Performance Analysis of Edge Computing for 5G and Internet of Things

Aigerim Ospanova, B. Eng. in Radioengineering, Electronics, and Telecommunications

Submitted in fulfilment of the requirements
for the degree of Master of Science
in Electrical and Computer Engineering

School of Engineering and Digital Sciences
Department of Electrical and Computer Engineering
Nazarbayev University

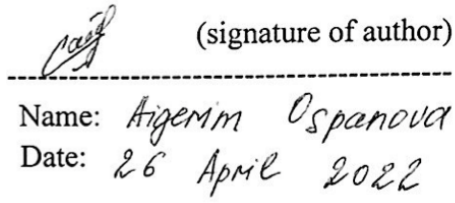
53 Kabanbay Batyr Avenue,
Nur-Sultan, Kazakhstan, 010000

Supervisors: Dr. Behrouz Maham, Dr. Aresh Dadlani

April 2022

Declaration

I hereby, declare that this manuscript, entitled *"title of thesis"*, is the result of my own work except for quotations and citations which have been duly acknowledged. I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.



Abstract

This thesis aims to explore edge computing paradigm for Internet of Things services and applications in the 5G era. Edge computing, with its processing and storage capabilities near end-users, can become a reasonable alternative to cloud computing. User device with constrained processing and storage capabilities offloads complex task to edge nodes for computing, and then, edge nodes transmit the outcomes back to the user. Though various requirements need to be met while deploying edge computing in smart applications, this work focuses on the most demanding and critical ones, such as latency and system reliability. It is crucial to minimize latency and enhance the reliability of the system in time-critical services, such as smart healthcare or transportation. Current research implements a single user – multiple edge nodes model with Rayleigh and Nakagami-m fading channels. It is worth noting that Nakagami-m fading channels are widely used in the fifth generation and beyond systems. In order to comprehensively investigate the topic of edge computing for Internet of Things, we have developed two different types of schemes. Specifically, selection schemes, in which user device offloads task to one edge node, and combining schemes, in which user device offloads task to several edge nodes concurrently. Concerning selection-based schemes, edge computing with cache-aided relay and cache-free relay are considered. In these schemes, offloading nodes are chosen based on the best computing capability, channel gain between user and edge node, or channel gain between the user device and relay node. Numerical results demonstrate that edge computing model with cache-free relay, where the best channel gain between the relay and edge node is selected, outperforms other models in system performance. At the same time, time-division multiple access, frequency-division multiple access, and capacity achieving schemes are introduced for combining case. Thus, it can be summarized that edge computing model with capacity achieving scheme demonstrates the highest results among the others in terms of the reliability metric. Moreover, performance analysis shows that system efficiency

can be affected by various parameters, such as transmit power, channel bandwidth, task size, latency threshold, number of edge nodes, and others. Numerical and simulation results are provided to validate analytical findings.

Acknowledgments

I would like to express my deep gratitude to my Lead Supervisor, Dr. Behrouz Maham, for his guidance and comprehensive explanations throughout the research process. Dr. Maham has provided me with valuable knowledge on the subject as well as motivation to learn more information related to my topic. His immediate feedback and detailed patient responses, despite the circumstances and time, allowed me to overcome challenges I encountered during the study process and greatly reduced my stress level.

Moreover, I would like to thank my Supervisor, Dr. Aresh Dadlani, and External Examiner, Dr. Sain Saginbekov, for their time and important advice on the content of the presentation. In addition, I am grateful to the Department of Electrical and Computer Engineering and the administrative staff at Nazarbayev University for the opportunity to pursue a Master's degree. With their great effort, we have obtained a solid education in the context of the pandemic.

Finally, I would like to thank my family and friends for their endless support and encouragement during all this time.

Table of Contents

Abstract 2
Acknowledgments 4
List of Abbreviations and Symbols 6

Chapter 1 – Introduction 7
1. 1 Background 7
1. 2 Aims and objectives 8
1. 3 Methodologies and techniques 9
1. 4 Thesis structure 10

Chapter 2 – Literature review 11
2. 1 Introduction 11
2. 2 Edge computing tools 12
2. 2. 1 Computation offloading 12
2. 2. 2 Resource allocation 16
2. 2. 3 Service caching 21
2. 3 Summary 24

Chapter 3 – Selection schemes for task offloading in edge computing 27
3. 1 Selection-based scheme for task offloading in edge computing under Rayleigh fading channels 27
3. 1. 1 System model 27
3. 1. 2 Performance analysis 29
3. 1. 3 Numerical results 32
3. 2 Selection-based scheme for task offloading in edge computing under Nakagami-m fading channels 34
3. 2. 1 System model 35
3. 2. 2 Performance analysis 37
3. 2. 3 Numerical results 41

Chapter 4 – Combining schemes for task offloading in edge computing 50
4. 1 TDMA model for task offloading 50
4. 1. 1 System model 50
4. 1. 2 Performance analysis 52
4. 1. 3 Numerical results 55
4. 2 FDMA model for task offloading 58
4. 2. 1 System model 58
4. 2. 2 Performance analysis 60
4. 2. 3 Numerical results 62
4. 3 Task offloading with capacity achieving codes 65
4. 3. 1 System model 66
4. 3. 2 Performance analysis 68
4. 3. 3 Numerical results 69

Chapter 5 – Conclusions 75
5. 1 Contributions 75
5. 2 Future work 76

Bibliography 77
Appendix A 81
Appendix B 87

List of Abbreviations and Symbols

AWGN	Additive white Gaussian noise
B	Task size
BCD	Block coordinate descent
c	speed of light
c_i	central processing unit cycle frequency
CDF	Cumulative distribution function
CDMA	Code-division multiple access
CPU	Central processing unit
d_i	distance between a user device and edge node
DDPG	Deep deterministic policy gradient
$E(.)$	Expectation
F	Fourier Transform
f_c	carrier frequency
FDMA	Frequency-division multiple access
g_i	channel gain
GenCOSCO	General cooperative service caching and computation offloading
ILP	Integer linear programming
IoT	Internet of Things
K	Cycle frequency for each bit of task execution
k	Euler's constant
m	Nakagami-m distribution parameter
MDP	Markov decision process
MEC	Mobile edge computing
MINLP	Mixed integral non-linear programming
N	Number of edge nodes
N_0	Spectral density of the noise
NOMA	Non-orthogonal multiple access
P	Transmit power
$Pr(.)$	Probability
PDF	Probability density function
QoS	Quality of services
R	Transmission rate
RV	Random variable
SC	Selection combining
SDMA	Space-division multiple access
SDN	Software-defined networking
SIC	Successive interference cancellation
SNR	Signal-to-noise ratio
SSC	Switching-and-stay combining
t_c	task computation time
$t_{tr,in}$	task transmission time
$t_{tr,out}$	outcome transmission time
TDMA	Time-division multiple access
UD	User device
URLLC	Ultra-reliable and low-latency communication
W	Channel bandwidth

Chapter 1 – Introduction

1. 1 Background

Internet of Things (IoT) concept enhanced various spheres of human life and industry. IoT represents the idea of connecting and managing heterogeneous things by means of Internet without human intervention [1]. IoT technology involves numerous services and applications, such as smart healthcare, transportation, power grid, industry, city, video surveillance, and many others. To support these advanced services with low latency, high reliability, and privacy requirements, novel technologies are engaged in the process of development [2]. Thus, 5G technology introduction for IoT services' deployment attracted the attention of many researchers in this area, as it can be a promising tool to ensure the aforementioned system properties. However, the functionality of IoT devices has considerable limitations, as they have scarce resources to solve complex problems. An elaborate paradigm such as cloud computing plays a vital role in IoT technology. Computational, storage, and communication capabilities for IoT devices are provided by central servers in cloud computing [3]. Nevertheless, the remote location of cloud data centers leads to inevitable drawbacks, specifically, undesirable latency, excessive bandwidth utilization due to the huge amount of generated data, lack of contextual information, privacy, and security vulnerabilities [3]. Latency is one of the major concerns in time-critical applications such as smart healthcare, smart vehicles, and others.

Edge computing is a new powerful technology that can solve these challenges. There are various types of edge computing technologies that differ in certain characteristics, the most common of them are cloudlets, mobile edge computing (MEC), and fog computing [3]. In edge computing, the computational capabilities are located closer to the end-users, hence the amount of redundant data transfer to central cloud servers is decreased [4]. It is noteworthy that edge computing is able to provide system requirements that are particularly useful for IoT applications, namely low latency, high reliability, improved privacy and security, context and

location awareness, lower energy consumption, and better bandwidth utilization [5]. Nevertheless, cloud and edge computing intend to complement each other, as both have certain advantages and disadvantages, thus it is unreasonable to substitute one with the other [4]. Although edge computing offers significant benefits for IoT applications, there are major challenges that need to be handled. Firstly, it is crucial to guarantee that a user receives computationally intensive tasks' outcomes on time after offloading it to the edge node for processing. Secondly, edge servers and end-users intend to maximize their benefits by getting the highest payment and the lowest price for resources respectively, consequently, the requests of both sides should be satisfied [6]. Finally, the number of IoT devices is increasing intensively, and as the result, the need for edge and cloud resources is also growing. Various economic and pricing methods can be used to address these open questions [7]. Nevertheless, edge resources are limited in comparison to cloud resources, therefore they should be allocated wisely. Consequently, the issues of optimal edge resource allocation, computation offloading, and service caching have become the subject of concern for many researchers.

1. 2 Aims and objectives

This research work aims to investigate edge computing paradigm, its characteristics, implementation methods, advantages, and challenges. Obviously, edge computing with its innovative tools of resource allocation, computation offloading, and service caching can ensure considerable benefits for smart technologies. As noted above, numerous techniques have been developed by scientists in this field to find the most efficient ways to apply these tools. There are various metrics such as latency, reliability, energy consumption, capacity, and scalability to assess their performance. However, the issue of effective edge computing deployment for IoT technology remains open, since new advanced practices and schemes, with corresponding benefits and drawbacks, are rapidly developing in this area.

In our research work, we implement different edge computing models and analyze how computation offloading efficiency in edge computing can be improved in terms of the reliability metric. We believe that using the reliability metric in the context of the delay-outage probability is beneficial for IoT scenarios, since these applications tend to be time-critical, and therefore, edge servers must fulfill tasks within a strict time frame. Moreover, the influence of different parameters on the system performance is explored in this work.

1. 3 Methodologies and techniques

We design edge computing model, in which a constrained IoT device offloads its complex task to edge nodes with sufficient computing and storage capabilities to process, and thus, time of task completion is reduced. Selection and combining schemes are proposed for a task offloading in edge computing system with a single user device – multiple edge nodes. In selection schemes, the user device offloads its task to the single most suitable edge node, whereas, in combining schemes, multiple edge nodes are used for this purpose. For the selection-based case, we consider edge computing with cache-aided relay and cache-free relay. Moreover, five selection criteria are derived to find the most effective scheme. For the combining-based case, we introduce time-division multiple access (TDMA), frequency-division multiple access (FDMA), and capacity achieving schemes. We evaluate proposed schemes' performance according to the reliability metric, which is represented by the probability that transmit and computation time of task offloading will exceed the established latency threshold. Then, we derive a closed-form solution for the delay-outage probability using probability density function (PDF) and cumulative distribution function (CDF). In addition, in both cases, we adjust system parameters, such as signal-to-noise ratio (SNR), transmit power, channel bandwidth, task size, number of edge nodes, and latency threshold. Thus, we can observe the system performance dependency on these values. Finally, analytical assumptions are verified by numerical outcomes.

For the literature review, we use reliable digital libraries, such as Elsevier, IEEE Xplore Digital Library, Scopus, and others. We also utilize Wolfram Mathematica software for performance evaluation parts, and MATLAB software for the simulations.

1. 4 Thesis structure

This research work is designed in the following manner. In Chapter 1, a schematic description of edge computing in IoT is recommended to help illustrate the main objectives of the proposed work. Chapter 2 represents a detailed literature review on strategies proposed by researchers in this area. Moreover, a description of the most common metrics for performance evaluation is considered in this chapter. Chapter 3 is devoted to selection schemes in edge computing with Rayleigh fading channels and Nakagami-m fading channels. This chapter includes system model, performance evaluation, and numerical results subsections. Chapter 4 is dedicated to combining schemes with Rayleigh fading channels and Nakagami-m fading channels. In particular, Section 4.1 examines TDMA scheme, Section 4.2 studies FDMA scheme, and Section 4.3 analyses capacity achieving scheme. All of these sections also include system model, performance evaluation, and numerical results subsections. Chapter 5, summarizes findings, identifies research gaps and limitations in research, and determines directions for future investigation.

Chapter 2 – Literature review

2. 1 Introduction

A significant advantage of edge computing over cloud computing is that the computing capabilities of the former are closer to the end-users. Therefore, the amount of redundant data transfer to the remote central server is reduced, which is not the case for the latter [4]. Being one of the main features of IoT technology, low latency in edge computing is based on its simplicity and less propagation delay due to the proximity of edge nodes to the customers [5]. At the same time, energy consumption in IoT devices is reduced through offloading complex tasks to edge nodes. Regarding context and location awareness, users can obtain information about desired services and products in accordance with their location and requests. Security and confidentiality of valuable data are also enhanced since there is no need to transfer data vulnerable to Internet attacks to central servers [5].

Despite the significant advantages offered by edge computing, there are still serious challenges for its deployment in IoT applications. One of the main concerns is the lack of incentive for edge node owners to contribute their limited resources, such as storage, computational capability, and energy, to the IoT system [8]. Different economic and pricing methods can solve this question, specifically, market-based, system utility gain, or auction and game theoretic pricings [7]. Conventional auction mechanisms, Vickrey, Vickrey-Clarke-Gloves (VCG), combinatorial, or double auctions have a great potential in this area. In addition, optimal edge resource allocation, computation offloading, and service caching is other concerning issue.

Thus, this literature review investigates strategies proposed by the authors for different edge computing architectures, and their performance evaluation metrics.

2. 2 Edge computing tools

2. 2. 1 Computation offloading

As noted above, computation offloading is a promising tool provided by edge technology to empower IoT services and applications. In particular, IoT devices with constrained capabilities offload complex assignments to nearby edge servers, after completion edge nodes transfer outcomes back to the users. Hence, offloading to edge nodes brings remarkable advantages to smart devices, such as increased battery life, reduced delay, and the ability to process multiple applications concurrently on a single device [9].

Latency is one of the most demanding features and is discussed in several research works, for instance [10] and [11]. The authors of these articles investigated the issue of computation offloading in a system with multiple user devices and edge nodes. Specifically, the researchers in [10] suggested a new approach of serverless computing that included both a traditional solution and a novel one with lambda function. Thus, in this new distributed design, users send their demands to the dispatcher, then the dispatcher chooses the most appropriate node for task completion. After completion, the outcome is sent back by the dispatcher to the users in an online mode. This approach demonstrated an advantage over traditional and online methods in guaranteeing less latency, increased scalability, and system reliability. Similarly, the authors of [11] intended to decrease system latency when offloading tasks to edge servers. They also argued that choosing the best edge node can contribute significantly to finding a feasible solution. However, in contrast to [11], the authors of [12] emphasized the necessity to take into account edge nodes' residual bandwidth, as well as users' placement while selecting offloading nodes in dynamic systems. Thus, the researchers formulated Markov decision process (MDP) problem, which included mentioned aspects, and solved it by means of an iterative approach. Exhaustive simulations showed that the developed model can outperform existing baseline approaches.

Another important metric for evaluating IoT services and applications with edge aid is energy consumption. For instance, [12] explored the issue of computation offloading in order to reduce energy expenditure. The authors implemented the strategy of dynamic partial offloading in their work, in which tasks can be performed by different entities simultaneously, based on their states. To embody a partial offloading strategy, the researchers utilized the integer linear programming (ILP) approach. Also, unlike other works, [12] considered energy harvesting user equipment and measured its power consumption during the experiment. At the same time, the authors of [13] were concerned not only with energy consumption but also with the latency criterion. According to the researchers, it is necessary to take into account the remaining battery power of user devices, in order to obtain better offloading results at these two metrics. To achieve the set goal, they stated mixed integral non-linear programming (MINLP) problem. Likewise [11], the authors of [13] used an iterative search mechanism to solve the problem, however, they included a penalty aspect in it. The researchers in [14] also claimed that finding a trade-off between latency constraint and energy consumption is crucial in cooperative edge computing. They formulated MINLP problem for this purpose and proposed to replace decision variables with real ones to alleviative its complexity. In addition, the researchers introduced branch and bound mechanism, as well as interior point mechanism to solve the specified problem. The work [14] demonstrated promising results in satisfying two critical performance metrics. Regarding optimal task offloading in edge computing, [15] also studied energy and latency trade-offs, however, unlike the works [12] – [14], the researchers focused on partial task offloading in a single device – single edge node model. Thus, as in [12], the authors partitioned computationally intensive assignment into several pieces to solve it simultaneously on the user device and edge server. To handle the decomposed problem, they used a block coordinate descent (BCD) approach, which outperformed the existing exhaustive algorithm in Monte Carlo simulations.

Reliability is also a critical measurement used for the performance evaluation of the suggested schemes. In many research works, reliability is utilized in the context of the delay-outage probability. It is important to guarantee high reliability, as well as low latency in time-critical IoT applications, such as smart transportation, industry, or smart healthcare [5]. For instance, collaborative MEC vehicles and stationary edge nodes were introduced in [16] for partial task offloading. Software-defined networking (SDN) enables the simultaneous functionality of the diverse edge computing entities here. It is important to note that SDN represents the idea of network softwarization, i.e., conducting processes on the software and dynamic reconfiguring the network architecture under certain conditions [17]. The researchers in [16] argued that their strategy of using heuristic algorithms can enhance the reliability of edge systems under latency constraints. However, they admitted that their mechanism needed further improvements, such as opportunistic node selection. Thus, many research papers, such as [18] – [21], focused on opportunistic node choosing for outage probability minimization. For instance, the authors of [18] considered relay nodes with embedded edge computing to assist user devices. The researchers used Rayleigh fading channels for the system modeling and determined its reliability under latency constraints. Thus, they developed an effective algorithm for the best relay node choosing, which showed promising results in reliability and diversity order improvements. The authors of [19] investigated the case of one user device and several processing access points. To determine the best access point for the partial task offloading, they developed two criteria for choosing: according to the processing capacity of the access point and according to channel characteristics. Moreover, the optimal size of the tasks for unloading was determined by the researchers. The performed estimations and modeling demonstrated the effectiveness of the selection schemes proposed in [19]. Inspired by [19], the authors of [20] also considered several processing access points for computation offloading, however, they extended the work in [19] and utilized multiple antennas for this aim.

In addition, Rayleigh fading channels were used in the former, while Nakagami-m fading channels were utilized in the latter. To gain better results in edge computing with relaying, the authors of [20] proposed two efficient techniques such selection combining (SC) and switching-and-stay combining (SSC), which also demonstrated effectiveness in a set goal. At the same time, [21] investigated total error probability under the reliability metric. The authors studied the subject of ultra-reliable and low-latency communication (URLLC) for MEC systems with several edge servers. Thus, to design URLLC for edge computing with task offloading the researchers formulated an appropriate edge node selection question, divided it into several sub-tasks, and solved them by means of a deep deterministic policy gradient (DDPG) approach. Interestingly, only the reference work [16] followed a many-to-many architecture, while the articles [18] – [21] used a model with one user device and several edge or relay nodes.

Capacity is another important feature of optimal computation offloading in edge computing. This metric characterizes the ability of the system to function under dynamic conditions when heterogeneous entities and their numbers change during the process of offloading [22]. Several research papers are considered in this literature review to obtain comprehension of this subject. First, in [23], the authors explored a novel approach of simultaneous task offloading to an edge server and device-to-device technology. According to the researchers, the problem of capacity maximization in such a scenario is a MINLP, so advanced techniques need to be introduced. To begin with, efficient energy distribution and partial computation offloading model was formulated to optimize the utilization of edge resources, then an exhaustive search algorithm was implemented to increase the processing capability of the entire system. The simulation results revealed that the suggested by the authors strategy can outperform some baseline approaches. The issue of capacity improvement from the minimized cost perspective was investigated in [24], however, in contrast to [23], the

researchers considered a cooperative cloud-edge system for task offloading here. This means that the tasks are offloaded from smart devices not only to edge or cloud servers but also between neighboring edge nodes. Then, as in [23], MINLP question was formulated and solved by means of the branch and bound approach. Modeling demonstrated the efficiency of the proposed scheme, in terms of a sufficient reduction in the service cost. Since the system's ability to function in unforeseen conditions is represented not only by the capacity metric but also by security and privacy measurements, it is important to examine these metrics as well. For example, another important IoT service, such as smart city, was considered in [25] with security and privacy enhancement goals. As the authors stated, it is challenging to guarantee privacy and security aspects in such cooperative systems, therefore, it is necessary to implement advanced tools. For this purpose, they delivered a strategy consisting of the following steps: determining the expected waiting time by a queueing mechanism, developing a smart offloading approach, simulating the proposed model, and comparing it with benchmark schemes. Simulation outcomes revealed the advantages of the proposed strategy; thus, the researchers intended to implement it in real-life conditions in the future. It can be seen that all these articles [23] – [25] correspond to one of the most common models, i.e., the many-to-many model.

2. 2. 2 Resource allocation

Edge computing with resource allocation is also used to improve the performance of IoT and 5G systems. Although many studies deal with the concepts of computation offloading and resource allocation together, some articles consider them separately. Thus, the same performance metrics can be implemented for the assessment of the developed strategies.

In [26], the authors attempted to decrease the latency during edge resource allocation and computation offloading in multiple users – single edge node model. In this case, numerous IoT devices contain a considerable amount of data that needs to be processed at the edge server,

so edge resources should be distributed wisely among users to reduce total latency. The researchers implemented partial computation offloading, multiple access, problem decomposition, and sub-gradient approach to obtain effective resource allocation with reduced users' total latency. This strategy is applicable for the scenario of scarce computation and sufficient communication capabilities. Resource allocation and computation offloading with reduced latency were also investigated in [27], however, the researchers considered fog computing in this case. The authors formulated an optimization problem, which included queuing and partial computation offloading concepts. This problem turned out to be MINLP, therefore the researchers used a hybrid genetic method to handle divided problems. The modeling revealed that the proposed approach can achieve high results in latency minimization under energy consumption constraints. Moreover, the authors intended to extend their work for multitask scenario in real-life services. At the same time, [28] considered edge computing resource allocation for a model with multiple users – multiple edge nodes, which is not the case for [26], [27], and [29]. Thus, complex tasks from user devices were simultaneously offloaded to several edge nodes for more reasonable resource utilization in [28]. To handle the mobility issue of the user devices, the authors suggested offloading tasks to edge nodes located in small areas. Further, an optimization question was defined by the researchers to obtain high reliability and low latency, which was then solved by utilizing the deep reinforcement method. Thus, this strategy turned out to be feasible for dynamic systems with limited latency requirements. The researchers of [29] developed a special approach, in which user devices were motivated to offload their tasks to MEC node, whereas MEC node provided power to the user devices as compensation. Moreover, the researchers were concerned with utility maximization and power usage reduction questions under limited latency. Thus, to fulfill this complex task, they used an iterative mechanism and Lagrangian approach, which allowed them to gain significant results in motivating the entities to participate in the process.

It was mentioned above, that in IoT and 5G systems equipped with edge computing, there is an inevitable trade-off between several desirable features. One of the most popular compromises is between energy consumption and latency constraint. For example, [30] considered the latency and energy consumption trade-off in edge computing with resource allocation and task offloading. In their work, the authors combined two cutting-edge concepts, such as MEC and cloud radio access network in IoT context. For this aim, they used network function virtualization. Based on the researchers' assumptions, it is challenging to handle this problem in its initial form. Therefore, they implemented Lyapunov approach to partition complex question into several small tasks and addressed them by matching game mechanism. Similar to [30], the authors of [31] investigated energy consumption reduction under reliability and latency requirements. The researchers analyzed the queueing pattern in their model using extreme value theory, as this pattern can influence on delay metric. They also used Lyapunov stochastic method to design computation offloading and resource distribution model for this system. Thus, these two works provided valuable insights into improving the performance of edge computing and demonstrated high potential in comparison to benchmark algorithms. It can be seen that both works [30] and [31] used many-to-many model for edge computing architecture in IoT systems. The researchers in [32] and [33] were mainly concerned with reduced energy consumption demand in edge-assisted IoT services and applications. These works attempted to efficiently allocate edge resources; however, the authors used different approaches in achieving the set goal. Namely, the work [32] efforted to reduce the transmission time and energy cost in resource allocation and task offloading, which turned out to be a MINLP question. Consequently, the authors divided it into two subproblems and solved them by an iterative method, also used in [11] and [13]. To confirm the high potential of the proposed mechanism, they performed a Monte-Carlo simulation. Thus, their approach is rather promising for the considered conditions and can outperform benchmark algorithms. Regarding

[33], the authors of this article also focused on the energy consumption aspect in edge computing. Thus, they highlighted the effectiveness of cooperative edge computing for dynamic systems. To raise the question of the simultaneous task processing by different entities for the total energy consumption decrease, it was proposed to use the MDP approach. According to the authors, MDP is a helpful tool when designing this kind of problem with stochastic processes and numerous conditions. Afterward, by using a stochastic optimization policy, the authors demonstrated that cooperative processing can significantly reduce system energy consumption. The simulation outcomes approved the correctness of the numerical results and demonstrated the effectiveness of the strategy. Evidently, the work [32] used a many-to-many model, while the work [33] considered a model with a single user device and several access points. Nevertheless, the authors of [33] claimed that a model with multiple user devices can be examined in the future.

The capacity metric for resource allocation evaluation has also attracted the attention of several academic articles, [34] and [35] are two of them. A multi-user and single edge node model was considered in these works to represent IoT systems with edge computing. In particular, [34] considered capacity measurement in terms of the quality of services (QoS) aspect. Thus, to enhance this parameter during resource allocation, the researchers emphasized that communication and computation queueing patterns should be taken into account. Similarly, the question of the appropriate queueing was considered in [25] and [31]. However, in [34], the distribution of the computation and communication edge resources was investigated with the aim of QoS improvement under latency limitations. This is a complex task and needs to be divided into small subtasks through an alternating direction algorithm. After thorough experiments, it was found that by efficient regulation of latency bound and allocated communication and processing resources, system functionality can be improved sufficiently. At the same time, [35] differs from [34] in several fundamental aspects. First, in [35], a single

user-single edge server and many-to-many architectures were implemented. Then, the capacity metric was defined as the edge computing resources available for user devices in a limited time. Thus, the researchers aimed to maximize the capacity of the system by appropriate resource distribution and partial computation offloading. Two scenarios were considered in [35], namely, with one user device and multiple user devices. Solution for the efficient resource allocation was given in the first scenario, furthermore, suboptimal, and water-filling approaches were derived to address the capacity maximization problem in the second scenario. Finally, Monte-Carlo simulations confirmed the analytical calculations of the researchers.

Regarding other critical performance metrics, such as system utility maximization, several research papers need to be reviewed. For instance, the research articles [29] and [36] developed advanced mechanisms to improve system utilization during computation offloading and resource allocation in edge computing. Thus, a many-to-many model was discussed in [29], whereas a single edge node – multiple user devices model was applied in [36]. A stochastic optimization question was stated by the authors of [36] to obtain proper results. Furthermore, similar to the authors of [30] and [31], the researchers used Lyapunov optimization approach to decouple this sophisticated question into small tasks, which then were solved directly or through advanced algorithms. Finally, the outcomes of these tasks were utilized to address the initial problem by means of the latency-aware resource distribution approach. Through simulations, the researchers demonstrated that higher values of utility metric under latency constraint can be achieved by their approach, in contrast to the heuristic algorithm. At the same time, the researchers in [37], made an effort to maximize the utility and satisfy the security and privacy requirements of edge technology in 5G scenario. This work included four basic stages in achieving the set goal: formulating a question with several objectives (e.g., optimal resource allocation), determining a balanced uploading mechanism, solving a problem with several objectives through advanced algorithms, and performing a

detailed simulation to validate analytical assumptions. Thus, this work provided useful insights according to optimal resource allocation with utility maximization aim in multiple user devices – multiple edge nodes model.

2. 2. 3 Service caching

Edge computing equipped with caching technology can facilitate IoT services and applications remarkably due to its specific architecture. According to [38], the rapid growth of multimedia content has led to the need for a more expensive and sophisticated network configuration, which is not appropriate in certain scenarios. In such a situation, caching becomes a helpful tool for satisfying user requests, while reducing the network load. The idea of caching is to store the necessary data at the edge of the network and provide it to the requesters when the backhaul channels are overloaded [38]. Thus, the integration of edge computing with caching has attracted the attention of many researchers in this area. In the vast body of articles about this paradigm, researchers were trying to find new ways to optimize it. This literature review discusses some of the research papers, such as [22], and [39] – [41], that are relevant to the research question.

The issue of latency minimization also plays a critical role in edge computing with caching. For example, the authors of [39] considered the question of caching, resource distribution, and assignment offloading collaboratively with latency reduction aim. The authors argued that IoT with edge computing aid can greatly benefit from task offloading to the appropriate node, rational edge resource allocation, and flexible caching based on user demands. They assume that this is a MINLP issue, therefore, handling it in its original form is challenging. To relieve the complexity of the problem, the researchers divided it into small tasks by means of auxiliary variables. Thereafter, they introduced advanced algorithm, such as general cooperative service caching and computation offloading (GenCOSCO), to solve divided problems. The modeling performed by the researchers showed that this strategy can

contribute to sufficient system utilization and minimized latency. The authors of [40] also studied the issue of edge computing with caching for IoT and 5G technologies' aid, however, they were more interested in the reliability metric under a time constraint. Likewise [18] – [20], relay aid was also involved in this system architecture. Thus, the researchers investigated the question of choosing the best edge node to improve system reliability. For this objective, they proposed to use two criteria for the model with the cache-assistance: according to the processing capability of the edge node and characteristics of the relay channel. For the cache-free model, they suggested utilizing three criteria: according to the edge node processing capability and characteristics of the direct and relay channels. Analytical and numerical results demonstrated delay-outage probability dependence on various criteria and system parameters. Eventually, the researchers aimed to expand their work by using deep and reinforced learning methods in the future.

In [22] and [41], the authors argued that capacity metric should also be used for performance evaluation of service caching in edge computing. Both works [22] and [41] investigated the issue of capacity maximization for a many-to-many model in dynamic systems with limited latency. However, according to the authors of [22] and [41], capacity measure in cache-aided edge computing differs from the traditional representation and depends on the popularity of the files, system states, and diversity of the entities. In [41], the authors admit that collaboration of edge servers can significantly increase capacity, however, the system budget needs to be taken into account. Thus, they changed the capacity optimization question to a transfer link reduction question. For this scenario, the researchers assumed that edge server distribution is based on the Poisson point process and used Lagrangian approach for the solution. The work [22] extended the research in [41] by including the file degree concept in the capacity maximization problem. A bipartite graph was used by the researchers to demonstrate the bonds between user devices and cached data. Then, transfer length, file query

probability, and file degree based on the user's dynamics were determined here. The researchers also used Lagrangian multiplier approach for the solution in this work. Modeling showed considerable results in achieving the set goal; however, the authors acknowledged that this study should be extended to other scenarios, for instance, congestion conditions [22]. Similar to [22], [39], and [41] considered a model with multiple user devices and edge nodes, while [40] focused on a model with one user device – several edge nodes.

Table 2.1 synthesizes considered works by implemented edge computing tools and performance metrics.

Table 2.1: Edge computing tools for IoT and 5G assistance.

Ref.	Edge computing tool	Performance metric	System model	Strategy
[10]	Computation offloading	Latency	Multiple UD – multiple EN	Serverless computing with lambda function
[11]	Computation offloading	Latency	Multiple UD – multiple EN	Value iteration algorithm
[12]	Computation offloading	Energy consumption	Multiple UD – multiple EN	ILP approach
[13]	Computation offloading, resource allocation	Latency, energy consumption	Multiple UD – multiple EN	Iterative search algorithm
[14]	Computation offloading, resource allocation	Latency, energy consumption	Multiple UD – multiple EN	Branch and bound, interior point algorithms
[15]	Computation offloading	Latency, energy consumption	Single UD – single EN	BCD approach
[16]	Computation offloading	Reliability	Multiple UD – multiple EN	FPSO-MR algorithm
[18]	Computation offloading	Reliability	Single UD – multiple EN	LBRS scheme
[19]	Computation offloading	Reliability	Single UD – multiple AP	CAP selection criteria
[20]	Computation offloading	Reliability	Single UD – multiple AP	RAS or MRC with SC or SSC protocols
[21]	Computation offloading	Reliability	Single UD – multiple AP	DDPG approach
[22]	Service caching	Capacity	Multiple UD – multiple EN	Lagrangian multiplier method
[23]	Computation offloading, resource allocation	Capacity	Multiple UD – multiple EN	Exhaustive search algorithm
[24]	Computation offloading	Capacity	Multiple UD – multiple EN	Branch-and-bound method
[25]	Computation offloading	Security and privacy	Multiple UD – multiple EN	IOM

[26]	Computation offloading, resource allocation	Latency	Multiple UD – single EN	Sub-gradient method
[27]	Computation offloading, resource allocation	Latency	Multiple UD – single EN	Hybrid genetic algorithm
[28]	Computation offloading, resource allocation	Latency	Multiple UD – multiple EN	DDPG approach
[29]	Computation offloading, resource allocation	Latency, energy consumption, utility	Multiple UD – single EN	Iterative algorithm and Lagrangian dual method
[30]	Computation offloading, resource allocation	Latency, energy consumption	Multiple UD – multiple EN	Convex decomposition methods and matching game
[31]	Computation offloading, resource allocation	Latency, energy consumption, reliability	Multiple UD – multiple EN	Lyapunov stochastic optimization
[32]	Computation offloading, resource allocation	Energy consumption	Multiple UD – multiple EN	Iterative algorithm
[33]	Computation offloading, resource allocation	Energy consumption	Single UD – multiple AP	Stochastic optimization theory
[34]	Resource allocation	Capacity	Multiple UD – single EN	Algorithm based on ADMM
[35]	Computation offloading, resource allocation	Capacity	Single UD – single EN, Multiple UD – multiple EN	Binary-search water-filling and suboptimal algorithms
[36]	Resource allocation	Utility	Multiple UD – single EN	DTCCRA algorithm
[37]	Computation offloading, resource allocation	Utility, security, and privacy	Multiple UD – multiple EN	BSOM
[39]	Computation offloading, resource allocation, service caching	Latency, utility	Multiple UD – multiple EN	GenCOSCO
[40]	Computation offloading, service caching	Latency, reliability	Single UD – multiple AP	Destination selection criteria
[41]	Service caching	Capacity	Multiple UD – multiple EN	Lagrangian multiplier method

2. 3 Summary

Thus, we can summarize that different metrics can be used to assess edge computing performance. However, in our case, the reliability metric is one of the most crucial. First, it is necessary to define the value of reliability as a measure of performance. Reliability has been a promising tool for evaluating computer networks and services, especially when the entities of the system are dynamic [42]. In general, reliability is the possibility that the system will execute a certain assignment correctly under unforeseen circumstances [43]. This metric is rather broad, and researchers in numerous scientific papers define it differently. As it was revealed in the

literature review, some research works use reliability metric to assess the system's ability to complete maximum assignments without overloading, while others use it to assess the system's ability to execute tasks within power limitations. Moreover, some authors determine reliability as a metric of system security or availability. The researchers in [44] define the reliability measurement in two ways: traditional and novel. The first definition reflects the system's fault resistance, while the second one relates to the delay threshold in a wireless environment [44]. Obviously, the reliability of the system can be deteriorated due to various external agents. The authors of [45] list several of them: channel utilization conflicts between users, frequency sharing by different participants, device mobility, system instability, and others. Figure 2.1 shows the main tools for reliability and latency improvement.

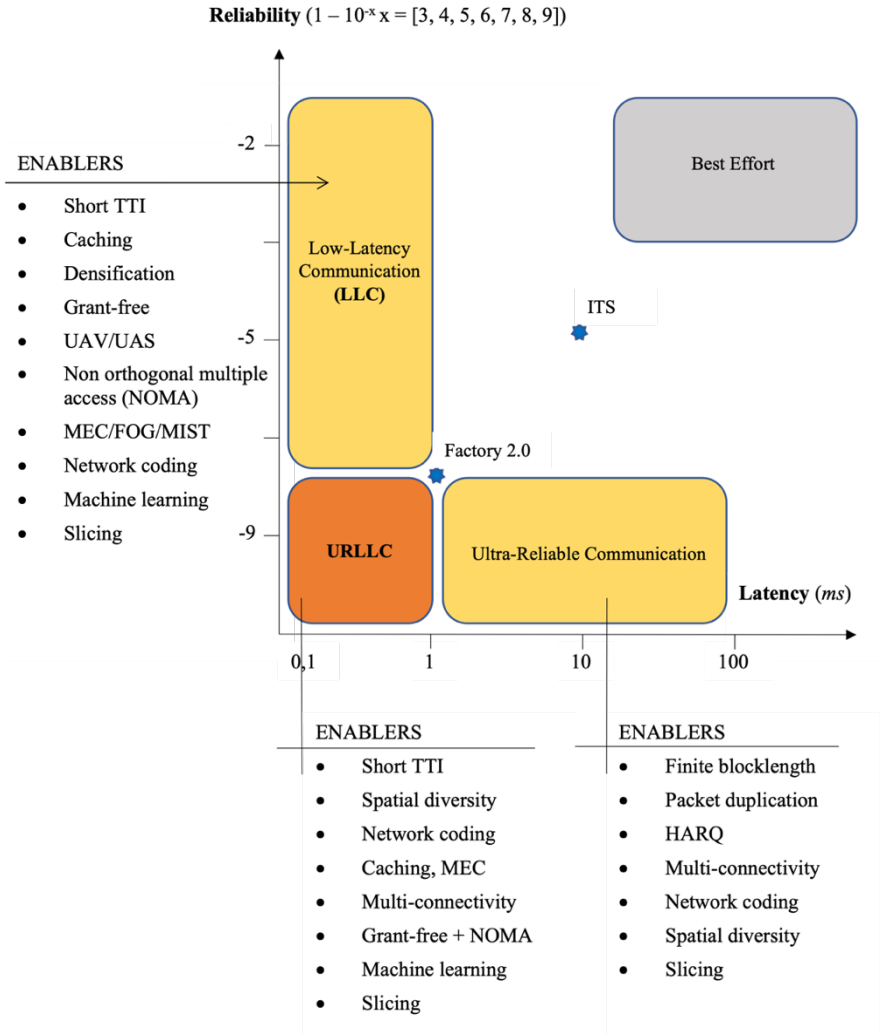


Figure 2.1: Main tools for reliability and latency improvements [45].

Recently, the concept of ultra-reliable and low-latency communication has emerged and attracted the attention of academia. For computing or communicating context, the reliability requirement is integrated with the latency limitation in URRLC [21]. In our work, we use reliability metric in its novel meaning, as a delay-outage probability.

Chapter 3 – Selection schemes for task offloading in edge computing

In selection schemes, the user device offloads its task to one edge node for completion. It is possible to enhance the reliability feature of MEC-assisted IoT services and applications under limited latency requirement by an appropriate edge node selection. Specifically, edge nodes can vary in processing capacity and channel characteristics, which can affect the system performance significantly [21]. Thus, computation offloading, resource allocation, and service caching schemes based on the different edge nodes selection with various parameters will be analyzed.

3. 1 Selection-based scheme for task offloading in edge computing under Rayleigh fading channels

For this scheme, we assume that the channels between the user device and edge nodes are the channels with Rayleigh fading, where channel gain has an exponential distribution. Fading is an unavoidable concept in wireless communication, which represents an influence of different obstacles on signal quality, such as transmission environment, weather, system instability, and many others [46]. Rayleigh, Nakagami-m, and Rician are some of the common types of channel fading.

3. 1. 1 System model

In our model, a user device with constrained capabilities needs to complete a computationally complex task, and thus, it offloads the task to the edge node. Figure 3.1 shows the proposed system with one user and several edge nodes. A wireless network can be used for the task offloading aim. One user device is defined as D, and a set of edge nodes having one antenna is defined as $N = \{1,\ldots, n\}$.

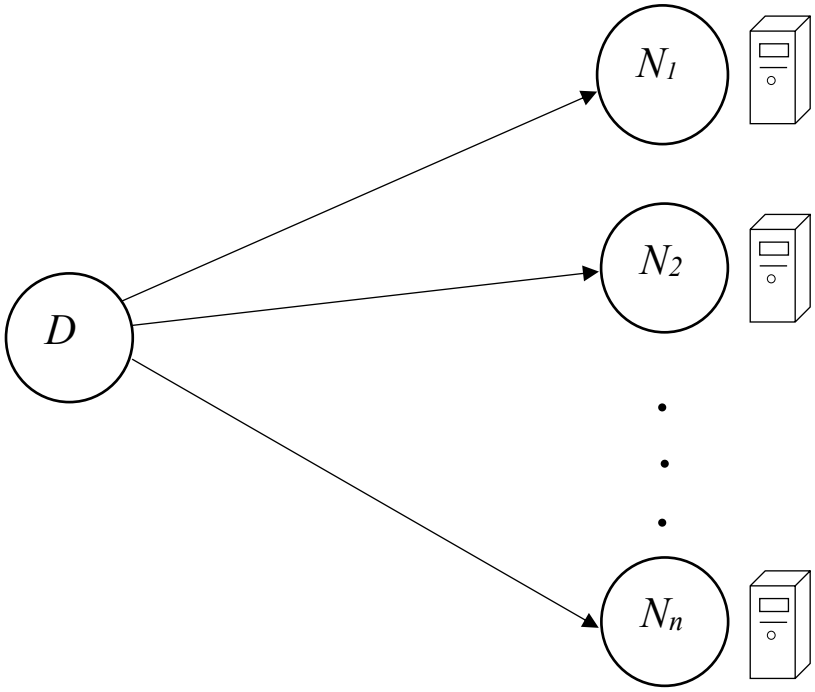


Figure 3.1: Edge computing offloading model.

First, it is necessary to determine offloading time of the system, which consists of parts: transmission time and computation time [21] and [47]:

$$t = t_{tr,in} + t_c + t_{tr,out}, \tag{3.1}$$

where $t_{tr,in}$ is a task transmission time between the user device and edge node, t_c is a task computation time of edge node, $t_{tr,out}$ is an outcome transmission time between the edge node and the user device. Figure 3.2 demonstrates the total time needed for task offloading.

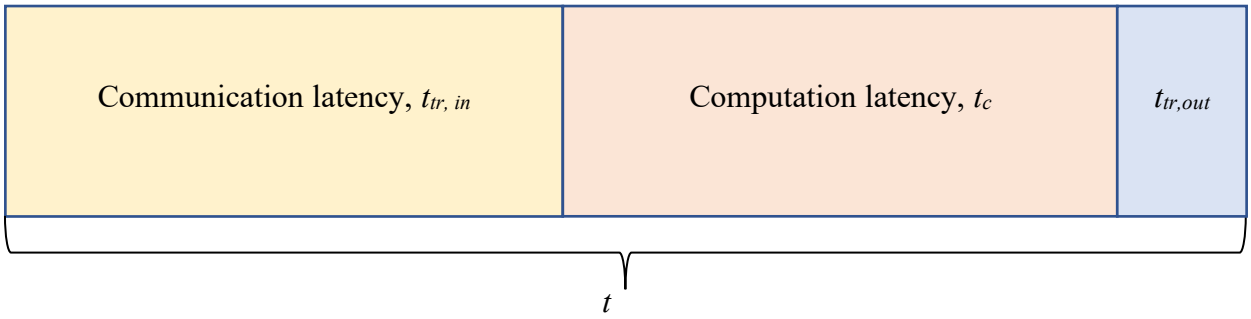


Figure 3.2: Edge computing total offloading time [20].

For the considered system, we will omit $t_{tr,out}$, as it is rather small due to the edge node's significant transmit power [21], [39], and [48]. Therefore, the total time of task offloading from the user device to an edge node for our system is obtained from (3.1) [21] and [48]:

$$t = t_{tr,in} + t_c. \tag{3.2}$$

Hence, the transmission rate should be found first, and then, the transmission time. For this aim, we find transmission rate by Shannon formula:

$$R = W\log_2\left(1 + \frac{P|g_i|^2}{\sigma^2}\right), \tag{3.3}$$

where W is a channel bandwidth, P is a transmit power of the user device, g_i is a channel gain from the user device to the edge node ($g_i \sim \mathcal{CN}(0, \Omega_i)$), for $i \in N$. σ^2 is an additive white Gaussian noise (AWGN).

Further, we can determine the delay-outage probability using the following formula [15], [18] – [20], and [40]:

$$Pr_{out} = Pr\,(t > t_{thr}), \tag{3.4}$$

where $Pr(\cdot)$ is a notation of probability and t_{thr} is a latency threshold, in particular, it is a constrained time for the task execution, which should not be exceeded. The delay-outage probability is determined by the likelihood that the actual task completion time is greater than this threshold [18] and [40].

3. 1. 2 Performance analysis

As was mentioned before, here we find transmission rate by Shannon formula:

$$R = W\log_2\left(1 + \frac{P|g_i|^2}{\sigma^2}\right). \tag{3.5}$$

In Rayleigh fading channels, the average channel gain Ω_i can be formulated through the path-loss function [49] – [51]:

$$\Omega_i = E(|g_i|^2) = \frac{c^2}{16\pi^2 f_c^2 d_i^\alpha}, \tag{3.6}$$

where $E(\cdot)$ is a notation of the expectation, c is the speed of light, f_c is a carrier frequency, d_i is a distance between a user device and edge node, α is a path-loss exponent. Some values of the path-loss exponent based on the medium are provided in Table 3.1.

Table 3.1: Values of path-loss exponent [52] and [53].

Medium	Path-loss exponent, α
Free space	2
In factories	2 – 3
Shadowed urban area	3 – 5
In buildings	4 – 6

An approximation of the average transmission rate for Rayleigh fading channels can be defined as $\overline{R} = E(R)$, thus we get from (3.5), (3.6), and Jensen's inequality [54]:

$$\begin{aligned}\overline{R} &= E\left(W\log_2\left(1 + \frac{P|g_i|^2}{\sigma^2}\right)\right)\\ &\approx W\log_2\left(1 + \frac{E(P|g_i|^2)}{\sigma^2}\right)\\ &= W\log_2\left(1 + \frac{P\Omega_i}{\sigma^2}\right).\end{aligned} \tag{3.7}$$

We can define task transmission time from the user device to an edge node using (3.5) [19], [20], and [40]:

$$t_{tr,i} = \frac{B}{W\log_2\left(1 + \frac{P|g_i|^2}{\sigma^2}\right)}, \tag{3.8}$$

where B bits is the size of the offloaded task.

An approximation of the average task transmission time from the user device to an edge node can be obtained using (3.7):

$$\overline{t}_{tr,i} = \frac{B}{W\log_2\left(1 + \frac{P\Omega_i}{\sigma^2}\right)}. \tag{3.9}$$

Computation latency of an edge node is defined as [19], [20], and [40]:

$$t_{c,i} = \frac{KB}{c_i}, \tag{3.10}$$

where K is a cycle frequency for one bit of task execution and c_i is an edge node's central processing unit (CPU) cycle frequency.

Thus, the total time of task offloading from the user device to an edge node can be determined from (3.2), (3.8), and (3.10) [19], [20], and [40]:

$$t_i = \frac{B}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} + \frac{KB}{c_i}. \quad (3.11)$$

Further, we can define the delay-outage probability for this scenario from (3.4) and (3.11) [19], [20], and [40]:

$$Pr_{outI} = Pr\left(\frac{B}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} + \frac{KB}{c_i} > t_{thr}\right). \quad (3.12)$$

After simplification of the formula (3.12) we obtain:

$$\begin{aligned} Pr_{outI} &= Pr\left(\frac{B}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} > t_{thr} - \frac{KB}{c_i}\right) \\ &= Pr\left(\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right) < \frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}\right) \\ &= Pr\left(|g_i|^2 < \frac{\left(2^{\frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}}-1\right)\sigma^2}{P}\right). \end{aligned} \quad (3.13)$$

The delay-outage probability can be determined from (3.13) through the CDF of the exponential random variable (RV) g_i [40]:

$$Pr_{outI} = \int_0^{\frac{\left(2^{\frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}}-1\right)\sigma^2}{P}} \frac{1}{\lambda} e^{-\frac{1}{\lambda}x}\, dx = 1 - e^{-\frac{\left(2^{\frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}}-1\right)\sigma^2}{\lambda P}}. \quad (3.14)$$

Therefore, it can be seen from (3.14) that the delay-outage probability is related to the CDF of exponential RV and depends on certain values, such as bandwidth and gain of the channel between the user device and edge node, transmit power of user device, AWGN, size of the task, cycle frequency for one bit of task, CPU capacity of the edge node, and latency threshold.

3. 1. 3 Numerical results

This subsection aims to validate the analytical assumptions regarding the delay-outage probability dependence on various system parameters in Rayleigh fading channels. Let us assume task size B = 50 Mbits, channel bandwidth W = 200 MHz, carrier frequency f_c = 28 GHz, path-loss exponent α = 3, cycle frequency for each bit of task execution K = 10, edge node's CPU cycle frequency c_i is 50 GHz, and the latency threshold is t_{thr} = 0.1 second. From Figures 3.3 and 3.4, we can observe the relationship between the system performance and transmit SNR. In this model, transmit SNR, i.e., P/σ^2, varies from 0 to 30 dB. It can be seen that average delay and the delay-outage probability curves decrease with higher values of transmit SNR.

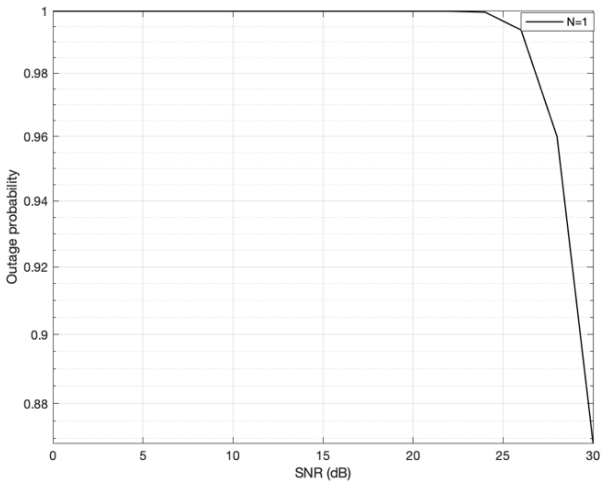


Figure 3.3: The outage probability curve of Rayleigh fading model with B = 50 Mbits, W = 200 MHz, and t_{thr} = 0.1 second versus SNR.

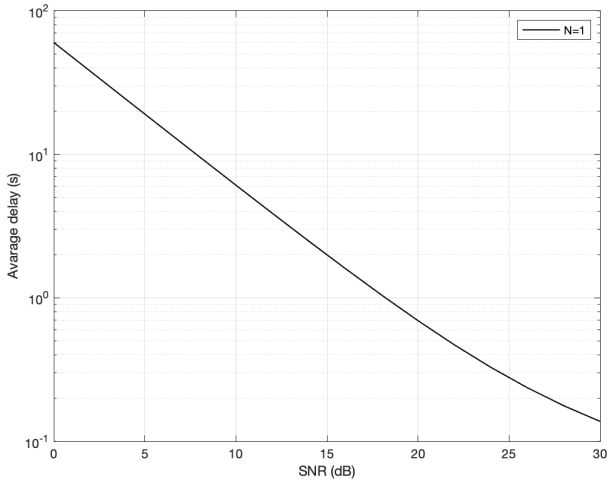


Figure 3.4: Average delay curve of Rayleigh fading model with B = 50 Mbits and W = 200 MHz versus SNR.

Figures 3.5 and 3.6 demonstrate the change in average delay and the delay-outage probability according to different task sizes. Here, we assume transmit SNR is P/σ^2 = 20 dB, while task size B varies from 20 to 200 Mbits. Evidently, task size increase leads to average delay and the delay-outage probability curves rise.

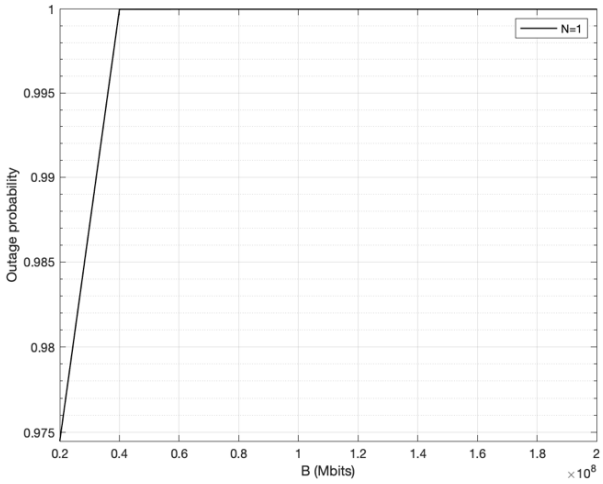


Figure 3.5: The outage probability curve of Rayleigh fading model with P/σ^2 = 20 dB, W = 200 MHz, and t_{thr} = 0.1 second versus task size.

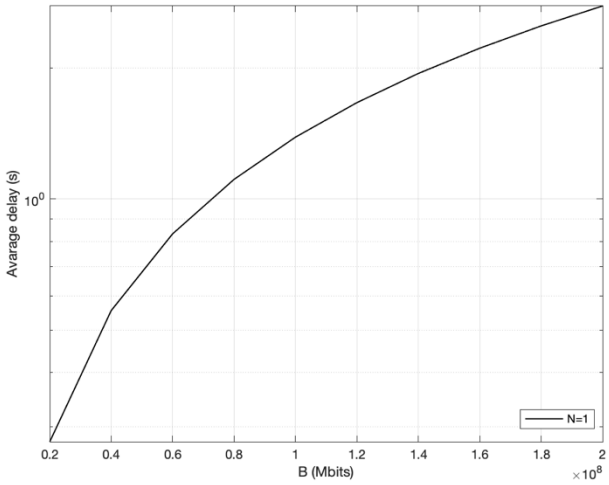


Figure 3.6: Average delay curve of Rayleigh fading model with P/σ^2 = 20 dB and W = 200 MHz versus task size.

From Figures 3.7 and 3.8, we can see how average delay and the delay-outage probability can be improved with higher values of channel bandwidth. For this model, we select channel bandwidth with values from 20 and 200 MHz.

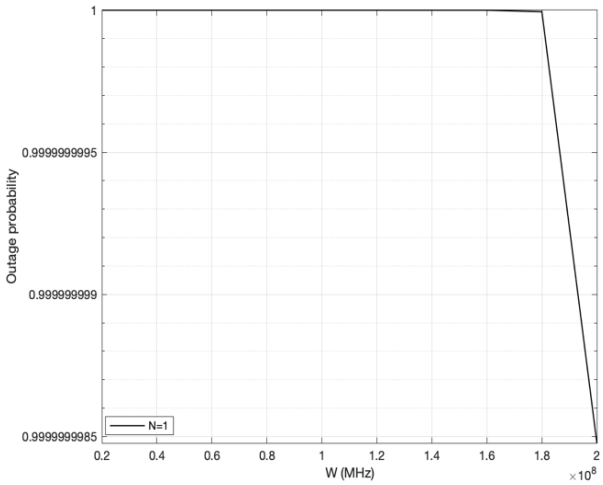


Figure 3.7: The outage probability curve of Rayleigh fading model with P/σ^2 = 20 dB, B = 50 Mbits, and t_{thr} = 0.1 second versus channel bandwidth.

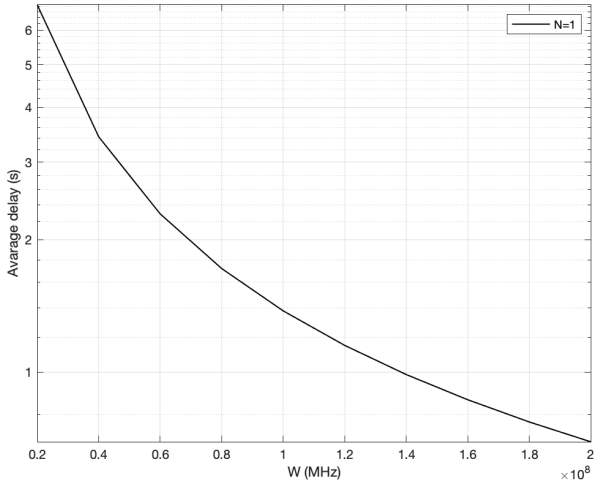


Figure 3.8: Average delay curve of Rayleigh fading model with P/σ^2 = 20 dB and B = 50 Mbits versus channel bandwidth.

In Figure 3.9, we can observe the system performance improvement with the latency threshold increase. Let us assume the latency threshold, i.e., t_{thr}, varies from 0.1 to 1 second.

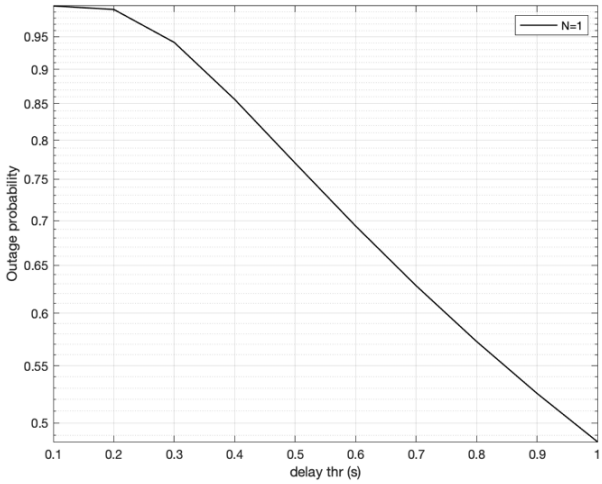


Figure 3.9: The outage probability curve of Rayleigh fading model with P/σ^2 = 20 dB, B = 50 Mbits, and W = 200 MHz versus latency threshold.

Therefore, it can be seen that the delay-outage probability can be reduced by selecting the edge node with the highest CPU capacity and channel gain [18], [19], and [40]. Moreover, by adjusting the other specified parameters, we observe how the delay-outage probability and, therefore, the system performance can be changed. In particular, different values of signal-to-noise ratio (SNR), task size, channel bandwidth, and latency threshold significantly affect the reliability of the system. Further, the selection model will be investigated using Nakagami-m channels.

3. 2 Selection-based scheme for task offloading in edge computing under Nakagami-m fading channels

Nakagami-m fading channels differ from Rayleigh fading channels by having two features, such as m and γ, which bring more precision to the system [55]. The channel gain has a gamma distribution in this scenario. Certain values of m parameter in the Nakagami-m distribution allow covering Rayleigh, Rician, and Gaussian fading channels [56]. Thus, Nakagami-m fading channels are suitable for urban and indoor environments [55] and [56]. In addition, Nakagami-m is an appropriate channel model for mmWave bands utilized in 5G and beyond mobile networks (see, e.g.,[49], [51], and [57]).

3. 2. 1 System model

In this scheme, inspired by [40], we consider a user device with constrained capabilities, which needs to fulfill a computationally complex task, and thus, it offloads the task to the edge node with the relay assistance. Figure 3.10 shows the proposed system with one user device *D*, one relay node *R*, and a set of edge nodes $N = \{1,\ldots,n\}$ having one antenna. However, in this subsection, we assume that the channels between the user device, relay, and edge nodes are the Nakagami-m fading channels.

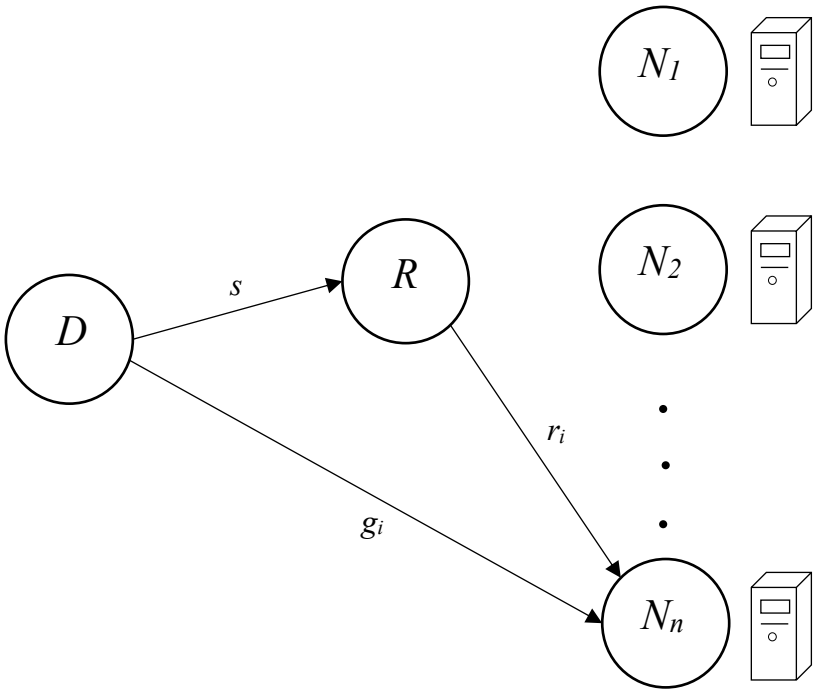


Figure 3.10: Edge computation offloading model with relay assistance [40].

To estimate the delay-outage probability, two models are implemented: relay with cache and relay without cache assistance systems. For the first model, the link between the user device and edge node can be decreased to the link between a relay and edge node [40]. Consequently, task transmission time from the user device to an edge node through cache-assisted relay obtained from (3.8) as:

$$t_{tr,i} = \frac{B}{W\log_2\left(1+\frac{P|r_i|^2}{\sigma^2}\right)}, \tag{3.15}$$

where r_i is a channel gain from the relay to the edge node ($r_i \sim \text{Nak}(m_i, \gamma_i)$) for $i \in N$.

The total time of task offloading from the user device to an edge node in this model can be determined from (3.2), (3.10), and (3.15) [19], [20], and [40]:

$$t_i = \frac{B}{W\log_2\left(1+\frac{P|r_i|^2}{\sigma^2}\right)} + \frac{KB}{c_i}. \tag{3.16}$$

Further, we can define the delay-outage probability for this scenario from (3.4) and (3.16) [19], [20], and [40]:

$$Pr_{outII} = Pr\left(\frac{B}{W\log_2\left(1+\frac{P|r_i|^2}{\sigma^2}\right)} + \frac{KB}{c_i} > t_{thr}\right). \tag{3.17}$$

The second model is represented by the cache-free relay system. In this model, the transmission time is composed of two parts: transmission of the assignment from the user device to relay and edge node simultaneously; processing and transmission of the task by relay to the edge node [40]. Furthermore, two links are integrated at the edge node. Consequently, the task transmission time from the user device to an edge node can be obtained from (3.8) [40] as:

$$t_{tr,i} = \frac{2B}{W\log_2\left(1+\frac{P\max\left(|g_i|^2,\ \min\left(|s|^2,|r_i|^2\right)\right)}{\sigma^2}\right)}, \tag{3.18}$$

where s is a channel gain between the user device and relay ($s \sim \text{Nak}(m,\ \gamma)$).

Thus, the total time of task offloading from the user device to an edge node in this model can be determined from (3.2), (3.10), and (3.18) [40]:

$$t_i = \frac{2B}{W\log_2\left(1+\frac{P\max\left(|g_i|^2,\ \min\left(|s|^2,|r_i|^2\right)\right)}{\sigma^2}\right)} + \frac{KB}{c_i}. \tag{3.19}$$

Further, we can define the delay-outage probability for this scenario from (3.4) and (3.19) [40]:

$$Pr_{outII} = Pr\left(\frac{2B}{W\log_2\left(1+\frac{P\max\left(|g_i|^2,\ \min\left(|s|^2,|r_i|^2\right)\right)}{\sigma^2}\right)} + \frac{KB}{c_i} > t_{thr}\right). \quad (3.20)$$

3. 2. 2 Performance analysis

For both models, we develop appropriate schemes for node selection and investigate system performance based on different architectures.

The first model is represented by two schemes [40]:

- Scheme I: Selection of the edge node with the best CPU c_i
- Scheme II: Selection of the channel between a relay and edge node r_i with the best power gain

The second model is represented by three schemes [40]:

- Scheme I: Selection of the edge node with the best CPU c_i
- Scheme II: Selection of the channel between a relay and edge node r_i with the best power gain
- Scheme III: Selection of the channel between the user device and edge node g_i with the best power gain

Regarding Scheme I of the first model, the edge node with the best CPU is utilized, i.e., maximum c_i, hence the delay-outage probability is based on high transmission time from relay to edge node [19], [20], and [40]. Thus, from (3.17) we obtain:

$$Pr_{outII} = Pr\left(\frac{B}{W\log_2\left(1+\frac{P|r_i|^2}{\sigma^2}\right)} > t_{thr} - \frac{KB}{c_i}\right)$$

$$= Pr\left(\log_2\left(1+\frac{P|r_i|^2}{\sigma^2}\right) < \frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}\right)$$

$$= Pr\left(|r_i|^2 < \frac{\left(2^{\frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}}-1\right)\sigma^2}{P} \right). \quad (3.21)$$

Moreover, analogously to the statement in the previous subsection, $\gamma_i = E(|r_i|^2)$ [57]. Thus, the delay-outage probability can be determined from (3.21) through the CDF of the gamma RV r_i , where $f_{|r_i|^2}(x) = \frac{{m_i}^{m_i} x^{m_i-1}}{{\gamma_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i x}{\gamma_i}}$ [56] and [57]:

$$Pr_{outII} = 1 - e^{-\frac{\gamma_{thr1,i}\sigma^2 m_i}{P\gamma_i}} \sum_{k=0}^{m_i-1} \frac{1}{k!}\left(\frac{\gamma_{thr1,i}\sigma^2 m_i}{P\gamma_i}\right)^k, \quad (3.22)$$

for $\gamma_{thr1,i} \geq 0$, where $\gamma_{thr1,i} = 2^{\frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1$.

According to Scheme II of the first model, the channel between a relay and edge node with the best power gain is chosen, i.e., maximum r_i. In addition, the likelihood of choosing an edge node with the best CPU, i.e., maximum c_i, is similar for every link [40]. Consequently, the delay-outage probability can be determined from (3.21) through the CDF of the maximum of independent gamma RVs r_i, where $f_{|r_i|^2}(x) = \frac{{m_i}^{m_i} x^{m_i-1}}{{\gamma_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i x}{\gamma_i}}$:

$$Pr_{outII} = \prod_{i=1}^{n}\left(1 - e^{-\frac{m_i\gamma_{thr1,i}\sigma^2}{\gamma_i P}} \sum_{k=0}^{m_i-1} \frac{1}{k!}\left(\frac{m_i\gamma_{thr1,i}\sigma^2}{\gamma_i P}\right)^k\right), \quad (3.23)$$

for $\gamma_{thr1,i} \geq 0$, where $\gamma_{thr1,i} = 2^{\frac{B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1$.

For Scheme I of the second model, the edge node with the best CPU is utilized, i.e., maximum c_i, therefore, the delay-outage probability can be expressed by high transmission time [19], [20], and [40]. Thus, from (3.20) we define the delay-outage probability as:

$$Pr_{outII} = Pr\left(\frac{2B}{W\log_2\left(1 + \frac{P\max\left(|g_i|^2,\ \min\left(|s|^2,|r_i|^2\right)\right)}{\sigma^2}\right)} > t_{thr} - \frac{KB}{c_i} \right)$$

$$= Pr\left(\log_2\left(1+\frac{P\max\left(|g_i|^2,\ \min\left(|s|^2,|r_i|^2\right)\right)}{\sigma^2}\right) < \frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}\right)$$

$$= Pr\left(\frac{P\max\left(|g_i|^2,\ \min\left(|s|^2,|r_i|^2\right)\right)}{\sigma^2} < 2^{\frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1\right). \quad (3.24)$$

Because of the independence of channels, we obtain from (3.24) [40]:

$$Pr_{outII} = Pr\left(\frac{P|g_i|^2}{\sigma^2} < 2^{\frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1\right)$$

$$\times\left(1 - Pr\left(\frac{P\min\left(|s|^2,|r_i|^2\right)}{\sigma^2}\right) > 2^{\frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1\right)$$

$$= Pr\left(\frac{P|g_i|^2}{\sigma^2} < 2^{\frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1\right)$$

$$\times\left(1 - Pr\left(\frac{P|s|^2}{\sigma^2} > 2^{\frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1, \frac{P|r_i|^2}{\sigma^2} > 2^{\frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1\right)\right). \quad (3.25)$$

Therefore, the delay-outage probability can be determined from (3.25) through the CDF of the gamma RVs g_i, s, and r_i, where $f_{|g_i|^2}(z) = \frac{{m_i}^{m_i} z^{m_i-1}}{{\beta_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i z}{\beta_i}}$, $f_{|s|^2}(y) = \frac{m^m y^{m-1}}{\alpha^m \Gamma(m)} e^{-\frac{my}{\alpha}}$, and $f_{|r_i|^2}(x) = \frac{{m_i}^{m_i} x^{m_i-1}}{{\gamma_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i x}{\gamma_i}}$:

$$Pr_{outII} = \left(1 - e^{-\frac{m_i\gamma_{thr2,i}\sigma^2}{\beta_i P}} \sum_{k=0}^{m_i-1}\frac{1}{k!}\left(\frac{m_i\gamma_{thr2,i}\sigma^2}{\beta_i P}\right)^k\right)$$

$$\times\left(1 - e^{-\frac{m\gamma_{thr2,i}\sigma^2}{\alpha P}} \sum_{k=0}^{m-1}\frac{1}{k!}\left(\frac{m\gamma_{thr2,i}\sigma^2}{\alpha P}\right)^k\right.$$

$$\left.\times\, e^{-\frac{m_i\gamma_{thr2,i}\sigma^2}{\gamma_i P}} \sum_{k=0}^{m_i-1}\frac{1}{k!}\left(\frac{m_i\gamma_{thr2,i}\sigma^2}{\gamma_i P}\right)^k\right), \quad (3.26)$$

for $\gamma_{thr2,i} \geq 0$, where $\gamma_{thr2,i} = 2^{\frac{2B}{W\left(t_{thr}-\frac{KB}{c_i}\right)}} - 1$.

For Scheme II of the second model, the channel between a relay and edge node with the best power gain is chosen, i.e., maximum r_i. In addition, the likelihood of choosing an edge node with the best CPU, i.e., maximum c_i, is similar for every link [40]. Hence, the delay-outage probability can be determined from (3.25) through the CDF of the gamma RVs g_i, s and the maximum of gamma RVs r_i, where $f_{|g_i|^2}(z) = \frac{{m_i}^{m_i} z^{m_i-1}}{{\beta_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i z}{\beta_i}}$, $f_{|s|^2}(y) = \frac{m^m y^{m-1}}{\alpha^m \Gamma(m)} e^{-\frac{my}{\alpha}}$, $f_{|r_i|^2}(x) = \frac{{m_i}^{m_i} x^{m_i-1}}{{\gamma_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i x}{\gamma_i}}$, and using order statistics:

$$Pr_{outII} = Pr\left(\frac{P|g_i|^2}{\sigma^2} < \gamma_{thr2,i}\right)\left(1 - Pr\left(\frac{P|s|^2}{\sigma^2} > \gamma_{thr2,i}\right) Pr\left(\frac{P\max|r_i|^2}{\sigma^2} > \gamma_{thr2,i}\right)\right)$$

$$= Pr\left(\frac{P|g_i|^2}{\sigma^2} < \gamma_{thr2,i}\right)\left(1 - Pr\left(\frac{P|s|^2}{\sigma^2} > \gamma_{thr2,i}\right)\left(1 - Pr\left(\frac{P\max|r_i|^2}{\sigma^2} < \gamma_{thr2,i}\right)\right)\right)$$

$$= Pr\left(\frac{P|g_i|^2}{\sigma^2} < \gamma_{thr2,i}\right)\left(1 - Pr\left(\frac{P|s|^2}{\sigma^2} > \gamma_{thr2,i}\right)\right)$$

$$\times\left(1 - Pr\left(\frac{P|r_1|^2}{\sigma^2} < \gamma_{thr2,i}, \frac{P|r_2|^2}{\sigma^2} < \gamma_{thr2,i}, \dots, \frac{P|r_n|^2}{\sigma^2} < \gamma_{thr2,i}\right)\right)$$

$$= Pr\left(\frac{P|g_i|^2}{\sigma^2} < \gamma_{thr2,i}\right)\left(1 - Pr\left(\frac{P|s|^2}{\sigma^2} > \gamma_{thr2,i}\right)\left(1 - \prod_{i=1}^{n} Pr\left(\frac{P|r_i|^2}{\sigma^2} < \gamma_{thr2,i}\right)\right)\right)$$

$$= \left(1 - e^{-\frac{m_i\gamma_{thr2,i}\sigma^2}{\beta_i P}} \sum_{k=0}^{m_i-1} \frac{1}{k!}\left(\frac{m_i\gamma_{thr2,i}\sigma^2}{\beta_i P}\right)^k\right)\left(1 - e^{-\frac{m\gamma_{thr2,i}\sigma^2}{\alpha P}} \sum_{k=0}^{m-1} \frac{1}{k!}\left(\frac{m\gamma_{thr2,i}\sigma^2}{\alpha P}\right)^k\right.$$

$$\left.\times\left(1 - \prod_{i=1}^{n}\left(1 - e^{-\frac{m_i\gamma_{thr2,i}\sigma^2}{\gamma_i P}} \sum_{k=0}^{m_i-1} \frac{1}{k!}\left(\frac{m_i\gamma_{thr2,i}\sigma^2}{\gamma_i P}\right)^k\right)\right)\right), \quad (3.27)$$

for $\gamma_{thr2,i} \geq 0$, where $\gamma_{thr2,i} = 2^{\frac{2B}{W\left(t_{thr} - \frac{KB}{c_i}\right)}} - 1$.

For Scheme III of the second model, the channel between the user device and edge node with the best power gain is chosen, i.e., maximum g_i. In addition, the likelihood of choosing an edge node with the best CPU, i.e., maximum c_i is similar for every link [40]. Consequently, the delay-outage probability can be determined from (3.25) using CDF of the gamma RVs s, r_i,

and maximum of gamma RV g_i, where $f_{|s|^2}(y) = \frac{m^m y^{m-1}}{\alpha^m \Gamma(m)} e^{-\frac{my}{\alpha}}$, $f_{|r_i|^2}(x) = \frac{{m_i}^{m_i} x^{m_i-1}}{{\gamma_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i x}{\gamma_i}}$, $f_{|g_i|^2}(z) = \frac{{m_i}^{m_i} z^{m_i-1}}{{\beta_i}^{m_i}\Gamma(m_i)} e^{-\frac{m_i z}{\beta_i}}$, and using order statistics:

$$Pr_{outII} = \left(1 - e^{-\frac{m\gamma_{thr2,i}\sigma^2}{\alpha P}} \sum_{k=0}^{m-1} \frac{1}{k!}\left(\frac{m\gamma_{thr2,i}\sigma^2}{\alpha P}\right)^k \times e^{-\frac{m_i\gamma_{thr2,i}\sigma^2}{\gamma_i P}} \sum_{k=0}^{m_i-1} \frac{1}{k!}\left(\frac{m_i\gamma_{thr2,i}\sigma^2}{\gamma_i P}\right)^k\right) \times \prod_{i=1}^{n}\left(1 - e^{-\frac{m_i\gamma_{thr2,i}\sigma^2}{\beta_i P}} \sum_{k=0}^{m_i-1} \frac{1}{k!}\left(\frac{m_i\gamma_{thr2,i}\sigma^2}{\beta_i P}\right)^k\right), \quad (3.28)$$

for $\gamma_{thr2,i} \geq 0$, where $\gamma_{thr2,i} = 2^{\frac{2B}{W\left(t_{thr} - \frac{KB}{c_i}\right)}} - 1$.

From the performance evaluation conducted in this subsection, it was revealed that caching in edge computing with relay can influence reliability metric sufficiently, as for cache-assisted schemes transmission time is decreased. In addition, there is flexibility in choosing the most efficient link between the user device and edge node in edge computing with cache-free relay, which also influences the reliability metric. Eventually, the delay-outage probability depends on the computational capacity of the chosen edge node and the number of edge nodes in the system. Thus, the impact of the aforementioned aspects will be observed during simulations. In the following chapter, we investigate the delay-outage probability for three different combining schemes.

3. 2. 3 Numerical results

In this subsection, numerical results also intend to confirm analytical assumptions obtained in the performance analysis subsection for Nakagami-m fading channels. To begin, we consider the first edge computing model, i.e., with cache-assisted relay, Scheme I and Scheme II. As was mentioned above, Scheme I represents edge computing with cache-assisted relay, where edge node with the best CPU is selected. At the same time, Scheme II represents

edge computing with cache-assisted relay, where the best channel gain between the relay and edge node is selected. Let us assume task size B = 50 Mbits, channel bandwidth W = 200 MHz, carrier frequency f_c = 28 GHz, path-loss exponent α = 3, Nakagami parameter m = 2, cycle frequency for each bit of task execution K = 10, and edge node's CPU cycle frequency c_i is 50 GHz.

In Figures 3.11 and 3.12, we can observe the relationship between the system performance and transmit SNR for Scheme I and Scheme II, respectively. For comparison, we use different parameters of the latency threshold. Thus, transmit SNR, i.e., P/σ^2, varies from 0 to 30 dB, while the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds. It can be seen that the delay-outage probability curves of the system decrease with higher values of transmit SNR and the latency threshold in both cases. Moreover, a larger number of edge nodes in Scheme II, namely N = 1, 2, 4, can also improve the system performance even more, which is obvious at higher values of SNR.

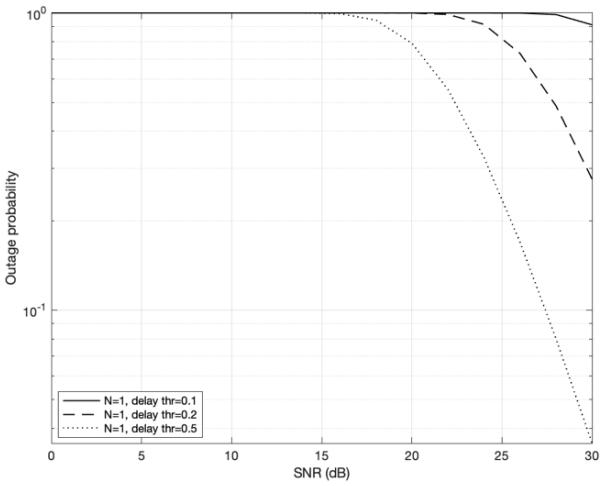


Figure 3.11: The outage probability curves of the first model, Scheme I, with B = 50 Mbits, W = 200 MHz, and m = 2 versus SNR for different delay threshold values.

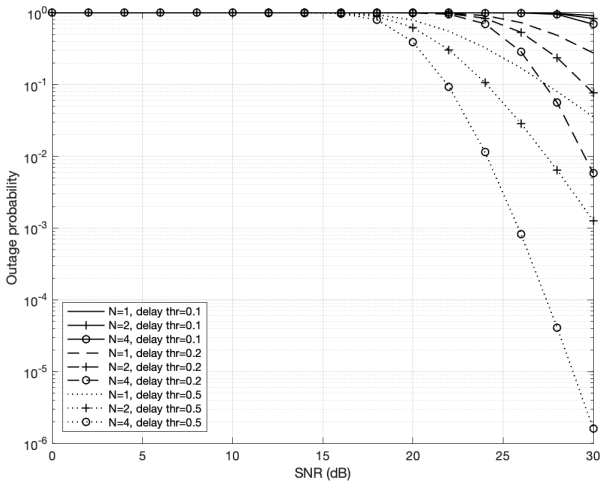


Figure 3.12: The outage probability curves of the first model, Scheme II, with B = 50 Mbits, W = 200 MHz, and m = 2 versus SNR for different delay threshold values and different number of edge nodes.

Figures 3.13 and 3.14 show the dependency of the delay-outage probability on task size. We assume transmit SNR is P/σ^2 = 20 dB. Here, task size B varies from 20 to 200 Mbits, and

the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds. Obviously, the delay-outage probability curves increase with the task size. However, the system performance can be improved by higher values of the latency threshold in both cases. While a larger number of edge nodes, that is, N = 1, 2, 4, can also enhance the system performance in Scheme II, this benefit is more noticeable for smaller task sizes.

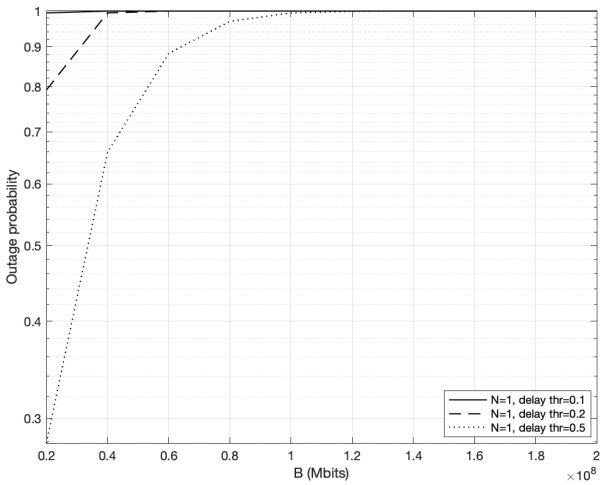


Figure 3.13: The outage probability curves of the first model, Scheme I, with P/σ^2 = 20 dB, W = 200 MHz, and m = 2 versus task size for different delay threshold values.

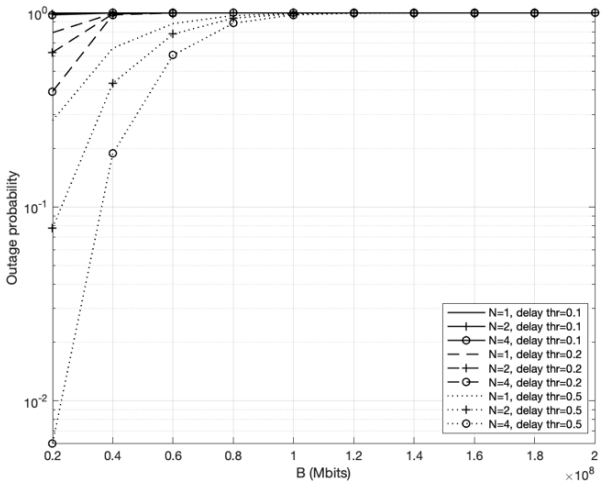


Figure 3.14: The outage probability curves of the first model, Scheme II, with P/σ^2 = 20 dB, W = 200 MHz, and m = 2 versus task size for different delay threshold values and different number of edge nodes.

From Figures 3.15 and 3.16, we can observe the delay-outage probability curves decrease with higher values of channel bandwidth W in Scheme I and Scheme II. For this model, we select channel bandwidth with parameters from 20 to 200 MHz, and the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds. Thus, higher values of the latency threshold also provide the delay-outage probability curves decrease in both cases. Moreover, with a greater number of the edge nodes, i.e., N = 1, 2, 4, the system performance improves remarkably.

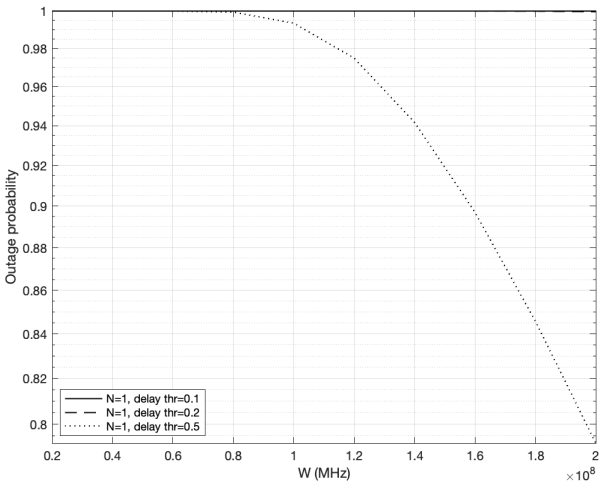


Figure 3.15: The outage probability curves of the first model, Scheme I, with $P/\sigma^2 = 20$ dB, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values.

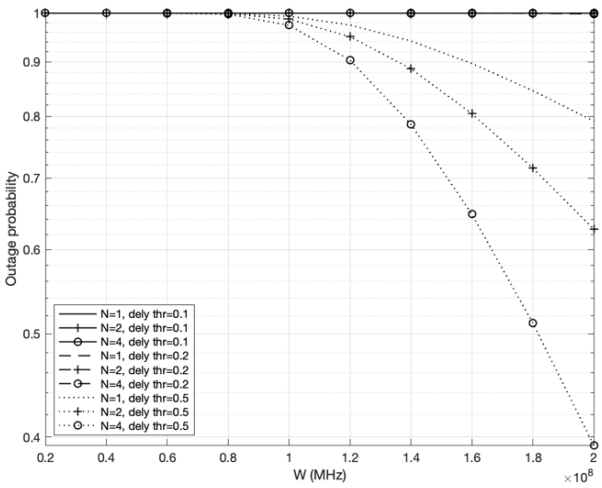


Figure 3.16: The outage probability curves of the first model, Scheme II, with $P/\sigma^2 = 20$ dB, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values and different number of edge nodes.

Next, we observe the second edge computing model, i.e., with cache-free relay, Scheme I, Scheme II, and Scheme III. Scheme I is a selection scheme that uses the edge node with the best CPU for task computing. At the same time, the best channel gain between the relay and edge node is selected in Scheme II. Finally, the best channel gain between the user device and edge node is chosen in Scheme III. In the same vein, we assume task size B = 50 Mbits, channel bandwidth W = 200 MHz, carrier frequency f_c = 28 GHz, path-loss exponent α = 3, Nakagami parameter m = 2, cycle frequency for each bit of task execution K = 10, and edge node's CPU cycle frequency c_i is 50 GHz.

From Figures 3.17, 3.18, and 3.19, we see that the system performance in this model is also related to SNR values. Specifically, when transmit SNR, i.e., P/σ^2, increases from 0 to 30 dB and the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds, the delay-outage probability curves decrease significantly in all schemes. For Scheme II and Scheme III, we utilize a different number of edge nodes, namely N = 1, 2, 4, which also leads to the reliability

enhancement. Evidently, the advantage of using more edge nodes is more noticeable for higher SNRs in Scheme II and Scheme III.

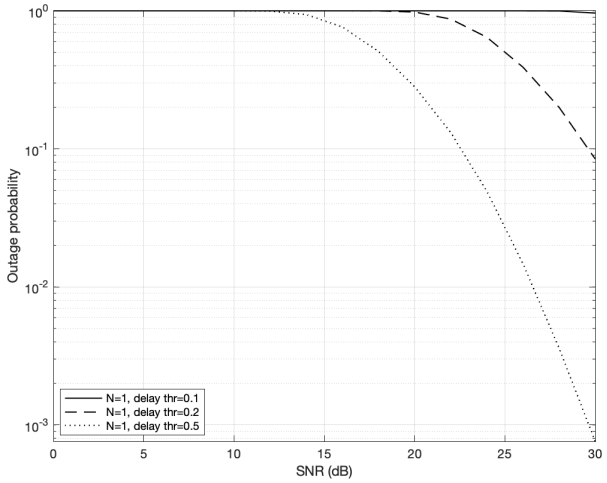


Figure 3.17: The outage probability curves of the second model, Scheme I, with B = 50 Mbits, W = 200 MHz, and m = 2 versus SNR for different delay threshold values.

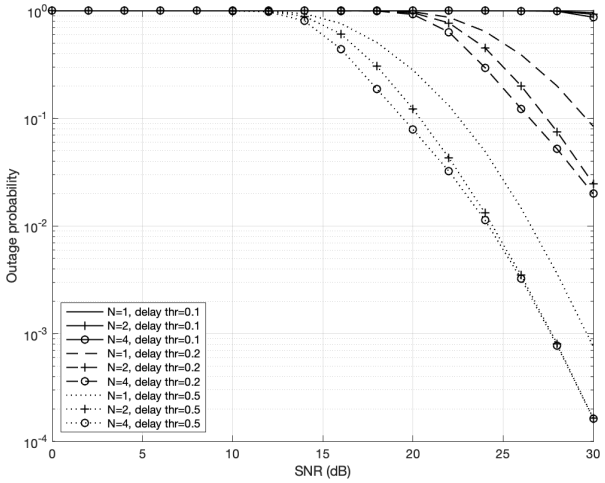


Figure 3.18: The outage probability curves of the second model, Scheme II, with B = 50 Mbits, W = 200 MHz, and m = 2 versus SNR for different delay threshold values and different number of edge nodes.

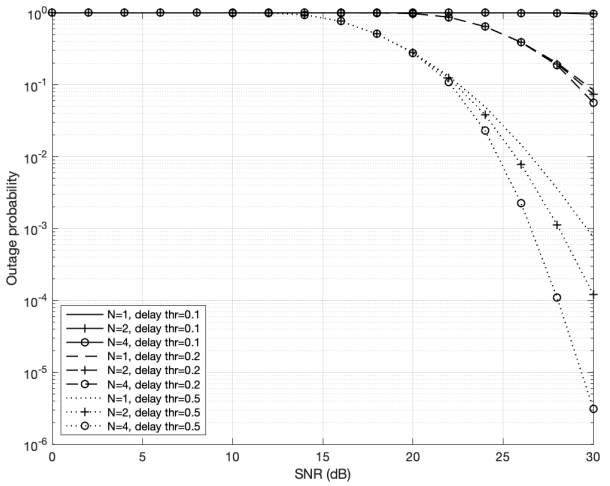


Figure 3.19: The outage probability curves of the second model, Scheme III, with B = 50 Mbits, W = 200 MHz, and m = 2 versus SNR for different delay threshold values and different number of edge nodes.

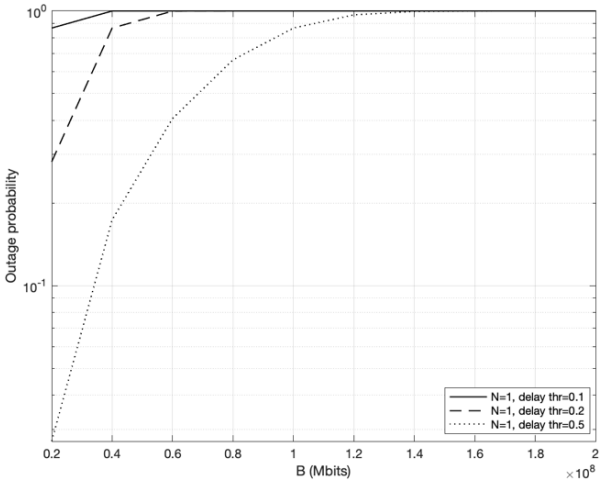


Figure 3.20: The outage probability curves of the second model, Scheme I, with P/σ^2=20 dB, W = 200 MHz, and m = 2 versus task size for different delay threshold values.

Figures 3.20, 3.21, and 3.22 demonstrate how the delay-outage probability depends on task size. Here, we assume transmit SNR is P/σ^2 = 20 dB and the latency threshold is t_{thr} =

0.1, 0.2, 0.5 seconds, while task size B varies from 20 to 200 Mbits. It can be seen that the system performance deteriorates in terms of the reliability metric with a larger task size. In contrast, higher values of the latency threshold result in better system performance in all schemes. In addition, the use of more edge nodes leads to the delay-outage probability curves decrease, though this improvement is reasonable for the smaller task size.

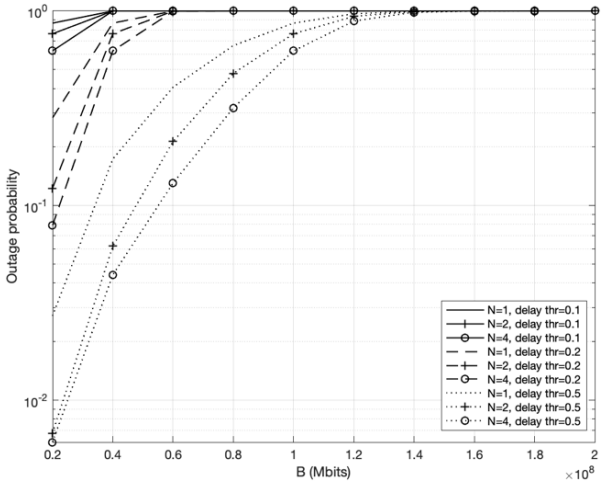


Figure 3.21: The outage probability curves of the second model, Scheme II, with P/σ^2=20 dB, W = 200 MHz, and m = 2 versus task size for different delay threshold values and different number of edge nodes.

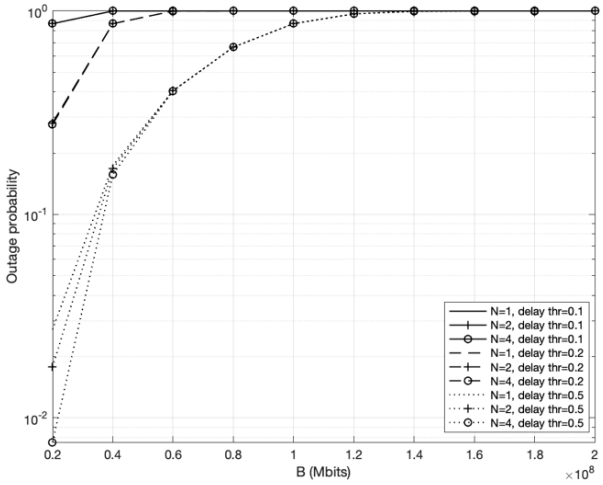


Figure 3.22: The outage probability curves of the second model, Scheme III, with P/σ^2=20 dB, W = 200 MHz, and m = 2 versus task size for different delay threshold values and different number of edge nodes.

Figures 3.23, 3.24, and 3.25 show the relationship between the delay-outage probability and channel bandwidth. In this experiment, channel bandwidth ranges from 20 to 200 MHz and the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds. We can observe the system performance improvement with higher values of channel bandwidth and the latency threshold in Scheme I, Scheme II, and Scheme III. The number of edge nodes, that is, N = 1, 2, 4, can also impact the system performance. Thus, the delay-outage probability curves decrease with a larger number of edge nodes, which is more noticeable in Scheme II.

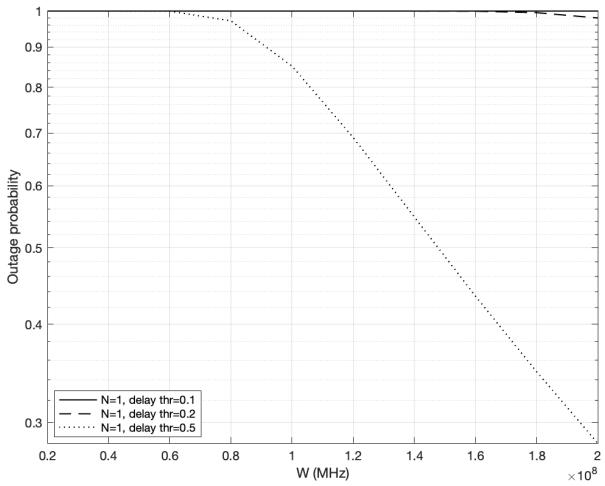


Figure 3.23: The outage probability curves of the second model, Scheme I, with P/σ^2 = 20 dB, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values.

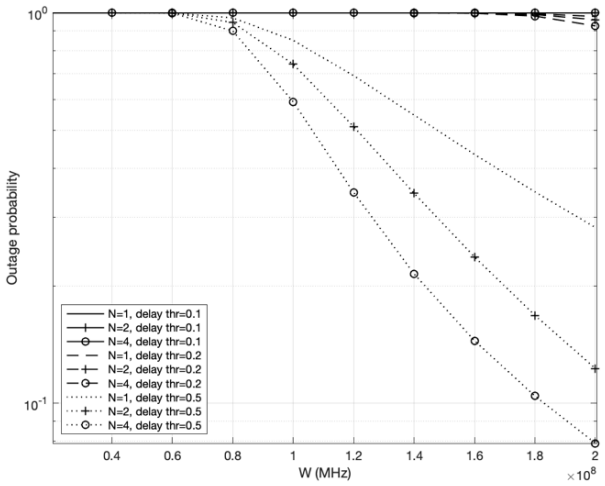


Figure 3.24: The outage probability curves of the second model, Scheme II, with P/σ^2 = 20 dB, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values and different number of edge nodes.

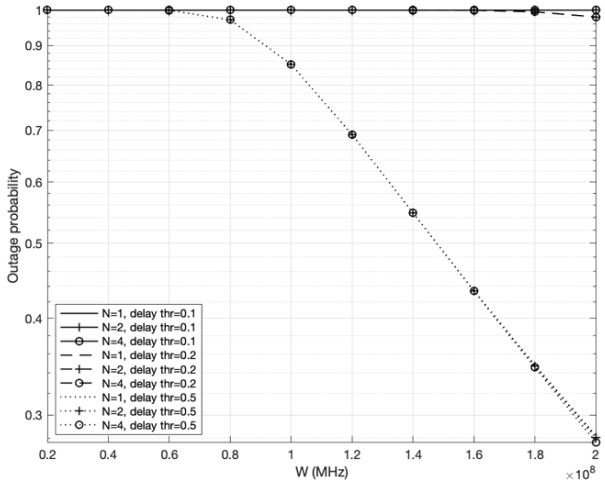


Figure 3.25: The outage probability curves of the second model, Scheme III, with P/σ^2 = 20 dB, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values and different number of edge nodes.

Hence, from the experiments above, it was revealed that the reliability of all schemes is highly dependent on SNR, task size, and channel bandwidth values. In addition, higher values of the latency threshold provide more opportunity for edge nodes to fulfill intensive tasks on time, therefore the delay-outage probability is decreased. Further, more edge nodes

improve the system performance to a certain extent in all cases, although this advantage differs depending on the models and schemes under consideration.

Eventually, we compare edge computing selection schemes with Nakagami-m fading channels presented above. For this aim, the delay-outage probability curves versus SNR, task size, and channel bandwidth are demonstrated in Figures 3.26, 3.27, and 3.28, respectively. In these schemes, the latency threshold is t_{thr} = 0.5 second and the number of edge nodes is N = 2. It can be seen that edge computing model with cache-free relay outperforms edge computing model with cache-assisted relay in terms of the reliability metric. This pattern can be justified by the flexibility in choosing the most efficient link between the user device and edge node in edge computing with cache-free relay. Interestingly that the system performance can be considerably improved in edge computing model, where the best channel gain between the relay and edge node is selected.

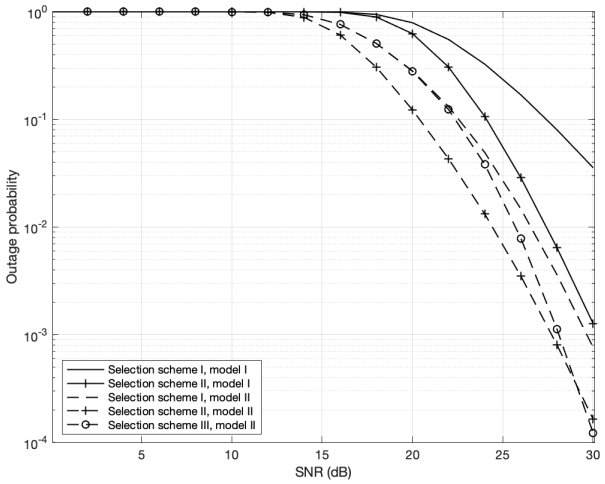


Figure 3.26: The outage probability curves of the Selection schemes, Nakagami-m fading channels, with B = 50 Mbits, W = 200 MHz, m = 2, t_{thr} = 0.5 second, and N = 2 versus SNR.

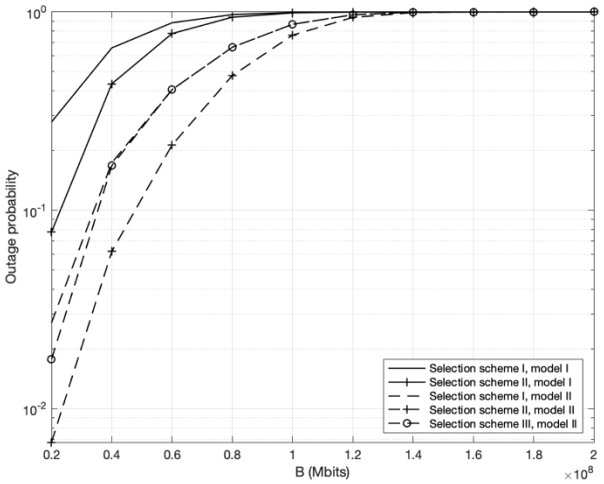


Figure 3.27: The outage probability curves of the Selection schemes, Nakagami-m fading channels, with P/σ^2 = 20 dB, W = 200 MHz, m = 2, t_{thr} = 0.5 second, and N = 2 versus task size.

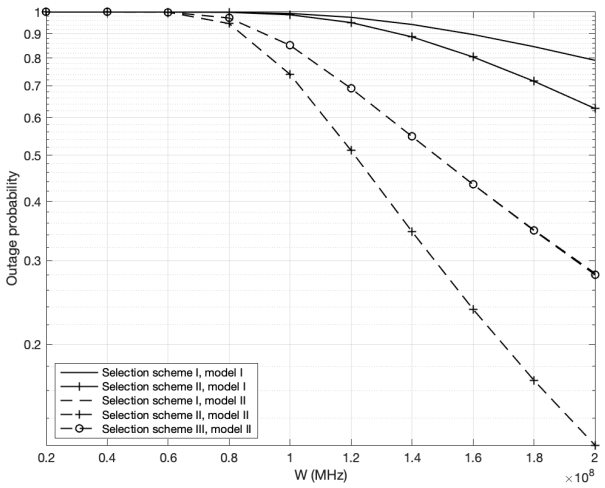


Figure 3.28: The outage probability curves of the Selection schemes, Nakagami-m fading channels, with P/σ^2= 20 dB, B = 50 Mbits, m = 2, t_{thr} = 0.5 second, and N = 2 versus channel bandwidth.

Chapter 4 – Combining schemes for task offloading in edge computing

Another approach is used in combining schemes, which has certain advantages over the strategy used in selection schemes. In this section, we investigate the delay-outage probability for IoT systems with edge computing assistance, in which all edge nodes are utilized simultaneously for task offloading. It is acknowledged that the main disadvantage of the wireless network in multiple access models is represented by the channels' interference. Time-division multiple access, frequency-division multiple access, space-division multiple access, and code-division multiple access turned out to be the solution to this problem [58].

Thus, we study three different scenarios of combining schemes in this chapter to assess the performance of computation offloading and resource allocation in edge computing by reliability metric.

4. 1 TDMA model for task offloading

In this model, the user device offloads a complex task to all edge nodes, at the same time, through the time interval allocated to each access [58].

4. 1. 1 System model

The usage of time-division multiple access model for edge computing is demonstrated in Figure 4.1. In this scheme, B bits of the offloaded task are equally divided between N edge nodes, and thus, each edge node processes b_i bits of the data, for $i \in N$.

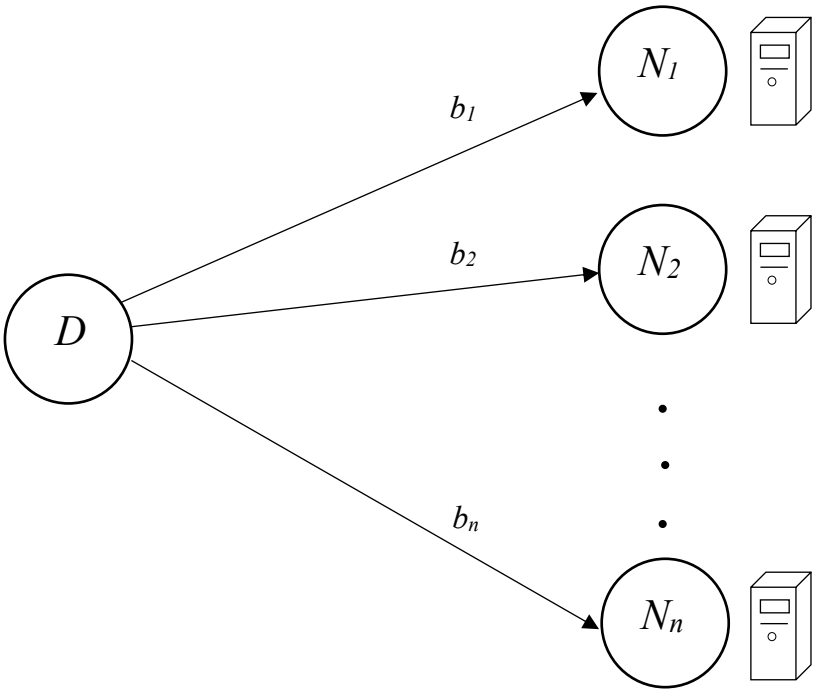


Figure 4.1: Edge computation offloading with TDMA model.

Hence, a transmission rate from the user device to each edge node can be obtained from Shannon formula:

$$R_i = W\log_2\left(1 + \frac{P|g_i|^2}{\sigma^2}\right). \tag{4.1}$$

Then, a task transmission time from the user device to each edge node is obtained using (4.1) [19], [20], and [40]:

$$t_{tr,i} = \frac{b_i}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)}. \tag{4.2}$$

In the TDMA based edge computing system, the user device shares the task between N edge nodes, therefore, each node obtains b_i bits of the data for computation, for $i \in N$. The transmission of the divided data from the user device to each node is performed at the assigned time interval. Then, edge nodes perform computation on the received data b_i and send the outcome back. However, the time of sending the results to the user can be neglected due to the small data size. Therefore, the total time of task offloading in TDMA model represents the sum of all time slots for transmitting divided data from the user device and computation time for the received data at the edge nodes [59]. It can be determined from (3.2), (3.10), and (4.2) as:

$$t = \sum_{i=1}^{n} \frac{b_i}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} + \frac{Kb_i}{c_i}. \quad (4.3)$$

We can define the delay-outage probability from (3.4) and (4.3) for this scenario as:

$$Pr_{out,I} = Pr\left(\sum_{i=1}^{n} \frac{b_i}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} + \frac{Kb_i}{c_i} > t_{thr}\right). \quad (4.4)$$

4. 1. 2 Performance analysis

The delay-outage probability can be obtained through the CDF of the summation of independent exponential RVs Y_i:

$$Pr_{out,I} = Pr(\sum_{i=1}^{n} Y_i > t_{thr}), \quad (4.5)$$

where $Y_i = \frac{b_i}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} + \frac{Kb_i}{c_i}$.

For this aim we can derive the CDF of Y_i for Rayleigh fading channels:

$$F_i(x) = \Pr(Y_i > x) = Pr\left(\frac{b_i}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} + \frac{Kb_i}{c_i} > x\right)$$

$$= Pr\left(\frac{b_i}{W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right)} > x - \frac{Kb_i}{c_i}\right)$$

$$= Pr\left(W\log_2\left(1+\frac{P|g_i|^2}{\sigma^2}\right) < \frac{b_i}{x-\frac{Kb_i}{c_i}}\right)$$

$$= Pr\left(|g_i|^2 < \frac{\left(2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}}-1\right)\sigma^2}{P}\right) = 1 - e^{-\frac{\left(2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}}-1\right)\sigma^2}{\lambda P}}. \quad (4.6)$$

Taking into account the following condition:

$$F_i(x) = \begin{cases} 1 - e^{-\frac{\left(2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}-1}\right)\sigma^2}{\lambda P}}, & x > \frac{Kb_i}{c_i} \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

Then, we can obtain PDF from CDF above by taking the derivative with respect to variable *x*:

$$f_i(x) = -\frac{\ln(2)\, b_i\, 2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}} e^{-\frac{\left(2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}-1}\right)\sigma^2}{\lambda P}}\sigma^2}{\lambda P W\left(x-\frac{Kb_i}{c_i}\right)^2}. \tag{4.8}$$

To determine the delay-outage probability, CDF from the sum of the densities of RVs should be found. In the case of independent RVs, the sum of their PDFs is equal to their convolution [60] – [62]. However, in many research works (see, e.g., [63] – [65]), it is proposed to use the characteristic function to alleviate the problem's complexity. Thus, using Fourier transform of PDF, we can obtain a characteristic function of Y_i:

$$\varphi_{Y_i}(\omega) = \left(F f_i(x)\right)^* = \int_{-\infty}^{\infty} f_i(x) e^{j\omega x} dx$$

$$= \int_{-\infty}^{\infty} -\frac{\ln(2)\, b_i\, 2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}} e^{-\frac{\left(2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}-1}\right)\sigma^2}{\lambda P}}\sigma^2}{\lambda P W\left(x-\frac{Kb_i}{c_i}\right)^2} e^{j\omega x} dx, \tag{4.9}$$

where *F* is Fourier Transform.

If RVs are independent, the characteristic function of their sum can be expressed as a product of their characteristic functions [60] and [66]:

$$\varphi_{\sum_{i=1}^{n} Y_i}(\omega) = \varphi_{Y_1}(\omega) \cdots \varphi_{Y_n}(\omega), \tag{4.10}$$

where $\sum_{i=1}^{n} Y_i = Y_1 + \cdots + Y_n$,and φ_{Y_i} is the characteristic function of the exponential RV Y_i. Therefore, we can obtain from (4.10):

$$\varphi_{\sum_{i=1}^{n} Y_i}(\omega) = \prod_{i=1}^{n} \varphi_{Y_i}(\omega)$$

$$= \prod_{i=1}^{n} \int_{-\infty}^{\infty} -\frac{\ln(2)\, b_i\, 2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}} e^{-\frac{\left(2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}}-1\right)\sigma^2}{\lambda P}}\sigma^2}{\lambda P W\left(x-\frac{Kb_i}{c_i}\right)^2} e^{j\omega x} dx. \quad (4.11)$$

Furthermore, using Fourier inversion formula, PDF of $\sum_{i=1}^{n} Y_i$ can be obtained from $\varphi_{\sum_{i=1}^{n} Y_i}(\omega)$:

$$f_{\sum_{i=1}^{n} Y_i}(x) =$$

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \left(\prod_{i=1}^{n} \int_{-\infty}^{\infty} -\frac{\ln(2)\, b_i\, 2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}} e^{-\frac{\left(2^{\frac{b_i}{W\left(x-\frac{Kb_i}{c_i}\right)}}-1\right)\sigma^2}{\lambda P}}\sigma^2}{\lambda P W\left(x-\frac{Kb_i}{c_i}\right)^2} e^{j\omega x} dx \right)^{*} e^{-j\omega x}\, d\omega. \quad (4.12)$$

Then, we can obtain CDF of $\sum_{i=1}^{n} Y_i$ from its PDF:

$$Pr_{outI} = \int_{0}^{t_{thr}} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \left(\prod_{i=1}^{n} \int_{-\infty}^{\infty} f_i(x)\, e^{j\omega x} dx \right)^{*} e^{-j\omega x}\, d\omega \right) dx, \quad (4.13)$$

for $t_{thr} \geq 0$. There is no closed-form expression for CDF, and we cannot derive an analytical expression for CDF. However, it can be computed numerically.

In addition, the authors of [63] – [65] proposed to use the approach implemented in [62] and composite trapezoidal rule to obtain CDF of $\sum_{i=1}^{n} Y_i$:

$$Pr_{outI} \approx \frac{1}{2} + \frac{1}{2\pi} \left(\frac{\omega_{max} - \omega_{min}}{n} \left[\frac{\Upsilon(x, \omega_{min}) + \Upsilon(x, \omega_{max})}{2} + \right. \right.$$

$$+\sum_{k=1}^{n-1}\Upsilon\left(x,\omega_{min}+k\,\frac{\omega_{max}-\omega_{min}}{n}\right)\Big]\Big), \tag{4.14}$$

where $\Upsilon(x,\omega)=\frac{e^{j\omega x}\varphi_{\sum_{i=1}^{n}Y_i}(-\omega)-e^{-j\omega x}\varphi_{\sum_{i=1}^{n}Y_i}(\omega)}{j\omega}$.

4. 1. 3 Numerical results

In this subsection, we introduce simulation results in order to evaluate the system performance numerically. For this aim, we consider TDMA model with Rayleigh fading channels and Nakagami-m fading channels. In corresponding simulations, we iterate 1000 000 times to reach a proper average. Let us assume task size B = 50 Mbits, channel bandwidth W = 200 MHz, carrier frequency f_c = 28 GHz, path-loss exponent α = 3, cycle frequency for each bit of task execution K = 10, noise power $\sigma^2 = 8 \times 10^{-10}$ mWatt, and edge node's CPU cycle frequency c_i is 50 GHz. Nakagami parameter m = 2 in Nakagami-m fading channels.

From Figures 4.2 and 4.3, we can observe how the system performance depends on transmit power in Rayleigh fading channels and Nakagami-m fading channels, respectively. For comparison, we use different parameters of the latency threshold and different number of edge nodes. Thus, transmit power, i.e., P varies from 0 to 20 dBm, the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds, and the number of edge nodes is N = 1, 2, 4. The delay-outage probability curves of the system decrease with higher values of transmit power and the latency threshold in both cases. The system performance also improves with an increase in the number of edge nodes and high transmit power for Nakagami-m fading case. Whereas this pattern is not substantial for Rayleigh fading case.

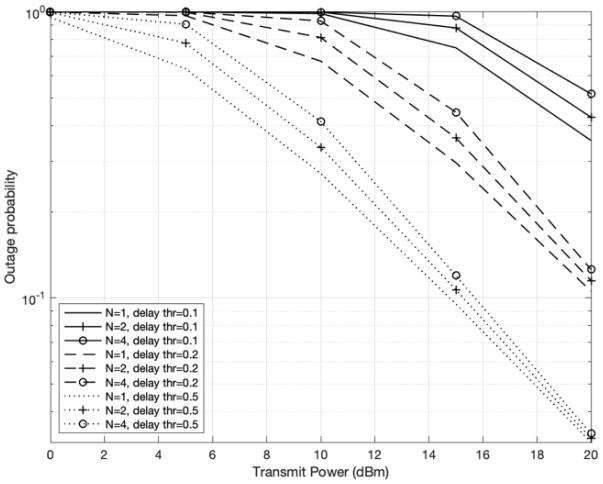


Figure 4.2: The outage probability curves of the TDMA model, Rayleigh fading channels, with B = 50 Mbits and W = 200 MHz versus transmit power for different delay threshold values and different number of edge nodes.

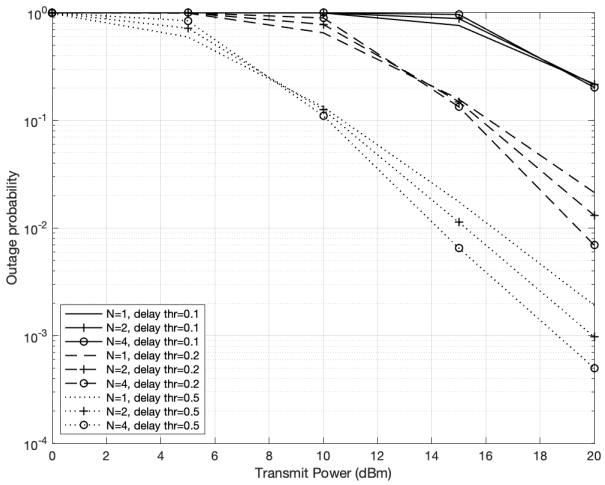


Figure 4.3: The outage probability curves of the TDMA model, Nakagami-m fading channels, with B = 50 Mbits, W = 200 MHz, and m = 2 versus transmit power for different delay threshold values and different number of edge nodes.

Figures 4.4 and 4.5 show the relationship between the delay-outage probability and task size. Here, transmit power is P = 20 dBm, while task size B varies from 20 to 200 Mbits. We also select different values of the latency threshold for this model, namely t_{thr} = 0.1, 0.2, 0.5 seconds, and different number of edge nodes, i.e., N = 1, 2, 4. It can be seen that the delay-outage probability curves increase with task size. At the same time, the system performance can be improved by using higher values of the latency threshold. Nevertheless, more edge nodes negatively affect the system performance in Rayleigh fading channels. This is not the case for Nakagami-m fading channels with smaller task sizes, where a larger number of edge nodes improve the system performance.

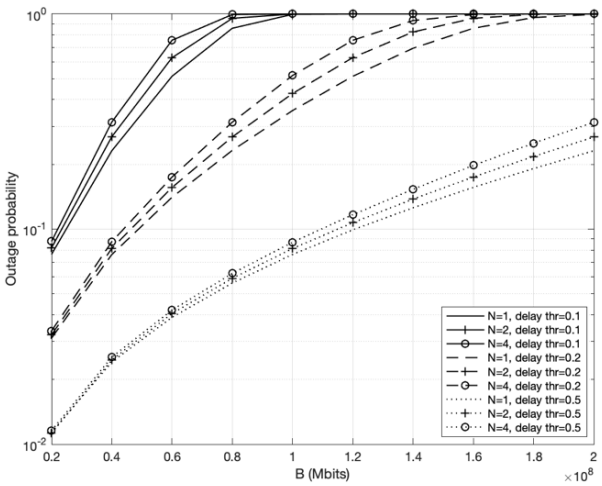


Figure 4.4: The outage probability curves of the TDMA model, Rayleigh fading channels, with P = 20 dBm and W = 200 MHz versus task size for different delay threshold values and different number of edge nodes.

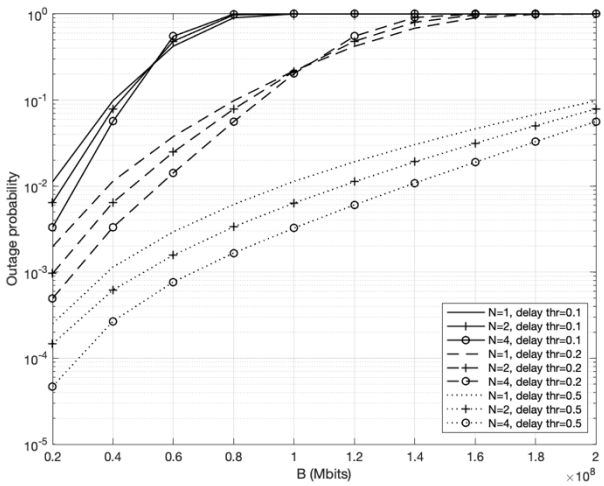


Figure 4.5: The outage probability curves of the TDMA model, Nakagami-m fading channels, with P= 20 dBm, W = 200 MHz, and m = 2 versus task size for different delay threshold values and different number of edge nodes.

Figures 4.6 and 4.7 demonstrate the delay-outage probability curves decrease based on higher parameters of channel bandwidth in TDMA model with Rayleigh and Nakagami-m fading channels. In this case, we choose channel bandwidth W with parameters from 20 to 200 MHz. In addition, the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds and the number of edge nodes, i.e., N = 1, 2, 4. Obviously, higher values of the latency threshold guarantee the delay-outage probability curves decrease. On the contrary, using more edge nodes worsens the system performance significantly in Rayleigh fading channels. For the Nakagami-m fading channels, a larger number of edge nodes improves the system performance at higher values of channel bandwidth and latency threshold.

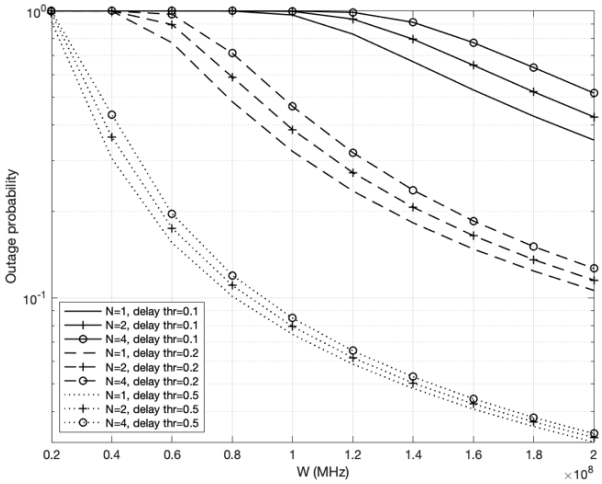


Figure 4.6: The outage probability curves of the TDMA model, Rayleigh fading channels, with P = 20 dBm and B = 50 Mbits versus channel bandwidth for different delay threshold values and different number of edge nodes.

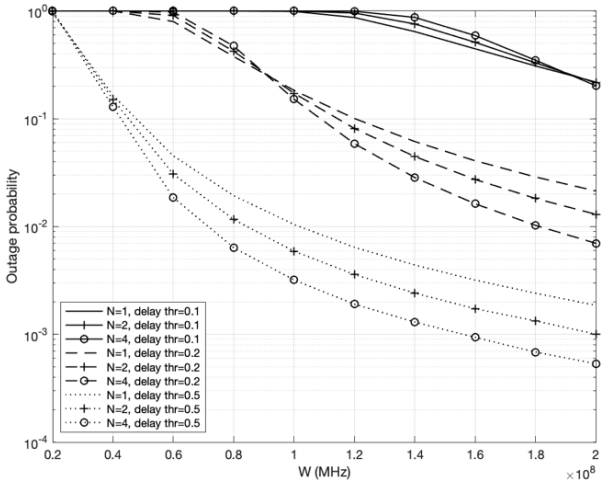


Figure 4.7: The outage probability curves of the TDMA model, Nakagami-m fading channels, with P = 20 dBm, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values and different number of edge nodes.

It can be concluded from the simulations above that the reliability metric in TDMA model with Rayleigh fading channels and Nakagami-m fading channels depends on transmit power, task size, and channel bandwidth values. Specifically, there is the delay-outage probability curves decrease based on higher values of transmit power and channel bandwidth. Conversely, the delay-outage probability curves increase with task size in both cases. In addition, higher values of the latency threshold improve the system performance considerably in both models. However, a larger number of edge nodes slightly improves the system performance in Nakagami-m fading channels with different parameters. While the utilization of more edge nodes leads to the system performance deterioration in Rayleigh fading channels.

4. 2 FDMA model for task offloading

In this model, the user offloads a complex task to all edge nodes at the same time through the appropriate frequency band designed for each link [58].

4. 2. 1 System model

The usage of the frequency-division multiple access model for edge computing is demonstrated in Figure 4.8. In this scheme, *B* bits of the offloaded task is divided between *N*

edge nodes, and thus, each edge node processes b_i bits of the data, for $i \in N$. However, in this scheme, channel bandwidth W and transmit power P of the user device are also divided between N edge nodes, and therefore, each link obtains bandwidth of w_i and transmit power of p_i, respectively, for $i \in N$.

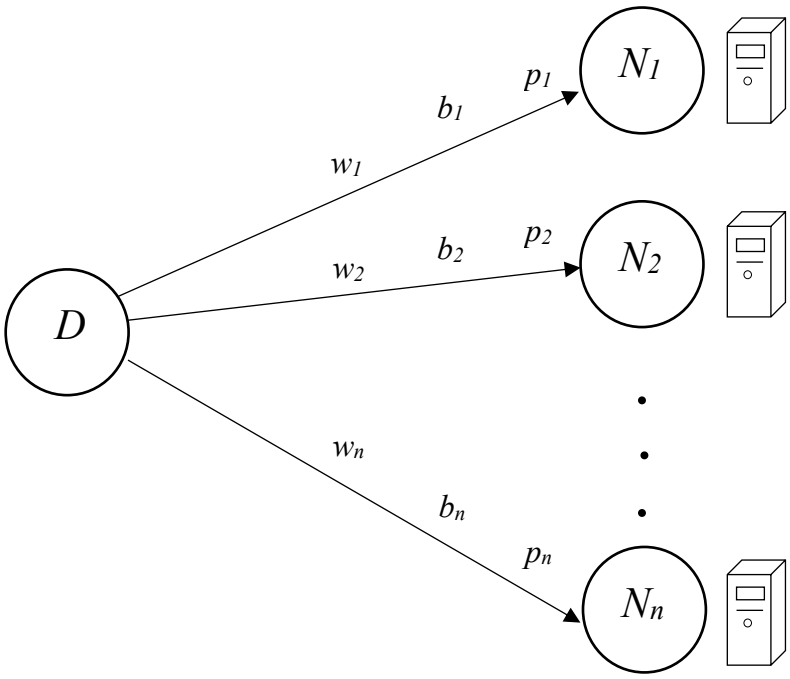


Figure 4.8: Edge computation offloading with FDMA model.

Hence, a transmission rate from the user device to each edge node can be obtained from Shannon formula [67]:

$$R_i = w_i \log_2 \left(1 + \frac{p_i |g_i|^2}{w_i N_0}\right), \tag{4.15}$$

where N_0 is the spectral density of the noise.

In the FDMA based edge computing system, the user device also divides the task between N edge nodes, and thus, each node obtains b_i bits of the data for computation, for $i \in N$. The transmission of the divided data from the user device to all edge nodes is organized simultaneously with the assigned power p_i and channel bandwidth w_i. After that, edge nodes process the received data b_i and send the results back. Nevertheless, the time of sending the results to the user device can be omitted due to the small data size. Consequently, the total time

of task offloading in FDMA model is the sum of the transmission time of the divided data from the user device and computation time of the received data at the edge nodes. Thus, the total time of task offloading from the user device to all edge nodes can be determined from (3.2), (3.10), and (4.15) as:

$$t_i = \max_i \left(\frac{b_i}{w_i \log_2\left(1+\frac{p_i|g_i|^2}{w_i N_0}\right)} + \frac{Kb_i}{c_i} \right). \quad (4.16)$$

The total transmit time is dominated by the MEC with the largest delay, therefore, we introduced $\max_i$ in this expression. Thus, we can obtain the probability of the delay-outage probability from (3.4) and (4.16):

$$Pr_{out,II} = Pr\left(\max_i \left(\frac{b_i}{w_i \log_2\left(1+\frac{p_i|g_i|^2}{w_i N_0}\right)} + \frac{Kb_i}{c_i} \right) > t_{thr} \right). \quad (4.17)$$

4. 2. 2 Performance analysis

Finding a closed-form solution for the expression (4.17) is not straightforward. Instead, we consider the special case where all parameters are equal, i.e., $b_i = b_0$, $c_i = c_0$, $p_i = p_0$, and $w_i = w_0$. In that case, MEC with the largest delay should have the worst channel gain g_i, therefore, we can obtain from (4.17):

$$Pr_{out,II} = Pr\left(\frac{b_i}{w_i \log_2\left(1+\frac{p_{i\min_i}|g_i|^2}{w_i N_0}\right)} > t_{thr} - \frac{Kb_i}{c_i} \right)$$

$$= Pr\left(\log_2\left(1+\frac{p_{i\min_i}|g_i|^2}{w_i N_0}\right) < \frac{b_i}{w_i\left(t_{thr}-\frac{Kb_i}{c_i}\right)} \right)$$

$$= Pr\left(\min_i |g_i|^2 < \frac{\left(2^{\frac{b_i}{w_i\left(t_{thr}-\frac{Kb_i}{c_i}\right)}}-1\right) w_i N_0}{p_i} \right)$$

$$= 1 - \prod_{i=1}^{n} Pr\left(|g_i|^2 > \frac{\left(2^{\frac{b_i}{w_i\left(t_{thr} - \frac{Kb_i}{c_i}\right)}} - 1 \right) w_i N_0}{p_i} \right). \quad (4.18)$$

Hence, for the case of Rayleigh fading channels, the delay-outage probability can be determined from (4.18) through the CDF of the exponential RV g_i, where $f_{|g_i|^2}(x) = \frac{1}{\lambda} e^{-\frac{1}{\lambda}x}$:

$$Pr_{out,II} = 1 - \prod_{i=1}^{n} \left(e^{-\frac{\gamma_{thr2,i} w_i N_0}{\lambda p_i}} \right), \quad (4.19)$$

for $\gamma_{thr2,i} \geq 0$, where $\gamma_{thr2,i} = 2^{\frac{b_i}{w_i\left(t_{thr} - \frac{Kb_i}{c_i}\right)}} - 1$.

With regard to Nakagami-m fading channels, the delay-outage probability can be obtained from (4.18) through the CDF of the gamma RV g_i, where $f_{|g_i|^2}(x) = \frac{m^m x^{m-1}}{\beta^m \Gamma(m)} e^{-\frac{mx}{\beta}}$:

$$Pr_{out,II} = 1 - \prod_{i=1}^{n} \left(e^{-\frac{m\gamma_{thr2,i} w_i N_0}{\beta p_i}} \sum_{k=0}^{m-1} \frac{1}{k!} \left(\frac{m\gamma_{thr2,i} w_i N_0}{\beta p_i} \right)^k \right), \quad (4.20)$$

for $\gamma_{thr2,i} \geq 0$, where $\gamma_{thr2,i} = 2^{\frac{b_i}{w_i\left(t_{thr} - \frac{Kb_i}{c_i}\right)}} - 1$.

Furthermore, we can determine the optimized value of the offloaded task size b_i in order to minimize the delay. Specifically, this optimization problem can be expressed as:

$$\min \max \left(\frac{b_i}{E(R_i)} \right) = \min \max \left(\frac{b_i}{E\left(w_i \log_2 \left(1 + \frac{p_i |g_i|^2}{w_i N_0} \right) \right)} \right) \quad (4.21a)$$

$$\text{s.t.} \sum_{i=1}^{n} b_i \geq B. \quad (4.21b)$$

Let us assume that the statistics of the channel gain, i.e., path-loss α of g_i, are known, and thus, the task size b_i can be defined as:

$$b_i = \left\lceil \frac{E(R_i)}{\sum_{i=1}^{n} E(R_i)} \right\rceil = \left\lceil \frac{E\left(w_i \log_2\left(1 + \frac{p_i |g_i|^2}{w_i N_0}\right)\right)}{\sum_{i=1}^{n} E\left(w_i \log_2\left(1 + \frac{p_i |g_i|^2}{w_i N_0}\right)\right)} \right\rceil, \tag{4.22}$$

where '⌈⌉' represents ceiling of the outcome.

4. 2. 3 Numerical results

In this part, we demonstrate simulation results to assess the system reliability numerically. To this end, we consider FDMA model with Rayleigh fading channels and Nakagami-m fading channels. The respective simulation includes 1000 000 trials to obtain a proper average. Let us assume task size B = 50 Mbits, channel bandwidth W = 200 MHz, carrier frequency f_c = 28 GHz, path-loss exponent α = 3, cycle frequency for each bit of task execution K = 10, spectral density of the noise N_0 = – 174 dBm/Hz, and edge node's CPU cycle frequency c_i is 50 GHz. For Nakagami-m fading case, Nakagami parameter m = 2.

Figures 4.9 and 4.10 show the impact of transmit power on the system performance with different parameters of the latency threshold and different number of edge nodes in Rayleigh and Nakagami-m fading channels, respectively. In our scheme, transmit power varies from 0 to 20 dBm, while the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds, and the number of edge nodes is N = 1, 2, 4. It can be seen, that the system performance improves with higher values of transmit power. Moreover, higher latency thresholds result in the delay-outage probability curves decrease for both cases. Nevertheless, the system performance degrades with more edge nodes involved in the process.

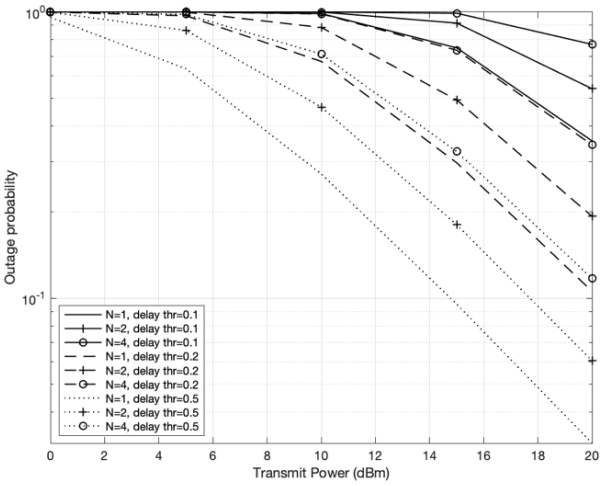


Figure 4.9: The outage probability curves of the FDMA model, Rayleigh fading channels, with B = 50 Mbits and W = 200 MHz versus transmit power for different delay threshold values and different number of edge nodes.

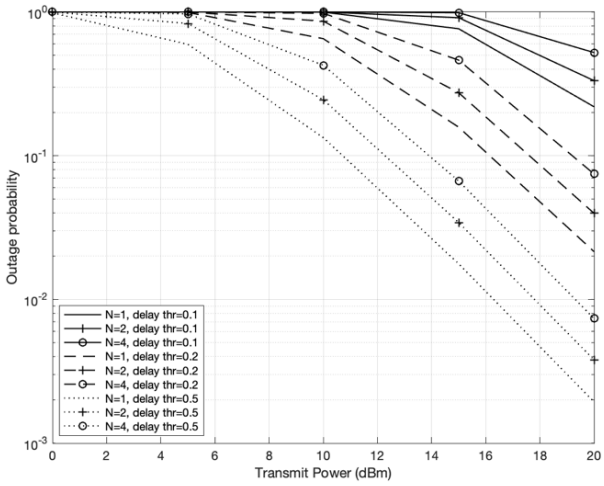


Figure 4.10: The outage probability curves of the FDMA model, Nakagami-m fading channels, with B = 50 Mbits, W = 200 MHz, and m = 2 versus transmit power for different delay threshold values and different number of edge nodes.

From Figures 4.11 and 4.12, we can observe the dependency of the system delay-outage probability on the task size. In these schemes, transmit power is 20 dBm, while task size B varies from 20 to 200 Mbits. The latency threshold also has different values, specifically t_{thr} = 0.1, 0.2, 0.5 seconds. Obviously, the delay-outage probability curves increase with task size. However, the system performance can be improved by means of the higher latency threshold. At the same time, more edge nodes also deteriorate the system performance in Rayleigh fading channels and Nakagami-m fading channels.

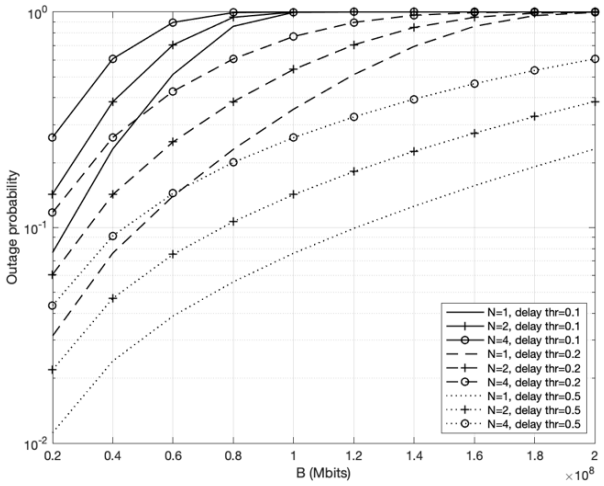


Figure 4.11: The outage probability curves of the FDMA model, Rayleigh fading channels, with P = 20 dBm and W = 200 MHz versus task size for different delay threshold values and different number of edge nodes.

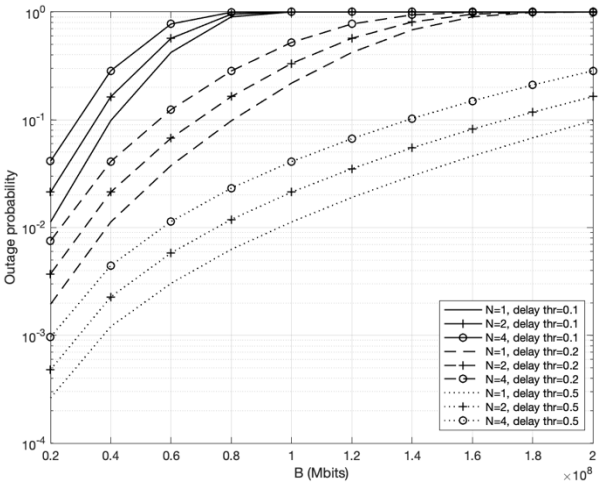


Figure 4.12: The outage probability curves of the FDMA model, Nakagami-m fading channels, with P = 20 dBm, W = 200 MHz, and m = 2 versus task size for different delay threshold values and different number of edge nodes.

In Figures 4.13 and 4.14, we can observe a slight decrease in the delay-outage probability curves based on the higher channel bandwidth parameters in FDMA model with Rayleigh and Nakagami-m fading channels. In these schemes, we assume channel bandwidth W with parameters from 20 to 200 MHz. Furthermore, the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds, and the number of edge nodes, that is, N = 1, 2, 4. It can be seen that the higher values of the latency threshold provide a significant improvement in the system performance. Conversely, the usage of more edge nodes negatively impacts the system performance in both cases.

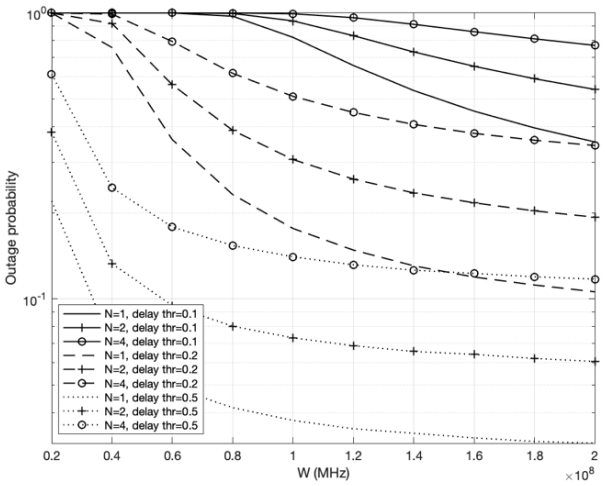


Figure 4.13: The outage probability curves of the FDMA model, Rayleigh fading channels, with P = 20 dBm and B = 50 Mbits versus channel bandwidth for different delay threshold values and different number of edge nodes.

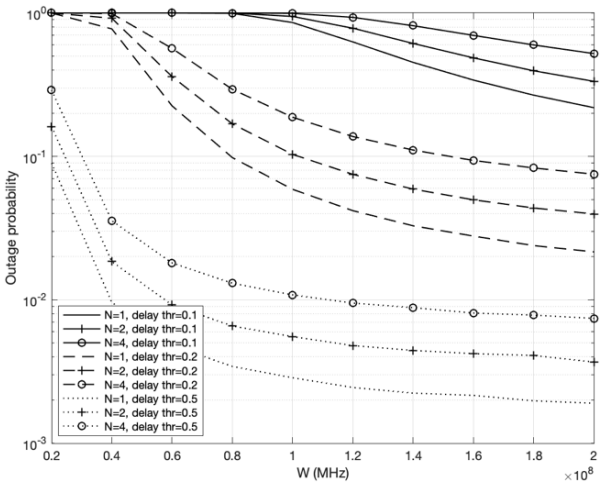


Figure 4.14: The outage probability curves of the FDMA model, Nakagami-m fading channels, with P = 20 dBm, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values and different number of edge nodes.

Thus, after obtaining numerical results, we can summarize that the reliability metric in FDMA model with Rayleigh fading channels and Nakagami-m fading channels also depends on transmit power, task size, and channel bandwidth values. In particular, the system performance improves considerably with higher values of transmit power. However, the delay-outage probability curves decrease with higher channel bandwidth values is less significant in FDMA model. Conversely, a larger task size deteriorates the system performance in both cases. With regard to the latency threshold, its higher values improve the system performance remarkably. This does not apply to the number of edge nodes, since a larger number of edge nodes, involved in the process, worsens the reliability of the system in Nakagami-m fading channels and Rayleigh fading channels.

4. 3 Task offloading with capacity achieving codes

In this model, the user device also offloads a complex task to all edge nodes at the same time and same frequency. This can be realized either by proper separation of space using the space-division multiple access scheme [59] or by efficient management of interference. Thus, the maximum transmission rate between the user device and edge nodes can be achieved. For

interference management, we can use superposition coding and successive interference cancellation (SIC) concepts which are also called non-orthogonal multiple access scheme [54], [59], [68], and [69].

4. 3. 1 System model

The usage of capacity achieving schemes for edge computing is demonstrated in Figure 4.15. In this scheme, the user device transmits complex task to all edge nodes via common channel bandwidth, i.e., each link obtains W bandwidth. At the same time, transmit power P of the user device is divided between N edge nodes, and thus, each link obtains transmit power of p_i, for $i \in N$.

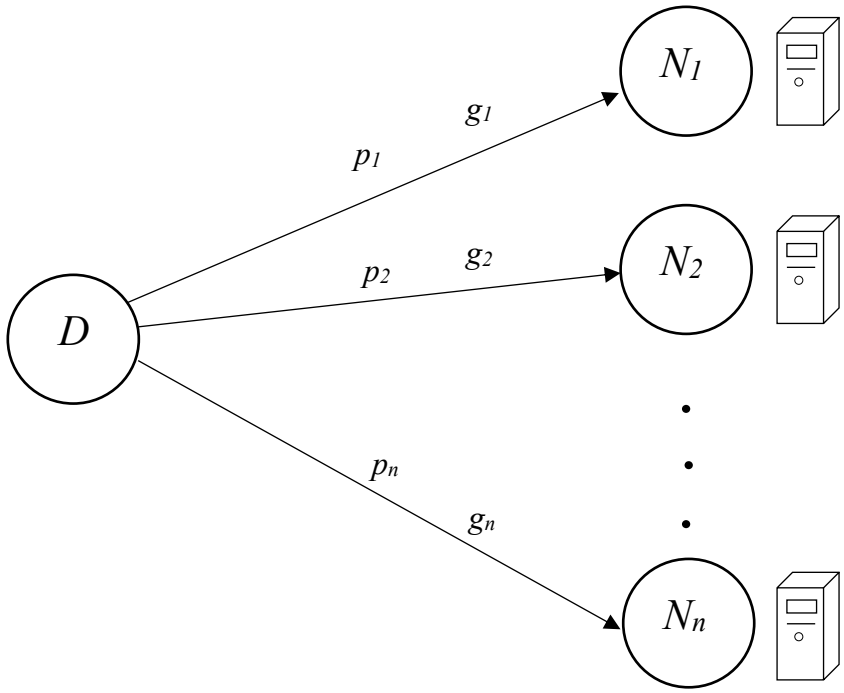


Figure 4.15: Edge computation offloading with capacity achieving codes.

An ergodic sum-rate is defined as $\overline{R} = E(R)$ and can be obtained from Shannon formula [54]:

$$\overline{R} = E\left(W\log_2\left(1 + \frac{\sum_{i=1}^{n} p_i|g_i|^2}{\sigma^2}\right)\right). \tag{4.23}$$

An upper-bound for the ergodic sum-rate can be obtained from (4.23) and Jensen's inequality [54]:

$$\overline{R} \leq W\log_2\left(1 + \frac{E\left(\sum_{i=1}^{n} p_i|g_i|^2\right)}{\sigma^2}\right)$$

$$= W\log_2\left(1 + \frac{\sum_{i=1}^{n} p_i\Omega_i}{\sigma^2}\right). \quad (4.24)$$

In addition, a lower-bound for the ergodic sum-rate can be also obtained from (4.23) and Jensen's inequality [54]:

$$\overline{R} \geq W\log_2\left(1 + \frac{\exp\left(E\left(\ln\left(\sum_{i=1}^{n} p_i|g_i|^2\right)\right)\right)}{\sigma^2}\right)$$

$$= \frac{1}{N}\sum_{i=1}^{n} W\log_2\left(1 + \frac{p_i\Omega_i}{\sigma^2}\exp\left(\sum_{i=1}^{n-1}\frac{1}{i} - k\right)\right), \quad (4.25)$$

where $i \in N$ and k is the Euler's constant.

By a tight approximation of the total achievable transmission rate, we obtain [54]:

$$R = \sum_{i=1}^{n} W\log_2\left(1 + \frac{p_i|g_i|^2}{\sigma^2}\right) = W\log_2\left(1 + \sum_{i=1}^{n}\frac{p_i|g_i|^2}{\sigma^2}\right). \quad (4.26)$$

Then, a task transmission time from the user device to edge node is obtained using (4.26):

$$t_{tr} = \frac{B}{W\log_2\left(1 + \sum_{i=1}^{n}\frac{p_i|g_i|^2}{\sigma^2}\right)}. \quad (4.27)$$

In the SDMA based edge computing system, the user device transmits task B to N edge nodes in separated dimensions, for $i \in N$. The transmission of the data from the user device to all edge nodes is organized simultaneously via common bandwidth W [58]. Further, edge nodes process the received data and send the results back. However, the time of sending the results to the user device can be ignored due to the small size. Consequently, the total time of task offloading in capacity achieving schemes is the sum of the transmission time of the data from the user device to all edge nodes and the computation times of the received data at the edge nodes. Thus, it can be determined from (3.2), (3.10), and (4.27) as:

$$t = \frac{B}{W\log_2\left(1+\sum_{i=1}^{n}\frac{p_i|g_i|^2}{\sigma^2}\right)} + \sum_{i=1}^{n}\frac{KB}{c_i}. \quad (4.28)$$

We can define the delay-outage probability from (3.4) and (4.28) for this scenario:

$$Pr_{out,III} = Pr\left(\frac{B}{W\log_2\left(1+\sum_{i=1}^{n}\frac{p_i|g_i|^2}{\sigma^2}\right)} + \sum_{i=1}^{n}\frac{KB}{c_i} > t_{thr}\right). \quad (4.29)$$

4. 3. 2 Performance analysis

For this scenario, we can obtain the delay-outage probability after simplification of the formula (4.29):

$$\begin{aligned} Pr_{out,III} &= Pr\left(\frac{B}{W\log_2\left(1+\sum_{i=1}^{n}\frac{p_i|g_i|^2}{\sigma^2}\right)} > t_{thr} - \sum_{i=1}^{n}\frac{KB}{c_i}\right) \\ &= Pr\left(\log_2\left(1+\sum_{i=1}^{n}\frac{p_i|g_i|^2}{\sigma^2}\right) < \frac{B}{W\left(t_{thr}-\sum_{i=1}^{n}\frac{KB}{c_i}\right)}\right) \\ &= Pr\left(\sum_{i=1}^{n} p_i|g_i|^2 < \left(2^{\frac{B}{W\left(t_{thr}-\sum_{i=1}^{n}\frac{KB}{c_i}\right)}} - 1\right)\sigma^2\right). \quad (4.30) \end{aligned}$$

For the Rayleigh fading channel case, we first consider a special case of channel gains with identical means. Therefore, to determine the delay-outage probability, the characteristic function can be utilized. Hence, the sum of independent exponential RVs will have Erlang distribution, namely $f_{\sum_{i=1}^{n} p_i|g_i|^2}(x) = \frac{\lambda^n x^{n-1}}{(n-1)!}e^{-\lambda x}$ [60] and [70]. In addition, CDF of the sum of independent exponential RVs, and therefore, the delay-outage probability can be obtained from PDF of Erlang distribution:

$$Pr_{outIII} = 1 - e^{-\frac{\gamma_{thr3,i}\sigma^2}{\lambda}}\sum_{i=0}^{n-1}\frac{1}{i!}\left(\frac{\gamma_{thr3,i}\sigma^2}{\lambda}\right)^i, \quad (4.31)$$

for $\gamma_{thr3,i} \geq 0$, where $\gamma_{thr3,i} = 2^{\frac{B}{W\left(t_{thr}-\sum_{i=1}^{n}\frac{KB}{c_i}\right)}} - 1$.

For the Nakagami-m fading channel case, to find the delay-outage probability, we need to derive CDF of the sum of independent gamma RVs, which in the general case can be obtained from the nested weighted sum of gamma CDFs [51], [57], and [71]:

$$Pr_{outIII} = \sum_{i=1}^{n} \sum_{k=1}^{m_i} A_{i,k} F_{p_i|g_i|^2}(x), \tag{4.32}$$

where $F_{p_i|g_i|^2}(x)$ is the CDF of gamma RV $p_i|g_i|^2$ and can be expressed $F_{p_i|g_i|^2}(x) = 1 - e^{-\frac{m_i \gamma_{thr3,i} \sigma^2}{\gamma_i}} \sum_{k=0}^{m_i-1} \frac{1}{k!} \left(\frac{m_i \gamma_{thr3,i} \sigma^2}{\gamma_i}\right)^k$. Whereas $A_{i,k}$ represents weights, which can be derived from the expression below [51] and [57]:

$$A_{i,m_i-k} = \frac{1}{k} \sum_{j=1}^{k} \sum_{q=1,q\neq i}^{n} \frac{m_q}{\gamma_i^j} \left(\frac{1}{\gamma_j} - \frac{1}{\gamma_q}\right)^{-j} A_{i,m_i-k+j} \tag{4.33}$$

with

$$A_{i,m_i} = \frac{\gamma_i^{m_i}}{\prod_{h=1}^{n} \gamma_h^{m_h}} \prod_{j=1,j\neq i}^{n} \left(\frac{1}{\gamma_j} - \frac{1}{\gamma_i}\right)^{-m_j}. \tag{4.34}$$

In the scenario with the analogous channel parameters (e.g., distance or path-loss), we can observe that the sum of independent gamma RVs also will have Erlang distribution. Consequently, we can define the delay-outage probability from the PDF of Erlang distribution:

$$Pr_{outIII} = 1 - e^{-\frac{m\gamma_{thr3,i}\sigma^2}{\gamma}} \sum_{k=0}^{nm-1} \frac{1}{k!} \left(\frac{m\gamma_{thr3,i}\sigma^2}{\gamma}\right)^k, \tag{4.35}$$

for $\gamma_{thr3,i} \geq 0$, where $\gamma_{thr3,i} = 2^{\frac{B}{W\left(t_{thr} - \sum_{i=1}^{n} \frac{KB}{c_i}\right)}} - 1$.

4. 3. 3 Numerical results

In this part, we present simulation outcomes to assess the system performance numerically. For this aim, we consider capacity achieving schemes with Rayleigh fading channels and Nakagami-m fading channels. Our simulation consists of 1000 000 trials in order to obtain an appropriate average. Let us assume task size B = 50 Mbits, channel bandwidth W = 200 MHz, carrier frequency f_c = 28 GHz, path-loss exponent α = 3, cycle frequency for each

bit of task execution $K = 10$, noise power $\sigma^2 = 8 \times 10^{-10}$ mWatt, and edge node's CPU cycle frequency c_i is 50 GHz. For Nakagami-m fading case, Nakagami parameter $m = 2$.

In Figures 4.16 and 4.17, we can observe the effect of transmit power on the system performance with different parameters of the latency threshold and different number of edge nodes in Rayleigh and Nakagami-m fading channels, respectively. In this case, transmit power varies from 0 to 20 dBm. At the same time, the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds and the number of edge nodes is N = 1, 2, 4. Obviously, higher transmit power parameters lead to system performance enhancement. In both schemes, the delay-outage probability curves decrease with higher latency thresholds. It can be seen that the reliability of the system improves as the number of edge nodes increases, especially for a higher latency threshold case, i.e., t_{thr} = 0.2, 0.5 seconds, where more edge nodes result in improved reliability.

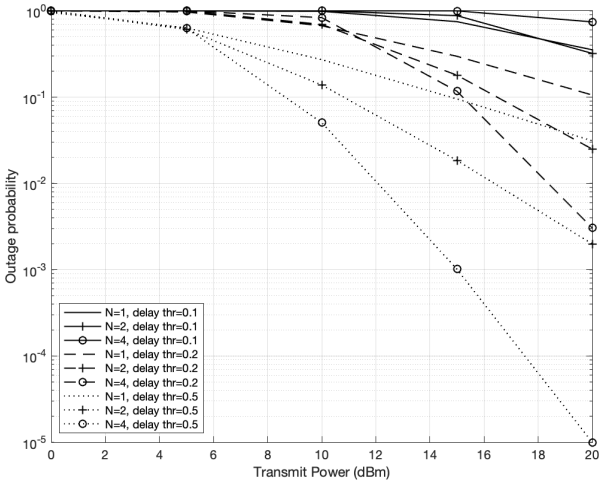


Figure 4.16: The outage probability curves of the SDMA scheme, Rayleigh fading channels, with B = 50 Mbits and W = 200 MHz versus transmit power for different delay threshold values and different number of edge nodes.

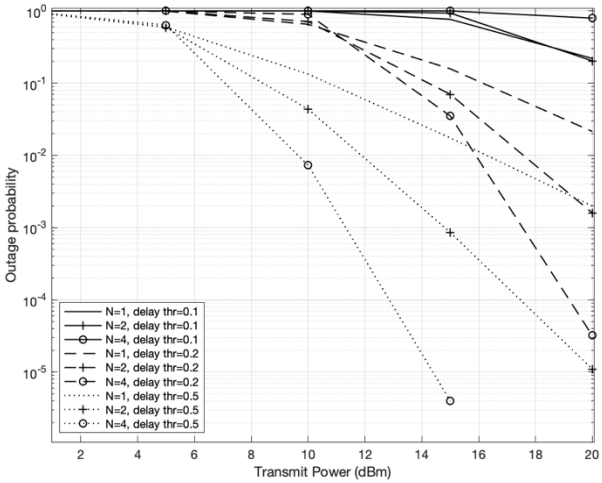


Figure 4.17: The outage probability curves of the SDMA scheme, Nakagami-m fading channels, with B = 50 Mbits, W = 200 MHz, and m = 2 versus transmit power for different delay threshold values and different number of edge nodes.

Figures 4.18 and 4.19 demonstrate the system performance dependency on a task size in Rayleigh and Nakagami-m fading channels, respectively. In these schemes, transmit power is 20 dBm, the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds, and the number of edge nodes is

N = 1, 2, 4. Whereas task size B varies from 20 to 200 Mbits in both schemes. Thus, we can observe the delay-outage probability curves increase with task size. At the same time, the reliability of the system improves with a higher latency threshold. In addition, the usage of more edge nodes enhances the system performance with a smaller task size.

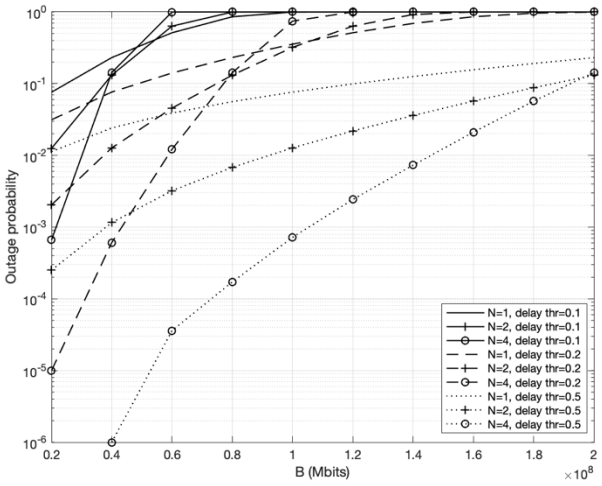


Figure 4.18: The outage probability curves of the SDMA scheme, Rayleigh fading channels, with P = 20 dBm and W = 200 MHz versus task size for different delay threshold values and different number of edge nodes

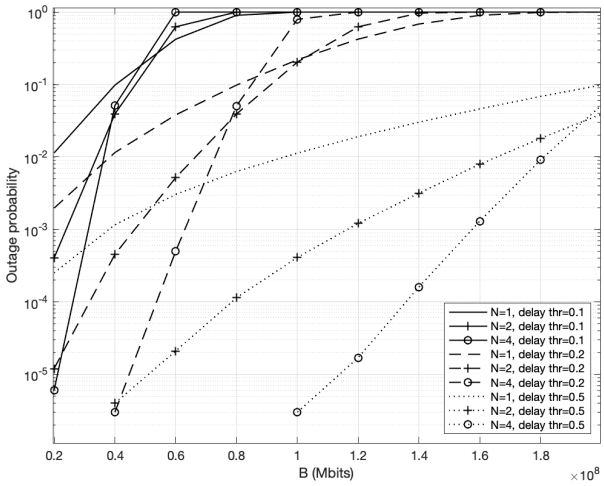


Figure 4.19: The outage probability curves of the SDMA scheme, Nakagami-m fading channels, with P = 20 dBm, W = 200 MHz, and m = 2 versus task size for different delay threshold values and different number of edge nodes.

From Figures 4.20 and 4.21, we can see the decrease in the delay-outage probability curves with higher channel bandwidth for capacity achieving schemes. In these schemes, we assume channel bandwidth W with parameters from 20 to 200 MHz, the latency threshold is t_{thr} = 0.1, 0.2, 0.5 seconds, and the number of edge nodes is N = 1, 2, 4. Clearly, higher values of the latency threshold enhance the system performance. Moreover, at higher latency threshold and channel bandwidth, more edge nodes significantly improve the system performance.

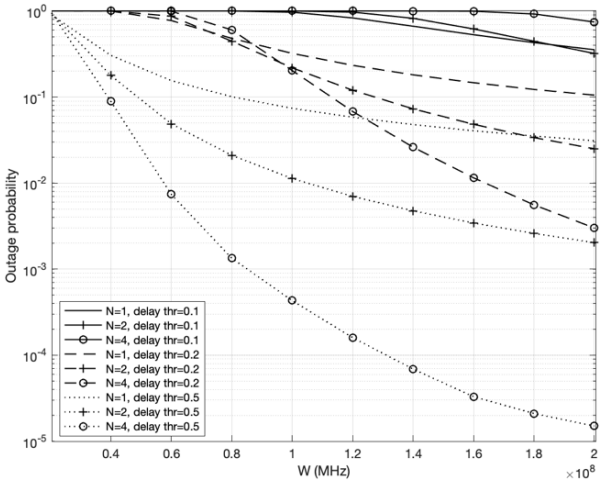


Figure 4.20: The outage probability curves of the SDMA scheme, Rayleigh fading channels, with P = 20 dBm and B = 50 Mbits versus channel bandwidth for different delay threshold values and different number of edge nodes.

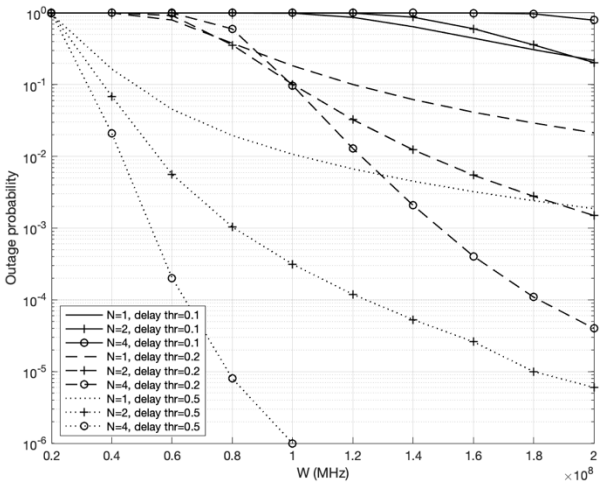


Figure 4.21: The outage probability curves of the SDMA scheme, Nakagami-m fading channels, with P = 20 dBm, B = 50 Mbits, and m = 2 versus channel bandwidth for different delay threshold values and different number of edge nodes.

Thus, it can be concluded that there is a tangible tie between the system performance and transmit power in capacity achieving schemes with Rayleigh fading channels and Nakagami-m fading channels. Moreover, the reliability of the system depends on the task size and channel bandwidth parameters. Specifically, the larger values of the transmit power and channel bandwidth improve the system performance, while the larger task size degrades it. Regarding the latency threshold, higher values improve the system performance. Additionally, a larger number of edge nodes enhances the reliability of the system in Nakagami-m fading channels and Rayleigh fading channels.

Finally, we observe differences in the system reliability based on several edge computing combining schemes, i.e., TDMA, FDMA, and capacity achieving schemes to obtain a comprehensive view of the subject. Figures 4.22, 4.23, and 4.24 show the delay-outage probability curves in Rayleigh fading channels versus transmit power, task size, and channel bandwidth, respectively. At the same time, Figures 4.25, 4.26, and 4.27 show the delay-outage probability curves in Nakagami-m fading channels. For all schemes, we select the latency threshold, i.e., $t_{thr} = 0.5$ second, and the number of edge nodes, namely $N = 2$. From the figures

below, we can conclude that edge computing model with capacity achieving scheme demonstrates the highest results in terms of the reliability metric. Since, in this scheme, all resources, namely frequency and time, are completely used for data transfer from the user device to edge nodes. This does not apply to edge computing with FDMA scheme, which is inferior in performance to the other two combining schemes.

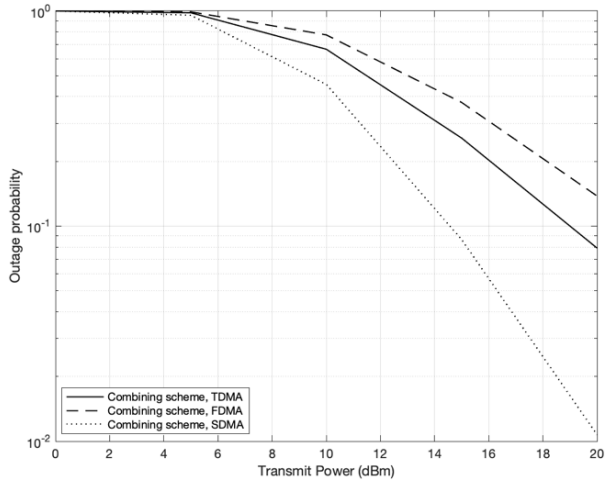


Figure 4.22: The outage probability curves of the Combining schemes, Rayleigh fading channels, with B = 50 Mbits, W = 200 MHz, t_{thr} = 0.5 second, and N = 2 versus transmit power.

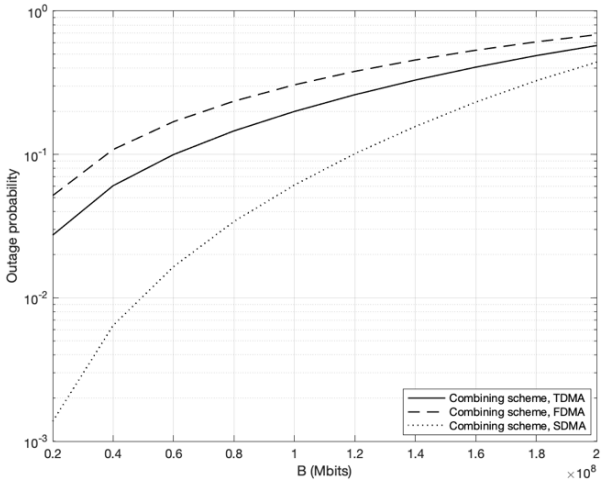


Figure 4.23: The outage probability curves of the Combining schemes, Rayleigh fading channels, with P = 20 dBm, W = 200 MHz, t_{thr} = 0.5 second, and N = 2 versus task size.

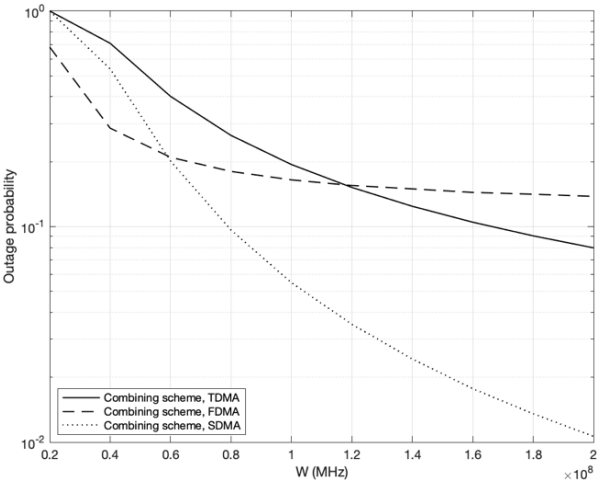


Figure 4.24: The outage probability curves of the Combining schemes, Rayleigh fading channels, with P = 20 dBm, B = 50 Mbits, t_{thr} = 0.5 second, and N = 2 versus channel bandwidth.

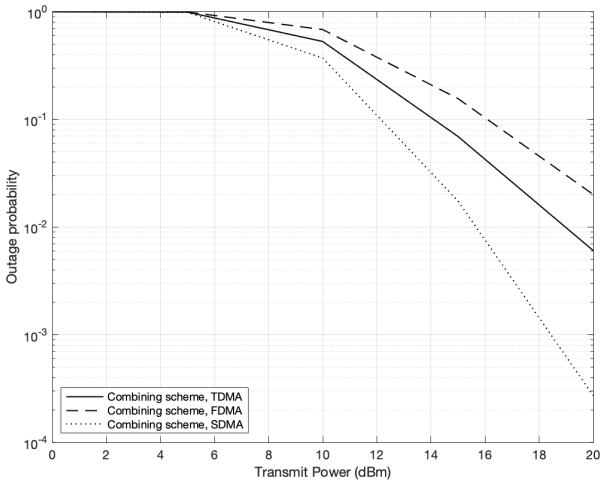


Figure 4.25: The outage probability curves of the Combining schemes, Nakagami-m fading channels, with B = 50 Mbits, W = 200 MHz, m = 2, t_{thr} = 0.5 second, and N = 2 versus transmit power.

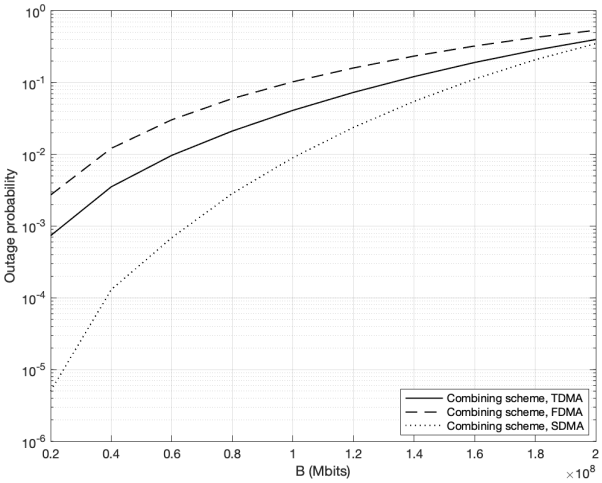


Figure 4.26: The outage probability curves of the Combining schemes, Nakagami-m fading channels, with P = 20 dBm, W = 200 MHz, m = 2, t_{thr} = 0.5 second, and N = 2 versus task size.

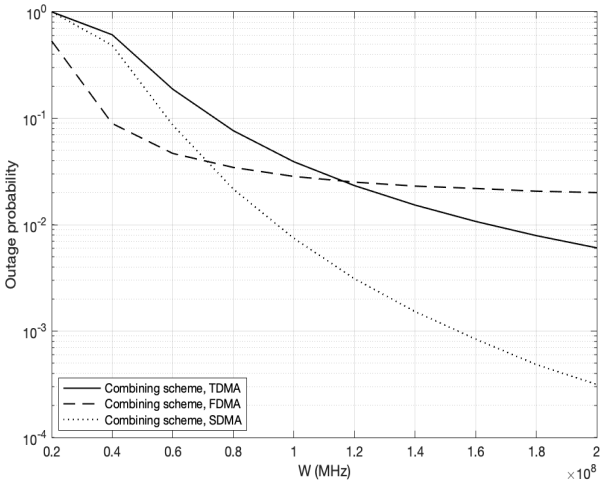


Figure 4.27: The outage probability curves of the Combining schemes, Nakagami-m fading channels, with P = 20 dBm, B = 50 Mbits, m = 2, t_{thr} = 0.5 second, and N = 2 versus channel bandwidth.

Chapter 5 – Conclusions

5. 1 Contributions

Edge computing has become a state-of-art means of equipping IoT devices with complementary computing and storage capabilities. In such a system, users partially or fully offload their assignments to edge nodes for completion, while edge nodes can distribute their resources among multiple user devices. As it was found during the process of investigation, the authors of previous works in this field developed various edge computing schemes with latency and energy consumption minimization, scalability, capacity, security, and privacy enhancement goals. However, due to the nature of IoT services and applications, latency is one of the most important and challenging characteristics. Therefore, in our research work, we implemented edge computing models to support IoT and assessed their performance against reliability metric in its delay-outage probability sense.

This thesis has accomplished three main objectives: studying edge computing models, evaluating the performance of selection schemes, and evaluating combining schemes for task offloading. The first part included an examination of the latest academic papers on task offloading, resource allocation, and service caching in edge computing. In this section, we investigated various edge computing schemes pursuing specific goals. Additionally, we identified a lack of research efforts to concurrently design different system architectures and compare them in terms of the delay-outage probability. Thus, in the second part, we designed selection schemes for task offloading with Rayleigh and Nakagami-m fading channels and examined them by changing system parameters. For the Nakagami-m fading case, we implemented edge computing with cache-free and cache-aided relay. Analytical and numerical results have demonstrated that edge computing model with cache-free relay outperforms edge computing model with cache-assisted relay according to the reliability. Moreover, edge computing model with cache-free relay, where the best channel gain between the relay and

edge node is selected, showed the highest results in this measurement. In the third part, we organized combining schemes with Rayleigh and Nakagami-m fading channels for task offloading and assessed their performance by adjusting system parameters. In this case, analytical and simulation results have demonstrated the advantage of capacity achieving schemes over TDMA and FDMA schemes in reliability. Therefore, in our work, two types of schemes with various parameters are considered simultaneously. As a result, we come to the conclusion that the latency of the task offloading from the user device to edge servers can be sufficiently reduced due to the appropriate architecture and favorable system and channel parameters.

5. 2 Future work

To get the initial insights into this subject, we used a single user device – multiple edge nodes model. Nevertheless, current work can be extended to a multiple user devices – multiple edge nodes model, where several user devices offload tasks to edge nodes simultaneously. In this model, edge servers can allocate resources using economic and pricing methods. Moreover, our work considered a static system, in which the system conditions or location remain unchanged, while this can be also spread to a dynamic system. In addition, to be closer to reality, one can also take into account the blockage effect when considering edge computing for IoT. Blockage is a random obstacle that threatens communication channels and is inherent in 5G technology. Partial task offloading, as well as local task offloading, can also be considered in future work. Finally, deep learning and reinforcement learning algorithms have attracted the attention of many research papers recently. Obviously, this topic is rather novel and broad, therefore, it has numerous research gaps for further investigation.

Bibliography

[1] T. K. Gannavaram V, U. M. Kandhikonda, R. Bejgam, S. B. Keshipeddi, and S. Sunkari, "A brief review on Internet of Things (IoT)," in *2021 Int. Conf. Comput. Commun. Inform. (ICCCI)*, 2021, pp. 1-6, doi: 10.1109/ICCCI50826.2021.9457009.

[2] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of Things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23022-23040, 2020, doi: 10.1109/ACCESS.2020.2970118.

[3] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1-36, Nov. 2019, doi: 10.1145/3362031.

[4] H. Ning, Y. Li, F. Shi, and L. T. Yang, "Heterogeneous edge computing open platforms and tools for internet of things," *Future Gener. Comput. Syst.*, vol. 106, pp. 67–76, May 2020, doi: 10.1016/j.future.2019.12.036.

[5] Y. Liu, M. Peng, G. Shou, Y. Chen, and S. Chen, "Toward edge intelligence: Multiaccess edge computing for 5G and Internet of Things," *IEEE Internet of Things J.*, vol. 7, no. 8, pp. 6722-6747, Aug. 2020, doi: 10.1109/JIOT.2020.3004500.

[6] F. Mashhadi, S. A. S. Monroy, A. Bozorgchenani, and D. Tarchi, "Optimal auction for delay and energy constrained task offloading in mobile edge computing," *Comput. Netw.,* vol. 183, pp. 1-10, Dec. 2020, doi: 10.1016/j.comnet.2020.107527.

[7] N. C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: A survey," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 2, pp. 954-1001, Secondquart. 2017, doi: 10.1109/COMST.2017.2647981.

[8]. H.Qiu, K. Zhu, N. Cong Luong, C. Yi, D. Niyato, and D. I. Kim, "Applications of auction and mechanism design in edge computing: A survey," *arXiv:2105.03559*, pp. 1-23, 2021. [Online]. Available: http://arxiv.org/abs/2105.03559.

[9] G. Premsankar, M. Di Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet of Things J.*, vol. 5, no. 2, pp. 1275-1284, Apr. 2018, doi: 10.1109/JIOT.2018.2805263.

[10] C. Cicconetti, M. Conti, and A. Passarella, "Architecture and performance evaluation of distributed computation offloading in edge computing," *Simul. Model. Pract. Theory*, vol. 101, pp. 1-22, May 2020, doi: 10.1016/j.simpat.2019.102007.

[11] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via Markov decision process in mobile-edge computing," *IEEE Internet of Things J.*, vol. 8, no. 4, pp. 2483-2493, Feb. 2021, doi: 10.1109/JIOT.2020.3033285.

[12] P. W. Khan, K. Abbas, H. Shaiba, A. Muthanna, A. Abuarqoub, and M. Khayyat, "Energy efficient computation offloading mechanism in multi-server mobile edge computing – An integer linear optimization approach," *Electronics*, vol. 9, no. 6, pp. 1-20, Jun. 2020, doi: 10.3390/electronics9061010.

[13] J. Zhang *et al.*, "Energy-latency trade-off for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things J.*, vol. 5, no. 4, pp. 2633-2645, Aug. 2018, doi: 10.1109/JIOT.2017.2786343.

[14] T. T. Vu, N. V. Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Offloading energy efficiency with delay constraint for cooperative mobile edge computing networks," in *2018 IEEE Glob. Commun. Conf. (GLOBECOM)*, 2018, pp. 1-6, doi: 10.1109/GLOCOM.2018.8647856.

[15] Q. Tang, H. Lyu, G. Han, J. Wang, and K. Wang, ''Partial offloading strategy for mobile edge computing considering mixed overhead of time and energy,'' *Neural Comput. Appl.*, vol. 32, no. 19, pp. 15383-15397, Aug. 2019, doi: 10.1007/s00521-019-04401-8.

[16] X. Hou *et al.*, "Reliable computation offloading for edge-computing-enabled software-defined IoV," *IEEE Internet of Things J.*, vol. 7, no. 8, pp. 7097-7111, Aug. 2020, doi: 10.1109/JIOT.2020.2982292.

[17] M. Condoluci and T. Mahmoodi, “Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges,” *Comput. Netw*., vol. 146, pp. 65–84, Dec. 2018, doi: 10.1016/j.comnet.2018.09.005.

[18] J. Liang, Z. Chen, C. Li, and B. Xia, “Delay outage probability of multi-relay selection for mobile relay edge computing system,” in *2019 IEEE/CIC Int. Conf. Commun. in China (ICCC)*, Aug. 2019, pp. 898-902, doi: 10.1109/ICCChina.2019.8855809.

[19] L. Chen, X. Kuang, D. Deng, F. Zhu, J. Xia, and L. Fan, “Multi-CAP assisted intelligent mobile edge computing networks for Internet of Things,” *IEEE Access*, vol. 8, pp. 137235-137243, 2020, doi: 10.1109/ACCESS.2020.3009686.

[20] J. Xia *et al.*, “Opportunistic access point selection for mobile edge computing networks,” *IEEE Trans. Wirel. Commun*., vol. 20, no. 1, pp. 695-709, Jan. 2021, doi: 10.1109/TWC.2020.3028102.

[21] Y. Zhu, Y. Hu, T. Yang, T. Yang, J. Vogt, and A. Schmeink, “Reliability-optimal offloading in low-latency edge computing networks: Analytical and reinforcement learning based designs,” *IEEE Trans. Veh. Technol*, vol. 70, no. 6, pp. 6058-6072, Jun. 2021, doi: 10.1109/TVT.2021.3073791.

[22] Y. Cai and P. Yuan, “Time-varying mobile edge computing for capacity maximization,” *IEEE Access*, vol. 8, pp. 142832-142842, 2020, doi: 10.1109/ACCESS.2020.3014275.

[23] Y. He, J. Ren, G. Yu, and Y. Cai, “D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks,” *IEEE Trans. Wirel. Commun*., vol. 18, no. 3, pp. 1750–1763, Mar. 2019, doi: 10.1109/TWC.2019.2896999.

[24] M. Thai, Y. Lin, Y. Lai, and H. Chien, “Workload and capacity optimization for cloud-edge computing systems with vertical and horizontal offloading,” *IEEE Trans. Netw*. *Serv. Manag*., vol. 17, no. 1, pp. 227-238, Mar. 2020, doi: 10.1109/TNSM.2019.2937342.

[25] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, “Intelligent offloading for collaborative smart city services in edge computing,” *IEEE Internet of Things J*., vol. 7, no. 9, pp. 7919-7927, Sept. 2020, doi: 10.1109/JIOT.2020.3000871.

[26] J. Ren, G. Yu, Y. Cai, Y. He, and F. Qu, “Partial offloading for latency minimization in mobile-edge computing,” in *IEEE Glob. Commun. Conf. (GLOBECOM)*, 2017, pp. 1–6, doi: 10.1109/GLOCOM.2017.8254550.

[27] Q. Wang and S. Chen, “Latency-minimum offloading decision and resource allocation for fog-enabled Internet of Things networks,” *Trans. Emerg. Telecommun. Technol*., vol. 31, no. 12, pp. 1-14, Jan. 2020, doi: 10.1002/ett.3880.

[28] H. Zhang, Y. Yang, X. Huang, C. Fang, and P. Zhang, “Ultra-low latency multi-task offloading in mobile edge computing,” *IEEE Access*, vol. 9, pp. 32569-32581, 2021, doi: 10.1109/ACCESS.2021.3061105.

[29] J. Feng, Q. Pei, F. R. Yu, X. Chu, and B. Shang, “Computation offloading and resource allocation for wireless powered mobile edge computing with latency constraint,” *IEEE Wirel. Commun. Lett.*, vol. 8, no. 5, pp. 1320-1323, Oct. 2019, doi: 10.1109/LWC.2019.2915618.

[30] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, “Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud RAN,” *IEEE Internet of Things J*., vol. 7, no. 4, pp. 3282-3299, Apr. 2020, doi: 10.1109/JIOT.2020.2967502.

[31] C. Liu, M. Bennis, and H. V. Poor, “Latency and reliability-aware task offloading and resource allocation for mobile edge computing,” in *IEEE Globecom Workshops (GC Wkshps)*, 2017, pp. 1-7, doi: 10.1109/GLOCOMW.2017.8269175.

[32] Y. Fu *et al.*, “Energy-efficient offloading and resource allocation for mobile edge computing enabled mission-critical internet-of-things systems,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2021, no. 1, pp. 1-16, Feb. 2021, doi: 10.1186/s13638-021-01905-7.

[33] S. Mu and Z. Zhong, “Computation offloading to edge cloud and dynamically resource-sharing collaborators in Internet of Things,” *EURASIP J. Wirel. Commun. Netw*, vol. 2020, no. 1, pp. 1-21, Dec. 2020, doi: 10.1186/s13638-020-01865-4.

[34] Y. Wang, X. Tao, Y. T. Hou, and P. Zhang, “Effective capacity-based resource allocation in mobile edge computing with two-stage tandem queues,” *IEEE Trans. Commun*., vol. 67, no. 9, pp. 6221-6233, Sept. 2019, doi: 10.1109/TCOMM.2019.2920835.

[35] M. Qin, L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Power-constrained edge computing with maximum processing capacity for IoT networks," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 4330-4343, Jun. 2019, doi: 10.1109/JIOT.2018.2875218.

[36] S. Li *et al.*, "Joint congestion control and resource allocation for delay-aware tasks in mobile edge computing," *Wirel. Commun. Mob. Comput.*, vol. 2021, pp. 1-16, Jan. 2021, doi: 10.1155/2021/8897814.

[37] X. Xu, X. Liu, Z. Xu, C. Wang, S. Wan, and X. Yang, "Joint optimization of resource utilization and load balance with privacy preservation for edge services in 5G networks," *Mob. Netw. Appl.*, vol. 25, no. 2, pp. 713–724, Apr. 2020, doi: 10.1007/s11036-019-01448-8.

[38] E. Recayte and A. Munari, "Caching at the edge: Outage probability," in *2021 IEEE Wirel. Commun. Netw. Conf. (WCNC)*, 2021, pp. 1-6, doi: 10.1109/WCNC49053.2021.9417339.

[39] S. Zhong, S. Guo, H. Yu, and Q. Wang, "Cooperative service caching and computation offloading in multi-access edge computing," *Comput. Netw.*, vol. 189, pp. 1-11, Apr. 2021, doi: 10.1016/j.comnet.2021.107916.

[40] J. Xia *et al.*, "Cache-aided mobile edge computing for B5G wireless communication networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2020, no. 1, pp. 1-10, Jan. 2020, doi: 10.1186/s13638-019-1612-0.

[41] P. Yuan, Y. Cai, X. Huang, S. Tang, and X. Zhao, "Collaboration improves the capacity of mobile edge computing," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 10610-10619, Dec. 2019, doi: 10.1109/JIOT.2019.2940067.

[42] M. S. Aslanpour, S. S. Gill, and A. N. Toosi, "Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks, and standards for future research," *Internet of Things*, vol. 12, pp. 1-20, Dec. 2020, doi: 10.1016/j.iot.2020.100273.

[43] T. Cui and S. Li, "System movement space and system mapping theory for reliability of IoT," *Future Gener. Comput. Syst.*, vol. 107, pp. 70-81, Jun. 2020, doi: 10.1016/j.future.2020.01.040.

[44] M. S. Elbamby *et al.*, "Wireless edge computing with latency and reliability guarantees," *in Proc. IEEE*, vol. 107, no. 8, pp. 1717-1737, Aug. 2019, doi: 10.1109/JPROC.2019.2917084.

[45] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," in *Proc. IEEE*, vol. 106, no. 10, pp. 1834-1853, Oct. 2018, doi: 10.1109/JPROC.2018.2867029.

[46] I. Khuda, S. Mir, M. Ebrahim, and K. Raza, "Simulation and modeling of fading gain of the Rayleigh faded wireless communication channel using autocorrelation function and doppler spread," *J. Appl. Sci. Eng.*, vol. 22, no. 4, pp. 637-644, Dec. 2019, doi: 10.6180/jase.201912_22(4).0005.

[47] N. di Pietro, M. Merluzzi, E. C. Strinati, and S. Barbarossa, "Resilient design of 5G mobile-edge computing over intermittent mmwave links," pp. 1-46, 2019. *Available online: http://arxiv.org/abs/1901.01894.*

[48] A. Islam, A. Debnath, M. Ghose, and S. Chakraborty, "A survey on task offloading in multi-access edge computing," *J. Syst. Archit.*, vol. 118, pp. 1-16, Sept. 2021, doi: 10.1016/j.sysarc.2021.102225.

[49] B. Maham, "Millimeter wave CoMP system with opportunistic cell selection under blockage and interference," in *ICC 2021 - IEEE Int. Conf. Commun.,* 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500461.

[50] Y. Sun, Z. Ding, X. Dai, and O. A. Dobre, "On the performance of network NOMA in uplink CoMP systems: A stochastic geometry approach," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5084-5098, Jul. 2019, doi: 10.1109/TCOMM.2019.2906307.

[51] B. Maham and T. Svensson, "Performance analysis of millimeter wave CoMP networks under blockage," in *2021 IEEE 93rd Veh. Technol. Conf. (VTC2021-Spring)*, 2021, pp. 1-5, doi: 10.1109/VTC2021-Spring51267.2021.9448743.

[52] T. Rappaport, *Wireless communications: Principles and practice*. NJ, USA: Prentice-Hall, 1996.

[53] J. Miranda *et al.*, "Path loss exponent analysis in wireless sensor networks: Experimental evaluation," in *2013 11th IEEE Int. Conf. Ind. Inform. (INDIN)*, 2013, pp. 54-58, doi: 10.1109/INDIN.2013.6622857.

[54] B. Maham and P. Popovski, "Cognitive multiple-antenna network with outage and rate margins at the primary system," *IEEE Trans. Veh. Technol.*, vol. 64, no. 6, pp. 2409-2423, June 2015, doi: 10.1109/TVT.2014.2345588.

[55] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2619-2692, Oct. 1998, doi: 10.1109/18.720551.

[56] M. K. Simon and M.-S. Alouini, *Digital Communications Over Fading Channels*. NY, USA: Wiley, 2001.

[57] B. Maham and P. Popovski, "Capacity analysis of coordinated multipoint reception for mmWave uplink with blockages," *IEEE Trans. Veh. Technol*., vol. 69, no. 12, pp. 16299-16303, Dec. 2020, doi: 10.1109/TVT.2020.3041054.

[58] W. Rhee and J. M. Cioffi, "On the capacity of multiuser wireless channels with multiple antennas," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2580-2595, Oct. 2003, doi: 10.1109/TIT.2003.817441.

[59] D. Tse and P. Viswanath, *Fundamentals of Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005, pp. 166–227.

[60] A. Papoulis and S. U. Pillai*, Probability, random variables, and stochastic processes.* NY, USA: The McGraw-Hill Companies, 2002, pp. 169-242.

[61] J. G. Proakis, *Digital Communications*. NY, USA: McGraw-Hill, Inc., Fourth ed., 2001, pp. 17-62.

[62] J. Gil-Pelaez, "Note on the inversion theorem," *Biometrika*, vol. 38, no. 3-4, 1951, pp. 481–482, Dec. 1951, doi: 10.1093/biomet/38.3-4.481.

[63] V. K. Trivedi, M. K. Sinha, and P. Kumar, "Simplified approach for symbol error rate analysis of SC-FDMA scheme over Rayleigh fading channel," *ETRI J.*, vol. 40, no. 4, pp. 537-545, Aug. 2018, doi: 10.4218/etrij.2017-0286.

[64] J. J. Sanchez-Sanchez, M. C. Aguayo-Torres, and U. Fernandez-Plazaola, "BER Analysis for zero-forcing SC-FDMA over Nakagami-m fading channels," *IEEE Trans. Veh. Technol.*, vol. 60, no. 8, pp. 4077-4081, Oct. 2011, doi: 10.1109/TVT.2011.2165091.

[65] J. J. Sanchez-Sanchez, U. Fernandez Plazaola, M. C. Aguayo-Torres, and J. T. Entrambasaguas, "Closed-form BER expression for interleaved SC-FDMA with M-QAM," in *2009 IEEE 70th Veh. Technol. Conf. Fall*, 2009, pp. 1-5, doi: 10.1109/VETECF.2009.5378758.

[66] E. Levy, "On the density for sums of independent exponential, Erlang, and gamma variates," *Stat. Pap*., vol. 179, pp. 1-29, Jul. 2021, doi: 10.1007/s00362-021-01256-x.

[67] L. Dong and X. Meng, "Energy efficiency in multiuser transmission over parallel frequency channels," *IEEE Trans. Commun*., vol. 66, no. 9, pp. 4234-4248, Sept. 2018, doi: 10.1109/TCOMM.2018.2827957.

[68] Z. Ding, P. Fan, and H. V. Poor, "Impact of non-orthogonal multiple access on the offloading of mobile edge computing," *IEEE Trans. Commun*., vol. 67, no. 1, pp. 375-390, Jan. 2019, doi: 10.1109/TCOMM.2018.2870894.

[69] S. S. Yilmaz and B. Özbek, "Multi-helper NOMA for cooperative mobile edge computing," *IEEE Trans. Intell. Transp. Syst.*, pp. 1-10, Oct. 2021, doi: 10.1109/TITS.2021.3116421.

[70] C. Forbes, M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*. Hoboken, NJ, USA: Wiley, Mar. 2011, pp. 109-113.

[71] G. K. Karagiannidis, N. C. Sagias, and T. A. Tsiftsis, "Closed-form statistics for the sum of squared Nakagami-m variates and its applications," *IEEE Trans. Commun*., vol. 54, no. 8, pp. 1353-1359, Aug. 2006, doi: 10.1109/TCOMM.2006.878812.

Appendices

Appendix A

This Appendix contains supplementary MATLAB codes for numerical evaluation of the system performance. There are codes for selection schemes with Rayleigh and Nakagami-m fading channels versus different values of SNR, task size, channel bandwidth, latency threshold, and the number of edge nodes.

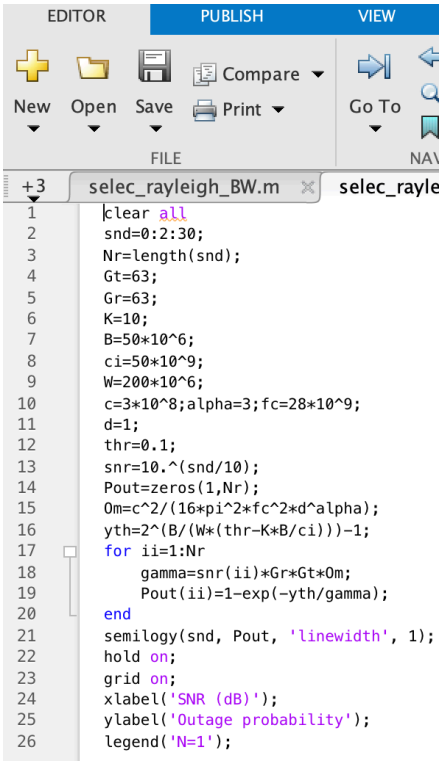


Figure A.1: Code of the delay-outage probability versus SNR, Rayleigh fading channels.

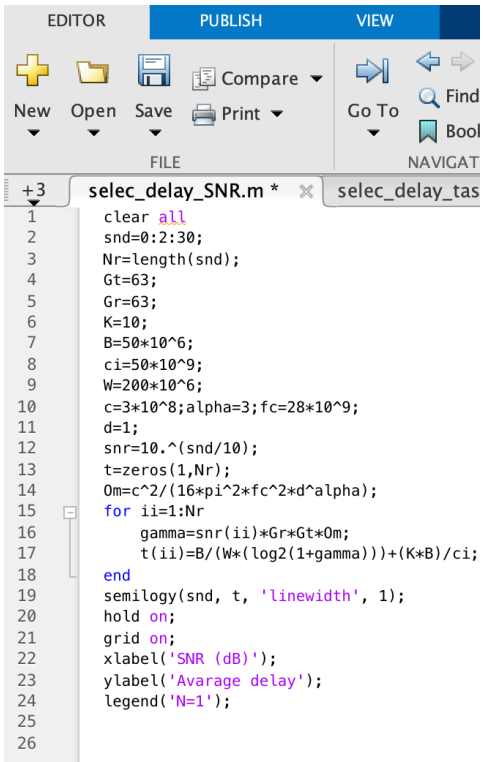


Figure A.2: Code of average delay versus SNR, Rayleigh fading channels.

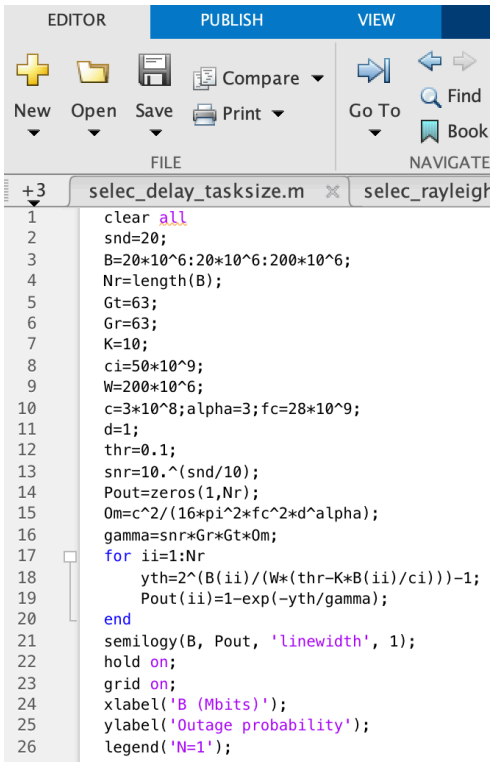


Figure A.3: Code of the delay-outage probability versus task size, Rayleigh fading channels.

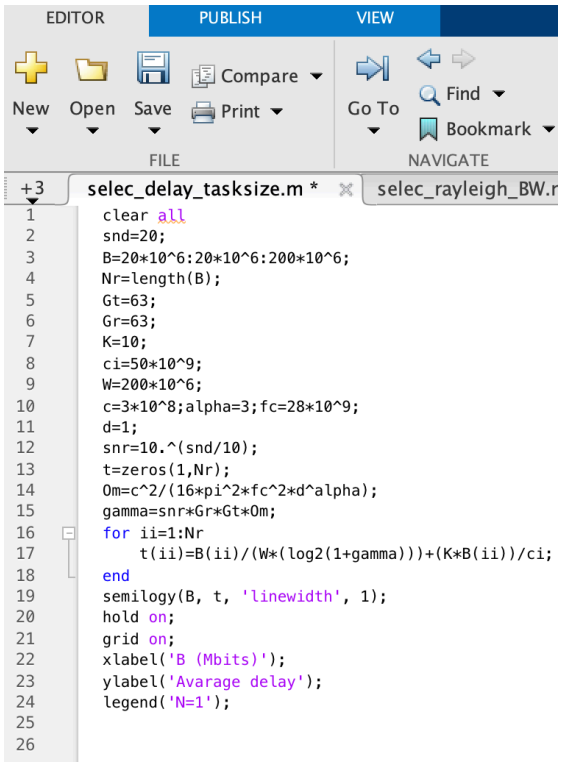


Figure A.4: Code of average delay versus task size, Rayleigh fading channels.

Figure A.5: Code of the delay-outage probability versus channel bandwidth, Rayleigh fading channels.

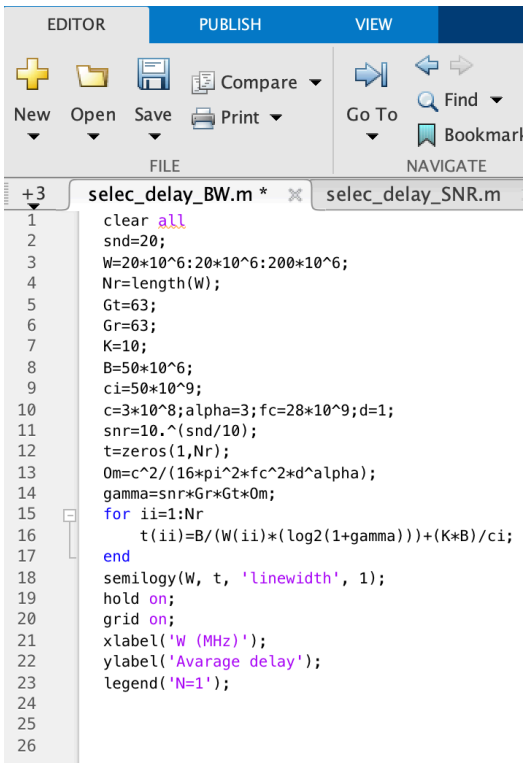
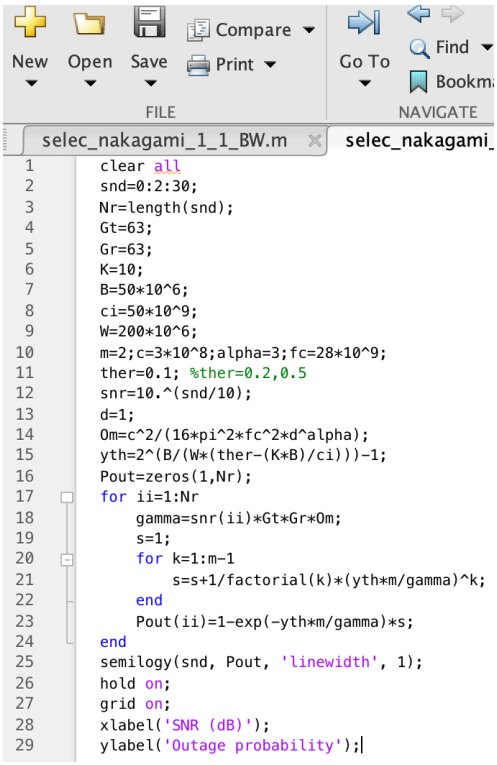
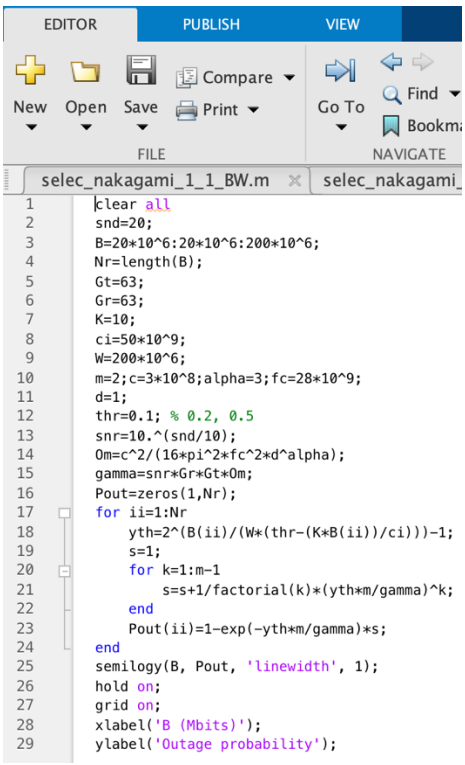


Figure A.6: Code of average delay versus channel bandwidth, Rayleigh fading channels.



```
clear all
snd=0:2:30;
Nr=length(snd);
Gt=63;
Gr=63;
K=10;
B=50*10^6;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
ther=0.1; %ther=0.2,0.5
snr=10.^(snd/10);
d=1;
Om=c^2/(16*pi^2*fc^2*d^alpha);
yth=2^(B/(W*(ther-(K*B)/ci)))-1;
Pout=zeros(1,Nr);
for ii=1:Nr
    gamma=snr(ii)*Gt*Gr*Om;
    s=1;
    for k=1:m-1
        s=s+1/factorial(k)*(yth*m/gamma)^k;
    end
    Pout(ii)=1-exp(-yth*m/gamma)*s;
end
semilogy(snd, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('SNR (dB)');
ylabel('Outage probability');
```

Figure A.7: Code of the delay-outage probability versus SNR, first model, Scheme I.

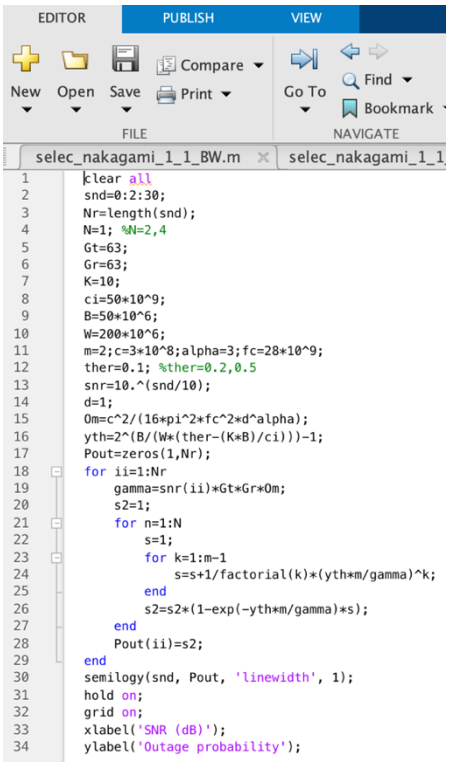


```
clear all
snd=20;
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
d=1;
thr=0.1; % 0.2, 0.5
snr=10.^(snd/10);
Om=c^2/(16*pi^2*fc^2*d^alpha);
gamma=snr*Gr*Gt*Om;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^(B(ii)/(W*(thr-(K*B(ii))/ci)))-1;
    s=1;
    for k=1:m-1
        s=s+1/factorial(k)*(yth*m/gamma)^k;
    end
    Pout(ii)=1-exp(-yth*m/gamma)*s;
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure A.8: Code of the delay-outage probability versus task size, first model, Scheme I.

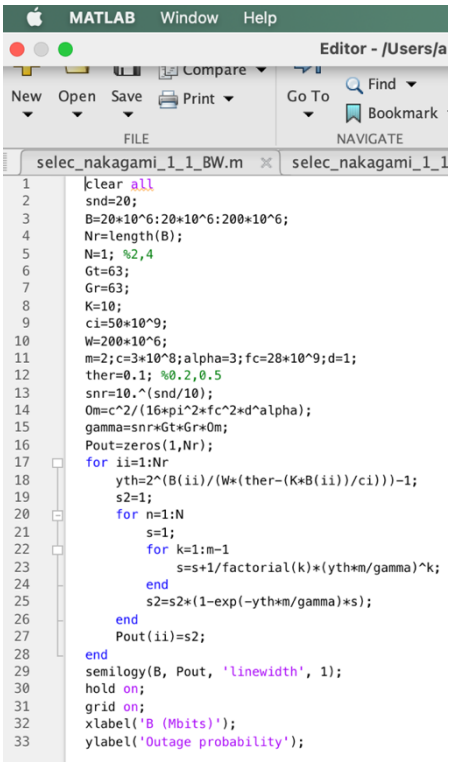
```
clear all
snd=20;
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d=1;
Om=c^2/(16*pi^2*fc^2*d^alpha);
gamma=snr*Gr*Gt*Om;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^(B/(W(ii)*(thr-(K*B)/ci)))-1;
    s=1;
    for k=1:m-1
        s=s+1/factorial(k)*(yth*m/gamma)^k;
    end
    Pout(ii)=1-exp(-yth*m/gamma)*s;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure A.9: Code of the delay-outage probability versus channel bandwidth, first model, Scheme I.



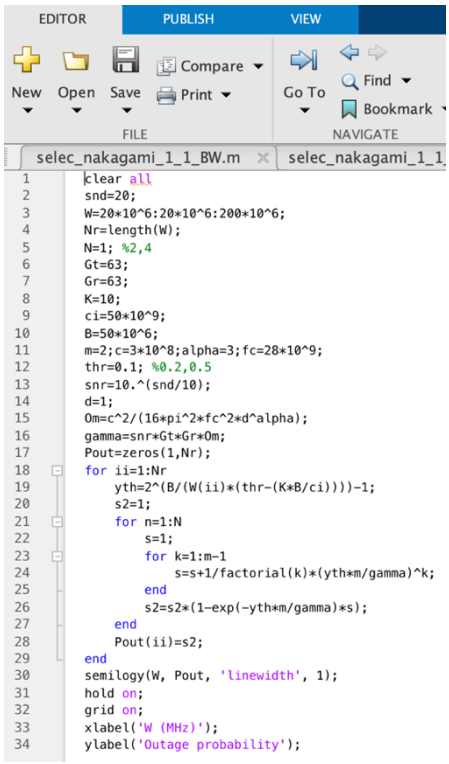
```
clear all
snd=0:2:30;
Nr=length(snd);
N=1; %N=2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
ther=0.1; %ther=0.2,0.5
snr=10.^(snd/10);
d=1;
Om=c^2/(16*pi^2*fc^2*d^alpha);
yth=2^(B/(W*(ther-(K*B)/ci)))-1;
Pout=zeros(1,Nr);
for ii=1:Nr
    gamma=snr(ii)*Gt*Gr*Om;
    s2=1;
    for n=1:N
        s=1;
        for k=1:m-1
            s=s+1/factorial(k)*(yth*m/gamma)^k;
        end
        s2=s2*(1-exp(-yth*m/gamma)*s);
    end
    Pout(ii)=s2;
end
semilogy(snd, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('SNR (dB)');
ylabel('Outage probability');
```

Figure A.10: Code of the delay-outage probability versus SNR, first model, Scheme II.



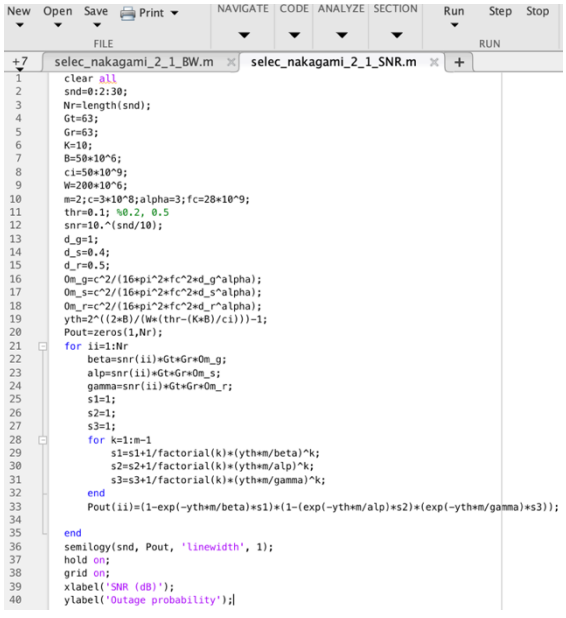
```
clear all
snd=20;
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=1;
ther=0.1; %0.2,0.5
snr=10.^(snd/10);
Om=c^2/(16*pi^2*fc^2*d^alpha);
gamma=snr*Gt*Gr*Om;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^(B(ii)/(W*(ther-(K*B(ii))/ci)))-1;
    s2=1;
    for n=1:N
        s=1;
        for k=1:m-1
            s=s+1/factorial(k)*(yth*m/gamma)^k;
        end
        s2=s2*(1-exp(-yth*m/gamma)*s);
    end
    Pout(ii)=s2;
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure A.11: Code of the delay-outage probability versus task size, first model, Scheme II.



```
clear all
snd=20;
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d=1;
Om=c^2/(16*pi^2*fc^2*d^alpha);
gamma=snr*Gt*Gr*Om;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^(B/(W(ii)*(thr-(K*B/ci))))-1;
    s2=1;
    for n=1:N
        s=1;
        for k=1:m-1
            s=s+1/factorial(k)*(yth*m/gamma)^k;
        end
        s2=s2*(1-exp(-yth*m/gamma)*s);
    end
    Pout(ii)=s2;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

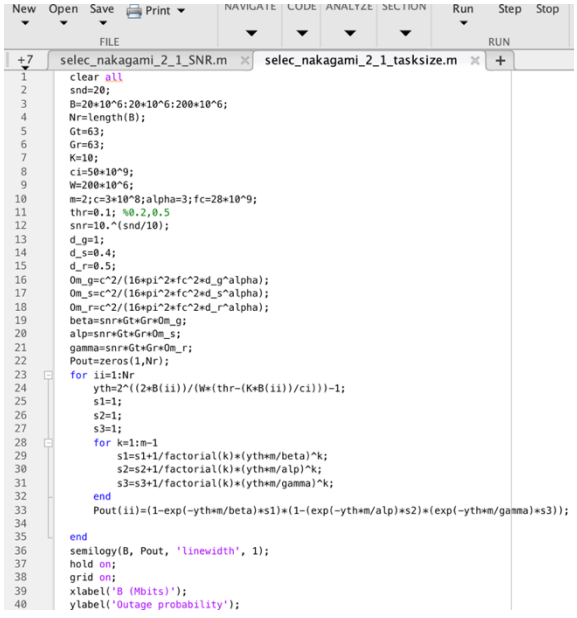
Figure A.12: Code of the delay-outage probability versus channel bandwidth, first model, Scheme II.



```
clear all
snd=0:2:30;
Nr=length(snd);
Gt=63;
Gr=63;
K=10;
B=50*10^6;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2, 0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
yth=2^((2*B)/(W*(thr-(K*B)/ci)))-1;
Pout=zeros(1,Nr);
for ii=1:Nr
    beta=snr(ii)*Gt*Gr*Om_g;
    alp=snr(ii)*Gt*Gr*Om_s;
    gamma=snr(ii)*Gt*Gr*Om_r;
    s1=1;
    s2=1;
    s3=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/beta)^k;
        s2=s2+1/factorial(k)*(yth*m/alp)^k;
        s3=s3+1/factorial(k)*(yth*m/gamma)^k;
    end
    Pout(ii)=(1-exp(-yth*m/beta)*s1)*(1-(exp(-yth*m/alp)*s2)*(exp(-yth*m/gamma)*s3));

end
semilogy(snd, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('SNR (dB)');
ylabel('Outage probability');
```

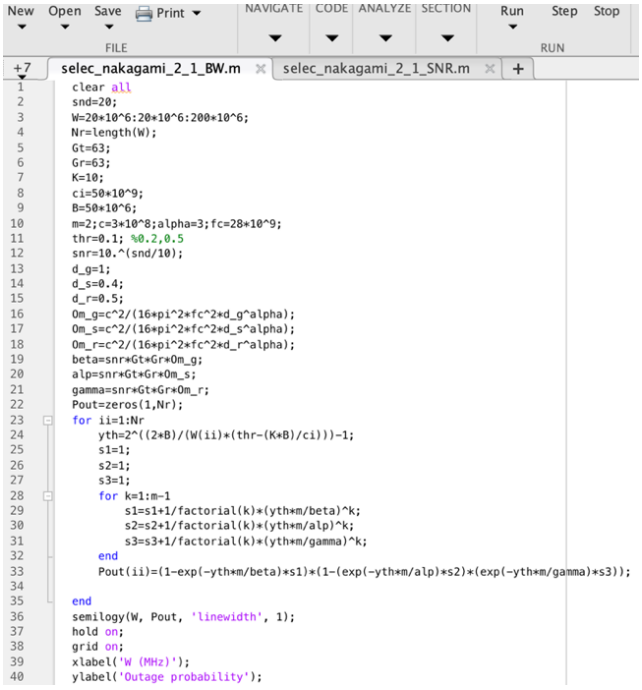
Figure A.13 Code of the delay-outage probability versus SNR, second model, Scheme I.



```
clear all
snd=20;
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
beta=snr*Gt*Gr*Om_g;
alp=snr*Gt*Gr*Om_s;
gamma=snr*Gt*Gr*Om_r;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^((2*B(ii))/(W*(thr-(K*B(ii))/ci)))-1;
    s1=1;
    s2=1;
    s3=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/beta)^k;
        s2=s2+1/factorial(k)*(yth*m/alp)^k;
        s3=s3+1/factorial(k)*(yth*m/gamma)^k;
    end
    Pout(ii)=(1-exp(-yth*m/beta)*s1)*(1-(exp(-yth*m/alp)*s2)*(exp(-yth*m/gamma)*s3));

end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure A.14: Code of the delay-outage probability versus task size, second model, Scheme I.



```
clear all
snd=20;
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
beta=snr*Gt*Gr*Om_g;
alp=snr*Gt*Gr*Om_s;
gamma=snr*Gt*Gr*Om_r;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^((2*B)/(W(ii)*(thr-(K*B)/ci)))-1;
    s1=1;
    s2=1;
    s3=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/beta)^k;
        s2=s2+1/factorial(k)*(yth*m/alp)^k;
        s3=s3+1/factorial(k)*(yth*m/gamma)^k;
    end
    Pout(ii)=(1-exp(-yth*m/beta)*s1)*(1-(exp(-yth*m/alp)*s2)*(exp(-yth*m/gamma)*s3));

end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure A.15: Code of the delay-outage probability versus channel bandwidth, second model, Scheme I.

```
clear all
snd=0:2:30;
Nr=length(snd);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
B=50*10^6;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; % 0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
yth=2^((2*B)/(W*(thr-(K*B)/ci)))-1;
Pout=zeros(1,Nr);
for ii=1:Nr
    beta=snr(ii)*Gt*Gr*Om_g;
    alp=snr(ii)*Gt*Gr*Om_s;
    gamma=snr(ii)*Gt*Gr*Om_r;
    s1=1;
    s2=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/beta)^k;
        s2=s2+1/factorial(k)*(yth*m/alp)^k;
    end
    s4=1;
    for n=1:N
        s3=1;
        for k=1:m-1
            s3=s3+1/factorial(k)*(yth*m/gamma)^k;
        end
        s4=s4*(1-exp(-yth*m/gamma)*s3);
    end
    Pout(ii)=(1-exp(-yth*m/beta)*s1)*(1-(exp(-yth*m/alp)*s2)*(1-s4));
end
semilogy(snd, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('SNR (dB)');
ylabel('Outage probability');
```

Figure A.16: Code of the delay-outage probability versus SNR, second model, Scheme II.

```
clear all
snd=20;
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
beta=snr*Gt*Gr*Om_g;
alp=snr*Gt*Gr*Om_s;
gamma=snr*Gt*Gr*Om_r;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^((2*B(ii))/(W*(thr-(K*B(ii))/ci)))-1;
    s1=1;
    s2=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/beta)^k;
        s2=s2+1/factorial(k)*(yth*m/alp)^k;
    end
    s4=1;
    for n=1:N
        s3=1;
        for k=1:m-1
            s3=s3+1/factorial(k)*(yth*m/gamma)^k;
        end
        s4=s4*(1-exp(-yth*m/gamma)*s3);
    end
    Pout(ii)=(1-exp(-yth*m/beta)*s1)*(1-(exp(-yth*m/alp)*s2)*(1-s4));
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure A.17: Code of the delay-outage probability versus task size, second model, Scheme II.

```
clear all
snd=20;
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
beta=snr*Gt*Gr*Om_g;
alp=snr*Gt*Gr*Om_s;
gamma=snr*Gt*Gr*Om_r;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^((2*B)/(W(ii)*(thr-(K*B)/ci)))-1;
    s1=1;
    s2=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/beta)^k;
        s2=s2+1/factorial(k)*(yth*m/alp)^k;
    end
    s4=1;
    for n=1:N
        s3=1;
        for k=1:m-1
            s3=s3+1/factorial(k)*(yth*m/gamma)^k;
        end
        s4=s4*(1-exp(-yth*m/gamma)*s3);
    end
    Pout(ii)=(1-exp(-yth*m/beta)*s1)*(1-(exp(-yth*m/alp)*s2)*(1-s4));
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

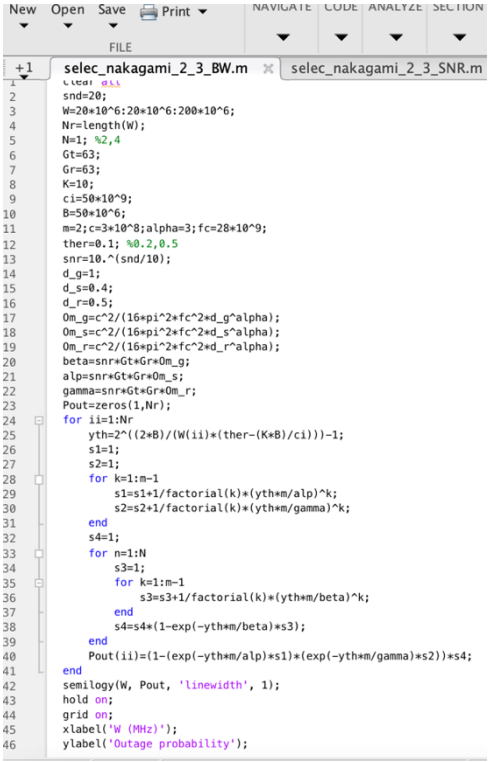
Figure A.18: Code of the delay-outage probability versus channel bandwidth, second model, Scheme II.

```
clear all
snd=0:2:30;
Nr=length(snd);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
B=50*10^6;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
yth=2^((2*B)/(W*(thr-(K*B)/ci)))-1;
Pout=zeros(1,Nr);
for ii=1:Nr
    beta=snr(ii)*Gt*Gr*Om_g;
    alp=snr(ii)*Gt*Gr*Om_s;
    gamma=snr(ii)*Gt*Gr*Om_r;
    s1=1;
    s2=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/alp)^k;
        s2=s2+1/factorial(k)*(yth*m/gamma)^k;
    end
    s4=1;
    for n=1:N
        s3=1;
        for k=1:m-1
            s3=s3+1/factorial(k)*(yth*m/beta)^k;
        end
        s4=s4*(1-exp(-yth*m/beta)*s3);
    end
    Pout(ii)=(1-(exp(-yth*m/alp)*s1)*(exp(-yth*m/gamma)*s2))*s4;
end
semilogy(snd, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('SNR (dB)');
ylabel('Outage probability');
```

Figure A.19: Code of the delay-outage probability versus SNR, second model, Scheme III.

```
clear all
snd=20;
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
thr=0.1; %0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
beta=snr*Gt*Gr*Om_g;
alp=snr*Gt*Gr*Om_s;
gamma=snr*Gt*Gr*Om_r;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^((2*B(ii))/(W*(thr-(K*B(ii))/ci)))-1;
    s1=1;
    s2=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/alp)^k;
        s2=s2+1/factorial(k)*(yth*m/gamma)^k;
    end
    s4=1;
    for n=1:N
        s3=1;
        for k=1:m-1
            s3=s3+1/factorial(k)*(yth*m/beta)^k;
        end
        s4=s4*(1-exp(-yth*m/beta)*s3);
    end
    Pout(ii)=(1-(exp(-yth*m/alp)*s1)*(exp(-yth*m/gamma)*s2))*s4;
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure A.20: Code of the delay-outage probability versus task size, second model, Scheme III.



```
clear all
snd=20;
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
N=1; %2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;
ther=0.1; %0.2,0.5
snr=10.^(snd/10);
d_g=1;
d_s=0.4;
d_r=0.5;
Om_g=c^2/(16*pi^2*fc^2*d_g^alpha);
Om_s=c^2/(16*pi^2*fc^2*d_s^alpha);
Om_r=c^2/(16*pi^2*fc^2*d_r^alpha);
beta=snr*Gt*Gr*Om_g;
alp=snr*Gt*Gr*Om_s;
gamma=snr*Gt*Gr*Om_r;
Pout=zeros(1,Nr);
for ii=1:Nr
    yth=2^((2*B)/(W(ii)*(ther-(K*B)/ci)))-1;
    s1=1;
    s2=1;
    for k=1:m-1
        s1=s1+1/factorial(k)*(yth*m/alp)^k;
        s2=s2+1/factorial(k)*(yth*m/gamma)^k;
    end
    s4=1;
    for n=1:N
        s3=1;
        for k=1:m-1
            s3=s3+1/factorial(k)*(yth*m/beta)^k;
        end
        s4=s4*(1-exp(-yth*m/beta)*s3);
    end
    Pout(ii)=(1-(exp(-yth*m/alp)*s1)*(exp(-yth*m/gamma)*s2))*s4;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure A.21: Code of the delay-outage probability versus channel bandwidth, second model, Scheme III.

Appendix B

This Appendix contains supplementary MATLAB codes for numerical evaluation of the system performance. There are codes for combining schemes with Rayleigh and Nakagami-m fading channels versus different values of transmit power, task size, channel bandwidth, latency threshold, and the number of edge nodes.

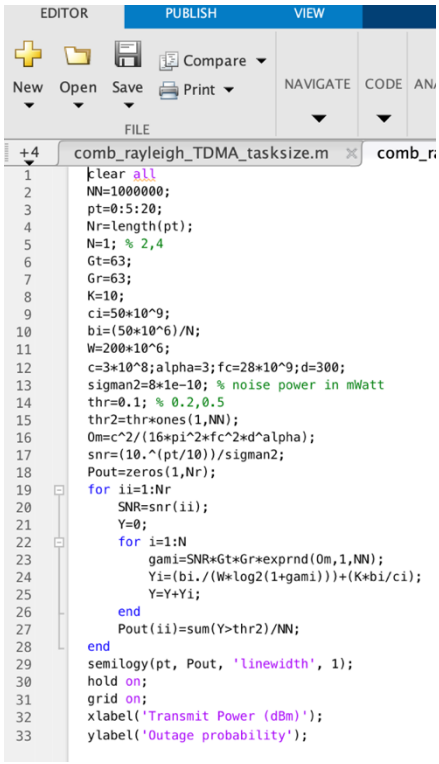


Figure B.1: Code of the delay-outage probability versus transmit power, TDMA model, Rayleigh fading channels.

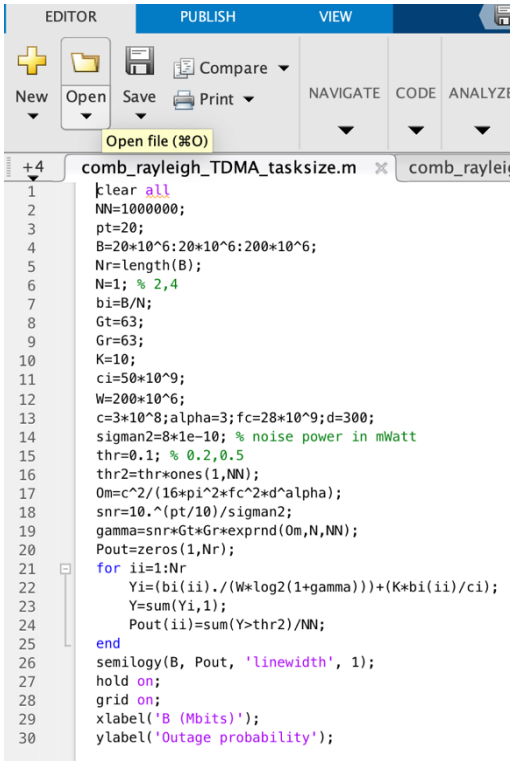
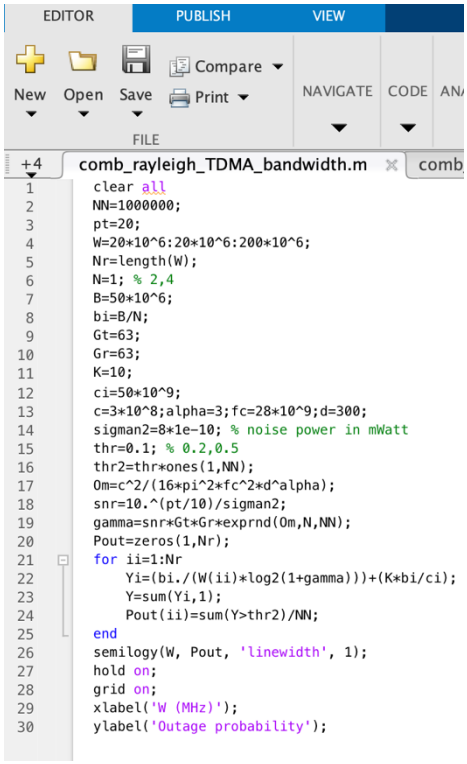
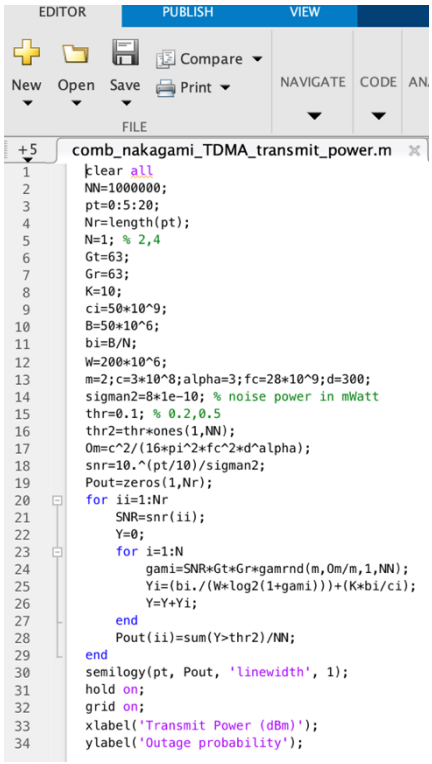


Figure B.2: Code of the delay-outage probability versus task size, TDMA model, Rayleigh fading channels.



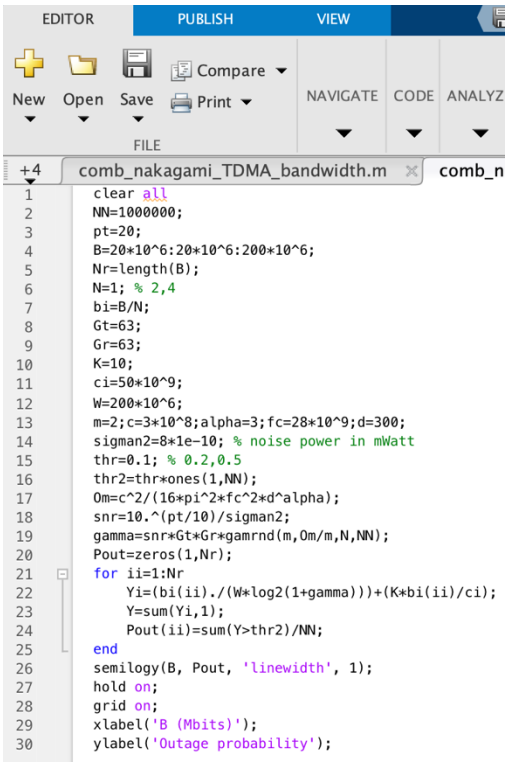
```
clear all
NN=1000000;
pt=20;
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
N=1; % 2,4
B=50*10^6;
bi=B/N;
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
c=3*10^8;alpha=3;fc=28*10^9;d=300;
sigman2=8*1e-10; % noise power in mWatt
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
snr=10.^(pt/10)/sigman2;
gamma=snr*Gt*Gr*exprnd(Om,N,NN);
Pout=zeros(1,Nr);
for ii=1:Nr
    Yi=(bi./(W(ii)*log2(1+gamma)))+(K*bi/ci);
    Y=sum(Yi,1);
    Pout(ii)=sum(Y>thr2)/NN;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure B.3: Code of the delay-outage probability versus channel bandwidth, TDMA model, Rayleigh fading channels.



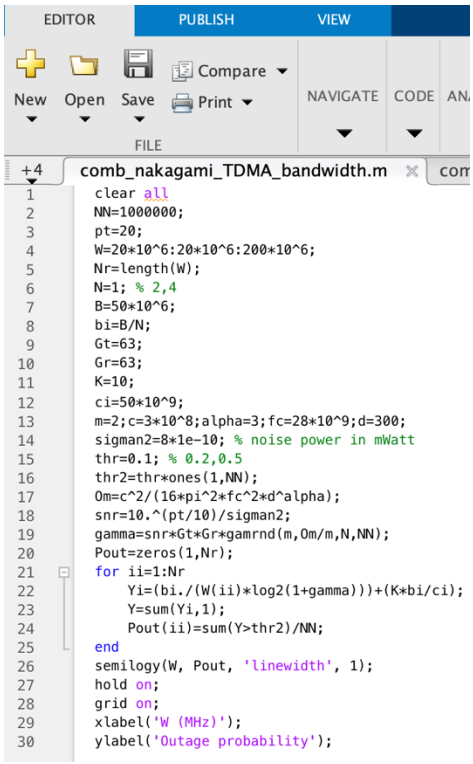
```
clear all
NN=1000000;
pt=0:5:20;
Nr=length(pt);
N=1; % 2,4
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
bi=B/N;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=300;
sigman2=8*1e-10; % noise power in mWatt
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
snr=10.^(pt/10)/sigman2;
Pout=zeros(1,Nr);
for ii=1:Nr
    SNR=snr(ii);
    Y=0;
    for i=1:N
        gami=SNR*Gt*Gr*gamrnd(m,Om/m,1,NN);
        Yi=(bi./(W*log2(1+gami)))+(K*bi/ci);
        Y=Y+Yi;
    end
    Pout(ii)=sum(Y>thr2)/NN;
end
semilogy(pt, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('Transmit Power (dBm)');
ylabel('Outage probability');
```

Figure B.4: Code of the delay-outage probability versus transmit power, TDMA model, Nakagami-m fading channels.



```
clear all
NN=1000000;
pt=20;
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
N=1; % 2,4
bi=B/N;
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=300;
sigman2=8*1e-10; % noise power in mWatt
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
snr=10.^(pt/10)/sigman2;
gamma=snr*Gt*Gr*gamrnd(m,Om/m,N,NN);
Pout=zeros(1,Nr);
for ii=1:Nr
    Yi=(bi(ii)./(W*log2(1+gamma)))+(K*bi(ii)/ci);
    Y=sum(Yi,1);
    Pout(ii)=sum(Y>thr2)/NN;
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure B.5: Code of the delay-outage probability versus task size, TDMA model, Nakagami-m fading channels.



```
clear all
NN=1000000;
pt=20;
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
N=1; % 2,4
B=50*10^6;
bi=B/N;
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=300;
sigman2=8*1e-10; % noise power in mWatt
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
snr=10.^(pt/10)/sigman2;
gamma=snr*Gt*Gr*gamrnd(m,Om/m,N,NN);
Pout=zeros(1,Nr);
for ii=1:Nr
    Yi=(bi./(W(ii)*log2(1+gamma)))+(K*bi/ci);
    Y=sum(Yi,1);
    Pout(ii)=sum(Y>thr2)/NN;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure B.6: Code of the delay-outage probability versus channel bandwidth, TDMA model, Nakagami-m fading channels.

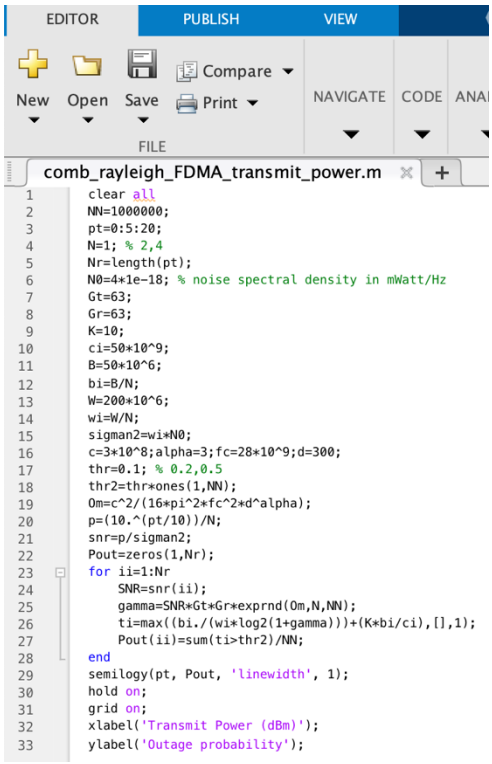


Figure B.7: Code of the delay-outage probability versus transmit power, FDMA model, Rayleigh fading channels.

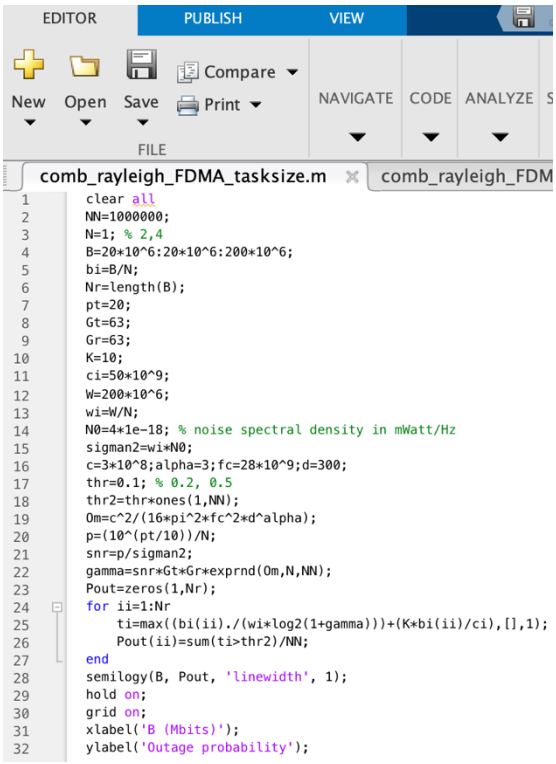


Figure B.8: Code of the delay-outage probability versus task size, FDMA model, Rayleigh fading channels.

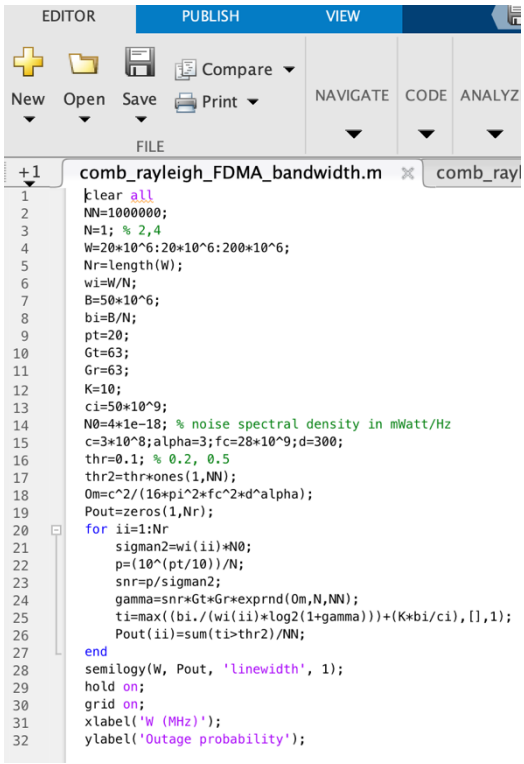


Figure B.9: Code of the delay-outage probability versus channel bandwidth, FDMA model, Rayleigh fading channels.

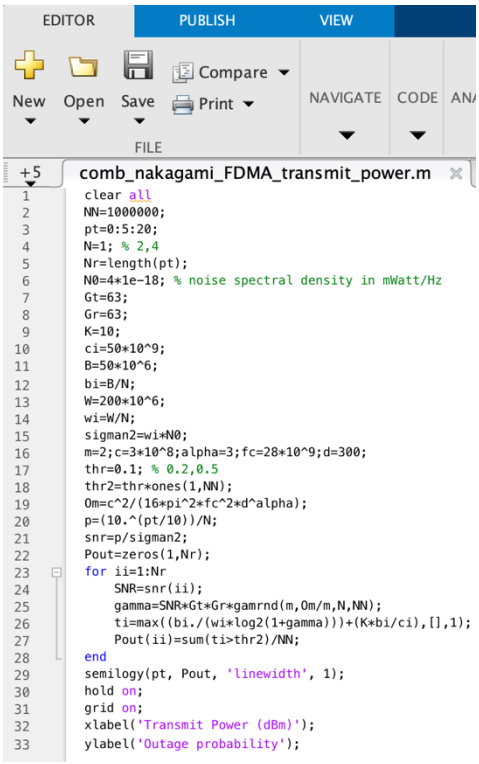
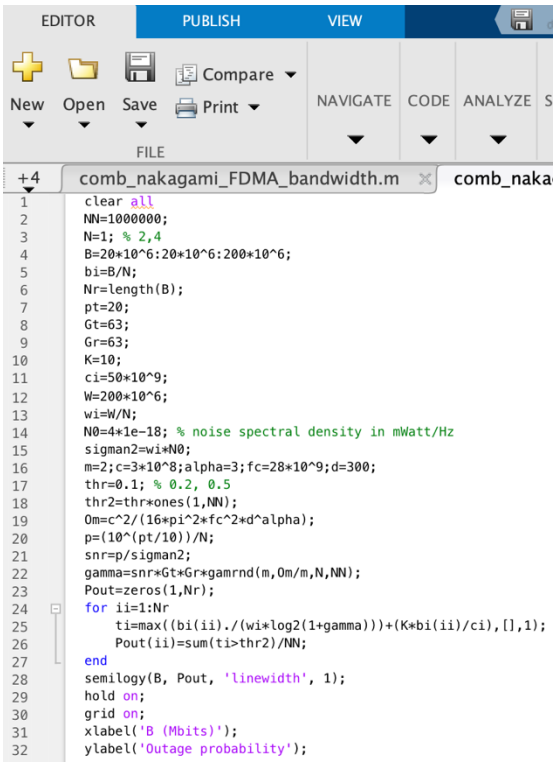
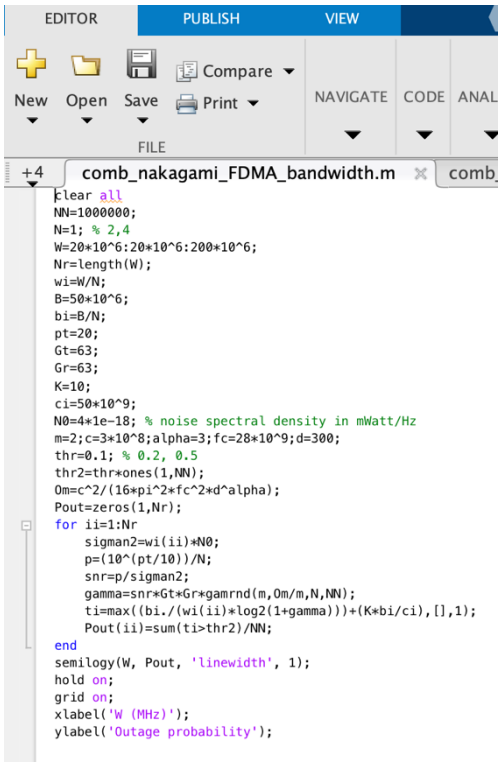


Figure B.10: Code of the delay-outage probability versus transmit power, FDMA model, Nakagami-m fading channels.



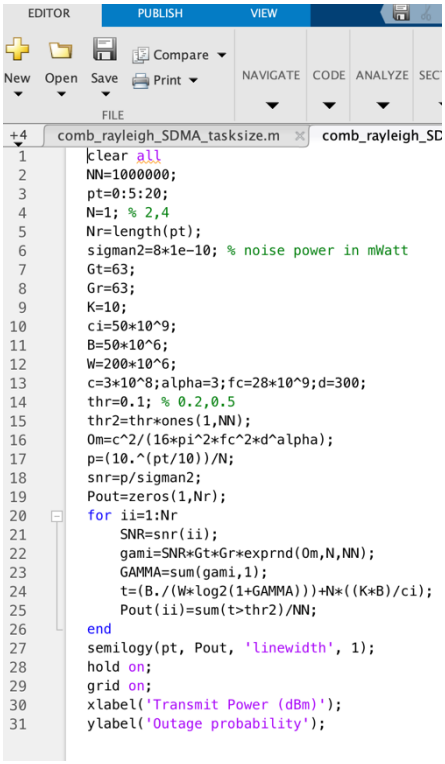
```
clear all
NN=1000000;
N=1; % 2,4
B=20*10^6:20*10^6:200*10^6;
bi=B/N;
Nr=length(B);
pt=20;
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
wi=W/N;
N0=4*1e-18; % noise spectral density in mWatt/Hz
sigman2=wi*N0;
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2, 0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
p=(10^(pt/10))/N;
snr=p/sigman2;
gamma=snr*Gt*Gr*gamrnd(m,Om/m,N,NN);
Pout=zeros(1,Nr);
for ii=1:Nr
    ti=max((bi(ii)./(wi*log2(1+gamma)))+(K*bi(ii)/ci),[],1);
    Pout(ii)=sum(ti>thr2)/NN;
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure B.11: Code of the delay-outage probability versus task size, FDMA model, Nakagami-m fading channels.



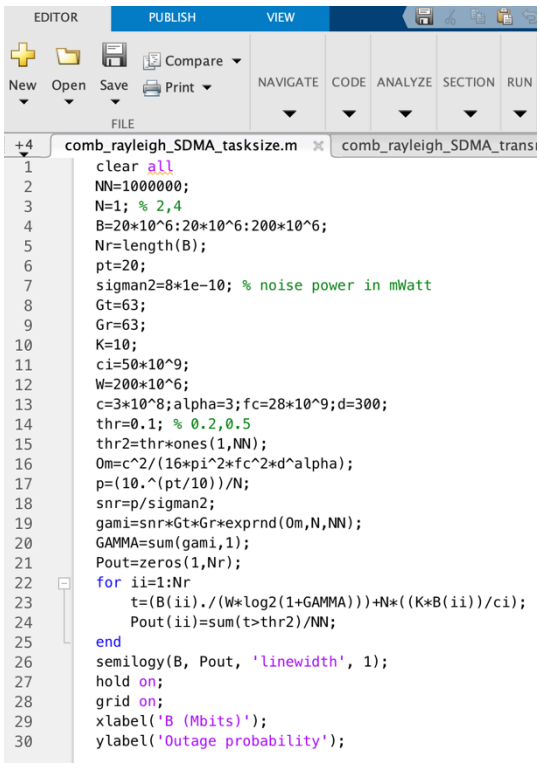
```
clear all
NN=1000000;
N=1; % 2,4
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
wi=W/N;
B=50*10^6;
bi=B/N;
pt=20;
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
N0=4*1e-18; % noise spectral density in mWatt/Hz
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2, 0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
Pout=zeros(1,Nr);
for ii=1:Nr
    sigman2=wi(ii)*N0;
    p=(10^(pt/10))/N;
    snr=p/sigman2;
    gamma=snr*Gt*Gr*gamrnd(m,Om/m,N,NN);
    ti=max((bi./(wi(ii)*log2(1+gamma)))+(K*bi/ci),[],1);
    Pout(ii)=sum(ti>thr2)/NN;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure B.12: Code of the delay-outage probability versus channel bandwidth, FDMA model, Nakagami-m fading channels.



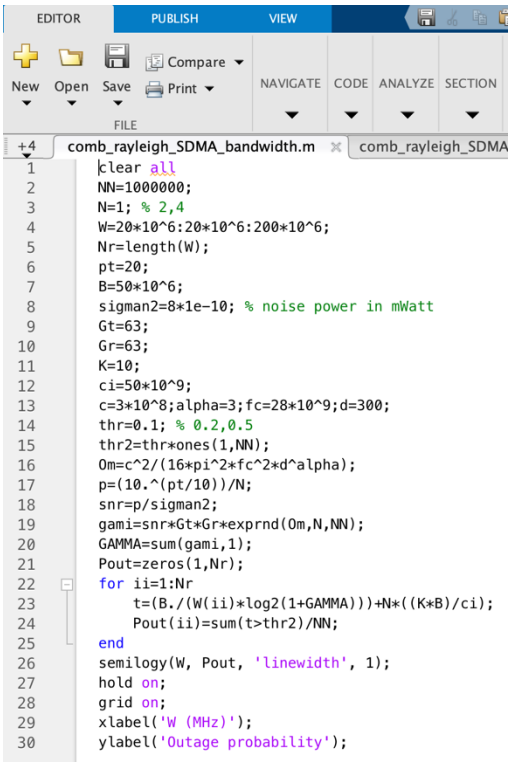
```
clear all
NN=1000000;
pt=0:5:20;
N=1; % 2,4
Nr=length(pt);
sigman2=8*1e-10; % noise power in mWatt
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
W=200*10^6;
c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
p=(10.^(pt/10))/N;
snr=p/sigman2;
Pout=zeros(1,Nr);
for ii=1:Nr
    SNR=snr(ii);
    gami=SNR*Gt*Gr*exprnd(Om,N,NN);
    GAMMA=sum(gami,1);
    t=(B./(W*log2(1+GAMMA)))+N*((K*B)/ci);
    Pout(ii)=sum(t>thr2)/NN;
end
semilogy(pt, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('Transmit Power (dBm)');
ylabel('Outage probability');
```

Figure B.13: Code of the delay-outage probability versus transmit power, SDMA model, Rayleigh fading channels.



```
clear all
NN=1000000;
N=1; % 2,4
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
pt=20;
sigman2=8*1e-10; % noise power in mWatt
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
p=(10.^(pt/10))/N;
snr=p/sigman2;
gami=snr*Gt*Gr*exprnd(Om,N,NN);
GAMMA=sum(gami,1);
Pout=zeros(1,Nr);
for ii=1:Nr
    t=(B(ii)./(W*log2(1+GAMMA)))+N*((K*B(ii))/ci);
    Pout(ii)=sum(t>thr2)/NN;
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure B14: Code of the delay-outage probability versus task size, SDMA model, Rayleigh fading channels.

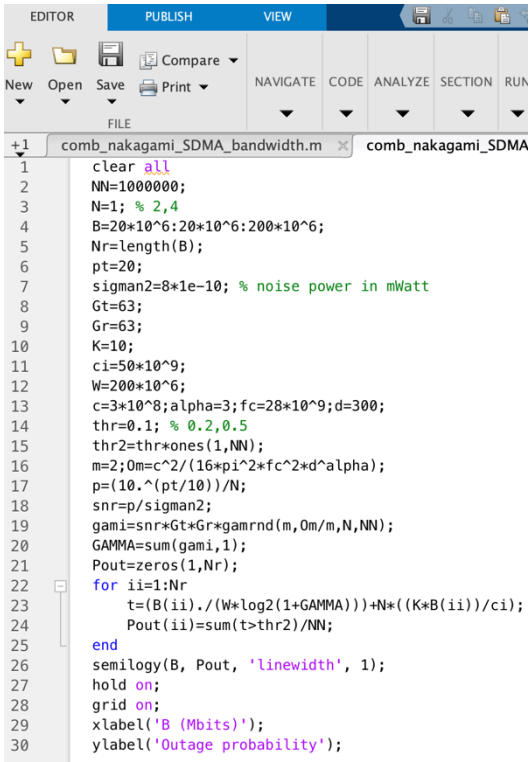


```
clear all
NN=1000000;
N=1; % 2,4
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
pt=20;
B=50*10^6;
sigman2=8*1e-10; % noise power in mWatt
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
p=(10.^(pt/10))/N;
snr=p/sigman2;
gami=snr*Gt*Gr*exprnd(Om,N,NN);
GAMMA=sum(gami,1);
Pout=zeros(1,Nr);
for ii=1:Nr
    t=(B./(W(ii)*log2(1+GAMMA)))+N*((K*B)/ci);
    Pout(ii)=sum(t>thr2)/NN;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure B.15: Code of the delay-outage probability versus channel bandwidth, SDMA model, Rayleigh fading channels.

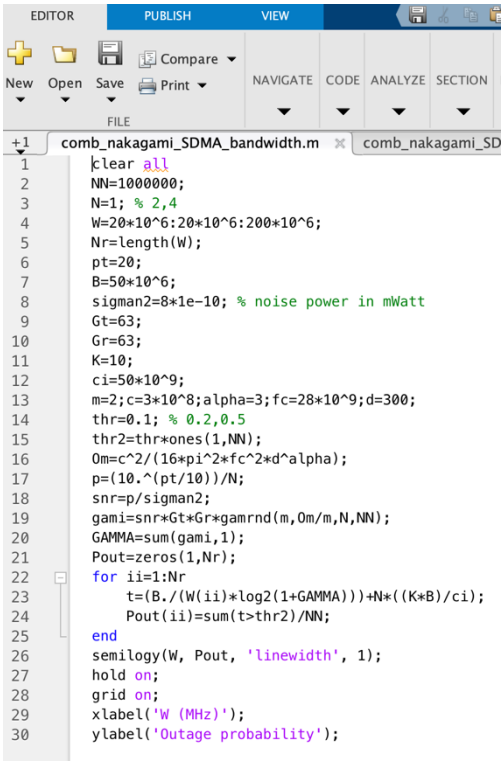
```
clear all
NN=1000000;
pt=0:5:20;
N=1; % 2,4
Nr=length(pt);
sigman2=8*1e-10; % noise power in mWatt
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
B=50*10^6;
W=200*10^6;
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
p=(10.^(pt/10))/N;
snr=p/sigman2;
Pout=zeros(1,Nr);
for ii=1:Nr
    SNR=snr(ii);
    gami=SNR*Gt*Gr*gamrnd(m,Om/m,N,NN);
    GAMMA=sum(gami,1);
    t=(B./(W*log2(1+GAMMA)))+N*((K*B)/ci);
    Pout(ii)=sum(t>thr2)/NN;
end
semilogy(pt, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('Transmit Power (dBm)');
ylabel('Outage probability');
```

Figure B.16: Code of the delay-outage probability versus transmit power, SDMA model, Nakagami-m fading channels.



```
clear all
NN=1000000;
N=1; % 2,4
B=20*10^6:20*10^6:200*10^6;
Nr=length(B);
pt=20;
sigman2=8*1e-10; % noise power in mWatt
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
W=200*10^6;
c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
m=2;Om=c^2/(16*pi^2*fc^2*d^alpha);
p=(10.^(pt/10))/N;
snr=p/sigman2;
gami=snr*Gt*Gr*gamrnd(m,Om/m,N,NN);
GAMMA=sum(gami,1);
Pout=zeros(1,Nr);
for ii=1:Nr
    t=(B(ii)./(W*log2(1+GAMMA)))+N*((K*B(ii))/ci);
    Pout(ii)=sum(t>thr2)/NN;
end
semilogy(B, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('B (Mbits)');
ylabel('Outage probability');
```

Figure B.17: Code of the delay-outage probability versus task size, SDMA model, Nakagami-m fading channels.



```
clear all
NN=1000000;
N=1; % 2,4
W=20*10^6:20*10^6:200*10^6;
Nr=length(W);
pt=20;
B=50*10^6;
sigman2=8*1e-10; % noise power in mWatt
Gt=63;
Gr=63;
K=10;
ci=50*10^9;
m=2;c=3*10^8;alpha=3;fc=28*10^9;d=300;
thr=0.1; % 0.2,0.5
thr2=thr*ones(1,NN);
Om=c^2/(16*pi^2*fc^2*d^alpha);
p=(10.^(pt/10))/N;
snr=p/sigman2;
gami=snr*Gt*Gr*gamrnd(m,Om/m,N,NN);
GAMMA=sum(gami,1);
Pout=zeros(1,Nr);
for ii=1:Nr
    t=(B./(W(ii)*log2(1+GAMMA)))+N*((K*B)/ci);
    Pout(ii)=sum(t>thr2)/NN;
end
semilogy(W, Pout, 'linewidth', 1);
hold on;
grid on;
xlabel('W (MHz)');
ylabel('Outage probability');
```

Figure B.18: Code of the delay-outage probability versus channel bandwidth, SDMA model, Nakagami-m fading channels.