



HAL
open science

A QoS-Aware Hybrid V2I and V2V Data Offloading for Vehicular Networks

Yasir Saleem, Nathalie Mitton, Valeria Loscri

► **To cite this version:**

Yasir Saleem, Nathalie Mitton, Valeria Loscri. A QoS-Aware Hybrid V2I and V2V Data Offloading for Vehicular Networks. VTC2021-Fall - 94th IEEE Vehicular Technology Conference, Sep 2021, Virtual, United States. hal-03318684

HAL Id: hal-03318684

<https://hal.inria.fr/hal-03318684>

Submitted on 10 Aug 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 QoS-Aware Hybrid V2I and V2V Data Offloading for Vehicular Networks

Yasir Saleem, Nathalie Mitton, Valeria Loscri
Inria Lille – Nord Europe, France - firstname.lastname@inria.fr

Abstract—In vehicular networks, RoadSide Units (RSUs) are not available everywhere, therefore, it is not possible for vehicles to stay connected with RSUs all the time and to send their data directly to RSUs anytime using Vehicle-to-Infrastructure (V2I) data offloading. Hence, in these cases, Vehicle-to-Vehicle (V2V) communications are used to offload data to RSUs through other vehicles. Data to offload could be urgent (e.g., accident data), therefore, it is important to consider the Quality of Service (QoS) provisioning. In this paper, we propose a QoS-aware data offloading scheme for vehicular networks with QoS provisioning (DOVEQ) that considers both V2I and V2V data offloading. DOVEQ models the connectivity of vehicles with RSUs and vehicles, offloading capacity and estimation of reaching RSUs. It provides QoS using traffic classification, overload control and admission control. The performance evaluation of DOVEQ shows that DOVEQ outperforms other schemes by offloading more amount of important data with lesser offloading delay and running time.

Index Terms—Data offloading, Road-side Unit (RSU), Quality of Service (QoS), Vehicular network, Vehicle-to-Infrastructure (V2I) communication, Vehicle-to-Vehicle (V2V) communication.

I. INTRODUCTION

In a vehicular network, RoadSide Units (RSUs) are equipped with edge servers and they analyze/process the data to take necessary actions or forward to the cloud for storage or further processing. Vehicles cannot always communicate with RSUs due to intermittent deployment of RSUs. In such a case, vehicles can leverage V2V data offloading.

In [1], the authors propose a Vehicle-to-Vehicle-to-Infrastructure (V2V2I) data offloading scheme that constructs a connected multihop V2V path towards RSUs. A vehicle relies on cellular network for data offloading, unless it is able to communicate with another vehicle (offloading agent) that has a k -hop V2V path towards an RSU. [2] investigates the Floating Car Data (FCD) that is a huge amount of data collected by vehicles and is sent to the control centers for processing through cellular communications that can overload the network. The authors introduce the uploader vehicles that receive FCD from vehicles using Dedicated Short-Range Communication (DSRC) technology, aggregate them using a distributed heuristic algorithm and send them to a control center through cellular network. [3] considers video data offloading from cellular networks to vehicles using RSUs that act as traffic managers to reduce the traffic load as an optimization problem. Using the vehicles' mobility prediction and throughput estimation, the authors design a storage time

aggregated graph and propose an iterative greedy algorithm for a sub-optimal solution with a polynomial time complexity. In Vehicle Assisted Offloading (VAO) [4], the goal is to reduce the load at cellular networks from pedestrians. Base Stations of light-loaded cellular networks offload delay-tolerant data to vehicles which further send the data to pedestrians after entering the street intersection and stopping at the red light. Smart Ranking-based Data Offloading (SRDO) [5] enables the vehicles to switch from cellular network to V2I whenever possible. SRDO is a centralized scheme that uses Q-learning for RSUs' selection and is based on Software Defined Networking (SDN). RSU-Assisted Backward Delivery (RABD) [6] investigates backward data delivery for V2V data transmissions under the traffic hole problem. RABD relays data among vehicles in backward direction and by RSUs. [7] proposes an adaptive QoS-based routing protocol for vehicular networks (AQRV) that considers three QoS metrics: delay, packet delivery ratio and probability of connectivity. It selects the routes that satisfy these QoS metrics by mathematically formulating as a constrained optimization problem using ant colony optimization. [8] investigates QoS issues in vehicular networks and provides three contributions. Firstly, it proposes a QoS-aware green, sustainable, reliable and available (QGSRA) algorithm for multimedia transmission in edge-based vehicular networks. Secondly, it implements a QoS optimization for QGSRA algorithm. Thirdly, it proposes the QoS metrics of energy efficiency, packet loss ratio and coverage for QGSRA algorithm. We have previously proposed a V2I data offloading scheme with QoS provisioning (V2I-Q) [9]. We extend V2I-Q in this paper by proposing DOVEQ that enables the vehicles to take advantage of V2V data offloading when V2I data offloading is not possible.

This paper proposes DOVEQ, a QoS-aware data offloading scheme with the objective of offloading the data to RSUs as soon as possible. Differently from previous works, we model the connectivity of vehicles with other vehicles heading on the same or opposite direction and the expected time to reach next RSUs. Whenever vehicles are in the coverage of RSUs, DOVEQ uses V2I data offloading, otherwise, it uses V2V using other vehicles that are going to meet RSUs sooner. DOVEQ provides the QoS using the three functions: i) traffic classification that categorizes the data into high, medium and low priorities; ii) overload control that defines the threshold values of maximum allowed load at RSUs for medium and low priority data to avoid RSUs from overloading; iii) admission control that enables RSUs and vehicles to stop servicing a

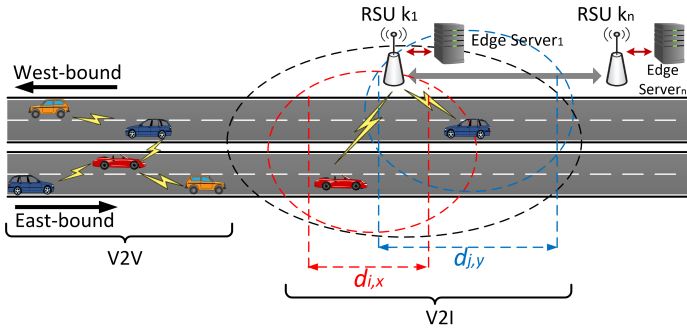


Fig. 1: System model.

vehicle offloading its lower priority data to allow another vehicle to offload high priority data. To the best of our knowledge, there is no existing scheme that offers a hybrid V2I and V2V data offloading with such QoS provisioning and priority data in their forwarding mechanisms.

This paper is organized as follows. The system and connectivity models are presented in Section II. DOVEQ is presented in Section III. The performance is evaluated in Section IV. Finally, this paper is concluded in Section V.

II. SYSTEM AND CONNECTIVITY MODEL

In this section we detail the specific scenario we considered and introduce our connectivity model.

A. System Model

Fig. 1 presents our system model that is comprised of a vehicular network having \mathcal{N} vehicles and \mathcal{K} RSUs. The vehicles have intermittent connectivity with RSUs. The vehicular network is a highway having east and west bound roads with multiple lanes. Each vehicle moves at a variable speed and carries data to be offloaded to RSUs. The vehicles and RSUs communicate together using DSRC technology and their connectivity varies from lane to lane. At one time, a vehicle can be connected to maximum one RSU. Each vehicle has the same coverage area, however the coverage area of RSUs is higher than vehicles. RSUs are connected to edge servers. Each RSU collects data from vehicles and forwards it to their underlying edge server for analysis/storage.

B. Connectivity Model

1) *Connectivity Time with RSU*: It estimates the time duration of a vehicle i to stay connected with an RSU k . It is used in data offloading by RSU k to calculate the amount of data to grant to i . It is calculated as:

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

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 k and $T_{i,v2i}^{wait}$ is the waiting time for i after sending data offloading request to k .

2) *Connectivity Time with Vehicles*: The connectivity time of two vehicles depends on their heading direction. Vehicles heading on the same direction have higher connectivity time than those heading on the opposite direction. The connectivity

time between Vehicles i and j (resp. l) heading on the same direction (resp. on opposite directions) is calculated as:

$$T_{i,j}^c = \frac{d_i}{|V_i - V_j|} - \tau_{i,j} - T_{i,v2v}^{wait} \quad (2)$$

$$T_{i,l}^c = \frac{d_i}{V_i + V_l} - \tau_{i,l} - T_{i,v2v}^{wait} \quad (3)$$

where d_i is the coverage area of Vehicle i , V_i , V_j , V_l are the speeds of Vehicles i , j and l , respectively, $|V_i - V_j|$ is the modulus of difference of speeds of both i and j , $\tau_{i,j}$ and $\tau_{i,l}$ are the registration time of i with j and l . $T_{i,v2v}^{wait}$ is the waiting time of i after sending V2V data offloading request.

3) *Offloading Capacity*: It is the maximum allowed data size that Vehicle i can offload to RSU k within a connectivity time $T_{i,k}^c$ and is directly proportional to $T_{i,k}^c$. 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:

$$S_{i,k}^{max} = T_{i,k}^c \times r_i \quad (4)$$

where r_i is the data rate of i .

4) *Expected Time To Reach RSU*: The distance l_j^{rsu} of last RSU encountered by j is exchanged among vehicles. This information is used to estimate the arrival time to next RSU (i.e., expected time to reach the RSU) by vehicles heading on the opposite direction and is calculated by Vehicle i as:

$$T_{i,RSU}^{arr} = \frac{l_j^{rsu}}{V_i} \quad (5)$$

III. OUR SOLUTION:DOVEQ

A. Overview

Vehicles carry data having $\mathcal{P} = \{high, medium, low\}$ priorities to be offloaded to RSUs through V2I or V2V data offloading with traffic classification (Section III-B1), with overload control at RSU (Section III-B2) and with admission control at both RSUs and vehicles (Section III-B3).

When a vehicle wants to offload its data, it sends a data offloading for each type of data in order to ensure that higher priority data is offloaded before lower priority data (Section III-C). Subsequently, RSUs or vehicles process the data offloading request and decide whether and how much data to grant and reply back to the vehicle (Section III-D). Finally, once permission is granted, the vehicle starts data offloading (Section III-E).

B. Quality of Service (QoS) Provisioning

QoS provisioning can be achieved using different functions [10]. DOVEQ applies traffic classification, overload control and admission control.

1) *Traffic classification*: Data is categorized into three priorities: high (urgent data), medium (standard data), and low (delay-tolerant data). For data having same priority, data with the largest size will be offloaded following a greedy approach.

2) *Overload control*: It is used to avoid RSUs from overloading and keeping their resources available for future vehicles with high priority data. We define threshold values of the maximum allowed load for medium and low priority data at RSU k (i.e., $\gamma_k^{max,med}$ and $\gamma_k^{max,low}$). RSU k only allows vehicles with medium (or low) priority data to offload

if its current load ϑ_k is below the threshold value $\gamma_k^{max,med}$ (or $\gamma_k^{max,low}$). There is no threshold for high priority data because it always needs to be serviced even though if some vehicles that are currently offloading low or medium priority data have to be terminated (see Section III-B3).

3) *Admission control*: The vehicular traffic under one RSU could be greater than that assumed by it, e.g., higher number of vehicles with lower priority data enter the network and occupy RSU's resources. The vehicles with higher priority data enter the network later. RSU cannot provide service to all of them because of being overloaded. This is why, admission control function is used and we use it in two ways. First, if the current load ϑ_k at RSU k reaches the maximum tolerable load ϑ_k^{max} , k can stop servicing a vehicle (offloading lower priority data) and allow a new one (having high priority data) to offload its data. Second, in V2V data offloading, a vehicle only offloads its high and medium priority data.

C. Data Offloading Request

A vehicle can request data offloading to RSU or neighboring vehicle. If a Vehicle i is connected to an RSU k , it sends a data offloading request to k containing its data rate r_i , speed V_i and the sizes of all the data it contains (i.e., S_i^{high} , S_i^{med} and S_i^{low}). It then waits for $T_{i,v2i}^{wait}$ duration for a reply.

Otherwise, if Vehicle i is connected to nb_i vehicles, it checks whether to request any of them for data offloading. It first estimates the expected time to reach the next RSU $T_{nb_i,RSU}^{arr}$ using Eq. (5) for all neighbor nb_i . Then it selects among them the Vehicle j having the least expected time to reach the next RSU. Vehicle i calculates its connectivity time $T_{i,j}^c$ with j using Eq. (2) or (3) in case of same or opposite heading direction, respectively. Subsequently, i checks the three conditions in which V2V data offloading is not preferred: (i) i can reach the next RSU earlier or at the same time as j (i.e., $T_{i,RSU}^{arr} \leq T_{j,RSU}^{arr}$), (ii) the difference of the expected time to reach RSUs for i and j is lower than the threshold value Ω_{RSU}^{arr} , (iii), the connectivity time of i and j is lower than the threshold value $\omega_{v2v}^{min,c}$ of the least allowed contact duration between vehicles for V2V data offloading (i.e., the connectivity time is too short). If none of the above conditions is met, then i sends the data offloading request to j , including its parameters and the size of high and medium priority data for V2V data offloading. Finally, it waits for a $T_{i,v2v}^{wait}$ duration to get a reply. i repeats this process until it finds a suitable neighbor or there is no more available neighbor.

D. Processing of Data Offloading Request

1) *Processing of Data Offloading Request at RSU*: When RSU k receives a data offloading request from Vehicle i , it calculates which and how much data it can allow i to offload. It first calculates its connectivity time $T_{i,k}^c$ and the offloading capacity $S_{i,k}^{max}$ (i.e., the maximum allowed data size) using Eqs. (1) and (4), resp. Then it checks whether admission control is required, i.e., if i has high priority data (i.e., $S_i^{high} > 0$) to offload but its current load ϑ_k is the maximum. If so, it selects a currently offloading Vehicle j and stops servicing to it (partially or completely) in order to reduce

Algorithm 1

Run at vehicle j upon reception of offloading request from i .

```

1: if Receiving data from Vehicle  $l$  then
2: if ( $S_i^{high} = 0$  and  $S_i^{high} > 0$ ) then
3:   Send CALCULATEMAXSIZES() to Vehicle  $i$ ;
4:   Stop servicing  $l$  and inform it to try again later;
5: else
6:   Inform Vehicle  $i$  to try again later later;
7: end if
8: else if Vehicle  $j$  is offloading its data to an RSU or another vehicle  $k$ 
9: then Inform Vehicle  $i$  to try again later;
10: else Send CALCULATEMAXSIZES() to Vehicle  $i$ ; end if
11: function CALCULATEMAXSIZES()
12:    $grant_{i,max}^{high} = 0$ ;  $grant_{i,max}^{med} = 0$ ;
13:   if ( $S_j^{max} > S_j^{high}$ ) then  $grant_{i,max}^{high} \leftarrow S_j^{max} - S_j^{high}$ ; end if
14:   if ( $grant_{i,max}^{high} > S_i^{high}$ ) then  $grant_{i,max}^{high} \leftarrow S_i^{high}$ ; end if
15:    $allocated \leftarrow S_j^{high} + grant_{i,max}^{high} + S_j^{med}$ ;
16:   if ( $S_j^{max} > allocated$ ) then
17:      $grant_{i,max}^{med} \leftarrow S_j^{max} - allocated$ ; end if
18:     if ( $grant_{i,max}^{med} > S_i^{med}$ ) then  $grant_{i,max}^{med} \leftarrow S_i^{med}$ ; end if
19:   Return  $grant_{i,max}^{high} \cup grant_{i,max}^{med}$ ;
20: end function

```

its current load. k repeats this process until its current load enables it to allow i to offload its high priority data. Note that if there is no Vehicle j offloading the low or medium priority data, it means that all the existing vehicles are offloading their high priority data, and hence k will inform i that it cannot offload its data.

Subsequently, by analyzing the sizes of each priority data that i has requested to offload, and the offloading capacity, k calculates which and how much data i can offload within $T_{i,k}^c$ by ensuring that it does not grant to i the medium and low priority data more than allowed current load $\gamma_k^{max,med}$ and $\gamma_k^{max,low}$, respectively. Then it sets the amount of each priority data into a granted set $\Phi_{k,i}^{grant}$ and sends it to i as data offloading reply. Finally, it updates its current load.

2) *Processing of Data Offloading Request at Vehicle*: Algorithm 1 presents the procedure of processing of V2V data offloading request at j . Vehicle j could be in four possible situations. Firstly, it could be receiving offloaded data from another Vehicle l . So, if the new Vehicle i requests to offload high priority data, while l has no high priority data, then j performs admission control and terminates the service to l . Then it calculates the granted data sizes of high and medium priority data it can allow to i using CALCULATEMAXSIZES() function (discussed next). Otherwise, (admission control is not preferred), it informs i to try again later. CALCULATEMAXSIZES() provides the maximum allowed data sizes of each priority data that j can allow i to offload. j first estimates the offloading capacity S_j^{max} using Eq. (4) that it can offload to an RSU (in the future). Based on S_j^{max} and the amount of requested high and medium priority data by i (i.e., S_i^{high} and S_i^{med}), j calculates the amount of maximum data it can allow i to offload.

In the second and third situations, j is offloading its data to RSU k and another Vehicle m , respectively. So, j calculates the time duration $T_{j,k}^{ofld}$ and $T_{j,m}^{ofld}$ that it requires to offload its data $S_{j,k}^*$ and $S_{j,m}^*$ to k and m , respectively. Accordingly, it informs i to try again later after $T_{j,k}^{ofld}$ and $T_{j,m}^{ofld}$ duration.

Algorithm 2 Run at vehicle i upon reception of data offloading reply containing granted data sizes from Vehicle j .

```

1: Compute  $T_{i,j}^c$ , either using Eq. (2) ( $H_i = H_j$ ) or Eq. (3);
2: Compute  $S_{i,j}^{max}$  using Eq. (4);
3: if ( $S_{i,j}^{high} \geq (grant_{i,max}^{high} + grant_{i,max}^{med})$ ) then
4:   /* Offload all the allowed data. */
5:   Offload  $S_i^{high} = grant_{i,max}^{high}$  &  $S_i^{med} = grant_{i,max}^{med}$ ;
6:   Update sizes of  $S_i^{high}$  &  $S_i^{med}$ ;
7: else if ( $S_{i,j}^{max} \leq grant_{i,max}^{high}$ ) then
8:   /* Offload high priority data equal to offloading capacity. */
9:   Offload  $S_i^{high} = S_{i,j}^{max}$ ; Update size of  $S_i^{high}$ ;
10: else if ( $S_{i,j}^{max} < grant_{i,max}^{high} + grant_{i,max}^{med}$ ) then
11:   /* Offload high priority data up to the granted high priority & medium
    data up to remaining capacity.*/
12:   Offload  $S_i^{high} = S_{i,j}^{max,high}$ ; &  $S_i^{med} = S_{i,j}^{max} - grant_{i,max}^{high}$ ;
13:   Updates sizes of  $S_i^{high}$  &  $S_i^{med}$ ;
14: end if

```

Finally, in the fourth situation, j is currently in a free state and thus, it computes the granted set $grant_{i,max}^{high}$ and $grant_{i,max}^{med}$ as data offloading reply to i .

E. Data Offloading Decision

1) *Reply from RSU*: When Vehicle i receives a data offloading reply from RSU k , it includes the actual size of each priority data by k that i should offload. Hence, i starts offloading each priority data equal to the size informed by k .

2) *Reply from Neighboring Vehicle*: When Vehicle i receives a data offloading reply from a Vehicle j , i first fetches the granted data sizes of high and medium priority data (i.e., $grant_{i,max}^{high}$ and $grant_{i,max}^{med}$), and calculates its connectivity time $T_{i,j}^c$ with j using Eq. (2) or Eq. (3), in case of same or opposite heading direction, respectively and then calculates the offloading capacity $S_{i,j}^{max}$ using Eq. (4). Subsequently, based on $S_{i,j}^{max}$, $grant_{i,max}^{high}$ and $grant_{i,max}^{med}$, it calculates the actual amount of high and medium priority data and starts data offloading to j . Algorithm 2 provides the detailed procedure.

IV. PERFORMANCE EVALUATION

A. Simulation Setup and Parameters

DOVEQ is implemented in network simulator OMNeT++ 5.5.1 [11] with Veins 5.0 [12] and SUMO 1.7.0 [13] frameworks. SUMO is used to generate the mobility of vehicles and scenarios, while OMNeT++ and Veins are used to simulate the vehicular communications using IEEE 802.11p standard. We built a 30km two-way highway scenario in SUMO with three lanes on each direction without intersections. 50-200 vehicles with maximum speed of 25m/s are fairly distributed on each lane and depart at a random time and location. They carry 6000Mb at 6Mbps data rate over 357m transmission range. 30 RSUs with maximum capacity of 10000Mb are uniformly distributed besides the highway. Vehicles and RSUs periodically send beacons every second and with the same transmission power. RSU's beacons are considered as acknowledgment by vehicles. $\gamma_k^{max,med}$ is 75% and $\gamma_k^{max,low}$ is 50%. The generated scenario is imported into OMNeT++ using Veins framework. DOVEQ is compared with V2I-Q [9] and with DOVE, a baseline of DOVEQ without QoS provisioning. Simulations for 2000 seconds or until all vehicles leave the network, for 20 times with different seeds.

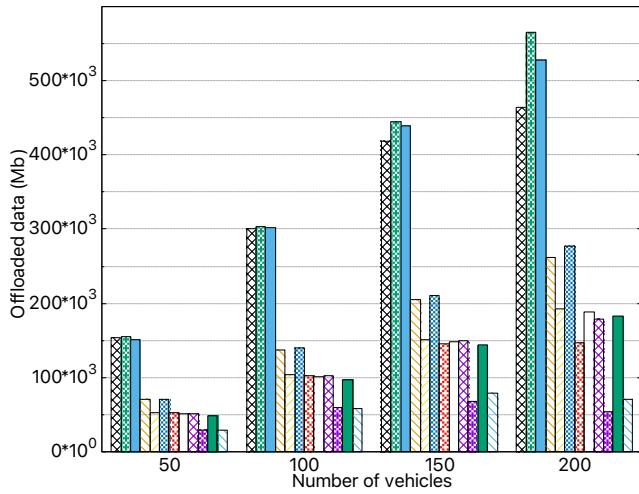
B. Results and Discussion

Fig. 2 presents the amount of offloaded data, average and maximum offloading delays and running time by DOVEQ, DOVE and V2I-Q under the varying number of vehicles. The amount of offloaded data increases with the number of vehicles (Fig. 2(a)) because the amount of data also increases with the number of vehicles. The total running of all schemes is similar (see Fig. 2(d)). DOVEQ and V2I-Q offload the highest amount of high priority data, thanks to QoS considerations, but at the cost of offloading less low priority data. DOVEQ achieves the lowest offloading delay for total and high priority data (Fig. 2(b)), and for it achieves the lowest maximum delay for offloading high priority data (Fig. 2(c)) and the lowest running time (Fig. 2(d)), fulfilling our objective of offloading the data as soon as possible to RSUs, thanks to QoS provisioning and V2V data offloading, but at the expense of lesser performance for low priority data. The maximum offloading delay of all data for DOVEQ is almost similar to V2I-Q because it corresponds to the cases when V2V data offloading in DOVEQ is either not possible or not preferred. Hence, vehicles use V2I data offloading, exhibiting similar maximum offloading delay as V2I-Q. V2I-Q achieves slightly higher offloading delay for total and high priority data, and higher maximum offloading delay and running time for high priority data than DOVEQ because of not taking advantage of V2V data offloading. DOVE offloads lower amount of high priority data than DOVEQ and V2I-Q, however it has lower average offloading delay of total and high priority data than V2I-Q which shows the significance of V2V data offloading, i.e., even without QoS provisioning, V2V data offloading can significantly help reduce the offloading delay. For medium priority data, all schemes perform similarly in almost all the aspects. For low priority data, DOVE achieves lower offloading delays and lower running time because of no QoS provisioning. DOVEQ and V2I-Q perform similarly because low priority data is offloaded only after higher priority data.

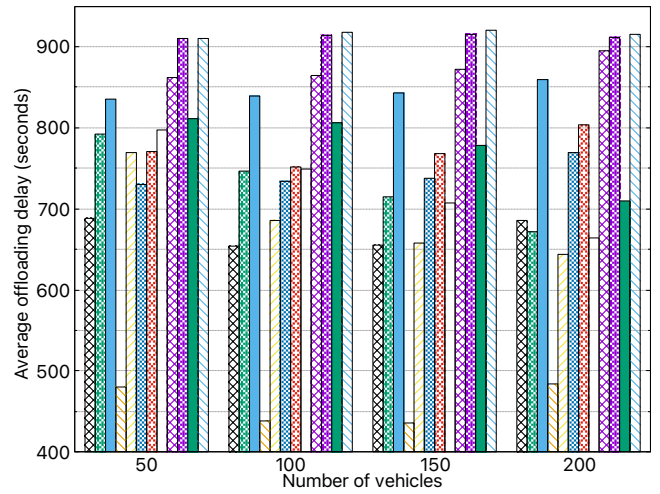
To summarize, DOVEQ performs the best by offloading the highest amount of high priority data similar to V2I-Q by achieving the lowest average and maximum offloading delays, however at the cost of less low priority data. DOVE causes higher maximum offloading delay for all priority data than DOVEQ and V2I-Q because of no QoS provisioning. DOVE achieves better performance for low priority data at the expense of less high priority data.

V. CONCLUSION

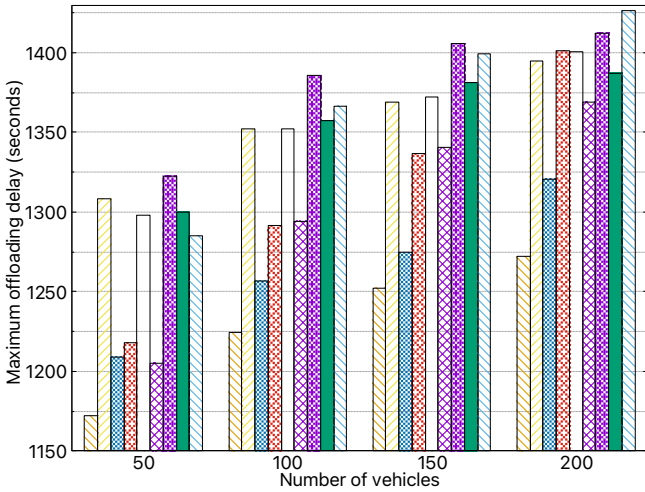
This paper proposed DOVEQ, a QoS-aware hybrid V2I and V2V data offloading scheme for vehicular networks. We modeled the connectivity of vehicles with RSU, with other vehicles heading on the same or opposite direction and the offloading capacity. We used traffic classification, overload control and admission control functions for QoS provisioning and proposed V2I and V2V data offloading procedures. The simulation results showed that DOVEQ outperforms V2I-Q and DOVE schemes by offloading more high priority data with lower delay and running time. In future, we plan to investigate



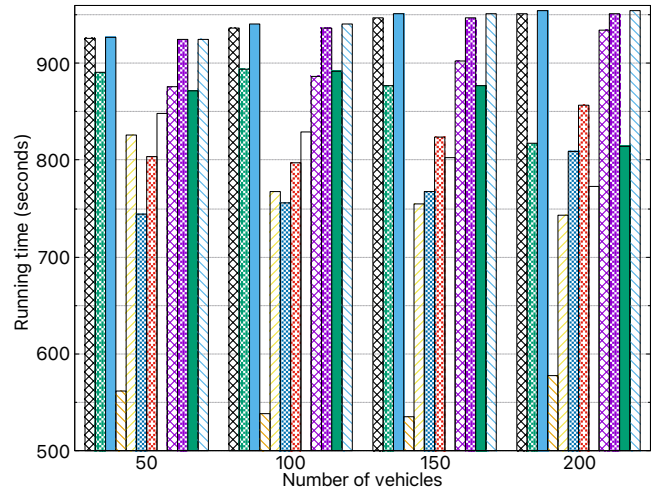
(a) Amount of offloaded data.



(b) Average offloading delay.



(c) Maximum offloading delay.



(d) Running time.



Fig. 2: Effects of number of vehicles.

the varying data rate, overlapping in the coverage of RSUs and an urban scenario.

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," *Veh. Commun.*, vol. 26, p. 100268, 2020.
- [2] S. Ancona, R. Stanica, and M. Fiore, "Performance Boundaries of Massive Floating Car Data Offloading," in *IEEE/IFIP Proc. 11th Annu. Conf. Wirel. On-Demand Netw. Syst. Serv.*, 2014.
- [3] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic Offloading for Online Video Service in Vehicular Networks: A Cooperative Approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7630–7642, 2018.
- [4] J. Feng and Z. Feng, "A Vehicle-Assisted Offloading Scheme for Hotspot Base Stations on Metropolitan Streets," in *IEEE Int. Symp. Pers. Indoor Mob. Radio Commun.*, 2018.
- [5] S. Guntuka, E. M. Shakshuki, A. Yasar, and H. Gharra, "Vehicular Data Offloading by Road-Side Units Using Intelligent Software Defined Network," *Procedia Comput. Sci.*, vol. 177, pp. 151–161, 2020.
- [6] C. Song, J. Wu, W. S. Yang, M. Liu, I. Jawhar, and N. Mohamed, "Exploiting Opportunities in V2V Transmissions with RSU-assisted Backward Delivery," in *IEEE INFOCOM WKSHPs*, 2017.
- [7] G. Li, L. Boukhatem, and J. Wu, "Adaptive Quality-of-Service-Based Routing for Vehicular Ad Hoc Networks with Ant Colony Optimization," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3249–3264, 2017.
- [8] A. H. Sodhro, M. S. Obaidat, Q. H. Abbasi, P. Pace, S. Pirbhulal, A.-u.-h. Yasar, G. Fortino, M. A. Imran, and M. Qarage, "Quality of Service Optimization in an IoT-Driven Intelligent Transportation System," *IEEE Wirel. Commun.*, vol. 26, no. 2, pp. 10–17, 2019.
- [9] Y. Saleem, N. Mitton, and V. Loscri, "A Vehicle-to-Infrastructure Data Offloading Scheme for Vehicular Networks with QoS Provisioning," *Int. Wirel. Commun. Mob. Comput. Conf.*, 2021.
- [10] "Quality of Service (in Mobile Networks), Last accessed: March 2021." [Online]. Available: <https://www.tu-ilmenau.de/fileadmin/public/iks/files/lehre/mobicom/MCN-08-QoS.pdf>
- [11] A. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," *Proc. 1st Int. Conf. Simul. Tools Tech. Commun. Networks Syst.*, 2008.
- [12] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Trans. Mob. Comput.*, vol. 10, no. 1, pp. 3–15, 2011.
- [13] P. L. et al., "Microscopic Traffic Simulation using SUMO," in *IEEE Conf. Intell. Transp. Syst.*, 2018.