CAN FD from an OEM point of view

M. Schreiner, H. Mahmoud, M. Huber – Daimler Research and Development
S. Koç, J. Waldmann – Daimler Truck Development

Abstract – The objective of this paper is to investigate the integration of CAN FD in today`s vehicle E/E-architecture from an OEM point of view. For this purpose physical layer capabilities are discussed and feasible scenarios of implementing CAN FD into mixed vehicle networks are addressed.

Introduction

Launching the first CAN-node in the Mercedes Benz S-Class in 1991 Mr. Moore`s prediction took also place in the vehicle networking context. A growing multitude of distributed E/E-systems made driving safer, more efficient and more emotional. In order to meet the likewise growing requirements in the commercial vehicle market the first CAN-node in a truck was only a matter of time. Only one year later in 1992 Daimler's truck division introduced the first CAN-based network in the heavy-duty segment - the "Schwere Klasse" truck was born. Since then, the number of ECUs and networks in vehicles and the amount of transmitted data over these networks has been constantly growing.
Figure 1 represents the different evolutional steps in terms of in-vehicle networks from its early days to the present using the example of the E-Class.

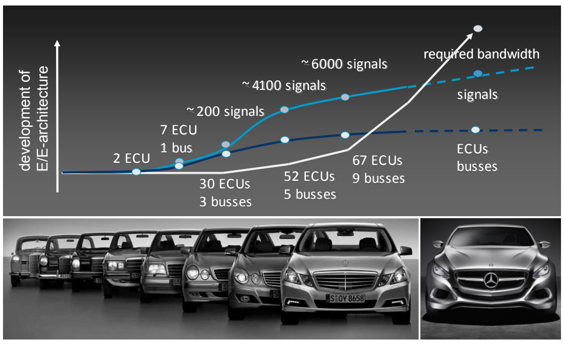


Figure 1: Evolution of in-vehicle networking

During an evolution of more than twenty years vehicle E/E-architectures that once used to be just one single CAN-bus have been extended using multiple CAN-buses interconnected by gateways and additional bus systems like LIN, FlexRay or MOST. Because vehicle networking is not an infrastructure that is directly perceptible by the customer, it is a balancing act between economical pressure and technical innovation that requires the adequate use of specific technologies. Along the lines of "as few as possible, as much as necessary". Figure 2 shows a simplified schematic of the E/E-architecture of the current S-Class (launched in 2013) and the current New Actros truck (launched in 2011) in figure 3.

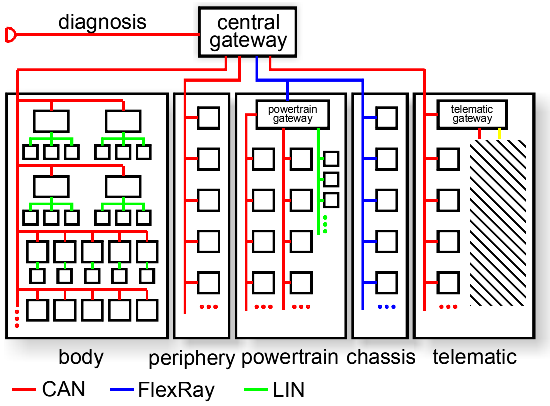


Figure 2: Current passenger car architecture

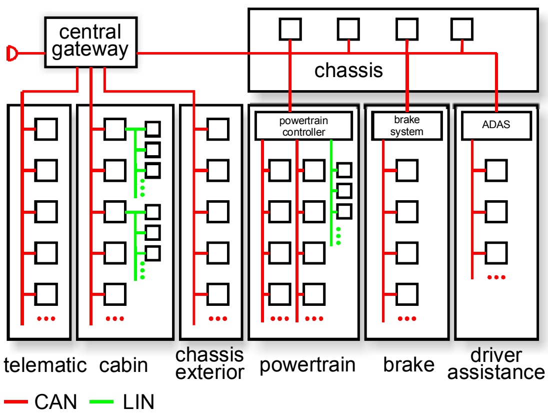


Figure 3: Current truck architecture

Both architectures follow the idea of grouping communicating systems of the same domain on own bus systems to reduce overall busload. Especially on truck systems still a large amount of inter-domain communication requirement (e.g. brake and powertrain system) remains which cannot be satisfied by introducing just another CAN bus system. On the other hand the introduction of non-CAN communication would require a big change of today's software and communication implemented within truck systems. The introduction of CAN FD [1] for these busload-critical CAN-busses seems to be a perfect solution to achieve higher bus capacity without large changes in the existing systems like brake systems or engine controllers.

Of course the next generation of passenger car E/E-architectures is already under development. Different concepts have been published in which a new bus system is discussed as a proper candidate: automotive Ethernet. Figure 4 shows a scenario of an E/E-architecture using Ethernet as a backbone interconnecting the different domains of the vehicle that are already visible in today's topologies. Such structures could be used in E/E-architectures as of approximately 2020. Even though an additional bus system will be introduced, it is noticeable that in many domains CAN will still remain the dominating bus system used by most common ECUs.

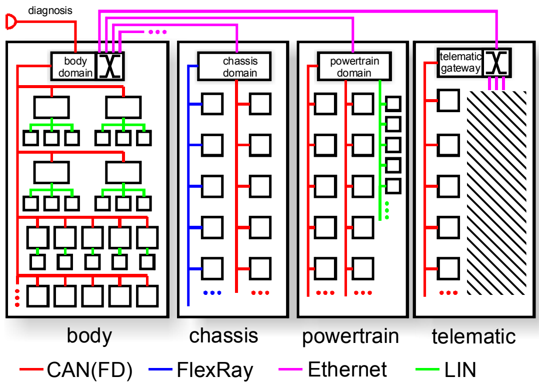


Figure 4: Future passenger car architecture

This next evolutional step will also have its impact on CAN. On the one hand there will be a need for more bandwidth in architectures like figure 4. On the other hand such a new architecture and especially Ethernet itself requires new communication mechanisms. Especially the usage of small data packets with at most 8 bytes is inefficient with a bus like Ethernet. Thus much larger payloads can be used. CAN-technology has to be integrated into these new mechanisms which is obviously a challenge for the grown structure of ECU's communication software stack. In the following it will be

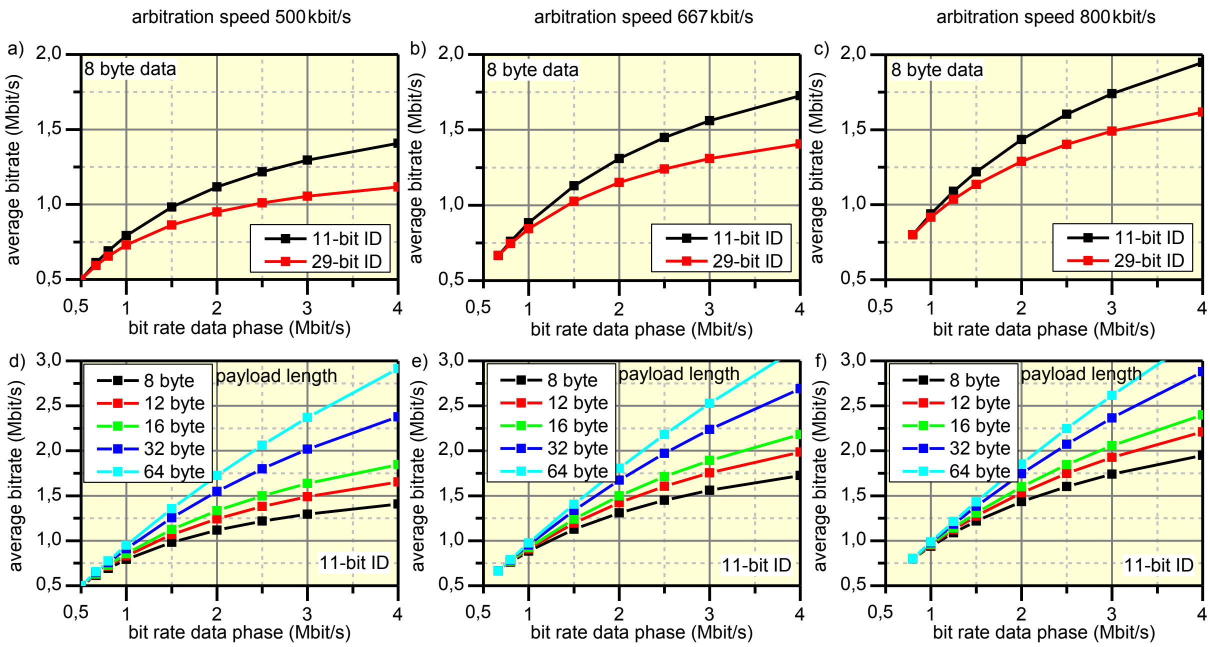


Figure 5: CAN FD average bit rate depending on data phase speed and ID length

shown that CAN FD [1] is not only a solution to increase the bandwidth, it also offers the possibility to use CAN-technology more efficiently in future E/E-architectures.

CAN FD Physical Layer Aspects

Estimation of average bit rates

CAN FD introduces in principle two major extensions to CAN2.0: 1. faster bit rate in the data field and 2. extended payload length up to 64 bytes. The average bit rate that can be achieved is determined by the speed of the control field and by the speed of the data field [2]. Additionally the average bit rate depends on payload length and identifier length (11-bit or 29-bit). The correlations are given in figure 5. The average bit rate which is shown in the graphs can be compared directly with today's CAN2.0 bit rates, e.g. 500 kbit/s. Please note that this estimation does not include stuff bits.
The upper graphs in figure 5 are plotted for arbitration speeds between 500 kbit/s and 800 kbit/s. The x-axis represents the bit rate in the fast data phase of a CAN FD frame, while on the y-axis the resulting average bit rate is plotted, assuming that only 8 byte payload frames are used. Figure 5a shows that the average bit rate could be nearly doubled by an arbitration speed of 500 kbit/s and 2 Mbit/s for the data phase using only 8-byte data frames and 29-bit IDs. There is more gain in average bandwidth using only 11-bit IDs.
The lower graphs in figure 5 show the effect of the extended payload length, assuming that all transmitted frames make completely use of the respective payload. It is evident, that the gain in average bit rate is maximized when frames with long payload are used: e.g. in a network with an arbitration speed of 500 kbit/s and a speed of 2 Mbit/s in the data phase using 8 bytes payload would give just a little less than 1 Mbit/s average bit rate. However using only 64 byte of payload yields a little more than 1,5 Mbit/s average data rate as shown in figure 5d. This means an increase of approximately 50% in average data rate.

As can be seen from the graphs in figure 5 all curves go into saturation which means that the accomplishable increase of average communication speed with CAN FD is limited.

How to evaluate and design CAN FD networks

The most important key parameter for evaluation of CAN2.0-networks is the propagation delay in the network. This value is limited by the CAN-protocol mechanisms and the respective bit time settings. In simple terms all nodes within a network need to receive the response of any other node to their own signal within a bit time. If the delay is too large CAN-arbitration and acknowledge mechanisms fail and as a consequence the communication on the CAN-bus breaks down completely. To make sure that this does not happen under any circumstance all communication relationships in a network between all nodes are assessed e.g. by means of measurement or physical layer network simulation. The maximum delay time in the network (TX to RX) is extracted and the signal integrity on the bus is checked as well in order to make sure that the predicted values are stable. If there is ringing in the network that could further enlarge the delay time the predicted delay values have to be adopted. Figure 6 shows how the evaluation can be done by means of a signal integrity chart. Of course there will always be an adequately defined safety margin to account for tolerances, EMC or temperature influence.

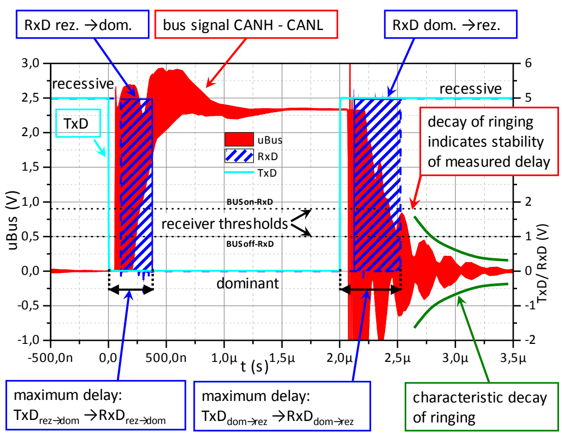


Figure 6: Signal integrity diagram

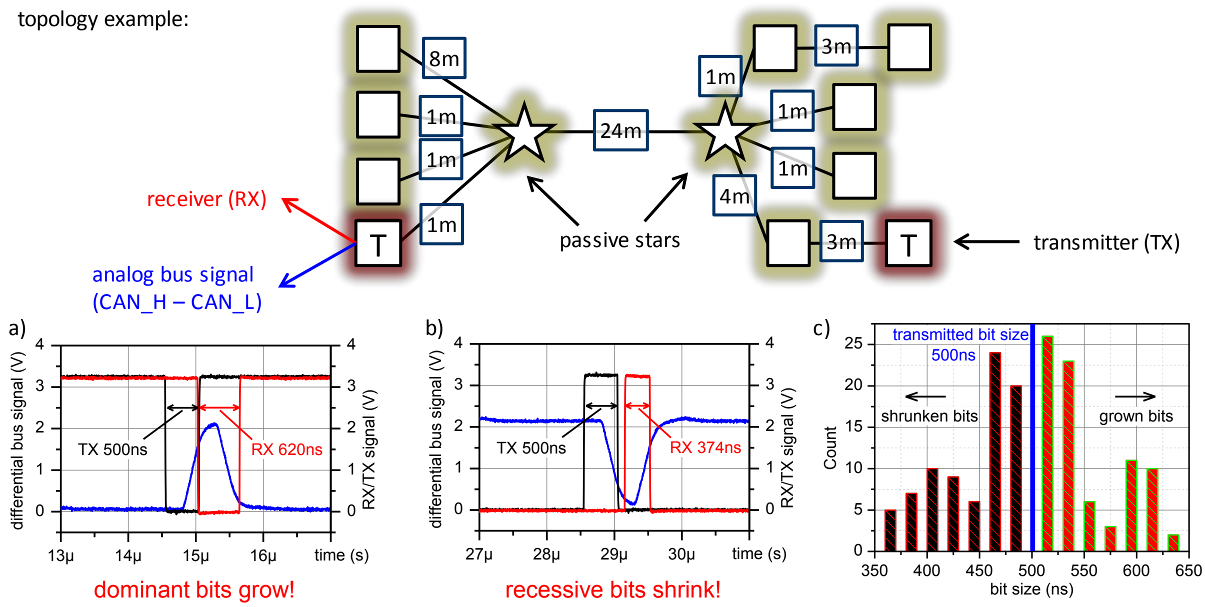


Figure 7: Example for bit asymmetry in a topology with CAN FD 2Mbit/s

Propagation delay is also an important key parameter for CAN FD networks since arbitration and acknowledge mechanisms are identical for CAN2.0 and CAN FD. That's why CAN FD networks are limited in terms of arbitration speed just like CAN2.0-networks are.

In the phase of accelerated data transmission of a CAN FD frame the delay values are not relevant as all other nodes are already synchronized and just listen to the transmitted data. However other key parameters can be identified for CAN FD frames that have not been considered for the CAN2.0 even though the effects are present in CAN2.0-networks as well. The most important is the asymmetric delay of the received signals in the network that becomes relevant especially for higher bit rates. This effect is due to the fact, that the rising and the falling edges of a dominant signal have different physical preconditions, i.e. the recessive to dominant edge is driven actively whereas the dominant to recessive edge is just released. In the end depending on the transceiver used dominant or recessive bits shrink or grow. The exact value may even be dependent on the previously transmitted signals. Figure 7 gives an example of bit asymmetry measured in a real network.

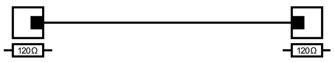
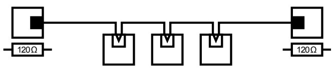
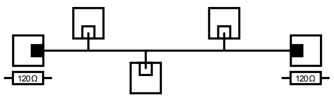
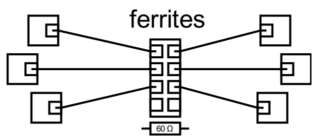
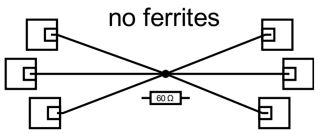
Depending on the bit time settings of the network bit asymmetry will cause communication errors due erroneous sampling of the bits. The total asymmetry is a combination of the intrinsic asymmetry of the transceivers and the specific characteristics of the topology. Up to now there is no official tolerance range of the intrinsic asymmetry of the transceivers themselves. Just like symmetric delay values in CAN2.0 implementations there has to be an adequately defined safety margin for the asymmetric delay to account for tolerances, EMC or temperature influence.

Rules of Thumb

The design rules given in table 1 should only serve as a rule of thumb for the reader. They show the typical operating range for arbitration and data speeds of CAN FD systems with typical topologies in an automotive environment. The maximum values can only be achieved if the limiting values are carefully confined. In practice most topologies cannot be handled with simple standardized topologies like those shown in table 1. That's why in practice all topologies have to be checked in the design process e.g. by means of physical network simulation.

It should be noted, that all communication speeds beyond 2Mbit/s in the data phase need newly designed transceivers with better controlled symmetry which are

Table 1: CAN FD topology design rules – range of operation depending on topology

topology	limiting values	arbitration speed (bit/s)	FD data speed (bits/s)
point 2 point 	• bus length	125k – 1M	1M – 4M
ideal bus without stubs 	• bus length • number of mid nodes	125k – 1M	500k – 2,5M
bus with stubs 	• bus length • maximum stub length • number of nodes • distance between nodes	125k – 667k	500k – 2M
passive star with ferrites 	• number of nodes • maximum length to nodes • total length of all nodes • type of ferrites used	125k – 667k	500k – 2M
passive star without ferrites 	not recommended with CANFD ≥ 1Mbit/s		

currently not yet available. Arbitration speeds on CAN FD networks beyond 500kbit/s are only possible, if the network does not need wake functionality, since current transceivers (ISO11898-5) would not wake up at higher arbitration speed.
Generally the data rates for CAN FD data phase given in table 1 are much lower than those demonstrated during exhibitions or in laboratory experiments. At a first glance the values are disappointing however they take account for all parasitical effects that vehicle electronics is exposed to, i.e. tolerances depending on temperature, manufacturing processes of semiconductors or cable harnesses, ageing, electromagnetic immunity and emission etc.
The values in Table 1 also show that CAN FD will never redundantize FlexRay, especially if the resulting average bit rates according to figure 5 are compared. Nevertheless with today's physical layer the average bit rate can be approximately doubled; if the topologies and transceivers would be optimized it may even be tripled or even quadruplicated.
Even more bandwidth can be achieved with CAN FD if an increased payload length is used. As it will be shown in the following chapter the combination of extended payload length in combination with a moderate increase of average bit rate makes CAN FD a very interesting technology for the evolution of vehicle E/E-architectures.

Scenarios for an effective integration of CAN FD into vehicles` E/E architecture

As shown in the previous section, the introduction of CAN FD offers additional communication capabilities. However the practical integration requires an adaptation of the entire software stack of ECUs. Depending on the respective solution

more or less parts of the ECU software are affected. Especially passenger cars use the AutoSAR (automotive open system architecture) software stack in their ECU software.
AutoSAR is following the principle: cooperate on standards, compete on implementation. With the increasing number of complex functions, the development of vehicle electronics is becoming broader in scope and more complicated in detail. AutoSAR was founded in 2003 by automotive OEMs and suppliers as a partnership with the objective to create and establish open standards for automotive E/E-architectures that will provide a basic infrastructure to assist developing vehicular software, user interfaces and management for all domains. This includes the standardization of basic systems function, scalability to different vehicles or platform variants as well as transferability throughout the network.
In the following three introduction scenarios for CAN FD into ECU software are discussed. In addition the impact on the ECU software is explained exemplarily for the AutoSAR software stack used in passenger cars in which the principle is also valid for other ECU software solutions. All Daimler vehicles (trucks, buses or passenger cars) make use of several bus systems that are interconnected via gateways. Thus not only the communication within one network but also the routing between several CAN-networks (which might be CAN FD and CAN2.0) or between other bus systems like Ethernet or FlexRay has to be considered for the introduction of CAN FD.

Scenario 1: Increase communication speed and maintain 8-byte payload per frame

This scenario called CANFD8 (=payload remains limited to 8-byte) will be introduced for the AutoSAR release 4.1.1. Figures 8, 10 and 14 show the AutoSAR software stack. Blue shaded boxes indicate that the respective component has to be adopted. In scenario 1 only the communication speed is increased, all other communication software mechanisms are maintained. As shown in figure 8 the only software component that is affected is the CAN-driver.

- Expansion of the CAN-driver to enable the configuration of the second baud rate
- Additional attributes required in the system description (bit time settings)

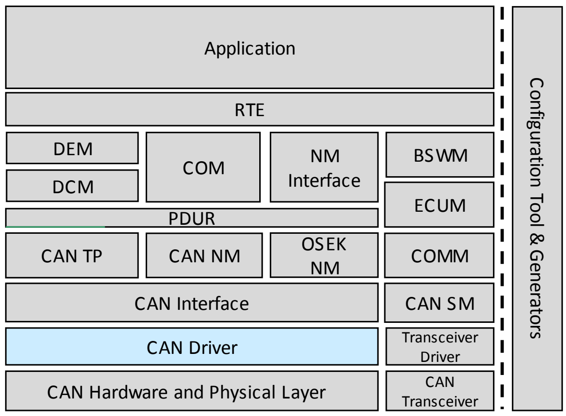


Figure 8: Software stack changes using CAN FD 8

Low busload reserves do not occur on all CAN-busses simultaneously which might result in a mixed CAN FD / CAN2.0 structure inside passenger cars or trucks. As long as a payload length of 8-byte is used on all CAN-networks, routing between CAN FD- and CAN2.0-networks is easy as shown in Figure 9:

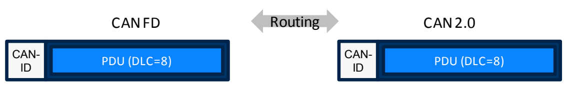


Figure 9: Routing scenario with CAN FD 8

PDUs (protocol data unit = bare payload of an application / CAN-frame without any additional control information) can be routed from CAN FD to CAN2.0 and vice versa without further considerations and even the same CAN-IDs could be used on both networks. This type of routing called “frame routing” is easy to implement and contained in communication standard software stacks. The routing mechanism to other bus systems such as Ethernet or FlexRay would be maintained as well without changes.

It has been shown that CAN FD 8 will approximately double the average baud rate compared to today's CAN. Thus scenario 1 might be a first step for the introduction of CAN FD.

Scenario 2: Increase communication speed and use 64-byte payload per frame

On the other hand as shown in figure 5 frames with extended payload (CAN FD 64) provide significant additional bandwidth. Because of this the extension of the payload to 64 bytes will be addressed in AutoSAR 4.2.1. As shown in figure 10 this scenario has an extensive impact on several software components in the ECU software stack.

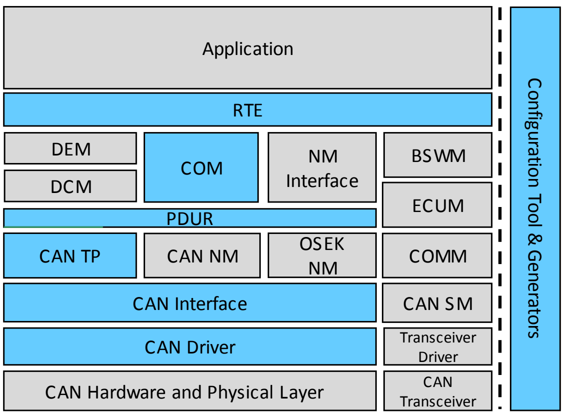


Figure 10: Software stack changes using CAN FD 64

In detail the following changes have to be applied:

- Expansion of the CAN-driver to enable the configuration of the second bit rate
- Expansion of the CAN-modules, PDUR, COM and RTE to support 64-byte payload.
- Expansion of the System Description / ECU extracts the ECU to support 64-byte payload
- Expansion of the configuration tools and generators to support 64-byte payload

As the payload length is limited to 8-bytes for CAN2.0, generally multiple messages have to be used to transport the original PDU from a CAN FD frame. Unfortunately this type of routing is not implemented in current communication standard software stacks, which means that every single CAN-signal contained in the original PDU has to be treated separately by the routing mechanisms. On the other hand, the possibility to realize separate and independent transmission cycles for the CAN-ID1 to CAN-IDn frames can be gained on the CAN2.0 network bought dearly by an increased router processor load. Ongoing standardization tends to implement this routing scheme within AutoSAR specification based on a signal routing scheme in the router.

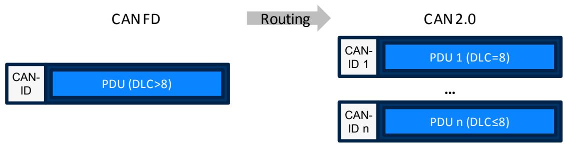


Figure 11: Routing scenario with CAN FD 64

A slightly different situation occurs if the PDU definition is changed on CAN FD networks. Originally, the PDU identifies the frame content of a CAN-message less the control information like e.g. message ID, acknowledge bit or stuff bits. For the simplification of the CANFD to CAN2.0 routing, the basic idea is a CAN FD frame containing multiple PDUs with a DLC (data length code) of 8 if compatible to CAN2.0 messages.

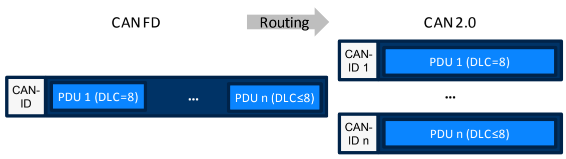


Figure 12: Routing scenario with CAN FD 64

As shown in Figure 12, routing scheme looks the same as in Figure 11 routing a CANFD frame to multiple CAN2.0 frames. But differently, a fixed relation between the PDUs in the CAN FD frames and the routed CAN2.0 PDUs/frames can be realized. This PDU routing scheme can be implemented within communication standard software stacks with a high reduction of routing processer interrupt load resulting in an increased routing

capacity. Furthermore, there is a static relation between CANFD frames and their contained PDUs. This procedure could be used as a migration strategy for vehicle architectures with CAN2.0 and CAN FD networks.
Currently there is the limitation in the AutoSAR software that only one single PDU can be mapped to one single CAN-frame. If this restriction will be maintained, it would be very difficult to use CAN FD effectively. In passenger cars, most applications currently using the CAN2.0 bus only generate a comparatively small amount of data that is designed to effectively use 8-byte PDUs today. Only a certain amount of applications could really generate large PDUs efficiently using 64-byte CAN FD frames.
Generally PDUs should be filled with signals that are transmitted with identical cycle times. On the contrary PDUs should not be filled by different software components in order to be able to map the relocatability of functions on a bus level. Due to different senders, transmission types, cycle times etc. there is no point in combining arbitrary signals of an ECU into large PDUs in the practical application.
In trucks and busses, communication is more based on cyclic signals and due to technical or legislative requirements, more data has to be exchanged between systems (e.g. exhaust relevant communication between engine controller and exhaust after treatment control unit to ensure Euro IV conformity).
Thus scenario 2 theoretically offers the possibility to make use of the extended payload frames CAN FD offers, however depending on the grown structure of the vehicles' software applications and the needed communication relations between systems, only a certain increase of bus capacity can be achieved. Therefore the ability to map multiple PDUs dynamically into one CAN FD frame appears to be an additional requirement.

Senario 3: Expand communication facilities with flexible PDU mapping

As a first step multiple 8-byte PDUs could be mapped statically into one single CAN FD frame. However the efficiency of such a solution would be poor with the transmission mechanisms specified in the present AutoSAR software package. As an example: In case that four PDUs are statically mapped to one CAN FD frame and only one of the PDUs has been updated this solution would imply that the entire CAN FD frame has to be transmitted including three PDUs without new information. Like scenario 2 this solution would make ineffective use of CAN FD possibilities. Therefore, as a second step, a so-called PDU-Header (similar to currently specified in Ethernet) could be introduced, see Figure 13.

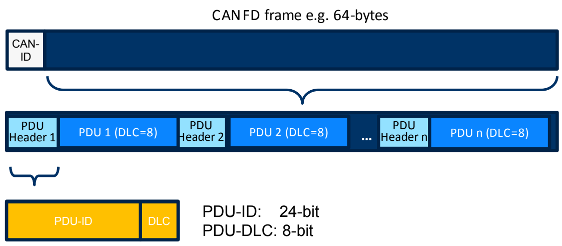


Figure 13: PDU header concept

The introduction of this PDU header allows a dynamic mapping of PDUs to CAN FD frames. In this case only those PDUs are transmitted in a CAN FD frame whose contents have actually changed. There will be no redundant transmission of unchanged PDUs. The PDUs that are contained in a current CAN FD frame can be identified clearly by means of the PDU header. Furthermore the length of the CAN FD frame can be adapted dynamically depending on the current communication needs. This method allows using the possibilities of the CAN FD technology in a quite effective manner, even though some bandwidth gets lost for the additional PDU headers. Secondly it fits into the grown structures of the ECU software structure. Figure 14 again highlights the software components that have to be adopted.
It has to be mentioned, that the PDU header concept also implies the loss of bandwidth due to the PDU headers in the CAN-message's payload. E.g. if five 8-byte PDUs are transmitted in an 64-byte CAN FD frame, 60-bytes of the frame are really used and another 20-bytes get lost

for the PDU headers resulting in an effective usage of 62,5% compared to the complete usage of 64-bytes payload. This would especially apply to the grown applications in the E/E-architecture whereas new applications could make use of larger PDUs where the loss is much less, e.g. approx. 93% efficiency for a single 60-byte PDU + 4-byte header.

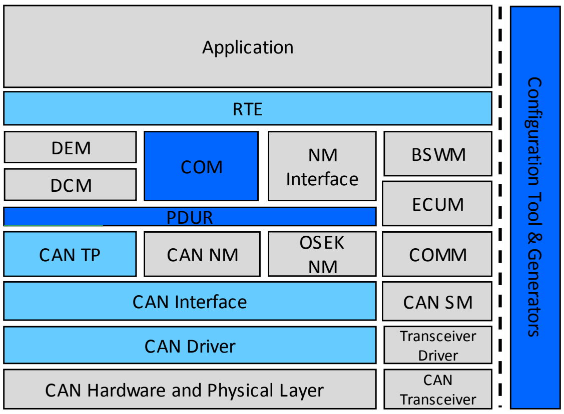


Figure 14: Software stack changes using CAN FD 64 and PDU routing

Routing of CAN-frames between networks containing CAN FD and CAN2.0 using the PDU header concept is shown in figure 15.

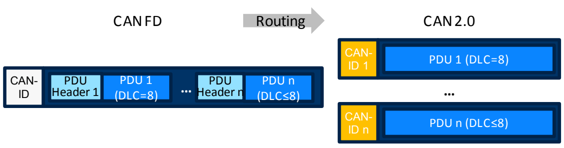


Figure 15: Routing scenario with CAN FD 64 and PDU header concept

E.g. Figure 15 shows the routing of a CAN FD frame with a DLC > 8 from a CAN FD network towards a CAN2.0 network. In this case, the router gains some more flexibility and reduces lookup table memory requirements as the CAN FD frame could carry the CAN2.0 destination CAN-ID information within the PDU header. Only in case of a multi-router with more than 2 CAN-networks, the router needs a lookup table for the selection of the destination network (e.g. CAN-ID1 should be routed to CAN2.0 network no. 1 while CAN-ID2 should be routed to CAN2.0 network no. 2 only). Of course this destination network information has to be available at the router also in the previous discussed routing schemes.

Discussing the routing schemes from a CAN2.0 network to a CAN FD network, the principle stays the same. A CAN FD frame is combined from multiple CAN2.0 frames depending on the PDU arrangement and the router scheme (PDU routing or signal routing). But another aspect rises up as for the case of DLC>8 or multiple PDUs on CAN FD: the timing of the arriving CAN2.0 frames has to be considered additionally. Multiple approaches may be used varying in routing latency for the different PDUs. Depending on the specific application of the routed signals, the appropriate routing timing scheme has to be selected (e.g. starting routing process at first incoming CAN2.0 frame, starting after arrival of last incoming CAN 2.0 frames). In every case, buffer memory for single signals or complete PDUs has to be provided within the router.

A very similar concept is currently under discussion for automotive Ethernet. Even though PDU headers and maximum PDU sizes are larger in this case the principle is the same. Thus scenario 3 described in this paper also enables the possibility to have common routing mechanisms for Ethernet and CAN FD.

Conclusion

During the evolution of in-vehicle E/E-networks there has always been an increasing demand for more communication bandwidth to meet the future requirements of new applications. As a next step new communication systems like automotive Ethernet will be introduced in future. Nevertheless this trend will not replace well established CAN-bus networks. CAN-networks will also play a major role for future vehicle networking. But it is evident, that the bandwidth on CAN-networks has to be increased as well. Secondly it would be desirable to keep CAN-networks compatible with new networking and routing concepts that will be introduced with systems like automotive Ethernet into the E/E-architecture.

The possibilities of CAN FD match very well with these requirements. As it has

been shown with today's physical layer and vehicle topologies it is possible to use data rates of approximately 1 Mbit/s up to 2 Mbit/s for the data part of a CAN FD message under vehicle ambience conditions.
Secondly the extended payload length of up to 64-bytes per CAN FD frame provides the necessary flexibility to adopt networking and routing concepts and fit CAN FD into the grown structure of ECU software. It has been shown, that the increase of system performance is limited if today's CAN-software mechanisms based on 8-byte messages are applied to CAN FD. Using the payload of 64-byte frames – with or without the concept of flexible communication mechanisms using PDU headers – enables network designers to make the most of CAN FD technology.

Dr.-Ing. Marc Schreiner
Daimler AG – Research and Development
Wilhelm-Runge-Straße 11
DE-89081 Ulm

Haytham Elwy Mahmoud
Daimler AG – Research and Development
Bela-Barenyi-Str. 1
DE-71063 Sindelfingen

Martin Huber
Daimler AG – Research and Development
Bela-Barenyi-Str. 1
DE-71063 Sindelfingen

Serhan Koç
Daimler AG – Truck Development
Mercedesstr. 123
DE-70372 Stuttgart

Dr.-Ing. Jan Waldmann
Daimler AG – Truck Development
Mercedesstr. 123
DE-70372 Stuttgart

References

[1] CAN with Flexible Data-Rate - Florian Hartwich, CAN in Automation, iCC 2012
[2] Safeguarding CAN FD for applications in trucks - M. Schreiner, H. Leier, M. Zerzawy, T. Dunke and J. Dorner, CAN newsletter March 2013