A Traffic Infrastructure-Enabled Task Scheduling in Vehicular Edge Computing Networks

Pratham Oza
prathamo@vt.edu
Virginia Tech

Thidapat Chantem
tchantem@vt.edu
Virginia Tech

ABSTRACT

Large-scale deployment of connected vehicles and traffic infrastructure along with the advances in vehicle-to-everything (V2X) technology has paved the way for novel automotive applications to enhance safety and driving experience. Though, such applications require increased computational capabilities with timeliness guarantees. To satisfy the resource demands, the vehicular edge computing (VEC) paradigm extends the computational abilities of the vehicles by enabling them to offload certain tasks onto the edge servers deployed at the road-side units (RSUs). However, the real-time performance of the VEC tasks heavily depends on the network coverage and vehicle mobility, especially in urban traffic scenarios. In this work, we motivate the requirement of a real-time task model and task offloading schemes that incorporate the readily available traffic flow and signal timing data to guarantee real-time performance of tasks offloaded on the edge.

1 INTRODUCTION

Advancements in wireless technology, vehicle-to-everything (V2X) connectivity, artificial intelligence and sensors have led to the proliferation of smart vehicles and intelligent traffic infrastructure. By leveraging on-board sensors such as cameras, LiDARs and RADARs, as well as the data aggregated from surrounding connected vehicles and traffic infrastructure, vehicles can now provide enhanced safety and efficiency. Similarly, by utilizing smart road-side units (RSUs) and relevant information from vehicles, the traffic infrastructure can enable efficient traffic control with minimal delays, prioritized emergency vehicle movements, dissemination of safety-critical messages, etc. [8]. V2X connectivity is also being used to provide media-rich infotainment solutions such as augmented reality (AR) and video streaming, to the vehicle users to improve the overall driving experience [3]. However, such data-intensive applications are accompanied with increased computational requirements where on-board processing units located in the vehicles are not sufficient to satisfy such increased demands.

Cloud computing offers centralized remote servers that the vehicles can utilize, instead of the on-board processors, to perform certain computationally intensive tasks such as, AR-based heads up displays (HUDs) for real-time safety warnings or natural language processing (NLP) based driver command cognition systems. Along with real-time processing, such media-rich applications also call for low-latency real-time communication, which the centralized cloud architecture fails to provide, leading to poor quality of service (QoS) [12].

As the number of RSUs are continuously increasing to provide seamless V2X connectivity, enhanced traffic detection, and traffic measurements, the vehicular edge computing (VEC) paradigm proposes deployment of cloud services on these RSU to bring computation closer to the vehicles [4]. VEC offers reduced communication latency along with increased computational capabilities and thereby satisfying both real-time processing as well as communication constraints. Through V2X technologies such as dedicated short range communication (DSRC) or C-V2X, vehicles can now *offload tasks* that are computationally-intensive and/or delay-sensitive, onto these RSUs to utilize the additional processing resources at low communication latency.

While RSUs offer additional computational resources, with increasing number of vehicles especially in urban areas, task offloading policies are necessary to manage the resources. Recent works have proposed various schemes of distributing the tasks on the RSUs within range, by maximizing the resource utilization of the system [3], minimizing the energy expenditure of the vehicle [12], or by maximizing the QoS for all road users [13]. However, these schemes lack the consideration of criticality and timeliness requirements of the tasks that are being offloaded on the edge.

Further, due to limited coverage area of the RSUs, the communication link between a vehicle and an RSU is interrupted as a vehicle moves out of coverage [11]. This indicates that the task offloading schemes must also account for vehicle mobility as well as the transmission and handover/ takeover time between the vehicles and the RSUs while allocating resources, to ensure that the tasks are offloaded and completed in time [7]. Most VEC task offloading schemes that incorporate vehicle mobility consider an idealised traffic pattern where vehicles move along a constant stream of straight moving traffic, as seen in highway-like driving [5]. But, in urban traffic scenarios, vehicle mobility is heavily impacted by intersections and dynamic traffic density, which in turn affects VEC resource management [7]. Urban driving also offers increased RSUs with multiple sensors, cameras, traffic lights and traffic controllers deployed across the road network which can be utilized to provide seamless connectivity and meet the resource requirements for the increased traffic demands.

Most state-of-the-art applications in traffic infrastructure are utilizing edge-based resources and data aggregated through RSUs to provide *temporal predictability* in traffic movements and traffic signal control in an urban road network [8, 10]. A closely related work [7] models task offloading for vehicles in an urban VEC environment with signalized intersections as an RSU-vehicle matching problem using car-following models to estimate mobility. However, they do not utilize the traffic flow data to provide a predictable offloading schedule. In this work, we propose using the predictive traffic behavior from the traffic flow data to influence the scheduling of tasks and allocation of resources in a VEC framework. Specifically, we present a task offloading scheme for opportunistically scheduling tasks on the available RSUs, by considering the

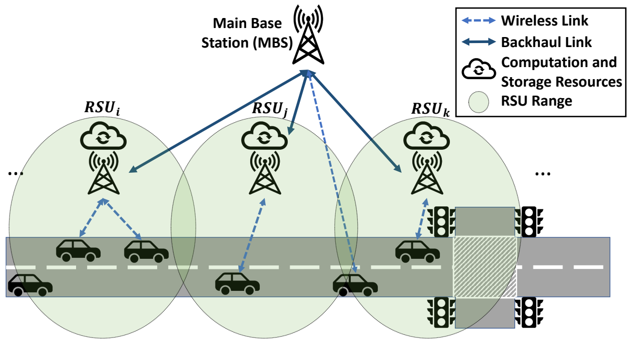

Figure 1: A typical vehicular edge computing framework

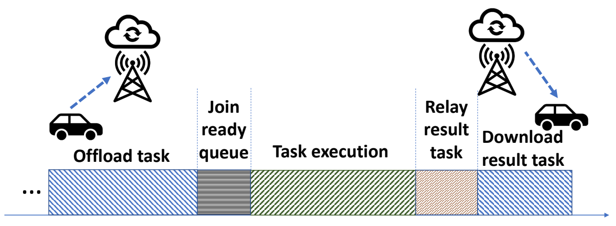

Figure 2: A task execution cycle

traffic signal timings and the wait times of the vehicles at multiple intersections.

Along with vehicle mobility and traffic flow parameters, the VEC task scheduling framework must also be aware of the properties of the tasks such as temporal requirements and deadlines, task criticality, effect on quality-of-service, etc. However, the recent work [5, 7, 9] lack a generic task model that can represent realistic workloads in the vehicular edge environment. Through this work, we also emphasise on the requirement of a real-time task model that characterizes a practical workload containing time-sensitive, critical tasks as well as delay-tolerant tasks with some effect on the quality of service of the system.

In all, this work highlights the need for a real-time model to represent VEC networks along with a task offloading scheme that:

- is deadline-aware and can cater to time-sensitive safety-critical tasks while providing timeliness guarantees for computationally-intensive data-rich tasks,
- incorporates readily available traffic data at intersections in modelling vehicle mobility through urban road networks to accuractely depict the delay overheads to VEC tasks, and,
- relies on a VEC task model that replicates realistic workloads in an urban environment.

Next, we will discuss the system model for the VEC framework and a motivating example of the advantages of incorporating the knowledge of traffic signal timings in task offloading process.

2 SYSTEM MODEL

2.1 VEC architecture and its components

Consider an urban traffic environment enabled with vehicular edge computing components distributed along the RSUs located by the road-side (Figure 1). The vehicles traveling along the road network utilize the processing resources at the RSUs equipped with VEC servers to meet their computation demands. The RSUs tend to have a smaller coverage area to enhance data transmission speeds. Coverage areas of consecutive RSUs have minimal overlap to expand the range of connectivity and reduce duplication of resources and communication. The VEC servers located with the RSUs provide increased computational capabilities with faster processing speeds as compared to the vehicles' on-board processors. RSUs are often equipped with multiple-input and multiple-output (MIMO) technology to communicate with multiple vehicles within its coverage area, at once.

A main base station (MBS) located along the urban network provides connectivity to the mobile devices, RSUs as well as the vehicles, and have a much larger coverage area encompassing multiple traffic intersections. Within the VEC paradigm, the MBS acts as a central coordinator that accepts offloading requests from the vehicles and schedules them on the RSUs for processing as per the underlying offloading policy. The MBS is connected with the RSUs through a wired (optical fiber) link that ensures stable network with reduced data transmission latency [4].

As the vehicles drive through the urban road network, they require the edge resources to compute multiple tasks originating from data-intensive applications or time-sensitive safety-critical applications. The vehicles require the edge resources to execute the tasks when either it lacks the on-board computational capability to complete the task execution or the tasks require data from other connected entities (vehicles and/or infrastructure) that is only available at the edge. These tasks have varying execution times and criticality, depending on the application. The vehicle sends the task information to the MBS after which the MBS assigns an appropriate RSU to execute the task as per the offloading policy. Further, the vehicles abide by the standardozed traffic rules and have ADAS features on-board to follow driving models such as Intelligent Driver Model (IDM) [6] for safe urban driving. The urban environment also consists of multiple signalized intersections through which the vehicles traverse, and are controlled using traffic lights that follow a green-yellow-red pattern. Novel traffic control techniques employ a centralized traffic controller deployed at the edge server to estimate the traffic flow and calculate the signal timings [8, 10]. With the VEC architecture and its individual components established, as shown in Figure 1, we will now discuss the need for accurately estimating vehicle mobility to guarantee real-time performance in the VEC environment.

2.2 Traffic mobility estimation for performance guarantees

When a vehicle requests the MBS for an edge resource to perform certain tasks, there are multiple network-related constraints that the MBS must consider before assigning an RSU resource to a vehicle. Note that the RSUs and the MBS are connected through a wired network and the MBS has the knowledge of the resource utilization of all RSUs within its coverage area.

- **Offloading task from a vehicle to an RSU:** A vehicle needs to be within the coverage area of the RSU before it can begin

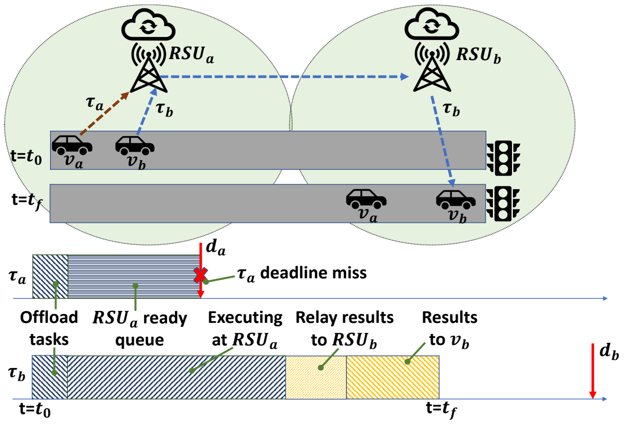

Figure 3: Traffic data unaware task offloading schemes

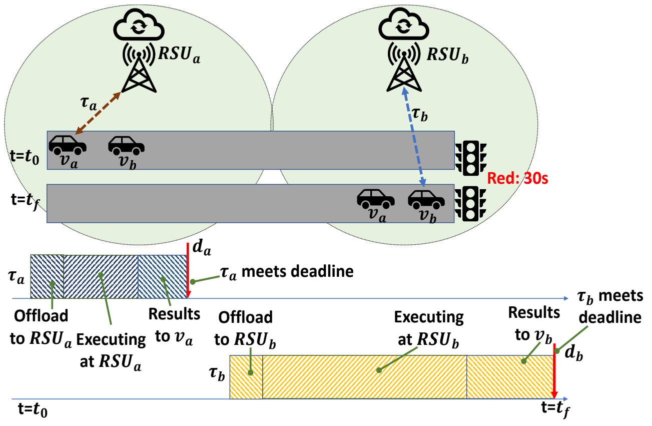

Figure 4: Proposed traffic data and deadline-aware task offloading scheme

offloading the task. The MBS must account for the time required for the vehicle to travel to the coverage area of an RSU before it can begin the offloading process.

- **Continuity of task offloading:** Once the vehicle is in the coverage area and starts offloading the task onto the RSU, the time it takes to offload a task depends on the size of the task. The larger the size, the longer it will take to offload the task. However, the vehicle is still in motion and may exit the coverage area of the RSU before the offloading completes. The RSUs then need to relay the already offloaded task information to subsequent RSUs while the vehicles continue offloading the remaining data to the next RSU. This incurs additional delays in re-transmission of data between the RSUs.
- **Wait time before execution:** Once the task is successfully offloaded onto the RSU, it joins the ready queue and begins execution only when the computation resource is available.
- **Offloading task from an RSU to the vehicle:** Once the task completes execution the resulting data must be offloaded back to the vehicle. Again, since the vehicle is already in motion, it may be the case that the task execution took place at an RSU which is away from the vehicle. The results must be transmitted to an appropriate RSU whose coverage area contains the vehicle currently, before the results are actually offloaded to the vehicle. The time delay associated with this transmission too, requires precise knowledge of the vehicles location and mobility.

Note that the transmission times and the delays mentioned here are over and above the actual task execution time at the edge resource (Figure 2). Clearly, all network-related delays require accurate estimation of vehicle mobility. In a highway setting, estimation of this time is trivial since the traffic flow is mostly constant. However, in an urban network, the travel time is not only influenced by the vehicle's own motion but is also impacted by the traffic lights and the traffic density through the road network. We therefore propose integrating the available traffic data and traffic signal timings for modeling the vehicle motion to more precisely estimate this travel time delay. We show in Section 3, that incorporating traffic data and signal timings to estimate vehicle's mobility and location helps in maximizing the resource utilization of the entire VEC system with more vehicles able to offload their desired tasks on the VEC network.

2.3 A task model for VEC workloads

Most existing task offloading policies [7, 12] are best-effort and do not consider task deadlines while scheduling and offloading tasks over the RSUs and hence provide no real-time performance guarantees. However, multiple standardization groups (European Telecommunications Standards Institute (ETSI) and the 3GPP), highlight the necessity for performance guarantees in VEC especially for safety-critical applications [2]. To ensure timeliness guarantees within the VEC paradigm, it is not only important to accurately estimate the communication overheads which rely heavily on vehicle mobility, but it is also necessary to consider task deadlines, especially for safety-critical tasks.

We denote the j^{th} task to be offloaded from i^{th} vehicle as $\tau_{i,j}$ which is represented by the tuple $\{b^o_{ij}, C_{ij}, d_{ij}, b^d_{ij}\}$ where,

- b^o_{ij}: size of the task (in Mb) to be offloaded from the vehicle to the RSU,
- C_{ij}: execution time of the task on the RSU,
- d_{ij}: absolute deadline of the task, and
- b^d_{ij}: size of the result (in Mb) to be offloaded from the RSU to the vehicle.

The existing works do not have a generic task model to represent workloads in VEC [7, 9, 12]. As explained in Section 2.2, the transmission times and communication delays depend on the size of the task and the computed results, along with the vehicle mobility. Hence, we want to emphasize the necessity of a real-time model to represent realistic workloads and provide novel scheduling schemes that enhance the predictability of the VEC environment and adhere to the performance requirements and timeliness guarantees standardized by the industry [2].

3 MOTIVATION AND CHALLENGES

As discussed in the previous section, the main base station (MBS) is a central entity to coordinate and schedule tasks on to the RSUs such that the delays involving data transmission to/from the RSUs as well as the execution time are accounted for while ensuring that the tasks meet their deadlines. The delays rely heavily on the MBS' estimate of the vehicles' location and its mobility. Previous works such as [3, 11, 13] do not consider dynamic traffic mobility especially in an urban environment where the demand for edge-based resources is high. [7] considers urban mobility through road networks with multiple intersections but does not incorporate traffic signal timing data in providing predictable offloading strategy. In all, none of these approaches guarantee real-time performance and are best-effort in either minimizing energy utilization, transmission costs, or transmission delays.

Clearly, the total execution times for the tasks to be offloaded depend heavily on the vehicle's mobility. Further, multiple safety-critical applications with strong reliance on cooperative edge-based processing, such as merging at blind turns, collision-free intersection management, etc. require hard deadline guarantees [1].

To motivate our work, let us consider a simple vehicular edge environment as shown in Figures 3 and 4 in which two vehicles, v_a and v_b require the RSUs to execute certain tasks. Here, v_a needs to offload a time-critical task, τ_a with a deadline d_a, and v_b needs to offload a data-intensive task, τ_b with a deadline d_b. The execution time of τ_a is shorter than that of τ_b. Additionally, there are two RSUs in the vicinity namely, RSU_a and RSU_b, which have the resource bandwidth of only performing one task at a time. Both v_a and v_b are approaching a signalized intersection. As mentioned in Section 2, both tasks can be uploaded to an RSU at once due to the MIMO functionality.

Task execution with state-of-the-art offloading scheme: Now, as shown in Figure 3, a task offloading scheme such as [7], which only considers a simplistic car-following model for vehicle mobility and neither utilizes exact traffic information, nor is it deadline-aware and hence offloads tasks τ_a and τ_b on RSU_a. Task τ_b starts execution while τ_a joins the ready queue. The state-of-the-art approaches assume a FIFO scheduling of the tasks. Since τ_b is a data-intensive task, it takes longer time to execute, leading to a missed deadline for a safety-critical task, τ_a. Such missed deadlines could hamper the driving and safety performance of the vehicles. Further, v_b moves out of the coverage area of RSU_a and stops at the red light within the coverage area of RSU_b leading to a longer total execution time of τ_b since the task results need to be transmitted to RSU_b from RSU_a and then to v_b.

Task execution with proposed offloading scheme: Now, consider a deadline-aware, traffic infrastructure-aware task scheduling scheme as in Figure 4. The task offloading policy knows that the upcoming traffic light is expected to stay red for 30 seconds, and vehicle v_b is going to stop for a prolonged period due to the red light. The policy therefore offloads only task τ_a onto RSU_a and due to its shorter execution time, it finishes execution and results transmission back to v_a while it is in the coverage area of RSU_a and thereby meeting its deadline. Additionally, even though τ_b is a computationally-intensive task, it successfully completes offloading to RSU_b, execution at RSU_b and transmission of results from RSU_b, all while it is waiting at the traffic light.

Thus, a criticality-aware VEC task model with a traffic infrastructure and deadline-aware task offloading policy does not only maximizes the resource utilization of the RSUs but also reduces the transmission delays between the RSUs, while ensuring that all tasks meet their deadlines.

4 CONCLUDING REMARKS

Through this work, we motivate the need for a real-time model for urban VEC environment with task scheduling policy that,

- exploits the readily available traffic data through RSUs and sensors, as well as the traffic signal timings information through connected infrastructure to predictably offload and schedule tasks on the edge servers,
- provides timeliness and performance guarantees by limiting the deadline misses and maximizing resource utilization, and
- relies on a deadline-aware task model specifically to replicate realistic workloads.

5 FUTURE WORK

We will formulate the proposed task model for realistic workloads that resemble deep-learning based tasks with timeliness and QoS requirements. Using a traffic simulator such as VISSIM, we will emulate large-scale traffic with connected vehicles on an urban roadmap with multiple intersections and traffic lights. This experimental setup will be used to deploy and evaluate our proposed task model and VEC offloading scheme.

REFERENCES

[1] S. Aoki and R. Rajkumar. 2019. V2V-based Synchronous Intersection Protocols for Mixed Traffic of Human-Driven and Self-Driving Vehicles. In *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. 1–11. https://doi.org/10.1109/RTCSA.2019.8864572

[2] Multi-Access Edge Computing. 2018. Study on MEC Support for V2X Use Cases. *Standard ETSI GR MEC* 22 (2018), V2.

[3] Yueyue Dai, Du Xu, Sabita Maharjan, and Yan Zhang. 2018. Joint load balancing and offloading in vehicular edge computing and networks. *IEEE Internet of Things Journal* (2018).

[4] Hisashi Futaki. 2020. RSU apparatus, base station apparatus, control node, and methods therein. US Patent 10,595,157.

[5] X. Huang, L. He, and W. Zhang. 2020. Vehicle Speed Aware Computing Task Offloading and Resource Allocation Based on Multi-Agent Reinforcement Learning in a Vehicular Edge Computing Network. In *2020 IEEE International Conference on Edge Computing (EDGE)*.

[6] Martin Liebner, Michael Baumann, Felix Klanner, and Christoph Stiller. 2012. Driver intent inference at urban intersections using the intelligent driver model. In *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 1162–1167.

[7] Pengju Liu, Junluo Li, and Zhongwei Sun. 2019. Matching-based task offloading for vehicular edge computing. *IEEE Access* (2019).

[8] P. Oza, T. Chantem, and P. Murray-Tuite. 2020. A Coordinated Spillback-Aware Traffic Optimization and Recovery at Multiple Intersections. In *2020 IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*.

[9] J. Sun, Q. Gu, T. Zheng, P. Dong, A. Valera, and Y. Qin. 2020. Joint Optimization of Computation Offloading and Task Scheduling in Vehicular Edge Computing Networks. *IEEE Access* 8 (2020), 10466–10477. https://doi.org/10.1109/ACCESS.2020.2965620

[10] S. Ucar, T. Higuchi, and O. Altintas. 2020. Signal Phase and Timing by a Vehicular Cloud. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*.

[11] Chao Yang, Yi Liu, Xin Chen, Weifeng Zhong, and Shengli Xie. 2019. Efficient mobility-aware task offloading for vehicular edge computing networks. *IEEE Access* (2019).

[12] Zhenyu Zhou, Pengju Liu, Zheng Chang, Chen Xu, and Yan Zhang. 2018. Energy-efficient workload offloading and power control in vehicular edge computing. In *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*.

[13] Chao Zhu, Giancarlo Pastor, Yu Xiao, Yong Li, and Antti Ylae-Jaeaeski. 2018. Fog following me: Latency and quality balanced task allocation in vehicular fog computing. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*.