An Automated Data Structure Migration Concept - From CAN to Ethernet/IP in Automotive Embedded Systems (CANoverIP)

Andreas Kern
and Thilo Streichert
Daimler AG
Hanns-Klemm-Straße 45
71034 Böblingen, Germany
{andreas.ak.kern,thilo.streichert}@daimler.com

Jürgen Teich
Department of Computer Science 12
University of Erlangen-Nuremberg
Am Weichselgarten 3
91058 Erlangen, Germany
teich@informatik.uni-erlangen.de

***Abstract*—In premium vehicles, the number of distributed comfort-, safety-, and infotainment-related functions is steadily increasing. For this reason, the requirements for the underlying communication architecture are also becoming stronger. In addition, the diversity of todays deployed communication technologies and the need for higher bandwidths complicate the design of future network architectures. Ethernet and IP, both standardized and widely used, could be one solution to homogenize communication architectures and to provide higher bandwidths. This paper focuses on a migration concept for replacing todays employed CAN-buses by Ethernet/IP-based networks. It highlights several concepts to minimize the protocol header overhead by using EA- and rule-based algorithms and presents migration results for currently deployed automotive CAN subnetworks.**

***Index Terms*—Ethernet, IP, UDP, CAN, migration, optimization, automotive, embedded, CANoverIP, XoverIP**

I. Introduction

Todays premium class vehicles have up to 70 *electronic control units* (ECUs) and five different communication technologies to implement and interconnect all distributed functions within a car. Current predevelopment activities indicate that the functionality and number of such distributed functions will further increase and this will lead to more complex system requirements for the underlying communication architecture. One reason is the growth of the communication relationships of functions between different communication domains inside the car. A *communication domain* is a subset of ECUs interconnected through a subnetwork and groups similar car functions. Typical communication domains are *powertrain*, *chassis*, *telematics*, and *body*. For example, the navigation system located in the telematics domain uses the current speed value distributed in the chassis domain to calculate the covered distance of the car within a tunnel, when no signals are available from the satellite-based system. The interconnection of different car domains and communication technologies is realized by so-called *gateways* which are special ECUs containing multiple communication interfaces.

The diversity of todays communication technologies such as *Local Interconnect Network* (LIN) [7], *Controller Area Network* (CAN) [6], *FlexRay* [2], *Media Oriented System Transport* (MOST) [9], and *Low Voltage Digital Signaling* (LVDS) [13] associated with the higher communication requirements mentioned above lead to a high complexity when designing new *electric and electronic architectures* (EE-architecture).

Ethernet [3], the *Internet Protocol* (IP) [12] and higher layer protocols like *User Datagram Protocol* (UDP) [11] could be one solution to meet the future requirements of next generation automotive EE-architectures. These technologies are standardized, support high bandwidths up to 10 GBit/s for consumer electronics, define a global address mechanism, and are even in operation in other business sectors like industrial automation, telecommunication as well as avionics. Furthermore, connectors, so called *switches*, are available to interconnect several Ethernet-based networks to build larger communication architectures.

Due to its complexity, automotive electronics is very legacy driven and thus, a hard switching from CAN-based networks to Ethernet/IP-based networks is very unlikely. Moreover, the introduction of Ethernet/IP-based communication can only be motivated, if advantages in the context of packaging, weight as well as system costs exist. In certain installation spaces, where CAN-based ECUs together with high bandwidth sensors are located, the multiplexing of CAN data and streaming data over one Ethernet connection can be beneficial.

Therefore, this paper focuses on a concept for introducing CAN over Ethernet/IP in automotive communication architectures on a data structure base. We present three concepts to replace a CAN network by an Ethernet/IP-based network with full duplex switched Ethernet connections and IP capable endpoints at design time. The concepts are evaluated with original CAN network configurations from compact-, middle-, and premium-class vehicles. In the end, we want to show that it is possible to use Ethernet/IP for transferring CAN traffic. We present algorithms to optimize the packaging problem and show evaluation results for different car networks. This is one first step to an entirely IP-based car communication architecture.

The remainder of this paper is organized as follows: Section II gives an overview of related work. A comparison of the two technologies CAN and Ethernet is presented in Section III. The proposed migration concepts are discussed in Section IV. Section V presents the migration results. Finally, Section VI concludes the paper and shows future research directions.

II. Related work

Most related research addresses the question of how to interconnect Ethernet and CAN communication networks to forward data from one network technology to another. In the automotive section, the *International Organization for Standardization* (ISO) [4] has found the working group *Diagnostics over IP* (DoIP) [5] which develops a standard for diagnostic communication over IP. They design a standardized vehicle interface which separates in-vehicle communication technologies from external test equipment. The implementation of the interface has to switch data between different in-car networks to the outside IP-based network of the test equipment and is thus working as a gateway. Beside this consideration there exists multiple concepts and implementations [1] for connecting industrial CAN-based plants to local factory networks. The authors present methods to bundle multiple CAN-frames into IP-packets in a dynamic fashion to reduce the protocol header overhead. In [10], the authors discuss a strategy to migrate from a CAN-network to a FlexRay-based solution. They present a solution to connect to a CAN-based network by using a special gateway and provide a concept for replacing a complete network through a FlexRay-based communication.

In summary, the discussed references concentrate on a dynamic behavior without using the information about the exact knowledge of the data which is being send on the network. This paper focuses on a static migration strategy to migrate to an IP-based network infrastructure at design time by using real network communication matrices.

III. Technology Differences

The basic technology differences of CAN and Ethernet are shown in Table I. In the automotive section, a CAN-frame typically includes a *message* and the message itself consists of a set of *signals*. Signals are physical or logical values like current acceleration, light on, or activate seat heating. In the case of Ethernet, IP and higher transport protocols like UDP are used to encapsulate user data into packets. These packets have a header length of 54 byte and a payload length between 18 and 1472 bytes. In contrast, a CAN frame has a header length of 47 bit and a payload length between 0 and 8 bytes which are much smaller compared to Ethernet. The addressing mechanism also differs: CAN uses an 11 bit message identifier which means that every node has to decide by the identifier, if a message belongs to it. On the other hand, Ethernet utilizes a 6 byte mac address which identifies a network interface card. Additional addressing is done by IP and UDP to support logical addresses and port-based addressing. On the physical layer, it is distinguished between shared media when using CAN and point-to-point connections if Ethernet is deployed. In addition, the physical bandwidths of Ethernet are at least ten times higher compared to CAN. The *payload/header-ratio* (p/h-ratio) also differs for CAN and Ethernet with the upper protocols IP and UDP. For example, to get the same payload/header-ratio as CAN, a UDP-based solution will need to use up to 75 payload bytes per packet which corresponds to about nine 8-byte CAN-messages.

TABLE I
The comparison of CAN and Ethernet.

attribute	CAN	Ethernet
packet layout	message $\longrightarrow$ signal	IP $\longrightarrow$ UDP
frame format	base frame format	Ethernet 802.3 frame
header size	47 bit	54 byte
payload size	0 - 64 bit	18 - 1472 byte
addressing	11 bit message identifier	6 byte mac address
topology	shared media	point-to-point
bandwidths	125, 500, 1000 kBit/s	10, 100, 1000 MBit/s
p/h-ratio	15% - 58%	1% - 96% (8% $\cong$ 8 byte)

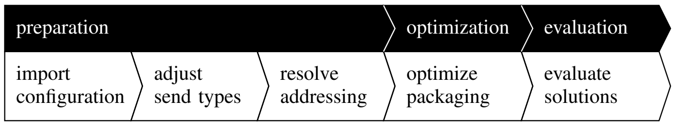


Fig. 1. The proposed migration concept.

In summary, we see three major problems for migrating the CAN communication to an Ethernet/IP-based communication:

- *adjustment of send types*
- *address resolution*
- *packaging of messages into packets*

IV. Data Structure Migration Concept

An overview of the considered migration concept is shown in Figure 1. It's separated in three basic steps: *preparation*, *optimization*, and *evaluation*. The following subsections describe each step and the idea behind each step in detail.

A. Preparation Phase

The tasks of the preparation phase which are common to all considered optimization variants, are *import configuration*, *adjust send types*, and *resolve addressing*.

1) Configuration Import:

The configuration import reads a current CAN network configuration from a configuration file and calculates the ECU-dependent message offsets from a network trace with an algorithm presented in [14]. The result is a set of ECUs E and a set of Messages M. Each message $m \in M$ is defined by the following parameters:

- $s_m \in \{\text{cyclic}, \text{cyclicIfActive}, \text{cyclicIfActiveFast}, \text{cyclicAndSpontanWithDelay}, \text{cyclicWithRepeatOnDemand}, \text{spontan}\}$ - the send type of m.
- $l_m \in \{1, ..., 8\}$ - the payload length of m in byte.
- $e_m^t \in E$ - the transmitting ECU of m.
- $E_m^r \subseteq E$ - the set of receiving ECUs of m.

Parameters which depend on the send type of a message are shown in Table II. For example, a cyclic message m consists of a cycle time t_m^{ct} and an offset t_m^o which define the time

TABLE II
THE MAPPING OF THE CAN SEND TYPES AND PARAMETERS TO THE CYCLIC SEND TYPE FOR THE ETHERNET-BASED SOLUTION.

CAN	**relevant parameters**							**Ethernet**
send type s	cycle time t^{ct}	offset t^{o}	cycle time active t^{cta}	delay time t^{dt}	no. of repetitions t^{nor}	cycle time $t^{ct'}$	offset $t^{o'}$	send type s'
cyclic	t^{ct}	t^{o}				t^{ct}	t^{o}	cyclic
cyclicIfActive		t^{o}	t^{cta}	t^{dt}	t^{nor}	t^{cta}	t^{o}	cyclic
cyclicIfActiveFast	t^{ct}	t^{o}	t^{cta}			t^{cta}	t^{o}	cyclic
cyclicAndSpontanWithDelay	t^{ct}	t^{o}		t^{dt}		t^{ct}	t^{o}	cyclic
cyclicWithRepeatOnDemand	t^{ct}	t^{o}		t^{dt}	t^{nor}	t^{ct}	t^{o}	cyclic
spontan		t^{o}		t^{dt}		t^{dt}	t^{o}	cyclic

interval between two messages and the local offset to all other messages of the same ECU.

2) Send Type Adjustment:

Current vehicle CAN networks use six different timing approaches which are shown in Figure 2a on the left side. These so-called *send types* of messages influence the optimization of the packaging of messages for an Ethernet-based solution, where it is tried to bundle *similar* messages. As a start, we decided to map all send types to a cyclic manner to simplify the timing analysis which is quite complex for a mixture of spontaneous and cyclic send types. Furthermore, about 80% of the messages are already cyclic. Figure 5 shows the distribution of the send types for all messages of different car types. Table II describes the adjustment of the parameters of all deployed send types. For example, the delay time t_m^{dt} of a spontaneous message m which describes the minimal time interval between two messages, is used as the cycle time $t_{m'}^{ct'} = t_m^{d}$ for the corresponding cyclic message m' in the Ethernet-based solution. As a result, we get a set of cyclic messages, each having a cycle time $t^{ct'}$ and an offset $t^{o'}$. The reduction of the send types to just a cyclic send type obviously leads to an oversizing of the system because messages are transmitted more often than actually necessary.

3) Address Resolution:

The function of the address resolution is the readjustment of the message-based addressing to a node-based address mechanism. Figure 2b shows the change over from a shared media bus topology to a full duplex switched Ethernet topology. Ethernet and IP support *uni-*, *multi-*, and *broadcast* to address one, several, or all end nodes within one subnet. We decided to use unicast addressing due to the following reasons: When using broadcast addressing, every ECU is forced to process all incoming packets to check if the packets belong to it, otherwise the packet is being discarded. In the case of multicast, we have to deal with the fact that most of the switches, in particular cheap ones, do not support multicast. They handle multicast addressing as broadcast addressing. In addition, multi- and broadcast addressing uses special address ranges which restrict the freedom of address distribution when designing a distributed system. The decision to use unicast leads to an increment of the number of messages to send, because when a message m in a CAN-based network is received by more than one end node, the Ethernet-based system has to send a packet to each receiver $e^r \in E_m^r$ of this message. The result of this preparation step is a transmitter/receiver-matrix containing all messages from one ECU to another within each cell.

B. Optimization Phase

Table III shows the three optimization variants which try to optimize the packaging within each cell of the transmitter/receiver-matrix. Variant 0 serves as the reference and uses a simple 1:1 transformation strategy which means that every message is encapsulated in its own UDP packet. Variants 1 and 2 use an n:1 transformation strategy to optimize the packaging of CAN frames into UDP packets. This means, that we try to bundle multiple messages into one UDP packet. The assignment is done by an evolutionary- and rule-based approach respectively. UDP is used as a transport protocol due to the reason that a retransmission of cyclic packets does not make sense. Furthermore, we use the UDP port field to identify the receiving processes of a packet p. The result of the optimization step is a set P of packets $p := \{m|\ m \in M\}$ containing an Ethernet, IP, and UDP header. The payload length l_p of a packet p in byte is calculated by Equation 1. In

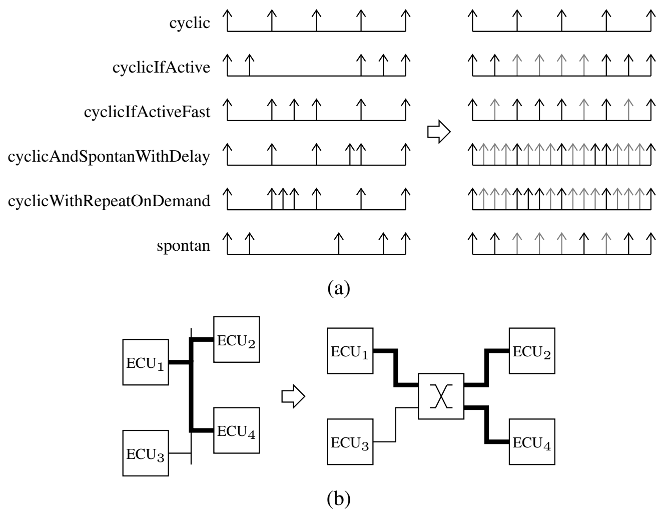


Fig. 2. (a) shows the send time behavior for the different send types on the left side. The adjusted send times for cyclic transmission are on the right side. (b) highlights the topology distinction which influences the address mechanisms.

TABLE III
THE OPTIMIZATION VARIANTS.

variant	0 (reference)	1	2
method	1:1 transformation	n:1 transformation	
	one-to-one mapping	EA-based solution	rule-based solution

the following, we use the term message for the CAN-based solution and packet to identify the Ethernet-based solution.

$$l_p := \sum_{m \in p} l_m(m) \tag{1}$$

1) Variant 0 - One-to-One Mapping:

This variant uses a simple one-to-one mapping which means that every CAN frame is encapsulated into one UDP packet, and is used as a reference implementation. This approach has several disadvantages like a huge protocol header overhead due to relatively large protocol headers of UDP-based systems and a large number of packets which are send on the network. This leads to a higher resource usage when processing more packets and sending more packets in the network. Equation 2 shows the resulting set $P^{1:1}$ of the one-to-one mapping:

$$P^{1:1} = \{p | \forall m_i \in M : p_i = m_i\} \tag{2}$$

2) Variant 1 - EA-based Solution:

They way of finding the optimal solution of the packaging of messages into UDP packets is similar to the set cover problem, which tries to find an optimal set of subsets covering all elements of a given set of objects. The optimization of such a problem is NP-hard, so we decided to use a meta heuristic optimization algorithm to get an approximated solution for our problem. Therefore, we utilized OPT4J [8], which is a Java-based framework implementing an evolutionary algorithm. An integer vector genotype $\vec{g} = \langle n_1, ..., n_i \rangle$, where i is the number of messages m to optimize and n represents the number of the corresponding packet, was chosen to map the memberships of all messages i to their packets in the Ethernet/IP-based solution. The parameters of the EA-algorithm were 500 generations with a population of 100, 50 parents, 25 children, and a cross over rate of 0.95.

3) Variant 2 - Rule-based Solution:

The rule-based approach uses a set of rules to find an approximated solution to our problem and is based on a greedy algorithm. The idea is to bundle similar messages into one UDP-packet with the main intention of minimizing the protocol header overhead. The approach uses a two step design. In the first step, all messages which are similar to each other are combined into one packet. Two messages m_1 and m_2 are defined to be similar, when the cycle times of both messages are identical $t^{ct}_{m_1} = t^{ct}_{m_2}$ and the offsets are within a specific range $\left|t^o_{m_1} - t^o_{m_2}\right| < r$. The first step result is a set P^{rb} of packets p:

$$\begin{aligned} P^{rb} = \{p | \forall m \in M : \exists p \in P^{rb} \wedge m \in p \\ \wedge \forall m_i, m_j \in p : t^{ct}_{m_i} = t^{ct}_{m_j} \wedge \\ \left|t^o_{m_i} - t^o_{m_j}\right| < r\} \end{aligned} \tag{3}$$

In the second step, this approach tries to combine packets p_1 and p_2 with different cycle times $t^{ct}_{p_1}$ and $t^{ct}_{p_2}$ where $t^{ct}_{p_2}$ is a multiple of $t^{ct}_{p_1}$ with $t^{ct}_{p_2} = \alpha \cdot t^{ct}_{p_1}$ and $\alpha \in \mathbb{N}$. With this idea, we can optimize the payload/header-ratio, although the packets with the slower cycle time are sent more often than necessary. Equation 4 calculates the total length $l^s(p_1, p_2, \alpha)$ of two packets sent separately containing all Ethernet, IP, and UDP headers $l^{header} = 54$ byte, fill bytes l^{fill}_p, and messages $m \in p$ in byte. The length of the combined sending $l^c(p_1, p_2)$ in byte is shown in Equation 5.

$$\begin{aligned} l^s(p_1, p_2, \alpha) = l^{header} + l^{fill}(p_1) + l_{p_1} + \\ \frac{1}{\alpha} \cdot \left(l^{header} + l^{fill}(p_2) + l_{p_2}\right) \end{aligned} \tag{4}$$

$$l^c(p_1, p_2) = l^{header} + l^{fill}(p_1 \cup p_2) + l_{p_1} + l_{p_2} \tag{5}$$

The number of fill bytes l^{fill}, which are potentially necessary to obtain the minimum length of an Ethernet-frame, is calculated with the following Equation 6:

$$l^{fill}(p) = \begin{cases} 18 - l_p & \text{, if } l_p < 18 \\ 0 & \text{, otherwise} \end{cases} \tag{6}$$

Figure 3 shows the relationship $q(p_1, p_2, \alpha)$ (see Equation 7) of two packets p_1 and p_2 with different cycle times $t^{ct}_{p_2} = \alpha \cdot t^{ct}_{p_1}$. For example, if there are two packets p_1 and p_2 with the payload lengths $l_{p_1} = 16$ byte and $l_{p_2} = 8$ byte and a cycle time factor of 10, it is better to combine these two packets into one new packet $p = p_1 \cup p_2$ with a cycle time of $t^{ct}_p = t^{ct}_{p_1}$,

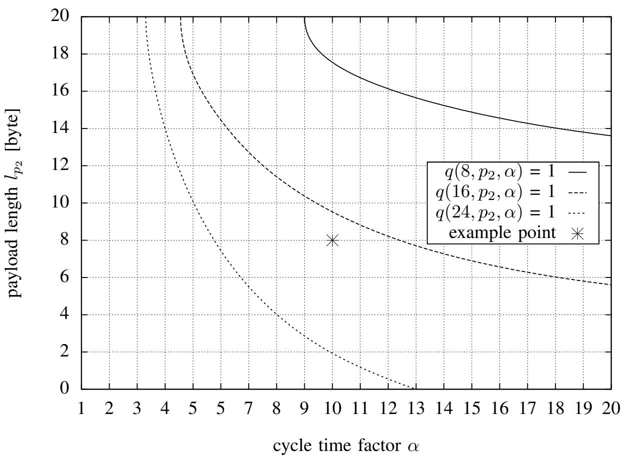


Fig. 3. The relationship $q(p_1, p_2, \alpha) = 1$ for different packet lengths of packets p_1 and p_2 and cycle time factors α. The area below the lines indicates a relationship $q > 1$, otherwise the relationship is $q < 1$.

because of getting a better payload/header-ratio (see example point in Figure 3).

$$q(p_1, p_2, \alpha) = \frac{l^s(p_1, p_2, \alpha)}{l^c(p_1, p_2)} \tag{7}$$

C. Evaluation Phase

The evaluation phase benchmarks the found solution by using several objectives. The considered objectives are the payload/header-ratio q^{ph} (see Equation 8), the maximum packet sizes, the number of packets, or the produced offsets when combining several messages with their different offsets into one packet.

$$q^{ph}(p) = \frac{l_p}{l^{header} + l^{fill}(p) + l_p} \tag{8}$$

Figure 4 shows several objectives for three possible packagings using a simple example with five messages m_1 to m_5. For example, solution number one represents the simple one-to-one mapping of these five messages with a resulting payload/header-ratio of 10 percent and a low offset rating. The combined sending of two messages m_1 and m_2 into one packet p can also produce additional message timing delays, when the two messages have different offsets $t^o_{m_1}$ and $t^o_{m_2}$. In such a case, the earlier message has to be buffered till the other message comes up. This behavior is called offset-rating.

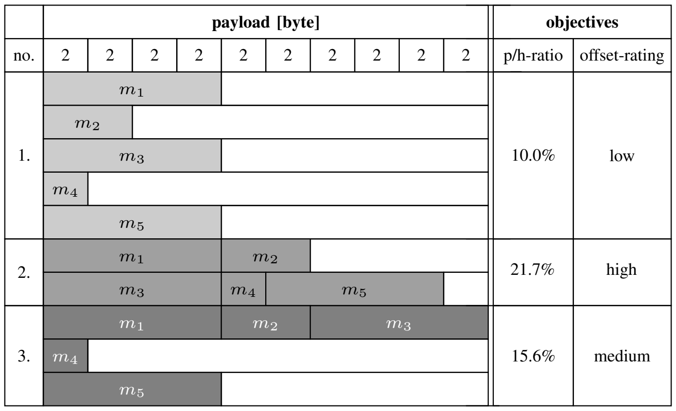
V. Migration Results

The migration results are based on current car network configurations from the BODY-CAN network from different classes of cars: A compact, middle, and premium car was chosen to show results from a minimum-, a middle-, and a maximum-sized real-world network configuration.

Figure 5 shows the results obtained from the preparation phase. The left bar of each car type shows the distribution of the different used send types. The number of different messages varies from 230 to 406 messages and the fraction of the spontaneous messages lies by about 15 percent. The send type adjustment reduces the send types to just a cyclic manner and the address resolution transforms the shared media message-based addressing to a unicast addressing in the Ethernet-based system. This results in a multiplication of the messages by a factor ranging from 2.4 to 2.7 which is represented by the right bar of each car type.

The achieved bandwidths for the different optimization variants are shown in Figure 6. The presented CAN-based values of the bandwidths are calculated after the send type adjustment to include the impact of the send type conversion. It is obvious that the one-to-one mapping leads to the highest bandwidths varying from 14 MBit/s to 25 MBit/s. A bisection of the values could be realized by using the EA- or rule-based optimization. The bandwidth ranges from 8 MBit/s to 11 MBit/s in the case of an EA-based optimization and between 8 and 10 MBit/s when using the rule-based algorithm. The offset range r was set to 40% of the respective cycle times.

Figure 7 shows the payload/header-ratio for the CAN-based solution and the different optimization variants. The



	payload [byte]										objectives	
no.	2	2	2	2	2	2	2	2	2	2	p/h-ratio	offset-rating
1.	m_1										10.0%	low
	m_2											
	m_3											
	m_4											
	m_5											
2.	m_1				m_2						21.7%	high
	m_3				m_4	m_5						
3.	m_1				m_2		m_3				15.6%	medium
	m_4											
	m_5											

Fig. 4. An example of different packagings for five messages m_1 to m_5 with the objectives payload/header-ratio and a offset classification.

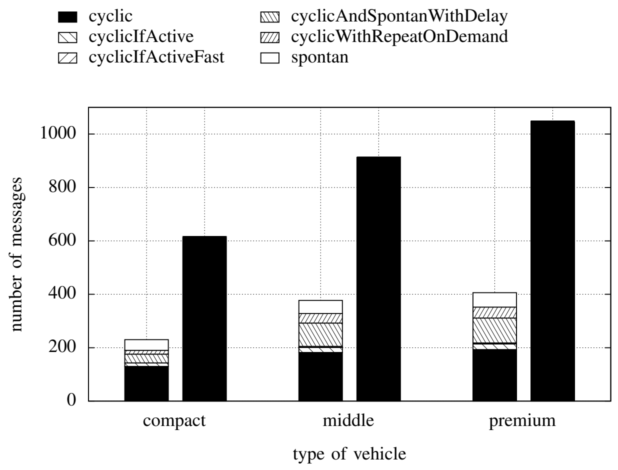


Fig. 5. The left bar shows the send type distribution of actual car network configurations and the right bar presents the results from the preparation phase, namely the send type adjustment and the address resolution.

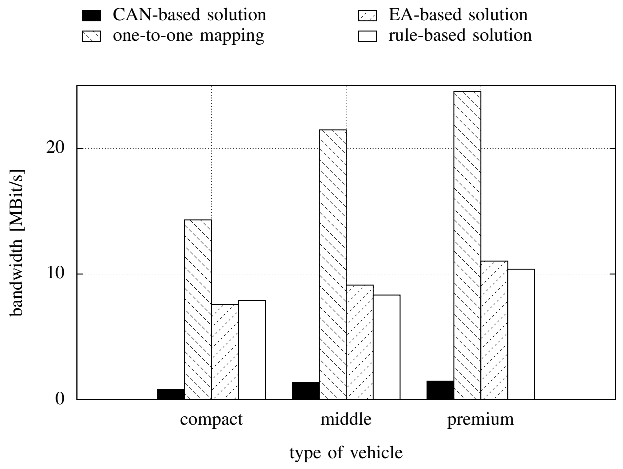


Fig. 6. The required bandwidth of the different optimization variants compared to the CAN-based network design.

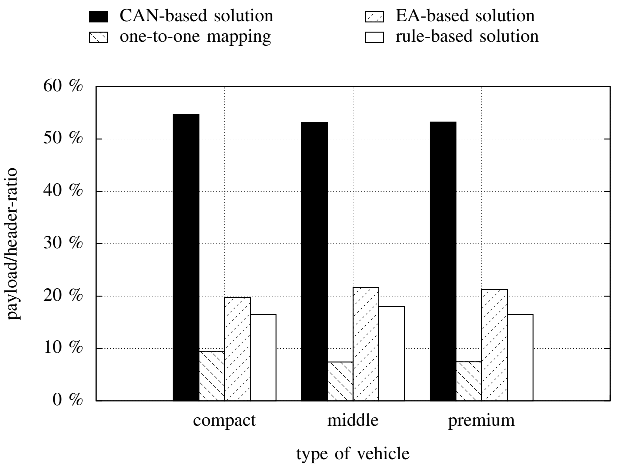


Fig. 7. The payload/header-ratio for the different optimization variants compared to the CAN-based network design.

CAN-based values are again calculated after the send type conversion and are over five times higher compared to the one-to-one mapping. In the case of the EA- or rule-based optimization, the payload/header-ratio values ranges from 16% to 22%.

Table IV summarizes the results obtained from the data structure migration algorithm for the different vehicle types.

VI. Conclusion and Outlook

In this paper, we presented a concept to migrate a CAN-based car network to an Ethernet-based approach. Different optimization concepts were introduced, implemented and evaluated. The shown results were based on current real car network configurations from Daimler of different classes of cars. We demonstrated that it is possible to migrate a complete car subnetwork to a Fast Ethernet network solution even with a very conservative approach, by just using cyclic send types and unicast addressing which results in an oversizing of the target system, when messages are transmitted more often then actually necessary.

TABLE IV
The calculated objectives.

objective	**vehicle type**		
	compact	middle	premium
one-to-one mapping			
bandwidth [MBit/s]	14.31	21.47	24.51
payload/header-ratio	9.37%	7.42%	7.45%
EA-based solution			
bandwidth [MBit/s]	7.56	9.12	11.03
payload/header-ratio	19.76%	21.64%	21.28%
average frames per packet	1.98	2.53	2.34
rule-based solution			
bandwidth [MBit/s]	7.91	8.33	10.39
payload/header-ratio	16.47%	17.98%	16.54%
average frames per packet	1.46	1.74	1.66

Future work will look at using more than just the cyclic send type and other address mechanisms like multicast within the target system. These will result in a better network utilization. But on the other hand, the configuration of the network becomes more complex. We also want to perform timing evaluations of the migration results based on a concrete underlying hardware architecture to measure latencies and jitters which depends on the entire system design by choosing different packet arbitration mechanisms within the network.

Acknowledgment

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) under project 01BV0905-SEIS. The authors are responsible for the content of this publication.

References

[1] CAN/Ethernet-Gateway: *SysTec-Eelectronics CAN/Ethernet-Gateway, http://www.systec-electronic.com/*
[2] FlexRay Consortium GbR (Hrsg.): *FlexRay Communications System – Protocol Specification*. 2.1 Rev. A. Industriestraße 6, 70565 Stuttgart, Germany: FlexRay Consortium GbR, Dezember 2005
[3] IEEE, Inc. (Hrsg.): *Local and metropolitan area networks – Requirements – Part 3: csma/cd detection method and physical layer specs*. IEEE Std 802.3, 2000. 3 Park Avenue, New York, USA: IEEE, Inc., Dezember 2000
[4] ISO: *International Organization for Standardization, http://www.iso.org/*
[5] ISO 13400: *ISO 13400 - Diagnostic communication between test equipment and vehicle over Internet Protocol (DoIP)*
[6] ISO und IEC: *ISO 11898: Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signallingrea network (CAN) for high-speed communication*. ISO 11898-1:2003. November 1993
[7] LIN Administration (Hrsg.): *LIN Specification Package*. 2.1. Bernhard-Wicki-Straße 3, 80636 Munich, Germany: LIN Administration, November 2006
[8] M. Glass, F. Reimann M. Streubühr J. Keinert C. H. M. Lukasiewycz L. M. Lukasiewycz ; Teich, J.: Dependability-Aware System Synthesis: SystemCoDesigner, OPT4J and JRELIABILITY. In: *DATE 2007 Conference*, 2009
[9] MOST Cooperation (Hrsg.): *Media Oriented System Transport – MOST Specification*. Rev. 3.0. 76185 Karlsruhe: MOST Cooperation, Mai 2008
[10] Murphy, Richard: *A CAN to FlexRay Migration Framework*. 2009
[11] Postel, J.: *User Datagram Protocol*. RFC 768. 4676 Admiralty Way, Marina del Rey, California 90291: Information Sciences Institute, University of Southern California, August 1980
[12] Postel, J.: *Internet Protocol – DARPA Internet Program – Protocol Specification*. RFC 791. 4676 Admiralty Way, Marina del Rey, California 90291: Information Sciences Institute, University of Southern California, September 1981
[13] Telecommunications Industry Association (Hrsg.): *Electrical characteristics of low voltage differential signaling (LVDS) interface circuits*. ANSUTIAEIA-644-1995 standard. 2500 Wilson Boulevard, Suite 300, Arlington, VA 22201, USA: Telecommunications Industry Association, März 1996
[14] Traub, Matthias: *Durchgängige Timing-Bewertung von Vernetzungsarchitekturen und Gateway-Systemen im Kraftfahrzeug*, Karlsruher Institut für Technologie, Deutschland, Diss., 2010