Retrofitting SDN to classical in-vehicle networks: SDN4CAN

Michael Doering
Robert Bosch GmbH - Corporate Research
Communication and Network Technologies
Renningen, Germany
Michael.Doering2@de.bosch.com

Marco Wagner
Robert Bosch GmbH - Corporate Research
Communication and Network Technologies
Renningen, Germany
Marco.Wagner3@de.bosch.com

***Abstract*—In-vehicle networking is currently undergoing radical changes. After decades of creating static, isolated networks, the vehicle's E/E architecture opens towards the world around it. Among other challenges, this development calls for the introduction of dynamic communication patterns into this formerly static world. While first enablers such as service-oriented communication are already on the market, the need for an adaptive network management functionality is still not resolved. In this talk the use cases and advantages of SDN-based concepts for in-vehicle networks are outlined, and an architecture for implementing these features while remaining compatible with non-SDN CAN-nodes is proposed.**

Keywords—SDN; Software-defined Networking; CAN; Controller Area Network; Automotive Networks

I. INTRODUCTION

For in-vehicle networks the Controller Area Network (CAN) is very similar to what Ethernet is for LANs: an established technology, that is quite mature but still evolving, and not going to disappear in the near future. Both technologies need to remain backward compatible when a new version (with increased performance or improved features) is standardized. However, there are huge differences between CAN and Ethernet when it comes to frame sizes (Bytes vs. Kilo-Bytes) and data rates (Kilo-Bits/s vs. Giga-Bits/s). Moreover, software-defined networking is an established technology for Ethernet-based networks, while there is (to the best of the authors' knowledge) not a single publication dealing with SDN concepts applied to CAN.

Compared to typical ICT-systems, vehicles have a longer lifecycle and higher safety requirements. In result, automotive development cycles are also longer, but more thorough. Sometimes proven concepts and technologies from ICT get adapted by the automotive industry after several years, e.g. service-oriented architectures, remote software updates and some elements of cloud computing. These examples have one thing in common that is quite different from current in-vehicle networks: they cause dynamic communication patterns. This is a huge challenge from the automotive perspective, because today's in-vehicle network traffic engineering relies on static communication. The advantage of this approach is reduced complexity, which is very congenial to safety, as it simplifies testing and validation. However, future use cases will cause or even require dynamic network loads.

II. AN EXAMPLARY USE-CASE

In order to illustrate the challenges and needs of such future dynamic communication patterns, we introduce a simplified use-case.

Providing novel IoT and Big Data services are considered as high business potential. Many of these services rely on accessing and collecting valuable data that is embedded in different Electronic Control Units (ECUs) (see Figure 1). However, today this data is deeply buried in the OEM specific E/E architecture. Since in-vehicle communication is statically configured, necessary adjustments would lead to the requirement of re-flashing relevant ECUs.

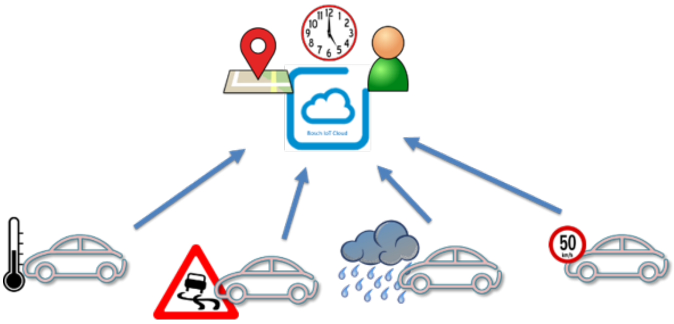
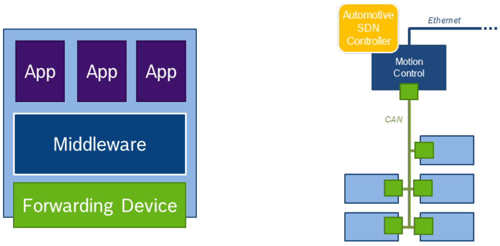


Figure 1 Providing cloud services by accessing valuable vehicle data.

Additionally, since the in-vehicle bandwidth is highly limited and the aggregation of IoT data leads to additional data traffic in the network, dynamic on-demand communication is of vital importance. This dilemma is partly addressed by the introduction of service-oriented communication into the automotive domain. It enables a cost/benefit trade-off by flexible usage of the limited in-vehicle bandwidth and on-demand acquisition of the data most valuable based on parameters like location, time and user. Especially considering the CAN bus, data that is sent can be dynamically chosen without re-configuration of the communication stack. The limited bandwidth can be managed in a way that a subset of services is transferred. This subset can be selected and changed during runtime. However, service-oriented communication does not contribute network management features that retain the dependability of a network when introducing dynamic communication patterns.

III. THE SDN4CAN APPROACH

SDN4CAN is a network management concept that aims to bring the benefits of a software-defined network to the CAN world. However, as CAN is based on a bus topology, a new approach has to be found that takes over the frame filtering, manipulation and forwarding tasks of SDN-enabled switches in Ethernet-based networks. In SDN4CAN, so called ‘Forwarding Devices’ are integrated into the CAN devices themselves (see Figure 2 (a)). They do not only provide access control functionality (e.g. whitelisting, blacklisting, bandwidth budgeting) but also allow to prioritize messages both locally (by building prioritized message queues) and network wide (by manipulating parts of the CAN identifier). In SDN4CAN controlled network, all participating nodes are equipped with such a Forwarding Device. Additionally, one or more Automotive SDN Controllers monitor and govern the communication by assigning forwarding rules to the Forwarding Devices participating in a communication path. Figure 2 (b) illustrates topology of such a network in principle.



(a) (b)

Figure 2 A SDN4CAN node (a) and a CAN network enhanced with SDN4CAN technology (b).

In order to create a new communication path, SDN4CAN has set up a simple and yet powerful procedure. In this procedure, an Application can request a new communication path at a dedicated interface of the underlying middleware (Figure 3, step 1). This request does not have to contain specific knowledge about for example the topology but can be quite abstract (e.g. in form of an intent). The request is received by the middleware instance and converted into a request that contains resource needs. This request is forwarded to the SDN controller using a dedicated southbound protocol (Figure 3, step 2). The SDN controller analyzes the request and examines whether the resources needed can be granted without jeopardizing the existing communication paths. If this is possible, it creates new forwarding rules and assigns them to the Forwarding Devices participating in the new path (Figure 3, step 3). Finally, the path is created after the acknowledgements have been forwarded back to the initializing application (Figure 3, steps 5-6).

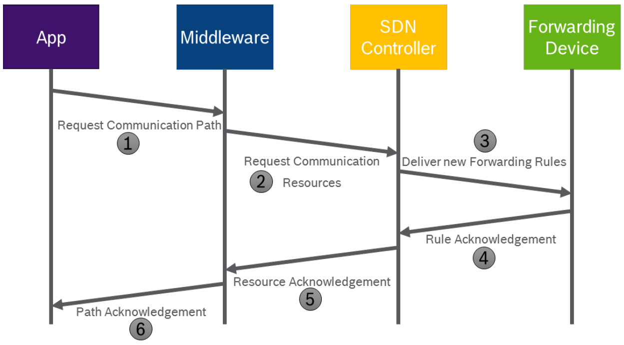


Figure 3 Provisioning of new communication paths in SDN4CAN.

IV. FUTURE WORK AND REMAINING CHALLENGES

In this paper, a basic concept called SDN4CAN has been described that enriches traditional CAN networks with the benefits of software-defined networking. While the overall architecture as well as the main procedures have been defined and already partly implemented, there is still a number challenges to be mastered.

One of these challenges is the strong requirement for availability within an automotive network. This is due to potential hazards caused by malfunction or failure of a device, and calls for a very high level of robustness even in unpredictable scenarios (e.g. spontaneous reboots of one or more participating devices) and a certain level of redundancy while keeping the costs in an acceptable range. Additionally, an automotive implementation of SDN should also contain measures for offline operation (e.g. persistency guarantees).

Besides these non-functional challenges, there are also demands derived from the technological surroundings. This includes for example the heterogeneity of automotive networks with a huge variety in topological styles (e.g. domain-, zone-, backbone-oriented topologies) and a significant number of network technologies (e.g. Automotive Ethernet, CAN, LIN, MOST, FlexRay). These circumstances call for a highly flexible SDN controller that is able to handle not only SDN4CAN sub-networks, but also bigger topologies with a multitude of sub-networks based on different technologies. Furthermore, the technical characteristics of some of the network technologies used create additional challenges, for example limited bandwidth capabilities (e.g. in CAN and LIN) or real-time issues (e.g. TSN or FlexRay). All of these challenges have to be seen in combination with the quite limited resources in many automotive computation devices since computation power and memory is quite expensive due to the harsh environment they face.

Finally, there is another group of challenges caused by the development and engineering environment within the automotive industry. This includes a lack of dependable planning processes for network traffic besides the widely-used best practices (e.g. thumb rule regarding bandwidth capacity utilization). Furthermore, the limited upgradability of devices in combination with the relatively long lifecycles of cars creates additional challenges.