

# In-Vehicle Software Defined Networking: An Enabler for Data Interoperability

Khalid HALBA<sup>\*†</sup>, Charif MAHMOUDI<sup>\*‡</sup>

<sup>\*</sup> National Institute of Standards and Technology, Gaithersburg, Maryland, USA

<sup>†</sup> Grenoble Alpes University, Grenoble, France

<sup>‡</sup> Algorithmic, Complexity and Logic Laboratory, University of Paris-Est Creteil, Creteil, France

{khalid.halba,charif.mahmoudi}@nist.gov

## ABSTRACT

As future transportation systems evolve, new in-vehicle network designs are required to handle the heterogeneous data generated by different Electronic Control Modules (ECUs). Enabling interaction between these data sources can trigger innovation and the emergence of new smart features significantly impacting upon security and riders experience. The interoperability between the ECUs is of high value in the context of autonomous transportation systems. Indeed, it enables different technologies to collaborate for achieving complex tasks. Without this interoperability, features like radar system connected to the Media Oriented Systems Transport bus (MOST) cannot trigger the electronic stability control connected to the Controller Area Network (CAN). These features allow the car to mitigate a high-risk situation using existing modules. In this work, we propose a Software Defined Network (SDN) approach that enables in-vehicle data sources interoperability that allows ECUs to share a medium. The benefits of the proposed approach are backed by the implementation of a relevant use case and the generation of simulation results.

## CCS Concepts

• Information systems → Information Integration  
• Information systems → Process control • Networks → Cyber-physical networks

## Keywords

In-vehicle Networks; Software Defined Networking; Interoperability.

## 1. INTRODUCTION

Taking advantage of the different and heterogeneous data sources in an in-vehicle network is of high importance as this renders the implementation of advanced vehicle features possible. However, in-vehicle networks differ in terms of the technology and the communication medium used [1]. This makes it impossible for the different data sources to interoperate. In fact, the Controller Area Network (CAN), Local Interconnect Network (LIN), Flex Ray, Media Oriented Systems Transport bus (MOST), or the Ethernet Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

*ICISDM '18*, April 9–11, 2018, Lakeland, FL, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-6354-9/18/04...\$15.00

<https://doi.org/10.1145/3206098.3206105>

buses are currently implemented in an island-like fashion. This means that these buses have no support for interoperability. As a result, they operate independently as illustrated in Figure 1. Such implementation limits the scope of possible use cases that can be driven by allowing interoperability between these buses.

In the context of autonomous vehicles, new features that require a combination of control messages and environment data cannot be implemented without a complete network redesign. The example of lane centering is of high relevance in this context: It requires a combination of environment sensors that make sure the vehicle is not bypassing the lane edges, plus a mechanism that communicates this information to steering, stability and braking systems [2]. Obstacle mitigation is another advanced feature which relies on specialized cameras and sensors that analyze the surroundings of the car whilst in motion which then sends information to braking, steering, and stability systems [3] to correct the cars direction or to limit its movement.

In this paper, we propose a solution that enables network interoperability between heterogeneous data sources (i.e.: the different Electronic Control Modules within a vehicle, for example: the Antilock Braking System and the Electronic Stability Control). We make two contributions to address this problem: i) we propose an SDN in-vehicle network architecture that enables interoperability between different automotive modules, and ii) we introduce a solution for adapting the protocols of different technologies during an interoperability use case. In other words, we design an SDN-based in-vehicle Network with an objective in mind: enhancing interoperability between Electronic Control Units regardless of the technologies and protocols used. The design we propose leverages advances in networking paradigms and network programmability capabilities to establish interoperability mechanisms. The remaining of this paper is organized as follows: Section 2 explains how legacy in-vehicle networks operate and we highlight some of their key technical specifications. Next, we describe SDN properties and we explain the importance of the Time Triggered Ethernet technology and how it fits industrial applications. In Section 3 we introduce the network design and data adaptation elements addressing the interoperability problem. Section 4 demonstrates a use-case where we show the usability of our processes upon which we implement and test our contributions.

Finally, we give highlights on our current and future work.

## 2. RELATED WORK

In this section, we discuss legacy in-vehicle network technologies and their properties and use cases. Next, we explain how Software Defined Networking enables interoperability at the network level. Finally, we discuss the importance of Time Triggered Ethernet

and how it represents a powerful backbone for handling future automotive networks.

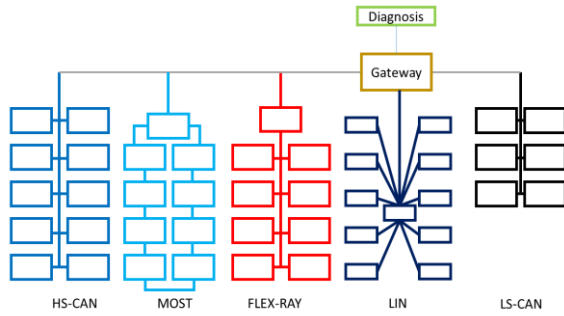


Figure 1. Different automotive buses composing a legacy in-vehicle network

## 2.1 Legacy In-Vehicle Networks

Modern day vehicles rely on a number of legacy buses to transport the control and multimedia traffic [1]. The widely used CAN

is responsible for carrying critical control messages. CAN is a double wire bus with resistance terminations protecting the carried signals from interference. CAN ECUs are connected to the bus through a communication controller and can send traffic with speeds up to 1 mbps, features such as steering, engine control, and braking are all handled by legacy High Speed CAN (HS-CAN) Technology, other

less critical features such as Body Control can be achieved using the Low Speed CAN (LS-CAN) [4]. The LIN is another legacy medium used for less critical operations. Functions such as door locking, mirrors and window control are managed by this technology and the speeds required for such features do not exceed 20 kbps [5]. Another bus which is mainly used for infotainment purposes is MOST, it is an optical ring capable of carrying data with speeds up to 157 mbps [6], allowing the bus to handle up to 64 channels ranging from audio and camera streams, to other functionalities like GPS. Flex Ray (FXR) is a relatively recent technology with a capacity of 10 mbps. It was designed to replace critical functionalities handled by the CAN bus as it offers more guarantees in term of real time requirements and speed. This bus enables advanced control features such as drive by wire, adaptive cruise control and active suspension [7]. Lastly, ethernet is used for diagnosis purposes via its OBD II interface [1]. These legacy networks are isolated (Figure 1) which limits the scope of possible usecases and features that can be driven by allowing them to interoperate and exchange data.

Another bus which is mainly used for infotainment purposes is MOST, it is an optical ring capable of carrying data with speeds up to 157 mbps [6], allowing the bus to handle up to 64 channels ranging from audio and camera streams, to other functionalities like GPS. Flex Ray (FXR) is a relatively recent technology with a capacity of 10 mbps. It was designed to replace critical functionalities handled by the CAN bus as it offers more guarantees in term of real time requirements and speed. This bus enables advanced control features such as drive by wire, adaptive cruise control and active suspension [7]. Lastly, ethernet is used for diagnosis purposes via its OBD II interface [1]. These legacy networks are isolated (Figure 1) which limits the scope of possible usecases and features that can be driven by allowing them to interoperate and exchange data.

## 2.2 Software Defined Networking

Networking is subject to continuous evolution with the emergence of new paradigms such as Software defined networking (SDN). SDN is a technology that allows separation of control and data planes and brings network programmability to the realm of advanced data forwarding mechanisms. This allows features such as network traffic filtering, fine grained forwarding, QoS, and failover [8] to be implemented and leveraged. In this paper, we take

advantage of the SDN network programmability to enable interoperability between ECU data sources and we leave the leverage of QoS and Failover for a future work. SDN will enable flows from heterogeneous data sources to be exchanged following

applications and rules running in the SDN controller as illustrated in Figure 2.

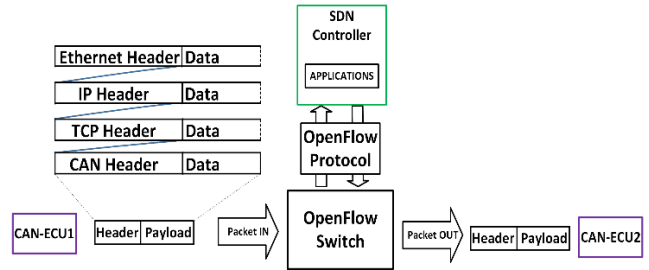


Figure 2. An ECU message as handled by the SDN controller and the OpenFlow capable Switch

## 2.3 Time Triggered Ethernet

Industrial applications such as transportation systems must respect specific requirements in terms of real time constraints, fault tolerance, allocated bandwidth, and resistance against electromagnetic interference. Ethernet protocols and hardware have been subject to a significant evolution in recent years. Ethernet flavors presently include options that are compatible with automotive requirements. Time Triggered Ethernet (TTEthernet) is a technology that achieves real time specifications [9]. It has been tested in automotive and aeronautic applications and was proven a viable solution to replace current legacy automotive buses [10]. Leveraging the benefits of TTEthernet will enable new possibilities in the automotive field.

## 3. DESIGN FOR INTEROPERABILITY

In this section we introduce an SDN solution that leverages network programmability and the benefits of TTEthernet to offer a reliable mean for interoperability between the ECUs. Such integration has been proven viable and promising for real time systems [11]. Next, we introduce a universal adapter that enables the adaptation of messages from a protocol to another when an interoperability scenario takes place.

### 3.1 Network Interoperability Design

We introduce the first element of our contribution: a network design that enables network interoperability between heterogeneous data sources.

Our proposed design is composed of three different planes as described in Figure 3: 1) the interoperable sources plane (ISP) composed of the heterogeneous data sources i.e.: ECUs, and environment sensors. These sources differ in terms of the protocols used and the nature of the data delivered. 2) The Interoperable Network Plane (INP) – this sits between the ISP and the Interoperable Control Plane (ICP), it delivers connectivity to the ISP components through OpenFlow Switches, and leverages a Time Triggered Ethernet Backbone capable of carrying the traffic generated by the ISP sources while respecting real time and industry requirements. 3) The Interoperable Control Plane: defines the set of advanced applications that require interoperability between heterogeneous data sources, it also populates OpenFlow switches with flows necessary to achieve these interactions. Figure 4 shows an internal perspective of the elements of each plane, each representing a component of the design: the ICP can be represented as an *Application* component which incorporates and manages the complex features domain. The *Application* component relies on the *Network* (INP) component to provide

mechanisms for data forwarding and protocols adaptation. Finally, the *Data Sources* component features the ISP plane. It represent the different ECUs and sensors exchanging heterogeneous messages.

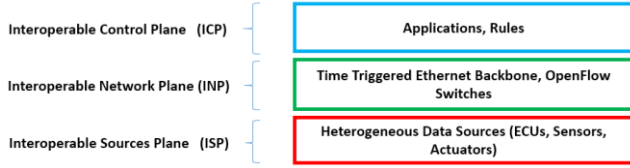


Figure 3. Planes composing the Network Interoperability Design

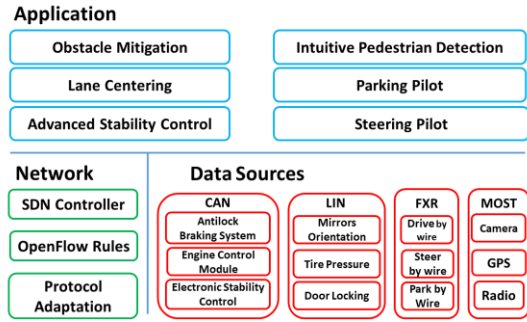


Figure 4. An inside perspective of the elements composing the Network Interoperability Design.

### 3.2 Protocols Interoperability: Introducing a Universal Protocol Adapter

After introducing the network interoperability design, we describe the second element of our contribution: a universal protocol adapter. We want to enable heterogeneous data sources to interact regardless of the protocols used to carry their information.

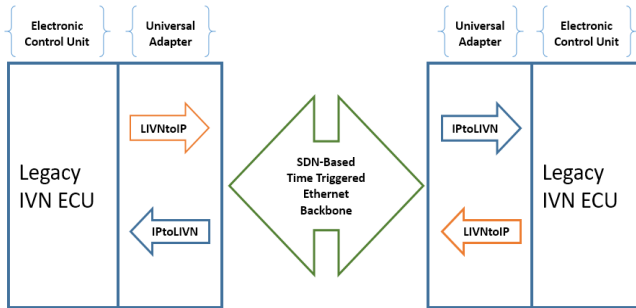


Figure 5. The role of the Universal Protocol Adapter at the ECUs edges.

As a part of our design, the protocol adapter is composed of the flowing elements: 1) an IP to Legacy-In-Vehicle-Network adapter (IPtoLIVEN adapter) and 2) a Legacy-In-Vehicle-Network to IP adapter (LIVENtoIP adapter). A representation of this adapter is described in Figure 5. These adapters allow heterogeneous sources to share information regardless of the legacy protocol used. Data messages generated at the legacy ECU is first adapted to the Time Triggered Ethernet backbone using the LIVENtoIP adapter component, the latter encapsulates the message in an Ethernet frame and adds IP and TCP headers. The packet then travels through the backbone and gets forwarded as defined in the

applications and rules running in the SDN controller. Finally, at the receiving node, the IPtoLIVEN component adapts the signal to the protocol used by the receiving node. It de-encapsulates the packet and extracts the original message sent by the source. Assuming ECUs have built in mechanisms to decipher the payload and react to the semantics of the different received messages, this solution enables two heterogeneous data sources to exchange information at the network level.

## 4. EXPERIMENTS

In this section we introduce the use case we will leverage to simulate our proposed solution, we define metrics to enable comparison between different designs, and we run simulations and we discuss the results.

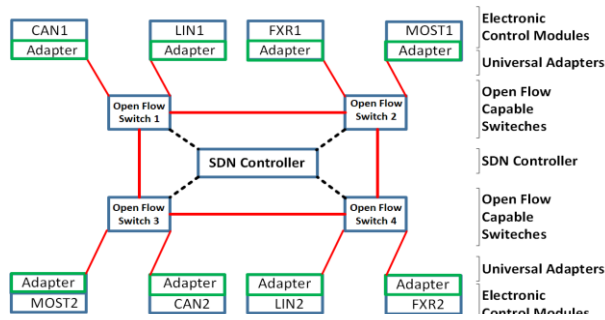
Regarding the use case, we simulate advanced autonomous features that require interoperability between different data sources such as the obstacle detection [12] and mitigation [13] feature. This use case fits for assessing our interoperability contributions. Diverse in-vehicle components interact to enable a self-driving car to mitigate an obstacle. Provided that a (MOST) special camera and front radar generate environment and awareness data of the surrounding of the vehicle, once an obstacle is detected, these data sources send a message through the time triggered Ethernet backbone. This message is then received and processed by the CAN buses responsible for Electronic Stability Control, Steering, and Anti-Lock Braking System, enabling the car to avoid a physical damage situation. We simulate features such as obstacle detection and mitigation, and we assess the interoperability in the proposed design (SDIVN) versus the LIVEN design.

For this end, we define the *Interoperability Cost (C)* as the time it takes for a message exchanged between two heterogeneous data sources to be adapted from one protocol to another. We do not include propagation delay. The interoperability cost in a Legacy in vehicle Network (Equation 1) can be expressed as the sum of the *Cost per Integration Hop (Ci)*, the latter represents the cost in terms of time to adapt a message from a protocol to another, which is done at the protocol adapter level.

$$C = \sum_{i=0}^n C_i (1)$$

$$C = 2 * C_i (2)$$

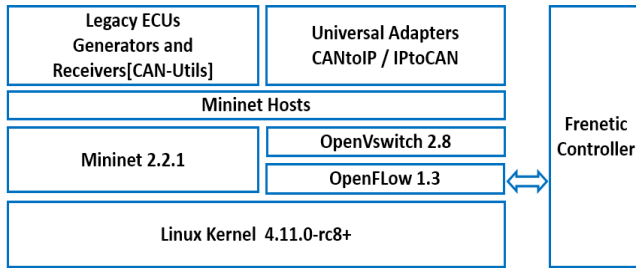
The design below (Figure 6) represents an implementation example of our use case. The obstacle detection feature for example can be implemented as follows: An environment perception component (camera or radar) located at MOST2 ECU generates data that reflects whether an obstacle represents a physical damage to the vehicle or not. In the case of a critical situation, the MOST2 ECU sends a message to both: the CAN1 and FXR1 ECUs where movement correction components are located (Antilock-Braking system (ABS), Steering Control, Stability Control). These streams of data represent the interactions between heterogeneous data sources that occur when implementing an advanced autonomous feature.



**Figure 6. 8 ECUs exchange in-vehicle messages and leverage the SDIVN**

We have implemented the simulation environment using Mininet that sits on a patched Linux Kernel (4.11.0-rc8+) to support CAN-Utils Package namespaces [14]. Additionally, we have built the protocol adapters based on TCP/IP Sockets and CAN-Sockets [15].

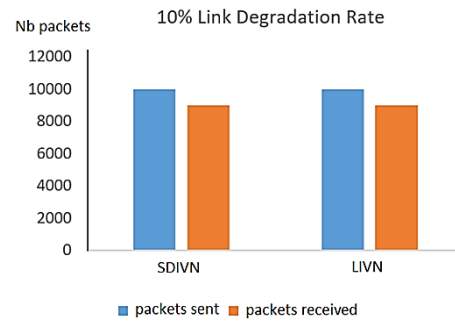
For the SDN implementation, Frenetic [16] was our pick for the controller as it ensures formal verification of the applications and properties [17]. A representation of the software stack we have leveraged is described in Figure 7.



**Figure 7. Developed and Open-Source components used in the simulation.**

After extensive simulations, we have generated interesting results that back the advantages of our proposed design. Comparing the Legacy In-Vehicle Network (LIVN) against the proposed Software Defined In-Vehicle Network (SDIVN) shows that the proposed design does not add overhead or degradation when implementing data exchanges that don't require protocol adaptation. Figure 8 represents the 8 ECUs when exchanging 10,000 messages during the interoperability use case, at the reception we measure how many packets were dropped. The results show that under the same link degradation of 10%, Legacy in-vehicle Network (LIVN) and Software Defined In-Vehicle Network are subject to the same message drop rates. Figure 9 shows the aggregated Data rates when 8 sources of the same technology exchange data messages. The data rate aggregation is also showing similar results for both LIVN and SDIVN.

Next, we assess the protocol adaptation in both designs: for a legacy in-vehicle network, adding a complex feature comes with the price of implementing more integration hops and adapters. Therefore, this leads to a linear increase of the interoperability cost. Simulation results show that there is a linear increase of ~1ms for each added feature in the legacy network to support new features that require protocol adaptation.

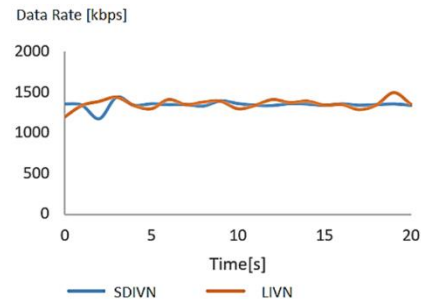


**Figure 8. Link degradation influence on Packet Drop: Number of packets received compared to packets sent.**

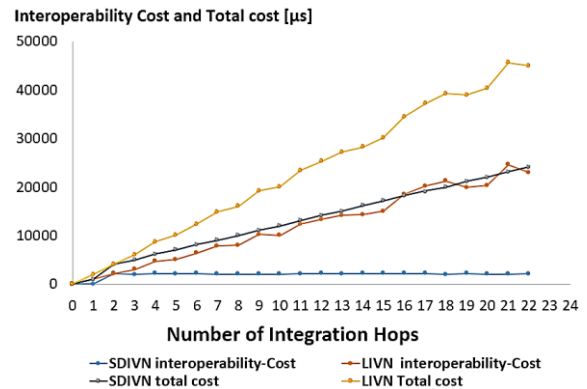
The proposed solution dubbed (SDIVN) shows promising results as adding more features to the design does not impact the interoperability cost. Every added feature in the proposed design consumes around ~2ms, which represents the cost of the adaptation at the edges of the two ECUs exchanging data (Equation 2). Figure 10 summarizes the results of the protocol adaptation in the case of a legacy in-vehicle network versus the network we have proposed. Total cost is also influenced by the interoperability cost:

$$Total\ Cost = Propagation\ delay + C$$

A higher Total Cost means that some critical features might not be achieved in a timely manner: this is intolerable in the automotive context. Figure 9 shows that the Total Cost in the case of the SDIVN is roughly equal to the interoperability cost in the LIVN case which proves the advantage of our design over the traditional design.



**Figure 9. Data rate: Aggregated data rates of ECUs exchanging messages.**



**Figure 10. Interoperability Cost: SDIVN vs Legacy Network.**

## 5. CONCLUSION AND FUTURE WORK

The proposed design has significant merits in terms of improving interoperability between heterogeneous automotive data sources. This will enable the implementation of a new spectrum of advanced

autonomous features. We intend for future work to explore the benefits of our solution in terms of improving in-vehicle network failover. At present, we are implementing fast failover mechanisms

and the preliminary results are promising. Additionally, we intend to conduct an in-depth business analysis. Formal verification is also

a domain of interest in our research as automotive applications require safety properties to be verified. Finally, a fractional factorial

statistical analysis of the experiments we have conducted is ongoing, the goal is to define which in-vehicle networks parameters

have the most influence level on certain properties of interest.

### DISCLAIMER

Any mention of commercial products or organizations is for informational purposes only; it is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the products identified are necessarily the best available for the purpose.

The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

## 6. REFERENCES

- [1] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi and L. Kilmartin, "Intra-vehicle networks: A review," *IEEE Transactions on Intelligent Transportation Systems*, pp. 534-545, 2015.
- [2] X. Xu, J. W. Grizzle, P. Tabuada and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Transactions on Automation Science and Engineering*, 2017.
- [3] M. N. Hasan, S. Didar-Al-Alam and S. R. Huq, "Intelligent car control for a smart car," *International Journal of Computer Applications*, vol. 14, pp. 15-19, 2011.
- [4] L.-m. Zhang, L.-h. Tang and Y. Lei, "Controller area network node reliability assessment based on observable node information," *Frontiers of Information Technology & Electronic Engineering*, pp. 615-626, 5 2017.
- [5] J. Takahashi, Y. Aragane, T. Miyazawa, H. Fuji, H. Yamashita, K. Hayakawa, S. Ukai and H. Hayakawa, "Automotive Attacks and Countermeasures on LIN-Bus," *IEEE Transactions on Automation Science and Engineering*, vol. 25, pp. 220-228, 2017.
- [6] I. A. Grzempa, MOST: the automotive multimedia network, Franzis Verlag, 2012.
- [7] T.-Y. Lee, C.-W. Kuo and I.-A. Lin, "High performance CAN/FlexRay gateway design for in-vehicle network," in *Dependable and Secure Computing, 2017 IEEE Conference on*, 2017.
- [8] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Communications Surveys & Tutorials*, vol. 16, pp. 1617-1634, 2014.
- [9] H. Kopetz, A. Ademaj, P. Grillinger and K. Steinhammer, "The time-triggered ethernet (TTE) design," in *Object-Oriented Real-Time Distributed Computing, 2005. ISORC 2005. Eighth IEEE International Symposium on*, 2005.
- [10] T. Steinbach, F. Korf and T. C. Schmidt, "Comparing time-triggered Ethernet with FlexRay: An evaluation of competing approaches to real-time for in-vehicle networks," in *Factory Communication Systems (WFCS)*, IEEE, 2010, pp. 199-202.
- [11] J. L. Du and M. Herlich, "Software-defined Networking for Real-time Ethernet," in *International Conference on Informatics in Control, Automation and Robotics*, 2016.
- [12] C. Häne, T. Sattler and M. Pollefeys, "Obstacle detection for self-driving cars using only monocular cameras and wheel odometry," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference*, IEEE, 2015, pp. 5101-5108.
- [13] J. Funke, M. Brown, S. M. Erlien and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Transactions on Control Systems Technology*, vol. 25, pp. 1204-1216, 2017.
- [14] CAN-Utills Team, "CAN-Utills," 11 2018. [Online]. Available: <https://github.com/linux-can/can-utils>.
- [15] H. Khalid and C. Mahmoudi, "can-2-ip-wrapper," 11 2018. [Online]. Available: <https://github.com/usnistgov/Intra-Vehicular-Networks/tree/master/can-2-ip-wrapper>.
- [16] C. Chen, L. Jia, W. Zhou and B. T. Loo, "Proof-based Verification of Software Defined Networks," in *Open Networkin Summit*, 2014.
- [17] D. Kozen, "NetKAT—a formal system for the verification of networks," *Asian Symposium on Programming Languages and Systems*, pp. 1-18, 2014.