A Deadline-Aware Offloading Scheme for Vehicular Fog Computing at Signalized Intersection

Beichen Yang*, Min Sun†, Xiaoyan Hong*, Xiaoming Guo*
*Department of Computer Science, †Department of Mathematics
The University of Alabama, Tuscaloosa, AL 35487, USA
byang12@crimson.ua.edu, msun@ua.edu, hxy@cs.ua.edu, xguo29@crimson.ua.edu

***Abstract*—Vehicular Fog Computing is a promising paradigm to exploit the computational resources on vehicles for providing computing services. Meanwhile, the unique mobility patterns of vehicles at the urban signalized intersection can be used to improve the performance of Vehicular Fog. In this paper, we present an offloading scheme tailored to the signalized intersection, by taking the vehicles' mobility pattern at the intersection. Our offloading scheme addresses the conditions of deadlines and the cost of replications. Predicting mobility is a key factor in further calculating the connection time and the link bandwidth for finding appropriate service vehicles at intersections. The scheme achieves maximum offloading success rate under given deadline constraints and budgets. Our evaluation uses realistic traffic data generated by traffic simulator SUMO. The results are compared with a few benchmarks given multiple parameters. The proposed scheme can achieve a satisfactory offloading success rate compared with the best-case offloading scenario.**

***Index Terms*—Connected Vehicles, Fog Computing, Task Offloading, Signalized Intersection**

I. Introduction

In recent years, the car manufacturing industry is moving toward the trend of connected vehicles. Compared with traditional vehicles, connected cars are equipped with On-BoardUnit (OBU) for wireless communication and embedded computers for computational purposes. Meanwhile, the development and deployment of Intelligent Transportation System (ITS) cooperated with connected vehicular technologies enables many new applications such as safety assistance, road navigation, and self-driving, etc. Those applications often require various on-vehicle sensors; for example, GPS, radar, dashcam, and will inevitably generate an enormous amount of data waiting to be processed. As research [1] points out, the scale of this vehicular data volume is usually in the order of terabytes per driving hour and still keeps growing rapidly. However, current OBUs usually do not have adequate comping power to deal with such a large amount of data. To address the issue of increasing computing demand of connected vehicles and other mobile devices, new computing paradigms such as Fog Computing and Mobile Edge Computing (MEC) are proposed as a possible solution.

Within the context of connected cars and vehicular network, the general idea of Vehicular Fog Computing (VFC) [2] and Vehicular MEC [3] is that when one vehicle generates a task that cannot be processed by itself due to lacking enough computing power, the task will be offloaded to another computation platform. The difference between Fog and MEC in terms of which computation entity is responsible to handle the task. In MEC, the tasks generated by vehicles are offloaded to the Edge servers connected with the base stations through a Vehicle to Infrastructure (V2I) manner. Those base stations are deployed alongside the Road Side Units (RSU). And the tasks could be either processed by edge servers themselves or be further offloaded to the remote Cloud data center via the backbone network. In VFC, the tasks will be offload to other vehicles who currently have available computing resources directly through Vehicle to Vehicle (V2V) communication. Our research is based on the Fog Computing paradigm where vehicles offload their computing tasks to other vehicles without the help of RSU and Edge servers.

In this research, we consider the urban signalized intersections as ideal places for hosting Vehicular Fog Computing. Our interest is based on two major reasons. Firstly, the high mobility of vehicles will inevitably introduce great difficulties for wireless communication and connectivity. But vehicles may stop at intersections for short periods of time due to red lights, which creates a more stable environment for wireless communication and connectivity. Secondly, intersections usually have high vehicle density compared to other road segments. The concentration of vehicles will provide a good resource pool for the vehicles to share computing tasks.

However, utilizing a signalized intersection to enhance the performance of task offloading is a non-trivial task due to a few reasons. First, we consider the case that computing tasks have to be constrained by deadlines [5] [4] [6]. To increase the possibility that an offloaded task will finish on time, replications of the same task will be used. But the number of replicated tasks should be limited due to budget constraints. Second, the vehicle mobility at intersections is largely impacted by signal control and the arrival time to the intersection. Predicting the connection time that would allow a vehicle to finish an offloaded task and return the results has to count all the potential moving directions. But no existing work has found on modeling the mobility.

In this paper, we study the VFC offloading problem with the consideration of deadlines and task replication with a limited offloading budget at the signalized intersections. We proposed

The work is supported partly by the National Science Foundation under Grants No.1719062. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

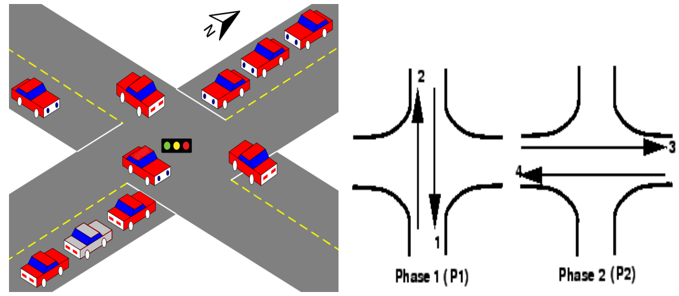
a location prediction method that exploits the traffic signal data alongside the current vehicle movement information to forecast the future position of vehicles. Based on the method, the distance between a task vehicle (TaV) and a candidate service vehicle(SeV) can be calculated. The latter allows further calculation of the condition and the data rate of the wireless connection between the two vehicles. In the paper, we formulate the problem as a deadline-aware offloading problem which aims to select appropriate offloading SeVs to achieve maximum offloading successful rate. The proposed scheme is evaluated and compared with other benchmarks. The results show that our scheme can achieve satisfactory offloading successful rate under various settings.

The rest of the paper is organized as follows. Section II reviews existing literature about various vehicular fog offloading problems. In Section III, we present the system model for the proposed offloading scenario at a signalized intersection, followed by the problem formation and the location prediction offloading scheme in section III. Experimental results are presented in Section IV and Section V concludes the paper.

II. Related Work

Vehicular task offloading has to consider vehicle computing resources, wireless connectivity, task deadlines, budget, etc. The general goal of the deadline constraint problem is to minimize the number of offloading attempts that have the total offloading latency that goes beyond the deadline. Both of [4] and [5] focus on the deadline constraint problem where RSUs get involved in data transmission between vehicles. [5] adopts a Markov Decision Process (MDP) to determine which vehicle should be chosen as the SeV according to their mobility information and the computing power. Chen et al. improve this offloading scheme in [4] by replacing MDP algorithm with a Multi-Armed Bandit (MAB) based learning algorithm to reduce the heavy computation load introduced by MDP. Sun et al. also discussed a deadline constraint task offloading problem in [6], and is the most relevant research compared with our current study. Like [4], [6] also adopts a MAB algorithm to solve the optimization problem. The difference is that [6] focuses on a highway scenario, where the TaV directly offloads its tasks to the SeVs that moving in the same direction without the help of the RSUs. Similar to [6], the vehicle also communicated in a directed V2V manner in this research. The difference is that we assume the offloading happens at the signalized intersection instead of the highway.

Our work differs from the existing works in terms of the focus on the signalized intersections, which calls for different and more sophisticated mobility modeling and prediction. While our previous research has shown that VFC can be host upon the vehicle crowd at the intersection with dynamics and inter-correlation with the traffic signal [7] [8], here we study the mobility patterns in relating to traffic signal information for a deadline-ware offloading scheme. Predicting the movement of an individual vehicle at an intersection is quite different from a general vehicle mobility model. A few recent works have used Long Short Term Memory (LSTM)



(a) Simple Intersection Scenario (b) Movements in Two Phase Signal System

Fig. 1: Single Intersection

based algorithm in predicting the vehicle movement trajectory. They can successfully be used for collision avoidance in self-driving vehicle applications [10] [11]. However, it is difficult for this approach to predict vehicle movements at a signalized intersection due to the factor that the future movement of a vehicle mainly depends on the traffic signal rather than its historical movement trajectory.

III. System Model and Problem Formulation

In this section, we introduce our system model and models used in formulating the task offloading problem and the problem itself.

A. System Model Overview

Without loss of generality, the vehicles can be divided into two categories in the system model. The first category is called task vehicles (TaVs) which have some tasks that can not be handled by themselves thus need to be offload to other vehicles. The second category is called Service Vehicles (SeVs) which process the tasks generated by TaVs and return the result to the TaVs. What's more, The role of TaV and SeV for each vehicle is interchangeable depending on whether the vehicle has a task to offload or whether it has resources to support others computing needs.

Fig. 1a shows a simple scenario of a signalized intersection. The grey car is the TaV, while, all the other red cars are potential SeVs. With a deadline constraint of a task, the TaV has to select a group of SeVs with considerations over the factors contributing to the completion latency of the task. In addition, the number of the selected SeVs needs to confine to a budget as it is often a consideration in real use. As such, for all the SeVs currently located at this intersection, the selection of the SeVs should be based on a prediction of their offloading completion latency with the budget limit. Offloading completion latency (or, offloading latency) is influenced by two major factors. One is task computing time, and the other is the data transmission time over the wireless link. Sevs will broadcast their computational information. A TaV thus learns and uses it in its decision. The data transmission time is related to the distance between TaV and SeV per the signal quality in urban areas [9]. Since vehicle mobility is highly dynamic at the intersection, predicting this distance is necessary for acquiring data transmission time.

B. Traffic Model

Consider the scenario described in Fig. 1a, the approaching traffic volumes from north, east, south, and west are Q^N, Q^E, Q^S, and Q^W respectfully. To simplify the model, we assume all of the vehicles arriving at this intersection will choose to go through. No turning movements will be made, as it is shown in Fig. 1b. Here, the cycle length of the traffic light is T_c. The green time of Phase 1 and Phase 2 are T_g^{P1} and T_g^{P2} accordingly. Note that the red time for one phase is the green time for the other phase, thus we can have $T_g^{P1} = T_r^{P2}$ and $T_g^{P2} = T_r^{P1}$. The yellow time is included in the green time in our model, therefore the relationship between the red/green time and the signal cycle length can be denoted as followed:

$$T_c = T_g^{P1} + T_g^{P2} = T_r^{P2} + T_r^{P1} \tag{1}$$

The discrete-time domain for the offloading process can be regarded as a set of time slots from start slot 0 to last slot T. Thus, the whole time domain can be represented as a sequence $\{0, 1, 2, ..., T\}$. When a vehicle i enters the intersection area and connects with the VFC in the time domain, a tuple of vehicle information will be generated. This tuple is denoted as $\left(Pos^i, \vec{s^i}\right)$. Here, Pos^i is the current position of the vehicle i described in x-y coordinates. $\vec{s^i}$ is the current vehicle speed vector. Traffic signal information is represented as a pair $\left(Ph^t, D^t\right)$. Ph^t is the current phase at time slot t, and D^t means how much time has passed after the current phase begins.

C. Task Offloading Model

To model the offloading procedure at the intersection, we first need to acquire a single offloading event. At time t, t $\in \{0, 1, 2, ..., T\}$, a TaV i arrives at the intersection and immediately generates a task, denoted as a tuple $\left(x^t, y^t, w^t, b^t, L^t\right)$ according to the definition by [4] and [6]. Here x^t is the size (in bits) of the task itself. y^t is the size (in bits) of the computing result of this task which generates by SeV and needs to be returned back to TaV. w^t is the computation intensity of the task, which can be representedby the number of CPU cycles that need to finish computing this task. b^t is the budget of this task. The more budget the task gets, the more replications and better SeV the TaV can afford. And L^t is the deadline for this task. The task arrival at the TaV follows the Poisson Distribution with an interval $1/\lambda$. During time slot t, there will be a set of available SeVs at the intersection for TaV to send its task replicas. We denote this set of potential available SeVs as $\mathcal{N}^t$ and the set of SeVs which the TaV actually chooses is $\mathcal{V}^t$. Once such a request is proposed by a TaV, the number of task replications and the number of chosen SeVs should never go beyond the budget limitation. Each individual SeV v in $\mathcal{V}^t$ will receive a task replica tuple $\left(x^t, y^t, w^t, b^t, L^t\right)$ from TaV. For SeVs, we use f_{SeV}^t to describe thecomputing resources that SeVs allocate to the task at time slot t. This $f_{SeV}^{t_1}$ are varied from time to time.

The offloading latency is denoted by d_v^t for offloading to v at time slot t. This latency consists of three parts.

1) Task Uploading Transmission Delay: According to [9], the transmission rate of a task from TaV to a selected SeV v at time t can be denoted as:

$$r_{t,v}^{(up)} = Wlog_2\left(1 + \frac{P^T A_0 (l_{t,v}^{TS})^{-2}}{\sigma^2 + I_{t,v}}\right) \tag{2}$$

Here, W is the channel bandwidth and P^T is the transmission power of the TaV. A_0 is the Line of Sight pass lost coefficient. $l_{t,v}^{TS}$ is the Euclidean distance between TaV and the selected SeV v. σ^2 and $I_{t,v}$ are the channel noise power and interference power respectively. Since the data size of the task is x^t , the task uploading transmission delay can be represented by:

$$d_t^{(up)}(t, v) = \frac{x_t}{r_{t,v}^{(up)}} \tag{3}$$

2) Task Computing Delay: Here we use $f_{t,v}$ to denote the CPU frequency that the SeV v assign to each task it receives. Therefore, the Task Computing Delay can be represented by:

$$d_t^{(comp.)}(t, v) = \frac{x^t w^t}{f_{t,v}} \tag{4}$$

3) Result Downloading Transmission Delay: The result downloading process from selected SeV v to the TaV is very similar to the task uploading process. Like 2, the result downloading transmission rate can be represented by:

$$r_{t",v}^{(down)} = Wlog_2\left(1 + \frac{P^S A_0 (l_{t,v}^{ST})^{-2}}{\sigma^2 + I_{t,v}}\right) \tag{5}$$

Here, $t"$ is the time that SeV is ready to return the result back to the SeV, and $t" = t + d_t^{(up+comp.)}$ where $d_t^{(up+comp.)} = d_t^{(up)} + d_t^{(comp.)}$. $l_{t,v}^{ST}$ is the distance between SeV and TaV. Note that this distance is unknown to TaV before the computing task is finished by targeted SeV. However, this distance is vital for TaV to make a good offloading decision. Thus we propose a distance prediction scheme based on the traffic signal information as well as speed and moving direction of TaV and SeV for estimating $l_{t,v}^{ST}$. The detailed prediction scheme will be discussed in the following subsection. Once TaV acquires the distance $l_{t,v}^{ST}$, the relationship between estimated result downloading transmission rate and downloading delay can be represented as,

$$d_t^{(down)}(t", v) = \frac{y^t}{r_{t",v}^{(down)}} \tag{6}$$

Subsequently, by adding up the waiting time and all three delays, we can get the total task offloading delay between TaV and selected SeV v as

$$d_v^t = d_t^{(up)}(t, v) + d_t^{(comp.)}(t, v) + d_t^{(down)}(t, v) \tag{7}$$

Since there is a deadline for each task offloading process, we consider the offloading process successful when $d_v^t < L^t$. The probability that the offloading process from TaV to selected

SeV v is successful is $p_v^t = Pr\{d_v^t < L^t\}$. Since v is the member of the target SeVs group $\mathcal{V}^t$, as long as one of the SeVs is successful, the task can be considered as successfully offloaded. If all of those offloading attempts can not reach the deadline, the whole task offloading is regarded as failed. Then the probability of successfully achieve at least one task offloading attempts can be denoted as

$$P(\mathcal{V}^t) = 1 - \prod_{v \in \mathcal{V}^t} \left(1 - p_v^t\right) \tag{8}$$

D. Problem Formulation

Here we consider queuing is not allowed in the proposed offloading scenario. Our assumption is that once TaV offloads a task replication to a chosen SeV v_i^t at time period t, this SeV v_i^t will not be available for service, i.e., will not appear in the set $\mathcal{N}^t$, until it finishes the current task. Based on this assumption, we can conduct optimization in a single time slot. This is to say that once a TaV comes to the intersection at time t_1, it will make a selfish offloading decision based on the current potential service vehicle set $\mathcal{N}^t$ without considering potential the TaVs coming after it. Then the problem of maximizing offloading success rate in a particular time period t under the budget constraint b can be represented as follows

$$\textbf{P1:} \quad \max_{v \in \mathcal{V}^t} P(\mathcal{V}^t) = \max_{v \in \mathcal{V}^t} \left[1 - \prod_{v \in \mathcal{V}^t} \left(1 - p_v^t\right)\right] \\ s.t. \quad \mathcal{V}^t \subset \mathcal{N}^t, \\ |\mathcal{V}^t| \leq b^t, \\ |\mathcal{V}^t| \leq |\mathcal{N}^t| \tag{9}$$

To achieve the maximum offloading success rate for the task generates at time slot t, we first calculate the offloading delay d_v^t for all SeV in $\mathcal{S}^t$. Then we use the brutal search to select b^t SeVs with lowest offloading delay and assign these SeVs to the offloading decision set $\mathcal{V}^t$. Meanwhile, total number of the vehicles in the target SeVs group $\mathcal{V}^t$ is strictly constrained by the budget b. If the $|\mathcal{N}^t| < b^t$, the size of offloading decision set $|\mathcal{V}^t|$ will be equal to $|\mathcal{N}^t|$. If there is no SeV available for the TaV to choose at all or TaV and SeV totally lose connection, we then assign a very large offloading delay, say 65535 seconds, to the offloading task and consider this offloading attempt a failure.

E. Distance Prediction Scheme

The most important factor affecting the task uploading and result downloading transmission rate is the distance between SeV and TaV as it is shown in equations 2 and 5. The larger the distance between TaV and SeV the lower the transmission rate it will be. When the TaV tries to select the most appropriate set of SeVs $\mathcal{V}^t$ from $\mathcal{N}^t$ at time point t, computing the equation 2 is straight forward since the TaV will get position information of all SeVs which are currently located at the intersection. However, computing equation 5 may not be as straight forward as it looks, since the result downloading procedure will happen in the future and the position of both SeV and TaV might already be changed. Due to the existence of traffic signal, the future location of a vehicle i can not be simply estimated by the displacement plus the current position like $newPos = d_t^{(up+comp.)} \cdot \vec{s} + oldPos$. Rather, the vehicle i may encounter a full stop because of the red light, or it may stop for a while then continues to move when red light turns green. Thus, when the vehicle is encountering a green light, this vehicle will be predicted as continuing move forward. The vehicle's future movement when encountering a red light depends on the remaining red time. If the vehicle is encountering a red light with longer remaining red time, the vehicle will have a full stop. In addition. If the remaining red time is relatively short, the vehicle will continue to move after red light turns green.

The detail of the vehicle location prediction algorithm is shown in **Algorithm 1**. In this algorithm, $oldPos$ is the position coordinate of the vehicle at this time point. $\vec{s}$ is the average moving speed vector of the vehicle. $isGreen(\vec{s}, t)$ is a function to query whether the vehicle is encountering a green light or not at certain timepoint, and $RedLeft(t)$ is a function to query how much red time is left when vehicle encountering a red light. This location prediction algorithm will be applied to both of the SeV and TaV to calculate the estimated distance between a SeV-TaV pair after the SeV finishing computing the offloading task.

Algorithm 1: Vehcile Location Prediction

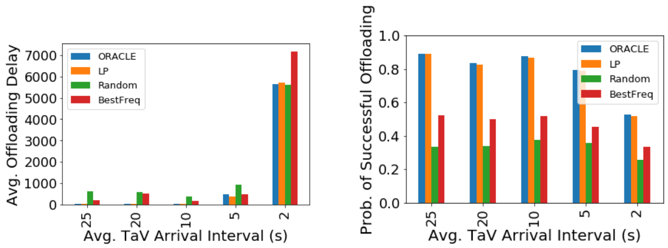
```
Input: oldPos, s⃗, d_t^(up+comp.), t
if Vehicle i is leaving this intersection then
    newPos = d_t^(up+comp.) · s⃗ + oldPos;
else
    if isGreen(t) then
        newPos = d_t^(up+comp.) · s⃗ + oldPos;
    else
        if d_t^(up+comp.) < RedLeft(t) then
            newPos = oldPos;
        else
            newPos =
              (d_t^(up+comp.) − RedLeft(t)) · s⃗ + oldPos;
        end
    end
end
return new-pos;
```

IV. Experiments and Results

In this section, we evaluate our offloading scheme under various task-related and traffic-related configurations against offloading latency and success rate as we previously defined in equations 7 and 8 respectively. The traffic simulation environment SUMO [12] is applied to generate realistic vehicle mobility at an intersection for the evaluation. We also built a customized Python-based time-slotted simulator program for simulating the whole Vehicular Fog task offloading process.

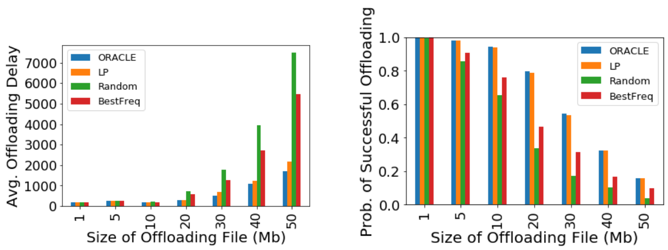
A. Simulation Settings

Our simulation uses one typical four direction intersection with an area of 200 x 200 meters. For simplicity, we assume



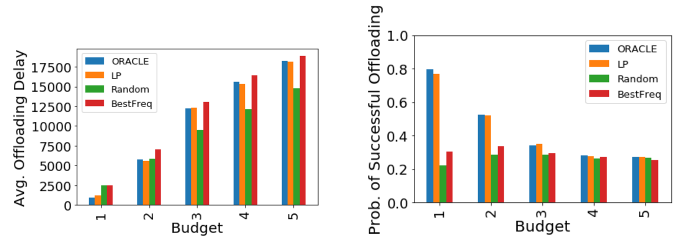
(a) Average Offloading Delay (b) Probability of Success

Fig. 2: Performance Under Different Task Arrival Intervals



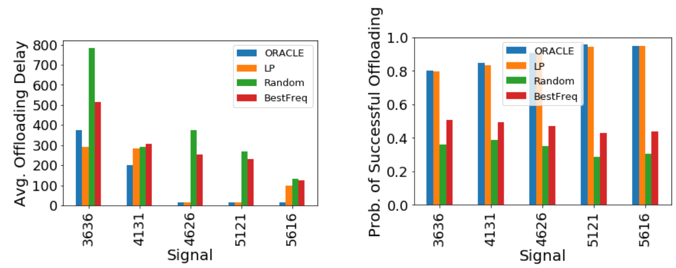
(a) Average Offloading Delay (b) Probability of Success

Fig. 3: Performance Under Different Task and Result Sizes



(a) Average Offloading Delay (b) Probability of Success

Fig. 4: Performance Under Different Budgets



(a) Average Offloading Delay (b) Probability of Success

Fig. 5: Performance Under Different Signal Settings

all four approaches have the same traffic input volume 360 veh/hour. Green time for both North-South direction and East-West direction is 36 seconds, thus the signal cycle length is 72 seconds. As for the TaV and SeV setting, we assume the task arrival interval $1/\lambda$ is 5 seconds. The size of the task x is 20 Mbits and the size of the computing result y is 10 Mbits. The computation intensity of w is 1000 CPU cycles per bit of the given task. Task deadline L is randomly chosen from the range 10 seconds to 25 seconds. The offloading budget of b is 2 copies per task. And the CPU frequency that SeV can allocate to the assigning task is randomly selected from the range of 1GHz to 5GHz. For the communication part, we assume all vehicles, be it a SeV or TaV, have the same wireless transmission power P^T = P^S = 0.1W and the same bandwidth W = 10MHz. The Line of Sight pass lost coefficient A_0 = -17.8dBm. Channel noise power σ^2 is 10^{-13}W and the average interference I is $6 * 10^{-9}$W. Our proposed offloading scheme is compared with three benchmarks as follows:

(1) Oracle: The Oracle has the knowledge about the exact position of a select TaV-SeV offloading pair at any given time. So when it makes the offloading decision, it can always find a set of the SeVs with minimum offloading delay.

(2) Random: The random offloading scheme will randomly select SeVs which currently located at the intersection to offload the task.

(3) Best Frequency: Best Frequency offloading scheme will choose SeVs with the best CPU frequency without considering the mobility pattern of SeVs and TaV.

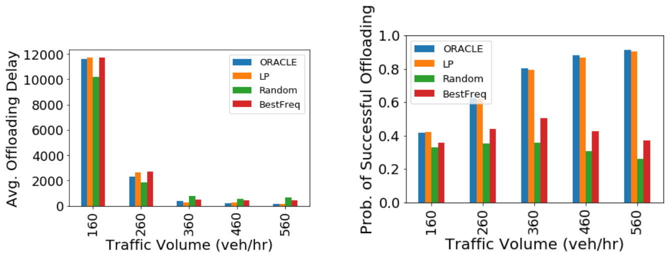
The Performance is evaluated in terms of the offloading delay and success ratio, following the related work [5] [6].

B. Performance Evaluation

We first compare the performance of our offloading scheme (noted as LP in the figures) with other benchmarks under different task arrival interval in Fig. 2. The shorter the interval is, the more frequent the task will arrive at the intersection. This creates a problem of TaV running out of available SeVs, since the previous tasks are still executed in the SeVs when the next several tasks arrive. The figure shows when the task arrival interval decrease, the average offloading delay will increase and the offloading success rate will deteriorate. Our proposed offloading scheme performs nearly identical with Oracle scheme and outperforms both Random and Best Frequency scheme.

Next, we investigate the influence of different task input size and result output size. In our experiment, we assume the size of the computing result is half of the size of the input task. The sizes of the task and result have a direct influence on the offloading delay. Thus, in Fig. 3, the larger the size of the task and result, the more delay and less success rate it will become. Still, our offloading scheme can maintain satisfactory performance and behave better than the Random and Best frequency scheme.

Then we conduct the experiment to test the impact of the budget on offloading. According to equation 9, it seems that if a task has an abundant budget, more replicas can be made thus the offloading will have a lower failure probability. However, our simulation results presented in Fig. 4 appear to contradict this assumption. The higher budget actually leads to heavier offloading delay and less success rate. This is because the higher budget will result in larger computation overhead and occupied more available SeVs and leave the following TaVs with less SeVs to choose. As a consequence, the offloading



(a) Average Offloading Delay (b) Probability of Success

Fig. 6: Performance Under Different Traffic Volumes

success rate will be dragged down by this overhead introduced by a redundant budget.

In Fig. 5, we present the results of the offloading performance under different traffic signal settings. Here we keep the cycle length the same at 72 seconds and changing the green time assign to the two phases. We start from evenly spilled the time cycle as 36 seconds for each phase and gradually increase the green time for phase 1 and reduce the green time for phase two. Changing the signal plan doesn't seem to have much impact on offloading the success rate of Random and Best frequency scheme, but it does slightly improve the offloading success rate of the Oracle and Location Prediction Scheme. When cycle length is fixed and all four approaches have the traffic input volume, evenly split the signal will improve the traffic flow movement for all directions. But this also means there will be less full-stop SeVs and TaVs at intersections. If the green time of a certain approach is reduced, the overall time that vehicles spend on waiting for red light will increase and introduce more stopped vehicles to the intersections. This situation is more favorable to Oracle and the location prediction offloading scheme since both of them cantake advantage of the stopped vehicles.

Last, we test our offloading schemes with different traffic volumes. Intuitively, more traffic volume at the intersection means the more potential SeV options for TaV to choose, and the higher chance that TaV will find a set of SeVs with lower delay and higher offloading success rate. The simulation result in Fig. 6 confirms this understanding.

V. Conclusion

In this paper, we studied the deadline-aware offloading problem in vehicular fog environment with a focus on the signalized intersection scenario. The deadlines and the cost of duplicate service are the main constraints. Predicting the unique mobility patterns generated by vehicles at the urban intersection is the major contributing factor in solving the problem. Our proposed offloading scheme can help TaV to make a decent offloading decision and achieve a satisfactory offloading success rate with adequate offloading delay.

References

[1] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat and R. W. Heath, ”Millimeter-Wave Vehicular Communication to Support Massive Automotive Sensing,” in IEEE Communications Magazine, vol. 54, no. 12, pp. 160-167, December 2016.

[2] X. Hou et al., ”Vehicular fog computing: A viewpoint of vehicles as the infrastructures”, IEEE Trans. Veh. Technol., vol. 65, no. 6, pp. 3860-3873, Jun. 2016.

[3] 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. 3644, 2017.

[4] Lixing Chen, Jie Xu, Task Replication for Vehicular Cloud: Contextual Combinatorial Bandit with Delayed Feedback, IEEE International Conference on Computer Communications. (IEEE INFOCOM), Paris, France, May 2019.

[5] Zhiyuan Jiang ; Sheng Zhou ; Xueying Guo ; Zhisheng Niu , ”Task Replication for Deadline-Constrained Vehicular Cloud Computing: Optimal Policy, Performance Analysis, and Implications on Road Traffic,” IEEE Internet of Things Journal, vol. 5, no. 1, pp. 93-107, 2018.

[6] Y. Sun, J. Song, S. Zhou, X. Guo, and Z. Niu, Task replication for vehicular edge computing: A combinatorial multi-armed bandit based approach IEEE Global Commun. Conf. (GLOBECOM), Abu Dhabi, UAE, Dec. 2018.

[7] Xiaoyan Hong, Yingyan Lou, Meng Kuai, Shuoping Wang, Quantify Self-Organized Storage Capacity in Supporting Infrastructure-less Transportation Operation, ACM MobiHoc 2015 workshops - The Second Workshop for Mobile Sensing, Computing and Communication (MSCC'15), June 22-25, 2015, Hangzhou, China.

[8] Meng Kuai, Pawan Subedi, Xiaoyan Hong, Alexander Hainen, ”Sustain Vehicle-Crowds via Traffic Signal Adjustments” in the proceedings of 2018 IEEE 88th Vehicular Technology Conference (VTC2018-Fall), 2730 August 2018, Chicago, USA.

[9] Mouhamed Abdulla, Erik Steinmetz, Henk Wymeersch, ”Vehicle-to-Vehicle Communications with Urban Intersection Path Loss Models” 2016 IEEE Globecom Workshops (GC Wkshps), 4-8 Dec. 2016, Washington, DC, USA.

[10] Deo, Nachiket, and Mohan M. Trivedi. ”Convolutional social pooling for vehicle trajectory prediction.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.

[11] N. Deo and M. M. Trivedi, ”Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs,” 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, 2018, pp. 1179-1184.

[12] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent Development and Applications of SUMO - Simulation of Urban MObility. International Journal On Advances in Systems and Measurements, 5 (3&4):128-138, December 2012.