Modeling and Minimizing Latency in Three-tier V2X Networks

Phi Le Nguyen*, Ren-Hung Hwang†, Pham Minh Khiem*, Kien Nguyen‡, Ying-Dar Lin§
*School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam
†Department of Computer Science and Information Engineering, National Chung Cheng University, Chia-Yi, Taiwan, 621
‡Graduate School of Engineering, Chiba University, Chiba, Japan
§Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan
Email: *{lenp@soict, khiem.pm170084@sis}.hust.edu.vn, †rhhwang@cs.ccu.edu.tw, ‡nguyen@chiba-u.jp, §ydlin@cs.nctu.edu.tw

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9348182

***Abstract*—Leveraging mobile cloud computing (MCC) and mobile edge computing (MEC) for offloading computational tasks is a promising approach to enabling delay-sensitive applications executing vehicles. Despite MCC and MEC's ability and complementary characteristics, most of the existing works on offloading focus on only either MCC or MEC. In this paper, we study their cooperation in a three-tier offloading model of a V2X network where a vehicle can offload computational tasks to cloud computing and MEC. Specifically, we investigate the optimal offloading probabilities of three offloading paths, including Vehicle-to-Infrastructure, Vehicle-to-Cloud, and Infrastructure-to-Cloud. Our contribution is twofold. First, we derive a mathematical model of task execution latency and a formulation to find an optimal solution for the minimum latency problem. Second, we propose an approximation algorithm based on the genetic algorithm toward the optimum. The experiment results show that by exploiting both MCC and MEC's complementary advantages, our proposed algorithm in the three-tier model can shorten the delay significantly compared to existing two-tier models. Depending on the traffic load and the number of Road Side Units, our proposal can reduce the delay by 93.75% on the average, and 99.9% in the best case.**

***Index Terms*—V2X, offloading, minimum latency, 3-tier, MEC.**

I. Introduction

Along with broadband Internet development, more and more computation-, delay-sensitive applications, such as intelligent transport systems and autonomous driving, are being used on vehicles. The applications require substantial computation resources to process a massive volume of sensory data and real-time actions. This demand poses a significant challenge to resource-constrained vehicles that have limited computation capability. As a solution, the emerging mobile edge computing (MEC) technology has been introduced into Vehicle-to-Everything (V2X) networks. In the MEC-enabled V2X systems, each Road-Side Unit (RSU) has a collocated MEC server that can provide computing services. The vehicles can offload computation tasks to the RSU through the Vehicle-to-Infrastructure (V2I) information exchange path. Because the MEC servers are in the proximity of vehicles, the MEC technology potentially provides low latency, high-reliability computing services via computation offloading.

In the literature, many efforts have been devoted to investigating the computation offloading issue in V2X networks. In [1], the authors studied the problem of resource allocation. The work in [2] addressed the allocation of transmit power, bandwidth, and computation resource to obtain system performance gains. In [3], [4], [5], the focus was on hybrid offloading of V2I and V2V. Nevertheless, as the RSUs are limited by a certain level of computation and the communication range, using only MEC servers for offloading tasks may induce intractable execution delay, especially under circumstances of large numbers of the vehicles or tasks.

Mobile cloud computing (MCC) has been known as a promising approach for providing scalable computing resources anytime, anywhere. In MCC-assisted vehicular networks, vehicles' computations can be offloaded to the cloud server through the so-called Vehicle-to-Cloud (V2N) information exchange path. Since the MCC server possesses an enormous computation resource, it can handle massive computational tasks. However, due to its inherent characteristics of centralized deployment and long distance from the vehicles, using MCC solely to mitigate vehicles' computations incurs an inevitably significant transmission delay.

Motivated by the MEC and MCC's complimentary, we investigate a three-tier V2X model that enables both MEC and MCC in this paper. Specifically, the model includes three exchange paths: V2I, V2N, and Infrastructure-to-Cloud (I2N). Note that I2N is responsible for offloading tasks from the RSUs to the cloud server. In the model, vehicles may offload tasks to their RSU or to the cloud server (hereafter, we name the gNB). Moreover, the RSU may either locally process the submitted task or offload it to the gNB. After performing the task, the RSU and the gNB send the result back to the vehicle. Under this three-tier model, we study how to optimize the offloading probabilities between three tiers, i.e., vehicles, RSUs, and the gNB, to minimize all tasks' average execution latency. We focus on two below research questions:

1) How much can the three-tier architecture improve the performance compared to the two-tier architectures (vehicle-RSU; vehicle-gNB)?

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

Paper	Target network	Offloading paths	Variables	Objectives
[1]	2-tier	V2I, I2I	RA, OF	min(Energy)
[6]	2-tier	V2I, I2I	OF	max(Utility)
[3]	2-tier	V2I, V2V	OF	min(Cost)
[4]	2-tier	V2I, V2V	OF	min(Energy)
[5]	2-tier	V2I, V2V	OF	min(Latency)
[7]	2-tier	V2I, I2N	RA, OF	max(Utility)
Proposal	3-tier	V2I, V2N, I2N	OF	min(Latency)

TABLE I: Comparison of existing offloading strategies
RA: Resource allocation, *OF:* Offloading decision

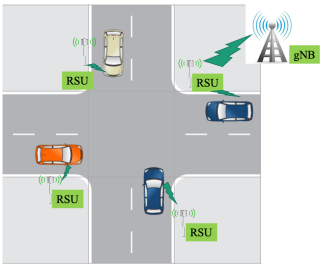


Fig. 1: The network model

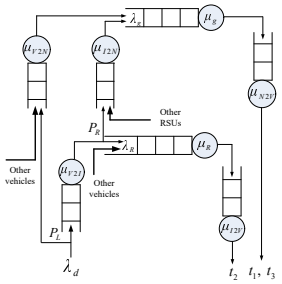


Fig. 2: The queueing model

2) How can optimal offloading probabilities affect the performance?

To the best of our knowledge, this is the first work addressing the offloading probability optimization in the three-tier V2X networks. The main contributions of this paper are as follows.

- We mathematically model the task execution latency in the three-tier model. We then derive an explicit formulation of the average latency as a function of the offloading probabilities. Moreover, we provide a mathematical formulation of the average latency minimization problem.
- We propose a genetic algorithm (GA) for approximately determining near-optimal offloading probabilities for the addressed problem.
- We extensively evaluate the three-tier model's effectiveness and the proposed optimal offloading probabilities determining algorithm by simulation.

The remainder of the paper is organized as follows. The next section briefly introduces the related works. In Section III, we describe the model of task execution latency and problem formulation. In Section IV, we propose a GA-based algorithm for determining near-optimal offloading probabilities. Section V includes performance evaluation. Finally, Section VI concludes the paper.

II. Related Work

Numerous works have studied MEC-based vehicular networks. In [1], Feng et al. jointly considered the computation offloading and resource allocation problems. Targeting at minimizing the power consumption of collaborative MEC servers and vehicles, the author divided the original problem into two sub-problems. The first one is how to optimally allocating the Ultra-reliable low-latency communication resource for multi-cells to multi-vehicles. The second one is determining the offloading decisions among local vehicles, a serving MEC server, and collaborative MEC servers. Similarly, the authors in [6] leveraged the collaboration between multiple MEC servers to achieve resource sharing. Specifically, they introduced a concept of double-MEC-layers, then exploit deep Q-learning to make the offloading decision for resource optimization.

Regarding V2V and V2I, the works in [3], [4], [5] addressed the offloading decision of collaborative task execution between platoons and a MEC server. Both [3], [4] studied how to decide whether the tasks should be executed locally, offloaded to the other platoon members, or the associated MEC server. However, [3] focused on minimizing the offloading cost, while [4] aimed to reduce the average total energy consumption. In [3], the authors first mathematically formulated the problem as a constrained shortest path problem on a directed acyclic graph. They then exploited the classical Lagrangian Relaxation-based Aggregated Cost (LARAC) algorithm to determine an optimal solution under the constraint of task execution deadlines. In [4], the authors leveraged Lyapunov algorithm to simplify the research objective and utilized the greedy approach to determine a sub-optimal solution. [5] aimed at reducing offloading latency between the vehicles, each of which necessarily maintains the information of its 1-hop neighbors. Whenever having a task, the vehicle calculates the offloading latency for all the relay hop candidates. Then, the neighbor vehicle with the minimum latency is chosen as the best relay node. We give a summary of related works in Table I. Although many works have been done, most of them rely on only the MEC servers for computation offloading.

Recently, Zhao et al. [7] tried to utilize the MEC and cloud computing resources simultaneously for offloading via two paths: V2I and I2N. Specifically, vehicles may offload their computation tasks to the MEC server or the cloud server through RSUs. The objective is to maximize the system utility by optimizing both the offloading strategy and resource allocation. Unlike previous works, this paper investigates a three-tier model that provides three offloading paths: V2I, V2N, and I2N. Our objective is to determine the offloading probabilities to minimize task execution latency.

III. Theoretical analysis

A. System modeling

1) Preliminaries: Figure 1 shows our investigated V2X network model. We consider an urban area, where N Roadside Units (RSUs) is located on the roadsides. The vehicles, each of which has computational tasks, run along the road. The tasks can be offloaded to either an RSU or the gNB. Moreover, the RSU may perform the offloaded task locally or forward it to the gNB. After completing the task, the RSU and the gNB send the result back to the vehicle. Accordingly, there are three types of tasks in the model: (1) tasks are directly offloaded to the gNB, (2) tasks are offloaded to the RSU and then forwarded to gNB, and (3) task are offloaded to RSUs. Example scenarios for type (1), (3) are intuitive, but the one

of type (2) could be as follows. A vehicle offloads a task to an RSU, but then leaves the RSU's coverage before the task is processed. Thus, the RSU forwards the task to the gNB as gNB could send back the result to the vehicle. Our objective is to determine the optimal values of the offloading probabilities to minimize all the tasks' average execution delay. A task's total execution latency comprises two components: the transmission delay and the processing delay. For the type (1) (and type (3)) tasks, the transmission delay is the time for transmitting packets from the vehicles to the gNB (and the RSU), and vice versa. In type (2), the transmission delay includes three transmission periods: vehicle-to-RSU, RSU-to-gNB, and gNB-back-to-vehicle. The processing time is time for performing the task at the RSU (concerning type (3)) or the gNB (concerning type (1) and (2)). Now, we define the notations and assumptions used throughout this paper. We denote by P_L and P_R the probability by that the vehicles and the RSUs offload to the gNB. We assume that the vehicles enter and leave an RSU's coverage according to the Poisson process with the arrival rate of λ_v and the departure rate of μ_v. The task generation at each vehicle also follows the Poisson distribution with the arrival rate of λ_d. We assume that the task processing time at RSUs and the gNB are exponentially distributed with the service capacities of each RSU and the gNB, denoted as μ_R and μ_g, respectively. When tasks are offloaded to RSUs or gNB, they are transmitted through communication links. Based on the above assumptions, tasks arrive at the links follow a Poisson process. We also assume the communication delays are exponentially distributed. The link capacities from a vehicle to an RSU and vice versa are denoted by μ_{V2I} and μ_{I2V}, respectively. Similarly, we denote the bandwidths for the vehicle-gNB links by μ_{V2N} and μ_{N2V}. The RSU-gNB link's capacities are denoted by μ_{I2N} and μ_{N2I}. Moreover, we use t_1, t_2, t_3 to denote the average latency of tasks belonging to type (1), (2) (3), respectively. The queuing model for our system is illustrated in Fig. 2.

2) Derivation of the latency: In the following, we derive the average latency of three task types. The average number of vehicles associated with an RSU is given by $E[V] = \frac{\rho_v}{1-\rho_v}$, where $\rho_v = \frac{\lambda_v}{\mu_v}$. As the probability for direct offloading to the gNB is P_L, vehicle offload tasks to RSUs with a probability of $1 - P_L$. Accordingly, the distribution of packets offloaded to an RSU can be seen as a Poisson process with the arrival rate of $\lambda_{V2I} = \frac{\rho_v}{1-\rho_v}\lambda_d\left(1-P_L\right)$, where $\frac{\rho_v}{1-\rho_v}\left(1-P_L\right)$ is the number of vehicles associated with an RSU; λ_d is the task arrival rate at each vehicle. As each vehicle offloads tasks to the gNB with the probability of P_L, the arrival rate at the vehicle-to-gNB link is $\lambda_{V2N} = \frac{\rho_v}{1-\rho_v}\lambda_d N P_L$. Similarly, the arrival rate at the RSU-to-gNB link is $\lambda_{I2N} = \frac{\rho_v}{1-\rho_v}\lambda_d N\left(1-P_L\right)P_R$. Hence, the packets at the gNB can be seen as a Poisson process with the arrival rate of $\lambda_g = \frac{\rho_v}{1-\rho_v}\lambda_d N P_L + \frac{\rho_v}{1-\rho_v}\lambda_d N\left(1-P_L\right)P_R$, where $\frac{\rho_v}{1-\rho_v}\lambda_d N P_L$ is the arrival rate of tasks offloaded by all vehicles, and $\frac{\rho_v}{1-\rho_v}\lambda_d N\left(1-P_L\right)P_R$ is the arrival rate of tasks offloaded by all RSUs. Let λ_{I2V} be the arrival rate of traffic from RSU to vehicles, then λ_{I2V} can be calculated as $\lambda_{I2V} = \frac{\rho_v}{1-\rho_v}\lambda_d\left(1-P_L\right)\left(1-P_R\right)$. Assuming that the tasks are homogeneous, we denote p, q as the size of the packet conveying a task, the one containing the results of a task, respectively.

Type 1: Tasks that are offloaded directly to the gNB

The transmission time spent to offload a task from a vehicle to the gNB is given by

$$t_{V2N} = \frac{1}{\mu_{V2N} - \lambda_{V2N}} = \frac{1}{\mu_{V2N} - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L}. \quad (1)$$

The time for handling a task at the gNB is calculated by $t_g = \frac{1}{\mu_g - \lambda_g}$. By substituting the value of μ_g, we have

$$t_g = \frac{1}{\mu_g - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L - \frac{\rho_v}{1-\rho_v}\lambda_d N\left(1-P_L\right)P_R}. \quad (2)$$

The transmission time from gNB to the vehicle is defined as $t_{N2V} = \frac{1}{\mu_{N2V} - \lambda_{N2V}}$, where λ_{N2V} is the arrival rate of packets from gNb to the vehicles. As the results of all task offloaded from both RSU and vehicle to the gNB then will be sent back to the vehicle, λ_{N2V} equals to λ_g. Consequently, we have

$$t_{N2V} = \frac{1}{\mu_{N2V} - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L - \frac{\rho_v}{1-\rho_v}\lambda_d N\left(1-P_L\right)P_R}. \quad (3)$$

From (1), (2) and (3), the execution latency for the tasks belonging to Type 1 is represented as

$$\begin{aligned} t_1 = & \frac{1}{\mu_{V2N} - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L} \\ & + \frac{1}{\mu_g - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L - \frac{\rho_v}{1-\rho_v}\lambda_d N\left(1-P_L\right)P_R} \\ & + \frac{1}{\mu_{N2V} - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L - \frac{\rho_v}{1-\rho_v}\lambda_d N\left(1-P_L\right)P_R}. \end{aligned} \quad (4)$$

Type 2: Tasks that are offloaded to the RSU and then processed at the RSU

The transmission time from the vehicle to the RSU is $t_{V2I} = \frac{1}{\mu_{V2I} - \lambda_{V2I}} = \frac{1}{\mu_{V2I} - \frac{\rho_v}{1-\rho_v}\lambda_d\left(1-P_L\right)}$. Let us denote by μ_R the service rate, and μ_R the arrival rate of the tasks at the RSU. The processing time of a task at the RSU is calculated by $t_R = \frac{1}{\mu_R - \lambda_R}$. As the arrival rate of tasks offloaded to a RSU is $\frac{\rho_v}{1-\rho_v}\left(1-P_L\right)\lambda_d$ and the probability that the RSU transfers a task to the gNB is P_R, the arrival rate of packets being processed by the RSU, i,e, λ_R is $\lambda_R = \frac{\rho_v}{1-\rho_v}\lambda_d\left(1-P_L\right)\left(1-P_R\right)$. The transmission time from the RSU to the vehicles is defined by $t_{I2V} = \frac{1}{\mu_{I2V} - \lambda_{I2V}} = \frac{1}{\mu_{I2V} - \frac{\rho_v}{1-\rho_v}\lambda_d(1-P_L)(1-P_R)}$. In consequence, the average latency of tasks belonging to Type 2 is deduced as

$$\begin{aligned} t_2 = & \frac{1}{\mu_{V2I} - \frac{\rho_v}{1-\rho_v}\lambda_d\left(1-P_L\right)} + \frac{1}{\mu_R - \frac{\rho_v}{1-\rho_v}\lambda_d\left(1-P_L\right)\left(1-P_R\right)} \\ & + \frac{1}{\mu_{I2V} - \frac{\rho_v}{1-\rho_v}\lambda_d\left(1-P_L\right)\left(1-P_R\right)}. \end{aligned} \quad (5)$$

Type 3: Tasks that are offloaded to the RSU and then forwarded

to the gNB

At the RSU, the packet has to be queued and then offloaded to the gNB. Therefore, the time from the RSU to gNB can be considered as

$$t_{I2N} = \left(\frac{1}{\mu_R - \lambda_R} - \frac{1}{\mu_R}\right) + \frac{1}{\mu_{I2N} - \lambda_{I2N}}. \quad (6)$$

where $\frac{1}{\mu_R - \lambda_R} - \frac{1}{\mu_R}$ is the waiting time at the RSU, and $\frac{1}{\mu_{I2N} - \lambda_{I2N}}$ is the transmission time from RSU to the gNB. By substituting the value of λ_{I2N} and λ_R into (6), we have

$$t_{I2N} = \left(\frac{1}{\mu_R - \frac{\rho_v}{1-\rho_v}\lambda_d (1-P_L)(1-P_R)} - \frac{1}{\mu_R}\right) + \frac{1}{\mu_{I2N} - \frac{\rho_v}{1-\rho_v}\lambda_d N (1-P_L) P_R}. \quad (7)$$

The total delay for the Type 3's tasks is defined as $t_3 = t_{V2I} + t_{I2N} + t_g + t_{N2V}$. Therefore, from (7), (2), and (3), we have

$$t_3 = \frac{1}{\mu_{V2I} - \frac{\rho_v}{1-\rho_v}\lambda_d (1-P_L)} + \left(\frac{1}{\mu_R - \frac{\rho_v}{1-\rho_v}\lambda_d (1-P_L)(1-P_R)} - \frac{1}{\mu_R}\right) + \frac{1}{\mu_{I2N} - \frac{\rho_v}{1-\rho_v}\lambda_d N (1-P_L) P_R} + \frac{1}{\mu_g - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L - \frac{\rho_v}{1-\rho_v}\lambda_d N (1-P_L) P_R} + \frac{1}{\mu_{N2V} - \frac{\rho_v}{1-\rho_v}\lambda_d N P_L - \frac{\rho_v}{1-\rho_v}\lambda_d N (1-P_L) P_R}. \quad (8)$$

B. Problem statement

Our objective is to determine optimal offloading probabilities for the vehicles and the RSUs to minimize all tasks' average delay. The probability for a packet belonging Type 1, Type 2, Type 3 are P_L, $(1-P_L)(1-P_R)$ and $(1-P_L)P_R$, respectively. Therefore, the average delay of all tasks is determined by

$$\bar{t} = P_L t_1 + (1-P_L)(1-P_R) t_2 + (1-P_L) P_R t_3.$$

Consequently, our optimization problem can be mathematically formulated as follows.

Minimize

$$P_L t_1 + (1-P_L)(1-P_R) t_2 + (1-P_L) P_R t_3 \quad (9)$$

Subject to

$$\frac{\rho_v}{1-\rho_v}(1-P_L)\lambda_d \le \mu_{V2I} \quad (10)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d (1-P_L)(1-P_R) \le \mu_{I2V} \quad (11)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d P_L N \le \mu_{V2N} \quad (12)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d N P_L + \frac{\rho_v}{1-\rho_v}(1-P_L)\lambda_d N P_R \le \mu_{N2V} \quad (13)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d (1-P_L) P_R N \le \mu_{I2N} \quad (14)$$

$$\frac{\rho_v}{1-\rho_v}(1-P_L)(1-P_R)\lambda_d \le \mu_R \quad (15)$$

$$\frac{\rho_v}{1-\rho_v}\lambda_d N P_L + \frac{\rho_v}{1-\rho_v}(1-P_L)\lambda_d N P_R \le \mu_g \quad (16)$$

Constraints (10), (11), (13) and (14) shows that the total traffic offloaded should not exceed the links' bandwidth. Specifically, (10) and (11) depict the constraint concerning the links between the vehicles and RSU; (13) refers to the vehicle to the gNB's link; and (14) represents the constraint regarding the link from gNB to the vehicle. (15) and (16) depict that the task arrival rates at the RSUs and the gNB should not exceed their service rates.

IV. GA-BASED OPTIMIZATION ALGORITHM

We propose a GA-based approximation algorithm to determine the optimal solution for the problem formulated in the previous section.

A. Chromosome representation, fitness function, and initialization

A chromosome consists of two genes representing the values of P_L and P_R, respectively. The fitness value is the average delay obtained when applying P_L and P_R, which can be calculated by using formula (9). We aim to determine the individual with a minimal fitness value and satisfying all constraints from (10) to (16). We randomly initialize N individuals, where N is a tunable parameter. To increase the population's diversity, we create 70% of individuals that satisfy the constraints from (10) to (16) and 30% of the individuals that do not. Moreover, we also include the boundary values (i.e., $[0,0], [1,1], [0,1], [1,0]$) to diversify the initial population.

B. Crossover and mutation

We propose a crossover algorithm that combines two approaches: *average crossover* and *swap crossover*. In the following, we denote $I_1 = \{I_1.x, I_1.y\}$, $I_2 = \{I_2.x, I_2.y\}$ as the two parents. Note that x and y represent the values of P_L and P_R, respectively. In the *average crossover*, we create a random number α in the range $[0,1]$ and take the α-weighted average of the parents. Specifically, the genes of the offspring $O = \{O.x, O.y\}$ is defined as: $O.x = \alpha I_1.x + (1-\alpha) I_2.x$ and $O.y = \alpha I_1.y + (1-\alpha) I_2.y$. In the *swap crossover*, we swap the parents' two genes and select among the two offsprings the one with better fitness value. Specifically, we first generate two offsprings $O_1 = \{I_1.x, I_2.y\}$ and $O_1 = \{I_2.x, I_1.y\}$. Then, we select among O_1 and O_1, the one whose fitness value is smaller. Consequently, by applying both *average crossover* and *swap crossover*, we obtain two offsprings after performing the crossover operation. Concerning the mutation operation, we mutate an individual by taking the individual's average with a random individual in the range of $[0,1]$. Specifically, let $I = \{I.x, I.y\}$ be the individual that will be mutated, then the mutated offspring $O = \{O.x, O.y\}$ is defined by $O.x = (x_0 + I.x)/2$ and $O.y = (y_0 + I.y)/2$. where x_0, y_0 are two random numbers belonging to the range $[0,1]$.

C. Selection and termination condition

Note that the offsprings obtained by the crossover and mutation operations may not satisfy the constraints. We do not remove these offsprings from the population but assign them a significantly tremendous fitness value. The reason is to maintain the diversity of the population. After performing crossover and mutation operations, we select N individuals whose fitness values are smallest among the current population. The algorithm terminates when the number of generations reaches a predefined number.

V. Performance evaluation

In this section, we compare the 2-tier and 3-tier models and investigate how much the offloading probabilities determined by our proposal can improve the 3-tier model's performance. We perform the evaluations with various configurations of P_L and P_R, as shown in Table II. In total, we have two 2-tier models and six 3-tier models. We conduct three experiments to see the impacts of the packet arrival rate, the number of the RSUs, and the vehicle arrival rates on the average delay. We build an in-house simulator that is written in Java language for the experiments. In our simulation, the vehicle arriving, packet processing, and packet transmission processes follow the Poisson process. Other parameters are presented in Table III, in which the configurations of the gNB, RSU, and vehicles are suggested by [8]. Moreover, the RSUs are placed evenly in a straight-lined road.

A. 2-tier vs. 3-tier and RSU vs. gNB

Figure 3 presents the impacts of the tasks' arrival rate, vehicles' arrival rate, and the number of RSUs on the average delay. From Fig. 3(a), 3(b), we can see that the offloading schemes all increase the average delay along with the increment of the arrival rate of tasks and vehicles, respectively. Increasing the number of RSUs improves the performance of the scheme that mainly relies on RSUs for offloading, i.e., $P_L = P_R = 0$, (as shown in Fig. 3(c)). For the schemes $(P_R = 0, P_L = 0.5)$ and $(P_R = 0.5, P_L = 0)$, we observe that the processing time at the gNB contributes the most to the delay; thus, increasing the number of RSUs cannot help to reduce the average delay. Another observation is that the 2-tier models suffer from significantly higher delay compare to 3-tier models. The model using only the gNB (i.e., $P_L = 1$) shows the worst performance (see Fig.3(a) and 3(b)). The 3-tier models with $(P_L = 0.5, P_R = 1)$ or $(P_L = 0, P_R = 1)$ are special cases where the RSUs only provide the communication but not computing. As computing contributes to the latency's main part, leveraging RSUs for only communication does not take many advantages. Consequently, the delay when using $(P_L = 0.5, P_R = 1)$ or $(P_L = 0, P_R = 1)$ is almost the same as using only the gNB. The 2-tier model using only RSUs (i.e., $(P_L = P_R = 0)$) improves the performance significantly compared to the one using only the gNB. Specifically, as shown in Fig. 3(a), the average delay when using only RSUs is only about half of those when using only the gNB. In comparison with the 2-tier models, our proposed algorithm reduces the delay up to 99.9%.

P_L	P_R	*Model*	*Meaning*
0	0	2-tier	All tasks are offloaded and processed at RSUs.
1	0	2-tier	Vehicles offload all tasks to the gNB.
0	0.5	3-tier	Vehicles offload all tasks to RSU. RSUs offload 50% of the tasks to the gNB.
0	1	3-tier	Vehicles offload all tasks to RSU. RSUs offload all tasks to the gNB.
0.5	0	3-tier	Vehicles offload 50% of the tasks to the gNB, 50% of the tasks to RSU; RSUs process all the tasks locally.
0.5	0.5	3-tier	Vehicles offload 50% of the tasks to the gNB, 50% of the tasks to RSUs; RSUs offload 50% of the tasks to the gNB.
0.5	1	3-tier	Vehicles offload 50% of the tasks to the gNB, 50% of the tasks to RSU; RSUs offload all the tasks to the gNB.
opt	*opt*	3-tier	the probabilities provided by our algorithm

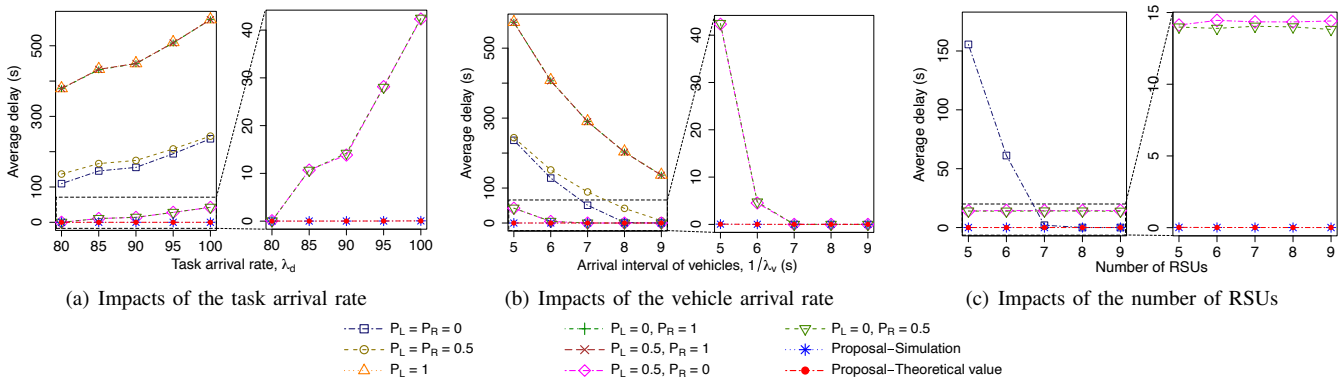
TABLE II: Offloading probability configurations

Factor	*Value*
the gNB's CPU	256 GHz
RSU's CPU	64 GHz
CPU cycles for a task	0.2 GHz
Mean packet size	500 kb
RSU-gNB's link bandwidth	10 Gbps
Vehicle-RSU's link bandwidth	1 Gbps
Vehicle-gNB's link bandwidth	500 Mbps
Vehicle arrival interval ($\frac{1}{\lambda_v}$)	$5 \sim 9$
Task arrival rate (λ_d)	$80 \sim 100$
Road length	1500 m
Vehicle speed	12 m/s

TABLE III: Simulation parameters

B. Comparison of various offloading probability configurations in the 3-tier model

In the following, we compare the offloading scheme determined by our algorithm and the others in the 3-tier model. As can be observed, our proposed algorithm achieves the best performance reflected by the smallest average delay. Especially, our proposal shows the superiority over the other schemes in the cases when the arrival rate of tasks and vehicles are significant (i.e., being higher than 85, less than 6 in Fig. 3(a), Fig. 3(b), respectively). The proposal shortens the delay by at least 4.5% and more than 90% in the best case compared to other offloading schemes. Figure 4 depicts the cumulative distribution function of the delay caused by various offloading strategies. As all the experiment scenarios show similar trends, we plot the results regarding the task arrival rate of 100, vehicle arrival interval of 5, and the number of RSUs of 5. The optimal values of P_L and P_R determined by our algorithm are 0.355 and 0.064, respectively. As the performance of $(P_L = 0.5, P_R = 1)$ and $(P_L = 0, P_R = 1)$ are almost the same, we plot only the delay of $(P_L = 0.5, P_R = 1)$. As shown, the delay values of three-tier with our proposal are all smaller than 0.1 seconds. Meanwhile, the offloading strategies using only the gNB computing capability (i.e., $P_R = 1$) incurs an extremely high latency (i.e., more than 80% of tasks endure the latency higher than 100 seconds). That is because the task arrival rate exceeds the service capacities of the gNB. By leveraging the resource of both the gNB



(a) Impacts of the task arrival rate (b) Impacts of the vehicle arrival rate (c) Impacts of the number of RSUs

Fig. 3: Average delay concerning various offloading probability configurations

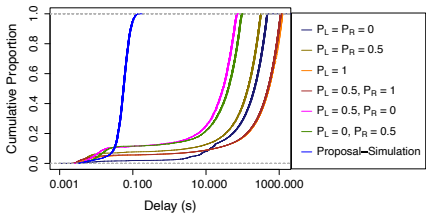


Fig. 4: Cumulative distribution function of delay

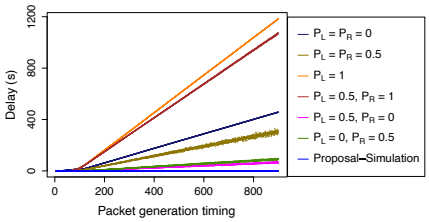


Fig. 5: Delay over time

and RSUs for offloading the computing, the schemes ($P_L = P_R = 0.5$), ($P_L = 0.5, P_R = 0$) and ($P_L = 0, P_R = 0.5$) shorten the latency significantly. However, in such cases, many packets (approximately 80%) have a higher latency than 1 seconds. Also, around 60% of those packets suffer from the latency of 10 seconds or above. Although the configuration ($P_L = 0.5, P_R = 0$) is quite close to those determined by our proposal, their performance gap is significant. The reason is that in ($P_L = 0.5, P_R = 0$), the task arrival rate at the gNB is slightly higher than the service capacity. Therefore, the processing time at the RSUs increases severely. Figure 5 plots the delay of the tasks over time. Interestingly, while the delay caused by other schemes gradually increases, that achieved by our proposal is quite stable. In ($P_L = P_R = 0$), ($P_L = P_R = 0.5$), ($P_L = 1$), and ($P_L = 0.5, P_R = 1$), most of the computational tasks are processed at only either RSUs and the gNB. This mechanism causes the bottleneck phenomenon and leads to overload as time goes on. As a consequence, the delay in these cases gradually increases. By leveraging the computational resource of both RSUs and the gNB, the configurations ($P_L = 0, P_R = 0.5$) and ($P_L = 0.5, P_R = 0$) balance the traffic loads better and thereby reduce the bottleneck phenomenon. It can be seen that the increasing slopes of the delay in ($P_L = 0, P_R = 0.5$) and ($P_L = 0.5, P_R = 0$) are much smaller compared to the others.

VI. Conclusion

In this paper, we have investigated the computational task offloading issue in the 3-tiers V2X network (i.e., vehicles, RSUs, and gNB). Specifically, we have addressed two problems: 1) performance comparison between the 3-tiers and 2-tiers model; 2) finding the optimal offloading probabilities for minimizing the average delay of the tasks. We derived the delay model to solve those problems, which is for the comparison and the mathematical formulation of the optimization problem. We have also proposed a GA-based approximation algorithm to determine near-optimal offloading probabilities. We have conducted simulations to validate the mathematical model's feasibility and evaluate the proposed algorithm's performance. The experimental results showed that our offloading algorithm could reduce the delay up to 99.9% in the best case, and 93.75% on the average.

Acknowledgment

This research is funded by Ministry of Education and Training of Vietnam under grant number B2020-BKA-13.

References

[1] L. Feng, W. Li, Y. Lin, L. Zhu, S. Guo, and Z. Zhen, "Joint computation offloading and urllc resource allocation for collaborative mec assisted cellular-v2x networks," *IEEE Access*, vol. 8, pp. 24 914–24 926, 2020.

[2] E. Pateromichelakis, C. Zhou, P. Keshavamurthy, and K. Samdanis, "End-to-end qos optimization for v2x service localization," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.

[3] X. Fan, T. Cui, C. Cao, Q. Chen, and K. Kwak, "Minimum-cost offloading for collaborative task execution of mec-assisted platooning," *Sensors*, vol. 19, no. 847, 2019.

[4] T. Cui, Y. Hu, B. Shen, and Q. Chen, "Task offloading based on lyapunov optimization for mec-assisted vehicular platooning networks," *Sensors*, vol. 19, no. 4974, 2019.

[5] H. Wang, X. Li, H. Ji, and H. Zhang, "Federated offloading scheme to minimize latency in mec-enabled vehicular networks," in *Proc. IEEE GLOBECOM Workshops*, 2018, pp. 1–6.

[6] G. Wang and F. Xu, "Regional intelligent resource allocation in mobile edge computing based vehicular network," *IEEE Access*, vol. 8, pp. 7173–7182, 2020.

[7] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, 2019.

[8] A. Weissberger, "Fastest 5G network in the U.S.? T-Mobile vs Verizon; Nokia's fastest 5G claim," https://techblog.comsoc.org/2020/05/20/fastest-5g-network-in-the-u-s-t-mobile-vs-verizon-nokias-fastest-5g-claim/, 2020, [Online; accessed 25-May-2020].