Connectivity aware tribrid routing framework for a generalized software defined vehicular network

Liyanage, Kushan Sudheera Kalupahana; Ma, Maode; Chong, Peter Han Joo

2019

Liyanage, K. S. K., Ma, M., & Chong, P. H. J. (2019). Connectivity aware tribrid routing framework for a generalized software defined vehicular network. *Computer Networks*, 152, 167–177. doi:10.1016/j.comnet.2019.01.040

<https://hdl.handle.net/10356/144530>

<https://doi.org/10.1016/j.comnet.2019.01.040>

© 2019 Elsevier B.V. All rights reserved. This paper was published in *Computer Networks* and is made available with permission of Elsevier B.V.

Downloaded on 26 Apr 2023 16:28:26 SGT

Connectivity Aware Tribrid Routing Framework For A Generalized Software Defined Vehicular Network

Kalupahana Liyanage Kushan Sudheera^{a,*}, Maode Ma^a, Peter Han Joo Chong^b

^a*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*

^b*Department of Electrical and Electronic Engineering, Auckland University of Technology, New Zealand*

Abstract

Data dissemination is a fundamental, yet one of the pressing issues in vehicular communication due to the associated high dynamicity. The vehicular network topology frequently keeps changing, limiting the lifetime of the links. This imposes serious difficulties in data transmission, especially in multi-hop applications as the vulnerability escalates when the packets are transmitted over multiple hops. The broadcasting based Vehicular Ad-hoc Network (VANET) routing protocols struggle to cope with this dilemma due to the lack of global network information. But, with novel Software Defined Vehicular Network (SDVN), link stability can be better scrutinized pertaining to the availability of global network view. Yet, the architectural challenges in SDVN can limit the availability of network information confining the empowerment of Software Defined Networking (SDN). Thus, in this paper, we introduce a link connectivity aware novel routing framework for a general SDVN acknowledging the limitations in data availability as well. The routing protocol comprises of both centralized and distributed routing techniques and makes use of unicast, broadcast, and store, carry and forward concepts. The resulting tribrid routing framework focuses on finding stable enough shortest routes that can deliver a given set of packets satisfying the required Quality of Services (QoSs) in terms of latency. In case of network uncertainties, the protocol incorporates broadcasting based distributed techniques along with unicast routing. In sparse network conditions, the model aims to deliver the packets in the optimal path with the least store and carry time within the QoS requirement. The routing protocol follows an incremental algorithm where extracted paths are tested for the feasibility on a case by case basis.

Keywords: Software defined networking, Vehicular ad-hoc networks, Software defined vehicular networks, Routing, Stability, Sparse networks, QoS

1. Introduction

Vehicular networks are associated with unique high mobility and it causes frequent topological changes. The links in the network are highly dynamic and their connectivity exists only for a limited amount of time [1, 2, 3]. This creates severe difficulties in data transmission applications, especially in multi-hop scenarios such as file sharing, file downloading, public transport scheduling, and congestion avoidance [3], as one or more links in the computed path, may not exist at the time the packets reach the corresponding links. In fact, 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 [4]. It is also relatively hard to find routes from one point to another in multi-hop scenarios, as the network topology frequently changes.

Conventional Vehicular Ad hoc Network (VANET) routing protocols struggle to efficiently manage this problem

as they lack a sense of global network image. The general attitude of routing in VANET is towards finding the shortest possible path [5]. However, several studies do address the link stability problem. Yet, their solutions are mostly based on finding the most stable path [6, 7, 8]. Both approaches carry several pros and cons. Ostensibly, the shortest path approach does not consider the route stability and thus, prone to the above mentioned link connectivity related problems. In contrast, the most stable path can lead to longer routes increasing the latency. At the same time, due to the high end to end delay associated with these paths, the vulnerability in link existence can escalate as well. Broadcasting based routing protocols [9, 10] find it hard to accurately discover stable enough short paths due to the unavailability of global network information, which we expect to overcome with the introduction of Software Defined Networking (SDN) into VANET.

Software Defined Vehicular Network (SDVN) recently emerged as a new paradigm and it has been attracting a lot of interest from the academia due to the empowerment of programmability and flexibility in vehicular networks [11, 12]. With the recent induction of 5G into the vehicular domain with multiple wireless interfaces, SDVN is making

*Corresponding author

Email addresses: KUSHANSU001@e.ntu.edu.sg (Kalupahana Liyanage Kushan Sudheera), EMDMA@ntu.edu.sg (Maode Ma), PETER.CHONG@aut.ac.nz (Peter Han Joo Chong)

its way for a promising futuristic vehicular architecture. The logically centralized manageability and availability of global network information of SDVN play key roles in this study in scrutinizing the link connectivity and stability [13, 14]. As such, one of the primary focuses of this paper is to find a set of short and stable enough routes that can successfully deliver the given set of packets.

The above approach assumes the availability of network information, yet, in reality, this is not always realized [15]. Due to certain circumstances such as absence of network coverage, the SDN control plane may not always receive updated network information and thus, may not be able to find the required paths [11, 16]. Also, fast changing networking areas such as highway and urban, makes the current network information obsolete very quickly. The above cases can generate black hauls (areas with obsolete or no any network information) in the topology. This is one of the concerns that we address in this paper to enhance the applicability of the routing protocol. For that, we incorporate some of the distributed routing techniques such as broadcasting, and store, carry and forward (SCF) in combination with a unicast protocol.

Another aspiration of this work is to make the routing protocol independent of the network conditions. Unlike other networks, vehicular networks can be dense, or sparse depending on the location, time, etc. The same area could be dense at peak hours and sparse at night. However, current VANET routing protocols generally target a specific network condition. But, in this paper, we articulate the generality in VANET, and propose an improved scheme to find optimal SCF routes with least store time, for sparse conditions.

Referring to the above conditions, we propose a tribrid routing protocol which consists of unicast, broadcast and SCF techniques for a general vehicular network. The focus is on maximizing the delivery of packets satisfying given latency based QoSs. The routing model adopts an incremental algorithm, where successive shortest paths are assessed for the feasibility in terms of stability and latency based QoS. The arising disturbances are treated with broadcast and SCF routing on a case by case basis.

The remainder of the paper is structured as follows. Section 2 surveys some of the existing work on routing in SDVN. In Section 3 and 4, we discuss the routing problem in detail and formulate the problem, respectively. Section 5 analyses the results, and finally, Section 6 concludes this paper.

2. Related Work

The conventional VANET routing protocols such as Ad hoc On-Demand Distance Vector (AODV) [9] lack global network information and generally based on the information exchange between vehicles in finding the routes. Therefore, despite been scalable, they struggle to maintain a competitive performance, especially in terms of packet re-

ception. This is repeatedly emphasized in previous researches such as [6, 13, 3].

Among various reasons, the inconsideration of path stability can be rooted as one of the major causes for the above performance deficiency. The studies that consider stability such as [6] which only utilizes the vehicles in the same velocity direction, focus on finding the most stable path which can lead to longer routes and higher delays. Further, the above protocol does not consider the future connectivity of links and thus, suffers heavily under sparse network conditions. Similarly, in [8], the relay node is selected based on the longest lifetime and additionally best signal quality. Also, the model works in a reactive manner, i.e. the relay node is selected after receiving all the signals, and analyzing the aforementioned parameters based on the received signal's characteristics.

The uniqueness in the routing protocols that targets sparse networks lies in the storing of packets until new connections are made. Generally, these protocols forward packets to the first encountered node rather than waiting for a more optimal forwarding [17]. This creates possibilities of longer routes or route failures. In contrary, studies such as [18] utilize a more optimal selection of forwarding nodes based on the vector directions of vehicular position and velocity.

However, lacking in the existing literature is a proper optimization on the selection of routes acknowledging the link connectivity and network conditions. The novel SDVN makes the global network information available to the controller [19], which could be used to enhance the performance by optimizing the link selection as emphasized above. Yet, the current researches in SDVN mostly focus on finding the shortest route as in [20]. Here, the packets are routed in the paths with the shortest distance and the route selection is based on the Dijkstra algorithm with the cost metric as the minimum optimistic time which symbolizes the time two nodes will take to meet each other.

In [21], the control plane and data plane both collectively work to route packets towards the destination. The control plane helps to send multiple copies of the packet to several selected neighbors which are closer to the destination. As the packets transverse in the network, the controller continues to guide and converge them towards the destination as in a prey process. But the involvement of the controller at each step can lead to higher communication delays and costs.

B. Dong et al. [23] have used a combination of distributed and centralized routing algorithms in a distributed SDVN architecture. In their system, RSUs which act as the local controllers, route the packets within road segments using centralized routing protocols based on the Bellman-Ford or Floyd algorithms. Within a local controller's span, the packets are transmitted following an optimized AODV scheme. Yet, there are several limitations in this protocol such as assumption of a dense and a connected network, and sole dependency on one way lanes, and RSUs at intersections, which is not always realized.

Figure 1: Proposed hierarchical distributed SDVN architecture (HD-SDVN [22]).

In compared to the existing literature, the uniqueness of our paper is in the scrutiny of path stability and optimization of latency. Further, it acknowledges the practical circumstances in SDVN such as partial availability of network information and makes use of broadcast based distributed techniques to overcome the disturbances and uncertainties. Proclaiming a general vehicular network including sparse network conditions, the model explores future connectivity between vehicular nodes as well. The storing and forwarding of packets is handled more accurately applying optimization on the global network information.

3. Problem Statement

In this section, we lay out the fundamentals related to the link connectivity problem in routing. First, we introduce the underlying SDVN system model that will be used in the paper. Next, we discuss some of the background related to the above problem with existing solutions.

3.1. System Model

SDVN decouples the control algorithms from the hardware allowing modularity and programmability in the vehicular networks. The SDN control plane together with the application layer facilitates the logically centralized algorithms and takes decisions 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 forwarded by the control plane.

According to the SDVN literature, the control plane is most commonly found at the Internet servers, and Long Term Evolution (LTE) is used to bridge the control and the data planes [24]. Due to the associated high latency in this architecture, we have previously proposed a hierarchical distributed SDVN architecture named as HD-SDVN

[19, 22], which splits the control plane and brings a layer of control to the RSU level minimizing the latency of reaching the controllers and allowing a competitive latency to prevail in data communication. The southbound communication exploits both LTE and Dedicated Short Range Communication (DSRC [25]) to access the top tier and bottom tier of controllers, respectively. The HD-SDVN architecture adopted in this paper is graphically illustrated in Fig. 1.

In addition to the periodical beacon message exchange between the vehicles, the current literature emphasizes the collection of network information by the control plane. However, this generates a considerable packet overhead in the network, and thus, we have proposed to offload this responsibility only to RSUs when the vehicles are under the RSU coverage. This information which includes node positions, velocities, directions and their neighbors, are used to build topological maps and vital in the decision making process. We further employ a source routing based flow instantiation operation to forward the flow rules to the respective data plane elements as discussed in [26].

3.2. Background of the Problem

As mentioned earlier, vehicular links are highly unstable and their connectivity exists only for a limited time due to the associated high mobility of vehicles. Therefore, the link connectivity should be a decisive factor in selecting a route in addition to the latency associated with the route [27]. Overall, all the links in a selected route should exist from the route discovered time, until the time the packet(s) passes each link and reaches the destination.

In the network instance shown in Fig. 2, the common selection of the shortest path between the vehicle S and D, would be S-3-4-5-D. At a macroscopic level, let us assume the residual lifetime of link 3-4 at the moment last network information update received, is 12 ms, and the time from

the update received moment until the respective flow rules reach the source node, which includes the route setup time, is 10 ms. Further, if the time taken to transmit a data packet in wireless 802.11p is 5 ms, then the respective data packet cannot be delivered via S-3-4-5-D, even though it is the shortest path. In this case, when the data packet reaches the node 3, the link 3-4 will no longer be connected. Therefore, the ideal path selection in this case would be S-1-2-4-5-D, given that it is stable enough.

Figure 2: A vehicular network instance.

The current literature to overcome the link stability problem is based on the selection of the most stable path. However, as mentioned above, this approach could lead to higher delays due to the possibility of selection of longer routes. However, the proposed system focuses on optimizing the latency associated with the route without violating the necessary link stability requirement.

As mentioned earlier, due to certain circumstances, a feasible route may not be able to realize using the above method. The concerning circumstances that we focus on this paper are listed below.

1. Obsolete network information
2. Unavailability of network information
3. Sparse network conditions

The above factors individually or in a combination can cause black hauls restricting the connectivity between the source and the destination as shown in Fig. 3. Therefore, they may develop situations such as,

1. Inability to find a route between the source and the destination
2. Inability to accurately estimate the stability of paths

Figure 3: Disturbance in network information.

This barrier (disturbance) could be found at different locations in the network as shown in Fig. 3, and different treatments are applied based on the location of the barrier. For an example, if the barrier is found at the location I, it implies that the source node has no neighbors, where as if

the barrier is at location II, it implies that the connectivity between the source and the destination is breached at some point either by obsolete or no network information or sparse network conditions. The barrier at location III refers to the situation where the destination's location is unknown. This is most likely due to the unavailability of network information.

The disturbance at location I and II are handled by SCF as accurate information including mobility and location of the nodes adjacent to the barrier is available. However, at location III, the SCF might not be the ideal solution as updated mobility and location information of the destination is not available, and thus, the computation of store time in this situation can lead to inaccurate estimations. Therefore, we incorporate a limited broadcast cycle in the area with no network information, assuming packets will reach the destination in a few broadcast hops.

4. Problem Formulation

In this section, we formulate the tribrid routing problem addressing the issues mentioned in the problem statement. Our focus here is to route a given set of packets in the shortest possible paths without violating the stability requirements. In case of uncertain conditions, we incorporate several distributed routing techniques. Overall, we make use of the following routing concepts in the system.

1. Unicast based centralized routing
2. Broadcast based distributed routing
3. Store, carry and forward routing

4.1. Centralized Unicast Routing

In this part, we assess the connectivity related stability of the routes. One main assumption made here is that updated network information is available at the control plane.

The proposed model routes a given set of packets via unicast in the vehicular network between the source and the destination under link stability and latency based QoS constraints. The motivation for the problem formulation is taken from Air Traffic Flow Management (ATFM) [28]. However, unlike in ATFM, here, the link connectivity is a function of the time. Further, the routing scheme is based on an incremental algorithm in which we incrementally find successive shortest paths and check the feasibility for the given packets.

4.1.1. 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, the source node, $s \in V$ and the destination node, $d \in V$, our objective is to successfully deliver a given set of packets, $P(|P| = n_p)$ with least delivery times without exceeding the dynamic link capacity values and delay tolerances of the packets. Each link from node i to j (refer to as ij) is associated with

three metrics, the lifetime (τ_{ij}^s), effective velocity (v_{ij}^{eff}) and transmission time ($t_{ij}^{hop} = \text{hop time}$).

The lifetime of link ij (τ_{ij}^s) refers to the residual lifetime of the link at the moment flow rules reaches the source node. v_{ij}^{eff} refers to the effective velocity associated with the link ij , and $v_{ij}^{eff} = \sqrt{\left[v_i^{\rightarrow ij}\right]^2 + \left[v_j^{\leftarrow ij}\right]^2}$ [25]. Here, $v_i^{\rightarrow ij}$ indicates the velocity component of the node i in the direction of the node j .

Depending on the link type, hop time t_{ij}^{hop} , will either be t_w^{hop} (wired) and t_{wl}^{hop} (wireless). We include the transmission (t_x^{trans}), propagation (t_x^{prop}), queuing (t_x^q) and contention (t_x^{cont}) times [22] 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 possess accurate information about all ongoing transmissions that can affect the considered transmission, as several transmissions do not require a controller's intervention. Therefore, we model t_w^{hop} and t_{wl}^{hop} with added uncertainties (Δt_w , Δt_{wl}) as in (1).

$$t_x^{hop} = t_x^{trans} + t_x^{prop} + t_x^q + t_x^{cont} + \Delta t_x, \quad (1)$$

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

Y_{ij}^p is the main decision variable used in the model, which indicates the packet p 's trajectory through the link ij . Moreover, $H_{s,r}^{w,p}$ and $H_{s,r}^{wl,p}$ represent the number of wired and wireless hops, respectively, from the source node, s to an intermediate node, r via the estimated route for the packet p . $T_{s,r}^p$ refers to the time taken by the p^{th} packet to reach an intermediate node, r from the source node, s via the estimated path. N refers to the number of nodes in the network. Further, N_i^{in} and N_i^{out} refer to the set of neighbors with incoming links to node i and outgoing links from node i , respectively. ϵ_{max}^p refers to the delay tolerance margin of the p^{th} packet.

4.1.2. Objective Function

The objective of this model is to maximize the delivery of packets by routing them in short and stable paths, and thus, the objective function can be formulated as in (2).

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1, i \neq j}^N \sum_{j=1}^N \sum_{p=1}^{n_p} [C_t \cdot t_{ij}^{hop} + C_v \cdot v_{ij}^{eff}] Y_{ij}^p \\ & - \sum_{p=1}^{n_p} \sum_{i \in N_d^{in}} C_p \cdot Y_{id}^p \end{aligned} \quad (2)$$

The term, t_{ij}^{hop} is used to incentivize the selection of shortest paths (paths with least travel times), and v_{ij}^{eff} to encourage less dynamic links. Studies such as [25, 8] have shown that when the links are associated with very high

effective velocities, there is a higher probability for packet errors, and thus, we discourage selection of such highly dynamic links.

Moreover, the last term, $\sum_{p=1}^{n_p} \sum_{i \in N_d^{in}} C_p \cdot Y_{id}^p$ is used to maximize the number of delivered packets. The values of the coefficients are selected as $C_p \gg C_t \gg C_v$ in order to symbolize their respective priorities.

4.1.3. Constraints

• Successive shortest paths

The following equations model the routing of the packets in the network. The constraints (3) and (5) are modeled as inequalities allowing packet cancellation options, as it is probable that all the packets may not be able to find stable paths to reach the destination.

$$\sum_{j \in N_{s,out}} Y_{sj}^p \leq 1, \quad (3)$$

$$- \sum_{i \in N_k^{in}} Y_{ik}^p + \sum_{j \in N_k^{out}} Y_{kj}^p = 0, \quad \forall k \in V \quad (4)$$

$$- \sum_{i \in N_d^{in}} Y_{id}^p \leq -1, \quad (5)$$

The equation (6) avoids loops in the path that can occur due to the bidirectional nature of the links in the vehicular networks.

$$Y_{ij} + Y_{ji} \leq 1, \quad \forall i, j \in V \quad (6)$$

The above problem can be solved with the help of Bidirectional shortest path algorithm (SPA) [29] with the cost metric (C_{ij}) as $C_t \cdot t_{ij}^{hop} + C_v \cdot v_{ij}^{eff}$. The shortest path algorithm is more efficient in solving the routing problem, as it does not scrutinize all the links in the network to find the optimal path unlike with an optimization solver. Furthermore, as the links in the vehicular environment are bidirectional, we can employ the bi-directional SPA. Unlike in the traditional Dijkstra algorithm, here we start to find the path from both source and destination until the circles meet in a middle node. The bi-directional SPA explores lesser number of nodes than the conventional SPA and thus, slightly faster.

• Feasibility assessment

The above extracted routes are then gone through a feasibility assessment to check the stability and the latency requirement for a given packet (the packets are incrementally considered from the given set P). The feasibility check basically assesses whether all the links in the selected path stay connected until the given packet passes through the corresponding links or not. For that, we formulate the variables related to packet arrival times as follows.

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

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

Equation (7) and (8) calculate the number of wired and wireless hops, respectively, from the source s to an 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 travels leading up to the node r . $L_{s,r}^{u,x}$ indicates the set of all upstream wired ($x = w$) and wireless ($x = wl$) links of node r from s .

The packet arrival time to an intermediate node r , $T_{s,r}^p$, can be modeled as follows. Primarily, the timing depends on the gap between consecutive packets, (t_{gap}). If the $t_{gap} \gg t_{wl}^{hop}$, then the channel access mechanism will not be influential¹ as the packets are adequately spanned. In this case, $T_{s,r}^p$ can be modeled as in (9).

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

When the packets are closely timed ($t_{gap} \approx t_{wl}^{hop}$), continuous packet transmission will be disturbed by channel access mechanisms, especially when there are multiple wireless consecutive links in a selected path as shown in Fig. 4. Multiple packets can not be transmitted at the same time in these consecutive wireless links without realizing a packet collision, and further, transmission patterns (packet order and timing) depend on the timings of the Request To Send/Clear To Send (RTS/CTS) packets. However, assuming the worst case in terms of timing, $T_{s,r}^p$ can be formulated as in (10).

Figure 4: Multiple wireless links in a selected path.

Let, the maximum number of consecutive wireless links in a selected path be n_b .

$$T_{s,r}^p = \begin{cases} n_b(p-1)t_{wl}^{hop} + H_{s,r}^{w,p}.t_w^{hop} + H_{s,r}^{wl,p}.t_{wl}^{hop}; & n_b < 3 \\ 3(p-1)t_{wl}^{hop} + H_{s,r}^{w,p}.t_w^{hop} + H_{s,r}^{wl,p}.t_{wl}^{hop}; & n_b \geq 3 \end{cases} \quad (10)$$

The residual lifetime of a common link ($r - r'$) at the moment the packet p reaches the starting node of that link ($\tau_{rr'}^{r,p}$) can be expressed as,

$$\tau_{rr'}^{r,p} = \tau_{rr'}^s - T_{s,r}^p \quad \forall rr' \in E \quad (11)$$

where $\tau_{rr'}^s$ is the residual lifetime of link rr' at the moment flow rules reach the source node. The link lifetime estimation process is discussed in depth in our previous research [13], thus, will not be analyzed in this paper.

For a route to be feasible, $\tau_{rr'}^{r,p}$ of all the links in the path should be higher than the transmission time of the

packet in the corresponding link ($r - r'$). The condition for a stable transmission in link rr' for the packet p is formulated as in (12).

$$\tau_{rr'}^{r,p} > t_{rr'}^{hop} \quad \forall rr' \in E \quad (12)$$

However, the above assessment need not be performed for all the links in the path. It can be narrowed to one necessary and sufficient condition, which is the assessment of the bottleneck link of the path. The bottleneck link is a qualified representation of the path's stability. If the bottleneck link passes the feasibility test for the packet p , it can be safely concluded that all the links in the path are stable for the packet p . The bottleneck in a path is the link with the minimum residual lifetime at the moment the respective packet reaches the starting node of the link, as expressed in (13).

$$\text{Bottleneck link} \rightarrow \text{link with min}:(\tau_{rr'}^{r,p}) \quad (13)$$

If the bottleneck link passes the feasibility assessment, we allocate packets as much as possible to the selected path, until the feasibility criteria is satisfied.

If the bottleneck link does not satisfy the stability criteria in (12), the model goes for the successive shortest paths. In order to find successive paths, we use a modified version of the Yen's SPA [30]. As we cannot accurately predict how many paths are required to be tested to find a feasible set of paths to transmit the given set of packet, we opt to find the next best shortest path incrementally rather than finding k-shortest paths at once. A successive shortest path is found by iteratively deleting the links in the previously found shortest path and finding the pool of shortest paths from the resulting networks. The path with the least cost in the shortest path pool will be selected as the successive shortest path. The feasibility study will be performed on each successive shortest path until either all the packets are allocated or the maximum permissible route exploration limit is reached.

We further check whether the selected paths can deliver the given set of packets within their delay tolerance margins from the condition (14).

$$T_{s,d}^p < \epsilon_{max}^p \quad (14)$$

The unicast routing protocol assumes the availability of accurate and updated network information. However, as stated previously, in practical scenarios, this is not always realized. In these scenarios, we apply different treatments based on the location of the disturbance in the network information, as discussed in the next subsection.

4.2. Distributed Broadcast Routing

We apply broadcast routing in two network instances as follows.

- **Case 1:** Computed path's stability is under scrutiny

¹The effect from other ongoing transmissions is encompassed into the packet travel time in (1).

Even though a considered route in the unicast approach may satisfy the stability criteria, uncertainties involved in vehicular network can cause deviations from the estimated link lifetime, and thus, we acknowledge the possibility of inaccurate assessment of link stability. Therefore, if any link merely satisfies the stability criteria such that, $(t_{rr'}^{hop} < \tau_{rr'}^s - T_{s,r}^p < t_{rr'}^{hop} + \delta)$ where δ indicates the allowed uncertainty in link lifetime estimation, rather than unicast through the corresponding link, we incorporate broadcast with a Time To Live (TTL) counter of 2. As a TTL of 2 is allowed, the packets can reach the end node of the corresponding link via an intermediate node. Respectively, the condition for unicast transmission can be stated as, $(t_{rr'}^{hop} + \delta < \tau_{rr'}^s - T_{s,r}^p)$.

Figure 5: Inclusion of broadcasting in a route.

For an example, in the Fig. 5, if the lifetime of link AB is under scrutiny, we will transmit the packet until A via unicast and broadcast the packet from A, expecting that via few broadcast hops, the packet will be received by the node B. The intuition here is to treat the link AB as a black haul and expect 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 exploit a limited broadcasting cycle.

- **Case 2:** Updated network information of the destination is not available

This is related to the disturbance implicated by the location **III** in Fig. 3. In this case, we find another node (agent node) that can replace the destination node in the centralized routing based on the destination node's last known location information as shown in Fig. 6.

Figure 6: Destination in a black haul.

This is executed by first calculating the distance between each node in a potentially interested area and the estimated location of the destination node, and then by finding the minimum out of all the distance values. Then,

we find a stable short path from the source node to the new agent node which replaces the destination node. If a feasible path is not found, the procedure is repeated until an agent node is found with a stable path from the source node. The procedure is summarized to the algorithm 1.

Algorithm 1 Selection of an agent node

Input: *source* - source node, *dest* - destination node
Output: *agent*, *path* ▷ agent node that replaces the destination and its path from the source

```

1: path ← null;
2: while ( path == null )
3:   minDist ← M;                                ▷ M - a large value
4:   destLoc ← FindLoc(destLoc, destVec, dueTime);
   ▷ estimates the new location of a given node based on
   its past location, velocity and time elapsed
5:   for ( ∀ node ∈ {V} - dest )
6:     nodeLoc ← FindLoc(nodeLoc, nodeVec, dueTime);
7:     dist ← FindDist(nodeLoc, destLoc);          ▷
   estimates the distance between given two nodes
8:     if ( dist < minDist ∧ node ∉ exploredAgents )
9:       agent ← node;
10:      minDist ← dist;
11:    end if
12:  end for
13:  path ← FindPath(source, agent);                ▷ finds the
   shortest stable path between given nodes via unicast
14:  if ( path == null )
15:    exploredAgents ← {exploredAgents, agent};
16:  end if
17: end while

```

Upon finding an agent node with a stable path, packets are forwarded until the agent node via unicast (or in a combination of both unicast and broadcast, if case 1 occurs). Then, a limited broadcasting cycle is used to forward the packets in the black haul network (network area with unknown network information). The agent node will start broadcasting the packets towards the estimated location of the destination, which essentially reflects a directional broadcast. The TTL count is adaptively set by the controller in the flow rules, and it is computed by dividing the estimated distance between the agent and the destination nodes by the transmission range of 802.11p given by the *2-ray ground reflection* model.

The main idea behind the above approach is to limit the broadcasting in the network by transmitting the packets to a closer node to the destination in an uncertain area. As the broadcasting starts from a closer node, a limited broadcasting cycle will incur in the network minimizing the congestion unlike in a conventional broadcasting routing protocol.

In addition to the above mechanism, we instruct the nodes in the black haul whom received the broadcasted data packets and the agent node to temporarily hold the

packets until their delay tolerance margins (ϵ_{max}^p) expire as, if the destination node comes into the transmission range of one of these nodes inside the above time period, a packet forwarding is possible. This acts as a backup mechanism since the broadcasting based approach cannot necessarily guarantee a successful transmission, especially as the area with the destination is a black haul with limited network information.

4.3. Store, Carry and Forward (SCF) routing

The SCF technique is solely used in places where a feasible path is not realized between the source and the destination, yet, location information about intermediate nodes are known. These scenarios refer to the location **I** and **II** in Fig. 3. SCF routing can yield better results in these situations as location information of nodes besides the barrier are known, and thus, an accurate estimation of SCF is possible.

The general practice in sparse network conditions is to store the packet until another node comes into its transmission range and then forward the packet to the new node expecting a route towards the destination. Few drawbacks can be found in this approach as discussed below.

The main issue is that the newly encountered node may not contain a path to the destination, and further route discovery setup can be needed. Critically, forwarding packets to a node which is far away from the destination can lead to longer routes and thus, higher delays and vulnerability on packet reception.

Vehicular nodes detect an incoming node into its transmission range by exchange of beacon packets, which occurs in periodic cycles. Therefore, node detection can be associated with a maximum delay up to the periodic time of beacon message exchange. This can also cause for a slight latency increment in SCF routing.

Alternatively, the objective here is to find the path with the least travel time including the store and forward time. In order to find it, we include the store time (t_{ij}^{scf}) of the link, if any, in the cost metric, as represented in (15). Then, the paths are found using a combination of bi-directional Dijkstra and successive shortest path algorithms similar to the unicast approach.

$$C_{ij} = C_t.(t_{ij}^{hop} + t_{ij}^{scf}) + C_v.v_{ij}^{eff} \quad (15)$$

In addition to the existing connections here, we explore possible future connections between nodes as well. However, the link travel time (transmission time + store time) should be within the delay tolerance margin of the packet. The amount of time a packet should be buffered until the ideal node comes into its transmission range is computed as follows with Fig. 7 and 8 taken as references.

The squared Euclidean distance between the nodes at a given future point in time can be stated as,

$$d_{ij}^2 = [P_{X_i} - P_{X_j} + (V_{X_i} - V_{X_j}).t]^2 + [P_{Y_i} - P_{Y_j} + (V_{Y_i} - V_{Y_j}).t]^2 \quad (16)$$

Figure 7: A general instance of two vehicular nodes in the network.

where P_{X_i}, P_{Y_i} and V_{X_i}, V_{Y_i} refer to the position and velocity components of node i , respectively, and t a general point in future time. Further, to maintain simplicity, let, $P_{X_i} - P_{X_j} = P_{X_{ij}}$, $P_{Y_i} - P_{Y_j} = P_{Y_{ij}}$ and $V_{X_i} - V_{X_j} = V_{X_{ij}}$, $V_{Y_i} - V_{Y_j} = V_{Y_{ij}}$.

Let, the time at two nodes comes into each other's transmission range be t' , and in turn d_{ij} be D_R . Then, an expression for t' can be derived as follows.

$$t'^2(V_{X_{ij}}^2 + V_{Y_{ij}}^2) + 2t'(P_{X_{ij}}V_{X_{ij}} + P_{Y_{ij}}V_{Y_{ij}}) + P_{X_{ij}}^2 + P_{Y_{ij}}^2 - D_R^2 = 0 \quad (17)$$

where D_R refers to the transmission range of a node via 802.11p. Finally, the packet store time can be evaluated as follows.

$$t_{ij}^{scf} = \begin{cases} 0, & \text{if } d_{ij} < D_R \\ t' + \lambda, & \text{if } d_{ij} > D_R \wedge t' \in \mathbb{R} \\ \bar{M}, & \text{otherwise} \end{cases} \quad (18)$$

Here, a λ time is allowed for the vehicles to cross the transmission boundary. $\bar{M}$ is a large value, and it helps to eliminate node pairs that will not be met in near future.

Unlike in conventional SCF, we do not necessarily forward the packet to the first encountered node, rather select an optimal route with the least travel time. However, it is imperatively important to assess whether all the connections exist at the times packets pass through the corresponding links.

Figure 8: A path with updated travel times for SCF.

For an example, in Fig. 8, connectivity of the link 2-3 does not exist at the route setup time, and both nodes will come into each other's transmission range at a time in future. Therefore, a store time is associated with that link, in addition to the transmission time. For a path to be feasible, the residual lifetime of all *connected* links should be higher than the total time the corresponding packets take to reach the end node of those links as modeled in (21). In this instance, the lifetime of link 3-4 should be

higher than the summation of all the travel times in preceding links and the current link. This is a vital condition in SCF routing more than in unicast routing, as packet store times are likely to dominate the residual lifetimes of links.

In this case, the expressions for the packet arrival times at intermediate nodes are slightly different from (9) and (10) due to the inclusion of packet store times. For a connected link $rr' \in E$, if $t_{gap} >> t_{wl}^{hop}$,

$$T_{s,r}^p = (p-1)t_{gap} + \sum_{\forall ij \in L_{s,r}^u} (t_{ij}^{hop} + t_{ij}^{scf})Y_{ij}^p \quad (19)$$

where $L_{s,r}^u$ indicates the set of all upstream links of node, r from the source node, s .

If $t_{gap} \approx t_{wl}^{hop}$,

$$T_{s,r}^p = \begin{cases} n_b(p-1)t_{wl}^{hop} + \sum_{\forall ij \in L_{s,r}^u} (t_{ij}^{hop} + t_{ij}^{scf})Y_{ij}^p; & n_b < 3 \\ 3(p-1)t_{wl}^{hop} + \sum_{\forall ij \in L_{s,r}^u} (t_{ij}^{hop} + t_{ij}^{scf})Y_{ij}^p; & n_b \geq 3 \end{cases} \quad (20)$$

Similarly,

$$\tau_{r,r'}^s - T_{s,r}^p \geq t_{rr'}^{hop} \quad (21)$$

The SCF routing scheme is summarized in algorithm 2.

Algorithm 2 SCF routing summarized

- 1: $pass \leftarrow false$;
 - 2: $newNetwork \leftarrow \text{UpdateCostMetric}(\text{oldNetwork}, \text{delay tolerance})$;
 - 3: **while** (! $pass$)
 - 4: $path \leftarrow \text{SuccessiveBiDirectionalSPA}(\text{newNetwork}, \text{source}, \text{dest})$; $\triangleright$ *SuccessiveBiDirectionalSPA* method finds the successive shortest paths between a given source and destination pair
 - 5: $pass \leftarrow \text{CheckFeasibility}(\text{path}, \text{packet})$; $\triangleright$ *CheckFeasibility* method assesses the feasibility of a given path under the stability and QoS constraints
 - 6: **end while**
-

The resulting flow rules are supplied to the data plane elements following a source routing based flow instantiation operation [26]. Therefore, all the path information is first forwarded to the source node via the southbound interface and subsequently to the respective nodes in the path via the data plane communication along with the data packets. In order to facilitate the new improvements, we incorporate the transmission method in the packet format as depicted in Fig. 9.

5. Performance Evaluation

In this section, we comparatively analyze the performance of the proposed tribrid routing protocol using the

Figure 9: Inclusion of next hop information in Flow Mod packet (*please note that Tx Method only require 2 bits and the characters are for the explanation purposes only).

NS-3 simulator [31]. A shortest path based SDVN routing protocol (SHORTEST), most stable path based distributed VANET routing protocol (ROMSGP [6]) and a modified store, carry and forward based distributed VANET routing protocol (SCF [18]) are chosen for the comparative study.

5.1. Simulation Configuration

We used a simplified version of the Nanyang Technological University (NTU), Singapore campus map with 15 scattered RSUs as the simulation environment as shown in Fig. 10. The simulation area covers around 1500*1500 m^2 . The road network traces are extracted from the OpenStreetMap [32] and simplified by removing the pedestrian lanes using the NETEDIT tool [33]. The vehicles are allowed to loop randomly in the road network and mobility traces are captured over a 500 sec. time duration via the SUMO software tool [33]. The total RSUs do not cover all the road network and thus, some areas create black hauls in the network. All the results except Fig. 11b are taken without the top tier of the control plane as it would be an unfair comparison given that other routing protocols do not rely on a centralized server.

Figure 10: Simulation network (NTU, Singapore campus map) [34].

Hello and beacon message interval is set to 500 ms. At each simulation run, quarter of the vehicles send randomly selected set of packets to randomly selected destinations.

The number of packets is selected based on an uniform distribution in between 1-8. The interval between set of packets is based on a Poisson distribution with a mean of 4 seconds. The upper bound latency requirement of packets is taken as 5 sec. Additional parameters and values used in the simulation are tabulated in Table 1.

Table 1: Simulation parameters.

Parameter	Value
Simulators	NS-3.25, NS-3-dev
Simulation time	400 seconds (at each run)
Transmission type	UDP
Data size	1500 bytes/packet
Wireless network	V2V and V2I = DSRC
Transmission speed	DSRC = 6 Mbps, Ethernet = 100 Mbps, Broadband = 1 Gbps
Transmission range	DSRC $\sim$ 200 m
Propagation model	Two ray ground propagation loss model
Propagation delay	RSU backbone = 100 μ s

We first analyze the improvement brought by the proposed treatments such as consideration of stability, broadcasting based routing and SCF routing in Fig. 11a. The plot labeled as SHORTEST only focuses on the shortest path, whereas STABLE graph scrutinizes the link stability/connectivity in addition to the shortest distance. The TRIBRID graph, which is the proposed routing framework of this work, includes broadcasting based distributed routing and SCF routing under uncertain/disconnected network conditions, in addition to the link stability.

As mentioned above, most of the results are taken only with local RSU level controllers, and yet, the proposed HD-SDVN architecture exploits a top tier of controllers with the southbound as LTE. The Fig. 11b depicts the results with the complete HD-SDVN architecture.

Moreover, we use the SCF success rate, which indicates the percentage of packets successfully delivered via the SCF technique (Fig. 12), as a metric to evaluate the accuracy of the proposed path selection scheme under the sparse network conditions and disturbances in network information. Finally, the key performance metrics such as the packet delivery ratio (PDR) (Fig. 13a), average end to end delay (Fig. 13b), average routing overhead, which refers to the additional packets utilized in discovering the routes per a single data packet delivery (Fig. 13c), and the average hop count, which symbolizes the average length of the selected routes (Fig. 13d), are comparatively evaluated. All the results are taken by varying the number of vehicles in the network from 20 to 100 in increments of 10.

5.2. Result Analysis

The Fig. 11a and 11b show the effects of the proposed improvements of the routing framework in stepwise. According to Fig. 11a, the shortest path approach in

SDVN displays a lower and degrading packet reception rate against increasing vehicle density. The reason for the degradation can be stated as the domination of the traffic density in a limited area which led to overwhelming communication traffic. However, each proposed improvement has upheld the PDR from the shortest path level which is used as a baseline in here. The consideration of link stability in route selection has increased the PDR by a considerable margin. Moreover, the inclusion of broadcasting under uncertain network conditions and optimized packet storing and path selection based on the residual link lifetime, which is a treatment for the disturbances in network information, have collectively yielded a slight improvement in packet reception as well. As discussed repeatedly, it is important to assess whether the residual lifetime of successive links in a selected path can carry the packet towards the destination, especially when the preceding links are associated with buffering of packets. This effect is explicitly analyzed in Fig. 12, in which the proposed SCF scheme dominates the tested existing SCF scheme by a significant margin.

The local RSU controllers do not cover the entire road network, and thus, in some areas, the vehicles do not have the capability to contact the control plane in Fig. 11a. Therefore, the packets under these circumstances failed to even discover a route. But, with the top tier control plane, these packets have better opportunities to find paths and reach the corresponding destinations as experimentally verified in Fig. 11b. Overall, a considerable PDR increment at around 9-10% is extracted in the simulated

Figure 11: PDR improvement with respect to applied treatments.

Figure 12: Effectiveness of the SCF scheme.

Figure 13: Performance comparison.

vehicular network with the complete HD-SDVN architecture.

The Fig. 13a which compares the packet reception effectiveness of the proposed routing protocol with respect to existing protocols, further verifies the accuracy of the protocol. The SCF protocol which targets sparse network conditions has immensely struggled under dense network conditions (when the number of vehicles is increased), due to the congestion created by the broadcast of data packets. Similarly, the most stable path based protocol (ROMSGP) has failed to display a competitive PDR due to the broadcasting of route discovery packets, and also exclusion of packet storing in sparse network areas. However, the proposed routing protocol scrutinizes the link stability, and acknowledges the generality in vehicular networks and thus, has managed to excel over existing routing protocols by significant margins.

The primary focus of our routing protocol is on the maximization of delivery of packets within given latency based QoSs. However, the protocol additionally thrives on minimization of the average end to end delay of packets by transmitting them via the shortest possible paths even under packet storing scenarios. The Fig. 13b provides proof of this attempt as the proposed protocol has managed to keep a competitive latency with the shortest path and most stable path protocols. When the vehicle density is low, which further empowers sparsity in the network, the proposed scheme has to depend on storing and carrying of packets, and thus, a higher latency is inevitable. However, when the vehicle density increases, the proposed model displays a competitive latency which is lower than the most stable path based approach. It is evident that the SCF and shortest path approach display the highest and lowest latency profiles due to their high dependency on storing of packets and usage of shortest paths without considering connectivity of links, respectively. The Fig. 13d agrees with the above explanation as understandably the average hops in the selected routes via the shortest path and most stable path have exhibited the lowest and highest values. As expected, the proposed scheme balances the average hop count closer to the lower bound.

In terms of routing overhead, both SDVN studies have displayed lower packet utilizations in finding routes as the

route setup generally includes a *PACKET IN* message (the packet requesting the flow rules) transmission to the controller and reception of a *FLOW MOD* message (the packet with the flow rules) only. However, the *FLOW MOD* messages in the proposed scheme include additional routing information such as transmission method, etc. and thus, a moderately higher overhead. Meanwhile, the distributed schemes rely on the broadcasting of route discovery packets and thus, higher overheads. The SCF scheme distributes copies of the data packet which are larger in size, instead of route request packets (RREP) and therefore, a significant routing overhead.

6. Conclusion

In this paper, we introduced a tribrid routing protocol for a general SDVN instance incorporating unicast, broadcast and store, carry and forward routing concepts. The routing problem is modeled following an incremental algorithm where the successive shortest paths are validated for the stability and latency based QoS for a given set of packets. Appropriate distributed routing techniques as in broadcast and SCF, are applied to the uncertainties and disturbances in the network information on a case by case basis. The proposed protocol makes use of limited broadcasting cycles in uncertain network conditions and optimal SCF in sparse network conditions. The simulation results comparatively verified the improvement extracted by the proposed routing framework.

References

- [1] F. Dressler, F. Kargl, J. Ott, O. K. Tonguz, L. Wischhof, Research challenges in intervehicular communication: lessons of the 2010 Dagstuhl Seminar, *IEEE Communications Magazine* 49 (5) (2011) 158–164.
- [2] G. G. M. Nawaz Ali, P. H. J. Chong, S. K. Samantha, E. Chan, Efficient data dissemination in cooperative multi-RSU Vehicular Ad Hoc Networks (VANETs), *Journal of Systems and Software* 117 (2016) 508–527.
- [3] C. M. Silva, B. M. Masini, G. Ferrari, I. Thibault, A Survey on Infrastructure-Based Vehicular Networks, *Mobile Information Systems* 2017 (2017) 28. doi:10.1155/2017/6123868.
- [4] J. J. Blum, A. Eskandarian, L. J. Hoffman, Challenges of intervehicle ad hoc networks, *Trans. Intell. Transport. Sys.* 5 (4) (2004) 347–351. doi:10.1109/tits.2004.838218.

- [5] J. Haerri, F. Filali, C. Bonnet, Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns, in: Proc. of 5th IFIP Mediterranean Ad-Hoc Networking Workshop (Med-Hoc-Net-2006), Lipari, Italy, Vol. 1045, 2006.
- [6] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, Y. Nemoto, A Stable Routing Protocol to Support ITS Services in VANET Networks, *IEEE Transactions on Vehicular Technology* 56 (6) (2007) 3337–3347. doi:10.1109/TVT.2007.906873.
- [7] C. Liu, Y. Shu, O. Yang, Z. Xia, R. Xia, SDR: A Stable Direction-Based Routing for Vehicular Ad Hoc Networks, *Wireless Personal Communications* 73 (3) (2013) 1289–1308. doi:10.1007/s11277-013-1278-6.
- [8] D. Abada, A. Massaq, Improving Relay Selection Scheme for Connecting VANET to Internet over IEEE 802.11p, 2015.
- [9] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC Editor, 2003.
- [10] D. A. Maltz, D. B. Johnson, The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4.
- [11] I. Ku, L. You, M. Gerla, F. Ongaro, R. L. Gomes, E. Cerqueira, Towards software-defined VANET: Architecture and services, in: 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), 2014, pp. 103–110.
- [12] M. Casado, ARCHITECTURAL SUPPORT FOR SECURITY MANAGEMENT IN ENTERPRISE NETWORKS, Ph.D. thesis, STANFORD UNIVERSITY (2007).
- [13] K. L. K. Sudheera, M. Ma, P. H. J. Chong, Link Dynamics Based Packet Routing Framework for Software Defined Vehicular Networks, in: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1–6. doi:10.1109/GLOCOM.2017.8254597.
- [14] K. S. K. Liyanage, M. Ma, P. H. J. Chong, Link Stability based Optimized Routing Framework for Software Defined Vehicular Networks, *IEEE Transactions on Vehicular Technology* (2019) 1–1doi:10.1109/TVT.2019.2895274.
- [15] K. L. K. Sudheera, M. Ma, P. H. J. Chong, Link Stability Based Hybrid Routing Protocol for Software Defined Vehicular Networks, in: 2018 IEEE International Conference on Communications (ICC), 2018, pp. 1–6. doi:10.1109/ICC.2018.8422399.
- [16] A. Alioua, S.-M. Senouci, S. Moussaoui, dsdivn: A distributed software-defined networking architecture for infrastructure-less vehicular networks, in: G. Eichler, C. Erfurth, G. Fahrnberger (Eds.), *Innovations for Community Services*, Springer International Publishing, Cham, 2017, pp. 56–67.
- [17] A. Vahdat, D. Becker, Epidemic Routing for Partially-Connected Ad Hoc Networks, Tech. rep. (2000).
- [18] Y. Ohta, T. Ohta, E. Kohno, Y. Kakuda, A Store-Carry-Forward-Based Data Transfer Scheme Using Positions and Moving Direction of Vehicles for VANETs, in: 2011 Tenth International Symposium on Autonomous Decentralized Systems, 2011, pp. 131–138. doi:10.1109/ISADS.2011.22.
- [19] K. L. K. Sudheera, M. Ma, G. G. M. N. Ali, 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. doi:10.1109/ICCS.2016.7833564.
- [20] M. Zhu, J. Cao, D. Pang, Z. He, M. Xu, SDN-Based Routing for Efficient Message Propagation in VANET, in: K. Xu, H. Zhu (Eds.), *Wireless Algorithms, Systems, and Applications*, 10th International Conference, WASA 2015, Qufu, China, August 10–12, 2015, Proceedings, Springer International Publishing, Cham, 2015, pp. 788–797.
- [21] Z. Ming, C. Zhiping, C. Jiannong, 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.
- [22] K. S. K. Liyanage, M. Ma, P. H. J. Chong, Controller placement optimization in hierarchical distributed software defined vehicular networks, *Computer Networks* 135 (2018) 226 – 239. doi:https://doi.org/10.1016/j.comnet.2018.02.022.
- [23] B. Dong, W. Wu, Z. Yang, 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), 2016, pp. 207–213. doi:10.1109/MSN.2016.041.
- [24] H. Li, M. Dong, K. Ota, Control Plane Optimization in Software-Defined Vehicular Ad Hoc Networks, *IEEE Transactions on Vehicular Technology* 65 (10) (2016) 7895–7904. doi:10.1109/TVT.2016.2563164.
- [25] F. Bai, D. D. Stancil, H. Krishnan, Toward understanding characteristics of dedicated short range communications (DSRC) from a perspective of vehicular network engineers, in: Proceedings of the sixteenth annual international conference on Mobile computing and networking, ACM, 2010, pp. 329–340.
- [26] K. S. K. L., M. Ma, 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, pp. 1–5. doi:10.1109/VTCFall.2017.8288201.
- [27] W. Su, S.-J. Lee, M. Gerla, Mobility prediction and routing in ad hoc wireless networks, *Int. J. Netw. Manag.* 11 (1) (2001) 3–30.
- [28] G. G. N. Sandamali, R. Su, Y. Zhang, Q. Li, Flight routing and scheduling with departure uncertainties in air traffic flow management, in: 2017 13th IEEE International Conference on Control Automation (ICCA), 2017, pp. 301–306. doi:10.1109/ICCA.2017.8003077.
- [29] G. Vaira, O. Kurasova, Parallel Bidirectional Dijkstra’s Shortest Path Algorithm, in: Proceedings of the 2011 Conference on Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference, DB&IS 2010, IOS Press, Amsterdam, The Netherlands, The Netherlands, 2011, pp. 422–435.
- [30] J. Y. Yen, Finding the k shortest loopless paths in a network, *management Science* 17 (11) (1971) 712–716.
- [31] G. F. Riley, T. R. Henderson, *The ns-3 Network Simulator*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 15–34. doi:10.1007/978-3-642-12331-3_2.
- [32] OpenStreetMap contributors, Planet dump retrieved from https://planet.osm.org 2017, https://www.openstreetmap.org.
- [33] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent Development and Applications of SUMO - Simulation of Urban Mobility, *International Journal On Advances in Systems and Measurements* 5 (3&4) (2012) 128–138.
- [34] Google, Google Maps (2016).

Kalupahana Liyanage Kushan Sudheera received his B.Sc. Eng. (First Class Honors) degree from University of Ruhuna, Sri Lanka in 2015. He is currently a PhD candidate at Nanyang Technological University, Singapore. He was previously a Lecturer in the Department of Electrical and Information Engineering, Faculty of Engineering, University of Ruhuna, Sri Lanka. His research interests mainly include Software Defined Networking, Vehicular Ad-hoc Networks, Internet of Things/Vehicles, Mobile/Wireless Networks, Optimization methods, Software Engineering, Embedded Systems.

Dr. Maode Ma received his Ph.D. degree in computer science from Hong Kong University of Science and Technology in 1999. Currently, Dr. Ma is an Associate Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore. He has extensive re-

search interests including network security and wireless networking. Dr. Ma has more than 380 international academic publications including more than 180 journal papers and about 200 conference papers. He serves as a Senior Editor or an Associate Editor for other 4 international academic journals. Dr. Ma is a Fellow of IET, a Senior Member of IEEE Communication Society and IEEE Education Society, and a Member of ACM.

Dr. Peter Han Joo Chong is currently a Professor and Head of Department of Electrical and Electronic Engineering at Auckland University of Technology, Auckland, New Zealand. He received the B.Eng. (with distinction) in Electrical Engineering from the Technical University of Nova Scotia, Halifax, NS, Canada, in 1993, and the M.A.Sc. and Ph.D. degrees

in Electrical and Computer Engineering from the University of British Columbia, Vancouver, BC, Canada, in 1996 and 2000, respectively. His research interests are in the areas of mobile communications systems including radio resource management, multiple access, MANETs/VANETs, multihop cellular networks and Internet of Things/Vehicles.