OCTANE: A Joint Computation Offloading and Resource Allocation Scheme for MEC Assisted 5G NR Vehicular Networks

Veerendra Kumar Gautam *, Chinmay Tompe†, Bheemarjuna Reddy Tamma*, Antony Franklin A*
* Department of CSE, Indian Institute of Technology Hyderabad, India
† Department of ECE, NYU Tandon School of Engineering, Brooklyn, New York
{cs18resch01003*, tbr*, antony.franklin* }@iith.ac.in
† chinmay.tompe@nyu.edu

***Abstract*—New vehicular applications like Augmented Reality (AR), Virtual Reality (VR), and High Definition Map (HD Map) have computational intensive and latency-sensitive traits and require collaboration among nearby vehicles. Computational offloading is used to improve the accuracy and performance of these applications, as it allows computational jobs to be processed on MEC servers at the cell edge. Here, the challenge is how to effectively take offloading decisions at the MEC server by considering wireless transmission delay and computational delay in the presence of time varying channel conditions due to vehicular mobility. In this work, we aim to maximize the number of jobs offloaded to the MEC server under application's deadline constraints while ensuring fairness among vehicles. First, we formulate the computational offloading as an integer linear programming (ILP) problem where both the transmission delay of 5G NR and MEC computational resources are taken into account. Then, we propose an online heuristic for joint computational offloading and resource allocation, OCTANE, that jointly takes 5G NR radio resources and computational resources into consideration while taking offloading decisions. Further, to provide fairness among vehicles, Transport Block Size (TBS) based Medium Access Layer (MAC) strategy is proposed for allocation of TDMA symbols in the 5G NR uplink. Finally, extensive simulations are performed in the NS-3 5G NR module with mobility traces taken from SUMO using OpenStreetMap to evaluate OCTANE and ILP model. Simulation results show that the proposed OCTANE scheme performs better than a state-of-the-art solution and is close to the ILP model in terms of offloading success rate.**

I. INTRODUCTION

The revolutionary era in the automotive industry begins with key milestone technologies like Vehicle to Everything (V2X) and Mobile Edge Computing (MEC). V2X technology is on the cusp of becoming essential in all the new vehicles launched in the market. Particularly, with the advent of 5G NR C-V2X, it is possible to transform driver experience to the next level. As a result, numerous automotive applications and services have been envisaged to provide a better driver experience like AR/VR and HD Maps. However, these resource-hungry applications consume immense computational and radio resources, and impose stringent latency-reliability requirements. Moreover, due to time varying channel conditions and limited availability of computational resources at vehicles, timely processing of sensor data of the vehicle and its surrounding vehicles to offer rich driving experience is very challenging. MEC aims to address this by providing computational resources in close vicinity to vehicles on the edge cloud. Here, vehicles intelligently leverage computational resources of MEC by offloading computationally intensive jobs to an MEC server over 5G NR.

In MEC, contention between vehicles for resources is fierce, as there are limited computational resources (e.g., in comparison to Cloud). A vehicle should perform efficient and effective management of offloading decisions. Since offloading jobs to the MEC server induces extra data transmission delay and remote computational delay, these need to be factored during the allocation of radio and computational resources to each job. Previously, this problem has been solved partially by providing offloading decisions [1], radio resources [2], or computational resources [3] for in quasi-static scenarios (e.g., for stationary users). Recently, the authors of [4] addressed the joint task offloading and resources optimization problem for mobile users in 4G networks. However, with advancements in 5G NR in terms of numerology options, MAC scheduling, and transport block (TB) sizes, guaranteeing QoS of the vehicular applications requires new mechanisms for job offloading and resource management. In this regard, radio and computational resources are inextricably linked together to complete the jobs in mobile environments. The challenge here lies in designing offloading mechanisms that take mobility, density, and QoS requirements of vehicular application into consideration while allocating radio and computation resources in a joint manner.

In this paper we propose an online heuristic named OCTANE for joint computational offloading and resource allocation in 5G NR based vehicular networks. We also propose a new strategy based on Transport Block Size (TBS) for allocation of TDMA symbols in uplink to meet QoS of vehicular applications in 5G NR based MEC system. Here, the objective is to maximize the number of jobs successfully completed for each vehicle with the help of the MEC server, while ensuring fairness among the vehicles. The key contributions of the paper are as follows:

1) We formulate an integer linear programming (ILP) model for job offloading that jointly considers computational and radio resource allocations in 5G NR C-V2X based MEC system.
2) We propose an online heuristic named OCTANE for joint computational offloading and resource allocation which tries to increase offloading success rate.

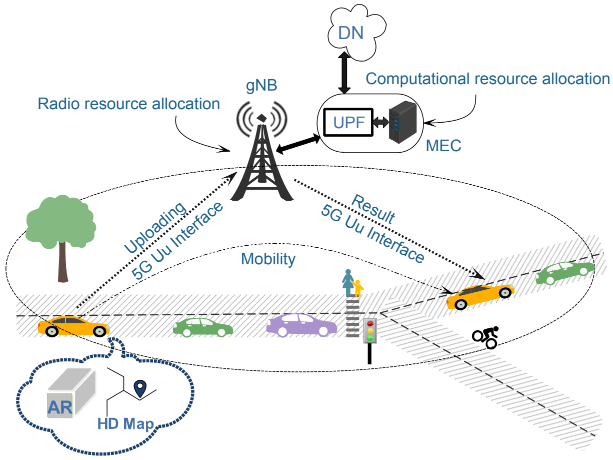

Fig. 1: System Model

II. Related Work

In recent years, synergy between computational offloading and radio resource allocation to reduce data processing time is drawing attention. The authors of [5] formulated the joint problem of offloading and radio resource allocation as an Integer programming (IP) problem and proposed a solution based on dynamic search algorithm for a Wi-Fi based MEC system. The author of [6] proposed a Multi-Dimensional Search and Adjust, an offline algorithm that sub-optimally solves the combined computation partitioning and radio resource allocation problem. They also proposed an online method called cooperative online scheduling. Factoring in the time varying channel conditions in vehicular networks, the authors of [7] put forward a convex optimization-based solution with the objective to minimize the total energy consumption of the users. The authors of [8] addressed the radio resource allocation problem using convex and quasi-convex optimization techniques and also proposed a novel heuristic algorithm for offloading. These works usually first calculated the expected delay or energy consumption for target jobs and then offloaded them to the mobile edge if the delay or energy consumption can be reduced further. The authors of [4] considered the problem of offloading decision and resource allocation to achieve the optimal system-wide user utility in a multi-user mobile scenario and proved that the problem is NP-Hard. Then they proposed a heuristic mobility-aware offloading algorithm (HMAOA) to obtain approximate optimal offloading decisions in polynomial time.

In this paper, we propose a novel joint computation offloading and computational resource allocation scheme named OCTANE in 5G NR C-V2X based MEC system. The proposed scheme jointly decides two things: i) subset of jobs to be offloaded by the vehicles and ii) computational resource allocations to the offloaded jobs at the MEC server. Moreover, we compare the proposed scheme with HMAOA algorithm to show its superiority.

III. System Model and Problem Formulation

In this section, we first describe the 5G NR C-V2X based MEC system model considered. Then, we introduce vehicle side local computing and MEC side (offloading) computing models. Thereafter, we formulate the job offloading decision problem as an ILP model by considering radio and computational resources. The notations used are summarized in Table I.

TABLE I: NOTATIONS

Symbol	**Description**
$\mathcal{V}$	Set of vehicles
$\mathcal{K}$	Set of jobs
$\mathcal{K}'$	Set of jobs offloaded to MEC Server
$\mathcal{W}_{v,k}$	Job tuple
$\alpha_{v,k}^{in}$	Input size of the job
$\beta_{v,k}^{cpu}$	CPU cycles required for the job
$\gamma_{v,k}^{delay}$	Deadline of the job
$T_{v,k}^{local}$	Local computational delay for the job
$T_{v,k}^{mec}$	MEC computational delay for the job
$T_{v,k}^{NR}$	5G NR Transmission Delay
f_{v}^{local}	Vehicle's processing capacity
F_{max}	MEC server's processing capacity
$x_{v,k}$	Number of RBs required for the job
$y_{v,k}$	Number of CPU cycles required for the job
rbg_{th}	Resource Block (RB) Threshold for the job
job_{th}	Job threshold per vehicle per second
N_{total}^{RB}	Total number of RBs per second
$t_{\text{slot}}^{\text{NR}}$	Slot duration in 5G NR
$\phi_{v,k}$	Vehicle side binary variable for offloading
$\eta_{v,k}$	MEC side binary variable for offloading

A. System Model

In this paper, to show the impact of mobility, we have taken a typical scenario of V vehicles navigating along a roadway. The set of vehicles are denoted by $\mathcal{V} = \{1,\ldots,V\}$ and indexed by $v \in \mathcal{V}$. All vehicles are equipped with 5G NR based On-Board Units (OBUs) which have their own local but limited computational capabilities. As shown in Fig. 1, we consider a 5G NR base station (gNodeB) which is connected to an MEC server that serves as a proximate cloud by which vehicles connected to the gNodeB extend their computing power by offloading their jobs. The assumption is that vehicles are connected to the MEC server over Vehicle-to-Network (V2N) in 5G NR C-V2X. Vehicles generate jobs with timing constraints that can be done locally. If local computing capacity is not sufficient or jobs are going to miss their deadlines, then jobs are considered for offloading to the MEC server which incurs an extra transmission delay. All the offloading requests made by vehicles are processed by the MEC server but some of them could be denied due to the following reasons: 1) job requires more computational resources than what the MEC server has, 2) total delay is more than the deadline of the job, and 3) jobs per vehicle threshold exceeds. The MEC server endeavors to maximize the number of jobs successfully offloaded and it makes offloading decisions i.e., accepting/rejecting requests coming from the vehicles in near real-time. However, the MEC server also has limited computational resources, hence it should consider both computational and radio resources jointly while making offloading decisions. If a vehicle is having poor channel conditions, then more computational resources should be allocated to its jobs at the MEC server to ensure that the jobs meet their timing constraints.

B. Computation Jobs and Data Arrival Models

It is assumed that an instance of vehicular application is installed on the vehicles and at the MEC server. Let there be $\mathcal{K}_v = \{1, \ldots, K\}$ jobs that are produced by a vehicle v ($v \in \mathcal{V}$) at each t ms and indexed by $k \in \mathcal{K}_v$. Vehicle v has a computational job tuple $\mathcal{W}_{v,k} = \{\alpha_{v,k}^{in}, \beta_{v,k}^{cpu}, \gamma_{v,k}^{delay}\}$. Here $\alpha_{v,k}^{in}$ is the size of the job's input data in KB and $\beta_{v,k}^{cpu}$ is the number of CPU cycles required to complete the job. $\gamma_{v,k}^{delay}$ is the deadline. $\phi_{v,k}$ denotes the offloading decision taken by the vehicle. $\phi_{v,k} = 1$ indicates that the job needs to be offloaded to the MEC server, while $\phi_{v,k} = 0$ indicates that the job is executed locally. If a vehicle chooses to offload a job to the MEC server, a job request is sent to the MEC server before offloading the job. If response is positive from the MEC server, then the vehicle will offload the job else it is marked as rejected due to resource scarcity.

C. Computation Model

1) Local Computation Model: If $\phi_{v,k} = 0$ (i.e., job is executed locally), then its computational delay is given by:

$$T_{v,k}^{local} = \left\lceil \frac{\beta_{v,k}^{cpu}}{f_v^{local}} \right\rceil \tag{1}$$

Where f_v^{local} is the local processing capacity (i.e., CPU clock frequency) of the vehicle v. $T_{v,k}^{local}$ should be less than $\gamma_{v,k}^{delay}$; otherwise the job needs to be checked for offloading to the MEC server by setting $\phi_{v,k} = 1$. Since the job processing result is much smaller than the input size of the job, we have not considered it in case of offloading [4].

2) MEC Computation Model: When a job is delegated to the MEC server, it allocates some CPU cycles to the job. Let $y_{v,k}$ is the allocated CPU capacity for the job. Computational delay for the job to be executed remotely is given by:

$$T_{v,k}^{mec} = \left\lceil \frac{\beta_{v,k}^{cpu}}{y_{v,k}} \right\rceil \tag{2}$$

D. 5G NR Transmission Delay

In order to calculate transmission delay in 5G NR, first we need to calculate number of Resource Blocks (RBs) that are required to offload a job to the MEC server. The required number of RBs can be calculated using this equation :

$$x_{v,k} = 14 \times \left\lceil \frac{\alpha_{v,k}^{in} \times 1024 \times 8}{n_{\mathrm{RE}}^{\mathrm{NR}} \times n_{\mathrm{nbpm}}^{\mathrm{NR}}} \right\rceil \tag{3}$$

There are 168 Resource Elements (RE) in a resource grid (i.e., 14 OFDM symbols in time domain in one slot and 12 sub-carriers in frequency domain in one resource block) in 5G NR. Demodulation reference signal (DMRS) is used to estimate the channel state, which could take 16 REs (mapping type =A with 3 additional symbols). So for data transmission the number of REs available ($n_{\mathrm{RE}}^{\mathrm{NR}}$) is reduced to 152 [9]. $n_{\mathrm{nbpm}}^{\mathrm{NR}}$ is the spectral efficiency corresponding to the chosen Modulation and Coding Scheme (MCS) value [10].

Transmission delay ($T_{v,k}^{NR}$) is determined by:

$$T_{v,k}^{NR} = \left\lceil \frac{x_{v,k}}{n_{\mathrm{RB-slot}}^{\mathrm{NR}} \times (1 - \mathrm{OH})} \right\rceil \times t_{\mathrm{slot}}^{\mathrm{NR}} \tag{4}$$

where ($t_{\mathrm{slot}}^{\mathrm{NR}}$) is the slot duration. $n_{\mathrm{RB-slot}}^{\mathrm{NR}}$ is the number of RBs given to a vehicle in a slot and OH is the uplink overhead (=8% [11]).

E. Proposed ILP Model

In this section, we formulate the joint problem of radio and computational resources for offloading decisions of jobs as an ILP model under the effect of mobility of vehicles in 5G NR based MEC system. The objective is to maximize the number of jobs admitted by the MEC server by taking care of fairness among vehicles and delay constraints of the jobs. The ILP model is formulated as follows:

$$\mathcal{P}_{\mathbf{MEC}} : \max \sum_{v=1}^{|\mathcal{V}|} \sum_{k=1}^{|\mathcal{K}'|} \eta_{v,k} \tag{5}$$

subject to:

$$\mathrm{C_{MEC}}1 : \sum_{v=1}^{|\mathcal{V}|} \sum_{k=1}^{|\mathcal{K}'|} x_{v,k} \times \eta_{v,k} \leq N_{total}^{RB}, \forall v \in \mathcal{V}, \forall k \in \mathcal{K}'$$

$$\mathrm{C_{MEC}}2 : \sum_{k=1}^{|\mathcal{K}'|} \eta_{v,k} \leq job_{th}, \forall v \in \mathcal{V}$$

$$\mathrm{C_{MEC}}3 : \sum_{v=1}^{|\mathcal{V}|} \sum_{k=1}^{|\mathcal{K}'|} y_{v,k} \times \eta_{v,k} \leq F_{max}, \forall v \in \mathcal{V}, \forall k \in \mathcal{K}'$$

$$\mathrm{C_{MEC}}4 : T_{v,k}^{NR} + T_{v,k}^{mec} \leq \gamma_{v,k}^{delay}, \forall v \in \mathcal{V}, \forall k \in \mathcal{K}'$$

$$\mathrm{C_{MEC}}5 : \eta_{v,k} \in \{0, 1\}, \forall v \in \mathcal{V}, \forall k \in \mathcal{K}'$$

The constraint $\mathrm{C_{MEC}}1$ ensures that the total radio resources required by the offloading jobs should not exceed maximum value N_{total}^{RB} of the 5G NR base station. $\mathrm{C_{MEC}}2$ stands for job threshold per vehicle. $\mathrm{C_{MEC}}3$ ensures that total computational resources of the jobs should not exceed maximum CPU capacity of MEC, F_{max}. $\mathrm{C_{MEC}}4$ states that transmission delay and computational delay should not exceed job's deadline. $\mathrm{C_{MEC}}5$ indicates the binary variable for offloading decision.

IV. OCTANE: Proposed Heuristic Scheme

Due to the computational complexity of the formulated ILP model in the previous section, we propose an online heuristic for joint computational offloading and resource allocation named OCTANE to solve the $\mathcal{P}_{\mathbf{MEC}}$ problem in near real-time. OCTANE is given in Algorithm 1. It has $\mathcal{O}(\mathcal{K}' \log \mathcal{K}')$ complexity iterative solution, which is put forward to solve the offloading decision problem by considering deadline, radio, and computational resources requirements of the jobs of the vehicles. Further, to provide fairness among vehicles while job offloading, a Transport Block Size-based strategy (Algorithm 2) is proposed for the MAC layer which runs in accordance with the Maximum Rate (MR) scheduler while allocating TDMA symbols to the vehicles in the uplink.

OCTANE takes offloading requests from $|\mathcal{V}|$ vehicles for offloading jobs to the MEC system. It first sorts $\mathcal{K}_v$ jobs for a vehicle v in ascending order of its job deadlines (i.e., each element k in the sorted list $\mathcal{K}_v^{sort}$ has $\gamma_{v,k}^{delay}$ less than or equal to $\gamma_{v,(k+1)}^{delay}$). It makes $|\mathcal{V}|$ iterations and calculates the maximum data transfer for a vehicle v in the next t_d ms by using 5G NR equation given in [11]. Next, it selects the maximum number of jobs and maintains fairness among vehicles. To do that, for each job in $\mathcal{K}^{sort}$, it calculates the total delay T_{delay} (which comprises of T_v^{NR} and T_v^{mec}) and admits a job which meets the job deadline, $\gamma_{v,k}^{delay}$. It maintains a job_{th}^v threshold for the number of admitted jobs per vehicle to maintain fairness among vehicles. Thereafter, it subtracts input data size $\alpha_{v,k}^{in}$ from D_{max} of the vehicle and CPU cycles $\beta_{v,k}^{cpu}$ from F_{max}. rb_{th} is the RB threshold given per job to calculate the delay of the job.

Algorithm 1 **OCTANE**

inputs : $\mathcal{V}$, $\mathcal{K}'$
output: Job Offloading Decision
$\mathcal{K}^{sort} \leftarrow$ `Sort` ($\mathcal{K}'$, $s.t.\ \gamma_{v,k}^{delay} < \gamma_{v,(k+1)}^{delay}$)
forall $v \in \mathcal{V}$ **do**
    if $mcs_v \geq mcs_{th}$ **then**
        $D_{max}^v \leftarrow$ `MaxDataTransferPerVehicle`(mcs_v)
    end
end
forall $k \in \mathcal{K}^{sort}$ **do**
    $T_{delay} \leftarrow$ `TotalDelayPerJob`($\mathcal{W}_{v,k}, rb_{th}$)
    if $k \leq job_{th}^v$ & $T_{delay} \leq \gamma_{v,k}^{delay}$ **then**
        if $(F_{max} - \beta_{v,k}^{cpu}) \geq 0$ & $(D_{max}^v - \alpha_{v,k}^{in}) \geq 0$ **then**
            `OffloadJob`(*1*) /* request offloading */
        else
            `OffloadJob`(*0*) /* reject job */
        end
    end
end

In 5G NR, TDMA-based scheduling is supported for uplink, in which symbols are assigned to vehicles and switching of beams is performed at the PHY layer. Algorithm 2 describes the allocation of symbols on the basis of current TBS values of the vehicles. TBS defines payload size, which passes between MAC and PHY layers of the 5G NR; TBS calculations are done by considering resource blocks, MCS, and numerology in 5G NR. In this strategy, we first calculate the maximum TBS. It creates the vehicle-symbol list which limits the number of symbols given per each vehicle by comparing the maximum TBS with the current TBS of the vehicle. The intuitive idea is to allocate more symbols to the vehicles with middle-level TBS values. Vehicles with higher-level TBS values will use higher MCS to send their data with fewer symbols. Vehicles with lower-level TBS values will not be able to send that much data because of poor channel conditions. That is why we have allocated one symbol for the vehicles with high and low TBS values. At the MAC layer, MR scheduler prioritizes the vehicles having better channel conditions from the list of active vehicles and allocates appropriate number of symbols in the Uplink to vehicles using the vehicle-symbol list. The vehicle-symbol list changes at a granularity of running Algorithm 2.

Algorithm 2 **TBS Strategy**

input : $\mathcal{V}$
output: TDMA Symbol Allocation for Vehicles
TBS_{max} = `GetMaxTBSsize` (mcs_{max})
forall $v \in \mathcal{V}$ **do**
    TBS_v = `GetTBSsize` (mcs_v)
    if $TBS_v \leq TBS_{max}/2$ & $TBS_v > TBS_{max}/3$ **then**
        $SymPerSlot_v$ = 2
    end
    if $TBS_v > TBS_{max}/2$ **then**
        $SymPerSlot_v$ = 1
    end
    if $TBS_v \leq TBS_{max}/3$ **then**
        $SymPerSlot_v$ = 1
    end
end

V. Simulation Setup and Performance Results

In this section, we first describe the simulation setup and then use the offloading success rate, offloading rate, and Jain's Fairness Index (JFI) as performance metrics to evaluate proposed heuristic solution along with the ILP model in different vehicular environments.

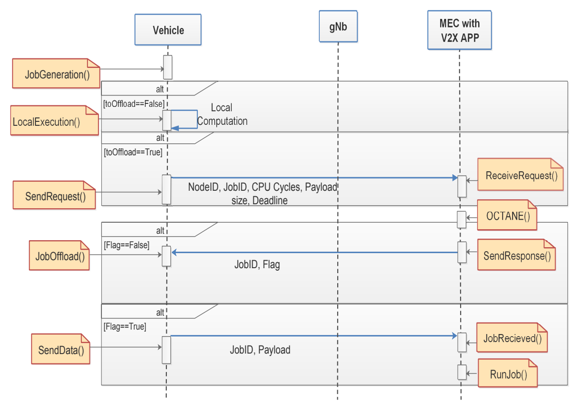

Fig. 2: Sequence Diagram of UdpOffloading Application.

A. Simulation Setup

The simulation scenario consists of a two-way Pembina Canada Highway segment of 250 meter length near Winnipeg which is served by a single 5G NR base station. Rapid Cellular Network Simulation Framework (RACE) [12] is used to generate vehicular traffic in the chosen highway segment. RACE uses cellular infrastructure dataset given by the Canadian organization of Innovation, Science and Economic Development (ISED)[1] which includes Canadian cellular providers like Telus,

[1] https://sms-sgs.ic.gc.ca/eic/site/sms-sgs-prod.nsf/eng/h_00010.html

Rogers, and Bell. RACE uses OpenStreetMap[2] to export maps and SUMO[3] for customized traffic generation. All simulation experiments are carried on the NS-3 [4]. To realize 5G links for vehicles in the 5G NR-based MEC system, we used the 5G NR module [13] developed by Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA).

In the simulated scenario, each vehicle generates different jobs related to HD Map like a collection of sensor data, sensor data analysis, and HD-map update [14]. Sensor data analysis is a compute-intensive job that needs to be offloaded to the MEC server over 5G NR. We developed a job offloading application named UdpOffloading, which is based on the Udp-Client-Server application of NS-3. The client application is deployed on vehicles, whereas the server application is deployed on the MEC server. The client application can be programmed to generate jobs with different input sizes, deadlines, and CPU cycle requirements at specified intervals. Vehicles can decide to send jobs to the MEC server or execute them locally. On the other hand, the MEC server receives requests from the vehicles in the coverage region of the 5G NR base station, and replies offloading decisions to the respective vehicles. The sequence diagram shown in Fig. 2 sketches the process of job offloading between vehicles and the MEC server over 5G NR. We use this UdpOffloading application to showcase the HD-map app. Herein, the latency of jobs are set according to 5GCAR deliverable [15] and other parameters are set according to [4] and [16] which are shown in Table II.

B. Performance Results

We compare the performance of proposed OCTANE with HMAOA [4] and ILP optimization model. The ILP model is solved using CPLEX solver. In this paper, we consider offloading success rate (OSR) as one of the key performance metrics. A job is treated as a success when it is offloaded to the MEC server and executed within its deadline. OSR is defined as the ratio of the number of successfully executed jobs by the MEC server to the total number of offloading jobs requests received by the MEC server. OSR is measured at the MEC server for different schemes in different vehicular scenarios. The offloading rate (OR) is another metric that we use. OR is the ratio of the number of jobs offloaded to the MEC server to the total number of requests received by the MEC server. The difference in OSR and OR is derived from the jobs which are offloaded but did not get successfully executed within their deadlines by the MEC server. This happens because of an increase in job offloading delay, for example, due to vehicles' poor channel conditions. If a vehicle decides to run a job locally, that job is excluded from OSR and OR calculations. Rejection rate is defined as the ratio of job requests rejected by the MEC server due to resource scarcity to the total number of requests received by the MEC server. Each simulation experiment is repeated for 10 seeds, and the results are presented with 95% confidence intervals.

[2]http://www.openstreetmap.org/

[3]http://www.sumo.dlr.de/userdoc/SUMO.html

[4]https://www.nsnam.org

TABLE II: Simulation Parameters

Parameter	**Value**
Scenario	Urban Macro Cell
Number Of Vehicles $\|\mathcal{V}\|$	10-25
Mobility Model	Krauss
Average Vehicle Velocity ($\mathcal{V}_{vel}$)	20-80 kmph
5G NR Base Station/Vehicle TX power	46/23 dBm
5G NR Base Station Antenna Pattern	Canadian dataset
5G NR Base Station Antenna Tilt	15°
5G NR Base Station/Vehicle Antenna Height	25 m / 1.5 m
Carrier Frequency	6 GHz
Channel Model	3GPP, LoS
Channel Condition	Line-Of-Sight
Channel Bandwidth	30 MHz
5G NR Numerology μ	1
CPU Clock Frequency Of MEC server (F_{max})	80 GHz
Vehicle's CPU Clock Frequency (f_v^{local})	2 GHz
Number of Jobs per Vehicle ($\mathcal{K}$)	[1,2]
Input Size of the job ($\alpha_{v,k}^{in}$)	Mean: 6 Kbytes
	Variance: 10 Kbytes
	Bound: 5 Kbytes
CPU cycles required per job ($\beta_{v,k}^{cpu}$)	[4,140] Mcycles
Deadline of job ($\gamma_{v,k}^{delay}$)	[100,150] msec
Resource Block Threshold per Job (rbg_{th})	1000 RB
Job Threshold per Vehicle per second (job_{th}^{v})	12
Interval For Maximum data transfer per vehicle (t_d)	100 msec
Job generation per vehicle t	0.1 Seconds
MCS Threshold mcs_{th}	5

1) Effect of number of vehicles: In Fig. 3, we show the variation in the offloading success rates (OSR) of the three schemes (OCTANE, ILP model, and HMAOA) by varying number of vehicles from 10 to 30 by keeping the average velocity of vehicles fixed at 60 kmph. Here, we observe that as the traffic load tends to increase, OSR begins to decrease for all the schemes under study. It is because more number of offloading requests coming to the MEC server increase the contention for computational resources at the MEC server as well as radio resources in 5G NR. Our proposed scheme OCTANE performs better than that of HMAOA for various densities of vehicles and comes close to ILP model. HMAOA, OCTANE and ILP model offer an ensemble average OSR of 57%, 75%, and 77%, respectively. In Fig. 4, job rejection rate is plotted by varying number of vehicles. The average job rejection rates of HMAOA, OCTANE and ILP model are 28.8%, 2.4%, 2.15%, respectively.

2) Effect of velocity of vehicles: To study the effect of velocity of the vehicles on OSR, we varied the average velocity of vehicles using acceleration and speed factor parameters of vehicles in SUMO and used these traces in NS-3 experiments. As shown in Fig. 5, increase in the average velocity of the vehicles leads to reduction in OSR for all the schemes under study. Here, ILP optimally selects the jobs, thereby achieving the highest OSR. In comparison, other schemes exhibit a drift from the optimal solution. With increase in velocity of vehicles, channel quality estimates become increasingly

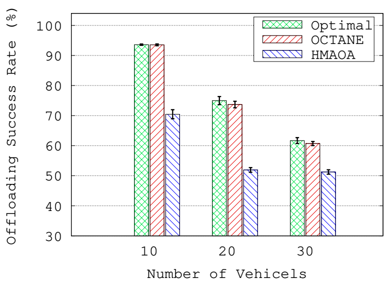

Fig. 3: OSR Vs numbers of vehicles with $\mathcal{V}_{vel} = 60kmph$.

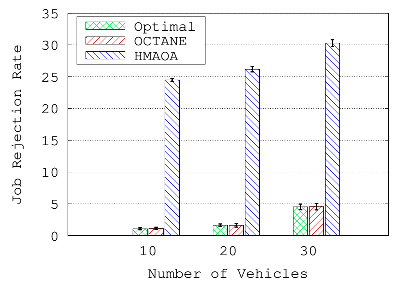

Fig. 4: Rejection Rate Vs numbers of vehicles with $\mathcal{V}_{vel} = 60kmph$.

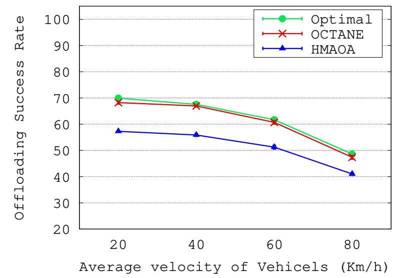

Fig. 5: OSR Vs velocity of vehicles for $\mathcal{V} = 30$.

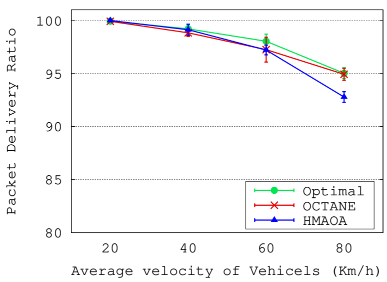

Fig. 6: PDR Vs velocity of vehicles for $\mathcal{V} = 30$.

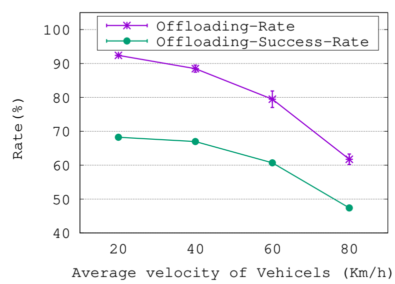

Fig. 7: OCTANE OSR and OR Vs velocity of vehicles for $\mathcal{V} = 30$.

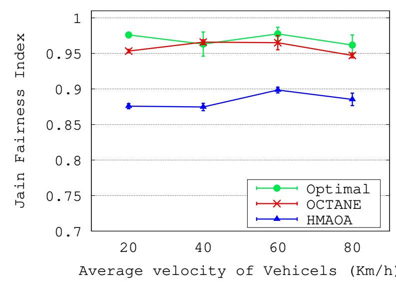

Fig. 8: JFI Vs velocity of vehicles for $\mathcal{V} = 30$.

unreliable. Due to high velocity and usage of UDP protocol for offloading the jobs, more packet loss happens while offloading jobs from the vehicles to the MEC server over 5G NR. As a result, the reassembly of jobs becomes increasingly difficult at the MEC server. Thus, OSR and OR rates deteriorate with increasing velocity. In Fig. 6, impact of velocity on different schemes is shown in terms of Packet Delivery Ratio (PDR). PDR drops with an increase in the velocity of vehicles, thereby decreasing the OSR. When the requested jobs are offloaded to the MEC server, MEC fails to complete some of the jobs within their deadlines. As a consequence, the difference in OR and OSR rates are visible in Fig. 7. Here, OCTANE offers OR and OSR of 79% and 60% respectively, for varying velocity from 20 to 80 kmph by keeping vehicle count fixed at 30. Under the effect of mobility, OCTANE gives fairness among vehicles. Fig. 8 shows offloading fairness among vehicles for different schemes. OCTANE, HMAOA, and ILP model have 0.95, 0.87 and 0.96 JFI for 30 vehicles by varying velocities from 20 to 80.

VI. Conclusions

In this paper, we studied the joint job offloading decision, radio resource allocation, and computational resource allocation problem for latency-sensitive vehicular applications in 5G NR based MEC system. We proposed OCTANE, which selects jobs for offloading by jointly considering deadlines, computational and communication delays of the jobs. Further, to provide fairness among vehicles, we implemented a TBS-based strategy for allocation of TDMA symbols in uplink. For the evaluation, we have used an HD-map application as a use case to study the effectiveness of OCTANE. The NS-3 simulations are performed for HD-map application in a highway scenario. The results show how OCTANE outperformed a state-of-art algorithm in terms of offloading success rate and other metrics.

References

[1] L. Yang, J. Cao, H. Cheng, and Y. Ji, "Multi-user computation partitioning for latency sensitive mobile cloud applications," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2253–2266, 2015.

[2] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.

[3] L. Yang, et al., "A framework for partitioning and execution of data stream applications in mobile cloud computing," vol. 40, no. 4, 2013.

[4] W. Zhan, et al., "Mobility-aware multi-user offloading optimization for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3341–3356, 2020.

[5] B. Dab, et al., "Joint optimization of offloading and resource allocation scheme for mobile edge computing," in *Proc. of IEEE WCNC*, 2019, pp. 1–7.

[6] L. Yang, et al., "Joint computation partitioning and resource allocation for latency sensitive applications in mobile edge clouds," in *Proc. of IEEE 10th CLOUD*, 2017, pp. 246–253.

[7] Y. Jang, et al., "Energy-efficient task offloading for vehicular edge computing: Joint optimization of offloading and bit allocation," in *Proc. of IEEE 91st VTC2020-Spring*, 2020, pp. 1–5.

[8] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.

[9] W. Anwar, et al., "Physical layer evaluation of v2x communications technologies: 5g nr-v2x, lte-v2x, ieee 802.11bd, and ieee 802.11p," in *Proc. of IEEE 90th VTC2019-Fall*, 2019, pp. 1–7.

[10] "3GPP TS 38.214, NR; Physical layer procedures for data." Tech. Rep.

[11] "3GPP TS 38.306, NR; User Equipment (UE) radio access capabilities." Tech. Rep.

[12] F. e. a. Jomrich, "Demo: rapid cellular network simulation framework for automotive scenarios (race framework)," in *Proc. of International Conference on Networked Systems (NetSys)*, 2017, pp. 1–2.

[13] P. Natale, et al., "An e2e simulator for 5g nr networks," *Simulation Modelling Practice and Theory*, vol. 96, p. 101933, 2019.

[14] "White paper: Operational behavior of a high definition mapapplication," Tech. Rep., May 2020.

[15] " Deliverable D2.1: 5GCAR scenarios use cases requirements and KPIs," Aug. 2017.

[16] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26 652–26 664, 2019.