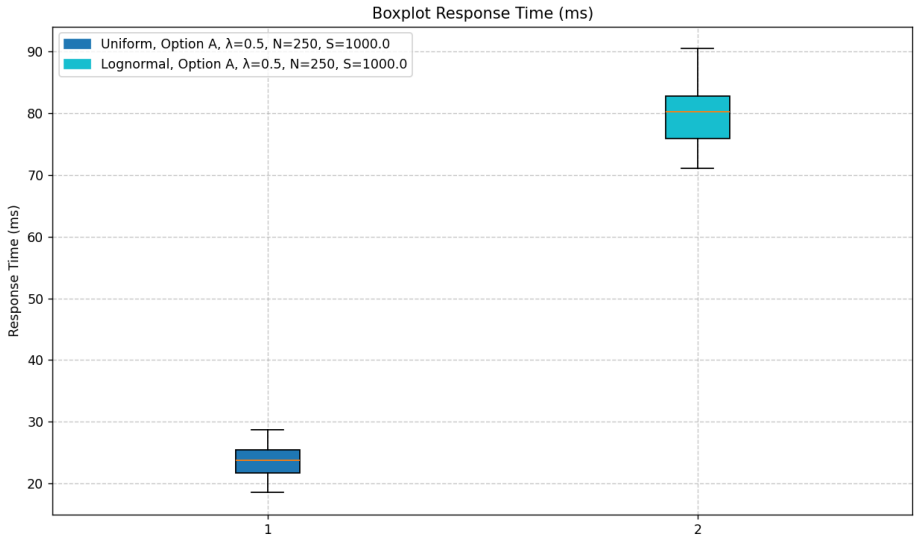

Figure 6.1: Comparison of $E[R]$ (Method A) between uniform and lognormal distributions for $N = 250$

From Figure 6.2, we observe that the mean queue length $\big(E[qlen]\big)$ is low for both distributions, and packet drops are practically nonexistent. With fewer users, each base station easily manages its own queue, leading to relatively stable performance irrespective of the arrival distribution.

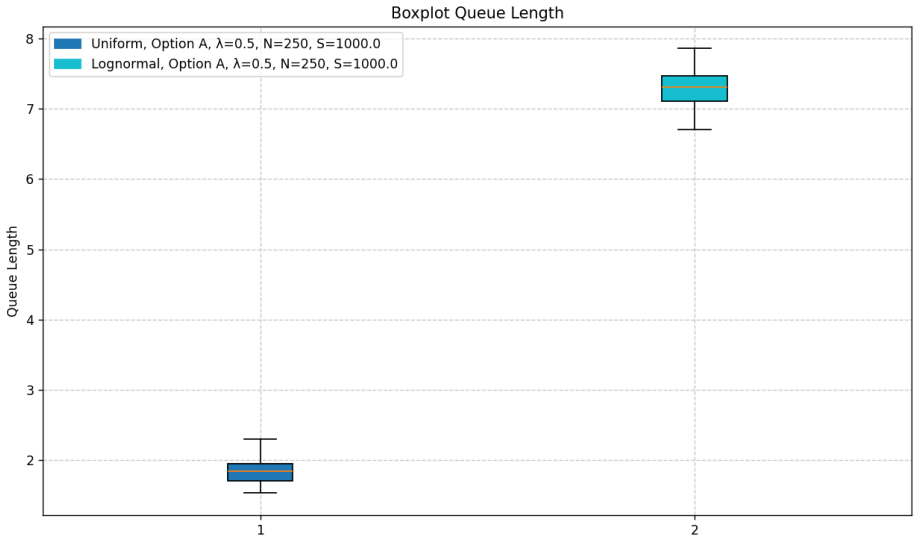

Figure 6.2: Comparison of $E[qlen]$ (Method A) between uniform and lognormal distributions for $N = 250$

High User Number: $N = 500$

As N increases to 500, the system encounters much heavier load when relying solely on local processing (Method A). Figure 6.3 shows that the lognormal distribution seems to yield a slightly better mean response time than the uniform distribution. This apparent advantage is misleading, as it stems from the higher number of packets being dropped due to the non-uniform nature of the arrivals.

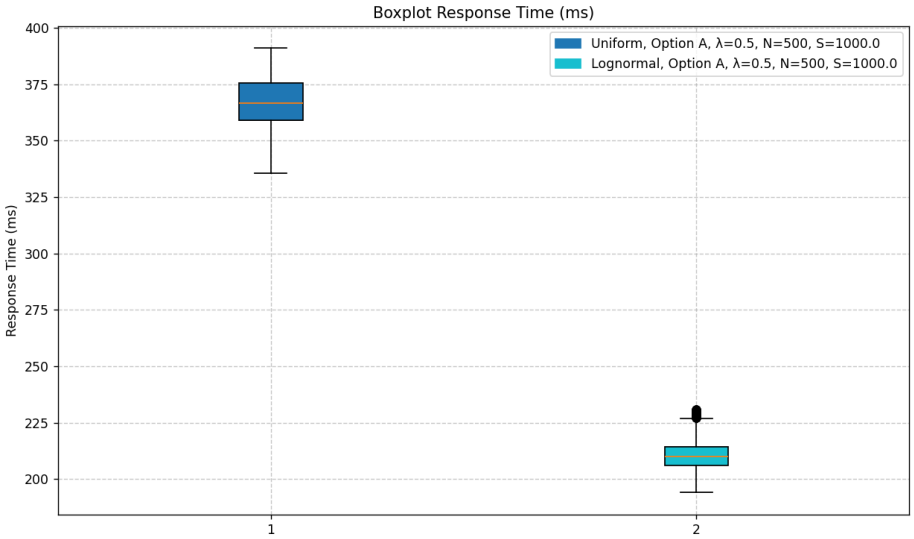

Figure 6.3: Comparison of $E[R]$ (Method A) between uniform and lognormal distributions for $N = 500$

In Figure 6.4, queue lengths $\big(E[qlen]\big)$ reveal that, under Method A, base stations can reach higher congestion levels when arrivals follow a lognormal distribution. This imbalance leads to occasional packet drops at overloaded stations.

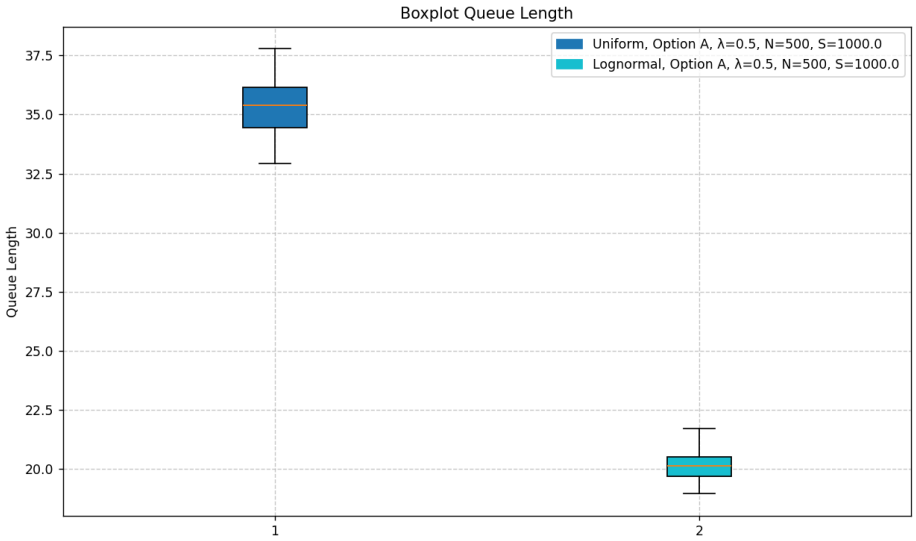

Figure 6.4: Comparison of $E[qlen]$ (Method A) between uniform and lognormal distributions for $N = 500$

To highlight the differences in dropped packets, Figure 6.5 provides a comparison of the dropped packet distributions between the two scenarios. As shown, the lognormal distribution causes significantly more packet drops due to its burstier arrival patterns, particularly under heavy load conditions.

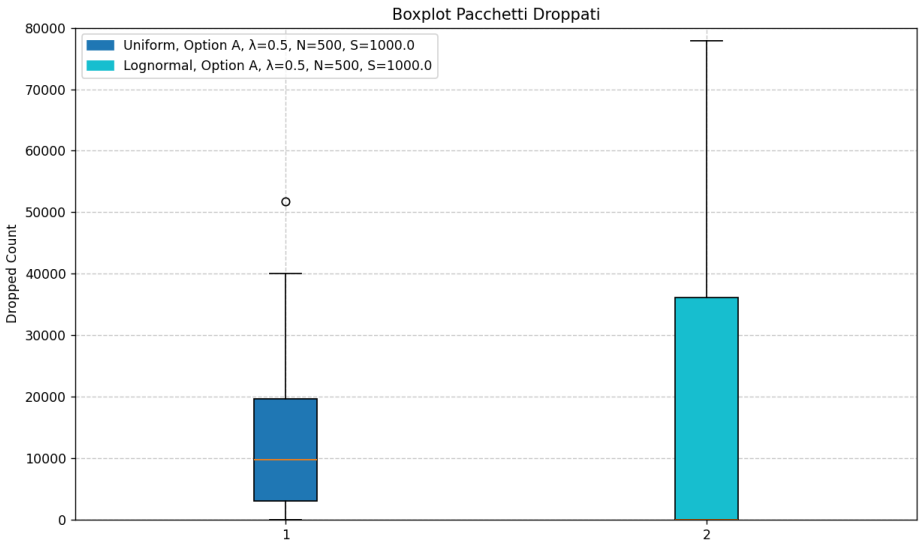

Figure 6.5: Boxplot of dropped packets (Method A) comparing uniform and lognormal distributions

Number of User variation - Method B

Low User Number: $N = 250$

Even at lower user densities ($N = 250$), the forwarding mechanism (Method B) equalizes the load across base stations. Figure 6.6 shows that both uniform and lognormal distributions yield comparable mean response times $\big(E[R]\big)$. With fewer users, the forwarding strategy is not heavily taxed, but it still prevents localized spikes in queue lengths for the lognormal case. However, the performance of the uniform distribution in case A are slightly better than the one in the case B due to the delay of the forwarding.

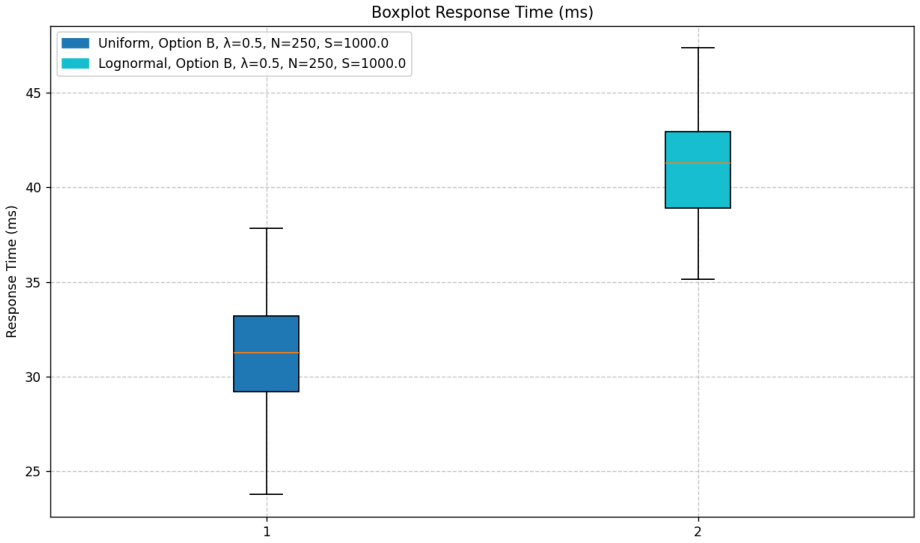

Figure 6.6: Comparison of $E[R]$ (Method B) between uniform and lognormal distributions for $N = 250$

In Figure 6.7, the mean queue lengths $\big(E[qlen]\big)$ for both distributions are again modest, and few—if any—packet drops occur. Offloading surplus tasks to neighboring base stations mitigates short-term congestion in lognormal arrivals, ensuring stable performance.

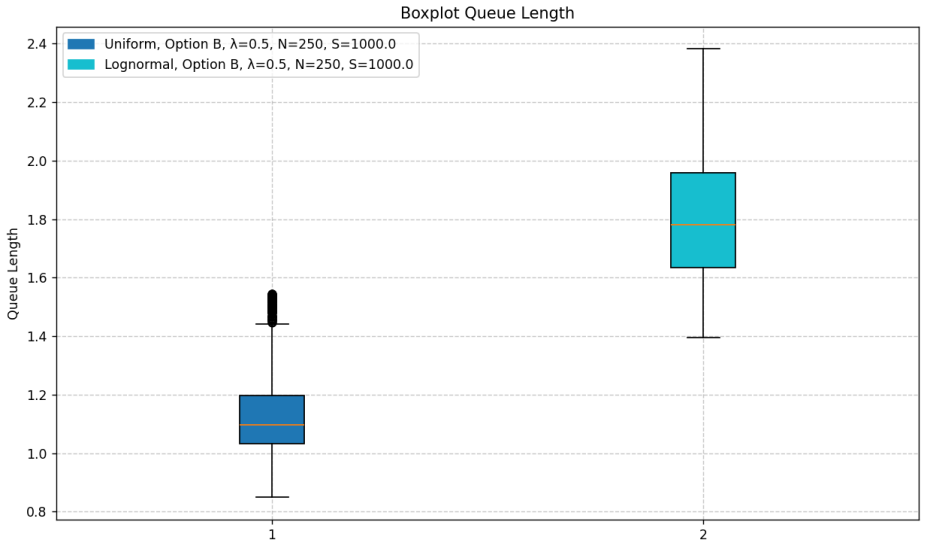

Figure 6.7: Comparison of $E[qlen]$ (Method B) between uniform and lognormal distributions for $N = 250$

High User Number: $N = 500$

At higher user densities ($N = 500$), Figure 6.8 indicates that Method B continues to effectively balance the load. Mean response times $\big(E[R]\big)$ remain relatively close for both the uniform and lognormal distributions. However, the lognormal case shows slightly better perfotmance than the uniform case, due to the more packets dropped, even tho this phenomenon is lower than the method A due to the balancing of the forwaring procedure.

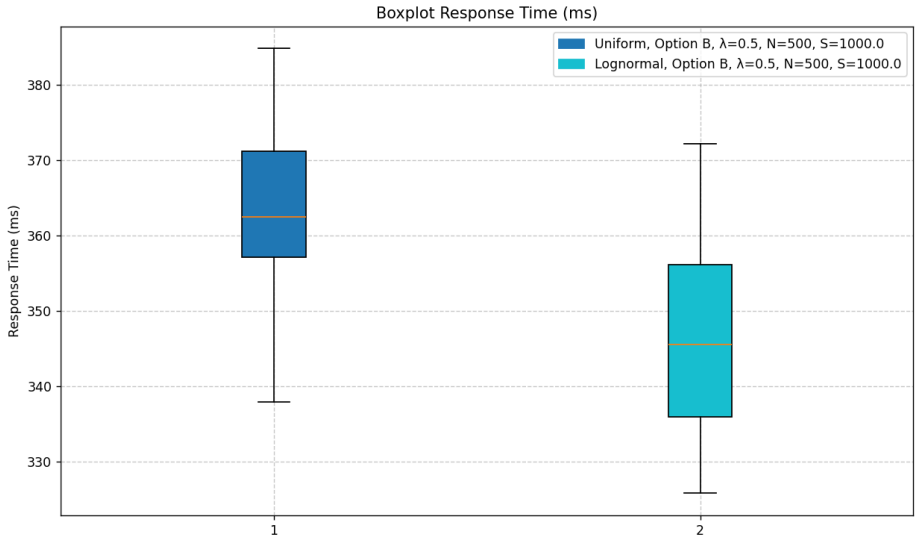

Figure 6.8: Comparison of $E[R]$ (Method B) between uniform and lognormal distributions for $N = 500$

Figure 6.9 shows how the queue lengths $\big(E[qlen]\big)$ remain balanced across the network in both arrival scenarios. Even there the phenomenon of the uneven distribution is present, minimized by the forwarding procedure.

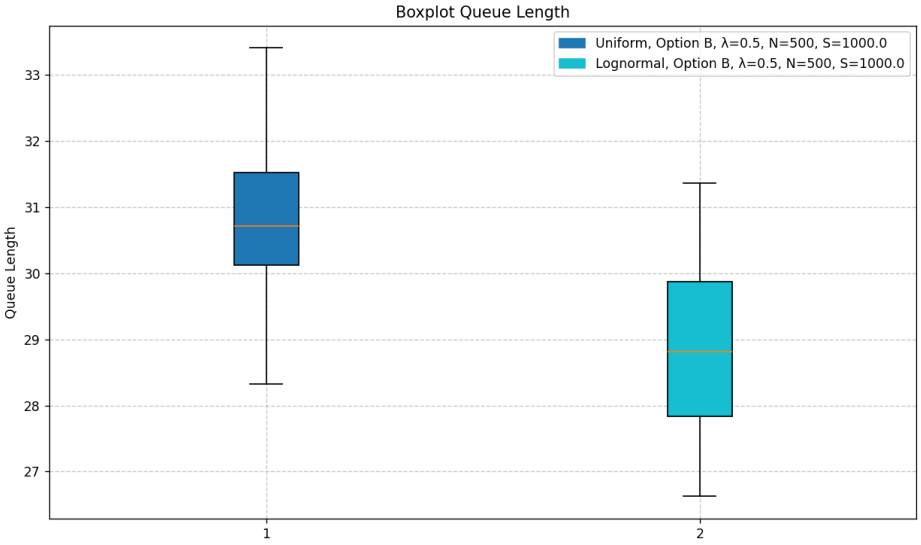

Figure 6.9: Comparison of $E[qlen]$ (Method B) between uniform and lognormal distributions for $N = 500$

Finally, Figure 6.10 presents the dropped-packet counts for the lognormal distribution under Method B at the highest load (i.e., $N = 500$). Drops for the lognormal distribution remain high but are way lower than the drops for tthe option A, demonstrating the forwarding mechanism's success in offloading traffic to avoid per-base-station saturation. The performance of the uniform case are slightly worse, due to the unnecessary forwarding delay.

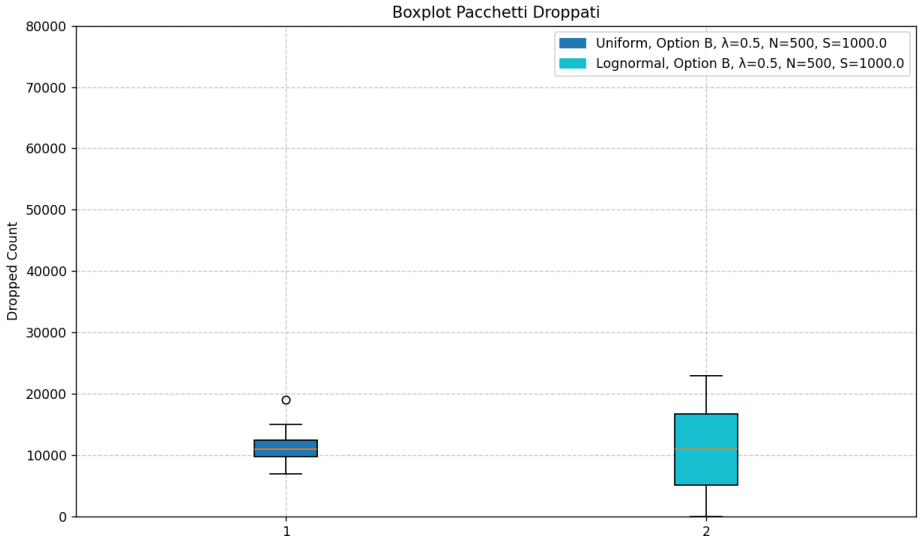

Figure 6.10: Dropped packets for the lognormal distribution at $N = 500$ (Method B)

6.3 The effect of varying the size rate

In this section, we investigate the performance of the system while changing the size rate, denoted by S.

6.3.1 Size rate variation - Method A

Moderate size rate: $\mu = \frac{1}{10^3}$

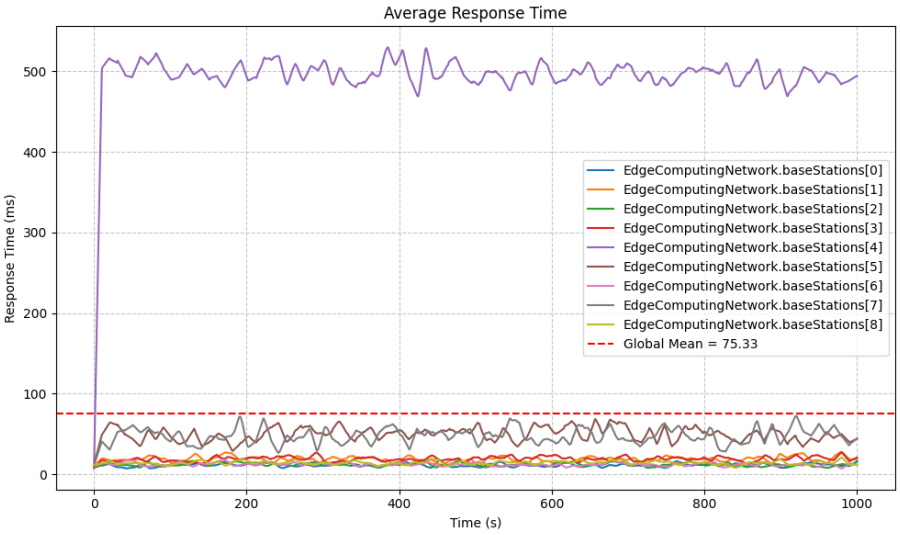

Figure 6.11: $E[R]$ for the lognormal distribution

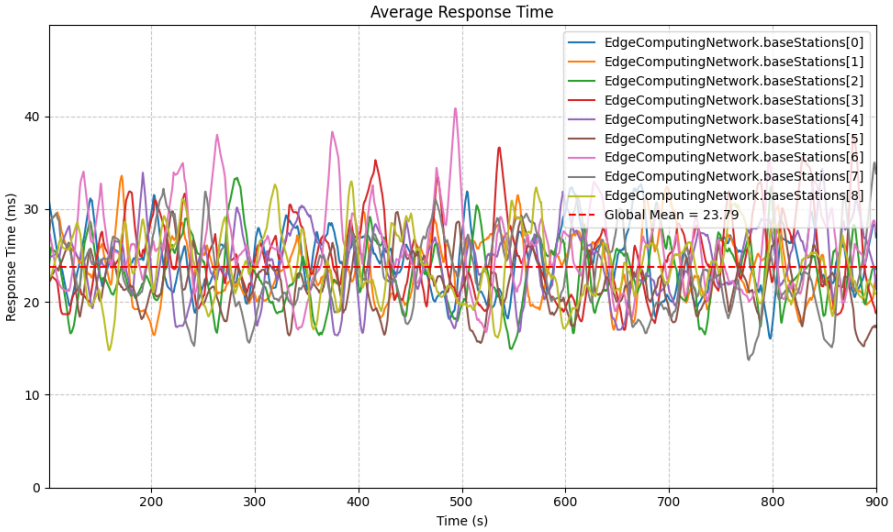

Figure 6.12: $E[R]$ for the uniform distribution

The difference between the two distribution is remarkable, as in the lognormal case we find a higher mean response time and queue length.

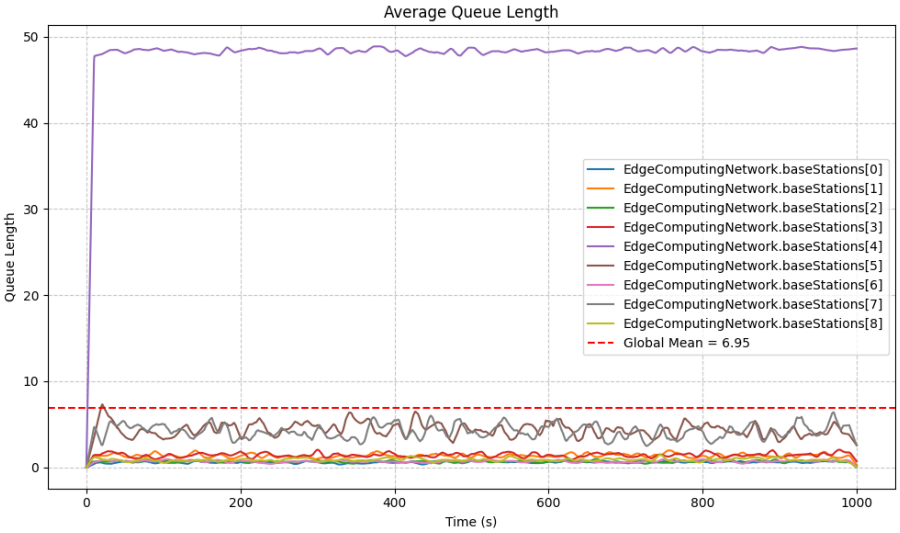

Figure 6.13: $E[qlen]$ for the lognormal distribution

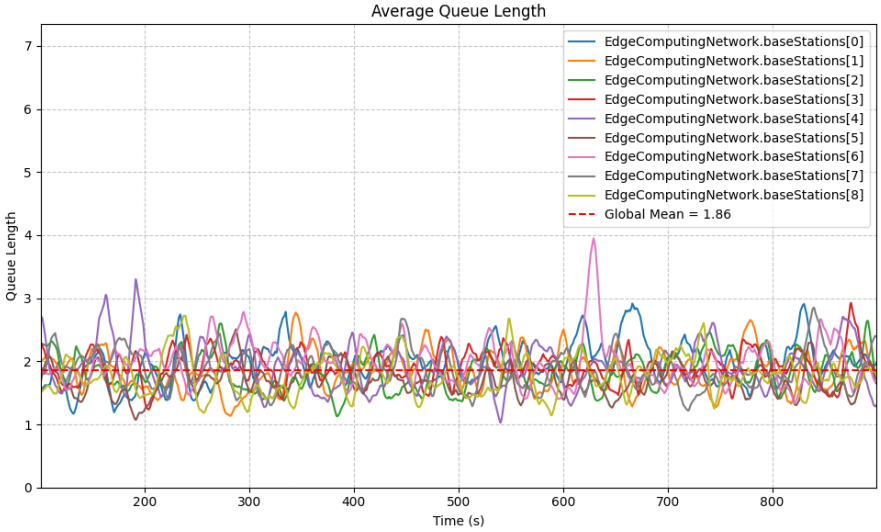

Figure 6.14: $E[qlen]$ for the uniform distribution

In the uniform distribution the values are more compact around the mean,

unlike the lognormal case, where is evident that there are one or more base stations dealing with a heavier load.

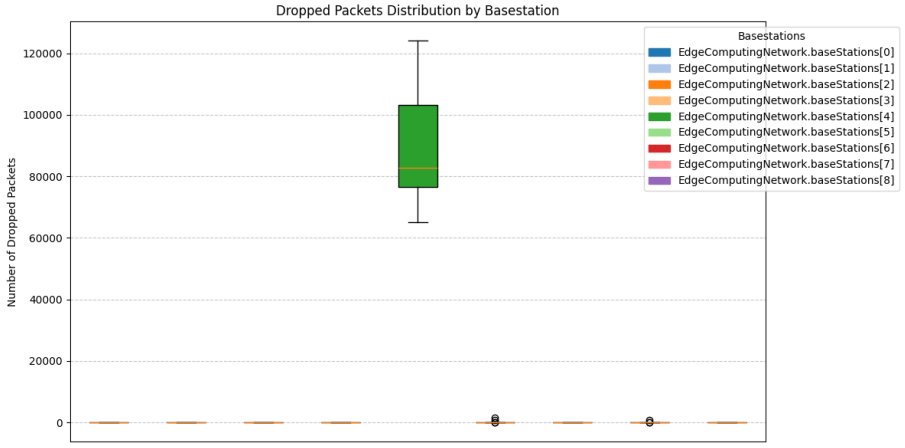

Figure 6.15: Dropped packets distribution (lognormal)

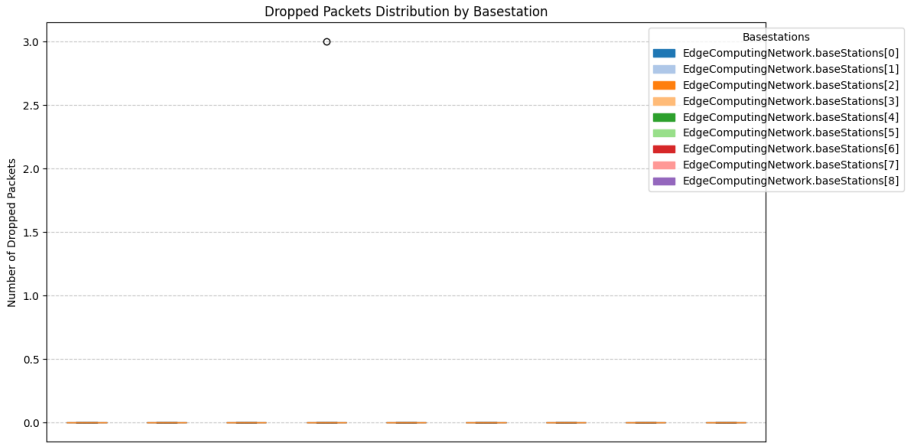

Figure 6.16: Dropped packets distribution (uniform)

Since base stations use the policy where packets are served locally, a higher queue length might mean a higher number of discarded packets. Even though, the system can still sustain the load in the uniform case as the number of discarded packets is lower than 10.

Not the same for the lognormal case, where the total of discarded packets is over 2.5×10^5 units.

High size rate: $\mu = \frac{1}{10^4}$

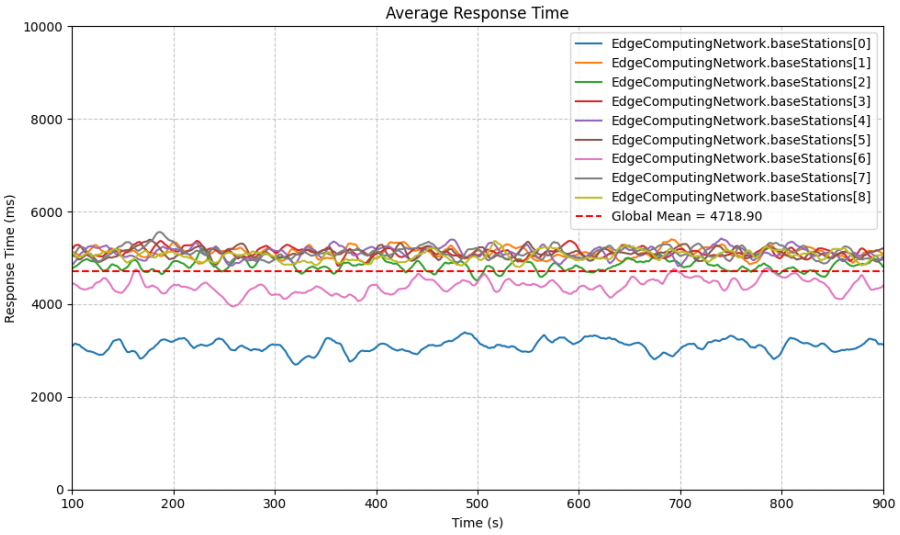

Figure 6.17: $E[R]$ for the lognormal distribution

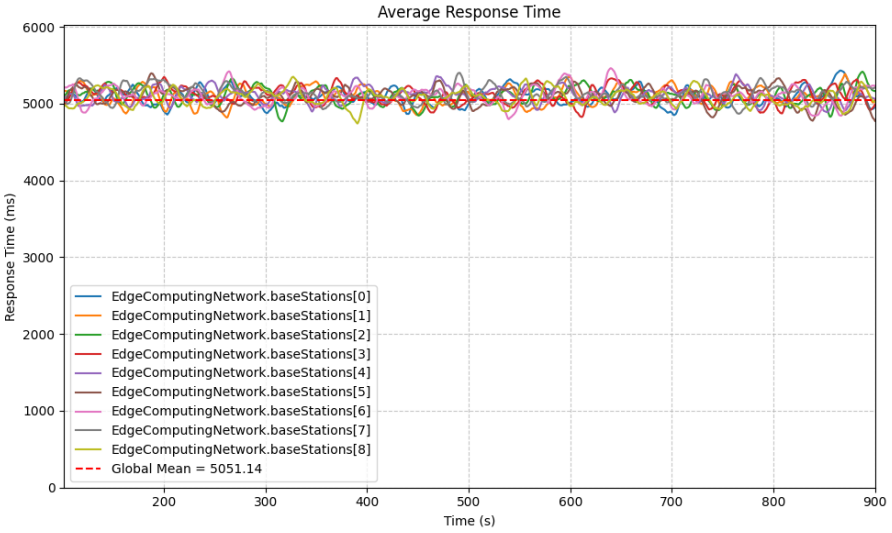

Figure 6.18: $E[R]$ for the uniform distribution

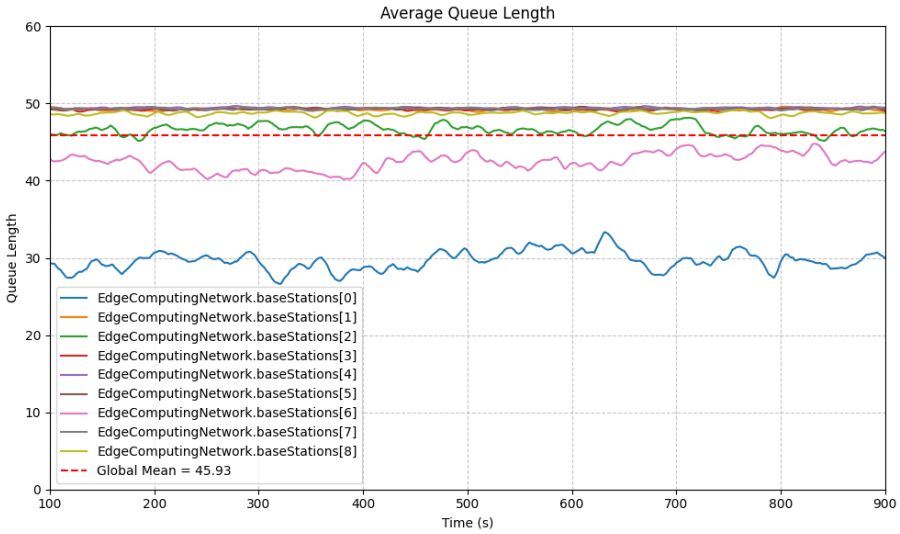

Figure 6.19: $E[qlen]$ for the lognormal distribution

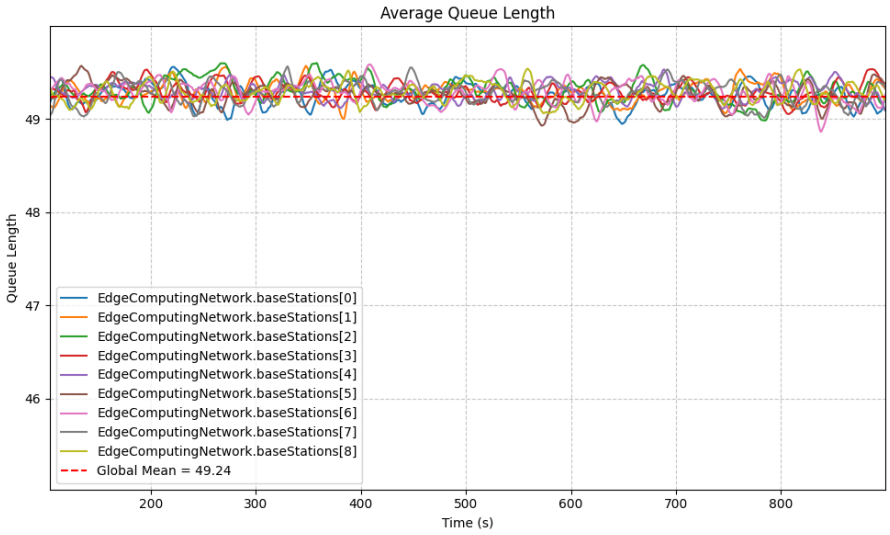

Figure 6.20: $E[qlen]$ for the uniform distribution

In this experiment, the results suggest that the lognormal case appears to perform better; however, this improvement is only apparent. This is due to the unbalanced load distribution among the basestations: while some basestations are overloaded, others remain underutilized. As the response time saturates

for the overloaded basestations, the contribution of the underutilized ones lowers the overall average response time. This phenomenon creates the illusion of better performance in the lognormal case.

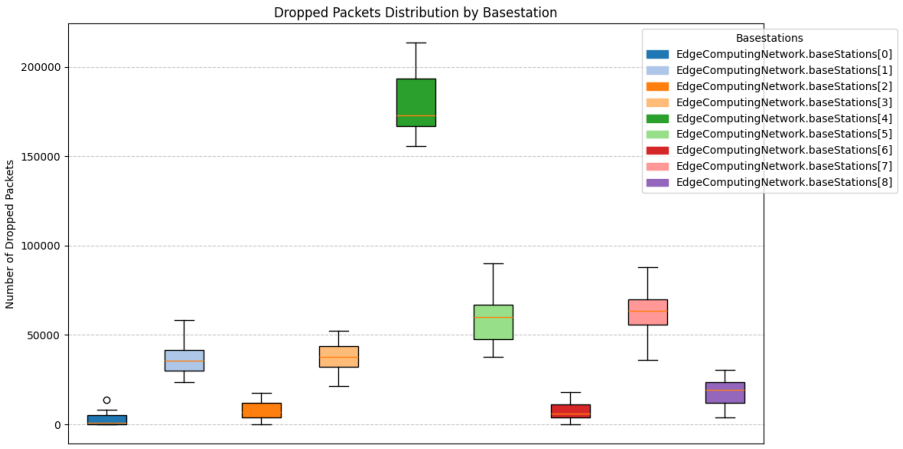

Figure 6.21: Dropped packets distribution (lognormal)

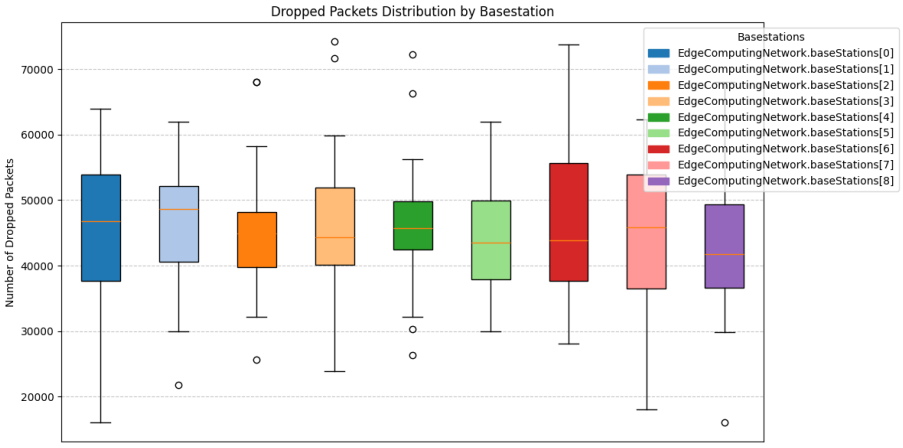

Figure 6.22: Dropped packets distribution (uniform)

However, as illustrated in these graphs, it becomes evident that a significantly higher number of packets are dropped in the lognormal case compared to the uniform case.

Moreover, the packet drops occur in a non-uniform manner, further highlighting the inefficiencies of the lognormal load distribution when compared to the uniform case.

6.3.2 Size rate variation - Method B

Moderate size rate: $\mu = \frac{1}{10^3}$

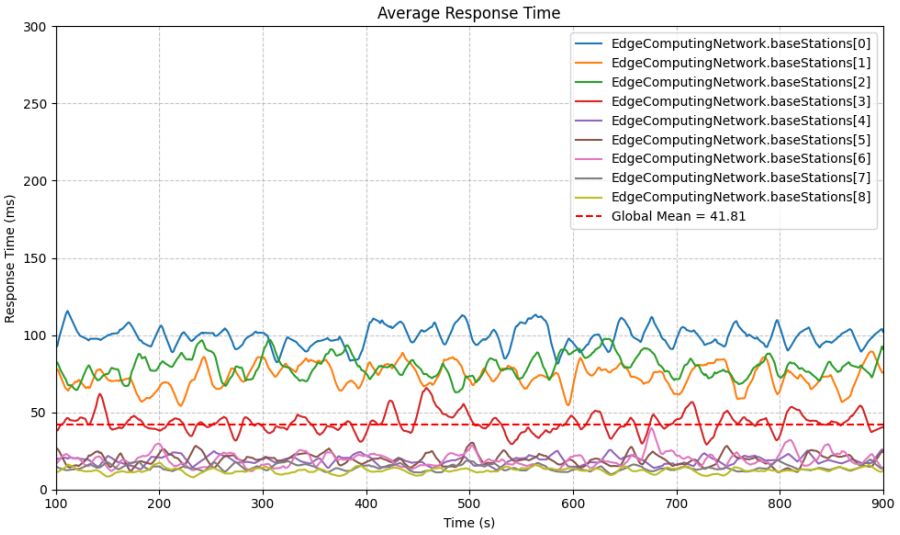

Figure 6.23: $E[R]$ for the lognormal distribution

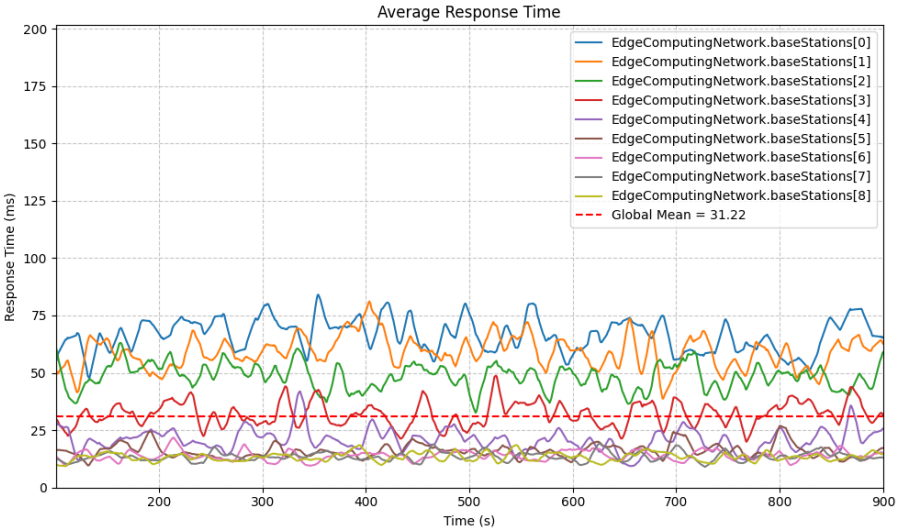

Figure 6.24: $E[R]$ for the uniform distribution

As we can observe, the performance of the system improves compared to method A only when dealing with the lognormal distribution. In the uniform case, we can see an higher response time because the load is already evenly distributed across the network and the method B add the forwarding delay. Conversely, in the lognormal case, the improvement is significant due to better balancing of the load that would otherwise remain unevenly distributed. This balancing effect enhances the overall efficiency of the network under the lognormal distribution.

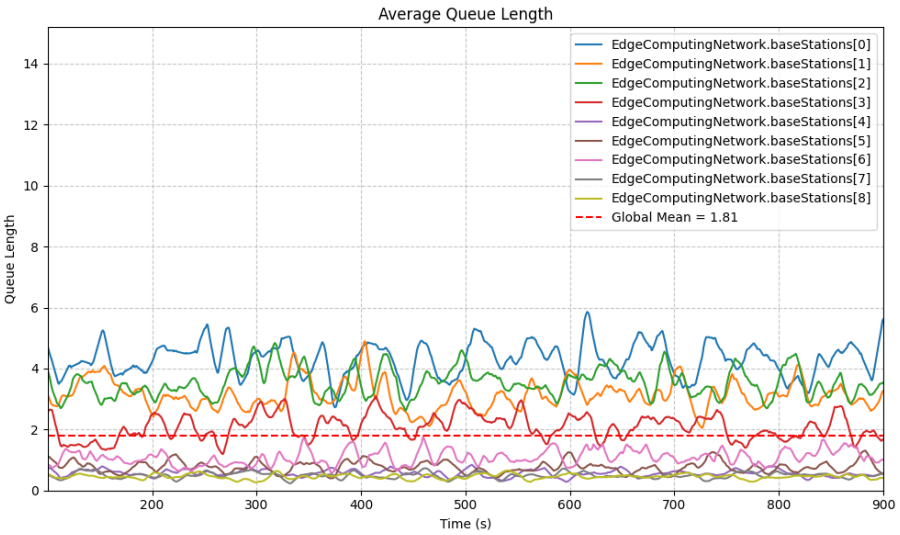

Figure 6.25: $E[qlen]$ for the lognormal distribution

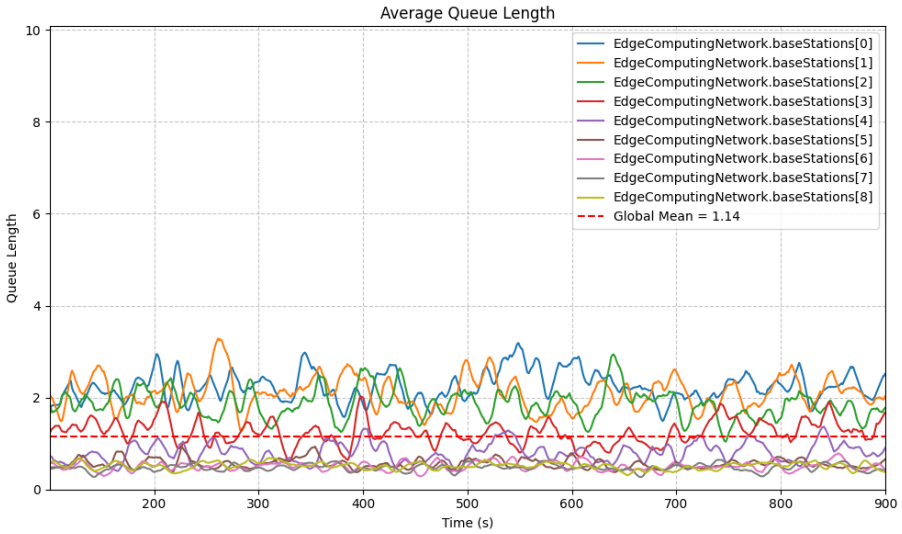

Figure 6.26: $E[qlen]$ for the uniform distribution

The same reasoning can be applied to the mean leangth of each queue, as the plots show the exact same behaviour.

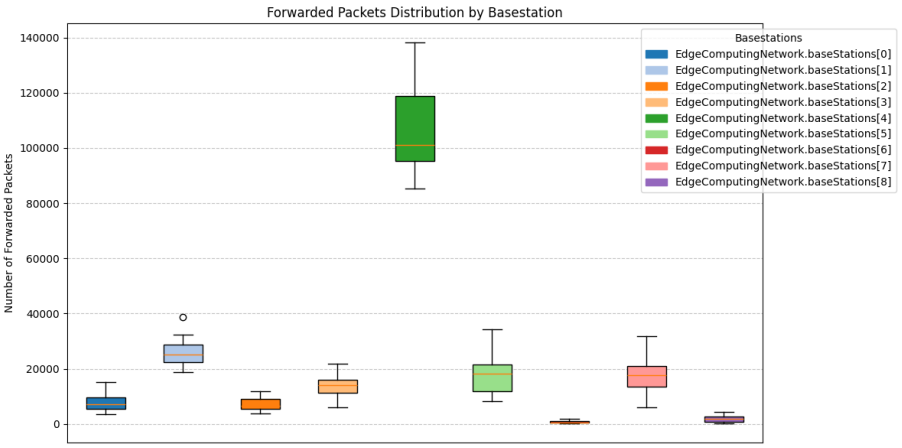

Figure 6.27: Forwarded packets distribution (lognormal)

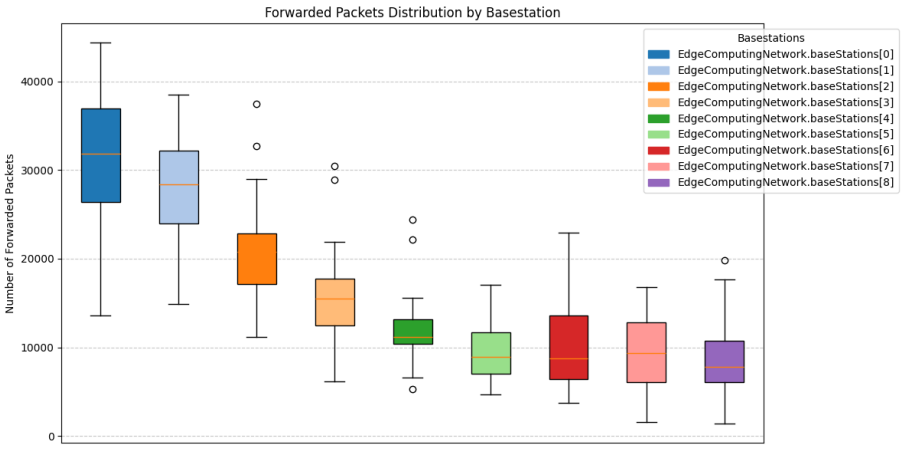

Figure 6.28: Forwarded packets distribution (uniform)

It is curious that in the lognormal case packets tend to be forwarded to one particular base station unlike the uniform case, where are forwarded more evenly. The lognormal case presents a higher number of forwarded packets (5.9×10^6 vs 4.3×10^6).

Dropped packets are not mentioned as there has been none throughout the experiment.

High size rate: $\mu = \frac{1}{10^4}$

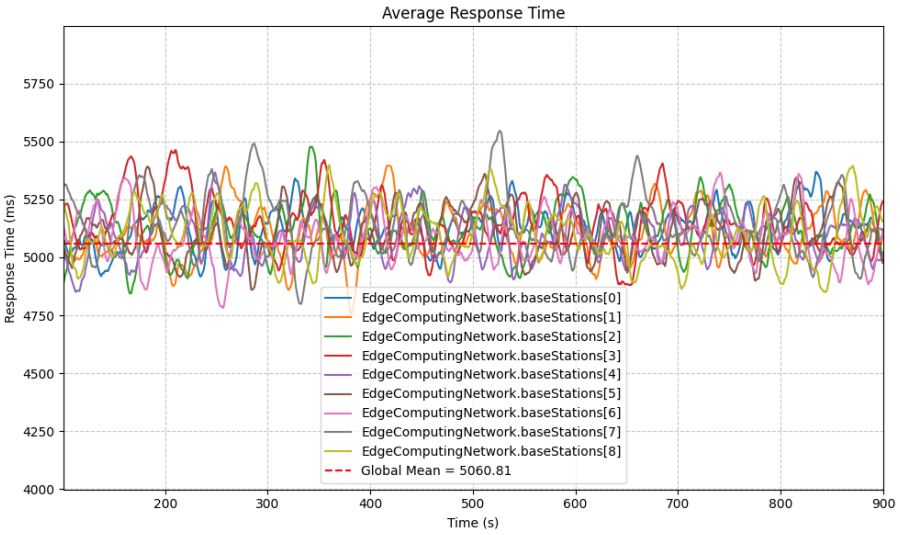

Figure 6.29: $E[R]$ for the lognormal distribution

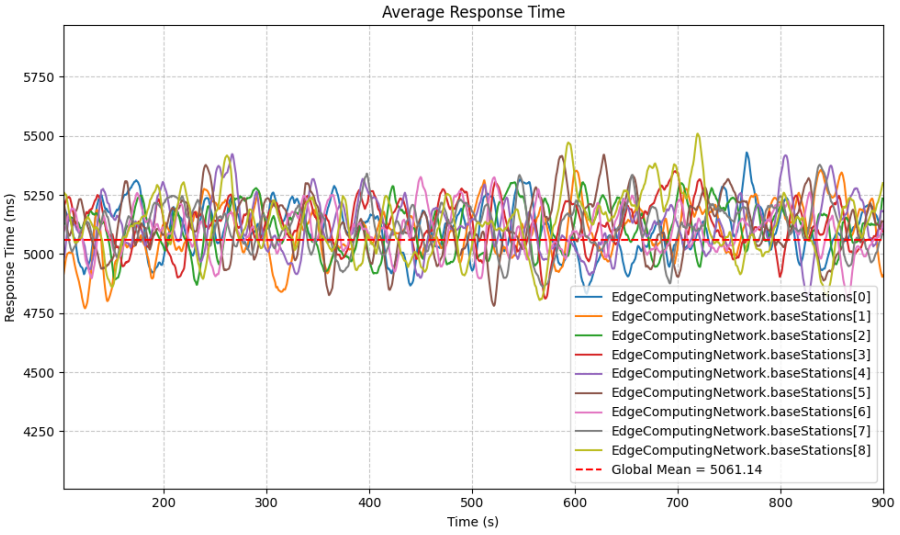

Figure 6.30: $E[R]$ for the uniform distribution

In this case, it is clear that the system is stressed out in both distributions, with almost equal mean response times. The queue is always full, leading to

a saturated response time. All the values of the basestations are concentrated around the global mean, which reflects the fact that all of them respond at nearly the same speed due to the saturation effect. As a result, the performance appears similar for both cases.

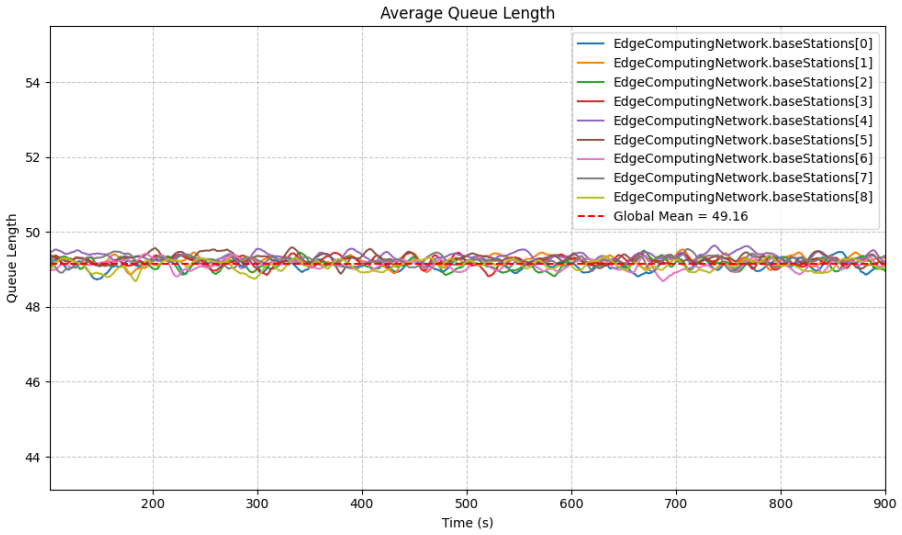

Figure 6.31: $E[qlen]$ for the lognormal distribution

The same reasoning can be applied to the mean leangth of each queue, as the plots show the exact same behaviour.

Since for both cases the plots are the same, it is shown the one relative to the lognormal distribution only.

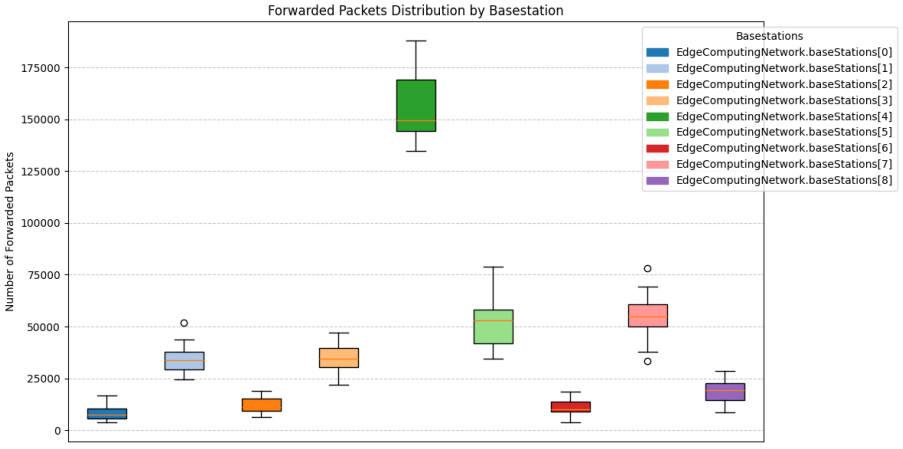

Figure 6.32: Forwarded packets distribution (lognormal)

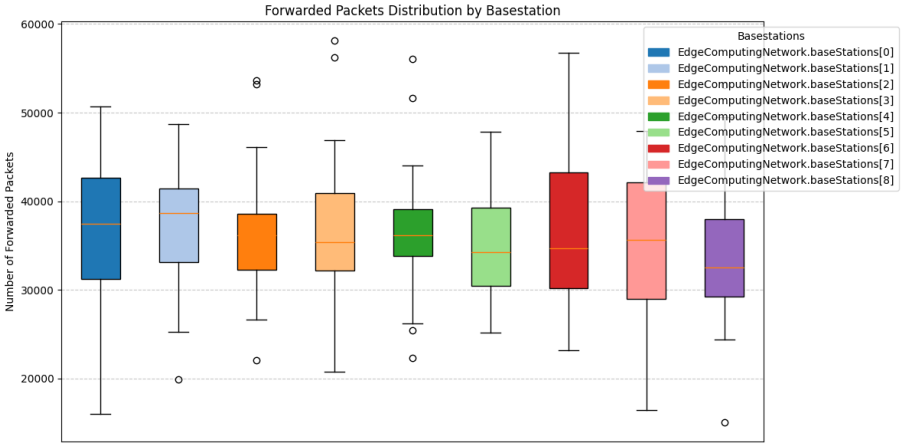

Figure 6.33: Forwarded packets distribution (uniform)

It is shown the same behaviour as in the method A, the only difference is that in the uniform case packets are forwarded more evenly among all the base stations.

We report a total of 9.7×10^6 forwarded packets for the uniform and 11.4×10^6 for the lognormal.

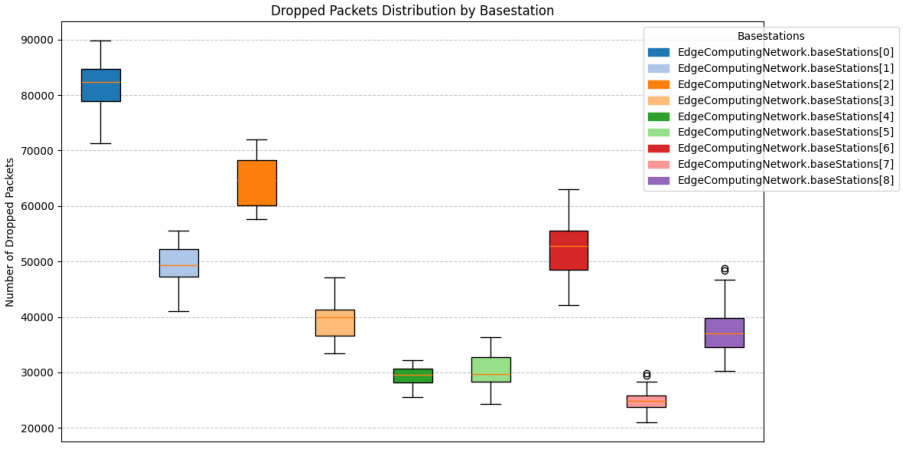

Figure 6.34: Dropped packets distribution (lognormal)

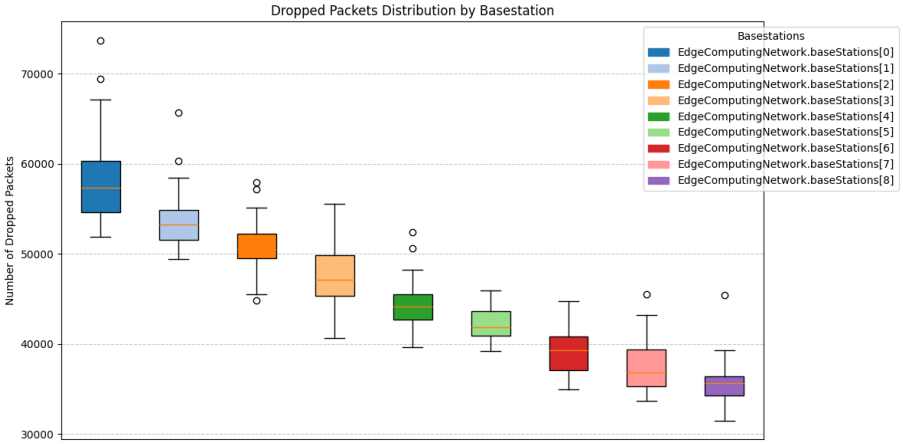

Figure 6.35: Dropped packets distribution (uniform)

As we can expect base stations that forward the majority of the packets are the ones that discard few of them and vice versa. We encounter this phenomenon for both the distributions.

We report the same number of discarded packets both for the lognormal and the uniform distribution.

6.4 The effect of varying the interarrival rate

In this section, we investigate the performance of the system while changing the interarrival rate, denoted by λ.

6.4.1 Interarrival rate variation – Method A

High interarrival rate: $\lambda = \frac{1}{0.1}$

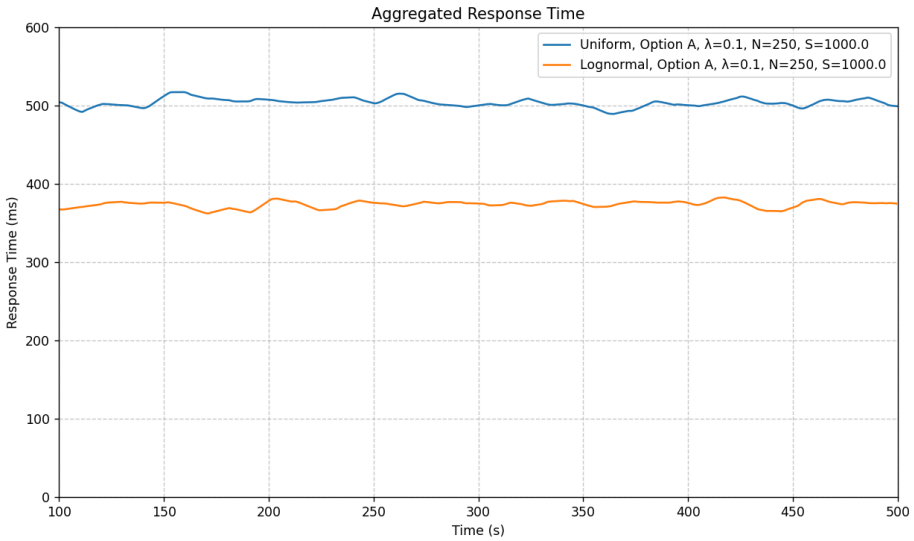

Figure 6.36: $E[R]$ comparison between Uniform and Lognormal distributions

The system experiences a significantly higher load with $I = \frac{1}{0.1}$, resulting in increased mean response times for both distributions. The *lognormal* case initially appears to perform better, as it exhibits a lower mean response time compared to the *uniform* distribution. However, this difference is misleading, as the response time saturates at 500 ms due to the limited queue size.

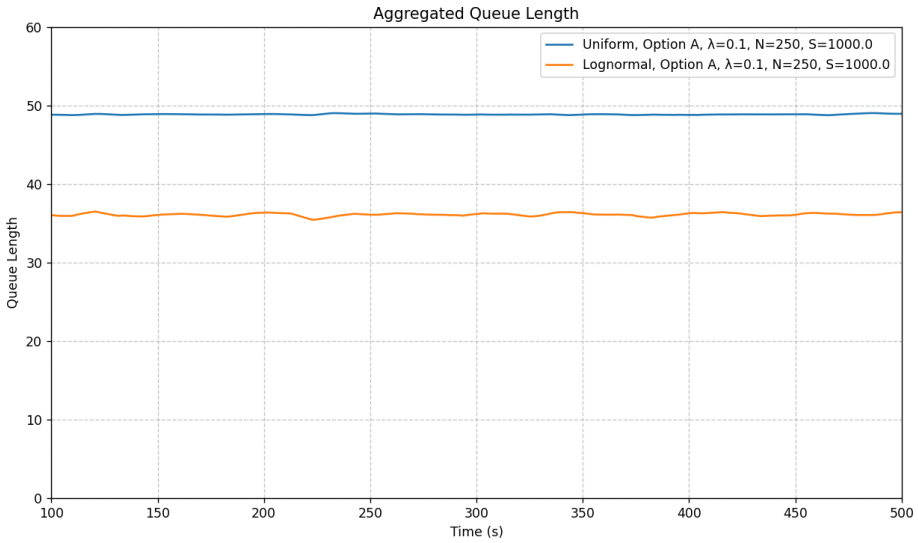

Figure 6.37: $E[qlen]$ comparison between Uniform and Lognormal distributions

Similarly, the mean queue length increases in both distributions due to the higher arrival rate. The *lognormal* distribution still shows a lower mean queue length compared to the *uniform* distribution, but this metric is also affected by the queue length saturation.

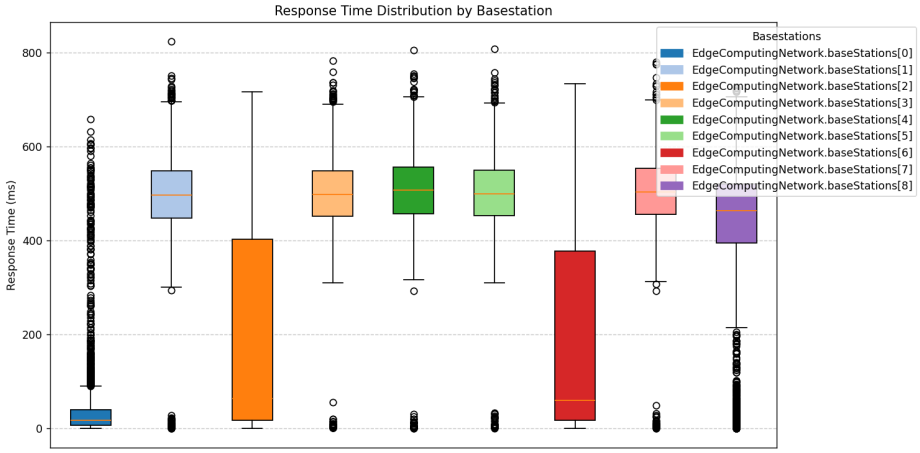

Figure 6.38: Response time for different basestation (Lognormal)

The boxplot analysis highlights the variability of the response time in the *lognormal* distribution, showing that some base stations experience heavy overloading while others remain underutilized. Consequently, the apparent

advantages of the *lognormal* distribution in terms of mean response time and queue length are invalidated, as they depend heavily on the saturation limit of the queue.

Moderate interarrival rate: $\lambda = \frac{1}{0.5}$

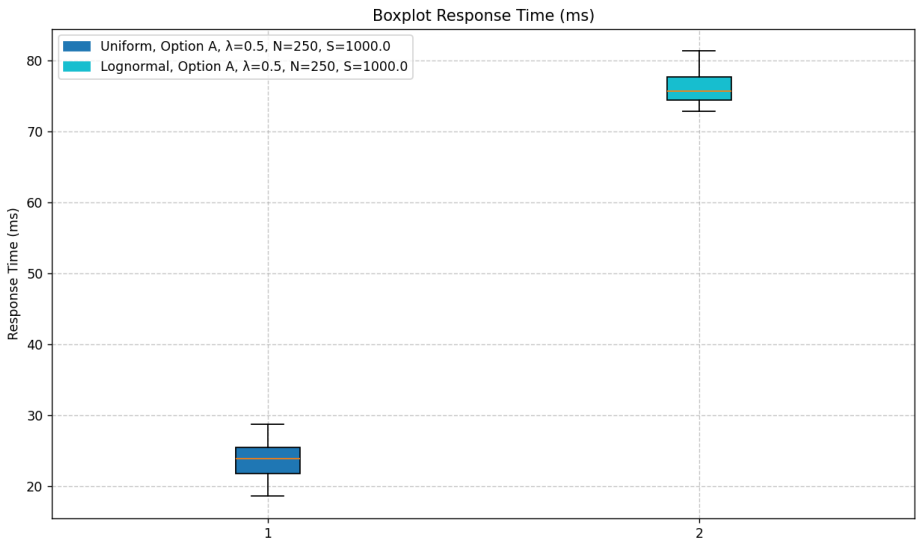

Figure 6.39: $E[R]$ comparison between uniform and lognormal distribution

In this experiment, the interarrival rate is moderate compared to the previous case, resulting in a less congested system. The comparison between the uniform and lognormal distributions shows that the uniform distribution yields better performance in terms of mean response time ($E[R]$). This is because the uniform distribution spreads arrivals more evenly over time, reducing the likelihood of sudden bursts that can temporarily increase congestion.

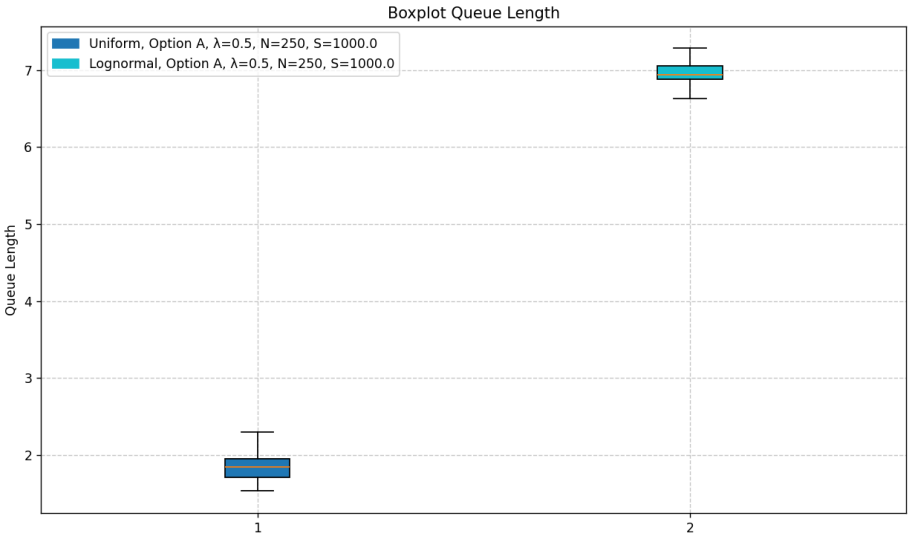

Figure 6.40: $E[qlen]$ for the lognormal distribution

The mean queue length ($E[qlen]$) remains relatively stable across the base stations, reflecting the moderate load of the system. Packet drops are infrequent in both distributions, as queues rarely reach their maximum capacity under these conditions. However, the uniform distribution exhibits a slight advantage by maintaining lower queue lengths overall. This is attributable to its more predictable arrival patterns, which mitigate the occurrence of sudden spikes in queue size.

6.4.2 Interarrival rate variation – Method B

High interarrival rate: $\lambda = \frac{1}{0.1}$

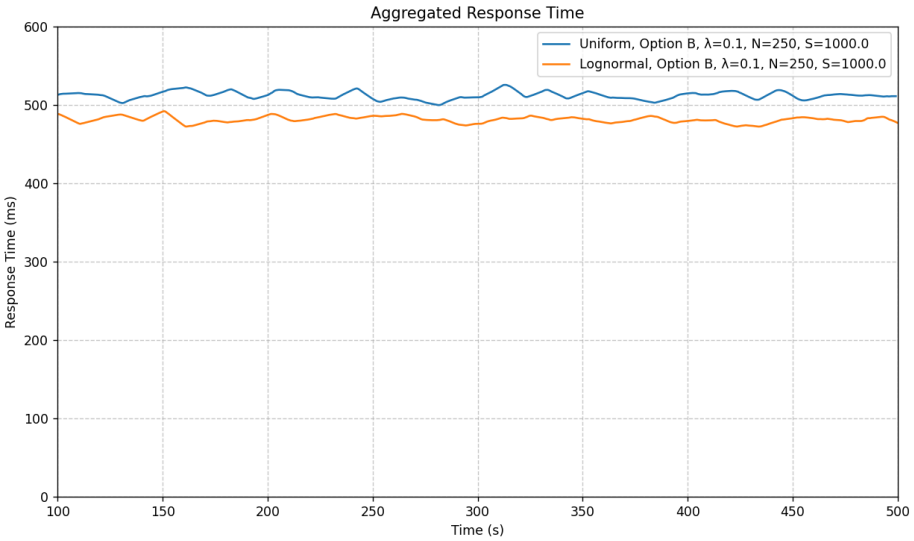

Figure 6.41: Comparison of $E[R]$ between uniform and lognormal distributions with Method B

Under Method B, the forwarding strategy effectively balances the system load, reducing congestion for both the uniform and lognormal distributions. As shown in the plot, the mean response time ($E[R]$) is very similar for the two distributions, with no significant differences. This demonstrates that the forwarding mechanism mitigates the effects of bursty arrivals typically associated with the lognormal distribution, aligning its performance closely with that of the uniform distribution.

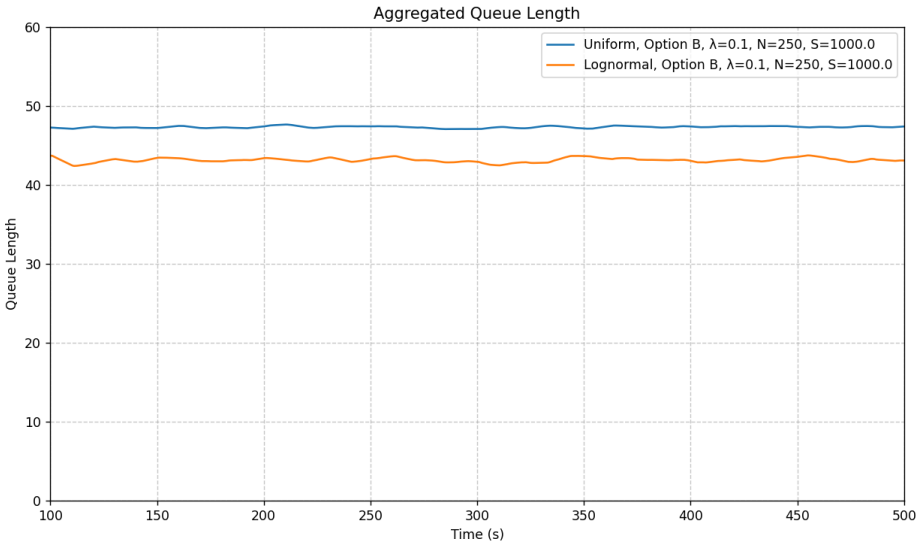

Figure 6.42: Comparison of $E[qlen]$ between uniform and lognormal distributions with Method B

The second plot illustrates the mean queue lengths ($E[qlen]$), which also show minimal differences between the uniform and lognormal distributions. The balanced load achieved through Method B ensures that queue lengths remain stable and comparable across both cases, preventing the queue-length imbalances observed in Method A.

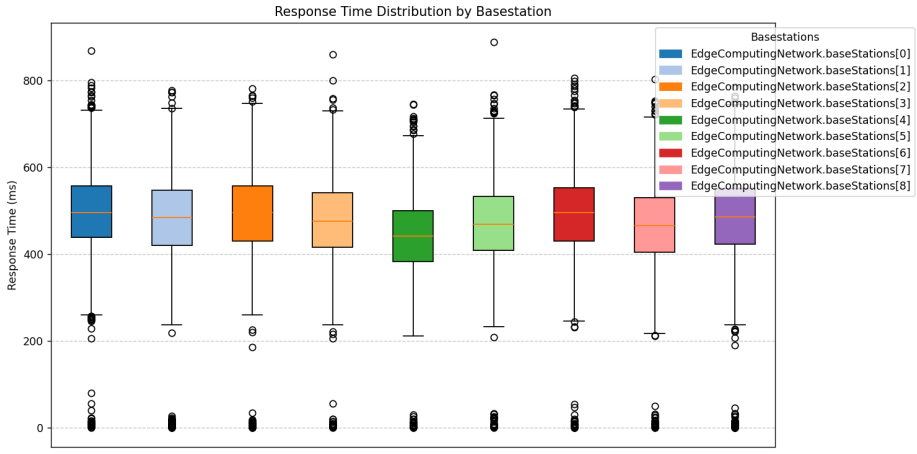

Figure 6.43: Packet forwarding distribution in the lognormal case with Method B

The third plot highlights the forwarding mechanism's impact on the lognormal

distribution. Packets are always directed to the less-loaded basestation, which efficiently handle the load and balance the packets into the network.

Moderate interarrival rate: $\lambda = \frac{1}{0.5}$

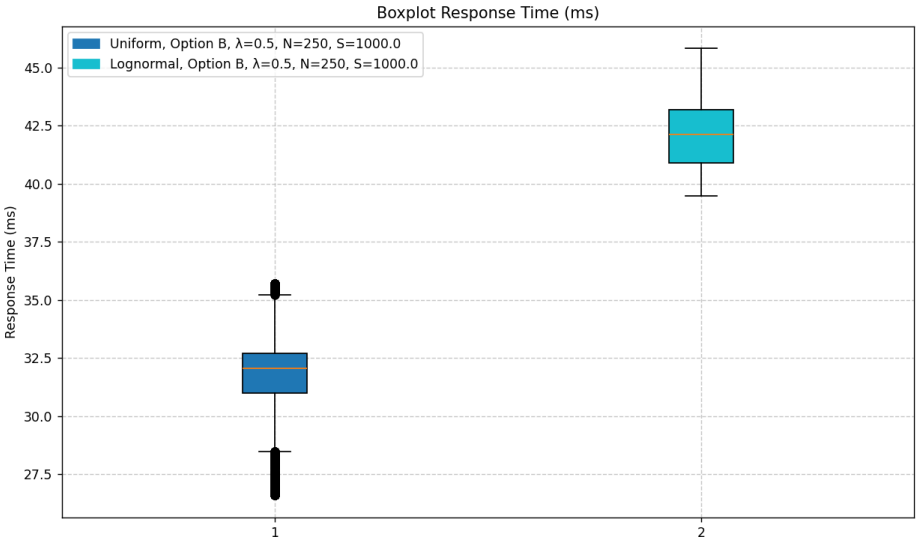

Figure 6.44: Comparison of $E[R]$ between uniform and lognormal distributions

The first plot shows the comparison of mean response times ($E[R]$) between the uniform and lognormal distributions under Method B. Here, the uniform distribution achieves slightly better performance, reflecting its ability to distribute the interarrival times more evenly and avoid response time spikes.

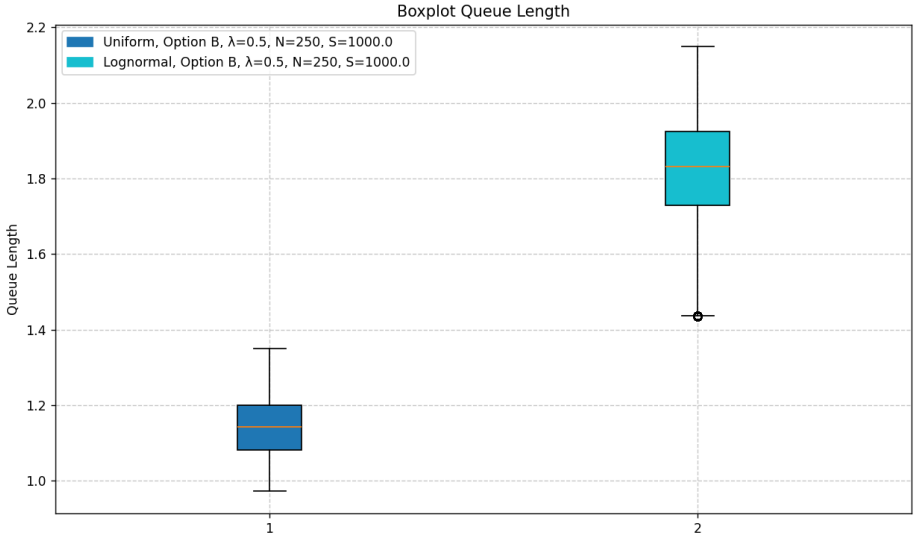

Figure 6.45: Comparison of $E[qlen]$ between uniform and lognormal distributions

The second plot illustrates the mean queue lengths ($E[qlen]$) for the two distributions. Similar to the response time results, the uniform distribution demonstrates slightly better performance, maintaining shorter queue lengths on average compared to the lognormal distribution.

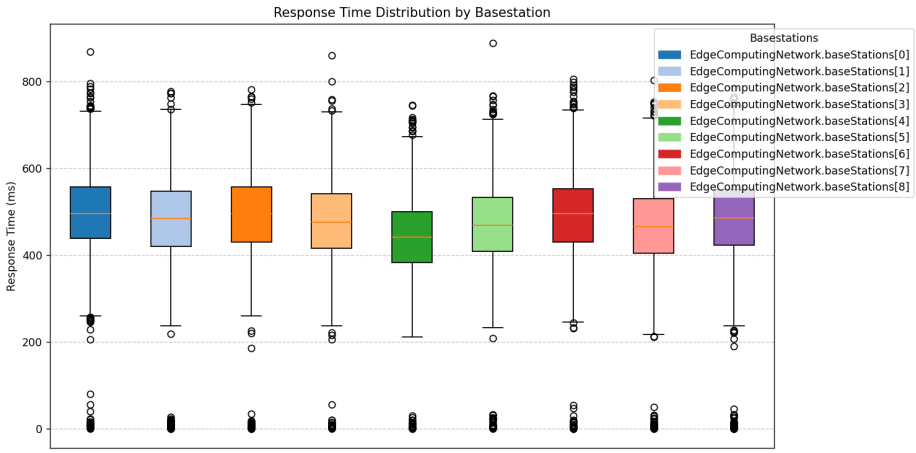

Figure 6.46: $E[R]$ balancing in the lognormal distribution with Method B

In the third plot, we observe that Method B effectively balances the mean response time in the lognormal case. The forwarding strategy ensures that

bursts of arrivals are handled more efficiently, preventing localized congestion and maintaining stable response times across the base stations.

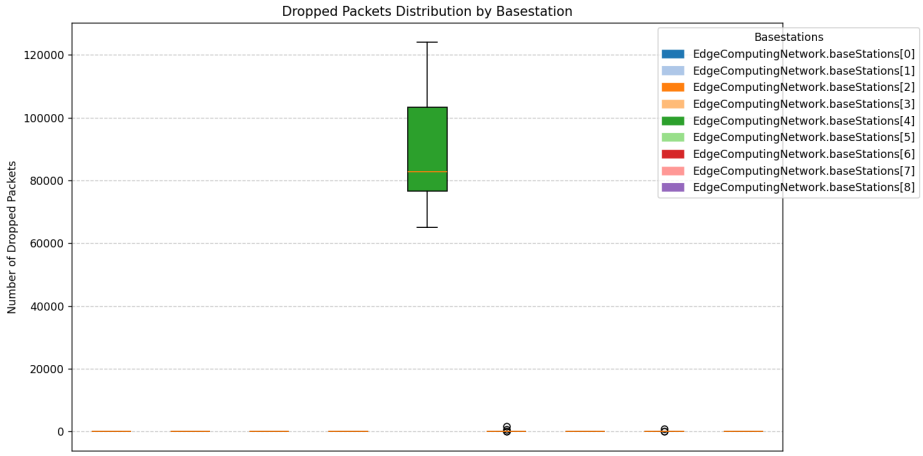

Figure 6.47: Dropped packets comparison for Method A

Finally, the fourth plot shows the mean number of dropped packets under Method A for individual base stations. While Method B, under a moderate interarrival rate, does not drop any packets due to its effective forwarding mechanism, Method A results in some base stations dropping packets. This is caused by the uneven load distribution within the network, which arises from the lognormal distribution of user arrivals.

Chapter 7

Conclusions

In this project, we evaluated the performance of an edge computing-enabled cellular network under two different user distributions (uniform and lognormal) and two service options (A: local service, B: service forwarding).

For the **uniform user distribution**, our results indicate that **Option A** is more efficient. Since users are already evenly distributed, forwarding requests to a less-loaded base station does not provide significant benefits. In fact, the additional latency introduced by the forwarding process in **Option B** results in increased response times. Consequently, when the user locations are uniformly spread, serving tasks locally (**Option A**) emerges as the preferable strategy.
Conversely, for the **lognormal distribution**, the load is highly imbalanced across the base stations when applying **Option A**. A few base stations receive a disproportionate share of requests, becoming overloaded, whereas others remain largely unused. The forwarding strategy of **Option B** alleviates this imbalance by distributing the workload more evenly, which substantially improves overall performance. As a result, under lognormal user distribution, **Option B** is the superior choice.
In addition to these findings, future research could explore hybrid strategies that combine local serving and forwarding in a more adaptive manner, potentially leveraging real-time network measurements. Furthermore, investigating more complex or realistic user distributions and refining the latency models for inter-base-station communication could provide deeper insights into the performance of edge computing-based systems in next-generation cellular networks.