A Collaborative Task Offloading Scheme in Vehicular Edge Computing

Muhammad Saleh Bute[1], Pingzhi Fan[1], Gang Liu[1], Fakhar Abbas[2], and Zhiguo Ding[3]
[1]School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China
[2]Center of Intelligent Networking & Communications, The University of Electronic Science and Technology of China
[3]School of Electrical and Electronic Engineering, The University of Manchester, Manchester M13 9PL, U.K.
Email: msbute@gmail.com

2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring) | 978-1-7281-8964-2/20/$31.00 ©2021 IEEE | DOI: 10.1109/VTC2021-Spring51267.2021.9448975

***Abstract*—The increase of mobile applications in the internet of vehicles (IoVs), necessitates the demand for higher computation capabilities. Vehicles can transfer related applications to another nodes for processing. In this paper, an efficient task offloading scheme for cellular vehicle to everything (C-V2X) is proposed to improve offloading reliability and latency. Vehicles are grouped into clusters, where vehicles in need of assistance can transfer their task to other vehicles for processing through the vehicle to vehicle (V2V) link, or transfer their task to the mobile edge computing (MEC) server via the vehicle to network (V2N) link. Matching theory is exploited for the task assignments. Simulation results reveals that the proposed scheme performs better than the existing schemes.**

***Index Terms*—Vehicle to vehicle (V2X), vehicular edge computing (VEC), matching, clustering.**

I. Introduction

As an enabling technology of the intelligent transportation systems (ITS), vehicular edge computing (VEC) supports the coordination and sharing of resources among vehicles. The increase of mobile applications triggers the explosive growth in the demand for high computational capabilities [1]. To support these applications, the idea of computation offloading came into existence, where a vehicle can transfer a task to another node for processing. The mobile edge computing (MEC) technology is a powerful paradigm for computation task offloading because computing servers are provided at the vehicle's proximity, such as the roadside unit (RSU) where the MEC servers are located. With the cellular vehicle to everything (C-V2X) technology, which is one of the crucial features of the fifth-generation (5G) networks [2]. The C-V2X is designed on two air interfaces; the Uu cellular interface, which is the conventional uplink/downlink interface that supports vehicle to network (V2N) communication, and the side-link (PC5) interface for direct communication among nodes. This interface supports the vehicle to vehicle (V2V) communication and vehicle to infrastructure (V2I) communication.

This paper proposes an efficient task offloading scheme to enhance latency and reliability based on existing schemes [3]. The key objective is to exploit the V2V and V2N communication links of the C-V2X for task offloading. Vehicles are grouped to form clusters based on V2V, where cluster heads are selected based on some vital metrics. Clustering is to enhance link reliability between nodes and minimize communication overhead on the cellular network. Vehicles in need of assistance can offload their computation task to vehicles in its cluster, neighboring cluster, or the MEC server. The V2V link is utilized for task offloading to other vehicles, while the V2N is used for offloading tasks to the MEC server.

The remainder of this paper is organized as follows. Section II describes the proposed system model. Section III presents the proposed computation offloading scheme. Section IV presents the task assignment algorithm. Performance evaluation is presented in Section V. Section VI concludes the paper.

II. System Model

Consider a multilane highway road with a random distribution of vehicles, having a total number of N vehicular nodes on the road, following a normal distribution with respect to their position and relative velocity. Assuming each vehicle is equipped with the Uu cellular interface for V2N communication, the PC5 interface for direct V2V communication, and a global positioning system (GPS) device. To ease the communication overhead on the cellular network, vehicles are grouped to form clusters, instead of each vehicle communicating directly to the network. Only the cluster head is responsible for communicating with the network through the eNodeB on behalf of other vehicles. As shown in Fig. 1, there are three categories of vehicles. The green color vehicles called the VEC servers, the VEC servers, are vehicles with idle computational resources that offer services to other vehicles. The ash color vehicles are the task vehicles. The task vehicles are vehicles with limited resources and are in need of a proxy server to offload their task for execution. The ordinary vehicles are in red color. The MEC server is co-located with the eNodeB, vehicles can access the MEC server through the eNodeB using the V2N connection.

III. Proposed Computation Offloading Scheme

A. Cluster Formation

Cluster is formed through V2V communication using a greedy iterative algorithm. At first, a set Q is obtained for several vehicles requesting to form a cluster, then a collection of some vital metrics for cluster head selection is introduced.

978-1-7281-8964-2/21/$31.00 ©2021 IEEE

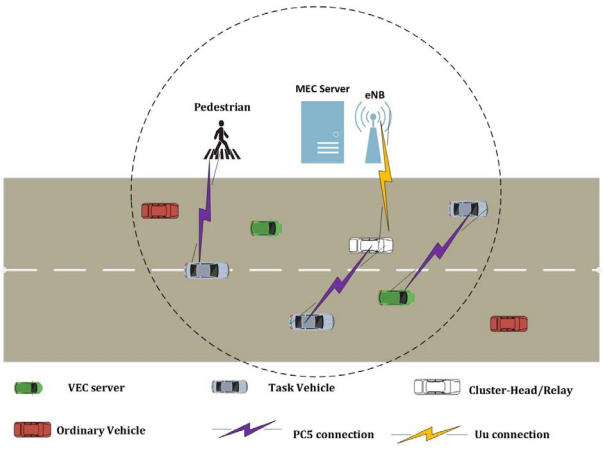

Fig. 1. System Model

The metric includes link lifetime, velocity, and mean distance between communicating vehicles. Each vehicle's metric is computed, the vehicle with the least value in the group emerges the cluster head (CH). The link lifetime [4], [5] between a pair of vehicles i and j can be given as

$$Lt = \frac{-(ef+gh) + \sqrt{(e^2+g^2)r^2 - (eh-fg)^2}}{e^2+g^2} \quad (1)$$

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 with transmission range r, 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 $\theta_j (0 \leq \theta_i, \theta_j \leq 2\pi)$ are their directions of motion, respectively.

From equation (1) we can deduce relative velocities and distance between pair of vehicles i and j as $Rv_{ij} = v_i - v_j$ and $Rd_{ij} = d_i - d_j$. Let Lt_{av}, Rv_{av} and Rd_{av} denotes average link lifetime, average relative velocity and average relative distance respectively. The CH selection value of each vehicle can be computed as

$$M = (1 - \frac{Lt_{av}}{Lt_{\max}}) + x(\frac{Rd_{av}}{Rd_{\max}}) + y(\frac{Rv_{av}}{Rv_{\max}}) \quad (2)$$

where x and y denotes the relative importance of position and velocity. $Lt_{\max}$, $Rd_{\max}$, and $Rv_{\max}$ represents the maximum value of each metric respectively.

B. Computation Task Processing Models

A computation task generated on a vehicle is divided into a number of subtasks before offloading, and each subtask is denoted as $T_i = (S_i, C_i, Z_i^{\max})$ here S_i denotes the size of the task, C_i denotes the amount of computing resources need to execute T_i, while $Z_i^{\max}$ is the maximum latency of the task. A vehicle can offload a computation task to a VEC server or the MEC server for processing.The offloading decision of a vehicle can be represented as $loc_i \in \{0,1\}, \forall_i$. $vec_i \in \{0,1\}, \forall_i$ denotes decision to offload a task to the VEC server, and $mec_i \in \{0,1\}, \forall_i$ represents decision to offload a task to the MEC server.

Algorithm 1: Cluster Formation Algorithm

inputs : $Q, Velocity, Position$
output: CH
initialization:
$i \leftarrow n$
$Nl = 0$
foreach $V_i \in Q$ **do**
  $broadcast$ hello message
  ▷ V_i denotes a vehicle, and Q is graph topology of vehicles
  if *message recieved == 1* **then**
    $V_i \in Nl$
    $Nl = Nl + 1$
  else
    message not recieved
    $V_i \notin Nl$
    ▷ Nl is the list of neighbors
  end
  return Nl
end
foreach $V_i \in Nl_i$ **do**
  $compute\ M$ using equation (2)
  $sort\ M$ in ascending order
  CH $\leftarrow min(M)$
end
return CH

1) Local Processing: In the local computing, the vehicles process their computing task with the available resources on the vehicle. The task processing latency can be given as

$$lat_{lc}^{sum} = \frac{C_i}{a_{loc}} \quad (3)$$

here C_i is the computation resources needed to execute task T_i, a_{loc} denotes the computation resources allocated by the vehicle.

2) VEC Processing: When a vehicle decides to offload a task to a VEC server $vec_i = 1$, in this case, the task is transmitted to a VEC server through the PC5 V2V interface for processing. Therefore, the spectral efficiency of the transmission link between vehicles [6] Vh_i and Vh_j can be given as

$$RVU_i = w\log_2(1 + \frac{P_i h_{i,j}}{I_{i,j} + \sigma^2}) \quad (4)$$

where w denotes the allocated bandwidth, P_i is the transmission power of vehicle Vh_i. The channel gain is represented by $h_{i,j}$, I_{ij} is the interference caused by other vehicles, and σ^2 denotes the noise power. Therefore, the offloading latency consists of task transmission time and task execution time, which can be expressed as

$$lat_{vc}^{sum} = \frac{S_i}{RVU_i} + \frac{C_i}{a_{vec}} \quad (5)$$

where S_i is the size of the task, RVU_i denotes the data rate, C_i is the computation resources needed to execute task T_i, a_{vec} denotes the computation resources allocated by the VEC server.

3) MEC Processing: If a vehicle decides to offload its task to the MEC server then $mec_i = 1$, a task is transmitted from a vehicle to the MEC through the V2N link. The spectral efficiency of the transmission link between vehicle Vh_i and MEC sever m_n can be expressed as

$$RMU_i = w\log_2(1 + \frac{P_i h_{i,n}}{I_{i,n} + \sigma^2}) \tag{6}$$

where w denotes the allocated bandwidth, P_i is the transmission power of vehicle Vh_i, the channel gain between vehicle v_i and MEC sever m_n is represented by $h_{i,j}$, $I_{i,n}$ is the interference from neighboring cells, and σ^2 denotes the noise power. The offloading latency, which includes the task transmission time and task execution time, can be given as

$$lat_{mc}^{sum} = \frac{S_i}{RMU_i} + \frac{C_i}{a_{mec}} \tag{7}$$

where S_i is the size of the task, RMU_i denotes the data rate, C_i represents the computation resources needed to execute task T_i, a_{mec} denotes the computation resources allocated by the MEC server.

C. Problem formulation

To minimize the total offloading latency of a task, this includes the transmission and execution latencies, which can be expressed as

$$lat_{i,j}^{sum} = lat_{i,j}^{tr} + lat_{i,j}^{ex} \tag{8}$$

The utility function presents the overall benefit that a task T_i can receive if assigned to server S_j. To reduce overall latency and task offloading cost, a utility function is defined as

$$x_{ij} = l_i - \alpha lat_{i,j}^{tr} - \beta C_i \tag{9}$$

where l_i is the reward if a task T_i is completed within delay requirement on the server S_j, β is the unit price per central processing unit cycle , α is the delay cost coefficient. The utility of server S_j for completing task T_i is given as

$$y_{ij} = \beta C_i - \alpha lat_{i,j}^{ex} \tag{10}$$

let a_{ij} denote a binary variable indicating whether a task is assigned to server. Therefore, maximizing overall utility is equivalent to minimizing overall offloading latency, hence, we have

$$\arg\max_{\{a_{ij}\}} \sum_{i=1}^{N}\sum_{j=1}^{M}(x_{ij} + y_{ij})a_{ij} \tag{11}$$

Consequently, the above maximization problem can be presented as

$$\begin{aligned}
\underset{\{a_{ij}\}}{\text{minimize}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}(lat_{i,j}^{tr} + lat_{i,j}^{ex})a_{ij} \\
\text{subject to} \quad & C1: \sum_{i=1}^{N} a_{ij} \leq 1, \forall_j \in M, \\
& C2: lat_{lc}^{sum}, lat_{vc}^{sum}, lat_{mc}^{sum} \leq Z_i^{\max}, \forall_i \in N, \\
& C3: loc_i + vec_i + mec_i = 1, \forall_i \in N, \\
& C4: x_{ij}a_{ij} \geq 0, \forall_i \in N, \\
& C5: y_{ij}a_{ij} \geq 0, \forall_i \in N
\end{aligned} \tag{12}$$

C1 guarantee that each task T_i is matched to at most one edge server S_j. C2 makes sure that a task is completed within the delay requirement. C3 guarantees that each task can be processed locally or offloaded either to the VEC server, or the MEC server. C4 and C5 guarantees that no utility value of either task T_i or edge server S_j should be negative. The optimization problem above is a binary linear programming (BLP) problem, which is NP-hard [7]. So many instances are intractable. Therefore, we apply a heurisitc algorithm to solve the problem.

IV. Matching Game Task Assignment

A task vehicle can offload a computation task to either a VEC server or a MEC server. We assume the MEC server possesses several virtual computing units (VU). Each MEC server has n number of virtual computing units, VU^{max}= $\{VU_1, VU_2, ..., VU_n\}$. The set of available VEC servers is given as $SV = \{SV_1, SV_2, ..., SV_m\}$. Time is divided into multiple time slots, while a task is divided into subtasks. The set of available edge servers, which includes the local processing unit on the task vehicle, VEC servers and MEC server's virtual computing units at time t in the network, is given as $S = \{S_0, S_1, ..., S_m\}$ and set of subtasks from a task vehicle is given as $T = \{T_1, T_2, ..., T_n\}$. Matching theory can be used in pairing a task with a server for processing [8], [9]. The matching function is defined as

$$\mu : \{T\} \leftrightarrow \{S\} \tag{13}$$

where the items in set T and S are matched satisfying constraints (C1-C5).

1) Stable Matching Process: A stable matching model operates on finite sets [10]. A member from one set is matched to a member from another set based on their preference list, which is obtained based on utility value. A member from one set needs to propose to get engaged with a member on the other set. Each member's preference list is vital, because in a matching algorithm, the preference list determines the partnership's stability between a pair of members.

A partner selection decision is based on the utility function defined in equation (9-10). A task ranks its preference based on high utility value if matched an edge server. Similarly, an edge server ranks its preference based on high utility value if matched with a task.

Algorithm 2: Proposed Stable Matching Algorithm

inputs : T, S, Pt, Ps
output: μ
initialization
$\mu = \emptyset$
while $T \neq \emptyset$ **do**
  foreach $i \in T$ **do**
    T_i proposes to its most preferred edge server
  end
  foreach $j \in S$ **do**
    if $j \in S$ *prefers* T_i *over* T'_i **then**
      $j \in S$ *rejects* T'_i and *accept* T_i
      ▷ $\mu : \{T_i\} \leftrightarrow \{S_j\}$ generates higher utility value
      update Pt_i, Ps_j
    else
      $j \in S$ *rejects* T_i
      ▷ $\mu : \{T'_i\} \leftrightarrow \{S_j\}$ generates higher utility value
      update Pt_i, Ps_j
    end
  end
  return μ
end

V. Performance Evaluation

In this section, simulation results are analyzed to evaluate the proposed scheme's performance in C-V2X networks. We consider the simulation of a highway scenario presented in 3GPP TS [11] and generate a multiple lane highway road running over a cell in which an eNodeB is situated. Vehicular nodes are randomly positioned on the road with initial velocity randomly generated within the range of [5-28] m/s for each node. All simulations are performed using network simulator NS 3.28.1 [12] in conjunction with MATLAB. Table I list the key simulation parameters [13], [14].

TABLE I
Summary of Simulation Parameters

Simulation Parmeter	Value
eNodeB Transmission	5km
Bandwidth	20Mhz
Simulation Area	$1000m \times 4000m$
Maximum Speed Limit	28m/s
Task Size	100Kb - 1Mb
Energy Consumption Level	20 - 85W
CPU Frequency	1 - 4Ghz
Number of Required CPU Cycles	[1 - 5] $\times 10^9$
Number of Vehicles	100
Task Delay Constraint	3s
Simulation Time	200s

A. Simulation Environment

The proposed scheme is compared against two other schemes: The next fit (NF) scheme [3], and the random allocation (RA) scheme [15], experiments were conducted to evaluate the performance of the proposed scheme.

B. Results and Discussions

In this section, we introduce and discuss the performance of the results obtained from simulations conducted.

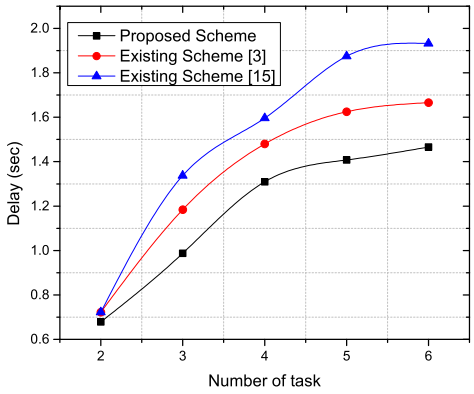

Fig. 2. Offloading delay over different number of tasks

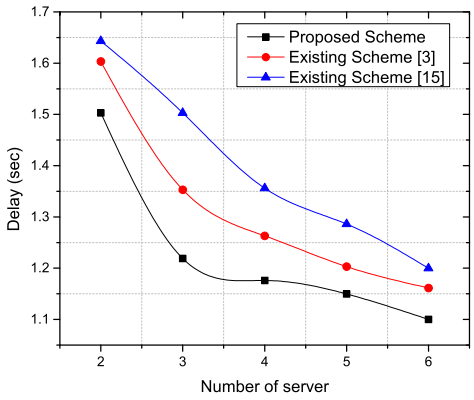

Fig. 3. Offloading delay over different number of servers

Fig. 2 shows that task offloading latency is proportional to the number of tasks in all schemes. As the number of tasks increases, the delay also increases because when the number of tasks is many, the input data size also increases, affecting data transmission and execution latency. The proposed scheme outperforms the two other schemes because a task is allocated to an appropriate server based on maximum utility. In the proposed scheme, PC5 V2V and Uu cellular links provide lower task transmission latency than the IEEE 802.11p used in the other two schemes [16], [17].

Fig. 3 shows that the offloading delay is inversely proportional to the number of edge servers in all schemes. When there are more available edge servers in the network, the lower the offloading latency because the larger number of edge servers engaged, the faster the execution time of all tasks.

Fig. 4 shows that energy consumption increases with an increase in the number of tasks in all schemes because energy consumption depends on the time consumed in the task offloading process, including task transmission and execution time. The higher the latency, the more energy is consumed in the network. The proposed scheme recorded

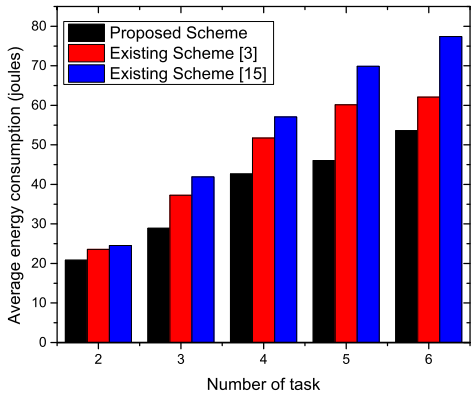

Fig. 4. Energy consumption with respect to different number of tasks

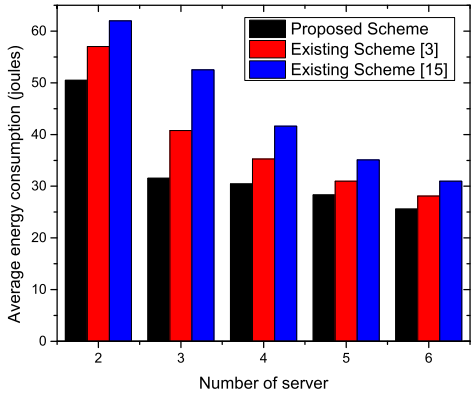

Fig. 5. Energy consumption with respect to different number of servers

better performance than [3] and [15] because, in the proposed scheme, vehicles are grouped into clusters which minimizes communication overhead.

Fig. 5 shows that energy consumption is inversely proportional to the number of servers in all schemes. As the number of servers increases, energy consumption reduces. When there are many edge servers available in the network, the energy consumption reduces because task offloading latency is grossly reduced.

VI. Conclusions

This paper proposes an efficient computation task offloading scheme for C-V2X networks by employing matching theory. The proposed scheme minimizes offloading latency, energy consumption and enhances offloading reliability by grouping vehicles in to clusters, which minimizes communication overhead and provides reliable routes for efficient task offloading using the PC5 V2V and Uu cellular links. The proposed scheme's performance has been compared with existing schemes under extensive simulations. The results confirmed that the proposed scheme has better performance over other schemes.

Acknowledgment

This work was supported by the NSFC Project No.61731017, No.62020106001, and the 111 project No.111-2-14. The work of Gang Liu was supported by NSFC Project No.61971359.

References

[1] C. Wu, X. Chen, T. Yoshinaga, Y. Ji and Y. Zhang, "Integrating Licensed and Unlicensed Spectrum in the Internet of Vehicles with Mobile Edge Computing," *IEEE Network*, vol. 33, no. 4, pp. 48-53, July/August 2019.

[2] F. Abbas, G. Liu, P. Fan, Z. Khan and M. S. Bute, "A Vehicle Density based Two-Stage Resource Management Scheme for 5G-V2X Networks," *in Proc. IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, 2020, pp. 1-5.

[3] Xu, Xiaolong, et al. "An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles" *Future Generation Computer Systems*, vol. 96, pp. 89-100, 2019.

[4] D. Zhang, H. Ge, T. Zhang, Y. Cui, X. Liu and G. Mao, "New Multi-Hop Clustering Algorithm for Vehicular Ad Hoc Networks,"*IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1517-1530, April 2019.

[5] Z. Khan and P. Fan, "A multi-hop moving zone (MMZ) clustering scheme based on cellular-V2X," *China Communications*, vol. 15, no. 7, pp. 55-66, July 2018.

[6] J. Zhao, Q. Li, Y. Gong and K. Zhang, "Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944-7956, Aug. 2019.

[7] Karp, Richard M. "Reducibility among combinatorial problems" *Complexity of computer computations*, pp. 85-103, Springer, Boston, MA, 1972.

[8] Y. Gu, W. Saad, M. Bennis, M. Debbah and Z. Han, "Matching theory for future wireless networks: fundamentals and applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52-59, May 2015.

[9] Z. Zhou, K. Ota, M. Dong and C. Xu, "Energy-Efficient Matching for Resource Allocation in D2D Enabled Cellular Networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5256-5268, June 2017.

[10] Pettersson, W., Delorme, M., García, S., Gondzio, J., Kalcsics, J. and Manlove, D., "Improving solution times for stable matching problems through preprocessing" *Computers & Operations Research*, pp.105128, 2020.

[11] Technical Specification, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol specification (Release 16)," document 3GPP 3GPP TS 25.331 V16.0.0 , Mar. 2020.

[12] Riley, George F., and Thomas R. Henderson. "The ns-3 network simulator" *Modeling and tools for network simulation*, Springer, Berlin, Heidelberg, pp. 15-34, 2010.

[13] J. Wang, W. Wu, Z. Liao, A. K. Sangaiah and R. Simon Sherratt, "An Energy-Efficient Off-Loading Scheme for Low Latency in Collaborative Edge Computing," *IEEE Access*, vol. 7, pp. 149182-149190, 2019.

[14] Y. Dai, K. Zhang, S. Maharjan and Y. Zhang, "Edge Intelligence for Energy-Efficient Computation Offloading and Resource Allocation in 5G Beyond," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12175-12186, Oct. 2020

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

[16] G. Cecchini, A. Bazzi, B. M. Masini and A. Zanella, "Performance comparison between IEEE 802.11p and LTE-V2V in-coverage and out of- coverage for cooperative awareness," *in Proc. 2017 IEEE Vehicular Networking Conference (VNC)*, Torino, 2017, pp. 109-114.

[17] F. Abbas, P. Fan and Z. Khan,"A novel low-latency V2V resource allocation scheme based on cellular V2X communications," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 6, pp.2185–2197, June 2019.