How to Leverage Mobile Vehicles to Balance the Workload in Multi-Access Edge Computing Systems

Yiqin Deng†, Zhigang Chen†*, Xianhao Chen‡, Xiaoheng Deng†, and Yuguang Fang‡, *Fellow, IEEE*

***Abstract*—In this correspondence, we investigate the joint relay and server selection problem for a novel vehicle-assisted multi-access edge computing (MEC) system, where vehicles on the road relay data from an end device to appropriate MEC servers for computing in a store-carry-forward manner. To harness the stochastic nature of task arrivals, vehicle availability, and resource availability at MEC servers, we design a relaying scheme to minimize the average task latency based on a Markov decision process. Our study shows that our proposed scheme is highly effective in balancing the communications and computing while minimizing the end-to-end latency for task completion.**

***Index Terms*—Multi-access edge computing (MEC), vehicular networks, relay selection, latency minimization, Markov decision process (MDP).**

I. Introduction

By offloading computation-intensive tasks from end devices to multi-access edge computing (MEC) servers, the performance at end devices in terms of task completion can be significantly enhanced [1], [2]. However, the resource limitation at MEC servers in terms of computing power, communications spectrum, and storage may become bottlenecks due to the ever-increasing number of end devices that scramble for MEC services [3]. For example, in surveillance video analytics for public safety applications in smart cities, a large volume of high-resolution videos should be transmitted from street cameras to surrounding MEC servers, which imposes great challenges to both computing and communication infrastructure. In this context, which videos are transported to and computed by which MEC servers becomes a critical design issue and hence task assignment to appropriate MEC servers under constraints on their communication-computing resources and geolocations has become an important and interesting research problem.

Yiqin Deng, Zhigang Chen, and Xiaoheng Deng are with the school of Computer Science and Engineering, Central South University, Changsha 410083, China (e-mail: dengyiqin@csu.edu.cn; czg@csu.edu.cn; dxh@csu.edu.cn).

Xianhao Chen and Yuguang Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, USA, FL 32611 (e-mail: xianhaochen@ufl.edu; fang@ece.ufl.edu).

There already exist intensive research efforts to address load balancing problems in MEC systems [4]–[7]. In [4], Rodrigues *et al.* studied an offloading decision problem in a co-existing system consisting of multiple MEC servers and a centralized cloud, determining whether users' requests are executed locally or routed to MEC/cloud servers to minimize the service delay. By leveraging the cooperation among MEC servers, Li *et al.* designed a multi-hop cooperative offloading mechanism in which tasks arriving at one MEC server can be fulfilled locally, or forwarded to one of the non-local but powerful MEC servers via backhaul links [5]. In [6], Yuan *et al.* investigated the joint vehicle route planning and task offloading problem under service delay constraints. Nevertheless, all these papers ignored the fact that tasks may fail to be uploaded to an MEC server directly due to poor channel condition or spectrum scarcity at the first hop, especially when the size of input data is large. An example is the scenario for the tasks generated by a wireless surveillance camera out of cellular coverage or within a congested cell. Recently, we have proposed to employ vehicles to beef up MEC systems [8]–[11], which can be regarded as "vehicle as a service (VaaS)". For example, by exploiting the capabilities of communication and storage on vehicles, Hui *et al.* proposed a relay selection scheme to extend the communication coverage of MEC servers, where a vehicle's computing task can be relayed to one of the MEC servers ahead of its driving direction via surrounding vehicles [12]. However, the dynamics of task generations at end devices and resource availability at MEC servers are not taken into consideration in the existing works on vehicle-assisted MEC systems. Considering a practical queuing model, tasks may have to be placed in buffers due to the surging service demands and the lack of communication and/or computing resources, which, however, cannot be captured by one-shot optimization methods.

Based on the above observations, this paper proposes a novel architecture to facilitate task uploading in MEC systems by introducing vehicles as store-carry-forward relays. Under this proposed architecture, a relay and server selection problem is investigated with the objective to minimize the end-to-end service delay. To this end, we build up a single-queue multi-server model and develop a Markov chain to characterize queue dynamics with the uncertainty of the availability of vehicular relays and resources at MEC servers. Finally, we

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3119189, IEEE Transactions on Vehicular Technology

design an efficient and optimal strategy based on the problem formulation.

II. System Model and Problem Formulation

A. Relay-based MEC framework

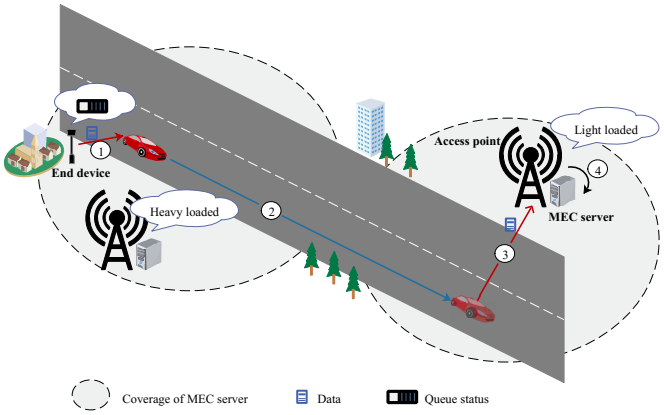

Fig. 1. Example of the relay-based MEC framework: data are delivered to the appropriate MEC server according to the queue status at the end device, availability of relay vehicles, and resource availability at MEC servers.

We consider a time-slotted system $t \in \mathcal{T} = \{0, 1, 2, \cdots, T\}$, where one end device intends to access MEC services by uploading data to one of J MEC servers. Since a device may not access the service of an MEC server because it is out of its communication range or the channel condition between them is poor, it may seek helps from vehicles on the move to relay its data, potentially in a store-carry-forward manner. Let p_j be the probability of resource availability at the j-th MEC servers in terms of communication, computing, and storage during each time-slot. The speed and probability of vehicles that reaches different MEC servers varies. For simplicity, we assume that the vehicles are categorized into V types according to their moving directions and speeds, i.e., vehicles with the same driving direction and speed are classified into the same category. It is assumed that vehicles of the same category have the identical moving times from the end device to the destination MEC server. Let $p'_{(v,j)}$ and $g_{(v,j)}$ respectively represent the probability that the v-th category of vehicles reaches the j-th MEC server and the time taking to move from the end device to the proximity of the j-th MEC server.

In addition, the availability of moving vehicles, which is assumed to remain unchanged during each time-slot, is modeled as a two-state Markov chain. Specifically, when a v-th category of vehicles is not available in the current time-slot, it will become available in the next time-slot with probability q_v, whose number is V_1. When a v-th category of vehicles is available in the current time-slot, it will not be available in the next time-slot with probability q'_v, whose number is V_2 where $V_1 + V_2 = V$.

As shown in Fig. 1, the proposed relay-based MEC framework consists of four phases: **i)** the data transfer between the end device and a relay vehicle, **ii)** vehicles' movement with stored data, **iii)** the data uploading between the relaying vehicle and the destination MEC server, and **iv)** task processing/computing at the MEC server.

B. Stochastic task arrivals

The task arrivals at the end device at the beginning of time-slot t, $a(t)$, is assumed to follow the distribution as

$$P\{a(t) = b\} = p_b, \tag{1}$$

where b is a non-negative integer. Based on this, the average rate of task generations at the end device is given by

$$\alpha_0 = \lim_{t\to\infty} \frac{1}{T} \sum_{t=0}^{T} a(t) = \sum_{b=0}^{B} b \cdot p_b. \tag{2}$$

where B is the maximal number of tasks generated in one time-slot, i.e., $b = 0, 1, 2, \cdots, B$, and $p_b \in [0, 1]$.

C. Single-queue multi-server model for task routing

Due to the uncertainty of the availability of relay vehicles and MEC servers, there could be three cases for data transmissions in the system: **i)** no data transmission when the resources on the selected MEC server are not available, **ii)** μ_0 data can be directly transmitted from the end device to the target MEC server when no vehicle is available while the MEC server is capable of receiving data, and **iii)** μ_1 data can be transmitted to a vehicle when the selected relay vehicle and MEC server are simultaneously available. Thus, the transmission rate between the end device and a vehicle or the associated MEC server (i.e., the service rate in the queueing model), at time-slot t, $\mu(v, j, t)$ can be summarized as

$$\mu(v,j,t) = \begin{cases} 0 & \text{if } j = 0; \\ \mu_0 & \text{if } v = 0, j \neq 0; \\ \mu_1 & \text{if } v \neq 0, j \neq 0, \forall\, t \in \mathcal{T}. \end{cases} \tag{3}$$

where both μ_0 and μ_1 are positive integers. We assume that $\mu_0 < \mu_1$ since in the considered scenario, the distance between the end device and relay vehicle is typically shorter than that between the end device and the MEC server.

The generated but not yet transmitted tasks are placed in the task buffer at the end device, whose buffer size is assumed to be a large value M. Let $Q(t)$ denote the sum length of tasks in the waiting line, i.e., the number of tasks queueing in the buffer, and tasks newly arrived $a(t)$ at the end device at the beginning of the t-th time-slot. According to queueing model, we have

$$Q(t+1) = \max\{Q(t) - \mu(v,j,t), 0\} + a(t+1), \forall t \in \mathcal{T}. \tag{4}$$

D. The MDP framework

In each time slot, the relaying decision is made based on the queue length and the availability of relay vehicles, which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3119189, IEEE Transactions on Vehicular Technology

are summarized as

$$\mathcal{S}(t) = (Q(t), N(t)), \tag{5}$$

where $N(t) = \{0,1\}^{1\times V}$ represents the availability of V categories of the vehicles during the t-th time-slot. In $N(t)$, the v-th element $N_v(t) = 1$ when at least one category-v vehicle available during the t-th time slot, and $N_v(t) = 0$ otherwise.

Based on system state $\mathcal{S}(t)$ and parameters, including $p'_{(v,j)}$ and p_j, the end device chooses its action $\eta(t)$ at each time slot. The available actions at the t-th time-slot are collected in a set $\mathcal{A} \subset \{(v,j)|v \in \{0,1,2,\cdots,V\}, j \in \{0,1,2,\cdots,J\}\}$, e.g., when $\eta(t) = (0,0)$, the data will still be waiting in the queue, when $\eta(t) = (0,j)$, the end device transmits its data to the j-th MEC server directly, and when $\eta(t) = (v,j)$ and $v \neq 0$, the end device transmits its data to the j-th MEC server via the v-th selected relay. The mapping from $\mathcal{S}(t)$ to $\eta(t)$ is called relaying policy in this paper, which is denoted by f, i.e., $f(\mathcal{S}(t)) = \eta(t)$. Specifically, we use $f^{(v,j)}_{(m,\boldsymbol{k})}$ to describe the probability of selecting the v-th relay and the j-th destination given the system state $Q(t) = m$ and $N(t) = \boldsymbol{k}$. The normalization constraint on $f^{(v,j)}_{(m,\boldsymbol{k})}$ is given by

$$\sum_{j=0}^{J}\sum_{v=0}^{V} f^{(v,j)}_{(m,\boldsymbol{k})} = 1,\ \forall\, m, \boldsymbol{k}. \tag{6}$$

Moreover, we should guarantee that the amount of transmitted data is no more than that of the remaining data, and also avoid the buffer overflow, yielding

$$f^{(v,s)}_{(q,\boldsymbol{k})} = 0,\ \text{if } k_v \cdot \mu(v,j) > q, \tag{7}$$

$$f^{(v,s)}_{(q,\boldsymbol{k})} = 0,\ \text{if } q - k_v \cdot \mu(v,j) > M - B, \tag{8}$$

where k_v is the v-th element in $\boldsymbol{k}$.

Since the transition of queue state from h to i, vehicle availability state from $\boldsymbol{k} = \{k_1, k_2, \cdots, k_v\}$ to $\boldsymbol{l} = \{l_1, l_2, \cdots, l_v\}$, whose probability are denoted by $\lambda_{(h,\cdot)\to(i,\cdot)}$, $\lambda_{(\cdot,\boldsymbol{k})\to(\cdot,\boldsymbol{l})}$, respectively, and resource availability at MEC servers, are independent, the state transition probability of this MDP, $\lambda_{(h,\boldsymbol{k})\to(i,\boldsymbol{l})}$, from the state $(h,\boldsymbol{k})$ to the state $(i,\boldsymbol{l})$, where $(h,\boldsymbol{k})$, $(i,\boldsymbol{l}) \in \mathcal{S}$, can be expressed as

$$\lambda_{(h,\boldsymbol{k})\to(i,\boldsymbol{l})} = \lambda_{(h,\cdot)\to(i,\cdot)} \cdot \lambda_{(\cdot,\boldsymbol{k})\to(\cdot,\boldsymbol{l})}, \tag{9}$$

where $\lambda_{(h,\cdot)\to(i,\cdot)}$ can be derived from Eq. (4), i.e.,

$$\lambda_{(h,\cdot)\to(i,\cdot)} = \begin{cases} \displaystyle\sum_{b=0}^{B}\sum_{j=0}^{J}\sum_{v=0}^{V} p_b p'_{(v,j)} q_j f^{(v,j)}_{(h,\boldsymbol{k})} \mathbb{1}_{\{h-\mu(v,j)+b=i\}} & \\ \qquad\qquad \text{if } i > 0; & \\ \displaystyle\sum_{b=0}^{B}\sum_{j=0}^{J}\sum_{v=0}^{V} p_b p'_{(v,j)} q_j f^{(v,j)}_{(h,\boldsymbol{k})} \mathbb{1}_{\{h-\mu(v,j)+b\le i\}} & \\ \qquad\qquad \text{if } i = 0; & \end{cases} \tag{10}$$

and the indicator function $\mathbb{1}_{\{\cdot\}}$ equals 1 if $\{\cdot\}$ holds, 0 otherwise. Based on the previous definition of the availability of vehicles, the transition probability $\lambda_{(\cdot,\boldsymbol{k})\to(\cdot,\boldsymbol{l})}$ can be expressed as

$$\lambda_{(\cdot,\boldsymbol{k})\to(\cdot,\boldsymbol{l})} = (q_v)^{V_1}(q'_v)^{V_2}. \tag{11}$$

Let $\pi_{(m,\boldsymbol{k})}$ and $\boldsymbol{\pi}_{(M+1)\times 2^V}$ represent the steady-state distribution probability of the state $(m,\boldsymbol{k})$ and the steady-state distribution of this Markov chain, respectively. The matrix $\boldsymbol{\pi}_{(M+1)\times 2^V}$ is then converted into a column vector, denoted by $\boldsymbol{\pi}$, which satisfies

$$\boldsymbol{H}\boldsymbol{\pi} = \boldsymbol{\pi}, \tag{12}$$

$$\mathbf{1}\boldsymbol{\pi} = 1, \tag{13}$$

where $\mathbf{1} = [1,1,\cdots,1]$, $\boldsymbol{H}$ is the transition matrix of the Markov chain, whose size is $2^V(M+1) \times 2^V(M+1)$, and its elements are defined by Eq. (9).

E. The average task latency

From the Little's Law [13], the average queue length equals the average task arrival rate multiplied by the average task queuing time. For a simplified expression, we introduce an integer parameter n and $1 \le n \le 2^V$. Therefore, the average waiting time at the end device D^q is given by

$$\begin{aligned} D^q &= \lim_{t\to\infty}\frac{1}{T}\sum_{t=0}^{T}[Q(t) - a(t)]/\alpha_0 \\ &= \frac{1}{\alpha_0}\sum_{n=1}^{2^V}\sum_{m=0}^{M} m \cdot \pi_{(m,n)} - 1. \end{aligned} \tag{14}$$

The average travel time of relay vehicles D^m from the transmission coverage of the end device to that of MEC servers can be given as

$$D^m = \sum_{n=1}^{2^V}\sum_{m=0}^{M}\sum_{j=0}^{J}\sum_{v=0}^{V} g_{(v,j)} f^{(v,j)}_{(m,n)} \pi_{(m,n)}, \tag{15}$$

where $g_{(0,j)} = 0$, corresponding to the case that no vehicle is chosen (hence the travel time equals 0).

The average transmission time between the end device and its relay vehicle can be neglected since the task can be transmitted to the vehicle while the vehicle moves toward MEC servers. The average uploading time between the relay vehicle and the associated MEC server and the average computing time at MEC server are assumed to be constant, the summation of which is denoted by D^e. There exist a lot of papers on resource allocation for task transmissions and computing at MEC servers, which is not the focus of this paper.

Since the download time of the task computing results from an MEC server is typically negligible, we neglect it and obtain the following expression for the average task latency D as follows

$$D = D^q + D^m + D^e. \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3119189, IEEE Transactions on Vehicular Technology

III. Latency-minimizing Relaying

In the considered scenario, intuitively, the task latency will be reduced if the end device forwards data to MEC servers for computation via relay vehicles, instead of directly transmitting data to remote MEC servers with a low transmission rate. However, the relay vehicles and the surrounding MEC servers might not be available, which may cause a prolonged waiting time at the end device. Thus, the central scheduler has to make proper decisions on task relaying, i.e., which relay and destination should be selected, according to the system states, such as the queue lengths, the availability of relay vehicles, and the resource availability at MEC servers.

In this paper, we are interested in minimizing the average task completion latency, which refers to the latency between the time the data is generated at the end device and the time the task completion result is successfully received. Based on the analysis of the average task completion latency and parameters of MDP in Section II, we formulate the average latency minimization problem (LMP) as follows

$$
\begin{aligned}
(\mathbf{LMP}) \min_{f^{(v,j)}_{(m,n)}} \quad & \frac{1}{\alpha_0}\sum_{n=1}^{2^V}\sum_{m=0}^{M} m\cdot\pi_{(m,n)} - 1 \\
& + \sum_{n=1}^{2^V}\sum_{m=0}^{M}\sum_{j=0}^{J}\sum_{v=0}^{V} g_{(v,j)} f^{(v,j)}_{(m,n)}\pi_{(m,n)} + D^e \quad (17)\\
s.t. \quad & \sum_{j=0}^{J}\sum_{v=0}^{V} f^{(v,j)}_{(m,n)} = 1,\ \forall\, m,n, \quad (17a)\\
& f^{(v,j)}_{(m,n)} \geq 0,\ \forall\, v,j,m,n, \quad (17b)\\
& \boldsymbol{H\pi} = \boldsymbol{\pi}, \quad (17c)\\
& \mathbf{1\pi} = 1, \quad (17d)\\
& \pi_{(m,n)} \geq 0,\ \forall\, m,n, \quad (17e)
\end{aligned}
$$

To solve $\mathbf{LMP}$, we use $y^{(v,j)}_{(m,n)} = \pi_{(m,n)} f^{(v,j)}_{(m,n)}$ to substitute $\pi_{(m,n)}$- and $f^{(v,j)}_{(m,n)}$- related variables in $\mathbf{LMP}$. Moreover, constraints (17c) and (17d) are converted into a matrix equation with constant matrix $\boldsymbol{G}$ and variables $y^{(v,j)}_{(m,n)}$, where matrix $\boldsymbol{G}$ consists of coefficients of $y^{(v,j)}_{(m,n)}$ after moving all variables to the left-hand side of the equations. Thus, we obtain an equivalent problem as follows

$$
\begin{aligned}
(\mathbf{LMP}') \min_{y^{(v,j)}_{(m,n)}} \quad & \sum_{n=1}^{2^V}\sum_{m=0}^{M}\sum_{j=0}^{J}\sum_{v=0}^{V}[\frac{m}{\alpha_0} + g_{(v,j)}]y^{(v,j)}_{(m,n)} \\
& - 1 + D^e \quad (18)\\
s.t. \quad & \sum_{n=1}^{2^V}\sum_{m=0}^{M}\sum_{j=0}^{J}\sum_{v=0}^{V} y^{(v,j)}_{(m,n)} = 1, \quad (18a)\\
& y^{(v,j)}_{(m,n)} \geq 0,\ \forall\, v,j,m,n, \quad (18b)\\
& \boldsymbol{Gy} = 0. \quad (18c)
\end{aligned}
$$

The equivalency between $\mathbf{LMP}$ and $\mathbf{LMP}'$ can be similarly derived as in [14]. The $\mathbf{LMP}'$ is a linear programming, whose optimal solution $y^{(v,j)^*}_{(m,n)}$ can be obtained efficiently. Therefore, the steady state distribution of the Markov chain can be given by

$$
\pi^*_{(m,n)} = \sum_{j=0}^{J}\sum_{v=0}^{V} y^{(v,j)^*}_{(m,n)}. \quad (19)
$$

Finally, we can derive the relaying policy $f^{(v,j)^*}_{(m,n)}$ as follows

$$
f^{(v,j)^*}_{(m,n)} = \begin{cases} \frac{y^{(v,j)^*}_{(m,n)}}{\pi^*_{(m,n)}} & \text{if } \pi^*_{(m,n)} \neq 0, \\ \frac{1}{|\mathcal{A}|} & \text{if } \pi^*_{(m,n)} = 0, \end{cases} \quad (20)
$$

where $|\mathcal{A}|$ is the cardinality of the feasible set of action (v,j) for state (m,n).

IV. Numerical results

In this section, we evaluate the performance of our proposed relaying policy. We set $V = 3$, $J = 5$, $M = 4$, $B = 3$, $p_v = [0.6, 0.3, 0.8]$, $q_j = [0.3, 0.8, 0.5, 0.7, 0.4]$, $p'_{(v,j)} = [0.2, 0.4, 0.1, 0.1, 0.2; 0.3, 0.2, 0.5, 0, 0; 0.1, 0.3, 0.2, 0, 0.4]$, $q'_v = [0.6, 0.3, 0.8]$, and the moving time function is set to $g(v,j) = 10/(v+j)$. The distribution of task generations at the end device is set to $p_0 = p_1 = p_2 = p_3 = 0.25$, and thus the average task arrival rate can be obtained according to Eq. (2), i,e., $a_0 = 1.5$.

In the simulation, the proposed *MDP-based relay policy* is compared with a baseline scheduling policy, namely *no-relay policy*, whose basic idea is to offload tasks to MEC servers directly. This approach does not make use of relay vehicles to assist data delivery between the end device and edge servers, which corresponds to what has been commonly done in the literature, i.e., the dynamic resource availabilities for communications and computing are not taken into consideration.

In Fig. 2, the average latencies achieved by the *MDP-based relay policy* and *no-relay policy* are compared under different average task arrival rates a_0. It can be observed that with the increase of average task arrival rate a_0, the average latencies for both two policies increase. The uncertainty information of the system is not utilized by the *no-relay policy*, which results into significant performance loss. Moreover, with the help of relay vehicles to deliver tasks between the end device and edge servers, the *MDP-based relay policy* can better utilize the spectrum resource and computing resource, hence achieving lower average latency under a given average task arrival rate, which validates the advantage of the use of relays.

Fig. 3 illustrates the average latency with different average task arrival rates a_0 and different vehicle availability probabilities p, where $p = \beta * [0.1, 0.2, 0.3]$ with $\beta = 1, 2, 3$, respectively. Intuitively, with the increase of β, more vehicles become available, leading to lower average task completion latency. It has been verified that the system performance, i.e., the average latency, can be improved significantly when vehicle availability probabilities increase.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3119189, IEEE Transactions on Vehicular Technology

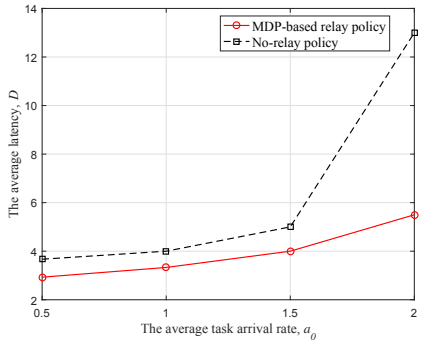

Fig. 2. The average latency vs. the average task arrival.

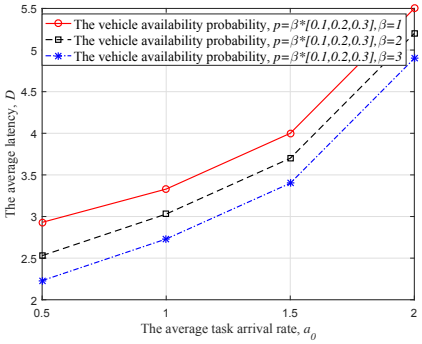

Fig. 3. The average latency vs. the average task arrival with different vehicle availability probabilities.

V. Conclusion

In this correspondence, we have investigated the relaying problem for vehicle-assisted edge computing systems to balance the workload at edge servers while better utilizing the spectrum resource from end devices to edge servers. The goal is to minimize the average task completion latency under given system resources. With the help of massive moving vehicles on the road, tasks can be forwarded to the appropriate edge servers in a timely fashion when the direct links from end devices to servers are unavailable or not preferable, thus greatly enhancing the quality-of-experience of MEC applications. By formulating the relaying problem based on an MDP framework, not only the information of relay vehicles can be efficiently utilized but also the average task completion latency can be analyzed. Based on the formulated problem, the optimal latency minimizing policy has been discovered. Finally, the performance enhancement with the assistance of relay vehicles has been validated with simulations.

References

[1] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, thirdquarter 2017.

[2] T. K. Rodrigues, K. Suto, H. Nishiyama, J. Liu, and N. Kato, "Machine learning meets computation and communication control in evolving edge and cloud: Challenges and future perspective," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 38–67, firstquarter 2020.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.

[4] T. Koketsurodrigues, J. Liu, and N. Kato, "Offloading decision for mobile multi-access edge computing in a multi-tiered 6g network," *IEEE Transactions on Emerg. Topics in Comput.*, to be published, doi: 10.1109/TETC.2021.3090061.

[5] Y. Li, X. Wang, X. Gan, H. Jin, L. Fu, and X. Wang, "Learning-aided computation offloading for trusted collaborative mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 19, no. 12, pp. 2833–2849, Dec. 2020.

[6] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo, and X. Shen, "A joint service migration and mobility optimization approach for vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9041–9052, Aug. 2020.

[7] Y. Deng, Z. Chen, X. Chen, and Y. Fang, "Throughput maximization for multi-edge multi-user edge computing systems," *IEEE Internet of Things J.*, to be published, doi: 10.1109/JIOT.2021.3084509.

[8] H. Ding and Y. Fang, "Virtual infrastructure at traffic lights: vehicular temporary storage assisted data transportation at signalized intersections," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12 452–12 456, Dec. 2018.

[9] H. Ding, C. Zhang, B. Lorenzo, and Y. Fang, "Access point recruitment in a vehicular cognitive capability harvesting network: how much data can be uploaded?" *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6438–6445, July 2018.

[10] H. Ding, Y. Guo, X. Li, and Y. Fang, "Beef up the edge: spectrum-aware placement of edge computing services for the Internet of Things," *IEEE Trans. Mob. Comput.*, vol. 18, no. 12, pp. 2783–2795, Dec. 2019.

[11] X. Chen, L. Zhang, Y. Pang, B. Lin, and Y. Fang, "Timeliness-aware incentive mechanism for vehicular crowdsourcing in smart cities," *IEEE Trans. Mob. Comput.*, Jan. 2021.

[12] Y. Hui, Z. Su, T. H. Luan, and C. Li, "Reservation service: Trusted relay selection for edge computing services in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2734–2746, Dec. 2020.

[13] J. D. Little and S. C. Graves, "Little's law," in *Building intuition*. Springer, 2008, vol. 115, pp. 81–100.

[14] D. Han, W. Chen, and Y. Fang, "Joint channel and queue aware scheduling for latency sensitive mobile edge computing with power constraints," *IEEE Trans. on Wireless Commun.*, vol. 19, no. 6, pp. 3938–3951, June 2020.