White Rose Research Online
eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Leeds.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/179974/

Version: Accepted Version

Article:
Bute, MS, Fan, P, Li, Z et al. (2021) An Efficient Distributed Task Offloading Scheme for Vehicular Edge Computing Networks. IEEE Transactions on Vehicular Technology, 70 (12). pp. 13149-13161. ISSN: 0018-9545

https://doi.org/10.1109/tvt.2021.3117847

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

An Efficient Distributed Task Offloading Scheme for Vehicular Edge Computing Networks

Muhammad Saleh Bute, *Student Member, IEEE,* Pingzhi Fan, *Fellow, IEEE,* Li Zhang, *Senior Member, IEEE,* and Fakhar Abbas, *Member, IEEE*

***Abstract*—With the recent advancement of vehicular ad-hoc networks (VANETs) or the internet of vehicles (IoVs), vehicles are getting more powerful and generating huge amount of traffic data, including computation-intensive and delay-sensitive applications in the vehicular edge computing (VEC) networks, which are difficult to be processed by an individual vehicular node. These resource-demanding tasks can be transferred to another vehicular node with idle computing resources for processing. Due to high mobility and limited resources of vehicular nodes, it is challenging to execute lengthy computation-intensive tasks until completion within the delay constraint. There is a need to provide an efficient task offloading strategies to support these applications. In this paper, an efficient distributed task offloading scheme is proposed to select nearby vehicles with idle computing resources, to process the tasks in parallel by considering some vital metrics, including link reliability, distance, available computing resources, and relative velocity. In order to complete the lengthy computation-intensive tasks in vehicular edge computing networks, a task is divided into several subtasks before offloading. The performance of the proposed scheme is evaluated in several VEC network conditions. Results show that the proposed computation task offloading scheme achieves better performance in latency, throughput, resource utilization and packet delivery ratio than the existing schemes.**

***Index Terms*—Computation-intensive, reliability, service vehicle, resource utilization, and infotainment.**

I. Introduction

THE evolution of the internet of vehicles (IoVs) brings about new technologies such as autonomous vehicles, remote driving, and vehicle platooning, requiring a scalable infrastructure to provide services adaptively. There are some heterogeneous vehicular applications of these emerging technologies, usually aiming at safety, infotainment, gaming, augmented reality and smart driving. However, due to limited resources to execute many of these tasks, vehicles have to offload their computation-intensive tasks to an edge server or remote cloud server for processing [1]. Thanks to the advancement in technology, vehicles are equipped with an onboard unit (OBU), which is a hardware that enables vehicle to everything (V2X) communication through the network interface card, installation of vehicular applications, task processing and other utility functions. The human machine interface (HMI) serves as an interface between the user and the machine for receiving input and displaying output. Some of the tasks generated by the vehicles are computation-intensive and delay-sensitive, which is beyond the computing power of a single vehicle. But offloading such tasks to a base station or remote cloud server will incur long transmission delay because of the huge distance from a moving vehicle to the server, and thus this approach can not meet the data rate and latency constraints of such tasks [2], [3]. The mobile edge computing (MEC) technology is adopted as a powerful paradigm for computation offloading because computing servers are located at the vehicle's proximity, such as the roadside unit (RSU) and the MEC server. Considering the high mobility of vehicles and the dynamic nature of vehicle edge computing network, the MEC servers are fixed at specific locations and can be off communication range of a vehicular node. In this case the idle computing resources on the vehicles can be utilized. Vehicles with such capability are referred to as service vehicles. They can act as mobile edge servers to augment the MEC servers and form the vehicular edge computing (VEC) network [4] with an enhanced computation offloading capacity. A number of technologies are available to support connectivity and data transmission [5], [6] between vehicular nodes in VEC network, including the IEEE 802.11p, dedicated short-range communication (DRSC) [7], device to device communication (D2D), and cellular networks such as the long-term evolution (LTE) and the fifth genertation network (5G) [8].

However, most of the previous works do not jointly consider vital metrics in selecting service vehicles for task processing. Instead, only one or two metrics are considered separately [12], [19]. These schemes might not complete the processing of lengthy computation intensive task within the delay constraint. [15]-[19] sacrifice computational resources for reliability, by replicating a single task and offloading them to several service vehicles within the communication range for processing, but the computation result from only one service vehicle will be used, while resources used by other service vehicles are squandered. A tradeoff between resource utilization and reliability is needed in the resource-constrained VEC network. In this paper, we discuss computation offloading in decentralized, self-organizing VEC network and propose an efficient distributed task offloading scheme, which focuses on how to select the optimal scattered idle computational

This work was supported by the NSFC Project No.61731017 and No.62020106001, National Key R&D Project No.2018YFB1801104, and the 111 project No.111-2-14.

M. S. Bute, and P. Fan are with the Key Lab of Information Coding & Transmission, Southwest Jiaotong University, Chengdu 610031, China (e-mail: msbute@my.swjtu.edu.cn; pzfan@swjtu.edu.cn).

L. Zhang is with the Institute of Communication and Power Networks, School of Electrical and Electronic Engineering, University of Leeds, Leeds LS2 9JT, U.K (email: l.x.zhang@leeds.ac.uk).

F. Abbas is with the Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad 46000, Pakistan (email: fakhar.14@hotmail.com).

resources. The proposed scheme jointly considers four vital metrics in service vehicle selection. These metrics includes both communication and computation factors in the VEC network. To improve the network capacity, the total task offloading cost is minimized. In order to encourage the service vehicles to augment the MEC, we introduce an incentive mechanism to reward the service vehicles for their services. An optimization problem is formulated, and it is solved in two stages. Firstly, the selection of service vehicles and then the task offloading decision. Moreover, to support the lengthy computation-intensive applications, a single task is divided into subtasks and offloaded to the selected service vehicles. The main contributions of this work are summarized as follows

- An efficient task offloading scheme for computation-intensive and delay-sensitive tasks is proposed to enhance resource utilization, task completion, reliability, and latency in the resource-constrained VEC network.
- The selection of optimal service vehicles is achieved using a fuzzy logic algorithm by jointly considering four metrics, which include link reliability, relative velocity, distance, and available computational resources. These factors influence the success of computation offloading in a VEC network.
- The effect of link lifetime and vehicular speed on lengthy task completion is investigated in order to reduce task waiting time and overall task offloading latency.
- The performance of the proposed scheme is evaluated by simulation. The simulation results show that the proposed scheme can significantly improve the performance in terms of latency, waiting time, resource utilization, and task offloading reliability.

The remainder of this paper is organized as follows. Section II presents the related works. In section III, we describe the proposed system model. Section IV presents the proposed task offloading scheme and problem formulation. Section V presents the simulation results and discussions. Section VI concludes the paper.

II. Related Work

The challenges of computation offloading in a high mobility vehicular scenario have attracted much research efforts over the years. Many offloading strategies aiming at optimizing communication and computation have been proposed.

MEC improves the vehicular ad-hoc networks' computational strength. In [9] and [10], the authors employ the vehicle to vehicle (V2V) communication between vehicular nodes to reduce the data traffic in the cellular network. If there exists a reliable route between the source and the destination nodes, then the traffic can be offloaded to the destination through the multi-hop V2V link. In such way, both the latency and the traffic burden are reduced. The authors in [11] utilize the vehicle to everything (V2X) communication to offload a task from the vehicle to MEC servers. Tasks are routed over the V2X link from the source vehicle to the MEC server. They optimize offloading delay and resource balancing on the MEC servers to improve system capability. However, the offloading strategies in [9]-[11] mainly depend on multi-hopping to access the MEC servers. Multi-hop links in VEC network with large hop count may not be reliable for efficient task delivery. The MEC network does not cover all road segments, and thus it is not accessible in some areas.

D. Souza et al, [12] employs the use of idle resources on vehicular nodes to offload computational tasks. In their approach, service vehicles are selected based on link duration and distance between nodes. Each computation task is then offloaded to a single service vehicle. The authors in [13] proposed a task scheduling scheme for task offloading in a vehicular cloud environment to minimize the task completion time on service vehicles in a quest to meet the delay requirements of these tasks. [14] investigates the effect of transferring delay-sensitive tasks to the MEC servers in an edge computing assisted vehicular network to minimize data transmission. The works in [12]-[14] studied computation offloading in vehicular edge computing network, where one or two metrics were separately considered in selecting service vehicles for task processing, thus the selected service vehicles might not be optimum.

The authors in [15]-[18] investigates the concept of task replication to improve task offloading efficiency and reliability, so that lengthy computationally intensive tasks can be completed before deadline. In this approach, a single task is offloaded to more than one service vehicle by choosing a number of service vehicles to host multiple processing of a single task. The result from only one service vehicle is returned. [19] employs the concept of flooding to offload a single task to more than one service vehicles. Then the result from the fastest service vehicle is transferred back to the task vehicle. The work in [20] highlighted the state of art computation offloading strategies in VEC network with emphasis on the effect of high mobility in the vehicular network, which is very difficult to model but also allows a vehicle to have contact with other vehicles frequently over a short period. They introduced a learning-based task replication algorithm for computation offloading. The concept of collaborative computation task offloading was presented in [21]. They proposed a two-stage computation offloading scheme in a vehicular edge environment. The concept of replicating a single task and transferring it to several service vehicles for processing improves offloading reliability and enables task execution until completion. However, considering the scarce resources in VEC, resource utilization is a crucial performance metric. The idea of multiple execution of a single task on several vehicles [15]-[19], tends to consume the limited resources in VEC network, incurring excessive communication and computation overheads.

III. System Model

Here we consider a multi-lane highway with a number of vehicular nodes. Vehicles are moving on the road following a normal distribution with respect to their positions and relative velocities. Assuming each vehicle is equipped with the IEEE 802.11p card, a cellular interface, and a global positioning system (GPS). There are two categories of vehicles moving on the road; the neighboring vehicles and vehicles in the need of assistance to process their tasks which are known as task vehicles (TV). At a particular time t, a TV can send an offloading request to its neighboring vehicles to assist in processing its task. The neighboring vehicle will return the execution result to the TV after processing.

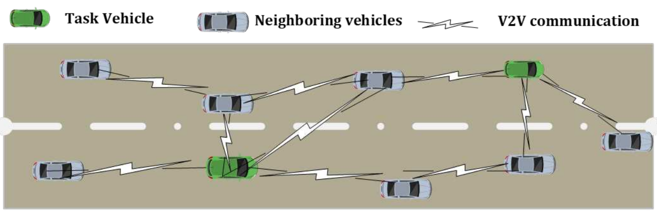


Fig. 1: System Model

As illustrated in Fig.1, a TV (green color vehicle) needs assistance to process a task, so it broadcasts an offloading request to the neighboring vehicles (vehicles in ash color). Upon reception of the request message, vehicles with idle computation resources known as service vehicles responds to the request message with some vital details like vehicle ID, moving direction, velocity, memory capacity, and available computational resources. Then the TV uses the received information to compute the performance value of each service vehicle using a fuzzy logic algorithm based on four metrics: distance, relative velocity, link reliability, and available computational resources. Then the vehicles with higher performance value will be selected as service vehicles for computation offloading.

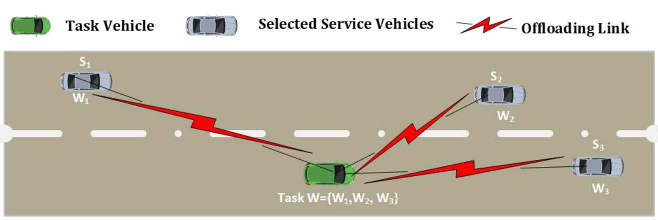


Fig. 2: Computation Offloading

IV. Proposed Computation Offloading Strategy

Consider a task vehicle TV_i initiating an offloading request by sending a broadcast message to neighbouring vehicles (NV) within its communication range, where $NV = \{nv_1, nv_2, nv_3, ..., nv_l\}$. It is always challenging to execute computation-intensive tasks, due to difficulties such as computation capability and wireless communication link breakage. Because of this, we divide each lengthy task into subtasks based on available service vehicles, and then forward each subtask from $W = \{w_1, w_2, w_3, ..., w_n\}$ to a service vehicle in $S = \{s_1, s_2, s_3, ..., s_m\}$ as illustrated in Fig 2. Each computation task can be defined in three terms as $w_i = \{d_i, c_i, T_i^{\max}\}$ where d_i is the total data size of the task, c_i represents the computing resources required to process the task, while $T_i^{\max}$ is the delay constraint of the task.

TABLE I: Main Notation Summary

Symbol	Description
W	Set of tasks
S	Set of service vehicles
NV	Set of neighboring vehicles
i	The task vehicle index $i \in n$
j	The service vehicle index $j \in m$
$r_t(l)$	The link reliability
$q_{i,j}$	The uplink data rate of each vehicle
P_i	The transmission power of each vehicle
$G_{i,j}$	The channel gain between two vehicles
B_0	The system bandwidth
d_i	The total data size of a task w_i
c_i	The computing resources required to process task w_i
$T_i^{\max}$	The delay constraint of task w_i
$L_{i,j}^r$	The task transmission latency
$L_{i,j}^e$	The task execution latency
$L_{i,j}^t$	The total task offloading latency
$LLT_{i,j}$	The link life time
$cst_{i,j}$	The total task offloading cost
$u_{i,j}$	Indicates whether task w_i is offloaded to service vehicle s_j
α	The weight of task offloading latency
β	The weight of task processing cost
γ_j	The unit cost of resource on service vehicle s_j

A. Service Nodes Selection

In this part, to achieve an optimal computation offloading decision in the VEC network, we present a fuzzy logic algorithm to select service vehicles. The service vehicle selection is affected by various factors in a VEC environment because of its dynamic nature and high mobility. Both communication and computation factors have to be considered jointly for reliable computation offloading. The fuzzy inference system (FIS) combines all input data to reflect the impact of the parameters in decision making. To make use of the fuzzy logic algorithm in service vehicle selection, it is crucial to find the factors that directly impact the vehicular nodes [22], [23]. Fig. 3 illustrates the proposed FIS model.

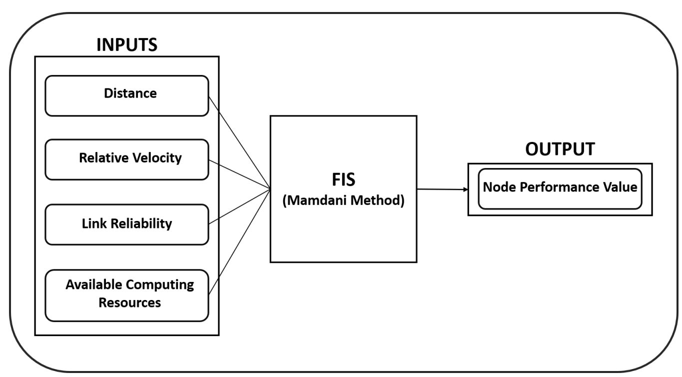


Fig. 3: Fuzzy System Control Model

B. Nodes Performance Value Using Fuzzy Logic

To calculate each service vehicle's performance value, four metrics: distance, relative velocity, link reliability, and available computation resources are measured by the task vehicle for all the service vehicles in its communication range that respond to its offloading request. The above metrics are transformed into fuzzy values and fuzzy rules based on the defined membership functions. Finally, fuzzy values are transformed into numerical values. Description of these metrics are given below

1) Distance (DT)

A lower relative distance between the vehicular nodes signifies lower packet transmission latency, and a more stable network connection. Service vehicles closer to the task vehicles have higher chance to be selected as service vehicles. When the distance is far between the vehicles, the connection between them is unstable and hence it is not suitable for task offloading. The distance between two nodes (Vh_i, Vh_j) can be computed using their respective coordinates (x_i, y_i) and (x_j, y_j).

$$D_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{1}$$

2) Link Reliability (LR)

Link reliability is the probability that a communication link between vehicles Vh_i and Vh_j will be available over a period of time [24], the link reliability $r(l)$ can be given as

$$r(l) = \Pr\{l \text{ is continuously available until } t+T_r\,|l \text{ exists at } t\} \tag{2}$$

where l denotes a communication link, and T_r represents the predicted interval for continuous existence of the link at t. The velocities of the vehicles are obtained to compute the link reliability, assuming the velocities follow normal distribution. The probability density function (pdf) of the velocity v of a vehicle [25] can be given as

$$h(v) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(v-\varphi)^2}{2\sigma^2}} \tag{3}$$

where φ is the mean value of velocity and σ^2 is the variance of velocity. Since the velocities of vehicle Vh_i and Vh_j follow normal distribution, their relative velocity $\Delta v_{i,j} = |v_i - v_j|$ is also a normally distributed variable. Let CR denote the communication range of a vehicle, then $2CR$ is the maximum communication distance between two vehicles. Therefore, the link duration T must satisfy

$$T = \frac{2CR}{\Delta v_{i,j}} \tag{4}$$

The expression above is monotonous and differentiable, based on equation (3) using the changing variable rules, we can obtain the pdf of T, which is given as

$$f(T) = \frac{2CR}{\sigma_{\Delta v_{i,j}}\sqrt{2\pi}} \frac{1}{T^2} e^{-\frac{\left(\frac{2CR}{T} - \varphi_{\Delta v_{i,j}}\right)^2}{2\sigma^2 \Delta v_{i,j}}} \quad \text{when } T \geq 0 \tag{5}$$

where $\sigma_{\Delta v_{i,j}}$ and $\varphi_{\Delta v_{i,j}}$ represent the mean and variance of the relative velocity respectively. Let T_r denote the continuous existence of communication link l between vehicles Vh_i and Vh_j, which can be expressed as

$$T_r = \frac{CR - D_{i,j}}{|v_i - v_j|} \tag{6}$$

where $D_{i,j}$ is the distance between vehicle Vh_i and Vh_j. $f(T)$ in (5) can be integrated from t to $t + T_r$, to obtain the probability that, at a certain time t the communication link l will continue to exist for a duration T_r. Therefore, the link reliability model can be expressed as

$$r_t(l) = \begin{cases} \int_{t}^{t+T_r} f(T)dT, \text{ if } T_r > 0 \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

Using the Gaussian error function erf, we can derive the integral in equation (7). Then we have

$$r_t(l) = \text{erf}\left(\frac{\frac{2CR}{t} - \varphi_{\Delta v_{i,j}}}{\sigma_{\Delta v_{i,j}}\sqrt{2}}\right) - \text{erf}\left(\frac{\frac{2CR}{t+T_r} - \varphi_{\Delta v_{i,j}}}{\sigma_{\Delta v_{i,j}}\sqrt{2}}\right) \tag{8}$$

3) Relative Velocity (RV)

The relative velocity $RV_{i,j}$ between two moving vehicular nodes is the velocity of vehicle Vh_i with respect to the vehicle Vh_j [26], [27]. Using the offloading request response from the one-hop neighboring service vehicles, the task vehicle TV_i will compute the relative velocity from each service vehicle. The relative velocity can be obtained using

$$RV_{i,j} = |v_i - v_j| \tag{9}$$

where v_i denotes the velocity of vehicle Vh_i and v_j denotes the velocity of vehicle Vh_j. A smaller value of relative velocity signifies a small variation in velocity between vehicle Vh_i and Vh_j. Hence network connection will be suitable for successful task transfer between the vehicular nodes.

4) Available Computational Resources (AC)

The available computational resource, is referred to as the computational capability of a vehicular node consisting of the random-access memory (RAM), processing unit (CPU) and the storage, this metric estimates the potential computation power of a vehicle. To determine ∂ which is the value of the idle resource, we use the ratio of the allocated resources to the total amount of resource on the service vehicle. The available computational resources on vehicle Vh_j can be expressed as

$$AC_j = \partial_j f_j \tag{10}$$

where f_j denote the CPU frequency of the vehicle. The available computional resources at time t on a service vehicle s_j, determines the computation delay of a task and whether the task can execute till completion within the delay constraint. The higher the available computional resources, the lower the task processing latency.

C. Fuzzy Sets

The major objective of the proposed fuzzy logic algorithm is to select reliable service vehicles based on the defined metrics. In conventional set theory, elements can either belong to a set or not. Whereas fuzzy theory extends this definition by introducing partial membership. A fuzzy set Z in a universe of discourse Y can be defined as

$$\mu_Z(y) : Y \to [0,1] \text{ where } y \in Y \tag{11}$$

The notation $\mu_Z(y)$ represents the membership degree of y in Z. $\mu_z(y) = 1$ if $y \in Z$ denotes full membership, $\mu_z(y) = 0$ if $y \notin Z$ denotes non membership, and $0 < \mu_Z(y) < 1$ indicates partial membership. A finite fuzzy set can be expressed as

$$Z = \{\mu_Z(y_1)/y_1 + \mu_Z(y_2)/y_2 + \mu_Z(y_3)/y_3 + ... + \mu_Z(y_n)/y_n\} \tag{12}$$

In fuzzification process, a crisp value is converted into a fuzzy value. The fuzzy logic uses a linguistic variable to represent an input parameter. The value of each linguistic variable is a real number within a defined range. Therefore, a linguistic variable can be denoted as $LV = \{Vb, Rn, \delta\}$ where Vb denotes the value of fuzzy input, Rn represents the range of the variable, and δ is the fuzzy set. Table II presents the fuzzy sets.

TABLE II: Fuzzification of Input and Output Variables

Input variable	**Fuzzy sets**
Distance (DT)	far, close
Relative Velocity (RV)	low, medium, high
Link reliability (LR)	reliable, adquate, not-reliable
Available computational resource (AC)	adquate, high
Output variable	**Fuzzy sets**
Performance value (PFV)	worse, bad, fair, average, good, execellent

D. Membership Functions

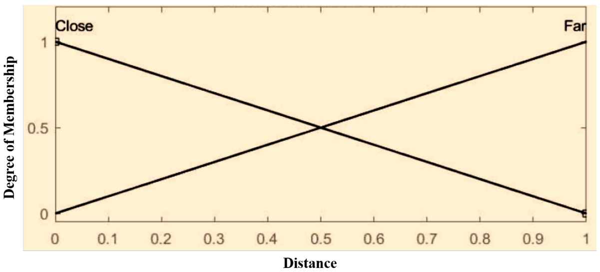
The membership function defines how each point in the input space is mapped to a membership value within the range of $[0, 1]$, which indicates the membership degree. Membership functions are designed by dividing each linguistic variable into an overlapping fuzzy sets, which were obtained through experiment. A set of membership function for each fuzzy variable can be expressed as

$$Z_F = \{(y, \mu_Z(y)) : y \in Y, \mu_Z(y) \in [0,1]\} \tag{13}$$

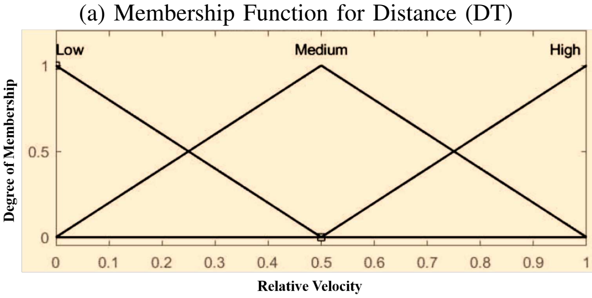
where $\mu_Z(y)$ is the membership function of Z, it also indicates the degree at which y belongs to Z. In this work, based on our input and output variables we have five membership function sets, the membership functions are designed in a triangular format due to its low complexity and flexibility, which can be expressed as

$$\mu_Z^{tri}(y) = \begin{cases} 0, & \text{if} \quad y < a \\ \frac{y-a}{b-a}, & \text{if} \quad a \le y \le b \\ \frac{c-y}{c-b}, & \text{if} \quad b \le y \le c \\ 0, & \text{if} \quad y > c \end{cases} \tag{14}$$

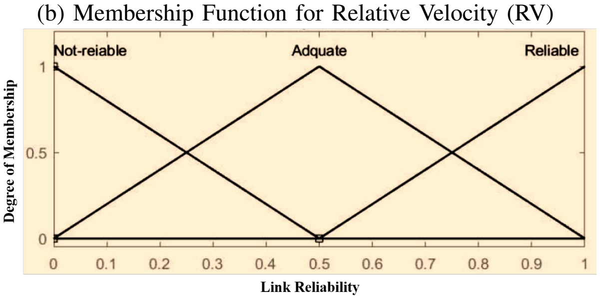
where a denotes the lower limit of the triangular curve, c denotes the upper limit of the triangular curve, and b denotes the modal value of the triangular curve. Fig. 4 illustrates the membership functions used in this work.



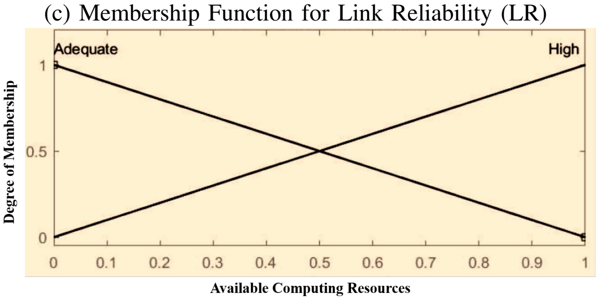
(a) Membership Function for Distance (DT)



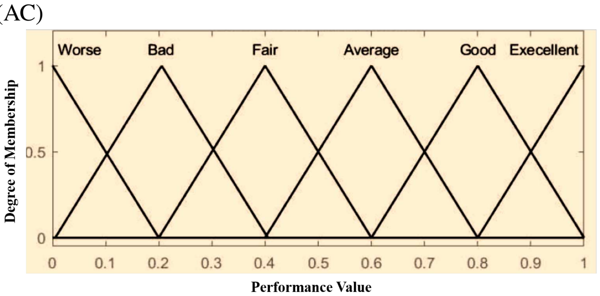
(b) Membership Function for Relative Velocity (RV)



(c) Membership Function for Link Reliability (LR)



(d) Membership Function for Available Computing Resources (AC)



(e) Membership Function for Performance Value (PFV)

Fig. 4: Membership Functions for Input and Output Fuzzy Variables

E. Fuzzy Rules

The fuzzy rules are defined based on the If-Then logical operation, each rule represents a fuzzy implication between a condition and conclusion. An output can only be generated after evaluation of the fuzzy rules, to obtain the desired result it is essential to carefully design the fuzzy rules. The set of fuzzy rules in Table III are formulated based on the Mamdani inference model [35]. A set with K number of If-Then rules is defined below.

$$Rl_x : \text{IF } o \text{ is } O_x, \text{Then } u \text{ is } U_x \tag{15}$$

In the expression above O_x and U_x are fuzzy sets, assuming $x \in \{1, 2, 3, ..., K\}$. The fuzzy inference system maps the values of the fuzzy input to output based on the defined rules. Lastly, in the defuzzification step, the result of the fuzzy inference is converted from linguistic value to a numeric value. For the proposed scheme, the center of gravity (COG) method is utilized for the defuzzification process, which can be expressed as

$$\eta = \frac{\int y.\mu_z(y)dy}{\int \mu_z(y)dy} \tag{16}$$

F. Communication and Computation Model

The computation offloading delay comprises two major aspects i.e. the transmission and the execution, which eventually determine the latency and the reliability of a task offloading scheme. In the task transmission process, the task vehicle communicates with service vehicles using a single hop network structure. We consider the frequency division multiple access method, where users are assigned a fraction of the total bandwidth. The data rate for transmission from a task vehicle to the service vehicle can be given as

$$q_{i,j} = B_0 \log_2 \left(1 + \frac{P_i G_{i,j}}{N_0}\right) \tag{17}$$

where B_0 denotes the allocated bandwidth, P_i is the transmission power of the task vehicle, $G_{i,j}$ is the channel gain between the communicating vehicles, and N_0 represents the additive Gaussian noise power. The task processing on the service vehicle involves task transmission and execution. The transmission latency can be expressed as

$$L^r_{i,j} = \frac{d_i}{q_{i,j}} \tag{18}$$

where d_i is the data size of the task. Then the task execution latency, can be defined as

$$L^e_{i,j} = \frac{c_i}{fz_j} \tag{19}$$

where c_i is the computing resources required to execute the task, and fz_j represents the computing resources allocated to process the task. Therefore, the total offloading latency which is the sum of transmission and execution latencies can be expressed as

$$L^t_{i,j} = L^r_{i,j} + L^e_{i,j} \tag{20}$$

TABLE III: Set of Fuzzy Rules

	Input				Output
Rules	**LR**	**RV**	**DT**	**AC**	**PFV**
1	Reliable	Low	Close	High	Excellent
2	Reliable	Low	Close	Adequate	Good
3	Reliable	Low	Far	High	Good
4	Reliable	Low	Far	Adequate	Average
5	Reliable	Medium	Close	High	Good
6	Reliable	Medium	Close	Adequate	Average
7	Reliable	Medium	Far	High	Average
8	Reliable	Medium	Far	Adequate	Fair
9	Reliable	High	Close	High	Average
10	Reliable	High	Close	Adequate	Fair
11	Reliable	High	Far	High	Fair
12	Reliable	High	Far	Adequate	Bad
13	Adequate	Low	Close	High	Good
14	Adequate	Low	Close	Adequate	Average
15	Adequate	Low	Far	High	Fair
16	Adequate	Low	Far	Adequate	Bad
17	Adequate	Medium	Close	High	Average
18	Adequate	Medium	Close	Adequate	Fair
19	Adequate	Medium	Close	High	Average
20	Adequate	Medium	Close	Adequate	Fair
21	Adequate	High	Far	High	Fair
22	Adequate	High	Far	Adequate	Bad
23	Adequate	High	Close	High	Fair
24	Adequate	High	Close	Adequate	Bad
25	Not-reliable	Low	Far	High	Bad
26	Not-reliable	Low	Far	Adequate	Worse
27	Not-reliable	Low	Close	High	Fair
28	Not-reliable	Low	Close	Adequate	Bad
29	Not-reliable	Medium	Far	High	Bad
30	Not-reliable	Medium	Far	Adequate	Worse
31	Not-reliable	Medium	Close	High	Fair
32	Not-reliable	Medium	Close	Adequate	Bad
33	Not-reliable	High	Far	High	Bad
34	Not-reliable	High	Far	Adequate	Worse
35	Not-reliable	High	Close	High	Bad
36	Not-reliable	High	Close	Adequate	Worse

Also, when a task is offloaded to a service vehicle for processing, the energy consumption in the task offloading process is computed based on the transmission cost for transferring a content to the service vehicle. Thus, the energy consumption for sending d_i bits of data from the task vehicle to the service vehicle is given as

$$E_{i,j} = P_{i,j}.L^r_{i,j} \tag{21}$$

G. Cost Function

Let γ_j denote the unit cost of a resource in the service vehicle and fz_j the processing resources assigned by the service vehicle, the offloading reward paid by the task vehicle to the service vehicle is given as

$$\varsigma_{i,j}^t = \gamma_j . fz_j \tag{22}$$

Therefore, the task offloading cost can be expressed as the sum of total latency and the price paid for the offloading services, then we have

$$cst_{i,j} = \varsigma_{i,j}^t + L_{i,j}^t \tag{23}$$

H. Problem Formulation

Time is divided into time slots $t \in \Psi = \{0, 1, 2, ..., t_n\}$. At each time slot, a subtask is allocated to at most one service vehicle for execution. A binary variable $u_{i,j}$ indicates whether subtask w_i is offloaded to a service vehicle s_j for execution.

$$u_{i,j} = \begin{cases} 1, & \text{if subtask } w_i \text{ is offloaded to service vehicle } s_j \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

Accordingly, the major objective is to minimize the task offloading cost, which includes communication and computation overhead. Here the offloading cost is defined as the weighted sum of latency and processing cost $\alpha L_{i,j}^t + \beta \varsigma_{i,j}^t$, where α and β are the weights of latency and processing cost respectively. The decision weights can be adjusted dynamically, depending on the preference of the user. When a task has a stringent delay requirement, the latency weight α is set to a larger value. Similarly, when the processing cost is of serious concern, then β is set larger. Therefore, the optimization problem can be formulated as

$$\begin{aligned} \underset{\{u_{i,j}, cst_{i,j}, E_{i,j}\}}{\text{minimize}} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} (\alpha L_{i,j}^t + \beta \varsigma_{i,j}^t) u_{i,j} \\ \text{subject to} \quad & \text{C1: } \sum_{i=1}^{n} \sum_{j=1}^{m} q_{i,j} \leq Q, \forall i \in n, j \in m, \\ & \text{C2: } E_{i,j} \leq E^h, \forall i \in n, \\ & \text{C3: } L_{i,j}^r + L_{i,j}^e \leq T_i^{\max}, \forall i \in n, j \in m, \\ & \text{C4: } r_t^{i,j}(l) > 0, \forall i \in n, j \in m, \\ & \text{C5: } 0 \leq fz_j \leq f^{\max}, \forall j \in m, \\ & \text{C6: } u_{i,j} \in \{0, 1\}, \forall i \in n, j \in m \end{aligned} \tag{25}$$

C1 defines the data rate constraint. C2 characterizes the maximum limit of energy consumption. C3 guarantees that the transmission and execution latency of a task should not exceed the delay requirement. C4 guarantees that the transmission link between the task vehicle and the service vehicle is reliable. C5 indicates that the computing resources allocated shouldn't exceed the maximum available resources. C6 ensures that a subtask cannot be allocated to more than one service vehicle.

To solve the optimization problem above, algorithms with exponential computation complexity have to be applied, which is not suitable for real-time decision making in a dynamic VEC network. Task allocation in the VEC network is a special case of 0/1 Knapsack problem [38], which is NP-hard. Therefore, we propose heuristic algorithms to obtain task offloading decisions and resource allocation. Algorithm 1 selects a set of service vehicles, and algorithm 2 provides the task offloading decision.

I. Computation Task Offloading

To meet delay requirements and completion of tasks, selecting appropriate service vehicle is necessary. It is crucial to optimize the communication latency, computation latency, and also avoid the system instability that may degrade system's performance. In the proposed scheme, both communication and computation factors are jointly considered in the fuzzy system to compute the performance value of each service vehicle. Vehicles with higher performance value are selected for task execution as presented in algorithm 1. For safety concerns, in algorithm 2 we consider social relationship which involves discovering the interaction pattern among users i.e., social networks. Using the pattern, we can estimate the level of trust among users based on previous their interactions. $X_{i,j}$ represents the level of trust between users, which is within the range $0 \leq X_{i,j} \leq 1$. Assuming the probability of secure communication is denoted by the degree of trust between the users, computation offloading request can only be accepted if $X_{i,j}$ is greater than or equal to a predefined threshold.

J. Task Completion and Waiting Time

A task can only be offloaded if a service vehicle is in a task vehicle's communication range. Otherwise, a generated task waits until when there is a service vehicle in the communication range. A task that cannot be processed within the delay requirement is declared failed and has to be re-initiated. Conventionally in a VEC network, computation offloading can be successful only if the link duration between a service vehicle and a task vehicle is higher than the task's maximum latency. Otherwise, the task has to wait for a service vehicle with its desired link duration. However, in a highway scenario as presented in Fig. 5, having a longer link duration is always difficult. In the proposed scheme, task offloading latency is optimized with consideration of link reliability, link lifetime, and task completion. A task is divided into subtasks and offloaded to service vehicles for execution, with the concept of multiprocessing even if the link duration is less than the delay requirement of a task. In this way, the task can be processed within its delay constraint.

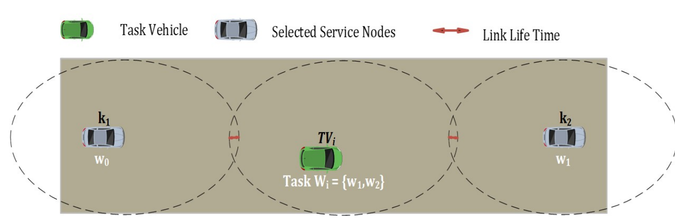


Fig. 5: Link Life Time

Algorithm 1: Service Vehicles Selection Algorithm

```
inputs: Ω, W
output: set of selected service vehicles S
i ← n, j ← m
foreach TV_i ∈ Ω do
    ▷ TV_i is a task vehicle, Ω is a vehicular graph topology
    broadcast task offloading request
end
foreach V_i ∈ Ω do
    ▷ V_i is a vehicle, Ω is a graph topology of vehicles
    if recieve offloading request ==1 then
        V_i is a neighbor
        V_i ∈ NV
        if V_i reply request ==1 then
            V_i is a service vehicle
            V_i ∈ S̄
            foreach s̄_j ∈ S̄ do
                calculate the value of each metric using equations (1-11)
            end
        else
            V_i ∉ S̄
            V_i is not a service vehicle
        end
    else
        recieve offloading request == 0
        V_i is out of communication range
    end
    N_s ← count(S̄)
        ▷ Number of available service vehicles
end
Service vehicle selection (S̄, N_s)
foreach s̄_j ∈ S̄ do
    compute PFV using fuzzy logic algorithm
    sort PFV in descending order
    return S
    ▷ S is the set of service vehicles with high PFV
    if |S| ≥ 1 then
        There are available service vehicles with higher performance value
    else
        |S| < 1
        no service vehicle with performance value above threshold
        retry offloading task later
    end
end
```

1) Link Life Time

The link lifetime (LLT) is an estimated duration for which a transmission link l will exist between vehicular nodes (Vh_i, Vh_j) [28], [29]. LLT can be obtained by the equation below

$$LLT_{i,j} = \frac{-(ef+gh)+\sqrt{(e^2+g^2)r^2-(eh-fg)^2}}{e^2+g^2} \tag{26}$$

where $e = v_i \cos\theta_i - v_j \cos\theta_j$, $f = x_i - x_j$, $g = v_i \sin\theta_i - v_j \sin\theta_j$ and $h = y_i - y_j$ while (i, j) are two vehicular nodes, v_i and v_j are their velocities, (x_i, y_i) and (x_j, y_j) are the positions of the nodes, θ_i and θ_j $(0 \le \theta_i, \theta_j \le 2\pi)$ are their directions of motion, respectively. Note that when vehicles are moving thesame direction $\theta_i = \theta_j$. When a task is generated, the task vehicle computes the link lifetime between a task vehicle and the service vehicle. The estimated link duration is used to determine whether to offload a task or to wait for a better link.

The link lifetime LLT can be used in measuring link stability between nodes. Consider a vehicle Vh_i in a set of vehicular nodes. The link stability of vehicle Vh_i with respect to vehicle Vh_j can be given as

$$LS_{i,j} = \frac{\sum\limits_{Vh_j \in NV_i} LLT_{i,j}}{\sum\limits_{Vh_j \in NV_i} \sum\limits_{Vh_k \in NV_j} LLT_{j,k}} \tag{27}$$

Where NV_i denotes a set of one-hop neighbors of vehicle Vh_i, NV_j denotes a set of one-hop neighbors of vehicle Vh_j, $LLT_{i,j}$ represents the link duration between vehicle Vh_i and Vh_j, and $LLT_{j,k}$ is the link duration between vehicle Vh_j and Vh_k. From equation (27) the link lifetime is proportional to the link stability. When the link duration is high, the link stability is also high [36], [37]. For a task w_i to be transmitted over a link with data rate $q_{i,j}$ between vehicle Vh_i and vehicle Vh_j, the probability of successful data transmission can be expressed as

$$Ps_{i,j} = Pr\{t'_{ij} \ge \frac{d_i}{q_{i,j}}\} \tag{28}$$

Assuming $t_{i,j} = \frac{d_i}{q_{i,j}}$ is the minimum required duration between vehicle Vh_i and Vh_j, $t'_{i,j}$ is the average link duration between the vehicles, and d_i represents the size of the task.

V. Simulation Results and Discussions

To evaluate the proposed computation task offloading scheme, the microscopic traffic simulator (SUMO v1.5) is used to generate vehicular traffic and mobility [30]. In the experiments a 2500 x $2500m^2$ area is considered. A multi-lane road is generated, vehicles are moving with random speed, position, and arrival time following a normal distribution. We follow the urban and highway simulation scenario, as stated in 3GPP TR [31]. Furthermore, for network simulation, the mobility traces obtained through SUMO were passed to the network simulator (NS3.8.1) [32] to simulate the vehicular ad-hoc network (VANETs) scenario with parameters stated in table IV [33], [20], [34].

Algorithm 2: Task Offloading Decision

inputs: S, TV_i, $threshold$, W
output: offloading decision
$i \leftarrow n$
$j \leftarrow m$
foreach $s_j \in S$ **do**
  if $X_{i,j} \geq threshold$ **then**
    ▷ social relationship exists
    $calculate\ LLT_{i,j}$
    ▷ LLT is the link life time
  else
    do not accept offloading request
  end
  $return\ LLT_{i,j}$
end
if $LLT_{i,j} \geq T_i^{\max}$ **then**
  $obtain\ Ps_{i,j}$
  TV_i offloads task w_i
  s_j processes and return results
else
  the value of $LLT_{i,j}$ is not enough
  request rejected
end

TABLE IV: Summary of Simulation Parameters

Parameter	Value
Simulation Area	2500 x $2500m^2$
Simulation Time	500s
Number of Vehicles	20-500
Service Vehicles Ratio	15%
Transmission Range	250-300m
Task Size	5-10mb
Vehicle Speed	40-130km/h
Connection Type	UDP

A. Simulation Environment

The proposed scheme is compared against other benchmark schemes.

- Distance aware offloading scheme [12]: in this scheme, service vehicles for task offloading are selected based on the relative distance between the task vehicle and the service vehicle, if the link duration is enough to complete a task.
- Task replication offloading scheme [20]: is a learning algorithm that replicates task and offloads tasks to all available service vehicles. The algorithm learns the average offloading delay of each service vehicle gradually before convergence.
- Two-stage task offloading scheme [21]: the algorithm operates in two phases; at the first stage, a cluster is formed. In the next phase offloading request is sent through the cluster head, which selects the service vehicles based on vehicular velocity and transmission link.

In the simulation, two types of experiments are conducted, with 30 simulations executed for each experiment to obtain averaged results.

Experiment 1: the proposed computation offloading scheme is evaluated with several vehicular densities. The performance metrics considered are as follows

- End to End Delay: is the time consumed to offload a task from task vehicle to service vehicle to process and return the result to the task vehicle. Delay is usually measured in seconds.
- Resource Utilization: is a performance metric that indicates how the resource in a network is efficiently utilized. Resource utilization is measured in percentage.
- Successful Offloading: is a performance metric that indicates the percentage of successful offloads over the total number of offloads.
- Communication Overhead: is a metric that indicates communication resources used by vehicles in the procedure of offloading a particular task.

Experiment 2: the proposed computation offloading scheme is evaluated with several vehicular speed. the performance metrics considered are as follows

- Throughput: is the actual amount of packets successfully transmitted between a task vehicle and service vehicle within time t. Throughput is presented in megabits per second $(mbps)$.
- Packet Delivery Rate (PDR): this is the ratio of the number of packets successfully delivered over the total number of packets sent from a task vehicle to the service vehicle. PDR is measured in percentage.
- Size of Task: is the size of the task offloaded to a service vehicle for processing. Task size is measured in a megabyte.

B. Discussions

In this section, we present and discuss the simulation results.

1) Performance Over Varying Number of Vehicles

In experiment 1, urban road traffic is considered with a density of 100-500 vehicles. The vehicle's maximum speed limit is set at 20m/s. To analyze the effect of computation-intensive and delay-sensitive task, a task of size $5Mb$ is generated on a random vehicle for offloading at a particular time slot after a specified interval. At each simulation session, five task offloading requests are successfully generated, and the result is recorded.

In Fig. 6, the offloading delay is analyzed with respect to vehicle density and the number of selected service vehicles. In this section, the number of selected service vehicles is two. Fig. 6 shows that as the number of vehicles increases in the network, the offloading delay reduces. This is because the relative distance between vehicles is higher when the vehicular density is lower. In such case, there are fewer number of service vehicles and the task vehicle may not be able to find an appropriate service vehicle. However, as the number of vehicles increases, the relative distance between nodes reduces, and a more significant number of service vehicles will be available, which will enable improved offloading decisions by jointly minimizing communication and computation latency. The proposed computation offloading scheme has much lower offloading latency than the existing schemes [12], [20], [21]. This is because the introduction of the four metrics in service

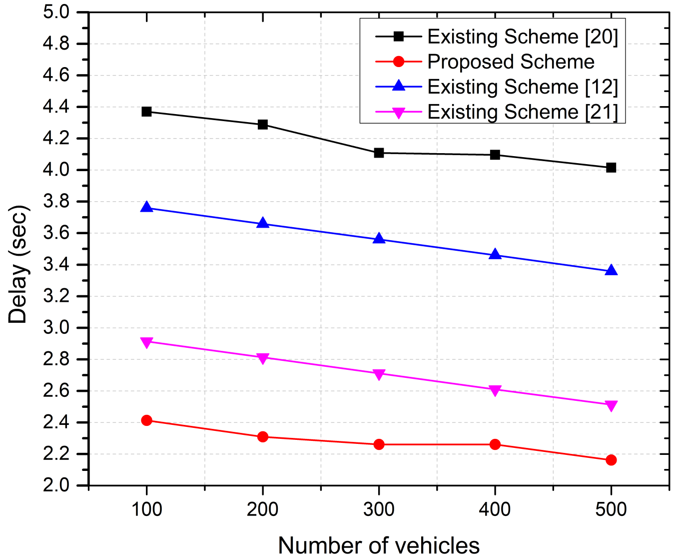


Fig. 6: Offloading delay with two selected service vehicles

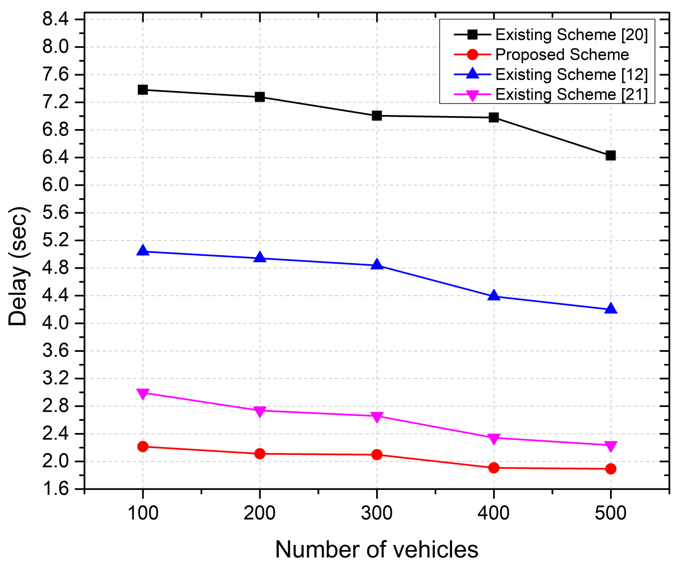


Fig. 7: Offloading delay with four selected service vehicles

vehicle selection, which is different with all other existing schemes, has immensely minimized the offloading delay.

Fig. 7 shows that when the number of selected service vehicles is increased to four, the delay in [20] increases exponentially because of the task transmission latency, which is related to its flooding nature where a single task is transmitted to all the available service. Also, in [12], the delay increases because of its single processing nature, in which the selected service node might not be optimal. In [21], delay reduces because of its multiprocessing approach, but this scheme's service node selection does not consider distance and link reliability, which are vital factors to consider in minimizing the transmission delay. The proposed scheme's delay reduces further because of the multiprocessing nature and the selection of service nodes based on four vital metrics, which jointly minimizes transmission and computation delay, supporting the completion of computation-intensive tasks.

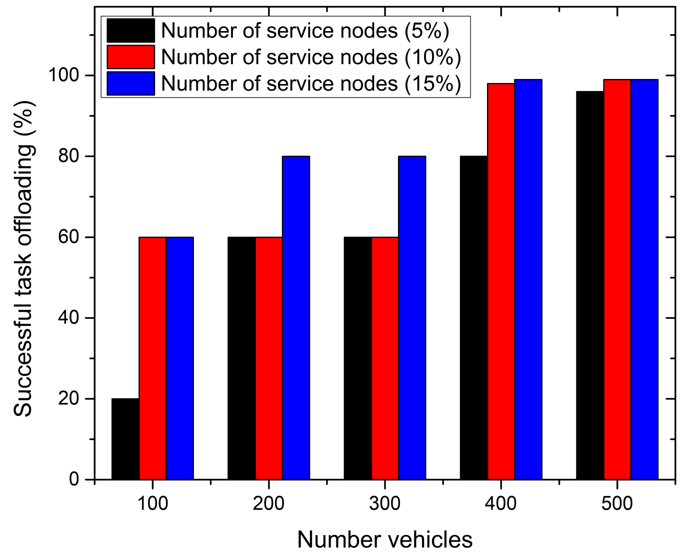


Fig. 8: Successful offloading with respect to the percentage of service nodes

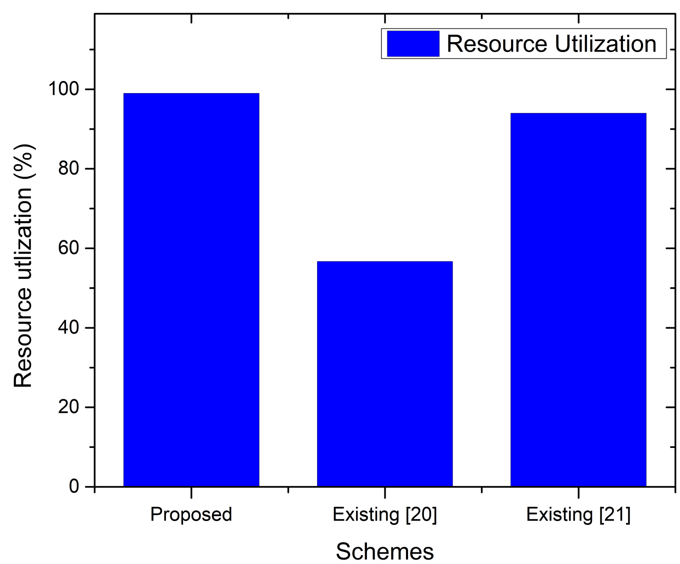


Fig. 9: Resource utilization

Fig. 8 shows the percentage of successful task offloading for different percentages of service vehicles in the network. It can be seen when the ratio of service vehicles is 5% the network recorded a low success of task offloading. When the number of vehicle is lower due to the vehicular sparsity in the network, service vehicles may not be available. Even if there is a service vehicle available, the relative distance is high, and the link between service vehicle and task vehicle is unstable. As the number of vehicles increases, the percentage of service vehicles also increases, providing more offloading opportunity, as a result, the success rate of offloading increases.

Fig. 9 shows that [20] recorded lower resource utilization because a task is offloaded to all available service vehicles, and the result from only one service vehicle is returned to the task vehicle. In this case communication and computation resources on other service vehicles are wasted. The proposed scheme achieves the highest resource utilization because service nodes are selected based on communication and computation metrics. [21] also recorded higher resource utilization because of the multiprocessing approach exploited by the scheme.

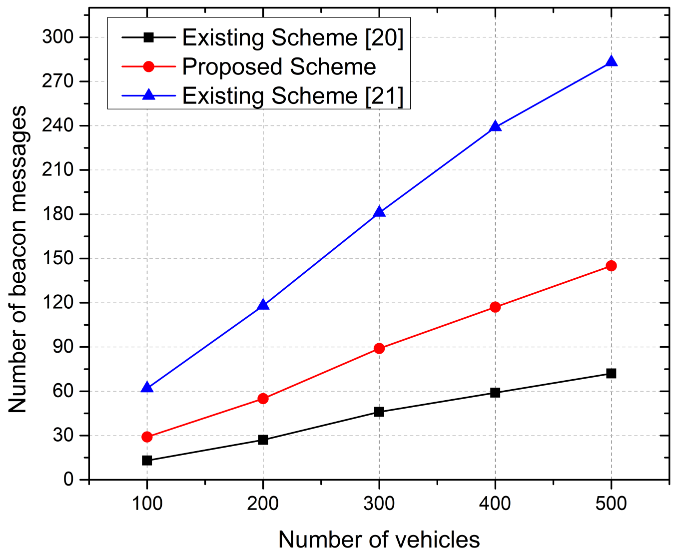


Fig. 10: Communication overhead

Communication overhead is related to the number of messages exchanged among vehicles to initiate task offloading. Fig. 10 shows that [20] recorded the lowest overhead because it is a learning-based scheme. A task is offloaded to all service vehicles in the communication range, without prior knowledge of available resources on the service nodes. The scheme learns and improves by the response of the environment. [21] incurs the highest communication overhead because of the two-stage offloading approach, where a task is offloaded through an edge vehicle to the service vehicles to process. And this involves much more beaconing between vehicles. The proposed scheme recorded a fair overhead because service vehicles are selected based on some vital metrics. The information is obtained through communication between vehicles before task offloading.

2) Performance with Different Vehicle Speeds

In experiment 2, highway road traffic is generated with a density of 20 vehicles. The vehicle's speed is between 20-34m/s. A task of size 5-10mb, with the delay constraint of 5-10s is generated on a random vehicle after a specified interval. At each simulation session, five offloading requests are successfully conducted, and the average result is recorded for analysis.

In Fig. 11, tasks offloading delay with respect to the size of a task is evaluated. As the size of the task increases, the offloading delay increases as expected. The task delay requirement is set between 5-10s to evaluate all schemes and analyze whether the tasks can be processed within the minimum delay requirement. The task's size determines the task waiting time. Suppose the estimated computation delay of

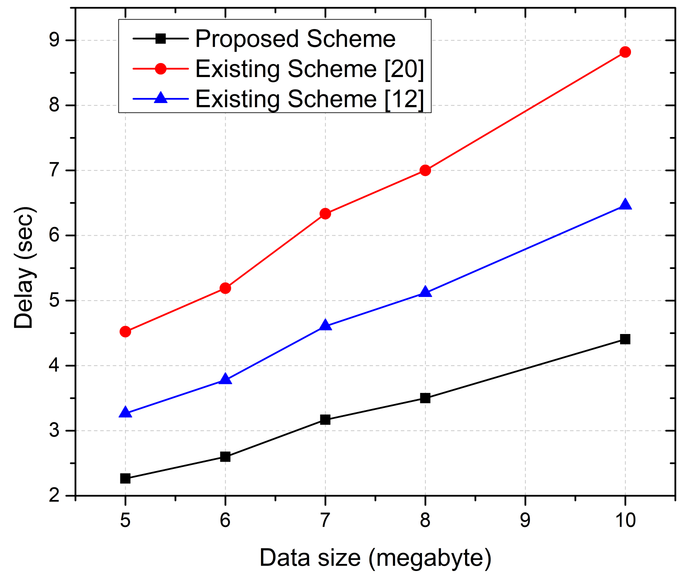


Fig. 11: Impact of waiting time on offloading delay

the task is higher than the link lifetime between two vehicles. The task has to wait for a link with a higher link duration to enable successful offloading. The proposed scheme recorded lower offloading latency because a task is divided into subtasks before offloading, thereby minimizing the waiting time of a task. Also, the scheme selects the optimal service vehicles, which reduces the task transmission latency. In comparison, [12] and [20] incur much waiting time.

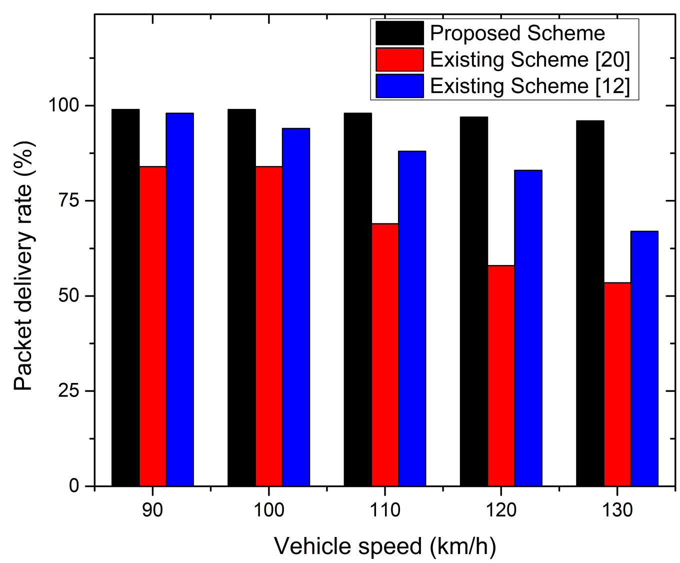


Fig. 12: Packet delivery rate

The effect of vehicle speed on the packet delivery rate is shown in Fig. 12. The packet delivery rate of [20] is lower because the packets need to be transferred to all available service vehicles. Therefore, the packet queue has to drop some packet that cannot be transferred to a service vehicle before the maximum queuing delay. [12] recorded less packet loss because a task is transferred to a single service vehicle, so the number of packets transmitted is not much compared

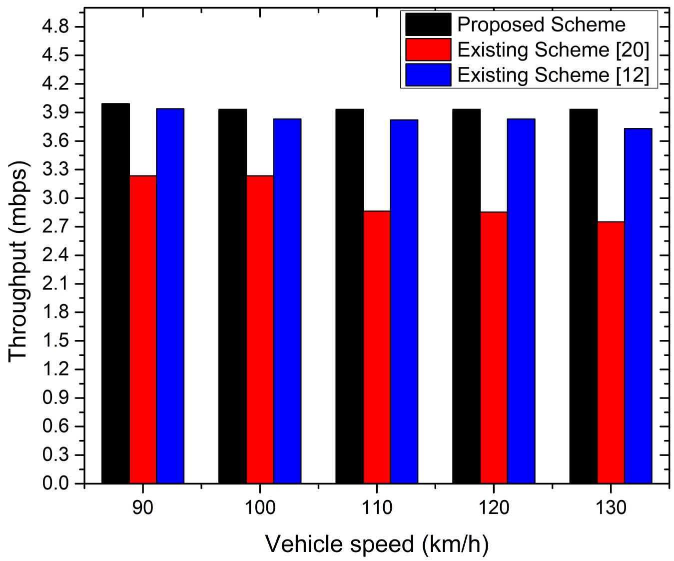


Fig. 13: Throughput

to [20]. The proposed scheme recorded the highest packet delivery rate because a task is divided into subtasks before transmission to respective service vehicles. Therefore, the number of packets is less for each destination, so packet loss is minimized. Fig. 13 shows that the proposed scheme achieves the highest throughput because it has the highest successful packet delivery than the other schemes.

VI. Conclusions

A distributive task offloading scheme is proposed to support task completion for delay-sensitive and computation-intensive tasks in vehicular edge computing networks. The fuzzy logic algorithm is applied to select the optimal number of service vehicles for task execution by jointly considering some vital metrics such as distance, relative velocity, link reliability, and available computation resources of the service vehicles. We further investigate the effect of vehicular speed and task waiting time on computation offloading in sparse vehicular density scenarios. Through extensive simulations, we have shown that the proposed scheme achieves significantly better performance in terms of throughput, latency, resource utilization and packet delivery ratio in comparison with existing schemes.

References

[1] Hou Xiangwang, et al,“Reliable Computation Offloading for Edge-Computing-Enabled Software-Defined IoV,” *IEEE Internet of Things Journal*, Vol. 7, No. 8, 2020.

[2] F. Jingyun, L. Zhi, W. Celimuge, and J. Yusheng, ”AVE: Autonomous vehicular edge computing framework with ACO-based scheduling,” *IEEE Transactions on Vehicular Technology*, Vol. 66, No. 12, 10660–10675, 2017.

[3] Cheng Ziqing, et al, ”Computation Offloading and Resource Allocation for Mobile Edge Computing,” *in Proc. IEEE SSCI*, pp. 2735–2740, 2019.

[4] L. Yi, Y. Huimin, X. Shengli, and Z. Yan,“Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks,” *IEEE Transactions on Vehicular Technology*, Vol. 68, No. 11, 11158–11168, 2019.

[5] Southwest Research Institute, ”Basic Infrastructure Message Development and Standards Support for Connected Vehicles Applications,” pp. 35, 2018.

[6] 5GAA, ”Timeline for deployment of C-V2X – Update,” White Paper, pp. 1–12, 2019.

[7] Gundlach, Michael, ”Use Cases for 3GPP Based V2X and Combined Solutions,” *in Proc. 10th ETSI ITS Workshop*, Sophia Antipolis, France, Mar. 2019, pp. 4-6.

[8] Technical Specification, ”ETSI TS 136 302 - V15.2.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Services provided by the physical layer (3GPP TS 36.302 version 15.2.0 Release 15),” pp. 0–32, 2019.

[9] H. Chung-Ming, C. Meng-Shu, D. Duy-Tuan, S. Wei-Long, X. Shouzhi, and Z. Huan, ”V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture,” *IEEE Access*, Vol. 6, 17741–17755, 2018.

[10] D. Zizhen, C. Zhen, and L. Mangui,“A Multi-Hop VANETs-Assisted Offloading Strategy in Vehicular Mobile Edge Computing,” *IEEE Access*, Vol. 8, 53062–53071, 2020.

[11] X. Xiaolong, X. Yuan, L. Xiang, Q. Lianyong, and W. Shaohua,“A computation offloading method for edge computing with vehicle-to-everything,” *IEEE Access*, Vol. 7, 131068–131077, 2019.

[12] A. B. de Souza, P. A. Leal Rego and J. N. de Souza, ”Exploring Computation Offloading in Vehicular Clouds,” *in Proc. IEEE 8th International Conference on Cloud Networking (CloudNet)*, Coimbra, Portugal, 2019, pp. 1-4.

[13] G. Minyeong, and A. Sanghyun, ”Computation Offloading-Based Task Scheduling in the Vehicular Communication Environment for Computation-Intensive Vehicular Tasks,” *in Proc. IEEE ICAIIC*, pp. 534–547, 2020.

[14] J. Huang, Y. Qian and R. Q. Hu, ”A Vehicle-Assisted Data Offloading in Mobile Edge Computing Enabled Vehicular Networks,” *in Proc. 2019 IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, 2019, pp. 1-6.

[15] J. Zhiyuan, Z. Sheng, G. Xueying and N. Zhisheng,“Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic,” *IEEE Internet of Things Journal*, Vol. 5, No. 1, 93–107, 2018.

[16] L. Chen and J. Xu, ”Task Replication for Vehicular Cloud: Contextual Combinatorial Bandit with Delayed Feedback,” *Proc. in IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Paris, France, 2019, pp. 748-756

[17] Y. Sun, J. Song, S. Zhou, X. Guo and Z. Niu, ”Task Replication for Vehicular Edge Computing: A Combinatorial Multi-Armed Bandit Based Approach,” *Proc. in 2018 IEEE Global Communications Conference (GLOBECOM)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1-7.

[18] Li. Kuikui, T. Meixia, and C. Zhiyong,“Exploiting computation replication for mobile edge computing: A fundamental computation-communication tradeoff study,” *IEEE Transactions on Wireless Communications*, Vol. 19, No. 7, 4563-4578, 2020.

[19] L. Bo, P. Ziyi, H. Peng, H. Min, A. Marco, and J. Gwanggil,“Reliability and capability based computation offloading strategy for vehicular ad hoc clouds,” *Journal of Cloud Computing*, Vol. 8, No. 1, 21, 2019.

[20] Z. Sheng, S. Yuxuan, J. Zhiyuan and N. Zhisheng,“Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks,” *IEEE Communications Magazine*, Vol. 57, No. 5, 49–55, 2019.

[21] B. Su, G. Siri, W. Celimuge, Z. Jiefang, Y. K. Alvin, and J. Yusheng, ”Collaborative Vehicular Edge Computing Towards Greener ITS,” *IEEE Access*, Vol. 8, 63935–63944, 2020.

[22] H. Abdulmughni, S. Mohammad, A. Omar and T. Eyad,“Energy-efficient fuzzy-logic-based clustering technique for hierarchical routing protocols in wireless sensor networks,” *Sensors*, Vol. 19, No. 3, 561, 2019.

[23] H. M. Delowar, S. Tangina, N. VanDung, N. Tri DT, H. Luan NT, H. Eui-Nam,“Fuzzy Based Collaborative Task Offloading Scheme in the Densely Deployed Small-Cell Networks with Multi-Access Edge Computing,” *Applied Sciences*, Vol. 10, No. 9, 3115, 2020.

[24] W. Xiufeng, W. Chunmeng, C. Gang and Y. Qing,“Practical link duration prediction model in vehicular ad hoc networks,” *International Journal of Distributed Sensor Networks*, Vol. 11, No. 3, 216934, 2015.

[25] A. Fakhar and F. Pingzhi,“Clustering-based reliable low-latency routing scheme using ACO method for vehicular networks,” *Vehicular Communications*, Vol. 12, 66–74, 2018.

[26] Z. Khan, P. Fan, F. Abbas, H. Chen and S. Fang,“Two-level cluster based routing scheme for 5G V2X communication,” *IEEE Access*, Vol. 7, 16194–16205, 2019.

[27] L. Won–Il, P. Jae–Young, L. Y. Sun, L. Sang–Woong,“Relative velocity based vehicle–to–vehicle routing protocol over ad–hoc networks,” *Inter-*

national Journal of Ad Hoc and Ubiquitous Computing, Vol. 12, No. 1, 14–22, 2013.

[28] Ahmed Izhar, et al,“Reliable coverage area based link expiration time (LET) routing metric for mobile ad hoc networks,” *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, Vol. 28 LNICST, pp. 446–476, 2010.

[29] K. Gaurav, C. S. Kumar, and S. Sieteng, ”Reliability evaluation of mobile ad hoc networks by considering link expiration time and border time,” *International Journal of System Assurance Engineering and Management*, Vol. 10, No. 3, 399–415, 2019.

[30] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzew, ”“Sumo–simulation of urban mobility: An overview,,” *in Proc. The Third International Conference on Advances in System Simulation (SIMUL)* , Vol. 14, pp. 63–68, 2011.

[31] Technical Report, ”3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Release 16 Description; Summary of Rel-16 Work Items (Release 16),” document 3GPP TR 21.916 V0.1.0, Sep. 2019.

[32] B. Sliwa, J. Pillmann, F. Eckermann, L. Habel, M. Schreckenberg and C. Wietfeld, ”Lightweight joint simulation of vehicular mobility and communication with LIMoSim,” *in Proc. IEEE Vehicular Networking Conference (VNC)*, pp. 81-88, 2017.

[33] Technical Specification, ”ETSI TS 123 285 V14.2.0 - Universal Mobile Telecommunications System (UMTS); LTE; Architecture enhancements for V2X services (3GPP TS 23.285 version 14.2.0 Release 14),” pp. 1–36, 2017.

[34] M. Sudip and B. Samaresh,“Soft-VAN: Mobility-Aware Task Offloading in Software-Defined Vehicular Network,” *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 2, pp. 2071–2078, 2019.

[35] Chen, G., Pham, T. T., and Boustany, N. M., ”Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems”, *ASME. Appl. Mech Rev.*, Vol 54, No. 6: B102–B103, pp. 57-109, 2001.

[36] Z. Zhang, P. Zhang, D. Liu and S. Sun, ”SRSM-Based Adaptive Relay Selection for D2D Communications,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2323-2332, Aug. 2018.

[37] J. Cheng et al., ”Accessibility Analysis and Modeling for IoV in an Urban Scene,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4246-4256, April 2020.

[38] C. Tang, X. Wei, C. Zhu, Y. Wang and W. Jia, ”Mobile Vehicles as Fog Nodes for Latency Optimization in Smart Cities,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9364-9375, Sept. 2020.

Muhammad Saleh Bute (Student Member, IEEE) received an M.Sc. degree in Computer Science from the University of Ilorin, Kwara State, Nigeria, in 2014. He is currently pursuing a Ph.D. degree with the Key Laboratory of Information Coding and Transmission, School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. He is also an academic staff with the Computer Science Department, Gombe State University, Gombe State, Nigeria. His research interests include mobile edge computing (MEC), vehicular ad-hoc networks (VANETs), 5G technologies, Cellular V2X, and radio resource management.

Pingzhi Fan (Fellow, IEEE) received the M.Sc. degree in computer science from Southwest Jiaotong University, Chengdu, China, in 1987, and the Ph.D. degree in electronic engineering from the University of Hull, Kingston upon Hull, U.K., in 1994. He is currently a Distinguished Professor and the Director of the Institute of Mobile Communications, Southwest Jiaotong University, China. He has been a Visiting Professor with the University of Leeds, Leeds, U.K., since 1997, and a Guest Professor with Shanghai Jiaotong University since 1999. He has authored or coauthored more than 300 research articles in various international journals and eight books. He is the inventor of 25 granted patents. His research interests include high mobility wireless communications, massive random access techniques, and signal design and coding. He is an IEEE VTS Distinguished Speaker from 2019 to 2022 and a fellow of IET, CIE, and CIC. He was a Board Member of IEEE Region 10, IET(IEE) Council, and IET Asia-Pacific Region. He was a recipient of the U.K. ORS Award in 1992, the NSFC Outstanding Young Scientist Award in 1998, the IEEE VTS Jack Neubauer Memorial Award in 2018, the IEEE SP Soc SPL Best Paper Award in 2018, the IEEE WCSP 10-Year Anniversary Excellent Paper Award from 2009 to 2019, and the IEEE or CIC ICCC Best Paper Award in 2020. He was the general chair or the TPC chair of a number of international conferences. He is the Founding Chair of IEEE VTS Beijing Chapter and IEEE ComSoc Chengdu Chapter, and IEEE Chengdu Section.

Li Zhang (Senior Member, IEEE) received her PhD in Communications at the University of York in 2003. Currently she is an Associate Professor and leads the Wireless Communication Group at the school of Electronic and Electrical Engineering, University of Leeds, UK. Her research interest is focused on wireless communications and signal processing techniques, such as massive MIMO, mmWave communications, Heterogeneous Network, Device to Device communications and 5G systems etc. She has served on the Technical Programme Committees of most major IEEE conferences in communications since 2006 and is an associate editor of IEEE journal. She has been selected as a member of the prestigious UK EPSRC Peer Review College since 2006, and regularly helps reviewing grant applications for Research councils and book proposals. She has been PhD examiner for numerous Universities. In 2005, she received a Nuffield award for a newly appointed lecturer. In 2006, she became a fellow of Higher Education Academy. In 2011, she was awarded as IEEE exemplary reviewer and in 2012 she was promoted as senior IEEE member.

Fakhar Abbas (Member, IEEE) received the Ph.D. degree in Information and Communication Engineering from the Key Laboratory of Information Coding and Transmission School of Southwest Jiaotong University (SWJTU), Chengdu China in 2020 and the M.S. degree in Information and Communication Engineering from Harbin Engineering University, Harbin, China, in 2015. He is currently working with the Computer Science Department, COMSATS Univeristy Islamabad(CUI), Islamabad Campus, Pakistan. He has been a reviewer for several IEEE journals and major conferences. His current research interests include wireless communications, vehicular ad-hoc network, 5G/6G cellular networks, C-V2X, routing protocols, Network Security, Mobile Edge Computing (MEC), offloading, IoTs, radio resource management and algorithm designs. He was a recipient of the Best Presentation Award of International Academy of Computer Technology (IACT) International Conference 2015. Conference 2015.