Analysing the Performance of SDN/OpenFlow Controllers in VANET

Abhijeet R Patil, Ravindra D. Patil, P M Mahajan, Kanchan S Bhagat

Abstract—One of the key enablers of the evolving 5G technology are Vehicular Ad hoc Networks (VANETs) and Software Defined Networking (SDN). The development of next generation intelligent vehicular networks includes integration of SDN in Vehicular Ad hoc Networks (VANETs). Researchers have focused on the development of the integrated technology including architecture and benefits of SDN based VANET services. In this paper we have simulated the Vehicular Ad hoc Networks (VANETs) using Mininet-wifi. The VANET is incorporated with SDN controller. The performance is evaluated by varying the speed of vehicles. We have evaluated the performance in two ways: performance evaluation of V2V communication in same RSU and performance evaluation of V2V communication from different RSU. We have evaluated the proposed system for POX and RYU SDN OpenFlow controllers. As demonstrated in the results the RYU controller of SDN performs better as compared to POX controller in terms of average delay and throughput.

Keywords—VANET, SDN, POX, RYU, Mininet-wifi

I. INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) have been one of the major are of research since few years. VANETS are of great interest due to many reasons including [1]

- Ability of improving vehicle road safety
- Enhancement of traffic and travelling efficiency
- Passengers and drivers are omfortable and feel convenient .

Intelligent Transport Services (ITS) is the best example which includes integration of information technology for improving quality of service and safety during transport. VANETs is one of the ITS system which offers several advantages to users.

- Providing real time information to users related to road situation. It includes identification of congestion on particular road due to traffic accidents.
- Providing alternative for the road in case of accidents on congestion.
- Hep users in avoiding accidents

As there is tremendous growth in the use of mobile devices and internet, two types of communications in demand are

- Vehicle-to-vehicle (V2V) and
- Vehicle-to-Infrastructure (V2I) communications

Lot of applications including safety as well as security can be provisioned by VANETs.

Revised Manuscript Received on May 21, 2020.

Abhijeet R Patil, PG Student, Department of E&TC J T Mahajan College of Engineering, Faizpur Jalgaon, India

Dr. P M Mahajan, J T Mahajan College of Engineering, Faizpur Jalgaon, India

Dr. Kanchan S Bhagat, Department of E&TC *J T Mahajan College of Engineering, Faizpur*. Jalgaon, India

Dr. Ravindra D. Patil, Principal J T Mahajan College of Engineering, Faizpur Jalgaon, India

Now a days, VANETS are implemented for providing variety of services and protocols. But still VANETS are facing major development challenges including [2]

- Flow traffic is unpredictable for multi path topology
- Network is not utilised properly during multi path topology
- Need of open and flexible VANETS for allowing researchers to test the solutions and improving the network resource management and applications.

In order to solve these issues we have integrated Software Defined Networks (SDN) in VANETs.[3] SDN is one of the emerging technologies which is able to control the network in flexible way. Open Flow protocol is used for communication. SDN is different from traditional networks due to separation of control plane and forwarding plane. The controller is responsible for taking decisions about forwarding or dropping of the packet while switch is responsible for only physical forwarding. The ability of the SDN controller to take decisions about forwarding the packets without knowing the underlying hardware details as well as flexibility of the SDN have attracted researchers to use SDN along with VANETs. SDN can satisfy the requirements of VANETs due to its flexibility and open source programmability. The network management is simple with SDN and new V2V and V2I services can be easily implemented.

Most of research on SDN/OpenFlow is on the wired networks. Application of SDN in wireless domain has been also a new research domain. However, it is necessary to understand and extend the application of SDN/OpenFlow in VANETs. This paper deals with the integration of SDN into VANETs.

- By integrating SDN into VANETs, we decouple the control and data plane in VANETs.
- This results in the centralisation of network intelligence.
- Abstraction of the underlying network infrastructure form the applications

Hence integration of SDN into VANETS will result in highly adaptive, flexible, programmable, and scalable VANETs environments. During this experimentation we present a simulated scenario for demonstrating the benefits of integration of SDN and VANETs. SDN VANETs offers benefits in terms of forwarding data in multipath topology.

The rest of the paper is organised as follows:

Section II explains the basic information of VANETs, SDN and preliminaries. Section III describes the architecture of SDN based VANETs and also lists the applications and challenges of integrating SDN into VANETs. Section IV explains the simulation and performance evaluation and Section V concludes the paper.

II. BACKGROUND

A. VANET

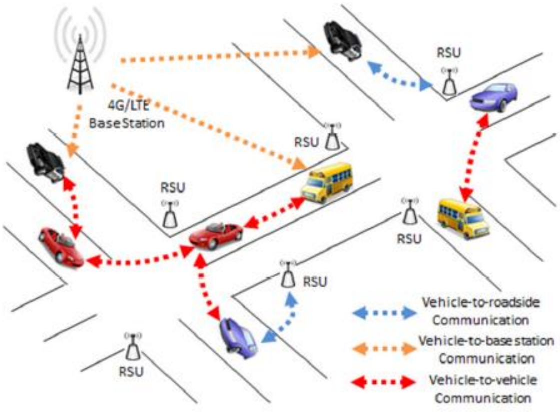

Figure 1: VANET Architecture

In VANETs communication between vehicles occurs using two ways Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications. Vehicle to Vehicle (V2V) communication occurs through vehicle On Board Units (OBU). MANTEs allow the wireless communication between OBU. In Vehicle to Infrastructure mode (V2I), vehicle communicate through Road Side Units (RSU).[4]

VANETs applications are divided into two categories, safety, and nonsafety applications. Safety applications deals with sending safety related messages. For example messages to vehicles on road for avoiding accidents. Traffic jam signal, emergency vehicle warnings, and road construction reports comes under vehicle safety messages. These safety application need low latency but high reliability. Nonsafety applications of VANET deals with improving the efficiency and comfortness in driving experience. There are two categories of nonsafety applications, traffic management and infotainment. The nonsafety applications used for improvement of traffic flow and reduction of congestion on road comes under Traffic management applications. Aim of informative applications is to provide passengers with the internet for entertaining them as well as for providing information. Examples of informative applications includes data storage, video calling and streaming. The challenge in VANET is to disseminate the critical information like accident quickly. It I very difficult to assure for the time and reliability of the critical information under dynamic vehicle network. If these critical messaged fails to deliver in time accurately, then this may lead to damage the nearby vehicles.

A large amount of traffic data is generated by the smart vehicles including video and sensor data. Traditional VANET architectures are centralised. They were unable to stream this data. Hence additional servers are required for handling of this large massive instantaneous data. One of the solution might be VANETs using cloud computing. But it was difficult to upload and download the data from the cloud. It consumes energy and time. The increasing traffic does not support the cloud environment in terms of location awareness, mobility support, and latency.

B. Software Defined Networks (SDN)

SDN is differentiated from traditional networks by the separation of control plane and data plane. The control plane is responsible for controlling several devices. SDN is dynamic, easy to manage, less costly and is able to provide with the high bandwidth. It is possible to directly program the network control due to the separation of network plane and forwarding functions. OpenFlow is one of the most widely used protocol standard for building SDN applications.[5]

SDN architecture is shown in figure 2. In SDN the network control is directly programmable due to separation from forwarding plane. The centralized controller is responsible for taking all decisions. It appears a single logical switch to programmers.

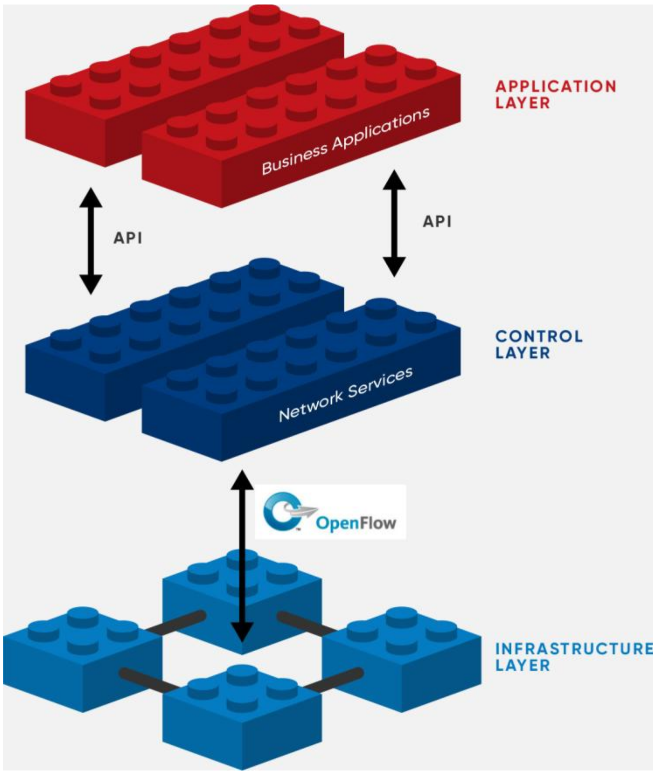

Figure 2: SDN Architecture

It becomes easy for network managers to configure and manage the network resources securely. The SDN implementations are in open standards. The instructions are specific to controller and not vendor or devices specific. Hence SDN offers simplified network design and operations.

C. POX SDN Controller

One of the most widely used and reliable open source python based SDN controller is POX. The rapid development and deployment capability of POX makes it more popular than NOX. Different characteristics of POX controller are listed below

- It is an open source controller written in python.
- Different sample applications are provided with the controller which can be modified for path selection and topology discovery.
- It is easy to deploy POX controller anywhere
- It works on Linux, Mac OS, and Windows.

- One of the most important features of POX is topology discovery.
- Same GUI and Visualization tools as that of NOX.
- The performance of POX applications is better than the performance of NOX applications written in python.

Figure the architecture of POX controller is shown in figure 3.[6]

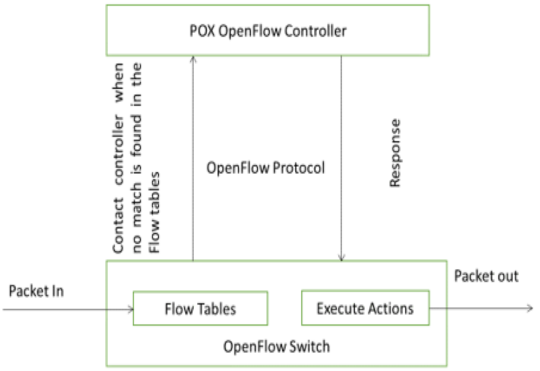

Figure 3: Architecture of POX

Pox architecture is simple. The communication in controller and network devices occur using OpenFow protocol. The controller is responsible for giving instructions to switches. Switches acts like a forwarding device. The switch contacts to the controller when it is ON. Flow tables are initially empty. Each switch maintains flow table. When the packet arrives, switch send the packet IN message to the controller. Flow entry will be inserted in the flow table of the switch by the controller. The controller will inform the switch about the packets handling. The switch will divide the flow entry into three parts, rule, action and counters. The switch only follow the flow entry. The switch does not disturb the controller again and again for packet forwarding. The packet will be dropped if the flow entry mismatches with the controller.

POX requires less memory space for operating. But as compared to other SDN controllers, POX is having low throughput.

D. RYU SDN Controller

RYU is the OpenFlow based SDN controller. RYU architecture is shown in figure 4.

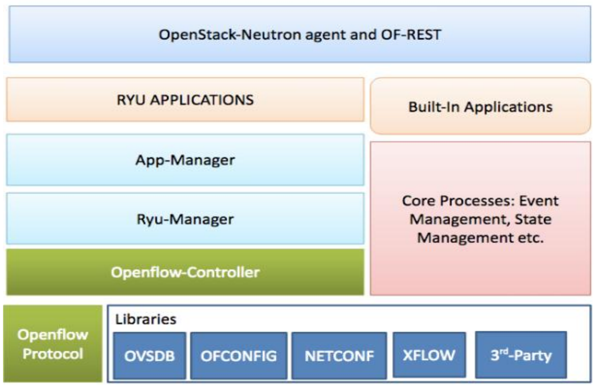

Figure 4: RYU Controller Architecture

RYU is a component-based software defined networking framework. RYU is provided with the well defined API. Network developers use it for creating different network management and control applications. Different protocols supported by RYU are OpenFlow, Netconf, OF-config. etc. Different network devices in RYU can be managed using these protocols. RYU support different OpenFlow extensions including 1.0, 1.2, 1.3, 1.4, 1.5 and Nicira Extensions. RYU is the python based open source controller. [7]

III. SOFTWARE DEFINED VANETS

This section explains the architecture Software-Defined VANET. The aim of the section is to describe the basic functionality of SDN VANET along with its benefits and challenges.

A. Architecture of SDN VANET

We have integrated the SDN in VANET. The architecture of SDN VANET incorporates following components:[8]

- SDN Controller: It is the central brain of the SDN VANET system. The controller is responsible for controlling the behavior of the entire network.
- SDN Wireless Node: The wireless nodes includes the vehicles receiving the messages from the controller.
- SDN RSU: The Road Side Units are the stationary data plane elements. They are controlled by the controller.

The architecture proposed here incorporates the SDN in VANET. In this architecture we used 10 vehicles with 4 RSU. 11 roads are used for communication. We have evaluated the performance of RYU and POX SDN controllers. SDN functionality is extended for operating in wireless mode of VANETs. The basic architecture of Software Defined VANETs is depicted on figure 5.

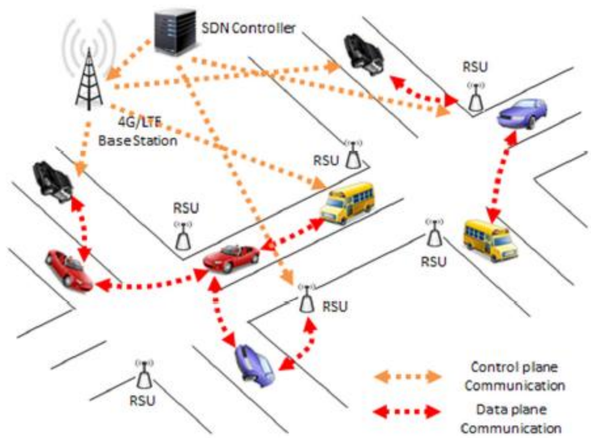

Figure 5. Software-Defined VANET Communications

B. Benefits of SDN VANETs

SDN enabled VANETs offers several benefits including:[9]

- Ability to configure the network rapidly
- Efficient resource utilization
- Low latency
- Security
- Improved user experience

C. SDN VANETs Challenges

During the deployment of SDN VANETs in large scale, the state of the art SDVNs face following challenges:[10]

- Dynamic network topology
- Security
- Detection of Malicious RSU behavior
- Attacks on security requirements: Hijacking of session, identity revealing, location tracking, denial of service.
- Availability of resources
- Scalability

IV. PERFORMANCE EVALUATION AND SIMULATION

This section explains the simulation environment used to carry out this experiment. We have implemented the SDN VANET architecture using Python. Mininet-wifi is used as an emulator. We considered the network topology consisting of

- 4 RSUs
- 10 vehicles
- 11 roads
- Varying the minimum vehicle speed from 0 to 10 and maximum vehicle speed from 5 to 55

We have evaluated the performance of the network for two controllers namely POX and RYU. VANET sample application provided by Mininet-wifi is modified for inclusion of SDN controllers.

Different simulation scenarios are done using 10 units of vehicles and 4 RSUs. For each scenario the performance is measured using delay and throughput. We have evaluated the performance in 2 scenario.

- V2V communication in same RSU
- V2V communication in different RSU

Figure 6 shows the designed topology of SDN VANET using Mininet-wifi.

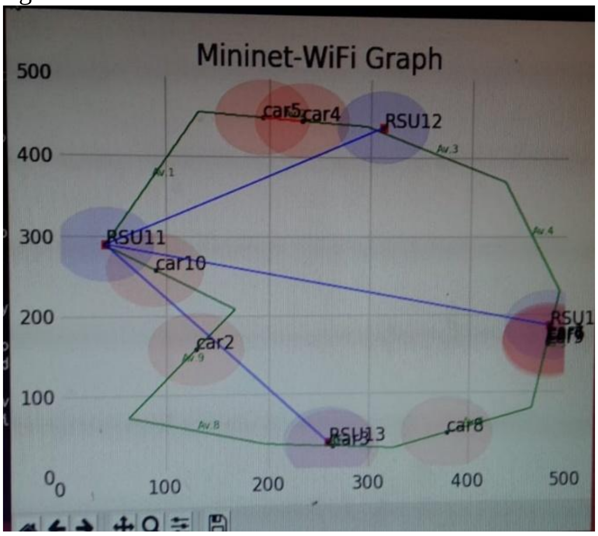

Figure 6: Mininet-Wifi Topology

A. Performance Metrics

The performance outcomes of SDN Vanets is measured by considering vehicles at different speed. The performance metrics used in this work include:

1. Throughput (TP): the ratio between total number of data packet size and the total simulation time. We have used iperf for measuring the throughput between two vehicles.

2. Packet delay time (E2E): the period from the sending time to the received time of certain packet. One may realize that the performance is better when E2E has a small value. We have used the ping command for measuring the delay. 10 Packets are sent between two vehicles.

B. Performance Evaluation Using Delay

Delay is measured using ping. We have increased the average vehicle speed from 2.5 to 32.5. Table 1 depicts the delay performance of POX and RYU controllers for V2V communication from the same RSU.

TABLE I. DELAY FOR V2V COMMUNICATION IN THE SAME SAME RSU

Min Vehicle Speed	*Max Vehicle Speed*	*Average Vehicle Speed*	*Delay RYU Controller*	*Delay POX Controller*
0	5	2.5	3.673	**2.82**
1	10	5.5	**1.977**	2.069
2	15	8.5	**2.78**	2.851
3	20	11.5	**1.683**	2.947
4	25	14.5	**2.666**	5.418
5	30	17.5	3.198	**2.39**
6	35	20.5	**1.527**	2.489
7	40	23.5	2.384	**2.259**
8	45	26.5	**2.314**	2.349
9	50	29.5	3.288	**1.546**
10	55	32.5	**2.669**	3.718
Average			**2.559909**	2.805091

As depicted in table I the average delay for SDN VANET using RYU controller for V2V communication in same RSU is less as compared to POX controller.

Table II shows the performance evaluation using delay for V2V communication in different RSUs.

TABLE II. DELAY FOR V2V COMMUNICATION IN DIFFERENT RSU

Min Vehicle Speed	*Max Vehicle Speed*	*Average Vehicle Speed*	*Delay RYU Controller*	*Delay POX Controller*
0	5	2.5	**4.0363**	4.681
1	10	5.5	5.604	**3.817**
2	15	8.5	**3.938**	5.272
3	20	11.5	4.597	**4.487**
4	25	14.5	**3.284**	4.744
5	30	17.5	**2.079**	2.796
6	35	20.5	5.295	**3.143**
7	40	23.5	**3.338**	4.077
8	45	26.5	**2.189**	3.557
9	50	29.5	2.857	**2.25**
10	55	32.5	**1.861**	4.404
Average			**3.552573**	3.929818

As depicted in Table II the delay for V2V communication in different RSU is less for RYU controller as compared to POX controller.

C.Performance Evaluation Using Throughput

Iperf is used for measuring the throughput. We have increased the average vehicle speed from 2.5 to 32.5. Table

III depicts the throughput performance of POX and RYU controllers for V2V communication from the same RSU.

TABLE III. THOUGHPUT FOR V2V COMMUNICATION IN THE SAME SAME RSU

Min Vehicle Speed	*Max Vehicle Speed*	*Average Vehicle Speed*	*Throughput RYU Controller*	*Throughput POX Controller*
0	5	2.5	28	25.2
1	10	5.5	30.5	29.6
2	15	8.5	27.6	29.9
3	20	11.5	29.6	29.8
4	25	14.5	29.2	24.5
5	30	17.5	29.2	24.8
6	35	20.5	31.5	26.6
7	40	23.5	19.1	28.2
8	45	26.5	31.9	27.2
9	50	29.5	30.5	25.6
10	55	32.5	25	28.2
Average			**28.37273**	27.23636

As depicted in table III the average throughput for SDN VANET using RYU controller for V2V communication in same RSU is more as compared to POX controller.
Table IV shows the performance evaluation using throughput for V2V communication in different RSUs.

TABLE IV. THROUGHPUT FOR V2V COMMUNICATION IN DIFFERENT RSU

Min Vehicle Speed	*Max Vehicle Speed*	*Average Vehicle Speed*	*Throughput RYU Controller*	*Throughput POX Controller*
0	5	2.5	4.23	6
1	10	5.5	5.83	4.48
2	15	8.5	8.05	7.9
3	20	11.5	7.64	5.79
4	25	14.5	29.4	7.52
5	30	17.5	8.45	20.4
6	35	20.5	15.1	7.25
7	40	23.5	18.2	11.1
8	45	26.5	28.7	8.75
9	50	29.5	11.5	11.3
10	55	32.5	17.4	10
Average			**14.04545**	9.135455

As depicted in Table IV the throughput for V2V communication in different RSU is more for RYU controller as compared to POX controller.
Figure 7 depicts the throughput performance of the network for V2V communication using same RSU vs V2V communication in different RSU.

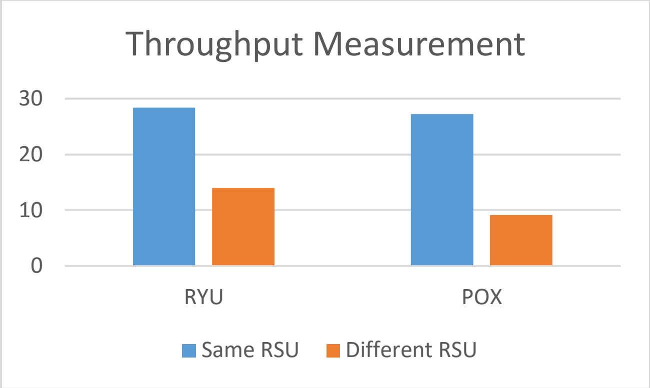

Figure 7 Throughput measurement

Figure 8 depicts the delay performance measure for V2V communication using same RSU vs V2V communication in different RSU.

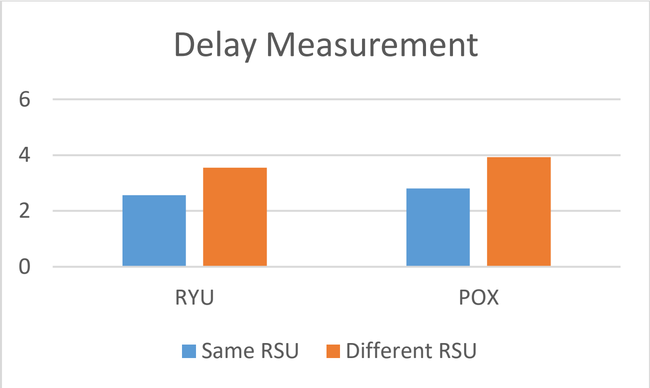

Figure 8. Delay Measurement

As depicted in figure 7 and figure 8 the performance of V2V communication in same RSU is better than the performance of V2V communication in different RSU.

V. CONCLUSION

In this research we have experimented with SDN based VANET architecture. We have evaluated the performance for POX and RYU controller. Mininet-wifi is used as an emulator with python programming language. Different performance parameters used are delay and throughput. The performance is evaluated in two scenarios, V2V communication in same RSU and V2V communication in different RSU. The vehicle speed is increased from 2.5 to 32.5. The V2V communication in same RSU occurs more precisely as compared to V2V communication in different RSU. The SDN VANET simulation comprises of 4 RSU and 10 vehicles with 11 roads. We have evaluated the performance of POX and RYU controller for SDN VANETS. The RYU controller performs better as compared with POX controller in terms of delay and throughput. As a future work we plan to design the V2V communication protocols considering no availability of RSU coverage.

ACKNOWLEDGMENT

The authors would like to thanks the management and authorities of J T Mahajan College of Engineering, Fazipur for providing the infrastructure to carry out this research.

REFERENCES

1. Ramon Fontes, Claudia Campolo, Christian Rothenberg.,"From Theory to Experimental Evaluation: Resource Management in Software-Define Vehicular Networks", Digital Object Identifier 10.1109/ACCESS.,January 2017
2. H.D. Moura, G.V.C. Bessa, M.A.M. Vieira, D.F. Macedo, "Ethanol: Software defined networking for 802.11 wireless networks", IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015, pp. 388–396
3. Neha Verma and Rakesh Kumar, "A Method for Improving Data Delivery Efficiency in Vehicular Adhoc Networks" International Journal of Advanced Science and Technology Vol. 44, July, 2012
4. Wenyang Guan, Xinghin Wang, Ye Jin, "Channel Congestion Performance Analysis for DSRC Vehicle Ad hoc Network", 10th International Conference on Communications and Networking in China (ChinaCom), 2015.
5. Chan, M.-C., Chen, C., Huang, J.-X., Kuo, T., Yen, L.-H., & Tseng, " OpenNet: A simulator for software-defined wireless local area network", Wireless Communications and Networking Conference (WCNC), 2014 IEEE pp. 3332-3336.
6. Lin D, Kang J, Squicciarini A, Wu Y, Gurung S, Tonguz, "Mozo: a moving zone based routing protocol using pure v2v communication in vanets", IEEE Trans Mob Comput 16(5):1357–1370
7. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review 38(2)(2008) 69-74.
8. W. Chen, C.-T. Lea, S. He, and Z. XuanYuan, "Opportunistic routing and scheduling for wireless networks," IEEE Trans. Wireless Commun., vol. 16, no. 1, pp. 320–330, Jan. 2017
9. G. Shang, P. Zhe, X. Bin, H. Aiqun, and R. Kui, "Flooddefender: Protecting data and control plane resources under sdn-aimed dos attacks," in IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1–9.
10. B. Dong,W.Wu, Z. Yang, and J. Li, "Software Defined Networking Based On-Demand Routing Protocol in Vehicle Ad Hoc Networks", 12th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2016, vol. 25, pp. 11–18, IEEE, Hefei, China, December 2016.