The following publication Z. He, J. Cao and X. Liu, "SDVN: enabling rapid network innovation for heterogeneous vehicular communication," in IEEE Network, vol. 30, no. 4, pp. 10-15, July-August 2016 is available at https://doi.org/10.1109/MNET.2016.7513858.

SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication

Zongjian He, Jiannong Cao, and Xuefeng Liu

Abstract—With the advances of telecommunications, more and more devices are connected to the Internet and getting smart. As a promising application scenario of carrier networks, vehicular communication has enabled many traffic related applications. However, the heterogeneity of wireless infrastructures and the inflexibility in protocol deployment hinder the real world application of vehicular communications. Software defined network (SDN) is promising to bridge the gaps through unified network abstraction and programmability. In this research, we propose an SDN based architecture to enable rapid network innovation for vehicular communications. Under this architecture, heterogeneous wireless devices, including vehicles and roadside units are abstracted as SDN switches with unified interface. In addition, network resources, such as bandwidth, spectrum can also be allocated and assigned by the logically centralized control plane, which provides a far more agile configuration capability. Besides, we also study several cases to highlight the advantage of the architecture, such as adaptive protocol deployment, and multiple tenants isolation. Finally, the feasibility and effectiveness of the proposed architecture and cases are validated through traffic trace based simulation.

1 Introduction

The advances of telecommunication and carrier network technologies have significantly improved the ubiquity of wireless communication. More and more devices are becoming connected with each other and to the Internet. Applying telecommunication technologies to vehicular scenario allows vehicles to connect to the cloud, roadside units (RSU), and ambient vehicles. Many new applications are enabled, such as intelligent transportation systems (ITS), urban computing and participatory sensing [1]. These applications have greatly improved the safety, efficiency and comfort of transportation. With the upcoming 5G wireless communication, which is envisioned to include direct device-to-device communication, massive machine communication and moving networks [2], the application of cellular communication in vehicular scenario is expected to grow rapidly, and bring new challenges to the architecture of carrier networks.

Cellular networks is the most widely used wireless media in today's vehicular communication. Millions of vehicles have been able to access Internet via UMTS/LTE [3]. However, imperfections remain for today's vehicular network to support a wide range of real world applications. Foremost, besides cellular networks, scores of other wireless communication technologies have been employed to vehicle-to-vehicle and vehicle-to-cloud communications. Such as Wi-Fi, DSRC, WiMAX and etc. The heterogeneity of these wireless technologies make the interconnection and interoperation an intractable task, which leads to network fragmentation and inefficiency of network resource utilization. Moreover, most of existing vehicular communication protocols are dedicatedly optimized for specific scenarios. For example, highway driving, urban driving, platoon driving, etc. However, the external context of a vehicle, such as traffic, road and weather conditions fluctuate rapidly. Unfortunately, the current hardwired implementation of protocol stack is difficult to switch protocols on-the-fly. Therefore, the inflexibility also restricts the popularity of vehicular communication.

Software defined networks (SDN) is an emerging network paradigm and has been increasingly diffused to different types of network systems. In SDN, the control plane and data plane are decoupled and network resources are managed by a logically centralized controller. Furthermore, devices from various vendors can communicate with each other via standardized interface. Therefore, it significantly simplifies the network management and offers a programmable and flexible network architecture.

Originally, SDN is designed and deployed in wired network environment with high-speed switches, such as data center network and campus network. In fact, the SDN architecture itself is general enough to be applied to the wireless scenarios. Researchers have proposed to use SDN to enhance many forms of wireless networks. E.g., carrier networks [4], WiFi networks [5], wireless sensor network [6], and wireless mesh networks [7]. Despite of the challenges of applying SDN to the wireless world, SDN indeed brings new insights and high potentials to improve the flexibility, programmability, efficiency, and evolvability of wireless networks [8].

With the benefits brought by SDN, we firmly believe that SDN is the right choice to bridge the gaps of vehicular application demands and today's limitations in vehicular networks. Yet, attempt exists to construct software defined vehicular ad-hoc networks (VANET) [9]. The

- *Zongjian He, Jiannong Cao and Xuefeng Liu are with Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China.*

pioneer explores reveal the potential of deploying SDN for vehicular communication. However, it only focuses on centralized control of vehicle-to-vehicle communication. In addition, the heterogeneity of vehicular network remains unsolved, as resources are statically assigned: use cellular network for control plane communication, and ad-hoc for data plane communication. We argue that vehicle-to-cloud communication is easier to be deployed and likely to get popularized than vehicle-to-vehicle communication. Hence, it should also be considered in SDN based vehicular networks.

In this paper, we take a step forward and present our work in SDN based heterogeneous vehicular networks. We name our system as SDVN, or Software Defined Vehicular Network. Firstly, we design an SDN based architecture for vehicular network, which integrates not only vehicular to vehicle communication, but also vehicle-to-infrastructure, and vehicle-to-cloud communication. With SDVN, rapid network innovation is enabled by configuring the network on-demand. Moreover, to mitigate the SDN management overhead, trajectory prediction based vehicle status update policy is employed. We also design several application scenarios as case study to demonstrate the feasibility and advantages of SDVN. We validate the proposed architecture and cases through traffic trace based simulation.

2 Opportunities and Challenges of Software Defined Vehicular Network

Besides the aforementioned advantages enabled by SDN that are common for both wired and wireless networks, the design philosophy of SDN brings some unique benefits to vehicular networks in many aspects. We emphasize the following key opportunities:

- **Heterogeneous network integration**: With network virtualization and abstraction, all the vehicles, RSUs and wireless infrastructures can be regarded equally as SDN switches and managed with a unified interface, which remarkably simplifies the integration of heterogeneous vehicular networks.
- **Improve network resource utilization**: With logically centralized controller, it is easier to coordinate among vehicles and allocate all kinds of network resource efficiently. E.g., the control plane can allocate dedicated wireless channels and frequency to high priority messages. Besides, the control plane can also adjust the wireless transmission power, which will affect the transmission range. With well collaborated transmission range, the probability of packet collision can be reduced.
- **Rapid network configuration**: The control plane can adaptively deploy routing protocols and adjust their parameters according to the fast changing external context. In this way, existing vehicular communication protocols can be better utilized.

On the other hand, applying SDN to vehicular network also poses challenges that need to be properly considered. SDN for vehicular communication will not be feasible without handling these challenges.

- **Highly dynamic mobility**: The key challenge is that the topology of vehicular network is highly dynamic due to vehicle mobility, which makes the control plane difficult to maintain the status of vehicles and hence increases SDN management overhead.
- **Broader rule definition**: The conventional SDN forwarding rules are generally designed for Ethernet with IP-based protocols. However, in vehicular network, many new vehicle specific factors can also impact the performance of the networks. For example, the geographical location of vehicles, the wireless media, and etc. Directly apply the flow rules in SDN to vehicular network may limit the application scenario. It is necessary to define broader rules and flows to better adapt the requirement of vehicular networks. E.g., to support the widely used geographical routing, it is necessary to forward the packets based on vehicles' locations rather than their MAC/IP addresses.

3 Case Studies

In this section, we present several cases to show how SDVN can enable rapid network innovation. The first case highlights the advantage of unified management of heterogeneous wireless devices. The second case shows the programmability of SDVN by selecting and deploying appropriate routing protocol on demand. The last case demonstrates the flexibility of using network slicing to isolate multiple tenants.

3.1 Heterogeneous Multi-hop Routing

Conventionally, a vehicle can only send message to another vehicle using multi-hop vehicle-to-vehicle ad-hoc communication or relay the message using RSU/cellular station. Unfortunately, both approaches are not perfect. Message may get lost in the ad-hoc communication, especially during sparse traffic condition, when the next hop is not always available. For cellular communication, the data transmission is ralatively costly.

With SDVN, it is possible to use multiple wireless interfaces collaboratively to send messages for better performance or lower cost. Fig. 1 illustrates an example of heterogeneous multi-hop to minimize the monetary cost. The leftmost vehicle wants to send messages to the rightmost one. Although messages can be delivered directly by cellular communication, considering the high cost of using cellular transmission for both up-link and down-link traffic, the control plane decides to choose a better approach: the sender firstly passes the message using ad-hoc communication (IEEE 802.11p) to its neighbor. Then the neighbor uploads the message to RSU (WiFi). Since the receiver only has cellular interface, the RSU then forwards the message to cellular base station (wired backbone), which finally delivers the message to

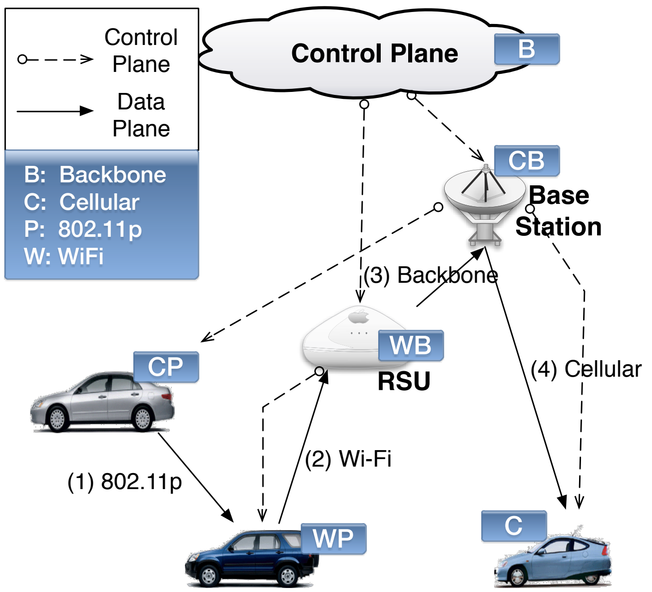

Fig. 1. Multi-hop routing using heterogeneous wireless interfaces (shaded blocks are the physical interfaces owned by each device).

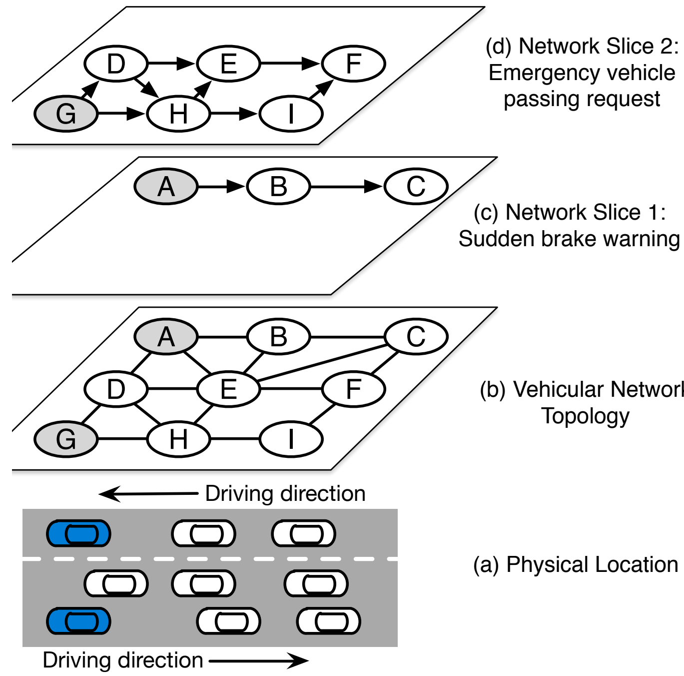

Fig. 2. Multiple tenants isolation via network slicing

the receiver (cellular network). During the entire process, cellular network is only used by the last hop down-link. Since today's WiFi and wired communication are cheap-as-free, half of the communication cost is saved.

3.2 Adaptive Protocol Deployment

Nowadays, a large amount of decentralized ad-hoc routing protocols for vehicular communication have been designed by researchers. They have significantly advanced the development of vehicular networks. When applying SDN to the centralized vehicular network, a natural question arises: *Instead of abandoning decentralized protocols and develop new centralized ones, is it possible for SDVN to make better use of existing protocols*?

We notice that most of existing routing protocols are dedicatedly customized for certain scenarios. For spatial dimension, there are protocols for highway driving and urban driving. For temporal dimension, there are protocols for time-critical applications and non-time-critical ones. However, the spatial and temporal context of a vehicle changes from time to time. With SDN controller, it is possible to identify the current context and the most suitable protocols accordingly. Instead of making the entire routing decision, the SDN controller may just instruct the vehicles to use the most suitable protocols to adapt current context. In this way, the overall network performance can be improved.

For decentralized routing, two categories of protocols are widely used: structure based and structure free [10]. Generally, structure based protocols need to maintain a data structure to organize all the vehicles, while structure free protocols do not. Therefore, structure based protocols are suitable for the scenarios that the network topology changes not so often. Otherwise, the structure maintenance overhead is not negligible, and structure free protocols are more suitable. The SDN control plane can firstly query the real-time traffic information of a region. If the region is congested and all vehicles are moving at low speed, the SDN controller will instruct the vehicles in that region to use structure based routing protocol for data delivery. On the other hand, structure free protocols can be applied to avoid the topology maintenance overhead.

3.3 Multiple Tenants Isolation

In some time-critical vehicular applications, such as emergency warning, broadcast are widely used. However, if multiple broadcast are sent simutaniously, the chances of packet collision are considerably high, and the packet delivery deadline may be missed.

Conventionally, VLANs are used to slice the network, or partition multiple tenants for distinct user groups to mitigate the interference. Recently, some SDN based slicing mechanisms, such as FlowVisor [11], have been proposed for network slicing. Differently from VLAN, SDN based slicing is implemented with software, and is more flexible and configurable. Therefore, it is more suitable for fast changing vehicular network. Moreover, with network slicing, the upper layer protocol design can be significanlty simplified.

An example of network slicing is illustrated in Fig. 2. Physically, several vehicles are driving in different lanes with difference directions (Fig. 2a). The network topology these vehicles formed are shown in Fig. 2b. Vehicle **A**

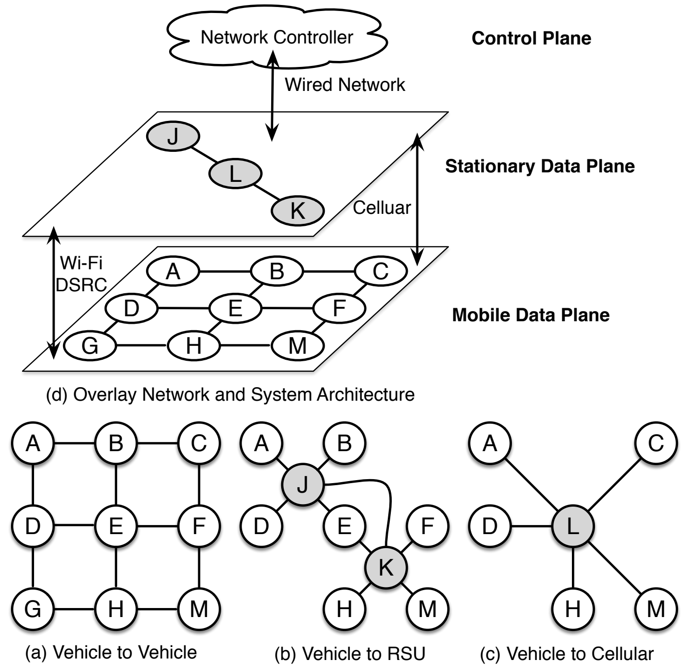

Fig. 3. Vehicular networks and the overlay abstraction

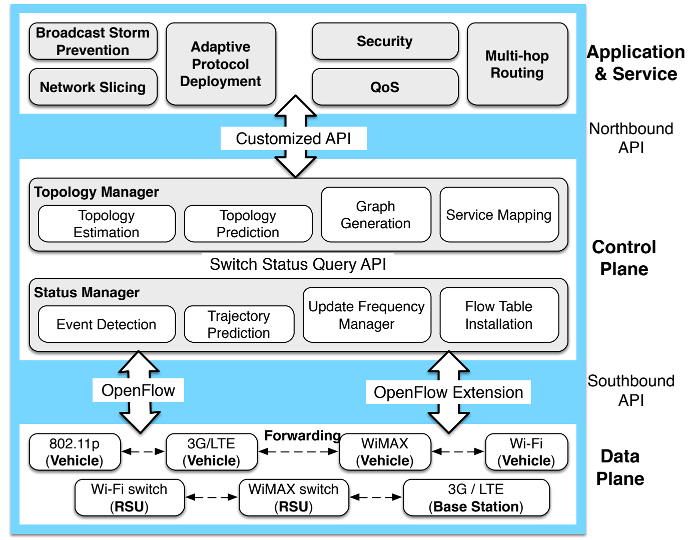

Fig. 4. SDVN System architecture

initiates a sudden brake and sends broadcast to notify the vehicles following it to avoid rear-end collisions. Naturally, it is pointless for vehicles at the opposite side of the road (Vehicle **D** and **E**) to receive the broadcast. Similarly, the ambulance **G** sends a broadcast to request other vehicles to give way. This broadcast should also not be received by vehicles on the opposite side of the road. In this case, SDN controller can simply slice the network according to the vehicles' driving directions. Without network slicing, to prevent broadcast storming, the broadcast protocol need to be carefully designed to determine when to re-broadcast, which will increase the complexity in design and implementation.

4 The SDVN System

4.1 System Architecture

Currently, the heterogeneity of existing network infrastructures have caused challenges in network management and integration. This situation is depicted in Fig. 3. Furthermore, the highly dynamic mobility increases the vulnerability of communication. Motivated by the concept of network virtualization [12], we propose to exploit SDN to tackle these challenges in vehicular networks. First of all, we need to make abstraction of the existing vehicular networks to adapt the concepts in SDN.

Data Plane: Data plane includes all data transmission network devices. In vehicular scenario, we construct the data plane with an overlay network to eliminate the heterogeneity of existing vehicular networks. All vehicles, roadside units, and base stations are abstracted as SDN switches. These SDN switches can be further categorized into *mobile data plane* and *stationary data plane*, according to their mobility. Roadside units and base stations belong to stationary data plane, while vehicles are in mobile data plane. Different management policies are applied to the two data planes.

Control Plane: Control plane maintains the status of all the switches and is responsible for making packet forwarding decisions based on it. Control plane is logically centralized, and the control of the networks is transferred from individual switches to the controller. In vehicular scenario, the switch status includes vehicle location, velocity, and network connectivity. Nowadays, almost all the vehicles have been equipped with localization devices like GPS that can provide such information.

Communication Interface: The control plane and data plane can communicate with each other with a unified interface (a.k.a., Southbound API), which includes some predefined control and notification messages. In generic SDN, OpenFlow is the dominating communication protocol. In vehicular scenario, standard OpenFlow need to be extended to adapt vehicular requirements. Network applications use northbound API to communicate with control plane. Currently, there is no standardized interface for northbound API. In our system, we also define customized interface.

With unified management interface, all network infrastructures in vehicular networks can be managed by the control plane equally. Thus, it is possible for upper layer protocols to take advantage of multiple physical networks for a single task. E.g., routing. Note that "unified interface" only means the same network management protocol, like OpenFlow. Physically, two peers still need to adopt the same physical standard to reach each other.

Fig. 4 shows the architecture of all the functional components of SDVN. They are divided into three layers. From bottom up, data plane, control plane, and application & service plane. Southbound API between control and data plane is implemented by extending OpenFlow standard. Northbound API is provided to the applications as a high level abstraction to communicate with SDN controller.

As control plane is the most important component in the architecture, we would explain the two modules: status manager and topology manager in detail.

4.2 Status Manager

To support logically centralized control of data plane, the control plane must collect and maintain the status information of all SDN switches, including vehicles and roadside units. Status manager is designed for this task.

Trajectory prediction is the key function of status manager, as it tackles two important challenges of applying SDN to mobile networks: reachability and mobility. To collect and maintain the status of all SDN switches, the reachability of a switch is crucial. Stationary data plane switches and mobile data plane switches with cellular interfaces can be reached with reliable connections [3]. Vehicles that are directly connected to the RSUs can also be reached by control plane. Special attentions need to be paid for the rest of vehicles, which can be temporary disconnected from the control plane. An intuitive way to address this issue is to "fall back" to the decentralized ad-hoc communication while disconnected with the control plane [9]. However, this approach will increase the complexity of both software and hardware design of switches. Differently, we employ *trajectory prediction* to estimate the possible positions when they are disconnected. In addition, even though all vehicles are fully reachable by control plane at all time, tracking the location of vehicles in real-time can still cause a remarkable management overhead. Because vehicles are highly dynamic and their status are fast changing. This issue can also be tackled by trajectory prediction.

The future trajectory of vehicles can be obtained in many ways. 1) For public transportation with fixed schedules (e.g., buses), their trajectories are predetermined. So the future trajectories can be easily obtained by looking up the scheduling table. 2) For vehicles using navigation systems, the drivers will normally drive following the suggested path from the navigation system, which can be regarded as the future trajectory. 3) For the rest of the cases, researchers have developed both macroscopic and microscopic models for trajectory prediction [13], which can be directly employed in this research.

Event detection and update frequency manager define the detailed switch status update policy. With trajectory prediction, the switch status update policies are designed as event driven rather than polling. When a specific event occurs (e.g., flow table miss, timeout), the controller will issue status update request. The request is implemented either by real vehicle position collection via network, or just re-calculation of the estimated position via prediction, depending on whether the predicted vehicle position is still valid. Whether a prediction is valid or not is determined by update frequency manager, which periodically mark predicted vehicle position longer than a specific threshold as invalid.

4.3 Topology Manager

The status information provided by status manager may not be directly used by upper layer network protocols and services, as they are usually more interested in network topology, which can be described as a graph. E.g., most of the routing protocols take network topology as input. Hence, the control plane also need to maintain the network toplogy information. Topology manager is designed to achieve this objective.

The key function of topology manager is to generate network topology using SDN switch status. Several approaches are applied to obtain the network topology: 1) For stationary data plane with wired communication, the network topology seldom changes. Hence, we treat the topology also as stationary. 2) For mobile data plane, the collected status includes neighbour information, which can be used to construct the network toplogy. 3) If the vehicle position is obtained via trajectory prediction, there is no neighbour information available. In this case, topology manager will estimate the network toplogy using mean transmission range.

Flow table management is another key function, collaboratively handled by both status manager and toplogy manager. The basic idea of flow table update policy is to pre-install routing table based on topology prediction. Specifically, when a vehicle is connected to the controller, the controller will predict the future trajectory of the vehicle, and estimate its topology change. Based on the predicted network topology, the SDN controller installs corresponding flow table, which will be used for a short period of time in the future. By doing this, the SDN management overhead can be significantly reduced. In addition, even if the vehicle is disconnected from control plane, it can still use the pre-installed flow table to perform tasks like routing. Technically, we adjust the *timeout* and *priority* value of each flow table entry to reflect the network topology change.

4.4 Comparison with Generic SDN

To make the generic SDN be applicable for vehicular communication. Several changes have been made. In this section, we summarize the similarity and differences between SDN and SDVN.

The design of SDVN adopt the fundamental architecture and concept of SDN, such as control plane, data plane and northbound/southbound API. Meanwhile, the network management is also transitioned from swithces to the logically centralized controller, where decisions are made based on the status of switches. In addition, like generic SDN, SDVN also manages heterogenious network devices using unified interface.

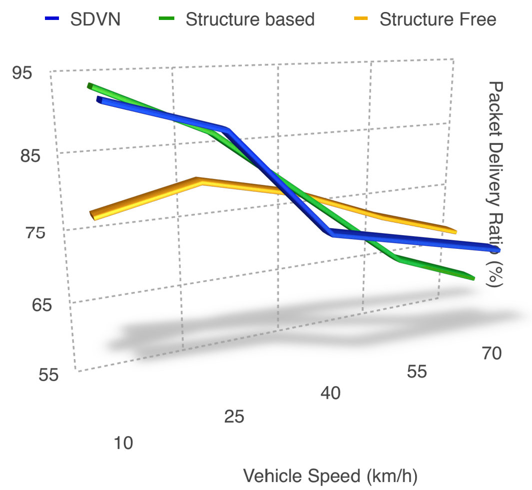

Fig. 5. Packet delivery ratio with adaptive protocol deployment

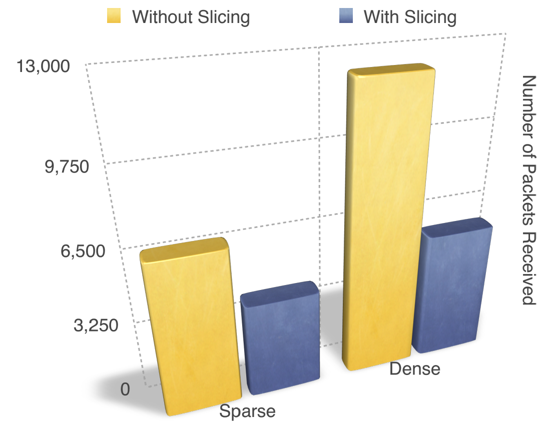

Fig. 6. Packet count of multiple tenants isolation via network slicing

To adapt the highly dynamic mobility of vehicles. Remarkable changes have been made. The differences can be summarized as follows:

- In SDVN, we define two subcategories of data plane: mobile data plane and stationary data plane, and handle the status collection with different strategies. While in generic SDN, all data plane components are stationary.
- In SDVN, to reduce controller overhead, switch status is maintained via both direct status collection and estimation. While in generic SDN, the status is directly collected from swichtes.
- In SDVN, several extensions are made based on OpenFlow standard. OpenFlow protocol can now support to report switch position and velocity; Flow table can support not only routing, but also adaptive protocol depolyment.
- In generic SDN, all the SDN switches have the same hardware interface (Ethernet). However, in SDVN, the heterogious devices have different wireless interface (WiFi, DSRC, cellular network).

5 Evaluation

To verify the feasibility and effectiveness of the software defined vehicular network architecture, we develop an evaluation platform using network simulator NS-3, traffic simulator SUMO, and open source SDN controller POX. To make the evaluation convincing, we adopt a typical urban traffic trace dataset collected from the TAPAS-Cologne project [14]. We extend the OpenFlow standard to support vehicle status update (position, velocity, neighbors) and routing protocol deployment (structure-based or structure free). We also add some road side units with uniform distribution.

First, we evaluate the performance of adaptive protocol deployment. We select two typical protocols widely used in vehicular network: OLSR as structure based protocols, and GPSR as structure free [10]. The traffic speed varies from time to time gradually, we adopt a simple protocol deployment policy: if the mean speed increases to 35 km/h, we deploy structure free GPSR, while if the speed decreases to 25 km/h, the controller switch the routing to structure based GPSR. The 10 km/h buffer is set to avoid frequent protocol switching, which may decrease the overall performance.

We select the packet delivery ratio as the performance metric, and the results is depicted in Fig. 5. From this figure, we can see that the overall packet delivery radio decreases with vehicle speed. This is mainly caused by the fact that when vehicle speed is lower, the density of vehicles is usually also not high. In this case, vehicles may have difficulty in finding next hop neighbors, which will cause packet delivery failure. Despite of this common factor, we can see that with the help of SDN and adaptive protocol switching, the overall performance are remarkably improved than any of the two protocols used individually.

Next, we evaluate the effectiveness of multiple tenants isolation. We simply slice the network according to the driving directions, and select random vehicles to send broadcast within certain ranges. The broadcast is expected to be received by only vehicles driving towards the same directions. We use data around 6 AM to represent sparse traffic and 8 AM as dense traffic.

Fig. 6 shows the evaluation results. We record the total number of packets delivered in the entire network. Clearly, with network slicing, the number of packets delivered are less than without it. Imagine if the number of vehicular networking applications is large, network slicing can significantly reduce the probability of packet collision and improve the bandwidth utilization.

6 Conclusion and Future Work

In this paper, we have studied the opportunities and challenges of applying the emerging SDN to vehicular networks scenario. We have presented our efforts in designing an SDN based vehicular network, which abstracts all the network components in vehicular networks as SDN switches in a unified way. By doing this, the heterogeneity of vehicular network can be mitigated. To reduce the SDN management overhead caused by the highly dynamic mobility of vehicles, the proposed system utilizes vehicle trajectory predictions to lower the frequency of status update. We have also presented some case studies to exhibit the strength of SDVN. Finally, we have evaluated the performance to validate the feasibility and effectiveness of our system.

Based on the proposed framework, there are lots of research issues to be further investigated. First, the architecture itself need to be enhanced, especially for northbound and southbound APIs. We plan to extend the OpenFlow standard to support more features in vehicular communication, including power adjustment, channel assignment, etc. Second, to fully utilize the SDVN platform, new protocols need to be redesigned to adapt the centralized and globally aware controller for better performance. For example, geographical cast, multicast, and etc. Last but not least, we want to exploit solutions for constructing a logically centralized but physically distributed SDN controller to improve the scalability [15], as in some city-wide vehicular networking applications, the number of vehicles involved are considerably large.

Acknowledgment

This research is financially supported in part under NSFC/RGC Joint Research Schemes 3-ZG1L, the NSF of China with Grant 61332004.

References

[1] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "Lte for vehicular networking: a survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, 2013.

[2] A. Gohil, H. Modi, and S. Patel, "5g technology of mobile communication: A survey," in *International Conference on Intelligent Systems and Signal Processing (ISSP)*, 2013, pp. 288–292.

[3] Y. Park, J. Ha, S. Kuk, H. Kim, C.-J. Liang, and J. Ko, "A feasibility study and development framework design for realizing smartphone-based vehicular networking systems," *IEEE Transactions on Mobile Computing*, 2014.

[4] A. Asensio, L. Gifre, M. Ruiz, and L. Velasco, "Carrier SDN for flexgrid-based inter-datacenter connectivity," in *International Conference on Transparent Optical Networks (ICTON)*, 2014, pp. 1–4.

[5] J. Schulz-Zander, L. Suresh, N. Sarrar, A. Feldmann, T. Hühn, and R. Merz, "Programmatic orchestration of WiFi networks," in *2014 USENIX Annual Technical Conference (USENIX ATC)*. USENIX Association, 2014, pp. 347–358.

[6] T. Luo, H.-P. Tan, and T. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.

[7] J. Chung, G. Gonzalez, I. Armuelles, T. Robles, R. Alcarria, and A. Morales, "Experiences and challenges in deploying OpenFlow over a real wireless mesh network," in *IEEE Latin-America Conference on Communications (LATINCOM)*, 2012, pp. 1–5.

[8] C. Chaudet and Y. Haddad, "Wireless software defined networks: Challenges and opportunities," in *IEEE International Conference on Microwaves, Communications, Antennas and Electronics Systems (COMCAS)*, 2013, pp. 1–5.

[9] I. Ku, Y. Lu, E. Cerqueira, R. Gomes, F. Ongaro, and M. Gerla, "Towards software-defined VANET: Architecture and services," in *Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2014.

[10] Z. He, "Structure based or structure free? topology management in VANETs," in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2012, pp. 1–4.

[11] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T.-Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, S. Seetharaman, D. Underhill, T. Yabe, K.-K. Yap, Y. Yiakoumis, H. Zeng, G. Appenzeller, R. Johari, N. McKeown, and G. Parulkar, "Carving research slices out of your production networks with OpenFlow," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 129–130, 2010.

[12] N. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, 2009.

[13] P. Pathirana, A. Savkin, and S. Jha, "Location estimation and trajectory prediction for cellular networks with mobile base stations," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 6, pp. 1903–1913, 2004.

[14] S. Uppoor and M. Fiore, "Large-scale urban vehicular mobility for networking research," in *IEEE Vehicular Networking Conference (VNC)*, 2011, pp. 62–69.

[15] X. Li, P. Djukic, and H. Zhang, "Zoning for hierarchical network optimization in software defined networks," in *Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–8.