Exploiting Moving Intelligence: Delay-Optimized Computation Offloading in Vehicular Fog Networks

Sheng Zhou, Yuxuan Sun, Zhiyuan Jiang, and Zhisheng Niu

Abstract

Future vehicles will have rich computing resources to support autonomous driving and be connected by wireless technologies. Vehicular fog networks (VeFNs) have thus emerged to enable computing resource sharing via computation task offloading, providing a wide range of fog applications. However, the high mobility of vehicles makes it hard to guarantee the delay that accounts for both communication and computation throughout the whole task offloading procedure. In this article, we first review the state of the art of task offloading in VeFNs, and argue that mobility is not only an obstacle for timely computing in VeFNs, but can also benefit the delay performance. We then identify machine learning and coded computing as key enabling technologies to address and exploit mobility in VeFNs. Case studies are provided to illustrate how to adapt learning algorithms to suit the dynamic environment in VeFNs, and how to exploit the mobility with opportunistic computation offloading and task replication.

Introduction

To satisfy the emerging need for autonomous driving, future vehicles will not only have rich onboard sensors like cameras and radars, but also be equipped with strong computing power to process the sensing data and make driving decisions. In addition, due to recent fatal accidents with standalone autonomous driving, it becomes evident that safe and reliable autonomous driving requires effective interaction and collaboration between vehicles, and between vehicles and roadside units (RSUs). Wireless technologies enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications that deliver critical information, such as safety warnings and road conditions. Accordingly, vehicles can extend their sensing capability to reach blind spots, and can also jointly process the sensing data and coordinate their driving decisions. The result can be precise recognition of the environment and robust control of vehicles, leading to safer autonomous driving and more efficient road traffic.

The large number of connected vehicles, each endowed with server-level computing power, form a network with an abundant amount of moving intelligence. Vehicles can contribute their computing resources, acting like fog nodes in the context of fog computing [1], and thus the whole network can be regarded as a vehicular fog network (VeFN). The VeFN can provide a wide range of applications beyond autonomous driving. For instance, passengers can utilize the excessive computing power on their own vehicle or neighboring vehicles for computation task offloading, overcoming the device limitations as in mobile edge computing (MEC) where computing resources are co-located with base stations (BSs) [2]. Pedestrians can also access the VeFN via RSUs. To this end, the VeFN combines the concepts of fog as a service [3] and vehicle as infrastructure [1], and is promising in the era of artificial intelligence (AI), which calls for computing anytime and everywhere.

For autonomous driving and other computation offloading applications, *delay*, accounting for both computing and transmission, is always the most demanding quality of service (QoS) requirement. In vehicular networks, the high mobility of vehicles and the ad hoc nature of networking make timely communication and computing quite challenging. Despite the existing research efforts on delay optimized ultra reliable V2V communications [4], the coupled communication and computing delays in task offloading are affected by more random factors, which is challenging to optimize. High mobility introduces difficulties in jointly adapting wireless and computing resources with respect to the time varying system conditions. Moreover, the adaptation requires fresh system state information of channels and computing power, which is unfortunately hard to obtain.

Nevertheless, mobility is not always an obstacle. As shown in the premier work by Grossglauser and Tse, mobility can increase the capacity of wireless ad hoc networks by increasing the probability of making contacts between nodes [5]. Moreover as shown in [6, 7], mobility can increase the successful downloading probability of files in caching systems, with more chances for end users to experience good channels and file holders. We believe that mobility is also beneficial to the task offloading in the VeFN. For example, the probability that vehicles with excessive computing resources appear in the vicinity of an end user can increase with the speed of vehicles [8]. In this context, how to guarantee

The authors review the state of the art of task offloading in VeFNs, and argue that mobility is not only an obstacle for timely computing in VeFNs, but can also benefit the delay performance. They identify machine learning and coded computing as key enabling technologies to address and exploit mobility in VeFNs.

Sheng Zhou, Yuxuan Sun, and Zhisheng Niu are with Tsinghua University; Zhiyuan Jiang (corresponding author) is with Shanghai University.

Digital Object Identifier:
10.1109/MCOM.2019.1800230

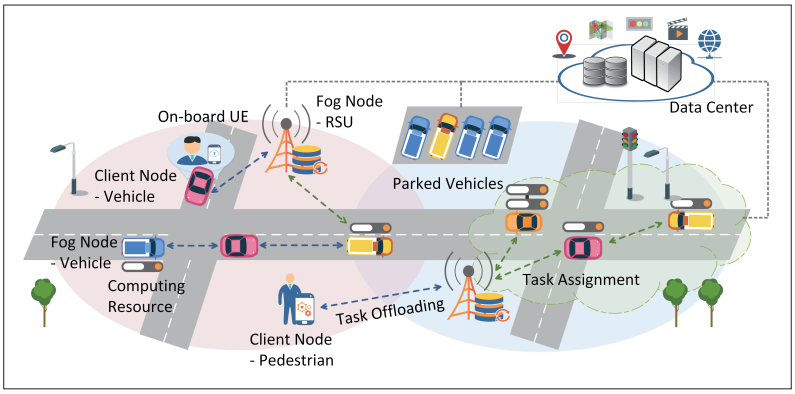

Figure 1. Illustration of task offloading in the VeFN.

Application	Type	Delay requirements
Cooperative collision avoidance	Safety	Delay bound, 10 ms
Vehicle platooning	Safety	Delay bound, 25 ms
Collective perception of environment	Safety	Delay bound, 500 ms
Vehicle scheduling	Non-safety	Average delay, 1 s
Virtual reality and augmented reality	Entertainment	Delay bound, 10 ms
Cloud gaming	Entertainment	Average delay, 100 ms~1 s
Road monitoring and flow optimization	IoT	Average delay, seconds~minutes

Table 1. Typical applications in VeFNs and the corresponding delay requirements.

offloading delay while at the same time exploiting the *diversity* brought by mobility becomes an intriguing research issue.

In this article, we first introduce the VeFN concept with the corresponding delay requirement, and review the state of the art for cloud computing under dynamic conditions in vehicular networks. We propose the VeFN architecture with three major offloading modes in the next section, and analyze the advantages and challenges of these modes. We then discuss why mobility is both a foe and a friend of computation offloading in the VeFN, and propose two key solutions, that is, learning while offloading and coded computing, to optimize the offloading delay. We carry out two case studies that address and exploit the mobility in VeFNs, respectively. In the first case study, the existing learning algorithms are revisited and modified so that they can adapt to the varying network topologies and workloads. Coded computing is combined with learning techniques to further improve the service reliability. In the second case study, the optimal task replication policy is derived, providing insights that balanced assignment optimizes the delay performance. Finally we conclude the article with an outlook on future research directions.

Vehicular Fog Networks: Architecture and State of the Art

The VeFN integrates the computing resources of vehicles and RSUs, and provides diverse fog computing services and applications for vehicles and mobile users. As shown in Fig. 1, RSUs and vehicles (moving and parked), which can provide computing resources, are regarded as fog nodes. Computation tasks with different workloads and delay requirements are generated by the client nodes, including vehicles requiring excessive computing support, onboard user equipments (UEs), pedestrians, and so on. These tasks are offloaded from the client nodes to the VeFN for processing. Note that each vehicle can act as either a fog node or a client node, denoted by *fog vehicle* and *client vehicle*, respectively. The role of each vehicle can change over time, depending on whether it has surplus computing resources to contribute to the network, or whether it requires support from other nodes for task offloading.

Some typical applications in VeFNs and their corresponding delay requirements are summarized in Table 1, where the data is from [2] and Third Generation Partnership Project (3GPP) TR 22.886 [9]. The key performance metric of task offloading is *delay*, consisting of three parts [2]: uploading delay related to the input data size of a task that needs to be processed, computing delay at the fog node, which is related to the computational complexity and the input data size, and downloading delay related to the output data size. All these delays are affected by the communication bandwidth used to transmit the data, and the computing power to process the task at the fog nodes. For delay-critical applications, the hard delay bound represents the longest allowable delay that cannot be violated. For delay-tolerant applications, tasks do not have an exact deadline, but timely feedback is still favorable. Such applications include traffic flow optimization, entertainment, and Internet of Things (IoT) applications, and average offloading delay can be used as the performance metric.

Architecture and Offloading Modes in VeFNs

The scattered computing resources in the VeFN bring a variety of offloading routes to the VeFN. To support data transmission between client nodes and fog nodes, multiple communication techniques are jointly used, including IEEE 802.11p-based dedicated short-range communications (DSRC) and LTE-V, which enable vehicle-to-everything (V2X) communications such as V2V, V2I, and vehicle-to-pedestrian (V2P) communications. Pedestrians get access to the RSUs through 3G or 4G LTE, and onboard UEs can offload tasks to the vehicle on which they ride via Bluetooth. As shown in Fig. 1, the computation task offloading in the VeFN is classified into three major modes.

Vehicle-Vehicle Offloading: Vehicles can directly offload their tasks (including the tasks offloaded by their passenger UEs) to neighboring fog vehicles. In this case, each client vehicle first discovers the available fog vehicles in its communication range. To keep a relatively long contact time, the moving directions and velocities should be considered, which can be acquired by V2X communication protocols. Multiple fog vehicles may be available at the same time. Offloading decisions about which fog vehicles to select are made by client vehicles independently in a distributed manner, since it is difficult to acquire global information about the vicinity, and there might not be a centralized entity to make such decisions.

Vehicle-RSU-Vehicle Offloading: When the surrounding fog vehicles cannot satisfy the computing needs of client vehicles, tasks are offloaded to nearby RSUs. RSUs may compute the tasks by their own computing resources, or further assign the tasks to other fog vehicles without direct wireless connection to the client vehicles. RSUs are able to master more information about communication bandwidth and computing resources. Hence, centralized task assignment can optimize the utilization of computing resources and the QoS. Computing results are finally transmitted back to the client vehicle via its associated RSU.

Pedestrian-RSU-Vehicle Offloading: The contact durations of pedestrians and vehicles are often very short, and the connectivities are quite unstable. Therefore, in VeFNs, RSUs can first collect the computation tasks from pedestrians. Then the tasks are handled by themselves or offloaded to fog vehicles. The computing results can be fed back by a fog vehicle to its nearby RSU, and then delivered back to the original RSU via the backhaul if the fog vehicle has already moved away from the original RSU.

State of the Art on Computing in Vehicular Networks

There are some recent efforts focusing on the resource management of communication and computation in the context of VeFNs. Researchers start by evaluating the feasibility of employing vehicles as fog nodes. Hou *et al.* [1] analyzed communication and computing capacity of vehicles using real traces of vehicles in Beijing and Shanghai. Simulation results show that the communication and computing resources of both parked and moving vehicles have great potential to enhance the static fog computing network.

For flexible computing resource management, a software-defined vehicular network architecture was proposed by Choo *et al.* [10] in which a centralized vehicular cloud (VC) controller periodically collects the mobility and resource status of fog vehicles, estimates their instantaneous locations and computation loads upon task requests, and allocates the computing and bandwidth resources for each task. In terms of resource allocation schemes, Zheng *et al.* [11] considered a dynamic VC consisting of moving vehicles. The arrival and departure of vehicles follow the Poisson process, and each vehicle is equipped with equal computing power. Tasks are collected by a central controller, and assigned to fog vehicles to maximize the average utility related to delay, energy consumption, and resource occupation. However, the basis of adopting centralized schemes is holding the instantaneous state information of the whole system, which may lead to high signaling overhead, and is thus hard to implement in real systems.

Task offloading decisions can also be made by each client node independently in a distributed manner, corresponding to the vehicle-vehicle offloading mode. Feng *et al.* [12] proposed a distributed VeFN architecture where RSUs and vehicles can offload tasks to their neighboring nodes based on their distributed decisions. They designed a task offloading algorithm based on ant colony optimization in order to maximize the sum utility of offloaded tasks related to delay, and evaluated it in a system-level simulator using real traces. However, the RSUs and vehicles are treated equally in [12], while in real systems, RSUs often know more about the network conditions, which should be more effectively exploited.

Mobility: Foe and Friend for the Offloading Delay

The *mobility* of vehicles makes the VeFN highly dynamic and volatile, which brings both challenges and opportunities for computation task offloading. In this section, we interpret the intuition on why mobility acts as both a foe and a friend for timely computing, and identify potential ways to address and exploit the mobility.

Mobility as a Foe

Mobility brings more randomness and uncertainties to the delay performance of the offloaded tasks. First, the VeFN can be viewed as an intermittently connected wireless network, in the sense that the network topology changes over time, and connection durations of V2X, including V2V, V2I, and V2P, are quite limited. This inherently stems from the physical limitations of the range of wireless communications and the high mobility of vehicles, and significantly limits the effectiveness of a VeFN. Second, the wireless channel states and thus the interference between V2X vary fast across time, depending on many factors such as relative speed, neighboring vehicles' transmit power, and surrounding scatters, and they are hard to model or predict. Third, similar to the MEC systems, the computation tasks are generated randomly with different delay requirements and workloads, and the computing power of RSUs and vehicles varies, producing highly dynamic and non-uniform computation loads. These factors bring challenges to collect information and to make optimal offloading decisions and resource allocation in a timely manner, which is critical for those safety-related applications with hard delay bounds.

Mobility as a Friend

However, the limitation due to intermittent connectivity, somewhat surprisingly, can be overcome by vehicle mobility. This can be illustrated by the following three factors.

Mobility Increases the Probability of Making Contact: As shown by several existing works such as [5], the mobility of nodes in an intermittently connected network can be beneficial since mobility creates more chances of contacts between nodes, and thus the probability of communication and task offloading between the nodes also increases.

Mobility Decreases Contact Time Interval: Many VeFN applications rely on consecutive contacts between V2V and V2I, such that the inputs and outputs of tasks can be communicated separately. Therefore, the offloading delay is related to the contact time interval significantly. In this regard, vehicle mobility decreases the time interval and hence reduces the offloading delay.

Mobility brings more randomness and uncertainties to the delay performance of the offloaded tasks. However, the limitation due to intermittent connectivity, somewhat surprisingly, can be overcome by the vehicle mobility.

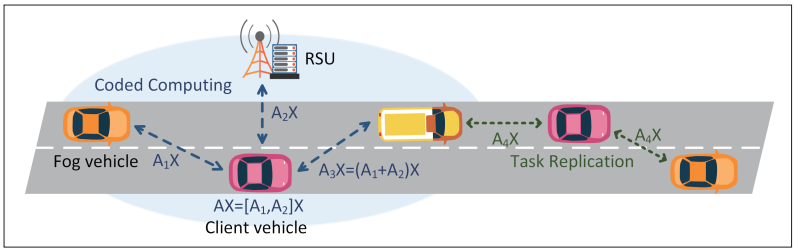

Figure 2. Illustration of coded computing and task replication in a VeFN.

Predictable Mobility: Despite the high mobility of vehicles, their trajectories are limited to roads, and their speeds are highly related to the traffic conditions. Thus, one can predict the mobility of vehicles to a certain extent and carry out prediction-based task offloading.

Address and Exploit Mobility

To release the aforementioned potentials brought by mobility while reducing the time overhead for information collection and online decisions, we resort to machine learning approaches to track the system dynamics. Moreover, we identify a set of solutions that falls into the concept of coded computing, which can enhance the reliability of computing with efficient resource utilization.

Learning While Offloading: Because both communication and computation environments depend on many factors and vary fast in VeFNs, the offloading delay is very complex to model and predict, especially for distributed task offloading. Instead of acquiring all the related state information to infer the offloading delay of candidate fog nodes before making the offloading decision, client nodes can try different candidates by offloading several tasks and observing the delay on the go. In other words, the environment is learned while tasks are being offloaded. Such learning algorithms should have low complexity and fast convergence to track the dynamic environment. They should also effectively balance the so-called *exploration-exploitation trade-off*: to explore more and get more accurate estimations about candidate fog vehicles, or to select the empirically best fog vehicle, hoping to minimize the instantaneous offloading delay. In the context of learning, the objective is to minimize the regret, that is, the performance loss of learning algorithms compared to the genie-aided optimal solution.

Multi-armed bandit (MAB) [13] is a promising method to perform learning on the go. In classical MAB, a decision maker faces a fixed number of candidate actions, whose rewards are governed by different distributions that are unknown a priori. The decision maker tries one action at a time, observes the reward, and gradually learns the performance of different candidates while minimizing the regret.

However, existing MAB algorithms cannot be applied in VeFNs directly because the network topologies are not fixed, with neighboring fog vehicles coming and leaving unexpectedly. In addition, the workloads of tasks vary over time, but such variations have not been considered in existing MAB problems. We revise the MAB-based learning algorithm to address the dynamic topology and task workloads in V2V offloading, which is illustrated in the first case study.

To exploit the mobility, supervised learning methods can also be adopted to learn the mobility of vehicles and predict their speeds and trajectories. Then both computing and bandwidth resources of fog vehicles can be better allocated. For example, vehicles can predict which neighboring vehicles may have longer contact duration, and RSUs can forecast the occurrence of handover and proactively fetch the computing data or migrate some computing services of client vehicles.

Coded Computing: As a foe, mobility makes the computing services at each fog node unreli able. But as a friend, mobility also brings opportunities for client nodes to meet more fog nodes. Equivalently, the computing resources of the VeFN become richer. To exchange the redundancy of computing resources for reliability, *coded computing* serves as an efficient tool. Consider an (n, m) maximum distance separable (MDS) coding scheme. If n fog nodes can provide computing services for a client node, a task can be decomposed into m subtasks with $m \le n$ using the minimum latency coding technique [14], encoded into n coded tasks, and then offloaded to these n fog nodes. Once the earliest m computing results are successfully delivered back, the task is completed. An example of coded computing in a VeFN is shown in Fig. 2. Three fog nodes (two fog vehicles and one RSU) can serve the client vehicle. The input data of a matrix multiplication task is decomposed as two submatrices, and encoded to be three subtasks. The original task is successfully computed if two of the three fog nodes complete their subtasks. As a result, fog nodes can cooperate with each other to better utilize their computing resources, and the uncertainties brought by mobility, such as intermittent connectivity, can be addressed. Coded computing can also be applied to share the computing resources of fog nodes for *multiple* client vehicles.

Coded computing can actually cover a large variety of mappings between computing resources and tasks, among which a special case is *task replication*, using the simplest repetition coding. Each task is directly offloaded to multiple fog nodes simultaneously and processed independently. If one of the selected fog nodes completes the task before the deadline, it is successfully executed. It is crucial to balance the reliability gain with more replications and the resource occupation alongside, and the offloading opportunities require nontrivial allocation among multiple clients. Our key contribution is that we derive the optimal task replication decisions for multiple clients in a VeFN in the second case study, providing the insight that balanced task assignment optimizes the delay performance. We also discuss how to combine MAB and coded computing in the first case study, and show the performance gain through coding.

Case Studies

In this section, we carry out two case studies that apply the aforementioned learning and coded computing methods, proving their great potential for timely computing in VeFN.

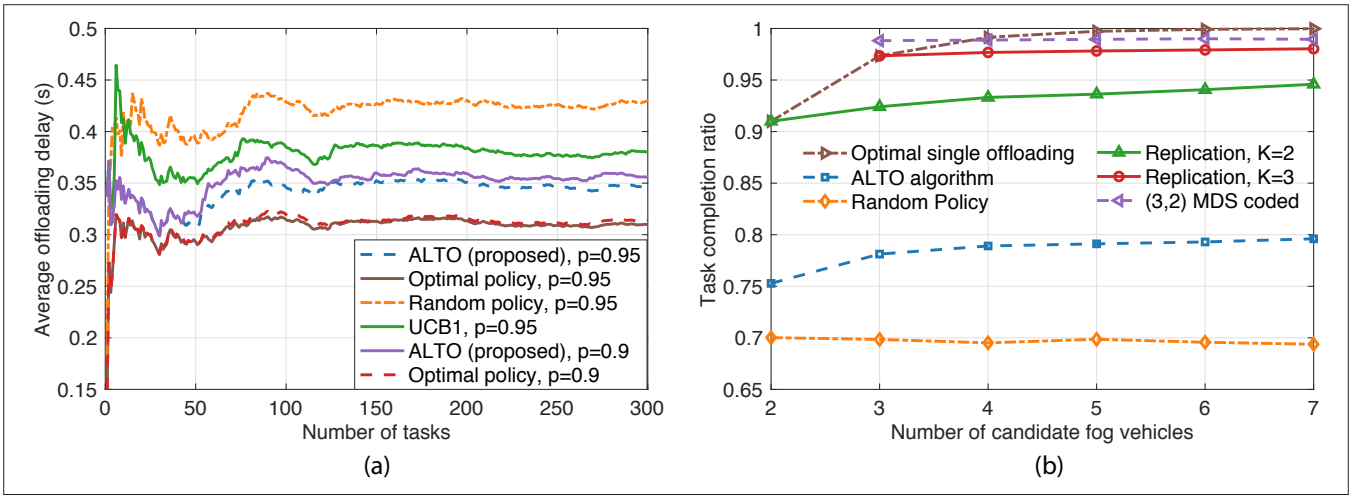

Figure 3. Delay performance of MAB-based task offloading algorithms: a) average delay of ALTO algorithm; b) task completion ratio.

Learning-Based Task Offloading in VeFN

To guide the task offloading in VeFN by using learning-based methods, we first focus on a distributed scenario with V2V offloading. Consider a client vehicle of interest who generates computation tasks in sequence, and it makes the offloading decisions on which fog vehicle to handle each task to minimize the average offloading delay.

It is difficult for each client vehicle to acquire the information about available computing resources and channel states for its own tasks; thus, it has no idea which fog vehicle performs best when making task offloading decisions. It has to learn the average delay of each candidate fog vehicle based on observations of the offloading delay associated with each candidate vehicle.

As mentioned above, an MAB-based learning method can be used to design the task offloading algorithm, but it still requires adaptations to fit the dynamic environment in VeFNs. To make MAB effective in VeFNs, we redesign the utility function of conventional MAB by considering the following three key factors:

- The empirical delay of the offloaded tasks, which is the delay performance of the candidate fog vehicles one has learned.
- The appearance time of each fog vehicle. The client vehicle should focus more on newly appearing fog vehicles, while exploiting what it has already learned about the existing fog vehicles.
- The workload of each task. Since the offloading delay is proportional to the workload, intuitively the client vehicle should try to exploit different fog vehicles when the workload is low so that the regret due to learning can be reduced, and vice versa.

We propose an Adaptive Learning-Based Task Offloading (ALTO) algorithm, proving that the complexity of the ALTO algorithm is linear with the number of candidate fog vehicles, and the regret grows sublinearly over time [15].

For simulations, we download the map of a 12 km stretch of the G6 Highway in Beijing from Open Street Map (OSM) and generate the traffic by Simulation of Urban Mobility (SUMO). Fog vehicles are equipped with heterogeneous computing power, with CPU frequencies in the range of [2, 5] GHz. The input data size of each task is uniformly distributed in [0.2, 1] Mbits. We assume that tasks are of equal computation intensity 1000 cycles/bit, and the size of the output data is neglected. The wireless connectivity is intermittent, with the probability of successful transmission p = 0.9 or 0.95.

As shown in Fig. 3a, the proposed ALTO algorithm is compared to three baselines: *UCB1* is the conventional MAB-based learning algorithm [13]. *Random Policy* is a naive policy in which the client node randomly selects a fog vehicle for each task. *Optimal Policy* is a genie-aided one and always selects the best fog vehicle with minimum offloading delay. Since fog vehicles may appear as candidates or leave, the average delay fluctuates over time. According to the simulations, UCB1 does not work well with moving vehicles and time-varying workloads, while our proposed ALTO algorithm performs better in terms of average offloading delay. This highlights the importance of revisiting machine learning methods to deal with the dynamics in VeFNs.

To further improve the QoS, we integrate learning with task replication and (3,2) MDS coding. Still, the global state information is unknown and needs to be learned, and intermittent connectivity is considered with p = 0.9. Figure 3b observes the ratio of tasks that are completed before a deadline 0.55s, and K is the number of replications. The optimal single offloading policy is the same as the Optimal Policy in Fig. 3a. It is shown that compared to the ALTO algorithm with single offloading, the service reliability is substantially improved through task replication, while light replication with K = 2 or K = 3 provides most of the gains. Meanwhile, the task completion ratio of MDS coding reaches over 98 percent with a small number of fog vehicles, and even outperforms the optimal single offloading policy. This is because MDS coding reduces the workload of each coded subtask, and can further exploit the computing resources of multiple fog vehicles.

Delay-Constrained Task Replication Exploiting Vehicle Mobility

In this case study, we focus on the vehicle/pedestrian-RSU-vehicle offloading mode with task replication. Each RSU collects tasks from pedestrians or vehicles, and then assigns them to the fog vehi-

Conventional encryption and authentication schemes may be too slow to perform in VeFN with high dynamics, especially for delay-critical applications. Guaranteeing security and privacy in task offloading call for novel designs, and may trigger a new dimension to understand mobility.

cles coming into its coverage. Tasks have hard delay bound that cannot be violated. Multiple tasks collected by each RSU waiting to be executed form a task queue. Each fog vehicle is assigned one task at a time, and task replication is used, that is, each task can be assigned to multiple fog vehicles and executed independently.

Our objective is to minimize the deadline violation ratio of tasks by deciding which task should be allocated to which fog vehicle. Assume that the arrival of fog vehicles follows a Poisson process, and the sojourn time (the duration of task assignment and result feedback) of each task at each fog vehicle follows exponential distribution with a homogeneous exponent. This enables a finite horizon Markov decision process (MDP) formulation of the problem, and we derive the optimal policy called balanced task assignment (BETA) [8]. The main intuition of BETA is that unfinished tasks with the fewest offloading replications should be scheduled first when a new fog vehicle arrives, and this *balanced* allocation of computing resources is optimal and avoids unnecessary service waste.

We further investigate how mobility affects the computing performance. Under a linear speed-density relationship widely used in vehicle traffic theory, the optimal vehicle speed that maximizes the traffic throughput (the number of vehicles that pass through the road) is $V_{max}/2$, where V_{max} is the maximum allowed speed on the road. Generalizations to a nonlinear speed-density relationship can be found in [8], while the linear model is sufficient to capture the essence. In the a VeFN system, the optimal vehicle speed which minimizes the deadline violation ratio is proved to be $2V_{max}/3$, meaning that when vehicles move faster within $2V_{max}/3$, the reliability of computing services increases [8]. Note that these results are from a statistical point of view, that is, the optimal speeds are averaged among all vehicles on the road. As shown in Fig. 4, when the vehicle speed increases, the deadline violation probability first decreases and then goes up. This is mainly because the mobility brings more opportunities for the RSUs to meet fog vehicles, and thus the reliability first increases. However, as the vehicle speed rises, the density of vehicles finally becomes too low to support the task requirements.

Conclusion and Outlook

In this article, we have presented the VeFN concept with the latest literature review, and discussed the role of mobility for timely computing in VeFNs as a foe and as a friend. Enabling technologies to address and exploit mobility, including machine learning and coded computing, are introduced, and their initial adoptions in VeFNs are illustrated through two case studies. While notable gains in terms of lower average delay and lower delay-bound violation probability are proved via a MAB-based learning scheme and task replication, more efforts are needed to truly realize the potentials of VeFN accompanied by mobility.

First, computation task partition plays an important role in task offloading, but it has been rarely covered. Considering the heterogeneity of computing resources, task partition helps to optimize the utilization of resources and balance workloads.

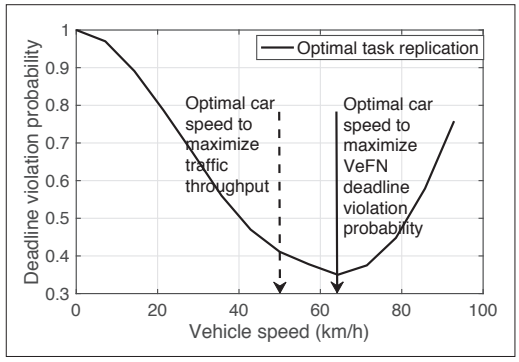

Figure 4. Task replication and corresponding performance analysis for the deadline violation probability in a VeFN.

Second, mobility prediction, either model-based or reinforcement-learning-based, can be exploited to reduce offloading delay by proactive resource provisioning or computation pre-fetching. Accordingly, coded computing over tasks generated at different times is also worth investigating.

Last but not least, conventional encryption and authentication schemes may be too slow to perform in VeFNs with high dynamics, especially for delay-critical applications. Guaranteeing security and privacy in task offloading calls for novel designs, and may trigger a new dimension to understand the mobility.

Acknowledgment

This work is sponsored in part by the National Key R&D Program of China 2018YFB0105005, the Nature Science Foundation of China (No. 61871254, No. 91638204, No. 61571265, No. 61861136003, No. 61621091), and the Intel Collaborative Research Institute for Intelligent and Automated Connected Vehicles.

References

[1] X. Hou *et al.*, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," *IEEE Trans. Vehic. Tech.*, vol. 65, June 2016, pp. 3860–73.
[2] Y. Mao *et al.*, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, 4th qtr. 2017, pp. 2322–58.
[3] N. Chen *et al.*, "Fog as a Service Technology," *IEEE Commun. Mag.*, vol. 56, no. 11, Nov. 2018, pp. 95–101.
[4] M. I. Ashraf *et al.*, "Towards Low-Latency and Ultra-Reliable Vehicle-to-Vehicle Communication," *Euro. Conf. Net. Commun.*, Oulu, Finland, 2017.
[5] M. Grossglauser and D. N. C. Tse, "Mobility Increases the Capacity of Ad Hoc Wireless Network," *IEEE/ACM Trans. Net.*, vol. 10, no. 4, Mar. 2002, pp. 477–86.
[6] S. Krishnan and H. S. Dhillon, "Effect of User Mobility on the Performance of Device-to-Device Networks with Distributed Caching," *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, Apr. 2017, pp. 194–97.
[7] S. Krishnan, M. Afshang, and H. S. Dhillon, "Effect of Retransmissions on Optimal Caching in Cache-Enabled Small Cell Networks," *IEEE Trans. Vehic. Tech.*, vol. 66, no. 12, Dec. 2017, pp. 11,383–87.
[8] 3GPP TR 22.886, "Study on Enhancement of 3GPP Support for 5G V2X Services," V15.1.0, Mar. 2017.
[9] Z. Jiang *et al.*, "Task Replication for Deadline-Constrained Vehicular Cloud Computing: Optimal Policy, Performance Analysis and Implications on Road Traffic," *IEEE Internet of Things J.*, vol. 5, no. 17, Feb. 2018, pp. 93–10.
[10] J. S. Choo *et al.*, "The Software-Defined Vehicular Cloud: A New Level of Sharing the Road," *IEEE Vehic. Tech. Mag.*, vol. 12, no. 2, June 2017, pp. 78–88.
[11] K. Zheng *et al.*, "An SMDP-Based Resource Allocation in Vehicular Cloud Computing Systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, Dec. 2015, pp. 7920–28.

[12] J. Feng *et al.*, "AVE: Autonomous Vehicular Edge Computing Framework with ACO-Based Scheduling," *IEEE Trans. Vehic. Tech.*, vol. 66, no. 12, Dec. 2017, pp. 10660–75.
[13] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-Time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, vol. 47, no. 2–3, May 2002, pp. 235–56.
[14] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coding for Distributed Fog Computing," *IEEE Commun. Mag.*, vol. 55, no. 4, Apr. 2017, pp. 34–40.
[15] Y. Sun *et al.*, "Learning-Based Task Offloading for Vehicular Cloud Computing Systems," *IEEE ICC*, Kansas City, MO, May 2018.

Biographies

Sheng Zhou [S'06, M'12] is an associate professor in the Electronic Engineering Department, Tsinghua University, China. He received his B.S. and Ph.D. degrees in electronic engineering from Tsinghua University in 2005 and 2011, respectively. His research interests include cross-layer design for multiple-antenna systems, mobile edge computing, and green wireless communications.

Yuxuan Sun [S'18] received her B.S. degree in telecommunications engineering from Tianjin University, China, in 2015. She is currently pursuing a Ph.D. degree in electronic engineering at Tsinghua University. Her research interests include mobile edge computing and vehicular cloud computing.

Zhiyuan Jiang [S'12, M'15] received his B.E. and Ph.D. degrees from the Electronic Engineering Department of Tsinghua University in 2010 and 2015, respectively. He is currently an associate professor with the School of Communication and Information Engineering, Shanghai University. His main research interests include sequential decision making in wireless networks, and design and implementation of massive MIMO systems.

Zhisheng Niu [M'98, SM'99, F'12] graduated from Beijing Jiaotong University, China, in 1985, and got his M.E. and D.E. degrees from Toyohashi University of Technology, Japan, in 1989 and 1992, respectively. During 1992–1994, he worked for Fujitsu Laboratories Ltd., Japan, and in 1994 joined Tsinghua University, where he is now a professor in the Department of Electronic Engineering. His major research interests include queueing theory, traffic engineering, radio resource management, and green communication and networks.