Link Stability Based Hybrid Routing Protocol For Software Defined Vehicular Networks

K. L. Kushan Sudheera*, Maode Ma†
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
*KUSHANSU001@e.ntu.edu.sg
†EMDMA@ntu.edu.sg

Peter Han Joo Chong
Department of Electrical and Electronic Engineering
Auckland University of Technology, New Zealand
PETER.CHONG@aut.ac.nz

Abstract*—The dynamic nature of vehicular networks imposes a lot of challenges in multi-hop data transmission as links are vulnerable in their existence. Thus, the packets frequently find it difficult to get through to the destination as links only exist for a limited amount of time. The broadcasting based conventional routing protocols struggle to cope with these situations due to the lack of global network information. But, with the novel Software Defined Vehicular Network (SDVN) architecture, link stability can be better scrutinized pertaining to the availability of global network view. However, due to the imperfections in the SDVN architectures and dynamicity in vehicular networks, the control plane may not possess all the network information at a given point in time. Considering all these factors, in this paper, we introduce a novel routing framework for SDVN which is composed of both centralized and distributed routing mechanisms. The resulting hybrid routing framework focuses on finding stable multiple short routes that can deliver a given number of packets, and in case of uncertain network conditions, a broadcasting approach is adopted. The overall problem is formulated as a minimum cost capacitated flow problem and the effectiveness is demonstrated comparatively via extensive simulations.

I. Introduction

Software Defined Vehicular Networks (SDVN) recently emerged as a new paradigm and now, it has been attracting a lot of interest from the academia due to the empowerment of programmability and flexibility in vehicular network [1]. With the recent applications introducing 5G into vehicular domain with multiple wireless interfaces, SDVN is making its way for a promising futuristic vehicular architecture.

The novel SDVN paradigm can, not only be used to empower programmability but also to resolve some of the pressing issues in Vehicular Ad-hoc Network (VANET) such as handling of dynamicity. As repeatedly emphasised in previous research [2], [3], the inherent dynamic nature of vehicular network imposes serious difficulties in successfully delivering packets. This is very crucial especially in applications such as file sharing, file downloading, public transport scheduling, and congestion avoidance, which requires transmission of sequences of packets over multiple hops. The dynamicity in links causes frequent topology changes as link existence ceases with time. Packets during multi-hop transmission may have a very small window opening to get through to the destination before the route breaches. By the time packets reach an intermediate node, the opening window of the link can disappear or not enough for the packets to get through, leading to packet drops [4]. It is found that the precipitous drop in route lifetime is likely to cause a route error when packets are sent over three to four hops in the network [2].

Conventional VANET routing protocols struggles to cope with this dilemma as they find it difficult to accurately estimate the route validity due to the lack of global network information, and the limitations put on by their general broadcast approaches in finding routes [3], [5]. Further, this becomes severe when a sequence of packets is to be transmitted and also if the computed route is very long, as in both situations it takes a lot of time to deliver all the packets, increasing vulnerability. There is a high probability for at least a single link in the path to break down as vehicles move continuously during this long time duration [2]. Even though several packets may be able to get through the computed path, it is highly possible that the packets at the end of the sequence to miss the opportunity to get through before the route terminate its existence [4]. This forces the system to frequently find new paths leading to higher delays and lower Packet Delivery Ratios (PDR).

However, SDVN provides a bird's eye view over the network and thus, the link stability can be better analyzed and optimized routes can be found to deliver more number of packets [1], [6]. Despite having new opportunities to effectively manage various routing attributes such as stability, researchers have mostly focused on delivering packets in the shortest possible path [1], [6], [7], except for [8], [9], where both control and data planes have worked collaboratively in routing the packets. Therefore, in this paper, we utilize the available local network view in finding more optimal routes by considering multiple factors such as distance and stability.

Even though, the current studies assume that the control plane has a global or at least a localized view over the network, yet, in reality, it may not always possess updated network information which is required to accurately find stable paths. Lack of cellular and infrastructure support (Road Side Units (RSUs)) in certain road areas (rural) can cause for an unreliable southbound interface. The control plane may not possess network information related to these areas (**blind areas**). Also, fast changing networking areas such as urban, makes the current network information obsolete very quickly. Therefore, the above conditions may create situations such as,

- Inability to find a route between the source and the destination due to unavailability of network information
- Inability to accurately estimate the stability of paths owing to outdated network information.

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

Referring to the above conditions, we further expect to incorporate some of the conventional VANET broadcasting techniques in an optimization based centralized routing framework, forming a hybrid routing protocol for SDVN which focuses on delivering a given number of packets in the network via multiple stable routes efficiently. The overall routing problem is formulated as a minimum cost capacitated flow problem by adhering the features of both maximum flow problem and minimum cost flow problem. The problem is solved with a less time complexity following an incremental packet allocation scheme.

The remainder of the paper is structured as follows. Section II discusses the background of the routing stability problem. In section III, we formulate both centralized and distributed segments of the hybrid routing framework. Section IV analyses the results, and finally, Section V concludes this paper.

II. Problem Statement

In this section, we layout the fundamentals related to the link instability problem in routing.

A. System Model

The main elements in SDVN are the SDN control plane and the data plane (application layer is integrated with the control plane). The SDN controller(s) contains all the logical functionalities and take decisions, including routing on behalf of the data plane elements. The SDN data plane consists of vehicle On Board Units (OBUs) and RSUs, and executes actions based on the flow rules supplied by the control plane [10]. The network information such as the node positions, velocities, directions and their neighbors are gathered periodically by the control plane via status update messages.

In the literature of SDVN architectures, the controllers are placed in various locations such as Internet and RSUs, and different wireless technologies are used for southbound interface to bridge the control and data planes [1], [7]. However, in this paper, we use a hierarchical distributed SDVN [11] architecture as shown in Fig. 1, where the control plane is split into two layers and the top layer is placed on the Internet while the bottom layer is at the RSU level. LTE is used to connect with the top tier while Dedicated Short Range Communication (DSRC) is used to connect with the bottom tier as well as other data plane elements. A source routing based operation is used in delivering flow rules to the data plane nodes [10].

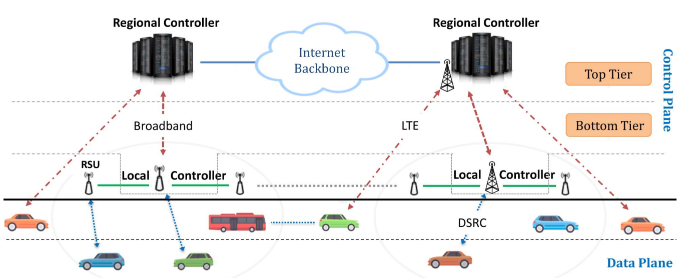


Fig. 1. SDVN system architecture [11].

B. Background of the Problem

As mentioned earlier, the links in vehicular networks are highly unstable due to the high mobility of vehicles. Thus, they only exist for a short amount of time. Therefore, it is imperatively important to analyze whether a selected path can deliver a given number of packets over the multiple hops before the route terminates its existence.

For an example in the network map shown in Fig. 2, if the lifetime of the link 9-8 (in the shortest path) is 8 ms and the total transmission time of a single hop is 5 ms, the packet will not be able to get through the link 9-8. When the packet arrives at node 9, link 9-8 only has a window opening of 3 ms which is not enough for the whole packet to get through. If more packets need to be transmitted in a sequence, ostensibly the shortest path may not be enough. In the same example, if the lifetime of the link 9-8 is 11 ms, it can only deliver 1 packet through the path of S-9-8-D. Any more packets via this path will result in unsuccessful transmissions. A better option would be to transmit the 1^{st} packet via S-9-8-D and the rest via S-2-3-4-5-6-D long path, if the path is stable enough.

Moreover, a bulk of packets cannot be transmitted consecutively just right after the other in a wireless environment, because of the employed channel access mechanisms. Carrier-Sense Multiple Access/Collision Avoidance (CSMA/CA) used in Enhanced Distributed Channel Access (EDCA) mechanism of VANET limits multiple packets transmitted over a wireless channel at the same time to avoid possible collisions. For an instance, in the network in Fig. 2, only a single packet transmission can take place at a time in the links of 1-2, 2-3, 3-4, as multiple transmission will cause for packet collisions. However, if the packets are scheduled to transmit with a reasonable gap between them, the neighboring packets will not have a significant effect on each other. At the same time, if two neighboring links belong to different channels such as wireless and wired, then packets can be transmitted in parallel in two links.

C. Possible Solutions

If the SDN control plane is given accurate and updated network information, then following centralized routing approaches can be used to find routes.

One possible way is to select a single path which is stable enough to transmit all the given packets [3]. Even though this

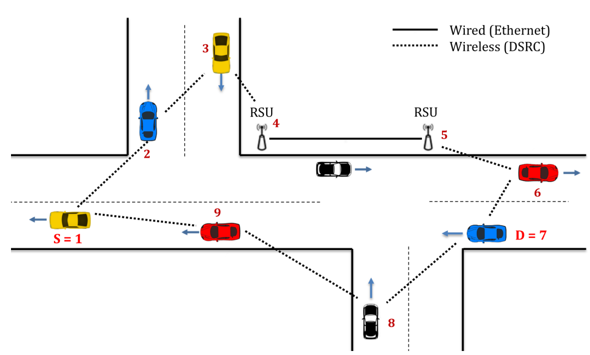


Fig. 2. An instance of a vehicular network.

approach may deliver more number of packets successfully, it may not be optimal as it can lead to longer routes resulting in higher communication delays. In reality, as the vehicular network topology is rapidly changing, there is also a possibility that this approach may not be able to find a feasible solution.

Alternatively, we can always find the shortest path without taking the route stability into consideration [1], [6]. As discussed earlier, with this approach, some packets may not be able to make it to the destination resulting in a relatively low PDR. Even though this seems to lead into a low communication latency, the fact that the nodes will be compelled to consult the controller when the supplied flow rules expire can limit the latency reduction. Moreover, frequent consultation of the control plane generates a higher communication overhead.

Deviating from above methods, in our routing protocol, we attempt to balance this trade-off between latency and PDR by routing the packets in multiple shortest paths which are collectively stable enough to deliver a given number of packets. However, as the control plane may not possess all the required network information at all times, and moreover, the information coming from highly dynamic areas soon become obsolete, it is not always possible to accurately estimate route expiration. In order to overcome these uncertainties, we instruct the nodes to rely on their neighbors by making use of broadcasting during these conditions. So a general routing decision may consist of both unicast and broadcast directives.

III. Problem Formulation

In this section, we will formulate and solve both centralized and distributed counterparts of the routing problem.

A. Centralized Routing

In the centralized part of the protocol, we route packets in stable enough paths via unicast. The motivation for the problem formulation is taken from Air Traffic Flow Management (ATFM) where flights are routed in the airspace within different source and destination airports under link capacity constraints [12]. Similarly, here we route individual packets between the source and destination nodes under link capacity (in terms of stability) constraints. However, unlike in ATFM, vehicular networks present several unique challenges such as, vulnerability of links (the capacity of the link is a function of time) and limitations on continuous packet transmission by channel access mechanisms.

1) Notations: Given a network graph, $G = (V, E)$ where V and E refer to the set of nodes (OBUs and RSUs), and edges connecting the nodes, respectively, and source node, $s \in V$ and destination node, $d \in V$, our objective is to successfully deliver a given number of packets (n) with the least cost without exceeding the dynamic link capacity values. Each link from node i to j (refer as ij) is associated with two metrics, cost (C_{ij}) and lifetime (T_{ij}). We use the transmission time of a reference packet as the cost metric (C_{ij}). Therefore, the cost is different in wired and wireless links. The link lifetime (T_{ij}) refers to the remaining lifetime of the link at the moment flow rules reaches the source node.

We use t^{w}_{hop} and t^{wl}_{hop} to represent the total time to send a packet within a single wired and wireless hop, respectively (hop time). We include the transmission ($t^{x}_{trans.}$), propagation ($t^{x}_{prop.}$), queuing ($t^{x}_{q.}$) and contention ($t^{x}_{cont.}$) times [11] in hop time (x represents the link channel as in wired (w) or wireless (wl)), and neglect the processing time assuming its effect is negligible. It is important to note that in reality, both t^{w}_{hop} and t^{wl}_{hop} are affected by other on-going transmissions in that particular area. However, in current SDVN systems, the controller does not have accurate information about all ongoing transmissions that can affect the considered transmission, as several transmissions do not require controller's intervention. Therefore, we model t^{w}_{hop} and t^{wl}_{hop} with added uncertainties ($\triangle t^{w}$, $\triangle t^{wl}$) as in (1).

$$t^{x}_{hop} = t^{x}_{trans.} + t^{x}_{prop.} + t^{x}_{q.} + t^{x}_{cont.} + \triangle t^{x} \tag{1}$$

where $x \in \{w, wl\}$.

We use X_{ij} to indicate the packet flow on link ij, and Y^{p}_{ij} to indicate whether the trajectory of the p^{th} packet lies on the link ij or not. $H^{p,w}_{s,r}$ and $H^{p,wl}_{s,r}$ represent the number of wired and wireless hops, respectively, from the source node, s to the intermediate node, r via the estimated route of the packet p. $T^{p}_{s,r}$ refers to the time taken for the p^{th} packet to reach the intermediate node, r from source node, s via the estimated path. $N(= |V|)$ and $L(= |E|)$ refer to the number of nodes and links in the network, respectively. Finally, U^{p}_{ij} represents the capacity of the link ij in terms of number of packets at the time packet p reaches node i.

2) Objective function: Our objective is to route individual packets from source s to destination d in a feasible path which has the least sum of costs. Following a minimum cost flow problem, the objective function can be formulated as in (2).

$$\text{minimize} \quad \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} C_{ij}.X_{ij} \tag{2}$$

3) Constraints:

- **Routing**

$$\sum_{\forall j \in N^{out}_{s}} Y^{p}_{sj} = 1, \quad \forall p \tag{3}$$

$$-\sum_{\forall i \in N^{in}_{k}} Y^{p}_{ik} + \sum_{\forall j \in N^{out}_{k}} Y^{p}_{kj} = 0, \quad \forall k, p \tag{4}$$

$$-\sum_{\forall i \in N^{in}_{d}} Y^{p}_{id} = -1, \quad \forall p \tag{5}$$

where N^{in}_{i} and N^{out}_{i} refer to the set of neighbors with incoming links to node i and outgoing links from node i, respectively. The above flow conservation based equations are responsible for routing the n packets in the network.

$$X_{ij} = \sum_{p=1}^{n} Y^{p}_{ij}, \quad \forall i, j \tag{6}$$

Equation (6) accumulates the number of individual packets on the link ij and equates it to the flow of that route.

- **Timing**

$$H_{s,r}^{p,w} = \sum_{\forall ij \in L_{s,r}^{u,w}} Y_{ij}^{p}, \quad \forall p, r \tag{7}$$

$$H_{s,r}^{p,wl} = \sum_{\forall ij \in L_{s,r}^{u,wl}} Y_{ij}^{p}, \quad \forall p, r \tag{8}$$

where $L_{s,r}^{u,w}$ and $L_{s,r}^{u,wl}$ indicate the set of all wired and wireless upstream links of node r from source s, respectively.

Equation (7) and (8) calculate the number of wired and wireless hops, respectively, from the source, s to the intermediate node, r for the packet p using the path variable, Y_{ij}^{p}. The binary variable Y_{ij}^{p} will be 1 in the links where the packet p travels leading up to the node r.

The value of $T_{s,r}^{p}$ primarily depends on the interval between consecutive packets, t_{gap}. If the t_{gap} is considerably high ($> 3.t_{hop}^{wl}$), then a considered packet transmission will not be disturbed by other preceding and successive packets. Thus, we can formulate ($T_{s,r}^{p}$) when $t_{gap} > 3.t_{hop}^{wl}$ as,

$$T_{s,r}^{p} = (p-1).t_{gap} + \sum_{\forall ij \in L_{s,r}^{u,w}} t_{hop}^{w}.Y_{ij}^{p} + \sum_{\forall ij \in L_{s,r}^{u,wl}} t_{hop}^{wl}.Y_{ij}^{p}$$

$$T_{s,r}^{p} = (p-1).t_{gap} + H_{s,r}^{p,w}.t_{hop}^{w} + H_{s,r}^{p,wl}.t_{hop}^{wl} \tag{9}$$

When the gap between consecutive packets is very small, it becomes very complex to accurately model the time a packet takes to reach a given node due to the involvement of randomness. However, assuming the worst case, it is safe to say that if the packets are transmitted with a gap of $3t_{hop}^{wl}$ between consecutive packets, then the channel access mechanism will not disturb the consecutive packet flow, as once a packet is due, the previous packet will be out of the corresponding node's coverage. Further, it will not be affected by the hidden node problem. Thus, when $t_{gap} < 3.t_{hop}^{wl}$,

$$T_{s,r}^{p} = 3(p-1).t_{hop}^{wl} + H_{s,r}^{p,w}.t_{hop}^{w} + H_{s,r}^{p,wl}.t_{hop}^{wl} \tag{10}$$

It should be noted that in the above derivations, we assume that two neighboring nodes in a selected path are on the edges of each other's transmission range as the system selects the shortest possible paths. Therefore, two nodes in a computed path that both have a common neighbor will not be in each other's transmission range.

- **Capacity**

We refer the capacity of a link as its ability to pass packets via its path before the link expires. Even though the link capacity is a function of time, we only consider the link capacity at discrete points (at packet arrival times), thus, the problem converts into a discrete time problem.

The capacity of link ij at the time packet p reaches node i, U_{ij}^{p}, can be expressed as in (11).

$$U_{ij}^{p} = \max(0, \lfloor \frac{T_{ij} - T_{s,i}^{p}}{t_{hop}^{ij}} \rfloor), \quad \forall i, j, p \tag{11}$$

where t_{hop}^{ij} refers to the packet transmission time of the considered link, ij. Depending on the medium of the link, it will be equal to t_{hop}^{w} or t_{hop}^{wl}.

Finally, the capacity constraint of link ij at the time packet p reaches node i can be represented as in (12).

$$Y_{ij}^{p} \leq U_{ij}^{p}, \quad \forall i, j, p \tag{12}$$

Overall formulation of the link dynamic based centralized packet routing problem falls into the minimum cost capacitated flow problem category and can be summarized as below.

$$\underset{X_{ij},\ Y_{ij}^{p},\ H_{s,r}^{p,w},\ H_{s,r}^{p,wl},\ T_{s,r}^{p},\ U_{ij}^{p}}{\text{minimize}} \quad \sum_{i=1}^{N} \sum_{j=1,\ i \neq j}^{N} C_{ij}.X_{ij} \tag{13a}$$

$$\text{subject to} \quad \varphi(.) \tag{13b}$$

where $\varphi(.)$ includes eq. (3, 4, 5, 6, 7, 8, 9, 10, 11) and (12).

B. Distributed Routing

The above centralized routing formulations are derived assuming possession of accurate network information. However, as stated previously, the control plane may not acquire all the required information at all times. In these cases, we divert to distributed mechanisms to find the remaining routes.

1) Case 1: Computed path's stability is under scrutiny

When a path is found from a source to a destination in the centralized routing section, it is possible that some links in the path hardly satisfy the minimum stability criteria. If the lifetime of a link in the path is between 0 and minimum stability limit + δ (uncertainty for stability), we will specify to use broadcasting in transmitting the packet in the link.

For an example, in the Fig. 3, if the lifetime of link AB is at scrutiny, we will transmit the packet until A via unicast and broadcast the packet from A, expecting that via few broadcast cycles the packet will be received by node B. The intuition here is to treat the link AB as a blind area and expect the other nodes' support on reaching the node B. Unlike conventional VANET routing protocols such as AODV where the path from S to D is discovered entirely based on broadcasting, here we only have a limited broadcasting cycle. The upper bound of the Time To Live (TTL) count is set to twice the estimated hop count from A to B, which we compute by dividing the distance between A and B by A's transmission range. The resulting path information can be represented as in Fig. 4.

2) Case 2: Inability to find a feasible path

Under this, we consider another two sub cases as follows.

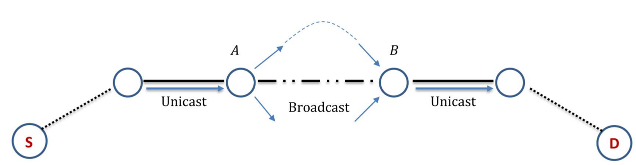


Fig. 3. Inclusion of broadcasting in a route.

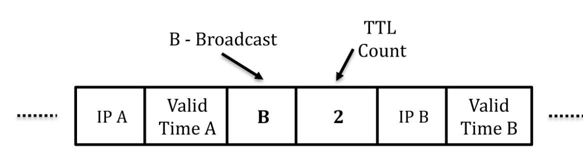


Fig. 4. Route information.

Subcase 1 : Updated position information about the destination node is not found

In this case, we find another node (agent node) that can replace the destination node in centralized routing based on the destination node's last known position. This is simply executed by finding the minimum out of distances between the destination node's estimated location and each other node's location in a potentially interested area.

Then, we find the shortest stable path from the source node to the new agent node. If a path is not found, the above procedure is repeated until a node is found with a feasible path. Upon finding the agent node and a stable path, we add a broadcast cycle from the agent node to the destination node with an adaptive TTL count based on their estimated distances.

The main idea behind the above distributed approaches is to limit the broadcasting in the network by transmitting the packets to a closer agent node in an uncertain area. As the broadcasting starts from a closer node, a limited broadcasting cycle will occur in the network minimizing the congestion.

Subcase 2 : Destination is known, but a path cannot be found.

This is most likely to occur if there is a blind area in between the source and the destination. In reality, a path could exist in between the considered nodes, but the control plane does not have enough information to find a path via centralized routing.

Motivated by Bi-Directional Dijkstra algorithm, here we instruct both source and destination nodes to start broadcasting towards each other directionally. In this way, it is possible that both broadcasting cycles will meet at an intermediate node limiting the broadcasting area as shown in Fig. 5. In order to limit the broadcasting cycle, TTL is set to $\lceil \frac{distance\ between\ S\ and\ D}{2*transmission\ range} \rceil + 1$.

Once the broadcasting circles meet at a common node (M), routes can be found by back tracing the traveled nodes. If M has connectivity with a controller, the route information can be updated in the control plane and the data transmission can be started along the discovered path. Otherwise, the computed path information can be sent to the source node via RREP messages along the data plane.[1]

IV. Performance Evaluation

In this section, we simulate our hybrid routing protocol in the NS-3 simulator and analyze the performance against a shortest path based SDVN routing algorithm (SHORTEST) which resembles [1], [6], [7], most stable route based modified conventional VANET scheme (ROMSGP*) [3] and AODV which does not take the link stability into consideration.

[1]In order to solve the routing problem efficiently, an incremental optimization technique [13] is followed, where successive shortest paths are incrementally searched and packets are allocated as much as possible without violating the stated constraints.

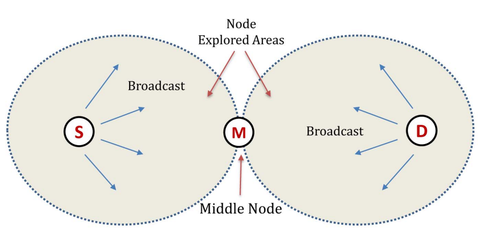


Fig. 5. Broadcasting from both source and destination.

A. Simulation Configuration

We used a simplified version of the Nanyang Technological University, Singapore campus map with 15 scattered RSUs as the simulation environment (1500*1500 m^2). However, the total RSUs did not cover all the road network and thus, some areas created blind spots in the network. Hello and beacon message interval was set to 250 ms. At each simulation run, quarter (1/4) of the vehicles sent packets to randomly selected destinations. Data transmission was based on a Poisson distribution with a mean of 4 seconds. Each transmission consisted of multiple packets and the number of packets was selected based on an uniform distribution between 1-10. The gap between consecutive packets was set to 100 ms. Additional parameters and values are tabulated in the Table I.

We used the average hop count (Fig. 6a), average route life time (Fig. 6b), total packet overhead (Fig. 6c), PDR (Fig. 7a) and average end to end delay (Fig. 7b) as the performance metrics to evaluate the efficiency of previously mentioned routing protocols. The results were taken by varying the average vehicular speed (number of vehicles was set to 60).

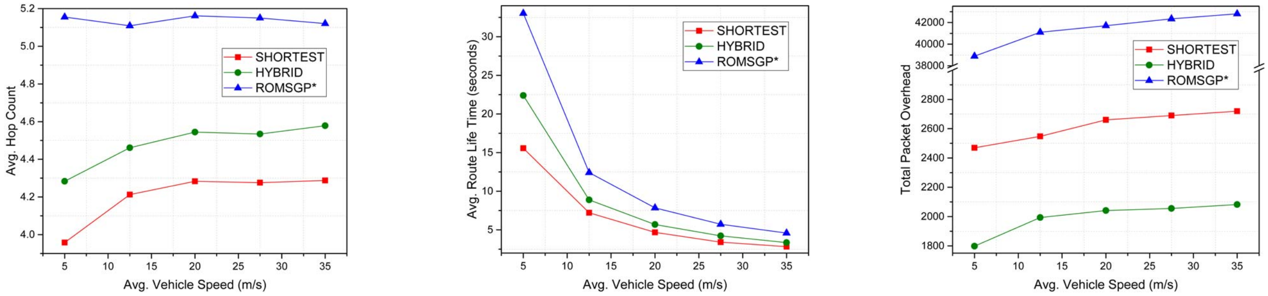
B. Results Analysis

According to Fig. 6a, as the name implies, the shortest path routes define the lower bound in hop count. The ROMSGP* focuses on finding most stable paths, thus a higher average hop count. In terms of route stability (Fig. 6b), as expected the ROMSGP* exhibits the highest average route lifetime, while the shortest path method displays the lowest due to the inconsideration of route expiration. However, the proposed hybrid routing protocol balances this trade-off by finding the shortest and at the same time sufficiently stable multiple paths. Hence, the extracted routes from the proposed method associate very low hop counts, which are closer to the lower bound and moderate route lifetimes, which are collectively stable enough to deliver a given number of packets.

More importantly, the proposed routing scheme yields the lowest packet overhead in discovering the routes, while ROMSGP* entertains an enormous overhead due to its broadcasting based route discovery phase. In the shortest path

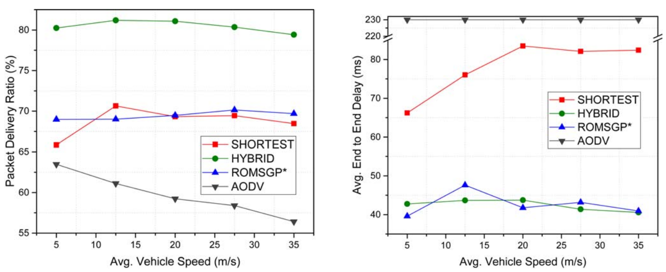
TABLE I. Simulation parameters.

Parameter	Value
Simulation time	400 seconds per run with 20 runs per each point
Packet sizes	Packet In = 32 bytes, Flow Mod = 56+ bytes, Data = 1500 bytes
Transmission	Type = UDP, Range = 200 m (DSRC)
Propagation model	Friis Propagation Loss Model
Wireless network	DSRC (IEEE 802.11p) = 6 Mbps
Wired network	Ethernet (IEEE 802.3) = 100 Mbps, Propagation delay = 1 ms
Internet Backbone	Data Rate = 100 Gbps, Propagation delay = 10 ms



(a) Average hop count (b) Average route life time (c) Total packet overhead

Fig. 6. Comparison of different routing attributes.



(a) PDR (b) Latency

Fig. 7. Comparison of different routing metrics.

approach, the fact that the SDN nodes consult the controller repeatedly as the given flow rules expire, causes for the shift in the packet overhead shown in Fig. 6c.

The competitive edge gained by the proposed scheme in above routing attributes can be stated as the driving factor in outperforming all the three reference protocols with significant margins in terms of PDR (Fig. 7a) at all speed levels. Naturally, as the vehicle speed increases, the vulnerability of links also escalate. As AODV does not consider the link stability, its PDR significantly drops at higher speeds, due to the selection of unstable links. Since, the ROMSGP* utilizes the most stable paths, it has maintained a steady packet reception rate against increasing mobility. Yet, the unavailability of global information and the broadcasting based route discovery approach which leads less optimized routes, have limited its PDR rate. Further, the shortest path protocol has maintained a competitive PDR with ROMSGP*, but both profiles are significantly lower than the hybrid scheme's PDR profile.

Even though, the routes explored from ROMSGP* are comparatively longer, the higher stability in routes caused less route setups, and helped to exhibit a lower latency for ROMSGP*. In the proposed scheme, all the routes are comparatively shorter and found beforehand. Negative side includes the controller consultation and processing times. Pertaining to above reasons, a lower competitive delay profile can be seen from the proposed hybrid scheme as well. In contrary, both AODV and shortest path methods are compelled to find routes multiple times as the links expire and thus, higher latencies.

V. Conclusion

In this study, we introduced a hybrid routing protocol for SDVN which incorporates both centralized and distributed routing techniques. The resulting routing scheme routes packets in the shortest possible and stable paths in multi-hop data transmission scenario. The protocol makes use of limited broadcasting in uncertain conditions where accurate network information is not available. The simulation results showed that the proposed routing framework outperforms the existing SDVN and VANET routing schemes in most cases.

References

[1] I. Ku, L. You, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, "Towards software-defined VANET: Architecture and services," in *Ad Hoc Networking Workshop (MED-HOC-NET), 2014 13th Annual Mediterranean*, 2014, pp. 103–110.

[2] J. J. Blum, A. Eskandarian, and L. J. Hoffman, "Challenges of intervehicle ad hoc networks," *Trans. Intell. Transport. Sys.*, vol. 5, no. 4, pp. 347–351, 2004.

[3] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "A Stable Routing Protocol to Support ITS Services in VANET Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337–3347, Nov 2007.

[4] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Netw. Manag.*, vol. 11, no. 1, pp. 3–30, 2001.

[5] C. Perkins, E. Belding-Royer, and S. Das, *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC Editor, 2003.

[6] M. Zhu, J. Cao, D. Pang, Z. He, and M. Xu, "SDN-Based Routing for Efficient Message Propagation in VANET," in *Wireless Algorithms, Systems, and Applications: 10th International Conference, WASA 2015, Qufu, China, August 10-12, 2015, Proceedings*, K. Xu and H. Zhu, Eds. Cham: Springer International Publishing, 2015, pp. 788–797.

[7] H. Zongjian, C. Jiannong, and L. Xuefeng, "SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication," *IEEE Network Magazine Special Issue on Software Defined Wireless Networks*, 2015.

[8] Z. Ming, C. Zhiping, C. Jiannong, and X. Ming, "Efficient multiple-copy routing in software-defined vehicular networks," in *2015 International Conference on Information and Communications Technologies (ICT 2015)*, 2015, pp. 1–6.

[9] B. Dong, W. Wu, Z. Yang, and J. Li, "Software Defined Networking Based On-Demand Routing Protocol in Vehicle Ad Hoc Networks," in *2016 12th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, Dec 2016, pp. 207–213.

[10] K. L. K. Sudheera, M. Ma, and P. H. J. Chong, "Efficient Flow Instantiation via Source Routing in Software Defined Vehicular Networks," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017.

[11] K. L. K. Sudheera, M. Ma, G. G. M. N. Ali, and P. H. J. Chong, "Delay efficient software defined networking based architecture for vehicular networks," in *2016 IEEE International Conference on Communication Systems (ICCS)*, 2016, pp. 1–6.

[12] G. G. N. Sandamali, R. Su, Y. Zhang, and Q. Li, "Flight routing and scheduling with departure uncertainties in air traffic flow management," in *2017 13th IEEE International Conference on Control Automation (ICCA)*, July 2017, pp. 301–306.

[13] A. M. Sharp, "Incremental algorithms: solving problems in a changing world," Ph.D. dissertation, Cornell University, 2007.