Controller placement optimization in hierarchical distributed software defined vehicular networks

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

2018

Kalupahana Liyanage, K. S., Ma, M., & Joo Chong, P. H. (2018). Controller placement optimization in hierarchical distributed software defined vehicular networks. *Computer Networks*, 135, 226-239. doi:10.1016/j.comnet.2018.02.022

<https://hdl.handle.net/10356/89738>

<https://doi.org/10.1016/j.comnet.2018.02.022>

© 2018 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 22:11:06 SGT

Controller Placement Optimization in Hierarchical Distributed Software Defined Vehicular Networks

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

Recently, a new paradigm has emerged, named as Software Defined Vehicular Network (SDVN) which applies the concept of Software Defined Networking (SDN) in Vehicular Ad-hoc Network (VANET), to overcome the shortcomings in vehicular networks. With the introduction of SDN, VANET has been provided with flexibility and programmability along with a performance improvement. However, the improvement comes at a cost of higher operational delay because, the controllers are placed far away from the data plane in the existing SDVN architectures. As an alternative, we have previously proposed to bring the control plane down to Road Side Unit (RSU). In this study, we further extend this work and introduce a hierarchical distributed controller architecture where the top tier of controllers are regionally distributed on the Internet and the bottom tier of controllers are placed in several selected RSUs closer to the vehicles so that the latency induced by the system becomes low. We further present a novel controller placement model for the RSU level controllers based on the p-median facility location problem with the delay and the significance of the RSU location as the factors to achieve the optimization heuristically as an integer quadratic programming problem. With the help of the simulation results, we show that our proposed controller placement model can optimize the placements of controllers with a lower latency compared to other possible controller placement methods including the existing SDVN architectures and conventional VANETs.

Keywords: Architecture, Optimization methods, Software defined networking, Software defined vehicular networks, Vehicular ad-hoc networks

1. Introduction

The transportation industry has been facing several serious issues such as increasing traffic accidents, long traffic congestions and damages to public property and human lives, etc. Continuous increasing amount of vehicles, limited space in road infrastructure and bad driving behaviors can be stated as the root causes that drive these problems [1]. However, these critical threats can be minimized by establishing a proper communication among vehicles, which is one of the fundamental objectives of the Vehicular Ad-hoc Networks (VANETs) [2].

The VANETs emerged nearly a decade ago and since then it has been getting an increasing attention both from the industry and academia due to its unique challenges. In VANET, vehicles will be equipped with On Board Units (OBUs) which generally facilitates a GPS device and a Dedicated Short Range Communication (DSRC) radio-transceivers [3]. In contrast to other mobile ad-hoc networks, VANETs exhibit several unique characteristics such

as predictable mobility patterns, large network size, frequent disconnections, and stringent delay requirements. Also, VANETs deploy unique stationary Road Side Units (RSUs) in the network with a purpose to disseminate important information such as road conditions, traffic light status, etc. to the passing vehicles [4].

But so far, VANETs have not been able to meet the desired expectations leaving a lot of challenges open such as inefficient information dissemination, security vulnerabilities, inability in coping with dynamic topology changes which result in low Packet Delivery Ratios (PDR) and high end to end delays, etc. [2, 5]. Some of these problems are very difficult to solve under current VANET architecture. Most importantly, current distributed VANET architecture cannot provide a sense of programmability to the system [6]. Therefore, it is very difficult to deploy and test new protocols and services in VANETs. A comprehensive reconfiguration in each vehicle is required for the purpose of deploying new services, prompting serious issues on the manageability and expandability of the existing VANETs.

In order to address these problems and enable programmability, researchers are focusing on incorporating Software Defined Networking (SDN) into VANETs [7, 8]. SDN has been brought forward to encourage innovation and provide programmability to traditional network sys-

*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)

tems [9]. The main concept behind SDN is the separation of the control plane and the data plane. All the control functionalities are extracted from the network devices and placed in a logically centralized controller. Thus, the network operators have access to easily implement new applications and change the network behavior according to various dynamic requirements [10].

Initial studies on SDN have mainly focused on wired networks. But now there has been an increasing attention on using SDN in wireless and ad-hoc domains [11]. Researchers have proposed to apply SDN into VANETs in different ways. Most of the Software Defined Vehicular Network (SDVN) architectures have allowed a full SDN control over both vehicles and RSUs [7, 12], while only few studies have introduced SDN to the RSU backbone adhering wired SDN concepts [13, 14]. However, there is a debate over the effectiveness of the second type of the systems.

On SDN controllers and their placements, the most of the solutions for the controller placement are employing a single logically centralized controller hosted on the Internet [7, 12]. The Fig. 1 illustrates such type of Internet based SDVN architecture. This configuration enables the controller to take better decisions based on the global view it has over the network. However, as the controller is usually far away from the vehicles, this architecture tends to produce a significant latency in its operations [15]. Especially, with a higher latency in a cellular network such as a fourth generation (4G) wireless network, [16, 17], getting flow information will take a considerable amount of time as highlighted in [18]. Thus, in case of routing, the route setup time will be very high in centralized controller based SDVN systems.

Figure 1: Existing Internet based SDVN architecture.

Furthermore, the system will generate a significant communication cost when the messages for the periodical status update and SDN control messages (*packet in*, *flow mod*) are delivered over the 4G wireless network. Although several studies have attempted to reduce the delay and the cost in this communication [18], the physical placement of the SDN controller at an Internet server blocks further delay reduction to a competitive level with the conventional VANETs.

Further, this architecture can also create problems on

the practicality and scalability of the system. Unlike other conventional networks, a VANET consists of hundreds thousands of OBUs and RSUs. So controlling them via a single controller or a cluster of centralized controllers is not feasible. It will impose a lot of stress on the control plane. Interestingly, many VANET applications such as traffic safety with collision avoidance and blind spot warning, parking assistance, road hazard warning, etc. only require the knowledge of the local or regional area. Therefore, it is not required to always place the controller on the Internet to have the entire global network view.

When applying the SDN into the vehicular domain, it is critical to consider the unique properties and requirements of VANETs. Many VANET applications such as safety (delay < 100 ms), and cooperative traffic efficiency (delay < 200 ms), are delay sensitive [19]. The above centralized architecture has provided flexibility and programmability to the VANETs by SDN features at the expense of higher delay as comparatively analyzed by [15]. The latency generated in the operation is mainly dictated by the placement of SDN controllers relative to the SDN nodes and the ways of communication. Thus, a higher consideration should be given to these factors in order to make the SDVNs a useful reality.

As a solution to some of above critical concerns, especially to reduce communication latency, in our previous study [20], we have proposed to bring the control plane down closer to vehicles. We believe that with the proposed improvement on the architecture, the SDVN can make its way to a promising futuristic architecture for VANETs. In this study, we further analyze the controller placement in the SDVNs and introduce a novel Hierarchical Distributed SDVN (HD-SDVN) architecture where the control plane is split into two layers based on certain conditions with distributed controllers at each layer. Most importantly, as suggested by [20], we bring one layer of control closer to RSU level to satisfy the stringent delay constraints of the VANETs.

Few studies have attempted to leverage the advantages of the RSUs in the SDVNs such as [21, 22]. The focus of [21] is to create a RSU cloud to support the Internet of Vehicles (IoV) applications, whereas [22] has brought a control layer down to the RSU level as a Fog cloud to support delay sensitive applications. However, none of the models has precisely analyzed to achieve the optimal placements of the controllers or the division of control functionalities at the RSU level. To answer this demand, we address the issue of the optimal placement of the SDN controllers and design a novel controller placement model for the HD-SDVN system based on the delay parameters and significance of the RSU locations. We also carry out a comparative study on the results generated by our proposed controller placement model against several other existing controller placement models, conventional VANETs and the existing SDVNs. To the best of our knowledge, the solution presented in this paper is the first piece of the research work to address the issue of the optimization of

the SDN controller placement for VANETs.

The remainder of the paper is structured as follows. Section II introduces the system model including the proposed SDVN architecture and its operation. In Section III and IV, we present the background related to controller placement problem, and then the controller placement model, respectively. Section V discusses the results and finally, section VI concludes the paper.

2. System Model

In this section, we introduce the proposed hierarchical distributed SDVN architecture with the operation approach in details. We take this system model as the basis for the controller placement optimization in next sections. It should be noted that in the context of this paper, we assume all vehicles are equipped with OBUs. Moreover, we associate the term 'message' with the control information and the term 'packet' with the payload information.

2.1. Hierarchical Distributed SDVN Architecture

As emphasized above, the controllers should be placed as close as possible to the SDN data plane elements when applying the SDN into the VANETs to reduce the latency experienced by the OBUs. In fact, the current trend in the research of SDNs is to minimize the extent of communication between the control and data planes as much as possible and also to bring the control plane closer to data plane [23]. Given that VANET has the advantage of unique fixed RSUs, we can bring the control plane down from the Internet to the RSU level as in [21]. It will reduce the time to reach controller and also allows using existing DSRC wireless technology to connect with controllers instead of using expensive cellular links.

However, controllers at each RSU will not hold an enough network view or controlling area for individual controllers to be effective. In this case, the controllers will be most likely handling single hop communication requests, which OBUs can already process by themselves via beaconing of status periodically. Further, it will induces a higher system cost as the number of controllers is too high [24]. Therefore, we propose to place some controllers at several selected RSUs giving them enough large controlling areas in the network to be effective, especially in multi-hop data transmission scenarios. Yet, we should make sure that the time to reach to the controller for OBUs should not go too high. The way to select RSUs for the controller placement will be discussed in section IV and V in depth.

Since some VANET applications such as congestion avoidance and traffic guidance require a certain regional network view which includes traffic statistics over that area, to function, we can always appoint a local RSU controller for this work and ask other local controllers to send network information to the appointed controller periodically. But this approach will most likely congest the RSU backbone. Instead, we appoint a few regional controllers

on the Internet but hosted locally, in the top tier of the control plane to handle the requests which require a regional knowledge.

The overall SDVN architecture consists of two controller tiers including top and bottom as shown in Fig. 2. In the top tier, regional controllers will be placed distributedly on the Internet and will be connected through the Internet backbone. The bottom tier accommodates local controllers in several selected RSUs connected through the RSU backbone via Ethernet. A local controller RSU in fact is a combination of a conventional RSU and a SDN controller. All the SDN requests such as *Packet In* messages are directed to the SDN controller. Data plane consists of OBUs and other non-controlling RSUs. Each OBU is equipped with both DSRC interface and 4G long term evolution (LTE) interface, which will only be used in the cases where the RSU coverage does not exist. In this way, the communication cost can be compensated in comparison to other SDVN systems with only a LTE interface.

There are several duties assigned to the local and regional controllers.

- Local Controllers - They handle the requests which only require a local knowledge such as multi-hop data transmission within few hops and lane change assistance, without the intervention of the regional controllers. Therefore, OBUs can get a quick response and do not have to use the expensive LTE links. The requests which need a broader network view will be passed to the top tier.
- Regional Controllers - These controllers handle the requests which require certain regional network information such as vehicular statistics, road conditions, and traffic statistics. These requests are either redirected by the local controllers, or received directly from the OBUs which are not under the RSU coverage. Upon estimating the responses for the incoming requests, the regional controllers will pass them to the OBUs either via the local controlling RSUs or directly via LTE depending on the type of the request such as real time, non-real time, and reachability of the OBUs whether they are existence in a RSU coverage or not. These controllers are further responsible for synchronizing the bottom tier controllers and their maintenance. In case of a local controller brake down or malfunction, the regional controller(s) should activate contingency plan to transfer the OBUs and the RSUs currently under the faulty controllers to other local controllers or take over the duties of the malfunctioned controller by itself.

There exist two main Data Ownership Models (DOM) in the literature on the research work on SDNs, *single* and *multiple* [25]. In the single DOM, data about the topology are stored in a single controller. Other controllers will

Figure 2: Proposed hierarchical distributed SDVN architecture.

access that data whenever needed by queries. However, in the multiple DOM, each controller will keep its separate data store inside and share the information between each other of the controllers. Both DOMs have their own pros and cons. For an example, the single DOM represents a single point of failure but the system is very simple. On the other hand, the multiple DOM can avoid single point of failure with each controller taking its own decisions. However, the frequent information exchange between controllers may flood the east-west SDN interface creating congestions in the control plane. In our system, we propose to use a hybrid DOM which entertains positives from both *single* and *multiple* DOMs. The bottom tier controllers will keep their own data storage as in multiple DOM and frequently update the top tier controller with the data as in the single DOM. The bottom tier controllers can make the decisions independently when they possess the information, and if not the request will be passed to the top tier. Here, the RSU level controllers do not have to exchange information between them, and thus, the RSU backbone will be less congested.

The resulting SDVN architecture resembles Orion [26] in the wired SDNs, while the physical placement of controllers, interaction between the control and data planes, and division of functionalities are unique to the vehicular networks. The overall architecture not only reduces the latency, but also adds scalability to the system. As the system does not depend on a single controller, single point of failure can also be avoided.

2.2. SDVN Operation

2.2.1. Acquiring Network Information

Conventional VANETs employ a periodical broadcasting mechanism to broadcast hello messages, which vehicles can use for basic safety such as collision avoidance. Now

with the SDVN, additionally, the OBUs at vehicles have to send status messages to the controller periodically. This information is used to build topology graphs and the frequency of status update should be high enough to get accurate decisions from the controller. Most of the existing SDVN systems, cellular links are used to update status, which generate an additional communication cost, while in our system, we take this pressure off from the OBUs and assign it to the RSUs. Basically, the RSUs will collect network information such as vehicle locations, velocities, etc. from the conventional hello message broadcasting and generate local topology maps based on their knowledge on the environment. Then, these topological information will be sent to the nearest local controller, which will combine all the information received from the RSUs and generate larger local area maps. Finally, they will be sent to the regional controllers where they generate the regional network maps. It is a sequential process which does not impose additional responsibilities on the OBUs. However, the OBUs who are out of a RSU coverage have to follow the existing status update mechanism and send the beacon messages to the regional controllers via LTE interface.

2.2.2. Proposed Operation

In a general operation, OBUs can find out their nearest local controllers via the hello message broadcasted by the RSUs as these messages additionally advertise information about nearby local controllers. For an instance, the hello message received by the yellow vehicle in the left hand side of Fig. 2 from the RSU which is connected via a red dotted line, will indicate that the yellow vehicle can reach a local controller with that particular RSU as the next hop. Thus, conventional RSUs act as the agents for the local controllers.

When vehicles are under RSU coverage, the OBUs use

Vehicle-to-RSU (V2I) and RSU-to-RSU (I2I) links to connect with local RSU controllers. The same links will also be used in the data communication and thus, this is called as **in-band** controlling. We propose to allocate a Service Channel (SCH) with a higher priority but a lower priority than the Control Channel (CCH), as immediate safety matters [27], out of six service channels for this communication.

A distance based simple handover mechanism is used when vehicles move between RSUs which are under different controllers. When vehicles move out of the RSU coverage, the OBUs on them use the dedicated LTE channel to connect with the regional controllers. As this channel is dedicated to the control-data plane communication, it is called as **out-of-band** communication. However, the costs associated with these links and the delays associated with the placement of regional controllers on the Internet impose the limitations on their usability.

In the data transmission phase, an OBU sends a *packet in* message to the nearest local controller when it receives any incoming information with an unknown destination. Then, the local RSU controller or top level controller depending on the type of request and availability of information computes the shortest path from the source to the destination using the Dijkstra algorithm. Unlike in *reactive flow instantiation* where the response is only sent to the requested node, the controller sends all the corresponding *flow mod* messages which carry the flow rules to each node¹ in the computed route including the source node, in parallel. This operational procedure is called as *proactive flow instantiation*, and it can avoid information holding at intermediate nodes until the flow rules are received and thus, less disruptive to the flow and the delay can be significantly lower compared to the reactive flow instantiation. Finally, the nodes upon receiving *flow mod* messages will update their routing tables based on the flow rules and route the packets towards the destination.

2.2.3. Fallback Mechanism

There could be situations, where both RSU coverage and cellular coverage do not exist. To handle these situations, the OBUs can be equipped with the Greedy Perimeter Stateless Routing (GPSR) protocol [28], to act as a fallback mechanism suggested by I. Ku. et al. [7]. The GPSR protocol does not need a route setup procedure but the OBUs will send the data to the next OBUs based on geographical location information. When the data is being transmitted through multiple hops, it is possible to initiate the SDN operation if the data is received by an intermediate OBU that has the connectivity with either a regional or a local controller. The Fig. 3 illustrates such a scenario with the sequence numbers representing the steps in the process.

Here, the vehicle A (OBU inside the vehicle) wants to send a packet to vehicle D. But, the vehicle A is not within

the coverage of a RSU or a base station of the cellular system. So it can use the GPSR protocol and send data to the OBU at the vehicle B via Vehicle-to-Vehicle (V2V) channel based on the geographical information (1). Now, the vehicle B has a connection with a RSU which it can use to initiate the SDN operation. Thus, the vehicle B will send a *packet in* message to the local controller requesting for route information (2). The local controller will then compute the entire path and sends flow mod messages to all the OBUs in parallel (3). Upon receiving the route information, the OBUs can update their routing tables and forward the messages towards D (4).

Figure 3: SDVN operation with fallback mechanism.

3. Controller Placement Problem

In this section, we discuss the background related to the controller placement problem in detail including factors affecting the controller placement and theoretical analysis on the delay parameters which is required to formulate the problem.

3.1. Factors Influencing in Controller Placement Selection

When placing local controllers at several selected RSUs, it is important to place them at optimal positions so that the latency experience by OBUs will be minimized. The controller placement problem depends on many factors, which has been analyzed in depth in wired SDNs [29]. The factors related to controller placement in the SDVNs are different from those in the wired SDNs due to the unique characteristics of the VANETs. Here, we discuss several important factors which will be considered in the proposed model. We assume that the SDN controllers including local and regional controllers can handle all the receiving requests without any performance limitations because the current commercial SDN controllers are capable of handling millions of client devices simultaneously [30].

3.1.1. Number of Controllers

One trivial way is to place controllers at each RSU so that the OBUs can directly contact the controllers. Since the controlling RSUs consist of two interfaces as DSRC and LTE, they are generally costly and complex. Obviously, placing the controllers at all RSUs increases the cost and the complexity of the total system. Also, high speed vehicles will only stay in the RSU coverage for a short

¹The term 'nodes' refers to both OBUs and RSUs

amount of time, which will lead to frequent handovers between the controllers, which is not desirable. At the same time, one RSU might not possess the minimum required global knowledge on the topology which is required to take accurate routing decisions. Therefore, most of the *packet in* messages might have to be diverted into the top tier controllers which again induce a higher latency. So it is desirable to have less number of controllers with larger controlling areas.

3.1.2. Latency

In contrary, if few controllers are used, several OBUs which are far away (in terms of hops) from the controlling RSUs may undergo a considerable amount of delay in route setup. Therefore, when placing the controllers, it is vital to minimize the delay experienced by the OBUs because many VANET applications are delay sensitive. It arises for a trade-off between the number of controllers and latency.

3.1.3. Geographical Location and Vehicular Parameters

The geographical location and vehicular parameters associated with a RSU are also important considerations in placing the controllers. Higher priorities have to be given to the RSUs situated near the intersections for placing controllers because usually vehicles tend to stop at intersections due to high traffic and traffic light systems, if present. Also, the speeds of vehicles are not very high at these intersections. Basically, the vehicles will use a considerable amount of time passing through the intersections with respect to other locations. So, if the controllers are placed at the intersection RSUs, the OBUs at the vehicles will have a higher direct contact with the controllers allowing them a better service in terms of delay.

It is vital to avoid placing controllers in the RSUs located at unrestricted roads with a higher mobility, because various studies have shown that high mobility affects the quality of experience (QoE) of vehicles negatively [31]. The ideal locations for placing controllers are the locations where vehicles display a lower mobility.

3.1.4. Neighboring RSU's Location Status

In addition to above facts, it is equally important to consider the distance based status of the neighboring RSUs whom could be assigned to the considered local RSU controllers in deciding the controller placement.

As an example in Fig. 4 (we only consider one-hop neighboring RSUs for the purpose of simplicity in explanation), RSU 7 is located at a 4-way intersection while RSU 10 is at a 3-way intersection. If we only consider the previous factor of significance of the location, the selection for controller would be RSU 7. However, three out of RSU 7's one-hop neighboring RSUs are in relatively unrestricted roads without intersections or traffic lights, where vehicles move fast. The vehicle OBUs within the coverage of the three RSUs which will be connected to the controller in RSU 7 (if selected) will not get a satisfactory QoE due to the high mobility of vehicles. In contrary, if

the controller is placed at RSU 10, not only the OBUs within the coverage of RSU 10 will get better QoEs, but also the OBUs within the coverage of its all three one-hop neighboring RSUs, which are located near multi-road intersections, will experience better services.

Figure 4: Simulation environment – NTU campus map [32].

3.1.5. Workload Distribution

It is also important to evenly distribute the workload among local RSU controllers, so that the individual controllers will not be too stretched. This is vital especially in highly dense road networks such as urban. The number of normal RSUs that a controlling RSU controls can be taken as a measure of its workload because higher the number of RSUs, larger the coverage area, and more number of vehicles.

3.2. Delay Analysis

The controller placement problem is associated with the delay generated over the links. Most of the time, *packet in* messages transmitted by OBUs go through both wireless channel (V2I) and wired channel (I2I) before reaching the controller. Therefore, we need to model the delay components over the wired and wireless links in order to evaluate the total delay which we should minimize in placing the controllers. Therefore, in this subsection, we model the delay parameters associated with SDVNs. For that, we follow the work of [33], where the delay associated with 802.11 networks has been analyzed in general.

Delay is an additive metric and thus, delay along a path is equal to the sum of delay components in each hop

in a multi-hop data transmission. The total delay, D , in a single hop can be represented as in (1).

$$D = t_{trans} + t_q + t_{cont} + t_{proc} + t_{prop} \quad (1)$$

where t_{trans} is the transmission delay, t_q is the queuing delay, t_{cont} is the contention delay, t_{proc} is the processing delay and t_{prop} is the propagation delay.

3.2.1. Transmission Delay

Transmission delay refers to the time taken to transmit a complete packet on to the channel and it depends on the packet size and the data rate of the interface as shown in (2).

$$t_{trans} = \frac{L \text{ (packet size)}}{R \text{ (data rate)}} \quad (2)$$

3.2.2. Queuing Delay

Each vehicle OBU and RSU maintain a single finite queue, which can be modelled by using the M/M/1/N queuing model. The packet arrival rate is assumed to be a Poisson distribution with a mean packet arrival rate, λ , and a service rate, μ . So the delay due to queuing can be represented as in (3) with N as the maximum queue size and $\rho = \lambda/\mu$.

$$t_q = \frac{\rho[1 - \rho^N - N \cdot \rho^{N-1} \cdot (1 - \rho)]}{\mu(1 - \rho)(1 - \rho^{N+1})} \quad (3)$$

3.2.3. Contention Delay

In the proposed architecture, packets go through different contention periods depending on the media access control (MAC) protocols. V2V and V2I communications use IEEE 802.11p standard in which the underlying MAC protocol is essentially Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). Between RSUs, the communication is through Ethernet and controlled by the CSMA with Collision Detection (CD). In order to model the contention delay for the two MAC protocols, a similar approach has been taken as in [33].

Let x be the number of retransmissions suffered by a given packet, p be the collision probability of the channel and c be the maximum number of attempts before the packet is dropped. The retransmission probability is given by,

$$P(x = k) = \begin{cases} p^k(1 - p), & k \leq c - 1 \\ p^k, & k = c \\ 0, & k > c \end{cases} \quad (4)$$

The expected number of retransmissions, $n = E[x]$, for a given frame is,

$$\begin{aligned} n &= \left[\sum_{k=1}^c k \cdot P(x = k) \right] \\ n &= \left[\sum_{k=1}^{c-1} k \cdot p^k \cdot (1 - p) \right] + c \cdot p^c \end{aligned} \quad (5)$$

CSMA/CA - The IEEE 802.11p standard employs Enhanced Distributed Channel Access (EDCA) as its MAC protocol with the same nature of the CSMA/CA to provide channel access. The nodes willing to transmit a packet will sense the medium first before transmission. If it is free for an Arbitration Inter-Frame Space (AIFS) [Access (AC)] time period, it will start to transmit the packet immediately. If the channel becomes busy during the AIFS, the node will defer the transmission to backoff.

In backoff, the node first selects a random backoff from the interval $[0, CW - 1[AC]]$ where the initial Contention Window (CW) equals $CW_{min}[AC]$. The CW size doubles if the subsequent transmission attempt fails until a maximum value of $CW_{max}[AC]$. Backoff value will decrease by a slot time, if the channel becomes free for a backoff slot time. When reaching a backoff value of 0, the packet will be sent immediately. After $C (= \log_2(\frac{CW_{max}}{CW_{min}} + 1))$ unsuccessful attempts, the node finally drops the packet.

In order to find the contention delay, first we calculate the average number of backoff slots as,

$$\begin{aligned} \overline{\text{backoff}} &= \sum_{k=0}^c P(x = k) \frac{\min(CW_{max}; 2^k \cdot CW_{min})}{2} \\ &= \left[\sum_{k=0}^{c-1} p^k(1 - p) \cdot \frac{2^k \cdot CW_{min}}{2} \right] + p^c \cdot \frac{CW_{max}}{2} \\ &= (1 - p) \cdot \frac{CW_{min}}{2} \cdot \frac{1 - (2p)^c}{1 - 2p} + p^c \cdot \frac{CW_{max}}{2} \end{aligned}$$

But, $CW_{max} = 2^{(c-1)} \cdot CW_{min}$,

$$\overline{\text{backoff}} = \frac{CW_{min}}{4} \cdot \left[\frac{2 - 2p - (2p)^c}{1 - 2p} \right] \quad (6)$$

There are two main reasons for the initiation of the backoff process in CSMA/CA protocol. The channel could be occupied by a transmission initiated by the same node from different queue or the channel could be occupied by a transmission initiated by a different node within its transmission range. For the simplicity of analysis, it is assumed that there is a 50% probability for the channel to become busy during AIFS[AC] period. Also, following the work of [33], the mean value of AIFS[AC] is taken as the start point of the backoff process in the derivations. So, the final expression of the contention delay incurred by the EDCA protocol can be stated as follows.

$$t_{cont} = \frac{3 * AIFS[AC]}{4} + \frac{\overline{\text{backoff}}[AC] * T_{slot}}{2} + n \cdot T_c \quad (7)$$

where T_c is the collision duration and T_{slot} is the slot time.

CSMA/CD - In the CSMA/CD protocol, a RSU who is willing to transmit a data, continuously senses the medium until it becomes idle (1-persistence). When the medium becomes idle, it immediately transmits the data and continues to sense the medium for possible collisions. If a

collision is detected, the RSU jams the network and defers the process to backoff.

In the backoff, as in the CSMA/CA scheme, a random backoff slot is selected from a $[0 : CW_{min} - 1]$ interval where $CW_{min} = 2$. After each consequent collision, contention window size doubles until a max of CW_{max} ($= 1024$). The packet will be dropped after c ($= 16$) unsuccessful attempts. The expected backoff value of CSMA/CD protocol can be modeled in a way similar to that of the CSMA/CA protocol as follows.

Let, c be the maximum number of attempts and c' ($= \log_2(\frac{CW_{max}}{CW_{min}}) + 1$) be the number of attempts until CW_{max} . The expected number of backoff value is given by,

$$\begin{aligned} \overline{\text{backoff}} &= \sum_{k=1}^c P(x=k) \frac{\min(CW_{max}; 2^{k-1} \cdot CW_{min})}{2} \\ &= \left[\sum_{k=1}^{c'-1} p^k (1-p) \cdot \frac{2^{k-1} \cdot CW_{min}}{2} \right] + \\ &\quad \left[\sum_{k=c'}^{c-1} p^k (1-p) \cdot \frac{CW_{max}}{2} \right] + p^c \cdot \frac{CW_{max}}{2} \\ &= p(1-p) \cdot \frac{CW_{min}}{2} \cdot \left[\frac{1 - (2p)^{c'-1}}{1 - 2p} \right] + \\ &\quad p^{c'} (1-p) \cdot \frac{CW_{max}}{2} \cdot \left[\frac{1 - p^{c-c'}}{1 - p} \right] + p^c \cdot \frac{CW_{max}}{2} \end{aligned}$$

But, $CW_{max} = 2^{(c'-1)} \cdot CW_{min}$,

$$\overline{\text{backoff}} = p \cdot \frac{CW_{min}}{4} \cdot \left[\frac{2 - 2p - (2p)^{c'}}{1 - 2p} \right] \quad (8)$$

$$t_{cont} = \overline{\text{backoff}}[\text{AC}] * T_{slot} + n \cdot (T_c + IFG) \quad (9)$$

where IFG represents the Inter-Frame Gap which allows the RSU to switch from the transmit mode to the receive mode.

3.2.4. Processing Delay

The theoretical processing delay depends on the number of instructions in the corresponding process and the speed of the processor. Since the OBU and SDN controller processor speeds are around several hundreds of MHz and GHz, respectively, the theoretical delay calculation leads to negligible values. Even though several studies such as [34], have numerically evaluated the centralized SDN controller latency to be in $100\mu s - 10ms$ range, it is not suitable to incorporate them in our theoretical calculations as it may lead to ambiguous results. It should be further noticed that the paper propose a distributed architecture with fair workload distribution among controllers. Thus, the processing delay in reality would be a very low, especially, given that the existing controller hardware can support millions of client devices, simultaneously [30].

3.2.5. Propagation Delay

The propagation delay depends on the distance between the RSUs and the speed of the signal propagation as in (10).

$$t_{prop.} = \frac{d \text{ (distance)}}{v \text{ (channel speed)}} \quad (10)$$

The parameter values considered in the theoretical calculations are tabulated in Table 1. It should be noted that the tabulated values such as distance and packet rate, are of a general VANET instance and does not dictate nor reflect the overall operation of the vehicular network. The distance between vehicles and the vehicle-RSU in the routing path is considered as 100 m, so as to reflect the closest controller placement in a 200 m transmission range network.

Table 1: Theoretical parameter values used in the delay analysis.

Parameter	Value
Distances	Distance between two vehicles and between vehicle and RSU in the routing path = 100m, Distance between neighboring RSUs = 400m
Propagation Speeds	Wireless channel = $3 * 10^8 ms^{-1}$, Wired channel = $2 * 10^8 ms^{-1}$
Data Rates	DSRC = 6 Mbps, Ethernet = 100 Mbps
Packet Sizes	$L_{RREQ} = 24$ bytes, $L_{RREP} = 20$ bytes [35], $L_{Packet_In} = 32$ bytes, $L_{Flow_Mod} = 56$ bytes (OpenFlow 1.3 [36]), $L_{Data} = 1000$ bytes, $L_{Beacon} = 400$ bytes
Queuing Parameters	$\lambda = 50$ packets/sec., $N = 15$, $L = 400$ bytes
CSMA/CA Parameters	$T_{slot} = 13 \mu s$, $SIFS = 32 \mu s$, $p = 0.3$, standard contention window values [33]
CSMA/CD Parameters	$T_{slot} = 51.2 \mu s$, $IFG = 9.6 \mu s$, $p = 0.3$, standard contention window values [33]

4. Proposed Controller Placement Model

In this section, we derive the proposed controller placement model as a modified p-median facility location problem which is known as a NP-hard problem [37]. One important assumption is made which is that the RSUs are located uniformly over the road network with nearly equal distances between them.

4.1. Notations

4.1.1. Network Map

Assume that there is a given network graph, $G = (V, E)$, and a distance matrix D_S , where $d_{ij} \in D_S$ represents the distance from i^{th} RSU to j^{th} RSU in terms of number of hops. The set V represents the RSUs interconnected by the edges in E . The terms r and k represent

the total number of RSUs and the number of controlling RSUs, respectively.

4.1.2. RSU Coefficients

The term, $C_i = \{C_1, C_2, \dots, C_r\}$ refers to the set of coefficient values of RSUs where $C_i \forall i \in V$ is assigned depending on the RSU's strategic location and vehicular statistics around the location as in (11).

Let R_n be the number of road segments passing through the RSU, V_{vp} be the average vehicle passing rate and V_{vs} be the average vehicle speed. Then the RSU weight coefficient is given by,

$$C_i = p * \frac{R_n * V_{vp}}{V_{vs}} \quad (11)$$

where p is a negative constant.

Incentive behind above weight assignment is to reward the selection of candidate RSUs as controllers which are located in high traffic areas (Ex: RSU 7 in Fig. 4). If the candidate RSU is located in a multi-road intersection with a relatively higher vehicle passing rate and a lower average vehicle speed, then placing a controller at this candidate RSU can be beneficial in terms of low latency because more OBUs get to directly consult the controller. The RSUs at an intersection with a higher number of roads connected to the intersection will have a high priority. For an example, a higher priority will be given to a 4-way intersection over a 3-way intersection as indicated by (11). Further, it builds up a stable southbound interface (interface between the control and data plane) with the controller for a considerable time, as vehicles move slowly in these areas.

4.1.3. Delay Parameters

D_{PI} and D_{FM} are used to indicate the time to transmit a *packet in* message and a *flow mod* message, respectively. These values are calculated based on the previous delay analysis, and can be formulated as follows.

$$D_{PI|FM} = (t_{trans} + t_q + t_{cont} + t_{proc} + t_{prop})_{PI|FM} \quad (12)$$

The notations, D_W and D_{WL} , (Fig. 5) are the two interpretations used to simplify the delay analysis for the wired and wireless transmission, respectively, as shown in below equations.

$$D_{WL} = (D_{PI} + D_{FM})_{Wireless} \quad (13)$$

$$D_W = (D_{PI} + D_{FM})_{Wired} \quad (14)$$

4.1.4. Controller Span

The span of a controller, in terms of number of consecutive neighboring RSUs outwards from the respective controller including the controlling RSU itself is represented by N_s . This value is associated with the values of D_W and D_{WL} (19, 20), and also depends on the considered delay constraint in the optimization problem.

If $N_s = 1$, the local controller will only control vehicle OBUs inside its own RSU coverage. So there will be a

SDN controller at each RSU. If $N_s = 2$, the controller will control the OBUs up to a span of one adjacent RSU as shown in Fig. 5. The maximum allowed controller span is indicated by S_{max} .

Figure 5: Positioning of normal RSUs with respect to the controlling RSU.

4.1.5. Controller Workload

y_j is used to indicate the workload of controller j and it is measured in terms of number of RSUs that the controller j governs.

4.1.6. Controller Selection

x_{ij} is used to represent the association between a normal RSU and a controlling RSU as follows.

$$x_{ij} = \begin{cases} 1, & \text{if the RSU } i \text{ is attached to the controller } j \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

The term x_{jj} is a special case of x_{ij} , and it indicates whether the j^{th} RSU is a controller or not.

$$x_{jj} = \begin{cases} 1, & \text{if point (RSU) } j \text{ is a controller} \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

We choose k, N_s, y_j and x_{ij} as our decision variables in the optimization problem.

4.2. Objective Function

Our objective of the controller placement is to minimize the number of controllers required to cover a given network map while placing them at the significantly located RSUs. Further, we encourage fair distribution of workload among local controllers as well. Based on the aforementioned requirements, the objective function can be formulated as in (17).

$$\begin{aligned} \text{minimize} \quad & C_c \cdot k + \sum_{i=1}^r \sum_{j=1}^r (N_s - d_{ij}) C_i x_{ij} + \\ & C_w \sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2 \end{aligned} \quad (17)$$

Out of the multiple objectives in (17), the major goal is to minimize the number of controllers (k) required to cover

the given network and thus, the weight coefficient related to the number of controllers, C_c , is given as a very high positive value. This helps the term, number of controllers, to make a significant impact in the objective function.

As the objective function is a minimization, the weight coefficients of RSUs (C_i) are selected as negative values and thus, the term $(N_s - d_{ij}).C_i$ is used to give higher priorities to the controlling RSUs while considering the neighboring RSUs' status connected to that particular controlling RSU with decreasing priorities. i.e. a higher priority is given to the one-hop neighbors over two-hop neighbors.

We can consider another possible controller placement method by considering all RSUs as equally significant. This can be used as a comparative measurement to evaluate the importance of prioritization of RSUs. If we place the controllers without considering RSU weight coefficients, the objective function will be changed from (17) to (18) and all other constraints remain the same.

$$\text{minimize } C_c.k + \sum_{i=1}^r \sum_{j=1}^r d_{ij}x_{ij} + C_w \sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2 \quad (18)$$

As stated in section III.A, we intent have a fair distribution of workload among all the local controllers. For that, we minimize the difference in the number of RSUs that each controller controls, with $\sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2$ term in the objective function. Yet, the workload distribution coefficient, C_w is assigned a lower positive value than C_c to reflect it's lower priority.

4.3. Constraints

As mentioned previously, there is a trade-off between the number of controllers and the latency. When the number of controllers in the system is minimized, the latency experienced by OBUs start to increase, as indicated by Fig. 5. In the proposed controller selection model, we chose to constraint this latency from increasing higher than a threshold value.

The time taken to transmit a data packet between a source and a destination in VANET undergoes two phases. The first one is the route setup phase where the source node employs either a broadcast or any other similar mechanisms to find the route to the destination. The second one is the actual data transmission phase where the discovered route is used to transmit the data. Given that two different routing protocols transmit a data between the same source and destination pair along the same path, the difference in delay occurs in the route setup.

In the optimization problem, this route setup time of the proposed SDVN system is constrained by the minimum route setup time of conventional VANETs in the multi-hop scenario. The Ad hoc On-Demand Distance Vector (AODV) [35] is selected as the reference routing protocol for the VANETs because it is a well-studied routing protocol with a similar route discovery phase to the SDVNs. The minimum route setup delay in AODV, D_{AODV} occurs when the source and the destination are apart from

each other by 2 hops in the multi-hop data transmission. In the proposed SDVN operation, the route setup delay is equal to the time to send a *packet in* message to the control plane and receive the respective *flow mod* message. When the source and destination are 2 hops away from each other, it is high likely that the local RSU controllers possess the required topology information to route packets between these 2 hops. Hence, it is assumed that the local controller-regional controller interface is not used in this case.

The minimum route setup time of the proposed system further depends on the position of the vehicle with respect to the position of the controller. Following up on this, we consider two delay aspects in the constraints.

Worst Case Delay Constraint,

$$D_{WL} + (N_s - 1).D_W \leq D_{AODV} \quad (19)$$

Average Case Delay Constraint,

$$D_{WL} + \frac{2}{2N_s - 1} \sum_{i=1}^{N_s} [(i - 1).D_W] \leq D_{AODV} \quad (20)$$

In (19), the worst case route setup delay which occurs when the vehicle is under the coverage of the furthest away RSU, is bounded by the minimum route setup delay in the AODV (Fig. 5). In the average case (20), the vehicle is considered to be with all the RSUs which are under the coverage of the same controller with equal probabilities. So the delay constraint is written by considering all the possible delay values that can occur when the vehicle is with each RSU.

By substituting the parameter values of Table 1, in theoretical delay expressions (2, 3, 7, 9, 10), we can get a numerical evaluation for D_W (13) and D_{WL} (14). With that, we can further simplify the worst case (19) and average case (20) delay constraints in the format of (21), where S_{max} is equal to **2.55** in the worst case and **3.68** in the average case.

$$N_s \leq S_{max} \quad (21)$$

We include the latency factor in the constraints rather than embedding in the objective function for the minimization and thus, the delay constraints are never to be violated in placing the controllers. Apart from that, few others constraints are considered as follows.

$$d_{ij}x_{ij} \leq N_s - 1 \quad \forall i, j \quad (22)$$

Equation (22) defines the controller span in terms of number of hops. In general sense, it restricts the selected controller RSUs governing conventional RSUs which are more than $N_s - 1$ hops away.

$$x_{ij} \leq x_{jj} \quad \forall i, j \quad (23)$$

Equation (23) indicates that a non-median (RSU), i can only be associated with a median (controller), j .

$$\sum_{j=1}^r x_{jj} = k \quad (24)$$

$$\sum_{j=1}^r x_{ij} = 1 \quad \forall i \quad (25)$$

Equation (24) accumulates the controlling RSUs and equates it to the number of controllers term, k . By the proposed model, a single non-controlling RSU and its coverage area is controlled by only one controlling RSU and it is modeled by (25).

$$y_j = \sum_{i=1}^r x_{ij} \quad \forall j \quad (26)$$

The controller workload, y_j is modeled as in (26). The equation essentially counts all the RSUs which are assigned to the controller RSU, j as its workload.

4.4. Solution Scheme

We have not used the number of controllers (k) as a hard constraint as in the conventional p-median facility location problem [37]. In fact, we have included it in the objective function as a soft constraint. Therefore, the model will always converge to a feasible solution for any given network map. The worst possible solution we can get is that all the RSUs are selected as controllers.

In terms of optimality, the objective function of the controller placement optimization model with prioritized RSUs is not convex, which can be verified by using the second derivative test with Hessian matrix. One of the Eigenvalues of the Hessian matrix is negative and thus, the function is concave. The results acquired by solving the optimization problem (17) directly via any Integer Quadratic Programming (IQP) or Mix Integer Quadratic Programming (MIQP) solver may not be optimal. In order to address this issue, we employ a heuristic approach as follows.

The concavity comes from the non-linear term, $N_s * x_{ij}$. Therefore, in order to convert the function into a convex format, we intend to fix the controller span, N_s . As our main objective in placing the controllers is to minimize the number of controllers required to cover the given network map, C_c is very high. It is clear that, to minimize the number of controllers, the controller span (N_s) should be maximized. Further, maximum controller span is bounded by the delay constraints (21). So we assign the highest possible controller span to N_s without violating the delay constraints as in (27).

$$N_s = \lfloor S_{max} \rfloor \quad (27)$$

Now, the controller span, N_s , is no longer a decision variable. Hence, the overall optimization problem is a convex Integer Quadratic Programming problem (IQP), and

can be solved optimally via a proper MIQP solver such as CPLEX. The overall controller placement selection problem for prioritized RSUs can be summarized as below.

$$\begin{aligned} \text{minimize}_{k, x_{ij}, y_j} \quad & C_c \cdot k + \sum_{i=1}^r \sum_{j=1}^r (N_s - d_{ij}) \cdot C_i \cdot x_{ij} \\ & + C_w \sum_{j=1}^r \sum_{j'=1}^r (y_j - y_{j'})^2 \end{aligned} \quad (28a)$$

$$\text{subject to} \quad d_{ij} \cdot x_{ij} \leq N_s - 1 \quad \forall i, j \quad (28b)$$

$$x_{ij} \leq x_{jj} \quad \forall i, j \quad (28c)$$

$$\sum_{j=1}^r x_{jj} = k \quad (28d)$$

$$\sum_{j=1}^r x_{ij} = 1 \quad \forall i \quad (28e)$$

$$y_j = \sum_{i=1}^r x_{ij} \quad \forall j \quad (28f)$$

$$x_{ij} = 0, 1 \quad \forall i, j \quad (28g)$$

$$k, y_j \in \mathbb{Z}^+ \quad (28h)$$

5. Results and Discussion

In this section, we apply the above controller placement model to a real world problem and compare the performance of the proposed approach with several reference methods. The simulation consists of two parts.

In the first part, we apply the proposed controller placement model with and without proper RSU coefficients as in (17) and (18), respectively, in a real world map for both average case delay constraint and worst case delay constraint. In addition, we generate another set of controller placements randomly as a comparative measurement.

In the second part, we simulate several custom traffic data sets on the map against the above mentioned controller placements and compare the performance of different systems. The custom traffic data sets have been composed based on the local knowledge that we have on the road network and also random number generators. The incentive behind using the custom traffic data sets is to test the significance of proposed candidate RSU prioritization method based on their strategic location in the local controller selection. We also make a comparative study on the latency, PDR, and packet overhead generated by the proposed HD-SDVN system with conventional vehicular networks and existing Internet based SDVN systems.

5.1. Simulation Model

We applied the above controller placement model for a simplified version of the campus map of Nanyang Technological University (NTU) in Singapore. The map is covered with a total of 19 RSUs as shown in Fig. 4.

The controller placement model for the above scenario consists of 381 decision variables and 722 inequality constraints and 39 equality constraints. We use the CPLEX Optimizer software to solve the two controller placement problems under different delay constraints. The obtained results **with** differential coefficients are shown in Fig. 6a and 7a for the worst case delay constraint and average case delay constraint, respectively. The Fig. 6b and 7b show the results for the same controller placement models **without** proper coefficients. The Fig. 6c and 7c show the **random** controller placement results using 6 and 3 controllers, respectively.

As the next step, we measure the performance of each controller placement method in terms of latency using the NS-3 simulator. In the simulation, the road network is a compressed version of the NTU campus road network, which span over nearly a $1500 * 1500 m^2$ area as shown in Fig. 4. As mentioned previously, we generate a set of custom mobility scenarios which resemble the actual traffic patterns in the NTU campus road network. The vehicle speed profiles are generated based on the Weibull random number generator with different shape and scale values depending on the road environment. Intersections deploy a lower speed profile and unrestricted straight roads entertain a higher speed profile with a maximum speed of $90kmh^{-1}$. Nodes which include both OBUs and RSUs broadcast Hello messages and update the controllers periodically at each 3 seconds. We selected these higher interval values to test the proposed system at extreme ends.

For the controller placement comparison, we measure the average route setup delay, which is the time taken to discover or/and setup the route from a source to a destination, and the average end to end delay, which is the time taken to find the route and transmit a packet from a source to a destination, for each type of controller placement against varying number of vehicles starting from 10 to 80 in an increment of 10. At each run, half of the OBUs acts as the sources and sends packets to the other half of the OBUs. The packet arrival is implemented using a Poisson distribution with a mean of 3 seconds. Each point in the graphs is an average over 50 simulation runs with different random numbers without outliers. The results for the worst case delay constraint and average case delay constraint are shown in Fig. 8 and 9, respectively.

Finally, we compare the performance of the proposed system under the worst delay constraint (WDC) against conventional VANETs, existing Internet based SDVN systems and also against reactive operation mode of SDVNs. For the VANETs, we employ conventional routing protocols of AODV and Optimized Link State Routing (OLSR) [38]. For the existing SDVNs, we recreate a simulation model for the SDVN architecture from [7] in NS-3 and connect a centrally placed SDN controller with OBUs via LTE interface. Then, we test the performance of these systems following the same mobility patterns. The performance has been measured in terms of average end to end delay and PDR by varying the number of vehicles, and

Table 2: Simulation Parameters.

Parameter	Value
Simulators	NS-3.25, NS-3-dev
Simulation time	600 seconds (at each run)
Simulation area	Simplified model of NTU-Campus Map ($1500 * 1500 m^2$)
Transmission type	UDP
Data size	1000 bytes/packet
Wireless network	V2V and V2I = DSRC, Vehicle to Base station = LTE
Transmission speed	Wireless = 6 Mbps, Wired = 100 Mbps, Internet = 100 Gbps
Packet Intervals	Hello message = 3 seconds, Status update message = 3 seconds
Transmission range	DSRC $\sim$ 200 m
Propagation model	Friis Propagation Model
Propagation delay	RSU backbone = $100 \mu s$, Internet backbone = 20 ms

the results are shown in Fig. 10a and 10b, respectively. We further test the total packet overhead of the proposed proactive based operation mode against existing reactive operation mode by varying the average vehicle speed at unrestricted roads with number of vehicles fixed to 40. The results are shown in Fig. 10c. The parameters and values used in the simulation are tabulated in Table 2.

5.2. Result Analysis

According to Fig. 8 and 9, it is clear that our proposed controller placement models with and without coefficients show considerably lower delays than those by random controller placement in each run of the simulation. In random controller placement, as we can see from Fig. 6c and 7c, several non-controlling RSUs are assigned to be the controlling RSUs which are far away in terms of hops in comparison to other controller placement schemes with the same number of controllers. Therefore, OBUs in the coverage of these non-controlling RSUs experience slightly higher delays compared to other OBUs. It has highlighted the importance of placing controllers with proper control on the delay induced in the systems.

Again, slightly lower delay in both route setup (Fig. 8a and 9a) and end to end delay (Fig. 8b and 9b) can be achieved by the proposed system with coefficients than those by the controller placement method without proper coefficients. It has proved our hypothesis that prioritizing the RSUs based on their location and statistics is important to the performance of the system in terms of latency. It is because the controllers are placed at the locations

Figure 6: Controller placement results for the worst delay constraint [32].

Figure 7: Controller placement results for the average delay constraint [32].

where most of vehicles show lower vehicular speeds and long vehicular traffic queues.

Previous researches have shown that OLSR can give a better performance in terms of delay in VANETs because it does not require any route setup time [39]. Yet, it struggles when it is applied to long source-destination pairs as the protocol needs a considerable time to converge. It is clearly shown in the results that a very low latency (Fig. 10a) and the lowest PDR (Fig. 10b) have been achieved by the OLSR scheme. On the other hand, the AODV exhibits a high successful packet delivery. However, it struggles to maintain that PDR as the number of vehicles increases (Fig. 10b). This is due to the fact that the AODV's broadcast approach in finding routes makes the network more congested when the number of vehicles increases. It has further led to the high increasing latency

by the AODV scheme (Fig. 10a).

Unlike the conventional VANETs, both SDVN systems have been able to maintain a promising PDR (Fig. 10b) against increasing number of vehicles, due to the global view they possess over the network. However, the existing SDVN systems with the control plane on the Internet (SDVN Internet) undergo a considerable amount of delay in the operation as shown in Fig. 10a due to its way of placing controllers and reactive based flow instantiation operation. In fact, these latency results for the Internet based SDVN systems can go up even higher depending on the placement of the control server in the Internet Service Provider (ISP) network. The latest statistics show that the round trip delay of the communication over the Internet between a single pair of a client and a server is around 100 ms on average [16, 17]. Thus, it is possible that a much

Figure 8: Latency results under the worst case delay constraint.

Figure 9: Latency results under the average case delay constraint.

higher actual latency could reach for the existing Internet based SDVN systems.

However, the upper bound delay requirements of active road safety applications and cooperative traffic efficiency applications in VANET are around 100 ms and 200 ms, respectively [19]. Therefore, this type of SDVN architectures prompts serious doubts on their feasibility in above areas.

In contrast, the proposed HD-SDVN system with coefficients (WDC) has managed to excel in both latency and PDR at all vehicle density levels. In terms of latency, our system which displays a delay range of 5 - 20 ms, can be applied in all the VANET applications with the excellent QoS. This is mainly due to the controlled way of placing controllers closer to vehicles. The latency is even lower than that achieved by the OLSR scheme in most cases (Fig. 10a). Further, the collective generation of global network view of the proposed HD-SDVN has managed to achieve a very high successful PDR, which is even slightly higher than the PDR by the Internet based SDVNs (Fig. 10b). We believe that the low route setup time of the HD-SDVN scheme is the root to lead to the PDR improvement. Since the communication links in VANETs are highly vulnerable in their existence, faster the accurate route information is supplied to the OBUs, higher the successful packet delivery rate.

In terms of communication overhead, the existing reactive based operation induces a very high overhead with a diminishing increment rate (Fig. 10c) when average vehicle speed increases, due to multiple consultation of the control plane in the multi-hop routing. If the control-data plane interface is composed of expensive LTE links as that in the Internet based SDVN, it further accounts for a huge communication cost as well. However, the proposed proactive based operation incurs a very low communication overhead (Fig. 10c) due to its limited consultation of the control plane, and further does not generate a significant communication cost as a result of the local placement of controllers and use of inexpensive DSRC links in the southbound interface.

6. Conclusion

In this paper, we have proposed a novel hierarchical distributed SDVN (HD-SDVN) architecture with a detailed controller placement model to optimize the placement of the controllers while preserving the stringent delay constraints in vehicular networks. The architecture splits the control plane into two levels and the top tier is placed on the Internet while the bottom tier is distributedly placed at the RSU level. The proposed controller placement model at the RSU level places local SDN controllers with the

Figure 10: Performance comparison.

consideration of several factors such as number of controllers, latency, significance of the RSU's geographical location, workload distribution and the vehicular statistics around the RSU location. By extensive numerical experiments, it is shown that our proposed controller placement method can incur a lower latency compared to other existing controller placement methods. It is also shown that our HD-SDVN architecture with the proposed controller placement method outperforms the existing Internet based SDVNs and the conventional VANETs with the lowest latency while maintaining a high PDR and a lower communication overhead.

Acknowledgment

This research is financially supported in part by the Development Of NTU/NXP-Intelligent Transport System Test-Bed Living Lab Fund, Singapore (Project Reference No. S15-1105-RF-LLF URBAN).

References

- [1] J.-P. Rodrigue, C. Comtois, B. Slack, *The geography of transport systems*, Routledge, 2013.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] M. Zhu, Z. P. Cai, M. Xu, J. N. Cao, *Software-defined vehicular networks: Opportunities and challenges*, in: *Energy Science and Applied Technology*, CRC Press, 2015, pp. 247–251.
- [7] 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.
- [8] X. Xiao, X. Kui, The Characterizes of Communication Contacts Between Vehicles and Intersections for Software-Defined Vehicular Networks, *Mobile Networks and Applications* 20 (1) (2015) 98–104.
- [9] W. Xia, Y. Wen, C. H. Foh, D. Niyato, H. Xie, A Survey on Software-Defined Networking, *IEEE Communications Surveys & Tutorials* 17 (2015) 27–51.
- [10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, OpenFlow: Enabling Innovation in Campus Networks, *SIGCOMM Comput. Commun. Rev.* 38 (2) (2008) 69–74. doi:10.1145/1355734.1355746. URL <http://doi.acm.org/10.1145/1355734.1355746>
- [11] C. Chaudet, Y. Haddad, *Wireless Software Defined Networks: Challenges and opportunities*, in: *Microwaves, Communications, Antennas and Electronics Systems (COMCAS)*, 2013 IEEE International Conference on, 2013, pp. 1–5.
- [12] H. Zongjian, C. Jiannong, L. Xuefeng, SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication, *IEEE Network Magazine Special Issue on Software Defined Wireless Networks*.
- [13] P. Duan, C. Peng, Q. Zhu, J. Shi, H. Cai, Design and analysis of software defined Vehicular Cyber Physical Systems, in: *Parallel and Distributed Systems (ICPADS)*, 2014 20th IEEE International Conference on, 2014, pp. 412–417.
- [14] L. Yu-Chun, C. Chien, S. Chakraborty, A Software Defined Network architecture for GeoBroadcast in VANETs, in: *Communications (ICC)*, 2015 IEEE International Conference on, 2015, pp. 6559–6564.
- [15] D. J. Deng, S. Y. Lien, C. C. Lin, S. C. Hung, W. B. Chen, Latency Control in Software-Defined Mobile-Edge Vehicular Networking, *IEEE Communications Magazine* 55 (8) (2017) 87–93. doi:10.1109/MCOM.2017.1601165.
- [16] I. Grigorik, *Latency: The New Web Performance Bottleneck* (2012). URL <https://www.igvita.com/2012/07/19/latency-the-new-web-performance-bottleneck/>
- [17] OpenSignal, *State of Mobile Networks: USA* (August 2016) (2016). URL <http://www.opensignal.com/reports/2016/08/state-of-the-mobile-network-usa>
- [18] 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.
- [19] Z. Hameed Mir, F. Filali, LTE and IEEE 802.11p for vehicular networking: a performance evaluation, *EURASIP Journal on Wireless Communications and Networking* 2014 (1) (2014) 1–15.
- [20] 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 Con-*

- ference on Communication Systems (ICCS), 2016, pp. 1–6. doi:10.1109/ICCS.2016.7833564.
- [21] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, Software-Defined Networking for RSU Clouds in Support of the Internet of Vehicles, *IEEE Internet of Things Journal* 2 (2) (2015) 133–144. doi:10.1109/JIOT.2014.2368356.
- [22] N. B. Truong, L. Gyu Myoung, Y. Ghamri-Doudane, Software defined networking-based vehicular Adhoc Network with Fog Computing, in: *Integrated Network Management (IM)*, 2015 IFIP/IEEE International Symposium on, 2015, pp. 1202–1207.
- [23] M. Yu, J. Rexford, M. J. Freedman, J. Wang, Scalable Flow-based Networking with DIFANE, *SIGCOMM Comput. Commun. Rev.* 40 (4) (2010) 351–362. doi:10.1145/1851275.1851224.
- [24] S. Mehar, S. M. Senouci, A. Kies, M. M. Zoulikha, An Optimized Roadside Units (RSU) placement for delay-sensitive applications in vehicular networks, in: *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, 2015, pp. 121–127. doi:10.1109/CCNC.2015.7157957.
- [25] T. Zhang, A. Bianco, P. Giaccone, The role of inter-controller traffic in SDN controllers placement, in: *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2016, pp. 87–92. doi:10.1109/NFV-SDN.2016.7919481.
- [26] Y. Fu, J. Bi, K. Gao, Z. Chen, J. Wu, B. Hao, Orion: A Hybrid Hierarchical Control Plane of Software-Defined Networking for Large-Scale Networks, in: *2014 IEEE 22nd International Conference on Network Protocols*, 2014, pp. 569–576. doi:10.1109/ICNP.2014.91.
- [27] K. A. Hafeez, A. Anpalagan, L. Zhao, Optimizing the Control Channel Interval of the DSRC for Vehicular Safety Applications, *IEEE Transactions on Vehicular Technology* 65 (5) (2016) 3377–3388. doi:10.1109/TVT.2015.2440994.
- [28] B. Karp, H.-T. Kung, GPSR: Greedy perimeter stateless routing for wireless networks, in: *Proceedings of the 6th annual international conference on Mobile computing and networking*, ACM, 2000, pp. 243–254.
- [29] B. Heller, R. Sherwood, N. McKeown, The Controller Placement Problem, in: *Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN '12*, ACM, New York, NY, USA, 2012, pp. 7–12. doi:10.1145/2342441.2342444.
- [30] CISCO, The Cisco Application Policy Infrastructure Controller - White Paper (2014).
URL <https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/unified-fabric/white-paper-c11-730021.html>
- [31] S. Singh, P. Kumari, S. Agrawal, Comparative Analysis of Various Routing Protocols in VANET, in: *2015 Fifth International Conference on Advanced Computing Communication Technologies*, 2015, pp. 315–319. doi:10.1109/ACCT.2015.113.
- [32] Google, Google Maps (2016).
- [33] C. Sarr, C. Chaudet, G. Chelius, I. G. Lassous, Bandwidth Estimation for IEEE 802.11-Based Ad Hoc Networks, *IEEE Transactions on Mobile Computing* 7 (10) (2008) 1228–1241. doi:10.1109/TMC.2008.41.
- [34] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee, DevoFlow: Scaling flow management for high-performance networks, *ACM SIGCOMM Computer Communication Review* 41 (4) (2011) 254–265.
- [35] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) Routing, RFC Editor, 2003.
- [36] O. N. Foundation, OpenFlow Switch Specification - Version 1.3.0 (2012).
URL <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.1.pdf>
- [37] A. Tamir, The k-centrum multi-facility location problem, *Discrete Applied Mathematics* 109 (3) (2001) 293 – 307. doi:https://doi.org/10.1016/S0166-218X(00)00253-5.
URL <http://www.sciencedirect.com/science/article/pii/S0166218X00002535>
- [38] T. Clausen, P. Jacquet, Optimized link state routing protocol

(OLSR), Tech. Rep. 2070-1721 (2003).

- [39] 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.

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, Mobile/ Wireless Networks, Optimization methods, Software Engineering, Embedded Systems, IC Design.

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 in Singapore. He has extensive research in-

terests including network security and wireless networking. Dr. Ma has more than 320 international academic publications including over 150 journal papers and more than 170 conference papers. He has delivered about 50 keynote speeches at various international conferences. He has served as conference chairs for over 80 international conferences.

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.