*Computer Systems Science & Engineering*
DOI: 10.32604/csse.2023.032316
***Article***

Tech Science Press

# Efficient-Cost Task Offloading Scheme in Fog-Internet of Vehicle Networks

**Alla Abbas Khadir[1], Seyed Amin Hosseini Seno[1,2,*], Baydaa Fadhil Dhahir[2,3] and Rahmat Budiarto[4]**

[1]Department of Electrical Power Techniques Engineering, Al-Hussain University College, Karbala, Iraq
[2]Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[3]Department of Computer Science, Thi-Qar University, Thi-Qar, Iraq
[4]Department of Computer Science, Mercu Buana University, Jakarta, Indonesia
*Corresponding Author: Seyed Amin Hosseini Seno. Email: hosseini@um.ac.ir
Received: 13 May 2022; Accepted: 24 June 2022

**Abstract:** Fog computing became a traditional OffLad Destination (OLD) to compute the offloaded tasks of the Internet of Vehicles (IoV). Nevertheless, the limited computing resources of the fog node leads to re-offload these tasks to the neighboring fog nodes or the cloud. Thus, the IoV will incur additional offloading costs. In this paper, we propose a new offloading scheme by utilizing RoadSide Parked Vehicles (RSPV) as an alternative OLD for IoV. The idle computing resources of the RSPVs can compute large tasks with low offloading costs compared with fog nodes and the cloud. Finally, a performance evaluation of the proposed scheme has been presented and discussed with other benchmark offloading schemes.

**Keywords:** RoadSide parked vehicles; offloading cost; deadline; budget

## 1 Introduction

Internet of Vehicles (IoV) plays an important role in intelligent transportation system (ITS) for their benefits in providing collision warning, traffic congestion detection, route planning, and infotainment services [1]. Recently, different IoV applications such as autonomous driving, navigation, gaming, and obstacle detection are expected to guide the driver and decrease the traffic accident rate, as well as improve traffic efficiency and traveling convenience. As a result, these applications need high computing resources for excellent processing [2–4]. Computational offloading enables tasks to be shared between IoT devices, fog nodes, and cloud servers [5]. Nowadays, the computation offloading IoV tasks has been developed efficiently like routing [6,7] based on different optimization techniques that have been proposed [8,9]. The centralized cloud is considered a computing paradigm that provides powerful resources, on-demand services, and processing massive information efficiently [10]. So, it can be used for solving the problems of data storage, data processing, and data analysis in the IoV [11]. However, offloading to the cloud has several drawbacks such as the high latency and cost of the data transmission, due to the long-distance issue [12,13], thus a serious degradation in the offloading efficiency [14]. Fog computing has been proposed to bring low latency and reduced bandwidth to the users by moving the resources (i.e., compute, storage, and services) to the edge network [15]. Notwithstanding, the limited computing resource of the fog node leads to re-offloading the tasks to the nearby fog nodes or cloud [16–18] which incurred additional transmission and processing costs [19]. The dense deployment of the RSPVs that have idle computing resources (likes CPUs) and rechargeable batteries [20] in the urban areas can make them a suitable computing resource for processing the tasks. Also, they have a low charging cost [12] compared with fog and cloud as well as they can communicate with any vehicle moving near them. In this paper, we propose a task offloading scheme by using the RSPVs as a computation offload destination. This scheme is better suited to the large urban area because of the following reasons. First, the number of parking vehicles on the roadside is reasonably high, so it is considered a suitable computing infrastructure. Second, these parked vehicles spend a long time in fixed locations, whereas their processing resources are idle. Besides, the close distance of this scheme reduces the cost of transmission and processing, thus minimizing the total offloading cost. The main contributions of this paper can be summarized as follows:

(1) We propose a RSPV offloading scheme in the Fog-IoV network, such that the idle computing resources of RSPV can be utilized effectively for processing the offloaded tasks.
(2) We formulate the assignment tasks from the IoV to the suitable RSPVs through a mathematical model to minimize the total offloading cost of the offloaded tasks under their budget and deadline constraints.
(3) We solve the mathematical model by CPLEX optimization software, in which the offloaded tasks can be assigned to the optimal RSPVs. Simulation results validate the effectiveness of our offloading scheme.

The rest of this paper is organized as follows: Section 2 compares our proposed scheme with some related works. Section 3 describes the system model and problem formulation. The proposed solution and example are presented in Section 4. Simulation results and discussions are presented in Section 5 and Section 6. Finally, Section 7 concludes the work and presents suggestions for the future scope of research. Tab. 1 explains the abbreviations that are used in the paper.

**Table 1:** List of abbreviations & acronyms

| | |
|---|---|
| DSRC | Dedicated short-range communications |
| IoV | Internet of vehicle |
| ITS | Intelligent transportation system |
| MEC | Mobile edge computing |
| MILP | Mixed integer linear programming |
| OLD | Offload destination |
| OPEX | Operational cost |
| RSPV | RoadSide parked vehicle |
| SDN | Software-defined network |

## 2 Related Works

This section describes existing work related to different aspects of task offloading cost. Several paradigms have been used to reduce the task offloading cost in terms of data transmission cost, task computation (processing) cost, or both. Du et al. [2] presented a dual-side optimization offloading decision problem to minimize the cost of both vehicles and corresponding MEC servers simultaneously, where TV white space bands have been used to reinforce the bandwidth for computation offloading. He et al. [21] utilized the SDN for heterogeneous vehicular networks, where different network resources are properly scheduled to minimize communications costs. The cost-efficient resource management problem has been addressed in [22]. They minimized the overall cost which includes the communication cost and VM deployment cost while satisfying the QoS requirement. Zhang et al. [23] proposed a cloud-based MEC offloading framework. They presented transmission offloading schemes by considering the time consumption of the computation task and the mobility of the vehicles. However, the cost-saving effectiveness is weak when the density of the vehicles on the road is small. Also, increasing the speed of vehicles can add more backhaul transmission costs to the total offloading costs. Pham et al. [24] investigated the resource limitation issue of the MEC server. They proposed an offloading policy to select the proper computing nodes and the number of computing resources for the tasks. The proposed policy did not consider the transmission cost of the task. Liu et al. [25] investigated the optimal offloading problem in the mobile fog computing system. Based on the theoretical analysis, a multi-objective optimization problem is formulated, which involves minimizing energy consumption, service delay, and payment cost. However, the transmission cost of the task has been ignored. To improve the limited coverage of the RSU, LiWang et al. [26] proposed a mechanism based on SDN through the offloading of moving vehicles under delay and cost constraints. This mechanism can reduce the monetary cost of the IoV, and guarantee the profits of the moving vehicles. However, the operational cost (OPEX) was high. Li et al. [27] proposed a load-aware MEC offloading method, in which each vehicle makes MEC server selection based on the predicted cost. This cost includes the cost of uploading the task, computing the task on the MEC server, and downloading the computation output, respectively. Wang et al. [28] considered that the end-user should pay a cost to MEC server for occupying the computation and bandwidth resource. They take the charge of both data transmission and task computation as one part of the total cost that is minimized. Tab. 2 shows a comparison between our proposed offloading scheme and the related works according to different parameters and paradigms that have been used.

**Table 2:** Comparison of related works

| Existing work | Transmission cost | Processing cost | Deadline | Budget | Paradigm used | Costly? |
|---|---|---|---|---|---|---|
| [2] | ✓ | ✓ | ✗ | ✗ | TV White Space | Yes |
| [21] | ✓ | ✗ | ✗ | ✗ | Software-defined vehicular network | Yes |
| [22] | ✓ | ✓ | ✓ | ✗ | Cellular Network Base Stations | Yes |
| [23] | ✓ | ✓ | ✓ | ✗ | Vehicular Cloud Computing | No |
| [24] | ✗ | ✓ | ✗ | ✗ | Vehicular Cloud Computing | No |
| [25] | ✗ | ✓ | ✗ | ✗ | Fog-Cloud Computing | Yes |
| [26] | ✗ | ✓ | ✓ | ✓ | Satellite and 5G Network | Yes |
| [27] | ✓ | ✓ | ✓ | ✗ | Vehicular Mobile Edge Computing | Yes |
| [28] | ✓ | ✓ | ✗ | ✗ | One-server MEC system | No |
| **This work** | ✓ | ✓ | ✓ | ✓ | RoadSide Parked Vehicle (RSPV) | No |

Note: ✓ = parameter has been considered, ✗ = parameter has not been considered

## 3 System Model and Problem Formulation

Fig. 1 shows our proposed RSPV offloading scheme. From the Figure, it can be seen that the scheme includes the moving vehicles (i.e., IoVs), RSPVs, and fog nodes. The RSPVs are grouped as a chain line on each road, whereas, the fog nodes are distributed in different locations and communicated with each other through a wireless backhaul. Moreover, all fog nodes have been connected with the cloud through a backbone network. The RSPV that has been interested to rent out its computing resource, sends the status information such as the vehicle number, computation capacity, cost per unit time of using processing resources, and the parking time to the closet fog node. The fog node records them in an RSPV table, which can be updated periodically through the feedback of the RSPVs. Whenever the moving vehicle decides to offload the tasks that cannot be accomplished locally, it sends an offloading query over 3G/LTE interface to the nearest fog node. The fog node responds to this offloading query by sending a response message that contains the RSPV table. According to this table, the moving vehicle can assign the tasks to the suitable RSPV over the low-cost short-range communication DSCR interface thus the cost of data transfer per unit time is different [29]. The RSPV has a low cost in terms of infrastructure and application deployment [30,31], so intuitively the processing cost of the RSPV will be relatively less compared with the traditional computing resources such as fog nodes and cloud [12].

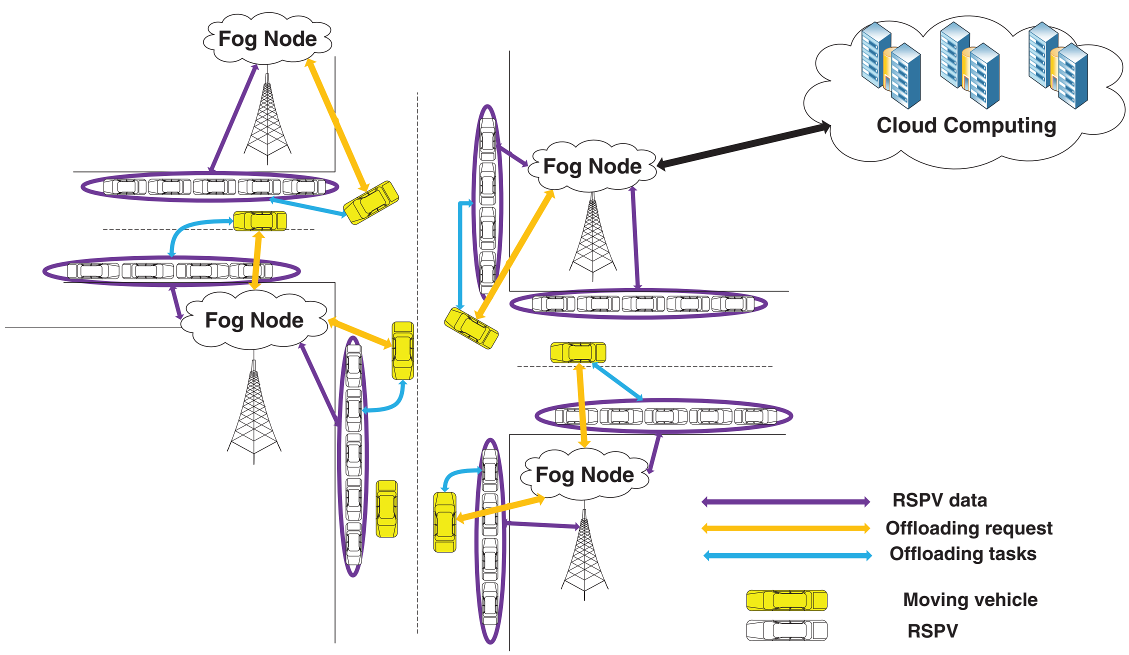


**Figure 1:** Proposed RSPV network architecture

Autonomous driving is the practical application that is assumed to be used with the proposed scheme. Autonomous driving has higher demands on data processing and storage capabilities and thus still requires more available resources [32]. So, let a moving vehicle has $\mathcal{N}$ set of computation tasks to be offloaded and each task is described as $T_i = \{l_i,\ t_i^{max},\ b_i,\ w_i\}$, where $i \in \mathcal{N}$. $l_i$, $t_i^{max}$ and $b_i$ are defined as the data size, the deadline, and the budget of the task respectively; $w_i$ is the number of CPU cycles required to complete the task. We assume there is $\mathcal{M}$ set of RSPVs in which $j \in \{1,\ 2,\ \ldots,\ M\}$, and the fog node can assign the tasks to the suitable RSPVs which can process them under their constraints. Then, the moving vehicle offloads the computation tasks to the RSPVs. Without loss of generality, we assume the task $i$ is atomic which means it should be executed on one RSPV and it cannot divide between more than one RSPV. After processing the tasks, the RSPVs will send back the data result of all tasks to the moving vehicle sequentially. The resulting data size is much smaller than the data size of the input data, thus the delay [32] and the cost [26] of transmission result can be ignored. Tab. 3 illustrates the significant notations used in the model.

**Table 3:** The main notations

| Notation | Description |
|---|---|
| $\mathcal{M}$ | The set of RSPVs |
| $\mathcal{N}$ | The set of tasks |
| $l_i$ | The data size of task $i$ |
| $t_i^{max}$ | The deadline of task $i$ |
| $b_i$ | The budget of task $i$ |
| $w_i$ | CPU cycles required to complete task $i$ |
| $r_j$ | The data transmission rate of RSPV $j$ |
| $\mu_j$ | The computation capacity of RSPV $j$ |
| $D_{ij}$ | The offloading delay of task $i$ at RSPV $j$ |
| $C_{ij}$ | The offloading cost of task $i$ at RSPV $j$ |
| $\Delta t^C$ | The total contact time of moving vehicles with RSPVs. |
| $\sigma_j$ | Cost per unit time for using channel bandwidth of RSPV $j$ |
| $\delta_j$ | Cost per unit time for using a computing resource of RSPV $j$ |

### *3.1 Offloading Delay Model*

In this subsection, we explain the offloading delay model of the offloaded computation tasks at the RSPV, where the offloading delay comprises two parts. The first part is the data transmission delay which represents the time consumed for transferring the data of the task to the selected RSPV. The second part is the task processing delay which represents the time consumed for executing the task on the RSPV. Formally, the offloading delay of the task $i$ at the RSPV $j$ can be expressed as (1).

$$D_{ij} = \frac{l_i}{r_j} + \frac{l_i\, w_i}{\mu_j} \tag{1}$$

where $r_j$ and $\mu_j$ are the data transmission rate and computation capacity of the RSPV $j$, respectively. We assume the data transmission rate $r_j$ remains the same for bidirectional communication. To ensure the success of the offloading process, the task offloading delay should not exceed the deadline of the task and total contact time of the moving vehicle with the RSPV, that is, $D_{ij} \leq t_i^{max} \leq \Delta t^C$, where the contact time $\Delta t^C$ can be calculated as [33]. Where $\tau$, $v$ are the communication range and speed of the moving vehicle respectively; $\lambda$ and $M$ are the intensity and the number of the RSPVs on the roadside. From Eq. (2), we can see the contact time is affected by the speed of the moving vehicle and the intensity of the RSPVs. For the sake of simplicity, we assume both of them (i.e., $\tau$ and $v$) are set to the suitable value, such that the contact time will be high.

$$\Delta t^C = 2\tau \left( \frac{1}{v} + \frac{M-1}{\lambda} \right) \tag{2}$$

### 3.2 Offloading Cost Model

Generally, the offloading cost consists of the data transmission cost and task processing cost [34]. To obtain the data transmission cost, we define $\sigma_j$ as the cost per unit time for using network channel bandwidth of the RSPV $j$. Hence, the data transmission cost to transfer the data of task $i$ to the RSPV $j$ can be given by (3).

$$C_{ij}^{trans} = \frac{l_i}{r_j}\ \sigma_j \tag{3}$$

Similarly, to calculate the task processing cost, we define $\delta_j$ as the cost per unit time for using the computing resource of the RSPV $j$. Thus, the processing cost for the task $i$ can be obtained by (4).

$$C_{ij}^{proc} = \frac{l_i\ w_i}{\mu_j}\ \delta_j \tag{4}$$

Finally, the offloading cost for the task $i$ at RSPV $j$ can be expressed as (5).

$$C_{ij} = C_{ij}^{trans} + C_{ij}^{proc} \tag{5}$$

### 3.3 Problem Formulation

Now, we formulate the assignment task offloading as an optimization problem. The goal is to minimize the total offloading cost under budget and deadline constraints. Therefore, we define an assignment optimization variable $x_{ij}$. Here, $x_{ij} = 1$ means that the fog node assigns the task $i$ to the RSPV $j$, and otherwise, $x_{ij} = 0$. The optimization problem can be written as (6).

$$\mathbf{P1} = \ \min \sum\nolimits_{i \in \mathcal{N},\, j \in \mathcal{M}} x_{ij}\ C_{ij}, \tag{6}$$

Subject to:

$$\sum_{j=1}^{M} x_{ij} = 1,\ \forall\, i\ \in \mathcal{N}, \tag{C1}$$

$$\sum_{i=1}^{N} x_{ij} = 1,\ \forall\, j\ \in \mathcal{M}, \tag{C2}$$

$$C_{ij}\ \le\ b_i, \forall\, i\ \in \mathcal{N},\ j\ \in \mathcal{M}, \tag{C3}$$

$$D_{ij} \le\ t_i^{max}\ \le\ \Delta t^C,\ \forall\, i\ \in \mathcal{N},\ j\ \in \mathcal{M}, \tag{C4}$$

$$x_{ij}\ \in \{0,\ 1\}\ ,\ \forall\, i\ \in \mathcal{N},\ j\ \ \in \mathcal{M}. \tag{C5}$$

Here, constraint C1 ensures that each task must be executed on only one RSPV. Constraint C2 states that each RSPV can receive at most one task at a given time. Constraint C3 enforces that the offloading cost at the RSPV should be less than or equal to the budget of the task, thus the total offloading cost will be less than the total budget of the tasks. According to constraint C4, the offloading delay at RSPV should not exceed the deadline of the task and the total contact time. The last constraint C5 denotes $x_{ij}$ is a binary variable.

## 4 Proposed Solution and Example

Minimization of the total offloading cost (P2) is considered a MILP problem which is solved by using CPLEX mathematical optimization software. The following example is performed to explain the effectiveness of the proposed model. Let the number of tasks ($N$), the number of the RSPVs $(M)$, and the total contact time $\Delta t^C$ are 3, 6 and 60 s respectively. The data of the related tasks and RSPVs are organized in Tabs. 4 and 5, respectively.

**Table 4:** The related data of tasks

| Parameter | Task 1 | 2 | 3 |
|---|---|---|---|
| $l_i$ (Bytes) | 5 | 10 | 15 |
| $w_i$ (cycle/Byte) | 2 | 4 | 5 |
| $t_i^{max}$ (s) | 2 | 3 | 4 |
| $b_i$ ($) | 0.19 | 0.25 | 0.37 |

**Table 5:** The related data of RSPVs

| Parameter | RSPV 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $\mu_j$ (cycle/s) | 20 | 30 | 40 | 50 | 75 | 80 |
| $\delta_j$ ($/s) | 0.0033 | 0.0082 | 0.0061 | 0.0017 | 0.0045 | 0.0041 |
| $r_j$ (Mb/s) | 10 | 15 | 5 | 10 | 10 | 12 |
| $\sigma_{j\}}$($/s) | 0.001 | 0.0032 | 0.001 | 0.0026 | 0.014 | 0.0018 |

The optimal solution of the **(P1)** in this example is 0.01139 $. The suitable assignment of the tasks in terms of the offloading cost associated with the offloading delay is shown in Tab. 6.

**Table 6:** The optimal assignment of the tasks

| Task $i$ to RSPV $j$ | $C_{ij}$ | $D_{ij}$ |
|---|---|---|
| 1 to 6 | 0.00645 | 0.48 |
| 2 to 5 | 0.00365 | 1.125 |
| 3 to 4 | 0.00129 | 1.8 |

## 5 Simulation Results

In this section, the proposed RSPV offloading scheme is evaluated to investigate its performance and effectiveness in comparison with three benchmark offloading schemes which are:

- **Fog-Cloud offloading scheme** [35]**:** The fog node will re-offload the tasks to the cloud for execution with a high cost of transmission and processing. In the simulation scenario, we assume the cost unit of data transmission and task processing for the cloud in the range (0.01–0.09) $/s. Also, the computation capacity and the data transmission rate are assumed 100 cycle/s and 50 Mb/s respectively.
- **Fog-Fog offloading scheme** [36]**:** The fog node will re-offload the tasks to the neighboring fog nodes with additional transmission costs. In the simulation scenario, we assume the cost unit of data transmission and task processing of the fog in the range (0.001–0.009) $/s and the cost unit of

data transmission between the fog nodes has a fixed value of 0.002 $/s. Also, the computation capacity and the data transmission rate are assumed 50 cycles/s and 25 Mb/s respectively.

- **Cost-aware offloading scheme** [14]**:** The vehicle makes MEC server selection based on the predicted cost and load distribution of MEC servers. In this scheme, the transmission cost represents the cost of the V2I transmission or multi-hop V2V relay transmission. Similarly, the cost of result feedback takes a multi-hop wireless backhaul relay between several MECs. In the simulation scenario, we assume the cost unit of data transmission and task processing of the MEC in the range (0.001–0.009) $/s and the cost unit of data transmission through the multi-hop wireless backhaul and V2V relay are 0.002 $/s and 0.001$/s respectively. Also, the computation capacity and the data transmission rate are assumed 50 cycles/s and 30 Mb/s respectively. We consider a simulation scenario with the following setting and the detail is shown in Tab. 7**.**

**Table 7:** Simulation parameters

| Parameter | Range of values |
|---|---|
| Number of tasks $(N)$ | 2–10 |
| Number of RSPVs $(M)$ | 5–15 |
| The data size of the task $(l_i)$ | 5–15 Bytes |
| CPU cycle of task $(w_i)$ | 2–5 cycle/Byte |
| Deadline of task $(t_i^{max})$ | 1–5 s |
| Budget of task $(b_i)$ | 0.19–0.37 $ |
| The data transmission rate of RSPV $(r_j)$ | 5–15 Mb/s |
| Computation capacity of RSPV $(\mu_j)$ | 20–80 cycle/s |
| Cost per unit time of channel bandwidth $(\sigma_j)$ | 0.0011–0.0033 $/s |
| Cost per unit time of computing resource $(\delta_j)$ | 0.0017–0.0082 $/s |

## 6 Results and Discussion

Fig. 2 compares our proposed scheme with other offloading schemes through the total offloading cost for a different number of tasks. From the Figure, it can be observed that when the number of tasks is increased, the total offloading cost of other schemes is significantly increased. In the Fog-Cloud scheme, the total offloading cost is very high because of the additional re-transmission cost of the task. This will be high when more than one task has been sent to the cloud. The same reason for the Fog-Fog offloading scheme, where re-transmission cost between the fog nodes will be less compared with the Fog-Cloud offloading scheme. For the Cost-aware offloading scheme, initially, the total offloading cost is low because the number of tasks is low and the V2I offloading mode has been used. In other words, there is no additional cost because the vehicle is in the coverage area of the MEC. With an increasing number of tasks, the total offloading cost is increased. That means the vehicle utilized the V2V offloading mode for transmitting the task to the target MEC and getting the result. This V2V cost is increased with increasing the hops of V2V transmissions. Meanwhile, our proposed scheme can always achieve the best total offloading cost. The reason is the low cost of data transmission time between the moving vehicle and RSPVs. This means the total offloading cost in this scheme represents the processing cost only.

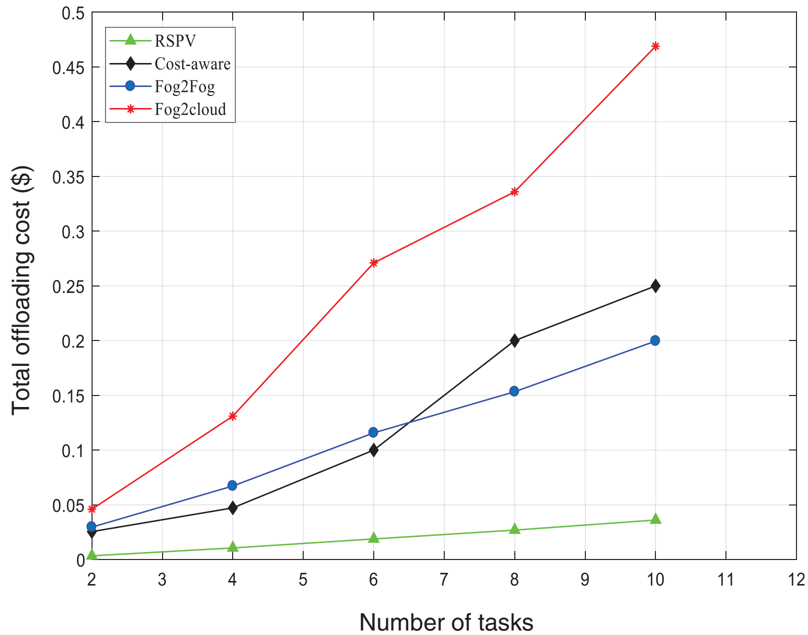


**Figure 2:** The total offloading cost *vs.* the number of tasks

Fig. 3 explains the relation between the total offloading cost and the deadline of the task. The deadlines that are used here are 1, 2, 3, 4 and 5 s. From the Figure, it can be seen that the proposed offloading scheme provides a total offloading cost lower than the other schemes when the deadline of the task increases. In other words, the proposed scheme can provide the ability for the fog node to consider the compatibility between the deadline of the task and the suitable offloading cost. Moreover, our proposed scheme becomes cost-effective when the task has a long time (2 s and more). The total offloading cost in the Cost-ware offloading scheme is reduced when the deadline is high (3 s and more). Also, it can be seen that the difference between the costs of the two schemes is small in some deadlines. The reason for that, when the deadline is high this scheme utilizes the V2I offloading mode that has a low transmission cost. Whereas, it utilizes the V2V offloading mode that has a high transmission cost when the deadline is low. In other words, when the deadline is high the vehicle waits to reach the cheaper MEC to use V2I mode instead of V2V mode, because the V2V mode meets the required deadline but with a high transmission cost. As a result, the Cost-aware offloading scheme considers this compatible. Meanwhile, Fog-Fog offloading scheme also considers this compatibility in case there is no more offloading for the neighboring fog nodes which can be seen at 3 s and more. Finally, Fog-Cloud offloading scheme has no consideration for compatibility. So, the total offloading cost has high and low values which are based on re-offloading some or all the tasks to the cloud. The high number of offloaded tasks from moving vehicles to a few RSPVs is the first limitation of this work. That will increase the offloading cost of the offloaded tasks according to the competition between the moving vehicles to get a response from the RSPVs. The second limitation is the big data size of the offloaded tasks that will increase the response time as well as the offloading cost.

Fig. 4 illustrates the revenue of the three schemes. Here the revenue represents the difference cost between the total budget of the tasks and the total offloading cost. From the Figure, we observe that our offloading scheme has the best revenue in all budgets because of the following reasons: First, the processing cost of the RSPVs will be relatively less because of the low cost of infrastructure and application deployment. Whereas the fog and cloud have a higher processing cost because of their high deployment and maintenance costs. Second, there is no backhaul transmission cost between the RSPVs as in the Fog-Fog and Fog-Cloud schemes which are highly expensive [27]. The Cost-aware offloading scheme did not consider the budget in the characteristic of the task. Therefore, this scheme has not been compared with our proposed scheme in terms of revenue.

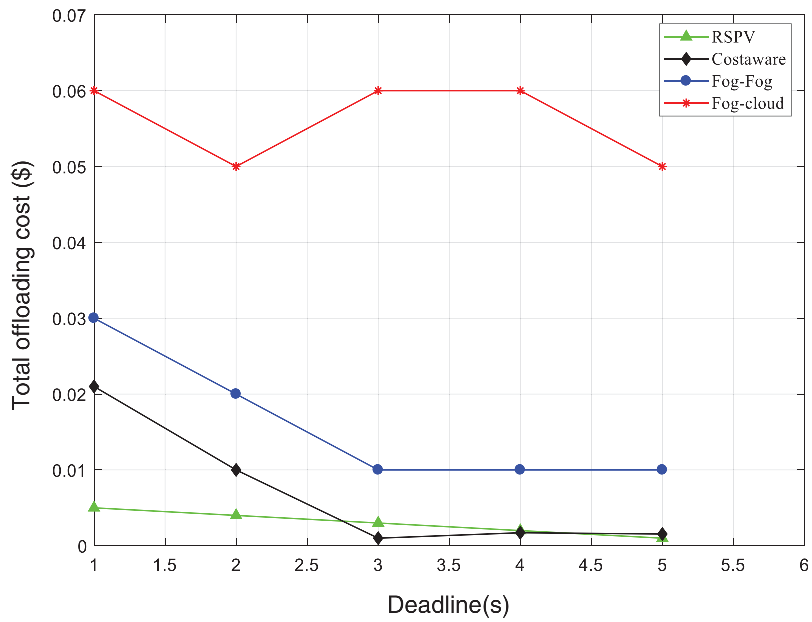


**Figure 3:** The total offloading cost *vs.* the deadline

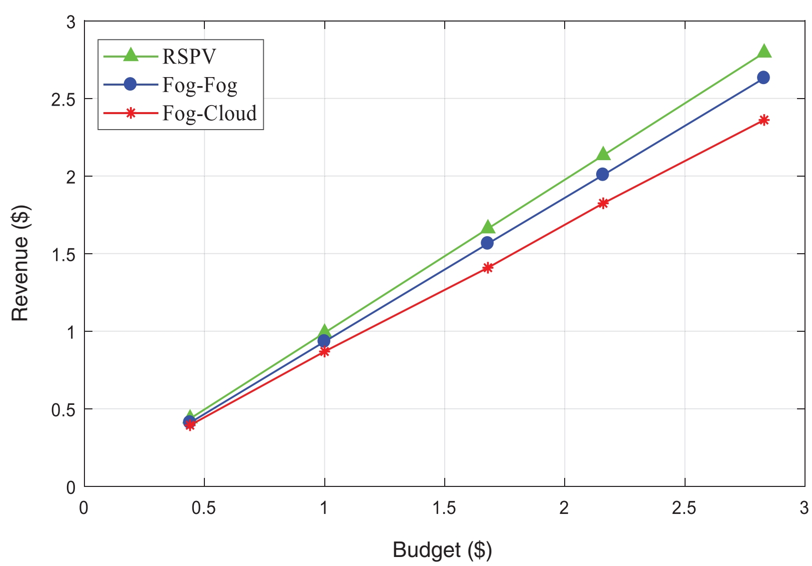


**Figure 4:** The revenue *vs.* the budget of tasks

## 7 Conclusion and Future Work

In this paper, we proposed the RSPV offloading scheme. In this scheme, the street parked vehicles have been utilized as a computing OLD instead of fog and cloud. In which the RSPVs can share their idle computing resources for executing the offloaded tasks of the moving vehicle (IoV). The simulation results have verified that our proposed offloading scheme can significantly reduce the total offloading cost. Also, it can grant the chance to moving vehicles for selecting the best computing resource that meets the predefined budget and deadline delay of the task. As a result, this offloading scheme is more suitable than the existing conventional offloading schemes. For future work, we plan to extend this work by increasing the number of offloaded tasks and the RSPVs. These tasks can be scheduled and assigned in a fast way to the suitable RSPVs through a centralized offloading process by a controller. Also, we can consider the parking time of the RSPVs to evaluate the performance of the proposed scheme under different conditions.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] X. He, Z. Ren, C. Shi, and J. Fang, "A novel load balancing strategy of software-defined cloud/fog networking in the internet of vehicles," *China Communications*, vol. 13, no. Supplement 2, pp. 140–149, 2016.

[2] J. Du, F. R. Yu, X. Chu, J. Feng and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1079–1092, 2019.

[3] S. Choo, J. Kim and S. Pack, "Optimal task offloading and resource allocation in software-defined vehicular edge computing," in *Int. Conf. on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), 2018.

[4] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.

[5] O. R. Faten Alenizi, "Dynamically controlling offloading thresholds in fog systems," *Sensors MDPI*, vol. 21, no. 7, pp. 2512, 2021.

[6] A. T. Albu-Salih, S. A. H. Seno and S. J. Mohammed, "Dynamic routing method over hybrid SDN for flying ad hoc networks," *Baghdad Science Journal*, vol. 15, no. 3, pp. 361–368, 2018.

[7] S. A. H. Seno and N. Noorani, "Routing in vanets based on intersection using SDN and fog computing," in *Int. Conf. on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, 2018.

[8] L. A. Phan, D. T. Nguyen, M. Lee, D. H. Park and T. Kim, "Dynamic fog-to-fog offloading in SDN-based fog computing systems," *Future Generation Computer Systems*, vol. 117, pp. 486–497, 2021.

[9] Q. Luo, C. Li, T. H. Luan, W. Shi and W. Wu, "Self-learning based computation offloading for internet of vehicles: Model and algorithm," *IEEE Transactions on Wireless Communications*, vol. 20, no. 9, pp. 5913–5925, 2021.

[10] C. S. RajkumarBuyya, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[11] S. PengfeiHu, "Survey on fog computing: Architecture, key technologies, applications and open issues," *Journal of Network and Computer Applications*, vol. 98, pp. 27–42, 2017.

[12] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin *et al.,* "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[13] R. A. Dziyauddin, D. Niyato, N. C. Luong, M. A. M. Izhar, M. Hadhari *et al.,* "Computation offloading and content caching delivery in vehicular edge computing: A survey," arXiv preprint arXiv:1912.07803, 2019.

[14] K. Zhang, Y. Mao, S. Leng, S. Maharjan and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *IEEE Int. Conf. on Communications (ICC)*, Paris, France, 2017.

[15] S. Liao, J. Li, J. Wu, W. Yang and Z. Guan, "Fog-enabled vehicle as a service for computing geographical migration in smart cities," *IEEE Access*, vol. 7, pp. 8726–8736, 2019.

[16] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa *et al.,* "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," *IEEE Vehicular Technology Magazine*, vol. 12, no. 3, pp. 55–64, 2017.

[17] M. Aazam, S. Zeadally and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Generation Computer Systems*, vol. 87, pp. 278–289, 2018.

[18] M. Mukherjee, S. Kumar, C. X. Mavromoustakis and G. Mastorakis, "Latency-driven parallel task data offloading in fog for industrial applications computing networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6050–6058, 2020.

[19] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang *et al.,* "QoS-Aware dynamic fog service provisioning," arXiv preprint, arXiv:1802.00800,available online: https://arxiv.org/abs/1802.00800 (2018).

[20] A. B. Reis, S. Sargento and O. K. Tonguz, "Parked cars are excellent roadside units.," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2490–2502, 2017.

[21] Z. He, D. Zhang and J. Liang, "Cost-efficient sensory data transmission in heterogeneous software-defined vehicular networks," *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7342–7354, 2016.

[22] L. Gu, D. Zeng, S. Guo, A. Barnawi and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, 2017.

[23] K. Zhang, Y. Mao, S. Leng, Y. He and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36–44, 2017.

[24] X. Q. Pham, T. D. Nguyen, V. Nguyen and E. N. Huh, "Joint node selection and resource allocation for task offloading in scalable vehicle-assisted multi-access edge computing," *Symmetry*, vol. 11, no. 1, pp. 1–17, 2019.

[25] L. Liu, Z. Chang, X. Guo, S. Mao and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2018.

[26] M. LiWang, S. Dai, Z. Gao, X. Du, M. Guizani *et al.,* "A computation offloading incentive mechanism with delay and cost constraints under 5G satellite-ground IoV architecture," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 124–132, 2019.

[27] L. Li, H. Zhou, S. X. Xiong, J. Yang and Y. Mao, "Compound model of task arrivals and load-aware offloading for vehicular mobile edge computing networks," *IEEE Access*, vol. 7, pp. 26631–26640, 2019.

[28] K. Wang, Z. Hu, Q. Ai, Y. Zhong, J. Yu *et al.,* "Joint offloading and charge cost minimization in mobile edge computing," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 205–216, 2020.

[29] L. Aminizadeh and S. Yousef, "Cost minimization scheduling for deadline constrained applications on vehicular cloud infrastructure," in *Int. Conf. on Computer and Knowledge Engineering (ICCKE)*, Mashhad, Iran, 2014.

[30] A. A. Khadir and S. A. H. Seno, "SDN-Based offloading policy to reduce the delay in fog-vehicular networks," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 1261–1275, 2021.

[31] V. G. Menon, "Moving from vehicular cloud computing to vehicular fog computing: Issues and challenges," *International Journal of Computer Science and Engineering*, vol. 9, no. 2, pp. 14–18, 2017.

[32] Y. Cui, Y. Liang and R. Wang, "Resource allocation algorithm with multi-platform intelligent offloading in d2d-enabled vehicular networks," *IEEE Access*, vol. 7, pp. 21246–21253, 2019.

[33] N. Liu, M. Liu, G. Chen and J. Cao, "The sharing at roadside: Vehicular content distribution using parked vehicles," in *proceedings IEEE INFOCOM*, Orlando, FL, USA, 2012.

[34] X. Guo, L. Liu, Z. Chang and T. Ristaniemi, "Data offloading and task allocation for cloudlet-assisted ad hoc mobile clouds," *Wireless Networks*, vol. 24, pp. 79–88, 2018.

[35] R. Deng, R. Lu, C. Lai, T. H. Luan and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[36] A. Yousefpour, G. Ishigaki, R. Gour and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, 2018.