Latency Analysis for the Cooperation of Event and Time-Triggered Networks

Sebastian Zug, Michael Schulze and Jörg Kaiser
Otto-von-Guericke University
Faculty of Computer Science
Department of Distributed Systems (IVS)
Universitätsplatz 2
D-39106 Magdeburg
{zug, mschulze, kaiser}@ivs.cs.uni-magdeburg.de

Abstract

The paper describes the analysis of cross network latencies occurring when federating an asynchronous, event-triggered CAN-Bus and a synchronous time-triggered TTP/C network. The connection of networks is motivated by both cost-efficiency trade-offs in various network types and reasons of isolating highly critical from less critical communication. The intrinsic latencies incurred when crossing network boundaries are evaluated under various conditions - either in some form of synchronisation with a global time or in completely unsynchronised networks. The results from the formal analysis are compared to experimental results obtained in a mobile robot where the critical reactive control components are connected by a TTP/C network while additional sensor information improving the quality of control is obtained via a CAN-Bus.

1 Introduction

Embedded systems often face problems caused by communicating across multiple, heterogeneous networks. One of the reasons is the cost-efficiency trade-off. Usually, complex embedded systems in the automotive field and industrial automation (e.g. cars, autonomous vehicles and robots) include multiple heterogeneous networks with specific requirements concerning bandwidth, delay, timeliness or dependability. Subsystems are grouped and connected according to these individual demands, and messages are exchanged via a gateway. The use of separate systems ensures local quality properties at adequate cost. On the other side, however, it complicates the combination and exchange of information because the messages may have to cross network boundaries. This is particularly difficult when different communication paradigms are used as a synchronous time-triggered and an asynchronous event-triggered network. Two well-known representatives for these opponents in the automotive area are the Time-Triggered Protocol (e.g. TTP/C) [1] and the Event-Triggered Protocol (e.g. CAN-Bus) [2]. It has been recognized, however, that the combination of both worlds would be beneficial, and protocols combining both communication paradigms in one system have been developed. TTCAN [3] and FlexRay [4] combine time-triggered and event-triggered communication in a single network. TTCAN, on the one hand, cannot ensure the high safety properties as the time-triggered part is a convention rather than enforced by independent hardware measures. It also abandons all positive CAN features concerning error detection and fault handling and no longer supportssporadic and aperiodic traffic very well. FlexRay, on the other hand, provides guardians but sacrifices many elegant fault-tolerance properties of TTP/C [5]. Another way of integration, introduced in [6], is to emulate an event-triggered network on top of a time-triggered one. In this approach, legacy CAN applications can run without modifications. However, even though such emulation is possible and certainly beneficial in specific application areas, the cost of adequate network support is high. Smart sensors and actuators will not have the resources and capacity to participate in such a network. This is the reason why the authors introduce physical gateways. These gateways allow the connection of low-cost sensor-actuator networks to the backbone network of the integrated architecture. Therefore, we investigate the case where networks are federated and connected via gateways while maintaining their specific properties and strength.

In our application scenario, the synchronous Time-Triggered Protocol (TTP/C) is used to realize the distributed safety critical motor control and collision avoidance of a mobile robot (see Fig. 1).

To adjust the trajectory of the robot continuously, the independent motor units not only receive obsta-

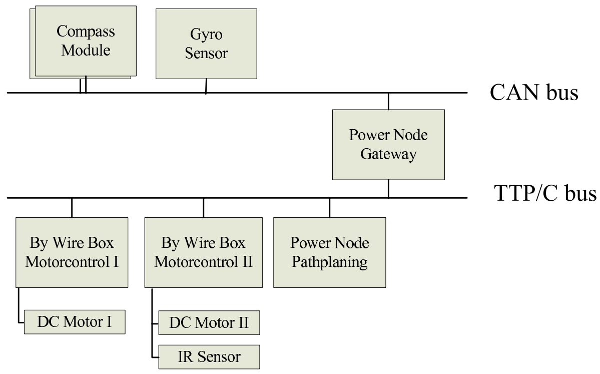

Figure 1. Robot Structure

cle warning and speed information of the other unit via TTP/C but also angular speed and absolute direction from a gyro and a compass module attached to a CAN-Bus (and connected via a gateway). This gateway translates message formats and performs address transformation. As an additional benefit, the gateway can control the information flow between the connected network domains and provide services like information filtering, scoping and - to a certain extent - fault isolation. In our robot application, the transferred sensor information is not safety critical and does not need real-time guarantees. It improves the quality of control but does not affect safety issues. Nevertheless, it is important to estimate the inherent latencies which may occur when transferring messages from the asynchronous, event-triggered CAN-Bus (or a similar event-triggered network) to the synchronous time-triggered network and vice versa. This is particularly important if all events (e.g. of the gyro) have to be transferred and any loss of messages would cause an error. Hence, the analysis of latencies is important to support the proper adjustment of periods and the assignment of message slots on the TTP network. The simplest way, of course, would be to synchronize message transfers on the CAN-Bus and the TTP. This, however, would require the additional overhead of introducing a scheme like real-time event channels [7] and a global time base across networks boundaries.

In this paper, we assume a raw CAN-Bus connected to a TTP/C network via a gateway. This gateway translates message formats and performs address transformation. As an additional benefit, the gateway can control the information flow between the connected network domains and provide services like information filtering and scoping and - to a certain extent - fault isolation. The networks are not synchronized and no global time is available. We analytically derive the latencies of the message transfer across the gateway assuming that the highest priority CAN message has to be transmitted under different load conditions. In this case, the latencies introduced by the gateway and the phasing with the schedule of TTP/C can be evaluated. The analytical results are then verified by an experimental assessment in the robot application.

2 Latency Analysis

2.1 General Description of the Delay

In this section, we describe the latencies when connecting different networks via a gateway. The basic structure is depicted in figure 2. The gateway supports two functionalities: the $ET2TT$ task transforms ET messages and inserts them into the respective TT communication slot $TTcom_1$. The way back is controlled by $TT2ET$. If a message reaches the gateway via $TTCom_2$, it will be transmitted on the ET network.

For the subsequent discussion, the two opposite directions for message transfers are called $ET \rightarrow TT$ and $TT \rightarrow ET$. If both are combined in a round trip time $ET \rightarrow TT \rightarrow ET$, we denote it as $ET \circlearrowleft TT$.

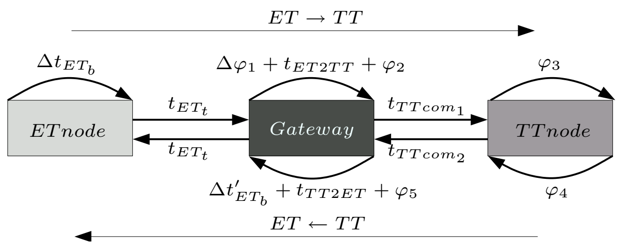

Figure 2. Gateway latencies

The entire duration of the message propagation from the ET network to the TT domain is determined by several parameters associated to each step in the figure.

Δt_{ET_b} - (ET bus blocked) Starting at the left hand side of figure 2, a message transfer may be delayed by an ongoing transmission. If an ET message has to be transmitted (with the highest priority), a loaded network can delay it by an ET bus transmission time at most. If a immediate bus access is possible, Δt_{ET_b} will be "0".

t_{ET_t} - (ET bus transmission time) The transfer of a message in the ET network depends on the transmission rate and the length of the message.

$\Delta\varphi_1$ - (phase displacement) In the gateway, the message has to wait for the appropriate execution slot of the conversion task, $ET2TT$. The message is now synchronized with the TDMA schedule.

t_{ET2TT} The $ET2TT$ task reads the ET message, transforms it into the TT message format and propagates it to the TT domain.

t_{TTcom_1} - (TT communication slot 1) The message is delayed by φ_2, waiting for the propagation task $TTcom_1$. The next step in the schedule is the target task for the message which is reached after the next latency of φ_3.

The time needed for a message propagation from an ET node to a TT node can therefore be represented by the following equations:

$$\begin{aligned} t_{ET \to TT} = & \Delta t_{ET_b} + t_{ET_t} + \\ & \Delta\varphi_1 + t_{ET2TT} + \varphi_2 + \\ & t_{TTcom_1} + \varphi_3 \end{aligned} \tag{1}$$

For the opposite direction in figure 2, the message transmission from TT to ET is similarly determined by the following parameters:

φ_4 Starting in a TT application, the message has to wait for the transmission task in $TTcom_2$.

t_{TT2ET} After φ_5 , the $TT2ET$ conversion task transfers the messages to the ET network.

$\Delta t'_{ET_b}$ The propagation into the ET network could be delayed if the ET bus is loaded. The plain ET transmission requires t_{ET_t}.

Hence, the end-to-end delay from a TT node to an ET node is modelled by:

$$\begin{aligned} t_{TT \to ET} = & \varphi_4 + t_{TTcom_2} + \\ & \varphi_5 + t_{TT2ET} + \Delta t'_{ET_b} + \\ & t_{ET_t} \end{aligned} \tag{2}$$

We assume now that a special TT task returns all incoming, transformed ET messages inside a single TT period. This scenario does not represent a usual application but a possibility to illustrate the resulting delays of synchronised and unsynchronised networks comparably. Therefore a round trip time has to combine $ET \to TT$ and $TT \to ET$. Dynamic terms (marked by a Δ symbol) change their values depending on the actual state of the TDMA schedule and ET bus load. The static values are derived from network configurations and from the schedule of the TT domain. For a more comprehensible notation, the static parts are separated. Hence, the static terms in equations 1 and 2 can be combined to

$$\begin{aligned} t_{ET \circlearrowleft TT_{static}} = & t_{ET2TT} + \varphi_2 + t_{TTcom_1} + \\ & \varphi_3 + t_{TTapp} + \varphi_4 + t_{TTcom_2} + \\ & \varphi_5 + t_{TT2ET} + 2 \cdot t_{ET_t} \end{aligned} \tag{3}$$

The entire duration of a round trip can be calculated by adding the dynamic terms of equations 1 and 2 to 3

$$t_{ET \circlearrowleft TT} = \Delta t_{ET_b} + \Delta\varphi_1 + \Delta t'_{ET_b} + t_{ET \circlearrowleft TT_{static}} \tag{4}$$

This equation describes the general representation of latencies caused by the gateway. In the following section, we discuss the consequences for special constraints.

2.2 Delay distribution function

The interpretation of equation 4 depends on two scenario parameters:

synchronised/unsynchronised networks If a global time exists throughout the system, it is possible to synchronise the ET message transmission in a way that the message reaches the gateway just before the respective conversion task starts. The next section shows that in this case the delay behaviour of the federated network can be determined exactly for the highest priority message.

ET bus without load / ET bus with load Additional ET bus load may delay the message transmission for the duration of one ET transmission. Without additional load, an immediately access can be assumed.

For different combinations of these parameters, two kinds of results for $t_{ET \circlearrowleft TT}$ are possible. The first one is a concrete latency value. Statistical distributed delays described by a probability density function $p(t)$ between t_{min} and t_{max} are more complicated to handle. The paragraphs below derive equations for the different constellations based on 4.

Case 1: Synchronised Networks - No Load The following examination of synchronised networks is based on two conditions:

- Both domains have access to a global time. Hence, it is possible to define a global schedule for the TT and ET domain. TT scheme is imposed onto the ET network [7].
- The most urgent message has the highest priority in the ET network and is sent without delay based on the global schedule.

In this case, the phase displacement becomes zero and there is no delay caused by ET network access. Equation 4 can be simplified to

$$t_{ET \circlearrowleft TT} = t_{ET \circlearrowleft TT_{static}} + 2 \cdot t_{ET_t} \tag{5}$$

The delay of a round trip can be exactly calculated.

Case 2: Synchronised Networks - With Load The conditions are now changed:

- Both domains have access to a global time.

- The respective message has the highest priority in the ET network and can only be delayed by an ongoing transmission.

To ensure that the correct TT time slot is met, the ET message has to send with a constant phase displacement of $\Delta\varphi_1 = 2 \cdot t_{ET_t}$. Accordingly, the equation 4 is simplified to

$$\begin{aligned} t_{ET\circlearrowright TT} =& t_{ET_t} + t_{ET\circlearrowright TT_{static}} + \\ & \Delta t'_{ET_b} + 2 \cdot t_{ET_t} \end{aligned} \tag{6}$$

Only the time for the ET bus arbitration on the way back introduces an uncertain dynamic element of $\Delta t'_{ET_b}$. The $t_{ET\circlearrowright TT_{min}}$ implies an immediate message transmission with $\Delta t'_{ET_b} = 0$ whereas $t_{ET\circlearrowright TT_{max}}$ considers the longest delay - in this case $\Delta t'_{ET_b} = t_{ET_t}$. Hence, $t_{ET\circlearrowright TT_{max}}$ represents the worst case.

$$\begin{aligned} t_{ET\circlearrowright TT_{min}} =& t_{ET\circlearrowright TT_{static}} + 3 \cdot t_{ET_t} \\ t_{ET\circlearrowright TT_{max}} =& t_{ET\circlearrowright TT_{static}} + 4 \cdot t_{ET_t} \end{aligned} \tag{7}$$

The delay may take all values between these two points in a uniform distribution, which can be defined by

$$p_{synch}(t) = \begin{cases} \frac{1}{T} & , t_{ET\circlearrowright TT_{min}} <= t < t_{ET\circlearrowright TT_{max}} \\ 0 & , else \end{cases} \tag{8}$$

with $T = t_{ET\circlearrowright TT_{max}} - t_{ET\circlearrowright TT_{min}}$. The representation of the delay by a probabilistic function includes the worst case. If the information is non-critical, the ratio between delay and success rate can be determined. If an application does not need each response to its requests, it will be possible to calculate the minimum waiting time to obtain a defined probabilistic percentage of all messages.

Case 3: Unsynchronised Networks Without any time synchronisation, the ET message can reach the gateway at arbitrary points in time, e.g. one moment after the conversion task slot has passed. In this case the delay caused by the phase shift φ_1 is equivalent to an entire $t_{TT_{period}}$. If the message is received just before the $ET2TT$ slot, it defines $\varphi_1 = 0$. Of course, φ_1 can take all intermediate values, hence a uniform probability density function describing the delay for this step can be derived by

$$p_{\varphi_1}(t) = \begin{cases} \frac{1}{t_{TT_{period}}} & , 0 <= t < t_{TT_{period}} \\ 0 & , else \end{cases} \tag{9}$$

Similarly, the ET bus arbitration produces a uniformly distributed delay between $0 \leq \Delta t_{ET_b} \leq= t_{ET_t}$ as shown in equation 7. This means

$$p_{\Delta t_{ET_b}}(t) = \begin{cases} \frac{1}{t_{ET_t}} & , 0 <= t < t_{ET_t} \\ 0 & , else \end{cases} \tag{10}$$

For both directions in the round trip, the ET bus has to be accessed. This means $p_{\Delta t_{ET_b}}(t)$ has to be considered twice. To obtain a single equation, the three distributions have to be combined and then added to the equation in 3. The resulting probability distribution of the sum of two or more independent random variables is the convolution of their individual distributions. In our case, this can be expressed by:

$$p_{unsync}(t) = p_{\Delta t_{ET_b}} * p_{\Delta t_{ET_b}} * p_{\varphi_1}$$

For three independent uniform distributions, a function depicted in figure 3 can be derived for the condition $2 \cdot t_{ET_t} < t_{TT_{period}}$.

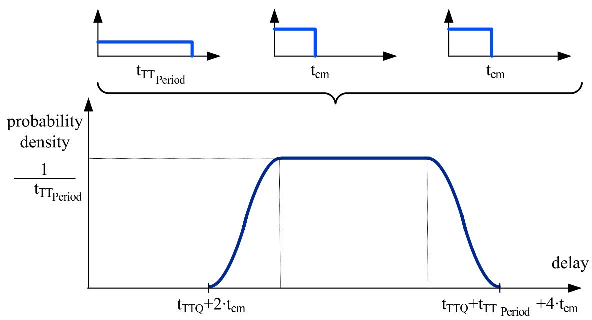

Figure 3. Probability Density Function for unsynchronised Networks

The figure shows that the time for a round trip is between $t_{ET\circlearrowright TT_{static}} + 2 \cdot t_{ET_t}$ and $t_{ET\circlearrowright TT_{static}} + t_{TT_{period}} + 2 \cdot t_{ET_t}$. Unlike the case 2 - equation 8, it is no longer a uniform distribution but exhibits varying probabilities. According to the general solution of this problem described in [8], we derived a probability distribution $p_{unsync}(t)$ for our approach function with a constant factor k

$$k = \frac{1}{t_{ET_t}^2 \cdot t_{TT_{period}}}$$

by equation 11.

Case 4: Other Configurations With an extension of the equation 4, it is also possible to describe the delays in federated networks for messages with arbitrary priority. This either requires a probabilistic model of message arrival time and priority distributions or an analysis scheme like Rate Monotonic Scheduling [9].

3 Measurements

To validate our analysis, we implemented an experimental setup embedded into the robot scenario depicted in figure 1. For the TT communication approach, we used a TTTech development system [10] consisting of 4 PowerNode PN312 components (MPC555 processor,1 MByte RAM, 4 MByte

$$p_{unsync}(t) = k \cdot \begin{cases} \frac{1}{2}t^2 & , 0 \le t < t_{ET_t} \\ -\frac{1}{2}t^2 + 2t_{ET_t}t + t_{ET_t}^2 & , t_{ET_t} \le t < 2t_{ET_t} \\ t_{ET_t}^2 & , 2t_{ET_t} \le t < t_{TT_{period}} \\ t_{ET_t}^2 - \frac{1}{2}\left(t - t_{TT_{period}}\right)^2 & , t_{TT_{period}} \le t < t_{TT_{period}} + t_{ET_t} \\ 2t_{ET_t}\left(t_{ET_t} + t_{TT_{period}} - t\right) + \frac{1}{2}\left(t - t_{TT_{period}}\right)^2 & , t_{TT_{period}} + t_{ET_t} \le t < t_{TT_{period}} + 2t_{ET_t} \\ 0 & , else \end{cases} \quad (11)$$

Flash). The ET network is a CAN bus with two AT90CAN128 AVR Atmel processors attached. The first node works as message source and as destination for the round trip measurement. For each round trip, the time between transmission and reception of the message is recorded. The crystal-controlled timer functions of the CAN nodes show a resolution of $1/16MHz$. The second node produces a defined load on CAN bus.

Figure 4 illustrates the TTP schedule and its temporal order. The TTP/C round has been adjusted to $5000\mu s$. This is, of course, an arbitrary value and the round is much longer than needed to allocate the two message slots and to execute the three TT tasks. With the selected arrangement, all possible phase shifts were integrated and allow the validation of a generalized exemplary schedule.

Column 2 of figure 4 illustrates one, periodically repeated TTP/C cycle $0 - 5000\mu s$ with known communication slots and tasks. For the concrete CAN and TTP network, the gateway tasks $ET2TT$ and $TT2ET$ become $CAN2TTP$ and $TTP2CAN$. The two tasks are executed on the gateway node and one - TTP_{App} - on an application node. Two dedicated slots are provided for communication. The time scale shows the duration of each slot in column 3. Column 1 illustrates the CAN domain, representing the measurement application and the bus traffic. The bold line depicts the way of a message through the networks.

The execution time of all TT tasks and communication slots can be calculated using:

$$\begin{aligned} t_{tasks} &= t_{CAN2TTP} + t_{TTP2CAN} + t_{TTP_{App}} \\ t_{slots} &= t_{TTcom_1} + t_{TTcom_2} \\ t_{tasks} + t_{slots} &= 1350\mu s \end{aligned}$$

The CAN message is based on an extended CAN frame with 8 bytes payload. Hence, the message has a constant length of $147Bit$ including stuff bits and has a transmission time of $t_{ET_t} = 588\mu s$ in the CAN network. The other parameters are listed in table 1.

Measurements on an unloaded bus This experiment should result in the latencies that are only attributed to the delays in the gateway and the arbitrary phase between the CAN and TT networks.

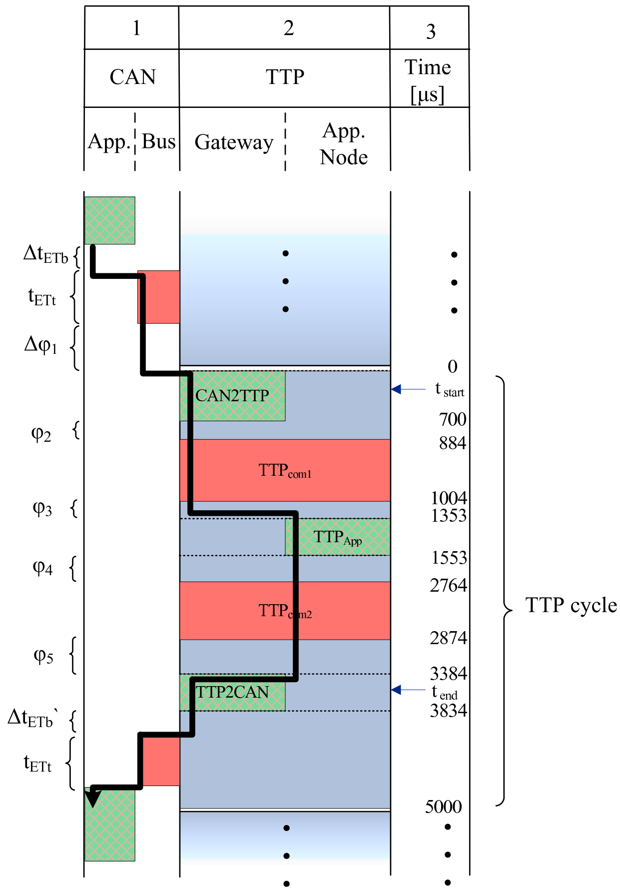

Figure 4. Measurement TTP/C tasks and slots

For the first measurement scenario without CAN bus traffic, the latency for a round trip is caused by mapping to the cyclic time slot in the TT domain only. In this case $\Delta t_{ET_b} = 0$ and $\Delta t'_{ET_b} = 0$ for equation 4. The CAN message is delayed until the start of the conversion task $CAN2TTP$ by the TDMA schedule as depicted in figure 4. Caused by the internal CAN buffer management, the message is read at $t_{start} = 174\mu s$. Hence, the deadline for acceptance in the current TT period is shifted to this value. If the CAN message is received just before the start of reading the CAN buffers, the latency (depicted as $\Delta\varphi_1$ in figures 2 and 4) will be quite small. However,

Denotation	Symbol	Value
TTP/C period	$t_{TTP_{Period}}$	$5000\mu s$
CAN2TTP task	$t_{CAN2TTP}$	$700\mu s$
TTP2CAN task	$t_{TTP2CAN}$	$450\mu s$
TT-Application task	$t_{TTP_{App}}$	$200\mu s$
CAN bit rate		$250kBit/s$
CAN message length		$147Bit$
CAN transmission time	t_{ET_t}	$588\mu s$

Table 1. Parameters of the setup

if the message arrives shortly after t_{start}, it cannot be read and thus message processing is delayed for one TDMA period $\Delta\varphi_1 = t_{TTP_{Period}}$. Therefore, the jitter is defined by the average half of a period.

In the opposite direction, the $TTP2CAN$ task transmits the message to the CAN network at $t_{end} = 3695\mu s$. This is a task specific value and was determined by measurements in advance.

Hence, the theoretically minimal round trip time $t_{min_{calc}}$ can be calculated according to equation 4 .

$$t_{min_{calc}} = t_{TTP_{Period}} + 2 \cdot t_{ET_t} \tag{12}$$

Based on figure 4 $t_{min_{calc}}$ can be calculated as follows

$$\begin{aligned} t_{min_{calc}} &= t_{end} - t_{start} + 2 \cdot t_{ET_t} \\ &= (3695 - 174 + 2 \cdot 588)\,\mu s \\ &= 4697\mu s \end{aligned} \tag{13}$$

For the round trip the maximum delay $t_{max_{calc}}$ is determined by $\Delta\varphi_1 = t_{TTP_{Period}}$ (by):

$$\begin{aligned} t_{max_{calc}} &= \Delta t_{TTP_{Period}} + t_{min_{calc}} \\ &= (5000 + 4697)\,\mu s \\ &= 9697\mu s \end{aligned} \tag{14}$$

For this scenario and based on equation 8, we expect a uniform distribution between $t_{min_{calc}}$ and $t_{max_{calc}}$ with $0 \le \Delta\varphi_1 \le t_{TTP_{Period}}$.

A measurement series for validation, that used the schedule described above, produces the results listed in table 2.

Denotation	Symbol	Value
Number of measurements		10.000
Minimal delay	$t_{min,meas}$	$4699\mu s$
Maximal delay	$t_{max_{meas}}$	$9702\mu s$
Difference		$5003\mu s$

Table 2. Measurement results without bus load

The calculated minimal delay $t_{min_{calc}}$ varies from the measurements illustrated in table 2 by $\Delta t_{min} = 2\mu s$.

The measured maximum delay exceeds the theoretical maximum by $\Delta t_{max} = 5\mu s$. These deviations are caused by inaccuracies of the time measurement in the AVR processor. To check this assumption, the gateway was substituted by another CAN node and we observed a delay of $4\mu s$.

Measurements with background traffic In a more general scenario, a second CAN node continuously transmits low priority CAN messages. The CAN message of the round trip wins the arbitration phase due to a higher priority.

The theoretically minimal round trip time is the same as calculated by equation 13 to $t_{min_{calc}} = 4697\mu s$. According to equation 4 and figure 3, the theoretical maximum delay $t_{max_{calc}}$ is determined by:

$$\begin{aligned} t_{max_{calc}} &= \Delta t_{TTP_{Period}} + t_{min_{calc}} + 2 \cdot t_{ET_t} \\ &= (5000 + 4697 + 2 \cdot 588)\,\mu s \\ &= 10873\mu s \end{aligned}$$

The measurement results can be summarised in the following table:

Denotation	Symbol	Value
Number of measurements		10.000
Minimal delay	$t_{min_{meas}}$	$4832\mu s$
Maximal delay	$t_{max_{meas}}$	$10861\mu s$
Difference		$6029\mu s$

Table 3. Measurements with bus load

The measured value is $(4832-4697)\mu s$ higher than the calculation result. This can be explained with probability property of the delay. Obviously, the 10,000 measurements do not include the theoretical minimal delay. A higher number of repeating (let converge) allows convergence of (the) $t_{min_{calc}}$ to $t_{min_{meas}}$ and $t_{max_{calc}}$ to $t_{max_{meas}}$.

The distribution of the measured 10,000 latencies is illustrated by a histogram in figure 5 . The calculation based on equation 11 is added with the bold line.

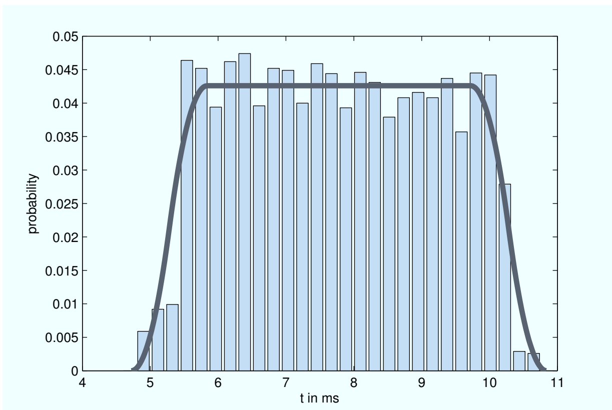

Figure 5. Latency distribution

The measurements show that the analysis models the real system behaviour adequately.

4 Conclusion and Future Remarks

The content of this paper is part of our research on middleware for embedded networked systems. The goal is a seamless and uncomplicated integration of components and networks into a decentralized control system. Apart from the problems related to masking the different addressing schemes, the temporal aspect of interaction across multiple networks is the focus of our interest. The analysis presented in this paper evaluates intrinsic latencies which are incurred when combining a time-triggered and an event-triggered network. TTP/C and the CAN-Bus can be regarded as adequaterepresentatives at both ends of the spectrum reaching from strictly synchronous to completely asynchronous networks. Latencies occur due to network-specific delays like arbitration and busy media as well as latencies in the connecting gateway. The contribution of this paper is firstly a formal probabilistic analysis of these latencies under various network conditions. Secondly, the results are confirmed by an experimental evaluation. The outcome of the paper can be exploited in two directions. Firstly, it provides worst-case latencies for synchronized and non-synchronized cases which can be used to adjust the static schedule in the TT domain. Secondly, it allows the determination of the coverage-latency trade-off. Therefore, it is possible to figure out which latency can be expected for a certain percentage of messages. As an example, an application accepts sensor readings within a certain time window. The respective probability of the temporal requirements being fulfilled can directly be derived from the analysis (cf. figure 5).

In our future work, we will investigate the COSMIC (Cooperating SMart devICes [11]) middleware in an application scenario of sensor/actuator networks. COSMIC provides event channels with different synchrony properties on the CAN-Bus, i.e. hard, soft and non-real-time channels. Additionally, COSMIC solves the problem of network transparency by offering a uniform naming and binding scheme thus masking the different addressing schemes and supporting the routing of messages. The analysis of this paper will be used to allocate appropriate time slots and deadlines to make real-time behaviour across the networks possible.

References

[1] H. Kopetz and G. Bauer, "The Time-Triggered Architecture", *Proceedings of the IEEE*, vol. 91, no. 1, pp. 112 – 126, Jan. 2003.

[2] Robert Bosch GmbH, *CAN Specification Version 2.0*, 1991.

[3] T. Führer, B. Müller, W. Dieterle, F. Hartwich, R. Hugel, and M. Walther, "Time Triggered Communication on CAN (Time Triggered CAN-TTCAN)", *7th international CAN Conference*, 2000.

[4] FlexRay Consortium, *FlexRay Communications System Protocol Specification Version 2.1 Revision A*, 2005.

[5] H. Kopetz, *Real-Time Systems Design Principles for Distributed Embedded Applications*, Kluwer Academic Publishers, Boston / Dordrecht / London, 1997.

[6] R. Obermaisser, "CAN Emulation in a Time-Triggered Environment", in *Proceedings of the 2002 IEEE International Symposium on Industrial Electronics (ISIE)*, volume 1, Jan. 2002. IEEE.

[7] J. Kaiser, C. Brudna, and C. Mitidieri, "COSMIC: A real-time event-based middleware for the CAN-bus", *Journal of Systems and Software*, vol. 77, no. 1, pp. 27–36, July 2005, Special issue: Parallel and distributed real-time systems.

[8] F. Killmann and E. von Collani, "A Note on the Convolution of the Uniform and Related Distributions and Their Use in Quality Control", *Economic Quality Control*, vol. 16, no. 1, pp. 17–41, 2001.

[9] K. Tindell and A. Burns, "Guaranteeing Message Latencies On Control Network (CAN)", *1st international CAN-Conference*, 1994.

[10] TTTech Computertechnik AG, "TTP-Powernode – The TTP Development Board", Product description, September 2005.

[11] J. Kaiser, C. Brudna, C. Mitidieri, and C. Pereira, "COSMIC: A middleware for event-based interaction on CAN", in *9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2003)*, September 2003, Lisbon, Portugal.