



# A Vehicle-to-Infrastructure Data Offloading Scheme for Vehicular Networks with QoS Provisioning

Yasir Saleem, Nathalie Mitton, Valeria Loscrì

## ► To cite this version:

Yasir Saleem, Nathalie Mitton, Valeria Loscrì. A Vehicle-to-Infrastructure Data Offloading Scheme for Vehicular Networks with QoS Provisioning. IWCMC 2021 - 17th International Wireless Communications and Mobile Computing Conference, Jun 2021, Harbin / Virtual, China. hal-03188360

**HAL Id: hal-03188360**

**<https://hal.inria.fr/hal-03188360>**

Submitted on 1 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Vehicle-to-Infrastructure Data Offloading Scheme for Vehicular Networks with QoS Provisioning

Yasir Saleem, Nathalie Mitton, Valeria Loscri  
Inria Lille – Nord Europe, France  
{yasir-saleem.shaikh, nathalie.mitton, valeria.loscri}@inria.fr

**Abstract**—In vehicular networks, vehicles carry various types of data that need to be offloaded to the RoadSide Units (RSUs) through Vehicle-to-Infrastructure (V2I) communications when vehicles come into their coverage. Since, RSUs are not widely deployed, vehicles have intermittent connectivity with RSUs. The data that vehicles carry to offload could be urgent data (such as accident data of nearby incident or emergency health data) that needs to be offloaded to the RSUs as soon as possible. Therefore, the consideration of Quality of Service (QoS) provisioning is imperative for data offloading in vehicular networks. In this paper, we propose V2I-Q, a V2I data offloading scheme with QoS provisioning by using three QoS functions: traffic classification, overload control and admission control. Traffic classification organizes the data into three priorities: high, medium and low. Overload control avoids overloading the RSUs to enable it to receive high priority data as soon as possible. Admission control allows RSUs to stop servicing existing vehicles offloading low priority data in order to receive high priority data from other vehicles. The performance evaluation shows that V2I-Q is able to offload more high priority data by incurring lower maximum offloading delay as compared to the traditional V2I data offloading schemes.

**Index Terms**—Data offloading, Road-side Unit (RSU), Quality of Service (QoS) provisioning, vehicular network; Vehicle-to-Infrastructure (V2I) Communication.

## I. INTRODUCTION

The vehicles in vehicular networks carry various types of data, such as accident data of nearby incident, emergency health data of patients, road traffic conditions, period health monitoring data, entertainment, backup data etc. Such data needs to be offloaded to RoadSide Units (RSUs) through Vehicle-to-Infrastructure (V2I) communications (i.e., V2I data offloading) when vehicles come into their coverage so that RSUs, which are equipped with edge servers, can analyze and process the data for taking required actions or forward the data to the cloud. The RSUs have been developed for more than ten years, however they are still not widely deployed [1]. Therefore, there is an intermittent RSUs connectivity and a vehicle cannot always be in the coverage of RSUs.

Some data can be very urgent and needs to be offloaded to RSUs as soon as possible. However, without traffic classification, there might be substantial delay in offloading such data to RSUs. In fact, the traffic classification alone is not sufficient because RSU might be overloaded and could not take more data. Therefore, the consideration of Quality of Service (QoS) is an important parameter to consider for data offloading in vehicular networks. There are various QoS considerations. For instance, since a vehicle carries various types of data, so

which data to offload first, how to avoid RSU from getting overloaded for smooth data offloading by vehicles and how to offload urgent data if RSU is overloaded.

In this paper, we propose V2I-Q, an efficient V2I data offloading scheme for vehicular networks with QoS provisioning. The main objective is to offload vehicles' data to RSUs as soon as possible. V2I-Q uses three QoS functions: 1) Traffic classification organizes the data into three priorities: high, medium and low; 2) Overload control tries to avoid overloading the RSUs by defining threshold values of maximum allowed load at RSUs for medium and low priority data; 3) Admission control operates when the load at RSU reaches the maximum capacity. In such a scenario, RSU can stop servicing vehicles that are offloading low or medium priority data in order to let a new vehicle offload high priority data. V2I-Q also models the connectivity of vehicles with RSUs to decide the amount of data to grant to vehicles for data offloading. To the best of our knowledge, there does not exist any V2I data offloading scheme in the literature that takes such QoS considerations into account.

The rest of the paper is organized as follows. Section II presents the related work. Section III presents the system model. Section IV presents the connectivity modeling. Section V presents V2I-Q, our proposed solution. Section VI presents performance evaluation, results and discussion. Finally, section VII concludes this paper.

## II. RELATED WORK

This section presents the state-of-the-art related work and discusses how we are different and novel from them. Offloading in vehicular networks have mainly been categorized into data offloading and task offloading. Most of the research works focus on task offloading in vehicular networks, however there are few studies that focus on data offloading in vehicular networks. Since our focus is on data offloading in this article, we only cover the works related to data offloading.

Song *et al.* [2] focused on V2V data transmissions under the problematic scenario of traffic hole. The authors rely on RSUs for data transmissions and proposed RSU-Assisted Backward Delivery with two methods: backward data relaying among vehicles and data relaying by RSUs. The authors evaluated their trade-offs and performance through simulations in NS-2 and SUMO using Taxi-ROMA dataset. However, firstly, this is a routing problem of data delivery and secondly, it is mainly focused on backward data delivery under the traffic

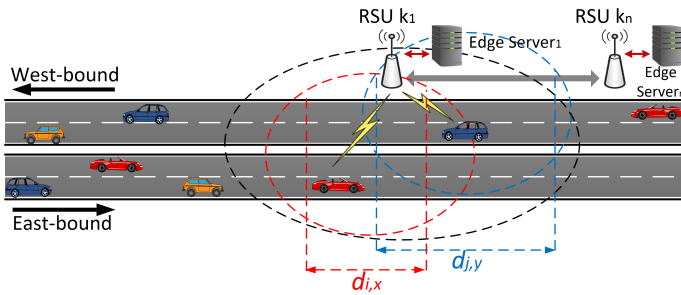


Fig. 1: System model.

hole problem, while we are mainly focusing on V2I data offloading from vehicles for RSUs.

Huang *et al.* [1] investigated data offloading by constructing multihop V2V paths connected towards RSUs and called it V2V2I offloading. A vehicle generally uses cellular network for data offloading. However, when it comes into the coverage of a vehicle (offloading agent) that has a connected  $k$ -hop V2V path towards an RSU, it uses this V2V path for data offloading. The authors studied the selection of offloading agent, the construction of  $k$ -hop V2V path using Mobile Edge Computing (MEC) and path repairing if the  $k$ -hop V2V2I offloading path is disconnected. However, this work does not consider the QoS provisioning and moreover, our focus is only on V2I data offloading.

Xiangming *et al.* [3] investigated the integration of cellular and opportunistic vehicular networks by considering the contact duration of vehicles for mobile data offloading. They proposed a mathematical framework for contact-aware optimal resource allocation offloading scheme that was formulated as an utility maximization problem under limited storage constraints of vehicles. The authors evaluated the performance in MATLAB by using Shanghai and Beijing vehicular mobility traces and compared with other schemes. This work is about data offloading from cellular network to vehicles in which the vehicles further disseminate the data to other vehicles in an opportunistic manner. However, we focus on V2I data offloading from vehicles using Dedicated Short-Range Communications (DSRC) technology.

Sun *et al.* [4] studied the cooperative downloading mechanism of online video content in vehicular networks as an optimization problem. This work is about video data offloading from cellular network while we are focused on data offloading from vehicles to RSUs. In [5], the authors propose Smart Ranking based Data Offloading (SRDO) algorithm for selecting an RSU and to improve the Quality of Service by relying on Software Defined Network (SDN) controller. Anyway, in our work, we do not rely on any centralized entity.

### III. SYSTEM MODEL

We consider a vehicular network, as presented in Fig. 1, having  $\mathcal{K} = \{1, 2, \dots, K\}$  RSUs and  $\mathcal{N} = \{1, 2, \dots, N\}$  vehicles such that  $\mathcal{K} < \mathcal{N}$ . The network has intermittent connectivity in which sometimes, a vehicle  $i \in \mathcal{N}$  can have

connectivity with an RSU  $k \in \mathcal{K}$ , however, sometimes, a vehicle enters an area with very low or no connectivity with RSUs. The considered scenario is a highway scenario with vehicles on east-bound and west-bound roads having multiple lanes on each direction. A vehicle carries various types of data for offloading to RSU. The vehicles move at a variable speed. The vehicles communicate with RSUs using Dedicated Short-Range Communication (DSRC) that is based on IEEE 802.11p standard [6] and offload their data to RSUs using V2I communications (i.e., V2I data offloading).

A vehicle can be connected maximum to one RSU at once. The connectivity of vehicles with RSUs varies from lane to lane as illustrated in the system model (e.g., the red Vehicle  $i$  moving on lane  $x$  is farther from the RSU  $k_1$  and has shorter connectivity than the blue Vehicle  $j$  moving on lane  $y$  nearer to the RSU  $k_1$ ). We assume all vehicles have the same coverage area, while RSUs have higher coverage area. Each RSU is connected with an edge server having storage and processing capability and connected through broadband connection. The RSUs collect data from vehicles and send this data to their connected edge server for storage/processing.

### IV. CONNECTIVITY MODELING AND CHARACTERIZATION

In this section, we model the connectivity time of a vehicle with the RSU and the offloading capacity.

#### A. Connectivity Time with RSU

The connectivity time estimates how long a vehicle  $i$  stays connected to an RSU (e.g., the duration while vehicle  $i$  is in RSU coverage area) and it is used in the calculation of how much data an RSU should grant  $i$  to offload. When  $i$  sends a data offloading request to RSU  $k$  with the amount of each type of data it wants to offload, its speed and data rate, as a first step, the RSU calculates the connectivity time with  $i$  before analyzing the data offloading request as follows:

$$T_{i,k}^c = \frac{d_{i,x}}{V_i} - \tau_{i,k} - T_{i,v2i}^{wait} \quad (1)$$

where  $d_{i,x}$  is the coverage area of Vehicle  $i$  with RSU  $k$  at lane  $x$  on which  $i$  is moving,  $V_i$  is the speed of  $i$ ,  $\tau_{i,k}$  is the registration time of  $i$  with RSU  $k$  before sending data offloading request and  $T_{i,v2i}^{wait}$  is the waiting time for  $i$  to get a reply after sending data offloading request to RSU  $k$ .

#### B. Offloading Capacity

Offloading capacity  $S_{i,k}^{max}$  is the maximum allowed data size that a vehicle  $i$  can offload to RSU  $k$ . It is directly proportional to the connectivity time  $T_{i,k}^c$ . The higher  $T_{i,k}^c$ , the higher the offloading capacity. It is a metric for limiting the size of data offloading and is calculated as:

$$S_{i,k}^{max} = T_{i,k}^c \times \min(r_i, r_k) \quad (2)$$

where  $r_i$  is the data rate of  $i$  and  $r_k$  is the data rate of RSU  $k$ . Note that vehicles might use different devices that support different data rates. For example, it can happen that Vehicle  $i$  supports higher data rate than RSU  $k$ , which can cause data loss. Therefore, we take the minimum data rate  $\min(r_i, r_k)$  between  $i$  and  $k$  for smooth data transfer.

## V. V2I-Q: OUR PROPOSED SCHEME

### A. Overview

A vehicle carries data featuring different  $\mathcal{P} = \{high, medium, low\}$  priorities that needs to be offloaded with respect to it to RSU with QoS provisioning, with overload control at RSU (see Section V-B2) and with admission control at RSU (see Section V-B3).

When a vehicle  $i$  has connectivity with RSU  $k$ , it first needs to obtain the permission grant from  $k$  by sending a data offloading request for each type data in order to ensure that higher priority data gets offloaded before lower priority data (see Section V-C). Subsequently,  $k$  processes the data offloading request and decides whether and how much data to grant and sends data offloading reply back to  $i$  (see Section V-D). Finally, on permission granted,  $i$  starts offloading its data (see Section V-E).

### B. Quality of Service (QoS) Provisioning

The QoS can be provisioned through various functions [7]. In this work, we use three functions: traffic classification, overload control and admission control for QoS provisioning.

1) *Traffic classification*: Data is classified into three priorities: high (such as accident and health emergency), medium (standard data, such as traffic conditions, periodic health monitoring), and low (delay-tolerant data, such as advertisement or updates). A vehicle first offloads high priority data, then the medium priority data and finally the low priority data. If it has multiple data with the same priority, it uses a greedy approach and first offloads the data having the largest size.

2) *Overload control*: We use overload control in order not to overload the RSUs and keep their resources available for future vehicles having high priority data. For this purpose, we define an offload criterion by considering threshold values of the maximum allowed load at RSU  $k$  for medium and low priority data (e.g.,  $\gamma_k^{max,med}$  and  $\gamma_k^{max,low}$  are the threshold values of maximum allowed medium and low priority data, respectively) for each vehicle. For instance, RSU  $k$  will allow a Vehicle  $i$  having medium priority data (resp. low priority) to offload its data if the current load  $\vartheta_k$  at RSU  $k$  is below the threshold value of maximum allowed medium priority data  $\gamma_k^{max,med}$  (resp.  $\gamma_k^{max,low}$ ), i.e.,  $\vartheta_k < \gamma_k^{max,med}$  (resp.  $\vartheta_k < \gamma_k^{max,low}$ ). Note that there is no threshold value of maximum allowed high priority data since RSUs will always try to service high priority data even if they need to remove other offloading (see next Section V-B3).

3) *Admission control*: In admission control, an RSU has the authority to stop servicing a vehicle that is currently offloading its data in order to provide service to another vehicle. As an example, a Vehicle  $i$  having high priority data wants to offload its data to RSU  $k$ . However, the current load  $\vartheta_k$  at RSU  $k$  is already equal to the maximum tolerable load  $\vartheta_k^{max}$  (i.e.,  $\vartheta_k = \vartheta_k^{max}$ ). In this case, RSU  $k$  will stop servicing one of the Vehicles  $j$  currently offloading medium or low priority data to reduce  $\vartheta_k$  and grant permission to  $i$  to offload high priority data.

### C. Data Offloading Request

When a vehicle  $i$  has connectivity with RSU  $k$ , it sends a data offloading request to RSU  $k$  containing its speed  $V_i$ , data rate  $r_i$  and the size of each priority data it wants to offload (i.e.,  $S_i^{high}$ ,  $S_i^{med}$  and  $S_i^{low}$ ). Subsequently, it waits for a time duration  $T_{i,v2i}^{wait}$  to receive a reply from RSU  $k$ .

### D. Processing of Data Offloading Request at RSU

Algorithm 1 presents the procedure of processing of data offloading request at RSU  $k$ . RSU  $k$  received data offloading request from Vehicle  $i$  and it first calculates the connectivity time  $T_{i,k}^c$  and offloading capacity  $S_{i,k}^{max}$  using Eqs. (1) and (2), respectively. In Part I of the algorithm, RSU  $k$  performs admission control for enabling  $i$  to offload high priority data. RSU  $k$  checks the amount of current high priority data  $S_{i,k}^{curr}$  that  $i$  can offload. If the size of high priority data  $S_i^{high}$  of  $i$  is less than the maximum allowed data size  $S_{i,k}^{max}$ , it sets the current amount of data  $S_{i,k}^{curr}$  to  $S_i^{high}$  (i.e.,  $S_{i,k}^{curr} \leftarrow S_i^{high}$ ), otherwise, it sets  $S_{i,k}^{curr} \leftarrow S_{i,k}^{max}$  since  $i$  cannot offload more data than the maximum allowed data size  $S_{i,k}^{max}$ . In that case, RSU  $k$  performs admission control if the sum of its current load  $\vartheta_k$  and the amount of current data  $S_{i,k}^{curr}$  is above the maximum tolerable load  $\vartheta_k^{max}$  (i.e.,  $\vartheta_k + S_{i,k}^{curr} \geq \vartheta_k^{max}$ ). RSU  $k$  stops servicing an existing Vehicle  $j$  that is offloading its data in order to reduce its current load so that the  $i$  can offload its high priority data instead. Vehicle  $j$ , chosen randomly, will be the one offloading its low priority data and if there is no vehicle offloading low priority data, then the Vehicle  $j$  will be the one offloading its medium priority data. RSU  $k$  informs Vehicle  $j$  about the service termination and updates its current load  $\vartheta_k$  by subtracting the current load  $S_{j,k}^{curr}$  of removed Vehicle  $j$  (i.e.,  $\vartheta_k \leftarrow \vartheta_k - S_{j,k}^{curr}$ ). RSU  $k$  repeats this process of removing one or more vehicles until its current load enables Vehicle  $i$  to offload its high priority data. Note that if there is no such Vehicle  $j$ , it means that all the existing vehicles are offloading high priority data, and hence the RSU  $k$  will inform Vehicle  $i$  that it cannot offload its data.

In Part II, RSU  $k$  analyzes the sizes of high, medium and low priority data that Vehicle  $i$  requested to offload and decides which priority data and how much data it can grant Vehicle  $i$  to offload. If the size of high priority data  $S_i^{high}$  requested by Vehicle  $i$  is higher than the maximum allowed data size  $S_{i,k}^{max}$  that Vehicle  $i$  can offload, the RSU  $k$  sets the granted set  $\Phi_{k,i}^{grant}$  (containing the sizes of different priority data that Vehicle  $i$  can offload to RSU  $k$ ) to be the size of high priority data equal to the maximum allowed data size (i.e.,  $\Phi_{k,i}^{grant} \leftarrow S_i^{high} \mid S_i^{high} = S_{i,k}^{max}$ ). Otherwise, it grants to offload all the high priority data and sets it into the granted set  $\Phi_{k,i}^{grant}$  (i.e.,  $\Phi_{k,i}^{grant} \leftarrow S_i^{high}$ ). Subsequently, if the size of granted set  $\Phi_{k,i}^{grant}$  is less than the maximum allowed data size  $S_{i,k}^{max}$  (i.e.,  $\Phi_{k,i}^{grant} < S_{i,k}^{max}$ ) and the sum of current load  $\vartheta_k$  at RSU  $k$  and granted set ( $\Phi_{k,i}^{grant}$ ) is below the threshold value  $\gamma_k^{max,med}$  of the maximum allowed medium priority data (i.e.,  $\vartheta_k + \Phi_{k,i}^{grant} < \gamma_k^{max,med}$ ), it means that there is a capacity for RSU  $k$  to take medium priority data of Vehicle  $i$ . Hence, if the

**Algorithm 1** Processing of data offloading request at RSU  $k$ .

```

1: Received data offloading request from Vehicle  $i$ ;
2: Compute  $T_{i,k}^c$  and  $S_{i,k}^{max}$  using Eqs. (1) and (2);
3:  $\Phi_{k,i}^{grant} \leftarrow \emptyset$ ;
4: /* Part I: Admission control.*/
5: if  $S_i^{high} > 0$  then
6:    $S_{i,k}^{curr} \leftarrow (S_i^{high} \leq S_{i,k}^{max}) ? S_i^{high} : S_{i,k}^{max}$ ;
7:   while  $(\vartheta_k + S_{i,k}^{curr} \geq \vartheta_k^{max})$  do
8:     Found  $\leftarrow j \mid S_i^{low} > 0$ ;
9:     If not Found then Found  $\leftarrow j \mid S_i^{med} > 0$ ; end if
10:    If not Found then Decline  $i$  request; Return; end if
11:    Inform Vehicle  $j$  about service termination and stop servicing  $j$ ;
12:     $\vartheta_k \leftarrow \vartheta_k - S_{j,k}^{curr}$ ;
13:  end while
14: end if
15: /* Part II: Check which priority and how much data to offload.*/
16: if  $S_i^{high} > S_{i,k}^{max}$  then
17:    $\Phi_{k,i}^{grant} \leftarrow S_i^{high} \mid S_i^{high} = S_{i,k}^{max}$ ;
18: else
19:    $\Phi_{k,i}^{grant} \leftarrow S_i^{high}$ ;
20: end if
21: if  $(\Phi_{k,i}^{grant} < S_{i,k}^{max})$  and  $(\vartheta_k + \Phi_{k,i}^{grant} < \gamma_k^{max,med})$  then
22:   if  $\Phi_{k,i}^{grant} + S_i^{med} > S_{i,k}^{max}$  then
23:      $\Phi_{k,i}^{grant} \leftarrow \Phi_{k,i}^{grant} \cup S_i^{med} \mid S_i^{med} = S_{i,k}^{max} - \Phi_{k,i}^{grant}$ ;
24:   else
25:      $\Phi_{k,i}^{grant} \leftarrow \Phi_{k,i}^{grant} \cup S_i^{med}$ ;
26:   end if
27: end if
28: if  $(\Phi_{k,i}^{grant} < S_{i,k}^{max})$  and  $(\vartheta_k + \Phi_{k,i}^{grant} < \gamma_k^{max,low})$  then
29:   if  $\Phi_{k,i}^{grant} + S_i^{low} > S_{i,k}^{max}$  then
30:      $\Phi_{k,i}^{grant} = \Phi_{k,i}^{grant} \cup S_i^{low} \mid S_i^{low} = S_{i,k}^{max} - \Phi_{k,i}^{grant}$ ;
31:   else
32:      $\Phi_{k,i}^{grant} = \Phi_{k,i}^{grant} \cup S_i^{low}$ ;
33:   end if
34: end if
35: Fetch  $S_i^{high}$ ,  $S_i^{med}$  and  $S_i^{low}$  from  $\Phi_{k,i}^{grant}$ ;
36:  $S_{i,k}^{curr} \leftarrow S_i^{high} + S_i^{med} + S_i^{low}$ ;  $\vartheta_k \leftarrow \vartheta_k + S_{i,k}^{curr}$ ;
37: Send  $\Phi_{k,i}^{grant}$  to  $i$ ;

```

sum of sizes of granted set (currently containing high priority data) and medium priority data is greater than the maximum allowed data size (i.e.,  $\Phi_{k,i}^{grant} + S_i^{med} > S_{i,k}^{max}$ ), the RSU  $k$  appends the partial medium priority data into the granted set  $\Phi_{k,i}^{grant}$  equal to the amount  $S_i^{med} = S_{i,k}^{max} - \Phi_{k,i}^{grant}$  (i.e.,  $\Phi_{k,i}^{grant} \leftarrow \Phi_{k,i}^{grant} \cup S_i^{med} \mid S_i^{med} = S_{i,k}^{max} - \Phi_{k,i}^{grant}$ ). Otherwise, it appends all the medium priority data into the granted set  $\Phi_{k,i}^{grant}$  (i.e.,  $\Phi_{k,i}^{grant} \leftarrow \Phi_{k,i}^{grant} \cup S_i^{med}$ ). Finally, if the size of granted set  $\Phi_{k,i}^{grant}$  is less than the maximum allowed data size  $S_{i,k}^{max}$  (i.e.,  $\Phi_{k,i}^{grant} < S_{i,k}^{max}$ ) and the sum of current load  $\vartheta_k$  at RSU  $k$  and granted set ( $\Phi_{k,i}^{grant}$ ) is below the threshold value  $\gamma_k^{max,low}$  of the maximum allowed low priority data (i.e.,  $\vartheta_k + \Phi_{k,i}^{grant} < \gamma_k^{max,low}$ ), it means that there is also a capacity for RSU  $k$  to take low priority data of Vehicle  $i$ . Hence, if the sum of sizes of granted set and low priority data is greater than the maximum allowed data size (i.e.,  $\Phi_{k,i}^{grant} + S_i^{low} > S_{i,k}^{max}$ ), the RSU  $k$  appends the partial low priority data into the granted set  $\Phi_{k,i}^{grant}$  equal to the amount  $S_i^{low} = S_{i,k}^{max} - \Phi_{k,i}^{grant}$  (i.e.,  $\Phi_{k,i}^{grant} \leftarrow \Phi_{k,i}^{grant} \cup S_i^{low} \mid S_i^{low} = S_{i,k}^{max} - \Phi_{k,i}^{grant}$ ). Otherwise, it appends all the low priority data into the granted set  $\Phi_{k,i}^{grant}$  (i.e.,  $\Phi_{k,i}^{grant} \leftarrow \Phi_{k,i}^{grant} \cup S_i^{low}$ ). Finally, RSU  $k$  fetches the granted sizes of high, medium and low priority

data from granted set  $\Phi_{k,i}^{grant}$ , calculates the current data size  $S_{i,k}^{curr}$  that Vehicle  $i$  will offload to RSU  $k$  by taking their sum (i.e.,  $S_{i,k}^{curr} \leftarrow S_i^{high} + S_i^{med} + S_i^{low}$ ) and updates its current load  $\vartheta_k$  by adding the current Vehicle  $i$ 's granted data size to its current load (i.e.,  $\vartheta_k \leftarrow \vartheta_k + S_{i,k}^{curr}$ ). It then sends the granted set  $\Phi_{k,i}^{grant}$  as data offloading reply to Vehicle  $i$ .

**E. Data Offloading Start**

When Vehicle  $i$  receives data offloading reply from RSU  $k$ , it also receives its granted set  $\Phi_{k,i}^{grant}$ . Hence, it fetches the granted sizes for each priority. Finally, it updates the sizes of its remaining data.

**VI. PERFORMANCE EVALUATION**

In this section, we evaluate the performance of V2I-Q through extensive simulations and present simulation setup, parameters, performance metrics, results and discussions.

**A. Simulation Setup and Parameters**

V2I-Q is implemented in network simulator OMNeT++ 5.5.1 with Veins 5.0 [8] and SUMO 1.7.0 (Simulation for Urban Mobility) [9] frameworks. SUMO is used to create the scenarios and mobility of vehicles, while OMNeT++ and Veins are used to simulate the vehicular communications using IEEE 802.11p standard. We built a 30km two-way highway scenario with three lanes on each direction without intersections. The vehicles are equally distributed on each lane and each vehicle is departed at a random time and location. The generated scenario is imported into OMNeT++ using Veins framework. The RSUs are uniformly distributed besides the highway. Vehicles data is randomly distributed among high, medium and low priority. Each vehicle and RSU periodically send beacons with an interval of one second. We assume that while sending beacons, the RSU uses the same transmit power as of vehicles. Hence, if vehicles can receive a beacon from RSU, then RSU can also receive a beacon from vehicles. However, RSU can use higher transmission power to broadcast some emergency messages to vehicles in order to cover larger area, which is not the scope of this work. Since we are not dealing with networking and MAC layer issues, after a vehicle is informed about the amount of data it can offload, it considers the received beacon as an acknowledgement and thus, calculates the amount of data that could be offloaded since the reception of last beacon using its speed and data rate. Subsequently, it updates its application data size. This helps evaluate the full performance of our data offloading scheme without having impact of external factors (such as collision). V2I-Q is compared with a traditional V2I offloading scheme in which vehicles offload their data to RSUs without QoS provisioning. Table I summarizes simulation parameters.

**B. Performance Metrics**

The performance metrics we considered for V2I-Q are :

- *Offloaded data* : the total amount of data successfully offloaded to RSUs.
- *Maximum offloading delay* : maximum time required for all the vehicles to offload their data per priority.

TABLE I: Simulation parameters and values.

Parameter	Value
Area	30km long highway
Number of RSUs	30
Number of vehicles	100
Maximum speed	25m/s
Vehicles depart position	Random
Vehicles depart time	Random
Simulation time	2000 seconds
Simulation runs	20
Maximum RSU capacity	1000Mb
$\gamma_k^{max,med}$	75%
$\gamma_k^{max,low}$	100%
Vehicle data	6000Mb
Data rate	6Mbps
Transmission power	10mW
Transmission range	357m
Communication technology	IEEE 802.11p
Beacon interval	1 sec

### C. Performance Evaluation

Fig. 2 shows the effect of varying number of vehicles from 50 to 250. When the number of vehicles increases, the amount of offloaded data and the maximum offloading delay for both V2I-Q and V2I schemes increase. This is very natural since as the number of vehicles increases, the amount of data in the network also increases. The number of RSUs being fixed, the maximum offloading delay increases. The total amount of offloaded data for both V2I-Q and V2I is almost similar except when the number of vehicles exceed 200. This is because the threshold for maximum medium and low priority data ( $\gamma_k^{max,med}$  and  $\gamma_k^{max,low}$ ) are reached. Overall, V2I-Q was able to offload the highest amount of high priority data with the least maximum offloading delay, similar amount and maximum delay for offloading medium priority data but at the cost of a lower amount of low priority data and a higher delay for this priority level.

Fig. 3 shows the effect of varying vehicles' maximum speed from 15m/s to 35m/s. When the vehicles maximum speed increases, the amount of offloaded data and maximum offloading delay for both V2I-Q and V2I reduces because with higher speed, the connectivity time of vehicles with RSUs reduces. The total amount of offloaded data for both V2I-Q and V2I is almost similar. Overall, V2I-Q was able to offload the highest amount of high priority data with the least maximum offloading delay. For medium priority data, the amount of offloaded data and maximum delay for both V2I-Q and V2I are similar. However, for low priority data, again V2I-Q offloaded lower amount and incurred higher delay at the cost of offloading more high priority data with lower offloading delay.

Fig. 4 shows the effect of varying RSUs capacity from 2000Mb to 10000Mb. When the RSUs capacity is 2000Mb, V2I-Q offloads lower amount of total data and incurred higher maximum offloading delay, as compared to V2I. This is because of insufficient RSUs capacity that restricted V2I-Q from offloading higher amount of medium and low priority data because of threshold values  $\gamma_k^{max,med/low}$ . However,

when the RSUs capacity increases, V2I-Q performs better than V2I and offloads more high priority data quicker. As expected, V2I-Q offloads similar amount of medium priority data as V2I, however it achieves a lower maximum offloading delay. Last, V2I-Q offloads less low priority data with higher delay. It is interesting to note that when the RSUs capacity reaches 6000Mb, the performance of both V2I-Q and V2I is similar for increasing RSUs capacity since in all cases all data can be offloaded. It means that in our current scenario, RSUs can be installed with a maximum capacity of 6000Mb.

Overall, for all the scenarios considered V2I-Q performs the best as compared to V2I. V2I-Q offloads more high priority data quicker, a similar amount of medium priority data with lower/similar delay, but less low priority data with higher delay.

## VII. CONCLUSION

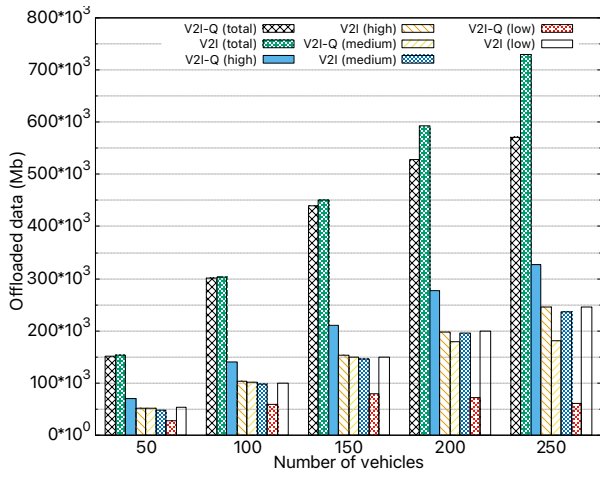
In this paper, we proposed V2I-Q, a V2I data offloading scheme for vehicular networks with QoS provisioning. We modeled the connectivity of vehicles with RSU and the offloading capacity to decide by RSU, the amount of data to grant to vehicles for offloading. We used three QoS functions of traffic classification, overload control and admission control, and provided detailed data offloading procedure and algorithms of V2I-Q, i.e., how a vehicle selects an RSU for requesting data offloading, how the data offloading request is processed at RSU and finally, how a requested vehicle processed data offloading reply to start data offloading. The simulation results confirmed that V2I-Q outperforms the traditional V2I data offloading scheme by offloading more high priority data by achieving lower delay. As future work, we will extend our scheme to V2V communications.

## ACKNOWLEDGMENT

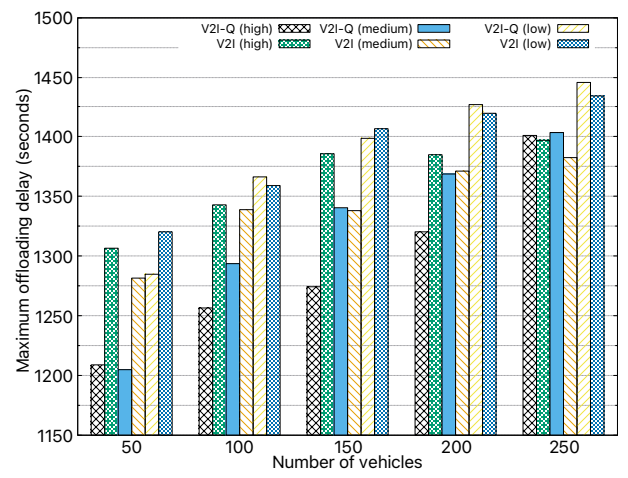
This work was partially supported by CPER DATA and ELSAT project.

## REFERENCES

- [1] C.-M. Huang, S.-Y. Lin, and Z.-Y. Wu, "The k-hop-limited V2V2I VANET data offloading using the Mobile Edge Computing (MEC) mechanism," *Vehicular Communications*, vol. 26, p. 100268, 2020.
- [2] C. Song, J. Wu, W. S. Yang, M. Liu, I. Jawhar, and N. Mohamed, "Exploiting Opportunities in V2V Transmissions with RSU-assisted Backward Delivery," *IEEE Conf. on Computer Communications Workshops, INFOCOM WKSHPs*, 2017.
- [3] X. Zhu, Y. Li, D. Jin, and J. Lu, "Contact-Aware Optimal Resource Allocation for Mobile Data Offloading in Opportunistic Vehicular Networks," *IEEE Trans. on Vehicular Technology*, vol. 66, pp. 7384–7399, 2017.
- [4] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic Offloading for Online Video Service in Vehicular Networks: A Cooperative Approach," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 8, pp. 7630–7642, 2018.
- [5] S. Guntuka, E. M. Shakshuki, A. Yasar, and H. Gharad, "Vehicular data offloading by road-side units using intelligent software defined network," *Procedia Computer Science*, vol. 177, pp. 151–161, 2020.
- [6] F. Lyu, H. Zhu, N. Cheng, H. Zhou, W. Xu, and et al., "Characterizing Urban Vehicle-to-Vehicle Communications for Reliable Safety Applications," *IEEE Trans. on Intelligent Transp. Systems*, pp. 1–17, 2019.
- [7] "Quality of Service (in Mobile Networks), Last accessed: July 2020." [Online]. Available: <https://www.tu-ilmnau.de/fileadmin/public/iks/files/lehre/mobicom/MCN-08-QoS.pdf>
- [8] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.
- [9] P. A. Lopez and et al., "Microscopic Traffic Simulation using SUMO," *IEEE Conf. on Intelligent Transportation Systems (ITSC)*, 2018.

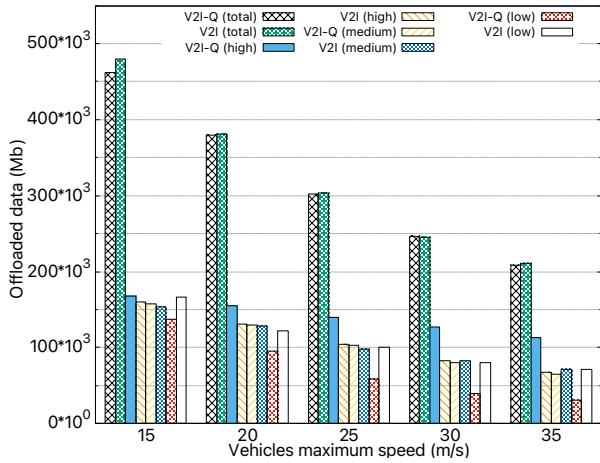


(a) Amount of offloaded data.

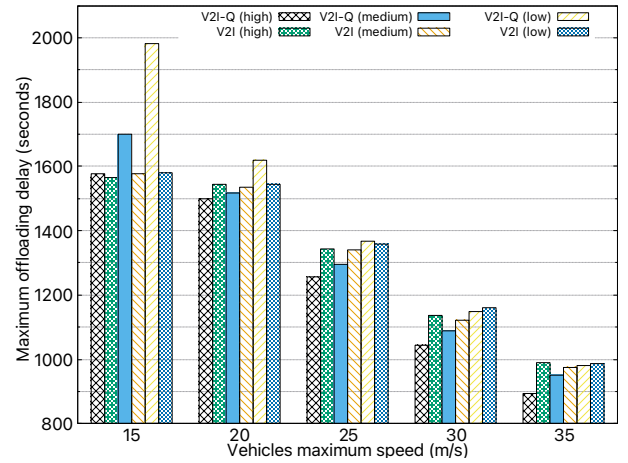


(b) Maximum offloading delay.

Fig. 2: Effects of number of vehicles.

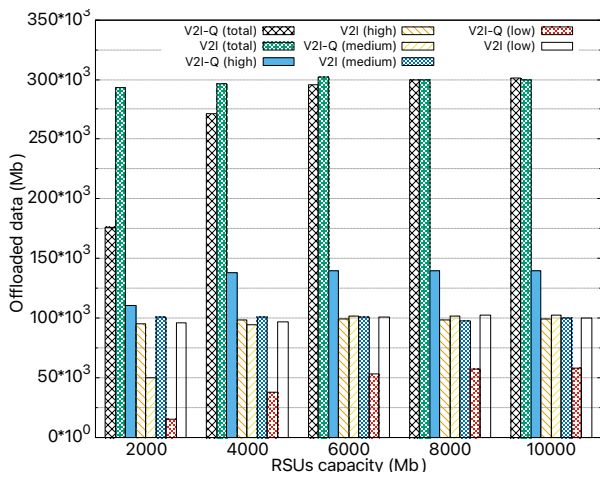


(a) Amount of offloaded data.

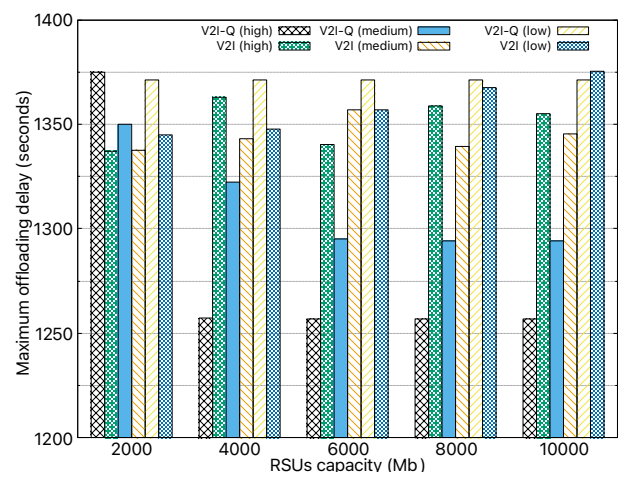


(b) Maximum offloading delay.

Fig. 3: Effects of vehicles maximum speed.



(a) Amount of offloaded data.



(b) Maximum offloading delay.

Fig. 4: Effects of RSUs capacity.