Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

V2V Data Offloading for Cellular Network based on the Software Defined Network (SDN) inside Mobile Edge Computing (MEC) Architecture

Chung-Ming Huang[1], Senior, IEEE, Meng-Shu Chiang[2], Duy-Tuan Dao[1], Wei-Long Su[1], Shouzhi Xu[3] and Huan Zhou[3]

1 Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan
2 Computer Science and Information Engineering, Far East University, Tainan 70101, Taiwan
3 College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China

Corresponding authors: Chung-Ming Huang (huangcm@locust.csie.ncku.edu.tw)

This work was supported by the Ministry Of Science and Technology (MOST), Taiwan (R.O.C.) under the grant number MOST 105-2221-E-006-063 and Natural Science Foundation of Hubei Province of China under the grant number 2017CFB594).

ABSTRACT Data offloading plays an important role for the mobile data explosion problem that occurs in cellular networks. This paper proposed an idea and control scheme for offloading vehicular communication traffic in the cellular network to V2V paths that can exist in vehicular ad hoc networks (VANETs). A software defined network (SDN) inside the mobile edge computing (MEC) architecture, which is abbreviated as the SDN_i-MEC server, is devised in this paper to tackle the complicated issues of VANET V2V offloading. Using the proposed SDN_i-MEC architecture, each vehicle reports its contextual information to the context database of the SDN_i-MEC server, and the SDN controller of the SDN_i-MEC server calculates whether there is a V2V path between the two vehicles that are currently communicating with each other through the cellular network. This proposed method (1) uses each vehicle's context, (2) adopts a centralized management strategy for calculation and notification, and (3) tries to establish a VANET routing path for paired vehicles that are currently communicating with each other using a cellular network. The performance analysis for the proposed offloading control scheme based on the SDN_i-MEC server architecture shows that it has better throughput in both the cellular networking link and the V2V paths when the vehicle's density is in the middle.

INDEX TERMS Cellular Networks; Mobile Edge Computing (MEC); Software Defined Network (SDN); VANET Offloading; V2V Communication.

I. INTRODUCTION

With the advance of computing and communication technology, information and communication technology (ICT)-enhanced vehicles, i.e., autonomous vehicles, connected vehicles, and the internet of vehicles (IOV), are emerging. For the communication aspect, a vehicular ad hoc network (VANET) has been researched in the past years [1][2]. Three types of communication paradigms discussed in VANET are vehicle to infrastructure (V2I), vehicle to vehicle (V2V) and V2I2V. In the V2I communication paradigm, a vehicle (i) has the data downloaded from a remote server or (ii) communicates with a peer entity that is in the infrastructure side through the cellular network base stations (BSs) or the IEEE 802.11p-based road side unit (RSU). In the V2V paradigm, two vehicles communicate with each other directly through a k-hop vehicle ad hoc path. For the V2I2V communication paradigm, two vehicles V1 and V2 communicate with each other through the path that is V1<-> BS/RSU<->core network/backbone network <-> BS/RSU<->V2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2820679, IEEE Access

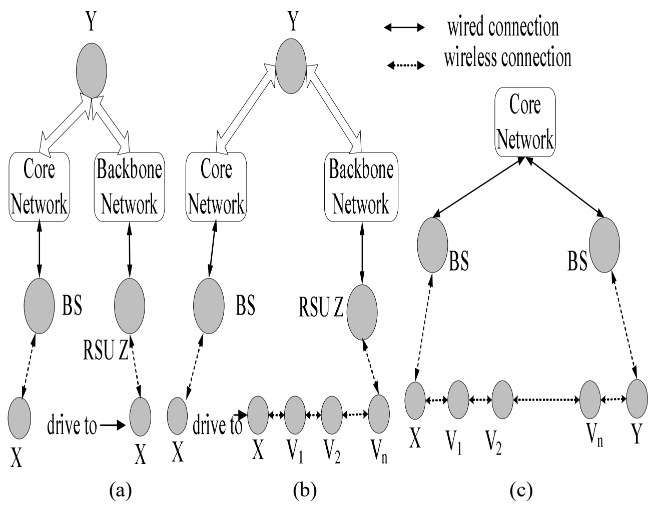


FIGURE 1. The configuration of VANET offloading.

Mobile data offloading [3][4] is a complementary technology of cellular networks. Using the offloading mechanism, the traffic in the cellular network and the expense for using the cellular network can be reduced. Currently existing VANET offloading techniques are mainly used for the infrastructure mode, i.e., for the V2I communication paradigm [5][6]. The goal of VANET offloading is to deliver data, which were originally through the cellular network, through VANET. Referring to Figure 1-(a), vehicle x is communicating with peer y, which is on the infrastructure side, using the cellular network normally. When X is in the signal coverage of an 802.11p RSU, X switches to connect with the RSU, and X switches back to connect with the BS when it is out of the 802.11p RSU's signal coverage. In [6], the authors proposed an expediting offloading model, whose transmission is through V2I and V2V, to offload more data packets. Referring to Figure 1-(b), when there is a V2V path that can connect vehicle x and RSU z, x switches to the VANET network to communicate with y, in which the communication path is $x \leftrightarrow v_1 \leftrightarrow v_2 \leftrightarrow \ldots \leftrightarrow v_n \leftrightarrow z \leftrightarrow y$. For convenience, the aforementioned scenarios are hereafter called V2I VANET offloading. In contrast, the scenario of the V2I2V offloading is considered in this work. Referring to Figure 1-(c), let the 802.11p OBUs of vehicles x and y, which are communicating with each other, be equipped with a cellular network interface and an 802.11p network interface. Vehicles x and y normally use the cellular network interfaces of their OBUs, which belong to the V2I2V communication. When there is a V2V path, which is $v_1 \leftrightarrow v_2 \leftrightarrow \ldots \leftrightarrow v_n$ in Figure 1-(c), between x and y, it can offload their cellular-based communication to the 802.11p-based V2V communication. For convenience, it is hereafter called V2V VANET offloading.

Main concerns that should be considered in V2V VANET offloading for the cellular network are as follows:

1) How to find that there is a V2V path between vehicles x and y that are communicating with each other using the cellular network? Can the finding of the V2V VANET path be made by a third party such that the computing overhead can be reduced and vehicles x and y can switch to the V2V path as soon as possible?
2) How to find the better V2V path for VANET offloading when there are more than one V2V paths that exist between vehicles x and y?
3) Is it possible to repair/recover a V2V path that is disconnected such that the V2V VANET offloading still can be kept?

With the availability of high bandwidth computer communication networks and the popular use of handheld devices, cloud computing and cloud servers are adopted in both wired and wireless networking environments. However, many applications and services in the wireless mobile environment, e.g., VANET, need shorter latency to have real time and more reliable response. Thus, the new computing paradigm called Mobile Edge Computing (MEC), for which some MEC servers are installed and available in the network edge instead of relying on the computing of the remote cloud server, was proposed to tackle the aforementioned real time requirements. Some standard bodies, e.g., 3GPP and ETSI, are defining services, architectures and APIs for MEC [8]-[10]. Additionally, many researchers have studied different aspects of MEC [11]-[13], and many the current study is for proposing architectures and/or derivation and analyzing the computation offloading from mobile nodes to MEC servers. Recently, some researchers are trying to apply MEC for VANET [14]-[17].

Software-Defined Networking (SDN) defines an abstract model for data transmission [18]. In SDN, the control plane and data plane are decoupled, for which the former sets some rules to the corresponding events for network routing and the latter is used to forward data packets according to the rules set by the control plane. The controller, which is a centralized control management for SDN, is in charge of collecting the information of those nodes inside its domain and then setting the rules accordingly.

In this paper, a V2V VANET offloading method based on the SDN inside the MEC (SDN_i-MEC) architecture is proposed to resolve the aforementioned issues for V2V VANET offloading in the highway environment. Using the SDN_i-MEC architecture, each vehicle X reports/transmits its context, including location, speed, direction and IDs of the neighboring vehicles that X can receive their beacons, to the context database of the SDN_i-MEC server.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

The SDN controller of the SDN_i-MEC server can keep calculating and exploring whether there are some V2V paths between vehicles *x* and *y* that are communicating with each other using the cellular network based on the received contexts, which are stored in the context database of the SDN_i-MEC server. If there is a V2V path between vehicles *x* and *y*, the SDN_i-MEC server will notify the peered vehicles *x* and *y* to switch from the cellular network to the VANET network for V2V VANET offloading. Figure 2 depicts the brief SDN_i-MEC architecture for V2V VANET offloading, for which each BS is a macro cell that is associated with an SDN_i-MEC server. Note that neighboring SDN_i-MEC servers can exchange the stored contents of their context databases such that more V2V paths can be explored.

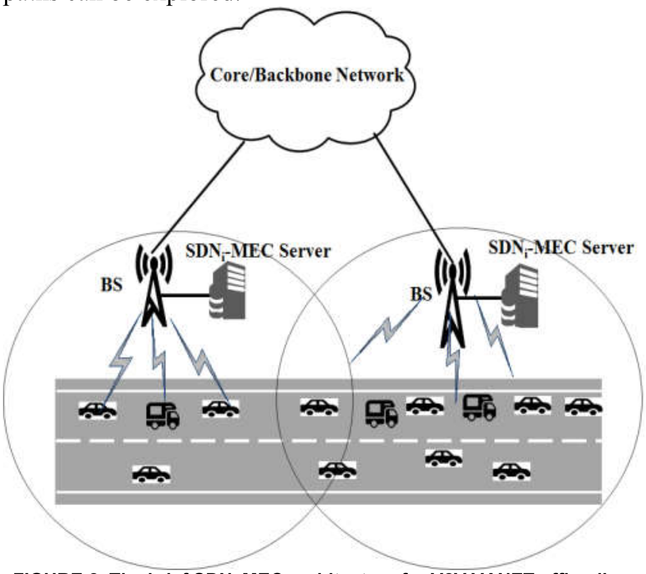


FIGURE 2. The brief SDN_i-MEC architecture for V2V VANET offloading.

To tackle the aforementioned problems that exist in V2V VANET offloading, this paper proposed a LifeTime-based Network State Routing (LT-NSR) algorithm and a LifeTime-based Path Recovery (LT-PR) algorithm that can be executed in the SDN controller of the SDN_i-MEC server. Using the proposed LT-NSR and LT-PR algorithms, all vehicle relationships are represented as a graph, in which (1) a node denotes a vehicle, (2) a link between two nodes denotes that these two vehicles are in mutual signal coverage and thus they can connect and communicate with each other directly, and (3) the weight of each link denotes the connection lifetime of the link, which can be derived based on their locations, speeds, and directions. Then, if there is a V2V ad hoc path between two peered vehicles, it means that they have a V2V routing path between them. When there is more than one V2V routing path, then select the path that has the longest connection lifetime as the V2V VANET offloading path.

The remaining part of this paper is organized as follows: Section II presents related works for V2V VANET offloading, Section III introduces the functional scenario of V2V VANET offloading in detail. Section IV presents the proposed LT-NSR and LT-PR algorithms and the control scheme for V2V VANET offloading. Section V gives results of the performance analysis. Section VI has the conclusion remarks.

II. RELATED WORK

In [19], the authors proposed a cooperative traffic optimization formulation in a joint 4G and VANET environment to evaluate how much traffic can be offloaded from a 4G cellular network and then they formulate the offloading decision to quantify the maximum data content using the V2I and V2V communication. The authors defined a solution to determine the data flows that could be retrieved through a VANET by considering the maximum fraction of data flows that have to be pre-fetched. H. Labiodet al [20] proposed a model named VOPP to determine the offloading potential of using VANET from the cellular network. It presented an analytical study by evaluating the extent to which a VANET could offload mobile data. The method considers the link availability between the infrastructure and the *downloader* for cellular traffic offloading. In [21], the authors provided an analytical study to discuss the capacity of the I2V and V2V links. The *downloader* requests content through VANET and offloads the data volume. At the same time, the proposed model can make a decision by considering the link quality. The aforementioned approaches [19]-[21] are fully distributed; thus, the detection of VANET offloading is evaluated by the vehicles themselves.

The author in [14] 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 to an MEC server using APIs to create a context-aware ad hoc collaboration system. The benefit of the MEC collaboration platform with 5G is to effectively reduce the handover latency data transmission and to avoid congestion at the core network. However, there is no performance analysis in the paper. The authors in [15] proposed an application-specific concept for a scalable SDN-enabled vehicular networking that is assisted by mobile edge computing (MEC) to provide reliable communication services over a heterogeneous V2X communication. An MEC server can support the cloud computing resources and provide delay-constrained feedback from clients and thus MEC can help reduce the round-trip time for packet transmission. The scalability and reliability of the proposed technique are examined by a case study in urban traffic management where traffic congestion always happens.

The authors in [16] 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 [17] have resolved the resource management issue for V2I cognitive radio-cloud-assisted access networks to tackle the mobility-induced abrupt changes. The target of this paper was to utilize the V2I connection and the remote cloud computing for data offloading done by road side units (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 vehicle client side.

Greedy Perimeter Stateless Routing (GPSR) [22] is one of the famous Geographic Routing protocols, which can search the nearest neighbor that is closest to destination and then forward packets. GPSR tries to solve the local maximum problem using greedy forwarding and the Right-Hand Rule. An improvement of GPSR was proposed in [23], in which authors added the PRedict protocol to predict vehicles' movements in order to improve the performance of traditional GPSR. In [24], an Energy Ability algorithm called EA-GPSR was proposed to optimize Greedy perimeter stateless routing for wireless network.

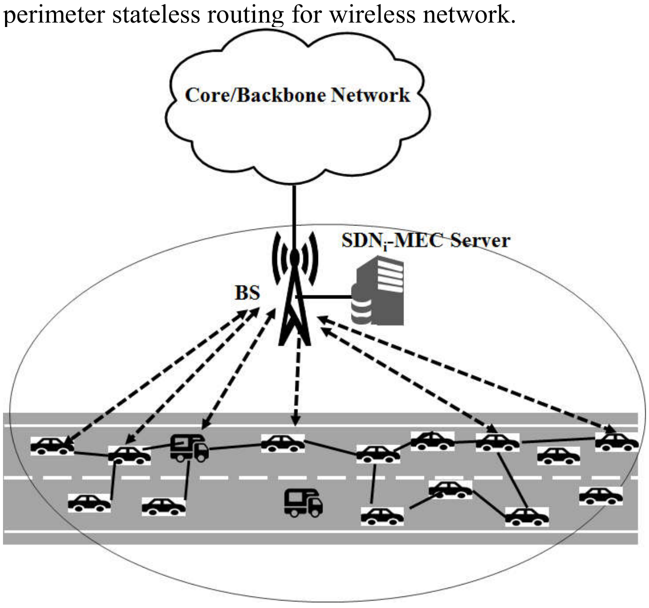


FIGURE 3. An illustrate V2V VANET offloading configuration uses SDN$_i$-MEC architecture

III. THE FUNCTIONAL SCENARIO OF THE PROPOSED METHOD

In this Section, an overview of the functional scenario of the proposed method is presented. For a convenient explanation, the illustrated case depicted in Figure 3 is used as an example. In the proposed architecture, each vehicle is equipped with a cellular network interface and an IEEE 802.11p network interface. Vehicles can send their context, including the location that is derived using GPS, speed, direction and IDs of those neighboring vehicles that can be sensed, i.e., those vehicles that are in proximity and whose beacons can be received, to the context database of the SDN$_i$-MEC server through BSs, which are equipped with the wireless OpenFlow protocol [25].

The functional scenario is as follows, and the corresponding execution procedure is depicted in Figure 4. Initially, let the peered vehicle x and vehicle y that want to communicate with each other use the cellular network. Vehicles x, y and others that want to adopt the V2V VANET offloading function report/transmit their contexts to the SDN$_i$-MEC server periodically.

1) SDN Controller of the SDN$_i$-MEC server keeps checking whether there is a suitable V2V path, which is $v_1 \leftrightarrow v_2 \leftrightarrow \ldots \leftrightarrow v_n$, between vehicles x and y or not. For example, the path $v_1 \leftrightarrow v_2 \leftrightarrow v_3$ that exists between x and y in Figure 3.

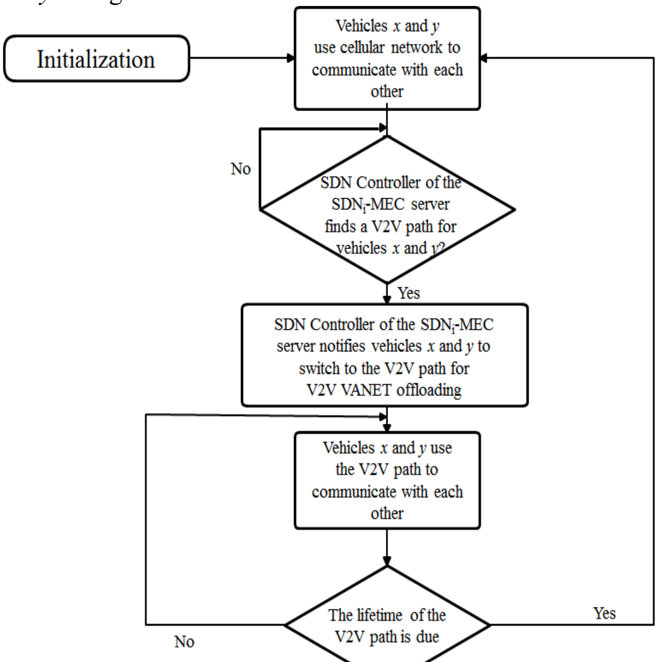


FIGURE 4. The execution procedure of having the V2V VANET offloading.

2) If it exists, then it denotes that v_1, v_2, …, v_n can play the relay nodes that are able to forward packets from vehicle x/y to vehicle y/x. Thereafter, SDN Controller in the SDN$_i$-MEC server notifies (i) vehicles x, v_1, v_2, …, v_n, y to establish a V2V path in their routing tables and (ii) vehicles x and y to be able to use the V2V path ($x \leftrightarrow v_1 \leftrightarrow v_2 \leftrightarrow \ldots \leftrightarrow v_n \leftrightarrow y$) for their communication.
3) Vehicles x and y then switch their communication path from the cellular network to the VANET network. At the same time, all of the vehicles x, v_1, v_2, …, v_n and y keep uploading/transmitting their context to the context database of the SDN$_i$-MEC server through BSs.
4) The V2V VANET offloading is ended (i) when the expected lifetime of path ($x \leftrightarrow v_1 \leftrightarrow v_2 \leftrightarrow \ldots \leftrightarrow v_n \leftrightarrow y$) is due, (ii) when the path is broken earlier, which results from one or more vehicles change their speed, or (iii) when the path is broken suddenly because a vehicle changes its direction and speed very fast.

5) When the V2V VANET offloading is ended, vehicles *x* and *y* switch the communication path from the V2V VANET network back to the cellular network. The execution procedure is changed to step 2).

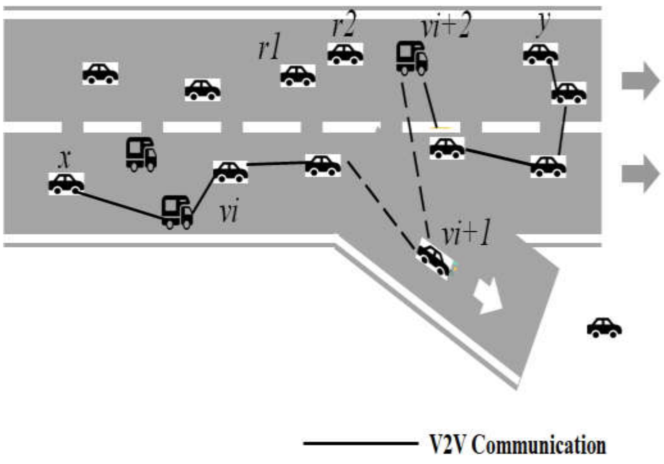


FIGURE 5. An illustrated example of the exception handling.

A vehicle may change its speed or direction, e.g., detour to the other road in the road interaction point, which results in breaking the V2V path. An illustrated example is depicted in FIGURE 5. Some recovery actions can be devised to tackle the exception handling. Let vehicle v_{i+1} be going to run away and vehicle v_i be the previous vehicle of v_{i+1} and vehicle v_{i+2} be the next vehicle of v_{i+1} in the V2V path. A mechanism can be devised to try to recover/repair the broken V2V path by replacing the drive-away vehicle v_{i+1} with the other neighboring vehicle, which can communicate with v_i and v_{i+2} directly, such that the V2V path can be kept. Note that the lifetime of the recovered/repaired V2V path may become longer or shorter after the path is recovered/repaired. The execution procedure of the path recovery is depicted in Figure 6 and explained as follows:

1) In the process of packet transmission, v_i is aware that the v_{i+1} runs away. At this time, v_i sends the repaired message to the SDN$_i$-MEC server through cellular network to repair the V2V path.
2) SDN Controller of the SDN$_i$-MEC server checks whether there is a repaired path ($v_i \leftrightarrow r_{i+1}^{new} \leftrightarrow v_{i+2}$), which means that r_{i+1}^{new} can communicate with v_i and v_{i+2} directly, to replace the broken path or not.
3) If it exists, then r_{i+1}^{new} replaces v_{i+1} and the V2V path between vehicles x and y can be kept, i.e., the path becomes $x \leftrightarrow v_1 \leftrightarrow v_2 \leftrightarrow \ldots v_i \leftrightarrow r_{i+1}^{new} \leftrightarrow v_{i+2} \leftrightarrow \ldots \leftrightarrow v_n \leftrightarrow y$.
4) If it doesn't exist, then vehicle *x* and *y* switch the communication path from the V2V VANET network back to the cellular network.

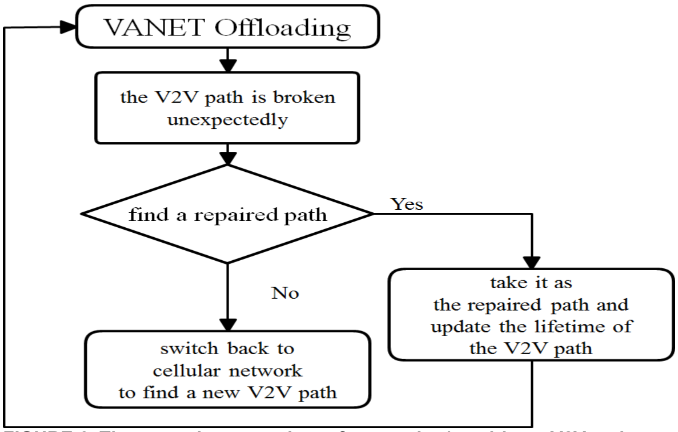
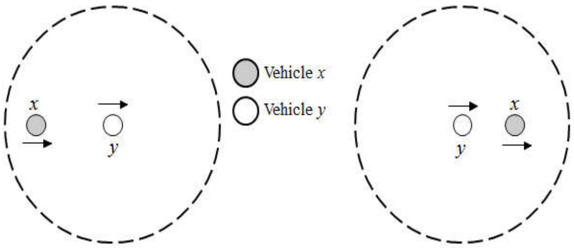


FIGURE 6. The execution procedure of recovering/repairing a V2V path.

Thus, the key issues are (1) how to find whether there is a V2V path for vehicles x and y based on the received context reported/transmitted from all of the vehicles that want to have the V2V offloading or not, (2) how to find the better V2V path when there are many paths and (3) how to repair/recover a broken path? In this work, the LifeTime-based Network State Routing (LT-NSR) algorithm and the LifeTime-based Path Recovery (LT-PR) algorithm are proposed to resolve the aforementioned 3 issues. Details of these two algorithms are presented in Section IV.

IV. THE PROPOSED METHOD

This Section presents our proposed method in detail. In the proposed method, each vehicle x reports/transmits its context, containing (1) the location derived using GPS, (2) speed, (3) direction, (4) IDs of the vehicles that x can detect beacons from, through BSs, to the context database of the SDN$_i$-MEC server periodically. The SDN$_i$-MEC server can generate rules for vehicles using messages and events. Table I and Table II contain the messages and events that are used. The messages include the initialized message (V2C), the reporting message (V2C), the switch message (C2V), the repaired message (V2C), the dropped message (V2C) and the end message (C2V); the events include the packet sending event, the reporting event, the VANET path event, the recovery event, the packet dropping event and the lifetime end event.



$V_x > V_y$	$\frac{r_{obu}+d_{x,y}}{V_x-V_y}$
$V_x < V_y$	$\frac{r_{obu}-d_{x,y}}{V_y-V_x}$

(a)

$V_x > V_y$	$\frac{r_{obu}-d_{x,y}}{V_x-V_y}$
$V_x < V_y$	$\frac{r_{obu}+d_{x,y}}{V_y-V_x}$

(b)

FIGURE 7. The configurations for connection's lifetime derivation.

TABLE I
MESSAGES AND PARAMETERS USED IN VEHICLES

Message	Parameters
Initialized Message (V2C)	(source, destination)
Reporting Message (V2C)	(location, speed, IDs of neighboring vehicles)
Switch Message (C2V)	(source, destination and relay vehicles' IDs)
Repaired Message (V2C)	(relay, forward, broken node)
Dropped Message (V2C)	(source, destination)
End Message (C2V)	(source, destination and relay's routing tables)

TABLE II
TRIGGERED EVENTS AND EXPLANATIONS

Triggered Event	Explanations
Packet Sending Event	The event is used for V2V data packet's transmission
Reporting Event	The event is used for reporting context
VANET Path Event	The event is used to switch network state
Recovery Event	The event is used to recover a V2V path which is broken
Lifetime End Event	The event is used when the lifetime of a V2V path is ended

Let $lt(u, v)$ denote the connection lifetime between vehicles u and v, where $lt(u, v) \geq 0$. After receiving the reported vehicle contexts, the SDN Controller of the SDN_i-MEC server can derive the connection lifetime $lt(x, y)$ between vehicles x and y. The connection lifetime $lt(x, y)$ between any two vehicles is varied and decided by their relative distance and relative velocity. Let vehicle x denote the sender and vehicle y denote the receiver. Referring to Figure 7 and TABLE III, five cases of the connection lifetime between vehicles x and y are as follows. Case 1: Vehicle x is behind vehicle y, and x can catch up with y in the same direction ($V_x > V_y$); referring to Figure 7-(a), the relative distance is $r_{obu} + d_{x,y}$ and the relative velocity is $V_x - V_y$. In this case, x is getting close to y until the relative distance $d_{x,y} = 0$, and then x is ahead of y thereafter. Case 2: When x is behind y, and they are leaving each other in the same direction ($V_x < V_y$); referring to Figure 7-(a), the relative distance is $r_{obu} - d_{x,y}$, and the relative velocity is $V_y - V_x$. Case 3: y is behind x, and they are leaving each other in the same direction ($V_x > V_y$); referring to Figure 7-(b), the relative distance is $r_{obu} - d_{x,y}$, and the relative velocity is $V_x - V_y$. Case 4: x is ahead of y, and y can catch up with x in the same direction ($V_x < V_y$); referring to Figure 7-(b), the relative distance is $r_{obu} + d_{x,y}$, and the relative velocity is $V_y - V_x$. Case 5. When x keeps the same distance with y, and they are in the same direction ($V_x = V_y$); $lt(x, y) = \infty$.

TABLE III
NOTATIONS USED IN THE PROPOSED SCHEME

r_{obu}	The coverage of an On-Board Unit (OBU) (m)
$d_{x,y}$	The relative distance between x and y (m)
V_i	The velocity of vehicle i (m/s)
$lt(u, v)$	The connection's lifetime between vehicle u and vehicle v (s)
V	The sequence of set
$P(s, d)$	The path from vehicle s to vehicle d
$P_{lt}(s, d)$	The lifetime of the path that is from vehicle s to vehicle d (m)

A. LIFETIME-BASED NETWORK STATE ROUTING (LT-NSR)

Let vehicles s and d start to communicate with each other using the cellular network. When vehicles transmit packets through cellular networks, the source vehicle s sends an initialized message, which represents a packet sending event, to the SDN_i-MEC server. This message/event triggers the SDN controller of the SDN_i-MEC server to find a V2V routing path from source vehicle s to destination node d. Hereafter, (i) data packets between s and d are transmitted and (ii) s, d, and other relay nodes report/transmit their contexts (location, speed, direction and IDs of neighboring vehicles) periodically to the context database of the SDN_i-MEC server using the reporting message, which is through the cellular network.

Let $P(v_1, v_n)$ denote a path that consists of a sequence of vehicles V = $\{v_1, v_2,..., v_n\}$, and the lifetime between the paired vehicles i and $i+1$ is denoted as $lt(v_1, v_2)$, $lt(v_2, v_3)$,..., $lt(v_{n-1}, v_n)$ respectively, i.e., $lt(v_i, v_{i+1})$, i = 1,..., n-1. The lifetime between s and v_1 is denoted as $lt(s, v_1)$, and the lifetime between v_n and d is denoted as $lt(v_n, d)$. $P_{lt}(s, d)$ denotes the connection lifetime of path $P(s, d)$ through $s, v_1, v_2, \ldots, v_n, d$, which is derived as follows:

$$P_{lt}(s, d) = \min\{ lt(s, v_1), lt(v_1, v_2), \ldots, lt(v_{n-1}, v_n), lt(v_n, d) \}$$

Since there are many vehicles running on the road at any time, there may not be just one V2V routing path between vehicle s and d. To find and ensure that the path is optimal, the LifeTime-based Network State Routing (LT-NSR) algorithm is proposed. The SDN Controller of the SDN_i-MEC server can use LT-NSR to derive the optimal k-hop V2V routing path, for which the k-hop V2V routing path has the longest lifetime based on the current VANET situation.

Algorithm 1 LT-NSR (s, d)

```
for each node v do // set the connection's lifetime of each neighbor of source s
    if v is a neighbor of s then
        D(v) = lt(s, v)
    else
        D(v) = -1
    end if
end for
S' = {s}
repeat
    find w not in S' such that D(w) is the maximum among all D(v)
    add w to S'
    // update nodes' lifetime
    for each neighbor u of w do
        if u ∉ S' and min{ D(w), lt(w, u) } > D(u) then
            D(u) = min{ D(w), lt(w, u) }
        end if
    end for
until d ∈ S'
return S'
```

Let $D(v)$ denote the set of connection lifetimes for all paths $P_{lt}(s, v)$ between vehicle s and each of the vehicles that have been considered and let S' denote a set of those vehicles that have been currently selected as member vehicles of the path. Algorithm 1 is the pseudo code of the LT-NSR algorithm.

At the beginning of exploration, in which the S' set only contains source s, it derives $D(v)$ for each neighbor v of s. After deriving the lifetime of each link, LT-NSR chooses

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2820679, IEEE Access

vehicle w, which is not in S', and $D(w)$ is the maximum among all $D(v)$, and then w is added to S'. Once vehicle w is added to the S' set, each of w neighboring nodes that are not in the S' set needs to update the lifetime of the path between s and itself. LT-NSR explores the topology repeatedly until destination d is contained in S'.

Theorem: *The offloading path constructed by LT-NSR(s,d) has the maximum lifetime of all offloading paths from s to d.*

Proof:

Let path $P_{final} = s \rightarrow V_1 \rightarrow V_2 \rightarrow \ldots \rightarrow V_k \rightarrow d$ be the path whose $lt(P_{final})$ has the longest path's lifetime. Let path $P_x = s \rightarrow V_1 \rightarrow V_2 \rightarrow \ldots \rightarrow V_m \rightarrow V_{m'}$, $m < (k-1)$, be a path that has been stop exploring because its path's lifetime $lt(P_x)$ is smaller than $lt(P_{final})$.

Let there be a path $V_{m'} \rightarrow V_{m'+1} \rightarrow \ldots \rightarrow d$.

(1) If $lt(P_x) < lt(V_{m'} \rightarrow V_{m'+1} \rightarrow \ldots \rightarrow d)$, then $lt(s \rightarrow V_1 \rightarrow V_2 \rightarrow \ldots \rightarrow V_m \rightarrow V_{m'} \rightarrow V_{m'+1} \rightarrow \ldots \rightarrow d) = lt(P_x)$, which is smaller than $lt(P_{final})$.

(2) If $lt(P_x) > lt(V_{m'} \rightarrow V_{m'+1} \rightarrow \ldots \rightarrow d)$, then $lt(s \rightarrow V_1 \rightarrow V_2 \rightarrow \ldots \rightarrow V_m \rightarrow V_{m'} \rightarrow V_{m'+1} \rightarrow \ldots \rightarrow d) = lt(V_{m'} \rightarrow V_{m'+1} \rightarrow \ldots \rightarrow d)$, which is smaller then $lt(P_x)$, and $lt(P_x)$ is smaller than $lt(P_{final})$.

Thus, $P_{final} = s \rightarrow V_1 \rightarrow V_2 \rightarrow \ldots \rightarrow V_k \rightarrow d$, which is found by *LT-NSR(s, d)*, is the path that has the longest path's lifetime from s to d.

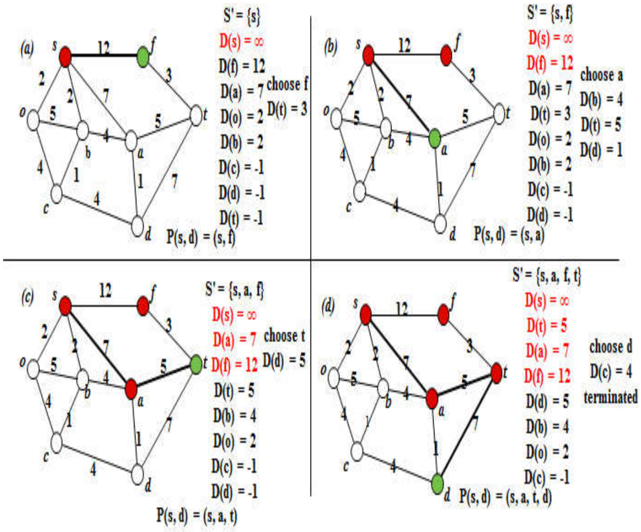


FIGURE 8. An example of executing the LT-NSR algorithm.

An example of using LT-NSR is depicted in Figure 8. Referring to FIGURE 8, the LT-NSR algorithm is expected to find a V2V routing path from source s to destination d. In the initialization stage, the currently known nodes are o, b, a, f, which are neighbors of s. The $D(v)$ of neighbors of s are computed directly, and the others are set to -1 because they are not directly reachable from s.

In the 1st iteration, which is depicted in FIGURE 8-(a), LT-NSR chooses node f and adds f to the S' set because $D(f)$ is the maximum among all $D(v)$. At this time, the lifetime of the path from s to each of the s neighboring nodes, which is t in Figure 8, is computed and then updated. The path (s, f, t) contains two pieces of connection lifetime, which are $lt(s, f) = 12$ and $lt(f, t) = 3$. $D(t)$ is taken from the $\min\{D(f), lt(f, t)\}$, which is in the illustrated topology depicted in FIGURE 8.

In the 2nd iteration, which is depicted in FIGURE 8-(b), node a is added into S' because $D(a)$ is the maximum among all $D(v)$. At the end of this iteration, node a is discovered and $D(b)$, $D(d)$, $D(t)$ are updated.

The exploration of the path $P(s, d)$ is ended in the 4th iteration, which is depicted in FIGURE 8-(d), and the selected path is (s, a, t, d), whose lifetime is 5.

The complexity of algorithm *LT-NSR (s, d)* is analyzed as follow. Let the number of vehicles between s and d be n. The *for* loop in Line 1 to Line 7 of *LT-NSR (s, d)* needs n iterations to finish. The *repeat* loop started from Line *9* is stop until d is included in S', which needs n iterations to finish; the *for* loop started from Line 13 needs n iterations to finish. As a result, the total complexity of the statements from Line *9* to Line *18* is $O(n*n)$ in the worst case. Thus, the complexity of *LT-NSR (s, d)* is $O(n^2)$.

B. V2V VANET OFFLOADING

When the SDN controller of the SDN_i-MEC server finds a routing path between s and d, it sets the new rule using the VANET path event and sends the switch message to inform vehicles. When the vehicles receive switch messages, they update their routing tables. When the connection between s and d is switched to a V2V path, the V2V path has its own connection lifetime that decreases over time. When the V2V path is ended, source s transmits the end message to the SDN_i-MEC server to trigger the lifetime end event. At that time, source s and destination d communicate with each other through the cellular network. Additionally, the SDN controller of the SDN_i-MEC server would try to find the other V2V routing path between s and d.

However, a V2V path may be broken because some vehicles run away unexpectedly. In this situation, the V2V path for V2V VANET offloading is broken and packets that are temporarily stored in relay vehicles should be dropped. The corresponding relay vehicle that dropped packets back-traces to source s and then s transmits the dropped message to the SDN_i-MEC server, which is denoted as a packet dropping event. This packet dropping event triggers the SDN controller of the SDN_i-MEC server to respond: FlowRemoved to remove the V2V path.

C. LIFETIME-BASED PATH RECOVERY (LT-PR)

The lifetime of the V2V path is derived and calculated by the SDN controller of the SDN_i-MEC server. In the process of offloading, some deviations may occur such that the V2V path is broken earlier then its originally expected lifetime.

Using the proposed recovery/repair method, the corresponding V2V path can be kept, which may result in extending or shortening the lifetime. LT-PR is triggered by the repair message, which represents the recovery event, sent to the SDN_i-MEC server. Algorithm 2 is the pseudo code for LT-PR.

Algorithm 2 LT-PR Pseudo Code

1: $R' = \{\}$
2: **for each** neighbor x of v_i **do**
3:     **if** x also receives v_{i+2}'s hello message **then**
4:         // x connects v_i and v_{i+2}
5:         add x into R'
6:     **end if**
7: **end for**
8: // Now there are one or more candidate vehicles can be selected
9: $R_{lt}(v_i, v_{i+2}) = lt(v_i, v_{i+2})$
10: $V_r \leftarrow v_{i+2}$
11: **for each** vehicle r in R' **do**
12:     **if** $R_{lt}(v_i, v_{i+2}) < \min\{ lt(v_i, r), lt(r, v_{i+2}) \}$ and $r \neq v_{i+1}$ **then**
13:         $R_{lt}(v_i, v_{i+2}) = \min\{ lt(v_i, r), lt(r, v_{i+2}) \}$
14:     $V_r \leftarrow r$
15:     **end if**
16: **end for**
17: **return** V_r

Let R' denote the set of candidate vehicles that can be selected, v_{i+1} denote the run-away vehicle, v_i denote the previous vehicle and v_{i+2} denote the next vehicle. Note that v_{i+1} may still receive the hello message from v_i *or* v_{i+2}, if v_{i+1} is still in the signal coverage of v_i *or* v_{i+2} or if v_i cannot communicate with v_{i+1} but v_{i+1} can still connect to v_{i+2} directly. However, vehicle v_{i+1} should not be a candidate because v_{i+1} is going to run away. These candidates are selected from the neighbors of v_i that are also the neighbors of v_{i+2}, i.e., those vehicles that can receive the hello messages from both v_i and v_{i+2}, and then are added into R'. There is a special case where v_i becomes able to connect to v_{i+2} directly. Thus, v_{i+2} is set as the target vehicle v_r and the temporary lifetime of this path is set as the lifetime between v_i and v_{i+2}, which is 0 if they are not connected directly, in lines 9-10. In lines 11-16, $lt(v_i, v_{i+2})$ is compared with other candidate path lifetimes and the maximum lifetime from these paths is $R_{lt}(v_i, v_{i+2})$, which denotes the lifetime of the repaired V2V path ($v_i \leftrightarrow v_r \leftrightarrow v_{i+2}$). If LT-PR cannot find a repaired V2V path, which means no candidate can be the repaired node, the packet dropping event is triggered by the SDN controller of the SDN_i-MEC server.

D. MESSAGE FLOW

FIGURE 9 shows the message flow of the proposed scheme. FIGURE 9-(a) depicts the message flow of the V2V offloading. At the beginning, each vehicle sends a hello message to a BS and the BS sends the HELLO message to the SDN_i-MEC server for the connection setup. When the connection is established, the SDN_i-MEC server sends back a HELLO message. Then, the communication between vehicle 1 and vehicle 2 begins. When vehicle 1 or 2 reports/transmits its context it is denoted as info. As shown in Figure 9, the BS sends PacketIn to the SDN controller of the SDN_i-MEC server. The PacketIn triggers the packet sending event to inform the SDN controller of the SDN_i-MEC server that a 4G/LTE path is established for data transmission between vehicle 1 and vehicle 2. In the OpenFlow protocol [25], each OpenFlow message has its own effort, e.g., construct, copy, compare, and print. The FlowMod message allows the SDN_i-MEC server to modify the states of the BSs, which are equipped with the OpenFlow protocol.

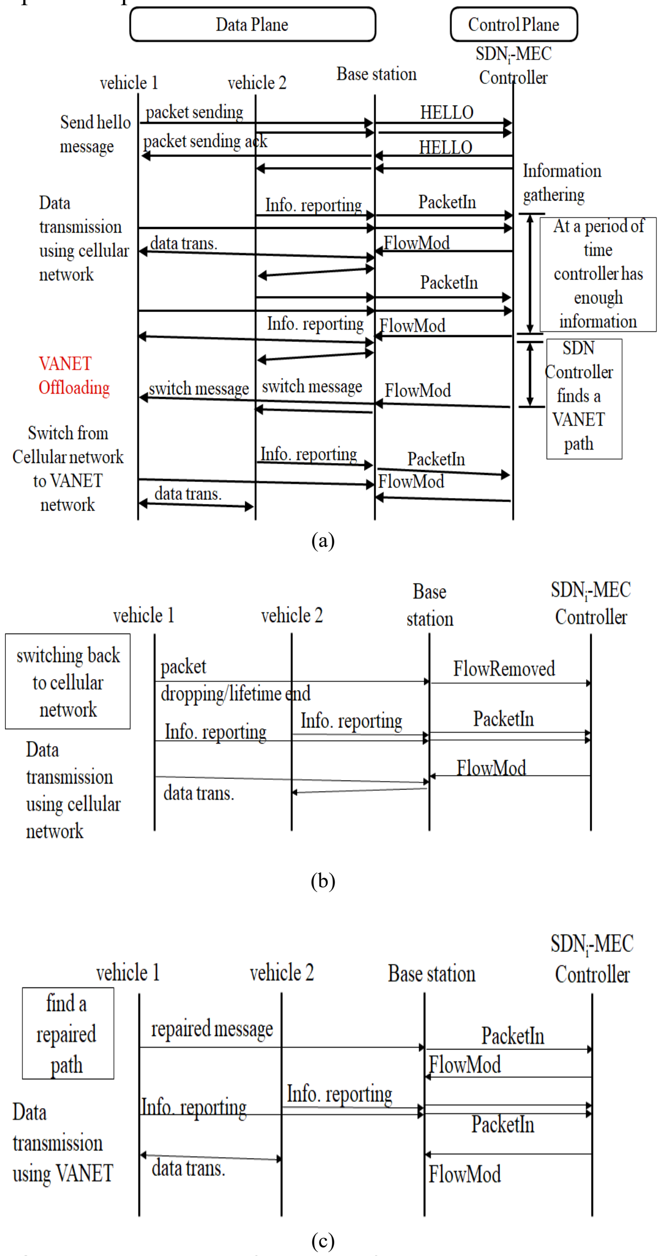


FIGURE 9. The message flow chart of proposed scheme. (a) The message flow of V2V VANET offloading. (b) The message flow of ending the V2V VANET offloading. (c) The message flow of disconnecting the V2V path unexpectedly.

The SDN_i-MEC server collects the reported context from the vehicles and then generates a graph for each pair of peered vehicles that are communicating with each other using the cellular network. The graph denotes the current network topology. Then, the SDN controller of the SDN_i-MEC server executes the proposed LT-NSR algorithm to find a V2V path between the peered vehicles for V2V VANET offloading. When a V2V path for V2V VANET offloading is found, the SDN controller of the SDN_i-MEC server sends FlowMod to modify the corresponding BS and the BS sends the switch message, which includes the V2V routing path, for modifying the corresponding vehicle routing tables.

FIGURE 9-(b) depicts the message flow of ending the V2V VANET offloading. When the V2V path ends, i.e., lifetime becomes 0, the vehicle sends the lifetime end message to the SDN_i-MEC server, which triggers the lifetime end event. This event also removes the V2V path using the FlowRemoved message. After receiving the message, the SDN controller of the SDN_i-MEC server sends the FlowMod message back to notify the peered vehicles 1 and 2 to use the cellular network to continue their communication.

FIGURE 9-(c) depicts the message flow of disconnecting the V2V path unexpectedly. When the V2V path is broken, the corresponding vehicle sends the repaired message to the SDN controller of the SDN_i-MEC server through a BS, which triggers the recovery event. After a repaired V2V path is found, the drive-away vehicle is replaced with the repaired one, which connects the previous vehicle v_i and the next vehicle v_{i+2}. Therefore, the corresponding routing path can be kept. The SDN controller of the SDN_i-MEC server sends the FlowMod message to modify the corresponding routing tables of the vehicles.

When the V2V path for V2V offloading is disconnected, the peered vehicles switch back to the cellular network. Thereafter, the SDN controller of the SDN_i-MEC server tries to find a new V2V path between the peered vehicles for V2V VANET offloading.

V. PERFORMANCE ANALYSIS

To evaluate the proposed method, the NS3 simulator and the VANET-highway mobility model [26][27] are used to simulate the V2V VANET offloading scenario. This highway mobility model provides realistic movement on highways, which includes the Intelligent Driver Model (IDM) [28] and multi-lane scenarios. With this highway mobility model, one can generate some realistic traffic data, which is taken as input in the proposed method. The simulation was carried out in a 5.0 km × 20.0 m region and Table IV shows the related parameters.

TABLE IV
VALUES OF PARAMETERS

Parameter	*Value*
Simulation time	300 seconds
RSU coverage	300 m
4G's available bandwidth	6 Mbps
VANET available bandwidth	6Mbps

Assuming the OBU in each vehicle has a cellular network interface and an IEEE 802.11p network interface, let two vehicles s and d be communicating with each other, for which s is the source, and d is the destination. In the process of transmission, packet loss occurs when one or more data packets are delivered across relays. Source s is always able to transmit packets to destination d through the BSs in our highway scenario. Note that the number of vehicles is derived based on the flow f_w, which denotes how many vehicles/sec are generated [26] in our highway scenario. In the evaluation, the performance metrics are the *offloading fraction*, *average throughput, average lifetime and delivered data volume* of the V2V path:

- *Offloading Fraction (%)*: It denotes the percentage of the V2V VANET offloading of all vehicles.
- *Average Throughput (Mbps)*: It denotes the average throughput of all vehicles. The average throughput is calculated based on the average data flow volume that each destination d (receiver) has received.
- *Average Lifetime (secs)*: It denotes the average time length of all V2V paths.
- *Delivered Data Volume (MB)*: It denotes the data volume transmitted using VANET network.

A. THE MOBILITY MODEL

The performance analysis is based on the highway scenario that has different numbers of vehicles at different times. When the traffic is congested on the highway, the speed of each vehicle also decreases. Let V_n denote the total number of vehicles on the highway. Each vehicle moves forward on the highway. At the start of the simulation, the speed of each vehicle is varied from 60 km/h to 120 km/h. Obviously, network quality and available bandwidth will become less when vehicle density increases.

In the simulation time, each vehicle starts from a fixed position and then moves from left to right. To simulate a realistic highway scenario, the speed and acceleration should be varied with time. Overtaking and lane changing are also supported.

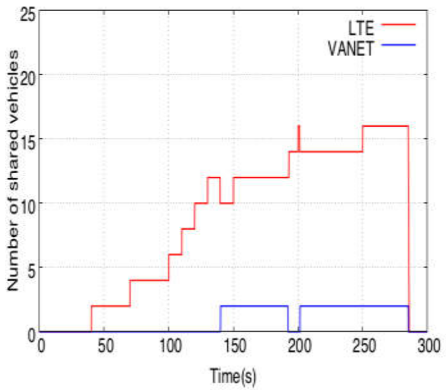
B. RESULTS

In this paper, LT-NSR with the path recovery function (LT-PR) is compared with the basic greedy routing (GD-NSR), which takes the minimum hop count to construct the V2V path for transmitting packets. The function of LT-PR is to try to repair/recover the broken V2V path such that the corresponding peered vehicles can still communicate with each other using the VANET network. The LT-PR function can repair a broken V2V path and make it possible for more mobile data to be offloaded continuously. In this part, each link is associated with a constant transmission rate of 2 Mbps.

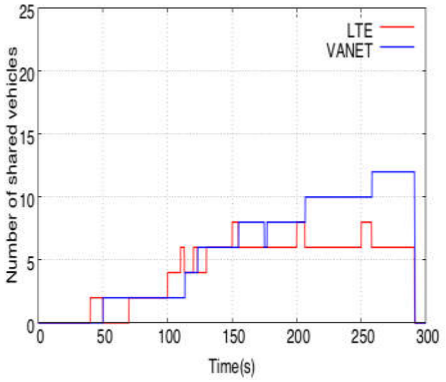
Figures 10-12 depict the relationships between (1) the numbers of paired vehicles that are communicating with each other and (2) the average throughput in the cellular network and VANET respectively, for which situations of different vehicle densities using the proposed LT-NSR

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2820679, IEEE Access

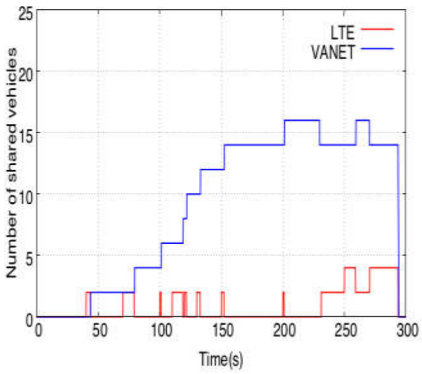
scheme with the path recovery function (LT-PR) are depicted.



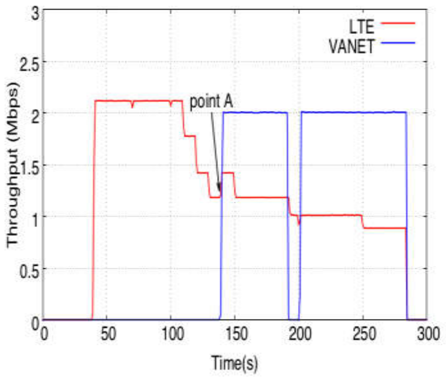
(a)



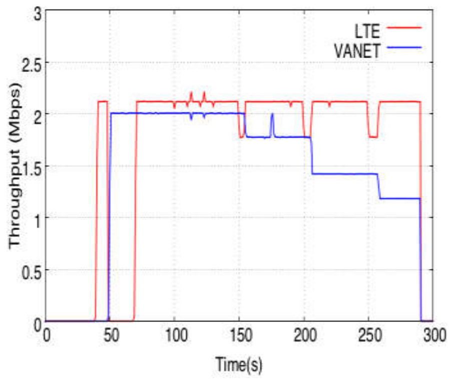
(a)



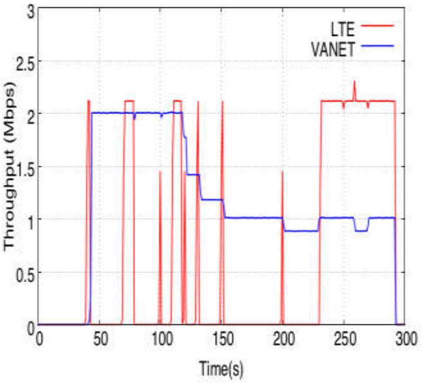
(a)



(b)



(b)



(b)

Figure 10. The average throughput for the situation of low vehicle's density using the proposed LT-NSR scheme with LT-PR. (a) Number of vehicles (f_w=0.07, V_n=20) (b) Average throughput (f_w=0.07, V_n=20)

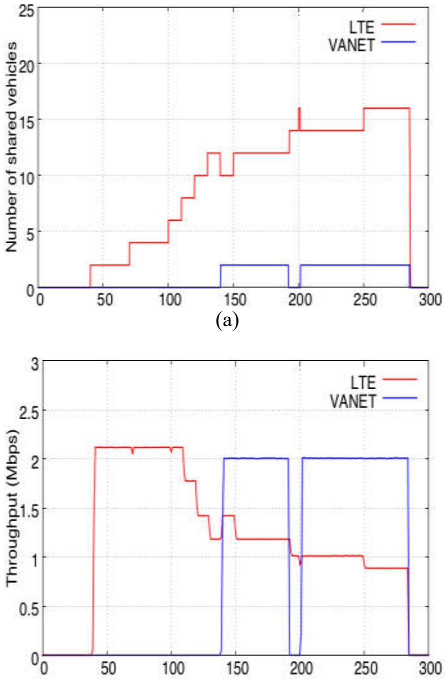
Figure 11. The average throughput for the situation of middle vehicle's density using the proposed LT-NSR scheme with LT-PR. (a) Number of vehicles (f_w=0.12, V_n=35) (b) Average throughput (f_w=0.12, v_n=35)

Figure 12. The average throughput for the situation of high vehicle's density using the proposed LT-NSR scheme with LT-PR. (a) Number of vehicles (f_w=0.17, V_n=50) (b) Average throughput (f_w=0.17, V_n=50)

In the test, links with the cellular network were established when the vehicles were generated and drove on the highway. The average throughput decreases when the number of paired vehicles that are communicating with each other increases. For the low density situation, Figure 10-(b) depicts that the average throughput in the cellular network is reduced as time goes by because the number of paired vehicles that are communicating with each other increases as time goes by, which is shown in Figure 10-(a). The V2V paths cannot be established easily because there are not so many vehicles that can be used to construct V2V paths in the situation of low density. An illustrated example is as follows. At around the 130th second, a new cellular networking link is started for a new pair of vehicles that are communicating with each other. After a period of time, a V2V path for offloading can be established, which is derived by the SDN controller of the SDN_i-MEC server, at point A (the 139th second). Thus, the data transmission of this cellular networking link can be offloaded to the V2V path and the average throughput in cellular network increases immediately. When this V2V path ends at the 192th second, the corresponding paired vehicles that are communicating with each other switch back to the cellular network and then the average throughput in the cellular network decreases. After a while the other V2V path starts at the 201th second.

Figure 11 depicts the situation of middle density. Referring to Figure 11-(a) and Figure 11-(b), the average throughput for the cellular networking link that each vehicle can get is much higher than the situation of low density. The reason is as follows. The number of paired vehicles that are communicating with each other in the situation of middle density is higher than in the situation of low density. Thus, compared with the situation of low density, it can find more V2V paths for more paired vehicles. As a result, the average throughput in the cellular networking link becomes higher than that in the situation of low density because more paired vehicles are offloaded to V2V paths. Additionally, the time length of using the V2V offloading in the situation of middle density is much longer than that of the situation of low density because more vehicles are available to become relays and do packet forwarding for the paired vehicles that want to offload to the V2V paths. A V2V path is broken at the 175th second in the situation of middle vehicle density. The proposed LT-PR repairs the broken path rapidly at the 175th second such that the V2V path can be kept. That is, LT-NSR with LT-PR can make the network state more stable.

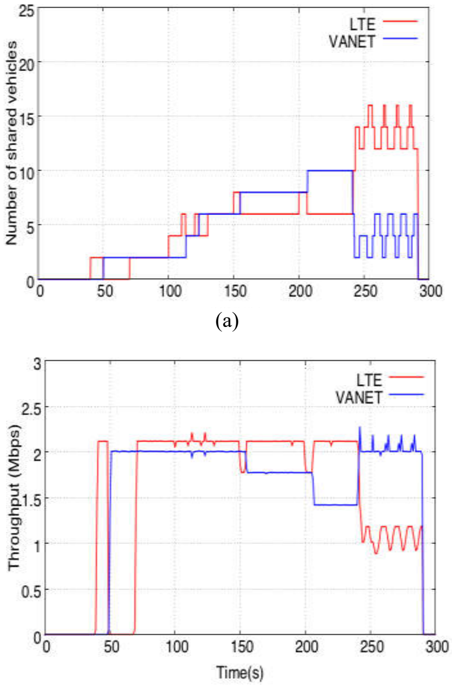
growing number of vehicles keep their V2V routing and share the VANET available bandwidth. This also leads to the average throughput of VANET decreasing as time goes by. A V2V path is broken at the 180th second and another one is broken at the 230th second. LT-PR successfully repairs the V2V path and continues the communication using the VANET network after the 180th second but the other one can't repair the broken path and the communication path is switched from the VANET network back to the cellular network. On the other hand, when the V2V path is broken at the 230th second, the left vehicle that results in the broken V2V path is the only one that can communicate with both its previous vehicle and its next vehicle. That is, without the left vehicle, LT-PR can't find a supplement vehicle that can repair the broken path.



(a)

(b)

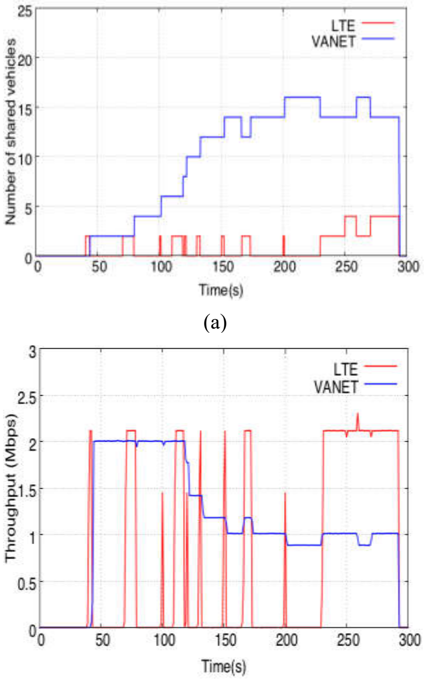
Fig. 13. The average throughput for the situation of low vehicle's density using the proposed GD-NSR scheme. (a) Number of vehicles (f_w=0.07, V_n=20) (b) Average throughput (f_w=0.07, V_n=20)



(a)

(b)

Figure 14. The average throughput for the situation of middle vehicle's density using the proposed GD-NSR scheme. (a) Number of vehicles (f_w=0.12, V_n=35) (b) Average throughput (f_w=0.12, v_n=35)



(a)

(b)

Figure 15. The average throughput for the situation of high vehicle's density using the proposed GD-NSR scheme. (a) Number of vehicles (f_w=0.17, V_n=50) (b) Average throughput (f_w=0.17, V_n=50)

Figure 12-(a) and Figure 12-(b) depict the average throughput of vehicles in the situation of high density. In this situation, many vehicles can provide relays and do forwarding. The higher probability of finding the V2V path (1) results in fewer vehicles staying in the cellular network because it is much easier to find V2V paths for offloading and (2) brings the low average throughput of the VANET network because more paired vehicles are offloaded to V2V paths. Obviously, the VANET network is congested when most of the vehicles switch from the cellular network to the VANET network. Due to the long lifetime of the V2V paths, which results from the higher vehicle density, a

Figures 13-15 depict the results for situations of different vehicle densities using the GD-NSR scheme. Comparing Figure 13-(b) with Figure 10-(b), there is nothing different between the LT-NSR scheme with the recovery function (LT-PR) and the GD-NSR scheme in the situation of low density. The reason is that there are just a few V2V paths that can be established. Thus, the hop counts and routing paths of existing V2V paths are almost the same such that there is no difference between the GD-NSR scheme and the LT-NSR scheme with the recovery function (LT-PR).

In the situation of middle density, many V2V paths that have short lifetimes exist more often such would switch between the cellular network and the VANET network more often. This makes the average throughput jump up and down. Figure 14-(a) depicts the number of vehicles that start the bouncing after around the 240th second. The bouncing situation means that many vehicles left the cellular network and switched to the V2V paths for offloading and then switched back to the cellular network again very soon. Therefore, the mobile data cannot be offloaded smoothly and causes some delays. Comparing Figure 11-(b) with Figure 14-(b), the LT-NSR scheme with the recovery function (LT-PR) outperforms the GD-NSR scheme for the average throughput. In Figure 14-(b), the average throughput in the cellular network declines rapidly and the average throughput in the VANET rises after the 242th second. The reason is that much bouncing occurs, which is depicted in Figure 14-(a). Even if a V2V path is established using the GD-NSR scheme, the lifetime of the V2V path is very short and the mobile data can't be offloaded to the VANET network smoothly.

In the situation of high density, there are many vehicles that can provide relays and do forwarding. In Figure 15-(a), the number of vehicles using VANET increases because of the high probability of finding a V2V path. This results in the average VANET throughput decreasing as time goes by, which is depicted in Figure 15-(b). Comparing Figure 12-(b) with Figure 15-(b), the situations of using LT-NSR with LT-PR and GD-NSR are similar. The reason is that it is easy to find V2V paths because there are many. Even though the hop counts of the V2V VANET offloading are different between the GD-NSR scheme and the LT-NSR scheme with the recovery function (LT-PR), the lifetime of the V2V paths for both are still long enough to allow mobile data to be offloaded.

This part shows the evaluation of the efficiency of transmission based on the aforementioned traffic patterns. After the V2V paths are established gradually, the maximum delivered data volume rises. Figure 16 depicts the maximum delivered data volume in the VANET network for different vehicle densities.

In the situation of low density, 30.8 MB is offloaded using the GD-NSR scheme and the LT-NSR scheme with the recovery function (LT-PR). The delivered data volumes of all peered vehicles that are communicating with each other using these three schemes are the same because there are few V2V paths that can exist in the VANET network for all of these three schemes. The reason is that each peered vehicle that switches to the V2V path can transmit their data with the maximum transmission rate, i.e., 2 Mbps. Since the offloading situations for the GD-NSR scheme and the LT-NSR scheme with LT-PR are the same, the delivered data in VANET are the same for both schemes.

In the situation of middle density, 128.5 MB and 154.5 MB are offloaded using the GD-NSR scheme and the LT-NSR scheme with the recovery function (LT-PR), respectively. Due to the aforementioned bouncing period of transmission in the GD-NSR scheme, which is depicted in Figure 14, the delivered data volume using the GD-NSR scheme is lower than that of using the LT-NSR scheme with the recovery function (LT-PR). The reason is that the LT-NSR scheme with the recovery function (LT-PR) repairs the broken V2V path rapidly to continuously keep offloading in the VANET network; thus, it can offload longer and have a higher delivered data volume.

In the situation of high vehicle density, 171.8 MB and 173.2 MB are offloaded using the GD-NSR scheme and the LT-NSR scheme with the recovery function (LT-PR), respectively. Due to the full utilization of the VANET network, the delivered data volumes are very close for both schemes. Additionally, since almost all vehicles use the VANET network instead of the cellular network in the situation of high vehicle density because many V2V paths can be established more easily, it allows more data to be offloaded.

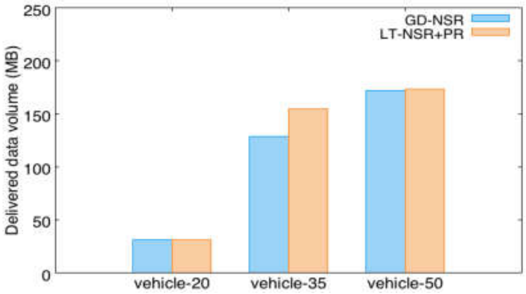


FIGURE 16. Delivered data volume in VANET for situations of different vehicle densities.

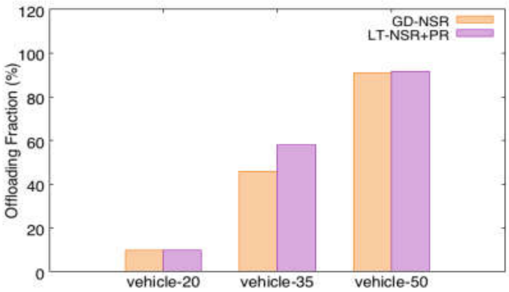


Figure 17. Offloading fractions for situations of different vehicle density.

C. ANALYSIS OF INTEGRATED INFORMATION

Figure 17 depicts the offloading fraction for situations of different vehicle density. In the situation of low density, only 9.7% of the mobile data can be offloaded for both schemes. In the situation of middle density, the offloading fractions are 45.8% and 58% for GD-NSR and LT-NSR with LT-PR, respectively. The proposed LT-NSR scheme with the recovery function (LT-PR) outperforms the GD-NSR scheme because there are more vehicles that stayed in the VANET network. In the situation of high vehicle density, the offloading fraction is up to 91.6% using the LT-NSR scheme with the recovery function (LT-PR); and the offloading fraction is 91% using the GD-NSR scheme. The reason is that there are plenty of vehicles such that V2V paths can be found easily for both schemes.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Figure 18 shows the average lifetime of the V2V paths using the GD-NSR scheme and the LT-NSR scheme with the recovery function (LT-PR). In the situation of low density, there is no difference for both schemes. The reason is that even though some V2V paths can be established in the situation of low density, the hop counts and the lifetime of these few V2V paths are almost the same for both schemes. In the situation of middle density and high density, the proposed LT-NSR scheme with the recovery function (LT-PR) tries to find a V2V path that has the maximum lifetime instead of having the minimum hop count. It is the reason that the average lifetimes using the LT-NSR scheme with the recovery function (LT-PR) are higher than that of using the GD-NSR scheme.

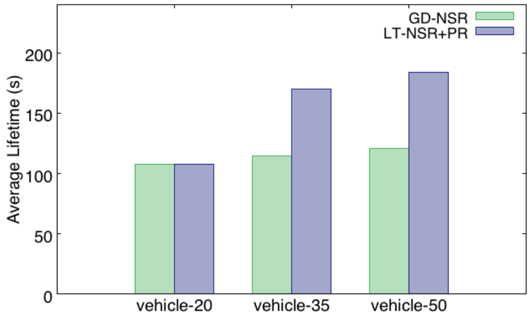


Figure 18. The average lifetime of V2V paths for situations of different vehicle density.

The average lifetime of V2V paths can become much higher when the vehicle density increases. For the GD-NSR scheme, the average lifetime of V2V paths is 108, 115 and 121 seconds in the situations of low, middle and high vehicle density, respectively. For the LT-NSR scheme with the recovery function (LT-PR), the average lifetime of V2V paths is 108, 170 and 184 seconds in the situations of low, middle and high vehicle density, respectively. The reason is that the higher density makes vehicles slow down and thus the communication topology is more stable making the average lifetime increase.

Referring to Figure. 17 and 18, the offloading fractions of these two schemes are almost the same in the situation of high density. The reason is as follows. Although the average lifetime of each V2V path using the LT-NSR scheme with the recovery function (LT-PR) is longer than that of using the GD-NSR scheme, the V2V paths can be found more easily using the GD-NSR scheme. Thus, the number of V2V paths that can be found using the GD-NSR scheme is more than that of using the LT-NSR scheme with the recovery function (LT-PR) in the situation of high density. It makes the offloading fraction of using the GD-NSR scheme keep up with that of using the LT-NSR scheme with the recovery function (LT-PR).

VI. CONCLUSION

When two vehicles are communicating with each other using cellular network, the data transmission can be offloaded to the V2V path that exists between these two vehicles. These two vehicles switch back to the cellular network to communicate with each other when the corresponding V2V path is broken. This paper has proposed the SDN_i-MEC architecture such that the calculation of the V2V path for data offloading can be derived using the centralized way in the network edge, instead of having it in a distributed way among vehicles. The LifeTime-based Network State Routing (LT-NSR) algorithm has been proposed and executed in of the SDN_i-MEC server to find the V2V routing path that has the longest life time based on the current network topology. Using the centralized model of the SDN_i-MEC architecture, the network topology and V2V paths can be derived and established from the periodically reported context, which was sent from vehicles and stored in the context database of the SDN_i-MEC server, using LT-NSR that is executed in SDN Controller of the SDN_i-MEC server. The LifeTime-based Path Recovery (LT-PR) algorithm has also been proposed to recover/repair a broken V2V path such that the corresponding V2V path can be kept. Comparing with the traditional GD-NSR scheme, which takes the minimum hop count to construct the V2V path for transmitting packets, the proposed method considers the maximum lifetime to construct the V2V path, which makes more mobile data be offloaded. The results of the performance evaluation shown that the proposed LT-NSR & LT-PR schemes outperform the GD-NSR scheme in the middle vehicle-density situation. For low and high vehicle-density's situations, LT-NSR & LT-PR and GD-NSR have similar performance. For the future work, it can be twofold: the first one is to consider the quality of service (QoS) issue for the offloading judgment to balance the networking situation of both cellular network and VANET network; the second one is how to have the reliable transmission, e.g., the split TCP concept adopted in the ad hoc network environment, over the MEC-based VANET network.

REFERENCES

[1] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of network and computer applications,* vol. 37, pp. 380-392, 2014.

[2] H. Zhou, S. Xu, D. Ren, C. Huang, and H. Zhang, "Analysis of event-driven warning message propagation in Vehicular Ad Hoc Networks," *Ad Hoc Networks,* vol. 55, pp. 87-96, 2017.

[3] R. Maallawi, N. Agoulmine, B. Radier, and T. B. Meriem, "A comprehensive survey on offload techniques and management in wireless access and core networks," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 3, pp. 1582-1604, 2015.

[4] Y. He, M. Chen, B. Ge, and M. Guizani, "On WiFi offloading in heterogeneous networks: Various incentives and trade-off strategies," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 4, pp. 2345-2385, 2016.

[5] A. Bazzi, B. M. Masini, A. Zanella, and G. Pasolini, "IEEE 802.11 p for cellular offloading in vehicular sensor networks," *Computer Communications,* vol. 60, pp. 97-108, 2015.

[6] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, "Opportunistic WiFi offloading in vehicular environment: A queueing analysis," *in Proceeding of IEEE Global Communications Conference (GLOBECOM),* pp. 211-216, 2014.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2820679, IEEE Access

[7] P. Kolios, C. Panayiotou, and G. Ellinas, "ExTraCT: expediting offloading transfers through intervehicle communication transmissions," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, no. 3, pp. 1238-1248, 2015.

[8] ETSI. (2015). Mobile-Edge Computing (MEC); Service Scenarios. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf

[9] ETSI. (2017). Mobile Edge Computing (MEC); General principles for Mobile Edge Service APIs [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/009/01.01.01_60/gs_MEC009v010101p.pdf

[10] ETSI. (2017). Mobile Edge Computing(MEC); Mobile Edge Platform Application Enablement. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/011/01.01.01_60/gs_MEC011v010101p.pdf

[11] 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, DOI: 10.1109/COMST.2017.2717482,* 2017.

[12] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," *IEEE Access,* vol. 5, pp. 6757-6779, 2017.

[13] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials, DOI: 10.1109/COMST.2017.2682318,* 2017.

[14] 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.

[15] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing," *IEEE Communications Magazine,* vol. 55, no. 7, pp. 94-100, 2017.

[16] 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.

[17] 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.

[18] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *in proceeding of IEEE,* vol. 103, no. 1, pp. 14-76, 2015.

[19] G. el mouna Zhioua, H. Labiod, N. Tabbane, and S. Tabbane, "VANET inherent capacity for offloading wireless cellular infrastructure: An analytical study," *in Proceeding of the 6th IEEE International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1-5, 2014.

[20] G. el Mouna Zhioua, J. Zhang, H. Labiod, N. Tabbane, and S. Tabbane, "VOPP: a VANET offloading potential prediction model," *in Proceeding of IEEE Wireless Communications and Networking Conference (WCNC),* pp. 2408-2413, 2014.

[21] G. E. M. Zhioua, H. Labiod, N. Tabbane, and S. Tabbane, "Cellular content download through a vehicular network: I2V link estimation," *in Proceeding of the 81th IEEE Vehicular Technology Conference (VTC Spring),* pp. 1-6, 2015.

[22] B. K. Kung and B. Karp, "Greedy perimeter stateless routing for wireless networks," *in Proceeding of ACM Conference on Mobile Computing and Networking, pp. 243-254, 2000.*

[23] Z. S. Houssaini, I. Zaimi, M. Oumsis, and S. E. A. Ouatik, "Improvement of GPSR protocol by using future position estimation of participating nodes in vehicular ad-hoc Networks," *in Proceeding of IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM),* pp. 87-94, 2016.

[24] S. Yi, X. Huang, and C. Wang, "EA-GPSR, a routing protocol for energy harvesting wireless sensor networks," *in Proceeding of IEEE International Conference on Computer Science and Network Technology (ICCSNT),* vol. 1, pp. 1029-1032, 2015.

[25] O. S. Brief, "OpenFlow™-Enabled Mobile and Wireless Networks," *white paper,* 2013.

[26] H. Arbabi and M. C. Weigle, "Highway mobility and vehicular ad-hoc networks in ns-3," *in Proceeding of IEEE Simulation Conference (WSC),* pp. 2991-3003, 2010.

[27] B. Dupont, "Improvements in VANET Simulator in ns-3," Masters project, Department of Computer Science, Old Dominion University, 2011.

[28] Treiber, Hennecke and Helbing, Intelligent driver model, 2000. https://en.wikipedia.org/wiki-/Intelligent_driver_model

CHUNG-MING HUANG (SM'–2007) received the B.S. degree in electrical engineering from National Taiwan University on 1984/6, and the M.S. and Ph.D. degrees in computer and information science from The Ohio State University on 1988/12 and 1991/6 respectively. He is currently a Distinguished Professor in Dept. of Computer Science and Information Engineering (CSIE), National Cheng Kung University (NCKU), Tainan, Taiwan, R.O.C. He (i) was Chair of Dept. of CSIE and Director of Institute of Medical Informatics, NCKU and (ii) served as the Chair of IEEE Vehicular Technology Society Tainan Chapter. He has published more than 300 referred journal and conference papers in wireless and mobile communication protocols, interactive multimedia systems, audio and video streaming and formal modeling of communication protocols. His research interests include wireless and mobile network protocol design and analysis, green computing and communication, media processing and streaming, and innovative network applications and services. He is a senior member of IEEE and ACM.

MENG-SHU CHIANG received the M.S. and Ph.d degree from Institute of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, R.O.C. He is currently an Assistant Professor of Department of Computer Science and Information Engineering, Far East University, Tainan, Taiwan. His research interests include Wireless and Mobile Networks.

DUY-TUAN DAO received the B.Sc. degree in electronics and telecommunications engineering from Danang University of Technology, Vietnam, in 2008. He got the M.S. degree in Computer Science and Information Engineering, National Cheng Kung University, R.O.C, in 2013. He is currently pursuing the Ph.D. degree in Computer Science and Information Engineering, National Cheng Kung University, R.O.C. His research interests include IP mobility and wireless communication and its applications in vehicles.

WEI-LONG SU received the B.S. degree in Computer Science Engineering from National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C. on 2015/06 and Master degree in Computer Science and Information Engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C. on 2017/2. His research interests include VANET network and Software-Defined Networking.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2820679, IEEE Access

SHOUZHI XU is a Professor of System Architecture in the College of Computer and Information Technology at China Three Gorges University, Yichang, Hubei Province, China. He received his Ph.D. degree in Information and Communication Engineering from Huazhong University of Science and Technology in China. He is a Fellow of the China Computer Federation. His current research interests include wireless sensor network, vehicle ad hoc network and network security

HUAN ZHOU received his Ph.D degree in Control Science and Engineering from Zhejiang University, Hangzhou, China, in 2014. Currently, he is a school hired full professor in the College of Computer and Information Technology, China Three Gorges University. He was a visiting scholar at the Temple University from November, 2012 to May, 2013. His research interests include mobile social networks, opportunistic mobile networks, and Vehicular Ad Hoc Networks.