Alexandru N. Onea

alexandru.onea@toporcomputing.com

Abstract

The automotive industry is rapidly expanding as new technologies are adopted. The infotainment domain is one of the most prominent domains, driving huge innovation and research efforts in the field of communication systems, further influenced by the development of ADAS and autonomous vehicles. This paper reviews common communication technologies used inside the vehicle. Both wired and wireless technologies are described together with their applications within the vehicle. Finally, the concept of wireless communication system used for in-vehicle control signaling is proposed. The potential issues of such an approach are discussed and a method to achieve end-to-end protection in wireless ad-hoc networks is detailed.

Review Of Automotive Communication System

Advanced Embedded System

Review of Automotive Communication System

Alexandru N. Onea, *Student, Polytechnic University of Timisoara*

alexandru.onea@toporcomputing.com

1. INTRODUCTION

Since the first modern automobiles produced by Karl Benz and Henry Ford, the automotive industry underwent continuous growth and development, becoming one of the largest industries worldwide. One of the most important drivers of the automotive industry is the constantly increasing demand for functionality, comfort, safety, and performance inside the vehicle, matched to the rapid development of electronics and software technologies [1]. During the last decades an increasing number of functions of the vehicle moved away from mechanical and hydraulically technologies and adopted electronic counterparts. Examples of such functions are the steering, transmission, and wipers. Moreover, new functions were added to the vehicle once certain embedded technologies became available, such as the electronic break system and the electronic stability control. In the same time, governmental regulations reading transportation safety and pollution led to the advancement of specific technologies aimed towards low power consumption and environmentally friendly vehicles. All these factors contributed to the development of embedded systems and software technologies for the automotive industry.

1. Early Stages

Early iterations of vehicle architectures comprised a relatively small number of electronic control units (ECUs). Each of the ECUs implemented a function of the vehicle completely. Although limited, the communication between ECUs was implemented using point-to-point connections [1]. In later years, the number of ECUs steadily increased as the technology became available, thus the number of connections grew proportionally. The development of hardware components made possible the integration of new functions and even domains into the vehicle, such as radio equipment for in-vehicle entertainment. Adding new features and functions had a direct impact on the hardware requirements of the embedded systems regarding processing speed and memory size. The software size within the vehicles rapidly increased from around 1KB to almost over 2MB in just 20 years [1]. Other features required completely new hardware to be integrated, such an example being the introduction of GPS in vehicles for safety and security, and much later for navigation purposes. Even though such hardware was available, it soon became obvious that it was not suitable for the road conditions in which the embedded systems within the vehicle should operate. Soon, silicon providers started producing automotive graded integrated circuits (ICs) which met the requirements of temperature, vibration, shock, electromagnetic compatibility, and electrostatic discharge of the automotive industry.

1. Current development

Vehicle became complex distributed embedded systems consisting in large numbers of sensors, actuators, and processing units. Each functions of the vehicle moved from being stand-alone implemented in a single ECU to being distributed across several electronic control units (ECUs). The number of ECUs in a vehicle was also subject to growth, averaging in current vehicle architectures to around sixty [1] [2]. Moreover, the data exchange demand increased as well since more functions required inputs from distinct parts of the vehicle. For example, modern vehicles employ dynamic volume control of the media player with respect to the vehicle speed. The volume control and the vehicle speed reading are implemented in different ECUs; thus, data exchange is required. The previous approach of point-to-point connections between ECUs proved inefficient and the need for large bandwidth communication busses emerged. The advantages that communication busses have over the traditional interconnections are numerous, including reduced weight of cables and, implicitly, reduced cost. In fact, the wiring and the harnesses required to connect various ECUs proved to be the most expensive component before the year 2000, wiring sometimes peaking at a few kilometres in length [3].

The early vehicle communication networks were not standardized and were based on proprietary circuitry and mostly Universal Asynchronous Receiver Transmitter (UART) interfaces. With the increasing vehicle functionality and competition on the market, the automakers started employing external suppliers to develop more complex and standardized communication interfaces [4]. The standardization of protocols and in-vehicle communication allowed for more complex systems to be integrated and led to the so-called open architecture of the modern vehicles. Bosch developed one of the first vehicle networks – the CAN (Controller Area Network) around the year 1980, which was first integrated into production vehicles by Mercedes-Benz in 1992.

1. VEHICLE ARCHITECTURES

In-vehicle data communication has been a key component to the modern vehicles. Whether it is inter-ECU or intra-ECU communication, the parameters of data transfer play a critical role in the performance of each individual function of the vehicle and the overall system. The communication technologies integrated into modern vehicles are standardized and described in detail at electrical, physical, software and mechanical level. The characteristics of the physical layer impact the throughput and the maximum data rates. The mechanical aspects of the connection contribute to the cost and weight reduction of the vehicle while also affecting the behaviour of the connection in the presence of shock, wide temperature ranges, vibrations, and other environmental constraints. The electrical aspects influence both the performance of the data transfer and the power consumption while the software design of the connection transmission and reception sides impact the resources consumption for the processing units (CPU load, memory, etc.).

The increase in vehicle functions both in the infotainment and the driver assistance systems domains exploits to the limit the currently available in-vehicle communication technologies in terms of bandwidth and performance. Following the development of advanced driver assistance systems (ADAS) and other technologies aimed towards autonomous driving, it became obvious that as the driver interaction with the vehicle will be reduced, the potential and opportunities for in-vehicle entertainment and even workspace functions will rapidly grow. Thus, the vehicle will need to support many more functions such as high-speed internet connectivity, high quality media streaming and playback, workspace applications and fully featured connectivity with the external environment. These features require more bandwidth than it is currently available using traditional communication and in-vehicle networking technologies.

In the traditional gateway vehicle architecture, the communication between different domains in routed through special ECU called the gateway. One of the purposes of the gateway ECU is to make sure that there is information flow across domains and across their different networks. In addition, the gateway performs conversions of messages between different protocols (e.g. CAN to LIN, MOST to CAN, etc.) when necessary. Furthermore, the gateway is responsible for delivering diagnostic requests and responses between domains and the On-Board Diagnostic (OBD) clients.

Historically, the advantages of such a gateway architecture are both in terms of lifetime and in terms of flexibility, enabling easy replacement of damaged parts as well as enhanced robustness for adjusting the system throughout the production stages [5]. However, the before described advantages add overhead to the communication between distinct parts of the vehicle and lead to overloading and interferences.

To solve the overloading problem and, possibly, to address the timing constraints imposed by the processing of various critical sensor information such as video camera inputs, a possible solution is to cluster vehicle functions into single ECUs [5]. With this approach, also depending on the various functions of the vehicle, the signal acquisition from sensors, the processing and the decision are performed within the same local system, thus reducing the amount of time required to perform actions based on sensor input. Moreover, this approach reduces the number of ECUs in the vehicle and the number of inter-connections, potentially saving production costs.

Such an architecture imposes two main disadvantages. Foremost, centralizing functions demands increased processing power to meet the functional requirements of the applications. Such processing power is not available at low cost and more expensive high-performance hardware platforms are necessary. Secondly, by grouping all functionality related to certain features of the vehicle into a single place, some redundancy is lost and must be dealt with while also complying with the automotive standards [5]

1. WIRED COMMUNICATION

Communication over wires has always been the primary and most reliable method of exchanging data between systems. The working principle of wired communication is the propagation of electrical signals through a physical conductor. The communication is, therefore, limited by the electrical properties of the conductor and its length. In addition, the communication is affected by signal encoding scheme which can further limit the data rates and the latency. Many encoding schemes and protocols have been proposed, each of them presenting different performance metrics. Since communication within the vehicle must also comply to the automotive standards, only a subset of the general communication protocols is suitable for in-vehicle use, especially when considering the increasing complexity of the current and future architectures. Moreover, once the point-to-point communication approach became infeasible and the need for multiplexed data transfer emerged, it became obvious that the available protocols are not sufficient to meet the stringent performance and timing constrains of the industry. This led OEMs and suppliers to research new methods for transferring data which satisfy the industry needs in terms of both performance and cost. The following sections describe several wired communication protocols both widely used and experimental.

1. CAN

Controller Area Network (CAN) is a serial communication bus developed in 1983 by Robert Bosch GmbH and later standardized as ISO 11898 in 1993 [5]. At the time of the standard publication, several silicon vendors already released CAN chips, including Motorola and Intel [6]. First integrated by Mercedes-Benz in 1991 and followed by the adoption of the protocol by other OEMs such as BWM, VW, Audi, PSA, Ford, and Volvo in their production vehicles, CAN rapidly became the industry-wide standard for in-vehicle communication. Today almost all vehicles sold worldwide are equipped with at least one CAN network [6].

CAN networks can operate at both high-speed and low-speed. Data rates at up to 1Mbps are possible for classical CAN while data rates above 1Mbps can be achieved with CAN-FD (CAN Flexible Data rate) – a variation of CAN which can carry more data in shorter time frames [7]. The flexibility in data rates makes CAN suitable for a range of in-vehicle applications without compromising cost or performance. In the power-train domain, CAN is used to deliver high-speed, low-latency messages between ECUs. A typical example is the engine and transmission control system which must respond to the driver input (pedal position) in critical time. Another example is the breaking system which activates both the breaks and the stop lights at the rear of the vehicle within a few milliseconds from the driver input trigger. Such applications have time constraints associated with them and a high-speed CAN network (500kbps) is employed to deliver the necessary messages in time. Other domains such as body and comfort do not have the same performance requirements and are usually implemented with low-speed CAN networks (100kbps to 125kbps) [6] [8] [9].

To meet the real-time constraints of the communication, CAN implements message arbitration based on priorities and non-pre-emptive transmission [5] [6]. Each node connected to the bus can send and receive messages and there is no differentiation between nodes. In fact, CAN does not identify the participants to the network but rather the messages being sent. In this sense, CAN is a multi-master bus and the participants are anonymous. The access to the bus must be arbitrated since there is no master-slave synchronization between nodes. Arbitration is done using the Carrier Sense Multiple Access/Collision Resolution (CSMA/CR) scheme. CAN distinguishes between four types of frame which control the network traffic: *Data frames*, *Remote Transmit Frame*, *Overload frame*,and *Error frame*. The standard data frame format always includes an 11-bit unique identifier which is used in the arbitration process. The identifier is also used by the receivers to filter unneeded messages. The standard data frame format is shown below.



Figure 1 - Standard CAN data frame structure

The RTR frame shares the same structure but does not include any data and has the RTR bit set. The Error and Overload frames violate the standard structure and are used to signal the state of the network to the participants. In addition to the standard format, CAN also support the extended format with 29-bit identifiers.

On the physical layer, CAN works with two states: *dominant* and *recessive*. The dominant state is represented by a value of 0 on the bus (i.e. the bus is in the low state) while a recessive state is represented by a value of 1 on the bus (i.e. the bus is in the high state). The succession of the two states is used by the network to perform the arbitration as well as to signal errors. As with other protocols with collision resolution (e.g. IEEE 802.11), each node must wait an arbitrary period (called *bus idle period*) before attempting to transmit. In the case of a collision (i.e. two nodes attempt to transmit in the same time), the nodes monitor the bus state and determine the relative priorities of the messages being transmitted. The node with the highest priority message will continue transmitting while the other nodes will stop and wait until the next idle period. Nodes determine the relative priority by monitoring the bus state bit by bit. A higher priority message will transmit a dominant bit when all other messages will attempt to transmit recessive bits. This scheme requires that nodes can overwrite recessive bits on the bus (i.e. a state of 0 will overwrite a state of 1 arrived the bus in the same time). To achieve the overwriting behaviour, each bit must be transmitted for a long enough time (called the propagation time) such that all nodes can detect it. The propagation time of a bit and, implicitly, the data rate, are directly affected by the bus length (i.e. total length of cable linking all nodes). In the table below, data rates are shown together with the maximum bus length requirements.

|  |  |
| --- | --- |
| Data rate | Maximum bus length in meters |
| 1 Mbps | 40 |
| 500 kbps | 100 |
| 100 kbps | 500 |
| 50 kbps | 1000 |

Table 1 - Maximum CAN bus length for various data rates

Nodes on the CAN bus do not share an implicit common clock. Thus, synchronization is required for nodes to properly monitor the bus. In this respect, CAN protocol specifies that each node must synchronize to the *start of frame* bit of any transmitting message. Moreover, each node can send a *start of frame* bit only after the bus has been in the idle state for at least the amount of time specified by the *inter-frame space* (the last 3 bits of the standard CAN frame).

Nodes can check for errors in message transmission in several ways. Each CAN data frame includes a 15-bit CRC which can indicate an *CRC error* in case it does not match the checksum calculated by the receiver. Nodes can also examine the fixed part of the frame (i.e. bits with fixed values depending on the frame type) and detect a *form error* in case they do not match. The ACK field of the frame can also indicate an *ACK error*. Finally, nodes can detect inconsistent bit patterns in the transmitted message causing a *bit stuffing*error. Upon detecting an error, the respective node transmits an *error flag* consisting in either the sequence 000000 for active errors or 111111 for passive errors. Because a sequence of 6 bits of the same polarity is used to signal errors, other messages transmitted on the bus most avoid the two patterns described above. In this respect, nodes employ a mechanism called *bit stuffing* which always inserts a bit of the opposite polarity after 5 bits of the same polarity in the variable parts of the message.

Based on the bit propagation time of the network and the worst-case bit stuffing scenarios, it is possible to calculate the maximum transmission time of a message as [6]

where is the maximum transmission time and is the number of data bytes of the message. The above equation is for standard 11-bit identifiers. In-vehicle CAN networks are usually used to transmit a fixed set of messages, each having a deadline associated (i.e. the latest time it should arrive for the system to still respond properly). Because worst-case response times can be calculated for CAN messages and based on priorities, it is possible to estimate in advance the performance and real-time behaviour of a CAN network. This property is extremely useful in the automotive industry and has been proven by [10] and later corrected by [6], leading to bus utilizations of up to 80% in modern vehicles, which can still guarantee the real-time behaviour.

1. LIN

The Local Interconnect Network (LIN) protocol was developed in 1998 by a consortium founded by several OEMs including Motorola, Audi, BWM, Daimler and Volkswagen. The protocol is intended as a low-cost alternative to CAN. Certain applications do not require the full features provided by CAN, such as connections between control units and smart sensors and actuators. In addition, the data rates required in many of these applications are much smaller. LIN is designed to be used with applications where CAN fails to satisfy the cost criteria and is under-utilized.

LIN operates as a serial, single-master, multiple-slave bus where communication is always initiated by the master. Because slaves cannot transmit messages without a request for transmission from master, collisions are not possible, and no arbitration is necessary. The number of slaves in a LIN network is limited to 16. The master initiates communication by sending a request message to which at least one slave will respond. Like CAN, each message has associated a unique identifier.

At the physical layer, each connection is realised via a single wire transporting signals at 12V. The data coding scheme is specified by the ISO 9141 NRZ standard. Synchronization of slave nodes with the master is done automatically through synchronization pulses. This feature of LIN reduces the protocol costs by removing the need for oscillator devices in slave nodes. The maximum data rate is limited to 20 kbps.

The master node governs and monitors the network. To reduce power consumption, the master node can switch individual slave nodes to the sleep state when their input is not required and wake them up when communication is necessary. This is particularly useful in applications such as smart sensors where the input from a sensor may not be always required and the sensors can be temporarily disabled by the master node. Slave nodes have the possibility to signal to the master that they need to wake up by sending *wakeup breaks*.

Each LIN message starts with a synchronization pulse sent by the master. Slaves wait for the synchronization pulse and use it to synchronize to the master clock. After synchronization, each slave node decides which action to perform next based on the unique identifier of the message. Possible actions are: *receive data*, *respond with data*, and *do nothing*. Data sent from slave nodes to the master can have lengths of 2, 4 or 8 bytes.

Messages are transmitted over LIN networks in frames. There are two types of frames specified: data frame and diagnostic frame. A special diagnostic frame with all-zeros value can be used by the master as a special request to switch all slave nodes to the *sleep* state. All frames contain an 8-bit checksum which is used by the receiving nodes to validate the frame content. The LIN frame format is composed of two sections: the message header (sent by the master to all slaves) and the message response (sent by the responding slave to the master). In the picture below a typical frame is described.



Figure 2 - LIN frame format

Each LIN frame header starts with a sequence of 13 dominant bits followed by a recessive delimiter bit, acting as a start of frame notice for all slave nodes. After this sequence, the master sends a synchronization pulse of 8 bits representing the patter 01010101 (i.e. the value 0x55) which is detected by the slave nodes and used to adjust their baud rates to the master clock. The last part of the frame header is the 8-bit identifier used by the slaves to decide the appropriate action. The unique identifiers also specify the type of frame. Each slave which transmits data to the master will send a response message containing the data followed by an 8-bit checksum. With the apparition of the LIN 2.0 specification, more frame types have been introduced. The difference between these frame types is the transmission mode and the data length.

1. MOST

With the rapid evolution of the infotainment domain, demands for bandwidth and high-speed increased and neither CAN nor LIN could meet the performance requirements for the new domain. Media Oriented Systems Transport (MOST) is a bus technology developed in 1998 to support automotive multimedia networks [11]. MOST busses are implemented using ring topologies and can support up to 64 nodes. MOST specifies both a method for transmitting audio, video, and signal data at data rates up to 150 Mbps and the higher-level abstractions necessary for implementing complex infotainment applications.

The MOST physical layer specifications describe the connection between two neighbouring nodes or Network Interface Controllers (NICs) in terms of optical-to-digital conversion and the physical connection to the optical medium. Each NIC incorporates a transceiver which can convert optical signals to electrical signals and vice-versa. The technology is based on the 1 mm core-diameter step-index Polymethylmethacrylate (PMMA) plastic optic fibre (SI-POF). Each transceiver is equipped with a 650 nm LED for transmitting optical pulses and a photodiode which senses the optical pulses sent by the neighbouring NIC.

The MOST bus usage is not limited to multimedia only as the protocol allows for encapsulation of ethernet frames and MAC addressing as well as high-speed SPI interfaces. Using optic transmission medium, MOST is more resilient to electromagnetic interference (EMI), this being an important advantage over other communication technologies based on transmission of electrical signals [9].

MOST can perform at higher data rates by using optimized Quadrature Amplitude Modulation (QAM) techniques, reaching up to 1 Gbps as shown in [12].

1. Automotive Ethernet / OABR

Ethernet is an IEEE standardized communication solution widely-available in the consumer and industrial domains which supports the bandwidth and data rate demands of the automotive industry. In addition, Ethernet comes with excellent hardware, software, and tools support which can be reused in the automotive industry as well, reducing the overall development costs. These factors made Ethernet a strong candidate for the future in-vehicle communication technologies. Nevertheless, Ethernet is not designed for use in the vehicles and optimizations must be made to comply with the automotive standards [13].

The IEEE 802 standards family specify various physical layers together with the MAC layer. Physical layers defined in the IEEE 802 standards family range from 10 Mbps up to 10 Gbps, giving high enough bandwidth and data rates for all automotive applications. The certification of the Automotive Ethernet is based on the 802.3u standard commonly referred to as 100Base-TX and is standardised as ISO 13400 for diagnostic over IP use. However, for comprehensive support for infotainment applications, the Automotive Ethernet standard must also specify higher layers of the OSI model. In this direction, the Audio/Video Bridging (AVB) Task Group [14] defined multiple IEEE standards addressing synchronization, timing, audio/video forwarding and queuing. Specific applications such as capturing video streams from multiple cameras for Advanced Driver Assistance Systems (ADAS) require their own standardizations and effort is already invested in specifying communication protocols for those applications (e.g. ISO 17215). The physical layer of the Automotive Ethernet is refined and specified by the Open Pair Ethernet (OPEN) Alliance while the software stack is addressed by the Automotive Open Architecture (AUTOSAR) Consortium. The joined effort aims to produce a complete standardization of the Automotive Ethernet leading to its deployment in production vehicles.

One of the major open points for Automotive Ethernet is its ability to exhibit real-time behaviour by guaranteeing message response times. Since Ethernet is not designed for multiple access based on time division (TDMA), higher level protocols must be developed to support the required capabilities. A variant of Ethernet called Time-Triggered Ethernet (TTEthernet) is available which uses distributed clock synchronization techniques to guarantee deterministic message transmission and reception.

The deployment of the Automotive Ethernet is scheduled in three generations. The first generation represents the use of Automotive Ethernet for On-Board Diagnostics (ODB) as well us for updating the software of ECUs. The 100Base-TX physical layer is used as interface between the vehicle and the test equipment while the protocol stack is described in ISO 13400 and ISO 14229 built on top of existing industry standards. Second generation of Automotive Ethernet introduces its use for ADAS and infotainment where audio and video streams are prominent. The current approach is based on LVDS links for video streaming. However, with an increasing number of cameras, the LVDS solution is no feasible because of high cabling costs. Moreover, the switching properties of Ethernet allow for seamless data aggregation, fusing and synchronization. Since low power consumption is required, the IEEE 802.3az standard (Energy Efficient Ethernet) is proposed for use with Ethernet-based cameras. Energy Efficient Ethernet introduces the ability to temporarily switch cameras to a Low Power Idle state when not in use and wake them up later. Cable cost is also reduced with Power-Over-Ethernet (POW) technology. For infotainment applications where high bandwidth is required, Automotive Ethernet specifies the higher-layer AVB protocol based on OPEN Alliance BroadR-Reach (OABR) physical layer. The final generation introduces Automotive Ethernet as a backbone for the domain-controlled architecture. Point-to-point connections link the domain controllers with a central ECU acting as a switch and routing inter-domain communications. One major advantage of the Ethernet backbone is the use of IP packets for both internal routing and external communication since most of the consumer electronics exchange data between them in this form already.

Although solutions for high data rates exist with Fast and Gigabit Ethernet, they cannot be used directly in the automotive industry. The symbol rate of 125M symbols per second lead to high electromagnetic emissions in the critical FM band. Therefore, such physical layers cannot be used without expensive shielded twisted pair cables since the automotive industry has strict requirements regarding electromagnetic compatibility (EMC). To address this issue, the OPEN Alliance developed the BroadR-Reach physical layer which uses a bi-directional communication scheme over a single pair of unshielded twisted pair cables. The PHY component of the OABR has been modified to replace the conventional 1:1 transformer (used to isolate nodes from the network cable) with a decoupling capacitor, also reducing the Bill of Materials (BOM). Below is the schematic representation of an OABR connection.

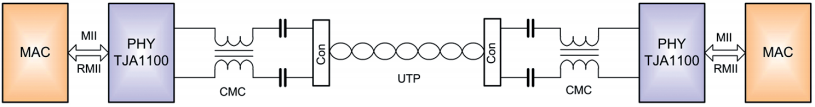


Figure 3 - OABR connection schematic [13]

Therefore, the Automotive Ethernet provides a cost effective and robust replacement for the available technologies such as CAN and FlexRay while also meeting the performance requirements.

1. WIRELESS COMMUNICATION

In recent years, infotainment and telematics applications gained increasing attention, including technologies focused on wireless communication. Mobile devices such as smart-phones, tables and laptops have developed in time as part of an ecosystem and the consumer market has learned to interact with them in an integrated fashion. Nowadays the same demands are placed for vehicles and OEMs are trying to make their products part of the same ecosystem. The successful integration of the next generation vehicles with the already integrated mobile devices used by consumers depends on wireless technologies and their adoption by the automotive industry [15].

1. Telematics

Vehicle integrations with external services over wireless links is referred to as *telematics*. Interconnected mobile devices such as smart-phones and laptops already used a series of wireless communication technologies. The automotive industry tries to integrate these technologies to explore new opportunities for the infotainment domain. Two major wireless technologies are deployed in vehicles today for telematics: Bluetooth and WLAN.

Bluetooth is a low-power, short range, ad-hoc wireless protocol specified by the IEEE 802.15.1 standard. Although not intended for automotive use, Bluetooth became interesting for in-vehicle use due to the increasing number of mobile phones and their associated connectivity. The primary feature offered by the protocol for vehicles is *hands free*, which enables mobile phones to automatically connect to the vehicle infotainment system and exchange data such as phonebook, short messages, call history and voice.

WLAN is mainly used for internet connection sharing between the vehicle and the mobile phone and for data transfer including photos, music, and documents. Moreover, WLAN is used at higher levels for applications integration using protocols such as AppleCar.

1. Wireless CAN

The Wireless CAN (WCAN) protocol is an experimental protocol in research aiming to provide the same benefits as wired CAN but over a wireless medium. The protocol is based on the same principles as the Wireless Token Ring Protocol (WTRP) to reduce retransmissions by avoiding collisions. The WTRP protocol works by sharing the same medium between multiple users which gain access to the medium in turn using a token. The token owner only can access the network while the other users wait. When the owner ends its turn, it passes the token to the next user. The ring topology guarantees fairness. The overall protocol architecture of WCAN is shown below.

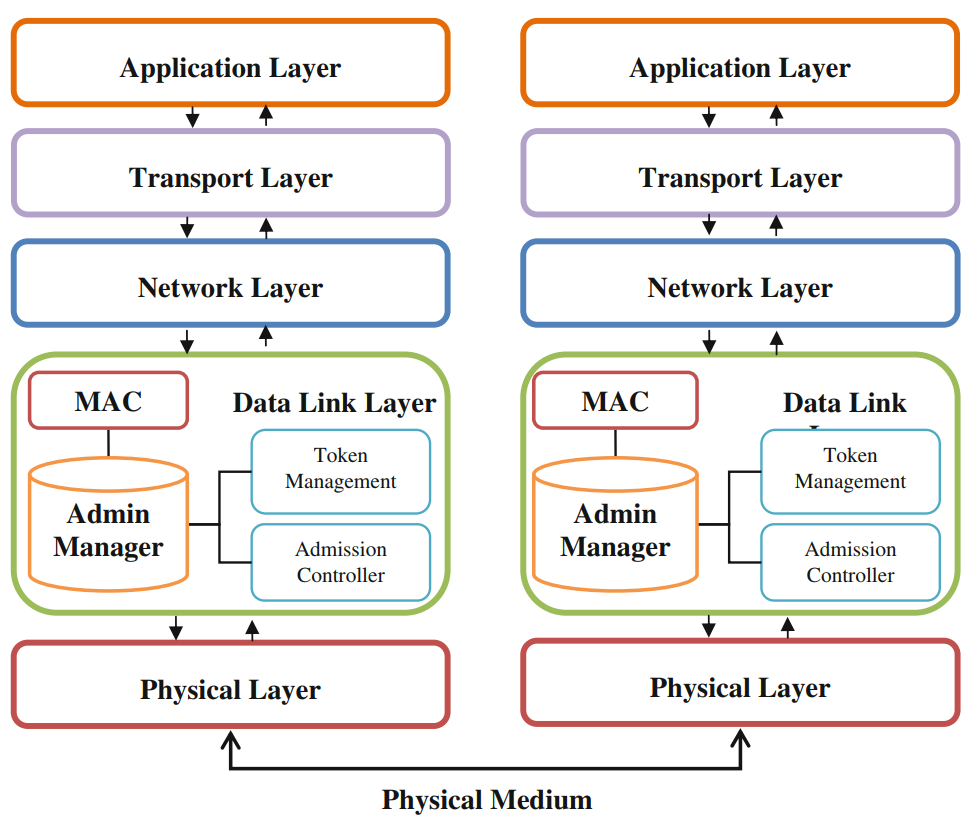


Figure 4 - WCAN protocol architecture [16]

1. Wireless power

Migrating communications to wireless technologies does not solve the power supply problem. Power is still transmitted over wires in most of the electronics used today. Wireless charging has reached the smartphone industry, but it is still not able to charge or power up more power demanding systems. Vehicles are power through a 12V or 48V high current battery. The 48V technology is especially designed for low current applications to reduce the wires gauge. The vehicle battery is the sole power supply of all the electronics and components. Automotive electronics are required to have low power consumption profiles and have optimized power usage. Wireless power transfer for electric vehicles is researched.

[[1]](#footnote-1)

Message authentication in automotive wireless networks using decentralized ledger

Alexandru N. Onea, *Student, Polytechnic University of Timisoara*

*Abstract*—Wireless communication technologies are adopted by the automotive industry for telematics and infotainment applications but not for vehicle control networking. Widely used wired networking technologies such as CAN are designed to exhibit real-time behavior at relatively low cost and power consumption. In terms of security, messages transmitted over wired connections do not need end-to-end protection since tempering with the network is possible only by wiretapping or access to a connected node. We propose a real-time wireless network for use in the automotive industry and we show how to implement end-to-end protection using a decentralized ledger. This paper also discusses open topics and future research directions.

*Index Terms*—automotive, communications, end-to-end protection, ledger, wireless

# INTRODUCTION

W

IRELESS communication technologies are slowly being adopted by the automotive industry as means to enhance the user experience in the infotainment and telematics domains. From the integration of Bluetooth in 1999 for the hands-free feature to the nowadays Wi-Fi in-vehicle connectivity, wireless technologies started to present interest and proved to be critical to the success of the future vehicle integrations [8]. However, wireless is only used for telematics while the in-vehicle control and critical real-time communication is still exclusively carried over wired mediums. The fist widely deployed wired communication protocol for automotive is CAN, designed specifically to reduce the interconnections between ECUs and to cut production costs [6]. The cost criterion is one of the most important criteria for the success of the industry and new protocols such as OABR have been developed to further reduce the Bill of Materials (BOM) and in the same time to meet the desired performance requirements [13]. Wireless communication has not been deployed outside of the telematics domain mainly because of two reasons: the reduced reliability of the communication and the elevated risk of unwanted network access. In addition, the risk of interference has also been considered. While the reliability of the wireless communication can be improved through various techniques, the risk of attacks remains because of the open nature of the network. Although numerous methods to encrypt and secure the communication exist, this considerably increases the overhead which is not compatible with the real-time requirements of the system.

In this paper we propose a reference implementation of a real-time wireless network suitable for automotive use and a method for validating the authenticity of messages sent on such a wireless network, based on a decentralized ledger. The applications of the presented method are not limited to the automotive domain.

# An Automotive Real-Time Wireless Network

Several critical aspects should be considered when designing a wireless network for automotive use. First, the most important acceptance criterion is the real-time behavior of the network. Being used for communicating safety-critical control signals, the network must exhibit deterministic behavior. This requires the network design and its protocols to allow for message scheduling and to guarantee their deadlines. Second, the security of the network is an important factor of decision regarding its performance. While wired networks are considered secure because access to the network cannot be obtained unless a potential attacker taps the wires or gains access to another network connected device, wireless networks are open (i.e. traffic can be easily intercepted). Therefore, special measures must be developed to ensure the security of the communication. Third, the feasibility of the network in terms of costs and power consumption is considered. Low power consumption requirements are placed on every vehicle component. Moreover, production costs should not exceed the costs of already existing communication technologies. Finally, the radio interference is another critical aspect. Wireless networks can co-exist in the same space as long as their frequencies do not overlap. While it is relatively easy to ensure the non-overlapping characteristic of the specifically designed networks, the electromagnetic interference (EMI) produced by all oscillating electrical signals can interfere with the networks on their respective frequency bands. Solutions must be devised to both minimize the EMI and guarantee the non-overlapping characteristic of the deployed networks [17].

Wireless technologies closely resembling deterministic behavior are already deployed in other industries such as telecommunications. The LTE network uses 1 millisecond scheduling intervals to grant users downlink and uplink access to the network. LTE also limits the access time for its users by implementing 0.5 milliseconds time slots. However, LTE uses a client-server topology where the server is implemented as the eNodeB (i.e. the base station). This approach allows centralized scheduling and that users receive fair access to the network. Moreover, centralized scheduling enables Quality-of-Service (QoS) dependent access, which is of interest for the automotive industry as well.

The disadvantage of the centralized scheduling is that the scheduling node must have enough computation power to support the heavy tasks of scheduling. In addition, the scheduling node must be able to transmit and receive message from all other nodes. This implies relatively high power consumption which, in the case of LTE, is not considered an issue due to the fact that the base station is not battery powered. Therefore, a distributed ad-hoc wireless network topology is more suitable for automotive use.

The Single Carrier Frequency Division Multiple Access (SC-FDMA) modulation scheme is used in LTE and other wireless networks as a low-power scheme used to transmit data over a single carrier. The multiple access is provided in frequency domain but can be further expanded to the time domain by employing an additional Time Division Multiple Access (TDMA) scheme over the existing frequency divisions. Such a combination of SC-FDMA and TDMA can be used as basis for an ad-hoc wireless communication network for use in automotive. We reference the proposed network as Real-Time Wireless (RTW).

Almost all in-vehicle control traffic has real-time constraints associated and the proposed wireless network must also behave deterministically. This can be achieved by combining the TDMA scheme discussed above with a distributed synchronization and scheduling algorithm. The scheduling is performed by each node individually based on a ground truth consensus which all nodes share. The criterion for scheduling messages is their priority and type. Each message priority will be mapped to a certain periodicity and deadline. Each transmitting node can determine the required deadline for a message prior to its sending and allocate the necessary resources on the network. To avoid collisions, the unique identifiers of the messages will also dictate which resources must be used by the sending node. A resource grid similar to the resource grid of LTE is used to predetermine to available resources while each node will individually manage the resources needed to transmit its messages.

Receiving nodes can also predetermine the resource grid allocation of their expected messages since the unique identifier of the messages is known in advance. Thus, receiving nodes can poll the resource grid element (i.e. the exact frequency and time slot) with the expected periodicity. Further optimizations can be obtained by designing the physical layer such that signaling of new messages is done automatically. Furthermore, message requiring more bandwidth can be mapped to more resource grid elements to ensure the rapid transmission.

RTW messages are transmitted in frames similar to CAN frames. A reference RTW frame is shown in Figure 5. Each frame has an additional field – AUTH – which is used in the authenticity validation protocol described in the next chapter.



Figure 5 - Reference RTW frame

# Message Authenticity

The advantage of wired networks is that unwanted access to the network can be granted only by physically tapping the wires or gaining access to an already connected node. With wireless networks this advantage is not applicable due to their open nature. Moreover, wireless networks allow much easier access to the network because a physical connection is not required. Therefore, a method to verify the authenticity of the transmitted data is mandatory.

The proposed protocol is based on a global decentralized ledger which is individually kept by each participating node. The ledger acts as a ground truth for all nodes by providing a method to verify whether new information entering the system is in accordance with the information already processed and determined as authentic. The ledger is internally stored as a SHA-256 hash by each node and it is updated with each network operation (referred to as a network transaction). Each network transaction is known to the entire network. In addition, each resource grid element has associated a transaction sequence number. The transaction sequence numbers are incrementally computed for each transaction occurring in the respective resource grid element by the transmitting node and they are not publicly transmitted in the RTW frame. However, they are used in the authenticity validation protocol. The transaction sequence numbers and the ledger hash are, in combination, the ground truth shared by each node which is used to distinguish authentic messages from intruding ones.

The transmission of a message is preceded by the calculation of a special hash by concatenating the current ledger hash and the next transaction sequence number of the respective resource grid element and applying the SHA-256 hashing function to the concatenation. This value is then truncated to 64 bits and included in the RTW frame of the message as the AUTH field. Even though the frame is publicly visible on the network, the AUTH field has no meaning to an attacker since it is the result of a hashing function.

After the reception of a message, the receiving node can verify the authenticity of the message by repeating the same calculations as the transmitting node and checking the results against the AUTH field of the message. Since both the sequence transaction number and the current ledger value are available to all receiving nodes, the validation of an incoming message is possible regardless of the sender and receiver, as long as both are trusted network participants. Since receiving nodes are expecting messages with certain unique identifiers and priorities (thus, actively polling the respective resource grid elements), they can update the value of the transaction sequence numbers in synchronization with the transmitting nodes.

In the case of matching AUTH values, the receiving node can conclude that the message must have been transmitted by a trusted network participant. In this case, the receiving node will update both the transaction sequence number of the corresponding resource grid element and the ledger value. While the transaction sequence number is simply incremented, the new value of the ledger hash is calculated as the result of the SHA-256 hashing function applied to the concatenation of the previous ledger value, the old transaction sequence number of the corresponding resource grid element, and the value of the AUTH field in the received message. The transmitting node performs the same updates upon transmission.

In the case of non-matching AUTH values, the message is regarded as not authentic and must be discarded. No update of the ledger hash or the transaction sequence number is performed.

To transmit a message as a non-trusted participant (i.e. an attacker), the transmitting node must know in advance both the ledger hash and the current transaction sequence number to calculate a valid AUTH value. Without knowing these values, the attacked is not able to send a message which will be considered by the rest of the network participants valid. Moreover, the attacker will not be able to temper with the values of the ledger hash and the transaction sequence number. Because of the chosen method of computing both the AUTH values and the updated ledger hashes, the attacker cannot study the progression of AUTH values sent on the network to deduce the inner working of the protocol or the current values of the parameters.

The proposed network is end-to-end protected by the proposed method at the cost of an overhead imposed by the fact that all nodes must actively poll all resource grid elements to have updated values of the transaction sequence numbers. Optimizations can be applied by embedding some of the computation related to the validity checking protocol in the network hardware.

# Open Points

Wireless communication systems are still being researched and there are numerous open points related to both their feasibility and their security. In terms of feasibility, the protocol must be optimized for low power consumption in both uplink and downlink. Automotive applications require message transmission periods of 1ms or less which leads to a high number of transmissions per time unit. Therefore, the power required to transmit a single frame should be sufficiently low so that the protocol becomes cost- and power-effective. In terms of security, the proposed method does not address sniffing attacks. Although the message authenticity can be verified, an attacker is able to monitor the network and intercept all traffic. This has direct security implications since it allows attackers to track the status of the vehicle through the messages sent. In terms of network performance, the proposed wireless technology does not consider collisions and, although the system can be designed in such a way that collisions are always avoided, an attacker can still disturb the system by intentionally transmitting colliding messages.

The overhead introduced by the active polling of each resource grid element must be carefully analyzed both in terms of time and power consumption as message reception and delivery is greatly impacted by processing time. The storage of the ledger hash and the transaction sequence number values is also critical because access to the memories of the ECUs is possible in theory.

# Conclusions

There are several still open points associated with wireless communication technologies in the context of vehicle control networking. However, recent research allowed the development of new methods to overcome most of the obstacles. Ad-hoc wireless networks can be designed to behave deterministically and can be secured against certain types of attacks. Nevertheless, more research is needed to fully specify an automotive compliant wireless control network.

# References

|  |  |
| --- | --- |
| [1] | N. Navet and F. Simonot-Lion, Automotive Embedded Systems Handbook, New York: CRC Press, Taylor & Fancis Group, 2009. |
| [2] | N. Navet and F. Simonot-Lion, "Trends in Automotive Communication Systems," in *Embedded Systems Handbook: Networked Embedded Systems - 2nd ed.*, New York, Taylor and Francis / CRC press, 2009, pp. 13.1-13.24. |
| [3] | G. Leen and D. Heffernan, "Expanding automotive electronic systems," *Computer,* vol. 35, no. 1, pp. 88-93, 2002. |
| [4] | G. Leen, D. Heffernan and A. Dunne, "Digital networks in the automotive vehicle," *Computing and Control Engineering Journal,* vol. 10, no. 6, pp. 257-266, 1999. |
| [5] | International Organization for Standardization, *Road vehicles -- Controller area network (CAN) -- Part 1: Data link layer and physical signalling,* 2015. |
| [6] | R. I. Davis, A. Burns, R. J. Bril and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems,* vol. 35, no. 3, pp. 239-272, 2007. |
| [7] | F. Hartwich, "CAN with Flexible Data-Rate," in *International CAN Conference*, 2012. |
| [8] | J. Leohold, "Communication Requirements for Automotive Systems," in *5th IEEE Workshop on Factory Communication Systems*, Vienna, 2014. |
| [9] | N. Navet and F. Simonot-Lion, "Trends in Automotive Communication Systems," in *Embedded Systems Handbook: Networked Embedded Systems - 2nd ed.*, 13.1-13.24, Taylor and Francis, CRC Press, 2009. |
| [10] | K. Tindell and A. Burns, "Guaranteeing message latencies on Controller Area Network (CAN)," in *1st International CAN Conference*, 1994. |
| [11] | O. Strobel, R. Rejeb and J. Lubkoll, "Communication in automotive systems: Principles, limits and new trends for vehicles, airplanes and vessels," in *12th International Conference on Transparent Optical Networks*, Munich, 2010. |
| [12] | S. Lee, F. Breyer, S. Randel, O. Ziemann, H. v. d. Boom and A. Koonen, "Low-Cost and Robust 1-Gbit/s Plastic Optical Fiber Link Based on Light-Emitting Diode Technology," in *8Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference*, San Diego, CA, 2008. |
| [13] | P. Hank, T. Suermann and S. Müller, "Automotive Ethernet, a Holistic Approach for a Next Generation In-Vehicle Networking Standard," in *Advanced Microsystems for Automotive Applications*, Berlin, Springer Berlin Heidelberg, 2012, pp. 79-89. |
| [14] | Avnu, "Avnu Alliance," Avnu, 2018. [Online]. Available: http://avnu.org/. |
| [15] | T. Nolte, H. Hansson and L. L. Bello, "Wireless Automotive Communications," in *Euromicro Conference on Real-Time Systems*, 2005. |
| [16] | W. L. Ng, C. K. Ng, B. M. Ali and N. K. Noordin, "Performance Evaluation of Wireless Controller Area Network (WCAN) Using Token Frame Scheme," *Wireless Personal Communications,* vol. 72, no. 4, p. 1847–1873, 2013. |
| [17] | D. Reinhardt and M. Kucera, "Domain Controlled Architecture, A New Approach for Large Scale Software Integrated Automotive Systems," 2013. |

1. [↑](#footnote-ref-1)