Relay-Assisted Task Offloading Optimization for MEC-Enabled Internet of Vehicles

Heli Zhang[1](✉), Haonan Zhang[1], Xun Shao[2], and Yusheng Ji[3]

[1] Beijing University of Posts and Telecommunications, Beijing 100876, China
zhangheli@bupt.edu.cn
[2] Kitami Institute of Technology, Kitami 0908507, Hokkaido, Japan
[3] National Institute of Informatics (NII), Chiyoda City, Tokyo, Japan

Abstract. Mobile edge computing (MEC)-enabled Internet of Vehicles (IoV) is a promising way to provide low latency and high computation functions to smart vehicles. Owing to the mobility of vehicles and unpredicted distribution of computation-intensive tasks, computational resources at the edge may be utilized with only low efficiency. To solve this problem, this study investigates a relay-supported task offloading scheme in MEC-enabled IoV. In this scheme, computational tasks produced by vehicles are predictively offloaded to MEC nodes through relays to improve the allocation of computational resources. A combinational problem is used to model relay selection for vehicles connected to the MEC. To solve the corresponding problem, a low-complexity algorithm that combines the Hungarian and the Greedy algorithms is designed. Simulation results show that the proposed scheme achieves better performance than existing schemes in terms of overall efficiency and offloading time.

Keywords: Vehicular network · MEC · Predictive offloading · Multihop relay

1 Introduction

With the development of the Internet of Vehicles (IoV), the number of vehicles connected to the network has been increasing exponentially. Considering the complexity of the traffic environment, wireless channel status, and user demand, dealing with the growth in vehicular services with high accuracy and making automatic decisions with low latency are major challenges [1].

Because IoV tasks are computation-intensive, mobile edge computing (MEC) technology has emerged as a promising tool to provide high computational capacity nearer to vehicle devices [2–4]. MEC can reduce the burden of the wireless backhaul and core network [5]. In MEC-enabled vehicular networks, tasks are offloaded to MEC nodes via the Vehicle-to-Infrastructure (V2I) mode. Owing to the instability of the wireless link and resource starvation, relay-assisted offloading schemes are widely used. Relay-assisted schemes are designed to optimize the offloading performance by transmitting a task to a better location. Recent research has identified and studied some key features in the realization of vehicular network architectures as well as solutions for some key

C. T. Calafate et al. (Eds.): MONAMI 2021, LNICST 418, pp. 152–164, 2022.
https://doi.org/10.1007/978-3-030-94763-7_12

challenges. For instance, a mobility-aware strategy was developed to realize resource sharing among servers in MEC-enabled vehicular networks [6]. Another study used the effective connectivity of vehicles, mobile devices, and infrastructure to overcome the adverse effects of mobility on reliable data transmission and introduced this concept into IoV applications using millimeter wave (mmWave) technology [7]. An autonomous hybrid edge/cloud framework for vehicular edge computing was proposed to significantly increase the computational capacity by utilizing the available computational resources in surrounding vehicles, roadside units (RSUs), and the cloud via multiaccess networks [8]. However, this model did not consider the relay mode. Another study considered a three-node relaying MEC system and an adapted transport protocol for realizing better energy performance [9]. From the abovementioned studies, it is obvious that selecting favorable servers in an offloading scheme remains a challenging task. As a vehicle moves, multiple RSUs are connected to different MEC servers. Existing studies rarely considered MEC server overload or downtime in a certain region. Because upgrading onboard units and realizing wider coverage of MEC servers requires a longer period of time, MEC-enabled vehicular networks may be unable to respond to a surge in demand for data processing; in other words, some vehicle tasks cannot be offloaded in time because of the low network scalability. Furthermore, low expansibility results in some vehicles suffering high delays. Thus, the present authors proposed a relay-assisted task offloading scheme to optimize the total latency and improve the scalability of the network. The present study describes the whole mission arrival process and defines it as a composite process in which task forecasting is relayed to a specified location and offloaded for computation. When the accessed server is busy, the vehicle terminal cannot receive calculation results in time. Therefore, multihop relay vehicles on its way are selected to offload the task data to a resource-rich server. Further, a joint selection algorithm is proposed to identify appropriate relay nodes for a part of the vehicles used in the proposed scheme. This study aims to minimize the total offloading latency of the terminals in the current area through reasonable matching between the source vehicles and each-hop relay vehicles.

The remainder of this paper is organized as follows. Section 2 describes the system model and problem formulation. Section 3 describes the developed optimal offloading scheme and joint selection algorithm. Section 4 presents the simulation results and discussions. Finally, Section 5 presents the conclusions of this study.

2 Introduction

This section describes the proposed relay-assisted MEC-enabled vehicular network system and discusses both the transmission model and the computation model.

2.1 Network Model

Figure 1 shows the vehicular network equipped with a continuous MEC server considered in this study. Typically, the RSUs installed along the unidirectional road collocate with the MEC server so that vehicle terminals can access the computing resources of the MEC server by communicating with the RSU. The coverage radius of each RSU is L.

For clarity, Fig. 1 shows only one routing path; in reality, multiple parallel paths will exist simultaneously. Vehicles with tasks are called source vehicles (SVs) and those acting as relay nodes traveling in the same direction are called relay vehicles (RVs).

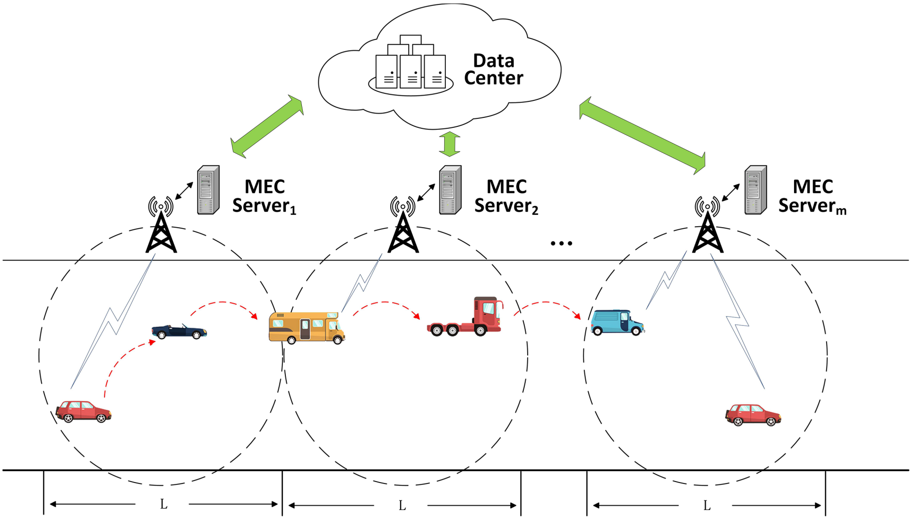

Fig. 1. System model

Analogous to cellular network users in communication systems, a connected vehicle determines whether the MEC server is idle by communicating with the current RSU [10]. One-to-one correspondence is used to satisfy the requirements of reducing signal interference and excessive resource overheads; in other words, each RV can serve only one SV through the Vehicle-to-Vehicle (V2V) mode, and each SV can only be served by one RV.

2.2 Transmission Model

The transmission model can also be interpreted as a communication model consisting of a direct connection and a multihop relay connection. This study considers environments where scatterers are sparse, such as suburban or expressway scenarios. As in similar studies in this area, signal fading caused by non-line-of-sight is ignored, and the influence of the relative distance between the vehicles on the signal strength of data transmission is considered. Pass loss is positively correlated with the distance between different devices and is given by

$$L(d(t)) = L(d_0) + 10\alpha \log_{10}\left(\frac{d(t)}{d_0}\right) \tag{1}$$

where $d(t)$ is the distance between the terminals; $L(d_0)$ is a path loss constant due to factors such as frequency, weather, and geology; and α is the path loss index from which the path loss L can be obtained [11].

1) Direct Connection

A task is offloaded from a vehicle terminal or a relay node to the RSU. The path-loss channel model is used for the link gain. For a given transmission power P_s and received power P_r,

$$L(d(t)) = 10\log_{10}\left(\frac{P_s}{P_r}\right) \tag{2}$$

This equation can be rewritten in terms of P_r as follows:

$$P_r = \frac{P_s}{10^{\frac{L(d(t))}{10}}} \tag{3}$$

N_0 is the Gaussian white noise power. The instantaneously available information transmission rate can be expressed as

$$R(t) = \log_2\left(1 + \frac{P_s 10^{-\frac{L(d(t))}{10}}}{N_0}\right) \tag{4}$$

The vehicle communicates with the RSU through the V2I mode based on the Long-Term Evolution Advanced specification. Let B_{V2I} denote the channel bandwidth and $d_{i,R}$, the distance between the vehicle carrying task i and the RSU. Then, the data rate is given by

$$R_{V2I}(t) = B_{V2I}\log_2\left(1 + \frac{P_t 10^{-\frac{L\left(d_{i,R}(t)\right)}{10}}}{N_0}\right) \tag{5}$$

2) Multihop Relay Connection

The V2V model can support multihop relay connections. Moving vehicles can share communication, computational, and storage resources with other nearby vehicles by using dedicated short-range communication (DSRC) technology following the IEEE 802.11p specification 8. The task requester in SVs are considered able to exchange position, velocity, and driving status information with RVs. Moreover, according to the DSRC link feature, the vehicle can update the status table in real time, including the variety of information obtained from nearby vehicles. Obviously, the vehicle may estimate the distance on the basis of its table as follows:

$$d_{i,j}(t) = \sqrt{\left(X_{i,j}(t)\right)^2 + \left(Y_{i,j}(t)\right)^2} \tag{6}$$

$$X_{i,j}(t) = X_{i,j}(0) + \left(v_j - v_i\right)t \tag{7}$$

$X_{i,j}$ is the parallel distance to the road between vehicles i and j at time t, and $Y_{i,j}$ is the lateral distance perpendicular to the road between the two vehicles and is assumed to be fixed. Given that the maximum communication distance according to the DSRC standard and actual link conditions between any two vehicles is D,

$$0 \le d_{i,j}(t) \le D \tag{8}$$

Then, within the communication distance, the end-to-end reachable information rate of $i \ni SV$ and $j \ni RV$ is

$$R_{i,j}(t) = B_{V2V} \log_2\left(1 + \frac{P_t 10^{-\frac{L(d_{i,j}(t))}{10}}}{N_0}\right) \tag{9}$$

2.3 Computation Model

For task i, c_i is the number of cycles required for completing task i, and d_i is the data size for transmitting it from the transmitter to the receiver. The tolerance delay for each task is $t_{i,local} > t_{i,\max}$. Briefly, the initial information of task i can be clearly denoted as $T_i = \{c_i, d_i, t_{i,\max}\}$. In addition, let f_i and f_m respectively be the CPU operating frequency at the vehicle terminal and at a MEC server (unit: number of CPU cycles per second). When the computing power of the SV is insufficient to complete the task calculation within the delay tolerance, that is,

$$t_{i,local} > t_{i,\max} \tag{10}$$

the MEC server is requested to assist in the calculation. $t_{i,local}$ is the time required to complete the task locally. When the vehicle leaves the RSU radio coverage area or the MEC is not expected to provide part of the computational resources for SVs, the next or even a later MEC server can be accessed through multihop V2V relay transmission. An analysis of this model shows that the total task processing delay involves three parts: transmission delay through the DSRC link, latency to offload to the MEC, and task execution time. In most task types, the amount of data obtained through the analysis is often much smaller than the input raw data; therefore, the delay in returning the calculation result is ignored in previous research.

3 Optimal Offloading Scheme and Joint Selection Algorithm

Based on the MEC-enabled IoV model discussed in Sect. 2, an optimal offloading scheme is designed to achieve the desired performance. Subsequently, a joint selection algorithm is used to solve the relay node selection problem of multiple vehicles.

3.1 Optimal Offloading Scheme

The offloading problem is considered for all vehicles in the area in light of the area covered by one of the RSUs as an example. First, under the influence of the computational task complexity ρ, the delay required by the vehicle's onboard unit to process the current task is calculated as

$$t_{i,local} = \frac{\rho c_i}{f_i} \tag{11}$$

Then, whether the computational capacity of the vehicle's onboard unit satisfies this task's delay requirement is judged using (10)–(11). If the task delay tolerance is met, the local mode is used as the first choice.

In the opposite case, the proposed multi-relay-assisted offloading scheme is applied for the uplink network shown in Fig. 1. Within the delay limit, the N_i - th RSU on which task i can be offloaded to the farthest continuous MEC is calculated as

$$\left\lfloor \frac{v_i t_{i,\max}}{L} \right\rfloor = N_i \tag{12}$$

where $\lfloor \cdot \rfloor$ means rounding down, and v_i is the velocity of the requester carrying task i. Because the vehicle transmits task data for a short time, the velocity change problem is not considered during the task offloading process.

The unique identifier of the source vehicle, N_i, and other information should be attached to task i. After the vehicle determines that the onboard unit fails to satisfy the needs, a resource is requested from the MEC, and the multihop relay connection process is used when the current server has no idle resources. According to the joint selection algorithm, a suitable RV is selected as the first-hop relay node. Let RV1 be the collection of first-hop relay terminals to which j belongs. Define a pairing $\theta = \{\theta_{i,j}\}_{\mathrm{SV}\times\mathrm{RV}}$ consisting of binary elements $\theta_{i,j} \in \{0, 1\}$ for $i \in$ SV and $j \in$ RV. Specifically, if SV i is paired with RV j, $\theta_{i,j} = 1$; otherwise, $\theta_{\mathrm{i,j}} = 0$. Each RV can only be paired with one SV and vice versa; therefore,

$$\begin{cases} \sum_{i=1}^{n_{SV}} \theta_{i,j} = 1, \ \ \forall i = 1, \ldots, \boldsymbol{SV} \\ \sum_{j=1}^{n_{RV}} \theta_{i,j} = 1, \ \ \forall j = 1, \ldots, RV \\ \theta_{i,j} = 0, \ \ otherwise \end{cases} \tag{13}$$

The time required for transmitting the task from i to j is $t_{i,j}$:

$$t_{i,j} = \frac{d_i}{R_{i,j}} \tag{14}$$

When task i reaches the relay node of the first-hop terminal, that terminal is reassigned to V_i and whether the connected MEC can support offloading is checked. As noted in [14] and [15], if it can support offloading, the offloading delay $t_{i,up}$ and the computing delay $t_{i,ex}$ are given by

$$\begin{cases} t_{i,up} = \dfrac{d_i}{R_{V2I}} \\ t_{i,ex} = \dfrac{c_i}{f_m} \end{cases} \tag{15}$$

$$t_i = \theta_{i,j} t_{i,j} + t_{i,up} + t_{i,ex} \tag{16}$$

If the MEC service is still unavailable, the second-hop relay process is continued and the abovementioned judgment conditions are rechecked and calculation methods are repeated. However, the limitation of N_i must be considered. Owing to the problem of communication distance, for more than N_i relay hops, the task can reach the coverage of the N_i-th MEC server. When an idle MEC server is still not found after N_i relay hops, the task has to be queued for calculation and offloading. Therefore, the maximum number of hops is set to N_i to ensure that the task is offloaded before the source vehicle arrives.

This scheme aims to minimize the total latency when the requester completes the application in an area where MEC computational resources cannot be obtained. The problem can be mathematically formulated as (17), where $C1$ is a prerequisite for relay offloading. hop_k indicates the k - th hop relay, and $k \leq N_i$ limits task i to pass the $t_{i,j,k} = +\infty$ jump at most. $t_{keep}^{i\to j}$ records the connectable time between the source vehicle and the relay vehicle. $C2$ and $C3$ ensure that tasks can be transmitted to the relay vehicle. $C4$ limits the one-to-one correspondence of V2V. $C5$ guarantees that the task is completed within the time delay requirement. By solving problem P1, the total latency is optimized.

$$\begin{aligned} &\mathrm{P1}: \min_{\theta_{i,j}} \sum_{i=1}^{n_{sv}} \sum_{k=1}^{N_i} t_i^{hop_k} \\ &\text{s.t.} \begin{cases} C1: t_{i,local} > t_{i,max}, \forall i \in \mathrm{SV} \\ C2: t_{keep}^{i\to j,hop_k} = \underset{t}{argmin}\left\{ d_{i,j}^{hop_k}(t) \leq D \right\}, \\ \forall i \in \mathrm{SV}, \forall j \in \mathrm{RV}, \forall k \in N_i \\ C3: t_{keep}^{i\to j,hop_k} > t_{i,j}^{hop_k} \\ C4: \sum_{i=1}^{n_{SV}} \sum_{j=1}^{n_{RV}} \theta_{i,j} = 1 \\ C5: \sum_{i=1}^{n_{SV}} \sum_{k=1}^{N_i} t_i^{hop_k} \leq t_{i,max} \end{cases} \end{aligned} \tag{17}$$

3.2 Joint Selection Algorithm

When more than one choice is available between SVs and RVs, Algorithm 1 is proposed to generate the optimal vehicle combination for relaying. All requesters of SVs monitor the nearby vehicle status information. First, when generating a new offloading program, the requester can apply for a RV vehicle. At the same time, surrounding vehicles will also connect with others and send relaying requests to them. Then, if a request is sent to multiple RVs, one of the requests will be selected using the proposed joint selection algorithm shown below. At the same time, the request should notify other RVs to disconnect. Ultimately, as soon as a trunk connection is established, one RV immediately stops announcing its availability until the service is terminated or disconnected.

In this light, an efficient and low-complexity scheduling algorithm is designed to solve problem 1, as shown below. This algorithm involves the following four steps. The delay is calculated for all $i = 1, \cdots, \boldsymbol{SV}$ required to transfer task data between each requester and the relay node it can connect to. Let the vehicle carrying the task i be V_i and the next hop relay node be V_j. The greedy algorithm and the quick sorting method will rapidly select the optimal vehicle j to join the RV set under the constraints of low complexity. When different requesters select the same relay vehicle, the vehicle V_j that can obtain a suboptimal delay is added to the RV, and so on. The optimal problem considered is for all requesters, in which case P1 is reduced to

$$\begin{gathered}\min_{\theta_{i,j}} \sum_{i=1}^{n_{SV}} \sum_{k=1}^{N_i} \theta_{i,j} t_i^{hop_k} \\ \text{s.t. } \mathrm{n}_{SV} = n_{RV}.\end{gathered} \tag{18}$$

Then, $\boldsymbol{W}^{hop_k} = \left\{t_{i,j,k}\right\}_{n_{SV} \times n_{RV}}, i.e.,$

$$\boldsymbol{W}^{hop_k} = \begin{bmatrix} t_{1,1,k} & t_{1,2,k} & \cdots & t_{1,j,k} \\ t_{2,1,k} & t_{2,2,k} & \cdots & t_{2,j,k} \\ \vdots & \vdots & \ddots & \vdots \\ t_{i,1,k} & t_{i,2,k} & \cdots & t_{i,j,k} \end{bmatrix} \tag{19}$$

where element $t_{i,j,k}$ indicates the transmission delay between V_i and V_j at the k - th hop. When V_i and V_j have no communication relationship, $t_{i,j,k} = +\infty$. This problem is then equivalent to minimizing the sum of delays by selecting exactly one element in each row and column of $\boldsymbol{W}$, where each V_i is only permitted to operate only one V_j. This is an assignment problem that can be solved as follows [17]. When task i does not get the opportunity to offload after the k-hop relay, it satisfies $t_{i,up}^{hop_k} = t_{i,ex}^{hop_k} = 0$, and $t_{i,up}^{hop_k}$ increases the relaying time of V_i to V_j in the k - th hop. Therefore, $\text{flag}_i = 0$ is set to indicate that it is not offloaded; otherwise, $\text{flag}_i = 1$ is set and the total latency t_i for task i is found when it finishes offloading after the k - th hop relay or reaches the upper limit of N_i - th hops. This can be expressed as

$$t_i = t_{i,wait} + \sum_{k=1}^{N_i} t_{i,up}^{hop_k} + t_{i,ex}^{hop_k} + t_{i,j}^{hop_k} \tag{20}$$

where $t_{i,wait}$ indicates the time when the task is queued for offloading when task i reaches the maximum number of relay hops.

This predictive offloading method can also prevent the pressure of the requester from driving to the wireless network through the wireless backhaul link owing to the fact that the requester has left the area covered by the RSU after the local queuing calculation is offloaded. When task i ends, a matching matrix $\boldsymbol{W}$ is reformed in the new one-hop relay until all tasks are offloaded. $\boldsymbol{W}$ is a square matrix. The assignment problem can be solved by the Hungarian algorithm [15] whose complexity is $\mathcal{O}(N^3)$. The proposed joint selection algorithm is shown below in Algorithm 1.

Table 1. Joint selection algorithm process

```
Algorithm 1 Joint Selection Algorithm
Initialization:
    a) Request vehicles $\mathbf{SV} = \{1, \cdots, i, \cdots n_{SV}\}$
    b) Maximal hops $N_i$
    c)Distance between $V_i$ and $V_j$
    d)Set $\mathrm{flag}_i = 0$, $i = \{1, 2, \cdots n_{SV}\}$;
for $V_i$ in $\mathbf{SV}$ do
    Choose $V_j$ and form $\mathbf{RV} = \{1, \cdots, j, \cdots n_{RV}\}$
    for $V_j$ in $\mathbf{RV}$ do
        Calculate $t_{i,j}^{hop_1}$
    end for
    Set $\mathbf{W}^{hop_1}$
    Calculate $t_{i,up}^{hop_1}, t_{i,\mathrm{ex}}^{hop_1}, t_i^{hop_1}$
end for
while $\mathbf{W} \neq 0$ do
    Set new $V_j$ in $hop_k$, $V_j \in \mathbf{RV}_k^*$
    Update distance between $V_i \in \mathbf{SV}_k^*$ and $V_j \in \mathbf{RV}_k^*$
    Calculate $t_{i,up}^{hop_k}$ and update $\mathbf{W}^{hop_k}$
    for $i$ in $n_{SV,k}^*$ do
        if $k = N_i$ or task $i$ satisfy constraint (21) then
        Set $\mathrm{flag}_i = 1$ and calculate $t_i$
        end if
        if $\mathrm{flag}_i = 0$ then
            Set $t_{i,up}^{hop_k} = t_{i,\mathrm{ex}}^{hop_k} = 0$
        end if
    end for
    $k = k + 1$
end while
Output: relay selection $\mathbb{RV} = \{\mathbf{RV}_1^*, \mathbf{RV}_2^*, \cdots, \mathbf{RV}_k^*\}$ and the optimal total latency.
```

4 Simulation Result and Discussion

This section presents some simulation results to illustrate the performance of the optimal offloading scheme and joint selection algorithm. A vehicular network in which SVs and RVs move along a unidirectional straight road is simulated. Each RSU is mounted at a position 15 m away from the curb, and it can cover a range of 200 m [12]. The wireless channel is modelled as a Rayleigh fading channel with V2I and V2V path-loss models given by [11], $L_{V2I}(d(t)) = 100.7 + 23.5\log_{10}\left(\frac{d(t)}{d_0}\right)$, $L_{V2V}(d(t)) = 63.3 + 17.7\log_{10}\left(\frac{d(t)}{d_0}\right)$, respectively. $P_t = 1.3 \quad W$ is set for a wireless bandwidth of 1 MHz to transmit data. The background noise is $N_0 = -100 \quad dBm$, DSRC coverage radius is $D = 150$ m and $\beta = 1.05$. The computational capacities of f_m and f_v are considered to be randomly distributed on the intervals $\left[10^6,\ 10^8\right]$ cycles/s and $\left[10^5,\ 10^7\right]$ cycles/s. For a given vehicle, the velocity is randomly selected in the range of 10–30 m/s.

To evaluate the proposed relay algorithm, two approaches are compared in the simulations: V2I offloading and partial V2V offloading. Partial V2V offloading is performed by splitting the computing tasks into different vehicle calculations. Figure 5 shows the performance of exhaustive search; the proposed scheme can optimize the total delay in multivehicle or multitasking situations compared to the other methods.

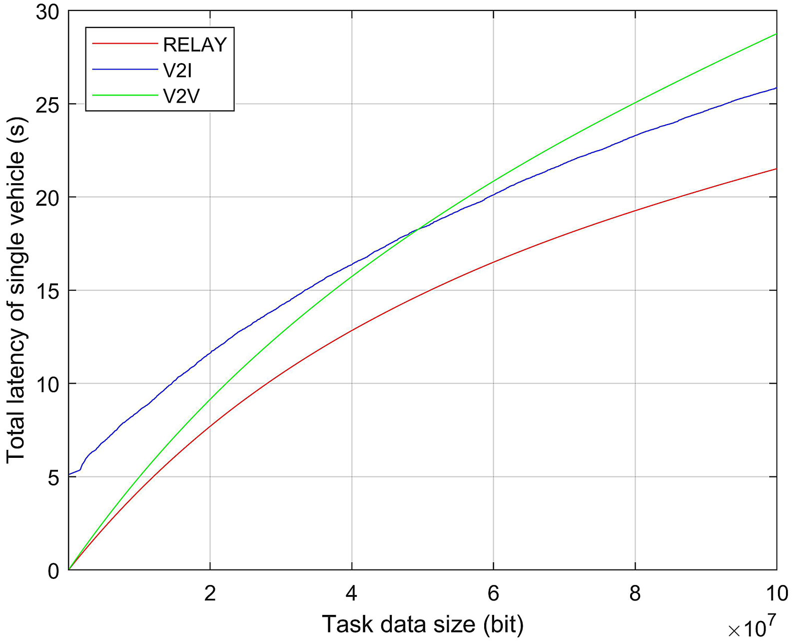

Fig. 2. Comparison of V2I and V2V optimal offloading schemes in terms of latency.

Figures 2 and 3 show comparisons of the delay gap under a single task request to more clearly analyze the effect. Figure 2 shows the total delay of the three offloading methods when the task data volumes are different. The data volume ranges from 10^5–10^8 bits. Because the local MEC server is busy in the considered scenario, even when the amount of task data is very low, the popular V2I mode still needs to be queued for offloading or be rejudged when entering the next RSU coverage, resulting in poor performance. As the amount of task data increases, the V2V mode needs to assign tasks to more vehicle calculations, resulting in increased delays. The other two unloading methods calculate that the total delay is relatively stable. Considering the fact that the vehicle velocity

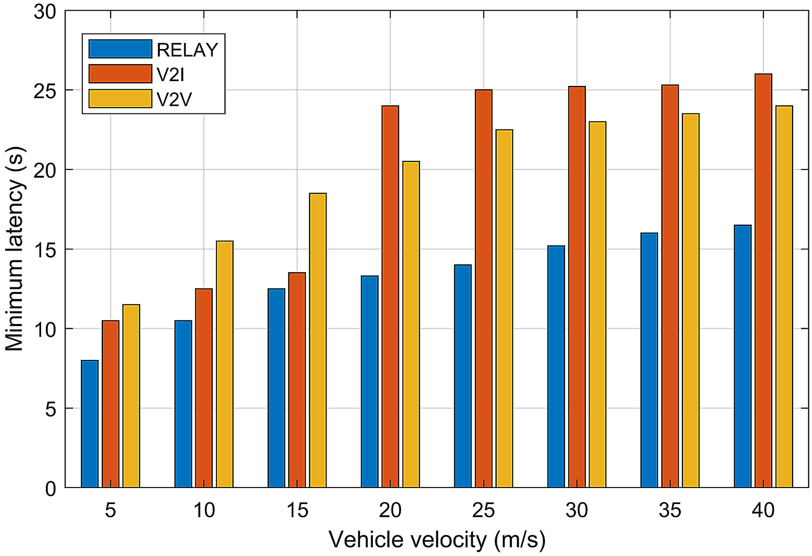

Fig. 3. Latency with vehicular speed for V2I and V2V optimal offloading schemes.

has always been an important factor in vehicle networks, the impact of mobility on the proposed scheme is analyzed with a task volume of 50M as an example. Figure 3 shows that when the velocity reaches a critical value, the V2I mode is offloaded in the current RSU, and the vehicle leaves the area before returning the result, causing the delay to rise suddenly. Similarly, the possibility of vehicles with assisted calculations returning the result to the SV in time is reduced when increasing the velocity in the V2V mode.

The proposed scheme does not occupy the computational resources of the relay car as long as communication factors such as connectivity are considered so as to achieve a lower total delay. In addition, the proposed scheme performs better and more consistently throughout the simulation, in line with the expectations that it increases network scalability.

Figure 4 shows the latency with the RSU coverage radius. According to previous simulation parameters, the vehicle velocity is in the range of 10–30 m/s. This figure shows that the delay is the minimum for a coverage radius of 200 m. Subsequently, because the coverage radius is much larger than the DSRC communication distance and the connection time with the current RSU is long, the computing task queues to offload. Then, the total delay for completing the task reaches a peak and tends to become stable as the coverage radius increases.

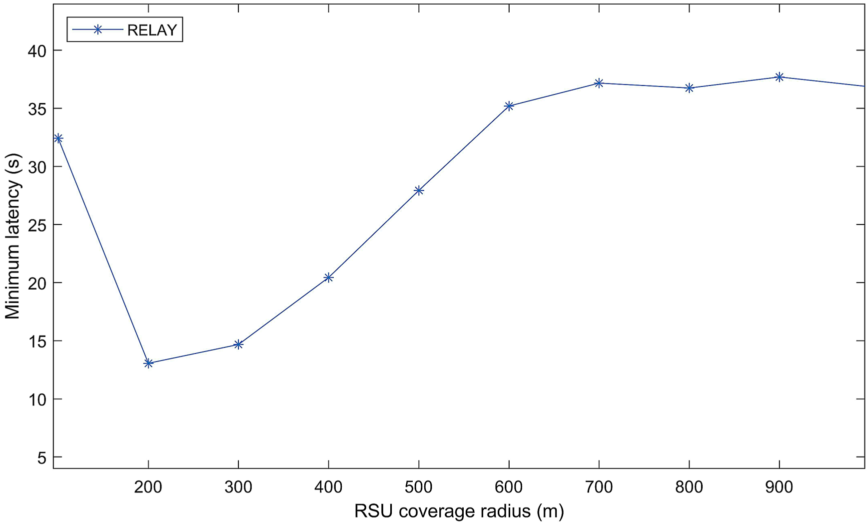

Fig. 4. Latency with RSU radius in proposed scheme.

5 Conclusion

In this study, the predictive offloading is investigated under the assistance of relay in MEC-enhanced vehicular networks. An optimal offloading scheme is proposed to reduce the total latency of computing tasks. And furthermore, a joint solution with greedy and hungarian algorithms is proposed to improve the scalability of vehicular networks. The proposed algorithm can help multiple vehicles select appropriate relay nodes to achieve higher completion efficiency of task processing. Simulation results demonstrated that our proposed optimal offloading scheme can work well. Compared with V2I and V2V without relaying, our algorithm can reduce the latency with 20%–40%.

Acknowledgement. This work is supported by ROIS NII Open Collaborative Research 2021 (21FA03).

References

1. Ma, M., He, D., Wang, H., Kumar, N., Choo, K.R.: An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks. IEEE Internet Things J. **6**(5), 8065–8075 (2019)
2. Zhang, K.,Mao, Y., Leng, Y., He, Y., Zhang, Y.: Mobile-edge computing for vehicular networks: a promising network paradigm with predictive off-loading. IEEE Veh. Technol. Mag. **12**(2),36–44 (2017)
3. Zheng, C., Feng, D., Zhang, S., Xia, X., Qian, G., Li, G.Y.: Energy efficient V2X-enabled communications in cellular networks. IEEE Trans. Veh. Technol. **68**(1), 554–564 (2019)
4. Tran, T.X., Hajisami, A., Pandey, P., Pompili, D.: Collaborative mobile edge computing in 5g networks: new paradigms, scenarios, and challenges. IEEE Commun. Mag. **55**(4), 54–61 (2017)

5. Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., Taleb, T.: Survey on multi-access edge computing for internet of things realization. IEEE Commun. Surv. Tutor. **20**(4), 2961–2991, Fourthquarter 2018
6. Yang, C., Liu, Y., Chen, X., Zhong, W., Xie, S.: Efficient mobility-aware task offloading for vehicular edge computing networks. IEEE Access **7**, 26652–26664 (2019)
7. Moltchanov, D., Kovalchukov, R., Gerasimenko, M., Andreev, S., Koucheryavy, Y., Gerla, M.: Socially inspired relaying and proactive mode selection in mmWave vehicular communications. IEEE Internet Things J. **6**(3), 5172–5183 (2019)
8. Feng, J., Liu, Z., Wu, C., Ji, Y.: Mobile edge computing for the internet of vehicles: offloading framework and job scheduling. IEEE Veh. Technol. Mag. **14**(1), 28–36 (2019)
9. Cao, X., Wang, F., Xu, J., Zhang, R., Cui, S.: Joint computation and communication cooperation for energy-efficient mobile edge computing. IEEE Internet Things J. **6**(3), 4188–4200 (2019)
10. Wang, Z., Zhong, Z., Zhao, D., Ni, M.: Vehicle-based cloudlet relaying for mobile computation offloading. IEEE Trans. Veh. Technol. **67**(11), 11181–11191 (2018)
11. Luoto, P., Bennis, M., Pirinen, P., Samarakoon, S., Horneman, K., Latva-aho, M.: Vehicle clustering for improving enhanced LTE-V2X network performance. In: 2017 European Conference on Networks and Communications (EuCNC), Oulu, pp. 1–5, June 2017
12. Wang, H., Li, X., Ji, H., Zhang, H.: Federated Offloading Scheme to Minimize Latency in MEC-Enabled Vehicular Networks 2018 IEEE Globecom Workshops (GC Wkshps), pp. 1–6. Abu Dhabi, United Arab Emirates (2018)
13. Guo, H., Liu, J., Zhang, J.: Efficient computation offloading for multi-access edge computing in 5G HetNets. In: 2018 IEEE International Conference on Communications (ICC), Kansas City pp. 1–6, May 2018
14. Guo, H., Liu, J.: Collaborative computation offloading for multiaccess edge computing over fiber-wireless networks. IEEE Trans. Veh. Technol. **67**(5), 4514–4526 (2018)
15. Desikan, K.E.S., Kotagi, V.J., Murthy, C.S.R.: Smart at right price: a cost efficient topology construction for fog computing enabled iot networks in smart cities. In: 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, pp. 1–7, September 2018