Road-Aware SDN Enabled Routing for the Internet of Vehicles (SD-IoV)

Elvis Yeboah-Duako
School of Computer Science and Engineering
University of Electronic Science and Technology of China
202124080120@std.uestc.edu.cn

***Abstract* — Proposing an optimal routing protocol for the internet of vehicles with low overhead has proven difficult due to the current architecture's inability to manage flexibility and scalability. As a result, the proposed architecture consolidates an evolving network standard known as software defined networking in the internet of vehicles. By separating the data plane from the control plane, it is able to handle highly dynamic networks in an abstract manner. The road-aware routing strategy is introduced first, which is a performance-enhanced routing protocol designed specifically for infrastructure-assisted vehicular networks. Roads are divided into road segments, and roadside units are used for multi-hop communication. The proposed protocol is unique in that it uses the cellular network to relay control messages to and from the controller with low latency. To achieve a real-time vehicle topology, the concept of edge controller is introduced as the operational backbone of the vehicle grid in the internet of vehicles. Last but not least, a novel mathematical model is estimated to assist the primary controller in finding not only the shortest but also the most durable path. The results demonstrate the proposed protocol's significant performance in terms of availability with minimal routing overhead. Furthermore, we discovered that the edge controller is primarily responsible for reducing network path failure.**

Keywords — Road-aware approach, Software Defined Networking (SDN), Internet of Vehicles (IoV), Network Function Virtualization (NFV), Edge Controller (EC).

I. INTRODUCTION

With the rapid advancement in technology over the past years, smart devices used to access network resources and applications are being tremendously modernized at an alarming rate. However, the core network that interconnects such devices has seen little to no technological development since its inception. This unfortunate reality stretched the limits of the network as it attempts to meet the demands of people and their smart gadgets as time progresses. In the wake of this, the Software Defined Networking (SDN) and the Network Function virtualization technology offers a way out by providing an all-complementary approach to offering a unique way to the design and management of the network [1]. The SDN technology provides a platform for experimenting with and implementing new inventive ideas while also investigating its programmability and centralized control mechanism that can be leveraged to enhance the performance of the network. To provide a centralized perspective of the distributed network, SDN separates the data plane from the control plane. [2]

The internet of vehicles is another rapidly growing technology that has attracted the attention of government agencies, industries, and researchers globally who have made significant efforts to develop and deploy an efficient vehicular communication system that will significantly contribute to the development and deployment of intelligent transportation systems (ITS). IoV has unique characteristics such as high computation capability, high-speed internet connectivity, predictable mobility, and variable network density [3]–[6] which are not available in MANETs due to limited battery power, random motion, and computation. IoV differs from vehicular ad hoc networks (VANETs) in that it has centralized management, making it better suited for ITS safety applications. Vehicular networks, on the other hand, allow vehicles on the road to act as access points while also offering connectivity to other vehicles. Exploring VANETs for large-scale traffic control and road safety applications is a difficult endeavor and as such, it requires a programable architecture to bridge this gap and provide current transportation services. IoV is a hybrid of VANETs and MANETs that is more powerful but also more difficult to implement. Designing an efficient routing system for data transfer in the Internet of Vehicles while keeping all of these dynamic elements of vehicles on the road in mind is a difficult issue [7]–[9]. This is because an ideal routing system must account for diverse node density and communication technologies, as well as intermittent connectivity and variable mobility.

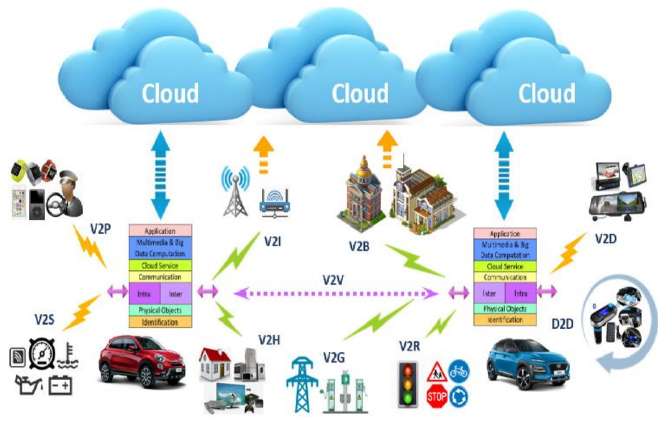

Fig 1: Conventional IoV for Smart Cities

The present vehicular network architecture fails to meet basic needs for advanced transportation systems and applications, such as routing protocol flexibility and scalability. And given the increase in the number of vehicles on the road and the frequency of accidents, large cities' traffic cannot be managed in a distributed manner. By incorporating Software-defined networking into the IoV architecture, IoV can be modernized to allow for more efficient and optimized routing that meets routing flexibility and scalability [10]–[14]. SDN enables the IoV to be handled in a logically centralized manner via heterogeneous networks, thanks to advancements in communication technologies (cellular network, RSUs, etc.) SDN is now mostly used for fixed network management, particularly in access networks and data centers. When applied to IoV, however, it can improve smart city traffic communication. Only recently has the use of SDN in vehicle networks been proposed. Preliminary research has been conducted, mostly at a high theoretical and architectural level, to demonstrate its potential for efficient network resource usage (VANETs) [15]–[17]. Practical implementations that analyze the extent to which SDN can support automotive networks are still lacking (IoV). Because a vehicle requires high-level communication due to its dynamic topology, the sort of wireless technologies utilized to enable connectivity between automobiles and SDN controllers is of particular interest. Our proposed architecture for implementing SDN with IoV paved the way for a centralized traffic management system to be realized. Different wireless technologies, such as LTE, are also considered to regulate the forwarding plane in order to accommodate bandwidth and short-range communication. The use of the cellular network for control messages is intended to relieve the network of heavy data traffic while ensuring its availability for traffic with low latency needs. IoV is gaining traction as a potential future that offers a closed and proprietary method of managing network devices. However, we are convinced that, given the benefits SDN can provide, it is the best option for bridging the gap between road safety applications and IoV. Figure 2 depicts an enhanced version of SDN for IoV.

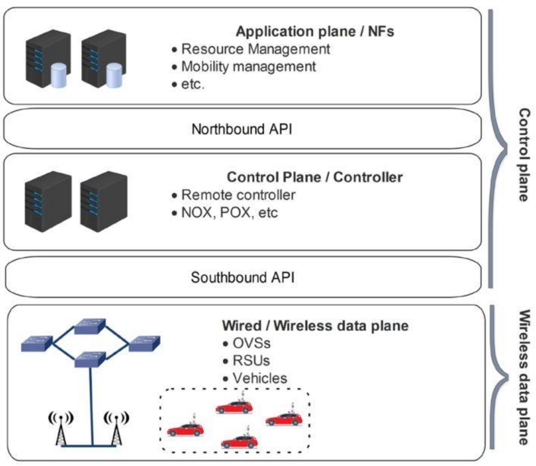

Fig. 2: SDN with IoV

The rest of the paper are sectioned as follows: Section II further discusses related literature. III provides a brief overview of the proposed protocol while IV gives an in-depth discussion of the SDN-IoV architecture and its working principle. Section V focuses on Evaluation and results from simulations and finally, VI concludes the research findings.

II. LITERATURE REVIEW

Routing protocols are important in IoV because they help locate the best available paths in a highly unstable vehicular environment [18]. A strong data quality protocol for data transfer in wireless networks can have little or no delay. [18]–[20] have all suggested routing methods for vehicle networks. However, depending on the working environment, each protocol has its own disadvantages and restrictions. Some protocols use the shortest path to forward data packets; however, due to rapid vehicle topology changes and limited link lifetimes, selecting the shortest path is not always possible. Furthermore, some protocols use a greedy forward approach, which might lead to dead ends [21]–[22]. Current routing protocols of VANETs can be classified into 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. Destination Sequence Distance Vector (DSDV), Dynamic Source Routing (DSR), and Ad hoc On-demand Distance Vector routing (AODV) are examples of topology-based protocols. Greedy perimeter coordinator routing (GPCR) and intersection-based geographical routing protocol are two examples of position-based protocols (IGRP). Geographic Source Routing (GSR) and Shortest-path-based Traffic-light-aware routing (STAR) are examples of map-based protocols. Finally, road-based routing methods include vehicle-assisted data delivery (VADD) among others [23].

Regardless of the previously proposed protocols for automotive networks, they cannot be used directly to the SDN-based IoV due to its ad-hoc nature, which necessitates particular adjustments. To provide innovative services, writers in [24] developed an SDN-based architecture for automotive networks. The suggested architecture captures the requirements and components needed for SDN deployment in a vehicle. After acquiring network topology information from the vehicles on the highways, the controller calculates reliable routing paths between them. Nevertheless, in the event of a controller failure, local agents inside the cars convert to GPSR routing mode to identify better paths to the destination, but a mobility problem is not taken into account, which has an impact on the SDN-based protocol as a whole with control overhead. This work proposes an edge controller-based architecture to solve this challenge, which aids the SDN controller in pre-processing vehicle input. After a specific amount of time, the edge controller gets vehicle mobility data and forwards it to the centralized controller if it contains useful information. The SDN controller's overhead is also reduced by receiving real-time vehicle information.

Authors in [25] presented an architecture for VANETs based on extending the SDN as a routing mechanism to achieve agile message forwarding with minimal routing overhead. Furthermore, a new routing parameter known as minimum optimistic time (MOT) is developed to calculate and maintain the shortest path with low latency between the cars. [26] offer a unique SDN-enabled architecture for fog computing that supports both safety and non-safety services. For the establishment of the SDN-based VANET's Fog framework, an orchestrator is added to the SDN controller. Although [25] and [26] focus mostly on theoretical and architectural elements, a detailed routing mechanism is still required to support their findings. Authors in [14] proposed SDN based framework for Internet of Things (IoT) to manage the devices more efficiently where SDN based data forwarding, security, and storage mechanism are proposed: SDN data forwarding, *SDNSec*, and *SDNStore*. Moreover, in order to solve the problem of low latency and manageability in IoT, authors in [12] proposed an architecture by the integration of latest technologies: SDN and Fog computing. Fog computing, as part of user-end data processing, is critical in reducing the latency of critical IoT applications.

Furthermore, a converged SDN framework for differential flow space allocation is proposed to meet QoS requirements for heterogeneous applications.

In [27], the author proposed an SDN-based geographical routing protocol for optimized data packet transmission. SDN has a comprehensive view of the underlying topology in the proposed architecture, allowing it to calculate the optimal paths in its vicinity. The SDN controller uses spatial data, such as OSM, to determine the shortest path between vehicles. And various parameters, such as distance, vehicle density, and vehicle speed, are used to estimate a stable path between the source and the destination. Although this work enables SDN to calculate the shortest path, it does not take into account the implementation of an analytical model to provide any relationship between the parameters for path calculation. In [28], the Hybrid road-aware routing protocol (HRAR) is specifically designed for data transmission in VANETs. Roads are divided into road segments based on road intersections and to reduce control routing overhead, HRAR introduces the concept of gateway vehicles. RREQ is not forwarded to every vehicle; rather, it is only sent to gateway vehicles, which are then responsible for finding the path in a multi-hop fashion. Furthermore, HRAR focuses solely on VANETs, which are considered a distributed management system, and does not consider infrastructure-assisted communication. These two protocols are used to compare with the proposed protocol, and the results show that the proposed protocol provides better packet delivery with less end-to-end delay and overhead.

In IoV, selecting the shortest path for communication is not always possible due to path duration. Paths with a longer link residual life are preferable to those with a shorter link life time. [29] explains a novel approach to path length in MANETs. The authors discovered a relationship between vehicle density and predictable path length. Even though the proposed approach discussed thus far works well for VANETs and MANETs, we cannot use the same approach indefinitely for IoV. The goal is that vehicle motion in IoV is limited to roads with fixed structure support. As a result, it serves as the inspiration for our research. Using a road-aware routing protocol, we anticipated the importance of route length between a source vehicle and a destination. In addition, an analytical model for path estimation for vehicles on the road is proposed. So far, no analytical model has been proposed in IoV, only simulations. Because of the dynamic movement of vehicles, predicting path duration is difficult. This mathematical solution for shortest path estimation is provided by this analytical model. Choosing the shortest path is not always possible; thus, the proposed model allows a vehicle to find a more suitable path based on various parameters for efficient communication.

III. PROPOSED PROTOCOL

In several ways, our proposed protocol differs from the previous protocols. To begin, our road-aware approach makes our protocol ideal for IoV due to road segmentation with gateway nodes (RSUs and vehicles near intersections) for path creation. This method of selecting paths based on road segment ID rather than vehicle id makes a path more durable. Second, various technologies are considered in order to efficiently forward data and control packets. The cellular network is used to send and receive packets between the SDN controller and the vehicles on the road. RSUs are being investigated for the purpose of forwarding data packets to both fixed and mobile destinations. The reason for using the cellular network for control messages is that they require less bandwidth and have a low latency. Its long-range coverage also allows vehicles on the road to receive emergency services in a matter of minutes with only a few hops. Normal data packet forwarding, on the other hand, can be done using RSUs, where the services are limited to entertainment, video streaming, and gaming, among other things. Furthermore, edge controllers are being investigated in order to process real-time data from vehicles arriving every $100\,ms$. This method not only reduces response time but also significantly reduces network packet overhead. Last but not least, the SDN controller employs a road-aware protocol with a path estimation model to determine the shortest but most durable communication path.

An edge controller (EC) is critical and is used in gathering real-time information from vehicles. It is important to have vehicle information, such as speed, position, and road ID, available without delay so that the primary SDN controller can process this data using the estimated model. However, because most of the time the vehicle generated data does not contain valuable information, it simply overloads the network with this redundant data. For example, in cities, vehicles do not change location frequently, so there is no need to update the controller every 0.1 s. In this case, the edge controller stepped in to remove the redundant data and forward only the data that contains valuable information.

IV. SD-IOV ARCHITECTURE AND WORKING PRINCIPLE

This section focuses on the proposed SD-IoV architecture, as well as the path estimation model and road-aware approach. The proposed architecture consists of a software program called the controller that investigates the underlying topology information to describe the rules for data forwarding, and vehicles that act as dumb forwarding devices. The proposed mechanism separates control traffic from data traffic by separating communication channels; RSUs are used for data forwarding and cellular networks are used for control traffic transmission. Each vehicle is recognized as an OVS in SD-IoV, with data flow rules installed in flow tables. Figure 3 depicts a detailed diagram of the proposed architecture.

Our proposed protocol divides roads into road segments with unique segment IDs to forward packets between source and destination (Sn). Vehicles on the road share information with the EC in the first level of road-aware routing. After receiving real-time topology from EC, the SDN controller discovers and maintains a path to the destination at the second level. Furthermore, the proposed protocol takes advantage of the fact that data traffic will be routed through RSUs or in a multi-hop fashion, i.e., vehicle-to-vehicle and vehicle-to-RSU communication. In general, keeping updated routes to neighboring RSUs is eventually necessary when compared to other mobile nodes because vehicles require frequent communication with RSUs. In this regard, it is assumed that each vehicle has two interfaces: Wi-Fi interface for

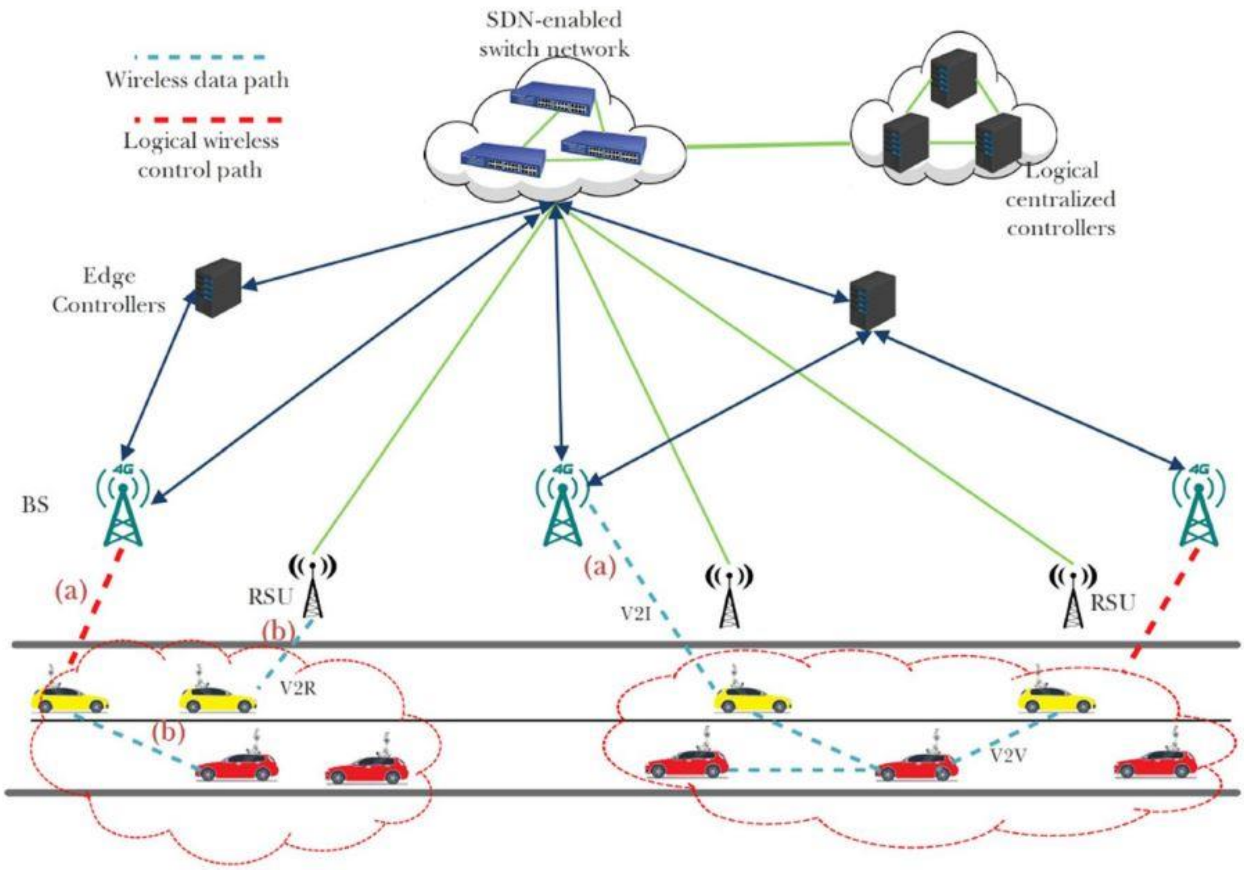

Fig. 3: Proposed SD-IoV architecture. (a) Control messages from vehicles to EC. (b) multi-hop communication (Vehicle to vehicle, vehicle to roadside unit)

communicating with RSUs and other vehicles, and cellular network interface for communicating with base stations for sending and receiving control messages to and from the controller.

A. *Functionality of SD-IoV*

In this section, the routing mechanism's detailed process is discussed, in which a data packet from the source vehicle is forwarded to the destination using the shortest path calculated by the SDN controller is discussed. SD-routing IoV's strategy is divided into two levels. The EC maintains a road level topology at the first level. Vehicles on each road segment communicate with EC about their vehicle id, road id, position, speed, and direction. RSUs and gateway nodes on each road segment are in charge of providing connectivity between the roads. Vehicles near road intersections serve as gateway nodes.

At the second level, the SDN controller keeps a table called the RAR topology table. After receiving vehicle and road information from EC, this table is updated on a regular basis. Using this table, the SDN controller can see the entire network topology. The SDN controller calculates the shortest path between source and destination for each road segment using Algorithm 2, and flow rules are installed to respective segments only for end-to-end connections. SDN finds all the paths for a road segment based on the shortest hop count, direction, and relative velocity and stores them in a table, with the shortest path at the top. The shortest path will be chosen as the optimal path only if it includes road segments with a vehicle density of 25–80 percent [30].

When a vehicle receives data from an incoming port for the destination, it searches its flow table for the destination IP address. When it finds the destination entry in the flow table, it forwards the data to its neighbor in the direction of the destination at the egress port, as shown in Algorithm 1. If the destination is located on a different road segment, the packet is routed either to the gateway vehicle or to the RSU.

Group tables, on the other hand, are available for additional actions. Within the OVS, the group tables include a number of action buckets that specify the list of actions to be performed on the packet. For example, a list of actions in bucket 1 can initiate a *packet in* event, which is then routed to the controller to look for a forwarding port. The data packet is not forwarded to the controller, but the size of the packet, the source and destination IP addresses, the ingress port, and the buffer ID where the packet is stored within the OVS are. As shown in Algorithm 1, the SDN controller responds to the *packet out* message by initiating the path estimation strategy to find the best available path to the destination. Initially, the shortest path with various road segments between the source and destination is chosen from among the available paths if it contains 25–80% vehicles. Furthermore, various parameters such as hop count, speed, and direction are taken into account to calculate a final path with a longer life time. Finding two vehicles with the smallest speed difference is important because they have more connection time. The controller then forwards the data packet to the specified port and updates the flow table to include the new flow entries. Another example of an action bucket in the group table is a scenario in which the connection to the SDN controller fails. When that happens, the EC, after some time, employs a hybrid road-aware routing (HRAR) approach to forward the data packet.

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

Notations:
GV: gateway vehicle
RSU: road side unit
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 neighboring vehicle
curr_R_{seg}: current road segment
dest_R_{seg}: road segment with destination
any_R_{seg}: any other segment aside *curr_R_{seg}* and *dest_R_{seg}*
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_des 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
```

Because vehicles are regarded as OVSs, a hard timeout is set in its database for each rule implemented by the controller. That specific entry is removed after a timeout or if the vehicle moves out of its range. The data is unicast from the source vehicle to the destination until the path expires. If the path expires before the data is completely transferred, the SDN controller is notified, and a new path is computed if no other link is available to continue the data forwarding. Before sending a packet to its destination, a vehicle checks the flow table for a valid flow entry, and if it finds one, the data packet is forwarded according to the specified rule. If no matching flow is found, the request is routed to the EC and then to the SDN controller. The controller will update the data plane with the shortest path to the destination based on information from vehicles between road segments. Only RSUs and vehicles along the path will receive the updated flow rule; no other vehicles will receive it. When a neighbor vehicle fails due to a change in topology, the source vehicle notifies the SDN controller and recomputes the flow entries. After receiving the failure notification, the SDN controller repeats the shortest path calculation process and updates the vehicle with the newly computed path.

When a vehicle leaves a road segment without engaging in any data forwarding process, its flow entry is removed after a *soft timeout*. If the value of the *hard timeout* is greater than the soft timeout, flow entry will remain until the value of the hard timeout falls to zero. However, if a vehicle leaves the vicinity of a road segment while data transmission continues, the controller selects an updated path from the topology table 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 30 – 80 %  traffic density
for (i=0 to length (available_path_list)) do
    for (i=0 to length (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)
```

B. Path Failure Notification

If a path expires in SD-IoV, a vehicle on the road sends a path failure notification to the EC, either due to a topology change or the removal of path flow entry. At EC, the SDN controller calculates and maintains various shortest paths for each road segment in its vicinity. When EC receives a failure notification, it first determines the type of failure. If a failed notification is received from within a road segment's vicinity, EC searches its table for the shortest path. A route request, on the other hand, is always forwarded to the SDN controller in the event of a path failure outside of the road segment, as shown in Algorithm 3. It is worth noting that EC can receive failure notifications from a variety of vehicles. After receiving the first notifications, EC will discard any remaining with a similar path ID.

C. Path Duration Estimation

A vehicular network can be viewed as a static network at any given time; however, based on a mobility model, changes in the topology can be predicted up to a certain time. Because a number of communication links between the source and destination are possible, estimating all possible paths is not always feasible. Given that the operation of "on demand" routing protocols is inextricably linked with the shortest path, the investigation of average path intervals based on the shortest path principle is appropriate and significant.

We have introduced a novel probabilistic model for estimating path duration for our SDN-based road-aware routing protocol in this section. The proposed protocol has the unique ability to find not only potential paths but also more durable and stable paths based on various parameters such as average number of hops, link connectivity, direction, and velocity.

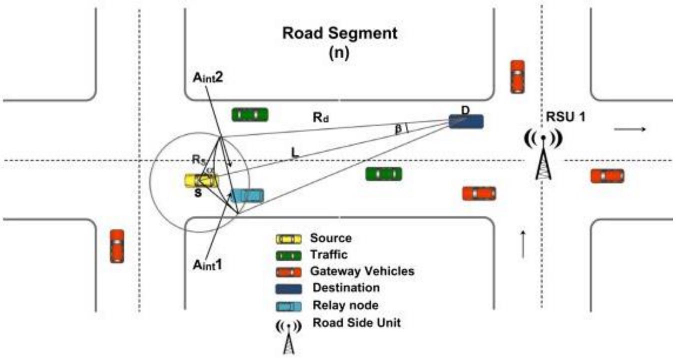

Fig. 4: Network model for selecting relay node for path estimation

To ensure reliable links, the SDN controller determines link duration for each path using discrete parameters. From the previously described beacons, each vehicle determines its neighbor's position and velocity. This data is then used to forecast how long two neighboring vehicles will be within communication range of each other (Fig. 4)

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 segement
Failure_notif: route failure notification
Upon receiving the packet at EC( )

```
if (notification_type == path_failure) then
  function path_faialed_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)
```

D. Mathematical Model

The goal of this section is to compute an expression for path duration between two vehicles using mathematical relationships such as link duration and average number of hops. In our estimation model, we used a traditional traffic flow principle to represent an efficient vehicular environment for data forwarding. Vehicles are assumed to follow Poisson distributed arrivals in our proposed model to obtain the probability distribution function (pdf).

Table 1: Description of variables used in mathematical model

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

1. Area for Next Hop

To find a stable path between the source and the destination, we need a communication link with the fewest hops to the destination. Because the node closest to the border line, towards the destination, covers the greatest distance, the number of hops between source and destination is reduced. This is why we chose the area for our next hop at the very end of the transmission range. The area that must be calculated is also known as the area of intersection of the circles with radius RS and Rd. It is worth noting that the area of the circular segment is equal to the area of the circular sector minus the area of the triangular portion. The standard formula for calculating the region's area is as follows:

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

The total area of both segments can be calculated using equation (2) below:

$$A_{Total} = A_{int1} + A_{int2} \tag{2}$$

However, with reference to fig. 4

$$A_{int1} \simeq \left[\frac{(\alpha - \sin(\alpha)) \cdot R_S^2}{2}\right] \tag{3}$$

And

$$A_{int2} \simeq \left[\frac{(\beta - \sin(\beta)) \cdot R_S^2}{2}\right] \tag{4}$$

By substituting (3) and (4) into (2), the total area expected for the relay node is:

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

Note that the A_{int1} and A_{int2} represents that region of the circle in which the source vehicle looks for the neighbouring vehicle.

2. Node Relative Velocity

The direction and speed of a vehicle are important factors in estimating path duration because the direction of a vehicle directly affects link duration. At this point, we are more interested in the relationship and derivation of relative velocity with its various cases. For our model, we consider a city scenario with moving vehicles in both directions. Assume we have two moving vehicles with velocities v_1 and v_2 , respectively, and the distance between them is d, whereas a vehicle's radio communication range is expressed as r. Four general cases for the velocities of these moving vehicles are considered in order to determine different velocities.

Case 1: When both vehicles are traveling in the same direction at the same velocity, a communication link exists between them for a long time T_1. The relative velocity between the vehicles, with velocities v_1 and v_2, can be calculated using the cosine law as follows:

$$|\overrightarrow{v_r}| = \sqrt{v_1^2 + v_2^2 - 2v_1v_2cos\theta} \quad (6)$$

When two vehicles have the same direction but different velocities, the vehicle with the greater velocity is represented as: v_1 which is λtimes greater than v_2. Whereas λ ranges from 1 to 4. And given that we consider the vehicles to be in the same direction with the same velocity, then:

$$V_1 = V_2 = V$$

And the angle

$$\theta = 0$$

Then:

$$|\overrightarrow{v_r}| = 0$$

Case 2: both vehicles travel in the same direction but with different velocities where V_1 is α times greater than V_2.

$$\Rightarrow V_2 = \alpha V_1 \ ; \ and \quad \theta = 0$$

Then

$$|\overrightarrow{v_r}| = V_1(a - 1)$$

Case 3: both vehicles have the same velocity but travel in opposite direction.

$$\Rightarrow \theta = \pi \quad ; \ and \ V_1 = V_2 = V$$

Then

$$|\overrightarrow{v_r}| = 2V$$

Case 4: both vehicles travel with different velocities and in opposite direction.

$$\Rightarrow V_2 = \alpha V_1 \ ; \ and \ \theta = \pi$$

Then

$$|\overrightarrow{v_r}| = V_1(\alpha + 1)$$

3. Probability Density Function of relative velocity

Based on previous findings, v_r can be represented as a random variable, and its expected relative velocity function can be calculated using the probability density function (pdf) as follows:

$$E(v_r) = \int_{\infty}^{-\infty} v_r \, dv_r \quad (7)$$

Further simplification to our scenario on (7) can be written as:

$$E(v_r) = \int_{vmin}^{vmax} \int_{vmin}^{vmax} \int_{0}^{\pi} fv_1 \cdot fv_2 \cdot f(\theta_1, \theta_2) * \sqrt{(v_1 + v_2 - 2v_1v_2cos\theta)} \, dv_1 \, dv_2 \, d(\theta_1, \theta_2) \quad (8)$$

The pdf for a relative velocity is represented by Equation 8. To be more specific, the pdf for each case can be derived as:

$$E(v_r) = \int_{vmin}^{vmax} \int_{vmin}^{vmax} (\lambda \pm 1) \cdot fv_1 \cdot fv_2 \, dv_1 \, dv_2 \quad (9)$$

When we have two vehicles moving in the same direction, we can use (9) with the minus sign and the positive sign when both vehicles are moving in opposite directions at velocities v_1 and v_2, respectively

4. Average number of neigbour nodes

The number of vehicles between the source and destination can be used to calculate the average number of neighbor nodes. To calculate the average number of hops, the total distance between the source and destination must be recognized. Vehicles on the road within the source transmission range use the Poisson distribution model. Furthermore, if the destination node is within the sender's transmission range, the probability of finding it is the same as the probability of finding the next hop node. The distance to the first next hop can be calculated as follows:

$$N_H = \frac{L}{D_L} \quad (10)$$

5. Link Connectivity

In this section, we will calculate the time for each vehicle's link duration in order to find the shortest route possible. The famous equation: $Time = \frac{Distance}{Speed}$ will now be used to calculate the link duration.

$$T_L = \frac{R_S - D_L}{V_S - V_{NH}} \quad (11)$$

where D_L is the total distance between the next hop source node and the source node RS. Furthermore, T_L demonstrates link connectivity, which holds the value of link residual life. The following formula can be used to calculate the distance D_L between the next hop and the source node.

$$D_L = \frac{n \cdot R_s}{n + 1} \quad (12$$

And the remaining link life is:

$$T_L = \frac{D_R}{R_V} \quad (13)$$

Where $D_R = R_s - D_L$ is the distance required by the next hop to move out of the transmission range of source node.

The pdf of T_L can be represented as:

$$F_T(T_L) = \int_0^V R_V f_{dR_V}(T_L R_V, V)\, dV \quad (14)$$

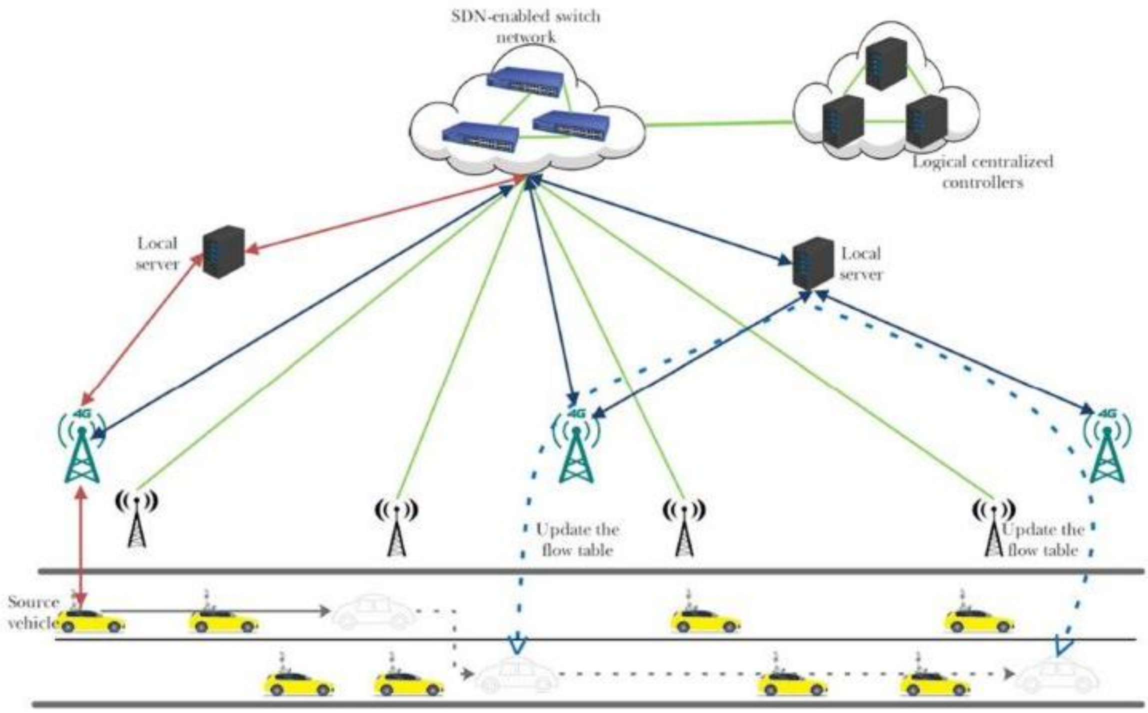

Fig. 5: SDN, EC Flow rule installation

E. Path time Estimation

In VANETS, a complete path estimation is one of the basic design parameters. We consider the remaining link life to determine the pdf of the path duration. If $T_{L1}, T_{L2}, T_{L3}, T_{L4}$ and $T_{L(NH)}$ are the remaining link time between the hops 1,2,3,4 and N_H, then the pdf for a path duration is calculated as:

$$T_{LPath} = MIN(T_{L1}, T_{L2}, T_{L3}, T_{L4}, T_{N_H} \quad (15)$$

In addition, the pdf of TL can be calculated using Baye's theorem [31].

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

Where $C_{(T)} = 1 - F_T$ is the complementary cumulative distribution function (CDF) of T_{LPath} and F_T. The average path duration can now be calculated using the following equation.

$$T_{Lpath}(average) = \int_0^\alpha T_L f(T_L)\, dT_L \quad (17)$$

F. Working Operation of the Edge Controller

This section of the paper focuses on how the proposed Edge Controllers (ECs) for the mobility problem work. Link breakage due to topology change is a critical issue in vehicular ad-hoc networks and it has a negative impact on overall network performance. Our goal in this section is to solve the mobility issue. As shown in Fig. 5, the concept of EC is used in this research to collect information from each vehicle in order to know its topology in real-time. Vehicles on the road send information to EC on a regular basis, including their $speed, direction, road\ ID$, and $position$, among other things. When this information reaches the EC, it searches its database for vehicle information. If a legitimate change in vehicle position occurs, EC updates the central SDN controller with the vehicle's most recent information; otherwise, EC updates its database only with the new vehicle position. This method of updating the SDN controller allows the network to handle less traffic. Furthermore, this information is provided to the EC at regular intervals of 0.1 seconds, which is calculated after running a few experiments, but this time cannot be fixed and can vary depending on the vehicle and road conditions. When the vehicle is moving quickly, information is sent more frequently. The mathematical formula for calculating hello packet interval is as follows:

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

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

where S is the distance traveled, V is the vehicle speed (m/s), and c is a fixed interval, such as 0.1 sec. There are two scenarios for sending hello packets to the controller. (a) Hello packets are forwarded after a fixed time interval, for example, 0.1 sec. (b) With regard to position, if position changes by 10m, a hello packet is sent. While eqn. (18) is for calculating the hello interval, eqn. (19), on the other hand, represents the type of interval that a vehicle will choose. In other words, a vehicle first computes the interval using eqn. 18, and then takes the minimum of the fixed interval, c, and the interval determined by vehicle speed and position. If the vehicle speed is 105 m/sec in the example above, the hello packet interval will be 0.095 sec. Meanwhile, if the vehicle's speed is less than 100m/s, the interval is set to the default value of 0.1 sec. Finally, providing real-time information about vehicles on the road forces the SDN controller to act quickly by installing rules on time.

V. Evaluation and Results

A. *Simulation Setup*

We used the SUMO simulator to evaluate the proposed SD-IoV-based road-aware approach. The road information, which includes traffic lights, speed limits, road directions, and junctions, with an area of $3000m \times 3000m$ is scaled down from the open street map (OSM) road topology, as shown in Fig. 6. The SUMO simulator is then used to generate trace files from the *.osm* file containing all of the road information. This is then fed into MININET-WiFi to run additional simulations [32]. Figure 7 depicts the system architecture used in our research.

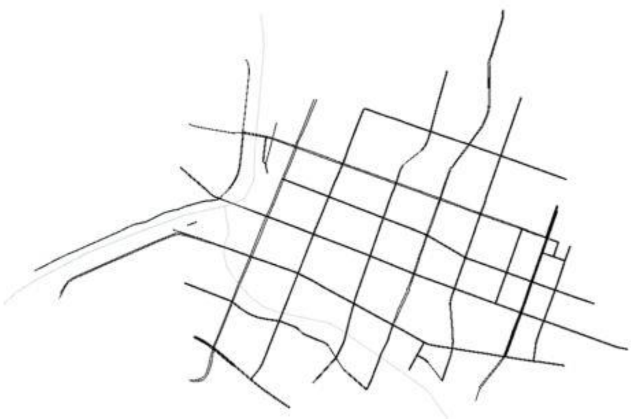

Fig. 6: Simulated Road topology of Techiman street

For the simulations, the log-distance propagation model with an exponent of 4.1 was used. The minimum vehicle speed was varied from $0\ to\ 4m/s$, while $5\ to\ 25m/s$ was used for the maximum speed. The number of RSUs used in the simulation ranges from 3 to 7 and finally, 4 eNodeBs were used. See Table 2 for more information.

Table 2: Simulation Setup

Parameters	Value
Number of Vehicles	20 – 100
Simulation Area	3000 × 3000m
Time for Simulation	1200s
Vehicle Speed (min)	0 – 4 m/s
Vehicle Speed (max)	5 – 25 m/s
Simulation scenario	Roads with various intersections
Propagation model	Log-distance model
Value of exponent	4.1
Traffic type	User Datagram Protocol (UDP)
Data Packet Size	512 bytes

B. *Compared Protocols*

We simulated the hybrid road-aware routing protocol with no infrastructure support with SD-IoV to prove the working of SDN with SD-IoV in order to evaluate the proposed protocol's functionality. HRAR (Hybrid Road-Aware Routing Protocol) is a data transfer protocol built specifically for VANETs. Roads are split into road segments in HRAR depending on their intersections. To reduce control routing overhead, HRAR proposes the notion of gateway vehicles. RREQ is not sent to every vehicle; instead, it is only sent to gateway vehicles, which are then responsible for finding the path in a multi-hop method. SDN-enabled routing protocol for VANETs [27] is also compared in Sect V (4 of C) to demonstrate the efficiency of the proposed method in terms of control overhead.

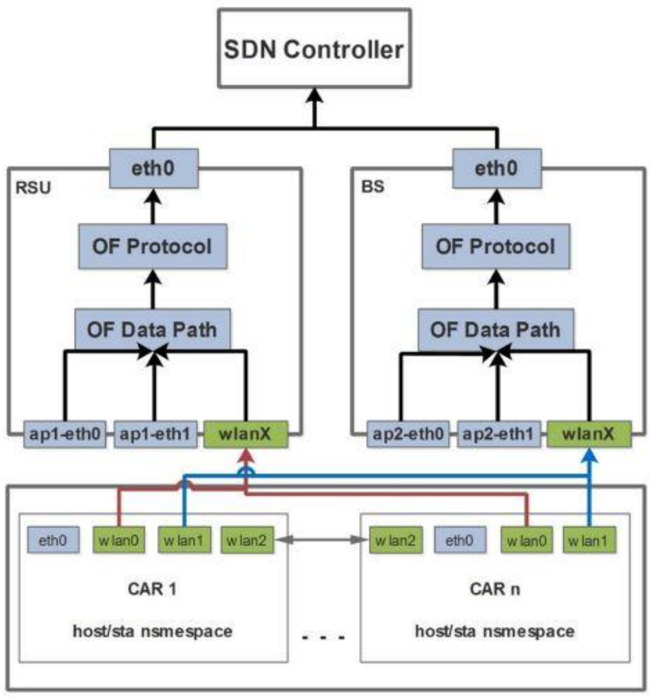

Fig. 7: Overview of system model, car architecture, and OpenFlow programmability in Mininnet-WiFi

C. *Metrics*

The performance of the proposed protocol for SDN-based IoV is evaluated using the metrics described below.

1. *Routing Overhead*

In Figure 8, the routing overhead for each protocol is calculated. It was observed that the total routing overhead increases as the average vehicle speed increases. And this is because redundant transmissions occur in highly congested road segments. Hence, more traffic is generated, thereby resulting in the increase in routing overhead for HRAR and SD-IoV as shown in Figure 8. Moreover, when the number of RSUs are increased, our SD-IoV has a relatively lower routing overhead. This is because the EC maintains the road segment level routing table and only discovers routes outside of the road segments when necessary; the SDN controller is solely responsible for this, and there is no need to transmit the RREQ to every road segment.

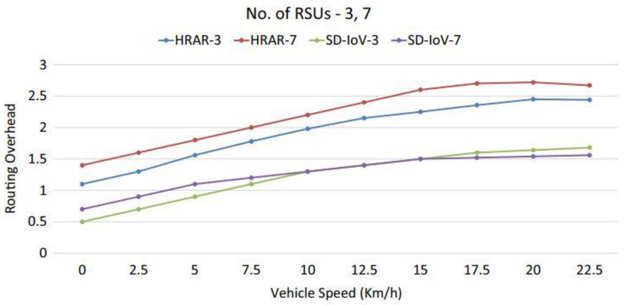

Fig. 8: HRAR and SD-IoV routing overhead Vs Vehicle Speed (Km/h)

2. Packet Delivery Ratio

We investigate the impact of varying average vehicle speed on the performance our proposed protocol in terms of packet delivery ratio in this section. The results of the delivery ratio for various RSUs with discrete node speed are shown in Figure 9. When compared to HRAR, SD-IoV outperforms at both low and high vehicle speeds, as seen in Figure 9. It shows that the packet delivery ratio rises as the number of RSUs along the route increases, however in our instance, performance rises with current RSUs when compared to others. Figure 9 demonstrates that the packet delivery for SD-IoV increases more as the number of RSUs increases. The rationale for this is because as the speed of the source vehicle increases, it will identify its neighbors more quickly, giving it more opportunity to transfer data packets to intermediate cars with a high probability of packet delivery success.

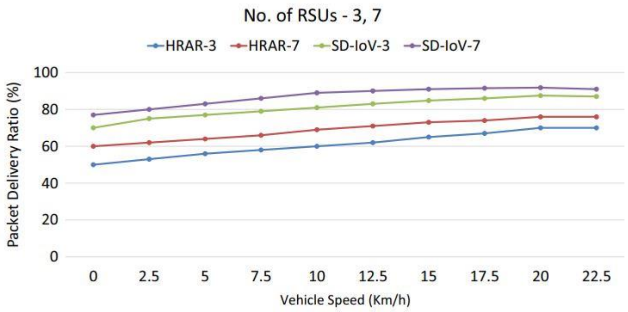

Fig. 9: HRAR and Sd-IoV packet delivery ratio vs Vehicle Speed (Km/h)

3. End-to-End Delay

End-to-end delay is determined by the number of hops and network congestion. SD-IoV performs better because the controller calculates and maintains pathways for each segment to each RSU in real time. Which helps with the rapid delivery of data packets. HRAR, on the other hand, exhibits more end-to-end delay than SD-IoV as average vehicle speed increases, as illustrated in Fig. 10. When the average vehicle speed changes from 4 km/h to 25 km/h, SD-IoV exhibits a very little delay.

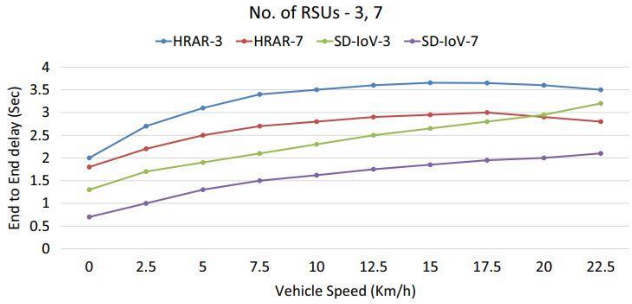

Fig. 10: HRAR and SD-IoV end-to-end delay vs vehicle speed (km/h)

4. *Control Messages vs Node density and vehicle speed*

Messages sent between the SDN controller and open flow switches, which operate as mobile nodes or vehicles with mobility, are referred to as control traffic. The control messages issued by the SDN controller to amend flow tables with new flow rules are seen in Figures 11 and 12. The SDN controller communicates with RSUs, base stations, and vehicles using the OpenFlow channel. Multiple channels can be supported by an SDN controller at the same time, and messages sent over these channels must adhere to OVS protocol specifications. In general, the control plane and data plane support a variety of messages, but in our simulations, we focused on three of them: packet in, packet out, and hello messages, as shown in Table 3. Initially, as shown in Figs. 11 and 12, we can see that as node density increases, the SDN controller must cater to additional vehicles and forward additional control messages. However, when compared to HRAR and SCGRP [27], SD-IoV shows fewer control messages between the controller and the switch because SCGRP forwards data traffic based on the shortest path calculated by the SDN controller and using the vehicle ID. When a vehicle moves out of range of the source vehicle, it notifies the controller of the link failure.

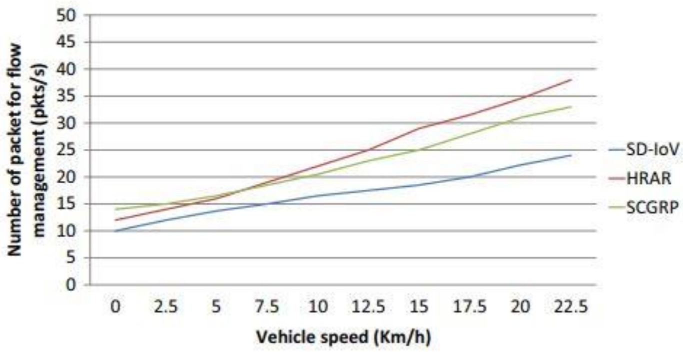

Fig. 11: Control traffic for SD-IoV, HRAR, and SCGRP vs Vehicle speed

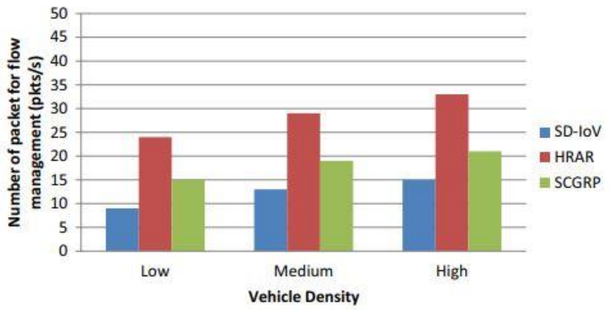

Fig. 12: Control traffic for SD-IoV, HRAR, and SCGRP vs vehicle density

Table 3: 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 actions, OF header.

VI. Conclusion

Software defined networking has gradually expanded its footprints not only in a single terrain, fixed network (e.g., wired networks, data centers), but also in dynamic networks such as wireless networks, Ad-hoc Networks, and internet of vehicles, among others. However, using wired network protocols for data dissemination in the highly dynamic environment of IoV is not appropriate and necessitates a number of changes. A novel road-aware routing protocol for SDN-based IoV is proposed in this paper to forward data packets in V2V and V2X modes. In the proposed protocol, roads are divided into road segments based on intersections and given unique identifiers. Within its vicinity, EC maintains road segment level routing. Using hello beacons, vehicles communicate with the EC about their location (vehicle speed, direction, road id, vehicle id, and position). This data is then sent to the SDN controller, which uses it to find the best available paths between the source and the destination. SD-IoV uses RSUs and neighbor vehicles to forward data packets, but only uses cellular networks (i.e. 4G/5G) to send control/emergency packets. This is due to the fact that cellular networks are more expensive while providing half the bandwidth of RSUs for data dissemination. To evaluate our proposed protocol for SDN-based IoV, we used Open Street Map (OSM) to obtain road information, SUMO to generate vehicular traffic on the roads, and MININET-WiFi to test our protocol. In MININET-WiFi, the default controller is used to install rules for each vehicle based on the information provided by it. The protocol has been rigorously tested using multiple scenarios at various speeds, dynamic deployment of RSUs and eNodeBs, and different road segment lengths. Finally, the proposed protocol is tested against the well-known routing protocol HRAR, and the results show that SD-IoV outperforms in almost all aspects of routing techniques. Because this topic is very active among researchers and industries these days, various studies can be conducted on it. Vehicle position and trajectory prediction, for example, can be very useful for better rule installation and heterogeneous communication using SDN technology in smart city applications. Other research topics include early collision warning, traffic flow forecasting, and, last but not least, the requirements for 5G deployment.

References

[1] Baofeng et al (2020), Survey on the Internet of Vehicles: Network Architectures and Applications. In IEEE Communications Standards magazine. Digital Object Identifier: 10.1109 /MCOMSTD .001. 1900053.

[2] Sanjeev Singh, Rakesh Kumar Jha (2016), A survey on Software Defined Networking: Architecture for Next Generation Network. Springer Science+Business Media Nee York 2016.

[3] 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

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

[5] 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

[6] 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

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

[8] Garavello M, Piccoli B (2006) Trafc fow on networks, vol 1. American institute of mathematical sciences, Springfeld.

[9] Daganzo C, Daganzo CF (1997) Fundamentals of transportation and trafc operations, vol 30. Pergamon, Oxford

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

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

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

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

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

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

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

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

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

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

[20] Manzo R, Piccoli B, Rarità L (2012) Optimal distribution of trafc fows in emergency cases. Eur J Appl Math 23(4):515–535 Muralidhar K, Geethanjali N (2013) A novel dead end and packet los

[21] 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

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

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

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

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

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

[27] 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

[28] Abbas Muhammad Tahir (2019), Two-level Hierarchical Hybrid Road-Aware Routing for Vehicular Networks". *KSII Transactions on Internet and Information Systems*. Korean Society for Internet Information (KSII). doi: 10.3837/tiis.2019.07.003.

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

[30] 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

[31] 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

[32] 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