A General and Practical Framework for Realization of SDN-based Vehicular Networks

Juan V. Leon*, Oscar G. Bautista*, Abdullah Aydeger†, Suat Mercan*, Kemal Akkaya*
* Department of Electrical & Computer Engineering, Florida International University, Miami, FL 33174
Email:{jleon148, obaut004, smercan, kakkaya}@fiu.edu
† Department of Computer Science, Southern Illinois University, Carbondale, IL, 62901
Email: aydeger@cs.siu.edu

2021 IEEE International Performance, Computing, and Communications Conference (IPCCC) | 978-1-6654-4331-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/IPCCC51483.2021.9679400

***Abstract*—With the recent developments of communication technologies surrounding vehicles, we will be witnessing the simultaneous availability of multiple on-board communication interfaces on vehicles. While most of the current interfaces already include Bluetooth, WiFi, and LTE, they will be augmented further by IEEE 802.11p and the 5G interfaces, which will serve for safety, maintenance, and infotainment applications. However, dynamic management of interfaces depending on application needs will become a significant issue that can be best addressed by Software Defined Networking (SDN) technology. While SDN-based vehicular networks have been promoted previously, none of these works dealt with their practical challenges. In this paper, we propose and develop a practical framework that will realize SDN-based vehicular networks for a wide range of applications. Through this framework, we demonstrate a platoon example which demonstrates the use of SDN for quick and efficient multi-hop messaging. The route from source vehicle to destination is computed with the help of the SDN Controller to transmit the Beacon Safety Messages through Road Side Units (RSUs) at the MAC layer without relying on IP for proper platooning operations. The results show the efficiency of the SDN-based approach compared to the traditional routing approaches.**

***Index Terms*—SDN, VANETs, SDN-based VANETs, Platooning, framework, routing**

I. Introduction

The number of vehicles on the roads has risen significantly with the increasing mobility of the people, and goods [1], which led to many research on traffic congestion, safety, and other transportation problems. However, with the developments in vehicles, IoT, and communication technologies, vehicles are now moving to a new era where they are touted as the next personal smart devices that will have a tremendous impact on the lives of people through the services offered by their availability. Added to this reality is the concept of self-driving or autonomous drivers, which will come with their own capabilities to disrupt the transportation industry in the next decades [2]. Thus, we will essentially see emerging of smart vehicles as an IoT-like device that is always connected while being aware of what is happening in the surroundings through its sensors and acting with its systems whenever needed.

The challenges surrounding vehicles are not new. Over the last two decades, there have been many efforts to alleviate the problem of traffic safety due to increasing vehicles such as road information signs, radio communication for hazards for the drivers in addition to vehicle-to-vehicle communication technologies. For enabling vehicles to communicate with each other (i.e., vehicle-to-vehicle, V2V) and with infrastructure (i.e., vehicle-to-infrastructure, V2I) or in more generic term vehicles to everything (i.e., V2X), there have been tremendous standardization efforts [3]. For instance, safety applications are based on short distance broadcasting to the neighbor vehicles to prevent accidents on the road. In order to satisfy these demands, the IEEE 802.11p standard, also known as DSRC, has been developed as standard network technology for V2V communication [4]. The IEEE 802.11p is the oldest technology mainly aimed at broadcasting basic safety messages between vehicles and/or roadside units (RSUs). Despite the mandate for DSRC from the US government, the V2X standard as part of 3GPP specifications is picking up much faster [5], especially with the rolling of 5G technology. Therefore, using both IEEE 802.11p and 4G/5G/LTE would complement each other, and thus they will co-exist.

In addition to these outside connection possibilities, there is also a lot of developments regarding insider communications within a vehicle as a cyber-physical system. For instance, the sensors within a vehicle, as well as other control units, can talk to each other using Ethernet-like technologies (i.e., CAN Bus) or Bluetooth. This adds other radio interfaces that should be available on modern vehicles. All in all, with such interfaces, a vehicle can get involved in many applications that span from safety, infotainment, crowdsensing, traffic optimization, vehicular forensics, etc., which are not only about the driver but also many other stakeholders that have involvement in such applications including the law enforcement, cities, manufacturers and insurance companies.

Exploiting such a diverse set of radio access for different applications, stakeholders, and environments for vehicles brings many challenges. For instance, the question of deciding which technology to use for each traffic type arises as one of the primary challenges [6]. In addition, this decision may need to be managed by external entities for the sake of application needs rather than leaving it merely to the vehicle's needs. This is where Software Defined Networking (SDN) [7] comes into play which is a great fit to remotely control traffic passing through the vehicles' on-board unit (OBU) and RSUs. SDN divides the control and data plane of the communication while providing central access to the network switches through a Controller. SDN-based vehicular networks (VANETs) can provide a smooth transition from one radio access to another by taking into consideration different metrics such as the type of application, location of the vehicles, density of the traffic in a given radio interface, etc. SDN can also facilitate multi-hop broadcast messages in case one of the safety messages is considered essential and useful for further distances.

While SDN-based VANETs have been proposed for different use cases in the past [8], there were two significant items missing in such studies: 1) They were very specific and focused on the networking aspect of DSRC; 2) None of the ideas were implemented and tested in a realistic environment to understand the engineering challenges.

Therefore, in this paper, we first propose a novel framework that provides multiple radio access technologies for each vehicle and RSU under an SDN-based switch and controller. Our framework proposes using 802.11p for safety applications and 5G for infotainment as well as self-driving purposes. Due to its wide coverage and availability, the SDN control channel utilizes LTE as a separate interface. In addition, LTE can serve as a backup for infotainment when 5G is not available. This is a comprehensive framework that can be customized based on the resources and needs of the vehicles.

Second, we demonstrate the applicability of this framework within a practical use case, namely *platooning*. Recently, there has been a great effort in enabling platooning for a variety of vehicles, including trucks [9]. Whenever these platoons are ready to be operated on the roads in the near future, this will bring some research problems that need to be investigated. For example, even though the platoons can be arranged in advance, these platoons might get separated due to the road's dynamic characteristics, or there can be new trucks that need to join another platoon. Given that a platoon would be managed by the same company, they can employ their own SDN controller. Thus, we propose the utilization of our SDN-based framework to tackle the connectivity restoration problem in platooning through the existing BSM messages. Specifically, we exploit multi-hop BSM messaging, which would enable two different platoons to talk and merge efficiently. SDN allows us to address this connectivity restoration within the MAC layer, which is not only faster but also does not bring any additional changes to the existing DSRC stack.

We developed our framework by implementing a module within the ns-3 simulation environment [10] to test its feasibility and effectiveness, which not only supports the underlying technologies but allows access from the SDN controller. Specifically, we extended the Openflow module [11] for ns-3 simulator to integrate 802.11p, LTE and 5G in addition to the default Ethernet ports to our scenarios. To the best of our knowledge, this is the first realistic and comprehensive implementation of SDN-based vehicular networks that can serve a wide variety of needs. Under this framework implementation, we compared the proposed SDN-based platoon merge approach to other existing solutions such as AODV [12] routing protocol. The experimental results show that SDN has a significant advantage in accommodating platoon merge operations quickly and efficiently.

The rest of this paper is organized as follows: In Section II, we explain some related work and give some background information in III. Section IV introduces our proposed framework and Section V explains our specific solution to Platooning. Section VI presents our experimental evaluations of our framework. We conclude the paper in Section VII.

II. Related Work

This section summarizes works on SDN-based vehicular networks and platooning these works and explains the differences of our work from them.

A. SDN in Vehicular Networks

In [8], vehicles and RSUs are considered as SDN switches. They leverage direct status collection and estimation through trajectory prediction schemes to determine vehicle positions. However, they do not specify how they handle the communication between the SDN controller and moving SDN switches. Their experiments do not show how they can manage heterogeneous radio access as well. In [13], the authors propose a geographic routing protocol for VANET by using SDN capabilities. However, their experiments do not consider SDN and vehicular network challenges. Instead, the authors only run their routing algorithm and compare it with other routing protocols. Thus, the paper does not offer a realistic evaluation of the proposed techniques.

Finally, there are only a few works in literature where the real testbed or simulation environment has been developed and used to show their solution can support realistic evaluations beyond the theoretical discussions and models. In [14], the authors develop an SDN-based testbed for WiFi integration into the VANETs by using Raspberry Pis as Openflow switches. They demonstrate that SDN simplifies resource management in the networks with limited bandwidth available. Their limitation is that their testbed was only able to support WiFi, which ignores SDN's wide use through wired networks for long distance control. Our work is the first to bring WAN-based wireless control to VANETs which is the only realistic option to control vehicles remotely.

B. Platooning

Platoons in the VANET environment are of extreme importance since they allow vehicles traveling together in the platoon formation to achieve higher energy efficiency. Even though most of the investigations are based on heavy trucks, it is also shown that up to a 10% increase in energy savings can be achieved on commercial vehicles like cars and small buses [15]. This increase in efficiency is due to the fact that only the leading vehicle would experience the full effects of drag, and the others trailing would be able to receive up to a 24% reduction on the drag experienced depending on the intra-vehicular distance [15]. Since the reduction in drag is affected by the intra-vehicular distance, different techniques have been developed to calculate it. The authors in [16] propose the different maneuverings vehicles can adopt to create, modify and dissolve the platoon formations. Our work considers the cases where inter-platoon connectivity is lost and needs to be restored. Thus, it offers a completely different solution.

III. Prelimenaries

A. SDN

The SDN architecture consists of the network switches and the SDN Controller as shown in Fig. 1. The SDN switches handle the packet forwarding with the rules and logic assigned by the SDN Controller through Openflow protocol [17]. The SDN controller has a full view of the network of switches (i.e., RSUs in our scenario) and can calculate the routes from one host to another.

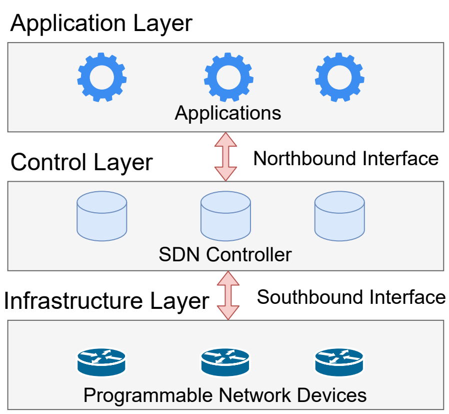


Fig. 1: SDN Infrastructure.

The primary motivation of using SDN in our platform is that SDN helps activate the switches' flow table rules in RSUs and the vehicles. The RSU switch is responsible for handling different tasks (i.e., carrying the platoon merge request from one platoon to another), and the switch in the vehicle facilitates the utilization of vehicles' communication interfaces smoothly and effectively.

B. Wave Protocol Stack (IEEE 1609)

Wireless Access in Vehicular Networks (Wave) protocol has been introduced to provide an interoperable communication interface for V2V and V2I message exchange. The protocol stack defines architecture, message format, and security mechanisms. It relies on 802.11p for physical channel access, which is a modified version of 802.11. This standardization enables a broad range of applications, including safety and traffic management. OBU and RSU are primary components designated to use this protocol stack. The range can go up to 1000 m, and the speed is around 27 Mbps. On top of the MAC layer, both IP protocol and Wave Short Message Protocol (WSMP) are supported. The latter one is only for the one-hop transmission though.

C. Platooning

With the help of efficient and secure communication capability among different brands of vehicles, it is possible to create applications where vehicles follow a leader using Cooperative Adaptive Cruise Control (CACC) technology with/without a driver. It is a group of vehicles moving together with a short distance between them, which reduces energy waste and fuel consumption while increasing safety. The system allows some operations such as *merge*, *split* and *lane change*. Relatively fixed positions of the vehicles, communication among them can be sustained. Moreover, the introduction and possibly wide adoption of self-driving cars will attract more attention to this driving scheme. The BSM and Micro-command packet structures are defined in [16], and they are shown in Fig. 2.

IV. Proposed Framework and Its Development

In this section, we introduce components of the framework and present their tasks and interactions.

In our framework, we propose an SDN Controller, vehicles with SDN switches as integrated into their OBUs, and RSUs (also with SDN switch) placed in different locations. SDN controller connects to SDN switches through 4G/LTE connections via OpenFlow protocol. Each vehicle is equipped with 802.11p, LTE, 5G (i.e., millimiterwave), and Bluetooth radio interfaces. Each RSU is capable of using 802.11p and LTE interfaces. Data connections from vehicles (i.e., either from users or other sensors) are established either through 5G or LTE, depending on the network conditions at a given time. The communication among the vehicles as well as RSUs is based on the 802.11p interface. Bluetooth interface enables connections with OBU sensors. The overview of our framework and the connections between the components are shown in Fig. 3. We explain each component below:

SDN Controller: The SDN controller calculates routes between vehicles and is connected to all vehicles and RSUs through LTE. It is assumed that the SDN Controller has direct access to the vehicles' location, which simplifies the route calculation and activation of the RSU switches.

Control Channel: The communication between SDN Controller and switches in RSU and vehicles is done through LTE based control channel. We established the LTE interface for this communication due to its wide coverage within the US and long-range communication support.

OBU: Each OBU hosts an SDN-based switch which is built upon the implementation of the OFSwitch13 module in ns-3 [18]. This implementation provides ns-3 with Openflow version 1.3 capabilities. This existing implementation allows the connection of only Ethernet (CsmaNetDevices) and virtual (VirtualNetDevices). We expanded the capabilities of this implementation to support the integration of several other options of radio access interfaces to the SDN switch for VANET applications, specifically 802.11p, LTE and mmWave.

RSU: An RSU is a wave device placed along the sides of the road or in other specific locations such as junctions or parking spaces. RSUs are equipped with the network device for DSRC communications in default, and different network devices can be integrated into them. In our specific framework, each RSU is equipped with an SDN-enabled switch, an LTE antenna for long-range communications, and the standard DSRC device. RSUs are the main devices used to provide multi-hop wireless communication in our platoon merge scenarios.

WSMP BSM

WSMP Header	Sender Address	Receiver Address	Position	Acceleration	Max Decel	Lane ID	Platoon ID	Platoon Depth

WSMP Micro-Command

WSMP Header	Sender Address	Receiver Address	Micro-command Type	Sending Platoon ID	Receiving Platoon ID	Value

Fig. 2: WSMP Packet Structure

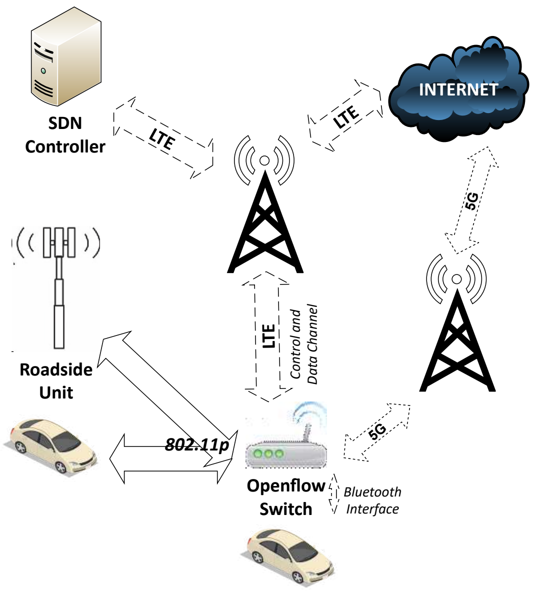


Fig. 3: Overview of SDN Framework

North-bound SDN applications: North-bound applications provide interaction between applications to operate different functionalities in the network and the SDN controller. In our framework, the North-bound application is also responsible for handling the physical location of the vehicles, which are then used to calculate routing paths. For our scenario, route calculations are done by the North-bound application, and the flow table rules are forwarded through the SDN Controller.

V. Platooning Use Case

In this section, we explain how we utilize our framework to address a specific problem (e.g., platoon merge) to showcase its feasibility.

A. Problem Definition

We consider a scenario that consists of two platoons. The first platoon (e.g., Platoon A) and the second platoon (e.g., Platoon B) are riding on the same highway with Platoon B leading with a distance more than the DSRC range as shown in Fig. 4. We assume that the platoon leaders want to merge, yet for them to successfully merge, they need to be able to communicate with each other by exchanging BSMs.

B. Leveraging SDN

Merge operation in platoon applications may not be performed using only short-range communication since the cars' initial position can be further than the DSRC transmission range. It requires involving other ways of communication (5G, LTE, e.g.) until they are within range. We assume that the cars are always in the range of an RSU, and RSUs can talk to other RSUs and SDN controller. However, RSUs lack the network topology information to correctly establish a route. To this end, we want to leverage the capabilities of SDN by adding an Openflow-enabled switch on to each Platoon leader and each RSU. Once the first BSM gets received by an RSU it will be sent to the controller as a PACKET_IN. Note that the location of the vehicles (e.g., trucks in the platoon) are known since GPS information can be obtained by SDN Controller through OpenFlow messages. The location information will be used in the route calculation for the join request from platoon A to B. The controller will respond by activating the correct RSU to rebroadcast the BSM from Platoon Leader A to Platoon Leader B. Then, the broadcast packets will be sent only to specific interfaces to route the messages from a vehicle in one Platoon to the vehicle in another platoon. This allows us to deliver BSM regardless of DSRC effective transmission range by forwarding the packet between RSUs through the route that the SDN Controller application calculates.

C. Merge Request

Initially two platoons are not aware of each other and they need to discover their presence to initiate the merge operation. Platoon leader B will broadcast its platoon ID to all RSU's in the vicinity to let Platoon leader A know its presence. RSUs are in charge of distributing the Platoon ID (Based on a Bloom filter) to other CACC-enabled vehicles. Since we consider the case where the lead vehicle in Platoon A wants to merge with Platoon B, Platoon A leader needs to send BSM to platoon leader B. BSM messages contains mobility details (position, velocity, acceleration, heading, steering wheel angle, yaw rate, etc.) which will later be used by the SDN to determine which RSUs will get updates on their flow tables. Since the first RSU does not know how to handle the BSM message, it will contact the SDN controller, which will compute the route by considering the information in the BSM message and activate designated RSUs by updating their flow tables.

D. Routing at MAC layer

The main novelty in our approach is to pull the routing function to layer 2 without relying on IP addresses. This is critical because we would like to keep RSUs as is so that they continue serve at the data link layer to handle DSRC messages. Involving them in distributed route computation will

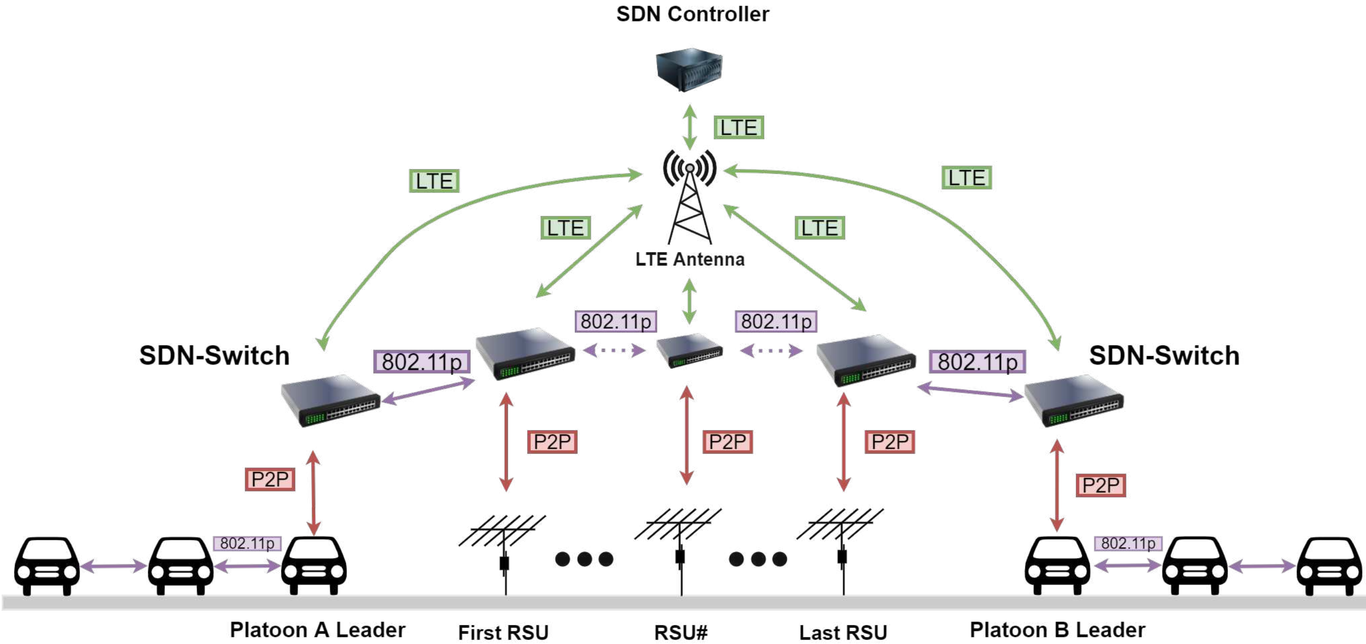


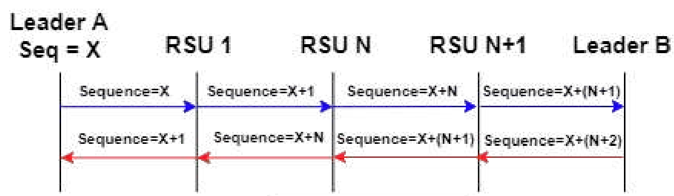
Fig. 4: Platoon Communication through RSUs

require relying on IP addresses which is not only costly but also bringing additional overhead for processing. Instead, we opt to rely on the SDN controller which can work with MAC addresses. In this way, by installing rules based on RSUs' MAC addresses, we are able to create a multi-hop connection from one vehicle to another. Note that the use of LTE here is for only control messages which are rare in the process. We refrain to use LTE for data communications since it will be costly and hence not available to everyone.

E. Loop Prevention

There is a problem that occurs where the packets are being forwarded back and forth by the RSUs, which causes the packets to be stuck in a loop forever. In order to avoid such loops, a *sequence number* field is added to the BSM packet. However, this new field will not be a standard Openflow Extendable Match (OXM) field. Thus, a modification to the SDN switch is performed to add the capability of accessing the sequence number within the packet. Adding this new Extendable Match field allows the controller to establish rules based on this number. Then switches will inspect the packet and drop it if its sequence number is higher than expected.

To elaborate on this issue, we use an example as shown in Fig. 5 where we can observe the specific behaviour given for each Platoon Leader and RSU. For Platoon Leader A we set the original sequence value to a random value of 'X'. As a can observe that at each hop the sequence number will be incremented. As a result, the sequence number at each node will be equivalent to the random value 'X' plus an offset equivalent to the relative position of the node. In Fig. 5 the blue arrows signify correct transmissions and the red arrows signify incorrect transmissions that cause loops. With this setup it is easy to observe that the only incorrect transmissions received



	Rx	Tx
Leader A	Seq = X+1	Seq = X
RSU 1	X+2	X+1
RSU N	X+(N+1)	X+N
RSU N+1	X+(N+2)	X+(N+1)

Fig. 5: Sequence Number at each RSU

at each node will always have a sequence number equal to random value 'X' plus the N^{th} relative position of the next node. By observing the incorrect transmissions, it is easy to formulate a rule at each node to avoid loops by simply dropping any packet whose sequence number is equivalent to the nodes N^{th} position plus one unit.

VI. Experimental Evaluations

To provide a proof-of-concept of our framework and show how it would work in a real platooning application, we implemented it on ns-3 [10] and conducted several experiments as detailed below.

A. Experimental Setup

Each vehicle in the platoons is 15m long and separated by a constant time gap distance of 1-3 m in order to take advantage of aerodynamics. This type of platoon is defined as a tightly-coupled platoon. We assume they will be working under CACC platoon management protocol as defined in [16]. In our experimental evaluations, we change the number of RSUs from 2 to 10 to measure various metrics under different network environments. To simulate the described scenario, we utilized NS-3.29 base distribution. From the base ns-3 distribution, we utilize the WAVE and LTE modules. The OFSWITCH13 module was used [11] to simulate the SDN controller and switches.

B. Experimental Metrics

We use the following metrics to measure the performance of proposed method.

- *Average Packet Delay:* It measures the time for a packet to arrive at the destination (leading platoon leader) vehicle from the source that it is generated. The delay is a crucial metric as the CACC messages older than 100ms are considered obsolete.
- *Overhead:* The utilization of broadcast packets among RSUs will create overhead on the network. The number of additional messages will be used to quantify this metric.
- *Setup Delay:* The first merge request message is routed to the SDN controller to find out the platoon's location and calculate routes between two platoons. The time spent on this process where the flow tables are initialized is referred as Setup Delay.
- *Packet Delivery Ratio (PDR):* It is defined as the ratio of received packets at the destination to the total number of packets sent at the source. We start measuring PDR as soon as the first BSM is sent.

C. Benchmark

We consider AODV (Ad hoc on-demand distance vector) [12], to compare with the proposed approach. AODV is a routing protocol that can be used to route packets within wireless environments, such as in VANETs. AODV protocol is an on-demand protocol because it does not maintain routes or focus on the exchange of routing information. Instead, its routes are created when they are needed, avoiding the extra resource needed to store routing tables.

D. Experiment Results

In this section, we present the results collected from ns-3 simulations for the metrics defined in different network environment setups.

1) Average Packet Delay:

As our first metric, we evaluated average packet delay. To this end, we varied the number of RSUs deployed. As shown in Fig. 6, we observe that the average delay is increasing almost linearly as the number of RSUs increases. The average delay is around 4 ms for ten hops, which is sufficient for the type of application we target (i.e., platoon merging). We also notice that the average delay for SDN is slightly lower than for AODV, the reason is due to the packets are being processed at the MAC layer vs. network layer with AODV. MAC layer communications eliminates additional processing at each node and save us time.

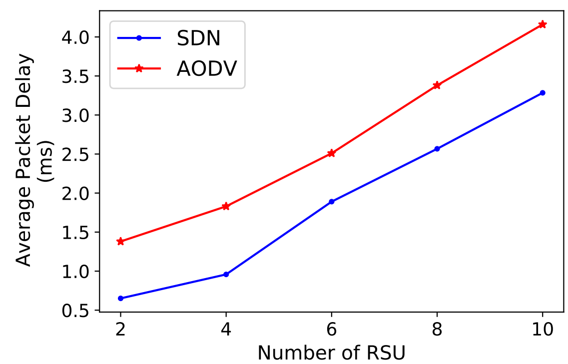


Fig. 6: Average Packet Delay

2) Network Overhead:

In the next experiment, we compared the network overhead in two different approaches. We compute the route in the SDN controller as a north-bound application and update the flow table in the RSUs, as discussed earlier. Thus, the number of packets increases keeps almost stable as shown in Fig. 7 which shows the scalability of our approach. On the other hand, AODV causes a much higher number of packets exchanged among the RSUs to calculate routes between vehicles and eventually transmit packets. Note that for AODV, the network overhead will grow significantly and thus eventually may congest the network to block/delay data transmissions.

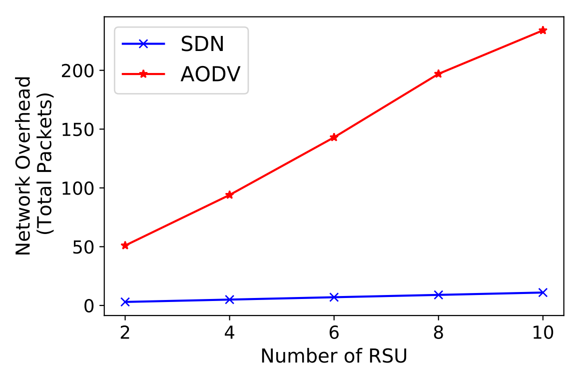


Fig. 7: Network Overhead

3) Setup Delay:

As the next experiment, we investigate the setup delay where we strive to find how long it takes for the first merge packet to be routed successfully. The comparisons of two approaches with different numbers of RSUs are displayed in Fig. 8. We specifically calculate the initial time needed to establish the route between two platoons. Since AODV requires peer-to-peer messaging among the nodes, the time increases at a faster pace with more RSUs placed in the simulated network. However, it is almost constant with SDN setup as all the RSUs are communicated at once and simultaneously. The only major delay with SDN setup is related to the software-based delay, which is negligible in our scenario.

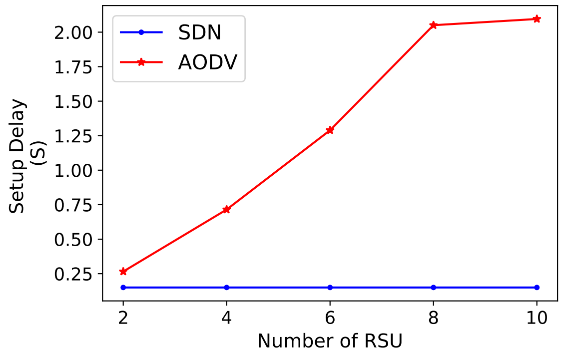


Fig. 8: Setup Delay

4) Packet Delivery Ratio:

Finally, we assessed the PDR for both approaches. As shown in Fig. 9, PDR for AODV is decreasing significantly with a higher number of RSUs when compared to our approach. This is because AODV requires more time to establish a route, and that causes the first few BSM packets to be dropped before the route is initialized. When there are more RSUs, it takes more time to establish the routes due to increased congestion and also increased geographical span of the area. This means more BSM messages will fail to reach the destination, reducing the overall PDR significantly. However, this is not the case with our SDN-based approach since the network overhead is much less, and there are very few extra packets transmitted within the network via the 802.11p interfaces.

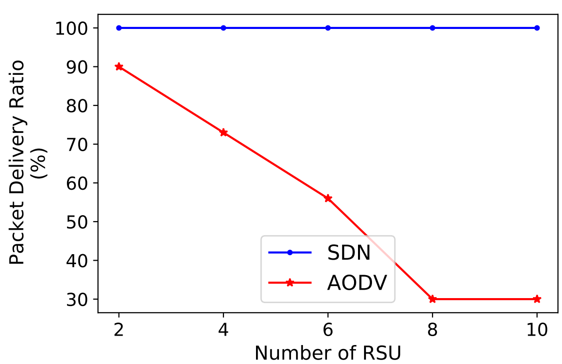


Fig. 9: Packet Delivery Ratio

VII. Conclusion

This paper introduces an SDN-based framework to manage multiple radio access technologies (i.e., LTE, 5G, and IEEE 802.11p) on vehicles and RSUs depending on the context. We developed a practical testing environment that can accommodate a comprehensive set of scenarios. We then chose and demonstrated platoon merge application within our framework by proposing a multi-hop merging capability through the use of BSM messages of vehicles without restoring to any network layer implementation. We then compared it with another alternative approach, namely AODV. Our experimental results from ns-3 implementation show that the SDN-based VANET is superior in terms of packet overhead and delay.

VIII. Acknowledgment

This work is supported in part by US National Science Foundation under the Award Numbers CNS-1816112 and SFS-1663051.

References

[1] P. Papadimitratos, A. De La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE communications magazine*, vol. 47, no. 11, pp. 84–95, 2009.

[2] S. R. Narla, "The evolution of connected vehicle technology: From smart drivers to smart cars to... self-driving cars," *Ite Journal*, vol. 83, no. 7, pp. 22–26, 2013.

[3] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: Architecture and challenges," *IEEE Network*, vol. 30, no. 4, pp. 72–80, 2016.

[4] D. Jiang and L. Delgrossi, "Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments," in *VTC Spring 2008-IEEE Vehicular Technology Conference*. IEEE, 2008, pp. 2036–2040.

[5] X. Wang, S. Mao, and M. X. Gong, "An overview of 3gpp cellular vehicle-to-everything standards," *GetMobile: Mobile Computing and Communications*, vol. 21, no. 3, pp. 19–25, 2017.

[6] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 2016.

[7] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.

[8] Z. He, J. Cao, and X. Liu, "Sdvn: Enabling rapid network innovation for heterogeneous vehicular communication," *IEEE network*, vol. 30, no. 4, pp. 10–15, 2016.

[9] "Autonomous trucks — self-driving trucks — driverless trucks," https://www.tusimple.com/, [Online; accessed 2021-7-12].

[10] "ns-3 — a discrete-event network simulator for internet systems," https://www.nsnam.org/, [Online; accessed 2020-6-4].

[11] L. J. Chaves, I. C. Garcia, and E. R. M. Madeira, "Ofswitch13: Enhancing ns-3 with openflow 1.3 support," in *Proceedings of the Workshop on Ns-3*, ser. WNS3 '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 33–40. [Online]. Available: https://doi.org/10.1145/2915371.2915381

[12] I. D. Chakeres and E. M. Belding-Royer, "Aodv routing protocol implementation design," in *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.* IEEE, 2004, pp. 698–703.

[13] X. Ji, H. Yu, G. Fan, and W. Fu, "Sdgr: An sdn-based geographic routing protocol for vanet," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData).* IEEE, 2016, pp. 276–281.

[14] G. Secinti, B. Canberk, T. Q. Duong, and L. Shu, "Software defined architecture for vanet: A testbed implementation with wireless access management," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 135–141, 2017.

[15] S. T. Kaluva, A. Pathak, and A. Ongel, "Aerodynamic drag analysis of autonomous electric vehicle platoons," *Energies*, vol. 13, p. 4028, 08 2020.

[16] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by vanet," *Vehicular Communications*, vol. 2, no. 2, pp. 110–123, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209615000145

[17] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[18] L. J. Chaves, V. M. Eichemberger, and I. C. Garcia, "Ofswitch13 module for ns-3." [Online]. Available: http://www.lrc.ic.unicamp.br/ofswitch13/