SPATIOTEMPORAL ANALYSIS ON DISTRIBUTED TASK OFFLOADING IN EXTREME EDGE DEVICES

By

MAHMOUD N. ABDELHADI

A thesis submitted to the Graduate Program in the School of Computing in conformity with the requirements for the Degree of Master of Science

Queen's University

Kingston, Ontario, Canada

May, 2022

Dedication

FOR THE SAKE OF ALLAH

to my parents, and beloved family

to Dr. Sameh Sorour, may his soul rest in peace

Abstract

With the rapid development of the Internet of Things (IoT), the number of smart devices connected to the Internet is exponentially increasing, resulting in large-scale data and inadequate resources, which has caused high congestion and slow response delay in legacy cloud computing models. Multi-access Edge Computing (MEC) is a computing paradigm that can facilitate both delay-sensitive, and data-intensive tasks associated with IoT applications. MEC provides low latency by pushing the resources closer to the applications. However, due to the increase in the number of devices that use MEC, the high congestion problem remains unsolved. A promising solution is to take advantage of the abundant and underutilized computing resources of the Extreme Edge Devices (EEDs). EEDs bring the computing service closer to the end-users, which could significantly reduce the delay caused by cloud execution. However, the success of such an extreme edge parallel computing paradigm is impacted by i) wireless device-to-device (D2D)communication performance, a requirement for the communication between the recruited EEDs and the task requester to perform the offloading process, ii) the computing capabilities of the EEDs, which governs the execution time of each offloaded task and iii) the reliability of the recruited EEDs. In this context, a novel spatiotemporal framework employing stochastic geometry and absorbing continuous time Markov chains (ACTMC) is developed to analyze the communication and computation performance of extreme edge computing systems. Using this framework, we study the influence of various system parameters on the average task response delay over a baseline system, where the devices are recruited randomly and do not fail during the execution. Extensive evaluations have shown that the EED-enabled system outperforms MEC in terms of the average response delay in some cases. Next, we developed an advanced model, where the possibility of failure of the recruited EEDs is considered, and the impact of the recruitment criteria of EEDs on the recruitment time is investigated. Our findings have revealed the optimal number of slices that the task should be divided into to minimize the total compu-

tation time, which will minimize the average response delay, and how that optimal number is affected by the various system parameters.

Acknowledgment

"Never forget what you are, the rest of the world will not. Wear it like armor, it can never be used to hurt you."

First, I would like to thank my Lord gratefully, I would never accomplish this work without his endless blessings and unconditional gaudiness.

I also would like to thank Dr. Sameh Sorour, may his soul rest in peace, for his inspiring influence on me, which made me complete this academic journey. You were like a father to me, you were a great supervisor and an amazing teacher, you have been there for us anytime, even when you could barely talk. You taught me new things and walked with me in this foggy path, I came as new Masters student and you guided me all the way. I have never thought that I will write this or even live this moment, thank you for everything, I will always be greatfull and thankful to you.

I would like to express my deepest gratitude to Dr, Hesham ElSawy and Dr. Hossam Hassanein for their unrelenting support, supervision, mentorship, and guidance in teaching me the intricacies of research. I also would like to thank Dr. Sara Elsayed, for helping me in writing my first paper, and for her help in reviewing this thesis.

I thank my family for their support and encouragement, as without them I would not have been able to have the perseverance to remain on this path nor the patience to learn. My family has taught me the notion of self-reliance and self-learning throughout my childhood, and it is that notion that had led me - by the support of Allah - to where I currently am.

Also, I would like to thank my dear friends, Mohammad Abu Zaid and Sofian Sammar, who have been holding my back, virtually, throughout my masters journey. For all of the endless IMessage calls and stories we shared. Hope we stay at each others sides forever.

I also thank my friend Shareef Azmi, for all of the advices he gave me, for the long talks and discussions. Thanks to the discord and basketball guys, for all the talks and the things we did together, which made my staying at Kingston bearable.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

IP	Internet Protocol
5G	Fifth Generation Mobile Networks
PPP	Poisson Point Processes
EC	Edge Computing
MEC	Multi-access Edge Computing
NOMA	Non-Orthogonal multiple access
OMA	Orthogonal multiple access
PM	Physical Machine
QoS	Quality of Service
mmWave	Millimeter wave
BS	Base Station
VM	Virtual Machine
AP	Access Point
CS	Computation Server
EED	Extreme Edge Device
IoT	Internet of Things
PC	Personal Computer
CTMC	Continuous Time Markov Chain

DTMC	Discrete Time Markov Chain
LCC	Local Computation Capability
ACTMC	Absorbing Continuous Time Markov Chain
EDTMC	Embedded Discrete Time Markov Chain
LoS	Line of Sight
NLoS	Non-Line of Sight
SINR	Signal to Interference plus Noise Ratio
D2D	Device to Device
STM	State transition matrix

List of Symbols

n	Number of task slices
$\mathbb{R}^2$	The plane of the system
Φ	Workers PPP
ν_w	Workers intensity
Ω	Requesters PPP
Ω_L	LoS Requesters PPP
Ω_N	NLoS Requesters PPP
ν_r	Requesters intensity
μ_f	Task finishing rate
$1/n\mu_f$	Task exponential distribution
R_L	LoS radius
P_l	Transmission power of worker l
α_L	LoS path loss exponent
α_N	NLoS path loss exponent
N_L	Nagakami fading LoS parameter
N_N	Nagakami fading NLoS parameter
M_x	Antenna main lobe gain
m_X	Antenna side lobe gain

θ_x	Signal beamwidth
ξ	SINR threshold
$\|h_0\|^2$	Channel power gain
C_L	Intercept of the LoS channel
C_N	Intercept of the NLoS channel
r_0	The distance between the requester and the worker
I_L	Interference from other active LoS requesters
I_N	Interference from other NLoS requesters
σ^2	Ambient noise power
τ_c	D2D average communication time
τ_h	Average EED hiring time
λ_h	Hiring rate
Q	State transition matrix
K_m	Hiring tracking matrix
$H_{m,m+1}$	Task finishing tracking matrix
$\mathcal{K}_m$	Hiring transition probability
$\mathcal{H}_{m,m+1}$	Finishing transition probability from 0 to $n-m$ devices
$t_{\mathbf{z}_i,\mathbf{z}_j}$	Average time to go from state z_i to z_j
T	Time until absorption vector
$\{X_l\}$	Set of outdoor workers

l	System reliability parameter
γ	Failure rate
m_L	Maximum number of allocated devices
$\mathcal{S}_A$	State space except the absorbing status
$\mathbf{P}_T$	Probability transition matrix without absorbing states rows and columns
n^*	Optimal number of task slices
$\mathbb{E}(\cdot)$	Expectation operator
$\mathbb{P}(\cdot)$	Probability operator
$\mathbb{I}(\cdot)$	Indicator function

Chapter 1
Introduction

1.1 Overview and Motivation

The evolution of mobile networks has been affected by the growing demand [1], content sharing, and user behavior, which are redefining the way networks are utilized [2]. The rollout of the Fifth-generation Mobile Network (5G) is expected to cause an increase in Device to Device (D2D) communications, along with the increase in the Internet of Things (IoT) traffic and services [3]. With the advent of the IoT, it is anticipated that 38.6 billion IoT devices will be connected to the Internet by 2025 [4]. In addition, it is expected that the IoT market size will rise up to $15 trillion by the same year [4]. This can trigger a broad spectrum of latency-sensitive IoT applications with strenuous Quality of Service (QoS) requirements [5]. Such requirements cannot be adequately satisfied by cloud computing, due to the distant geographical location of cloud data centers, as well as the huge traffic influx imposed at backhaul links [5].

The widespread use of IoT devices such as smartphones, tablets, and wearables, along with improved wireless network capabilities, has prompted an extensive study on wireless communication to address the issues that have arisen [6]. However, despite ongoing advancements in hardware components, most of the available IoT devices cannot completely satisfy the needs of future computation-intensive and delay-sensitive applications [7]. Therefore, achieving the network objectives of low latency, reliable communication, and efficient computing relies heavily on effective network design, analysis, and optimization, where a combined communication and computation aspect must be taken into account. Multi-access Edge Computing (MEC) has emerged as a propitious computing paradigm that can bring the computing service within close proximity to end devices, thus significantly reducing the delay and successfully satisfying the soaring demands of IoT applications [8].

In MEC, efficient task offloading decisions are pivotal to achieve promising performance

gains. Most existing MEC platforms depend on the availability of computationally capable Base Stations (BSs) to perform the offloaded computational tasks [8]. Recently, various research efforts [9–11] have explored the potential of leveraging the drastic surge in IoT devices, also referred to as Extreme Edge Devices (EEDs) [12], and exploiting their collective processing capabilities to further improve the performance.

Harvesting abundant yet underutilized computational resources at EEDs can break the monopoly caused by the fact that most EC paradigms, including MEC, are controlled solely by cloud service providers and/or network operators. Breaking this monopoly can democratize the edge and enable more players to construct and control their own edge cloud. In addition, in EED-enabled computing environments, EEDs are recruited to amplify the compute resource pool, perform parallel computing, and enhance the task offloading service, which can enable further improvement of the delay. However, achieving network objectives, such as low latency, reliable communication, and efficient computing, relies heavily on effective network design, analysis, and optimization, where a combined communication and computation perspective must be taken into consideration.

1.2 Challenges

The presence of demanding calculations within the network resulted from the astounding number of active applications and services. One solution is to allow such computations to be performed in a distant data center (the cloud, for example). However, due to the imposed network delay for offloading data to the cloud and computation dependencies between data generated by nearby sensors, such an approach is inefficient due to bandwidth constraints. Moreover, it hinders the performance of time-sensitive and location-aware applications [13]. A reasonable solution would be to bring the compute resources closer to the devices, this will reduce the network latency and maximize the use of extra resources capabilities, but doing that might be expensive and not applicable in some scenarios. Another solution is to utilize the massive number of nearby devices to come up with EED-enabled computing

environments. Despite its advantageous potential, task offloading in EED-enabled computing environments is associated with a number of challenging issues. Such issues can be summarized as follows:

- Spatial randomness: which emerges from the highly dynamic network topology, leading to a shortage in the number of available EEDs in specific scenarios. Note that spatial randomness can vary based on the number of EEDs, as well as their arrival rate.
- Reliability of EEDs: which is affected by the fact that EEDs are prone to failure, due to some inside or outside effects. Thus, EED-enabled computing systems need to account for such reliability issues.
- Temporal randomness: which can be witnessed mainly in the device recruitment time. Such time depends on many factors, including distance and signal power. Temporal randomness can also be witnessed in the task execution time, which differs according to task size.

1.3 Contributions and Objectives

The main objective in this thesis is to come up with spatiotemporal analysis model, to calculate the average task response delay in EED-enabled computing environments. This can be divided into the following objectives: a) model the network spatially in a way that can capture the randomness in the nature of the network, b) model the temporal dynamic in the network based on captured temporal parameters, and c) propose a novel mathematical model to calculate the average task response delay in case of offloading the task slices to surrounding EEDs.

The contributions of this thesis are as follows:

- A novel spatiotemporal analysis framework that investigates the average task response delay at EEDs. We consider a computational task that can be divided into smaller

slices, referred to as jobs, to be offloaded at EEDs for faster execution and less computational delay at each device. Consequently, the response delay includes a) the D2D communications delay to recruit and offload jobs to EEDs, and b) the computation delay to execute the offloaded jobs at the recruited EEDs. To this end, an absorbing continuous time Markov chain (ACTMC) is constructed to model the temporal part. The ACTMC is used to track the sequential recruitment of EEDs via D2D communications, as well as the parallel task execution at the recruited EEDs. To capture the interwoven communication and computation delays, the recruitment rate of the ACTMC is computed via tools from stochastic geometry, which we use to spatially model our network, to account for D2D communications success probability. The results reveal the existence of an optimal number (n^*) of EEDs that minimizes the response delay. Going beyond (n^*) leads to an overwhelming communication latency that dominates the reduced computation latency of each job. Current literature models do not count on both spatial and temporal models to calculate the average task response delay, which supports the novelty in our proposed model.

- We advance the baseline model to account for crucial system parameters, where the number of Line of Sight (LoS) EEDs is finite, and requesters (i.e., task initiators) compete on the available EEDs. We then introduce a location-aware hiring technique. The requester will recruit the closest EEDs rather than selecting random EEDs; we will show how this will reduce average task response delay compared with the previous hiring technique. Furthermore, we consider the failure-prone nature of the EEDs by investigating the impact of their failure on the average task response delay. To this end, we introduce the system reliability rank that reflects the reliability degree of the available EEDs in the system.

1.4 Thesis Outline

The rest of this thesis is organized as follows: Chapter 2 introduces spatial and temporal modeling in wireless networks and gives an overview of existing spatiotemporal analysis frameworks in the literature. Chapter 3 presents our proposed baseline spatiotemporal analysis model. It also provides the method used to evaluate our mathematical model used to calculate the success recruitment probability and the average task response delay, our mathematical model validation, and a few numerical results, along side a comparison between offloading in EED versus MEC-enabled systems. Chapter 4 describes our advanced model, where instead of picking a random device to recruit, EEDs are recruited based on their distance to the requester, followed by our failure model. Chapter 4 also illustrates the incorporation of the failure model into the proposed ACTMC, and the way it is considered while calculating the average task response delay. Finally, Chapter 5 concludes our work and outlines some potential future directions.

Chapter 2

Computing Paradigms and Wireless Networks Modeling: Literature Review

The wireless networks modeling and analysis is a comprehensive topic that involves a massive number of building blocks, especially for the envisaged applications that combine the computation and the communications aspects of the proposed networks. To conclude the overview related to this thesis research topic, which is a spatiotemporal analysis framework, we focus in this chapter on tools that are employed in the following technical chapters. In this chapter, we provide an overview of the most prominent computing paradigms, including cloud computing, Multi-access Edge computing, and mist computing (which hereafter is referred to as extreme edge computing). Considering that wireless networks modeling can help analyze task offloading in EED-enabled computing environments, we discuss wireless networks spatial and temporal modeling and provide a literature review of various techniques. Finally, we provide a literature review of spatiotemporal analysis and modeling in edge computing.

2.1 Cloud, Multi-access Edge, and Extreme Edge Computing

In cloud computing, computational tasks are offloaded to distant cloud centers to be executed, and the output is sent back to the task requester. The evolution of cloud computing offered a solution for many different problems related to data processing and task execution, such as high task execution latency, and big data processing. Most of those problems surfaced after the emergence of IoT technology, which was first introduced to the community in the late 1999s [14]. However, IoT was firstly introduced as a system specializing in supply management. Now with the development of IoT applications, there is a huge amount of data and computational power being generated by devices, but these devices were not considered as a source of computational power in the past. Moreover, IoT devices are now integrated

into many fields, such as health care systems [15,16] and autonomous vehicles [17,18], which require ultra-reliable communication and low latency in task execution when it comes to the sensitivity of the generated data and tasks.

With the statistics reported by Cisco, it is estimated that by 2025 there will be 38.6 billion connected devices worldwide [4]. Some of these IoT devices will demand very low latency, and some may generate a significant amount of data, which may cause problems in the networks. Although efficient in many aspects, cloud computing alone cannot accommodate such applications and process the vast amount of generated data and the installed IoT applications constraints. Moving the computational power closer to the users will save time and alleviate the traffic load at backhual links, reduce delay, and satisfy the stringent QoS constraints associated with these applications.

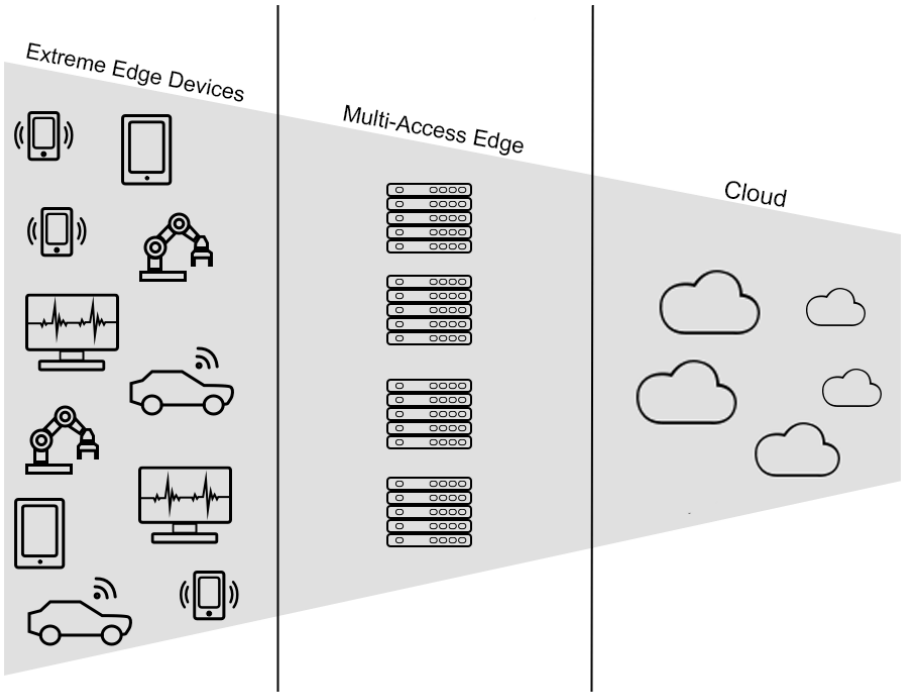

Figure 1: Extreme edge computing compared to MEC and cloud computing

Multi-access Edge Computing (MEC) has emerged as a promising computing paradigm that brings the computing service in close proximity to end devices, thus significantly reducing the delay and successfully satisfying the soaring latency demands of IoT applications [8].

In MEC, efficient task offloading decisions are pivotal to achieving promising performance gains, as offloading the task to the best computational unit is required in order to satisfy the tasks' demanded requirements. Most existing MEC platforms depend on the availability of computationally capable edge servers (i.e., Base Stations (BSs)) to perform the offloaded computational tasks [8].

Recently, various research efforts [9–11] have explored the potential of leveraging the drastic surge in IoT devices, also referred to as Extreme Edge Devices (EEDs) [12], and exploiting their collective processing capabilities to further improve the computing performance further. EEDs are recruited to expand the compute resource pool, perform parallel computing, and enhance the task offloading service in EED-enabled computing environments, which can also be called Extreme Edge Computing (EEC). While the vast majority of developed Edge computing models and implemented platforms consist of dedicated infrastructure-based edge nodes (e.g., edge servers, base stations, and smart access points) that are solely controlled by cloud service providers and network operators, EEC can democratize the edge and break such monopoly. This can allow more players to develop and control their own cloud, which can affordably support many Internet of Things (IoT) applications, thus enabling more businesses and enterprises to enter this sector. This can have ground-breaking impacts on speeding up the deployment of feasible pervasive smart home/building/transportation/city applications.

Task offloading is one of the most challenging issues in edge computing environments. To enable efficient task offloading, the spatial and temporal network parameters need to be considered. Towards that end, the use of wireless networks spatial and temporal modeling can be leveraged.

2.2 Wireless Networks Spatial and Temporal Modeling

In this section, we provide an overview and a literature review of the tools and mathematical models associated with wireless networks spatial and temporal modeling.

2.2.1 Wireless Networks Spatial Modeling

To fully address the heterogeneity in a network and in order to provide significant insights, network spatial parameters (e.g., device location, number of devices) should be modeled and optimized using a collective cross-layer architecture. The main goal of the network spatial modeling phase is to develop mathematical expressions that describe and characterize networks random behavior and architecture. For example, the location of the devices in the network does not follow a certain pattern when it is distributed. Typically, the input for the network modeling are the networks parameters, which mainly describes the nature of the network. After building a spatial model that reflects the nature of the desired network, a performance analysis can be performed to understand and study the system behavior and how the system is expected to react after changing some of those parameters.

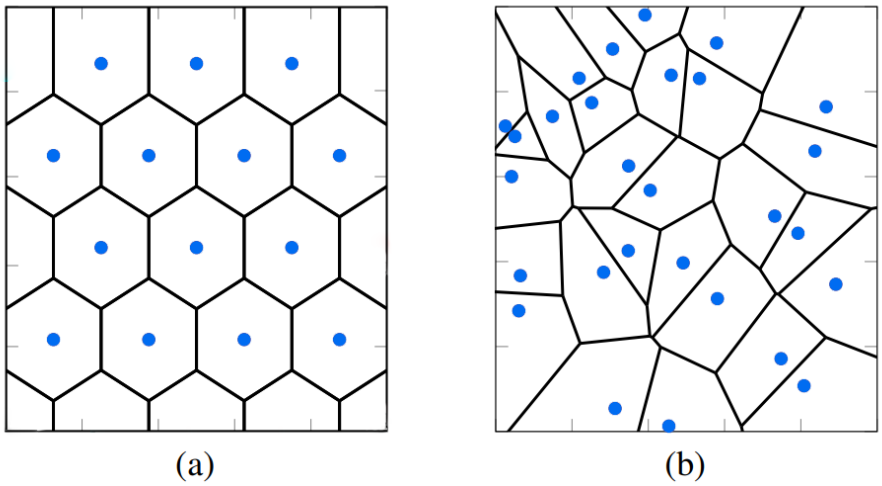

Figure 2: Wireless networks spatial modeling using (a) Grid-based model and (b) Stochastic Geometry

A widely used spatial model is the grid-based model, which is good for large-scale infrastructure-based wireless networks [19]. As depicted in Figure 2(a), the devices are deployed on a shaped architecture. Real world networks implementation is far from being standard as shown in [20]. Moreover, the gap between theoretical and actual installations grows in case of heterogeneous devices deployments, where each cell has its coverage area,

and BS capabilities in case of BS deployment.

An alternative effective model is network simulations, in which the network is simulated many times in order to average out all of those simulations and get average results, and to eliminate any source of randomness. However, simulations can be time and cost consuming due to the need of doing the simulation so many times, in order to get the most accurate results after averaging. In this context, stochastic geometry is a powerful tool that can be used to capture the randomness in the networks, and to offer tractable and accurate mathematical expressions that can be used further more in the analysis [20]. Specifically, stochastic geometry studies the geographic average performance of a network in which nodes (i.e., BSs, devices, or both) are distributed according to a specified distribution over a large enough spatial plane [21], and each spatial deployment is weighted by its chance of occurrence [22]. Furthermore, to simulate the geographical distribution of the network nodes, point processes are used [23]. Due to its mathematical tractability, the Poisson point process (PPP) is the most popular point process for modeling wireless networks [23]. Figure 2(b) depicts an example of a PPP-based network deployment. When it comes to generating mathematical expressions for complicated network scenarios, PPP-based models offers simpler mathematical models than the grid model.

2.2.2 Wireless Networks Temporal Modeling

Due to the existence of temporal dynamics in the network devices and the tasks itself (i.e., devices recruitment time, devices arrival rate, task finishing time) in this context, queuing models are utilized to account for that nature. The complexity of the design of any queuing model may vary from one system to another depending on the number of features used and the temporal dynamic of these features, which may not be very common in other types of models [24]. One approach used to express queuing model is Markov chain. Markov chain is a mathematical model that experiences transitions from one state to another based on some probabilities or rates. For example, a system can go from state i to state j based on

probability p, or after spending time r at that state. A first order Markov chain, which is the used in our model, is distinguished by the fact that the potential future states are fixed, regardless of how the process arrives to its current state. In other words, the chance of transitioning to any given state is purely determined by the present state only, and any previous transitions are not counted while deciding to move to the next state [25]. Generally, Markov chains can be expressed in two types [25]:

Discrete-time Markov chain (DTMC), which models the systems where the transitions involve discrete time steps, and any change in the system will happen in one of those discrete time steps [25]. In this type of Markov chains, the transition is usually defined as the probability to go from the current state to all the future states. The summation of those probabilities must equal one.

Continuous-time Markov chain (CTMC), which models the systems where the transitions involve continuous time steps [25]. Any change in the system happens at any moment while the system is running. In this type, the transitions are usually described by the rates to go from the current state to all the future states [25].

An absorbing continuous time Markov chain (ACTMC) is a special case of Markov chains. This type of Markov chains allows the system to continue transitioning between states until it reaches a set of states (often referred to as a class) that, once entered, cannot leave [26]. This ACTMC is chosen to track the temporal dynamics in our system parameters (e.g., devices hiring time, task finishing time, etc.).

The following is a description of some of the essential key concepts of the DTMC, which also holds for the CTMC and ACTMC. Let $X_0, X_1, ..., X_n; n \in \mathbb{N}$ be a discrete time stochastic process, with state space $Z = \{z_1, z_2, ..., z_n\}$. If $\mathbb{P}\{X_{i+1} = z_{i+1} | X_i = z_i, X_{i-1} = z_{i+1}, ..., X_0 = z_0\} = \mathbb{P}\{X_{i+1} = z_{i+1} | X_i = z_i\}$ holds for any i and Z, then X_i is said to be DTMC [24]. The following $n \times n$ grid in (1) represents an example of ACTMC, where state n is the absorption state, and Λ_i is the summation of all the elements in the i_{th} row

$$\mathbf{Q} = \begin{array}{c} z \\ 0 \\ 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & \cdots & n \end{array} \\ \begin{bmatrix} -\mathbf{\Lambda_0} & \lambda_{\mathbf{0,1}} & \lambda_{\mathbf{0,2}} & \cdots & \lambda_{\mathbf{0,n}} \\ \lambda_{\mathbf{1,0}} & -\mathbf{\Lambda_1} & \lambda_{\mathbf{1,2}} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{\Lambda_2} & \ddots & \lambda_{2,n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \lambda_{\mathbf{n-1,n}} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \end{array} \tag{1}$$

2.3 Spatiotemporal Modeling and Analysis in Edge Computing

Task offloading in MEC environments is largely dependent on the availability and reachability of the resources, and their resilience to failures [27]. Service interruptions triggered by failures of physical machines (PMs) and virtual machines (VMs) are addressed in [28]. In [29], the scalability of the network is explored in wireless-based task offloading, and both the communication and computation performance bounds are determined. The authors investigate and analyze the communication latency and computation latency under a variety of network parameters, and they focused on the architecture of the network and how this will affect the communication latency. However, they assumed that the task is not divided into slices/jobs and users are not able to offload to more than one access point (AP). In addition, they only considered fixed-sized tasks. The work in [30] analyzes task offloading under heterogeneous computational resources by estimating the network-wide outage probability. To perform energy-efficient task offloading, the spatial and temporal network parameters are considered in [31].

The authors in [32] proposed an offloading decision strategy for a simultaneous wireless information and power transfer mobile edge computing system, where the devices are considered low power devices. The offloading decision is made based on three trade-offs: energy, local computation, and offloading to the edge. A computation and communication trade-off

emerged after performing Monte Carlo simulations using different (SINR) values. In [33], the authors investigate the impact of applying non-orthogonal multiple access (NOMA) on improving the computation offloading performance of a mobile edge computing network. The authors develop a mathematical framework to analyze the impact of NOMA on MEC. This is done using stochastic geometry to model the system and extract an expression to evaluate the computation offloading probability based on the distance between the user and the serving BS. Results have shown that NOMA-based MEC outperforms the normal orthogonal multiple access (OMA) in certain task arrival rates. However, only one type of tasks has been considered, and without parallelization in the computation process [33].

The authors in [34] consider an MEC system that utilizes parallel computing to execute tasks on both the mobile device and the MEC server. The computing tasks operating on the mobile device can be executed and computed at the device itself or the the MEC server or both in parallel. The authors devise a Markov chain to determine the average latency of each activity, as well as the average power consumption of the mobile device. This input is used to formulate a searching algorithm to find the optimal stochastic computation offloading policy. Parallel computing is allowed between the device and the MEC server and the device is not allowed to offload to more than one MEC.

In [35], a spatiotemporal model is proposed for large MEC networks called SGedge. The model analyzes the network latency performance based on network performance indicators, while analyzing the trade-off between the communication and computation latency. Stochastic geometry and queuing theory are utilized to calculate the communication and computation latency and analyze the total network latency.

The above reviewed body of works either adopted a dependability perspective of the network [27, 28] or a spatiotemporal perspective [29–31]. Recent efforts have combined queuing theory with stochastic geometry to generate a complete large-scale networks' spatiotemporal characterization [34–39]. This spatiotemporal network approach has sparked a slew of new research lines aimed at modeling, evaluating, and optimizing networks from both spatial and

temporal perspectives. A combined view of both perspectives is provided by the spatiotemporal framework presented in [40]. The authors considered the joint limitation of network interference and parallel computing by multiple failure-prone VMs that reside on the same edge server. However, feasible and dependable task execution that accounts for the joint D2D communications under network-wide interference, as well as parallel computing at the EEDs, has been overlooked. Also, in this paper [41], the authors studied the network performance of a large-scale MEC wireless network, where the tasks can be computed locally by the local computation capabilities (LCCs) or offloaded to the MEC servers. The network is modeled using stochastic geometry and 2D discrete time Markov chain, the DTMC is used to characterize the task execution process and to keep track of the time that the task will take until it finishes execution, both locally and on the edge execution is included in the DTMC.

In this thesis, we propose a novel spatiotemporal framework that characterizes the task response time in EEC networks. The developed model accounts for interwoven communication and computation delays. We used an ACTMC to track the hiring and offloading to EEDs. The ACTMC tracked the parallel execution of tasks at the hired devices. The offloading rate is obtained by utilizing stochastic geometry analysis to obtain the successful hiring probability. We also proposed two ways of hiring new EEDs; it can be done by either choosing a random EED or choosing EEDs based on their distance order. We discuss how choosing the EED hiring method have a fast task response delay and a better optimal number of task slices, using various system parameters that cover many cases. Besides that, we also present our failure tracking model, which keeps track of the hired EEDs and the rate that those EEDs are going to fail. In case of failure, the requester hires a new EED based on the desired EED selection method. Finally, we validated our failure model and tested the proposed numerical model using different system parameters.

Chapter 3
Spatiotemporal Analysis For Randomly Recruited EEDs

Harvesting copious yet underutilized computational resources of the EEDs is foreseen as a promising endeavor. Such EEDs offer a unique opportunity to bring the computing service closer to IoT devices to curtail delay. However, the efficacy of extreme-edge parallel computing paradigm is profoundly impacted by i) wireless device-to-device communication performance that is required for task offloading; and ii) computing capabilities of EEDs, which govern the execution time of each task. In this context, we propose a novel spatiotemporal framework that employs stochastic geometry and continuous time Markov chains to jointly analyze the interwoven communication and computation performance of extreme edge computing systems. Based on the incorporated framework, we study the influence of various system parameters on the task response delay. In this chapter, we provide a detailed description of the system model. We then present the proposed spatiotemporal analysis. Next, we discuss the numerical results. Finally, we conclude and summarize the discussion.

3.1 System Model

The computationally capable EEDs, which hereafter are referred to as workers, are modeled via a PPP $\Phi \subset \mathbb{R}^2$ with intensity ν_w. The EEDs offer their computational services to resource-constrained devices (e.g., IoT), which hereafter are referred to requesters. The requesters are spatially distributed according to an independent PPP $\Omega \subset \mathbb{R}^2$ with intensity ν_r. There is an *edge orchestrator* that can be a base station or an access point, which organizes the offloading process between workers and requesters. In particular, the EEDs that have any available computational power register their availability at the edge orchestrator, which in turn informs each requester about the availability of proximate EEDs. Specifically, when a requester decides to offload a task to the surrounding EEDs, it asks the edge orchestrator to dispatch the available resources around it. In that context, the edge orchestrator does not have the location information, so it sends the devices in a random order. It is assumed

in this model that $\nu_w \gg \nu_r$, and hence, the edge orchestrator avoids conflicting the workers with more than one task slice. To utilize parallel computing and reduce response delay, the requester divides each computational task into n smaller and equivalent jobs to be offloaded and executed at different EEDs. Due to the heterogeneity of the computational powers of the EEDs, the finishing time of each job is exponentially distributed with mean $\frac{1}{n\mu_f}$, where μ_f is the task execution rate if computed at a single worker.

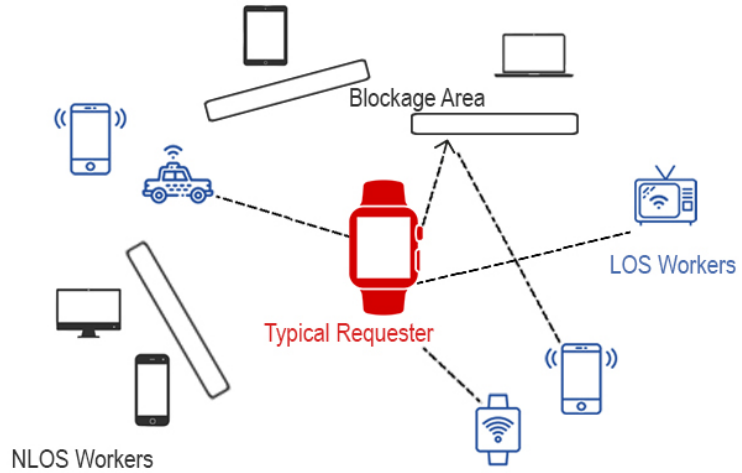

Figure 3: System Model

In compliance with 5G and beyond systems, the requesters utilize millimeter wave (mmW) for D2D communications to offload jobs to their proximate workers. The high vulnerability of mmW communications to blockage is considered via the general line of sight (LoS) ball blockage model [42, 43]. The devices within the distance of R_L from the requester are considered LoS devices, and otherwise, any device located after that point is considered non-Line of Sight (NLoS) devices. Distance-dependent power-law path-loss is considered with exponents α_L and α_N for LoS and NLoS devices, respectively. All transmissions experience Nagakami multi-path fading. Hence, the channel power gains have independent and identical gamma distribution parameters N_L for LoS devices and N_N for NLOS devices. We also ignore the fading in the frequency selective, as measurements show that the delay spread is generally small [44]. Also, results indicated that small-scale fading at mmWave is less severe than that in conventional systems when narrow beam antennas are used [44]. Thus, we can use a large

Nagakami parameter N_L to approximate the small-variance fading as found in the LOS case.

Universal frequency reuse and constant transmit power P is utilized via all requesters. The requester and workers deploy antenna arrays for mmW beamforming. The array patterns are approximated by the sectored antenna model with main lobe gain of M_x, side lobe gain of m_x, and 3 dB beamwidth of θ_x, where the subscript $x \in \{w, r\}$ to differentiate between the antenna patterns of the requesters and workers, Figure 4 illustrates the beamforming model used.

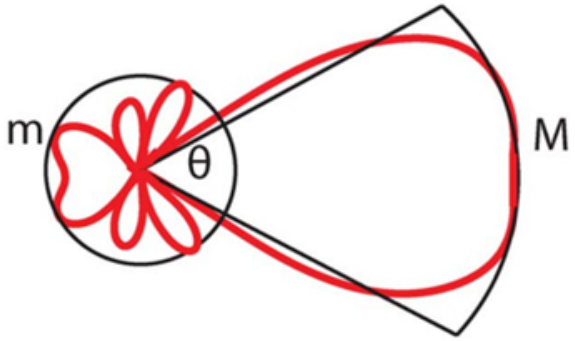

Figure 4: Beamforming Model

Without loss of generality, we consider that the picked requester is located at the origin and can establish D2D links with proximate LoS EEDs only. Therefore, perfect antenna alignment is considered for the intended D2D link, and uniform random antenna alignment is considered for the interfering links. A pictorial illustration of the system model is shown in Figure 3. The antenna gains for the interfering links and their corresponding probabilities are given in Table 1.

The requester is assumed to have one task sliced into n smaller and equal slices called jobs. The jobs are encapsulated into n packets transmitted via D2D communications to different proximate workers. Since a single mmW interface is available at the requester, the workers are sequentially recruited. The workers are selected either randomly among the list of available LoS EEDs provided by the edge orchestrator. The communication between the requester and worker is subject to errors, and hence, the requester may need several attempts to deliver the job packet and recruit the worker successfully. The time required for

each packet transmission attempt via D2D communications is τ_c seconds. The worker starts executing the job immediately upon the successful reception of the job. The requester is then notified to offload the pending jobs to other available LoS workers. All notifications are assumed to be sent over the perfect feedback channel. Once the worker finishes executing the task, the results are returned to the requester. Note that the communication time needed to return the results back to the requester is negligible since the results tend to be too small.

3.2 Successful Random EEDs Recruitment Probability

In order to calculate the average task response delay, we first need to obtain the average worker recruitment time, as recruitment is considered a part of the total execution time. We say that the worker correctly receives the job if the signal-to-interference-plus-noise ratio (SINR) is above a given threshold ξ. Otherwise, the job has to be re-transmitted to another LoS worker. Hence, the first step in investigating the response delay is to find the D2D communication success probability between the requester and the randomly selected LoS worker to recruit. As we will see in the next section, such probability is then utilized within an ACTMC to find the average response delay. The received SINR at the intended worker is represented as given by Eq. 2.

$$\text{SINR} = \frac{|h_0|^2 M_r M_w C_L r_0^{-\alpha}}{\sigma^2 + I_N + I_L} \tag{2}$$

The successful D2D transmission of the job probability can be expressed as

$$p_s = \mathbb{P}\left\{\text{SINR} > \xi\right\} = \mathbb{P}\left\{\frac{h_0 M_r M_w C_L r_0^{-\alpha_L}}{\sigma^2 + I_N + I_L} > \xi\right\} \tag{3}$$

where h_0 is the intended channel power gain, C_L is the intercept of the LoS channel, r_0 is the distance between the requester and the intended LoS worker, I_L is the aggregate interference from other active LoS requesters, I_N is the aggregate interference from other active NLoS requesters, and σ^2 is the ambient noise power. Let $\Omega_L \subset \Omega$ and $\Omega_N = \Omega \setminus \{(\Omega_L) \cup (0,0)\}$ be

Table 1: Directivity gain probability and value

k	1	2	3
a_k	M_w^2	$M_w m_w$	m_w^2
b_k	c^2	$2c(1-c)$	$(1-c_r)^2$

the point process of the LoS and NLoS requesters seen from the origin, respectively. Then, the LoS and NLoS interference terms are expressed as as given by Eq. 4,

$$I_L = \sum_{i>0:\mathbf{x_i}\in\mathbf{\Omega_L}} h_i D_i C_L \left\|\mathbf{x_i}\right\|^{-\alpha_L}, \tag{4}$$

and

$$I_N = \sum_{i>0:\mathbf{y_i}\in\mathbf{\Omega_N}} g_i D_i C_N \left\|\mathbf{y_i}\right\|^{-\alpha_N}, \tag{5}$$

where h_i is the i^{th} LoS interfering link channel power gain, g_i is the i^{th} NLoS interfering link channel power gain, C_N is the intercept of the NLoS channel, $\|\cdot\|$ is the Euclidean norm, and D_i is the antenna gain width for the i^{th} interfering requester in Ω_L or Ω_N. Due to the sectored antenna model along with the uniformly random antenna alignment, D_i is a discrete random variable with the probability distribution defined as $\mathbb{P}\{D_i = a_k\} = b_k$ with k $\in$ {1, 2, 3}, where a_k and b_k are constants defined in Table 1 and $c = \theta_r/2\pi$.

The D2D transmission success probability given by Eq. 3 is characterized in the following lemma.

Lemma 1. *The spatially averaged successful recruitment probability via mmW D2D communications for a randomly selected LoS worker out of Φ_w is given by Eq. 6,*

$$p_s = \int_0^{R_L} \sum_{n=1}^{N_L} \binom{N_L}{n} \frac{2r(-1)^{n+1} e^{M_n(\xi)\sigma^2 - W_n(\xi) - Z_n(\xi)}}{R_L^2} dr \tag{6}$$

where $M_n(\xi) = -\frac{\eta_L n r_0^{\alpha_L} \xi}{C_L M_r M_w}$, while $W_n(\xi)$ and $Z_n(\xi)$ are given by Eq. 7 and Eq. 8

$$W_n(\xi) = 2\pi\nu_r b_k \int_0^{R_L} \left(1 - \frac{1}{\left(1 + \frac{\eta_L \bar{a}_k n \xi (\frac{r_0}{x})^{\alpha_L}}{N_L}\right)^{N_L}}\right) x dx \tag{7}$$

and

$$Z_n(\xi) = 2\pi\nu_r b_k \int_{R_L}^{\infty} \left(1 - \frac{1}{\left(1 + \frac{n_L \bar{a}_k n \xi C_N r_0^{\alpha_L}}{C_L x^{\alpha_N} N_N}\right)^{N_N}}\right) x dx \tag{8}$$

Proof. The coverage probability as implied in Eq. 3 is described as the probability of having $P(\xi) = \mathbb{P}(\text{SINR} \geq \xi)$ where ξ is predefined threshold. Let g be a normalized gamma random variable with parameter N. For a constant $\mu > 0$, the probability $\mathbb{P}(g > \mu)$ can be tightly upper bounded as give by Eq. 9, where $a = N(N!^{-1/N})$ [45]

$$\mathbb{P}(g < \mu) < [1 - e^{-a\mu}]^N \tag{9}$$

Based on that, we can rewrite Eq. 3 as given by Eq 10, where $M_n(\xi) = -\frac{\eta_L n r_0^{\alpha_L} T}{C_L M_r M_w}$.

$$\begin{aligned}
p_s &= \mathbb{P}\left\{h_0 \geq \frac{R_0^{\alpha} \xi (\sigma^2 + I_L + I_N)}{(C_L M_r M_w)}\right\} \\
&= 1 - \mathbb{E}_{\Phi}\left\{\left(1 - e^{M_n(\xi)(\sigma^2 + I_L + I_N)}\right)^{N_L}\right\} \\
&\overset{(a)}{=} \sum_{n=1}^{N_L} (-1)^{n+1} \binom{N_L}{n} \mathbb{E}_{\Phi}\left[e^{M_n(\xi)(\sigma^2 + I_L + I_N)}\right] \\
&\overset{(b)}{=} \sum_{n=1}^{N_L} (-1)^{n+1} \binom{N_L}{n} e^{M_n(\xi)\sigma^2} \mathbb{E}_{\Phi_L}\left[e^{M_n(\xi) I_L}\right] \mathbb{E}_{\Phi_N}\left[e^{M_n(\xi) I_N}\right]
\end{aligned} \tag{10}$$

Eq. 10(a) follows from the Binomial theorem based on the assumption that N_L is an integer, Eq. 10 (b) comes from the fact that Φ_L and Φ_N are independent PPPs. To work on that more, we apply some concepts from stochastic geometry to compute the LoS interfering term $\mathbb{E}_{\Phi_L}\left[e^{M_n(\xi) I_L}\right]$ in Eq. 10 as given in Eq. 11, where g in Eq. 11 (c) is normalized gamma

random variable with parameter N_L, $\bar{a}_k = \frac{a_k}{M_r M_w}$, b_k and a_k for $1 \leq k \leq 3$ is defined in Table 1; Eq. 11 (c) follows from the probability generating functional of the PPP Φ_L [46], and Eq. 11 (d) follows from computing the moment generating function of the gamma random variable g, and $\eta_L = N_L(N_L!)^{-1/N_L}$.

$$
\begin{aligned}
&\mathbb{E}_{\Phi_L}\left[\exp\{-\frac{\eta_L n r_0^{\alpha}\xi I_L}{C_L M_r M_w}\}\right] \\
&= \mathbb{E}_{\Phi_L}\left[\exp\{-\frac{\eta_L n r_0^{\alpha}\xi \sum_{i>0:x_i\in\Phi_L} |h_i|^2 D_i \left\|x_i\right\|^{-\alpha_L}}{M_r M_w}\}\right] \\
&\overset{(c)}{=} \exp\left\{-2\pi\lambda_r \sum_{k=1}^{3} b_k \int_{r_0}^{R_L} \left(1 - \mathbb{E}_g\left[e^{-nT\eta_L g \bar{a}_k (\frac{r_0}{x})^{\alpha_L}}\right]\right) dx\right\} \\
&\overset{(d)}{=} \prod_{k=1}^{3} \exp\left\{-2\pi\lambda_r b_k \int_{r_0}^{R_L} \left(x - x/\Big(1 + \frac{\eta_L \bar{a}_k n \xi (\frac{r_0}{x})^{\alpha_L}}{N_L}\Big)^{N_L}\right) dx\right\} \\
&= e^{-W_n(T)}
\end{aligned} \tag{11}
$$

Likewise, for the NLoS interference $\mathbb{E}_{\Phi_N}\left[e^{M_n(\xi) I_N}\right]$, the small-scale fading term $|g_i|^2$ is a normalized gamma random variable with parameter N_N. Thus, it can be computed as Eq. 12.

$$
\begin{aligned}
&\mathbb{E}_{\Phi_N}\left[\exp\{-\frac{n\eta_L r_0^{\alpha_L}\xi I_N}{C_L M_r M_w}\}\right] \\
&= \mathbb{E}_{\Phi_N}\left[\exp\{-\frac{n\eta_L C_N r_0^{\alpha_L}\xi \sum_{i>0:y_i\in\Phi_N} |g_i|^2 D_i \left\|y_i\right\|^{-\alpha_N}}{C_L M_r M_w}\}\right] \\
&= \prod_{k=1}^{3} \exp\left\{-2\pi\lambda_r b_k \int_{R_L}^{\infty} \left(x - x/\Big(1 + \frac{\eta_L \bar{a}_k n \xi C_N r_0^{\alpha_L}}{C_L x^{\alpha_N} N_N}\Big)^{N_N}\right)\right\} \\
&= e^{-D_n(\xi)}
\end{aligned} \tag{12}
$$

After that, combining Eq. 11 and Eq. 12 with Eq. 10 will give us the closed form to

calculate the successfully recruitment probability between requester x and worker y as given by Eq. 13

$$p_{s_{xy}}(T) = \sum_{n=1}^{N_L} (-1)^{n+1} \binom{N_L}{n} e^{M_n(T) - W_n(T) - D_n(T)} \tag{13}$$

The distance between the devices is a random variable denoted by r_0, the recruitment probability for a randomly picked device is presented as given by Eq 14

$$p_s = \int_0^{R_L} \sum_{n=1}^{N_L} \binom{N_L}{n} \frac{2r_0(-1)^{n+1} e^{M_n(\xi)\sigma^2 - W_n(\xi) - Z_n(\xi)}}{R_L^2} dr_0 \tag{14}$$

□

3.3 Average Task Response Delay

The average task response delay is defined as the total time that n job slices take from the moment the requester decides to offload these job slices to the recruited EEDs until they are successfully executed. The total time that one job takes to be executed is represented as given by Eq. 15, where τ_h is the average recruitment time that the requester takes to recruit a randomly selected EED, and τ_f is the average time the job takes to be executed at the intended EED. The recruitment and finishing events can happen at any given moment, so in order to represent them, we model the system as an ACTMC and use it to estimate the average task response delay for a given number of jobs n. The successful recruitment probability estimated in Lemma 1 is a core building block of the ACTMC, the average recruitment rate $\lambda_h = 1/\tau_h$, where $\tau_h = p_s/\tau_c$, p_s is the probability given in Lemma 1, and τ_c is the average D2D communication time in mmW networks.

$$T_o = \tau_h + \tau_f \tag{15}$$

In the following subsections, we discuss the underlying CTMC and EDTMC used.

3.3.1 CTMC and EDTMC

The states set of the ACTMC is represented as $\mathcal{S} = \{\mathbf{z} = (x_f, x_c) | \sum_j x_j \leq n; j \in \{f, c\}\}$, where $x_f \in \{0, 1, 2, \cdots, n\}$ denotes the number of workers that have finished their assigned task successfully, and $x_c \in \{0, 1, 2, \cdots, n\}$ denotes the number of recruited workers that are recruited to execute the assigned job.

For each task, ACTMC starts at the state $\mathbf{z} = (0, 0)$, where the requester has a task that is sliced to n jobs, but has not yet recruited any worker. Each time the requester succeeds to recruit a LoS EED via mmW D2D transmission, a transition occurs from state $\mathbf{z}_i = (x_f, x_c)$ to state $\mathbf{z}_j = (x_f, x_c + 1)$. Also, Each time a worker is retired because of a job completion, a transition from state $\mathbf{z}_i = (x_f, x_c)$ to $\mathbf{z}_j = (x_f + 1, x_c - 1)$ occurs. Since the requester needs only n workers, then $x_c + x_f \leq n$ and $\mathbf{z} = (n, 0)$ is the absorbing state that implies the termination of the ACTMC. Following the aforementioned criterion, jobs offloading and execution at the EEDs can be tracked with an ACTMC with the following two-level hierarchical generator matrix

$\mathbf{z}_i = (x_f, x_c, x_d)$

$\mathbf{z}_j = (x_f, x_c + 1, x_d)$

$\mathbf{z}_j = (x_f + 1, x_c, x_d)$

$$\mathbf{Q} = \begin{array}{c} x_f \\ 0 \\ 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & \cdots & n \end{array} \\ \begin{bmatrix} \mathbf{K_0} & \mathbf{H_{0,1}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{K_1} & \mathbf{H_{1,2}} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K_2} & \mathbf{H_{2,3}} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{K_{n-1}} & \mathbf{H_{n-1,n}} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \end{array},$$

where $\mathbf{Q}$ is a block matrix of size $(n + 1) \times (n + 1)$ that tracks the number of finished workers x_f. Since the task is finished upon the completion of the n jobs, then the state

$x_f = n$ is the absorbing state that indicates the termination of the edge computing. Within each level of $\mathbf{Q}$, the sub-matrices $\mathbf{K}_m$ and $\mathbf{H}_{m,m+1}$ track the number of recruited workers x_c. Exploiting the fact that $x_c + x_f \leq n$, the matrix $\mathbf{H}_{m,m+1}$ is of size $(n-m) \times (n-m-1)$ that tracks x_c due to the completion of a job by any of the workers. Let $\mathbf{H}_{m,m+1}(i,j)$, with $i \in \{0,1,2,\cdots,n-m\}$ and $j \in \{0,1,2,\cdots,n-m-1\}$, denote that (i,j)-th element of the matrix $\mathbf{H}_{m,m+1}$. Then, due to the parallelism in the computing at the EEDs along with the fact that only one worker can finish at a given instance, the matrix $\mathbf{H}_{m,m+1}$ is given by

$$\mathbf{H}_{m,m+1}= \begin{array}{c} X_C \\ 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ (n-m-1) \\ (n-m) \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & n-m-2 & n-m-1 & n-m \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \mu_f & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2\mu_f & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 3\mu_f & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & (n-m-1)\mu_f & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & (n-m)\mu_f & 0 \end{bmatrix} \end{array} \tag{16}$$

Using similar argument, the matrix $\mathbf{K}_m$ is of size $(n-m+1) \times (n-m+1)$ that tracks x_c upon recruiting new workers. Let $\mathbf{K}_m(i,j)$, with $i,j \in \{0,1,2,\cdots,n-m\}$ denote that (i,j)-th element of the matrix $\mathbf{K}_m$. Then, due to the sequential worker recruitment, we have

$$\mathbf{K}_m= \begin{array}{c} X_C \\ 0 \\ 1 \\ 2 \\ \vdots \\ n-m-1 \\ (n-m) \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & 3 & \cdots & n-m-1 & n-m \end{array} \\ \begin{bmatrix} -\lambda_h & \lambda_h & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\lambda_h-\mu_f & \lambda_h & 0 & \ddots & 0 & 0 \\ 0 & 0 & -\lambda_h-2\mu_f & \lambda_h & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & -\lambda_h-(n-m-1)\mu_f & \lambda_h \\ 0 & 0 & 0 & 0 & 0 & 0 & -(n-m)\mu_f \end{bmatrix} \end{array} \tag{17}$$

where $\lambda_h = p_s/\tau_c$ is the recruiting rate, p_s is the D2D transmission success probability given

in Eq. 6, and τ_c is the time required for each D2D transmission attempt.

The average task response delay cannot be directly obtained for the matrix $\mathbf{Q}$. Instead, we first need to obtain the embedded discrete time Markov chain (EDTMC) of $\mathbf{Q}$ and the average sojourn time at each state. The EDTMC of $\mathbf{Q}$ is given by

$$\mathbf{P} = \begin{array}{c} \begin{matrix} x_f & 0 & 1 & 2 & 3 & \cdots & n \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{matrix} \begin{bmatrix} \mathcal{K}_{\mathbf{0}} & \mathcal{H}_{\mathbf{0,1}} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathcal{K}_{\mathbf{1}} & \mathcal{H}_{\mathbf{1,2}} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{K}_{\mathbf{2}} & \mathcal{H}_{\mathbf{2,3}} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathcal{K}_{\mathbf{n-1}} & \mathcal{H}_{\mathbf{n-1,n}} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \end{array} \tag{18}$$

where $\mathcal{K}_m$ and $\mathcal{H}_{m,m+1}$ track the transition probabilities due to worker recruitment and job completion, respectively. The matrices $\mathcal{K}_m$ and $\mathcal{H}_m$ are given by

$$\mathcal{K}_m = \begin{array}{c} \begin{matrix} X_C & 0 & 1 & 2 & 3 & \cdots & (n-m-1) & (n-m) \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ (n-m-1) \\ (n-m) \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{\lambda_h}{\lambda_h+\mu_f} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \frac{\lambda_h}{\lambda_h+2\mu_f} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{\lambda_h}{\lambda_h+(n-m-1)\mu_f} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \end{array}$$

and

$$\mathcal{H}_{m,m+1}= \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ (n-m-1) \\ (n-m) \end{array} \begin{array}{c} \begin{array}{ccccccccc} X_C & 0 & 1 & 2 & 3 & \cdots & (n-m-2) & (n-m-1) & (n-m) \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \frac{\mu_f}{\lambda_h+\mu_f} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \frac{2\mu_f}{\lambda_h+2\mu_f} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \frac{3\mu_f}{\lambda_h+3\mu_f} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{(n-m-1)\mu_f}{\lambda_h+(n-m-1)\mu_f} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \end{array}$$

3.3.2 Average Time until Absorption

After showing how we represent the ACTMC and how we obtain the matrices $\mathbf{Q}$ and $\mathbf{P}$, we now utilize those to calculate the average task response delay. Our ACTMC has an absorbing state, which after reaching the n jobs will be successfully executed at the recruited EEDs. Based on that, the average task response delay is equivalent to the average time until absorption in that state. To calculate the average time until absorption, let $x_{c_i} \in \mathbf{z}_i$ be the number of recruited workers in state $\mathbf{z}_i$, then the average sojourn time $t_{\mathbf{z}_i,\mathbf{z}_j}$ is given by

$$t_{\mathbf{z}_i,\mathbf{z}_j} = \begin{cases} \frac{1}{x_{c_i}\mu_f}, & \text{if the transition from } \mathbf{z}_i \text{ to } \mathbf{z}_j \\ & \text{is due to job completion} \\ \frac{1}{\lambda_h}, & \text{if the transition from } \mathbf{z}_i \text{ to } \mathbf{z}_j \\ & \text{is due to worker recruitment} \end{cases} \tag{19}$$

Equipped with $\mathbf{P}$ and $t_{\mathbf{z}_i,\mathbf{z}_j}$, the average task response delay is given in the following Lemma.

Lemma 2. *The average task response delay in the extreme edge computing networks with mmW D2D communications and n recruited workers is given by Eq. 20*

$$T_A = \alpha(\mathbf{I} - \mathbf{P}_T)^{-1}\mathbf{w}, \tag{20}$$

where $\alpha = [1, 0, 0, \cdots, 0]$, $\mathbf{I}$ is the identity matrix, $\mathbf{P}_T$ is the transition probability of the transient states only in $\mathbf{P}$, given in 18, which obtained by excluding the transitions to the absorbing state (the last row and column of $\mathbf{P}$). The column vector $\mathbf{w}$ contains the average sojourn times at states $\mathbf{z}_i$, which are given by $w_{\mathbf{z}_i} = \sum_{\mathbf{z}_j} \mathbf{P}(\mathbf{z}_i, \mathbf{z}_j) t_{\mathbf{z}_i, \mathbf{z}_j}$, where $\mathbf{P}(\mathbf{z}_i, \mathbf{z}_j)$ is the transition probability from state $\mathbf{z}_i$ to $\mathbf{z}_i$.[1]

Proof. Let $\mathcal{S}_A = \mathcal{S} \setminus (n, 0)$ be the entire state space of the ACTMC excluding the absorbing state. Then, the time to absorption from a state $\mathbf{z}_i \in \mathcal{S}_A$ is given by Eq. 21, where $\mathbf{P}$ is the EDTMC given in (18), $t_{\mathbf{z_i}}$ is given from (19), and $T_{\mathbf{z_j}}$ is the time until absorption starting from state $\mathbf{z_j}$.

$$T_{\mathbf{z}_i} = \sum_{\mathbf{z}_j \in \mathcal{S}_A} \mathbf{P}(\mathbf{z}_i, \mathbf{z}_j)(t_{\mathbf{z}_i, \mathbf{z}_j} + T_{\mathbf{z}_j}). \tag{21}$$

For every $\mathbf{z_i} \neq \mathbf{z_s}$, where $\mathbf{z_s}$ is the absorption state, $\mathbf{T} = (T_{\mathbf{z_i}})_{\mathbf{z_i} \neq \mathbf{z_s}}$ solves Eq. 22, where $\mathbf{P}_T = \mathbf{P}(\mathbf{z_i}, \mathbf{z_j})_{\,0 \leq i,j \leq n-1}$ is the transition probability of the transient states, and $\mathbf{w}$ is the average state sojourn time column vector, where $\mathbf{w}_{\mathbf{z_i}} = \sum_{\mathbf{z_j}} \mathbf{P}(\mathbf{z_i}, \mathbf{z_j}) / \mathbf{Q}(\mathbf{z_i}, \mathbf{z_j})$.

$$\mathbf{T} = \mathbf{w} + \mathbf{P}_T \mathbf{T} \tag{22}$$

Eq. 22 can also be written as given by Eq. 23.

$$\mathbf{T} = (\mathbf{I} - \mathbf{P}_T)^{-1}\mathbf{w}, \tag{23}$$

$\mathbf{T}$ is a column vector which its values are the average time until absorption starting from any state $\mathbf{z_i}$. To get the time from the first state, we multiply $\mathbf{T}$ by α to get Eq. 20. □

[1] In consistence with the hierarchical structure of $\mathbf{P}$, we utilize two dimensional indexing for its elements. Particularly, $\mathbf{P}(\mathbf{z}_i, \mathbf{z}_i) = \mathbf{P}\big((x_{f_i}, x_{c_i}), (x_{f_j}, x_{c_j})\big)$ is the (x_{c_i}, x_{c_j}) element of the matrix (x_{f_i}, x_{f_j}) sub-matrix in $\mathbf{P}$.

3.4 Optimal Time and Number of Recruited Devices

Upon calculating the average task response delay, we utilize it to calculate the optimal number of jobs slices n the requester should divide the desired task to yield the minimum response time. Towards that end, we rely on the fact that the computation and recruitment latency are profoundly affected by the number of jobs and the number of EEDs to which these jobs should be offloaded. However, on the other end, having a high number of job slices require many devices that must be recruited to complete those n jobs; this causes a high recruitment latency due to the high number of needed devices. Hence, we introduce Algorithm 1; this algorithm is used to calculate the optimal number of job slices based on the fact that, at some point, the computational latency and recruitment latency will reach their lowest. After that, increasing the number of jobs increases the recruitment delay more than it decreases the computational delay.

Algorithm 1 Optimal number of task jobs

Require: $(\mathbf{Q}, \mathbf{P})$
0: Set $n = 0, T(n) = \infty$
0: $\mathbf{w} = \sum_{\mathbf{z}_j} \mathbf{P}(\mathbf{z}_i, \mathbf{z}_j) t_{\mathbf{z}_i, \mathbf{z}_j}$
0: $\mathbf{P}_T = \mathbf{P}(\mathbf{z_i}, \mathbf{z_j})_{\,0 \le i,j \le n-1}$
0: **while** $T(n) \le T(n-1)$ **do**
0: $\quad n = n + 1$
0: $\quad$ Compute $T(n)$ from (20)
0: **end while**
Ensure: $T^* = T(n)$ and $n^* = n$ =0

3.5 System Analysis and Simulation Results

This section provides numerical and simulations results to validate the developed spatiotemporal model and illustrate the trade-off between the computation value and communication cost in extreme edge computing networks. Unless otherwise specified, the list of underlying

Table 2: Simulation Parameters

Parameter	Value
Workers Intensity (ν_w) / 10 km^2	$7 * 10^{-4}$
Requester Intensity (ν_r) / 10 km^2	$1 * 10^{-4}$
LOS and NLoS path loss exponent (α_L, α_L)	2, 4
Fading values for LoS and NLoS (N_L, N_N)	3, 2
Noise (σ^2)	-114 dBm
Maximum radius for LoS devices (R_L)	100m
SINR coverage probability threshold (ξ)	-10 dB
D2D communication time (τ_c)	1 second
Task finishing rate (μ_f)	0.02 task/second
System reliability rank (l)	3
Number of task slices (n)	5 slices

network parameters utilized in this section is summarized in Table 2. The Monte Carlo simulation are conducted over an area of 10 km^2. In each simulation run, a requester in the origin utilizes D2D communication to recruiter proximate LoS workers and the successful recruitment probabilities as well as the task response delay are recorded. The simulation results are then averaged over 10^5 runs. Figure 5 is a snapshot from one of the runs; it shows how the system is modeled in the desired area.

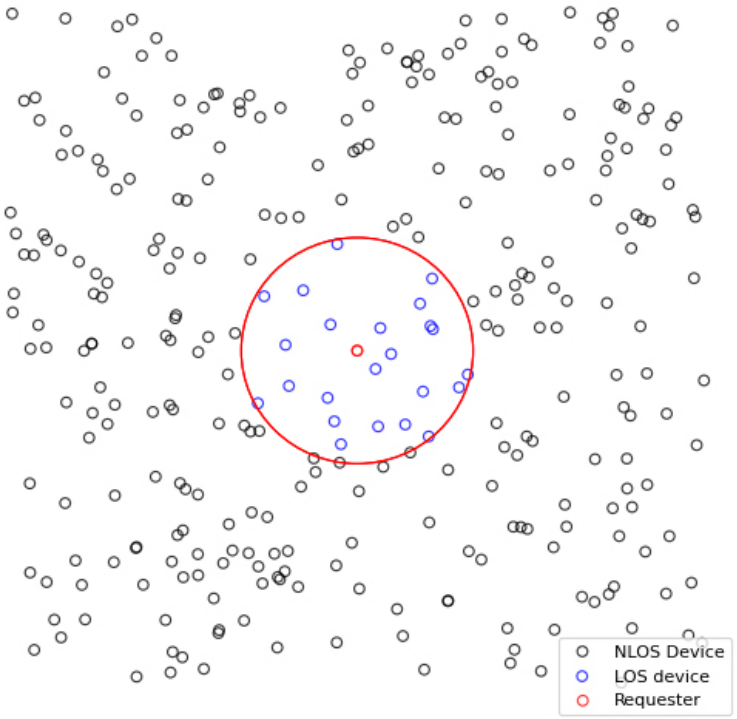

Figure 5: Simulated system model

3.5.1 Mathematical Model Validation

Using the default parameters mentioned in Table 2, this section will show our proposed model's validity alongside the simulation results obtained by simulating the system.

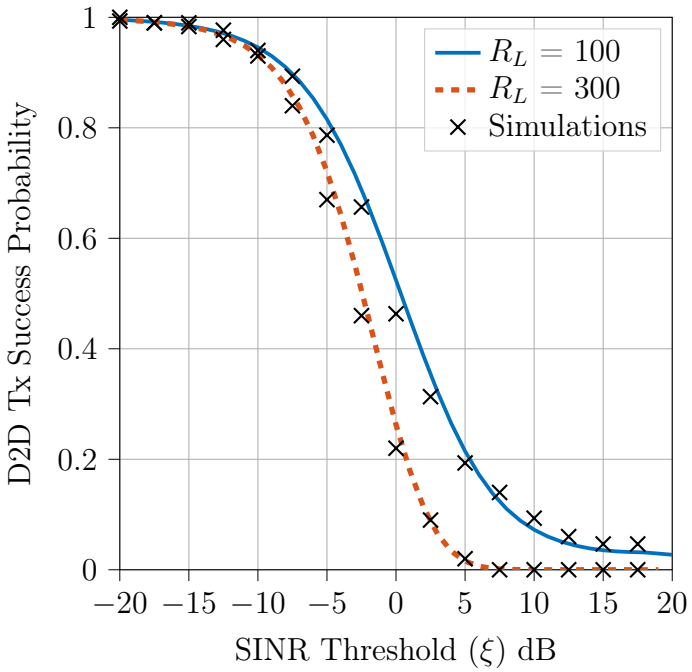

Figure 6: D2D recruitment success probability vs SINR threshold ξ

Figure 6 shows the successful recruitment probability p_x as a function of the desired SINR threshold ξ for different radius R_L values that enclose LoS devices. The close match demonstrated between the simulation, and the proposed analytical framework validates Lemma 1. The figure shows that the successful recruitment probability p_x is inversely proportional to ξ due to the increased link quality requirement. Hence, increasing ξ leads to more attempts to successfully offload a task to a worker, which increases the communication cost. The figure also shows that a larger R_L also increases the communication cost due to i) the higher probability of longer D2D distance between the requested and the workers and ii) the increased interference from other LoS requesters.

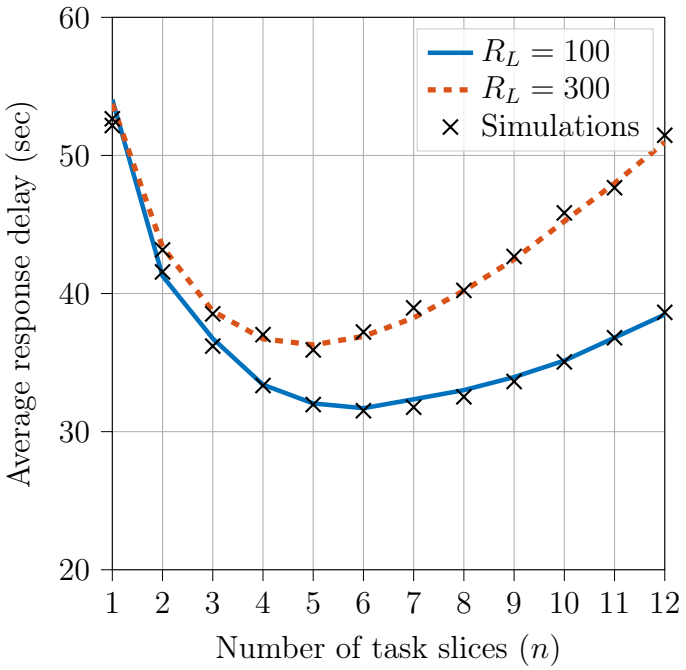

Figure 7: Average task response delay vs the number of task slices

Figure 7 depicts the system performance in terms of the average task response delay over a varying number of recruited workers given different values of R_L. The simulation and the proposed analytical framework closely coincide, validating Lemma 2. The figure reveals an important trade-off between the communication cost and the computation value in extreme edge systems. As the number of recruited workers increases, the total communication time increases, whereas the total computation time decreases. This is due to the increase in the number of collaborating devices among which the task is divided and with which communication occurs. As a result, the average task response delay continues to decrease as long as the reduction in computation delay is significantly predominant. This persists until reaching a point beyond which the increase in communication delay becomes too intense that it dominates the reduction in computation delay, thus causing the task response delay to start increasing. This indicates an optimal number of recruited workers that minimizes the task response delay.

3.5.2 Varying System Parameters

After the validation of the analysis has been demonstrated, we then conduct several experiments to show the impact of varying different system parameters on the average task response delay, as depicted in Figures 8, 9, 10.

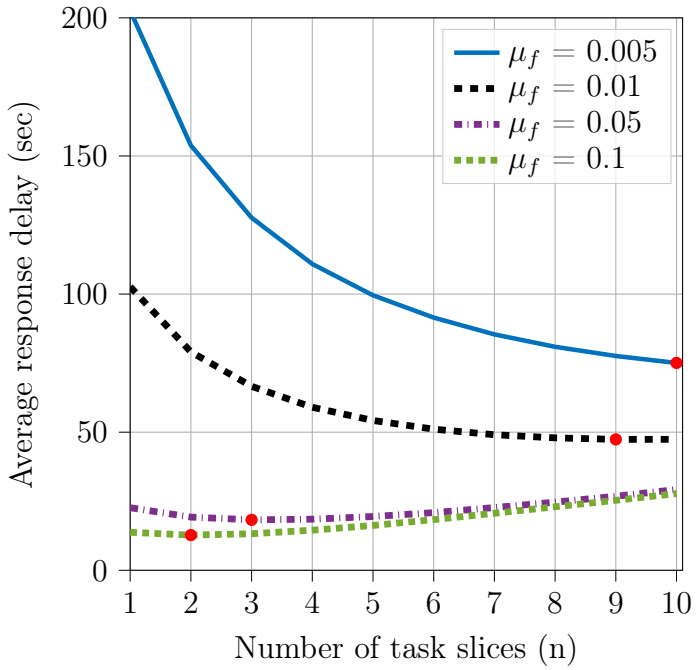

Figure 8: D2D recruitment success probability vs SINR threshold ξ

In this context, Figure 8 demonstrates the optimal number of task slices for different values of the task finishing rate (μ_f). As μ_f increases, the number of task slices that must be offloaded to EEDs to reach the minimum response time decreases. This can be attributed to the fact that the higher μ_f is, the lower the computation delay. This causes the system to reach the minimum response time with a fewer number of slices. Note that as the number of slices increases beyond this number, the number of recruited workers increases, causing the recruitment delay to dominate the reduced computation delay, which causes the average response delay to increase.

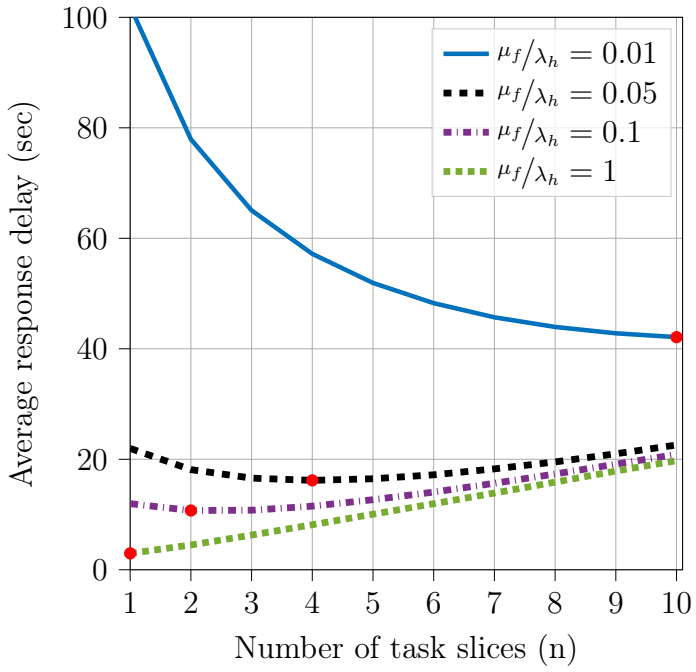

Figure 9: Average task response delay vs the number of task slices

Figure 9 shows the average task response delay over varying values of the ratio between the finishing rate and the recruitment rate, μ_f/λ_h. Note that as the ratio μ_f/λ_h decreases, the average task response delay increases. This is because the lower the ratio, the lower μ_f compared to λ_h, and the lower recruitment latency. This increases the total computation latency and causes it to be predominant over the communication latency, thus increasing the task response delay. In contrast, as the ratio increases, μ_f increases compared to λ_h, which reduces the total computation latency, and causes the total communication latency to dominate the reduced computation delay as the number of recruited workers increases.

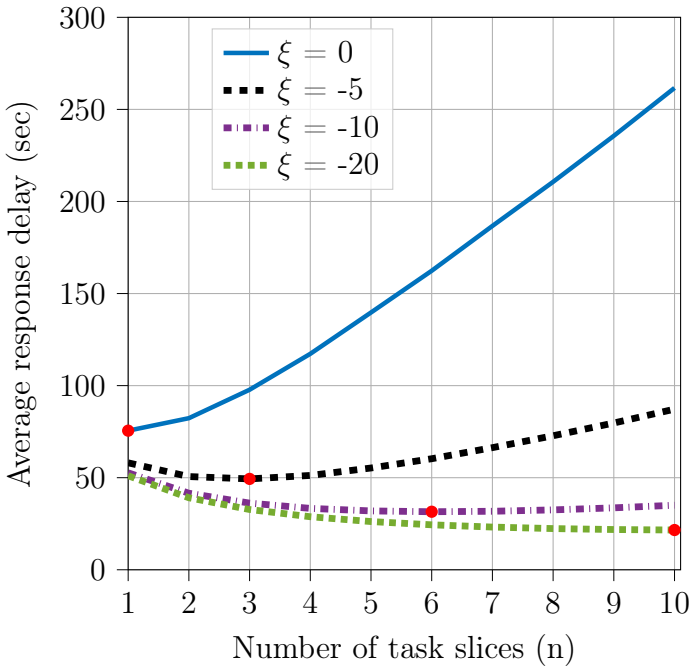

Figure 10: D2D recruitment success probability vs SINR threshold ξ

Figure 10 depicts the average task response delay over varying coverage probability thresholds ξ. As ξ decreases, the average response delay decreases. This is because as ξ decreases, the successful recruitment probability p_x increases, which increases the recruitment rate. By that happening, the network will be shifted into computing dominant, which will decrease the average task response time.

3.5.3 Benchmarking Against Conventional MEC

Using the BS as a computing resource by appending a PM with high computational power alongside it proved its effectiveness. However, the efficiency of the PM is affected by many parameters. For example, the distance between the BS and the requester plays a role in the communication latency, and the congestion on the PM plays a role in the computation latency. In that context, we consider a PM with computational power ten times better than the computational power available in the EED. PM can receive tasks from the surrounding requesters using mmWave. We will investigate different BS distance parameters R, and

different intensity $\nu_{r_{MEC}}$ values of the requesters use that PM. The PM communication latency is calculated the same way we calculate the communication latency for the EEDs, and the average computation latency at the PM is considered as an exponential random variable, with a rate of $\mu_{f_{MEC}} = 10 * \mu_f$.

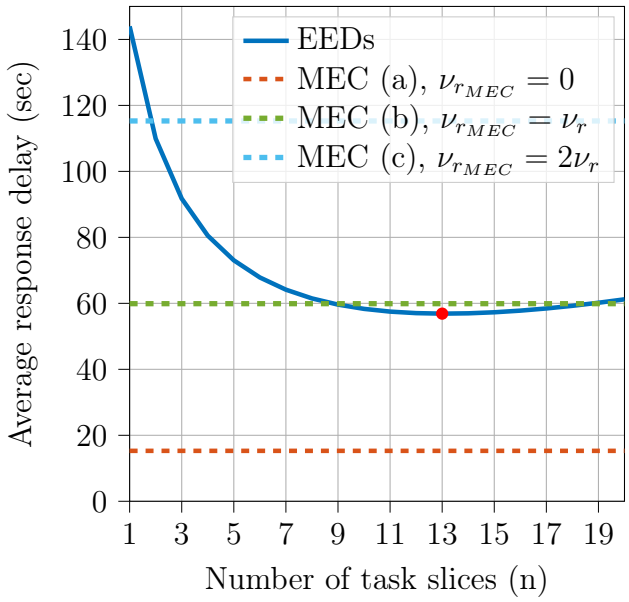

Figure 11: MEC and EEDs average response delay using varying BS congestion parameters

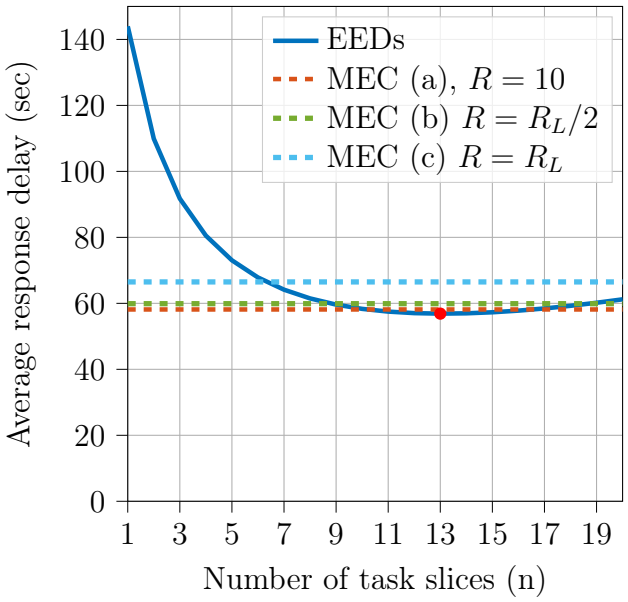

Figure 12: MEC and EEDs average response delay using varying BS distance parameters

Figure 11 and 12 illustrate the average task response delay from our model, alongside the response delay obtained from offloading the task as to the PM. The task used in this context is $\mu_f = 0.007$, and the intensity of the available requesters is $\nu_r = 5 \times 10^{-5}$ (the rest of the used parameters are specified in Table 2), and the requester will not divide the task before sending it, as it will be computed in the PM.

Figure 11 shows the average response delay using three different PM congestion cases: (a) the PM is currently not serving any other users ($\nu_{r_{MEC}} = 0$), in that case, the average task delay will be less than the optimal value when the task is offloaded to the EEDs, due to the availability of the computational resources at the PM. (b) the number of served requesters equals to all of the available requesters ($\nu_{r_{MEC}} = \nu_r$), but the PM is not fully congested,

this will lead to increasing the average task response delay, which happened because of the increase in the computational time due to the increase of the demand on the PM. (c) the PM is serving two times more than the the available requesters ($\nu_{r_{MEC}} = 2\nu_r$), and the PM is fully congested, in that case, offloading the task to the EEDs will be better than offloading it to the MEC, due to the response delay, the MEC will take compared to the delay from the EEDs.

Figure 12 shows the average response delay using three different distance values with $\nu_{r_{MEC}} = \nu_r$: (a) the distance between the requester and the BS $R = 10$, so the communication time will be at its best the BS is close to the requester. (b) the distance between the BS and the requester $R = R_L/2$, this will increase the communication time a little due to the increase in the distance, which aggravates the impact of fading and interference. Finally, (c) the BS is located on the farthest LoS point from the requester $R = R_L$, the communication time will be at its worst due to the low successful recruitment probability. Combining the three cases, we see that the distance increases the average response delay, and the communication time in the MEC case does not have a big effect.

3.5.4 Analysis on the results

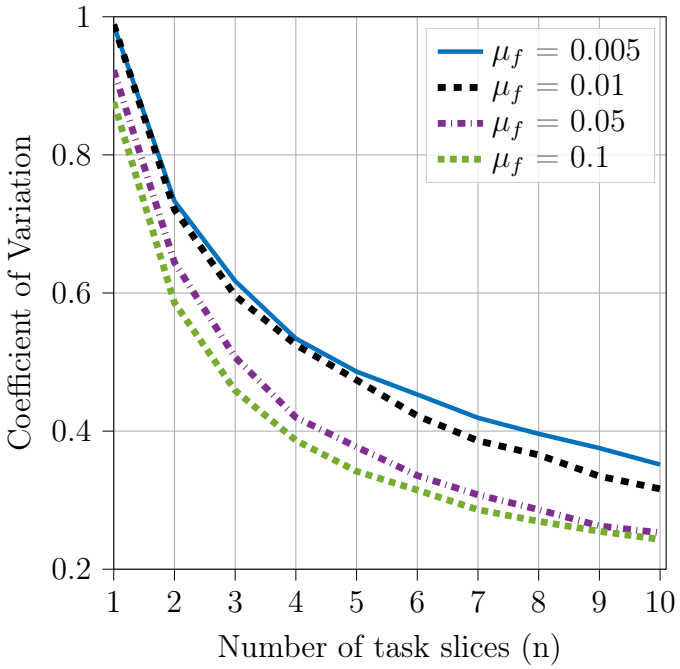

Figure 13: Coefficient of Variation of the Average time until absorption vs the number of task slices (n)

Figure 13 depicts the results after calculating the Coefficient of variation $CV = \sigma/\mu$ of the average time until absorption for different μ_f values. The CV is a measurement used to measure of dispersion in the results. The results show that the higher the task finishing rate μ_f, the lower the dispersion in the results. Also, the results show that the higher the number of task slices, the lower the dispersion in the results, which motivates the task slicing technique.

3.6 Conclusion and Discussion

This chapter presents the base of our spatiotemporal framework that characterizes the average task response delay in extreme edge computing networks. The developed model accounts for the interwoven communication and computation delays by constructing an ACTMC to track the sequential offloading and parallel execution of tasks at the EEDs. The offloading

rate is obtained via stochastic geometry analysis. The numerical results validate the analysis and reveal an optimal number (n^*) of recruited EEDs that minimizes the underlying task response time. Operating below n^* leads to underutilized edge computational resources and prolongs the task response delay. Exceeding n^* leads to a dominating recruitment delay that prolongs the response delay. To this end, it is demonstrated that the optimal number of recruited EEDs significantly varies with the network parameters as mentioned in the simulations section. We also showed that in the case of congesting the PM, our EEDs recruiting model could outperform offloading the task to the MEC.

Chapter 4

Location-Aware Spatiotemporal Analysis Over Prone to Failure EEDs

In the previous chapter, the presented model operates under the assumption that there is no scarcity in the number of EEDs, and the edge orchestrator does not have information about the EEDs locations. Thus, when a requester wants to recruit EEDs for task execution, it does so by randomly selecting devices from an infinite pool of EEDs. Also, it has been assumed that once the task once it is assigned to an EED, it will be executed successfully without any failure or interruption. To address the aforementioned deficiencies, we present an advanced model that considers the impact of failure, where EEDs might fail while executing a task slice. In addition, we investigate the impact of location awareness. Furthermore, we examine the impact of EEDs intensity on the average task response delay.

This chapter is organized as follows: First, we present the system model used. Second, we introduce the new recruitment criterion, and the effect of selecting the closest EEDs on the recruitment probability equation. Third, we present the failure model, and the modifications applied on ACTMC and EDTMC in order to track failure events. Finally, we validate the model and discuss the results under varying system parameters.

4.1 System Model

The system model in this chapter shares some similarities with the previously proposed model; the workers and the requesters are still modeled as PPPs with intensities ν_w and ν_r, respectively. The requester can only recruit the LoS EEDs due to the existence of blockages, and the recruitment process will be done after fetching the LoS EEDs information from the edge orchestrator, which is the entity that has the EEDs availability information.

In the baseline model, the edge orchestrator was not fully aware of the EEDs' locations in the network, so selecting an EED to recruit was done by picking a random EED from the LoS

devices pool. Also, we assumed that there would be no scarcity in the number of available EEDs, so at any time, if a requester is asking for an EED to recruit, it is guaranteed that there will be an available one. Furthermore, EEDs that are likely to fail while recruitment and executing a job were not included in the previous analysis. If an EED received the offloaded job, it was assumed that it would successfully execute that job without any failure or interruption in the process.

In addition to the similarities between the two models, there are multiple additional features that distinguish the advanced model. In particular, we assume that there is a limited number of available devices to each requester. Thus, when a requester decides to offload a task, the orchestrator is considered to have a pool that contains a maximum of m_L EEDs, where an average value of $m_L = \nu_w \pi R_L^2$. When a requester probes the orchestrator for information about the surrounding LoS EEDs, the orchestrator includes their location among the sent EEDs information. As depicted in Figure 14(a), we can take advantage of the the acquired location information and recruit the closest devices. Accordingly, instead of picking a random device, the requester tends to recruit the closest i_{th} device for the i_{th} recruitment action. This new recruitment technique leads to an increase in the successfully recruitment probability, which is because when the requester tries to recruit a close device, the signal quality will increase, and the path loss will decrease, which will increase the successful recruitment probability and the recruitment rate.

Capturing and handling EEDs failure is also an important aspect in a more realistic contexts. Thus, we will mutate our ACTMC and EDTMC in order to add the failure event, such that if any EED failed, the requester would recruit a new one to replace it, as illustrated in Figure 14(b). We will also track the number of EEDs that fail while executing the assigned job, and the probability that the task is successfully computed. Note that single job execution time is not going to be the same in all cases. When a device executes a job that takes more time than that executed on another device, the former device's rate of failure tends to be higher than that of the latter, since it operates for more time. Consequently, the job failure

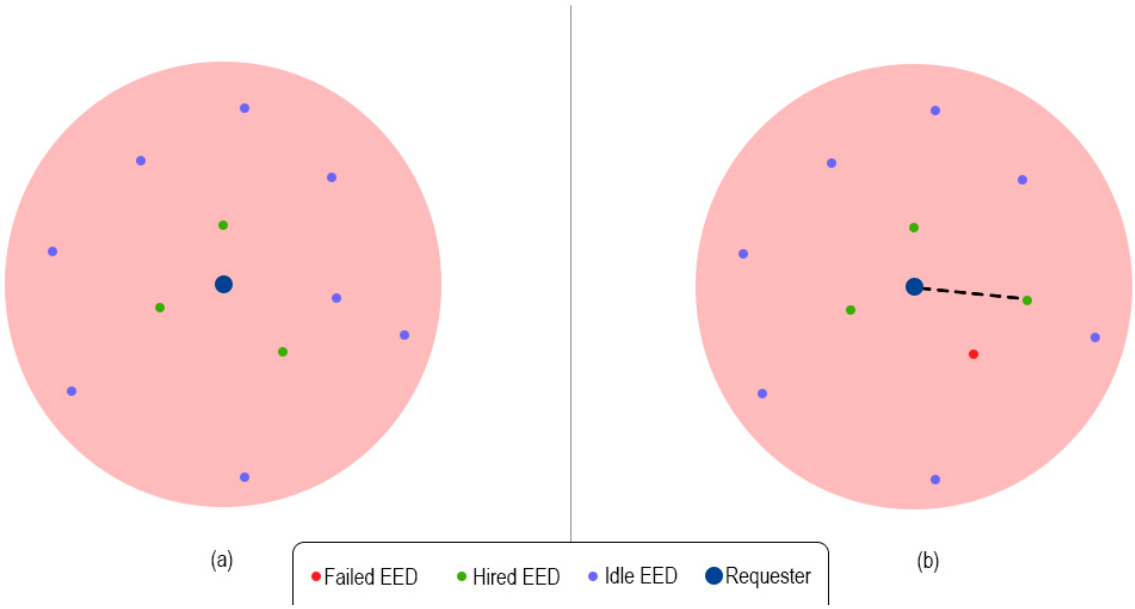

Figure 14: Advanced System Model, where (a) the requester will recruit the closest EEDs, and (b) if one of those EEDs fail, it will go for the closest idle one.

rate is directly related to the time that the job slice takes to be executed. Based on that, let $\gamma = \mu_f / l$ be the failure rate, where l represents the system reliability parameter, which means that, on average, an EED fails l times less than the rate of the task being successfully executed. For n task slices, the failure rate for each device i is $\gamma_i = \gamma / n$.

4.2 Distance-based Successful Recruitment Probability

As the previous section explained, new EEDs recruitment will be done by selecting the closest device to that requester. Let $\mathbf{R} = \{R_{(0)}, R_{(1)}, R_{(2)}, ..., R_{(k)}, ..., R_{(n)}\}$ be the sorted distance vector of all the LoS EEDs, where k represents the rank of the EED in the sorted vector, such that $R_{(0)} = \mathbf{min}\{\mathbf{R}\}$ and $R_{(n)} = \mathbf{max}\{\mathbf{R}\}$. The following Lemma represents the new successful recruitment probability when recruiting a new device based on its rank:

Lemma 3. *The spatially averaged successful recruitment probability via mmWave D2D communications for a distance-based selected LoS worker out of Φ_w is given Eq. 24*

$$p_{s_{(k)}} = \int_0^{R_L} \sum_{n=1}^{N_L} \binom{N_L}{n} (-1)^{n+1} e^{M_n(\xi)\sigma^2 - W_n(\xi) - Z_n(\xi)} f_{(k)}(r) dr \tag{24}$$

$M_n(\xi) = -\frac{\eta_L n r_0^{\alpha_L} \xi}{C_L M_r M_w}$, $W_n(\xi)$, $Z_n(\xi)$ are given in Eq. 6, $f(x) = 2r/R_L^2$, $F(x) = r^2/R_L^2$ and $V = \pi \nu_w R_L^2$, and $f_{(k)}(x)$ is given by Eq. 25.

$$f_{(k)}(x) = \frac{V^k e^{-V} f(x) F(x)^{k-1}}{(k-1)!} e^{-V[F(x)-1]} \tag{25}$$

Proof. The PDF of an ordered EEDs based on their distance is given Eq 26, where, $f_X(x)$ and $F_X(x)$ are the PDF and the CDF of the distance, n is a Poisson random variable with mean $\nu_w \pi R_L^2$, which represents the average number of LoS EEDs, k is the order of the desired EED.

$$f_{X_{(k)}}(x) = \frac{n!}{(k-1)!(n-k)!} f_X(x) [F_X(x)]^{k-1} [1 - F_X(x)]^{n-k} \tag{26}$$

Since n follows the Poisson distribution, this can be written as Eq 27, where $V = \pi \nu_w R_L^2$.

$$\begin{aligned} f_{X_{(k)}}(x|n) &= \mathbb{E}[f_{X_{(k)}}(x|n)] \\ &= \sum_{i=0}^{\infty} \frac{V^i f_r(x|i)}{i!} e^{-V} \end{aligned} \tag{27}$$

By embedding the order statistic PDF mentioned in Eq. 26 in Eq. 27, $f_{X_{(k)}}(x|n)$ can be given by Eq. 28.

$$\begin{aligned} f_{X_{(k)}}(x|n) &= \sum_{i=0}^{\infty} \frac{V^i \frac{i!}{(k-1)!(i-k)!} f_X(x) [F_X(x)]^{k-1} [1 - F_X(x)]^{i-k}}{i!} e^{-V} \\ &= \frac{V^k e^{-V} f(x) F(X)^{k-1}}{(k-1)!} \sum_{i=0}^{\infty} \frac{V^{i-k} [1 - F(x)]^{i-k}}{(i-k)!} \end{aligned} \tag{28}$$

Let $a = i - k$. Thus, when $i = 0$, $a = -k$. Consequently:

$$\sum_{i=0}^{\infty} \frac{V^{i-k}[1-F(x)]^{i-k}}{(i-k)!} = \sum_{a=0}^{\infty} \frac{V^{a}[1-F(x)]^{a}}{(a)!} + \sum_{a=-k}^{-1} \frac{V^{a}[1-F(x)]^{a}}{(a)!} \tag{29}$$

This part follows Taylor Series expansions of exponential functions. Accordingly, the final form of $f_{X_{(k)}}(x|n)$ is given by Eq. 30. Note that the remaining proof has already been provided in the proof of Lemma 1.

$$f_{X_{(k)}}(x|n) = \frac{V^{k}e^{-V}f(x)F(X)^{k-1}}{(k-1)!}e^{-V[F(x)-1]} \tag{30}$$

□

The aforementioned recruitment probability (given by Eq. 24) requires changing the ACTMC to be a level dependent ACTMC, which means that any recruited device has its own recruitment rate. This recruitment rate is represented as $\lambda_{h_k} = p_{s_k}/\tau_c$, where p_{s_k} is the successful recruitment rate of the k_{th} closest EED to the requester. The change in the ACTMC model only affects matrix $\mathbf{K}$ in $\mathbf{Q}$ and $\mathcal{K}$ in $\mathbf{P}$, which are the matrices responsible for tracking any new EED recruitment transition. The new matrices are given as follows:

$$\mathbf{K}_m = \begin{array}{c} \\ X_C \\ 0 \\ 1 \\ 2 \\ \vdots \\ n-m-1 \\ (n-m) \end{array} \begin{array}{c} \begin{array}{ccccccc} \quad & 0 & 1 & 2 & 3 & \cdots & n-m-1 \quad\quad & n-m \end{array} \\ \begin{bmatrix} -\lambda_{h_1} & \lambda_{h_1} & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\lambda_{h_2}-\mu_f & \lambda_{h_2} & 0 & \ddots & 0 & 0 \\ 0 & 0 & -\lambda_{h_3}-2\mu_f & \lambda_{h_3} & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & -\lambda_{h_{n-m}}-(n-m-1)\mu_f & \lambda_{h_{n-m}} \\ 0 & 0 & 0 & 0 & 0 & 0 & -(n-m)\mu_f \end{bmatrix} \end{array}$$

and

$$\mathcal{K}_m = \begin{array}{c} \begin{matrix} X_C & 0 & 1 & 2 & 3 & \cdots & (n-m-1) & (n-m) \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ (n-m-1) \\ (n-m) \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{\lambda_{h_2}}{\lambda_{h_2}+\mu_f} & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \frac{\lambda_{h_3}}{\lambda_{h_3}+2\mu_f} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{\lambda_{h_{n-m}}}{\lambda_{n-m}+(n-m-1)\mu_f} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \end{array}$$

4.3 Modeling Failure

As explained in the system model, $\gamma = \mu_f / l$ is defined as the failure rate, where l represents the system reliability parameter. For n task slices, the failure rate of each recruited EED i is defined as $\gamma_i = \gamma / n$. To reflect those changes on the proposed model, the new matrices $\mathbf{K_m}$ and $\mathbf{H_m}$ are represented as follows:

$$\mathbf{K}_m = \begin{array}{c} \begin{matrix} X_C & 0 & 1 & 2 & \cdots & (m_L-1) & (m_L) \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ \vdots \\ (m_L-1) \\ (m_L) \end{matrix} \begin{bmatrix} -\lambda_{h_1} & \lambda_{h_1} & 0 & \cdots & 0 & 0 \\ 0 & K_{F_1} & K_{H_2} & \cdots & 0 & 0 \\ 0 & 0 & K_{F_2} & K_{H_3} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & K_{F_{m_L-1}} & K_{H_{m_L}} \\ 0 & 0 & 0 & \cdots & 0 & K_{F_{m_L}} \end{bmatrix} \end{array}$$

and

$$\mathbf{H}_{m,m+1}= \begin{array}{c} X_C \\ 0 \\ 1 \\ 2 \\ \vdots \\ (m_L-1) \\ (m_L) \end{array} \begin{array}{c} \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & (n-m-1) & (n-m) \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & H_{F_1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & H_{F_2} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & H_{F_{m_L-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & H_{F_{m_L}} \end{bmatrix} \end{array}$$

where

$$\mathbf{K}_{\mathbf{F}(k,k)}(i,j) = \begin{cases} (X_C - X_F - X_D)\gamma_i, & i = j+1 \ \ \& \ \ i < m_L \\ -(X_C - X_F - X_D)(\gamma_i + \mu_f) - \lambda_{h_i}, & i = j \\ 0, & \text{otherwise} \end{cases}.$$

$$\mathbf{K}_{\mathbf{H}(k,k)}(i,j) = \begin{cases} \lambda_h, & i = j+1 \ \ \& \ \ (X_C - X_F - X_D) < m_L \\ 0, & \text{otherwise} \end{cases}.$$

$$\mathbf{H}_{\mathbf{F}(k,k)}(i,j) = \begin{cases} (X_C - X_F - X_D)\mu_f, & i = j \ \ \& \ \ (X_C - X_F - X_D) < m_L \\ 0, & \text{otherwise} \end{cases}.$$

Considering that $\beta_{(X_F,X_C,X_D)} = (X_C - X_F - X_D)(\gamma_{X_C} + \mu_f) + \lambda_{h_{X_C}}$, the matrices $\mathcal{K}_{\mathbf{m}}$ and $\mathcal{H}_{\mathbf{m,m+1}}$ can be described as follows:

$$\mathcal{K}_m = \begin{array}{c} X_C \\ 0 \\ 1 \\ 2 \\ \vdots \\ (m_L-1) \\ (m_L) \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & (m_L-1) & (m_L) \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & \mathcal{K}_{F_1} & \mathcal{K}_{H_2} & \cdots & 0 & 0 \\ 0 & 0 & \mathcal{K}_{F_2} & \mathcal{K}_{H_3} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathcal{K}_{F_{m_L-1}} & \mathcal{K}_{H_{m_L}} \\ 0 & 0 & 0 & \cdots & 0 & \mathcal{K}_{F_{m_L}} \end{bmatrix} \end{array}$$

and

$$\mathcal{H}_{m,m+1} = \begin{array}{c} X_C \\ 0 \\ 1 \\ 2 \\ \vdots \\ (m_L-1) \\ (m_L) \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & \cdots & (n-m-1) & (n-m) \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \mathcal{H}_{F_1} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \mathcal{H}_{F_2} & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \mathcal{H}_{F_{m_L-1}} & 0 \\ 0 & 0 & 0 & \cdots & 0 & \mathcal{H}_{F_{m_L}} \end{bmatrix} \end{array}$$

where

$$\mathcal{K}_{F(k,k)}(i,j) = \begin{cases} (X_C - X_F - X_D)\gamma_i/\beta, & i = j+1 \ \ \& \ \ i < m_L \\ \\ 0, & \text{otherwise} \end{cases}.$$

$$\mathcal{K}_{H(k,k)}(i,j) = \begin{cases} \lambda_h/\beta, & i = j+1 \ \ \& \ \ (X_C - X_F - X_D) < m_L \\ \\ 0, & \text{otherwise} \end{cases}.$$

$$\mathcal{H}_{F(k,k)}(i,j) = \begin{cases} (X_C - X_F - X_D)\mu_f/\beta, & i = j \ \ \& \ \ (X_C - X_F - X_D) < m_L \\ \\ 0, & \text{otherwise} \end{cases} .$$

Let ρ_t be the probability that all the n task slices are successfully executed at the recruited EEDs, which represents the probability that the ACTMC model absorbs in one of the success states. This probability is calculated iteratively and can be given by

$$\rho_{\mathbf{t_i}} = \begin{cases} 1, & \text{j is success state} \\ \\ 0, & \text{j is failure state} \\ \\ \sum_j P(i,j) * \rho_{t_j} & \text{otherwise} \end{cases} . \tag{24}$$

4.4 System Analysis and Simulation Results

In this section, we provide a detailed discussion and analysis of the numerical results. Then, like what we did in the previous chapter. We validate the proposed spatiotemporal model compared to simulation results obtained from Monte Carlo simulations. In addition, the impact of different system parameters on the task response delay is demonstrated. Unless otherwise specified, the list of network parameters used in this section is summarized in Table 2.

4.4.1 Mathematical Model Validation

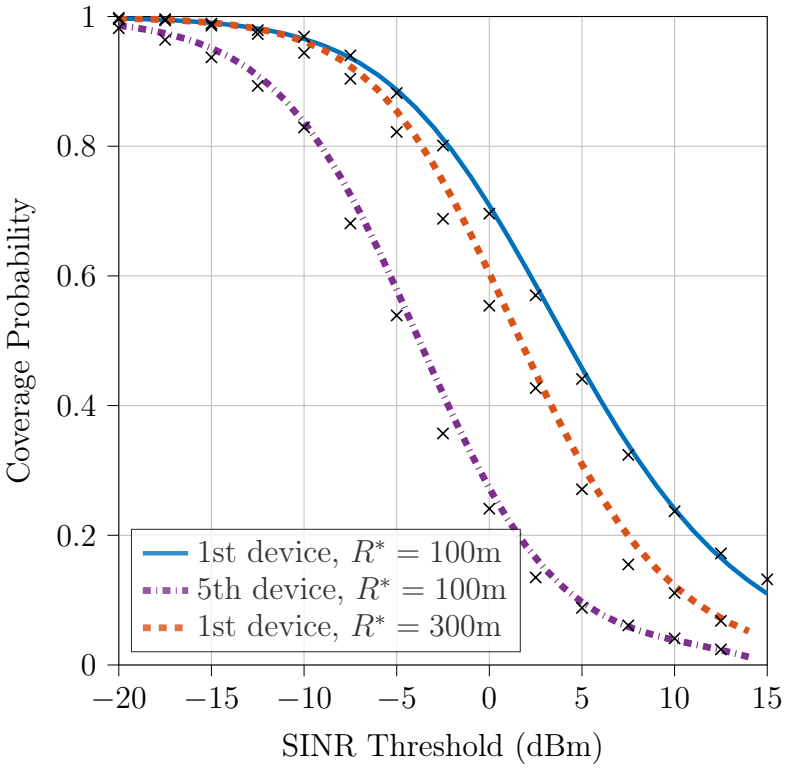

Figure 15: D2D recruitment success probability vs SINR threshold ξ over distance ordered EEDs

Figure 15 shows the advanced successful recruitment probability p_{x_r} as a function of the desired SINR threshold ξ and the selected device rank k. The analysis is done using different R_L values that enclose LoS devices, and different device rank k values. The close match illustrated between the simulation and the proposed framework validates Lemma 3. The figure shows that the new successful recruitment probability p_{x_r} gives better results than the one in Lemma 1 for the nearby devices. This is due to the small distance between the requester and its closest device, which results better signal strength due to the decrease in path loss. The results also show that the lower the device rank, the better its coverage probability. This can be attributed to the difference in the distance between the devices and the requester. In addition, as depicted in the Figure, a larger radius R_L increases the

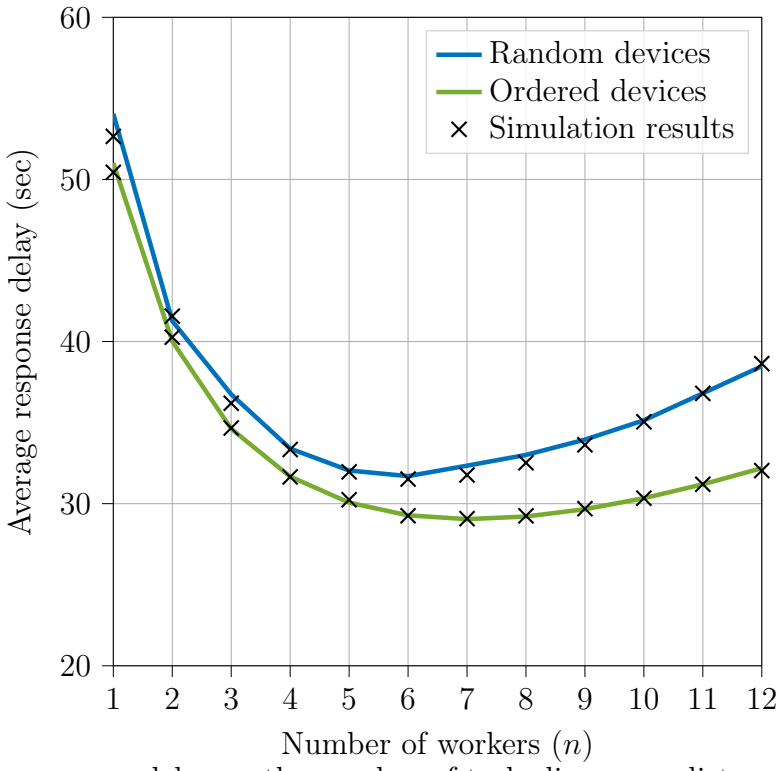

Figure 16: Task response delay vs the number of task slices over distance ordered EEDs

communication cost, due to the increase in the interference from other LoS requesters.

Figure 16 demonstrates the average task response delay the random and ordered recruitment approaches over varying number of recruited workers. The close match rendered between the simulation and the analytical model validates the model. On average, the task response delay keeps decreasing until it reaches a point where the delay of recruiting a new EED exceeds the time spared by the requester if the task involves more slices. The Figure also shows that recruiting EEDs based on their distance to the requester reduces the response delay by up to 20% , compared to the random recruitment approach. This can be attributed to the fact that the decrease in the recruiting time resulting from the increase in the coverage probability of the closer devices. This also can occasionally increase the number of optimal task slices, due to the decrease in the recruiting time.

Next, we validate the average task response delay after including the failure model, as well as the probability that the n task slices will be executed successfully. We validate the failure model in both the random and ordered recruitment approaches.

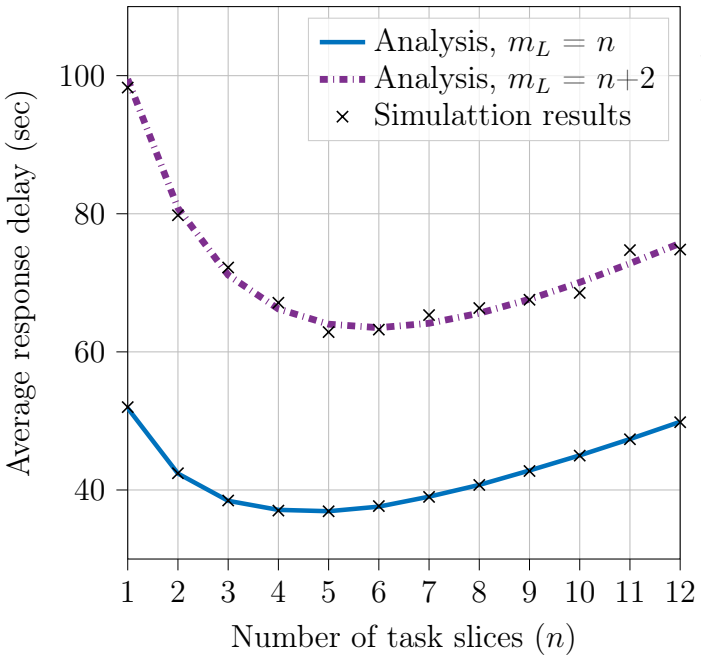

Figure 17: Average response delay over two different m_L values

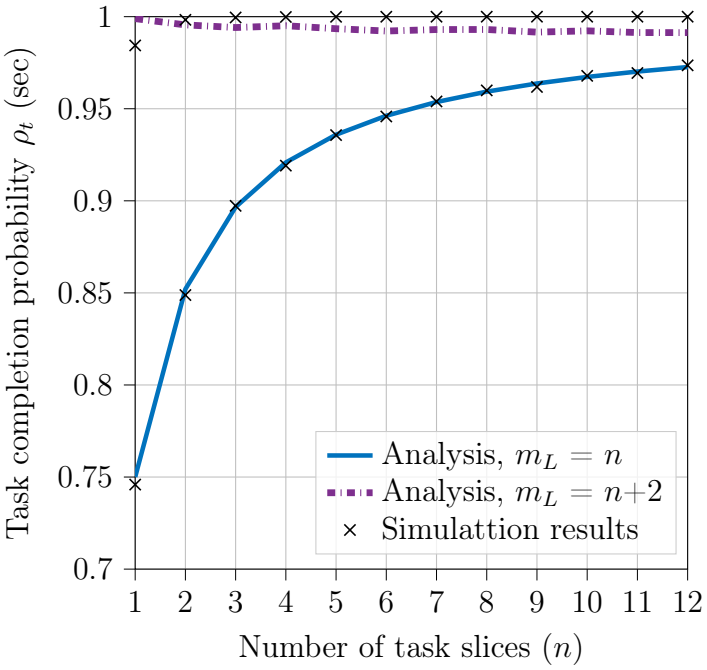

Figure 18: Task successful completion probability over two different m_L values

Figure 17 and Figure 18 show the average task response delay after including the failure model using the random recruiting model. The results are obtained using two different values of the maximum numbers of available EEDs, m_L. The average task response delay when $m_L = n$ is similar to the results where the failure model is not applied. This is because when an EED fails at any time, the requester is not be able to recruit any new one. This implies that the only way to execute all the n task slices successfully, none of the recruited EEDs should fail. This triggers a low success probability, as depicted in Figure 18, where the lower the maximum number of available devices, the lower the probability. As the number of task slices increases, the probability of task successful completion increases, due to the fact that the slice size is inversely proportional to the failure rate. Having a higher value of m_L increase both the average task response delay and the success probability. This is because when an EED fails, the requester can immediately recruit a new one and offload the failed slice to it.

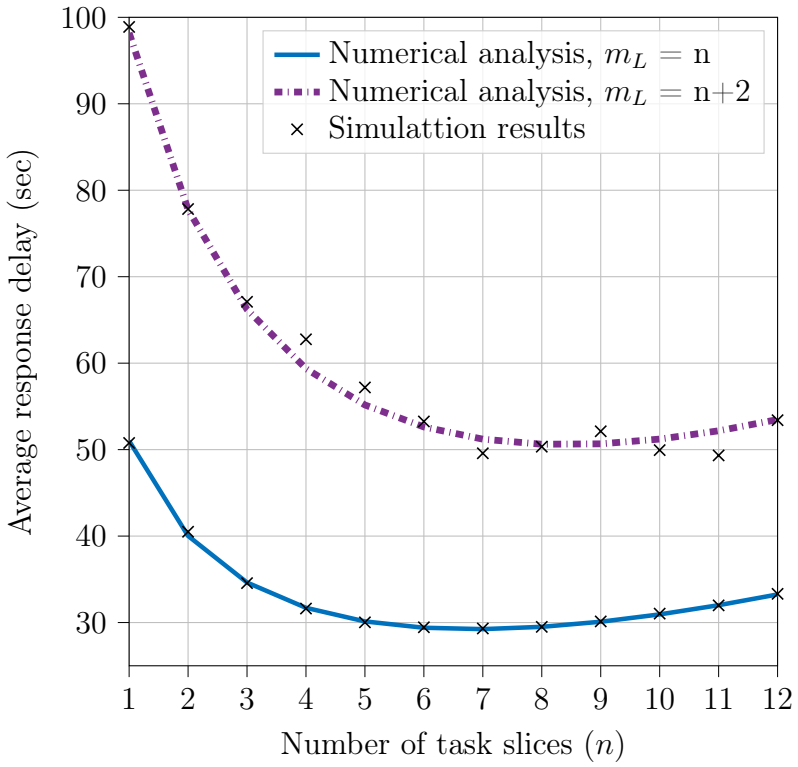

Figure 19: Average response delay over two different m_L values and distance ordered EEDs

Figure 19 the average response delay of the failure and recruitment model, where EEDs are prone to failure, and the requester starts recruiting the nearest EEDs. In this model, when an EED fails, the requester hires the closest non-hired device to the requester. For example, if the requester has five recruited EEDs and one of them fails, the requester recruits the sixth closest device to it. This also leads to a faster task response delay compared to the model where location-awareness is not considered. This is since in the latter model, a random device is recruited each time an EED fails, while the former model recruits the nearest device. Similar to Figure 17, when m_L is equal to the number of tasks slices, the average task response delay is the same as the one without failure in Figure 16, whereas when m_L increases, the average delay increases, and the task execution success probability increases too. Note that the yielded task success probability for this model is the same as the one introduced in Figure 18.

4.4.2 Varying System Parameters

After validating the model, including both extensions, we discuss the effect of those extensions on the average task response delay. Then, starting with the first extension, we show the average response delay in hiring ordered EEDs, using multiple system parameters. After that, we will compare the results with the random recruitment technique.

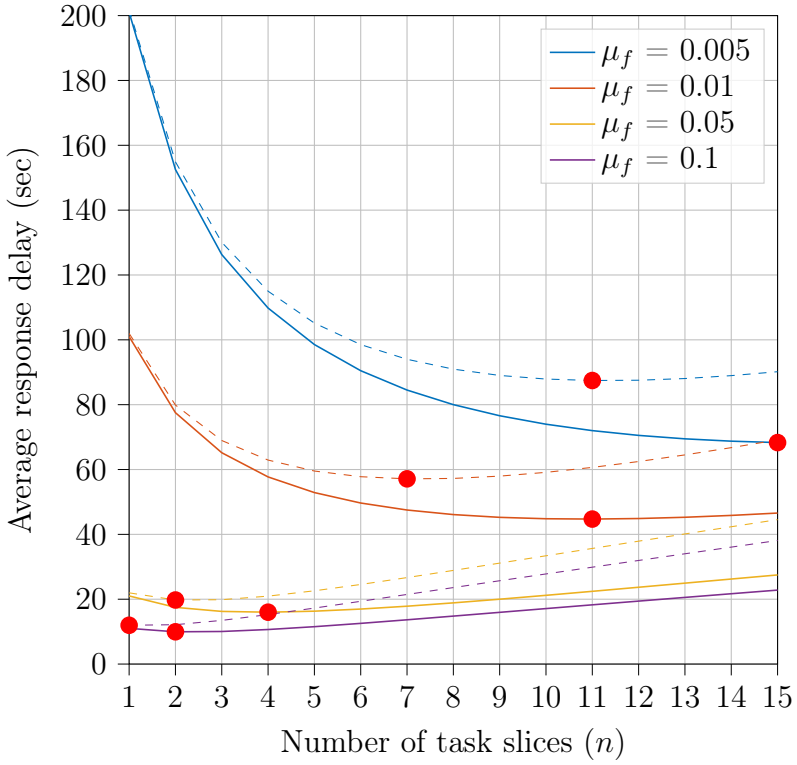

Figure 20: Average response delay of random versus ordered recruited devices under varying μ_f, where the dashed lines represent the random recruiting, and the strong lines represent the ordered recruiting

Figure 20 shows the average response delay using different task finishing rate μ_f values, where the dashed lines represent the delay after recruiting random EEDs, and the solid lines represent the average delay after recruiting ordered devices. The values of μ_f represent different task loads, where the small numbers reflect a high task load, and vice versa. In all results, the ordered EEDs recruiting model yields a smaller average task delay and higher value of optimal number of task slices n, which are the red dots in the Figure. This can be

attributed to the significantly reduced recruitment time rendered by the ordered recruitment model due to recruiting the nearest devices. This reflects on both the average task response delay and the optimal number of recruited workers.

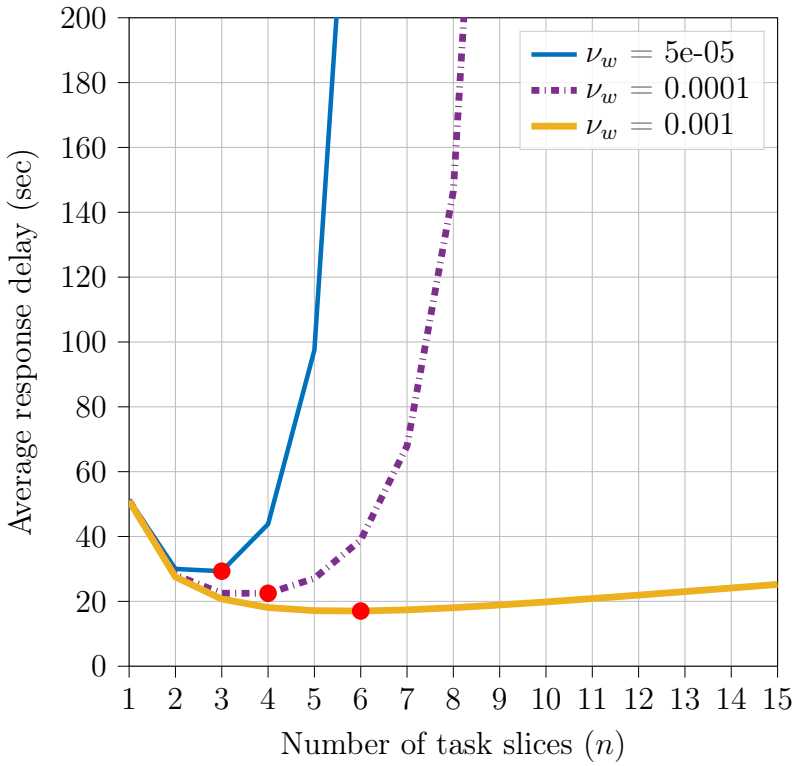

Figure 21: Average Response delay of the ordered recruitment model over varying ν_w

We conduct another experiment to evaluate the impact of the workers intensity values ν_w on the response delay, Figure 21 shows the average task response delay over varying values of ν_w. The results shows that the lower the workers intensity, the higher the task response delay and the lower the optimal number of task slices. This is due to the fact that the lower the value of ν_w the smaller the number of available LoS devices. As depicted in the Figure, after reaching a certain amount of task slices, the average response delay starts to increase indefinitely when ν_w has smaller values. This is due to the low probability of having another LoS device under low intensity.

4.4.3 Failure Impact on Delay

In this experiment, we evaluate the impact of embedding the failure model on the response delay under varying system parameters. Unless otherwise specified, the maximum number of recruited EEDs, m_L, is the number of task slices $n + 2$, and the reliability rank $l = 2$.

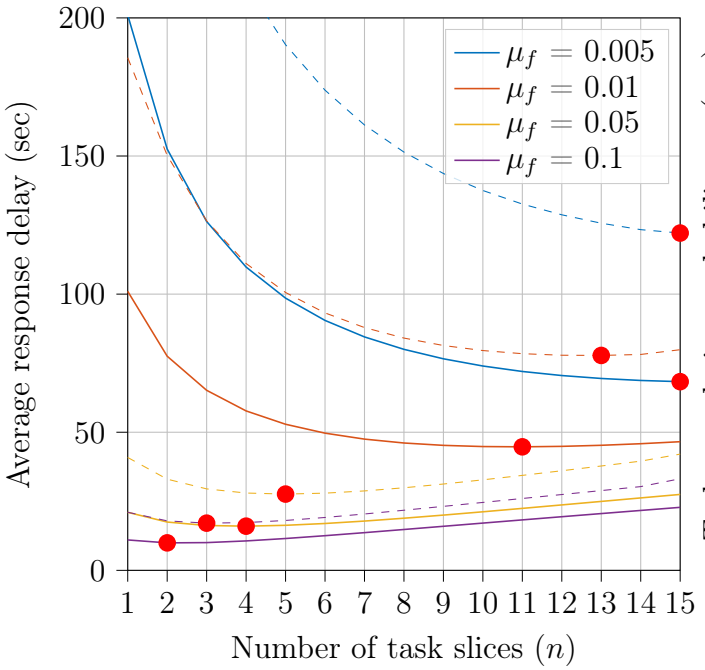

Figure 22: Average task response delay over varying task finishing rate μ_f values

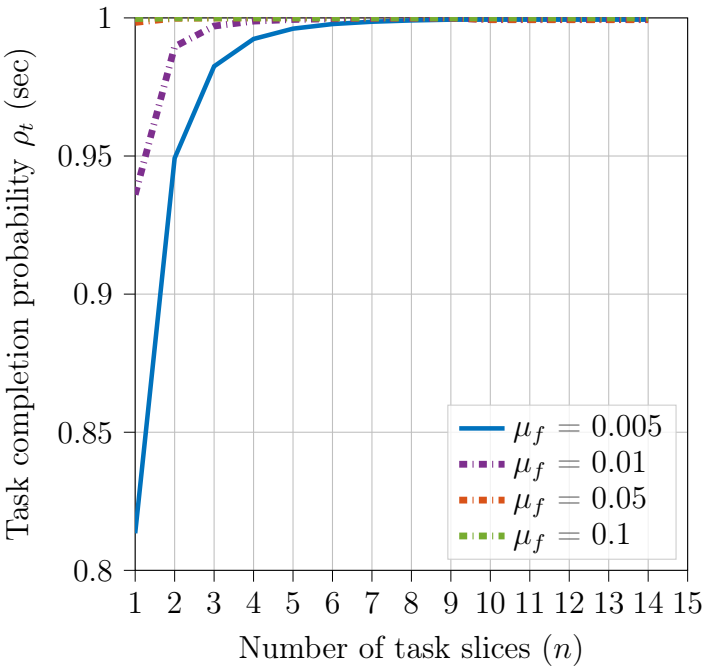

Figure 23: Task successful completion over varying task finishing rate μ_f

Figures 22 and Figure 23 show the task response delay and successful execution probability over different μ_f values. In Figure 22, the dashed lines represent the model that accounts for failure, and the solid lines represent the model that does not consider failure. At lower task finish rates μ_f values (i.e., higher task finishing time), the requester tends to split the task more till it reaches the optimal number of task slices and optimal delay. This results from the fact that the fail rate γ value is linked to the μ_f value, and thus more slices leads to lower γ_i values and less failing events. This persists until the requester reaches a limit where splitting the task more requires recruiting more devices, which adds extra recruitment time. Figure 23 depicts the success probability over varying finish rate. As shown in the Figure, the success probability decreases while having more computing requirements. This is since

the value of of γ_i depends on the task finishing rate, and thus, a low μ_f with a low number of task slices increases the probability that the recruited devices fail. Based on the used system reliability value, after splitting the task into more jobs, the success probability converges to be almost one due to having small task portions and small device failure probability.

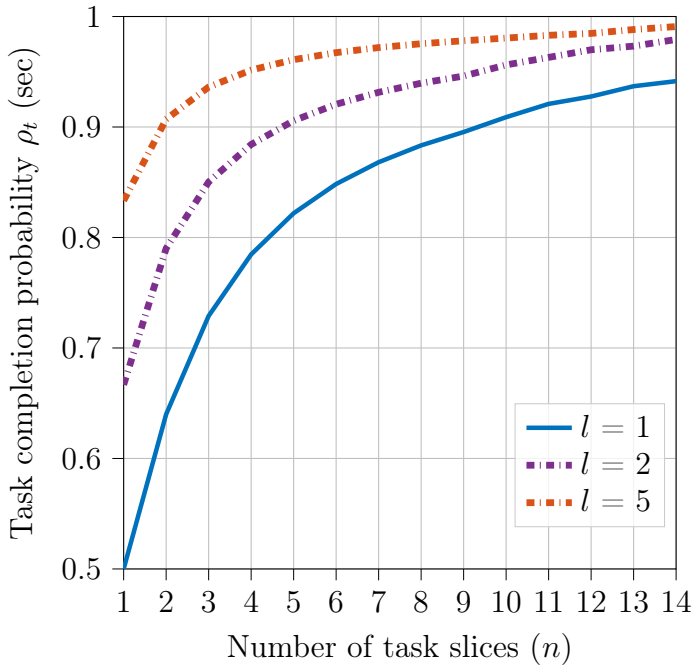

Figure 24: Task completion probability over different EEDs reliability l values

Figure 24 depicts the task execution success probability over different reliability rank values l. The results show that the lower the reliability rank the lower the success probability. This can be attributed to the inverse relation between l and γ, and thus lower values of k values leads to higher γ, which lowers the success probability.

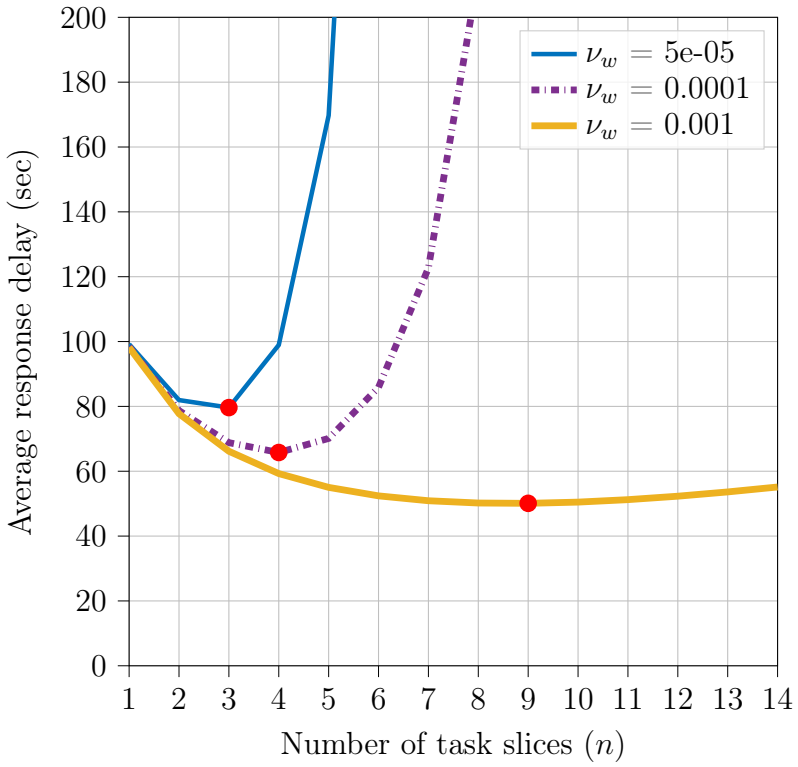

Figure 25: Average response delay over different ν_w values, by hiring ordered devices with the failure model

Finally, Figure 25 demonstrates the average response delay over different workers intensity ν_w values. As illustrated in Figure 21, having lower workers intensity leads to having fewer optimal task slices and higher task response delay. Also, having small ν_w leads to an increase in the average time indefinitely due to the lack of enough LoS devices to handle those available slices.

4.5 Conclusion and Discussion

In this chapter, we have introduced our advanced system model, where we investigated the affect of having the distance information while recruiting new EEDs. This decreases the average task response delay due to decreasing the successful recruitment probability. Also, we have investigated the prone to failure EEDs in the introduced ACTMC; then we showed how the adjustments made to the ACTMC model to embed the failure model and used it

to calculate the task success execution probability. We have validated both new models by comparing them with Monte Carlo simulations. We then evaluated the model using a variety of system parameters. The results revealed that in some cases, the needs to be should be aware of not having enough devices due to the low worker intensity in the LoS range. Also, the non-reliable systems were considered, where the failure rate was equal to the finishing rate. In that case, the task execution success probability tends to be low. Thus, the requester might need to divide the task into more slices and refrain from applying the optimal number of slices in order to increase the task success probability.

Chapter 5
Conclusions

5.1 Summary and Conclusion

This work presents a novel spatiotemporal framework that characterizes the task response delay in extreme edge computing paradigms. The developed framework accounts for interwoven communication and computation delays. We have used an Absorbing Continuous-Time Markov Chain (ACTMC) to track workers recruitment and task offloading to Extreme Edge Devices (EEDs). The ACTMC model considers the parallel execution of tasks at the recruited devices. The offloading rate is obtained by utilizing stochastic geometry analysis to obtain the successful recruitment probability, which is directly related to the number of attempts needed to offload a task successfully. We have also proposed two ways of recruiting new EEDs; by either choosing a random EED each time or choosing that EEDs based on its distance order. We have discussed and shown the positive impact of the ordered EED recruitment method on the task response delay and the optimal number of task slices under varying system parameters that reflect different scenarios. In addition, we have introduced a failure tracking model to keep track, of the recruited EEDs and the rate at which they tend to fail. In case of failure, the requester recruits a new EED based on the desired EED selection method. We validated the proposed failure model and conducted extensive evaluations under varying system parameters.

5.2 Recommendations and Future Work

Numerical results validate the analysis and reveal the existence of an optimal number of task slices that minimizes the underlying task response time. Operating below the optimal number leads to underutilized edge computational resources and prolongs the task response delay. Exceeding the optimal number leads to a dominating communication delay that prolongs the response delay. Note that the optimal number of recruited EEDs significantly

differs under varying network parameters. When incorporating the failure model into the spatiotemporal analysis, the maximum number of devices that a requester can recruit is an important aspect that plays a significant role in rendering a high successful task execution probability and a reduced average task response delay.

In the future, we plan to considering a system with heterogeneous EEDs, where each EED has its own available computational capability. Given such heterogeneity, we plan on developing a spatiotemporal analysis model that recruits EEDs based on their computational capability. In addition, scenarios where multiple requesters contend for the available resources will be modeled.

References

[1] J. Steward, "Cisco visual networking index: Global mobile data traffic forecast update," 2019. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[3] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of things realization," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.

[4] J. Steward, "21 Internet of things statistics, facts & trends for 2021," 2021. [Online]. Available: https://findstack.com/Internet-of-things-statistics/

[5] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[8] L. Peterson, T. Anderson, S. Katti, N. McKeown, G. Parulkar, J. Rexford, M. Satyanarayanan, O. Sunay, and A. Vahdat, "Democratizing the network edge," *SIGCOMM Comput. Commun. Rev.*, vol. 49, no. 2, p. 31–36, may 2019. [Online]. Available: https://doi.org/10.1145/3336937.3336942

[9] R. Olaniyan, O. Fadahunsi, M. Maheswaran, and M. F. Zhani, "Opportunistic edge computing: Concepts, opportunities and research challenges," *Future Generation Computer Systems*, vol. 89, pp. 633–645, 2018.

[10] Y. Sahni, J. Cao, S. Zhang, and L. Yang, "Edge mesh: A new paradigm to enable distributed intelligence in Internet of things," *IEEE Access*, vol. 5, pp. 16 441–16 458, 2017.

[11] W. Zhang, H. Flores, and P. Hui, "Towards collaborative multi-device computing," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2018, pp. 22–27.

[12] J. Portilla, G. Mujica, J.-S. Lee, and T. Riesgo, "The extreme edge at the bottom of the Internet of things: A review," *IEEE Sensors Journal*, vol. 19, no. 9, pp. 3179–3190, 2019.

[13] A. Jain and P. Singhal, "Fog computing: Driving force behind the emergence of edge computing," in *2016 International Conference System Modeling Advancement in Research Trends (SMART)*, 2016, pp. 294–297.

[14] K. Ashton, "That Internet of things thing," vol. 22, no. 7, 2009.

[15] M. Hartmann, U. Hashmi, and A. Imran, "Edge computing in smart health care systems: Review, challenges, and research directions," *Transactions on Emerging Telecommunications Technologies*, vol. 33, 03 2022.

[16] W. Yu and J. Choi, "Human identification in health care systems using mobile edge computing," *Transactions on Emerging Telecommunications Technologies*, vol. 31, p. e4031, 12 2020.

[17] H. T. Mouftah, M. Erol-Kantarci, and S. Sorour, *Connected and Autonomous Vehicles in Smart Cities*. CRC Press, 2020.

[18] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-Cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. New York, NY, USA: Association for Computing Machinery, 2019, p. 88–100.

[19] V. H. M. Donald, "Advanced mobile phone service: The cellular concept," *The Bell System Technical Journal*, vol. 58, no. 1, pp. 15–41, 1979.

[20] J. G. Andrews, F. Baccelli, and R. K. Ganti, "A tractable approach to coverage and rate in cellular networks," *IEEE Transactions on Communications*, vol. 59, no. 11, pp. 3122–3134, 2011.

[21] H. ElSawy, A. Sultan-Salem, M.-S. Alouini, and M. Z. Win, "Modeling and analysis of cellular networks using stochastic geometry: A tutorial," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 167–203, 2017.

[22] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 7, pp. 1029–1046, 2009.

[23] H. ElSawy, E. Hossain, and M. Haenggi, "Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 996–1019, 2013.

[24] A. S. Alfa, *Applied Discrete-Time Queues*, 2nd ed. Springer Publishing Company, Incorporated, 2015.

[25] M. K. N. T.-K. S. Wai-Ki Ching, Ximin Huang, *Markov Chains, Models, Algorithms and Applications*. Springer New York, NY, 2013.

[26] A. S. Alfa, *Applied Discrete-Time Queues*. Springer New York, NY, 2015.

[27] S. Bagchi, M.-B. Siddiqui, P. Wood, and H. Zhang, "Dependability in edge computing," *Commun. ACM*, vol. 63, no. 1, p. 58–66, Dec 2019. [Online]. Available: https://doi.org/10.1145/3362068

[28] R. Birke, I. Giurgiu, L. Y. Chen, D. Wiesmann, and T. Engbersen, "Failure analysis of virtual and physical machines: Patterns, causes and characteristics," in *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2014, pp. 1–12.

[29] S.-W. Ko, K. Han, and K. Huang, "Wireless networks for mobile edge computing: Spatial modeling and latency analysis," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5225–5240, 2018.

[30] H.-S. Lee and J.-W. Lee, "Task offloading in heterogeneous mobile cloud computing: Modeling, analysis, and cloudlet deployment," *IEEE Access*, vol. 6, pp. 14 908–14 925, 2018.

[31] H. Ko, J. Lee, and S. Pack, "Spatial and temporal computation offloading decision algorithm in edge cloud-enabled heterogeneous networks," *IEEE Access*, vol. 6, pp. 18 920–18 932, 2018.

[32] A. I. Akin, H. Mirghasemi, and L. Vandendorpe, "Swipt-based mobile edge computing systems: A stochastic geometry perspective," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–7.

[33] L. Lin, W. Zhou, and Z. Zhao, "Analytical modeling of NOMA-based mobile edge computing systems with randomly located users," *IEEE Communications Letters*, vol. 24, no. 12, pp. 2965–2968, 2020.

[34] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 1451–1455.

[35] Y. Chen, J. Liu, and P. Siano, "SGedge: Stochastic geometry-based model for multi-access edge computing in wireless sensor networks," *IEEE Access*, vol. 9, pp. 111 238–111 248, 2021.

[36] Y. Zhong, T. Q. S. Quek, and X. Ge, "Heterogeneous cellular networks with spatio-temporal traffic: Delay analysis and scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1373–1386, 2017.

[37] M. Gharbieh, H. ElSawy, H.-C. Yang, A. Bader, and M.-S. Alouini, "Spatiotemporal model for uplink iot traffic: Scheduling and random access paradox," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8357–8372, 2018.

[38] J. M. F. Benkhelifa, H. ElSawy and M. Alouini, "Recycling Cellular Energy for Self-Sustainable IoT Networks: A Spatiotemporal Study," 1 2020. [Online]. Available: https://www.techrxiv.org/articles/preprint/Recycling_Cellular_Energy_for_Self-Sustainable_IoT_Networks_A_Spatiotemporal_Study/11604147

[39] H. H. Yang and T. Q. S. Quek, "Spatio-temporal analysis for SINR coverage in small cell networks," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5520–5531, 2019.

[40] M. Emara, H. ElSawy, M. C. Filippou, and G. Bauch, "Spatiotemporal dependable task execution services in MEC-enabled wireless systems," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 211–215, 2021.

[41] Y. Gu, Y. Yao, C. Li, B. Xia, D. Xu, and C. Zhang, "Modeling and analysis of stochastic mobile-edge computing wireless networks," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 051–14 065, 2021.

[42] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2196–2211, 2015.

[43] N. Deng, M. Haenggi, and Y. Sun, "Millimeter-wave device-to-device networks with heterogeneous antenna arrays," *IEEE Transactions on Communications*, vol. 66, no. 9, pp. 4271–4285, 2018.

[44] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[45] H. Alzer, "On some inequalities for the incomplete gamma function," *Math. Comput.*, vol. 66, no. 218, p. 771–778, apr 1997. [Online]. Available: https://doi.org/10.1090/S0025-5718-97-00814-4

[46] F. Baccelli and B. Blaszczyszyn, *Stochastic Geometry and Wireless Networks*, 2009, vol. I—Theory. Delft, The Netherlands.