

Article

# Task Offloading Strategy and Simulation Platform Construction in Multi-User Edge Computing Scenario

Guilu Wu <sup>1,2,3,4,\*</sup>  and Zhongliang Li <sup>4</sup><sup>1</sup> State Key Laboratory of Millimeter Waves, Southeast University, Nanjing 210096, China<sup>2</sup> State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China<sup>3</sup> School of Information and Communication Engineering, Hainan University, Haikou 570228, China<sup>4</sup> School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China; lizhongliang@stu.jiangnan.edu.cn

\* Correspondence: wgl@seu.edu.cn

**Abstract:** Various types of service applications increase the amount of computing in vehicular networks. The lack of computing resources of the vehicle itself will hinder the improvement of network performance. Mobile edge computing (MEC) technology is an effective computing method that is used to solve this problem at the edge of network for multiple mobile users. In this paper, we propose the multi-user task offloading strategy based on game theory to reduce the computational complexity and improve system performance. The task offloading decision making as a multi-user task offloading game is formulated to demonstrate how to achieve the Nash equilibrium (NE). Additionally, a task offloading algorithm is designed to achieve a NE, which represents an optimal or sub-optimal system overhead. In addition, the vehicular communication simulation frameworks Veins, SUMO model and OMNeT++ are adopted to run the proposed task offloading strategy. Numerical results show that the system overhead of the proposed task offloading strategy can degrade about 24.19% and 33.76%, respectively, in different scenarios.

**Keywords:** mobile edge computing; task offloading; game theory; Nash equilibrium



**Citation:** Wu, G.; Li, Z. Task Offloading Strategy and Simulation Platform Construction in Multi-User Edge Computing Scenario. *Electronics* **2021**, *10*, 3038. <https://doi.org/10.3390/electronics10233038>

Academic Editor: George A. Tsihrintzis

Received: 15 November 2021

Accepted: 2 December 2021

Published: 5 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the development of Internet of Things (IoT) communication technology, vehicular networks have been the most promising application in IoT and provide vehicle users with comprehensive information services on transportation, safety, and entertainment [1,2]. However, these services are usually computationally intensive with too great a burden to the terminal equipment. The limited computing resources of the user terminal restrict the response to the user service application. Fortunately, the cloud computing technology as an emerging technology, can apply to data processing, which can break through the drawbacks of limited computing resource of end-user devices [3,4]. Hence, many IoT applications have achieved rapid growth driven by the cloud computing.

However, some typical applications are not suitable for using cloud computing to solve the problem of limited computing resources. For example, the delay-dependent messages sent by vehicles need to be transmitted with extremely low transmission delay in the vehicular networks. This demand makes the vehicular networks must to satisfy ultra-low delay [5,6]. By definition, cloud computing is a new paradigm of computing, which refers to the network management, storage and centralized computing in clouds for providing resources [7]. This mechanism may bring large delay due to long-distance transmission of the calculation result for user service. The cloud computing resource which is brought to the edge of network, that is mobile edge computing (MEC) technology, closer to the user terminal, can effectively reduce transmission delay. Edge computing is between physical entities and industrial connections, or at the top of physical entities.

Additionally, cloud computing can still access the historical data of edge computing [8]. The task offloading is an important part of the MEC technology.

The nature of task offloading is to exchange communication resources for computing resources. Generally, data are increasingly produced at the edge of the network, so it would be more efficient to process data at the edge of the network [9–11]. As a rule of thumb, task offloading process includes three steps, denoted as task transmission, task execution and result retrieval [12]. MEC is a novel network architecture, where tasks can be executed in the edge server nearby the vehicle. This also reduces response time of data communication [13]. Normally, the BS allocates multi-channels to multiple mobile users in most wireless networks. A key challenge is an efficient wireless channel coordination when multiple mobile users operate computation offloading. In this process, the interference may occur among multiple mobile users and reduce the data rates for computation offloading. Hence, the energy efficiency and data transmission time is reduced further. In this case, the task offloading with MEC cannot bring benefit for the mobile users. The efficient task offloading strategy with MEC is required to achieve high wireless access efficiency in mobile wireless networks.

In this paper, we design a task offloading scheduling scheme based on game theory, followed by an efficient task offloading algorithm in vehicular networks. Additionally, none vehicle will change current offloading decision. In this way, the system can remain stability and reduce the system overhead compared to the state where computational task is executed locally. The main contributions of this paper are summarized as follows:

1. The system overhead is calculated through time cost on task offloading in vehicular networks. The communication and computing models are considered to construct system performance characteristics. Specifically, the reliable uplink transmission rate is defined as the communication model, which is the function of offloading decision profile. Then, the computing model focuses on local computing and edge computing for a computationally intensive task;
2. The optimization problem on system overhead is established to deliver computing goal when the data transmission rate reaches the minimum threshold data transmission requirement. A novel game theory is applied to this strategy for reaching an optimal solution and reducing the computational complexity and achieving system stability. Meanwhile, the specific multi-user task offloading scheduling algorithm which includes channel interference and offloading decision update is designed in vehicular networks;
3. To obtain the vehicle trajectory data, the vehicular communication simulation frameworks Veins, road traffic simulator SUMO and OMNeT++ tools are combined together to simulate vehicle behavior in real scenario. Additionally, on this basis the proposed multi-user task offloading strategy is verified and analyzed at aspect of reducing system overhead.

In addition to providing insight into the behavior of the task offloading strategy, our simulation platform constructed can be used as an enabling component in vehicular networks validation platform such as MEC scenario simulators and emulators. In particular, simulation platform has verified actual geographic scenario based our proposed task offloading strategy in this paper. At the same time, the selection of Veins, SUMO and OMNeT++ in simulation platform constructed demonstrates strong operability.

The rest of this paper is organized as follows. After a review of related works in Section 2. We formulate in Section 3 the system model on multi-user and multi-server task offloading in vehicular networks. Section 4 derives an efficient task offloading strategy using the system model. After verifying and discussing the simulation platform building and system overhead results in Section 5, we conclude in Section 6.

## 2. Related Works

In recent years, due to the advent of MEC technology, the issues of data processing efficiency and delay have received increased attention again. We briefly review task offloading scheduling problems in MEC scenario which are directly related to our study.

With cloud computing and mobile cloud computing, References [14–16] formulate an advanced task offloading model from resource allocation, data availability, scalability, performance augmentation, energy saving, and security and privacy. How to manage the exponential data traffic increase is always a key challenge to improving network performance. For example, Reference [17] presents a series of data offloading techniques in wireless networks. The delay problem is considered to satisfy content delivery requirements. In vehicular networks, Zhou et al. in [18] focuses on data offloading framework design, the data transmission algorithm, and data offloading optimization problem. Specifically, delay and processing capabilities are important to vehicular networks for enhancing vehicular applications. Data offloading techniques through vehicular networks mainly include three different communication patterns among vehicle and infrastructures, such as vehicle to infrastructure (V2I) communication, vehicle to vehicle (V2V) communication, and vehicle to everything (V2X) communication. In addition, device to device (D2D) communication technology is another promising application in term of network control through communication session [19]. D2D function enables to overcome the delay and timing issues from V2V communication. This is similar to the concept of offloading. In [19], a comprehensive survey of D2D communication and the requirements for V2V communication are presented. More importantly, high latency and huge load problem are also discussed and solved for vehicle communication. A content delivery management system is designed in [20] for D2D data offloading. Additionally, this management system is suitable for vehicular environments, where the network topology changes in real time. However, these works ignore specific task offloading strategy and scheduling mechanism. This will degrade system performance.

Against this background, there has been many works focusing on task offloading strategies to achieve specific optimization of network performance [21–24]. In [21], the optimal resource allocation scheme is formulated as a convex optimization problem for minimizing the sum energy consumption under the constraint on computation latency. Moreover, a sub-optimal resource allocation scheme is also proposed to reduce the computation complexity as of the cloud with finite capacity. With the development of novel application, more and more tasks are computation-intensive and data-intensive. These features can be delay-sensitive [22]. In order to meet the low latency demand in ultra-dense network, MEC technology is introduced as an effective solution. However, the distributed computation resource in edge cloud makes it difficult to offload tasks for users. Hence, Chen et al. propose a task offloading policy for MEC in software-defined ultra-dense networks [23]. In [23], the task offloading problem is investigated and formulated to a mixed integer non-linear problem which is NP-hard. Additionally, the distributed task offloading algorithm is proposed to solve it based on game theory. In fact, topology and schedules of user devices are often omitted in process of task offloading. This leads to the performance of network degradation and the edge resources under utilization. The task can be divided into a series of sequential subtasks and the subtask is determined to be offloaded [24]. In real-world practices, Reference [25] improves the double auction mechanism which can assign the tasks of users to the edge servers. In the proposed efficient mechanism, the resource allocation problem is converted to a minimum cost flow problem to achieve the optimal social welfare. Considering the mobility of vehicular environments, it is crucial for the optimal task offloading decisions. A novel dynamic task offloading scheme for multiple subtasks is proposed to make the utility minimization in [26]. The computation resources of MEC server can be reasonably allocated to meet the differences computation intensity of each vehicle. Furthermore, Reference [26] combines the task offloading strategy and computation resource allocation to design a dynamic task offloading decision scheme for improving the utility of vehicular networks. In addition, Reference [27] proposes a

reinforcement learning-based algorithm to solve the task offloading problem in vehicular edge computing networks. Additionally, the proposed scheme considers time-varying offloading decisions facing dynamic topology and the actual road traffic situation. Considering transmission latency and energy efficiency, a deep learning approach for energy efficient computational offloading in MEC is designed in [28] to improve low network congestion and application performance. When the mobile user has the limited energy capacity, the performance of MEC is restricted in operating task computation. In [29], the jointed optimized scheme is proposed to consider the energy transmission beamforming at the access point, the CPU frequencies, the number of offloading data bits at the user and the time allocation among the users in a multi-user wireless powered MEC system. Reference [30] formulates an optimization problem to discuss the tradeoff between energy efficiency and delay in wireless powered MEC systems by deep learning when the cost function is ignored. To achieve a joint task offloading and time allocation simultaneously, a joint time allocation and offloading strategy is proposed in [31]. Additionally, both energy consumption and the time delay of the mobile user are also considered in [31] by designing a cost function for a mobile user.

In these processes, the specific communication performance is omitted when the task offloading is focused on algorithm or scheme design for practical networks, such as vehicular networks. Relying on actual scenarios to verify the proposed scheme, more sophisticated simulation techniques are needed to simulate a network traffic and a road traffic [32–34]. In order to develop more efficient Inter-Vehicle Communication (IVC) protocols [32], Veins is mainly used for evaluating the performance of vehicular ad hoc networks. Veins provides comprehensive IVC models, which can be used for simulation frameworks to simulate practice application scenarios. Reference [33] develops a simulation framework OMNeT++ which is a simulation environment to model realistic communication patterns of vehicles nodes. Additionally, the microscopic road traffic simulation package SUMO can simulate traffic simulation [34]. In this paper, we use Veins, combining with OMNeT++ and SUMO, as our simulation framework where road-side units (RSUs) receive message vehicles transfer by IVC protocols to evaluate the performance of proposed task offloading scheme.

### 3. System Model

The main object of this section is to formulate the system overhead model based on multi-users and multi-servers task offloading process in vehicular networks. To achieve this goal, we first built communication model to ensure reliable data transmission rate. We then introduce computation model from two aspects local computing and edge computing.

#### 3.1. Overview

We consider the multi-user and multi-server task offloading architecture in MEC scenario, as shown in Figure 1. The base station (BS) and RSUs are deployed in road side randomly. These V2I and V2V communication links agglomerate together to form vehicular communication networks. We assume the edge node could be other vehicles, RSUs and BS. Two problems are presented in this scenario. The first is that the tasks are to be offloaded or not. The second is that how to choose the appropriate channel to improve the efficiency of task offloading.

We consider a large number of vehicle users, denoted as  $\mathcal{N} = \{1, 2, \dots, i, \dots, N\}$  in vehicular networks, where each vehicle has a computationally intensive task to be completed. In order to obtain an tractable analysis process and results, we assume a quasi-static scenario where some parameters (e.g., vehicle numbers or channel state) remain unchanged during a task offloading period (The number or channel state may change across different periods). The communication and computation models will be introduced in following, respectively. All the notations used in this section are displayed in Table 1.

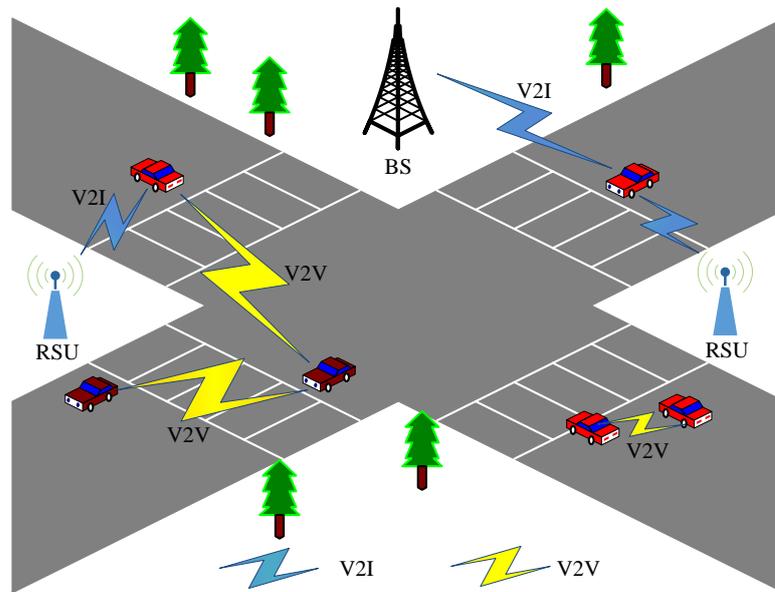


Figure 1. Task offloading architecture in vehicular networks.

Table 1. List of notations used.

Notation	Meaning
$N$	Number of vehicles
$M$	Number of ES
$a_i$	The $i$ -th vehicle
$W$	Channel bandwidth
$p_i$	The $i$ -th vehicle transmission power
$h_{ij}$	The channel gain for link between $i$ -th vehicle and $j$ -th ES
$r$	The uplink transmission rate
$\mathbf{a}$	The function of offloading decision
$m_i$	The size of computation input data
$n_i$	The total numbers of CPU cycles to accomplish task
$f_i^{local}$	Computing capacity
$t_i^{local}$	The local execution time of task
$e_i^{local}$	The energy consumption of local computing
$\delta_i$	The energy consumption coefficient of local computing
$E_i^{local}$	The integrated circuit energy consumption for local computing
$K_i^{local}$	The overhead of local computing
$\lambda_i^t$	The factor of delay
$\lambda_i^e$	The factor energy consumption
$f_j^{edge}$	The computing capability of $j$ -th ES
$t_{i,exe}^{edge}$	The time of task executed in the ES
$t_{i,trans}^{edge}$	The transmission time of task offloading to the ES
$t_i^{edge}$	The total time of edge computing
$e_i^{edge}$	The energy consumption between vehicle and ES
$E_i^{edge}$	The energy consumption when the vehicle maintains communication link
$E_0^{edge}$	The energy consumption threshold when the vehicle maintains communication link
$K_i^{edge}$	The overhead of edge computing
$K_{sys}$	The system overhead

### 3.2. Capacity Model

Herein, the capacity model is introduced to transmit vehicle offloading task to the edge server (ES) through uplink communication and the impact of downlink communication is ignored [35]. There are  $\mathcal{M} = \{1, 2, \dots, j, \dots, M\}$  ESs deployed in system. The offloading decision of  $i$ -th vehicle is denoted as  $a_i$ , where  $a_i = 0$  and  $a_i = j$  represent the vehicle  $i$  chooses to execute task locally with its own CPU and offload task to ES  $j$  for execution, respectively.

We define the offloading strategy profile  $\mathbf{a} = (a_1, a_2, \dots, a_N)$ . For a certain offloading strategy profile, different vehicle users can access to the same ES at the same time and frequency (e.g., CDMA). The noise and interference will accompany the data propagation process. Therefore, we define reliable uplink transmission rate of  $i$ -th vehicle to  $j$ -th ES according to Shannon theorem as [36]

$$r_i(\mathbf{a}) = W \log_2 \left( 1 + \frac{p_i h_{ij}}{\Psi + \sum_{i' \in N; a_{i'}=a_j} p_{i'} h_{i'j}} \right). \quad (1)$$

Due to multiple access in the channel, Equation (1) introduces interference term  $p_{i'} h_{i'j}$ . We notice that the uplink transmission rate  $r$  is the function of offloading decision  $\mathbf{a}$ . When large number of vehicles access to the same ES, the rate  $r$  will decrease.

### 3.3. Computation Model

For a computationally intensive task, the task can be executed locally or offloaded to ES. The task offloading efficiency is represented by two indicators, denoted as  $m_i$  and  $n_i$ . In the following, we focus on the computation model of local computing and edge computing.

(1) *Local Computing*: For the task executed locally, let  $f_i^{local}$  denotes the computing capability (e.g., CPU revolutions per second) of  $i$ -th local computer. In heterogeneous vehicular networks, the local execution time of task for different vehicles can be defined as

$$t_i^{local} = \frac{n_i}{f_i^{local}}, \quad (2)$$

Additionally, the energy consumption is proportional to the CPU cycles [12], and it is can be expressed as

$$e_i^{local} = \delta_i n_i + E_i^{local}, \quad (3)$$

where, the value of  $E_i^{local}$  is constant in this paper. The overhead of local computing is written by

$$K_i^{local} = \lambda_i^t t_i^{local} + \lambda_i^e e_i^{local}. \quad (4)$$

According to different applications, we can set  $\lambda$  flexibly. For example, when executing time sensitive application, we can set  $\lambda_i^t = 1$ , and when the vehicle energy is low, we can set  $\lambda_i^e = 1$ .

(2) *Edge Computing*: If the vehicle choose offload task to the ES, the delay and energy consumption can be analyzed in a similar way. The time of task executed in the ES is

$$t_{i,exe}^{edge} = \frac{n_i}{f_j^{edge}}, \quad (5)$$

During the period of transmitting data, delay and energy consumption will be introduced. According to the communication model introduced in Section 3.2, transmission time of task offloading to the ES can be expressed as

$$t_{i,trans}^{edge} = \frac{m_i}{r_i(\mathbf{a})}, \quad (6)$$

Certainly, we assume that all ESs are equipped with sufficient computing resource. Then, the total time of edge computing by combining with Equations (5) and (6) can be derived by

$$t_i^{edge} = t_{i,exe}^{edge} + t_{i,trans}^{edge}. \quad (7)$$

At aspect of energy consumption, we only consider their own energy consumption of vehicle users. The energy consumption between vehicle and ES can be formulated as

$$e_i^{edge} = \alpha_i t_{i,trans}^{edge} + E_i^{edge} = \frac{\alpha_i m_i}{r_i(\mathbf{a})} + E_i^{edge}. \quad (8)$$

and we have

$$E_i^{edge} \geq E_0^{edge}. \quad (9)$$

Similarly, the total overhead of edge computing according to (7) and (8) can be written as

$$K_i^{edge} = \lambda_i^t t_i^{edge} + \lambda_i^e e_i^{edge}. \quad (10)$$

Equations (4) and (10) show the overhead of different computing scenarios. Therefore, the system overhead can be expressed as

$$K_{sys} = \sum_i K_i^{local}(a_i = 0) + \sum_i K_i^{edge}(a_i \neq 0). \quad (11)$$

#### 4. Efficient Task Offloading Strategy

After the basic components of the model is derived, we now consider an efficient task offloading strategy for multi-user task offloading. First, we describe a general scenario of task offloading. Then, the problem that how to update decision based on game theory is discussed. Lastly, an efficient task offloading algorithm is proposed to solve multi-user task offloading problem in vehicular networks.

##### 4.1. Problem Statement

From the system model in Section 3, the wireless channel will be idle, if tasks are all executed locally (e.g.,  $\mathbf{a} = (0, 0, \dots, 0)$ ). At this time, if one vehicle chooses to offload its task to the ES, the overhead of this vehicle will be reduced, then it will choose to change its current offloading decision. In addition, when a large number of vehicles choose to offload tasks to the ESs, the transmission rate  $r$  will decrease from Equation (1). According to Equations (6) and (8), the transmission time and energy consumption will increase, and the performance of task offloading will be worse correspondingly. Therefore, the minimum data transmission rate  $R$  is introduced to eliminate communication link interruption.

Considering a task offloading scenario with  $N$  vehicles and  $M$  ESs, task offloading strategy profile  $\mathbf{a}$  is in  $n$ -dimensional space, where each space includes  $M + 1$  possible values. If we need to transmit all possible values, there are  $(M + 1)^N$  situations. The minimum overhead in system is expressed as

$$\begin{aligned} & \min K_{sys} \\ & s.t. \quad r_i(\mathbf{a}) \geq R \quad \forall i \in \{0, 1, \dots, N\} \end{aligned} \quad (12)$$

From the previous description, the complexity of calculation will increase exponentially with the increase in vehicle numbers. This characteristic does not apply to current vehicular network environment. Hence, traditional optimization problem solutions are difficult to achieve the desired results on the Equation (12).

##### 4.2. Game Formulation

Game theory is a mathematical theory and method for studying conflicting or competitive phenomenon [37]. Taking into account the limited computing resources of the vehicle and the complexity of multiple tasks from Section 4.1, a distributed task offload-

ing algorithm is needed to achieve lower computational complexity. For task offloading decision  $a_i$  of vehicle user  $i$ , we define the task offloading strategy of other vehicles as  $a_{-i} = (a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_N)$ . The objective of vehicle user  $i$  is to minimize its overhead. The specific scheme is designed by

$$\begin{aligned} & \min_{\substack{a_i \in \{0,1,\dots,M\} \\ i=0,1,\dots,N}} K_i(a_i, a_{-i}) \\ \text{s.t. } & K_i(a_i, a_{-i}) = \begin{cases} K_i^{\text{local}}(a_i = 0) \\ K_i^{\text{edge}}(a_i \neq 0) \end{cases} \end{aligned} \quad (13)$$

For classical game theory application, all vehicles,  $\Gamma = \{1, 2, \dots, N\}$ , denote participants, and task offloading strategy,  $\{A_i\}_{i=\{1,2,\dots,N\}}$  ( $A_i = 0, 1, \dots, M$ ), denotes strategy space and the overhead,  $K_i(a_i, a_{-i})$ , denotes payment vector of each participant. Therefore, a task offloading game can be described as  $G_i = (\Gamma, \{A_i\}, K_i(a_i, a_{-i}))$ . The Nash equilibrium (NE) in task offloading scenario will be discussed in the following.

For any vehicle  $i$ , the task offloading strategy profile  $\mathbf{a}^* = (a_1^*, \dots, a_i^*, \dots, a_N^*)$  requires to achieve through using game theory. When the vehicle  $i$  cannot change its own offloading decision  $a_i^*$  to reduce its own overhead  $K_i(a_i^*, a_{-i}^*)$ , the task offloading strategy profile  $\mathbf{a}^*$  is a NE of multi-user task offloading game. The result can be written by

$$\begin{aligned} & K_i(a_i^*, a_{-i}^*) \leq K_i(a_i, a_{-i}) \\ & i = 0, 1, \dots, N, \\ & a_i = 0, 1, \dots, M. \end{aligned} \quad (14)$$

Although the NE point cannot be sure that the solution will reach optimal, any vehicle will not change its current offloading decision. All vehicles will maintain current offloading strategy. Nonetheless, the NE solution can ensure system stability, which applies to practice application scenario.

#### 4.3. The Proposed Task Offloading Algorithm

We assume the offloading strategy between different vehicles are mutually coupled. Each vehicle is a decision-maker during task offloading in the distributed multi-user edge networks. The specific task offloading process includes two stages as follows:

(1) *Channel Interference*: For any vehicle  $i$ , we assume the offloading decision  $a_i$  is known and the overall received power  $P_j(a_i(t), a_{-i}(t))$  of ES  $j$  is also known. If the vehicle  $i$  accesses to the ES  $j$ , the interference  $I_j(a_i(t), a_{-i}(t))$  numerically equals to the overall received power. Otherwise, the interference numerically equals to the overall received power minus except for the vehicle  $i$ . The wireless channel interference of vehicle  $i$  to the ES  $j$  is calculated as

$$I_j(a_i(t), a_{-i}(t)) = \begin{cases} P_j(a_i(t), a_{-i}(t)), & a_i(t) \neq j \\ P_j(a_i(t), a_{-i}(t)) - q_i g_{i,j}, & a_i(t) = j \end{cases} \quad (15)$$

It is worth noting that each vehicle can obtain the wireless channel interference from Equation (15) without knowing the offloading decision of other vehicles.

(2) *Offloading Decision Update*: The next stage is the current offloading decision of vehicle will be updated at this moment. When the wireless channel interference  $\{I_j(a_i(t), a_{-i}(t)), j = 1, \dots, M\}$  is known, the vehicle  $i$  can update its strategy based on the optimal offloading decision as

$$\begin{aligned} \mathbf{U}_i = & \{a_i^*; K_i(a_i^*, a_{-i}(t)) \leq K_i(a_i(t), a_{-i}(t)), \\ & a_i^* = 0, 1, \dots, M; r_i(a_i^*, a_{-i}(t)) \geq R\} \end{aligned} \quad (16)$$

Each vehicle can get the update set  $\mathbf{U}$  by Equation (16) in wireless communication. When the condition is  $a_i(t) \in \mathbf{U}_i$ , the vehicle  $i$  will remain its current decision, such as  $a_i(t+1) = a_i(t)$ . Otherwise, the vehicle  $i$  will send out a request message to update its current decision.

The specific operations on multi-user task offloading are elaborated in vehicular networks as follows: the ESs  $\mathcal{M}$  send out the overall received power to the vehicles  $\mathcal{N}$ . The vehicles  $\mathcal{N}$  can get wireless channel interference according to Equation (15). Then, these vehicles decide whether to change current offloading decision by Equation (16) and send out a request message to the ES. The ES randomly chooses a request frame and returns a response frame to the corresponding vehicle. This demonstrates the vehicle is permitted to change its current decision and the other vehicles remain the current decision. The task offloading scheduling can be accomplished, until the ES cannot receive any request frame. A multi-user edge network task offloading scheduling algorithm is designed in vehicular networks. The corresponding pseudocode is shown in Algorithm 1. It is noted that the data processing of this algorithm can be operated in edge node, such as BS or vehicles. In addition, the MEC technology could provide computing resources for all vehicles.

---

**Algorithm 1** Multi-user edge network task offloading scheduling algorithm.

---

**Require:** *timeslot* and initial offloading decision  $\mathbf{a}(0)$

**Ensure:** overhead and final offloading decision  $\mathbf{a}$

```

1: Initialization;
2: for  $t = 1; t \leq \textit{timeslot}$  do
3:   ES broadcasts received power  $P_j(a_i(t), a_{-i}(t))$ ;
4:   Vehicle calculates channel interference  $I_j(a_i(t), a_{-i}(t))$ ;
5:   Vehicle calculates optimal decision sets  $\mathbf{U}_j$ ;
6:   if  $\mathbf{U} \neq \emptyset$  then
7:     Randomly select a vehicle  $i$ ;
8:      $a_i(t+1) \in \mathbf{U}_i$ ;
9:      $t = t + 1$ ;
10:  else
11:    System reaches NE;
12:    Task offloading scheduling accomplished;
13:  end if
14: end for

```

---

## 5. Simulation Results

In this section, we evaluate the proposed task offloading strategy by simulation platform built. The simulation platform is built with reference to the actual physical scenario. Subsequently, when the simulation platform runs the proposed task offloading strategy, the system overhead is analyzed accompanying with different slot time, number of vehicles and RSUs.

### 5.1. Simulation Platform Building

The vehicular communication simulation framework Veins, road traffic simulator SUMO and network simulator OMNeT++ tool are combined together to realize the proposed multi-user task offloading scheduling scheme. Against this background, the SUMO is adopted to provide map and vehicle trajectory information, which can describe simulation scenario well, and the OMNeT++ is introduced to realize the communication between the vehicles and roadside infrastructure (e.g., RSUs and BS). For supporting in dynamic communication environment, Veins enables wireless access in vehicular environments (WAVE) standards, which standardizes the communication between physical (PHY) layer and media access control (MAC) layer. The Veins provides the corresponding function, which can realize the whole communication from vehicle to RSU. In addition, Table 2 shows mainly system parameters in multi-user and single-server scenario, where vehicles randomly travel in different crossroads and RSUs are located at the center of crossroads.

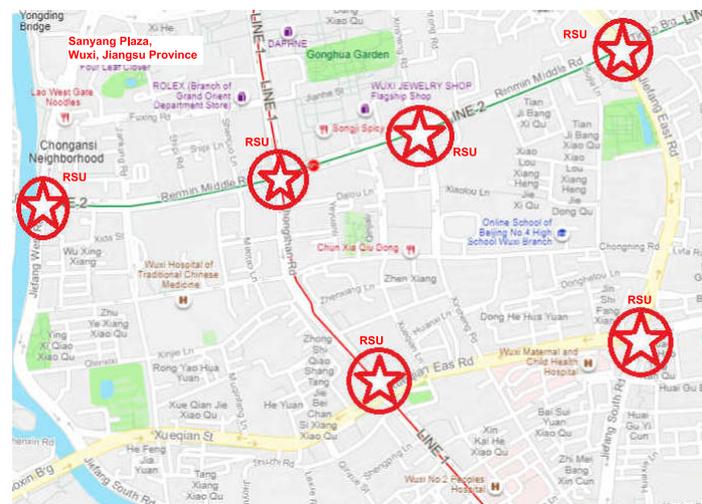
**Table 2.** Simulation Parameters.

System Parameters	Value
Vehicle numbers ( $N$ )	10
RSU number ( $M$ )	1 or 6
Channel Bandwidth ( $W$ )	5 MHz
Background noise power ( $\Psi$ )	−100 dBm
Local computing capability ( $K^{local}$ )	{0.5, 0.8, 1.0} GHz
Edge computing capability ( $K^{edge}$ )	10 GHz
Total numbers of CPU cycles ( $n$ )	100 Megacycles
Size of input data ( $m$ )	200 KB
Delay factor ( $\lambda^t$ )	0.5
Energy consumption factor ( $\lambda^e$ )	0.5

### 5.2. Results Analysis

In order to verify our model, the data from a real scenario with multi-user and multi-server is considered to display work conditions. Formally, we get the map of Sanyang Plaza in Wuxi, Jiangsu province, China from Open Street Map (OSM), as shown in Figure 2. The messages from each vehicle can be received by any number of RSU (even no RSU). In a nutshell, we make some modifications with “JOSM” software. The corresponding simulation scenario is built in Figure 3. Some parameters also need to be changed, where we set vehicle numbers to 120, and RSU numbers to 6, and channel bandwidth to 20 MHz, and size of input data to 200 KB, which benefits to task offloading. The OMNeT++ supports to export resulting data to CSV format, suitable for importing into Python’s Pandas or R language. We choose Python’s Pandas to process data and achieve task offloading strategy scheduling.

The local computing capability of each vehicle is defined as 0.5, 0.8, or 1. The SUMO builds a network simulation model and generates network file, traffic file and poly file. This model is imported to the OMNeT++ and the vehicles are to instantiate mobile network nodes. RSU begins to operate task offloading scheduling after receiving the messages of all vehicles. The offloading vehicle numbers of each RSU increase initially due to free channel, and then settle out during a period of jitter. Specifically, the system overhead of each time slot is verified in Figures 4 and 5 when  $RSU = 1$  and  $RSU = 6$ , respectively. From the Figure 4, it is obvious that the system overhead reduces about 24.19% compared to the initial state when it maintains stabilize after four times of update operation. When the multiple RSUs are deployed in networks, the system overhead fluctuates quite a bit initially and reaches a steady state finally, which can reduce about 33.76% compared to the initial state from the Figure 5. It shows that network resource can be used effectively by task offloading scheme.

**Figure 2.** The actual deployment of the RSUs in real scenario.

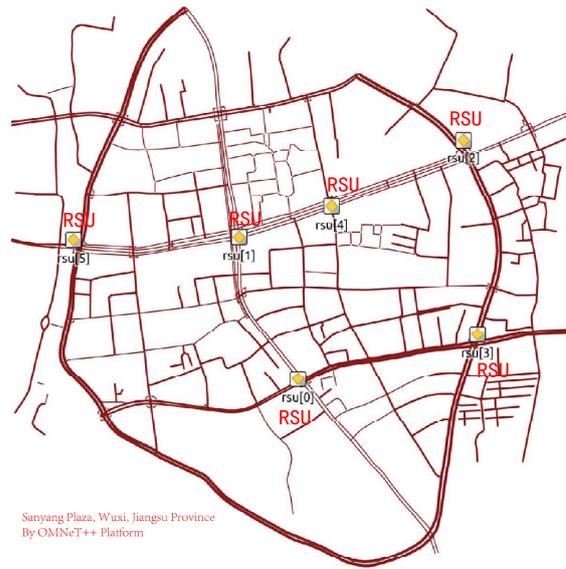


Figure 3. The simulation of the RSUs in OMNeT++ platform.

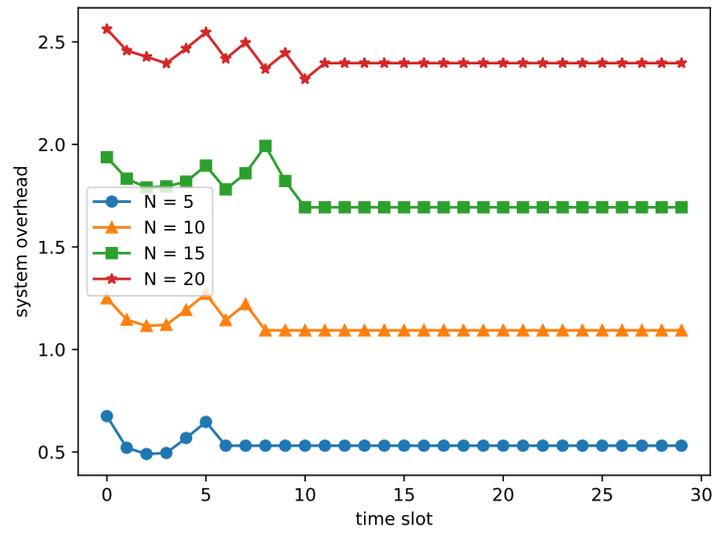


Figure 4. System overhead of each time slot in  $RSU = 1$  scenario.

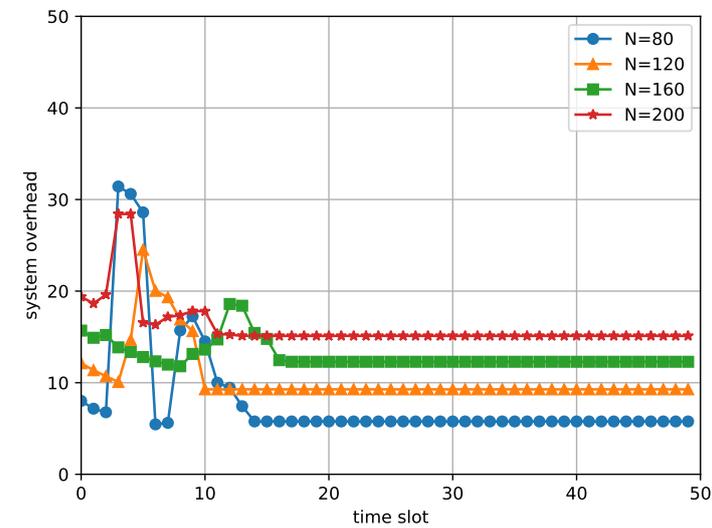
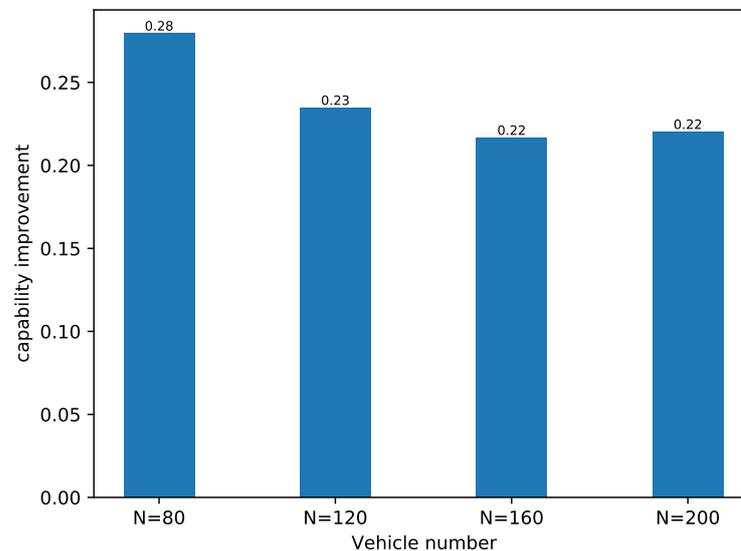


Figure 5. System overhead of each time slot in  $RSU = 6$  scenario.

In addition, the system capacity of the vehicular networks is improved when the proposed offloading scheme is adopted in Figure 6. With the increase in the number of vehicles, the proposed task offloading scheme can enhance the vehicles to access channel freely and to facilitate communication through V2I and V2V pattern.



**Figure 6.** Capacity improvement at different number of vehicles.

## 6. Conclusions

In this paper, we propose an efficient task offloading scheduling strategy based on game theory in multi-user and multi-server scenario for MEC. We formulate the problem as a multi-user task offloading game and develop a multi-user edge network task offloading scheduling algorithm for NE. Furthermore, the road traffic simulator SUMO and network simulator OMNeT++ are combined together to achieve task offloading scheduling. On the basis of Veins, vehicles can transmit message to RSUs, and then RSUs begin to operate the task offloading strategy. Simulation results show that the proposed algorithm can get a stable offloading decision and achieve optimal or near-optimal of task offloading performance for vehicular networks.

In addition, the more complex task offloading scenario will be discussed in future work. The proposed scheme can be embedded into Veins and handled task offloading in real-time. Additionally, when the task is divided into different subtasks, the system model is also suitable for task offloading of subtask.

**Author Contributions:** Conceptualization, G.W.; methodology, G.W.; software, Z.L.; validation, Z.L.; formal analysis, G.W. and Z.L.; investigation, Z.L.; resources, G.W.; data curation, Z.L.; writing—original draft preparation, Z.L.; writing—review and editing, G.W.; visualization, G.W.; supervision, G.W.; project administration, Z.L.; funding acquisition, G.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Jiangsu Planned Projects for Postdoctoral Research Funds (2020Z113), the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) (SKLNST-2021-1-13), the Open Foundation of National Mobile Communications Research Laboratory, Southeast University (2020D17) and the Fundamental Research Funds for the Central Universities project (JUSRP12020).

**Acknowledgments:** The authors would like to thank the anonymous reference and reviewers for their helpful comments that have significantly improved the quality of the presentation.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Qureshi, K.N.; Din, S.; Jeon, G.; Piccialli, F. Internet of vehicles: Key technologies, network model, solutions and challenges with future aspects. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1777–1786. [\[CrossRef\]](#)
2. Abbasi, S.; Rahmani, A.M.; Balador, A.; Sahafi, A. Internet of vehicles: Architecture, services, and applications. *Int. J. Commun. Syst.* **2021**, *34*, e4793. [\[CrossRef\]](#)
3. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [\[CrossRef\]](#)
4. Siriwardhana, Y.; Porambage, P.; Liyanage, M.; Ylianttila, M. A survey on mobile augmented reality with 5G mobile edge computing: Architectures, applications, and technical aspects. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1160–1192. [\[CrossRef\]](#)
5. Gina Rose, G.; Gunasekaran, R.; Aishwarya, G. Mobility management for critical time and delay tolerant applications in vehicular networks. In Proceedings of the 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 13–15 December 2018; pp. 344–348.
6. Naseer Qureshi, K.; Bashir, F.; Iqbal, S. Cloud computing model for vehicular ad hoc networks. In Proceedings of the 2018 IEEE 7th International Conference on Cloud Networking (CloudNet), Tokyo, Japan, 22–24 October 2018; pp. 1–3.
7. Zhang, Q.; Cheng, L.; Boutaba, R. Cloud computing: State-of-the-art and research challenges. *J. Internet Serv. Appl.* **2010**, *1*, 7–18. [\[CrossRef\]](#)
8. Corcoran, P.; Datta, S.K. Mobile-Edge Computing and the Internet of things for consumers: Extending cloud computing and services to the edge of the network. *IEEE Consum. Electron. Mag.* **2016**, *5*, 73–74. [\[CrossRef\]](#)
9. Akpakwu, G.A.; Silva, B.J.; Hancke, G.P.; Abu-Mahfouz, A.M. A survey on 5G networks for the Internet of things: Communication technologies and challenges. *IEEE Access.* **2018**, *6*, 3619–3647. [\[CrossRef\]](#)
10. Xavier, G.P.; Kantarci, B. A survey on the communication and network enablers for cloud-based services: State of the art, challenges, and opportunities. *Ann. Telecommun.* **2018**, *73*, 169–192. [\[CrossRef\]](#)
11. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [\[CrossRef\]](#)
12. Shu, C.; Zhao, Z.; Han, Y.; Min, G. Dependency-aware and latency-optimal computation offloading for multi-user edge computing networks. In Proceedings of the 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), Boston, MA, USA, 10–13 June 2019; pp. 1–9.
13. Zhang, Z.; Li, S. A survey of computational offloading in mobile cloud computing. In Proceedings of the 2016 4th International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Oxford, UK, 29 March–1 April 2016; pp. 81–82.
14. Fatemi Moghaddam, F.; Ahmadi, M.; Sarvari, S.; Eslami, M.; Golkar, A. Cloud computing challenges and opportunities: A survey. In Proceedings of the 2015 1st International Conference on Telematics and Future Generation Networks (TAFGEN), Kuala Lumpur, Malaysia, 26–28 May 2015; pp. 34–38.
15. Kumar, S.; Tyagi, M.; Khanna, A.; Fore, V. A survey of mobile computation offloading: Applications, approaches and challenges. In Proceedings of the 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), Paris, France, 22–23 June 2018; pp. 51–58.
16. Wu, G.; Li, Z.; Jiang, H. Quality of experience-driven resource allocation in vehicular cloud long-term evolution networks. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, e4036. [\[CrossRef\]](#)
17. Rebecchi, F.; Dias de Amorim, M.; Conan, V.; Passarella, A.; Bruno, R.; Conti, M. Data offloading techniques in cellular networks: A survey. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 580–603. [\[CrossRef\]](#)
18. Zhou, H.; Wang, H.; Chen, X.; Xu, S. Data offloading techniques through vehicular ad hoc networks: A survey. *IEEE Access.* **2018**, *6*, 65250–65259. [\[CrossRef\]](#)
19. Nshimiyimana, A.; Agrawal, D.; Arif, W. Comprehensive survey of V2V communication for 4G mobile and wireless technology. In Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSpNET), Chennai, India, 23–25 March 2016; pp. 1722–1726.
20. Pescosolido, L.; Conti, M.; Passarella, A. D2D data offloading in vehicular environments with optimal delivery time selection. *Comput. Commun.* **2019**, *146*, 63–84. [\[CrossRef\]](#)
21. You, C.; Huang, K.; Chae, H.; Kim, B.-H. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 1397–1411. [\[CrossRef\]](#)
22. Ren, J.; Yu, G.; He, Y.; Li, G.Y. Collaborative cloud and edge computing for latency minimization. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5031–5044. [\[CrossRef\]](#)
23. Chen, M.; Hao, Y. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 587–597. [\[CrossRef\]](#)
24. Shu, C.; Zhao, Z.; Han, Y.; Duan, Y.H. Multi-user offloading for edge computing networks: A dependency-aware and latency-optimal approach. *IEEE Internet Things J.* **2020**, *7*, 1678–1689. [\[CrossRef\]](#)
25. Kim, K.; Lynskey, J.; Kang, S.; Hong, C.S. Prediction based sub-task offloading in mobile edge computing. In Proceedings of the 2019 International Conference on Information Networking (ICOIN), Kuala Lumpur, Malaysia, 9–11 January 2019.
26. Huang, X.; Xu, K.; Lai, C.; Chen, Q.; Zhang, J. Energy-efficient offloading decision-making for mobile edge computing in vehicular networks. *EURASIP J. Wirel. Commun. Netw.* **2020**, *35*, 1–16. [\[CrossRef\]](#)
27. Zhang, J.; Guo, H.; Liu, J. Adaptive task offloading in vehicular edge computing networks: A reinforcement learning based scheme. *Mob. Netw. Appl.* **2020**, *25*, 1736–1745. [\[CrossRef\]](#)

28. Ali, Z.; Jiao, L.; Baker, T.; Abbas, G.; Abbas, Z.; Khaf, S. A deep learning approach for energy efficient computational offloading in mobile edge computing. *IEEE Access*. **2019**, *7*, 149623–149633. [[CrossRef](#)]
29. Wang, F.; Xu, J.; Cui, S. Joint offloading and computing optimization in wireless powered mobile-edge computing systems. *IEEE J. Sel. Areas Commun.* **2018**, *17*, 1784–1797. [[CrossRef](#)]
30. Mao, S.; Leng, S.; Maharjan, S.; Zhang, Y. Energy efficiency and delay tradeoff for wireless powered mobile-edge computing systems with multi-access schemes. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 1855–1867. [[CrossRef](#)]
31. Irshad, A.; Abbas, Z.H.; Ali, Z.; Abbas, G.; Baker, T.; Al-Jumeily, D. Wireless powered mobile edge computing systems: Simultaneous time allocation and offloading policies. *Electronics* **2021**, *10*, 965. [[CrossRef](#)]
32. Sommer, C.; German, R.; Dressler, F. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Trans. Mob. Comput.* **2011**, *10*, 3–15. [[CrossRef](#)]
33. Varga, A. The OMNeT++ discrete event simulation system. In Proceedings of the European Simulation Multiconference (ESM'01), Prague, Czech Republic, 6–9 June 2001.
34. Krajzewicz, D.; Hertkorn, G.; Rossel, C.; Wagner, P. SUMO (Simulation of Urban Mobility): An open-source traffic simulation. In Proceedings of the Fourth Middle East Symposium Simulation and Modelling (MESM'02), Sharjah, United Arab Emirates, 28–30 October 2002; pp. 183–187.
35. Chen, X.; Jiao, L.; Li, W.; Fu, X. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans. Netw.* **2016**, *24*, 2795–2808. [[CrossRef](#)]
36. Tse, D.; Viswanath, P. *Fundamentals of Wireless Communication*; Cambridge University Press: Cambridge, UK, 2005; pp. 175–178.
37. Wu, G.; Jiang, H. Spectrum sharing with dynamic cournot game in vehicle-enabled cognitive small-cell networks. *J. Comput. Netw. Commun.* **2019**, *2019*, 4835923. [[CrossRef](#)]