Software Defined Networking for Emergency Traffic Management in Smart Cities

Nikbakht Bideh, Pegah; Paladi, Nicolae; Hell, Martin

Published in:
Vehicular Ad-hoc Networks for Smart Cities

DOI:
10.1007/978-981-15-3750-9_5

2020

Software Defined Networking for Emergency Traffic Management in Smart Cities

Pegah Nikbakht Bideh[1], Nicolae Paladi[1,2], and Martin Hell[1]

[1] Lund University, Department of Electrical and Information Technology, Sweden
{pegah.nikbakht_bideh, nicolae.paladi, martin.hell}@eit.lth.se
[2] RISE Research Institutes of Sweden

Abstract. Vehicle traffic management is becoming more complex due to increased traffic density in cities. Novel solutions are necessary for emergency vehicles, which despite growing congestion must be able to quickly reach their destination. Emergency vehicles are usually equipped with transmitters to control the traffic lights on their path and warn other vehicles with sirens. Transmitters are operated manually and, like sirens, have a limited range. Smart cities can make use of novel network models to facilitate traffic management. In this paper, we design a traffic management application leveraging Software Defined Network controllers for traffic preemption. The proposed application leverages the logical centralization of the SDN control plane to improve traffic management. Results from evaluating the application under five different scenarios indicate that emergency vehicles can reach their destination much faster, with very little effect on the surrounding traffic.

1 Introduction

Road traffic congestion caused by increasing traffic has become a major problem in big cities. However, increasing connectivity creates opportunities to radically improve road traffic management.

Smart transport systems, such as Vehicular Ad-hoc Networks (VANETs), have the potential to fulfil the needs of traffic management in big cities. VANETs enable communication between vehicles as well as between vehicles and fixed Road Side Units (RSU); this includes vehicle to vehicle, vehicle to RSU and RSU to RSU communications. RSUs are fixed access points along the roads which help communication among vehicles. VANET infrastructure is usually deployed by automobile manufacturers to provide services for vehicle owners [5].

Safety and traffic management are the main attributes of VANETs since they can directly affect the lives of people travelling on the road. Vehicles acting as nodes in VANETs can form a vehicular network without prior knowledge of each other. There are two types of applications available in VANETs: namely *comfort applications* and *safety applications* [15]. Traffic prioritization is unnecessary in comfort applications - messages can be delivered to the destination in the order they arrived in the network. Comfort applications include traffic information systems, weather information, and gas station information. Safety applications are intended for emergency and unsafe situations - messages for safety applications have higher priority in the network. Safety applications include emergency vehicle warning, SOS services, and post-crash warnings.

Traffic management and safety applications are usually operated by government agencies [14]. In safety application scenarios - such as emergency situations - a vehicle receives complete coverage of all vehicles in the network using broadcasting [16]. This can lead to the broadcast storm problem [6] in VANETs. This occurs when nearby vehicles send a large number of broadcasts, causing packet loss due to collisions.

To increase the efficiency of VANETs and address issues such as the broadcast storm problem, support for Software Defined Networking (SDN) was introduced in VANET [9]. SDN-based VANET helps address the limitations and challenges of traditional VANET systems. A core advantage of an SDN-based VANET is the global overview of the network which it provides. This can be used to manage the entire network communications more efficiently. Managing the overall network load through a central controller, as done in SDN-based VANETs, can help making more informed routing decisions. But in simple VANETs, because of the focus on shortest path routing, the traffic can easily become unbalanced. Moreover, SDN-based VANETs offer additional advantages: 1) There is no need to configure each network device manually, 2) the path-recovery latency decreases, and 3) the programmability of the network improves through external applications.

In this paper, we propose an emergency traffic management application for SDN controllers. The application provides an efficient and fast route for emergency vehicles, moving inside the VANET, in case of emergencies. Our application combines two main approaches to reach its goal. First, which is the main contribution of this paper, we control traffic lights to create a clear and fast route for emergency vehicles. This technology is called *Emergency Vehicle Signal Preemption* and we aim to improve the performance of emergency signal preemption through an SDN controller. Second, we improve warnings to other vehicles in the network on the path of emergency vehicles through a targeted delivery warning. This allows targeted vehicles to leave the path as quickly as possible (e.g., keeping to the right lane). These are building blocks of our system for emergency traffic management with the help of SDN.

We explain emergency signal preemption and traffic lights control in Section 2. In Section 3, we explain SDN and its properties. We present the designed application in Section 4, and we demonstrate the implementation of the system in Section 5. We evaluate the application in Section 6. We present related works in Section 7. Finally, we conclude the paper in Section 8.

2 Background

Traffic lights usually have two control modes: *fixed time* and *dynamic control* [1]. In the fixed time mode, a fixed time is assigned to traffic lights and after a given time, they change colour. A dynamic traffic light instead has a detector which communicates with the traffic light and informs it about real-time traffic conditions, such as the number of cars on the road. Some other traffic lights are equipped with wireless LAN that can send information, including waiting time until next colour change, to the approaching vehicles, allowing drivers to adjust their driving pattern.

More complex dynamic control can be coordinated or synchronized [4]. In synchronized control, all changes are done at the same time. These are only used in special cases or older systems. Coordinated lights are usually controlled from a master controller and can change lights in cascade order so platoons of vehicles can proceed through a series of green lights. The controller is usually placed on a corner of an intersection. It receives information from the detector and changes the traffic signal based on this information.

While dynamic control traffic lights perform better than fixed time lights they nevertheless have some drawbacks. A controller is needed in each intersection, and failures in controllers or detectors are difficult to troubleshoot.

Traffic lights work on the concept of phases. Phases of traffic lights are groups of directions of movements existing at an intersection. Traffic lights follow the predefined phases patterns repeatedly, and when a traffic light receives a green signal, if the current phase is not green it shifts the phase to green immediately.

Traffic signal preemption [12] allows manipulating traffic signals in the path of an emergency vehicle. Signal preemption enables emergency vehicles to move more quickly and more safely since emergency vehicles can move on a green path and avoid traffic congestion. The signal preemption system uses a receiver mounted on the traffic light and a transmitter mounted inside the emergency vehicle. In case of emergency, the transmitter inside the emergency vehicle can be activated and causes compatible traffic lights in the path of the vehicle to change the colour to green immediately. Traffic signal preemption can work well in emergencies but it needs to be manually activated and only works within a limited distance from the traffic light. The existence of a central controller, using SDN, can significantly improve the performance of signal preemption since it has knowledge about the path of the emergency vehicle in advance and the control is independent of the vehicles' distance to the traffic light.

3 Software Defined Networking

The explosive growth of data traffic has made the limitations of traditional networking obvious. In traditional networking, each network device has local control and a local data plane. Devices, such as switches and routers, are often vendor- and application-specific, resulting in a complex (re-)configuration and management. The control plane and data plane inside networking devices, which are respectively responsible for policy definition and traffic forwarding, reduce networking flexibility and hinder the evolution of the networking infrastructure.

There are many definitions of SDN and the most well-accepted one is from the ONF (Open Networking Foundation) organization. There it is defined as "The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices" [11]. Around this network architecture, SDN has four main characteristics [8].

- **Separation**
  The control and data planes are decoupled. Control functionality is removed from network devices, so they act as simple packet forwarding entities. The control plane is responsible for policy definition and management while data plane devices forward data according to installed rules.
- **Central Control**
  Control logic is moved to an external entity (SDN controller), or a more complex control system with multiple controllers. It now has central control and management of multiple devices and an overall view of the network situation, allowing better policies.
- **Virtualization**
  Devices can be physical or virtualized and implementation details can be abstracted. Devices can be configured to act as an arbitrary network device.
- **Open Interfaces and Programmability**
  SDN-based devices now have open interfaces and software applications running on top of the controller which could control the devices by high-level programming without concern about the trivial details of the devices, bringing easier configuration and updates. This is a fundamental characteristic of SDN and the main value proposition.

4 SDN-based Emergency Traffic Management Application

In this section, we describe the main contribution of this work, an SDN-based emergency traffic management application. This novel approach for emergency signal preemption makes use of the global network view of an SDN controller to create a green corridor for emergency vehicles, without increasing the latency for other vehicles on the road. The proposed approach can replace the traditional signal preemption method and radically simplify the current approach.

4.1 System Components and Communications

The components of our designed system are the SDN controller, SDN nodes, and SDN RSUs [15]. The SDN controller is used to control the whole network and the traffic lights. SDN nodes, in our system, are vehicles that are constantly moving, making the system more dynamic in comparison to stationary SDN nodes. An SDN RSU is a physical device attached to roadsides and responsible for communication between vehicles, traffic lights and with the SDN controller.

A vehicle can interact with its environment using the following types of communication:

- **V2V (Vehicle to Vehicle)** in which both parties are vehicles.

– **V2I (Vehicle to Infrastructure)** is a type of communication in which one part is a vehicle and the other part is RSU.
– **I2I (Infrastructure to Infrastructure)** in which different RSUs communicate with each other or RSU is communicating to SDN controller.

4.2 System Design

The general overview of the system is depicted in Fig. 1. The SDN controller has a global overview of the network, e.g, emergency vehicles' current position, their destinations and the paths they are taking to their destinations (the available shortest path). Moving vehicles and traffic lights are connected to their nearest RSUs. Here we make the assumption that the network has high coverage and there are RSUs available all around the network. The system is constructed from two sub-systems described below.

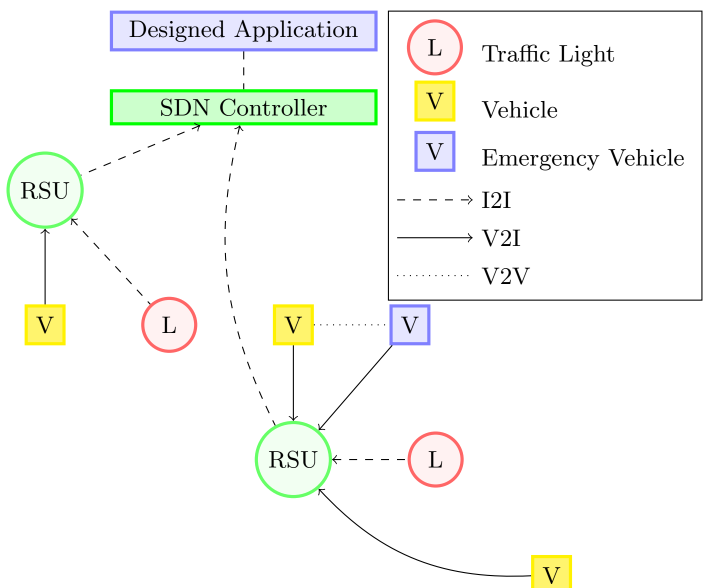

Fig. 1. General overview of SDN-based emergency traffic management

(i) Vehicle Traffic Preemption
This sub-system aims to preempt traffic lights on the path of emergency vehicles. When an emergency vehicle approaches a traffic light on its path, if the phase of the traffic light is not green, the SDN controller sends a change phase command to the traffic light via the nearest RSU. This command changes the phase of the traffic light and turns it to green sooner than the specified phase state. Later, the traffic lights return to their earlier phase

pattern. Vehicle traffic preemption helps emergency vehicles to move on the green path.

Considering that other vehicles on the road are controlled by drivers, control commands issued over a network cannot be enforced. However, this feature may become available in autonomous vehicles. Vehicle traffic preemption can be complemented with control commands to vehicles on the road to help provide a green corridor for moving emergency vehicles. However, traffic control commands to autonomous vehicles are outside the scope of this paper.

(ii) Target Vehicle Notification

Target vehicle notification aims to notify other available vehicles on the path of emergency vehicles. Vehicles on the path of emergency vehicles are identified by the SDN controller and a "notify message" is sent to them. After receiving the "notify message" by a vehicle, if there are two lanes available on the road, the driver keeps right and slows down to make the road clear. There is no need to send traffic lights information to other vehicles on the road since they just stop if the light is red or pass if it is green. Target vehicle notification helps in providing a congestion-free path for emergency vehicles.

5 Implementation

Based on the system model described in Section 4, we here describe the implementation of the SDN-based traffic management application. As mentioned in Section 3, in SDN networks the data plane is separated from the control plane. We used the RYU SDN controller for our implementations since it supports almost all OpenFlow versions and has good documentation available [17].

Along with the SDN controller, we used a network emulator to create a network of virtual hosts, switches, controller, and the links between them. We used the Mininet [2] emulator for this purpose. It provides a simple testbed to develop OpenFlow applications, enables complex topology testing, and provides an easy way to achieve correct system behaviour.

Furthermore, we used SUMO (Simulation of Urban MObility) [10] version 1.1.0 to simulate road traffic in a smart city. SUMO is an open-source and portable road traffic simulator which has been designed to handle large road networks. SUMO uses the TraCI module (Traffic Control Interface) to retrieve attribute values of simulated vehicles, traffic lights and to manipulate their behaviour online. SUMO can be connected to Mininet via a Python API. Hosts, taking the role of vehicles are created in Mininet based on the number of vehicles in SUMO. Then, Mininet connects to RYU, our SDN controller. In the emergency traffic management application, live data such as vehicle state, position, accident spot, and emergency vehicles' location need to be sent to the SDN controller. The controller should also be able to send packets to different vehicles, traffic lights, and devices such as RSUs. Since Mininet does not support data packet communication with RYU, in our implementation we instead used sockets to send data packets between RYU and Mininet. Whenever data need to be sent, a socket is open between different devices and also between Mininet and

RYU. Sockets stay open and devices can use them for upcoming data transfers if needed.

For the emergency traffic management application, vehicles on the road and traffic lights along with RSUs have been simulated in SUMO. Since SUMO can not be connected directly to an SDN-based controller, we used Mininet between SUMO and the RYU controller. Thus, for each available vehicle and traffic light in SUMO, we add a node and map them to the relevant hosts in Mininet. There are many typologies such as Single Switch, Tree and Linear available in Mininet. We selected Linear topology and based on this topology the relevant number of switches is added in Mininet and hosts are connected to the switches.

As illustrated in Fig. 2, both SUMO and Mininet form the infrastructure layer in our implementation. The mapping and communication between SUMO and Mininet are also shown in Fig. 2. Then, the infrastructure layer connects to, and is controlled by, the RYU controller and our designed emergency traffic management application runs on top of RYU.

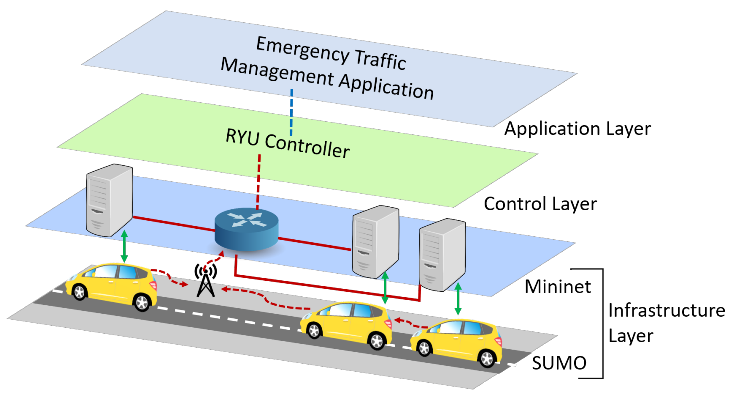

Fig. 2. SUMO and Mininet mapping in emergency traffic management application

In the course of a simulation, whenever an accident occurs, a vehicle identifier along with a lane identifier is sent to the SDN controller through the nearest RSU. This information can be extracted from SUMO using the TraCI module. The SDN controller knows all vehicles on the road and also their positions. Then, the controller finds the ambulance closest to the accident spot and sends the ambulance the position of the accident.

In SUMO, the shortest path algorithm is used to find the shortest route to the destination. Thus, an ambulance takes the available shortest path to the accident spot. While the ambulance is moving, it connects to its nearest RSUs and its position information is sent to the controller online. When the ambulance approaches a traffic light, the SDN controller checks the phase of the light. If it is not green, the SDN controller sends a "change phase" command to the traffic light through the nearest RSU. Also, the SDN controller knows which vehicles are moving on the ambulance route and sends notification messages to them. After receiving a notify message, if there are two lanes available, the vehicle keeps right, slows down and lets the ambulance take over. If there is only one lane available, the vehicle stops at the roadside and the ambulance can take over.

Based on the severity of the accident, other emergency vehicles such as a fire truck, police or another ambulance can be sent to the accident spot. In Section 6, we test the system with different emergency vehicles.

The SDN controller has a global view of the network and signal preemption can be achieved more efficiently by allowing earlier preemption and reducing delays of emergency road traffic. The preemption is done through the RSUs instead of sending the green signal via a transmitter mounted inside the vehicle.

6 Performance Evaluation and Results

To evaluate the performance of our proposed emergency traffic management application, we defined five different scenarios of varying complexity.

- **Scenario 1:** One accident spot, one ambulance drives to the accident spot, and 10 other vehicles on the road.
- **Scenario 2:** One accident spot, one ambulance drives to the accident spot, and 100 other vehicles on the road.
- **Scenario 3:** One accident spot, one ambulance, one fire truck, and one police all drive to the accident spot, and 100 other vehicles on the road. Firetruck and police face each other in an intersection.
- **Scenario 4:** One accident spot, two ambulances drive to the accident spot and they face each other in an intersection, and 100 other vehicles on the road.
- **Scenario 5:** Two accident spots, two ambulances each drives to a different accident spot, and 100 other vehicles on the road.

We implemented and simulated each of the above scenarios in SUMO. We used a part of the New York city map (Chelsea, Manhattan) in our implementations and there were 63 separate roads available in the map. The map was exported from OpenStreetMap [3] and imported into SUMO. Among these roads, 21 of them were one-way roads with an ending edge. To avoid such traps, only 42 out of the 63 roads were used in our simulation. We simulated the first, second and last scenario 42 times with the 42 different starting road positions for ambulances. Since in the third and fourth scenarios the emergency vehicles' paths must intersect, these starting points were not randomized.

In the application, we encoded a set of traffic priority rules, based on traffic rules. First, when two crossing emergency vehicles at the intersection have different priorities, the one with higher priority should cross the intersection first. For example, if a firetruck and a police car face in an intersection, the firetruck crosses first. Second, when two crossing emergency vehicles at the intersection have the same priority we prioritise the vehicle whose traffic light was due to turn to green sooner. For example, if ambulance 1 and ambulance 2 face an intersection and the remaining time to the green phase for ambulance 1 is 5 seconds and for ambulance 2 is 10 seconds, ambulance 1 has priority. The SDN controller has knowledge about different types of emergency vehicles and their priorities and the encoding rules can be easily applied.

To evaluate the performance of the emergency traffic management application, we measure the arrival time of the different emergency vehicles before and after applying the SDN-based traffic management application. The arrival times before applying our application are shown in Table 1, while the arrival times resulting from applying our application can be found in Table 2. Comparing the arrival times in the two tables shows that with our SDN-based application, the arrival time is significantly decreased for the emergency vehicles.

Table 1. Arrival time to accident spot without using emergency traffic management application

Scenarios	Arrival time to Accident Spot (Seconds)			
	Ambulance 1	Ambulance 2	Firetruck	Police
Scenario 1	297.49	-	-	-
Scenario 2	310.31	-	-	-
Scenario 3	178.30	-	416.40	235.60
Scenario 4	267.70	356.80	-	-
Scenario 5	357.90	412.80	-	-

Table 2. Arrival time to accident spot with using emergency traffic management application

Scenarios	Arrival time to Accident Spot (Seconds)			
	Ambulance 1	Ambulance 2	Firetruck	Police
Scenario 1	159.50	-	-	-
Scenario 2	168.90	-	-	-
Scenario 3	160.90	-	246.60	272.10
Scenario 4	175.90	181.00	-	-
Scenario 5	183.10	193.20	-	-

In our simulations, starting points and destinations of other vehicles on the road are fixed points and they are equal for both cases (before and after applying our application). We do not apply the randomization of starting points for normal vehicles on the road. To make sure the emergency traffic application does not increase latency for other vehicles on the road, we calculated the Mean, Median, Variance and Standard Deviation of arrival times of all vehicles to their destination before and after applying emergency traffic management application. These values are shown in Table 3 and 4.

Table 3. Mean, Median, Variance and Standard Deviation of arrival time of all vehicles to their destination without using emergency traffic management

Scenarios	Without using emergency traffic management			
	Mean	Median	Variance	Standard Deviation
Scenario 1	227.87	208.8	7293.02	90.57
Scenario 2	309.60	282.2	16369.62	128.70
Scenario 3	292.70	282.2	12108.11	110.73
Scenario 4	289.47	280.05	10774.48	104.45
Scenario 5	313.24	285.45	16162.16	127.87

Table 4. Mean, Median, Variance and Standard Deviation of arrival time of all vehicles to their destination with using emergency traffic management

Scenarios	With using emergency traffic management			
	Mean	Median	Variance	Standard Deviation
Scenario 1	223.32	212.5	4451.12	69.97
Scenario 2	315.82	286.65	16952.22	130.94
Scenario 3	284.29	282.6	8812.15	94.46
Scenario 4	306.28	294.7	14205.18	119.88
Scenario 5	318.03	295.3	15084.80	123.54

As illustrated in the tables, the difference between the mean arrival times before and after using our emergency traffic management application is relatively small. Thus, applying our application does not have a significant impact on the latency of other vehicles on the road.

7 Related Work

Several SDN-based approaches to emergency traffic control are available from earlier work. In [13] the authors proposed an SDN-based algorithm to control emergency traffic. In case of an emergency, the SDN controller calculates the shortest route to the destination area and gives priority to emergency vehicles to reach their destination as quickly as possible. The controller also diverts normal traffic to alternative routes to avoid congestion. The authors tested the proposed method by using Mininet, with the cars represented as packets and the traffic lights emulated as OpenFlow switches. This is not a very realistic emulation since it is not possible to emulate real-time vehicular traffic behaviours such as slow down, take over, and change of lights using Mininet. Instead, in our emulations, we used SUMO in which all vehicular behaviors can be simulated easily.

In [7], the authors proposed a solution for adaptive traffic management for emergency services in smart cities. The goal was to reduce the latency of emergency vehicles with minimum disruption to the regular traffic. The designed traffic management system consists of a set of traffic management controllers, each of them controlling traffic in a specific area. One of the advantages of this method is the ability to authenticate emergency vehicles (hospitals, fire stations, etc) upon receiving emergency notifications. On the other hand, the system needs many traffic management controllers and they need to coordinate with each other, increasing the complexity of the system. In our designed system, there is only a central controller available and there are RSUs all around the city. Each RSU can communicate with the SDN controller and there is no need to coordinate RSUs since they are directly controlled by the SDN controller. Also, in our system it is not necessary to program different controllers, so the complexity of the system decreases.

In [18], an SDN-enabled hybrid emergency message transmission architecture on the Internet of Vehicles (IoV) was proposed. The authors applied SDN to a vehicular network to obtain rapid and reliable transmission of emergency messages. In the proposed method, the SDN controller sends the emergency message to relevant RSU switches. Then the RSU switches take relevant action based on their flow table. The designed architecture can only help in forwarding emergency messages in the vehicular network and it does not help emergency vehicles itself to reach their destination faster. Our designed system aims to improve emergency vehicular traffic management by providing a green and congestion-free path for emergency vehicles.

8 Conclusions and Future Work

In this paper, an SDN-based emergency traffic management application was designed to improve signal preemption through a central controller. The application can provide a green path for emergency vehicles to their destination. It can also be used to notify other vehicles on the road about an approaching emergency vehicle. We define different scenarios and compute the arrival time for vehicles. The results show that by using our designed SDN-based emergency traffic management application, we can clear the road for emergency vehicles and decrease their arrival times. Moreover, our application does not have a significant impact on the latency of other vehicles. Thus, having a central SDN controller with a global network view can reduce the latency of emergency vehicles with little negative impact on other vehicles. On the other hand, a central controller can also be a Single Point of Failure. This problem can to some extent be mitigated by having a backup SDN controller available. Deploying such an SDN-based VANET traffic management application in smart cities requires collaboration between government agencies and automobile manufacturers.

A future work could be to consider autonomous vehicles on the road and extend the application to control commands instead of notification messages.

Acknowledgements This paper was partially supported by the Swedish Foundation for Strategic Research, grant RIT17-0035, and partially supported by the Wallenberg Autonomous Systems and Software Program (WASP).

References

1. How Traffic Light Control Systems Work. https://www.autoevolution.com/news/how-do-traffic-light-control-systems-work-41839.html (June 2019)
2. Mininet. http://mininet.org/ (July 2019)
3. Open Street Map. https://www.openstreetmap.org (June 2019)
4. Traffic signal design terminology. https://www.traffic-signal-design.com/terminology_main.htm (June 2019)
5. Bhatia, A., Haribabu, K., Gupta, K., Sahu, A.: Realization of flexible and scalable VANETs through SDN and virtualization. In: 2018 International Conference on Information Networking (ICOIN). pp. 280–282. IEEE (2018)
6. Chen, R., Jin, W.L., Regan, A.: Broadcasting safety information in vehicular networks: issues and approaches. IEEE network **24**(1), 20–25 (2010)
7. Djahel, S., Salehie, M., Tal, I., Jamshidi, P.: Adaptive traffic management for secure and efficient emergency services in smart cities. In: 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops). pp. 340–343. IEEE (2013)
8. Kreutz, D., Ramos, F., Verissimo, P., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: A comprehensive survey. arXiv preprint arXiv:1406.0440 (2014)
9. Ku, I., Lu, Y., Gerla, M., Gomes, R.L., Ongaro, F., Cerqueira, E., et al.: Towards software-defined VANET: Architecture and services. In: Med-Hoc-Net. pp. 103–110 (2014)
10. Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic Traffic Simulation using SUMO. In: The 21st IEEE International Conference on Intelligent Transportation Systems. IEEE (2018), https://elib.dlr.de/124092/
11. ONF: Software-Defined Networking (SDN) Definition. https://www.opennetworking.org/sdn-definition/ (July 2019)
12. Paniati, J.F., Amoni, M.: Traffic signal preemption for emergency vehicle: a cross-cutting study. US Federal Highway Administration (2006)
13. Rego, A., Garcia, L., Sendra, S., Lloret, J.: Software defined networks for traffic management in emergency situations. In: 2018 Fifth International Conference on Software Defined Systems (SDS). pp. 45–51. IEEE (2018)
14. Salmon, P.M., Read, G.J., Stevens, N.J.: Who is in control of road safety? A STAMP control structure analysis of the road transport system in Queensland, Asustralia. Accident Analysis & Prevention **96**, 140–151 (2016)
15. Shafiq, H., Rehman, R.A., Kim, B.S.: Services and security threats in SDN based VANETs: A survey. Wireless Communications and Mobile Computing **2018** (2018)
16. Yousefi, S., Mousavi, M.S., Fathy, M.: Vehicular ad hoc networks (VANETs): challenges and perspectives. In: 2006 6th International Conference on ITS Telecommunications. pp. 761–766. IEEE (2006)
17. Zhu, L., Karim, M.M., Sharif, K., Li, F., Du, X., Guizani, M.: SDN controllers: Benchmarking & Performance Evaluation (2019)
18. Zhu, W., Gao, D., Zhao, W., Zhang, H., Chiang, H.P.: SDN-enabled hybrid emergency message transmission architecture in internet-of-vehicles. Enterprise Information Systems **12**(4), 471–491 (2018)