Part 2: Comparing CAN, CAN FD, and Ethernet

This analysis compares Classical CAN, CAN FD, and Ethernet systems with focus on a decentralized battery management system. Part 1 of this article was published in the CAN Newsletter magazine issue 2-2021.

In Part 1 the test environment setup as well as the evaluation criteria for the comparative analysis were introduced. Part 2 provides related results and discussion.

Results and discussion

Frame processing time comparison: The frame processing times were first measured for different user data lengths of 0 byte, 8 byte, 64 byte, and 1 500 byte aligned with the maximum frame sizes of CAN, CAN FD, and Ethernet (Table 1). The frame processing time was measured using CAN with 500 kbit/s, CAN FD with 500 kbit/s, 1 Mbit/s, and 4 Mbit/s as well as Ethernet with 100 Mbit/s.

The highest frame processing duration for a user data transmission of 0 byte up to 8 byte is consistently CAN FD with 500 kbit/s. At the identical transmission rate as CAN, CAN FD is on an average 27 µs slower because of the longer control field (9 bit vs. 6 bit) in the CAN FD frames. The main advantage of CAN FD is the bit-rate switch (BRS), i.e. the increased data rate compared to the arbitration rate. This is not exploited in case of a constant transmission rate of 500 kbit/s, but at a transmission speed of 1 Mbit/s and higher, the transmission time is reduced and CAN FD reveals its advantage over Classical CAN.

Since only the transmission time of the data phase is reduced, the effect of the BRS increases with an increasing number of user data (Table 1). The increased data rate is used for the user data and the CRC (cyclic redundancy check) field. A further increase of the CAN FD data rate to 4 Mbit/s shows a considerable improvement of the frame processing time, which is especially effective with a high amount of transmitted data bytes.

Table 1: Measured frame processing times for various transmission rates and user data size (Source: OTH Regensburg)

Message processing time in μs					
User data bytes	CAN (500 kbit/s)	CAN FD (500 kbit/s// 500 kbit/s)	CAN FD (1 Mbit/s// 1 Mbit/s)	CAN FD (1 Mbit/s// 4 Mbit/s)	Ethernet (100 Mbit/s)
0	99.2	126.2	91.3	69.5	44.3
1	114.5	140.7	99.3	71.2	43.8
2	133.1	158.5	108.7	73.5	43.7
4	163.4	190.9	125.2	78	44
8	227.6	257.4	158.4	86.7	44
16	-	389.5	222.6	103.9	44.1
32	-	657.2	355.8	139.4	46.8
64	-	1171.4	611.8	208.3	53.4
1500	-	-	-	-	314.7

The frame processing time of CAN FD is at any time significantly higher than the one of Ethernet. At 64 user data bytes, the time difference between CAN FD (4 Mbit/s) and Ethernet has increased to 155 µs. These differences become even more striking when the payload exceeds the frame size, i.e. multiple segments are necessary.

With a focus on the fastest possible data transmission, Ethernet is the communication technology of choice. Even with a payload of up to 64 byte, the data transmission is faster than with CAN FD. In addition, Ethernet enables user data transmission of up to 1 500 byte per data frame, which is higher than the maximum of 8 user data byte to be transmitted with CAN and the maximum of 64 user data byte with CAN FD (Figure 1). Comparing CAN FD and CAN, it becomes clear that CAN FD only shows advantages over CAN when bit-rate switching is used and the data rate is increased. Furthermore, it is shown that the effect of bit-rate switching becomes more significant as the payload increases.

When selecting the communication technology, the number of user data byte to be transmitted and the ratio between user data byte and overhead due to the data frame should be taken into account. If the objective is to transmit as much user data as possible within minimum time, the advantages of Ethernet outweigh those of CAN and CAN FD. For the transmission of fewer user data byte in short intervals, as it is the case for example within the DBMS (decentralized battery management system), CAN FD is definitely an option, especially if the transmission rate is further increased up to the maximum of 8 Mbit/s.

Processor workload analysis: The processor time is measured to compare the processor load caused by the respective communication technology. A part of the measurement is the required processor time for initialization. The initialization is executed once at system startup or after a reset. The initialization time for CAN (FD) is approx. 40 µs, whereas Ethernet requires 1,7 s and is over the factor 40 000 greater than the initialization time of CAN (FD) (Figure 1). Especially if an unexpected restart of the micro-controller and the applied real-time system occurs, the long downtime of the subscriber has consequences, including loss of information and the associated effects on system control and coordination. The Ethernet initialization time of 1,7 s (Figure 1) is quite long compared to the one for CAN (FD). The reason for this is the built-in Ethernet PHY (physical layer) on the applied evaluation board. By replacing the Ethernet PHY, the Ethernet initialization time could lie within the millisecond range. ▷

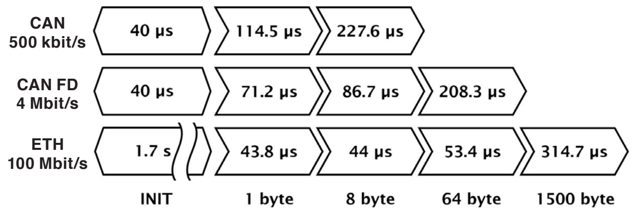


Figure 1: Initialization and frame processing time for varying user data size (Source: OTH Regensburg)

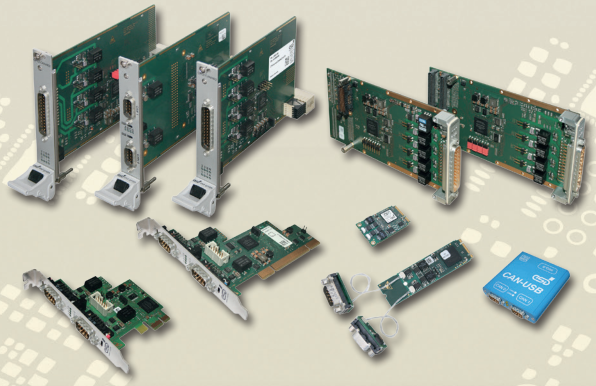
Next, the processor load for CAN (FD) communication for sending and subsequent receiving of 1 000 data frames is measured. Varying numbers of frames showed a linear behavior between the CAN FD process and the number of frames. The processor time of the CAN FD process increases with the number of frames.

The processor times were measured for intervals of 0,2 ms, 0,3 ms, 1 ms, and 10 ms. For shorter intervals between frames, the processor time share for the CAN FD thread increases slightly. With the shortest frame interval of 0,2 ms, the share of the CAN FD thread process is still only 1,5 %. The measurements show that the CAN FD communication only places a low overall load on the processor and cannot overload the processor due to communication channel restrictions limiting the transmission speed beforehand.

Measurements for CAN communication showed that it requires 10 µs less processor time for 1 000 data frames compared to CAN FD communication. Examining the percentage processor utilization, there are no differences compared to CAN FD communication. Therefore, only the CAN FD measurements are used for the comparison with Ethernet. Receiving and unpacking CAN (FD) data frames is handled by an interrupt service routine and depends on the length of the user data. For example, for 8 byte of user data it takes 3,3 µs whereas it takes 7,8 µs for 64 byte (Figure 5).

The measurement of the processor times for sending of Ethernet frames shows that the processor times do not behave linearly, especially for a few frames, in contrast to the processor times for CAN (FD). The reason for this are threads, which are executed in a fixed time interval independent of the number of frames. The shorter the frame interval, the more processor time is required by the Ethernet threads. With a send interval of 10 ms the Ethernet communication requires only 1,8 % of the processor time while with a send interval of 0,2 ms it already requires 23,8 % of the processor time. If the increase is linear to the send intervals, it is possible that the processor utilization limit occurs before the minimum send interval limit of Ethernet, which is 5 µs. This behavior was not investigated in this work because such a short send interval is not necessary for the DBMS.

The measurements show that Ethernet communication requires significantly more processor time than CAN (FD) communication. At a transmission interval of 0,2 ms, Ethernet requires 23,8 %, while CAN (FD) requires only 1,5 %. The processor times are additionally strongly dependent on the transmission intervals. The processor load therefore depends on the communication technology and additionally on the selected frame interval. For the selected minimum frame interval, neither Ethernet nor CAN (FD) communication significantly loads the processor. ▷



For receiving and unpacking of Ethernet frames, a separate thread and an interrupt service routine is used. For 8 byte of user data it takes 18 µs, which is significantly longer compared to the duration of 3,3 µs for CAN (FD) frames. With the maximum of 1 500 byte of user data, receiving and unpacking the frame takes 83,9 µs.

Energy consumption comparison: Since the same peripheral units are used for CAN and CAN FD communication, the energy consumption is combined for both types of communication. The processor operates constantly with a frequency of 400 MHz and an operating core voltage of 1,2 V. The operating voltage of the peripherals is 3,3 V. The total energy consumption is calculated in each case, including the necessary peripheral units, system components, and the supply of external peripherals. For Ethernet communication, significantly more peripheral units and system components are required and additionally the cache is utilized.

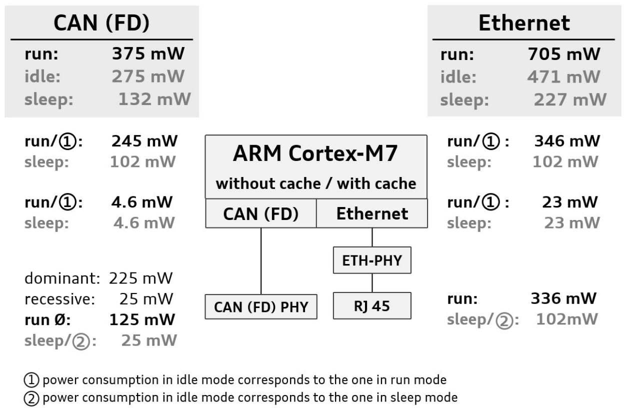


Figure 2: Energy consumption of CAN (FD) and Ethernet for diverse peripheral units in run, sleep and idle mode [5], [9]–[12] (Source: OTH Regensburg)

The energy consumption is calculated in run, idle, and sleep mode (Figure 2 and Figure 3). CAN (FD) communication requires significantly less power than Ethernet communication in all three modes. In run mode, the power required for CAN (FD) of 375 mW is just 53,2 % of the power required for Ethernet. Similarly, in sleep and idle mode, the power required for CAN (FD) is only about 58 % of the power required for Ethernet. For Ethernet and CAN (FD) the sleep mode is significantly more efficient than the idle mode. Only 48 % of the power required for idle mode is consumed in sleep mode.

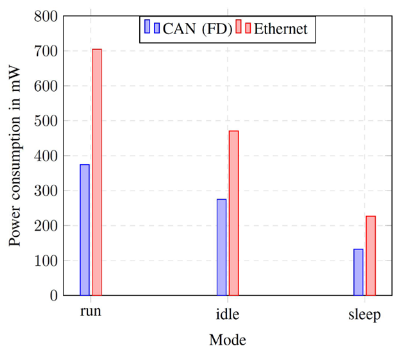


Figure 3: Power consumption comparison [5], [9]–[12] (Source: OTH Regensburg)

For energyefficient applica-tions, CAN (FD) communication shows clear advantages with almost 50 % lower power consumption than Ethernet. For all three communication technologies, using sleep mode instead of idle mode shows a significant improvement in energy efficiency. If Ethernet is required due to its significantly higher transmission rate, the sleep mode should be considered with regard to energy efficiency.

Error rates: The error rates provide information about the reliability and the correctness of data transmission. The RER (residual error rate) of CAN (FD) communication is officially specified as 4,7 · 10^{-11} [13], [14], whereas no official data is available for 100BaseT Ethernet. Direct code analysis (DCA) is used to determine the RER, which depends mainly on the polynomial of the CRC, the BER (bit error rate), and the frame length [15], [16]. DCA generates all possible error patterns, resulting in a sharp increase in computational effort with data length. Figure 4 shows the upper and lower limits of the RER for CAN with an 8-byte data frame and a 15-bit CRC as well as for Ethernet with a 42-byte data frame and a 23-bit CRC [15].

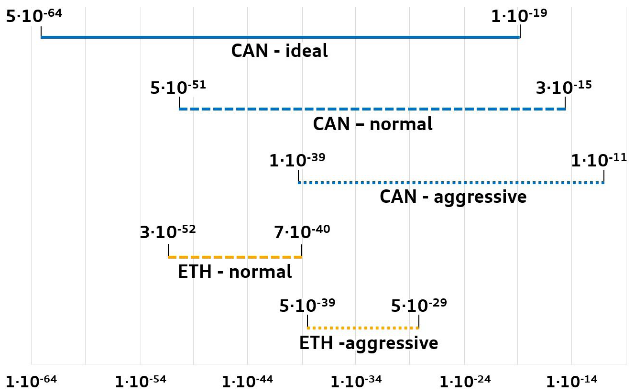


Figure 4: Upper and lower limits for RER of CAN [17] and Ethernet [16] in different environments determined by using DCA [15] (Source: OTH Regensburg)

To determine the BER, a large number of frames are sent under different conditions and the number of bit-erroneous frames is obtained. Table 2 shows the BER for CAN at eight user data byte [17] and for Ethernet at 1 468 byte of user data [16].

Table 2: BER of CAN and Ethernet for different environments [16] (Source: OTH Regensburg)

Environment	BER of CAN	BER of Ethernet
Ideal	$3.0 \cdot 10^{-11}$	-
Normal	$3.1 \cdot 10^{-9}$	$3.35 \cdot 10^{-10}$
Aggressive	$2.6 \cdot 10^{-7}$	$3.0 \cdot 10^{-8}$

Ethernet has a calculated maximum RER of 7 · 10^{-40} while CAN has a RER of up to 3 · 10^{-15}. Ethernet shows clear advantages in the RER, which has a greater impact due to non-detection, compared to the BER. All RER values are significantly below the limit value of 10^{-7} defined in ISO 61508 and ISO 26262, which has to be observed for communication technologies in critical applications. The BER of Ethernet is also smaller than that of CAN by a factor of about 10. Ethernet therefore shows an advantage with regard to the error rate.

Rx-Fifo load: To test the Rx-Fifo load, random frames were generated. For CAN, CAN FD, and Ethernet no noticeable Rx-Fifo load occurs. Table 3 shows the frame processing time, which includes frame generation, transmission, and unpacking of the frames, compared to the pure frame receiving and unpacking time. When ▷

comparing these times and taking into account that all communication participants share a common communication bus, it is evident that no significant Rx-Fifo load occurs (Figure 5).

Table 3: Comparison of the frame processing time and the frame unpacking time (Source: OTH Regensburg)

	CAN	CAN FD	Ethernet
Transmission rate:	500 kbit/s	4 Mbit/s	100 Mbit/s
Send interval:	300 µs	200 µs	120 µs
User data:	8 byte	64 byte	1500 byte
Message processing time:	227.6 µs	208.3 µs	314.7 µs
Message receiving and unpacking time:	3.3 µs	7.8 µs	83.9 µs

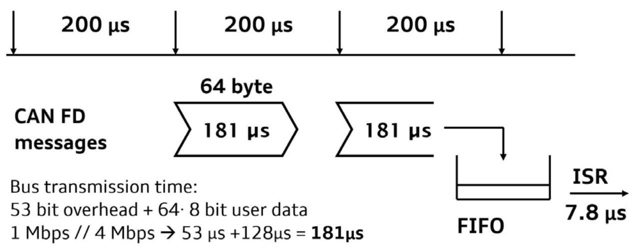


Figure 5: Comparison of the transmission time and the receiving and unpacking time of CAN FD frames with 64 user byte at an interval of 200 µs. The arbitration bit-rate is 1 Mbit/s and the data transmission rate is 4 Mbit/s (Source: OTH Regensburg)

The Rx-Fifo utilization increases with the frame receiving and unpacking time. CAN is particularly advantageous here with a receiving and unpacking time of 3,3 µs for the maximum 8 byte of user data. Ethernet takes 83,9 µs for receiving and unpacking 1 500 byte of user data, which is significantly longer in comparison to CAN (FD). The ratio of the unpacking time to the frame processing time of 314,7 µs and the minimum possible transmission interval of 121 µs, still indicates that the Rx-Fifo is not significantly loaded.

Conclusion

The application of the communication technologies Classical CAN, CAN FD, and Ethernet in networked control systems and especially in the DBMS was evaluated on the basis of the criteria frame processing time, processor load, power consumption, error rate, and Rx-Fifo load. In terms of frame processing time, Ethernet showed significant advantages due to its high transmission rate of 100 Mbit/s (Table 4). When transmitting 8 byte of user data, Ethernet is approximately 80 % faster than CAN and 50 % faster than CAN FD. CAN FD already shows clear benefits over Classical CAN. For transmission of 8 byte of user data, CAN FD with a data rate of 4 Mbit/s is 60 % faster than CAN.

None of the investigated communication technologies noticeably loads the processor. With a transmission interval of 200 µs, CAN (FD) communication requires only 1,5 % of the processor load, while Ethernet requires 24 %. For energy-efficient applications, CAN (FD) is preferred as it consumes 50 % less power than Ethernet. In terms of ▷

Table 4: Comparative analysis of CAN, CAN FD, and Ethernet (Source: OTH Regensburg)

	CAN	CAN FD	Ethernet
Payload data	low	low - medium	high
Message processing time	+	++	+++
Processor workload	+++	+++	+
Energy consumption	+++	+++	+
Error rate	++	++	+++
Rx-FIFO load	+++	+++	++

error probability, Ethernet offers advantages with at least a 10-fold lower error occurrence, whereas all communication technologies are suitable for safety-critical systems due to their low error probability. CAN, CAN FD, and Ethernet did not show any noticeable load on the Rx-Fifo due to the short receive and unpacking times in relation to transmission times.

Ethernet offers excellent characteristics in the transmission of large data amounts and in the error rate, but it requires significantly more power and processor time. The long initialization time of Ethernet is critical for real-time systems and networked control systems. For these applications, an Ethernet implementation without lwIP has to be considered. CAN FD has significant benefits in terms of frame processing time, even with few user data, and requires the same amount of power and processor time as CAN. In addition, CAN FD offers the possibility to transmit up to 64 user data byte and to increase the data rate even further. For these reasons, CAN FD is evaluated as the most appropriate communication technology for the DBMS. For applications with a higher number of user data byte, Ethernet is suitable.

Outlook

In further comparative analyses, CAN XL [20], Energy Efficient Ethernet (EEE) [22], and Ethernet Time Sensitive Networking (TSN) will be considered. CAN XL, with a maximum user data transfer of 2 048 byte and data transfer rates of up to 10 Mbit/s, is the successor to CAN FD. With the higher data rates and number of user data, the gap between CAN and Ethernet is steadily closing. EEE is an extension of Ethernet with the aim of reducing power requirements. Ethernet TSN is a standard extension of the IEEE with the aim of achieving real-time capability of Ethernet. Among other things, real-time capability is enhanced by time synchronization, prioritization, scheduling, traffic shaping, and resource reservation [23] and is promising for networked control systems. ◀

This article is split in two parts. If you have missed it, in the June issue of the CAN Newsletter magazine you can read Part 1. This article was originally presented as a paper at the Embedded World Conference 2021 Digital.

References

[5] Datasheet, STMicroeletronics: STM32H742xI/G STM32H743xI/G, https: //www.st.com/resource/en/datasheet/stm32h743bi.pdf, 2019.
[6] Datasheet, Microchip: MCP2561/2FD, http://ww1.microchip.com/downloads/en/devicedoc/20005167c.pdf, 2014.
[7] Datasheet, Ixxat: USB-to-CANFD.
[8] VN16xx,Datasheet,Vector, https://assets.vector.com/cms/content/products/VN16xx/docs/VN16xx FactSheet DE.pdf
[9] STMicroelectronics: UM1718 User Manual. Rev32. 2020.
[10] Microchip: MCP2561/2FD. 2014.
[11] Microchip: LAN8742A/LAN8742Ai. 2013.
[12] STMicroeletronics: RM0433 Reference manual. Rev 7. 2020.
[13] Bosch: CAN Specification, 1991.
[14] Bosch: CAN with Flexible Data-Rate Specification, 2012.
[15] M. Rahmani, W. Hintermaier, B. Muller-Rathgeber and E. Steinbach,"Error Detection Capabilities of Automotive Network Technologies and Ethernet - A Comparative Study", 2007 IEEE Intelligent Vehicles Symposium, Istanbul, 2007, pp. 674-679, doi: 10.1109/IVS.2007.4290194.
[16] A. Kern: "Ethernet and IP for Automotive E/E-Architectures", Doktorarbeit, Universit at Erlangen-N urnberg, 2012.
[17] J. Ferreira: "An Experiment to Assess Bit Error Rate in CAN", Polytechnisches Institut, International Workshop on Real-Time Networks RTN 2004, Castelo Branco, Spain, 2004.
[18] A. Schön: "Und am Anfang war das Ethernet... die Unterlage des Internet", http://www.franken.de/fileadmin/mediapool/kongress/2001/ethernet.pdf, KNF Kongress, 2001.
[19] Vector "VN16xx—Factsheet—Vector", [Online] Available at: https://assets.vector.com/cms/content/products/VN16xx/docs/VN16xxFactSheet DE.pdf
[20] A. Mutter, CAN in Automation, "CAN XL specifications and test plans", https://www.can-cia.org/groups/technical-groups/technical-committee-tc/ig-lower-layers/sig-can-xl/, 2020.
[21] P. Decker, O. Garnatz, "CAN XL für zukünftige Fahrzeugarchitekturen", Hanser Automotive 4-5/2020, https://assets.vector.com/cms/content/ know-how/technical-articles/CAN XL Introduction Hanser automotive 202007 Pressarticle DE.pdf, 2020.
[22] N. Balbierer, T. Waas, J. Noebauer and J. Seitz, "Energy consumption of Ethernet compared to automotive bus networks", 2011 Proceedings of the Ninth International Workshop on Intelligent Solutions in Embedded Systems, Regensburg, Germany, 2011, pp. 61-66
[23] L. Lo Bello and W. Steiner, "A Perspective on IEEE Time Sensitive Networking for Industrial Communication and Automation Systems", in Proceedings of the IEEE, vol. 107, no. 6, pp. 1094-1120, June 2019, doi: 10.1109/JPROC.2019.2905334..

Authors

Andrea Reindl, Daniel Wetzel,,Norbert Balbierer, Hans Meier, Michael Niemetz
Faculty of Electrical Engineering and Information Technology - Ostbayerische Technische Hochschule Regensburg
Contact the authors via andrea.reindl@st.oth-regensburg.de
www.oth-regensburg.de
Sangyoung Park
Smart Mobility Systems - Technical University of Berlin
sangyoung.park@tu-berlin.de
www.sms.tu-berlin.de


KVASER
Rugged,
Powerful,
Intuitive.
CAN FD to USB.
Introducing
THE KVASER U100
LEARN MORE
TX
RX
01000177
KVASER