

Comparing Time-Triggered Ethernet with FlexRay: An Evaluation of Competing Approaches to Real-time for In-Vehicle Networks

Till Steinbach, Franz Korf, Thomas C. Schmidt
Department of Computer Science
Hamburg University of Applied Sciences, Germany
{till.steinbach, korf, schmidt}@informatik.haw-hamburg.de

Abstract

FlexRay is considered the next generation state-of-the-art technology for in-car networks, while time-triggered Ethernet emerges with the promise to integrate real-time and best-effort traffic into one homogeneous backbone. This paper contributes a competitive analysis of FlexRay and time-triggered Ethernet. By showing that it is possible to transfer a fully utilized FlexRay system to a system based on time-triggered Ethernet, it is demonstrated that time-triggered Ethernet is a suitable replacement of current in-vehicle bus-systems. Further it is shown that a switched system has advantages in bandwidth utilization over a shared bus, when using group communication.

1. Introduction

Today's vehicles are complex distributed real-time systems with a high demand of broadband communication links. In such systems, the communication flow between modules has significant impact on safety, reliability, and comfort of the vehicle. New proposals for automotive systems require supervising control units for specific tasks that are connected across a backbone network. These networks need to carry heavy load, while keeping message delays predictable.

Currently, FlexRay [1] gains importance as state-of-the-art technology for a backbone of in-vehicle networks designed to meet hard real-time constraints. It is the first automotive protocol that allows offline scheduling for time-triggered traffic and randomly occurring event-triggered data in parallel. For each message, the predefined schedule defines an exact, system-wide timestamp at which this message will be visible on the FlexRay bus. FlexRay complies to the enhanced temporal requirements of the new vehicle chassis control applications, which could not be satisfied with busses like CAN [2].

A new approach to in-vehicle networks is based on Ethernet, which has proven to be a flexible, highly scalable protocol. Current Ethernet — in contrast to FlexRay — is a technology based on switching that allows to increase the amount of traffic simultaneously transferred by using segregated communication in groups. However, due to its randomised access and best effort approach, it does not provide reliable temporal performance bounds. There

are many attempts to overcome those obstacles like token-based, bandwidth-limiting or time-triggered systems.

For a smooth integration into current designs, this paper focuses on time-triggered Ethernet systems [3]. The results of this paper are based on the time-triggered Ethernet (TTEthernet) specification by TTEch [3] that is currently proposed for standardisation by the Society of Automotive Engineers [4]. Nonetheless, the results are mostly transferable to other time-triggered Ethernet protocols, as well. TTEthernet supports several traffic classes with different qualities for the various real-time related metrics. This offers the chance to decrease the heterogeneity of today's in-vehicle communication systems, that consist of several CAN, FlexRay or MOST [5] busses connected via gateways, by aggregating several systems into one Ethernet based communication network.

This paper offers a competitive analysis of FlexRay and TTEthernet by showing the differences and commonalities of both technologies. Based on a mathematical model, the general eligibility of TTEthernet for in-vehicle applications is shown in the scenario, when a fully utilised FlexRay system is replaced by a time-triggered Ethernet substitute. Further, it is discussed that a switched system has advantages in utilization, when using group communication. Besides general differences like components and topology, the analysis covers comparisons for real-time relevant metrics like latency, jitter and bandwidth.

The remainder of this paper is organised as follows. In section 2 related work and the base techniques are introduced. Section 3 compares the performance of the protocols under common real-time related metrics. Finally, section 4 concludes and gives an outlook.

2. Background

Related Work In process automation, there is already a large collection of products that rely on Ethernet and promise real-time communication. For in-vehicle communication, the use of Ethernet is a novelty.

In general, there are *token-based, bandwidth limiting and time-triggered protocols* to ensure a predictable delay and jitter in packet delivery and prevent real-time packets from simultaneously traversing a switch interface or line card. Time-triggered protocols are well known within the automotive industry and based on previous experiences

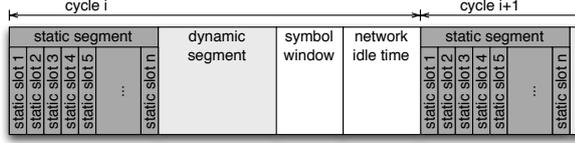


Figure 1. Composition of FlexRay cycle

with TTCAN [6] and FlexRay. Therefore they are in focus of this paper. All senders in time-triggered protocols are required to operate in a synchronised way. A schedule defines permissible transmission times. In process automation, these protocols dominate deployment. Examples are Profinet [7], SynqNet [8] or RTnet [9].

FlexRay FlexRay has been developed by the FlexRay consortium [1] starting in the year 2000. Especially new requirements by the chassis control led to a system with synchronisation and a predefined schedule that allows operation without collision and thus lower bounds for latency and jitter.

The FlexRay protocol is centred about periodic cycles. Each cycle is composed of a static and a dynamic segment (see figure 1), governed by different bus access policies. For time-triggered communication, each node is assigned to communication slots in the static segment. Delivery depends on a global time across all participants and thus a precise synchronisation. The dynamic segment is intended for prioritised event-triggered communication. It is not intended for hard real-time communication and is therefore not subject of this paper.

FlexRay allows for bus, star or mixed topologies and defines two separate channels for transmission. Both can be used simultaneously, leading to a total bandwidth of 20 Mbit/s or with one channel for redundancy at 10 Mbit/s.

Time-Triggered Ethernet TTEthernet [3] combines standard Ethernet based best-effort network traffic and hard real-time communication on the same infrastructure. It adds, similar to FlexRay, time slotting to standard switched Ethernet. On layer 2 a global out of band schedule that defines a plan for transmission or relaying in the time-triggered domain, is shared among all TTEthernet participants and switches. The TTEthernet specification defines a failsafe synchronisation protocol to distribute a global time among all senders. Besides the time-triggered traffic, TTEthernet offers rate-constrained traffic based on the AFDX-Protocol [10] for communication with less rigid temporal requirements, similar to the messages in FlexRays dynamic segment and best-effort traffic.

TTEthernet relies on switched Ethernet. Any topology is formed of switches that relay the messages. Redundancy can be achieved by multiple redundant channels.

3. Comparison

Based on the configuration given in table 1, time-triggered functionality of FlexRay and TTEthernet are compared. This scenario represents the maximal topology of FlexRay [11]. Analysis of other scenarios that base on current car networks is subject of upcoming work. The framework given here can be adapted to these topologies.

Table 1. Sample configuration

	FlexRay	TTEthernet
bus speed (V_B)	10 $\frac{Mbit}{s}$	100 $\frac{Mbit}{s}$
max payload size (l_P)	254 <i>Byte</i>	1500 <i>Byte</i>
min payload size (l_P)	1 <i>Byte</i>	46 <i>Byte</i>
topology	two active stars/switches	
wire length (l_W)	72 <i>m</i>	
divergence of quartz (Δ_Q)	200 <i>ppm</i>	
cycle time (t_C)	16 <i>ms</i>	

Latency Since FlexRay is a wired bus, latency consists of the signal runtime from sender to receiver and the transfer time for the message. Equation 1 calculates the signal runtime t_S for the FlexRay sample configuration. It is composed of the propagation delay of the wire t_{WD} , maximum delays for sending and receiving at bus drivers t_{DD} , and the product of the number of active stars n_S and their maximum delay t_{SD} . Considering our sample configuration with a wire length of l_W and a topology with two active stars n_S we get a delay of $t_S = 1420 ns$.

$$\begin{aligned}
 t_S &= t_{WD} * l_W + 2 * t_{DD} + t_{SD} * n_S & (1) \\
 &= 10 \frac{ns}{m} * 72 m + 2 * 100 ns + 250 ns * 2 \\
 &= 1420 ns
 \end{aligned}$$

The transfer time depends on frame size and bus speed. In FlexRay, the frame size l_F is composed of the transmission start sequence l_{TS} , the frame start sequence l_{FS} , the protocol overhead l_{PO} , the frame end sequence l_{FE} and the payload l_P . For synchronisation of FlexRay controllers each byte is encoded in 10 *bit*. The transmission start sequence must fill the time needed to change state from idle to active at sender, receiver and at each active star and the time to transmit one bit. For the sample configuration, a sequence of $l_{TS} = 15 bits$ is suitable.

$$\begin{aligned}
 l_F &= l_{TS} + l_{FS} + l_{PO} + l_P * 10 bit + l_{FE} & (2) \\
 &= 15 bit + 1 bit + 80 bit + l_P * 10 bit + 2 bit \\
 &= 98 bit + l_P * 10 bit
 \end{aligned}$$

$$l_{Fmin} = 98 bit + (1 * 10 bit) = 108 bit \quad (3)$$

$$l_{Fmax} = 98 bit + (254 * 10 bit) = 2638 bit \quad (4)$$

With the bit time $t_b = \frac{1}{V_B} = 0.1 \frac{\mu s}{bit}$, the transfer time t_T can be calculated.

$$t_T = l_F * t_b \quad (5)$$

$$t_{Tmin} = 108 bit * 0.1 \frac{\mu s}{bit} = 10.8 \mu s \quad (6)$$

$$t_{Tmax} = 2638 bit * 0.1 \frac{\mu s}{bit} = 263.8 \mu s \quad (7)$$

Together with the signal runtime $t_S = 1420 ns$ a total latency between 12.2 μs and 265.2 μs was calculated.

Latency in TTEthernet is caused by switching delays. TTTech indicates the switching delay t_{SD} for their current implementation between 1 μs and 2.4 μs . Total latency can be calculated as the sum of propagation delay $t_{WD} * l_W$, transmission time $l_F * t_b$, and the sum of each switch delay between sender and receiver. Due to store and forward switching the transmission time must

be added for each switch. The bit time for 100 Mbit/s is $t_b = \frac{1}{V_B} = \frac{1}{100 \frac{Mbit}{s}} = 0.01 \frac{\mu s}{bit}$.

$$t_L = t_{WD} * l_W + (n_s + 1) * l_F * t_b + \sum_{i=1}^{n_s} t_{SD_i} \quad (8)$$

The time for message transmission $l_F * t_b$ ranges from $5.8 \mu s$ for the smallest Ethernet frame to $122 \mu s$ for a frame with maximum payload. The latency for TTEthernet t_L for the topology with two switches can be approximated by equation 8 with $24 \mu s$ to $372 \mu s$.

Jitter We calculate an upper bound of possible jitter for FlexRay by evaluating the maximum time difference in network traversal. The receiving node can experience jitter at the scale of clock deviations between the nodes. Since FlexRays clock synchronisation takes place after each cycle, equation 9 calculates the drift with the precision of the quartz Δ_Q and the cycle time t_C .

$$\begin{aligned} t_{Jmax} &= \frac{2 * t_C * \Delta_Q}{(1 - \Delta_Q)} \\ &= \frac{2 * 16 \text{ ms} * 0.0002}{1 - 0.0002} = 6.4 \mu s \end{aligned} \quad (9)$$

In TTEthernet each forwarding instance in the path of a message has a schedule for transmission time. The latency can vary when the local clocks of sender and forwarding switches drift apart. The longest runtime between sender and receiver is given when the clock of the sender gains and the clock of the forwarding switch loses. The shortest runtime is obtained when the senders clock loses and the forwarding switches clock gains. Accordingly, the maximum jitter for TTEthernet messages t_{Jmax} can be calculated as twice the maximum clock deviation t_{Dmax} .

$$t_{Jmax} = 2 * t_{Dmax} \quad (10)$$

TTTech indicates t_{Dmax} for the current TTEthernet implementation in the low unary microsecond span. Hence the jitter bounds are comparable (see table 2).

Bandwidth Since only real-time communication is of interest for the comparison, the optional dynamic segment and symbol window of FlexRay are omitted to maximise bandwidth for static transmission slots. The size of the static segment t_S can then be calculated as the cycle time t_C without the network-idle time t_I (see figure 1).

$$\begin{aligned} t_S &= t_C - t_I \\ &= 16000 \mu s - 12 \mu s = 15.988 \text{ ms} \end{aligned} \quad (11)$$

To calculate the static slot size t_{ST} , the channel idle delimiter l_{CI} and the action point offset t_A are needed. The action point offset allows inaccuracies in synchronisation. Based on the precision, an action point offset of $t_A = 4 \mu s$ is chosen. To account for the optimal ratio of payload to protocol overhead, the largest payload of $l_P = 254 \text{ Byte}$ is chosen. This leads to a frame size of $l_{Fmax} = 2638 \text{ bit}$, as shown in equation 4. t_b is the time needed to transfer one bit at 10 Mbit/s. In FlexRay the slot size is defined in macroticks (MT). In the sample configuration one

macrotick is $t_{MT} = 1 \mu s$.

$$\begin{aligned} t_{ST} &= 2 * t_A + \frac{(l_{Fmax} + l_{CI}) * t_b + t_{WD} * l_W}{t_{MT} * (1 - \Delta_Q)} \quad (12) \\ &= 2 * 4 \text{ MT} \\ &\quad + \frac{(2638 \text{ bit} + 11 \text{ bit}) * 0.1 \frac{\mu s}{bit} + 0.9 \mu s}{1 \mu s * (1 - 0.0002)} \\ &= 8 \text{ MT} + 265 \text{ MT} = 273 \text{ MT} \end{aligned}$$

With equation 13, the maximum number of slots n_S can be calculated.

$$n_S = \left\lfloor \frac{t_S}{t_{ST}} \right\rfloor = \left\lfloor \frac{15988 \text{ MT}}{273 \text{ MT}} \right\rfloor = 58 \quad (13)$$

Equation 14 calculates the net bandwidth B with the maximum payload of $l_P = 254 \text{ Byte}$ for a fully utilised FlexRay bus in the sample configuration.

$$\begin{aligned} B &= \frac{n_S * l_P}{t_C} = \frac{58 * 254 * 8 \text{ Bit}}{16 \text{ ms}} \quad (14) \\ &= 7.366 \frac{\text{Mbit}}{\text{s}} \end{aligned}$$

To show the feasibility of embedding the traffic of the sample FlexRay configuration, TTEthernet is configured with the same cycle time t_C , number of slots n_S and payload size l_P . Together with the Ethernet protocol overhead (header, preamble and trailer) the total frame size evaluates to $l_F = 280 \text{ Byte}$. The time between two consecutive frames t_A is extended from the standard Ethernet inter frame gap of $0.96 \mu s$ to $t_A = 4 \mu s$ to allow inaccuracies in synchronisation. With equation 15, the slot size t_{ST} for one TTEthernet real-time message can be calculated. For the desired $n_S = 58$ slots this results in a total time for real-time communication of $t_S \approx 1.764 \text{ ms}$. The TTEthernet cycle of the sample configuration is utilised with approximate $n_C = 11\%$. As expected due to bus speed the FlexRay real-time traffic can be embedded in TTEthernet.

$$\begin{aligned} t_{ST} &= \frac{l_F}{V_B} + 2 * t_A \quad (15) \\ &= \frac{280 * 8 \text{ Bit}}{100 \frac{\text{Mbit}}{\text{s}}} + 2 * 4 \mu s = 30.4 \mu s \end{aligned}$$

$$t_S = n_S * t_{ST} = 58 * 30.4 \mu s \approx 1.764 \text{ ms} \quad (16)$$

Discussion Figure 2 shows the available number of real-time slots for frames with payload sizes between 1 Byte and the maximum payload size, as derived from the previously defined parameters. Based on the available slots, the possible bandwidth for FlexRay and TTEthernet can be evaluated as is shown in figure 3. Due to the overhead of both protocols, a gain of bandwidth can be seen for packets with higher payload. Due to the larger protocol overhead and padding of small Ethernet frames, TTEthernet benefits especially from frames bigger than the minimum frame size. The padding can be seen in figure 2 as a constant interval. For the same interval, the increase of bandwidth is linear (figure 3). FlexRays bandwidth can be directly read off the given plot, since the static segment only allows for one fixed frame size at all static slots. TTEthernet allows to vary frame sizes for all messages. This leads to a mixed calculation over all used frame sizes.

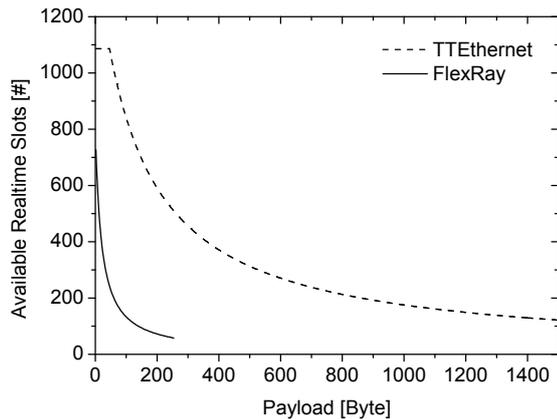


Figure 2. FlexRay and TTEthernet slots at payload size (16ms cycle)

Most in-vehicle networks can be partitioned in groups with communication restricted to group members, and messages that are passed only within those groups. While FlexRay is limited to broadcast communication, TTEthernet inherits multicast group communication from the Ethernet standard. As long as the links between the members of different groups remain disjoint, messages can be relayed concurrently. Plotted results therefore reflect the worst case results.

For the maximum payload of FlexRay (254 Bytes) we showed that a comparable TTEthernet design is only utilised by approximately 11%, while all messages remain of broadcast type. This corresponds to empirical observations considering the ten times higher bandwidth of 100 Mbit/s Ethernet and the slightly higher protocol overhead. TTEthernet is not offered in a 10 Mbit/s version. Because of the larger protocol overhead, TTEthernet with 10 Mbit/s would only keep up with FlexRay with a certain amount of traffic transmitted in parallel.

As shown, latency in TTEthernet is strictly predictable for each time-triggered message, but in general depends on the path from sender to receiver. In contrast to FlexRay, the usable bandwidth is not affected by the latency. Latency remains low as compared with the usual cycle time. Besides the topology, the latency depends on the switching technology and the precision of clock synchronisation. Jitter depends on the precision of clock synchronisation, as well. This holds for both FlexRay and TTEthernet.

Table 2. Sample configuration results

	FlexRay	TTEthernet
latency min. payload	12.2 μ s	24 μ s
latency max. payload	265.2 μ s	372 μ s
jitter bounds	6.4 μ s	< 10 μ s

4. Conclusion & Outlook

Real-time Ethernet is an emerging technology for in-vehicle communication. We have shown by calculations of characteristic FlexRay and TTEthernet scenarios that FlexRay real-time traffic can be embedded in real-time Ethernet. Jitter and latency are comparable (table 2). It is

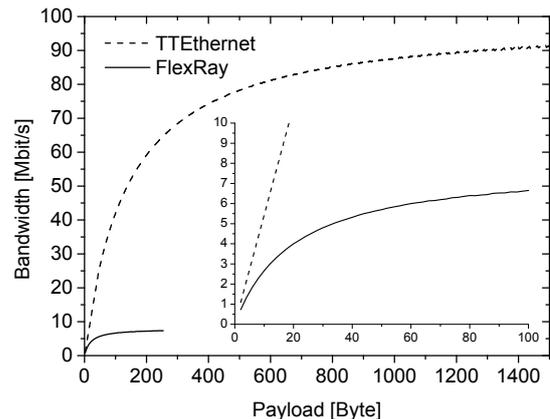


Figure 3. FlexRay and TTEthernet net bandwidth at payload size (16ms cycle)

thus possible to transfer a FlexRay based in-vehicle network to an Ethernet network.

This work defines a basis for further analysis. The calculations given can be adapted for other time-triggered Ethernet technologies. Currently, we simulate TTEthernet in-vehicle networks based on concrete automobile topologies and application data and build a mockup of an in-vehicle network, based on TTEthernet for measurement and load analysis. Future work will analyse how event-triggered traffic, segmentation and priority functionalities of Ethernet can guarantee a smooth integration of time-triggered Ethernet systems into current in-vehicle communication infrastructures.

References

- [1] FlexRay consortium, “Protocol Specification”, Specification 2.1, Stuttgart, Germany, Dec 2005.
- [2] International Organization for Standardization, “Road vehicles - Controller Area Network (CAN)”, ISO 11898, ISO, Geneva, Switzerland, 2003.
- [3] W. Steiner, “TTEthernet Specification”, TTTech Computertechnik AG, Vienna, Austria, Nov 2008.
- [4] Society of Automotive Engineers, “SAE International”, Warrendale, Pennsylvania, USA.
- [5] MOST Cooperation, “Protocol Specification”, Specification 3.0 E1, Karlsruhe, Germany, Oct 2009.
- [6] International Organization for Standardization, “Road vehicles - Controller Area Network (CAN) - Part 4: Time-triggered communication”, ISO 11898-4:2004, ISO, Geneva, Switzerland, 2004.
- [7] PROFIBUS & PROFINET International, “Profinet”, Karlsruhe, Germany.
- [8] SynqNet Interest Group, “SynqNet”, Santa Barbara, CA, USA.
- [9] J. Kiszka and B. Wagner, “RTnet - a flexible hard real-time networking framework”, *10th IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2005)*, vol. 1, pp. 456–463, 2005.
- [10] Aeronautical Radio Incorporated, “Aircraft Data Network”, Standard 664, ARINC, Annapolis, MD, USA, 2002.
- [11] M. Rausch, *FlexRay: Grundlagen, Funktionsweise, Anwendung*, Carl Hanser Verlag, München, 2008.