Rapport d'avancement du projet E-horizon sur la thèse #20 "Vers des réseaux véhiculaires programmables via la technologie SDN"

Soufian Toufga, Slim Abdellatif, Philippe Owezarski, Thierry Villemur

Projet E-horizon

Rapport d'avancement du projet E-horizon sur la thèse #20 “Vers des réseaux véhiculaires programmables via la technologie SDN”

Doctorant : Soufian Toufga

Encadrants au LAAS : Slim Abdellatif, Philippe Owezarski, Thierry Villemur

Référent à CDSF : Bruno Coudoin

Date de remise de la version finale : 28/03/2019

Thesis presentation

Towards Software Defined Vehicular Networks

Despite the technological and strategic advancements in terms of road safety, the majority of world's transportation systems still suffer from serious safety and comfort problems. The latest report published by the "French Observatory of Road Safety" [18] shows that in 2016, the France's mortality rate on the road has increased for the third year in a row, an increase that particularly affects cyclists and pedestrians with more than 9% and 19% respectively, compared to 2015. Study in [16], shows that Germans spend on average 36 hours per year in traffic jams and the manner of driving has an impact on the fuel consumption up to 20 %. The economic cost of road accidents represents 1.4% of Germany's GDP, 3.2% in Austria, while it reaches 6% in the USA.

To overcome these problems, ITS (Intelligent Transport System) is conceived to provide the vehicles and the transport infrastructure with communication capabilities, in an effort, to improve their safety, efficiency and quality[1]. Therefore, the next generation of vehicles will be connected and equipped with one or more interfaces allowing it to communicate with other elements of the Intelligent Transportation System, includings other vehicle, pedestrians, infrastructure (RSU, BS, Cloud) forming a vehicular network.

These networks represents a crucial component of an ITS system, they are expected to support large volume of traffic originating from a variety of services with different QoS expectations. Moreover, flexibility is desired in order to promote the emergence of new services. Despite the attention that vehicular network research got recently, current solutions lack flexibility and efficiency in network resource utilization. The global aim of this PhD is to investigate the application of the SDN concept to vehicular networks. The expected contributions are:

(1) Design an SDN based architecture for vehicular networks that meet the QoS requirements of ITS services, (2) Identify the SDN basic services and network functions control to be extended/ adapted for vehicular network context, (3) Explore the possibilities of using the data collected by the other actors involved in the system (ITS service providers, Road authority, Weather forecast center) in order to achieve an intelligent effective network control. Machine learning and Artificial Intelligence approaches will be considered.

The proposed solutions and use cases will be evaluated using network emulation such as Mininet-wifi [15] combined with real mobility model using SUMO [19].

Introduction

Vehicular networks are one of the cornerstone of an Intelligent Transportation System (ITS). They are expected to provide ubiquitous network connectivity to moving vehicles while supporting various ITS services, some with very stringent requirements in terms of latency and reliability. Two vehicular networking technologies are envisioned to jointly support the full range of ITS services : DSRC (Dedicated Short Range Communication) for direct vehicle to vehicle/Road Side Units (RSU) communications and cellular technologies. To the best of our knowledge, approaches from the literature usually divide ITS services on each of these networks according to their requirements and one single network is in charge of supporting the each service. Those that consider both network technologies to offer multi-path routing, load balancing or path splitting for a better quality of experience of ITS services assume obviously separately controlled networks.

In a first part, We propose a hybrid architecture combining these technologies that follow the SDN paradigm with the intention of achieving integrated network control and improved along with "on the fly" and dynamic network programmability.

A crucial piece of any SDN system is the topology discovery service, implemented by the controller, whose objective is to build and maintain a wide view of the underlying network from which customized views are exposed to network control functions. The second part of our work focuses on the design of such service for the vehicular networks context, which represents a crucial step towards programmable vehicular communications.

Technical objectives

Under the umbrella of SDN (Software Defined Networking), we propose a hybrid network architecture that enables the joint control of the networks providing connectivity to multi-homed vehicles and, also, explore the opportunities brought by such an architecture. We show through some use cases, that in addition to the flexibility and fine-grained programmability brought by SDN, it opens the way towards the development of effective network control algorithms that are the key towards the successful support of ITS services and especially those with stringent QoS. We also show how these algorithms could also benefit from information related to the environment or context in which vehicles evolve (traffic density, planned trajectory,...), which could be easily collected by data providers and made available via the cloud.

The topology discovery service is a crucial component of this architecture, recent studies have highlighted many limitations of the "de facto" discovery protocol (Openflow and OFDP : Openflow Discovery Protocol) when applied to wireless networks in terms of security, overhead and capabilities [20] [21] [22]. The second objective of this work is to further analyze the limitations of Openflow and OFDP with respect to some of the challenging specificities of vehicular network, namely the high density of wireless nodes and their potential high mobility. From these limitations, we outline some of the design guidelines for an efficient topology discovery service for vehicular networks.

State of the art

In the last few years, with the emergence of new ITS services, vehicular networks are attracting more attention. Various scientific research organizations, industries and standardisation bodies are interested in improving them by proposing new architectures and mechanisms in order to effectively support these new services.

To that end, "I. Ku" and al. [6] propose to apply SDN to VANET networks in order to control the inter-vehicle ad-hoc communications and vehicle to RSU communications. A routing algorithm that exploits the global view of the SDN controller is proposed and performance tests show better results compared to traditional VANET routing protocols, which motivates the application of the SDN paradigm in that context. To address the problem of resiliency raised by the use of a single controller, a fallback mechanism based on a local controller embedded in the vehicle is also proposed to run traditional routing protocols in case of connectivity loss with the controller. However, this approach introduces an additional user cost. Based on the architecture of [6] and the global vision provided by the SDN controller, other routing algorithms were also proposed in [7, 8] and compared, by simulation, to traditional VANET routing protocols to show the performance benefits of SDN even when the programmable nodes are moving vehicles. In the same direction, "Y. C. Liu" and al. [9] propose a Software Defined Network (SDN) architecture for GeoBroadcast in VANETs in which the RSU entities are openflow enabled, and connected via Openflow switches, all under the control of an SDN controller. Performance tests show that with programmable RSUs, better performance is achieved in comparison to the GeoNetworking protocol [10] used in traditional ITS architectures. "N. B. Truong" and al. [11] explore the use of Fog computing in an SDN-VANET architecture in order to effectively support low latency services. In [12], a Decentralized Software-Defined VANET Architecture is proposed where the SDN controller is distributed to address the scalability issue inherent to very dense environment. Results show that, control plane distribution improves network scalability, in addition to keep better delivery packets delays.

Our proposition differs from previous works in two main directions, Firstly, by applying SDN to the global ITS communication architecture including not only ad-hoc or RSU networks, but also the cellular network in a hierarchical manner in order to enforce a scalable network. This paves the way to the development of new network control mechanisms that take advantage of the ability to simultaneously control all available network resources. Secondly, we leverage the data collected by cloud platforms of the ITS service providers [13] to devise wise, proactive and effective network control algorithms that are aware of the environment and the context in which vehicles evolve (density and speed of vehicles, weather conditions,...) and may evolve in the near future. Indeed, with these data, an estimate of the network topology, network and node loads can be predicted and potential changes in network conditions can be anticipated and treated proactively.

Reference	**Programmable components of the architecture that can be used for routing**			**SDN Controller**	**Using Data present in the cloud**
	Vehicle	**RSU**	**BS**		
[6]	✓	✓		centralized	
[7]	✓			centralized	
[8]	✓			centralized	
[9]	✓	✓		centralized	
[12]	✓	✓		hierarchical	
[11]	✓	✓	✓	hierarchical	
Our proposition	✓	✓	✓	hierarchical	✓

Table I . positioning w.r.t related work

Uncertainty, scientific and technical locks to overcome

In this section, We first present the global communication architecture of an ITS, the SDN Paradigm, and the topology discovery service . Then, we analyze the limitations of the Openflow-based topology discovery approach for SDVN (Software Defined Vehicular Networks).

- *Global communication architecture of an Intelligent Transport System*

The global architecture consists of three main parts as shown in Fig.1: The vehicle, the network infrastructure managed by an Internet Service Provider (ISP), and the cloud platform controlled by an ITS service provider :

❖ Vehicle: A vehicle is equipped with a set of sensors and systems (GPS, Radar, Lidar, Advanced Driver-Assistance System (ADAS) camera, etc.) enabling it to collect several information about its environment (position, speed, neighboring vehicles, temperature, etc.). Depending on its location, it can be reached, as shown in Figure 1, only by a Road Side Unit (RSU) (vehicle A), or only by a Base Station (vehicle B), or both (vehicle C), or it may be out of any network coverage (vehicle D). A vehicle can be equipped with several interfaces allowing it to interact with the various components of the system: (1) a 3/4G interface enabling it to benefit from different functionalities offered by the cellular network (Internet access, communication with other parts of the system (Vehicles, Cloud, etc.)), (2) a Dedicated Short Range Communication (DSRC) Interface enabling it to communicate with the RSU entities as well as other vehicles equipped with the same interface, and (3) a short range wireless Interface (e.g. Bluetooth) allowing it to communicate with the connected objects that surround it, as well as with the different User Equipments (UE) handled by the pedestrians, as illustrated in Figure 1. A vehicle acts not only as an end node, but also as a router to transmit information to other vehicles.

- Network Infrastructure: The network infrastructure is composed mainly of two parts: RSU and cellular network
  - Road Side Unit: A RSU entity represents one of the dedicated components for an ITS system. It may be implemented in a base station, or in a dedicated stationary entity installed along the road. It is mainly equipped with a DSRC interface, with which, it can communicate with any component equipped with the same interface (vehicle, RSU, etc.). Its communication range depends on the environment and the technology used. For example an RSU entity that supports the DSRC standard can have a communication range of 300 m in urban environments, and a communication range up to 1km in rural environnements. The Road Side Units may be interconnected via a wired or wireless medium, and they can not only provide a local service but also a cloud service and/or Internet access to the different vehicles.
  - Cellular Network: The cellular network represents one of the main technologies that may support the different vehicular communications. It has a very high network capacity enabling it to support applications requiring high throughput/bandwidth demands. It has a very high network capacity enabling it to support applications requiring high throughput/bandwidth demands. Moreover, it is characterized by a wider communication range, which allows a base station to maintain connectivity with a network node (vehicle) as long as possible,thereby limiting handover operations. In addition, It offers Multicast/ Broadcast transmission services (MBMS/eMBMS) and D2D communication technology, which can be used extensively in an ITS system.
- Cloud/ Fog computing : The cloud computing is the intelligent part of the system. It has a high storage and processing capabilities to massively collect data, and process them to provide customized ITS services to different vehicles, whereas Fog computing represents a distributed data-center whose computing devices are closer to the end users in order to provide real-time services that require a very low latency.

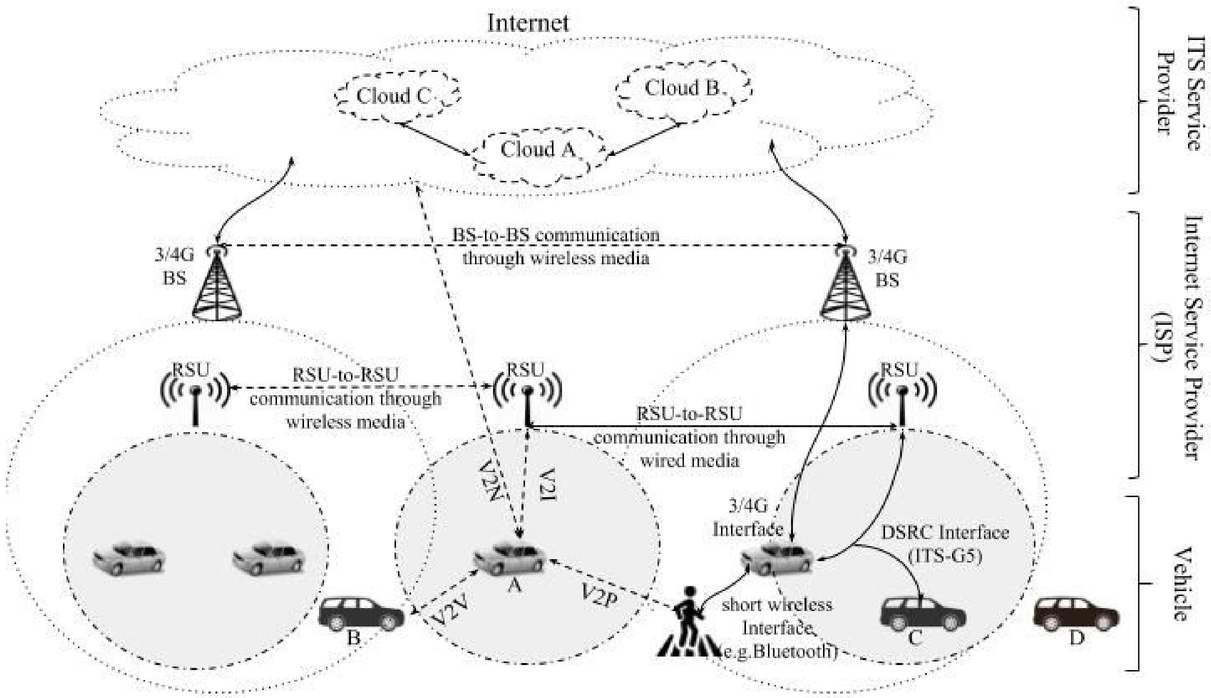

Fig 1 . Global communication architecture of an ITS

- *SDN Paradigm*

In a network, each device is composed of data plane and control plane. the main objective of data plane is the data forwarding , while the control decisions are taken by the control plane, for example, to decide from which interface each packet will be forwarded. In Conventional networks, the data plane and control plane are embedded in the same equipement and the decisions are taken locally by each device. However, SDN, software defined network, advocates the idea of taking control plan functions out of network forwarding elements, and relocating them on remote external devices called SDN controllers, as shown in Figure 2. The network intelligence and state are logically centralized [4].

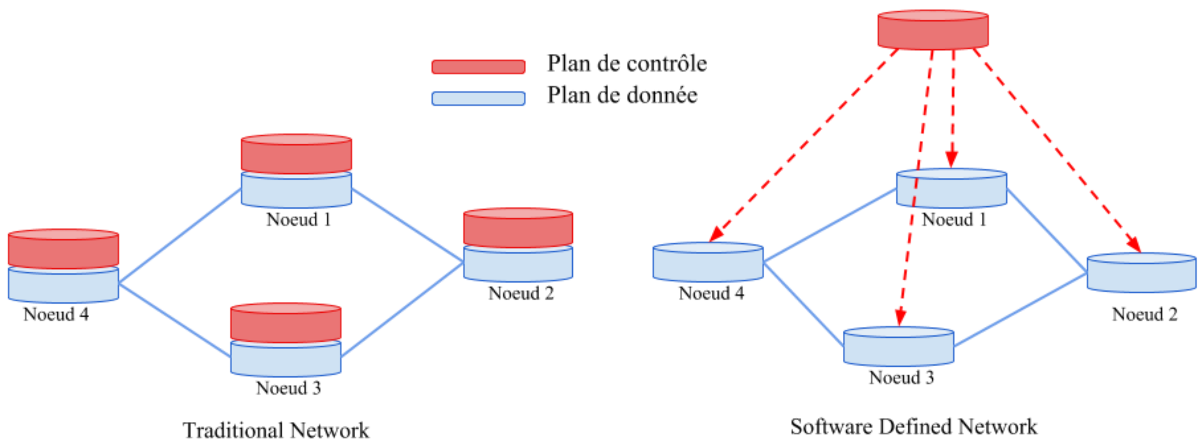

Fig 2 . SDN paradigm

The SDN controller communicates with the different network nodes using a southbound interface protocol, i.e. the widely used OpenFlow standard [5], while applications explicit their requirements to the SDN controller using Northbound Interface (API), as presented in Figure 3. In this architecture, the network nodes forward packets according to the rules installed on network devices by the SDN controller in a proactive or reactive manner.

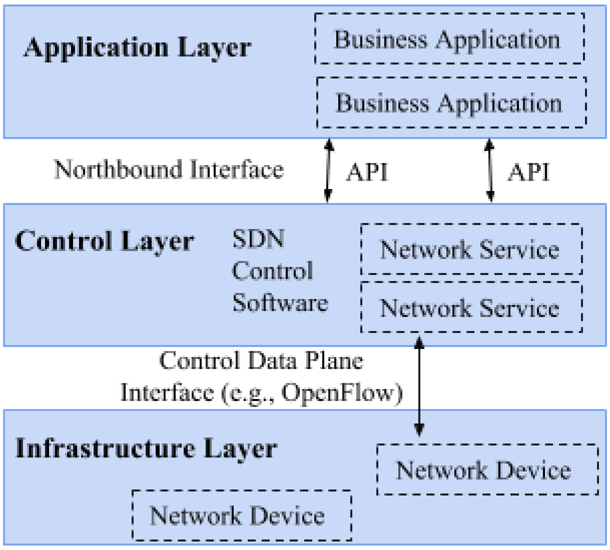

Fig 3 . Software Defined Network Architecture [4]

- *Topology discovery in SDN/Openflow networks*

A crucial piece of any SDN system is the topology discovery service, implemented by the controller, whose objective is to build and follow-up the network state and topology, from which, customized representations of the underlying network are exposed to network control functions, as shown on the figure below.

Network topology discovery covers node discovery and link discovery. In Openflow networks, node discovery is performed by the Openflow protocol while link discovery is provided by OFDP.

- Node Discovery: During its boot process, an Openflow switch establishes an Openflow session with its preconfigured SDN controller. During the session initiation, the switch transmits to the controller information about its identity (Datapath ID), its characteristics (e.g. capacity of the flow tables), its active ports and the functionalities supported through the exchange of "Features Request/Reply" messages. From there, a periodic Echo (Request / Reply message) procedure initiated by the controller is used to maintain the state of the node. Port-Status messages may be initiated by an OF switch to inform the controller in case a state port change.
- Link Discovery: From the information obtained during session initiation, the SDN controller is aware of the different active ports of each active switch. Following the OFDP procedure, the controller sends regularly LLDP (Link Layer Discovery Protocol) packets encapsulated in Openflow messages on a per active port basis. The Openflow message that encapsulates the LLDP packet (that conveys the switch ID and the concerned port ID) instructs the switch, via an Openflow rule, to relay the LLDP packet via the chosen port. When received by the adjacent switch, the LLDP is encapsulated in an Openflow message and sent back to the controller with its switch ID and the receiving port ID. By doing so, a link connecting both switches via the sending and receiving ports is discovered.

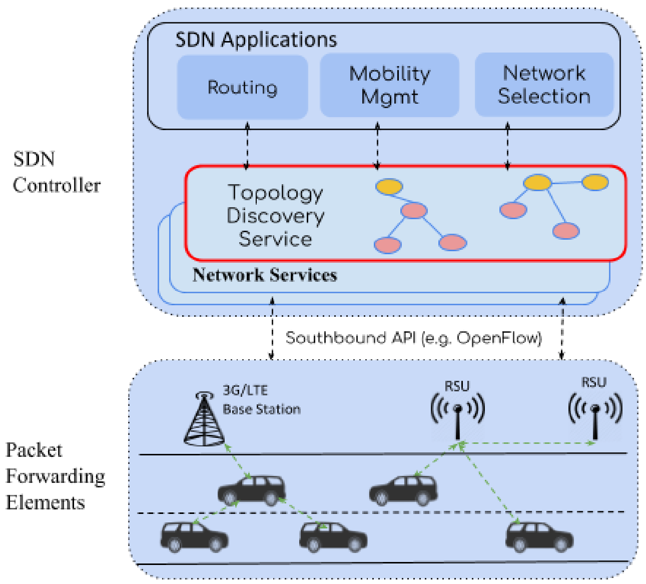

Fig. 4. Topology discovery service as a crucial component of SDVN

- *Limitations of Openflow/OFDP based topology discovery service for SDVN*

The Openflow/OFDP based topology discovery service was originally proposed with wired networks in mind. It clearly suffers from some limitations when it comes to wireless networks with mobile nodes. These are presented below.

❖ *Node Discovery*:

➢ Preconfigured SDN controller ID: The Openflow/OFDP discovery service relies on a continuous living Openflow session between each active switch and its associated controller(s). As shown in figure 5, in a vehicular network, a moving vehicle can leave the area where its initial SDN controller operates (Area A in figure 5) to get into a second area under the responsibility of another controller (e.g. zone B, in Figure 5), jeopardizing its Openflow session and consequently the topology discovery procedures described above. An SDN controller discovery mechanism and a handover mechanism are needed to ensure Openflow session survival and the exchange between controllers of vehicles related topology discovery attributes.

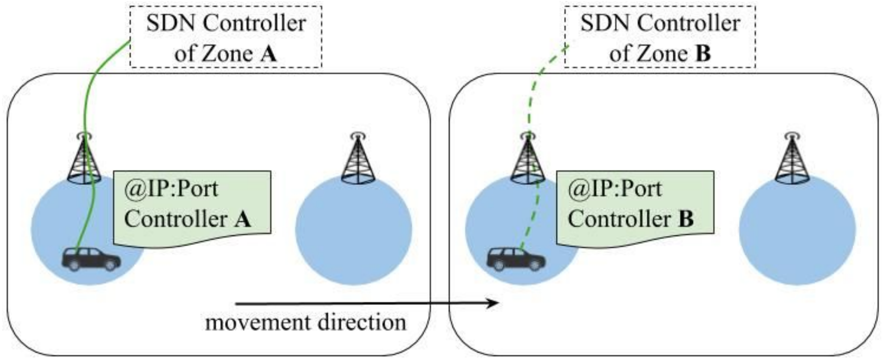

Fig. 5. SDN controller change caused by car movement

➢ Presence of non-openflow nodes or loss of connectivity with the controller: In both cases, the proposed service, does not report such nodes into the network topology built on the controller despite their presence and their ability to support and forward network traffic.

❖ *Link Discovery*:

- ➢ Presence of Heterogeneous links: In a vehicular network architecture, some Routing Nodes (RSUs and BSs) have wired as well as wireless links. Referring to [23] , the wired link discovery mechanism is a bit different from that used for wireless links. The Link discovery procedure must take into account the presence of different types of links in the same node and compute them differently.
- ➢ Link discovery with the presence of Openflow and non-Openflow nodes: The link discovery procedure of OFDP is operational when nodes are OpenFlow enabled. When the network includes non-openflow enabled nodes, [24] proposes that the SDN controllers use the Broadcast Domain Discovery Protocol (BDDP) instead of LLDP. The specificity of these messages is that their destination address is set to broadcast instructing the non-Openflow nodes to flood the packet (received from the transmitting adjacent node) until it is received by an openflow-compatible node where it is treated as a normal LLDP packet. Broadcasting topology discovery messages is prohibitive in terms of resource wastage in a vehicular network context.
- ➢ Scalability: Link discovery but also node discovery in Openflow/OFDP require respectively a periodic transmission of Openflow messages on a per active port and a per node basis. This periodicity should be adapted to the degree of node mobility. The higher the mobility, the lower the periodicity, and the more significant the discovery traffic. Even if a modification of the OFDP mechanism has been proposed in the literature in order to send a single packet to each switch[22], the generated traffic remains significant and, also, useless when the network does not experience any change.
- ❖ *Lack of useful node and link attributes:* Some vehicular network control functions (routing, mobility management, load balancing, network technology selection, etc.) may benefit from information related to nodes mobility (speed, position, direction) or wireless port characteristics (operating channel, transmission power, receiver sensitivity, etc.) as well as wireless link attributes (latency, reliability, etc.) to make informed control decisions. The Openflow/ODFP discovery service does not collect any of such attributes.

Field of exploration

In this section, We first describe the SDN proposed architecture and its benefits through a representative use case. Then, we assess the overhead of openflow/OFDP network discovery service in a vehicular network context, and we propose some directions to follow in order to design an efficient topology discovery service more suited to vehicular networks.

○ *Proposed architecture*

In our approach, we apply SDN to the global architecture of an ITS, including, not only the ad hoc network as proposed in [6], but also the RSU and cellular networks. This approach responds to the limitations of current architectures, by opening the road to the development of novel network control algorithms that take advantage of: (1) a vision of the state of all three above cited communication networks; (2) the ability to jointly control these networks; and (3) the knowledge of the environment in which vehicles evolve, which is derived from the data present in the cloud. For example, vehicles status information (position, direction, speed) can be used to predict the number of vehicles that will be present in a given region at a given time, allowing the estimation of the potential network load of a routing node (BS/RSU). Moreover, the dynamic nature of vehicular networks requires an adaptable network, SDN brings this flexibility to dynamically program the network according to network conditions. The SDN controller, which is a new component, is added to the architecture, as illustrated in Figure 6. Typically, We consider three controllers: one to manage the cellular network, another to manage the RSU-based network and a last one to coordinate between the different controllers. The main controller builds a global view of the communication infrastructure using the information sent by the controller of each network coupled with the data present in the cloud. It sends to each controller the global rules which describe the general behavior of the network, while the BS and RSU controllers define the specific rules to be installed in each network device. The communication between the SDN controllers is done using a specific interface known as East-West Interface e.g. AMQP, while the communication between the SDN controllers and the cloud is performed through specific APIs.

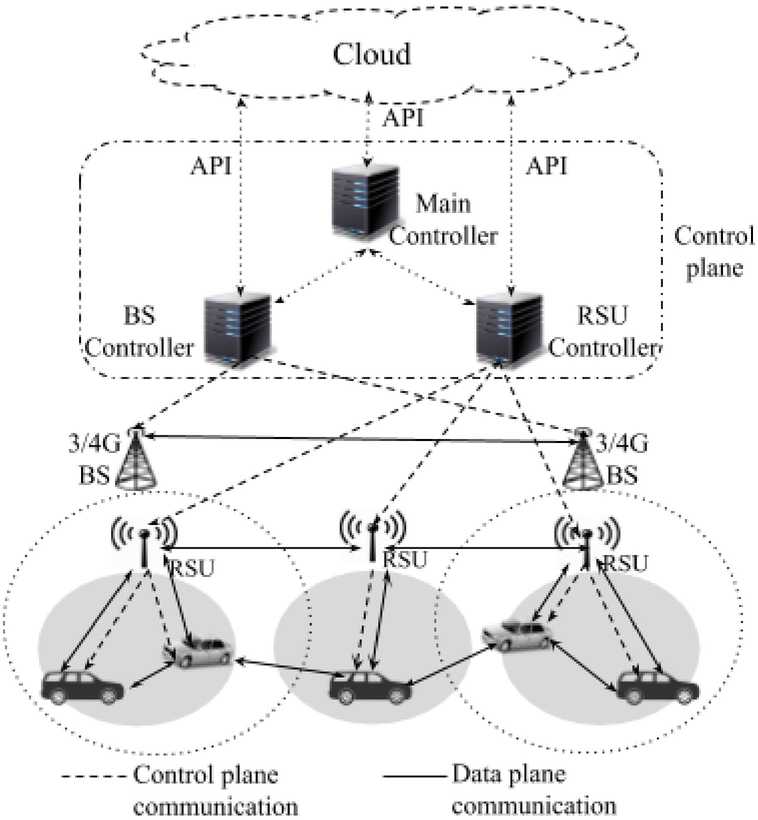

Fig 6 . The proposed SDN Architecture

Among the opportunities brought by this architecture:

- QoS aware routing with potential environmental inputs in a multi homed context: The SDN controller provides the best routing path according to the services requirements through efficient routing algorithms that are aware of the QoS requirements of each ITS services and the environment in which vehicles evolve, and which take advantage of the presence of several networks.
- Mobility Management: The global vision of the network allows the SDN controller to provide a better coordination of handover operations, moreover, the collected data present in the cloud allows it to predict the mobility of surrounding vehicles in order to anticipate some control operations.
- Enhanced QoS Management: The QoS management can be improved thanks to the fine grained as well as on-line programming capabilities offered by SDN. Efficient and dynamic QoS support can be achieved. Joint control algorithms (routing, topology control, etc.) can be developed to that end.
- Network Load Balancing and flow splitting.

○ *Use Case : Cooperative collision avoidance*

In order to show the benefits of our approach, we consider the "cooperative collision avoidance" service as one of the main safety services of "self-driving". The primary goal of this service, as its name suggests, is to help vehicles avoid crashes. The Vehicles continuously exchange information about their trajectories and status (position, velocity, direction). Therefore, each vehicle uses these information to compute the optimal collision avoidance actions and apply them in a cooperative manner. As specified in [2] and [3], the communications between vehicles must be made within a maximum latency of 100 ms, and shall not fail with a probability higher than 10-5 (the service tolerates the loss of one packet among 105 packets sent) which represents a significant challenge for the network. Let's refer the traffic of this service as (A1) and let's consider that the vehicles simultaneously execute other services (A2, A3) which have no stringent requirements on latency and reliability compared to A1.

In legacy vehicular ad-hoc networks [14], a priority based medium access scheme is defined with four priority levels named Access Categories (AC). Traffic is directed to the appropriate AC according to its priority and is statistically given priority in comparison to lower priority traffic originating from other véhicles. Despite this priority scheme, with no admission control (which is hard to set up in an ad-hoc context), there is no guarantee that the highest priority traffic (A1) will receive its requested network performance, particularly when entering a crowded area with high priority traffic from multiple vehicles contending for transmission.

With our proposed SDN based architecture, thanks to the centralized global view of the network, network resource allocations can be envisioned in order to provide the required QoS to each service, with the added ability to keep up with varying QoS requirements (e.g. referring to the figure below, a channel has been allocated dynamically to A1, while A2 and A3 share the same channel). Thanks to the multihomed nature of vehicles, these allocations can simultaneously apply to the different active connecting networks (cellular, RSU...) which paves the way to effective QoS provisioning algorithms.

There is another advantage that traditional architectures cannot achieve compared to our proposed architecture: the ability to reconfigure the network according to the changes of the network conditions (link disruption, failure of a network node, interference because of node density). We consider the case where the quality of wireless transmissions is degraded due to weather changes, which directly affects the transmission reliability and hence the functioning of the service.

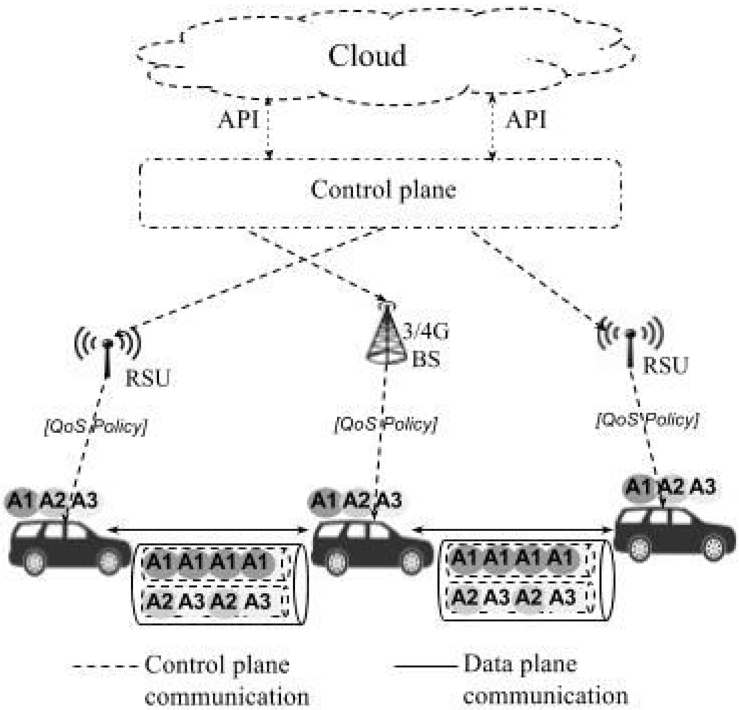

Fig 7 . Cooperative collision avoidance - scenario 1

In traditional architectures, the vehicle can detect these changes in a reactive manner by continuously monitoring the quality of wireless links (SNR, BER). However the decisions that it can take are predefined and limited to its local knowledge. For example, it can duplicate the critical traffic on two different channels, or change the transmission parameters (modulation technique...), which may not be relevant in some cases.

In our proposed architecture, the SDN controller can detect proactively these changes using data from the cloud. For example, the weather forecast in a given region can be exploited by network control applications to predict forthcoming changes in transmission conditions (wireless link quality), and take the appropriate actions to overcome this problem as, for example, duplicating the A1 traffic on two different paths in order to increase the packet delivery probability and consequently, provide A1 with the expected transmission reliability (see figure 8).

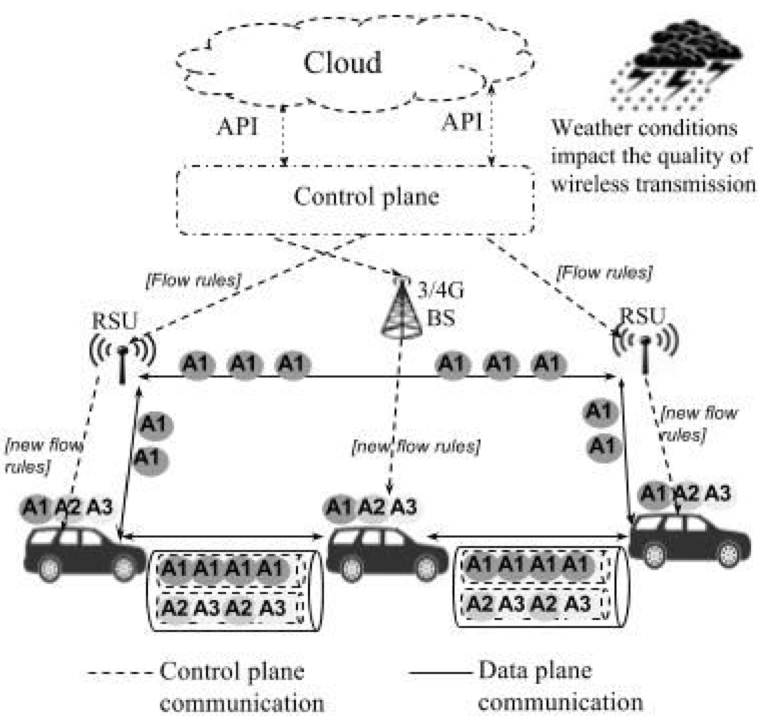

Fig 8 . Cooperative collision avoidance - scenario 2

○ *Experimental Results*

The goal of the experimentation is to demonstrate how the global network view established at the controller combined and enriched with information brought from the cloud enables a wiser and more efficient control of network behaviour with, at the end, a service with enhanced performance provided to the user. To that end, we consider the "Bird's eye view" scenario [2], in which the vehicles equipped with sensors such as cameras, radars, lidars can share their views with the neighboring vehicles. and show through evaluations how the SDN controller can leverage its global view of current and forthcoming (from the cloud) network loads to guide the node in the selection of the point of attachment to the network with the best expected performance.

A. Simulation description

In order to simulate our scenario, we use the MiniNet-WiFi emulator [15], which is an extension of MiniNet [16] to emulate 802.11 wireless networks programmable via SDN. In an area of 2000 × 2000 m2, we consider a network topology composed of 4 RSU entities, each with a communication range of 600 m, interconnected via wired connections; All are under the control of an SDN/Openflow controller.

The simulation plays out in two phases, a first one where the network has an average load, and a second one where the network is overloaded. In the first phase, we use 10 vehicles running an udp client-server session using the Iperf tool [17] generating a given network traffic. The vehicle "car1" (server) attached to the RSU1, stream a video traffic that vehicles "car2" and "car3" (clients) covered respectively by "RSU2" and "RSU4" consume as shown in the Figure 9. Table V shows the characteristics of this traffic. The Vehicles covered by several RSU entities, select the network to which they attach according to the power of the received signal "RSSI".

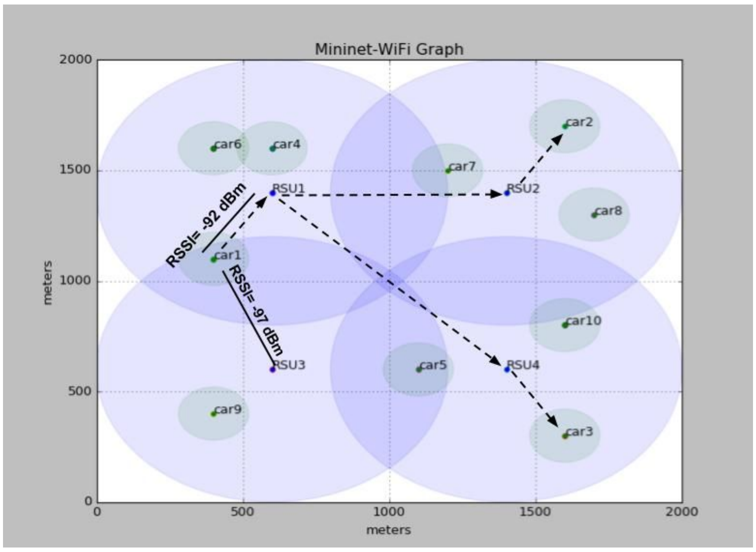

Fig 9 . Phase 1 : Network with an average load

In the second phase, we add 5 vehicles in order to overload the RSU1 to which "car1" is associated (as shown in Figure 10). This network condition change will be anticipated by the SDN controller in order to apply some network control actions in a proactive way. In this scenario, the idea is to prompt vehicle "car1" to attach to RSU3. New flow rules will be installed in the RSU entities by the SDN controller to support this new flow.

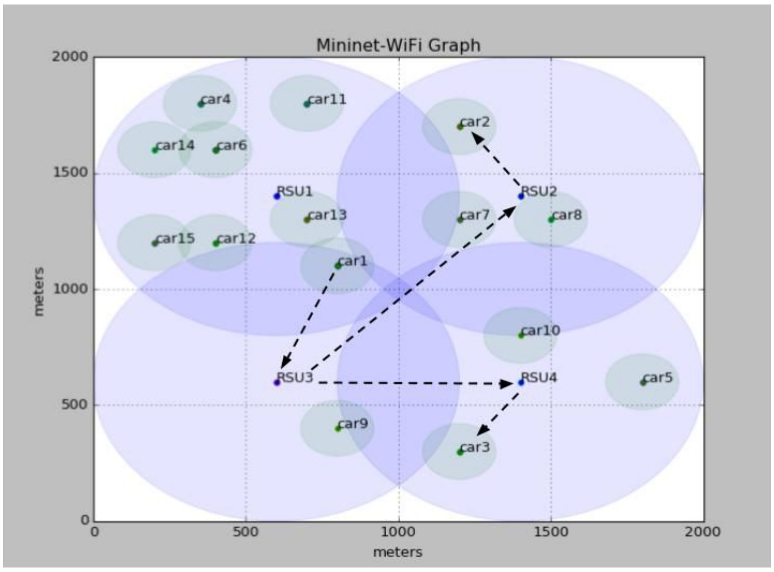

Fig 10 . Phase 2 : Network with an average load

B. Simulations results

Two metrics are considered in our evaluations :

- ➢ RTT : Round Trip Time, is the time required for a packet to travel from a specific source to a specific destination and back again. This metric is measured using the ping utility.
- ➢ PDR : Packet Delivery Ratio, This metric represents the ratio of the delivered packets to the destination to those produced by the source node, which represents the reliability of the transmission, this metric that we measure using the Iperf tool.

Figure 11 and 12 show the performance results for the communication between the "car1" and "car2" during the two simulation phases (average load and overloaded) and in both cases with (solid lines) and without (dotted lines) SDN controller presence. They respectively show the RTT and PDR as a function of time.

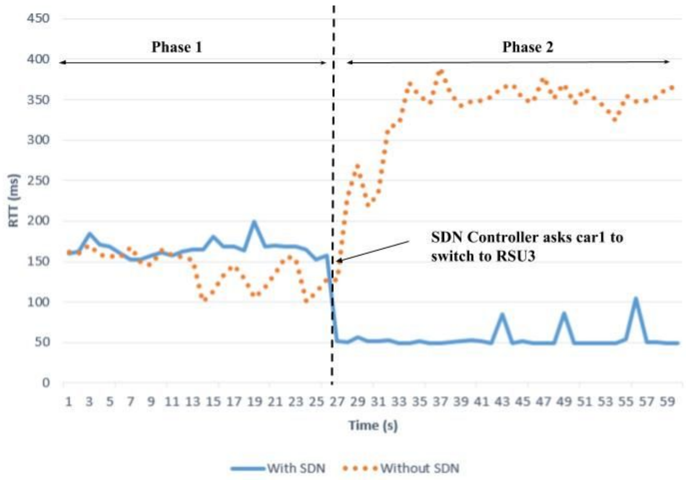

Fig. 11. RTT (Round Trip Time)

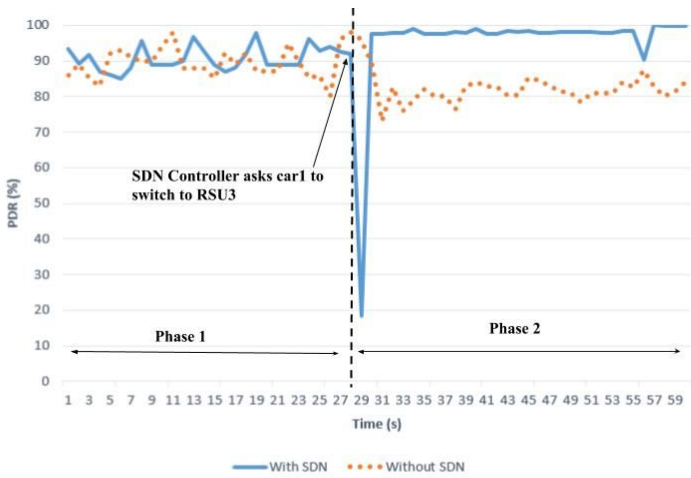

Fig. 12. PDR (Packet Delivery Ratio)

We notice that during the second phase, when the RSU1 entity covering the vehicle “car1” is overloaded, the network performance degrades, the average RTT increases from 140 ms to 353 ms and the PDR decreases by 10%.

With the SDN controller which triggers the change of the RSU entity to which vehicle "car1" is attached, we remark that the network performances is improved, the PDR remains around 98% and the average RTT is decreased by 87 ms. However, this handover action has a cost in terms of network performance as we notice that the PDR drops to 20% during the RSU entity change.

- *Assessing the overhead of openflow/OFDP network discovery service in a vehicular network context*

The goal is to analyze the impact of node mobility and node density, peculiar to vehicular networks, on the network traffic overhead generated by the Openflow/OFDP discovery service as well as the required computing resources at the Openflow controller. The rationale of these experimentations is to demonstrate that the increase of the number of vehicles and their velocity induces scalability issues.

A. Simulation Settings

The simulated network consists of 13 RSU entities interconnected via wired links, each with a communication range of 500 m covering a map area of 2000 x 2000 m2, all under the control of an SDN controller. The road network is a grid type network with each road segment of 500 m length (see Fig. 13). We use 10-30 vehicles (step of 5) with varied velocity 5-25 m/s (step of 5). Each vehicle starts at a random location and then moves with the configured velocity following the road network by randomly choosing a direction at each intersection.

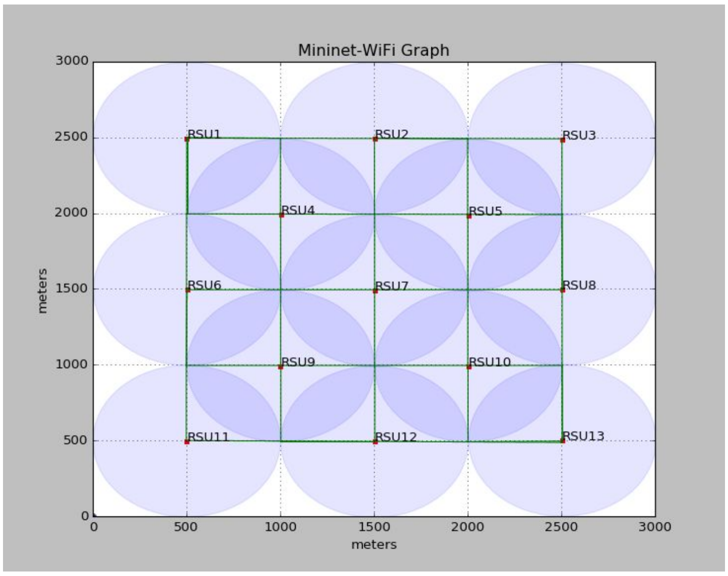

Fig. 13. Simulated vehicular network infrastructure

To the best of our knowledge and at the time of writing this paper, the only emulator that integrates real SDN/ OpenFlow Controller with programmable wireless network is Mininet-Wifi [15], and given that the current OFDP implementations are not compatible with the wireless link extensions added in MiniNet-Wifi, we used MiniNet considering the following assumptions: Each vehicle / RSU entity represents an OpenFlow switch, and the vehicle movement from one RSU to another results in the removal of the link established with the first RSU, and the creation of a new link with the second RSU. In addition, the existence of a vehicle in another's communication range enforces the creation of a V2V link between them.

In our scenario, we are interested in the dynamicity of the topology (link addition and removal caused by node mobility as explained above). This network model was implemented, using the MiniNet API and we used Floodlight (v1.2) [25] SDN Controller with its implementation of the Openflow/OFDP topology discovery service. The SDN controller and the mininet emulator (v 2.2.2) [16] are running on separated Ubuntu 16.04 VMs with 2 Go RAM and a dual core CPU (Intel ® Core (TM) i5-4310U 2 Ghz). We measure the CPU utilization and the amount of control data exchanged between the controller and all active nodes. Each set of simulations is averaged over 3 runs, each running for 5 minutes.

B. Simulation Results

Two kind of simulations were first realized. Firstly, all the vehicles move with the same speed (10 m/s) and their number varies from 10 to 30 with a step of 5. In a second phase, we keep the same number of vehicles (20) and their speed varies from 5 to 25 m/s with a step of 5. Figures 14(a,b) and 14(c,d) show respectively the variation of the CPU utilization and network overhead as a function of the number and the speed of vehicles.

We notice that the CPU utilization increases in a linear way and proportionally to the number of vehicles. This is explained by the fact that OFDP operates in "one-to-one communication" mode (controller to each node) which multiplies by the number of controlled nodes the number of actions performed by the SDN controller to discover and maintain the topology. This also holds for network overhead that depends mainly on the number of messages exchanged with each node and the number of created links.

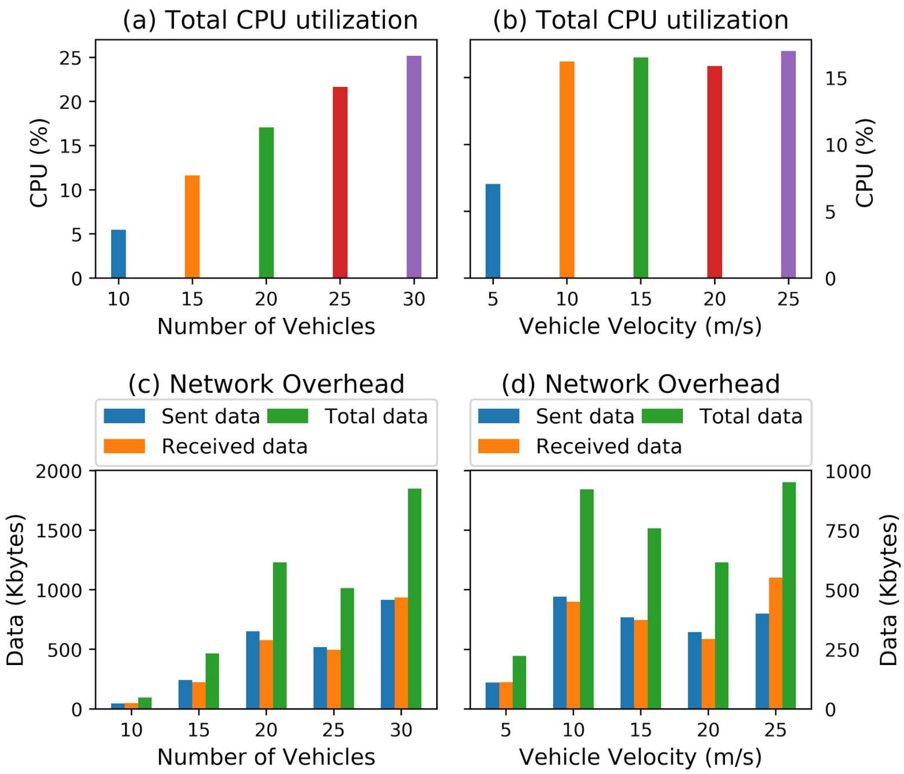

Fig. 14. Simulated vehicular network infrastructure

As expected, when increasing the speed of vehicles, the CPU utilization and the network overhead increase as well. They do not grow linearly as a function of vehicle velocity. This is

explained by the fact that the speed variation induces a longer variation in the number of added or removed links. In fact, this number increases according to the vehicle speed for V2I links but remains random for the V2V links that depend mainly on the location of each vehicle at a given time. We also notice that the variation in the number of vehicles (density) has a more significant impact in terms of scalability than their mobility.

- *Design guidelines*

In order to avoid the node discovery limitations mentioned in the previous section, a dynamic discovery of SDN controllers appears more suited in a vehicular network context. Another approach is the use of a paradigm that separates network node identification and localization, such as the LISP protocol.

In addition, to avoid the scalability issue of existing discovery approaches, we need to rethink the link discovery process. Indeed, the use of a ”one-to-one communication” approach presents scalability issues (as demonstrated in the previous section). One research direction could be the adoption of a ”one-to-many” communication approach, by defining a vehicle cluster with a Cluster Head that handles the communication with the controller, instead of communicating with each node separately. By doing so, the network overhead between the controller and the network nodes is reduced. However, additional traffic will be added between the vehicles. Hence, an efficient mechanism is needed for aggregating the information to be sent to the controller while reducing the load and maintaining a consistent discovery.

Another point to consider in the design of a topology service is the LLDP message sending interval. This frequency is statically configured according to the dynamicity of the network: the more dynamic the network, the higher its frequency. However in a vehicular network, the dynamicity of network varies depending on several parameters (the density and speed of vehicles (vary according to the time of day (day, night, peak hours)) and also as a function of the environnement (urban /suburban areas)). An interesting direction will be to have an adaptable frequency depending on the environment. On the other hand, considering the use of an event-based approach instead of a periodic approach can be more interesting, as far as only information about added/ removed links will be notified to the controller.

Finally, some mobility-related data (velocity, trajectory, etc.) required by the network control functions can be recovered from the ITS service provider clouds instead of being sent through the SDN control plane. Indeed, these data are already collected by these service providers on behalf of the development of their proposed services.

Retained solutions and acquired knowledge

In this work, we proposed a new architecture based on the SDN paradigm combining the RSU and the cellular networks in order to efficiently support the QoS requirements of the envisioned ITS services, combined with the data collected in the cloud, we argue that novel network control algorithms can be devised to improve the efficiency and QoS capabilities of vehicular networks.

We demonstrated through use case and simulation the manner how the data collected by the ITS providers which can be used in order to develop proactive and effective network control algorithms. This requires collaboration between ISP and ITS service providers.

We have also demonstrated that the “de facto” OpenFlow based topology discovery service is not adapted for Software Defined Vehicular Networks and we proposed some directions to follow in order to address the highlighted limits and to design an efficient topology discovery service more suited to vehicular networks.

The proposed architecture is a first step towards a software defined vehicular network (SDVN). However, the main components of the SDN architecture (Southbound Interface, basic services) need to be adapted/extended to vehicular network context. In this purpose, we are now working on the development of "QoS estimation service", which is a complementary topology discovery service, whose objective is to estimates the potential network state and topology. This view is exposed to network control functions allowing them to make more "intelligent" decisions in a proactive way.

From the architectural point of view, The SDN controllers placement which is crucial for the proposed architecture remain a challenge to be addressed.

References

[1] ITS Definition, ETSI, available [Online]. Available at : http://www.etsi.org/images/files/ETSITechnologyLeaflets/IntelligentTransportSystems.pdf.

[2] 5G-PPP, 5G Automotive Vision, white paper, October 20, 2015.

[3] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions, ETSI TR 102 638 V1.1.1 (2009-06)

[4] Open Networking Foundation, Software-Defined Networking: The New Norm for Networks, ONF White Paper April 13, 2012.

[5] Open Networking Foundation, OpenFlow Switch Specification, version 1.3.4 (protocol version 0x04) March 27,2014.

[6] I. Ku, Y. Lu, E. Cerqueira, R. Gomes, M. Gerla, “Towards Software-Defined VANET: Architectures and Services”, Mediterranean Ad Hoc Networking Workshop, 2014.

[7] D. K. N. Venkatramana, S. B. Srikantaiah, J. Moodabidri, “SDN-enabled connectivity-aware geographical routing protocol of VANETs for urban environment”, IET Netw., Vol. 6 Iss. 5, pp. 102-111, 2017.

[8] X. Ji, H. Yu, G. Fan, W. Fu “SDGR: An SDN-based Geographic Routing Protocol for VANET”,IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016.

[9] Y. C. Liu, C. Chen and S. Chakraborty, "A Software Defined Network architecture for GeoBroadcast in VANETs," 2015 IEEE International Conference on Communications (ICC), London, 2015, pp. 6559-6564.

[10] Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration;Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols, ETSI TS 102 636-6-1V1.1.1 (2011-03).

[11] N. B. Truong, G. M. Lee and Y. Ghamri-Doudane, "Software defined networking-based vehicular Adhoc Network with Fog Computing," 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, 2015, pp. 1202-1207.

[12] fA. Kazmi, M. A. Khan and M. U. Akram, "DeVANET: Decentralized Software-Defined VANET Architecture," 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW), Berlin, 2016, pp. 42-47.

[13] Frank Försterling, “Electronic Horizon How the Cloud improves the connected vehicle”, Wien, 2015

[14] Felipe Domingos da Cunha, Azzedine Boukerche, Leandro Villas, Aline Carneiro Viana, Antonio A. F. Loureiro. Data Communication in VANETs: A Survey, Challenges and Applications. [Research Report] RR-8498, INRIA Saclay; INRIA. 2014.

[15] Fontes, R. R., Afzal, S., Brito, S. H. B., Santos, M., Rothenberg, C. E. “Towards an Emulator for Software Defined Wireless Networks“. In EAI International Conference on Software Defined Wireless Networks and Cognitive Technologies for IoT. Rome, Italy, Oct 2015

[16] Mininet Emulator. Available at https://www.mininet.org, accessed on November 2017

[17] Iperf Tool. Available at https://iperf.fr/ accessed on November 2017

[18] ONISR, Observatoire national Interministériel de la sécurité routière, “Bilan de l'accidentalité de l’année 2016”

[19] SUMO, Simulation of Urban MObility. Available at http://sumo.sourceforge.net/ , accessed on November 20117

[20] S. Khan, A. Gani, A. W. A. Wahab, M. Guizani, and M. K. Khan, “Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art,” IEEE Communications Surveys Tutorials, vol. 19, pp. 303–324, First quarter 2017.

[21] A. Azzouni, R. Boutaba, T. M. T. Nguyen, and G. Pujolle, “softdp: Secure and efficient topology discovery protocol for SDN,” CoRR, vol. abs/1705.04527, 2017.

[22] F. Pakzad, M. Portmann, W. L. Tan, and J. Indulska, “Efficient topology discovery in OpenFlow-based software defined networks,” Computer Communications, vol. 77, pp. 52–61, mar 2016.

[23] L. Chen, S. Abdellatif, P. Berthou, B. Nougnanke, and T. Gayraud, “A Generic And Configurable Topology Discovery Service For Software Defined Wireless Multi-Hop Network,” in 15th ACM International Symposium on Mobility Management and Wireless Access, (Miami, United States), p. 4p., Nov. 2017

[24] C.-P. C. F.-F. A. Ochoa-Aday, L., “Current trends of topology discovery in openflow-based software defined networks,” 2015.

[25] Floodlight project, Available at http://www.projectfloodlight.org/floodlight/. Accessed: 2018-08-01