The Mobile Edge Computing (MEC)-based VANET Data Offloading using the Staying-Time-oriented k-hop away Offloading Agent

Chung-Ming Huang, Zhong-You Wu
Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan

Shih-Yang Lin
School of Transportation and Vehicle Engineering, Shandong University of Technology, Shandong, China

Correspondence: huangcm@locust.csie.ncku.edu.tw

Abstract — **This paper proposed a Mobile Edge Computing (MEC)-based k-hop away IEEE 802.11p Road Side Unit (RSU) Vehicular-to-Infrastructure (V2I) offloading method. Let a vehicle X regularly use LTE cellular network to communicate with the peer entity in the infrastructure side, i.e., in Internet. When there is a k-hop V2V path that can connect X with the ahead RSU, the V2I offloading can be enabled; additionally, when X has left the signal coverage of the corresponding RSU, if there is a k-hop V2V path that can connect X with the rear RSU, the V2I offloading can still be continued. In the proposed method, a MEC server is used to compute whether the k-hop V2I offloading can be enabled and maintained or not. This work also shows the performance analysis for different offloading scenarios in situations of different vehicular density.**

Keywords: VANET data offloading; Mobile Edge Computing (MEC); Road Side Unit (RSU); V2I and V2V.

I. INTRODUCTION

In the past decade, a lot of research has been done for vehicle ad hoc network (VANET), which is essentially based on IEEE 802.11p [1][2][3]. In the considered VANET's configuration, vehicles (i) communicate with other entities in the infrastructure side, i.e., Internet, through IEEE 802.11p Road Side Unit (RSU), which is called Vehicle-to-Infrastructure (V2I) communication, and (ii) communicate with each other directly using the Dedicated Short Range Communications (DSRC) concept without the involvement of the infrastructure side, i.e., through the IEEE 802.11p On Board Units (OBUs) that are on a sequence of vehicles, which is called Vehicle-to-Vehicle (V2V) communication. Since IEEE 802.11p RSUs are not deployed everywhere and thus it is not feasible to have the aforementioned V2I communication paradigm, i.e., all of the V2I communications are through IEEE 802.11p RSUs. Thus, the V2I communication still needs to be done through the use of LTE cellular network.

The scenario of the V2I communication that adopts mobile data offloading from LTE cellular network to IEEE 802.11p RSUs is as follows: 1) A vehicle X regularly uses LTE cellular network, i.e., through Base Station (BSs), to communicate with entities in the infrastructure side, i.e., in Internet. 2) When X is inside the signal coverage of an IEEE 802.11p RSU, X switches to use IEEE 802.11p network, i.e., through RSU. 3) When X is outside the signal coverage of the IEEE 802.11p RSU, X switches back to LTE cellular network. In this way, the IEEE 802.11p offloading for VANET can be achieved. This work considers the following functional scenario to enlarge the offloading effect:

1) Before vehicle X enters into the signal coverage of its ahead IEEE 802.11p RSU, if there is a k-hop V2V path between X and the signal coverage area of the ahead IEEE 802.11p RSU, i.e., the other end vehicle of the k-hop V2V path is inside the signal coverage of the ahead 802.11p RSU, then X can have the IEEE 802.11p offloading using the k-hop V2V path.

2) When vehicle X leaves the signal coverage of the IEEE 802.11p RSU, if there is a k-hop V2V path between X and the signal coverage area of the rear 802.11p RSU, i.e., the other end vehicle of the k-hop V2V path is inside the signal coverage of the rear 802.11p RSU, then X still can maintain the IEEE 802.11p offloading using the k-hop V2V path.

The key concern of the aforementioned functional scenario is how to find the k-hop V2V path, if it exists, before the vehicle enters into the signal coverage of the ahead RSU or after the vehicle leaves the signal coverage of the rear RSU and have the related actions as soon as possible. In this work, the Mobile Edge Computing (MEC) mechanism [4][5] is adopted to have the MEC-based 802.11p offloading using the k-hop V2V path.

The remaining part of this paper is organized as follows: Section II presents related works for offloading. Section III introduces the functional scenario of the MEC-based k-hop VANET offloading in detail. Section IV presents the proposed control schemes and algorithms for the k-hop VANET offloading. Section V gives the performance analysis results. Section VI has the conclusion remarks.

II. RELATED WORKS

In [6] the authors proposed a VANET offloading potential prediction (VOPP) model to calculate the offloading potential from LTE cellular to VANET. The VOPP is an analytical study based on the optimization

This work was supported by the Ministry Of Science and Technology (MOST), Taiwan (R.O.C.) under the grant number MOST 107-2221-E-006-139.

problem and uses link availability between the infrastructure and the downloader to find a routing path whose offloading volume is the maximum.

In [7] the authors proposed a Floating Car Data (FCD) application in LTE cellular and VANET network to efficiently execute the collection of geo-located information process. The FCD application scheme can disseminate messages across the VANET network and reduce the required LTE channel capacity for data transmission.

The authors in [8] proposed the formation of a real-time context aware ad-hoc collaboration system for emergency situations and remote robotic medical care using the combination of 5G and MEC technologies. The characteristics of 5G can be supported in an MEC server using APIs to create a context-aware ad hoc collaborations system. The benefit of the MEC collaboration platform with 5G is to effectively reduce the handover latency of data transmission and to avoid congestion at the core network.

The authors in [9] proposed the mobile-edge cloud-enabled vehicular networks using MEC servers, whose main role is to accomplish the computation tasks between remote clouds and local vehicular terminals. MEC servers receive the task-input messages from vehicles and then predict the processing time to finish the offloading tasks. Thus, the proposed MEC architecture can help reduce the offloading transmission costs and the delay time of the offloading process.

The authors in [10] tried to resolve the resource management issue for V2I cognitive radio-cloud-assisted access networks to tackle the mobility-induced abrupt changes. The target of the work was to utilize the V2I connection and the remote cloud computing for data offloading done by road side unit (RSU). The authors exploited the distributed and scalable traffic offloading scheme that has the adaptive controller to optimize energy and to manage the access time windows at the RSU side and traffic flows at the vehicular client's side.

III. THE FUNCTIONAL SCENARIO OF THE PROPOSED SCHEME

In this Section, an overview of the functional scenario is presented. Four main components in our proposed system architecture are MEC server, Vehicles, RSU and BS. In the proposed architecture, each BS is associated with an MEC server and the MEC server is in charge of those RSUs that are inside the signal coverage of the corresponding BS. Each vehicle is equipped with an OBU that has a LTE cellular network's interface and an IEEE 802.11p network's interface. Each vehicle can (1) use the LTE cellular network's interface to have V2I communication over the LTE cellular network or (2) use the IEEE 802.11p network's interface (i) to have V2I communication, which is through RSU, or (ii) to have V2V communication, which is through OBU. Each vehicle reports its context information (speed, direction, location, neighboring vehicles' IDs, IDs of sensed RSUs, etc.) to the MEC server using the LTE cellular network. The MEC server is in charge of checking whether the k-hop V2V path exists for a vehicle or not and notifies related entities to do the associated actions if the k-hop V2V path does exist. The functional scenario is as follows.

Initially, vehicle X is using the LTE cellular network to communicate with its peer P_X, which is in Internet, and all of the vehicles report their contexts to the MEC server periodically. When the MEC server finds a suitable offloading path for vehicle X, the MEC server sends messages to the constituent vehicles of the offloading path to help X forward data. All of the constituent vehicles of the offloading path report their contexts to the MEC server synchronously, for which the timing can be done through the use of GPS.

Five phases of the proposed MEC-based k-hop offloading method are as follows:

1) Phase 1: Find the initial k-hop V2V offloading path for vehicle X when X is communicating with its peer P_X, which is in the infrastructure Internet side, using the LTE cellular network and is driving toward an IEEE 802.11p RSU. A k-hop V2V offload path's selection algorithm is needed for phase 1.

2) Phase 2: When vehicle X is using a k-hop V2V offloading path, the offloading path needs to be modified periodically, for which the k-hop V2V offloading path normally needs to be shrank when the offloading agent is leaving the RSU's signal coverage. A Detection and Shrinking (DAS) algorithm is needed for phase 2.

3) Phase 3: When vehicle X is inside RSU's signal coverage, vehicle X connects with the RSU directly to have the offloading by itself.

4) Phase 4: When vehicle X has left RSU's signal coverage, i.e., RSU is on the rear of vehicle X and vehicle X is driving away RSU more and more, vehicle X can use a k-hop V2V offloading path that connects with X's rear RSU to continue the offloading. A Detection and Extending (DAE) algorithm is needed for phase 4.

5) Phase 5: When vehicle X is using a k-hop V2V offloading path, the offloading path may be disconnected unexpectedly because a constituent vehicle V_i drives away unexpectedly. Then, one of vehicle V_i's neighboring vehicles, which is also a constituent vehicle of the offloading path, tries to find a vehicle to repair the offloading path. A Path Recovery (PR) algorithm is needed for phase 5.

IV. THE STAYING TIME-BASED CONTROL SCHEME

In this Section, the proposed staying time-based control scheme is presented. The main principle is as follows: among those vehicles that are inside RSU's signal coverage, select the one that has the longest staying time inside RSU's signal coverage as the offloading agent; among those k-hop vehicular ad hoc paths between vehicle S, which wants to have the VANET offloading, and the offloading agent, selecting the one that has the longest offloading time, which is equal to the minimum of (i) the path life time of the k-hop vehicular ad hoc path between S and the offloading agent and (ii) the offloading agent's staying time inside RSU's signal coverage.

Phase 1. The Construction of the Initial Offloading Path

The Staying Time-based Selection (STS) algorithm uses a greedy way to find an offloading agent with the corresponding offloading path for vehicle X. Using the STS algorithm, the MEC server uses reported contexts from vehicles to (i) find the vehicle Y, which has the longest staying time among all vehicles that are inside the ahead RSU's signal coverage, as the offloading agent and (ii) construct the offloading path between vehicle X and offloading agent Y for X when vehicle X is outside RSU's signal coverage and is driving toward the ahead RSU. Algorithm 1 depicts the STS algorithm.

The MEC server uses the STS algorithm to construct the initial offloading path. For convenience, Fig. 1 is used to explain the operation of STS.

Algorithm 1 STS (S)

1: **for each** vehicle V that is in RSU_k **do**
2: Staying_Time $_{V2I}$(V) = *staying_time* (V, RSU)
3: **end for**
4: choose the vehicle V that has the maximum Staying_Time $_{V2I}$(V)
5: Selected_set(V) = {V}
6: Offloading_path = {S}
7: **for each** vehicle V_R that is between S and V and is V*'s* neighboring vehicle **do**
8: Path_life_time (V_R) = *connected_time* (V_R, V)
9: *In*(V_R) = V
10: **end for**
11: **repeat**
12: find vehicle X that is not in Selected_set(V) and Path_life_time (X) is the maximum of all Path_life_time (V_r)
13: add X to Selected_set(V)
14: **for each** neighbor Y of X **do**
15: **if** Y $\notin$ Selected_set(V) and min{ Path_life_time (X), *connected_time* (X,Y)} > Path_life_time (Y) **then**
16: *In*(Y) = X
17: Path_life_time(Y) = min {Path_life_time (X), *connected_time* (X, Y)}
18: **end if**
19: **end for**
20: **until S** $\in$ Selected_set (V) or all vehicle X is in Selected_set(V)
21: **if** Path_life_time (S) $\neq$ -1 **then**
22: Time_of_offloading_path (S, V) = min (Staying_time $_{V2I}$ (V), Path_life_time (S))
23: offload $_{\textbf{node}}$ = S
24: **repeat**
25: add *In* (offload $_{\textbf{node}}$) to Offloading_vehicle
26: offload $_{\textbf{node}}$ = *In* (offload $_{\textbf{node}}$)
27: **until** V $\in$ Offloading_path
28: **return** Offloading_ path
29: **else**
30: **return** null
31: **end if**

Let the connected time between two vehicles A and B denote the time length that A is inside B's OBU signal coverage and B is inside A's OBU signal coverage. Let the staying time of Y denote the time length that Y is inside the signal coverage of a RSU. In Fig. 1, each node denotes a vehicle and each link is associated with a number, which denotes the connected time length between two vehicles or the staying time of a vehicle inside RSU's signal coverage.

At first, STS decides the offloading agent, which has the longest staying time inside RSU's signal coverage among all vehicles that are inside RSU's signal coverage. Lines 1 to 3 of STS calculate the staying time of each vehicle inside RSU's signal coverage, in which function *staying_time* calculates the connected time of each vehicle V with the corresponding RSU.

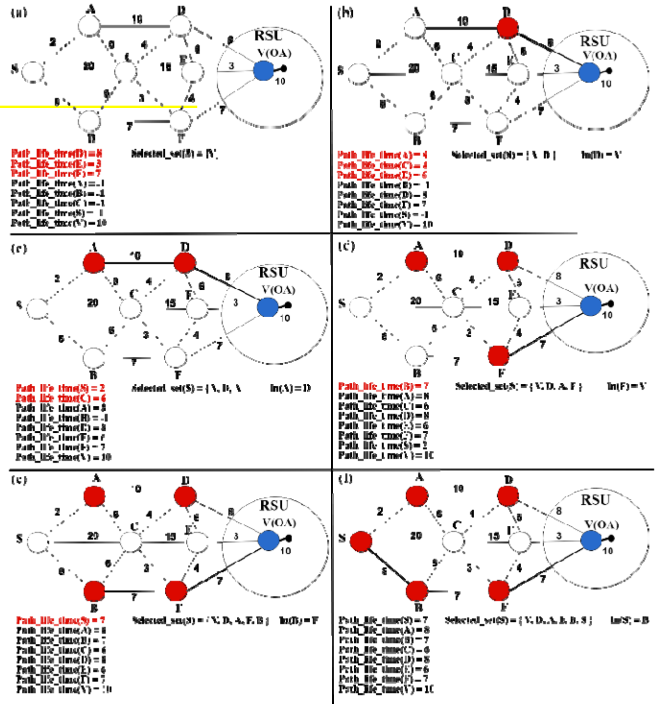


Fig. 1 An example of executing the STS algorithm.

Lines 4 to 6 of STS choose vehicle V that has the longest staying time as the offloading agent, put it in set *Selected_set(V)* and put the vehicle S that wants to have k-hop VANET offloading in set *offloading_path*. In the for loop of Lines 7 to 10, STS starts to search the offloading path by exploring all of vehicle V's neighboring vehicles, for which each candidate path's life time is equal to the connected time between V and each of its neighboring vehicle, the *connected_time* function returns the V2V connection duration, Fig. 1-(a) denotes the results of these actions. Lines 11 to 20, which are the key part of STS, explore the potential paths from V to S to find the one that has the longest path life time.

Referring to Fig. 1-(b), in the 1st iteration of Lines 11 to 20 of the STS algorithm, the MEC server selects vehicle D that has the maximum path life time, i.e., path (V, D) whose life time is 8 and is the maximum among path (V, D), path (V, E) and path (V, F), and add D to Selected_set after executing Lines 12 to 13 of the STS algorithm. Then, in Lines 14 to 19 of the STS algorithm, STS extends the path from vehicle D to each of its neighboring vehicles, which are A, C and E in Fig. 1, and updates the life time of the path from V to each of its neighboring vehicles A, C and E. If the minimum of path (V, D)'s life time and the connected time of D to the corresponding neighboring vehicle, i.e., the new path's life time, is longer then the corresponding neighboring vehicle's original path's life time, then the path and the path's life time are updated. As a result, *path_life_time* (A) is updated from -1 to 8 (V, D, A), *path_life_time* (C) is updated from -1 to 4 (V, D, C) and *path_life_time* (E) is updated from 3 (V, E) to 6 (V, D, E) in Fig. 1-(b). Note that the updating of the path life time only considers those neighboring vehicles for which the path's life time of each one of the corresponding extended paths is longer than the original one of the path from V to the associated neighboring vehicle.

After executing Lines 12 to 13 of the STS algorithm in the 2nd iteration, which is depicted in Fig. 1-(c), STS selects vehicle A to extend because A's associated path's life time is the longest one among those of (V, D, A), (V, D, C), (V, D, E) and (V, F), which are depicted in Fig. 1-(b) and are 8, 4, 6 and 7 respectively. That is, path (V, D, A) has the longest path's life time. Then STS extends the path to vehicle A's qualified neighboring vehicles, which are C and S, and updates the path's life time of these two extended paths, which are 6 (V, D, A, C) and 2 (V, D, A, S) respectively after executing Lines 14 to 17 of the STS algorithm.

After executing Lines 12 to 13 of the STS algorithm in the 3rd iteration, which is depicted in Fig. 1-(d), vehicle F is selected to extend the path because Path_life_time (F), which is 7, is longer than the other currently available ones, i.e., 2 (S), 6 (C) and 6 (E). Among F's neighboring vehicles, vehicle B is the qualified vehicle to be extended. Thus, the path's life time from V to B is updated to 7 and the path becomes (V, F, B) after executing Lines 14 to 17 of the STS algorithm. In the 4th iteration, which is depicts in Fig. 1-(e), vehicle B is selected to extend the path because Path_life_time (B), which is 7, is longer than the other currently available ones, i.e., 2 (S), 6 (C) and 6 (E). Among B's neighboring vehicles, vehicle S is the qualified vehicle to be extended because path's life time from V to S is updated from 2 (V, D, A, S) to 7 (V, D, B, S) after executing Lines 14 to 17 of the STS algorithm. Thus, the extended path (V, F, B, S) is considered in the next iteration. In the 5th iteration, which is depicted in Fig. 1-(f), vehicle S is selected and the iteration of Lines 11 to 20 of the STS algorithm is stop because the source S has been chosen.

After finding the k-hop ad hoc path, which is S↔ B↔ F↔ V, that has the longest path's life time from S to V, i.e., the OA, STS compares the corresponding path's life and the staying time of V to set the offloading path's offloading time, which is the smaller one of the aforementioned two values. Lines 21 to 22 of the STS algorithm are for the corresponding calculation. In Lines 23 to 31 of the STS algorithm, STS outputs the constituent vehicles of the offloading path to set offloading_path.

Phase 2. The Detection and Shrinking (DAS) mechanism

When the offloading agent of the current offloading path is leaving RSU's signal coverage, the MEC server needs to find the other offloading agent and modify the offloading path accordingly. Two main issues of the Detection and Shrinking (DAS) mechanism that need to be tackled are as follows:

1) When to trigger the re-selection of the new offloading agent?

2) How to select the new offloading agent and modify the offloading path accordingly?

The DAS mechanism is event-driven, which is triggered when the current offloading agent is going to leave RSU's signal coverage or leaves RSU' signal coverage unexpectedly. The agent re-selection method is as follows:

(i) Let $S \leftrightarrow V_1 \leftrightarrow \ldots \leftrightarrow V_i$ be those vehicles that are outside RSU's signal coverage of the offloading path, i.e., V_1 connects with vehicle S that is having the VANET offloading, $V_i \leftrightarrow \ldots \leftrightarrow V_k \leftrightarrow V_{OA}$ be those vehicles that are inside RSU's signal coverage of the offloading path.

(ii) The MEC server selects vehicle V_x that is inside RSU's signal coverage and has the longest staying time among those vehicles that are inside RSU's signal coverage as the new offloading agent.

(iii) The MEC server selects the sub-pathM (V_i, V_x) such that the offloading time of the modified offloading path $S \leftrightarrow V_1 \leftrightarrow \ldots \leftrightarrow V_i \leftrightarrow$sub-pathM($V_i$, V_x) is the maximum among all offloading time of offloading paths $S \leftrightarrow V_1 \leftrightarrow \ldots \leftrightarrow V_i \leftrightarrow$sub-pathP($V_i$, V_x), P = 1..n.

Phase 3: Offloading by the vehicle Itself

When vehicle X can receive RSU's signal, which denotes that vehicle X is inside RSU' signal coverage, vehicle X should connect with RSU and communicate with the peer PX through RSU. That is, vehicle X can have the RSU offloading by itself. Two main issues are as follows:

1) How to know that the offloading can be handled by the vehicle itself without the help of a k-hop V2V offloading path, i.e., when to terminate the help of the k-hop V2V offloading path?

2) When to trigger the help of the k-hop V2V offloading path for the follow-up offloading for which vehicle X is leaving and is to be away the corresponding RSU more and more?

For the 1st issue, when vehicle X can receive RSU's signal, which denotes that vehicle X can transmit/receive packets to/from the RSU directly, vehicle X notifies the MEC server that it can have the RSU offloading directly by itself. For the 2nd issue, it can be handled as follows:

(i) When vehicle X's next reporting context time point is outside RSU's signal coverage, the MEC server starts the DAE mechanism, which is depicted in phase 4, to find the k-hop V2V offloading path.

(ii) Let vehicle X's geo location on the next context reporting time point be still inside RSU's signal coverage. If vehicle X suddenly speeds up and thus is outside RSU's signal coverage before its next context reporting time point, then vehicle X would disconnect with RSU and need to notify the MEC server that it is not able to have the offloading by itself.

(iii) Then, the MEC server starts the DAE mechanism to find the k-hop V2V offloading path.

Phase 4: The Detection and Extending (DAE) mechanism

When the offloading agent of the current offloading path is leaving RSU's signal coverage, the MEC server needs to find the other offloading agent and modify the offloading path accordingly. Two main issues are as follows:

1) When to trigger the re-selection of the new offloading agent?

2) How to select the new offloading agent and modify the offloading path accordingly?

The timing that triggers the MEC server to execute the agent reselection procedure is based on the reporting time points of offloading agent's contexts:

(1) When the offloading agent's geo location on the next reporting time point is outside RSU's signal coverage, the MEC server starts to find the new offloading agent.

(2) Based on the unexpected event: Let the offloading agent's geo location on the next context reporting time point be still inside RSU's signal coverage. If the current offloading agent suddenly speeds up and thus is outside RSU's signal coverage before its next context reporting time point, then the current offloading agent would disconnect with RSU and need to notify the MEC server that

it is not able to be the offloading agent through the LTE cellular network.

(3) Then the MEC server will find a new offloading agent for the corresponding offloading path.

The DAE mechanism is event-driven, which is triggered when vehicle X is going to leave RSU's signal coverage or leaves RSU' signal coverage unexpectedly. The agent re-selection method is as follows:

i) Let $V_i \leftrightarrow \ldots \leftrightarrow V_1 \leftrightarrow S$ be those vehicles that are outside RSU's signal coverage of the offloading path, V_{OA} be the offloading agent that is inside RSU's signal coverage of the offloading path, i.e., V_{OA} connects with V_i.

ii) The MEC server selects vehicle V_x that is inside RSU's signal coverage and has the longest staying time among those vehicles that are inside RSU's signal coverage as the new offloading agent.

iii) The MEC server selects the sub-pathM (V_x, V_{OA}) from all of the possible sub-paths between V_x to V_{OA} such that the offloading time of the modified offloading path sub-pathM (V_x, V_{OA})$\leftrightarrow V_i \leftrightarrow \ldots \leftrightarrow V_1 \leftrightarrow S$ is the maximum among all offloading time of the offloading paths sub-pathP (V_x, V_{OA})$\leftrightarrow V_i \leftrightarrow \ldots \leftrightarrow V_1 \leftrightarrow S$, P = 1..n.

Phase 5: The Path Recovery mechanism

When the offloading path is constructed, a constituent vehicle V_{i+1} may speed up and driving away unexpectedly, then vehicle V_i, which is V_{i+1}'s neighboring vehicle, needs to find the other neighboring vehicles and repairs the offloading path accordingly. Two main issues are as follows:

i) When to trigger the path recovery to repair the offloading path?

ii) How to repair/modify the offloading path accordingly?

When the offloading path is constructed, let V_{i+1} be the constituent vehicle in the partial offloading path $V_i \leftrightarrow V_{i+1} \leftrightarrow V_{i+2}$ and V_{i+1} be driving away the OBU's signal coverage of V_i unexpectedly. The timing of the Path Recovery mechanism is triggered when V_i cannot receive V_{i+1}'s hello message. When V_i found V_{i+1} is driving away, V_i finds the other vehicle that is in OBU's signal coverage of V_i and also in OBU's signal coverage of V_i+2 to substitute V_i+1.

V. PERFORMANCE ANALYSIS

In our simulation environment, we use SUMO and NS-3 simulator to simulate a VANET offloading scenario. The SUMO provides urban vehicle's mobility with collision avoidance, customized road environments and randomize driving behaviors. The vehicle is driving in 5.0 km × 10 m region with multiple intersections for which SUMO is used to generate more realistic mobility patterns. Table I lists the parameters and their values in our simulation environment.

TABLE I THE OFFLOADING PARAMETERS

Parameter	Value
RSU coverage	300 m
Velocity limit	60 km/h
Lane	3
Vehicle generate time	200 s

Let one vehicle be the offloading source and communicate with its peer using LTE at the simulation beginning and all vehicles report their contexts to the MEC server periodically, When the offloading source reports its context, the MEC server uses the STS algorithm to finds an offloading agent and its corresponding offloading path. The simulation ends when the offloading source is leaving the simulation region. With the urban vehicular characteristic, the offloading source starts at a fixed position and move from left to right; the starting and ending positions of each of the other vehicles are randomly distributed using SUMO.

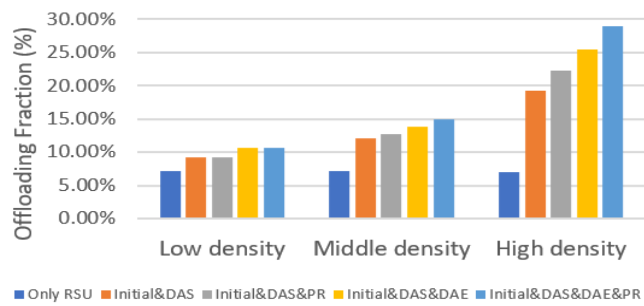
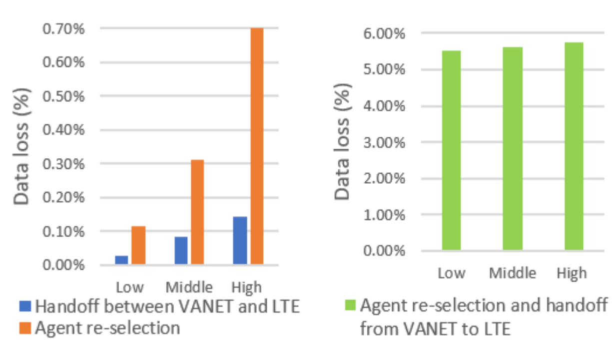


Fig. 2. Offloading fractions with difference vehicle's density.

In this paper, 5 scenarios that are compared are as follows: (1) the traditional offloading scenario for which the offloading source is allowed to have offloading only when it is inside RSU's signal coverage, (2) the offloading scenario consisting of phases 1/2/3, (3) the offloading scenario consisting of phases 1/2/3/5, (4) the offloading scenario consisting of phases 1/ 2/3/4, and (5) the offloading scenario consisting of phases 1/2/ 3/4/5 of the proposed method. That is, the adopted offloading function are Initial with DAS, Initial with DAS and Path Recovery, Initial with DAS and DAE and Initial with DAS and DAE and Path Recovery for scenarios 2, 3, 4 and 5 respectively. Each scenario was executed in situations of low/middle/high vehicle density for thirty times. The corresponding result is the average of these thirty times' experiments.

Fig. 2 depicts the offloading fractions for different situations of vehicle's density. Scenario 1 only has 7.1% of data can be offloaded. In the situation of low/middle/high density, the offloading fraction are (9.2%, 9.2%, 10.65%, 10.65%)/ (12%, 12.8%, 13.9%, 14.9%)/ (19.3%, 22.3%, 25.4%, 28.9%) for scenarios 2, 3, 4 and 5 respectively. Comparing with the traditional offloading method, i.e., scenario 1, the proposed k-hop offloading method can offload 1.29 to 1.5 / 1.69 to 2.09 / 2.71 to 4.07 times of the data in low/middle/high density's situation depending on the adopted method. The results show that the higher density it is, the more offloading fraction it has. The reason is that more vehicles can be forwarders and offloading agent when the density is increased and thus the probability of constructing a k-hop offloading path is also increased.



(a) (b)

Fig. 3. Loss fractions with difference vehicle's density

Fig. 3-(a) depicts the average data loss rate, which is equal to the lost packets resulted from (i) the handoff from LTE cellular network/k-hop 802.11p VANET to k-hop VANET /LTE cellular network, for which those data that are

stored and on-the fly in the k-hop VANET are lost in the later case, and (ii) the offloading agent re-selection, for which those data that are stored in the previous offloading agent and the associated forwarders that are inside RSU's signal coverage are lost, divides by the total transmitted data from the source to the peer for scenarios 2, 3, 4 and 5. The results show that the higher density it is, the more offloading loss it has. The reason is that the offloading probability becomes higher when the density is increased; the more chance to have offloading, the more chance to have the handoff and agent re-selection and thus the data loss rate becomes bigger. But, in fact, the loss rates are similar if it divides by the total transmitted data in the k-hop VANET, i.e., the total amount of the offload data, which is depicted in Fig. 3-(b).

VI. CONCLUSION

This paper has proposed a MEC-based k-hop away VANET offloading method, for which the data offloading can be executed k-hop away, before or after, the corresponding IEEE 802.11p RSU. Through the help of the MEC server, the k-hop V2V offloading path, including the constituent forwarders and offloading agent, can be derived and monitored. The simulation results have shown that, comparing with the traditional offloading method, i.e., a vehicle is allowed to have data offloading only when it is inside RSU's signal coverage, the proposed k-hop away VANET offloading method can have 1.29 to 4.07 times of data offloading depending on the adopted method and the vehicle's density; the increased data loss is about 0.7%. In this work, it only considers those vehicles that are inside an individual MEC server's domain. For the future work, it can consider the inter-MEC server's situation, i.e., the k-hop away V2V path is expanded into 2 BSs'/MEC servers' domains.

REFERENCES

[1] K. Abboud, H. A. Omar and W. Zhuang, "Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9457–9470, December 2016

[2] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang and Y. Zhou, "Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2377–2396, June 2015

[3] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang and J. Liu, "A Survey on Large-scale Software Defined Networking Testbeds: Approaches and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 891-917, November 2016

[4] A. C. Baktir, A. Ozgovde and C. Ersoy, "How Can Edge Computing Benefit from Software-Defined Networking: A Survey, Use Cases & Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359 - 2391.

[5] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628-1656, 2017.

[6] G. El Mouna Zhioua, J. Zhang, H. Labiod, N. Tabbane and S. Tabbane," VOPP: A VANET Offloading Potential Prediction Model," *in Proceedings of IEEE Wireless Communications and Networking Conference (WCNC),* pp.2408–2413, April 2014.

[7] P. Salvo, I. Turcanu, F. Cuomo, A. Baiocchi and I. Rubin, "LTE Floating Car Data application off-loading via VANET driven clustering formation," *in Proceedings of the 12th Wireless On-demand Network Systems and Services (WONS),* pp. 192–199, January 2016

[8] S. Nunna et al., "Enabling real-time context-aware collaboration through 5G and mobile edge computing," *in Proceeding of the 12th IEEE International Conference on Information Technology-New Generations (ITNG),* pp. 601-605, 2015.

[9] K. Zhang, Y. Mao, S. Leng, Y. He and Y. Zhang, "Mobile-Edge Computing For Vehicular Networks: A Promising Network Paradigm with Predictive Off-Loading," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36-44, 2017.

[10] N. Cordeschi, D. Amendola and E. Baccarelli, "Reliable adaptive resource management for cognitive cloud vehicular networks," *IEEE Transactions on Vehicular Technology,* vol. 64, no. 6, pp. 2528-2537, 2015.