Journal of Ambient Intelligence and Humanized Computing (2020) 11:1265–1280
https://doi.org/10.1007/s12652-019-01319-w

ORIGINAL RESEARCH

SD-IoV: SDN enabled routing for internet of vehicles in road-aware approach

Muhammad Tahir Abbas[1] · **Afaq Muhammad**[2] · **Wang-Cheol Song**[3]

Received: 20 December 2018 / Accepted: 8 May 2019 / Published online: 20 May 2019

Abstract

Proposing an optimal routing protocol for internet of vehicles with reduced overhead has endured to be a challenge owing to the incompetence of the current architecture to manage flexibility and scalability. The proposed architecture, therefore, consolidates an evolving network standard named as software defined networking in internet of vehicles. Which enables it to handle highly dynamic networks in an abstract way by dividing the data plane from the control plane. Firstly, road-aware routing strategy is introduced: a performance-enhanced routing protocol designed specifically for infrastructure-assisted vehicular networks. In which roads are divided into road segments, with road side units for multi-hop communication. A unique property of the proposed protocol is that it explores the cellular network to relay control messages to and from the controller with low latency. The concept of edge controller is introduced as an operational backbone of the vehicle grid in internet of vehicles, to have a real-time vehicle topology. Last but not least, a novel mathematical model is estimated which assists primary controller in a way to find not only a shortest but a durable path. The results illustrate the significant performance of the proposed protocol in terms of availability with limited routing overhead. In addition, we also found that edge controller contributes mainly to minimizes the path failure in the network.

Keywords Software defined networking (SDN) · Internet of vehicles (IoV) · Road-aware approach · Edge controller (EC)

1 Introduction

Over the past two decades, with the increased number of new technologies, we have seen extensive modernization in smart devices, we use to access network services and applications. However, the fundamental network that relates such devices has remained unchanged since its formulation. The truth is that with passage of time, the requirement of people and devices using the network are stretching its limits. Network function virtualization (NFV) and software defined networking are all complementary approaches while offering a unique way to design and manage the networks. SDN technology offers a platform for testing and implementing new innovative ideas while exploring its programmability and centralized control mechanism. It separates the data plane from the control plane for the sake of providing a centralized view of the distributed network.

Internet of vehicles is another technology growing rapidly and great endeavors have been made by the government agencies, industries and researchers towards an efficient vehicular communication which would considerably contribute in the development and deployment of intelligent transportation system (ITS). The exclusive characteristics of IoV include high computation ability, connectivity with the high-speed internet, predictable mobility, and variable network density (Saleet et al. 2011; Abbasi et al. 2014; Salkuyeh and Abolhassani 2016; Yaqoob et al. 2019), which is not available in MANETs where we have limited battery power, random motion and computation. IoV is different from the Vehicular Ad-hoc networks (VANETs) by having a centralized management which makes it more suitable for

✉ Muhammad Tahir Abbas
tahir.abbas@kau.se

Afaq Muhammad
afaq.csit@suit.edu.pk

Wang-Cheol Song
philo@jejunu.ac.kr

[1] Department of Computer Science, Karlstad University, Karlstad, Sweden

[2] Department of Computer Science and IT, Sarhad University of Science and Information Technology, Peshawar, Pakistan

[3] Department of Computer Engineering, Jeju National University, Jeju City, South Korea

ITS safety application. On the other hand, vehicular networks only enable vehicles on the roads to turn into access points while providing connectivity to the other vehicles. 20.8 million vehicles are going to be expected till 2030 only in USA. Exploring VANETs for providing road safety applications and traffic management at such a large scale is a hard task. To bridge this gap, VANETs requires a programable architecture to fulfil modern transport services. IoV, is the evolution form of VANETs and MANETs, its is more powerful, yet more challenging to implement.Keeping all these dynamic aspects of vehicles on the road (Kerner 2004); Garavello and Piccoli 2006; Daganzo and Daganzo (1997), designing an efficient routing protocol for data transmission in IoV is a challenging task. This is because that an optimal routing protocol has to consider the heterogeneous node density and communication technologies, intermittent connectivity, and varying mobility Daganzo and Daganzo (1997).

The current architecture of vehicular networks does not fulfill the basic requirements for the advance transportation system and its applications such as flexibility and scalability of routing protocols. A new technology named as software defined networking (Hakiri and Berthou 2015; Kreutz et al. 2015; Diro et al. 2018; Jain and Paul 2013; Jararweh et al. 2015) modernize the IoV architecture for an efficient and optimized routing methodology. With the increase in number of vehicles and road accidents, one cannot manage a huge traffic of big cities in a distributed manner.With the advance in communication technologies, SDN enables the IoV to be managed in a logically centralized fashion through heterogeneous networks (cellular network, RSUs, etc.) These days, SDN has been considered mainly for the fixed network management, especially in access networks and data centres. However, it can also boost the smart city traffic communication, if applied to IoV. Employment of SDN in vehicular networks has been proposed in recent years only. Particularly, preparatory research has been made, mainly at high theoretical and architectural level, to present its potential for efficient utilization of network resources (VANETs) (Salahuddin et al. 2015; He et al. 2016; Zheng et al. 2016). The practical implementations are still missing that significantly assess to which extent SDN can assist vehicular networks (IoV). Specifically the type of wireless technologies used to provide the connectivity between vehicles and SDN controller since a vehicle requires a high level connectivity due to its dynamic topology. Our proposed architecture for the implementation of SDN with IoV paved a path towards the realization of centralized traffic management system. In addition to this, different wireless technologies i.e. LTE are considered to control forwarding plane to cater bandwidth and short-range communication. The reason behind using the cellular network for control messages is to offload the network from massive data traffic while confirming its availability for the traffic with low latency requirements. IoV is emerging out as a promising future, the closed and proprietary way of managing network devices these days. But we firmly believe that due to the benefits SDN can bring, it is the right choice to bridge the gap between the road safety applications and IoV. An extended version of SDN into IoV is shown in Fig. 1.

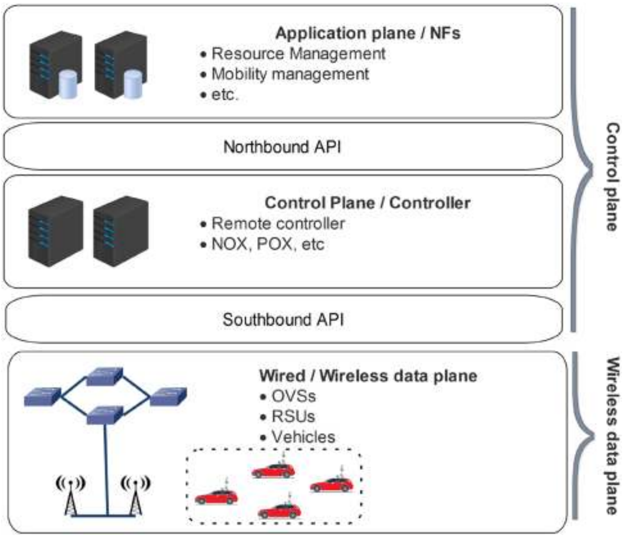

Fig. 1 SDN with IoV

In IoV, routing protocols plays a key role in the finding of best available paths in highly unstable vehicular environment (Cascone et al. 2010; Cutolo et al. 2012; Manzo et al. 2012). A suitable protocol for data transmission in wireless networks can have a good data quality with less or no delay. A number of routing protocols have been proposed for vehicular networks so far (Devangavi and Gupta 2017; Lin et al. 2017; Ding et al. 2016). However, each protocol has its own drawback and limitations according to their working environment. Some of the protocols takes the shortest path to forward the data packet, however, selection of the shortest path is not always feasible due to swift vehicle topology changes with short link lifetime. Morover, a number of protocols follows greedy forward approach which may results in dead end (Bazzi et al. 2017; Muralidhar and Geethanjali 2013). Current routing protocols of VANETs can be classified into following different categories based on the type of information needed: Positions based protocols, geographical based protocols, map based protocols, road based protocols and topology based protocols. The example of topology based protocol includes destination sequence distance vector (DSDV), dynamic source routing (DSR) and Ad hoc On-demand distance vector routing (AODV). Position based protocols includes greedy perimeter coordinator routing (GPCR) and intersection-based geographical routing protocol (IGRP). Map based protocols encompasses geographic source routing (GSR) and shortest-path-based traffic-light-aware routing (STAR). Last but not least, road based routing protocols consists of vehicle-assisted data delivery (VADD) (Zhao and Cao 2008) etc.

Irrespective of the previously proposed protocols for vehicular networks, they cannot directly apply on the SDN based IoV and requires special changes in it because of the ad-hoc nature. In Ku et al. (2014), authors proposed an SDN based architecture for vehicular networks in order to provide the innovative services. The proposed architecture captures the requirement and components required for the deployment of SDN in vehicular environment. Reliable routing paths between the vehicles are calculated by the controller after obtaining network topology information from the vehicles on the roads. However, in case of controller failure, local agents inside the vehicles are then switch to GPSR routing mode to find better paths towards the destination but a mobility problem is not considered which effects the SDN based protocol as whole with control overhead. An edge controller based architecture is proposed in this paper to overcome this problem which helps the SDN controller in a way to pre-process the vehicle data. Edge controller receives a vehicle mobility data after certain intervals of time and if forwarded further towards the centraliszed controller if it have valuable information. This reception of real-time vehicle information also reduces the overhead from the SDN controller.

Authors in Zhu et al. (2015) proposed an architecture by extending the SDN into a routing mechanism for the VANETs to get agile message forwarding with minimized routing overhead. Moreover, in order to calculate and maintain the shortest path with low latency between the vehicles, a new routing metric named as minimum optimistic time (MOT) is designed. A distinctive SDN-enabled architecture is proposed in Truong et al. (2015) for the Fog computing with the support of both the serverise such as safety and non-safety. An orchestrator is added into the SDN controller for the creation of SDN-based VANET's Fog framework. Authors in Zhu et al. (2015) Truong et al. (2015) mainly focuses the theoretical and architectural aspects, a detailed routing mechanism is still needed to support their results. Authors in Jararweh et al. (2015) proposed SDN based framework for Internet of Things (IoT) to manage the devices more efficiently. SDN based data forwarding, security, and storage mechanism is proposed: SDN data forwarding, SDNSec, SDNStore. Moreover, in order to solve the problem of low latency and manageability in IoT, authors in Diro et al. (2018) proposed an architecture by the integration of latest technologies: SDN and Fog computing. Being a part of user end data processing, fog computing plays a crucial role in reducing the latency of the critical IoT applications. Furthermore, to maintain QoS requirements for hetrogenous application, a converged SDN framework is proposed for differential flow space allocation.

In Venkatramana et al. (2017), SDN based geographical routing protocol is proposed for the optimized transmission of data packets. In the proposed architecture, SDN has the comprehensive view of the underlying topology, hence able to calculate the optimal paths in its vicinity. Authors claims that the SDN controller obtains shortest path between the vehicles using the spatial data i.e. OSM. A stable path between the source and the destination is estimated using various parameters, such as distance, vehicle density, speed of a vehicle. Although this work enables SDN to calculate a shortest path, it does not consider the implementation of an analytical model for it to provide any relation between the parameters for path calculation. In ? Hybrid road-aware routing protocol (HRAR) is designed specifically for data transmission in VANETs. Roads are divided into road segments based on road intersection in HRAR ?. HRAR introduces the concept of gateway vehicles to reduce the control routing overhead. RREQ is not forwarded to every vehicle, instead it is only send towards the gateway vehicles and gateway vehicles are further responsible to find the path in a multi-hop fashion. Moreover, HRAR targets the VANETs only, which is considered as a distributed management system and does not consider the infrastructure-assisted communication. These above mentioned two protocols are used to compare with proposed protocol and results proves the better packet delivery with reduce end-to-end delay and overhead.

In IoV, choosing shortest path for communication is not always feasible in case of path duration. Paths with more link residual life are preferred over the shorter link life time. A novel approach for path length in MANETs is explained in Namuduri and Pendse (2012). Authors have derived an association for vehicle density for the predictable path length. Even though, the proposed approach discussed so far operates significantly for VANETs and MANETs but we cannot use the same approach unswervingly for IoV. The purpose is that, motion of vehicles in IoV is limited to roads with the support of fixed structure. Hence, it is the inspiration for our research. In our approach, using road-aware routing protocol, we have anticipated the significance for route length among the source vehicle and a destination. Also, an analytical model for path estimation is proposed for the vehicles on the road. There is no analytical model proposed in IoV so far, but the simulations. It is a challenge to predict path duration due to the dynamic movement of vehicles. This analytical model provides a mathematical form of solution for shortest path estimation. Selecting a shortest path is not always feasible, hence, proposed model enable a vehicle to find a more suitable path based on various parameters for efficient communication.

Our proposed protocol is different from the previous protocols in a number of ways. Firstly, road-aware approach makes our protocol very suitable for IoV due to road segmentation with gateway nodes (RSUs and vehicles near

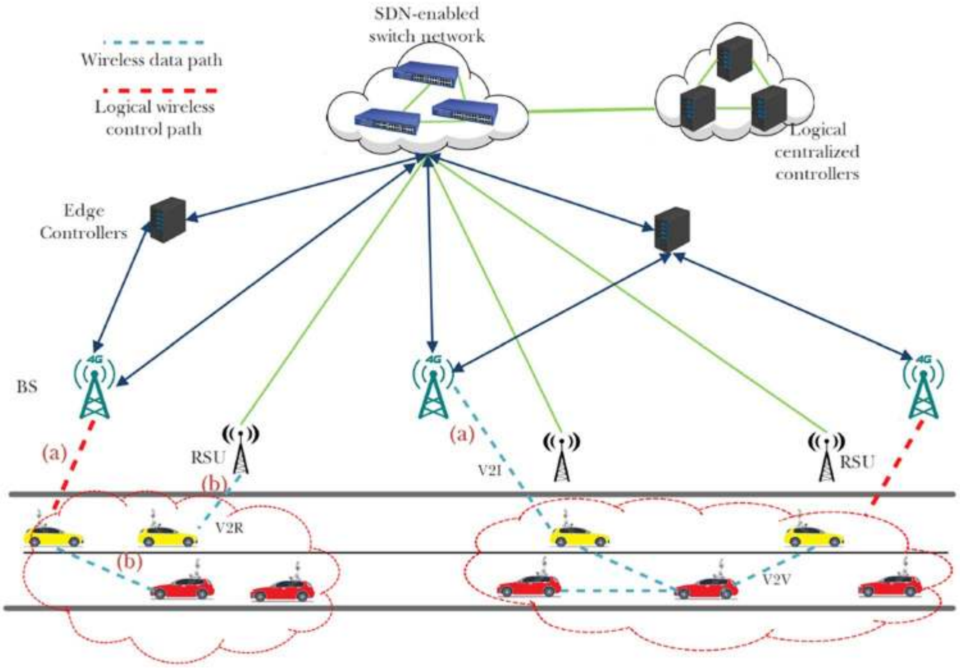
Wireless data path
Logical wireless control path
SDN-enabled switch network
Logical centralized controllers
Edge Controllers
BS
RSU
(a)
(b)
V2R
V2I
RSU
V2V

Fig. 2 Proposed SD-IoV architecture. **a** Control messages from vehicles to EC. **b** Multi-hop communication(vehicle to vehicle, vehicle to road side unit)

intersection) for path creation. This approach of selecting paths using road segment id instead vehicle id makes a path durable. Secondly, different technologies are considered to efficiently forward the data and control packets. Cellular network is used to forward and receive packets to and from SDN controller and vehicles on the road. RSUs are explored to forward the data packet to fixed as well as mobile destination. The reason of using the cellular network for control messages is that these messages requires less bandwidth with low latency for its delivery. Also its long range coverage assists the vehicles on the road to have emergency services with few hops in no time. On the other hand, normal data packet forwarding can be done using the RSUs where the services are limited to entertainment, video streaming and gaming etc. In addition to this, edge controllers are explored to process the real-time data from the vehicles coming after every 100 ms. This approach not only reduces the response time but also a huge packet overhead from the network. Last but not least, SDN controller runs the road-aware protocol with path estimation model for finding the shortest but durable path for communication. A detailed path estimation model is proposed in Sect. 2.3.

Edge controller plays an important role in gathering the realtime information from the vehicles. It is very important to have vehicle information i.e. speed, position and road id without any delay so that the primary SDN controller can process this data by applying the estimated model. However, most of the time the vehicle generated data dose not contains a valuable information hence it just overload the network with this redundant data i.e. in cities vehicles do not change its location much, so there is no need to update the controller after every 0.1 s. In this situation, edge controller came forward to remove this redundant data and forwards only the data which contains some valuable information. It is assume that each edge controller manages a specific area i.e. 6 base stations.

The rest of the paper is organized as follows: Sect. 2 illustrates detailed SD-IoV architecture and working of the protocol. Section 3 outlines the results and Sect. 4 concludes the paper.

2 SD-IoV enable road-aware approach

This section emphasize more on the proposed architecture for the SD-IoV along with path estimation model and road-aware approach. The proposed architecture consists of a software program named as controller which explores the underlying topology information in order to describe the rules to forward the data, and vehicles act as a dumb forwarding devices. Proposed mechanism splits the control traffic from the data traffic by the separation of communication channels, RSUs are utilized for data forwarding and cellular network is used for the transmission of control traffic. In SD-IoV, each vehicle is recognized as an OVS with data flow rules installed in flow tables. A detailed diagram of the propoased architecture is shown in Fig. 2.

Our proposed protocol takes a road-aware approach to forward packets between source and destination by splitting roads into road segments with unique segment ID (Sn). In the first level of road-aware routing, vehicles on the road share their information with the EC. In second level, after getting real-time topology from EC, SDN controller discover and maintains path towards the destination. In addition, proposed protocol exploits the fact that data traffic will be forwarded through RSU or in a multi-hop fashion i.e. vehicle-to-vehicle, and vehicle-to-RSU communication. In general, maintaining updated routes to adjacent RSUs is eventually essential as compared to other mobile nodes because vehicles demand acquaintance to RSUs at an immense rate. In this regard, it is assumed that every vehicle is available with two interfaces,as shown in Fig. 5, WiFi interface for providing the connection with RSUs and other vehicles and cellular network interface to provide the connection with base stations for sending/receiving control messages to and from the controller.

2.1 Functionality of SD-IoV

In this section, a detailed process of the routing mechanism is discussed by which a data packet from the source vehicle is forwarded to the destination using the shortest path calculated by the SDN controller. SD-IoV takes a two level approach for routing strategy. At the first level, a road level topology is maintained by the EC. Vehicles on each road segment share their information with EC that includes $vehicle_id$, $road_id$, *position*, *speed*, and *direction*. RSUs and gateway nodes, on each road segment, takes the responsibility of providing a connectivity between the roads. Gateway nodes are the vehicles near the road intersections.

In the second level, SDN controller maintains a table called $RAR_topology_table$. This table is updated periodically after an interval, with the vehicle and road information, after receiving it by EC. Using this table, SDN contrller have a complete topology of the network. Shortest path between source and destination for each road segment is calculated by the SDN controller, using Algorithm 2 and the flow rules are installed to respective segments only for end-to-end connection. SDN attains all the paths for a road segment based on minimum hop count, direction, and relative velocity and stored them in a table with shortest one at the top. The shortest path will be only selected as an optimal path if it comprises of road segments with with 25–80% value of vehicle density (Abbas and Song 2017).

Algorithm 1 : Process of packet forwarding from source to destination and set of rules for it by the controller in SD-IoV

Notations:
ip: Internet Protocol for vehicle identity
ip_curr: current vehicle IP address
ip_src: source vehicle IP address
nxt_rsu: corresponding RSU of a road segment
nxt_vehicle: neighbor vehicle
$curr\_R_{seg}$: current road segment
$dest\_R_{seg}$: road segment with destination
any_R_{seg}: Segment which is not a $curr\_R_{seg}$ neither $dest\_R_{seg}$
GV: gateway vehicle
RSU: road side unit
Input: *packet_in*
Output: *nxt_rsu*, *nxt_vehicle*, *enodeb*
Upon receiving the packet at vehicle(*packet_in*)

```
if (ip_curr==ip_src) then
    curr_vehicle = src_vehicle
    pkt_receiving_queue(packet_in)
else if (ip_curr ≠ ip _src)&(ip_dst exists_in_any_R_seg) then
    if (curr_R_seg==dest_R_seg) then
        find _nxt _vehicle()
        send _pkt _to _nxt _vehicle(packet_out)
    else
        if (selected path contains GV) then
            find _nxt _vehicle _to _GV()
            send _pkt _to _nxt _GV(packet_out)
        else
            find _nxt _vehicle _to _RSU()
            send _pkt _to _RSU(packet_out)
        end if
    end if
else
    find _enodeb _to _controller()
    send _pkt _to _controller(packet_out)
end if
```

Every time, a vehicle gets a data from the incoming port for the destination, it will look for the destination IP address inside its flow table. Upon finding the destination entry inside the flow table, it forwards the data at the egress port to its neighbor in the direction towards the destination, as shown in Algorithm 1. In case of the destination available on a different road segment, packet is forwarded either towards the gateway vehicle or RSU.

On the other hand, group tables are available to perform further actions. The group tables, inside the OVS, incorporates a number of action buckets which specifies the list of actions to be performed on the packet. For Example, list of actions in bucket 1 can start a $packet_in$ event and is then send towards the controller to look for the forwarding port. Data packet itself is not forwarded to the controller but size of packet, source and destination IP, ingress port and the buffer id where the packet is stored inside the OVS. SDN controller reply with the $packet_out$ message by initializing the path estimation strategy to find out the best available path towards the destination, as shown in Algorithm 1. In the begning, from the available paths, the shortest path with various road segments between the source and destination are selected if consisting of 25–80% vehicles. Further, various paramaters, i.e. hop count, speed, direction, are considererd to calculate final path with more life time. The importance of finding two vehicles with leats speed difference is that they have more connction time. Vehicles then forwards the data packet to the specified port by the controller and also it updates the flow table to add the new flow entries. In the group table, another example of action bucket can be a scenario where the connection towards the SDN controller fails. In that case after waiting for a while, EC takes a hybrid road-aware routing(HRAR) approach to forward the data packet ?.

Vehicles are considered as an OVSs, so a hard timeout is set in its database for every rule made by the controller. After a timeout or if the vehicle move out of its range, that specific entry is removed. A source vehicle continues to unicast the data towards the destination until the path expires. If the path expires before the data completely transferred, SDN controller is notified with the path failure and a new path is recomputed if no other link is available to continue the data forwarding. A vehicle, before sending the packet towards the destination, investigate the flow table for a valid flow entry and if it does, data packet will be forwarded accordingly to the specified rule. However, in case of $no_matching_flow$() a request will be forwarded to the EC and later to SDN controller. Based on the information from the vehicles between the road segments, controller will update the data plane with shortest path towards the destination. RSUs and vehicles along the path will only receive the updated flow rule, no other vehicles will get this update. Due to change in topology, when a neighbor vehicle went down, the source vehicle update the SDN controller with the failure message to recompute the flow entries. After receiving the failure notification, SDN controller repeat the process of shortest path calculation and update the vehicle about the newly computed path.

Whenever a vehicle leaves a road segment without having any process of data forwarding, flow entry of that vehicle will be removed after waiting for the $soft_timeout$. On the other hand, if the value of the $hard_timeout$ is more than the $soft_timeout$, flow entry will remain there until the value $hard_timeout$ declines to zero. However, if a vehicle leaves the road segment vicinity, and still there a data transmission going on, then an updated path is selected form the topology table by the controller for further data transmission.

Algorithm 2 : Packet handling for path calculation at SDN controller

```
Input: $open\_flow\_packet\_in$
Output: $open\_flow\_rule\_installation$
Upon receiving the packet at SDN controller($OF\_packet\_in$)

find dst in database($RAR\_topology\_table$)
get available paths($RAR\_topology\_table$)
compute _traffic _density()
select path with traffic density 30-80%
for (i=0 to lenght($available\_path\_list$)) do
    for (i=0 to lenght($next\_hop\_list$)) do
        select vehicle at the edge of range()
        function $Calculate\_link\_lifetime$
            $hop\_count()$
            $relative\_velocity()$
            $sort\_vehicle\_with\_min\_speed\_diff()$
            $select\_vehicle\_with\_min\_speed\_diff()$
            $link\_connectivity()$
            $total\_path\_duration()$
        end function
    end for
end for
select path with long duration()
return $OF\_write\_flow(ingress\_port, egress\_port)$
```

2.2 Path failure notification

In SD-IoV, a vehicle on the road forwards a path failure notification towards the EC if a path expires, either due to topology change or the removal of path flow entry. SDN controller calculate and maintain various shortest paths at EC for each road segment under its vicinity. Each time a failure notification is received by EC, it first checks the type of failure. EC analyze its table for a shortest path if the failed notification received from inside of the road segment, of its vicinity. On the other hand, a route request is always forwarded to the SDN controller in case of path failure outside of the road segment, as shown in Algorithm 3. It is worth mentioning that, EC can have failure notifications form a number of vehicles. After receiving first notifications, remaining with similar path ID will be discarded by EC.

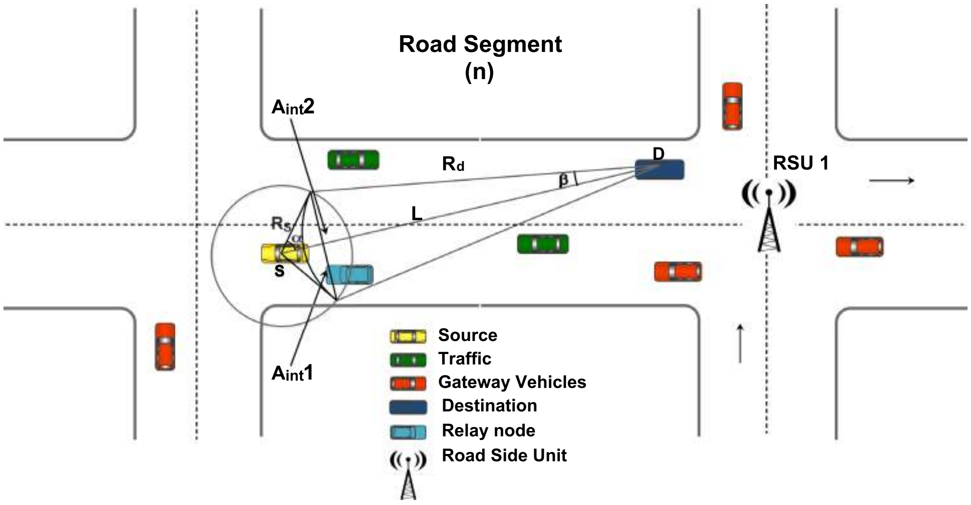

Fig. 3 Network model for selecting relay node for path estimation

Algorithm 3 : Path failure notification handling at EC and SDN controller

Input: *path_failure_notification*, *route_request*
Output: *open_flow_rule_installation*
Notations:
rd_seg: road segment
failure_notif: route failure notification
Upon receiving the packet at EC()

```
if (notification_type==path_failure) then
    function path_failed_notification()
        pkt _from _road _segment()
        if (failure_notif -¿ inside_rd_seg) then
            find new path in EC(road_segment_table)
            select path with long lifetime()
            generate flow rule()
            remove old flow entry()
            write new flow entry()
            discard request from same source & ID()
        else if (failure_notif -¿ outside_rd_seg) then
            send path request to SDN()
            find new path in SDN(RAR_topology_table)
            calculate link lifetime() // apply algorithm 2
            select path with long lifetime()
            generate flow rule()
            remove old flow entry()
            write new flow entry()
        end if
    end function
end if
return OF_write_flow(ingress_port, egress_port)
```

2.3 Path duration estimation

A vehicular network, at a particular interval of time, can be viewed as a static network, however, based on a mobility model, change in the topology can be predicted upto certain time. Since a number of communication links between the source and destination could be possible and the estimation of all the paths is not always reasonable. Given that the conduct of "on demand" routing protocols is strictly related with the shortest route, the exploration of average path interval based on shortest path principle is suitable and significant.

In this section, we have introduced a novel probabilistical model for the estimation of path duration for our SDN based road-aware routing protocol. A remarkable property of the proposed protocol is to find not only the potential paths but also the durable and more stable based on various parameters, which incorporate the average number of hops, link connectivity, direction and velocity.

To provide reliable links, SDN controller determines link duration for every path incorporating discrete parameters. Each vehicle perceives its neighbor's position and velocity from the beacons characterized earlier. This information is further used to predict a time span for which a two neighbor vehicles remain in the communication range of each other (Fig. 3).

2.3.1 Mathematical model

Aim of this portion is to reckon an expression for path duration between two vehicles by deriving mathematical relation such as link duration and average number of hops. We have used adopted a traditional traffic flow principle in our estimation model to represent an efficient vehicular environment for data forwarding. In our proposed model, vehicles are considered to follow Poisson distributed arrivals for obtaining the probability distribution function (*pdf*).

Variables	Description
S	Source node
D	Destination node
L	Distance between source and destination
R_S	Source vehicle range
R_D	Distance from destination to R_S
A_{int1}	Area of intersection 1
A_{int2}	Area of intersection 2
A_{Total}	Total area for expected neighbor node
A_S	Area of sub-segment of road
D_L	Source to relay node distance
R_V	Relative velocity
V_S	Source node velocity

Variables	Description
V_{NH}	Velocity of relay node
N_H	Expected number of hops
$f_{RV}(RV)$	PDF of relative velocity
λ	Constant integer
α	Angle between two lines (source to destination)

2.3.2 Area for next hop

To find the stable path between source and destination, we need a communication link with minimum number of hops towards the destination. Since the node which is closer to the border line, towards the destination covers maximum distance, reduce the number of hops between source and destination. This is the reason that we have chosen the area for our next hop at the extreme end of the transmission range. Area that needs to be calculated is also known as the area of intersection of the circles with the radius of R_S and R_d respectively. Note that the area of circular segment of is equal to the area of circular sector minus the are of triangular portion. To find the area of the region we have the following standarad formula:

$$A = \left[\frac{(\theta - sin(\theta)) \cdot R^2}{2}\right] \tag{1}$$

Total area of both the segments can be calculated using the above formula:

$$A_{Total} = A_{Int1} + A_{Int2} \tag{2}$$

However

$$A_{Int1} \simeq \left[\frac{(\alpha - sin(\alpha)) \cdot R_s^2}{2}\right] \tag{3}$$

And

$$A_{Int2} \simeq \left[\frac{(\beta - sin(\beta)) \cdot R_s^2}{2}\right] \tag{4}$$

Now, we have the entire region for expected relay node

$$A_{Total} = \left[\frac{(\alpha - sin(\alpha)) \cdot R_s^2}{2}\right] + \left[\frac{(\beta - sin(\beta)) \cdot R_s^2}{2}\right] \tag{5}$$

Therefore, we can say that the A_{int1} and A_{int2} represents that region of the circle in which the source vehicle looks for the neighbor vehicle.

2.3.3 Node relative velocity

Direction and speed of a vehicle plays a significant role for the estimation of path duration, it is because direction of a vehicle directly effects the link duration. At this step, we are enthusiastic more in relation and derivation for the relative velocity with its various cases. A city scenario is considered for our model with the moving vehicles in both the direction. Lets assume a scenario that we have two moving vehicles with velocities v_1 and v_2, respectively, and the distance between them is d while the range for radio communication of a vehicle is expressed as r. In order to determine different velocities, four general cases for the velocities of these moving vehicles are considered:

Case 1: when both the vehicles have same direction with same velocity then communication link is available for longtime T_1 between them. Relative velocity between the vehicles, with velocity v_1 and v_2 respectively, can be calculated using the following cosine law as:

$$\left|\vec{v_r}\right| = \sqrt{v_1^2 + v_2^2 - 2v_1v_2Cos\theta} \tag{6}$$

When both the vehicles have same directions but different velocities, the vehicle with greater velocity can be represented as: v_1 which is λ times greater then v_2. Whereas the value of λ varies from 1 to 4.

As we have considered same velocity for both the vehicles with same direction therefore,

$V_1 = V_2 = \mathrm{V}$

And Angle: $\theta = 0$

Then: $\left|\vec{V_r}\right| = 0$

Case 2: when both the vehicles have same directions but different velocities, V_1 is α times greater then V_2.

$\alpha V_1 = V_2$

And Angle: $\theta = 0$

Then: $\left|\vec{V_r}\right| = V_1(\alpha - 1)$

Case 3: when both the vehicles having the same velocity with opposite direction.

$V_1 = V_2 = \mathrm{V}$

And Angle: $\theta = \pi$

Then: $\left|\vec{V_r}\right| = 2V$

Case 4: when both the vehicles having different velocity and both are opposite direction.

$\alpha V_1 = V_2$

And Angle: $\theta = \pi$

Then: $\left|\vec{V_r}\right| = V_1(\alpha + 1)$

2.3.4 Probability density function of relative velocity

From previous results, it is observed that v_r has different values so it can be represented as a random variable and according to probability density function (*pdf*), we can find it's expected relative velocity function as:

$$E(v_r) = \int_{-\infty}^{\infty} v_r f v_r dv_r \tag{7}$$

For further simplification to our scenario, above equation can be written as:

$$E(v_r) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} \int_{0}^{\pi} fv_1 \cdot fv_2 \cdot f(\theta 1, \theta 2) * \sqrt{v_1^2 + v_2^2 - 2v_1 \cdot v_2 \cdot Cos\theta} \quad dv_1 \quad dv_2 \quad d(\theta 1, \theta 2) \tag{8}$$

Equation 8 represents the *pdf* for a relative velocity. To be more specific, *pdf* for each case can be derived as:

General case

$$E(v_r) = \int_{v_{min}}^{v_{max}} \int_{v_{min}}^{v_{max}} (\lambda \pm 1) v_1 f v_1 f v_2 dv_1 dv_2 \tag{9}$$

Above formula can be used with minus sign when we have two vehicles moving in the same direction. On the other hand, the same formula can be used with positive sign if both the vehicles are moving in the opposite direction with velocities v_1 and v_2, respectively.

2.3.5 Average number of neighbor nodes

Average number of neighbor nodes can be defined as the number of vehicles between the source and destination. It is necessary to recognize total distance between source and destination in order to calculate average number of hops. Poisson distribution model is followed by vehicles on the road available within source transmission range. In addition, the probability of finding destination node is the same as the probability of finding $next - hop$ node, if destination node is present in the senders transmission range. The distance to first $next - hop$ can be calculated as:

$$N_H = \frac{L}{D_L} \tag{10}$$

In Eq. 10 , D_L is the distance between source and relay node.

2.3.6 Link connectivity

In this section, we will calculate the time for link duration of every vehicle for the sake of finding a route with maximum duration. Now, the equation for time and speed will be, Time = Distance/Speed.

$$T_L = \frac{R_S - D_L}{V_S - V_{NH}} \tag{11}$$

Whereas, D_L is total span between the next hop source node, accessible within the scope of source node R_S. Moreover, T_L exhibits the link connectivity that holds the value of link residual life. Distance D_L between next hop and source node can be determined by using the following formula.

$$D_L = \frac{n \cdot R_S}{n + 1} \tag{12}$$

And the remaining link life is,

$$T_L = \frac{D_L}{R_V} \tag{13}$$

where, D_R is the distance required by the next hop to move out of the transmission range of source node and is calculated as $D_R = R_S - D_L$. Now the *pdf* of T_L can be represented as:

$$F_T(T_L) = \int_{0}^{V} R_V f_{dR_V}(T_L R_V, V) dV \tag{14}$$

2.4 Path time estimation

Complete path estimation in VANETs is one of the fundamental design parameter. Remaining link life is considered in order to determine the *pdf* of path duration. If $T_L 1, T_L 2, T_L 3, T_L 4$ and $T_{L(N_H)}$ are the remaining link time between the hops 1,2,3,4 and N_H , *pdf* for a path duration is calculated as

$$T_{LPath} = MIN(T_L 1, T_L 2, T_L 3, T_L 4 T_{LN_H}) \tag{15}$$

Also, the *pdf* of T_L can be determined using Baye's theorem (De et al. 2006) and chapter 6 in Papoulis and Unnikrishna Pillai (2002),

$$F(T_L) = N_H . D_L . C_{T_L}^{N_H - 1} \tag{16}$$

Here, $C_{(T)} = 1 - F_T$ illustrates the complementary cumulative distribution function (*CDF*) of T_{LPath} and F_T. Henceforth, average path duration can be known using the following equation:

$$T_{LPath}(average) = \int_{0}^{\alpha} T_L f(T_L) dT_L \tag{17}$$

2.5 Working of edge controller

This part of the paper focuses on the operation of the proposed Edge Controllers (ECs) for the mobility problem. In vehicular ad-hoc networks, link breakage due to change in topology is critical issue which effects the overall

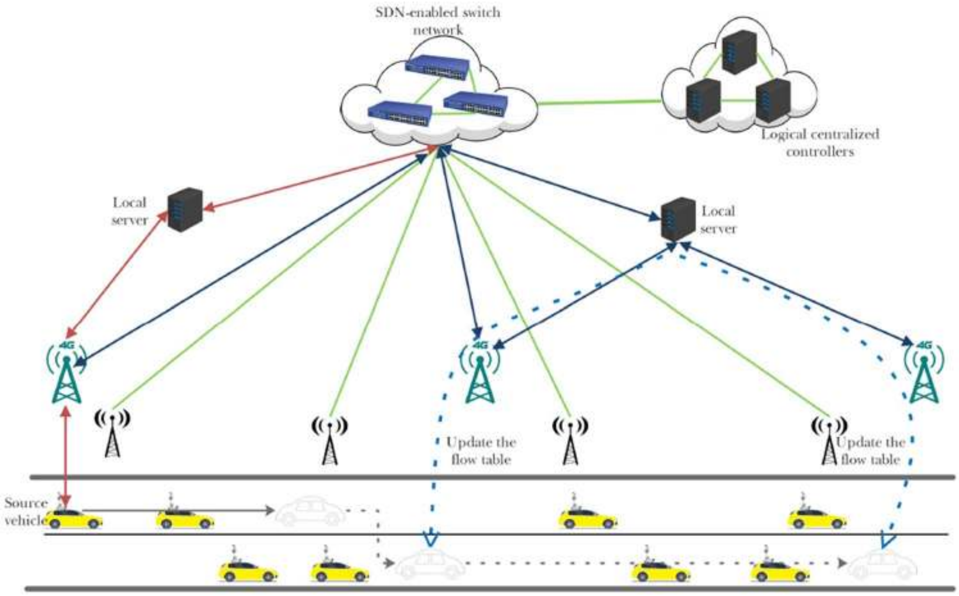

Fig. 4 SDN, EC flow rule installation

performance of the network badly. In this section, our aim is to overcome the mobility problem. In this research, the concept of EC is used to accumulate the information from every vehicle in order to know its topology in real-time, as shown in Fig. 4. Vehicles on the road periodically send their information which includes the *speed*, *direction*, $road_{I}D$, and *position* etc. towards EC. Once this information reaches the EC, it checks the vehicle information in its database. If there is legitimate change of vehicle position, EC updates the central SDN controller with the vehicle's latest information, otherwise EC update its database with new vehicle position only. This approach of updating SDN controller allows the network to handle less traffic overhead. Moreover, this information is provided to EC at the regular intervals, which is 0.1 seconds which is calculated after running few experiments, however this time cannot be fixed and can be changed according to the vehicle and road conditions. If the vehicle is moving with high speed, information will be send more frequently. The mathematical formula for the calculation of hello packet interval can be derived as:

$$hellointerval(t) = \frac{S}{V} \tag{18}$$

$$hellointerval(t) = MIN\left(c, \frac{S}{V}\right) \tag{19}$$

In Eq. 19, 'S' represents the distance covered, V represents the vehicle speed (m/s) and c is a fixed interval e.g. 0.1 second. Sending the hello packets towards the controller have two different scenarios. (a) Hello packets are forwarded after a fixed time interval, for example hello packet is sent after fixed interval e.g. 0.1 sec (b) w.r.t. position that if position is changed by 10m then hello packet shall be sent. Eq. 18 represents a formula for the calculation of hello interval. Where the interval is estimated based on vehicle velocity and distance covered in a certain period of time. On the other hand, Eq. 19 represents the type of interval to be selected by a vehicle. A vehicle first calculates the interval based on the Eq. 18 and then it takes the minimum of both, fixed interval which is 0.1 second and is represented by 'c' or the interval determined by vehicle speed and position. In the above case, if the vehicle speed is 105 m/sec than hello packet interval will be 0.095 sec. In the meanwhile, if the vehicle is moving with speed below 100m/s, interval is set to the default value which is 0.1 sec. Last but not least, providing real-time information of vehicles on the road makes the SDN controller to perform quick action by installing the rules on time.

3 Evaluation and results

3.1 Simulation setup

For the evaluation of the proposed SD-IoV based road-aware approach, we have used the SUMO simulator. The road information, which includes traffic lights, speed limits, road directions and junctions, with an area of 3000 m × 3000 m is downscaled from the open street map (OSM) the road topology as shown in Fig. 6. The *.osm* file with all the road information is then processed using the SUMO simulator to get trace files. Which is then injected to MININET-WiFi (Lantz and O'Connor 2015) to perform further simulations. The system architecture used in our research is presented in Fig. 5.

Log distance propagation model with the exponent value of 4.1 is used for the simulations. Minimum vehicle speed

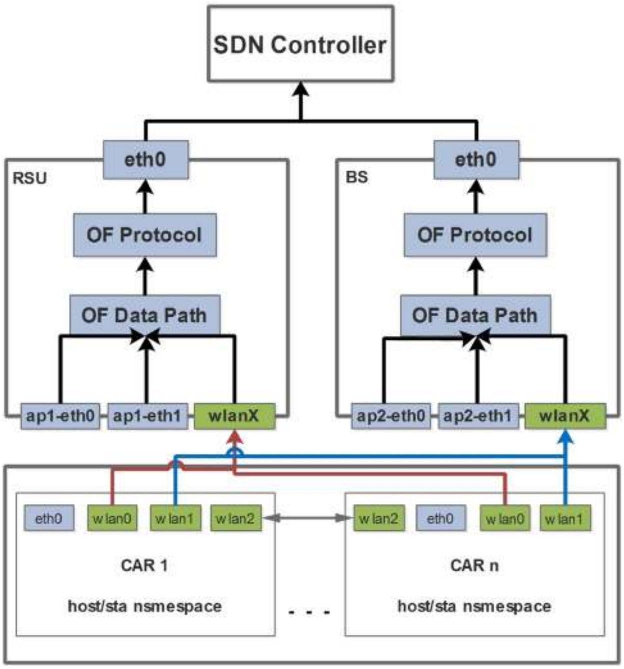

Fig. 5 An overview of system model, car architecture, and OpenFlow programmability in Mininet-Wifi

Table 1 Simulation Setup

Parameters	Value
Number of vehicles	20–100
Simulation area	3000 × 3000 m
Time for simulation	1200 s
Vehicles speed (min)	0–4 m/s
Vehicles speed (max)	5–25 m/s
Simulation scenario	roads with various intersections
Propagation model	Logdistance
Value of exponent	4.1
Traffic type	User datagram protocol
Data packet size	512 bytes

varies from 0–4 m/s, however, the maximum speed varies from 5–25 m/s. The number of RSUs used for the simulation varies from 3 to 7, however fixed number of eNodeB are used (4 eNodeB).For more details, see Table 1.

3.2 Compared protocols

In order to evaluate the working of the proposed protocol, we have simulated the hybrid road-aware routing protocol with no infrastructure assistance with SD-IoV to prove the working of SDN with SD-IoV. Hybrid road-aware routing protocol (HRAR) is designed specifically for data transmission in VANETs. In HRAR, roads are divided into road segments based on road intersection. HRAR introduces the concept of gateway vehicles to reduce the control routing overhead. RREQ is not forwarded to every vehicle, instead it is only send towards the gateway vehicles and gateway vehicles are further responsible to find the path in a multi-hop fashion. In addition, to prove the efficiency of the proposed solution in terms of control overhead, SDN-enabled routing protocol for VANETs (Venkatramana et al. 2017) is also compared in Sect. 3.4.

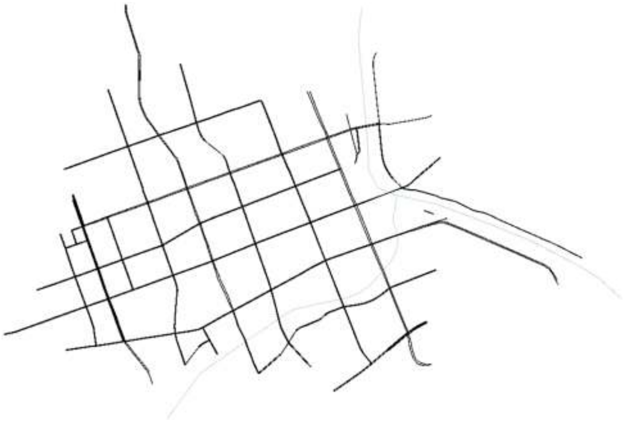

Fig. 6 Simulated road topology of Gangnam street

3.3 Metrics

The following metrics are considered in order to evaluate the performance of the proposed protocol for SDN-based IoV.

3.3.1 Routing overhead

In Fig. 7, routing overhead is determined for all the protocols. And it is observed that the total routing overhead is escalated with average vehicle speed. Figure 7 shows that routing overhead for HRAR and SD-IoV increases because of the fact that the redundancy will generate more traffic in highly dense road segments. Routing overhead for SD-IoV is less when compared with the more RSUs. This is because of maintaining the road segment level routing table by EC and finding routes outside the road segments only when needed and it is purely determined by the SDN controller and there is no need to send the RREQ to every road segment.

3.3.2 Packet delivery ratio

In this portion, we study the effect of varying average vehicle speed on the performance of our proposed protocol in relation of packet delivery ratio. Figure 8 shows the outcomes of the delivery ratio for various RSUs with discrete node speed. Figure 8 illustrates that SD-IoV outperforms

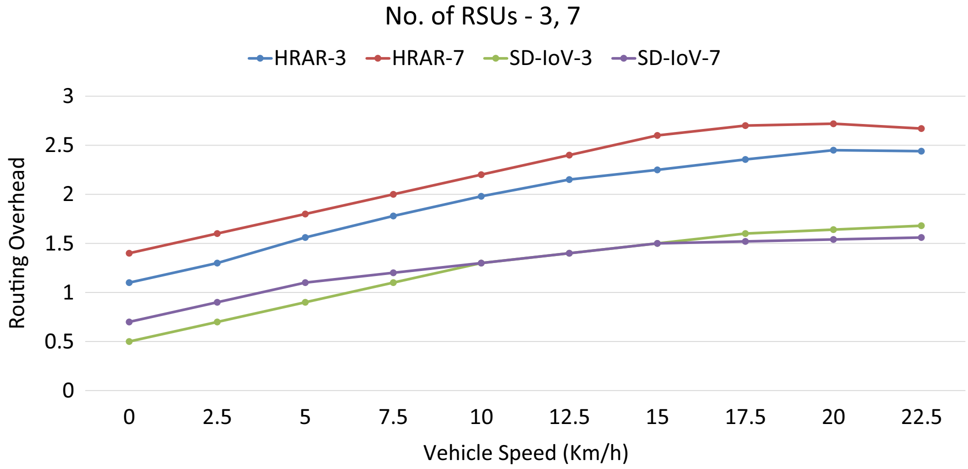

Fig. 7 HRAR and SD-IoV-routing overhead vs. vehicle speed (km/h)

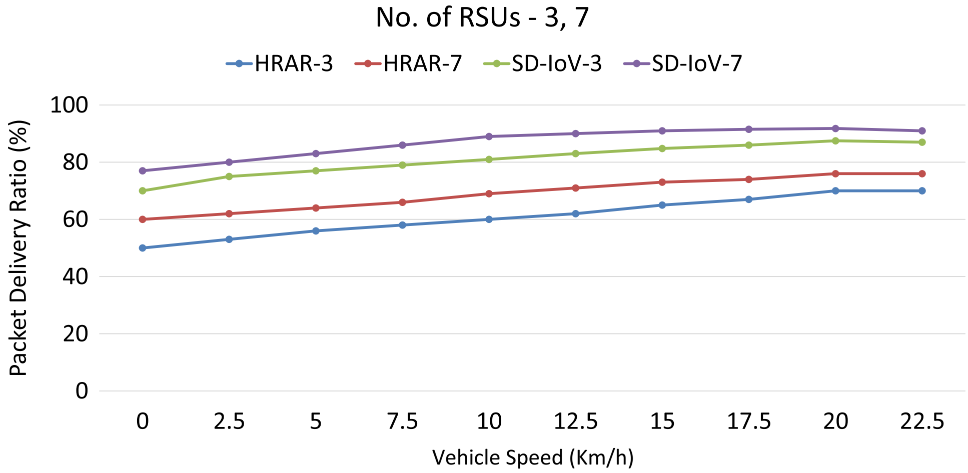

Fig. 8 HRAR and SD-IoV packet delivery ratio vs vehicle speed (Km/h)

when compared to HRAR at low as well as at high vehicle speed. It outlines that the packet delivery ratio increases with an increase in the number of RSUs along the roads but in our case the performance increases with the existing RSUs when compared to others. On the other hand, Fig. 8 shows that the packet delivery for SD-IoV increases more with the increase in RSUs. The reason is that due the escalation of speed, source vehicle will find the neighbors rapidly, and hence, have more chances to deliver the data packet to intermediate vehicles with the high probability of successful packet delivery.

3.3.3 End to end delay

End to end delay depends on number of hops and congestion on the network. SD-IoV show better performance because paths are calculated and maintained re-actively for each segment to each RSU by the controller. Which benefits in the distribution of data packets quickly. On the other hand, with the increase in average vehicle speed, HRAR shows more end-to-end delay when compare to the SD-IoV, as shown in Fig. 9. However, SD-IoV shows very small delay when the average vehicle speed varies from 4 km/h to 25 km/h.

3.4 No. of control messages vs node density and vehicle speed

Control traffic includes the messages transferred between the SDN controller and the open flow switches, which act as a mobile nodes or vehicles with mobility. Figures 11 and 10 represents the control messages spawned by the SDN controller to revise flow tables with new flow rules. Openflow is a channel that connects the SDN controller with RSUs, base stations and vehicles. SDN controller can support multiple channels at a same time and the messages transmitted over these channels have to follow the standards made by OVS protocol. In general, control plane and data plane provide support for various messages but in our simulations we have explored three of them i.e. packet in, packet out and hello messages, as shown in Table 2. Initially,as shown in Figs. 11 and 10, we can perceive that as node density rises, the SDN controller requires to cater additional vehicles and have to

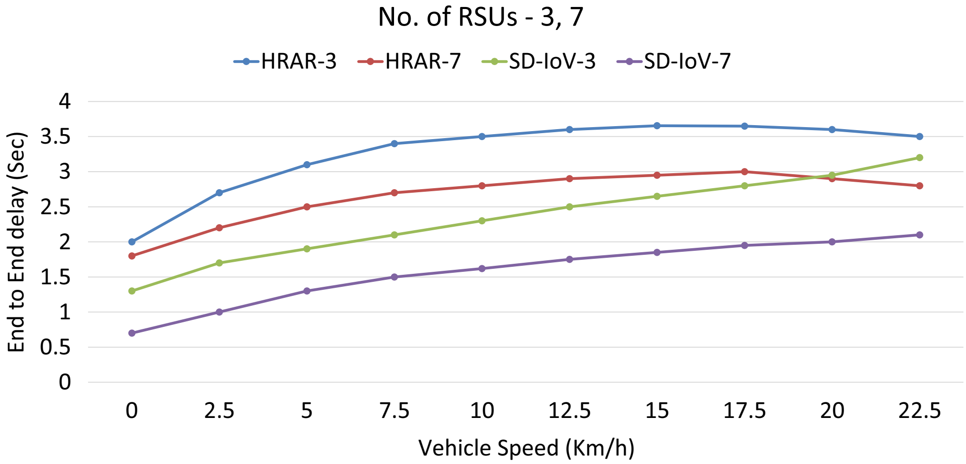

Fig. 9 HRAR and SD-IoV end to end delay vs vehicle speed (Km/h)

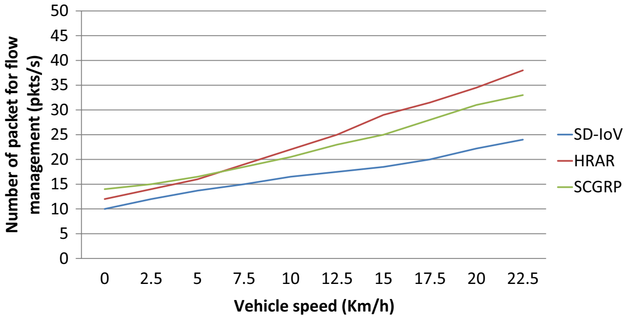

Fig. 10 Control traffic for SD-IoV, HRAR, and SCGRP vs vehicle speed

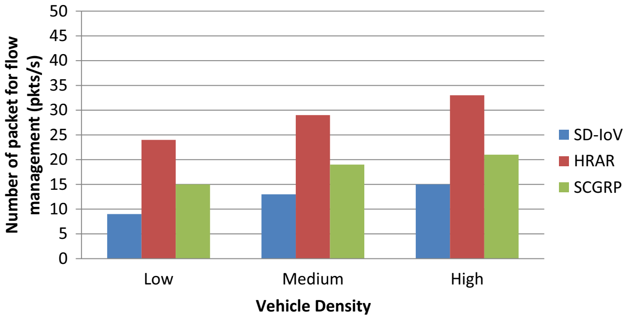

Fig. 11 Control traffic for SD-IoV, HRAR, and SCGRP vs vehicle density

Table 2 Type of control messages used

Message type	Parameters
Hello	OpenFlow header, vehicle location, road id, vehicle id, vehicle speed
Packet in	buffer id, size, reason to invoke, table id, cookie
Packet out	buffer id, no of action, actions, OF header

forward further control messages. However, SD-IoV shows less control messages between the controller and the switch when compared to HRAR and SCGRP (Venkatramana et al. 2017) because SCGRP forwards data traffic based on the shortest path calculated by SDN controller and using the vehicle ID. Source vehicle notifies the controller about the link failure whenever a vehicle moves out of its range. On the other hand, SD-IoV uses road ID instead of vehicle ID to forward the data traffic and in addition SDN controllers provides the best path for each road segments which reduces the chances of link failure. From the Fig. 11 low density represents 0–15 vehicles, medium shows 15–35 and high shows 35–50. Also, we see in Fig. 10 that the control traffic escalates with the speed because the controller have to update himself with rapidly change of the vehicle topology. Generally, as these messages do not transmit bulky payloads, at 30packets/s the overhead traffic flow is controllable.

4 Conclusion

Software defined networking has been progressively extending its footsteps not only in a single terrain, fixed network (e.g. wired networks, data centers) but also in dynamic networks i.e. wireless network, Ad-hoc Networks and internet of vehicles etc. However, applying the wired network protocols inside highly dynamic environment of IoV for data dissemination is not appropriate and requires a number of changes in it. In this paper, a novel road-aware routing protocol for SDN based IoV is proposed to forward data packets in V2V and V2X manners. In the proposed protocol, roads are divided into road segments based on the intersections and are assigned with unique ids. EC maintains road segment level routing within its vicinity. Vehicles share their information (vehicle speed, direction, road id, vehicle id, position) with the EC using the hello beacons. This information is then further sent towards the SDN controller in order to find the best available paths between the source and the destination. SD-IoV explores the RSUs and neighbor vehicles to forward the data packets, however, it utilizes cellular network i.e. 4G/5G to send the control/emergency packets only. This is because of the fact that cellular network costs higher while providing half the bandwidth of RSUs for data dissemination. In order to evaluate our proposed protocol for SDN based IoV, we have used Open Street Map (OSM for getting the road information, SUMO for generating the vehicular traffic on the roads, and MININET-WiFi to test our protocol. Default controller is used in MININET-WiFi to install rules for every vehicle based on the information provided by it. Protocol is rigorously tested using multiple scenarios, at various speeds, dynamic deployment of RSUs and eNodeB, with different length of road segments. Last but not least, the proposed protocol is tested with well-known routing protocol HRAR, and the results show that SD-IoV outperforms in almost all the aspects of routing techniques. Since, this topic is very active among the researcher and industries these days, so various researches can be made on to it. For example, vehicle position and trajectory prediction for better rule installation and heterogeneous communication using SDN technology can be very useful for smart city applications. Other research topic includes early collision warning, traffic flow predictions, and, last but not least, the requirements for 5G deployment can also be recognized.

Acknowledgements This work was supported by the research grant of Jeju National University in 2018.

References

Abbas MT, Song WC (2017) A path analysis of two-level hierarchical road, aware routing in VANETs. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, pp 940–945

Abbasi IA, Nazir B, Abbasi A, Bilal SM, Madani SA (2014) A traffic flow-oriented routing protocol for vanets. EURASIP J Wirel Commun Netw 2014(1):121

Bazzi A, Slock DT, Meilhac L (2017) A Newton-type Forward Backward Greedy method for multi-snapshot compressed sensing, in 2017 51st Asilomar Conference on Signals, Systems, and Computers, pp 1178–1182

Cascone A, Marigo A, Piccoli B, Rarità L (2010) Decentralized optimal routing for packets flow on data networks. Discrete Cont Dyn Syst-Ser B (DCDS-B) 13(1):59–78

Cutolo A, De Nicola C, Manzo R, Rarità L (2012) Optimal paths on urban networks using travelling times prevision. Model Simul Eng 2012:7

Daganzo C, Daganzo CF (1997) Fundamentals of transportation and traffic operations, vol 30. Pergamon, Oxford

De S, Caruso A, Chaira T, Chessa S (2006) Bounds on hop distance in greedy routing approach in wireless ad hoc networks. IJWMC 1(2):131–140

Devangavi AD, Gupta R (2017) Routing protocols in VANET–a survey. In: 2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon), IEEE, pp 163–167

Ding Q, Sun B, Zhang X (2016) A traffic-light-aware routing protocol based on street connectivity for urban vehicular ad hoc networks. IEEE Commun Lett 20(8):1635–1638

Diro AA, Reda HT, Chilamkurti N (2018) Differential flow space allocation scheme in sdn based fog computing for iot applications. Journal of Ambient Intelligence and Humanized Computing, 1–11

Garavello M, Piccoli B (2006) Traffic flow on networks, vol 1. American institute of mathematical sciences, Springfield

Hakiri A, Berthou P, (2015) Leveraging sdn for the 5g networks: trends, prospects and challenges. arXiv preprint arXiv:1506.02876

He Z, Cao J, Liu X (2016) Sdvn: enabling rapid network innovation for heterogeneous vehicular communication. IEEE Netw 30(4):10–15

Jain R, Paul S (2013) Network virtualization and software defined networking for cloud computing: a survey. IEEE Commun Mag 51(11):24–31

Jararweh Y, Al-Ayyoub M, Benkhelifa E, Vouk M, Rindos A et al (2015) Sdiot: a software defined based internet of things framework. J Ambient Intell Hum Comput 6(4):453–461

Kerner B (2004) The physics of traffic. understanding complex systems. Springer Berlin Heidelberg, Berlin, Heidelberg. doi 10, 978–3

Kreutz D, Ramos FM, Verissimo P, Rothenberg CE, Azodolmolky S, Uhlig S (2015) Software-defined networking: a comprehensive survey. Proc IEEE 103(1):14–76

Ku I, Lu Y, Gerla M, Gomes RL, Ongaro F, Cerqueira E, et al (2014) Towards software-defined VANET: Architecture and services., in Med-Hoc-Net, pp 103–110

Lantz B, O'Connor B (2015) A mininet-based virtual testbed for distributed SDN development. In: ACM SIGCOMM Computer Communication Review, vol. 45, ACM, pp 365–366

Lin D, Kang J, Squicciarini A, Wu Y, Gurung S, Tonguz O (2017) Mozo: a moving zone based routing protocol using pure v2v communication in vanets. IEEE Trans Mob Comput 16(5):1357–1370

Manzo R, Piccoli B, Rarità L (2012) Optimal distribution of traffic flows in emergency cases. Eur J Appl Math 23(4):515–535

Muralidhar K, Geethanjali N (2013) A novel dead end and packet loss avoidance scheme for geographic forwarding in manets

Namuduri K, Pendse R (2012) Analytical estimation of path duration in mobile ad hoc networks. IEEE Sens J 12(6):1828–1835

Papoulis A, Unnikrishna Pillai S (2002) Probability, random variables, and stochastic processes. Tata McGraw-Hill Education

Salahuddin MA, Al-Fuqaha A, Guizani M (2015) Software-defined networking for rsu clouds in support of the internet of vehicles. IEEE Inter Things J 2(2):133–144

Saleet H, Langar R, Naik K, Boutaba R, Nayak A, Goel N (2011) Intersection-based geographical routing protocol for vanets: a proposal and analysis. IEEE Trans Veh Technol 60(9):4560–4574

Salkuyeh MA, Abolhassani B (2016) An adaptive multipath geographic routing for video transmission in urban vanets. IEEE Trans Intell Trans Syst 17(10):2822–2831

Truong NB, Lee GM, Ghamri-Doudane Y (2015) Software defined networking-based vehicular adhoc network with fog computing. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, pp 1202–1207

Venkatramana DKN, Srikantaiah SB, Moodabidri J (2017) Scgrp: Sdn-enabled connectivity-aware geographical routing protocol of vanets for urban environment. IET Netw 6(5):102–111

Yaqoob S, Ullah A, Akbar M, Imran M, Shoaib M (2019) Congestion avoidance through fog computing in internet of vehicles. J Ambient Intell Hum Comput 1–15

Zhao J, Cao G (2008) Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. IEEE Trans Veh Technol 57(3):1910–1922

Zheng K, Hou L, Meng H, Zheng Q, Lu N, Lei L (2016) Soft-defined heterogeneous vehicular network: architecture and challenges. IEEE Netw 30(4):72–80

Zhu M, Cao J, Pang D, He Z, Xu M (2015) SDN-based routing for efficient message propagation in VANET, in International Conference on wireless algorithms, systems, and applications, Springer, pp 788–797

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.