UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING
DEGREE PROGRAMME IN WIRELESS COMMUNICATIONS ENGINEERING

MASTER'S THESIS

TOWARDS RELIABLE AND LOW-LATENCY VEHICULAR EDGE COMPUTING NETWORKS

Author	Sadeep Batewela
Supervisor	Associate Prof. Mehdi Bennis
Second Examiner	Dr. Himal A. Suraweera
Technical Advisor	Chen-Feng Liu

July 2019

Batewela S. (2019) Towards Reliable and Low-Latency Vehicular Edge Computing Networks. University of Oulu, Faculty of Information Technology and Electrical Engineering, Degree Programme in Wireless Communications Engineering, 40 p.

ABSTRACT

To enable autonomous driving in intelligent transportation systems, vehicular communication is one of the promising approaches to ensure safe, efficient, and comfortable travel. However, to this end, there is a huge amount of application data that needs to be exchanged and processed which makes satisfying the critical requirement in vehicular communication, i.e., low latency and ultra-reliability, challenging. In particular, the processing is executed at the vehicle user equipment (VUE) locally. To alleviate the VUE's computation capability limitations, mobile edge computing (MEC), which pushes the computational and storage resources from the network core towards the edge, has been incorporated with vehicular communication recently. To ensure low latency and high reliability, jointly allocating resources for communication and computation is a challenging problem in highly dynamics and dense environments such as urban areas. Motivated by these critical issues, we aim to minimize the higher-order statistics of the end-to-end (E2E) delay while jointly allocating the communication and computation resources in a vehicular edge computing scenario. A novel risk-sensitive distributed learning algorithm is proposed with minimum knowledge and no information exchange among VUEs, where each VUE learns the best decision policy to achieve low latency and high reliability. Compared with the average-based approach, simulation results show that our proposed approach has the better network-wide standard deviation of E2E delay and comparable average E2E delay performance.

Keywords: 5G and beyond, URLLC, risk sensitive, mobile edge computing, vehicular networks.

TABLE OF CONTENTS

FOREWORD

This thesis is focused on improving the performance of low latency and high reliability vehicular edge computing and communication networks. I express my sincere gratitude to my supervisor and mentor Associate Prof. Mehdi Bennis for his assistance and guidance throughout the period of the research. I would like to extend my gratitude to my technical and immediate supervisor Chen-Feng Liu for his continuous supervision, constant encouragement and giving me great insights of research. I would also like to thank my supervisor Dr. Himal A. Suraweera from University of Peradeniya for his assistance in my research. I express my deep gratitude to Prof. Nandana Rajatheva for his valuable support and guidance through out this master program and also double degree coordinator Matti Isohookana for his support during the masters studies. I would like to thank all the lectures from University of Peradeniya for their contribution in making inaugural double degree masters program a success.

Finally, I thank my parents and my siblings for their immense support, love and kindness throughout this journey.

Oulu, 30th July, 2019

Sadeep Batewela

LIST OF ABBREVIATIONS AND SYMBOLS[1]

5G	Fifth Generation
CPU	Central Processing Unit
DL	Down-Link
E2E	End-to-End
EVT	Extreme Value Theory
GPS	Global Positioning Systems
IoT	Internet of Things
IT	Information Technology
KKT	Karush–Kuhn–Tucker
LOS	Line of Sight
MEC	Mobile Edge Computing
QoE	Quality of Experience
QoS	Quality of Service
QSI	Queue State Information
RRM	Radio Resource Management
RSU	Rode-Side Unit
URLLC	Ultra-Reliable and Low-Latency Communication
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad-Hoc Network
VUE	Vehicle User Equipment

1 INTRODUCTION

1.1 Background and Motivation

With the evolution of information technology (IT), we are moving towards a digitized society. Under this evolution, traditional transportation systems are being upgraded to intelligent transportation systems to mitigate the drawbacks of the traditional systems such as traffic jams, road accidents, etc. Transition can also improve traffic safety and achieve more reliable and efficient transportation [1]. However, it is not easy to facilitate the requirements of intelligent transportation in communication perspective because there should be an effective way to provide real-time information for road users, intelligent vehicles, and transportation system operators to make better decisions. Therefore, fast and reliable information exchange and data processing are really critical in applications such as autonomous driving and smart cars. To address these issues, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication have been introduced to enable those services [2].

V2V communication can be used to improve the performance of transportation systems in three main areas, i.e., safety, comfort, and efficiency [3]. When autonomous driving is considered, its safety totally depends on the the sensor data and information received from other vehicles and the infrastructures. One of the major problems in current transportation systems is a vast number of accidents happening in daily basis. According to [4], nearly 1.25 million people die in road accidents each year. If there is a way to communicate or exchange information among vehicles, it can benefit vehicles to take intelligent decisions such as avoiding traffic jams and accidents, and it might even be a help to minimize the human error and save lives.

In V2V communication, vehicles exchange their information such as the speed, location, direction of travel and braking. In V2I communication, vehicles communicate with infrastructures such as surveillance cameras, intelligent traffic signal posts, intelligent road blocks, etc [5], [6]. To achieve the performance of V2V and V2I communication, it is mandatory to have a reliable and low latency communication. However, with high mobility and a vast number of vehicles, it is very difficult to ensure quality of service (QoS) with limited resources. Vehicular ad-hoc networks (VANETs) has been introduced to achieve a reliable communication which consists of road side unit (RSU) and vehicles equipped with processing capabilities, various sensors, cameras, the global positioning system (GPS), radio transceivers, and other equipments [3].

Ultra reliable and low latency communication (URLLC) is one of the three pillars of fifth generation (5G), which is the main requirement of the mission-critical and low-latency applications such as V2V and V2I communication. Delivering information with low latency and ultra reliability significantly improves the performance of V2V and V2I communication.

Today, at the edge of 5G, driven by 5G specifications such as URLLC, there is a huge trend towards MEC, where computation and processing happen closer to the edge rather than in the cloud so that latency and traffic in the core network can be minimized [7]. At the same time, the resource limitations at vehicle nodes such as computational capabilities, power, and storage will also be released.

V2V and V2I communication requires very high QoS requests. Moreover, during a traffic jam there can be a huge number vehicles close to an intersection in an urban area [2] which makes it difficult to allocate the limited resources such as power and spectrum in an optimum manner. Due to high dynamics and dense urban environment, it is very difficult to achieve the strict requirements of latency and reliability.

Motivated by the above factors, we analyze the system performance of a vehicular edge computing network, which consists of V2I communication and MEC, in an urban environment. Our main objective is to achieve the low latency and high reliability requirements with V2I communication and MEC, where VUEs are able to offload the data for processing. Furthermore, we propose a joint utility and policy estimation-based learning approach. Firstly, each VUE observes its channel state to all the entities in the network and decides the communication action. Then each VUE observes its instantaneous utility, estimates the utility function, and builds a decision policy. In this thesis, the proposed approach is based on the risk-sensitive metric. Instead of minimizing the average delay in conventional communication design, we aim at minimizing the higher-order statistics of the delay distribution. Thus, the probability distribution function of E2E delay is compacted towards the mean E2E delay, and the probability of occurring higher delays is minimized resulting in minimal information loss. The proposed approach improves the reliability while minimizing the E2E latency so that stricter QoS requirements in V2I communication can be satisfied.

1.2 Structure of the Thesis

The thesis is structured as follows. First we thoroughly refer the related work and background in Chapter 2 including few related concepts, and theories, which is used in our problem formulation and system model. Then as mentioned in Section 1.1, the system model and problem formulation are introduced in the Chapter 3. In system model V2I communication scenario is developed, and MEC is embedded into the system model. Next we formulate the problem, setting our objective as satisfying the QoS requirements. Here, we introduce a risk-based latency minimization problem, which will take into account not only the average latency but also higher-order statistics of the latency distribution. To ensure high reliability, it is not enough to pay attention only on minimizing the latency. Therefore, our proposed method which minimizes the higher-order statistics of the delay distribution will ensure high reliability. There are two parts of the problem formulation. Firstly, we develop the optimization problem for each VUE in which the VUE's objective is to find out the decision policy that minimizes E2E latency while maximizing the reliability. Then we develop another optimization problem at the MEC server, where the MEC server finds the optimum power allocation for all the VUEs that minimizes the exponential sum of E2E latencies. In Chapter 4, we propose a risk-based distributed learning algorithm that each VUE learns the best decision policy with time, so that the VUE's optimization problem can be solved. Then we introduce our baseline approach which we use to compare our proposed algorithm. The last chapter is about the simulation and results. There, we compare the results of the proposed algorithm with the baseline approach and analyze performance of the proposed approach varying system parameters, and based on results, conclusion is developed. According to the results it can be seen that the proposed approach performs way better than the baseline approach, and standard deviation of the delay distribution is low compared to all other approaches, so that reliability is high. Finally all the references are listed.

2 LITERATURE AND RELATED CONCEPTS

2.1 Literature

Most of the research work in V2V and V2I communication consider VANETs with computing capabilities integrated into the system model. Since the MEC architecture extends computing, storage, and applications to the network edge, it helps to reduce traffic in back-haul and E2E latency while ensuring high reliability. The authors in [8] consider a scenario, where vehicles and fixed road infrastructures are integrated to VANET to build a edge computing-enabled VANET. Therein, to ensure high reliability in the considered architecture, the authors propose a reliable computation uploading strategy considering partial offloading, task allocation, and re-transmission processing.

RSUs play a key role on enabling various vehicular applications such as autonomous driving, road safety, infotainment, and collaboration services with high throughput and low latency. In the work [9], the authors study the viability of the solar-powered RSU, consisting of small cell base stations and MEC servers. Since the solar power has spatial and temporal fluctuations, and data traffic demand also varies with time, there might be a mismatch between the RSU's power consumption and solar power generation, leading severe QoS. The authors jointly study the RSU's power consumption minimization problem, the temporal energy balancing problem, and the spatial energy balancing problem. Subsequently, three algorithms are proposed, which decide battery charging and user association control for minimizing the QoS loss under the delay constraint of the computing tasks.

Local vehicular computing solely cannot satisfy the computational needs of vehicular networks, which is very sensitive to computational power. Additionally, it is very challenging to guarantee the reliability of vehicular computation offloading due to the dynamic and random nature of vehicular networks. The authors in [10] study a reliability-oriented stochastic optimization for V2I-based computation offloading, which improves the reliability performance in stochastic situations and can be used to design a threshold-based decision making policy for computation offloading. The proposed approach is based on the dynamic programming for computation offloading in the presence of the deadline constraint on application execution. The authors improve the reliability of computation offloading by maximizing the lower bound of the expected successful probability of data transmissions.

Edge computing has been proposed to address the challenges of the conventional cloud computing paradigm. The major drawback of conventional cloud computing is the delay caused by the limited backhaul capacity and excessive network hops. Edge computing requires a large scale deployment of edge computing servers to successfully cater the stringent requirements of QoS and quality of experience (QoE) of vehicular applications which are both delay and computational intensive [11]. The large scale deployment of edge computing servers also causes management and operational problems that might add additional cost. One of the promising solutions is vehicular fog computing [12]. Therein, base stations and edge computing servers are able to offload their overloaded tasks to nearby vehicles with under-utilized computational resources. However, despite the above mentioned advantages, it is difficult to find an optimal task assignment strategy in a way that fulfills the low-latency and high-reliability requirements [13]. There are many mathematical tools to address the task assignment problem including matching theory

[14], coalitions game [15], and Stackelberg game [16]. In [13], the authors present a low-latency massive-connectivity vehicular fog computing framework using two-dimensional matching algorithm to deal with the task assignment problem between vehicular fog nodes and user equipments (UEs). The proposed pricing-based stable matching algorithm is used to drive the stable matching.

Performance of wireless vehicular communication systems highly depend on efficient radio resource management (RRM), but stringent QoS-based V2V communication requirements make it challenging [17]. The quality of the wireless links in V2V and V2I communication varies due to the mobility of vehicles. If queuing latency is considered, it varies due to dynamic nature of traffic arrivals and service rates. There are few works focusing on bounding the maximal queue length within a threshold value and do the radio resource management (RRM) [18], [19].

Some works focus on latency and reliability. The work [18] proposes a proximity and QoS-aware resource allocation approach considering the queuing latency and reliability requirements. The main problem is decoupled into two interrelated sub-problems. Firstly, the RSU groups vehicles into zones based on their physical proximity using the proposed virtual clustering mechanism and allocates resource blocks in each zone given the vehicles' traffic demands and their QoS requirements. Then the power minimization problem of each VUE is formulated subject to probability constraints on data queue length which is a measure of queuing latency and reliability. Even though there is a reduction in average queue length, it does not cover the reliability aspects of V2V communication.

The authors in [19] model a power minimization problem considering the network-wide maximal data queue length. The problem is subject to both first and second-orders statistics of latency and reliability. To avoid incurring signalling overhead by exchanging queue state information (QSI) among vehicles and the RSU, authors leverage principles of extreme value theory (EVT) locally to estimate the maximal queue length. This semi-centralized and distributed dynamic power allocation solutions combining tools from Lyapunov stochastic optimization and EVT minimize the mean and variance of the maximal queue length. However, both works [18] and [19] only consider the queue length and queue delay statistic to define the reliability which is a very small part when computational and transmission delay is considered.

To our best knowledge, the formulated framework to achieve high reliability while minimizing the latency considering E2E delay profile, i.e., computation and communication, has not been investigated in the literature. At the same time, there are many distributed learning approaches, which learn the optimal decision policy with minimum or no information exchange [20], [21]. Nevertheless, this is the first work in which the distributed learning approach is used to model a V2I communication problem considering the effect of the random variations of the channel between vehicle and the other entities. Therefore, the proposed approach is very suitable for scenarios in V2V and V2I communication since the overhead information exchange can also be minimized ensuring low latency and high reliability.

2.2 Related Concepts

2.2.1 Manhattan Mobility Model

When the performance of the VANET is analyzed, there are so many characteristics that affect performance of the network including movement of vehicles, positions, cross roads, traffic lights, vehicle density, speed restrictions, road side obstacles, speed variations, etc. [22].

Mobility models have been introduced to represent these characteristics that affect the communication between V2V and V2I. The Manhattan mobility model is used to model the movements of vehicles in urban areas.

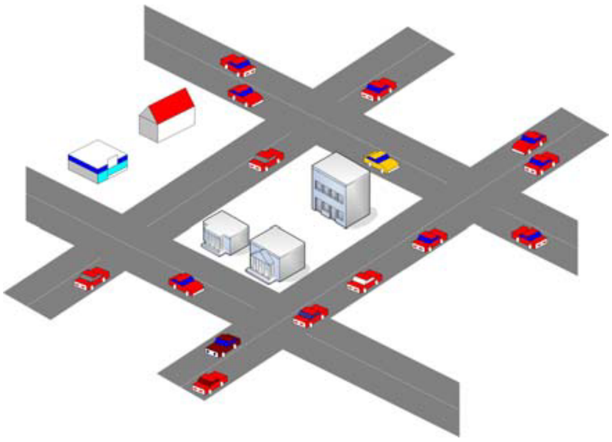


Figure 2.1. Manhattan mobility model [22].

Figure 2.1 represents a map of an urban area in which each street has a two lanes for each direction and vehicles move on these horizontal and vertical streets. When a vehicle reaches an intersection, it can move with 0.5 probability on same street, 0.25 probability of turning to left, and 0.25 probability of turning to right [23]. The velocity of the vehicles is restricted, and there is a velocity dependency between two vehicles. Vehicles are allowed to change their lanes and there are high temporal and spatial dependencies.

2.2.2 Rayleigh Fading

Rayleigh fading is a statistical model which is used to model the effect of transmission medium, i.e., propagation environment, which varies according to the Rayleigh distribution. This model is mostly used to model urban environments, where there is no dominant propagation path between the transmitter and the receiver, i.e., non-line-of-sight propagation. If there is a dominant path, i.e., line-of-sight (LOS) propagation, Rician fading is used rather than Rayleigh fading. When the radio signal heavily scatters along the way due to the objects such as buildings, trees, and other objects, according to the central limit theorem, the channel impulse response can be modeled as a Gaussian distribution irrespective of the distribution of individual components. Thus, the Rayleigh fading channel can be modeled with the following probability distribution

$$P_r(x) = \frac{x}{\sigma^2} e^{(\frac{-x^2}{2\sigma^2})}, x \geq 0. \tag{1}$$

2.2.3 Path Loss

Path loss is the variation of the received signal power over the distance when a radio signal propagates through space. This happens because of dissipation of the transmitted power due to the effects such as free-space loss, refraction, diffraction, reflection, and absorption. Path loss is influenced by the environment, characteristics of the space, and also the distance between transmitter and receiver. Path loss in the linear scale is defined as the ratio of transmitted power to received power and expressed as follows:

$$P_L = \frac{P_t}{P_r}. \tag{2}$$

In addition, we write path loss in the log-scale as

$$P_L(dB) = 10 \log_{10}(\frac{P_t}{P_r}). \tag{3}$$

There are many path loss models which are being used to model the transmitted power loss. The simplest one is the free space path loss model that assumes LOS and no obstacles between the transmitter and receiver. Ray tracing, two-ray model, and dielectric canyon (ten-ray model) [24] are some other models which consider the complex environment effects such as shadowing and multi-path effects. But for the general analysis, it is convenient to use a simplified path loss model which captures the essence of signal propagation. Following the simplified and commonly-used path loss model [24], which is a function of distance, the attenuated power is expressed as

$$P_r = P_t K [\frac{d_0}{d}]^\gamma. \tag{4}$$

Similarly, the attenuated power in dBm is

$$P_r(dBm) = P_t(dBm) + 10 \log_{10}(K) - 10\gamma \log_{10}(\frac{d}{d_0}). \tag{5}$$

Here, K is a constant, d_0 is a reference distance for the antenna far-field, and γ is the path loss exponent.

2.2.4 Edge Computing

Emerging technologies such as Internet of things (IoT), V2V and V2I communication generate a huge amount of data that need to be processed, subject to low latency and high reliability requirements. This makes cloud computing and other conventional approaches inefficient due to bandwidth limitations and the large response time.

The main objective of edge computing is to push computing, network control, and storage closer to the location where it is needed (e.g., base stations and access points) promising dramatic reduction in latency and high reliability [25].

MEC is one solution which enables IT and cloud computing capabilities within the radio access networks. Mobile devices offload computational tasks to MEC servers due to lack of computational power and also energy saving purposes [25]. Evaluating computational performances of mobile and edge nodes are very important since it directly affects the E2E latency of whole communication. Moreover, the central processing unit (CPU) cycle frequency of the MEC server or the mobile device is the main performance indicator when execution latency of particular task is calculated.

Given that L is the task input data size (in bits), and X is the computational workload intensity (in CPU cycles per bit), the execution latency (T_c) can be calculated as follows:

$$T_c = \frac{LX}{f_s}. \tag{6}$$

As per the above equation, high CPU clock speed is preferred for low latency.

3 SYSTEM MODEL AND PROBLEM FORMULATION

3.1 System Architecture and End-to-End Delay

We consider a Manhattan mobility model as shown in Figure 3.1, in which a set $\mathcal{C}$ of C cameras equipped with transceivers are installed at an intersection. A set $\mathcal{V}$ of V vehicular user equipments (VUEs) are randomly located on two perpendicular roads under the coverage of a single RSU, which is installed close to the intersection. The cameras at the intersection monitor the nearby street view, and all cameras' images should be synthesized to have the full view of the intersection. All VUEs use the synthesized image to make intelligent decisions in order to sustain the traffic safety. Synthesizing procedure (processing) is unique for each VUE. A MEC server is installed at the RSU to provide the computational services, and all the cameras are connected to the RSU through a fiber link.

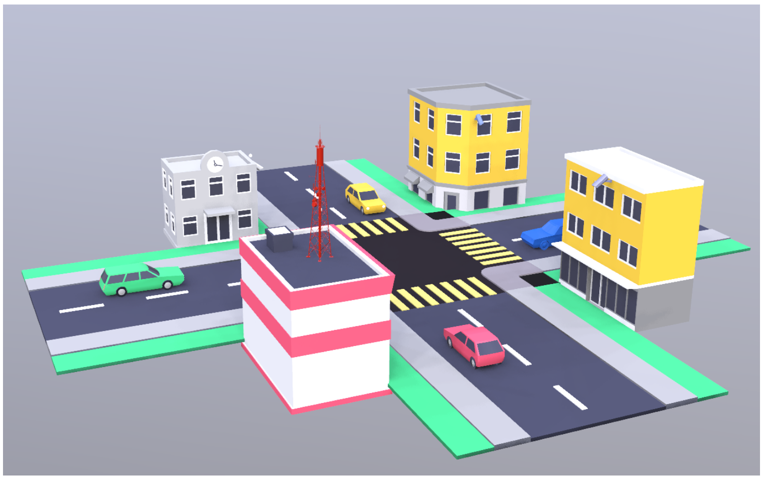


Figure 3.1. System model.

VUEs also have their own computational capabilities. Compared to the MEC server, VUE's processing capability is weaker. Therefore, VUEs are in favour of receiving synthesized images through MEC server rather than directly receiving raw images from cameras and synthesizing on their own. However, if all the VUEs intend to offload, the MEC server's computational, power, and spectrum resources will be shared among the VUEs. The system performance such as computational and transmission latency will increase, resulting in inefficient communication. When the performance of the whole system is considered, there is a trade-off between offloading to the MEC server and local computation.

We let $\alpha_i \in \{0, 1\}$ to express VUE i's decision, either for task-fetching which is directly receiving the images from cameras and compute locally (by $\alpha_i = 0$), or offloading which is receiving the synthesized image from MEC server (by $\alpha_i = 1$) as illustrated in Figure 3.2. Moreover, h_{ji} denotes the channel between camera j and VUE i, and the downlink (DL) channel from the MEC server to VUE i is h_{si}. P_c is camera's transmit power which is identical for all cameras. Each camera has its own dedicated bandwidth W_c. Hence, when a camera sends its image to the VUEs, there is no interference from the other cameras. The value of α_i is decided by the each VUE having intention of minimizing the E2E latency while ensuring the reliability.

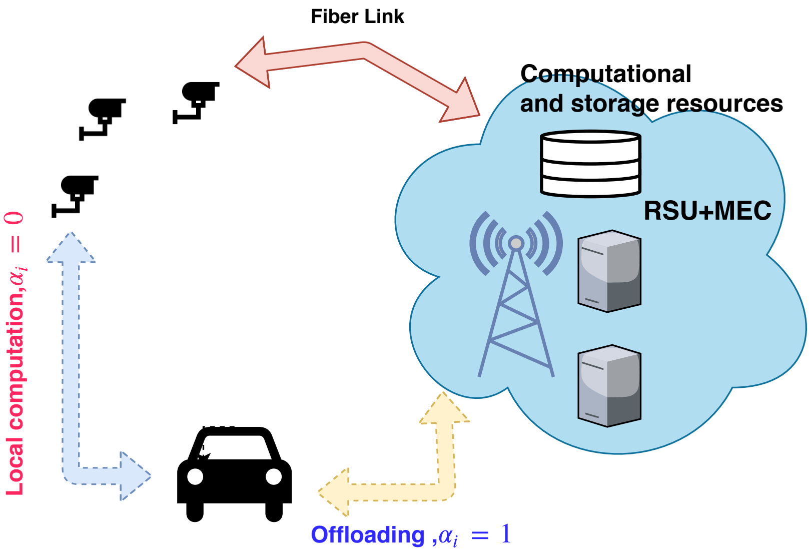

Fiber Link
Computational and storage resources
RSU+MEC
Local computation,$\alpha_i = 0$
Offloading ,$\alpha_i = 1$

Figure 3.2. System model overview.

We assume that each VUE knows its full channel state information to all cameras and to the MEC server. All the cameras know their channel states to all the VUEs, and MEC server also knows its channel state information to all the VUEs. Before communication starts, VUE i sends its decision α_i to the cameras (if α_i=0) or to the MEC server (if α_i=1), as shown in the Figure 3.3 and Figure 3.4.

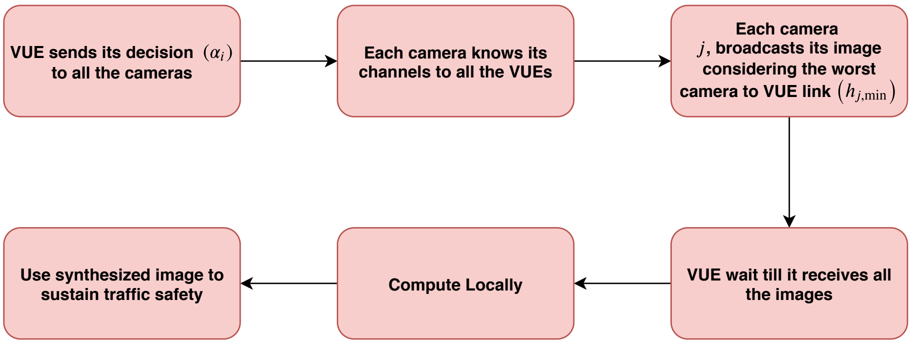

VUE sends its decision (α_i) to all the cameras
Each camera knows its channels to all the VUEs
Each camera j, broadcasts its image considering the worst camera to VUE link $(h_{j,\min})$
VUE wait till it receives all the images
Compute Locally
Use synthesized image to sustain traffic safety

Figure 3.3. End to end communication flow given α_i=0.

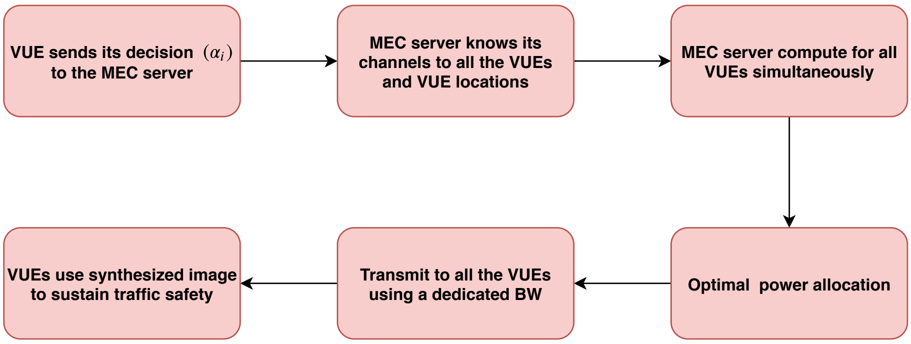

VUE sends its decision (α_i) to the MEC server
MEC server knows its channels to all the VUEs and VUE locations
MEC server compute for all VUEs simultaneously
Optimal power allocation
Transmit to all the VUEs using a dedicated BW
VUEs use synthesized image to sustain traffic safety

Figure 3.4. End to end communication flow given α_i=1.

Let us denote $\mathcal{V}_{\mathrm{f}} = \{i \in \mathcal{V} | \, \alpha_i = 0\}$ as the set of all task-fetching VUEs. In order to ensure that all VUEs in $\mathcal{V}_{\mathrm{f}}$ can correctly receive the image, each camera j sends its data with the rate R_j, which is decided by considering the minimal channel coefficient among all corresponding camera-VUE links, to ensure that all VUEs in V_{f} would be able to receive the information without a loss. Accordingly, the data rate from camera j to VUEs is

$$R_j = W_{\mathrm{c}} \log_2 \Big(1 + \frac{P_{\mathrm{c}} h_{j,\min}}{W_{\mathrm{c}} N_0}\Big), \tag{7}$$

where N_0 is the noise variance. Then the the transmission delay from camera j to VUEs is

$$T_j^{\mathrm{Tx}} = A/R_j. \tag{8}$$

Here, A is the size of a camera image. Since VUE i has to wait till it receives all cameras' images before start synthesizing, the transmission delay in the task fetching phase is

$$T^{\mathrm{Fet}} = \max_{j \in \mathcal{C}} T_j^{\mathrm{Tx}}. \tag{9}$$

We assume all camera images are in the same size. L is the required CPU cycles per bit for computation, i.e., the processing density, and f_i is the CPU cycle frequency (in cycles per second) of each VUE. The computation delay at the VUE i is calculated as,

$$T_i^{\mathrm{Comp}} = CAL/f_i. \tag{10}$$

If $\alpha_i = 0$, VUE i's E2E delay T_i^{E2E} includes the transmission delay and the local computation delay, i.e.,

$$T_i^{\mathrm{E2E}} = \max_{j \in \mathcal{C}} \left\{ \frac{A}{W_{\mathrm{c}} \log_2 \left(1 + \frac{P_{\mathrm{c}} h_{j,\min}}{W_{\mathrm{c}} N_0}\right)} \right\} + \frac{CAL}{f_i}. \tag{11}$$

For task offloading, let us analogously denote the set of VUEs with the task-offloading decision as $\mathcal{V}_{\mathrm{o}} = \{i \in \mathcal{V} | \, \alpha_i = 1\}$. When α_i=1, the communication and computation procedures happen in three steps. First all cameras send their images to the MEC server through a fiber link and transmission delays from cameras to the server are negligible compared to the other delays in the network. When MEC server receives all the camera images, it starts to synthesize for all VUEs in $\mathcal{V}_{\mathrm{o}}$. Here we consider that the MEC server computes all the VUEs' dedicated images simultaneously so that total computational resources are shared among VUEs equally. The computational delay at the MEC server is expressed as

$$T_{\mathrm{s}}^{\mathrm{Comp}} = |\mathcal{V}_{\mathrm{o}}| \frac{CAL}{f_{\mathrm{s}}}. \tag{12}$$

Subsequently, the MEC server sends the synthesized image to each VUE $i \in \mathcal{V}_{\mathrm{o}}$ using a dedicated bandwidth W_s and the transmit power P_i. The power allocation mechanism will be detailed in Section 3.3. Accordingly, the the corresponding DL rate (from the MEC server) to the VUE i is expressed as

$$R_i = \frac{W_\mathrm{s}}{|\mathcal{V}_\mathrm{o}|}\log_2\Big(1+\frac{P_i h_{\mathrm{s}i}|\mathcal{V}_\mathrm{o}|}{W_\mathrm{s}N_0}\Big). \tag{13}$$

Then the DL transmission delay from MEC server to the VUE i is

$$T_i^{\mathrm{DL}} = B/R_i, \tag{14}$$

where B denotes the synthesized image size. If $\alpha_i = 1$, VUE i's E2E delay includes the computation delay at the MEC server and the DL transmission delay, i.e.,

$$T_i^{\mathrm{E2E}} = \frac{|\mathcal{V}_\mathrm{o}|CAL}{f_\mathrm{s}} + \frac{B|\mathcal{V}_\mathrm{o}|}{W_\mathrm{s}\log_2\left(1+\frac{P_i h_{\mathrm{s}i}|\mathcal{V}_\mathrm{o}|}{W_\mathrm{s}N_0}\right)}. \tag{15}$$

3.2 Risk Minimization for the VUE's End-to-End Delay

URLLC promises to reduce E2E latency while improving the reliability. Latency-sensitive and mission-critical applications like V2V and V2I communication strongly depend on the low latency and ultra reliability. Motivated by V2V and V2I communication's latency and reliability requirements, we pay attention to the E2E delays of VUEs. As a reliability measure, we leverage the concept of risk from financial mathematics, where risk is closely associated with gaining or losing something valuable. Since higher delays can result in losing the information along the way putting traffic safety at a stake, we aim at minimizing the risk which means minimizing the probability of occurrence of higher delays. To do that we use the entropic risk measure which is defined as $\frac{1}{\rho}\ln(\mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})])$, with a risk sensitivity parameter $\rho > 0$. By taking the Maclaurin series expansion [19], we can get

$$\frac{1}{\rho}\ln(\mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})]) = \mathbb{E}[T_i^{\mathrm{E2E}}] + \frac{\rho}{2}\mathrm{Var}(T_i^{\mathrm{E2E}}) + \mathcal{O}(\rho^2), \quad \rho \in (0,1). \tag{16}$$

Therefore, we formulate our problem as a risk minimization problem of VUE's E2E delay. Our approach focuses on not only minimizing the latency, but also reducing the risk so that high reliability can be ensured. We define the channel state vector of VUE i as

$$\mathbf{h}_i = [h_{ji}, h_{\mathrm{s}i} : j \in \mathcal{C}] \in \mathcal{H}_i, \tag{17}$$

We simplify (16), making sure that simplification does not affect our objective and it is solvable. Thus, each VUE's optimization problem can be defined as follows:

$$\begin{aligned}
&\underset{\Pr(\alpha_i|\mathbf{h}_i)}{\text{minimize}} \quad \mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})] && && \text{(18a)}\\
&\text{subject to} \quad \sum_{\alpha_i\in\{0,1\}} \Pr(\alpha_i|\mathbf{h}_i) = 1, && \forall\,\mathbf{h}_i \in \mathcal{H}_i, && \text{(18b)}\\
&\qquad\qquad \Pr(\alpha_i = 0|\mathbf{h}_i) \geq 0, && \forall\,\mathbf{h}_i \in \mathcal{H}_i, && \text{(18c)}\\
&\qquad\qquad \Pr(\alpha_i = 1|\mathbf{h}_i) \geq 0, && \forall\,\mathbf{h}_i \in \mathcal{H}_i. && \text{(18d)}
\end{aligned}$$

Our objective function can be expanded using Maclaurin series as follows:

$$\mathbb{E}[\exp(\rho T_i^{\mathrm{E2E}})] = 1 + \rho\mathbb{E}[T_i^{\mathrm{E2E}}] + \frac{\rho^2}{2!}\mathbb{E}[(T_i^{\mathrm{E2E}})^2] + \frac{\rho^3}{3!}\mathbb{E}[(T_i^{\mathrm{E2E}})^3] + \cdots. \tag{19}$$

As per (19), while minimizing the objective function, all the higher-order statistics of the VUE i's E2E delay are taken into account along with the average delay. In the above optimization problem, each VUE's objective is to find the optimal decision policy $\Pr(\alpha_i|\mathbf{h}_i)$, knowing the channel state $\mathbf{h}_i$.

3.3 Transmit Power Allocation at the MEC Server

Referring to the motivation of considering (18a), we formulate the MEC server's power allocation problem as

$$\underset{P_i}{\text{minimize}} \quad \sum_{i\in\mathcal{V}_\mathrm{o}} \exp(\rho T_i^{\mathrm{DL}}) \tag{20a}$$

$$\text{subject to} \quad \sum_{i\in\mathcal{V}_\mathrm{o}} P_i = P_{\max} \text{ and } P_i \geq 0, \ \forall\, i \in \mathcal{V}_\mathrm{o}, \tag{20b}$$

with the MEC server's total transmit power budget $P_{\max}$. In the objective (20a), we consider the DL transmission delay since the allocated transmit power only affects this delay. Our objective function (20a) places emphasis on decreasing higher delays rather than treating all the delays equally.

To find a closed-form solution for the (20), the convexity of the objective function should be examined. To this end, we firstly simplify our problem as follows:

$$\underset{P_i}{\text{minimize}} \quad \sum_{i\in\mathcal{V}_\mathrm{o}} \exp\Big(\frac{\theta}{\ln(1+P_i\kappa_i)}\Big), \tag{21a}$$

where

$$\theta = \frac{\rho B|\mathcal{V}_\mathrm{o}|\ln 2}{W_\mathrm{s}} \qquad \text{and} \qquad \kappa_i = \frac{h_{\mathrm{s}i}|\mathcal{V}_\mathrm{o}|}{W_\mathrm{s}N_0}. \tag{22}$$

Taking the first and second derivative subject to our optimization variable P_i, we have

$$f'(P_i) = \frac{-\kappa_i\theta}{(1+\kappa_iP_i)}\frac{\exp(\frac{\theta}{\ln(1+\kappa_iP_i)})}{[\ln(1+\kappa_iP_i)]^2}, \tag{23}$$

$$f''(P_i) = \frac{\kappa_i^2\theta}{(1+\kappa_iP_i)^2}\frac{\exp(\frac{\theta}{\ln(1+\kappa_iP_i)})}{[\ln^2(1+\kappa_iP_i)]^2}\Big(P_i + \frac{\theta}{[\ln(1+\kappa_iP_i)]^2} + \frac{2}{\ln(1+\kappa_iP_i)}\Big). \tag{24}$$

Considering the second derivative, $f''(P_i) > 0$, our objective function satisfies the convexity criteria [26]. Then, the Lagrangian and Karush–Kuhn–Tucker (KKT) conditions can be used to find a closed-form solution to the problem (20). Given that

λ_i and μ are the Lagrangian multipliers, the Lagrangian of the problem is written as follows:

$$L(P_i, \lambda_i, \mu_i) = \sum_{i \in \mathcal{V}_\mathrm{o}} \exp(\frac{\theta}{\ln(1+\kappa_i P_i)}) - \sum_{i \in \mathcal{V}_\mathrm{o}} \lambda_i P_i + \mu(\sum_{i \in \mathcal{V}_\mathrm{o}} P_i - P_{\max}). \tag{25}$$

The KKT conditions include

1) Primal feasibility,

$$-P_i \leqq 0, \qquad \sum_{i \in \mathcal{V}_\mathrm{o}} P_i - P_{\max} = 0. \tag{26}$$

2) Dual feasibility,

$$\lambda_i \geq 0. \tag{27}$$

3) Complementary slackness,

$$\lambda_i(-P_i) = 0. \tag{28}$$

4) Equate the derivative of the Lagrangian with respect to P_i to 0, i.e.,

$$L'(P_i, \lambda_i, \mu_i) = \frac{-\kappa_i \theta}{(1+\kappa_i P_i)} \frac{\exp(\frac{\theta}{\ln(1+\kappa_i P_i)})}{[\ln(1+\kappa_i P_i)]^2} - \lambda_i + \mu = 0. \tag{29}$$

Our goal is to find the power allocation which satisfies (26)−(29). From (29), we can find $P_i \neq 0$. Otherwise, there do not exist any finite solutions for λ_i and μ. Thus, the only possibility is $P_i > 0$ with $\lambda_i = 0$ according to (27) and (28). Then, (29) is rewritten as

$$\mu = \frac{\kappa_i \theta}{(1+\kappa_i P_i)} \frac{\exp(\frac{\theta}{\ln(1+\kappa_i P_i)})}{[\ln(1+\kappa_i P_i)]^2}. \tag{30}$$

The power allocation is that the MEC server allocates the transmit power $P_i^* > 0, \forall\, i \in \mathcal{V}_\mathrm{o}$, which satisfies

$$\frac{\theta \kappa_i \exp\left(\frac{\theta}{\ln(1+P_i^* \kappa_i)}\right)}{(1+P_i^* \kappa_i)[\ln(1+P_i^* \kappa_i)]^2} = \mu. \tag{31}$$

Here, μ is chosen such that $\sum_{i \in \mathcal{V}_\mathrm{o}} P_i^* = P_{\max}$.

4 RISK MINIMIZATION FOR THE VUE'S END-TO-END DELAY

4.1 Regret-Based Risk-Sensitive Approach for Task Fetching and Offloading

VUE's optimization problem (18) can not be solved using conventional optimization techniques since the objective function depends on the network circumstances that particular VUE has no control. Therefore, we consider a regret minimization-based learning approach assuming that each VUE has the perfect information of the network such as other VUEs' wireless channels and task-fetching and offloading decisions. Let $\boldsymbol{\omega}_i$ denote all unobservable uncertainties of VUE i's, including the other VUEs' wireless channels and task-fetching and offloading decisions, the value of $\mathbf{h}_i$ belongs to a finite set $\mathcal{H}_i$, i.e.,

$$\mathcal{H}_i = \{\mathbf{h}_i^1, \cdots, \mathbf{h}_i^l, \cdots, \mathbf{h}_i^{|\mathcal{H}_i|}\}. \tag{32}$$

We assume that the communication timeline is slotted and indexed by t. In addition, referring to the objective (18a), we denote $u_i(t)$ as VUE's utility (u_i) in time slot t, given the specific values of $\alpha_i(t)$, $\mathbf{h}_i(t)$ and $\omega_i(t)$ in slot t, which is defined as

$$u_i(t) = u_i(\alpha_i(t), \mathbf{h}_i(t), \boldsymbol{\omega}_i(t)) = -\exp(\rho T_i^{\mathrm{E2E}}(t)). \tag{33}$$

Let's assume that a given VUE i compares the time average of its utility observations, obtained by constantly changing its action following a particular decision policy $\Pr(\alpha_i|\mathbf{h}_i)$, with the case where it would have taken the same decision (α_i^m) in all previous time instances, while other VUEs use their current decision policies. We assume that all VUEs are interested in choosing the probability distribution $\beta(\mathbf{r}_{\mathbf{h}_i^l}(t); \alpha_i^m)$ that minimizes its regret not having played action α_i^m from $\tau = 1$ to time t which is calculated as

$$r_{\mathbf{h}_i^l}(t; \alpha_i^m) = \frac{1}{\sum_{\tau=1}^{t} \mathbb{1}_{\{\mathbf{h}_i(\tau)=\mathbf{h}_i^l\}}} \times \sum_{\tau=1}^{t} \Big[u_i(\alpha_i^m, \mathbf{h}_i(\tau), \boldsymbol{\omega}_i(\tau)) - u_i(\tau)\Big] \times \mathbb{1}_{\{\mathbf{h}_i(\tau)=\mathbf{h}_i^l\}}, \tag{34}$$

denoting,

$$\mathbf{r}_{\mathbf{h}_i^l}(t) = [r_{\mathbf{h}_i^l}(t; 0), r_{\mathbf{h}_i^l}(t; 1)]. \tag{35}$$

As the time elapses, the empirical distribution $\beta(\mathbf{r}_{\mathbf{h}_i^l}(t); \alpha_i^m)$ provides a solution to the risk minimization problem which is a solution of the following optimization problem:

$$\beta(\mathbf{r}_{\mathbf{h}_i^l}(t); \alpha_i) = \underset{\Pr(\alpha_i|\mathbf{h}_i^l)}{\arg\max} \left\{ \sum_{\alpha_i^m=0}^{1} \left[\Pr(\alpha_i^m|\mathbf{h}_i^l) r_{\mathbf{h}_i^l}(t; \alpha_i^m) + \frac{1}{\xi} \Pr(\alpha_i^m|\mathbf{h}_i^l) \ln\left(\frac{1}{\Pr(\alpha_i^m|\mathbf{h}_i^l)}\right) \right] \right\}, \tag{36}$$

where, $\xi > 0$ is the temperature parameter which balances between exploitation (by maximizing average regret) and exploration (by maximizing information entropy). The resulting probability distribution for VUE i is given by [20],

$$\beta(\mathbf{r}_{\mathbf{h}_i^l}(t); \alpha_i^m) = \frac{\exp\left(\xi r_{\mathbf{h}_i^l}^{+}(t; \alpha_i^m)\right)}{\sum_{\alpha_i^m=0}^{1} \exp\left(\xi r_{\mathbf{h}_i^l}^{+}(t; \alpha_i^m)\right)}, \tag{37}$$

in which, $r_{\mathbf{h}_i^l}^{+}(t; \alpha_i^m) = \max\{r_{\mathbf{h}_i^l}(t; \alpha_i^m), 0\}$.

Regret learning relies on the assumption in each iteration t, VUE i is able to evaluate its instantaneous utility and calculate the utility in which the same action is constantly taken. However, to this end, VUE i should be aware of $\boldsymbol{\omega}_i$, all the other VUEs' wireless channels and task-fetching and offloading decisions. In reality, these required information is not available at VUEs, and it is very complex to gather in a practical wireless network since there is no information exchange among VUEs. Due to lack of information, the VUE is unable to find policy (37).

4.2 Distributed Risk-Sensitive Approach for Task Fetching and Offloading

Since regret learning approach is not practical, we must rely on an approach that only depends on the information available at each VUE. Therefore, we propose a distributed learning approach [20] which only relies on the utility observations and channel state information available at the VUE. Note that VUEs do not need the information about $\boldsymbol{\omega}_i$.

As shown in Figure 4.1, at each time instant, each VUE autonomously selects an action α_i, knowing the channel state $\mathbf{h}_i$, and observes the E2E delay. The observed E2E delay is used by each VUE to estimate its utility and the regret of constantly taking a specific decision knowing the channel state.

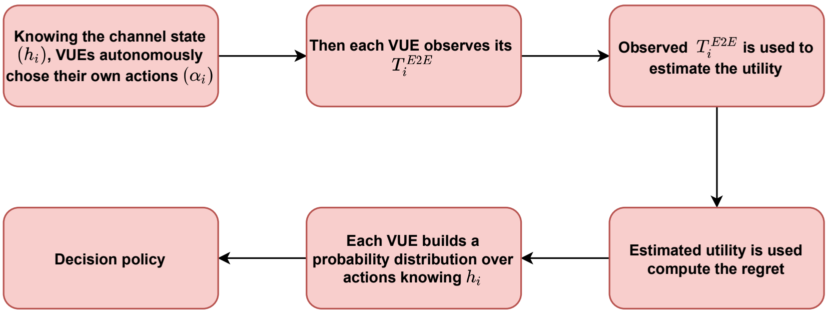

Knowing the channel state (h_i), VUEs autonomously chose their own actions (α_i)
Then each VUE observes its T_i^{E2E}
Observed T_i^{E2E} is used to estimate the utility
Estimated utility is used compute the regret
Each VUE builds a probability distribution over actions knowing h_i
Decision policy

Figure 4.1. Distributed learning flow diagram.

At each time instant, given the channel state and the decision taken by the VUE i at time t, the estimation of utility, regrets, and probability distribution function carried out by each VUE are updated for all α_i^m as follows:

$$\begin{cases} \hat{u}_{\mathbf{h}_i^l}(t;\alpha_i^m) = \hat{u}_{\mathbf{h}_i^l}(t-1;\alpha_i^m) + \eta_u(t) \times \mathbb{1}_{\{\mathbf{h}_i(t)=\mathbf{h}_i^l\}} \\ \qquad \times \mathbb{1}_{\{\alpha_i(t)=\alpha_i^m\}} \times \Big[u(t) - \hat{u}_{\mathbf{h}_i^l}(t-1;\alpha_i^m)\Big], & \text{(38a)} \\ \hat{r}_{\mathbf{h}_i^l}(t;\alpha_i^m) = \hat{r}_{\mathbf{h}_i^l}(t-1;\alpha_i^m) + \eta_r(t) \times \mathbb{1}_{\{\mathbf{h}_i(t)=\mathbf{h}_i^l\}} \\ \qquad \times \Big[\hat{u}_{\mathbf{h}_i^l}(t;\alpha_i^m) - u(t) - \hat{r}_{\mathbf{h}_i^l}(t-1;\alpha_i^m)\Big], & \text{(38b)} \\ \pi_{\mathbf{h}_i^l}(t;\alpha_i^m) = \pi_{\mathbf{h}_i^l}(t-1;\alpha_i^m) + \eta_\pi(t) \times \mathbb{1}_{\{\mathbf{h}_i(t)=\mathbf{h}_i^l\}} \\ \qquad \times \Big[\beta(\hat{\mathbf{r}}_{\mathbf{h}_i^l}(t);\alpha_i^m) - \pi_{\mathbf{h}_i^l}(t-1;\alpha_i^m)\Big]. & \text{(38c)} \end{cases}$$

For $\beta(\hat{\mathbf{r}}_{\mathbf{h}_i^l}(t);\alpha_i^m)$ in (37) and (38c), we fix $\xi(t) = D$, where D is a constant. Thus, this distributed non-regret learning algorithm puts the same emphasis on exploring the utilities of different actions and leveraging the discovered information. Additionally, the learning rates $\eta_u(t)$, $\eta_r(t)$, $\eta_\xi(t)$, and $\eta_\pi(t)$ should satisfy

$$\lim_{N\to\infty} \sum_{t=1}^{N} \eta_u(t) = \infty, \qquad \lim_{N\to\infty} \sum_{t=1}^{N} \eta_r(t) = \infty, \tag{39}$$

$$\lim_{N\to\infty} \sum_{t=1}^{t} \eta_\pi(t) = \infty, \qquad \lim_{N\to\infty} \sum_{t=1}^{N} [\eta_u(t)]^2 < \infty, \tag{40}$$

$$\lim_{N\to\infty} \sum_{t=1}^{N} [\eta_r(t)]^2 < \infty, \qquad \lim_{N\to\infty} \sum_{t=1}^{N} [\eta_\pi(t)]^2 < \infty, \tag{41}$$

$$\lim_{t\to\infty} \frac{\eta_r(t)}{\eta_u(t)} = 0, \qquad \lim_{t\to\infty} \frac{\eta_\pi(t)}{\eta_r(t)} = 0, \tag{42}$$

whose values can be chosen by referring to p-series. At the time slot $t+1$, VUE i decides optimum decision policy as follows:

$$\Pr(\alpha_i = \alpha_i^m | \mathbf{h}_i = \mathbf{h}_i^l) = \pi_{\mathbf{h}_i^l}(t;\alpha_i^m). \tag{43}$$

The converged distribution $\pi_{\mathbf{h}_i^l}(\infty;\alpha_i^m)$ provides a solution to our studied problem (18).

4.3 Average Delay Minimization for the VUE's End-to-End Delay

Since our main goal is to decrease the higher-order statistics while minimizing the average delay, to emphasize the reliability improvement through our proposed approach, we consider the average delay minimization approach for the VUE's E2E delay. Analogously to our proposed case, we define average delay minimization problem for the VUE's E2E delay. Then we use the distributed learning approach for task-fetching and offloading. After that we formulate the transmit power allocation problem of the MEC server based on the average delay. Finally we compare the results of proposed and baseline approaches in Chapter 5.

We formulate our baseline problem as

$$\begin{aligned}
&\underset{\Pr(\alpha_i|\mathbf{h}_i)}{\text{minimize}} && \mathbb{E}[T_i^{\text{E2E}}] && && \text{(44a)}\\
&\text{subject to} && \sum_{\alpha_i\in\{0,1\}} \Pr(\alpha_i|\mathbf{h}_i) = 1, && \forall\, \mathbf{h}_i \in \mathcal{H}_i, && \text{(44b)}\\
& && \Pr(\alpha_i = 0|\mathbf{h}_i) \geq 0, && \forall\, \mathbf{h}_i \in \mathcal{H}_i, && \text{(44c)}\\
& && \Pr(\alpha_i = 1|\mathbf{h}_i) \geq 0, && \forall\, \mathbf{h}_i \in \mathcal{H}_i. && \text{(44d)}
\end{aligned}$$

To solve the above VUE's optimization problem, we use the distributed average delay-based approach for task fetching and offloading. We define the utility as follows:

$$u_i(t) = u_i(\alpha_i(t), \mathbf{h}_i(t), \boldsymbol{\omega}_i(t)) = -T_i^{\text{E2E}}(t). \tag{45}$$

The same distributed learning procedure as described in Section 4.2 is carried out at each VUE to find out the optimal decision policy $\Pr(\alpha_i|\mathbf{h}_i)$, knowing the channel state $\mathbf{h}_i$. Then we define the MEC server optimization problem for the baseline as follows:

$$\begin{aligned}
&\underset{P_i}{\text{minimize}} && \sum_{i\in\mathcal{V}_\text{o}} T_i^{\text{DL}} && \text{(46a)}\\
&\text{subject to} && \sum_{i\in\mathcal{V}_\text{o}} P_i = P_{\max} \text{ and } P_i \geq 0,\ \forall\, i \in \mathcal{V}_\text{o}. && \text{(46b)}
\end{aligned}$$

To find a closed form solution for the (46), the convexity of the objective function should be examined. We can simply the our problem as follows:

$$\underset{P_i}{\text{minimize}} \quad \sum_{i\in\mathcal{V}_\text{o}} \frac{\theta}{\log_2(1+P_i\kappa_i)}, \tag{47a}$$

where,

$$\theta = \frac{B|\mathcal{V}_\text{o}|}{W_\text{s}} \qquad \text{and} \qquad \kappa_i = \frac{h_{\text{s}i}|\mathcal{V}_\text{o}|}{W_\text{s}N_0}. \tag{48}$$

Let's take,

$$g(P_i) = \log_2(1+kP_i) \quad \text{and} \quad h(g(P_i)) = \frac{\theta}{\log_2(1+\kappa_i P_i)}. \tag{49}$$

Here $g(P_i)$ is a concave function of P_i and $h(g(P_i))$ is non-increasing and convex function of $g(P_i)$. Therefore, $f(P_i)$ is also a convex function [26] which is given by

$$f(P_i) = h(g(P_i)) = \frac{\theta}{\log_2(1+\kappa_i P_i)}. \tag{50}$$

Thus, Lagrangian and KKT conditions can be used to find a closed-form solution to the problem. Given that λ_i and μ are the Lagrangian multipliers, the Lagrangian of the problem is written as follows:

$$L(P_i, \lambda_i, \mu_i) = \sum_{i\in\mathcal{V}_\text{o}} \frac{\theta}{\log_2(1+\kappa_i P_i)} - \sum_{i\in\mathcal{V}_\text{o}} \lambda_i P_i + \mu \sum_{i\in\mathcal{V}_\text{o}} \Big(P_i - P_s\Big). \tag{51}$$

The KKT conditions include
1) Primal feasibility,

$$-P_i \leqq 0, \qquad \sum_{i\in\mathcal{V}_\mathrm{o}} P_i - P_{\max} = 0. \tag{52}$$

2) Dual feasibility,

$$\lambda_i \geq 0. \tag{53}$$

3) Complementary slackness,

$$\lambda_i(-P_i) = 0. \tag{54}$$

4) Equate the derivative of the Lagrangian with respect to P_i to 0, i.e.,

$$L'(P_i, \lambda_i, \mu_i) = \frac{-\kappa_i\theta}{(1+\kappa_i P_i)[\log_2(1+\kappa_i P_i)]^2} - \lambda_i + \mu = 0. \tag{55}$$

Our goal is to find the power allocation which satisfies (52)−(55).

$$\frac{-\kappa_i\theta}{(1+\kappa_i P_i)[\log_2(1+\kappa_i P_i)]^2} - \lambda_i + \mu = 0. \tag{56}$$

From (55), we can find $P_i \neq 0$. Otherwise, there do not exist any finite solutions for λ_i and μ. Thus, the only possibility is $P_i > 0$ with $\lambda_i = 0$ according to (53) and (54). Then, (55) is rewritten as

$$\mu = \frac{\kappa_i\theta}{(1+\kappa_i P_i)[\log_2(1+\kappa_i P_i)]^2}. \tag{57}$$

The power allocation is that the MEC server allocates the transmit power $P_i^* > 0, \forall\, i \in \mathcal{V}_\mathrm{o}$, which satisfies

$$\frac{\theta\kappa_i}{(1+P_i^*\kappa_i)[\log_2(1+P_i^*\kappa_i)]^2} = \mu. \tag{58}$$

Here, μ is chosen such that $\sum_{i\in\mathcal{V}_\mathrm{o}} P_i^* = P_{\max}$.

5 NUMERICAL RESULTS

In this chapter we compare the performance of the proposed approach and the other baselines. All approaches are implemented and simulated in MATLAB [27]. At the same time, the performance of the proposed approach is investigated varying different parameters such as the network density, computational capability of the server, computational capability of the VUE, server's bandwidth and camera's bandwidth. Simulation parameters are listed in Table 5.1.

Table 5.1. Simulation parameters

Parameter	Description	Value
V	Number of VUEs	60
C	Number of cameras	4
A	Camera's image size	20kbits
W_c	Camera's bandwidth	100kHz
W_s	Server's bandwidth	20MHz
f_i	VUE's CUP cycle frequency	1GHz
f_s	Server's CUP cycle frequency	200GHz
B	Synthesized image size	60kbits
N_0	Thermal noise	-174dBm
ρ	Risk sensitivity parameter	10
ξ	Temperature parameter	10

In the first baseline which is described in Section 4.3, each VUE's intention is to minimize its average E2E delay, while in the proposed approach, the VUE's intention is to minimize its average exponential E2E delay (i.e., risk).

Figure 5.1 shows the complementary cumulative distribution function (CCDF) of the E2E delay of the network for both proposed and baseline approaches. As shown in the figure, the tail of the delay distribution in the proposed approach decays faster. That means its occurrence probability of very high delay is lower compared to the baseline approach. In addition, we can straightforwardly find that the proposed approach has a lower standard deviation but a higher average performance. To further elaborate this, the standard deviation of the E2E delay and average E2E delay are plotted separately with respect to the VUE index in Figure 5.2.

In the proposed approach, all the statistics of the E2E delay distribution are taken into account in the risk minimization problem as per (19). Thus, the effect of minimization on the average E2E delay is less compared to baseline approach in which the only focus is on minimizing average E2E delay.

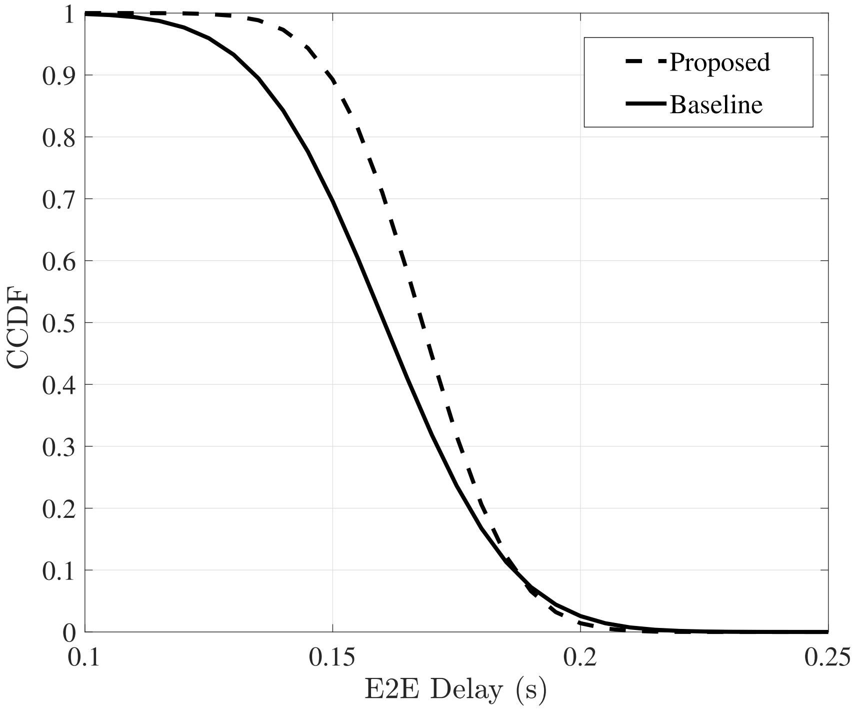
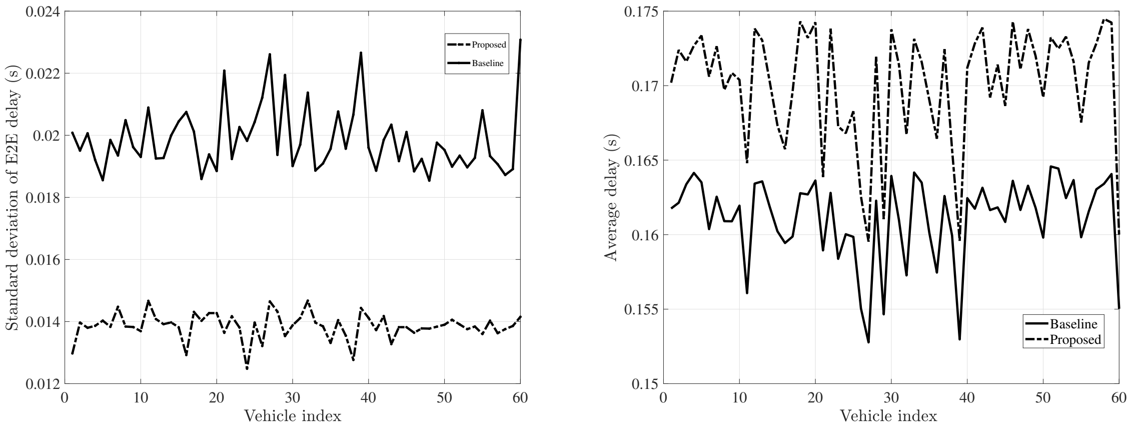


Figure 5.1. CCDF of both proposed and baseline approaches.

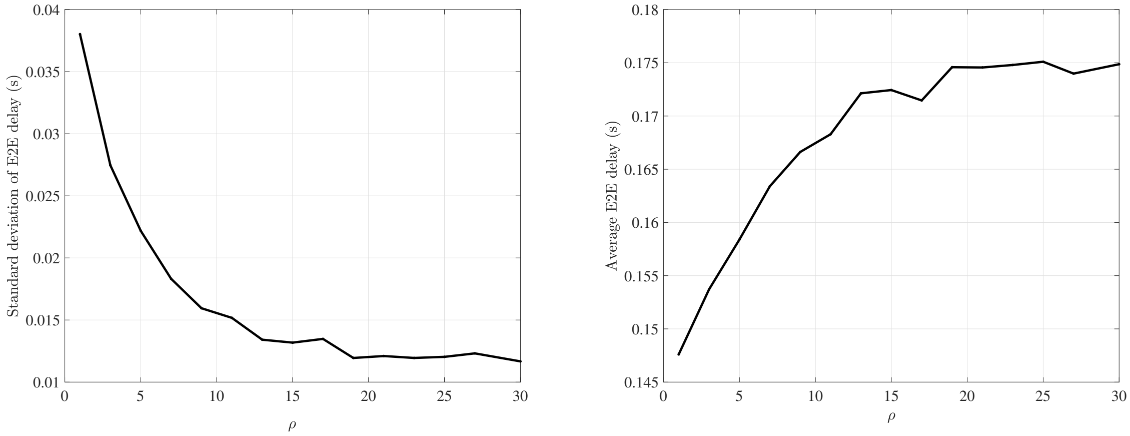


(a) Standard deviation

(b) Average E2E delay

Figure 5.2. Standard deviation of E2E delay and average E2E delay over vehicle index.

The risk-sensitive parameter ρ has a influence on the proposed approach as per (19). When ρ increases, the emphasis on standard deviation is higher. Therefore, standard deviation is a decreasing function of ρ. In contrast, when ρ is small, the average performance is lower since the emphasis on the higher-order statistics vanishes. Figure 5.3 shows how the standard deviation of the E2E delay and the average E2E delay vary with ρ.



(a) Standard deviation

(b) Average E2E delay

Figure 5.3. Variation of standard deviation and average E2E delay over ρ.

Now we analyze the system performance of the proposed approach varying the system settings. Moreover, we investigate how the VUE's geographic location affects the system performance.

Figure 5.4 shows the convergence of the decision of offloading ($\alpha_i = 1$) for particular VUE and for a particular state ($\mathbf{h}_i^l$) for given parameters in Table 5.1. The VUE has a higher probability of offloading, since the sever computational capability is higher compared to the VUE's computational capability. Therefore, the VUE can achieve low latency and high reliability by offloading rather than computing locally.

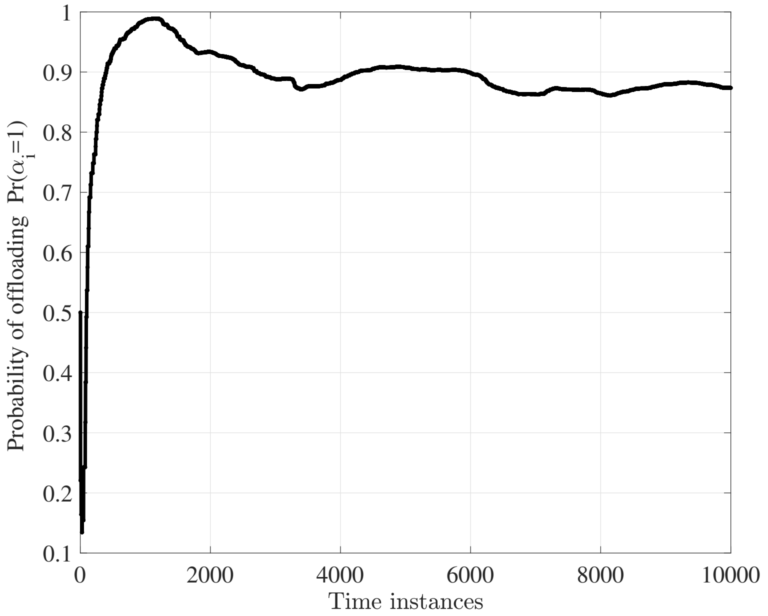


Figure 5.4. Convergence of the learning process.

Now we fix the distances of four VUEs to the server and the cameras as in Table 5.2 while letting VUEs be randomly located.

Table 5.2. A, B, C, D vehilces' locations

VUE	Distance to the cameras (m)	Distance to the server (m)
A	5	5
B	5	90
C	90	5
D	90	90

Firstly, we vary the number of VUEs in the network and observe how average E2E delay varies. As shown in Figure 5.5, average E2E delay of all four VUEs increase with the number of VUEs in the network. This is because when the number of VUEs in the network increases, the number of VUEs which access the server also increase. Then the server's computational and spectrum resources are shared among VUEs, resulting in higher computational and transmission delay. VUE A and C has the lowest average E2E delay among all four, when number of VUEs increases. This is because they are closer to the server than other two, and the path loss affects on the server-to-VUE transmission delay.

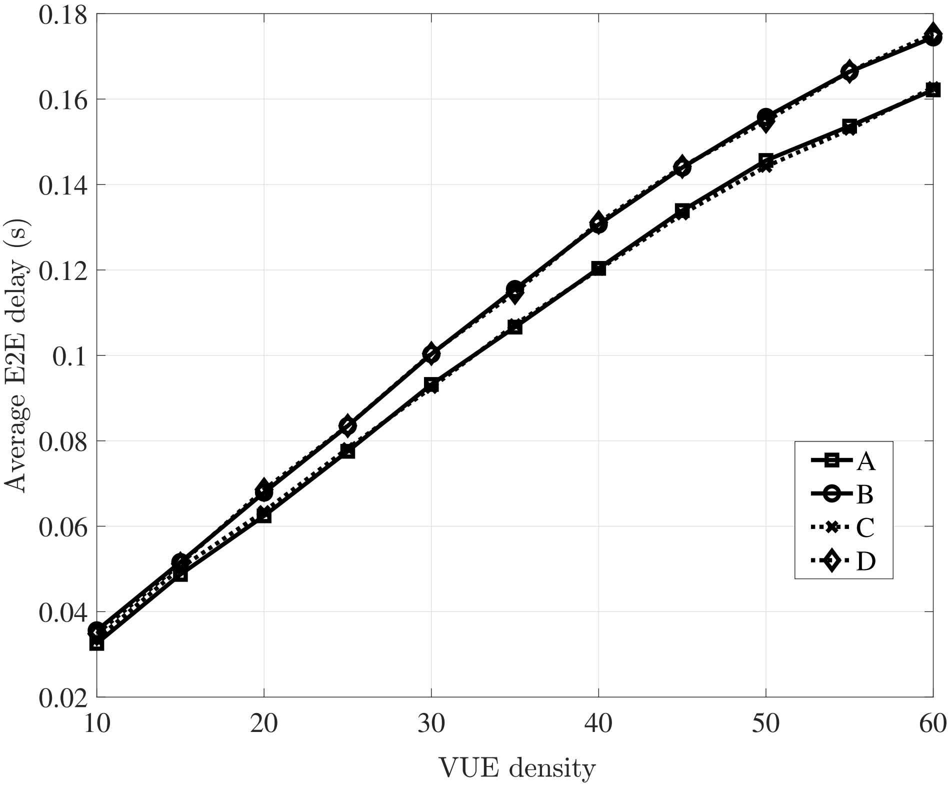
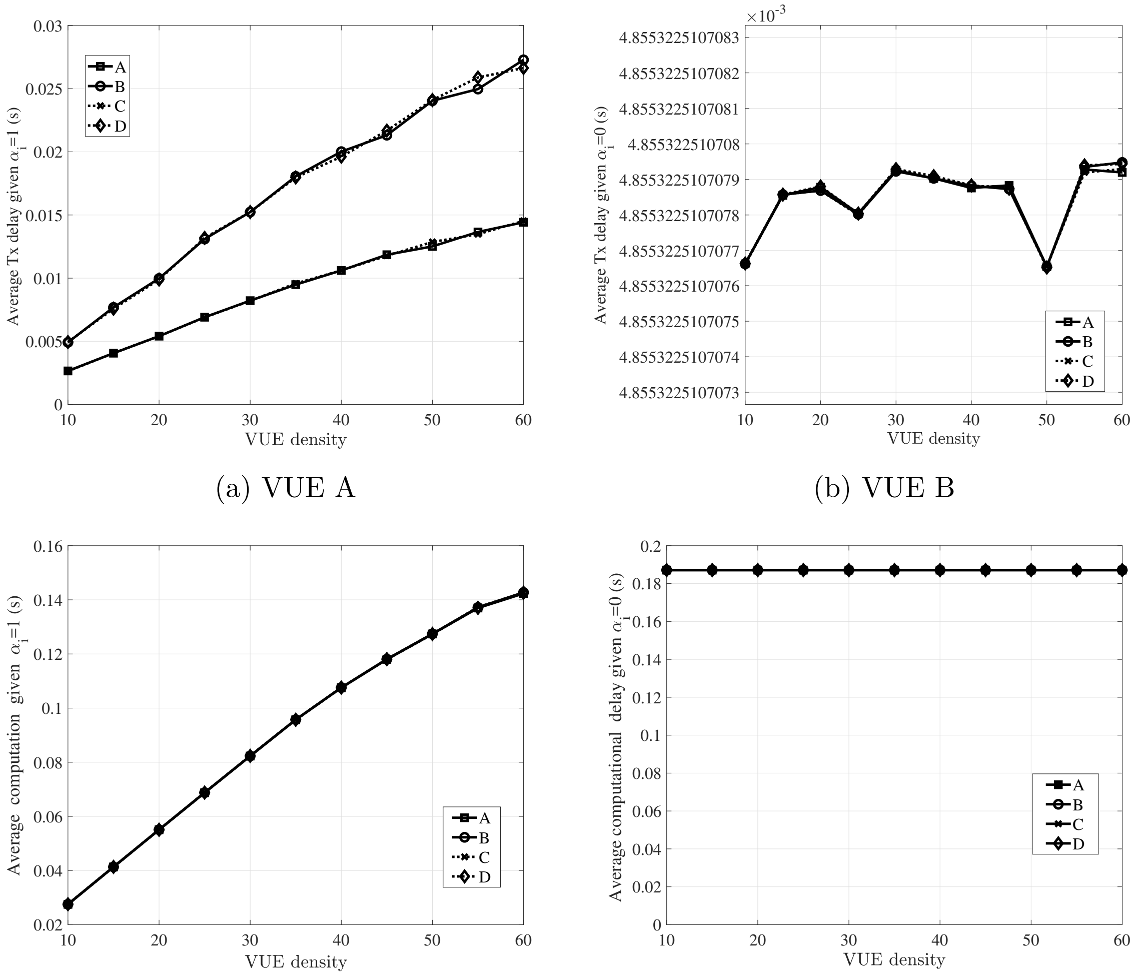


Figure 5.5. Variation of average E2E delay over VUE density

Subsequently, we break down average E2E delay into average computational delay and average transmission delay for both $\alpha_i = 1$ and $\alpha_i = 0$ cases and observe how these delays vary with the number of VUEs in the network. Figure 5.6 shows the variation of average transmission and computational delay for both cases for the VUEs A, B, C, and D. Average transmission delay given $\alpha_i = 1$ increases with the number of VUEs because transmission bandwidth is shared among the VUEs. Intuitively, when the distance to the server is higher, delay is higher. Average computational delay given $\alpha_i = 1$ linearly increases with the number of VUEs in the network but does not depend on the VUE's location. Average transmission delay given $\alpha_i = 0$ does not vary with number of VUEs because it only depends on the minimum channel gain among all the links between the cameras to VUEs. Since path loss is also taken into account when calculating the minimum channel gain, average transmission delay remains identical for all VUEs despite of the number of VUEs in the network.Average computational delay given $\alpha_i = 0$ does not depend on other network parameters, since VUE's computational capability is the same for all VUEs.



(a) VUE A

(b) VUE B

(c) VUE C

(d) VUE D

Figure 5.6. Variation of delay components over VUE density.

Next we fix the four VUEs A, B , C, and D at the same location and observe how converged probability of offloading $\Pr(\alpha_i = 1)$ changing over the network density. Figure 5.7 shows how the converged probability varies with the VUE density. When the number of VUEs in the network is low, server resources are under utilized, and VUEs have a higher probability of offloading. When the number of VUEs increases server resources are over-utilized. Thus, the probability of offloading decreases. In the figure, we can find that VUEs A and C, in general, have higher probabilities, since they are close to the server than others.

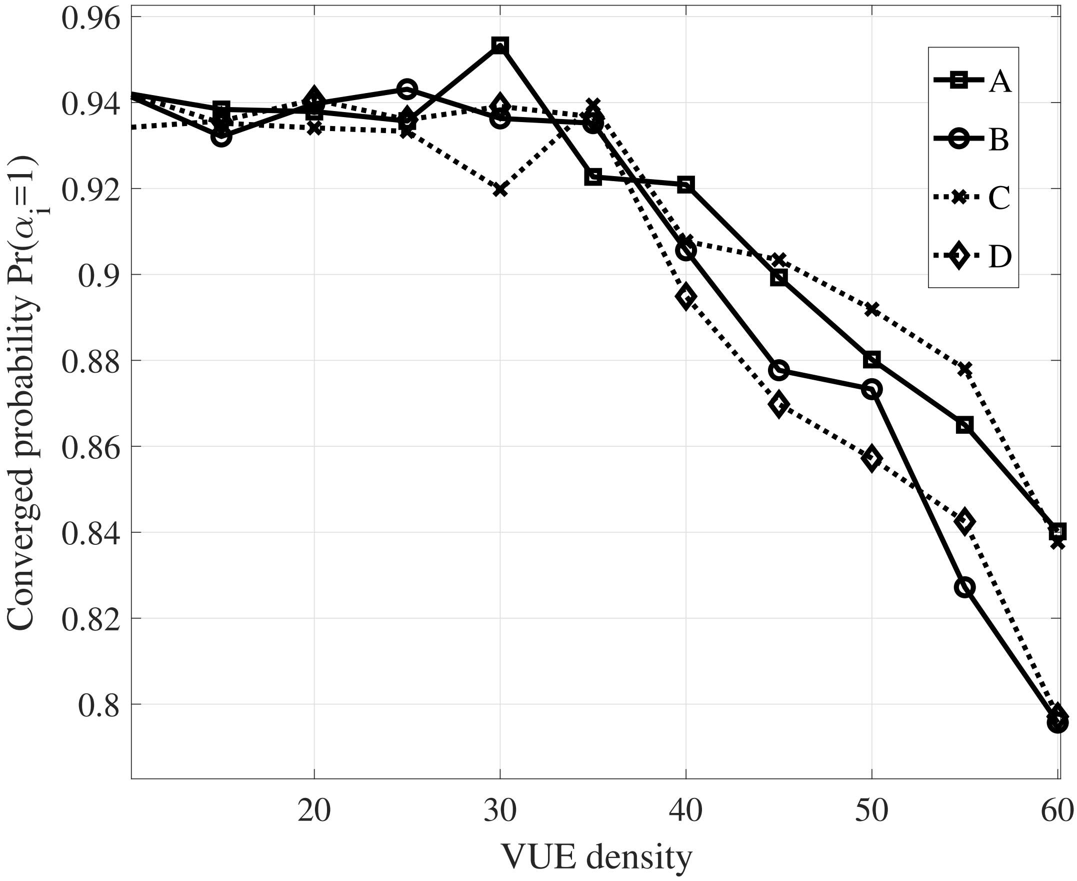


Figure 5.7. Variation of converged probability over VUE density

Now we see how converged probability $\Pr(\alpha_i = 1)$ changes with the computational power of the server. When computational power of the server is increased, VUEs have a higher probability of offloading than local computation. As shown in Figure 5.8, VUEs A and C have a logarithmic variation with computational power while VUEs B and D have a linear variation at low computational power region. When server's computational power is high, all VUEs' offloading probabilities converge to the same value.

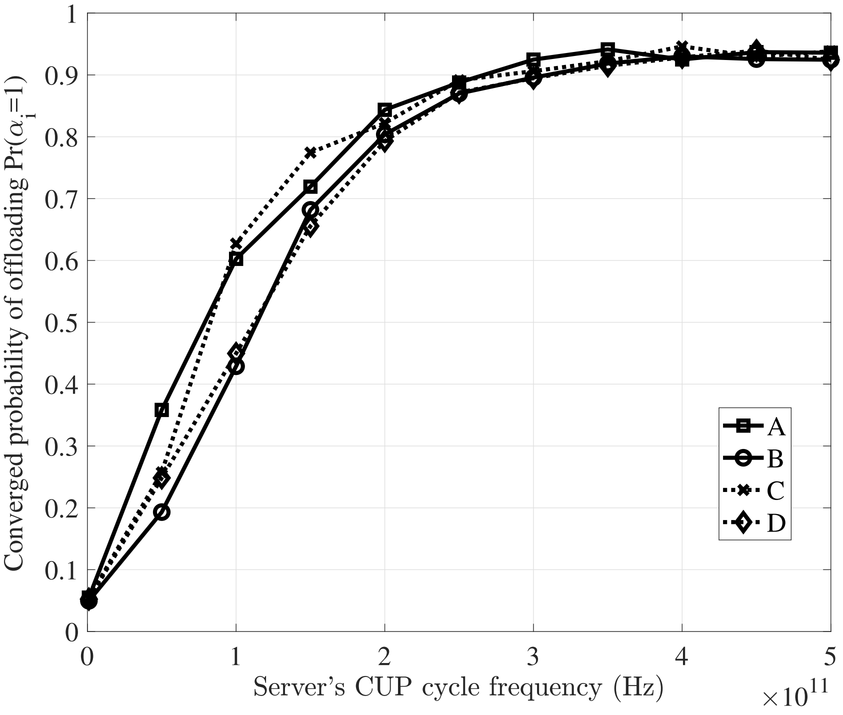


Figure 5.8. Variation of converged probability over server's computation power

Then we vary the server's bandwidth and observe how the converged probability of offloading changes. Figure 5.9 shows that when server bandwidth increases, VUEs have probabilities close to one despite of their locations. But when server's bandwidth is small, VUEs that are not close to the server (B and D) are interested in local computation rather than offloading.

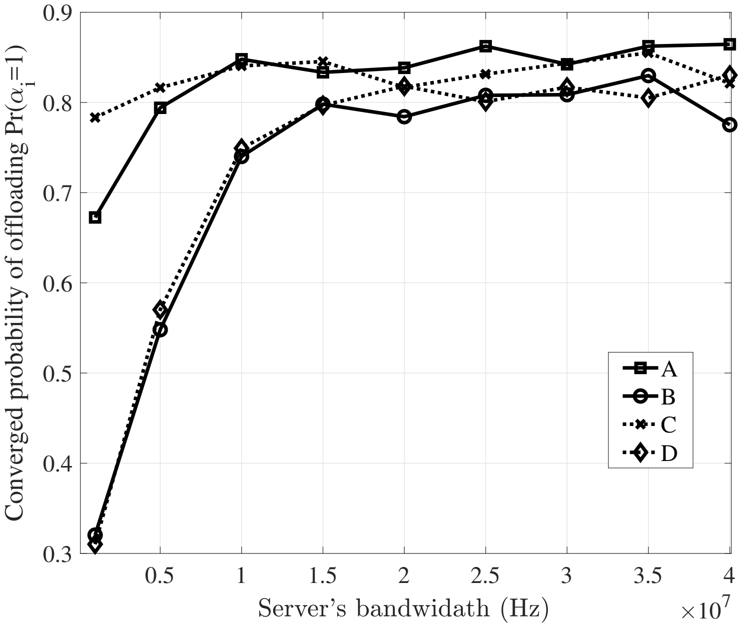


Figure 5.9. Variation of converged probability over server's bandwidth

Figure 5.10 shows the variation of the converged probability of offloading for the VUEs A, B, C, and D with the VUE's computational capability. When local computational power of the VUE is low, the probability of offloading is high. The probability of offloading decreases with the local computational capability as expected despite of VUEs' location.

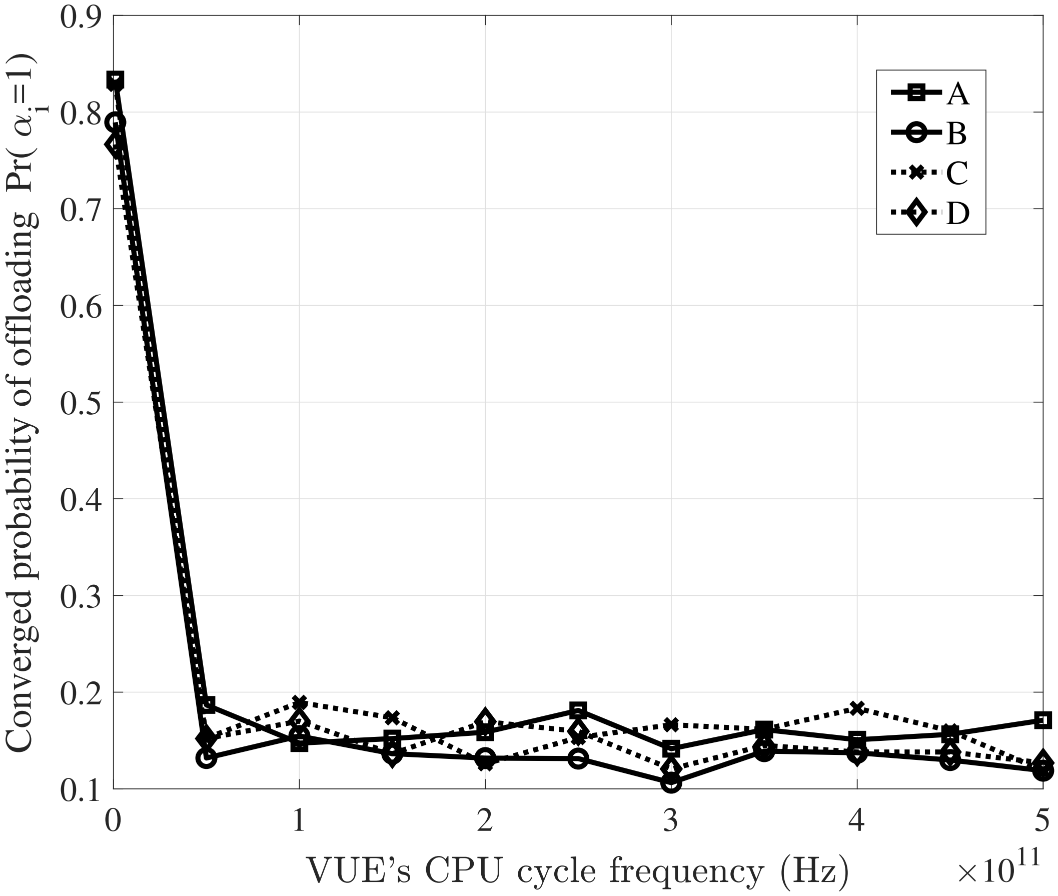


Figure 5.10. Variation of converged probability over VUE's computation power

Figure 5.11 shows the variation of the converged of probability of offloading for the VUEs A, B, C, and D with camera's bandwidth. Camera's bandwidth does not affect much on converged probability since transmission delay from camera to VUE is low compered to computational delay. However, the VUEs that are close to the cameras (B and D) have comparatively low probability for offloading since local computation is more beneficial for them.

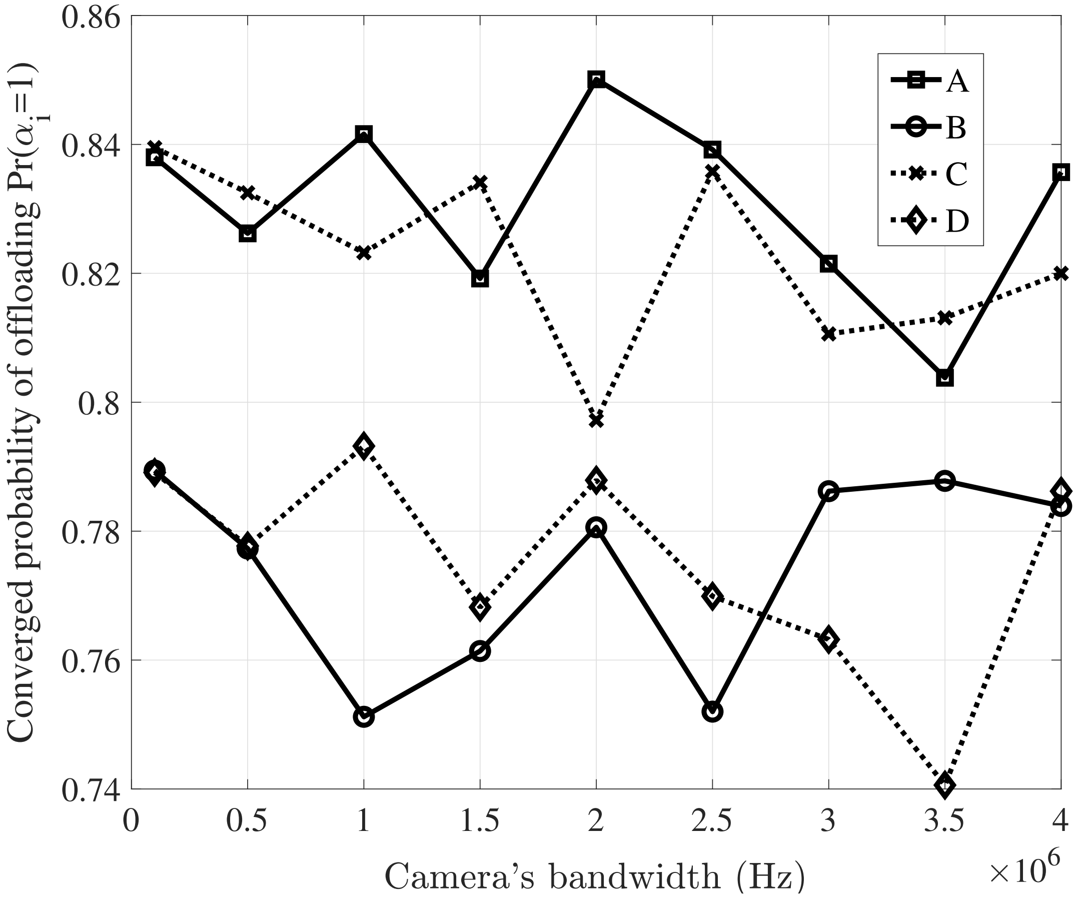
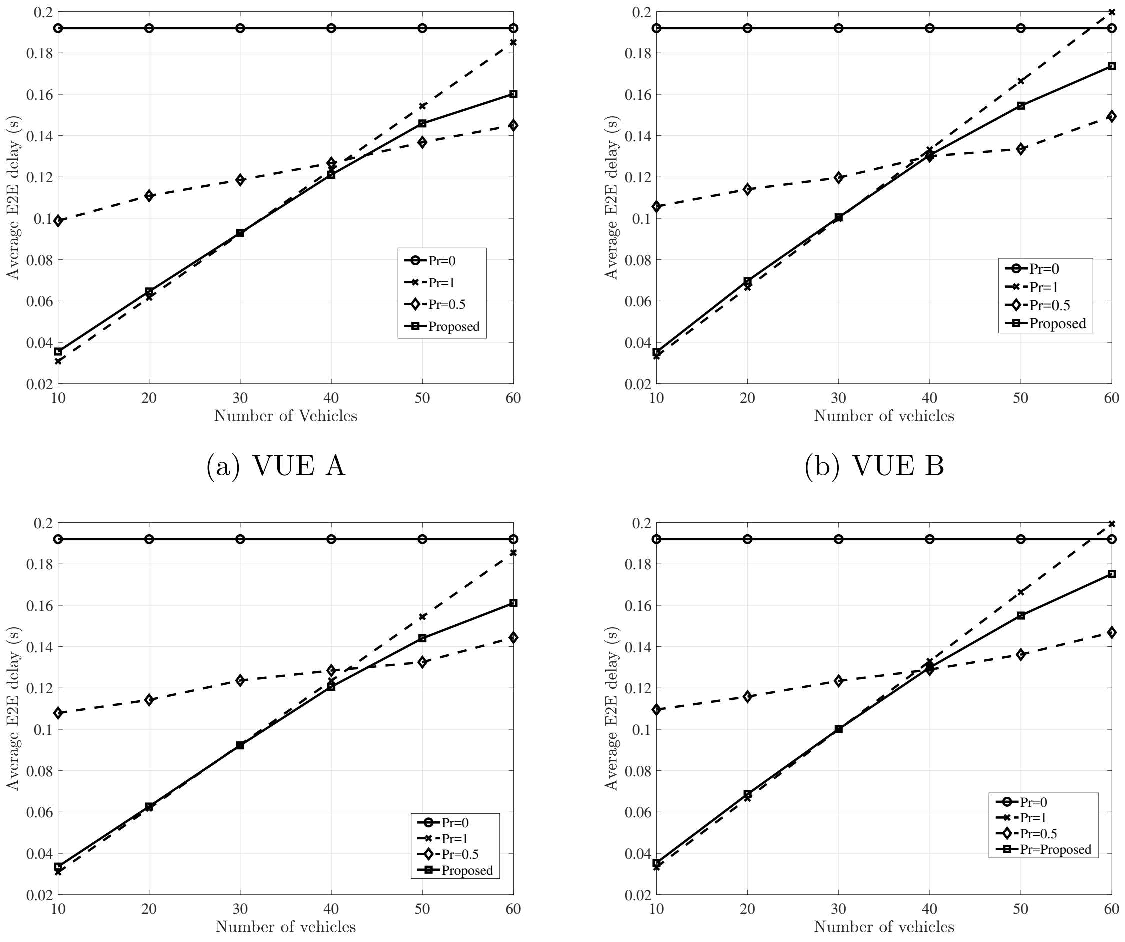


Figure 5.11. Variation of converged probability over camera's bandwidth

Furthermore, we compare our proposed approach with the other three baselines, in which probabilities of local computation and offloading are fixed with predetermined value. The three cases are shown as follows.

- All VUEs offload i.e., $\Pr(\alpha_i = 1) = 1$.
- All VUEs compute locally i.e., $\Pr(\alpha_i = 0) = 0$.
- All VUEs have the equal probabilities of offloading and local computation i.e., $\Pr(\alpha_i = 1) = \Pr(\alpha_i = 0) = 0.5$.

Figure 5.12 shows the performance of the proposed approach compared to three baselines for the VUEs A, B, C, and D by varying the VUE density of the network. When the number of vehicles in the network is lower than 40, fully-offloading in all VUEs can achieve the minimal E2E delay. The fully-fetching scheme in all VUEs has the maximal E2E delay. When the VUE density is high offloading E2E delay exceeds the delay for local computation for B and D VUEs which is far from the server, this is due to path loss. E2E delay of offloading linearly increases with number of VUEs, because computational delay when $\alpha_i = 1$, linearly depends on the number of VUEs. When VUE density is high, having the same probability of offloading and local computation i.e., $\Pr(\alpha_i = 1) = \Pr(\alpha_i = 0) = 0.5$ is also good. Our proposed approach performs well in all the VUE densities and despite of VUE's location.



(a) VUE A

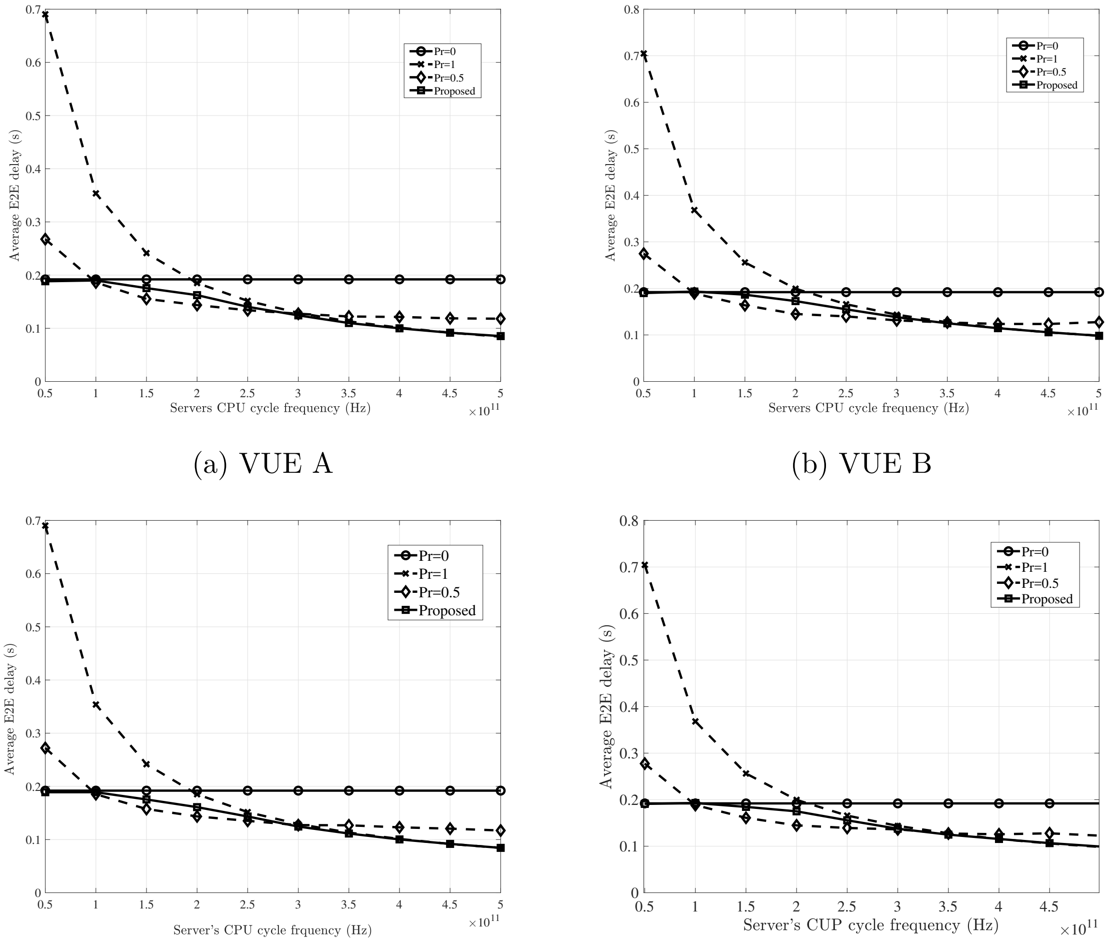
(b) VUE B

(c) VUE C

(d) VUE D

Figure 5.12. Variation of E2E delay of the VUEs A, B, C, and D with number of VUEs in the network.

Figure 5.13 shows performance of the proposed approach compared to the three baselines for the VUEs A, B, C, and D with different computational capabilities of the server. When server computational capability is weaker, fully-fetching scheme in all VUEs has the minimal average E2E delay. Server computational capability is high, all VUEs A, B, C, and D can achieve the minimum E2E delay by offloading. Most of the time following proposed approach, all the VUEs can archive the minimal E2E latency despite of its location in the network.



(a) VUE A

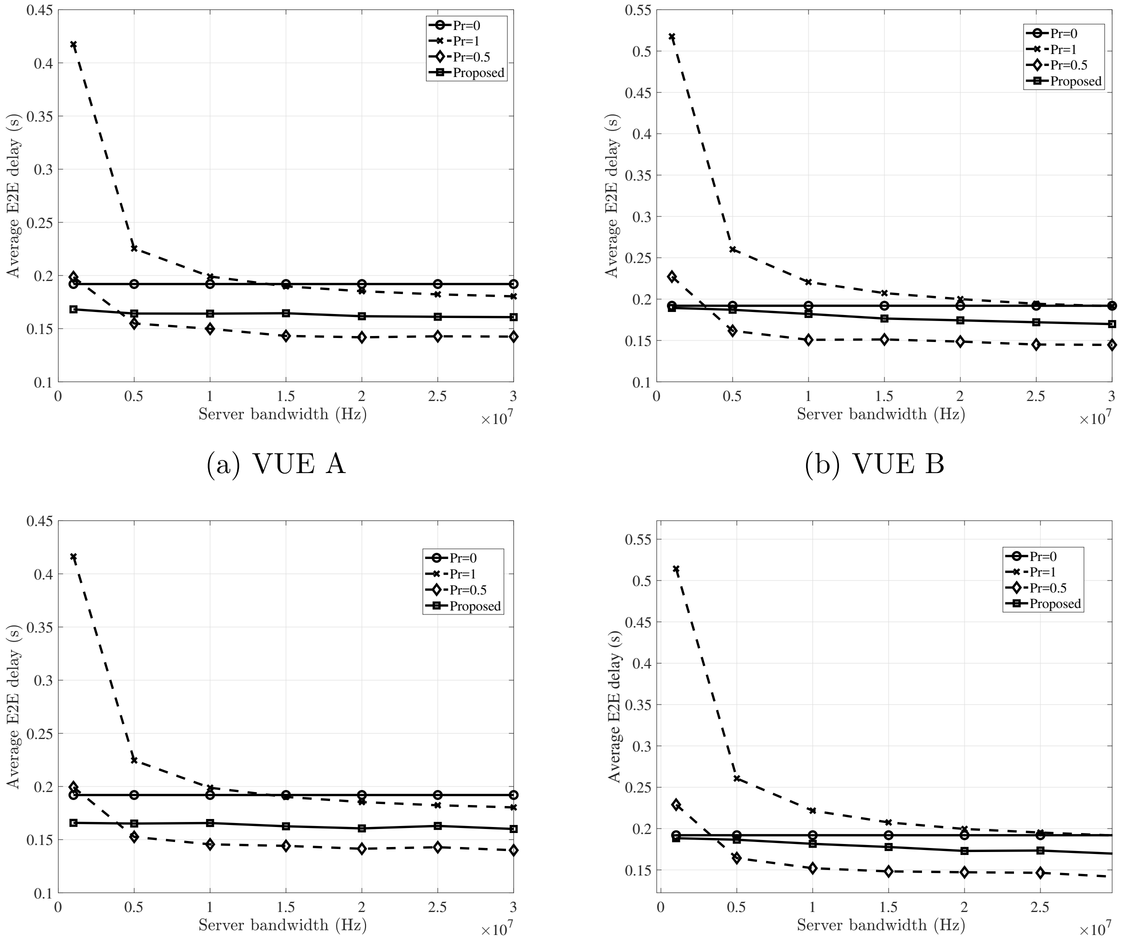
(b) VUE B

(c) VUE C

(d) VUE D

Figure 5.13. Variation of E2E delay of the VUEs A, B, C, and D with server's computational capability.

Figure 5.14 shows performance of the proposed approach compared to three baselines for the VUEs A, B, C, and D with the server's bandwidth. There is no significant effect from server bandwidth since computational delay is more dominant than the transmission delay. Minimum average E2E delay can be achieved when all the VUEs have the equal probabilities of offloading and local computation, i.e., $\Pr(\alpha_i = 1) = \Pr(\alpha_i = 0) = 0.5$. However, when the server bandwidth is low the proposed approach performs better. Server bandwidth increases, following the proposed approach, all four VUEs A, B, C, and D can archive a comparatively low E2E delay.



(a) VUE A

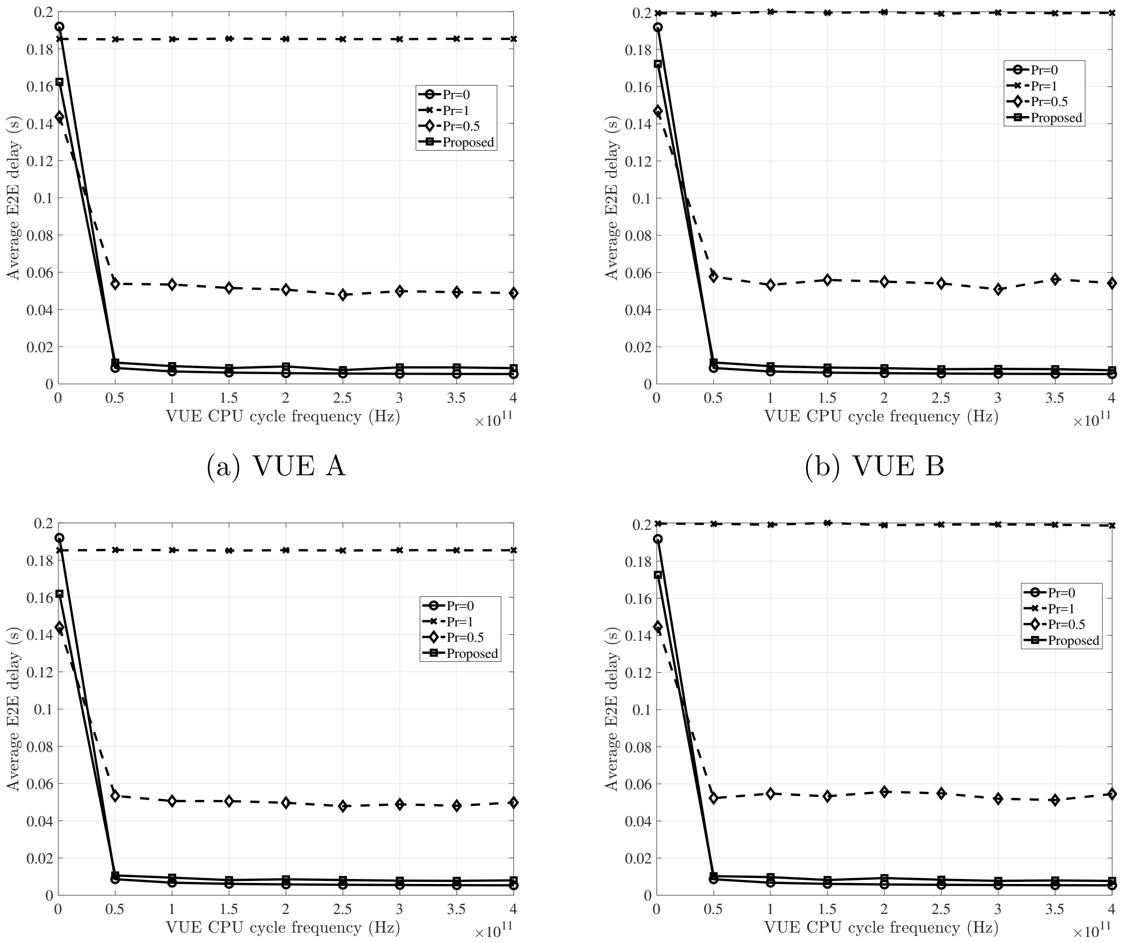
(b) VUE B

(c) VUE C

(d) VUE D

Figure 5.14. Variation of E2E delay of the VUEs A, B, C, and D with server's bandwidth.

Figure 5.15 shows performance of the proposed approach compared to three baselines for the VUEs A, B, C, and D with VUE's computational capability. If a VUE has a high computational power, there is no use of offloading. Minimum E2E delay can be achieved by fully-fetching scheme in all VUEs. In addition, the proposed approach also give the same performance that means the proposed approach suggests VUEs to compute locally. VUEs B and D have a higher E2E delay for offloading since they are far from the server.



(a) VUE A

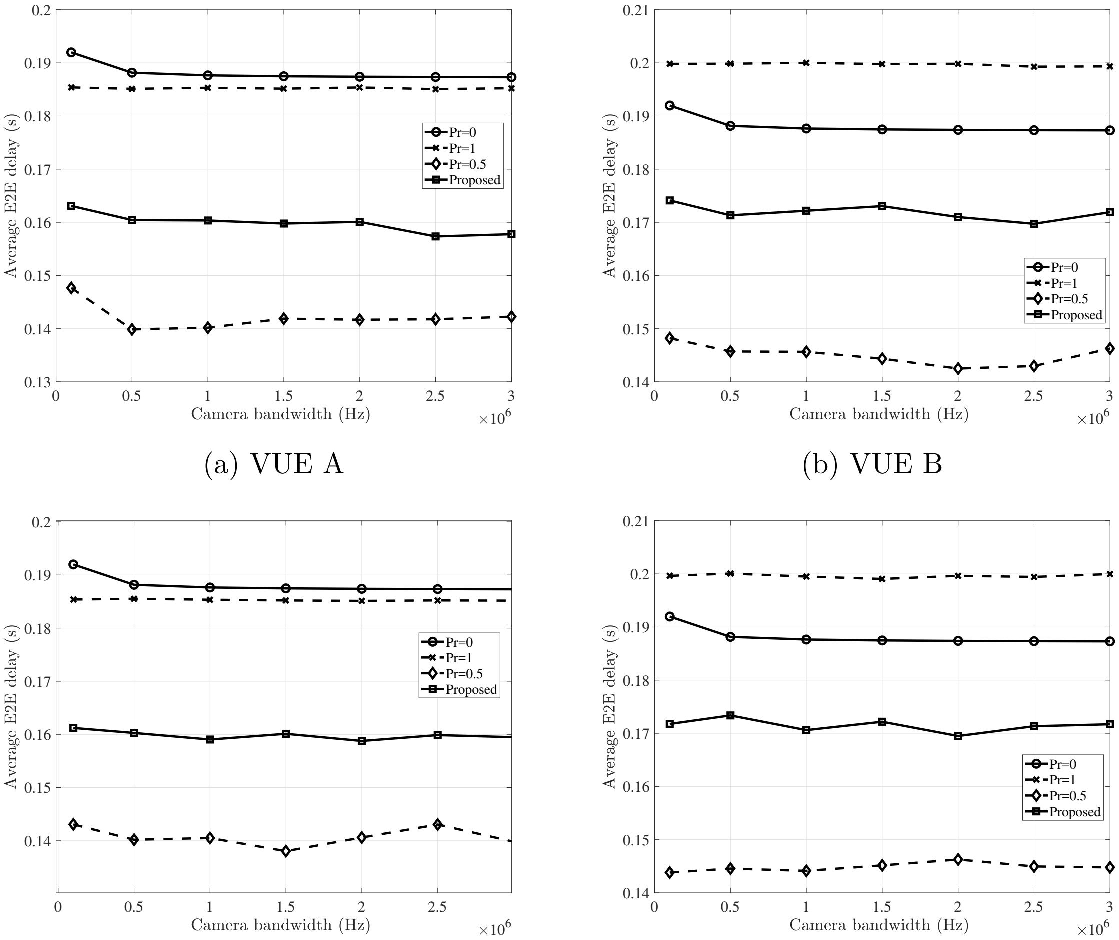
(b) VUE B

(c) VUE C

(d) VUE D

Figure 5.15. Variation of E2E delay of the VUEs A, B, C, and D with VUE's computational capability.

Figure 5.16 shows performance of the proposed approach compared to three baselines for the VUEs A, B, C, and D with VUE's bandwidth. Minimum average E2E delay can be achieved by the partially-offloading and fetching scheme. The proposed algorithm also has a comparatively less average E2E delay. The performance of the proposed approach is better when the VUE is close to the server.



(a) VUE A

(b) VUE B

(c) VUE C

(d) VUE D

Figure 5.16. Variation of E2E delay of the VUEs A, B, C, and D with camera's Bandwidth.

6 CONCLUSION

In this thesis we have proposed a novel approach to ensure low latency and high-reliability requirements of a vehicular edge computing network which consists of V2I communications and MEC considering an urban environment scenario. Our objective was to minimize the higher-order statistics of the E2E delay while jointly allocating the communication and computation resources in the considered vehicular edge computing scenario. A novel risk-sensitive distributed learning algorithm is proposed with minimum knowledge and no information exchange among VUEs, where each VUE learns the best decision policy to achieve low latency and high reliability.

We have compared the performance of the proposed method with the average-based baseline method and the simulation results show that the proposed approach improves the reliability while minimizing the end to end latency, so that stricter QoS requirements in V2I communication can be satisfied. The proposed approach is observed to have a better network-wide standard deviation of E2E delay and a comparable average E2E delay performance.

Simulations also investigate the performance of the proposed method when different system parameters such as the number of VUEs in the system, computational capability of the server, computational capability of the VUEs, server's bandwidth and camera's bandwidth etc. are changed. Simulation results shows the solidity of our proposed learning algorithm in different system parameters. VUE's location also has a effect on the decision taken by the VUE.

It can be seen that systems which has either local computation or offloading are very inefficient and can't satisfy the strict QoS requirements. Therefore, the proposed approach which has incorporated MEC with V2I is the best approach, and it has succeeded in allocating the communication and computation resources in an optimize way while VUEs are able to learn the best decision policy subject to latency and reliability constraints with minimum information and no information exchange among VUEs.

7 REFERENCES

[1] Khekare G.S. (2014) Design of emergency system for intelligent traffic system using vanet. In: International Conference on Information Communication and Embedded Systems (ICICES2014), pp. 1–7.

[2] Toor Y., Muhlethaler P., Laouiti A. & La Fortelle A.D. (2008) Vehicle ad hoc networks: applications and related technical issues. IEEE Communications Surveys Tutorials 10, pp. 74–88.

[3] Dar K., Bakhouya M., Gaber J., Wack M. & Lorenz P. (2010) Wireless communication technologies for its applications [topics in automotive networking]. IEEE Communications Magazine 48, pp. 156–162.

[4] W.H.O (2019), Road traffic injuries. URL: `https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries`, read 25.07.2019.

[5] Maslekar N., Boussedjra M., Mouzna J. & Labiod H. (2011) Vanet based adaptive traffic signal control. In: 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), pp. 1–5.

[6] Gradinescu V., Gorgorin C., Diaconescu R., Cristea V. & Iftode L. (2007) Adaptive traffic lights using car-to-car communication. In: 2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring, pp. 21–25.

[7] (2014) Mobile-edge computing– introductory technical white paper. Tech. Rep. Issue 1, European Telecommunications Standards Institute. URL: `https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1`.

[8] Xu X., Xue Y., Qi L., Yuan Y., Zhang X., Umer T. & Wan S. (2019) An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. Future Generation Comp. Syst. 96, pp. 89–100.

[9] Ku Y., Chiang P. & Dey S. (2018) Quality of service optimization for vehicular edge computing with solar-powered road side units. In: 2018 27th International Conference on Computer Communication and Networks (ICCCN), pp. 1–10.

[10] Zhou J., Tian D., Wang Y., Sheng Z., Duan X. & Leung V.C.M. (2019) Reliability-oriented optimization of computation offloading for cooperative vehicle-infrastructure systems. IEEE Signal Processing Letters 26, pp. 104–108.

[11] Zhou Z., Feng J., Tan L., He Y. & Gong J. (2018) An air-ground integration approach for mobile edge computing in iot. IEEE Communications Magazine 56, pp. 40–47.

[12] Hou X., Li Y., Chen M., Wu D., Jin D. & Chen S. (2016) Vehicular fog computing: A viewpoint of vehicles as the infrastructures. IEEE Transactions on Vehicular Technology 65, pp. 3860–3873.

[13] Xu C., Wang Y., Zhou Z., Gu B., Frascolla V. & Mumtaz S. (2018) A low-latency and massive-connectivity vehicular fog computing framework for 5g. In: 2018 IEEE Globecom Workshops (GC Wkshps), pp. 1–6.

[14] Zhou Z., Gao C., Xu C., Zhang Y., Mumtaz S. & Rodriguez J. (2018) Social big-data-based content dissemination in internet of vehicles. IEEE Transactions on Industrial Informatics 14, pp. 768–777.

[15] Wang X., Chen X., Wu W., An N. & Wang L. (2016) Cooperative application execution in mobile cloud computing: A stackelberg game approach. IEEE Communications Letters 20, pp. 946–949.

[16] Zhou Z., Yu H., Xu C., Zhang Y., Mumtaz S. & Rodriguez J. (2018) Dependable content distribution in d2d-based cooperative vehicular networks: A big data-integrated coalition game approach. IEEE Transactions on Intelligent Transportation Systems 19, pp. 953–964.

[17] Anpalagan A., Bennis M. & Vannithamby R. (2015) Design and deployment of small cell networks.

[18] Ashraf M.I. & C. Liu and M. Bennis and W. Saad (2017) Towards low-latency and ultra-reliable vehicle-to-vehicle communication. In: 2017 European Conference on Networks and Communications (EuCNC), pp. 1–5.

[19] Liu C. & Bennis M. (2018) Ultra-reliable and low-latency vehicular transmission: An extreme value theory approach. IEEE Communications Letters 22, pp. 1292–1295.

[20] Bennis M., Medina Perlaza S. & Debbah M. (2012) Learning Coarse Correlated Equilibria in Two-Tier Wireless Networks. In: IEEE ICC 2012, Ottawa, Canada, pp. 1592 – 1596.

[21] Samarakoon S., Bennis M., Saad W. & Latva-aho M. (2013) Backhaul-aware interference management in the uplink of wireless small cell networks. IEEE Transactions on Wireless Communications 12, pp. 5813–5825.

[22] Alam M., Sher M. & Husain S.A. (2009) Integrated mobility model (imm) for vanets simulation and its impact. In: 2009 International Conference on Emerging Technologies, pp. 452–456.

[23] Bai F., Sadagopan N. & Helmy A. (2003) The important framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks. Ad Hoc Networks 1, pp. 383–403.

[24] Goldsmith A. (2005) Wireless Communications. Cambridge University Press.

[25] Mao Y., You C., Zhang J., Huang K. & Letaief K.B. (2017) A survey on mobile edge computing: The communication perspective. IEEE Communications Surveys Tutorials 19, pp. 2322–2358.

[26] Boyd S. & Vandenberghe L. (2004) Convex Optimization. Cambridge University Press.

[27] MATLAB (2019) version 9.6.0.1072779 (R2019a). The MathWorks Inc., Natick, Massachusetts.