Enhancing Uplink Scheduling in 5G Enabled Vehicular Networks: A Cross-Layer Approach with Predictive Buffer Status Reporting

Veerendra Kumar Gautam[$], Venkatarami Reddy Chintapalli[†], Bheemarjuna Reddy Tamma[$], and C. Siva Ram Murthy[⋆]
[$]Indian Institute of Technology Hyderabad, India, [†]National Institute of Technology Calicut, India
[⋆]Indian Institute of Technology Madras, India
e-mail:cs18resch01003@iith.ac.in, venkataramireddyc@nitc.ac.in, tbr@cse.iith.ac.in, murthy@iitm.ac.in

***Abstract*—Enabling widespread adoption of resource-intensive vehicular applications such as Extended Reality (XR) and High Definition map (HD Map) necessitates further enhancements in 5G, which is anticipated with 5G-Advanced. These applications, sensitive to latency, prompt researchers to propose offloading vehicles' complex computations to nearby edge clouds, aiming to minimize latency and meeting the Quality-of-Service (QoS) demands of these applications. However, the uncertainties arising from spatio-temporal factors due to vehicle mobility and the dynamic nature of application behaviour pose significant challenges in deciding the efficient offloading decision for minimizing latency. To tackle this challenge, this paper introduces a cross-layer framework that bridges the Radio Access Network (RAN) scheduler with the Mobile Edge Computing (MEC) scheduler. The proposed framework facilitates the exchange of vehicle ranks and channel condition information between schedulers, strategically aimed at reducing Head-Of-Line (HOL) delay for efficient computational offloading. Furthermore, the MAC layer incorporates the prediction of the Buffer Status Report (BSR) using Machine Learning (ML) to further reduce the queuing delay experienced by the offloading jobs of the vehicles in uplink. Simulation results using the NS-3 gym demonstrate that the proposed cross-layer framework achieves a higher Offloading Success Rate (OSR) than the state-of-the-art QoS scheduler by effectively reducing HOL delay for HD Map vehicular application.**

I. Introduction

VEHICULAR applications, including High-Definition map (HD Map), Augmented Reality (AR), Virtual Reality (VR), and services enabled by the Vehicle-to-Everything (V2X) network, play a pivotal role in reshaping traffic dynamics in 5G and beyond networks. This transformation involves a shift away from predominantly downlink (DL) traffic towards a more balanced distribution of DL / uplink (UL) traffic, with a Radio Access Network (RAN) delay below 1 *msec* [1]. Moreover, the advent of Mobile Edge Computing (MEC) technology has brought in computational capabilities closer to the Base Station (gNodeB), effectively bridging the gap between the limited computational resources of vehicle and the compute-intensive demands of vehicular applications. Offloading computationally intensive workloads (a.k.a. jobs) from vehicle to MEC server can enhance a vehicle's computational capabilities thereby reducing execution latency. However, the successful implementation of computational offloading to an MEC server critically hinges on the End-to-End (E2E) delay between MEC server and the vehicle.

In an effort to minimize the E2E delay, 3GPP has introduced an array of technologies within 5G New Radio (NR), including flexible numerology, Massive MIMO, Bandwidth Parts (BWPs), service multiplexing, and mini-slotting. Recent field tests of 5G NR in [2] indicate that the existing 5G infrastructure can adequately meet the fundamental requirements of vehicular applications. However, this study also underscores the need for further improvements to reduce E2E delay for wider adoption of vehicle applications and services. To address this need, 3GPP emphasizes the vital integration of vehicular application awareness within the MAC scheduler [3], which should encompass additional information about vehicular traffic, including application identifiers, Packet Data Units (PDUs) sets, and vehicle ranks, aimed at reducing E2E delay. In this context, a cross-layer mechanism was proposed for Extended Reality (XR) in [4].

The 5G NR RAN scheduler present at the gNodeB relies on Buffer Status Reports (BSRs) sent by the UEs for uplink scheduling. These BSRs convey details on the current RLC queue sizes at vehicles (UEs)[1] to the gNodeB so that RAN scheduler could allocate uplink radio resources among different contending UEs. Consequently, current BSR overlooks incoming data at UE, potentially resulting in increased Head-Of-Line (HOL) delay of packets in the uplink. Here, HOL refers to the phenomenon where specific packets encounter congestion or delay at the vehicle's Radio Link Control (RLC) queue, causing subsequent packets to be held back. To address this, the proposed solution involves predicting RLC queue size at the UE by leveraging historical BSR values at the gNodeB, which allows for allocating additional radio resources beyond what the current BSR indicate, and as shown in [5] it can reduce the overall HOL delay of applications deployed at the vehicles. However, to the best of the authors' knowledge, the influence of the combined effect of cross-layer optimization and BSR prediction on the E2E delay in vehicular environments has not been thoroughly investigated. Therefore, it is necessary

[1]Throughout this paper we use the terms vehicles and UEs interchangeably.

to examine cross-layer optimization and BSR prediction in vehicular networks to enhance the overall performance of vehicular applications.

In a vehicular environment, predicting BSR and employing a cross-layer-based scheduler approach pose challenges due to the high-speed nature of vehicles, resulting in rapid channel variations and frequent fluctuations in Signal-to-Interference-plus-Noise Ratio (SINR) values. In dynamic environments, accurate radio resource allocation for vehicles relies on crucial roles played by BSR predictions and cross-layer information, both of which are essential in minimizing E2E delay and enhancing Packet Delivery Ratio (PDR). Additionally, a careful adjustment of vehicle ranks based on cross-layer information, provided by vehicular application at appropriate intervals, becomes essential for better utilization of radio resources. The main contributions of this work are:

- We propose an inclusive cross-layer framework intended to facilitate seamless information exchange between the RAN scheduler and the MEC scheduler. This framework is designed to promote a cooperative system by enabling efficient communication and data sharing between the gNodeB and the MEC server used for offloading of jobs.
- We propose the utilization of a Bi-directional Long Short-Term Memory (Bi-LSTM) Machine Learning (ML) model to predict the RLC queue size of vehicles based on historical BSR messages received from the respective vehicles. The Bi-LSTM model is trained using the Berlin V2X dataset [6]. Integrating this model into the UL RAN scheduler decreases signaling overhead, specifically minimizing the transmission of BSR messages.

II. Background

This section provides the necessary background on 5G NR scheduling timings, grant-based UL scheduling, Radio resource scheduling and MEC scheduling algorithms.

A. Scheduling Timings and Processing Delays in 5G NR

- *K2 timer:* It serves to schedule the transmission of the Physical Uplink Shared Channel (PUSCH) after receiving an UL grant through the Physical Downlink Control Channel (PDCCH). It begins its countdown when a UE receives an UL grant on the PDCCH and initiates a transmission on the PUSCH. In simpler terms, K2 represents the number of time slots allocated by the gNodeB to a UE for the tasks of decoding the UL grant and transmitting UL data on the PUSCH during the specified scheduling opportunity.
- *L2L1 processing:* The duration referred to as the encoding delay corresponds to the time it takes for the gNodeB PHY/MAC layers to encode control and/or data channels. More precisely, it signifies the delay from the time the MAC layer obtains control/data from the RLC layer to the time that control/data are ready for transmission over the air [7].
- *Decode latency:* It represents the delay involved in acquiring data from the air by the PHY layer and making the data block available for processing at the MAC layer. In the context of UL, it occurs at the gNodeB.
- T_{in}: If gNodeB identifies a UE that remains inactive without transmitting or receiving packets for the duration of the inactivity timer (T_{in}), it terminates the connection.

B. Grant-based UL Procedure: Dynamic Scheduling

In 5G NR, UL transmission for UEs connected to the gNodeB via the Uu interface follows a grant-based approach, illustrated in Fig. 1, referred to as Mode-1 resource allocation. In this resource allocation mode, an UE requests UL radio resources from gNodeB when it has data in its RLC buffer and requires a Scheduling Grant (SG) to transmit the pending data. To initiate this process, the UE sends a Scheduling Request (SR) request to the gNodeB, typically when it needs the SG. The SR request is essentially a signal sent to gNodeB through PUCCH to establish communication. In response, gNodeB issues a minimal UL grant to the UE via PDCCH, where PDCCH carries a DCI (Downlink Control Information). This DCI includes essential details such as Modulation and Coding Scheme (MCS), Resource Block (RB) allocation, and HARQ configuration, which the UE uses for its initial UL data transmission. Simultaneously with its first UL transmission, the UE transmits a BSR. This BSR contains a quantized value representing the number of bytes pending in its Logical Channel Groups (LCGs). Subsequently, the gNodeB responds with SG messages to allocate an appropriate amount of UL radio resources, in terms of RBs, to the UE. The overall E2E delay in this process depends on factors such as the packet size, Transport Block Size (TBS) of the first UL scheduling assignment, which can result in either a 3-step process (SR → UL grant → UL data) or a 5-step process (SR → UL grant → UL data + BSR → UL grant → UL data). Furthermore, processing delays, including *L2L1 processing* and *decode latency*, along with the *K2 timer*, also contribute to latency.

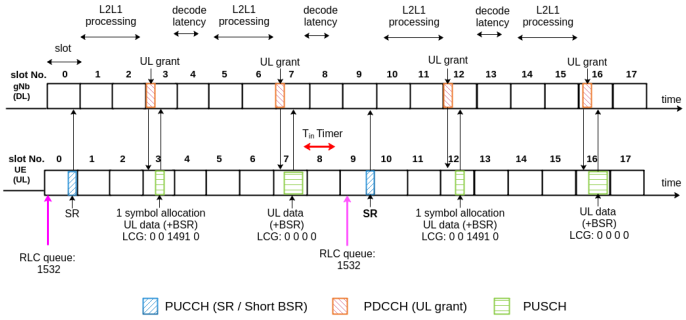


Figure 1: An instance of grant-based UL procedure under numerology 0 showing transmission of a two packets (1500 bytes with 32-byte header).

C. MEC scheduling algorithms and Radio resource scheduling

- *OCTANE* [8]: *OCTANE* is an online heuristic based MEC scheduling algorithm that takes into account job deadlines, job input sizes, computational resources of the MEC platform, and communication delays of vehicles when making job offloading decisions.

- *Proportional Fair (PF)*: The PF scheduling is a radio resource allocation strategy which is employed to strike a balance between optimizing overall system throughput and ensuring fairness among UEs. This balance is achieved by taking into account both the present channel conditions and the prior resource allocations to the UEs. The set of $\mathcal{V}$ UEs is represented by $\mathcal{V} = \{1,\ldots,v,\ldots,V\}$, with each UE indexed by $v \in \mathcal{V}$. The PF metric for UE v within a Transmission Time Interval (TTI) is represented as $PF_v(t)$ and is defined in Eqn. 1.

$$PF_v(t) = \left[\frac{T_v^{\alpha'}}{R_v^{\beta'}}\right] \tag{1}$$

  In this Eqn., $R_v^{\beta'}$ represents the historical average throughput of UE v, while $T_v^{\alpha'}$ signifies its instantaneous data rate. Parameters $0 \leq \alpha' \leq 1$ and $0 \leq \beta' \leq 1$ can be adjusted to strike a balance between optimizing throughput and ensuring fairness within the PF metric.
- *RETALIN* [9]: It uses a modified PF metric that takes the probability of SR and backlogs of UEs into account to decide the number of RBs to be assigned to different UEs for their UL transmissions in 5G NR. *RETALIN*, designed as a queue-aware radio resource scheduler, assesses the probability of SR for each UE by scrutinizing dynamic queue behavior. *RETALIN's* aim is to effectively manage backlogs, reducing the incidence of SR and mitigating the negative impact of numerology on UL traffic, thus reducing the E2E delay. The *RETALIN* considers the normalized backlog ratio (α_v), which is defined in Eqn. 2.

$$\alpha_v = \left[\frac{\eta_v}{\sum_{v=1}^{|\mathcal{V}|} \eta_v}\right] \tag{2}$$

  Where α_v represents a drift in the UL buffer length of UE v that is (η_v) as compared to the aggregated UL buffer length of all UEs in a TTI.
  Further, *RETALIN* also considers the probability of not generating an SR by a UE in a TTI, given by:

$$P_{v,NSR}(t) = [1 - P_{SR}] \tag{3}$$

  where P_{SR} is the probability of generating an SR in a TTI derived in [9]. *RETALIN* uses the utility metric $U_v(t)$ defined in Eqn. 4, which has three different components as shown below.

$$U_v(t) = \alpha_v \times P_{v,NSR}(t) \times PF_{v,r}(t) \tag{4}$$

  $U_v(t)$ is calculated for each vehicle v in every TTI. The vehicle with the highest value will be allocated radio resources first, followed by others in descending order.

III. System Model

We examine a typical scenario in a highway environment where $\mathcal{V}$ vehicles are within the coverage of a gNodeB, as illustrated in Fig. 2. All $\mathcal{V}$ vehicles are equipped with On-Board Units (OBUs) based on 5G NR, each with limited local computational capabilities. As depicted in Fig. 2, a MEC system enables vehicles connected to the gNodeB to augment their computational capabilities through job offloading. In this setup, we assume that the vehicles establish a connection with the MEC system over Uu interface of 5G NR. Vehicles run a V2X client application which generates jobs with specific timing constraints. These jobs consist of data packets with varying periodicity (i.e., Inter-Packet Arrival Time (IPAT)) with fixed data sizes. If the local computing capacity is inadequate or jobs are at risk of missing their deadlines, they are considered for offloading to the MEC system, incurring additional transmission delay. On the other hand, the MEC system runs a V2X server application which strives to maximize the number of successfully offloaded jobs, making real-time offloading decisions based on requests from the vehicles. However, it is important to note that the MEC system has its limitations in terms of computational resources. Hence, the MEC system must consider both the computational and channel conditions of a vehicle while making an offloading decision. For vehicles with poor channel conditions, more computational resources should be allocated to their jobs at the MEC system to ensure that the jobs meet their timing constraints as there is an extra RAN delay in offloading the jobs to the MEC system.

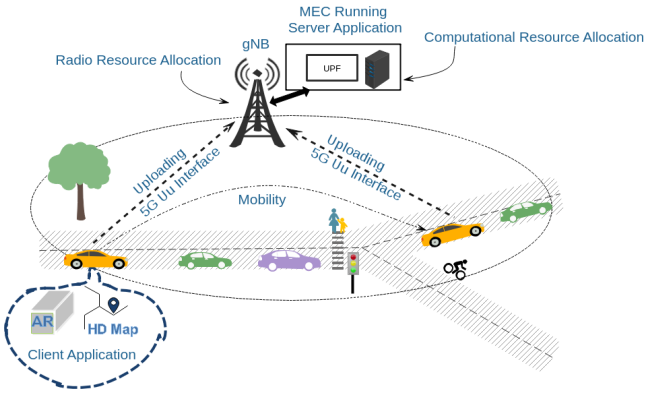


Figure 2: System model.

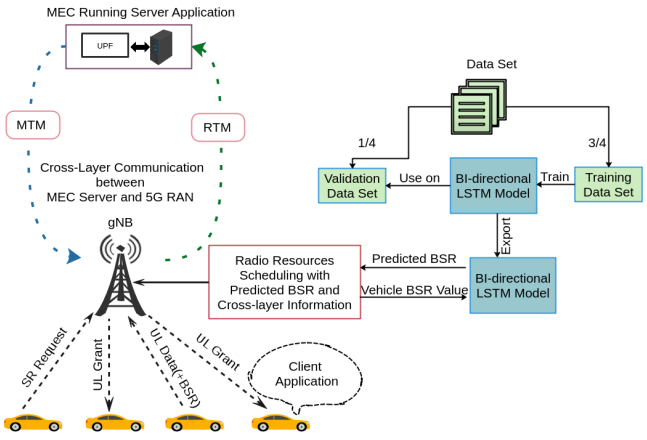


Figure 3: Cross-layer framework and BSR prediction.

IV. Cross-layer Framework and BSR Prediction in Vehicular Scenario

In this work, we propose a cross-layer framework that facilitates information exchange between the MEC scheduler and the RAN scheduler for efficient allocation of radio and computation resources in vehicular environments, as shown in Fig. 3. The proposed cross-layer framework aims to minimize the HOL delay for job offloading by fostering collaboration between the two schedulers through message exchanges at predefined intervals. The RAN scheduler provides essential parameters (i.e., MCS and transmission rates) of the UE to the MEC scheduler, enabling informed decisions about job selection by the MEC scheduler. Simultaneously, the MEC scheduler offers ranking information for vehicle selection, enhancing radio resource allocation by the RAN scheduler. The process involves the MEC scheduler selecting jobs from the pool of offloading requests based on job size, deadline, and data rate of the UE—calculated using Modulation and MCS values provided by the RAN scheduler through the RAN Trigger Message (RTM). Subsequently, responses are sent to selected vehicles to initiate job offloading. Following this, the vehicles transmit the required data to execute these offloaded jobs at the MEC system. The MEC scheduler then notifies the RAN scheduler through the MEC Trigger Message (MTM), which includes the ranking and Radio Network Temporary Identifier (RNTI) values of the selected vehicles for offloading. Upon receiving the MTM, the RAN scheduler prioritizes vehicles for data offloading. Both schedulers trigger each other by exchanging MTM and RTM messages, detailed in Fig. 3. The core of this cross-layer design lies in the cooperative nature of the schedulers. Their collaboration extends beyond mere message-based notifications and encompasses strategies for adjusting rankings, where changes by one scheduler can impact the other's performance. By strategically designing these adjustment mechanisms, the cross-layer approach aims to balance radio and edge cloud resources, ultimately reducing job queuing delay.

In general, RAN scheduler relies on current BSR values indicated by the UE for assigning uplink radio resources. The current BSR does not encompass information on the arrival of new data after BSR transmission. As a result, the RAN scheduler allocates resources without considering newly arrived data, leading to increased HOL delay, signaling overhead, and thereby increasing overall E2E delay. To reduce HOL delay, we leverage ML techniques, specifically a Bi-LSTM model trained on real-world measurements [6] to anticipate UL traffic. The deployed model learns from traffic patterns and evolving channel conditions. Subsequently, the RAN scheduler proactively allocates more resources than indicated by the BSRs in the UL grant, preempting the need for SR and ensuring timely grants before the data arrives. These grants adapt to changing channel conditions using MCS and allocate sufficient radio resource blocks while upholding QoS requirements. This proactive approach ensures that when a UE is ready to transmit a lot of pending data, it already possesses a grant, significantly reducing HOL delay and associated E2E delay.

Prediction of BSR can occur in three ways: conservative (under prediction), precise (right prediction), or aggressive (over prediction), as shown in Fig. 4. The gNodeB performs the BSR prediction for the subsequent cycle using the ML model chosen. The predicted $\widehat{B}$ value represents an additional grant provided to the vehicle, which could either be less than, exactly equal to, or more than the current UE buffer level. Over allocation of radio resources leads to wastage if the prediction exceeds the actual UE buffer capacity, while excessive grants may result in the RLC buffer being emptied unnecessarily. Conversely, reducing the grant to match or be less than the UE buffer level can significantly increase the HOL delay. Accurate BSR predictions pose a challenge, but the ML model aims to forecast values equal to or less than the UE buffer level, ensuring high spectral efficiency while reducing HOL delay.

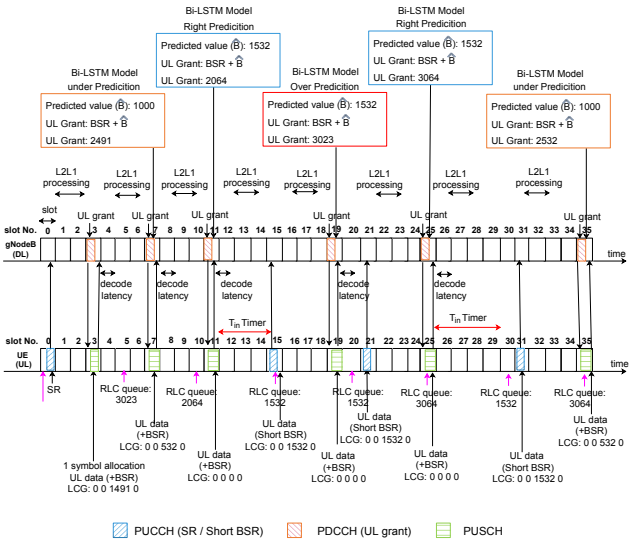


Figure 4: Grant-based UL procedure with BSR prediction in case of numerology 1 for packets of size 1500 bytes (with 32 bytes of header) with $IPAT = 5$ slots and $T_{in} = 5$ slots.

V. Cross-Layer Scheduling Algorithms

When a vehicle initiates an application, the vehicle sends a job request to the MEC server. The MEC scheduler then evaluates whether to execute the requested job on the MEC server or not, as outlined in Algorithm 1. The MEC scheduling algorithm starts by receiving job requests from all vehicles, along with their respective MCS values (i.e., use to calculate data rate of vehicles). Initially, the algorithm initializes an empty ranking set R_V for each vehicle v within the set of vehicles $\mathcal{V}$, denoted as $R_V = \emptyset$. Upon receiving the RTM message, the MCS values MCS_V are extracted using `extractMCSofVehicles()` (line 2). Subsequently, by calling the `calculateRanking()` procedure, which utilizes the *OCTANE* algorithm [8], the scheduler selects jobs for execution (line 3). The algorithm iterates through all vehicles and their respective selected jobs to compute the total size of jobs (*i.e.,* $JobSizeTotal$) to be offloaded from the vehicles to the MEC server, as well as the total size of jobs for each

vehicle (*i.e.,* $JobSize_V$) (lines 11 - 16). Using $JobSizeTotal$ and $JobSize_V$, the algorithm calculates a ranking R_V for each vehicle and communicates this ranking information to the RAN scheduler using the message MTM (lines 18-22). Here, ranking is calculated based on the amount of data to be offloaded to the MEC server. Higher rankings are assigned to vehicles that need to offload less data to the MEC server. Following this, the algorithm waits for new job requests to be received.

We modified the utility metric $U_v(t)$, as defined in Eqn. 4 of *RETALIN*, which is the PF variant. This modification involves incorporating the ranks of UEs, denoted as R_v, utilized for UL scheduling, sent by the MEC scheduler using MTM message. The modified utility metric, $U'_v(t)$, comprises four distinct components, as detailed below.

$$U_v(t) = \left[\frac{\alpha_v \times P_{v,NSR}(t) \times PF_v(t)}{R_v}\right] \quad (5)$$

Algorithm 1 Scheduler running at MEC Server

inputs: $V_{\text{allJobs}} = \{J_1, J_2, \ldots, J_V\}$, $J_v = \{j_1, j_2, \ldots, j_n\}, (v \in \mathcal{V})$, $V_{\text{SelectedJobs}} = \{\}, R_V = \{\}$

1 **if** *receive the message RTM* **then**
2 $MCS_V \leftarrow$ `extractMCSofVehicles`(RTM)
3 `calculateRanking`($V_{\text{allJobs}}, MCS_V$)
4 **else**
5 **if** *new job request is received* **then**
6 `calculateRanking`($V_{\text{allJobs}}, MCS_V$)
7 **else**
8 GOTO line 1
9 **end**
10 **end**
procedure calculateRanking($V_{\text{allJobs}}, MCS_V$)
// OCTANE selects the jobs to be offloaded
11 $V_{\text{SelectedJobs}} \leftarrow$ `OCTANE`($V_{\text{allJobs}}, MCS_V$)
12 $JobSizeTotal \leftarrow 0$, $JobSize_V = \{\}$
13 **forall** $i \in V_{SelectedJobs}$ **do**
14 $JobSize_v \leftarrow$ `sumOfJobs`(i)
15 $JobSizeTotal \leftarrow JobSizeTotal + JobSize_v$
16 $JobSize_V \leftarrow JobSize_V \cup JobSize_v$
17 **end**
18 **forall** $v \in \mathcal{V}$ **do**
19 $R_v \leftarrow \left[\frac{JobSizeTotal - JobSize_v}{JobSizeTotal}\right]$
20 $R_V \leftarrow R_V \cup R_v$
21 **end**
22 Send *MTM* message with R_V to the RAN scheduler
end procedure

The RAN scheduling algorithm employs historical BSR data from vehicles as input for a Bi-LSTM model to predict the future queue size of the UE. The algorithm also uses the ranking information provided by the MEC scheduler to allocate radio resources, as described in the Algorithm 2. The algorithm initializes with empty sets for predicted queue size and rankings for each UE v within the set $\mathcal{V}$, denoted as $QueueSize_v^{pred} = \emptyset$ and $R_v = \emptyset$, respectively. Upon receiving the MTM message, the algorithm extracts the ranking of each UE v into R_v using `extractRanksOfVehicles()` (line 2). If there are no updates in the information, the algorithm retains the previous rankings of UEs. The algorithm calls `callRETALIN()` procedure to predict the BSR and to allocate radio resources using `RETALIN()` scheduler (line 3). Afterward, the algorithm sends an RTM message containing MCS values of vehicles to the MEC scheduler (line 4). `callRETALIN()` procedure iterates through all UEs in V_{allBSR}, where each UE $v_i = \{bsr_1, bsr_2, \ldots, bsr_{Bi-LSTM_w}\}$ contains previous BSR values ($Bi-LSTM_w$ represents the window size). For every UE v in V_{allBSR}, the algorithm predicts the queue size using the Bi-LSTM model (`Bi-LSTMpredict()`) and stores these predictions in $QueueSize_v^{pred}$ (lines 7 - 9). Finally, the algorithm calculates the number of RBs for radio resource allocation (`RETALIN()`) [9] for the vehicle v, using the predicted BSR values in BSR_v and the rankings of vehicles R_V (line 10).

Algorithm 2 Scheduler running at gNodeB

inputs: $V_{\text{allBSR}} = \{BSR_1, BSR_2, \ldots, BSR_v, \ldots, BSR_V\}$, $BSR_v = \{bsr_1, bsr_2, \ldots, bsr_{Bi-LSTM_w}\}, (v \in \mathcal{V})$, $R_V = \{\}$, $QueueSize_v^{pred} = \{\}, (\text{v} \in \mathcal{V})$

1 **if** *receive the message MTM* **then**
2 $R_V \leftarrow$ `extractRanksOfVehicles`(MTM)
3 $MCS_V \leftarrow$ `callRETALIN`(V_{allBSR})
4 Send *RTM* message with MCS_V to the MEC scheduler
5 **else**
6 `callRETALIN`(V_{allBSR})
procedure callRETALIN(V_{allBSR})
// Predict BSR values using Bi-LSTM model
7 **forall** $v \in V_{allBSR}$ **do**
8 $\widehat{B} \leftarrow$ `Bi-LSTMpredict`(BSR_v)
9 $QueueSize_v^{pred} \leftarrow (bsr_{Bi-LSTM_w} + \widehat{B})$
10 $MCS_V \leftarrow$ `RETALIN`(R_v, $QueueSize_v^{pred}$)
11 *Return* MCS_V
end procedure

VI. Simulation And Performance Evaluation

We created a simulated scenario, where each vehicle generates various tasks related to HD Map, such as sensor data collection, sensor data analysis, and HD Map updates. Sensor data analysis, being computationally intensive, requires offloading to the MEC server through the 5G NR network. To facilitate this, we developed a job offloading application called *UdpOffloading*, built upon NS-3's Udp-Client-Server application. The client application can be configured to generate jobs with different characteristics, including input sizes, deadlines, and CPU cycle requirements, at specified intervals. The deadlines of the jobs are configured according *OCTANE* [8], while other parameters are set according to *RETALIN* [9], as detailed in Table I. Configurable parameters of the Bi-LSTM model include feature dimensions, timestamps, lead-time, and the learning rate (η). We chose the default value of 0.01 for the learning rate η, as specified in the Keras library [10]. After extensive experimentation, we determined that the ideal number of training epochs for the Bi-LSTM model is 100. We

considered the Berlin V2X dataset [6] and split the dataset into two parts for training and testing, allocating one-fourth of the data for testing purposes. The Bi-LSTM model has been integrated as a gym with the NS3-gym interface. Subsequently, we developed a MAC scheduler interface based on OpenAI gym. The evaluation is carried out within a highway scenario, specifically utilizing road segments sourced from Winnipeg, Canada, that encompass a 250-meter stretch of the Pembina Canada Highway. To simulate customized vehicular traffic, we employed the RACE [11]. RACE utilizes SUMO and OpenstreetMap to replicate realistic vehicle traffic patterns. For real data on cellular infrastructure, we utilized datasets provided by the Canadian Organization for Innovation, Science, and Economic Development (ISED) throughout our evaluation, each simulation is repeated with 10 different random seeds, and the results are presented with 95% confidence intervals.

Table I: Simulation Parameters

Parameter	**Value**
Scenario	Urban Macro Cell
Number of Vehicles $\lvert\mathcal{V}\rvert$	30
Mobility Model	Krauss
Average Vehicle Velocity ($\mathcal{V}_{vel}$)	60 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, Line-Of-Sight
Channel Bandwidth	30 MHz
5G NR Numerology μ	0, 1, 2
Channel model	UMa_LoS
MEC Task scheduler	*OCTANE* [8]
5G NR MAC Scheduler	*RETALIN* [9], PF QoS-aware [12]
5G QoS Identifier (5QI)	75, GBR_V2X
$Bi-LSTM_w$	50
Packet size (L)	1000 Bytes
Job generation per vehicle	0.1 sec

A. Comparison Schemes and Performance Metrics

1) Comparison Schemes: In order to test the performance of our proposed algorithm, we compare it with the following state-of-the-art and baseline radio resource scheduling schemes and MEC scheduling algorithms.

- *OCTANE [8] + PF*: At the MEC server, *OCTANE* operates as an MEC scheduling algorithm. Meanwhile, the RAN scheduler employs the PF scheduling strategy.
- *OCTANE [8]+ QoS-aware scheduler [12]*: At the MEC server, *OCTANE* operates, while a state-of-the-art QoS-aware scheduler functions as the RAN scheduler, as outlined in [12]. QoS-aware scheduler introduces a delay budget factor (D) that represents the weight sensitive to delay, considering HOL delay and Packet Delay Budget (PDB). The calculation for this factor is given as D = PDB / (PDB - HOL). Further, the QoS-aware scheduler utilizes multiple factors including the default priority level of the flow, the PF metric and D to allocate the radio resources.
- *OCTANE+ RETALIN (cross-layer)*: At the MEC server, *OCTANE* is operational, while modified *RETALIN* [9] functions as a RAN scheduler. *OCTANE* and *RETALIN* engage in cross-layer communication using MTM and RTM messages, with *RETALIN* utilizing predicted RLC queue value to reduce HOL delay.

The following performance metrics are used to evaluate the performance of the proposed scheme.

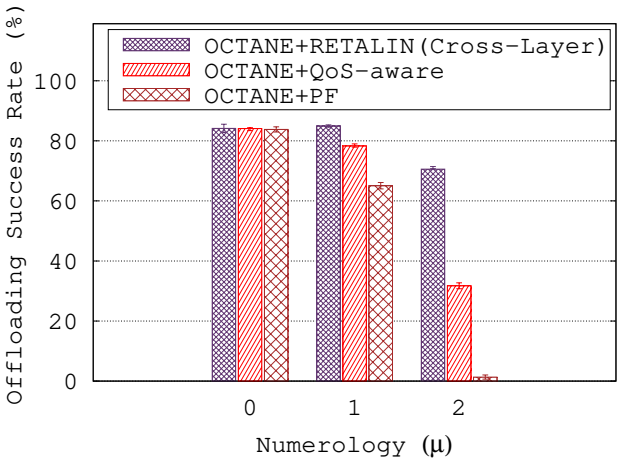
- *Offloading Success Rate (OSR)*: A job is considered successful if the job is offloaded to the MEC server and completed within its designated deadline. OSR is calculated as the ratio between the number of jobs successfully executed by the MEC server and the total number of job offload requests received by the MEC server.
- *HOL delay*: It represents the delay experienced by a packet at the head of a queue waiting to be transmitted. HOL delay is a measure of the time a packet spends in a UE queue before it is transmitted.
- (%) of SR (SR_p): It represents the ratio of SRs, such as the control or signaling overhead, to the total number of packets exchanged within the network, expressed as a percentage.
- BSR_{avg}: It is the average number of BSR messages used to transmit a packet in the network.

B. Performance Results

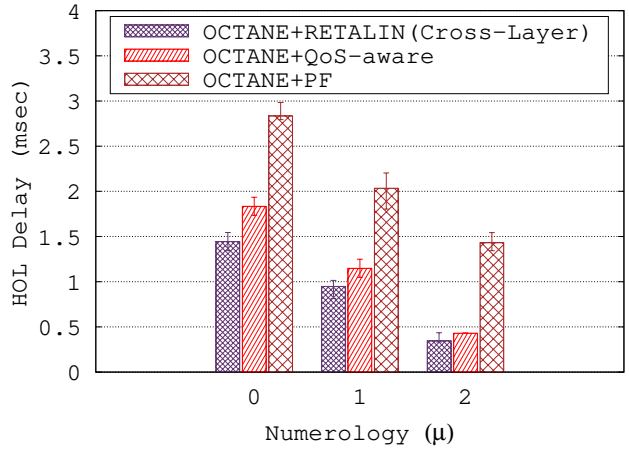
In Fig. 5(a), the OSR variation is depicted for *OCTANE+PF*, *OCTANE+QoS-aware* and *OCTANE+RETALIN (cross-layer)* across numerologies, setting $\mathcal{V}_{speed} = 60\ kmph$ and $L = 1000$ bytes. With an increase in numerology from $\mu = 0$ to $\mu = 2$, a consistent decrease in OSR is observed. Here, *OCTANE+PF*, *OCTANE+QoS-aware* and *OCTANE+RETALIN (cross-layer)* have OSR of 83%, 64%, 2% and 84%, 78%, 31% and 84%, 84%, 70% in case of $\mu = 0$, $\mu = 1$, $\mu = 2$, respectively. The declining OSR with higher numerologies indicates an increasing number of jobs failing to meet the deadline of the HD Map application. *OCTANE+RETALIN (cross-layer)* notably enhances the OSR by 19% over the state-of-the-art QoS-aware scheduler for HD Map applications. Additionally, *OCTANE+RETALIN (cross-layer)* experiences less HOL delay compared to *OCTANE+PF* and *OCTANE+QoS-aware* as shown in Fig. 5(b), due to cross-layer information exchange facilitated by RTM and MTM messages, which improves OSR. The results shown in Fig. 6(a) and Fig. 6(b) clearly indicate that *OCTANE+RETALIN (cross-layer)* is capable of achieving a better trade-off between SR and BSR for higher numerologies for the application of the HD Map. However, the OSR is low for $\mu = 2$ compared to $\mu = 1$ for all schemes due to the increase in packet fragmentation in $\mu = 2$, attributed to the reduced slot time.

VII. Conclusions

This work presented a novel cross-layer framework that facilitated information exchange between RAN and MEC schedulers, leveraging ranking and channel condition data for efficient task offloading in case of V2X applications. The utilization of a Bi-directional LSTM, trained on the Berlin V2X dataset, enhanced the model's capability to learn traffic inter-arrival patterns and predict future UL grants, assisting the RAN

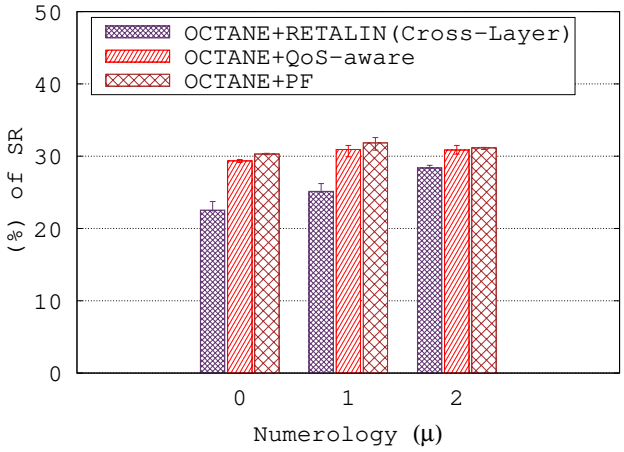


((a)) OSR.

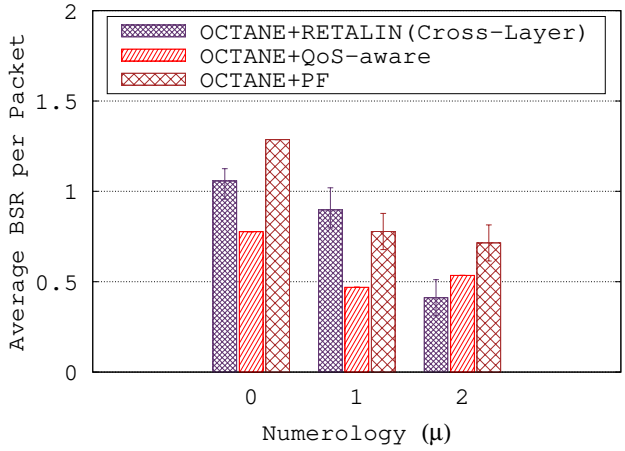


((b)) HOL Delay.

Figure 5: Result observed for HD Map application by varying numerology for $V = 30$ with $\mathcal{V}_{speed} = 60$ $kmph$ where $L = 1000$ bytes and $Bi - LSTM_w = 50$.



((a)) (%) of SR (SR_p).



((b)) BSR_{avg}.

Figure 6: Result observed for HD Map application by varying numerology for $V = 30$ with $\mathcal{V}_{speed} = 60$ $kmph$ where $L = 1000$ bytes and $Bi - LSTM_w = 50$.

scheduler. The integration of this cross-layer framework and RLC queue prediction proved particularly impactful in HD Map applications. In the context of the cross-layer framework, where RETALIN (our previous RAN scheduler work) and OCTANE (our previous solution for task offloading to an MEC server) work together, it indicates an increased offloading success rate of 19% and a reduction of 25% in HOL delay compared to state-of-the-art QoS-aware scheduler.

ACKNOWLEDGEMENT

This research work was supported by the Science and Engineering Research Board, New Delhi, India. Grant number: JBR/2021/000005.

REFERENCES

[1] E. TR, "5G: Study On Scenarios and Requirements for Next Generation Access Technologies (3GPP TR 38.913 Version 14.2. 0 Release 14)," *ETSI TR 138 913*, 2017.

[2] P. Pérez, D. Corregidor, E. Garrido, I. Benito, E. González-Sosa, J. Cabrera, D. Berjón, C. Díaz, F. Morán, N. García, J. Igual, and J. Ruiz, "Live free-viewpoint video in immersive media production over 5g networks," *IEEE Transactions on Broadcasting*, vol. 68, no. 2, pp. 439–450, 2022.

[3] "3GPP TR," *Study on XR enhancements for NR, Release 18, v0.0.1 ed., Abr.*, 2022.

[4] B. Bojović, S. Lagén, K. Koutlia, X. Zhang, P. Wang, and L. Yu, "Enhancing 5g qos management for xr traffic through xr loopback mechanism," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 6, pp. 1772–1786, 2023.

[5] K. Boutiba, M. Bagaa, and A. Ksentini, "On using deep reinforcement learning to reduce uplink latency for urllc services," in *Proc. of IEEE GLOBECOM Conference*, 2022.

[6] R. Hernangómez *et al.*, "Berlin V2X: A machine learning dataset from multiple vehicles and radio access technologies," *arXiv preprint arXiv:2212.10343*, 2022.

[7] N. Patriciello, S. Lagen, L. Giupponi, and B. Bojovic, "5g new radio numerologies and their impact on the end-to-end latency," in *Proc. of IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2018.

[8] V. K. Gautam, C. Tompe, B. R. Tamma, and A. F. A, "Octane: A joint computation offloading and resource allocation scheme for mec assisted 5g nr vehicular networks," in *Proc. of IEEE ANTS*, 2021.

[9] V. K. Gautam and B. R. Tamma, "Retalin: A queue aware uplink scheduling scheme for reducing scheduling signaling overhead in 5g nr," *IEEE Access*, vol. 12, pp. 16 632–16 651, 2024.

[10] F. Chollet *et al.*, "Keras. received from https://keras. io," 2018.

[11] F. e. a. Jomrich, "Demo: rapid cellular network simulation framework for automotive scenarios (race framework)," in *Proc. of NetSys*, 2017.

[12] K. Koutlia, B. Bojovic, S. Lagén, X. Zhang, P. Wang, and J. Liu, "System analysis of QoS schedulers for XR traffic in 5G NR," *Simulation Modelling Practice and Theory*, vol. 125, p. 102745, 2023.