Towards Software-Defined VANET: Architecture and Services

Ian Ku, You Lu, Mario Gerla
Department of Computer Science
University of California Los Angeles
Los Angeles, USA
{ianku, youlu, gerla} @cs.ucla.edu

Francesco Ongaro
Department of Computer Science and Engineering
University of Bologna
Bologna, Italy
francesco.ongaro@studio.unibo.it

Rafael L. Gomes
Institute of Computing
University of Campinas (UNICAMP)
Campinas, Brazil
rafaellgom@ic.unicamp.br

Eduardo Cerqueira
Institute of Technology
Federal University of Pará
Belém, Brazil
cerqueira@ufpa.br

Abstract*— Vehicular Ad Hoc Networks (VANETs) have in recent years been viewed as one of the enabling technologies to provide a wide variety of services, such as vehicle road safety, enhanced traffic and travel efficiency, and convenience and comfort for passengers and drivers. However, current VANET architectures lack in flexibility and make the deployment of services/protocols in large-scale a hard task. In this paper, we demonstrate how Software-Defined Networking (SDN), an emerging network paradigm, can be used to provide the flexibility and programmability to networks and introduces new services and features to today's VANETs. We take the concept of SDN, which has mainly been designed for wired infrastructures, especially in the data center space, and propose SDN-based VANET architecture and its operational mode to adapt SDN to VANET environments. We also discuss benefits of a Software-Defined VANET and the services that can be provided. We demonstrate in simulation the feasibility of a Software-Defined VANET by comparing SDN-based routing with traditional MANET/VANET routing protocols. We also show in simulation fallback mechanisms that must be provided to apply the SDN concept into mobile wireless scenarios, and demonstrate one of the possible services that can be provided by a Software-Defined VANET.

Keywords-Software-Defined Networking; VANET; wireless networks;

I. INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) have received a lot of interest in recent years. It is an active area of research because of its potential to improve vehicle road safety, enhance traffic and travel efficiency, and provide convenience and comfort for passengers and drivers. One example would be the Intelligent Transport Services (ITS) [1].

With the expected growth in mobile devices and mobile traffic, Vehicle-to-vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are expected to become more in demand and will continue to grow [2]. VANETs can be used to provide a wide range of services, including both safety and non-safety related applications. Examples include vehicular safety traffic management services, surveillance services, and mobile vehicular cloud services.

VANETs are now a reality and support a variety of new services and protocols. However, there are still challenges in the deployment of VANETs' applications, such as unbalanced flow traffic among multi-path topology, and inefficient network utilization. Therefore, open and flexible vehicular architectures are key requirements to allow experimenters to test their solutions in productive environments, as well as to improve the management of network resources, applications, and users.

To address these challenges, we look at Software-Defined Networking (SDN) [3]. Nowadays, SDN has emerged as a flexible way to control the network in a systematic way, with OpenFlow [4, 5] as the most commonly used SDN protocol. The flexibility of SDN makes it an attractive approach that can be used to satisfy the requirements of VANET scenarios. Applying SDN principles to VANETs will bring the programmability and flexibility that is lacking in today's distributed wireless substrate, while simplifying network management and enabling new V2V and V2I services.

While most SDN/OpenFlow based advances has been focused on wired networks, especially for datacenters, increase attention has been made on incorporating SDN/OpenFlow into the wireless domain [6]. While its scope has been primarily focused on carrier backbones and access networks, the usage of SDN in other wireless scenarios has been gaining attention from both academia and industry. OpenRoads [7] envisions that users will move between wireless infrastructures. CloudMAC [8] proposes virtualized access points. The Wireless & Mobile Working Group (WMWG) [9] in ONF focuses on wireless backhaul, cellular Evolved Packet Core (EPC), and unified access and management across enterprise wireless and fixed networks (e.g., campus Wi-Fi).

Other works on wireless SDN include OpenFlow in wireless mesh environments [10], OpenFlow in smartphone as an application [11], OpenFlow in wireless sensor networks [12], SDN in heterogeneous networked environments [13], and SDN for handover management in heterogeneous networks [14]. However, it is necessary to understand and extend the usage of SDN/OpenFlow in VANETs.

In this paper, we focus on applying SDN into VANETs. In specific, we look at the architecture, operations, and benefits of Software-Defined VANET services and new functionalities to support them. By decoupling the control and data planes in VANETs, network intelligence and state can be logically centralized and the underlying network infrastructure is abstracted from the applications. Thus, it will be possible to have highly adaptive, flexible, programmable, and scalable VANETs environments. A use-case on routing is presented to demonstrate the benefits of integrated SDN VANETs architectures in forwarding data in multipath scenarios.

The reminder of the paper is structured as follows. Section II provides background information on VANET and SDN/OpenFlow. Section III describes the architecture and operations of a Software-Defined VANET. Benefits and services for Software-Defined VANETs are presented in Section IV. Section V presents simulation evaluation and Section VI concludes the paper.

II. Background and Terminologies

In this section we describe some background information and terminologies on VANET and SDN/OpenFlow used through the paper.

A. VANET

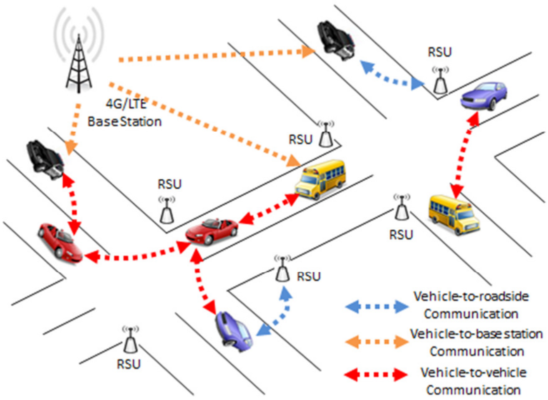

Figure 1. VANET Component and Communications

In a typical VANET, Vehicles communicate with each other through V2V communication in Ad hoc fashion, and V2I communication through road-side-units (RSU) and mobile broadband (e.g. 4G/LTE). Fig. 1 shows the components and communications with a typical VANET.

Traditional VANET services include vehicle and road safety services, traffic efficiency and management services, and infotainment services. Vehicle and road safety services are those that target the decrease of traffic accidents and loss of life to vehicle occupants. Traffic efficiency and management services aim to improve traffic flow, traffic coordination, and to provide local and map information. Infotainment services aims to provide information and entertainment such as multimedia data transfer and global Internet access.

B. SDN/OpenFlow

The core concept of SDN [3] is the separation between the control plane and the data plane. The latter is used for the data forwarding while the other is exploited for the network traffic control.

OpenFlow [4, 5] is the most commonly used protocol for communication between the SDN control plane and data plane. Fig. 2 shows a high level concept of SDN. In this paper, OpenFlow is used as base and is integrated to VANET wireless environments.

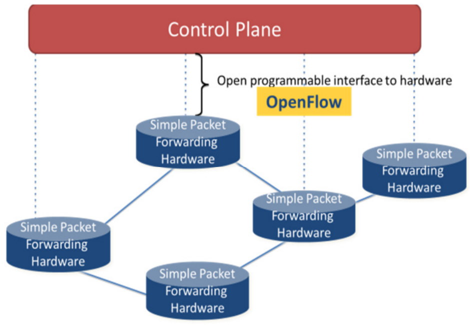

Figure 2. Software Defined Networking Concept

Fig. 3 shows an OpenFlow network comprising the two main components: the OpenFlow controller and several OpenFlow-enabled switches that communicate using a secure channel.

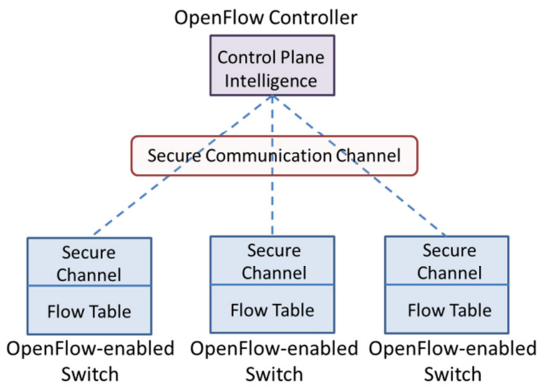

Figure 3. OpenFlow components

The controller is a software program element that is used to modify the content of flow tables, which are located in each OpenFlow-enabled switch. Each flow table contains a list of flow entries, which consist of Match Fields, Priority, Counters, Instructions, Timeouts, Cookie, and actions associated. When a packet comes to the switch there is a lookup into the matching field of each entry. If there is a match, the packet will be processed according to the actions of the matching flow entry. On the other hand, if a packet does not match, a table-miss occurs. In this case, different actions can be taken as specified in the table-miss flow entry, for instance, the OpenFlow-enabled switch can encapsulate the packet and send it to the controller through the secure channel or directly drop the packet.

OpenFlow controller controls the behavior of the network by sending *flowmods* packets to modify the content of flow tables. The controller has two ways to add rules in the switch: (I) proactively, where the controller takes the initiative and adds rules before packet arrival into the network; or (II) reactively, where the controller reacts because of an in the network, such as a previously unrecognized packet. An example of such an operation would be an Access Point (AP)/switch that sends a data packet to the controller (by encapsulating the packet in an OpenFlow control packet called *packet-in*) because it does not know how to deal with it. Then, the controller sends the *flowmods* packet to the APs/switches with instructions.

One notable feature in an OpenFlow network is that once a specific traffic flow matches the flow table, the OpenFlow-enabled switch "knows" how to treat this flow and does not need further interactions with the OpenFlow controller. While this allows switches to forward traffic efficiently, issues arise when the flow table rules are no longer consistent with the network condition. In other words, if network conditions such as topology have changed, until the controller inserts/updates the flow table entry, an OpenFlow-enabled switch will use the old (and potentially incorrect) rule. Section V shows more details about this issue.

In mobile networks, the introduction of SDN and OpenFlow will enable the programing of base stations' wireless data plane and enhance the functionalities of the core networks. Thus, it will improve the management of resources and mobile devices and create a great opportunity for new services and control functions. In dynamic wireless mobile environments, such as VANETs, the use of SDN can reduce interference; improve the usage of channels and wireless resources, as well as the routing of data in multi-hop and multi-path scenarios. We describe the benefits of Software-Defined VANETs later in Section IV.

III. SOFTWARE DEFINED VANET

In this section, we describe the architecture of a Software-Defined VANET and its operations. The goal is to describe how VANETs take advantage of SDN concepts and functionalities to improve resource utilization, select best routes, and facilitate network programmability.

A. Architecture Overview

To enable a SDN based VANET system, our architecture incorporates the following SDN components:

- **SDN controller**: The logical central intelligence of the SDN based VANET system. The SDN controller controls the network behavior of the entire system.
- **SDN wireless node**: The data plane elements that are controllable by the SDN controller. They are the vehicles that receive control message from the SDN controller to perform actions.
- **SDN RSU**: Stationary data plane elements that are controllable by the SDN controller. They are the infrastructure RSUs that are deployed along road segments.

Our proposed architecture extends SDN to operate in mobile wireless VANET scenarios. In our architecture, we choose to use different wireless technologies for controlling and forwarding planes as expected in future VANET systems: Long range wireless connection i.e., LTE/Wimax for control plane, and high bandwidth wireless connection i.e., Wi-Fi for data plane. The practical reason is that in VANETs not all nodes are easily reachable from the Infrastructure via RSUs. Fig. 4 shows the communication between components in our Software-Defined VANET.

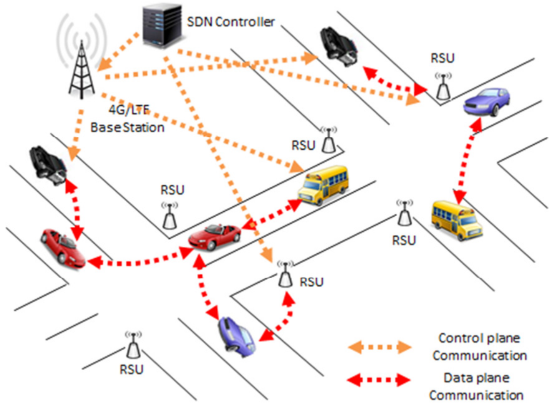

Figure 4. Software-Defined VANET Communications

Fig. 5 shows the internal components of a SDN wireless node. It contains all the functionality of an OpenFlow-enabled switch in traditional OpenFlow networks, plus additional intelligence to enable different modes of operation in VANET environments. The number of WiFi interfaces used as data channel is based on configuration and the service that the SDN wireless node is required to support. The SDN module is the combination of packet processing and the interface that accepts input from a separated control plane.

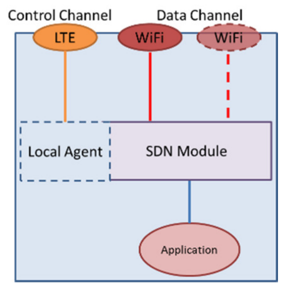

Figure 5. SDN wireless node internals

Each SDN wireless node has a local SDN agent, which functionality depends on what features are enabled on the SDN wireless node. This local SDN agent can either be the backup controller when connection to the SDN controller is lost or the primary SDN intelligence while receiving input from the SDN controller. Traditional Ad hoc routing protocols (e.g., GPSR, AODV, DSDV, and OLSR) are supported in agents as fallback mechanisms, to allow the SDN network to revert back to Ad hoc network operation even in the case where SDN controller communication is unavailable. In scenarios where the connection to a SDN controller is stable and has full control, this SDN agent has minimal intelligence.

One distinct characteristic of Ad hoc networks is that the nodes act both as Hosts (sending/receiving traffic) and Routers (forwarding traffic on behalf of other nodes). An SDN wireless node is therefore both an SDN data plane forwarding element and an end-point for data. Traffic from any wireless node (e.g. application traffic) will run through its own SDN module before being sent, which allows the SDN controller to determine the access of user traffic into the network.

B. Operations Overview

While the concept of SDN is the separation of control and data plane, there are differences in how a Software-Defined VANET can operate based on the degree of control of the SDN controller. We classify our architecture into three operational modes:

- **Central Control Mode:** This is the mode where the SDN controller controls all the actions of underlying SDN wireless nodes and RSUs. In specific, all the actions that the SDN data element performs are explicitly defined by the SDN controller. As shown in Fig 6, the SDN controller will push down all the flow rules on how to treat traffic.

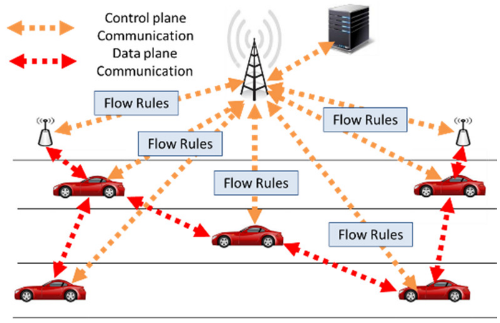

Figure 6. Central Control Mode

- **Distributed Control Mode:** This is a mode where underlying SDN wireless nodes and RSUs do not operate under any guidance from the SDN controller during data packet delivery. This control mode in essence is very similar to the original self-organizing distributed network without any SDN features, except that the local agent on each SDN wireless node controls the behavior of each individual node (e.g., run GPSR routing), as shown in Fig 7.

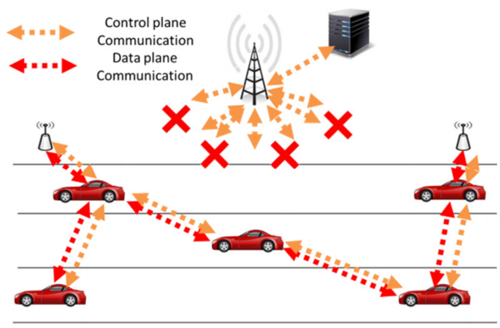

Figure 7. Distributed Control Mode

- **Hybrid Control Mode:** This mode includes all the operational modes of a system where the SDN controller exerts control anywhere between full and zero. Fig 8 shows an example, where the SDN controller does not hold complete control, but instead can delegate control of packet processing details to local agents. Therefore control traffic is exchanged between all SDN elements. One example would be that instead of sending complete flow rules, the SDN controller instead sends out policy rules which define general behavior, while the SDN wireless nodes and SDN RSUs use local intelligence for packet forwarding and flow level processing. In specific, the SDN controller instructs SDN wireless nodes and RSUs to run a specific routing protocol with certain parameters.

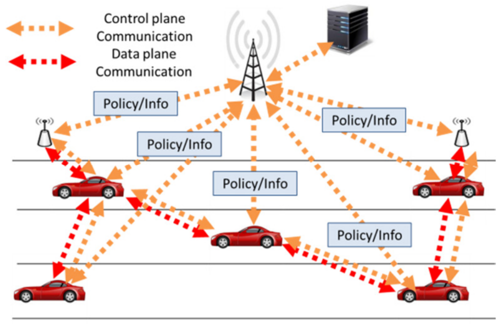

Figure 8. Hybrid Control Mode

The central control mode behaves similar to that of wired SDN architectures, where the SDN controller will insert all the rules. However, since the inherent problem of wireless channel is its reliability/availability, there are always potential communication losses between mobile nodes with the controller. This is the reason why a Software-Defined VANET must have failure recovery mechanisms that guarantee that the system can still function, even if at a reduced level, when SDN controller communication is lost or disrupted. The local agent located on each SDN wireless node has the intelligence to deal with such disruptions. For example, when communication to the SDN controller is lost, the system can revert back to running a traditional routing protocol, such as GPSR. In Section V, we demonstrate in simulation how this fallback mechanism can maintain good packet delivery even during SDN controller disruption.

Learning the network topology is important for the SDN controller to make intelligent decisions. We utilize beacon messages, a common feature in VANET systems. Each SDN wireless node will be exchanges beacon messages to learn information about immediate neighbors. This neighbor information is periodically updated to the SDN controller, which uses this information to build a node connectivity graph to make decisions, such as choosing paths to route packets through the network. Exploiting this feature can provide a lot of advantages for the management of mobility in a VANET scenario. We choose to use this beacon method over the Link Layer Discovery Protocol (LLDP) actually used in wired SDN systems.

IV. Software-Defined VANET Benefits and Services

As a separated control plane, SDN brings flexibility and programmability into the network. This brings awareness into the system so that it can adapt to changing conditions and requirements. In specific, this awareness allows a Software-Defined VANET to make better decisions based on the combined information from multiple sources, not just individual perception from each node. Also, dynamic and flexibility can react to sudden events, suitable for reacting to emergencies and changing requirements. In this section, we describe the benefits of Software-Defined VANETs, and describe several services that can be enhanced by utilizing these benefits.

A. *Software-Defined VANET benefits*

We classify benefits of a Software-Defined VANET into three individual areas:

- **Path Selection**: The awareness of SDN allows the system to make more informed routing decisions. For example, in a VANET scenario, data traffic can become unbalanced, either because the shortest path routing results in traffic focusing on some selected nodes, or because the application is video dominant which occupy big bandwidth on the path. When this situation is discovered by the SDN controller, it can start a reroute traffic process to improve network utility and reduce congestion.
- **Frequency/Channel selection**: When a SDN wireless node has multiple available wireless interfaces or configurable radios such as cognitive radios [15, 16], a SDN-based VANET can allow better coordination of channel/frequency used. For example, the SDN controller can dynamically decide at which time what type of traffic will use which radio interface/frequency. This can be used to reserve channels for emergency traffic for VANET emergency services.
- **Power selection**: Because of the awareness, a SDN-based VANET will have the information to decide whether changing the power of wireless interfaces, and therefore its transmission range, is a logical choice. For example, the SDN controller gathers neighbor information from SDN wireless nodes and determines that node density is too sparse and commands all nodes to increase power to achieve more reasonable packet delivery and reduce interference.

B. *Software-Defined VANET services*

Based on the benefits that we described earlier, we present services that can be enhanced using a Software-Defined VANET.

- **SDN Assisted VANET Safety Service**: Improving road safety through the use of V2V communications is one of the primary use cases of VANETs. We show how a Software-Defined VANET can improve the services when compared to traditional methods. SDN can be used to reserve or limit specific frequencies so that emergency traffic (or otherwise privileged traffic, such as security) uses this reserved path. The difference between this and traditional emergency channels is that reservation in our architecture is configurable dynamically. The SDN controller can assign flows to these channels or remove them based on current traffic conditions and application requirements. This can also be used to offer different level of services based on policies. The way this can be done is by changing rules during an emergency period. Emergency traffic gets priority over the remaining traffic.
- **SDN-based On Demand VANET Surveillance Service**: Surveillance service for emergency/authority vehicles is another area in which a Software-Defined VANET can be deployed. In traditional architectures, a requester (e.g. police car) must send out a request for the surveillance data (or even a broadcast for the data if the holder of the data is unknown to the requester). In a SDN-based system, this request is done by the SDN controller. The SDN controller simply inserts flow rules for the surveillance data to reach the requesting nodes. Also, when there are several requests for the same surveillance data, such as when multiple police request for video surveillance feed, the SDN controller inserts rules so that the same copy is sent to multiple destinations.
- **Wireless Network Virtualization Service**: Network virtualization services aims to provide abstract logical networks over shared physical network resources. SDN has already been used in data centers to provide network virtualization services, and we can apply the same idea for Software-Defined VANETs. The idea is to let different flows choose different radios/interfaces using different frequencies. If the radio frequencies used by each individual network is different, individual network's traffic are isolated from each other and we have thus effectively sliced the networks and created virtual wireless networks. One method would be the grouping of wireless nodes and RSUs, so each RSU only forwards traffic from a selected group of wireless nodes. Another more advance method would be to incorporate time slicing. The control of which network uses which radio interface/frequency for which time period is done by the SDN controller, which makes the allocation of network traffic a programmable fashion. Time slicing for efficient OFDM spectrum allocation used for LTE networks can be applied in the Software-Defined VANET to support one virtual wireless network per time slot. If multiple radio interfaces are available, multiple virtual networks can be supported in the same time slot. For example, ITS traffic is exchanged on frequency channel *f*1; MPEG DASH video is transmitted on frequency channel *f*2. Note that while the video packet broadcast on channel *f*2 is picked up by all neighbors tuned on *f*2, the nodes that will receive and forward the video packet is determined by SDN controller intelligence. Additionally, the SDN controller can set filters on node inputs so that some nodes, say, may reject certain traffic classes. This could be used, for example, to restrict the propagation

of video surveillance traffic to law enforcement vehicles. This input filtering is an SDN feature unique of wireless networks where broadcast is used, and can be used in combination with VANET surveillance services.

V. Simulation Evaluation

In this section we describe our simulation setups, configurations, and results. We model the architecture using the NS-3 simulator [17]. The goal of the simulations is to evaluate the feasibility of implementing services in a Software-Defined VANET.

A. *Comparison of SDN vs Traditional Ad Hoc Routing*

In this evaluation we compare SDN-based routing implemented for Software-Defined VANET and compare it to traditional Ad hoc routing. The simulation is performed over a SUMO [18] generated road network shown in Fig. 9, the road network is a grid type network that spans an area of 1000 x 1000 m2, with each road segment = 200m. Node density varies is 50 nodes in the simulation. The SDN controller LTE access is placed in the center of the simulation area where it is in wireless range of all SDN wireless nodes. Each SDN wireless node has multiple wireless interfaces; short range using 802.11with the Friis propagation loss model to limit the transmission range to 250m, and long range using LTE. Each simulation run features a pair of random nodes in the topology running a NS3 echo client-server streaming session, with a packet generation rate of 4 packets/s and packet size of 1024 byte. Beacon message interval is 500ms. SDN wireless nodes will update neighbor information to the SDN controller at intervals of 1s Simulation parameters were chosen based on MANET comparison studies [19]. Each set of simulations is averaged over 10 runs each running for 5 minutes.

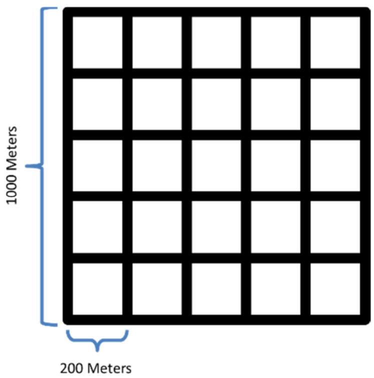

Figure 9. Grid Road Network

Fig 10 shows the comparison of SDN-based VANET routing to other traditional MANET/VANET routing protocols, including GPSR, OLSR, AODV, and DSDV. We use this evaluation to demonstrate the feasibility of a Software-Defined VANET.

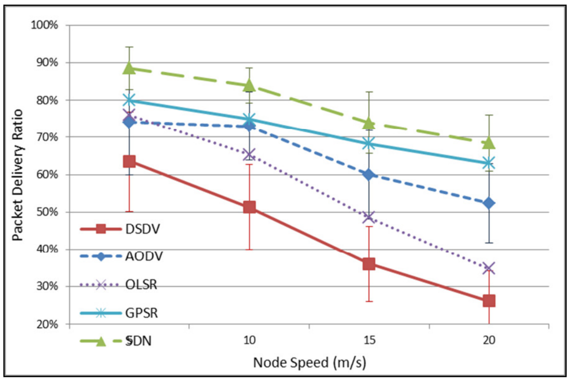

Figure 10. PDR comparison: SDN vs Traditional Ad hoc routing

We can see that our SDN-based routing outperforms the other traditional Ad hoc routing protocols. The aggregated knowledge that the SDN controller has is the major reason. As SDN wireless nodes update the SDN controller about neighbor information, the SDN controller immediately detects that there is topology change and sends out control messages as needed. Therefore our SDN-based system responds much faster to topology change.

B. *Failure Recovery from SDN Controller Connection Loss*

In this evaluation we demonstrate how fallback mechanisms utilized by the local agent in SDN wireless nodes can still provide good packet delivery even when communication to SDN controller is lost. Once again, the simulation is performed over the SUMO generated grid road network using the same experiment parameters. Fig 11 shows the scenario where there is a controller failure for 100 seconds, as shown by the dash lines.

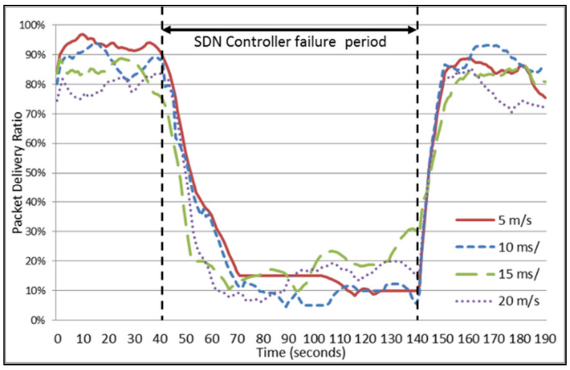

Figure 11. SDN controller failure

We can see that Packet delivery ratio immediately starts to drop as SDN controller no longer insert fresh rules for SDN wireless nodes. This demonstrates that operating a Software-Defined VANET in central control mode is dangerous if SDN controller communication is not reliable enough. The nature of a VANET is that nodes will move around quickly, and stale rules will become obsolete much more quickly compared to a scenario where node mobility is low.

We then show the same scenario, except that this time the fallback mechanism, running GPSR routing, is triggered when SDN controller communication is lost. Fig. 12 shows the packet delivery ratio of this scenario.

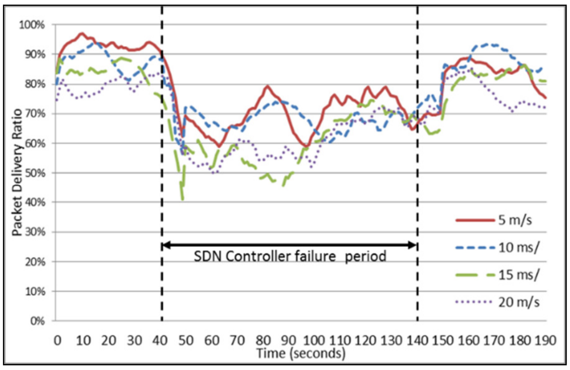

Figure 12. SDN controller failure with GPSR as fallback

We can see that after an initial drop (after the loss of communication with SDN controller and before GPSR is activated), good packet delivery ratio is restored. Once communication with SDN controller is restored, the system once again reverts back to SDN-based routing.

C. Adaptive Transmission Range Selection

In this evaluation, we demonstrate how allowing the SDN controller to dynamically control the transmission power of SDN wireless nodes can improve packet delivery. The simulation again is performed over the SUMO generated grid road network; however the node density is 30 nodes. Fig. 13 shows the result, with the dash line marking the time when the SDN controller raises the transmission power so that transmission range is now approximately 400 meters (up from the previous 250 meters).

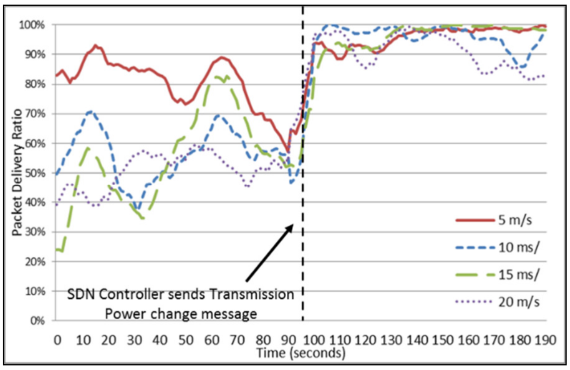

Figure 13. Adaptive Transmission Range Selection

We can see how immediately how packet delivery ratio becomes higher, as the new transmission range provides much better connectivity between nodes. The SDN controller can make the judgment to adjust transmission power because it has global information on the entire VANET scenario. In specific, in our evaluation the SDN controller increases transmission power because based on the information gathered from the SDN wireless nodes, it determines that the connectivity between SDN wireless nodes is too low.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose the architecture and services toward a Software-Defined VANET. The architecture captures the components and requirements needed to deploy SDN in VANET, and we described several different operational modes and the services that can be provided. We demonstrate in simulation several points: (I) the feasibility of a Software-Defined VANET by comparing SDN-based routing with traditional MANET/VANET routing protocols, (II) how fallback mechanism is a key feature that must be provided to apply the SDN concept into mobile wireless scenarios, and (III) transmission power adjustment as one of the possible services that can be provided by Software-Defined VANET.

For future work, there are several directions that we intend to explore. First, although we demonstrated how a fallback mechanism can maintain good packet delivery in the case of SDN controller failure, there are more advanced scenarios that should be considered. For example, there are cases of partial SDN controller connectivity loss, where only a subset of mobile nodes loses communication to the controller. In this case, the isolated nodes might need to form their own SDN cluster, or nodes with connectivity to the SDN controller can act as relay nodes to relay rules. The best action to take will depend on several factors such as node density, mobility pattern, or others.

Second, although we demonstrated the feasibility of a Software-defined VANET, our current architecture still requires infrastructure support (e.g. LTE). Therefore, we would like to investigate alternate Software-Defined VANET architectures such as where SDN controller transmits control traffic in P2P mode using WiFi channels. While this allows to build a wireless SDN system that is completely distributed and thus does not need infrastructure support, the communication with the SDN controller can be delayed and even interrupted causing new complications.

Third, we intend to investigate on additional Software-Defined VANET services and their performances. Such as the safety and surveillance services that we proposed, and also other new services that can be deployed by taking advantages of the benefits of SDN.

ACKNOWLEDGMENT

This work was partially supported by CNPq, CAPES, and FAPESP (Grant 2012/04945-7).

REFERENCES

[1] M. Gerla. “Vehicular Cloud Computing”, VCA 2012 Proceedings, Cyprus, June 2012

[2] C. Harsch, A. Festag, and P. Papadimitratos. “Secure position-based routing for VANETs.” In Proceedings of IEEE 66th vehicular technology conference (VTC-2007), Fall 2007 (pp. 26–30), September 2007

[3] N. Mckeown, “Software-defined networking.” INFOCOM keynote talk, Apr, 2009

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, 2008.

[5] OpenFlow Switch Specification, Version 1.4.0 (Wire Protocol 0x05). [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.4.0.pdf

[6] ONF Solution Brief, “OpenFlow-Enabled Mobile and Wireless Networks”, September 2013

[7] K.-K. Yap, M. Kobayashi, R. Sherwood, T.-Y. Huang, M. Chan, N. Handigol, and N. McKeown, “OpenRoads: empowering research in mobile networks,” SIGCOMM Comput. Commun. Rev., vol. 40, no. 1, pp. 125–126, January 2010.

[8] J. Vestin, P. Dely, A. Kassler, N. Bayer, H. Einsiedler, and C. Peylo, “CloudMAC: towards software defined WLANs,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 16, no. 4, pp. 42–45, 2013.

[9] Wireless & Mobile Working Group (WMWG). [Online]. Available: https://www.opennetworking.org/images/stories/downloads/working-groups/charter-wireless-mobile.pdf

[10] P. Dely, A. Kassler, and N. Bayer. “Openflow for wireless mesh networks.” In Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN), pages 1{6. IEEE, 2011.

[11] P. Baskett, Y. Shang, W. Zeng, and B. Guttersohn. “SDNAN: Software-Defined Networking in Ad hoc Networks of Smartphones”, Consumer Communications and Networking Conference (CCNC), 2013 IEEE

[12] T. Luo, H. Tan, and T. Quek, “Sensor openflow: Enabling sofware-defined wireless sensor networks,” Communications Letters, IEEE, Vol. 16 , Issue: 11, 2012

[13] M Mendonca, K Obraczka, and T Turletti, “The Case for Software–Defined Networking in Heterogeneous Networked Environments”, ACM converence on CoNEXT student workshop, 2012.

[14] C. Guimarães, D. Corujo, R. L. Aguiar, F. Silva, and P. Rosa, "Empowering Software Defined Wireless Networks Through Media Independent Handover Management", Proc. 2013 IEEE Globecom, Atlanta, USA, Dec 2013

[15] P. Pawelczak, R. Venkatesha Prasad, L. Xia, and I. Niemegeers. “Cognitive radio emergency networks-requirements and design.” In the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN), pages 601{606. IEEE, 2005.

[16] I. Akyildiz, W. Lee, and K. Chowdhury. “Crahns: Cognitive radio ad hoc networks.” Ad Hoc Networks, 7(5):810{836, 2009.

[17] NS-3, http://www.nsnam.org/

[18] Simulation of Urban Mobility (SUMO), http://sumo-sim.org/

[19] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva. “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols.” In Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’98), pages 85–97, Dallas, TX, October 1998. ACM