Impact of Time-Triggered Transmission Window Placement on Rate-Constrained Traffic in TTEthernet Networks

Florian Heilmann and Gerhard Fohler
Chair of Real-Time Systems
Technische Universität Kaiserslautern, Germany
{heilmann,fohler}@eit.uni-kl.de

ABSTRACT

Safety-critical Ethernet-based networks are receiving significant attention in avionics, automotive and industrial domains. Time-Triggered Ethernet (TTEthernet AS6802) provides safety-critical transmission guarantees via a high priority, time-triggered (TT) traffic class and a lower priority, rate-constrained (RC) traffic class. TT traffic is transmitted between synchronized nodes of a TTEthernet network in offline scheduled TT transmission windows. In this work, we analyze the impact of different placement strategies for these TT transmission windows on end-to-end delay and jitter of RC messages on the same path segment. We show that, depending on the placement of TT transmission windows in a schedule, the end-to-end delay and jitter of RC messages can vary significantly. We further introduce link-based offsets, a new placement strategy for TT transmission windows which allows to reduce the impact of TT transmission windows on RC traffic. In this strategy offsets are applied to all TT transmission windows in a physical link schedule to reduce the amount of time that an RC message on the same physical link is delayed by TT traffic. The link-based offsets strategy can be implemented in the TTEthernet scheduler and does not require hardware modifications. We show that the link-based offset strategy can reduce the end-to-end delay and jitter of RC traffic, and evaluate our claims using an *OMNET++* simulation.

Keywords

Time-Triggered Networks; Scheduling; Time-Triggered traffic; Rate-Constrained traffic

1. INTRODUCTION

Switched Ethernet is at the center of attention as the base technology for next generation avionics, automotive and industrial interconnects due to its high bandwidth and maturity. However, regular switched Ethernet is not suitable for safety-critical applications, as it provides neither delivery guarantees nor delay bounds. To address these issues, extensions to the switched Ethernet protocol have been proposed to enable its use in safety-critical environments.

One such extension is the Time-Triggered Ethernet (TTEthernet, AS6802 [7]). TTEthernet defines a global synchronization scheme and a means to transmit safety-critical messages during time-triggered (TT) transmission windows. These transmission windows are scheduled offline, provide tight guarantees for end-to-end delay and reduce the transmission jitter of TT messages. In addition to this offline scheduled TT traffic TTEthernet implements an unscheduled, rate-constrained (RC) traffic class used for safety-critical, event-triggered messages. Consequently, a TTEthernet network can accommodate both time-triggered and event-triggered safety-critical messages simultaneously.

In a TTEthernet network RC traffic has a lower priority than TT traffic. Thus, a node in a TTEthernet network may preempt or block RC messages in favor of scheduled TT transmissions. Individual RC messages may experience varying amounts of preemptions and blocking delay because RC traffic is not strictly periodic. This can result in a high worst-case end-to-end delay and high worst-case transmission jitter of RC traffic, depending on the TT traffic and its associated TT transmission windows in the network.

In this work, we analyze the impact of TT transmission window placement on RC traffic. We show that the placement of TT transmission windows in a physical link schedule can impact the end-to-end delay and jitter of RC messages which traverse the same physical links. We further introduce our own TT transmission window placement strategy: link-based offsets. Link-based offsets are applied to all TT transmission windows in a physical link schedule. The resulting schedule reduces the amount of preemptions and blocking of RC messages due to TT transmission windows, which, in turn, reduces RC end-to-end delay and jitter. The link-based offset strategy can be applied offline during the schedule generation of the network and does not require hardware modifications.

The contribution of this work is threefold. First, we analyze TTEthernet schedules at the window level, and discuss the impact of two placement strategies for TT transmission windows on RC messages competing for the same physical links. Second, we introduce link-based offsets, a TT transmission window placement strategy which arranges TT transmission windows in such a way, that their impact on RC message delay and jitter is reduced. Third, we evaluate the presented transmission window placement strategies by providing simulation results. These results show that the link-based offsets strategy is superior to the other presented strategies in terms of RC end-to-end delay and jitter.

The remainder of this work is structured as follows: Section 2 discusses related work, while Section 3 introduces the networked system model. In Section 4 we compare different integration strategies for TT and RC traffic, as well as different TT transmission window implementations. Section 5 describes two strategies to place TT transmission windows and details their issues. In Section 6 we introduce the link-based offsets strategy and present simulation results comparing the two previously described strategies and the link-based offsets strategy in Section 7. Finally, Section 8 concludes this work.

2. RELATED WORK

The state of the art contains several works that investigate the co-existence of TT traffic and RC traffic on the same network. Suethanuwong [10] details how the fragmentation of RC messages in a TTEthernet network can improve throughput of RC traffic, however, this work does not take end-to-end delay and jitter into account. Our previous work [6] presents a method to integrate a higher number of TT transmission windows into a schedule to facilitate mode changes. However, while this work focuses on the TT transmission windows, it does not consider RC traffic in the analysis.

Tămaş-Selicean [11], Zhao [12] and Zhou [13] investigate the interaction between TT and RC traffic and the resulting impact on timing analysis. These works, however, focus on the analysis of the co-existence of TT and RC traffic and provide no recommendations or methods to schedule the two traffic classes. Boyer [4] performs analysis on the impact of varying TT loads on RC traffic, but doesn't discuss approaches to minimize the impact of TT traffic on RC traffic. Abuteir [3] introduces a backtracking heuristic to schedule TT and RC messages on the same network, however, this heuristic also doesn't operate at the window level. The strategies presented in this paper could serve as a guidance to the algorithm in [3] to find schedules faster.

Steiner [8] aims to improve the delay and jitter of RC traffic by inserting blank intervals into the TT schedules. While this approach shares the focus and goal of this work, it does not consider how TT transmission windows are placed on multiple, subsequent links, but instead focuses on one link at a time.

3. SYSTEM MODEL

For our system model we partially adopt the notation described in [5] and model the network as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{L})$ where nodes are represented by the vertices $\mathcal{V}$ and physical links between the nodes are represented by the edges $\mathcal{L} \subseteq (\mathcal{V} \times \mathcal{V})$. A physical link $l \in \mathcal{L}$ is defined by an ordered pair (v_a, v_b) which represents the connection from $v_a \in \mathcal{V}$ to $v_b \in \mathcal{V}$. Because switched Ethernet uses bi-directional links it also holds that $\forall (v_a, v_b) \in \mathcal{L} \Rightarrow (v_b, v_a) \in \mathcal{L}$.

Messages are transmitted through the network using virtual transmission channels called virtual links. A virtual link vl can either be a Time-Triggered (TT) virtual link or a Rate-Constrained (RC) virtual link. TT and RC virtual links share the following parameters:

$$\langle src_{vl}, dst_{vl}, period_{vl}, dl_{vl}, lmax_{vl}, path_{vl} \rangle$$

where $src_{vl} \in \mathcal{V}$ is the source node of the virtual link and $dst_{vl} \subseteq \mathcal{V} \setminus src_{vl}$ is the set of destinations of the virtual link. $period_{vl}$ and dl_{vl} are the period and deadline of the virtual link, respectively. For TT virtual links, $period_{vl}$ describes the fixed time span between two scheduled TT transmission reservations. In case of RC virtual links, $period_{vl}$ is the *Bandwidth Allocation Gap (BAG)* which defines the minimum inter-arrival time (assuming no jitter) between two consecutive messages on the same virtual link. This means that, for every BAG, at most one message is transmitted on an RC virtual link. $lmax_{vl}$ denotes the maximum payload size of a message on the respective virtual link, and $path_{vl}$ describes the path of the virtual link through a sequence of links l from the virtual link source node src_{vl} to its destination node(s) dst_{vl}. TT virtual links additionally define $\forall l_i \in path_{vl}$ a TT transmission window $w_{vl,i} = (s_{vl,i}, e_{vl,i})$ for TT messages on the given physical link where $s_{vl,i}$ and $e_{vl,i}$ denote the start and end times of the TT transmission window for vl on l_i, respectively. This window acts as a reservation for pending TT messages of the virtual link. The length of a window $\text{len}(w)$ is the difference between the start and end times, i.e. $\text{len}(w_{vl,i}) = e_{vl,i} - s_{vl,i}$. In this work, we assume that all TT transmission windows have an equal length. As RC virtual links can compete for a physical link due to their unscheduled nature, an additional maximum jitter parameter $j_{\max}$ is used to limit this contention delay of the RC virtual links on any physical link.

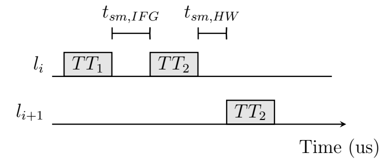


Figure 1: Safety Margins in TTEthernet schedules

TTEthernet supports both multicast and unicast virtual links. For the sake of simplicity, in this work, we model multicast virtual links as a set of unicast virtual links — one per destination. This simplification allows us to represent $path_{vl}$ as an ordered sequence of links $l_i \in \mathcal{L}$, i.e. $[l_0, l_1, \dots, l_n]$. We also assume that $dl_{vl} = period_{vl}$. The transmission time of a virtual link message on a link l_i is computed by dividing the message length by the link bandwidth, i.e. $\tau_{msg} = \frac{len(msg)}{bw(l_i)}$. Analogously, the time to transmit one bit on the network is the inverse of the link bandwidth: $\tau_{bit} = bw(l_i)^{-1}$.

In switched Ethernet, two consecutive messages on a physical link l_i are separated by an *Inter-Frame-Gap* τ_{IFG} which is $96 \times \tau_{bit}$ long. Additionally, we denote the time it takes the hardware of a node to internally forward a message from the input port to the output port as τ_{HW}. TTEthernet schedules need to account for τ_{IFG} and τ_{HW} by inserting appropriate safety-margins between any two messages on the same physical link ($\tau_{sm,IFG}$) and messages of the same virtual link on two consecutive physical links ($\tau_{sm,HW}$), respectively. See Figure 1 for a visual illustration of $\tau_{sm,IFG}$ and $\tau_{sm,HW}$ on two consecutive links l_i and l_{i+1} with two TT virtual links TT_1 and TT_2.

We define the network end-to-end delay of a message as the time difference from the point in time at which the producer task on the sending node releases the message to the network to the point in time where the network hands over the message to the consumer task on the destination node. For TT traffic, upper and lower bounds of the end-to-end delay depend on the placement of the TT transmission windows $w_{vl,l}$ and the integration strategy (see Section 4) and can be easily computed. For RC traffic, previous works such as [11], [12] and [13] provide methods to compute worst-case delays of RC traffic in presence of TT traffic using frameworks like network calculus. We further define the jitter of a TT or RC virtual link as the difference between the

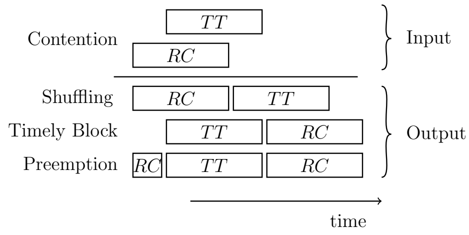


Figure 2: Integration Strategies for TTEthernet [9]

worst-case delay and the best-case delay of messages on the respective virtual link. Note that this jitter is not the same as the $j_{\max}$ parameter of RC virtual links.

TTEthernet defines two additional message classes: Best-Effort (BE) messages, which represent non-safety-critical traffic with the lowest priority, and Process Control Frames (PCF) which are used to establish synchronization of the network and have the highest priority.

4. BACKGROUND

In this section, we discuss the required concepts for the placement strategies presented in Section 5 and the link-based offsets strategy described in Section 6. Section 4.1 describes integration strategies for TT and RC traffic in a TTEthernet network, while Section 4.2 details TT transmission window implementations. Our analysis focuses on the interaction between TT and RC traffic, hence, we do not consider PCF and BE traffic in this work.

4.1 Integrating TT and RC traffic

In a TTEthernet network, TT and RC messages may contend for a common physical link l of a node on their respective virtual link paths $path_{vl}$ to their destinations. To integrate TT and RC messages, several integration strategies are available: *Shuffling* and *Preemption* are defined by the AS6802 standard [7], while [9] also presents *Timely Block* and *Resume Preemption*. Although not part of AS6802, we include timely block in this work, as it is the integration strategy used in the *CoRE4INET* model suite [1] we use in *OMNET++* [2] to simulate TTEthernet. Timely block is also implemented in existing TTEthernet hardware [9].

Shuffling: Shuffling allows RC messages to be transmitted, even if their transmission overlaps with the transmission window of a TT message. The benefits of shuffling are higher bandwidth utilization and lower RC end-to-end delay, but at the same time, the end-to-end delay and jitter of TT messages may be increased, as they are delayed until the RC messages finish transmission.

Timely Block: Nodes implementing timely block will delay the transmission of an RC message if its transmission would overlap with the transmission window of a TT message. While this improves the jitter and end-to-end delay of TT messages, RC messages may experience a larger end-to-end delay and higher jitter. Furthermore, the physical link may remain idle even though RC messages are pending, which reduces throughput and wastes bandwidth.

(Resume) Preemption: Under the preemption strategy, an RC message is preempted if, during its transmission, the TT transmission window of a pending TT message begins. After the TT message finishes transmission, the RC message is resumed or retransmitted under resume preemption and preemption strategies, respectively. Preemption reduces end-to-end delay and transmission jitter of TT messages, but increases the end-to-end delay and transmission jitter of RC messages similar to timely block.

Figure 2 shows the output sequence for the shuffling, timely block and preemption strategies when an RC and a TT message are in contention for a physical link. Shuffling delays the transmission of the TT message, while timely block and preemption delay and preempt the RC message, respectively, and re-transmit it after the TT message has finished transmission.

4.2 TT Window Implementations

This section presents two transmission window implementations as they are used in TTEthernet Hardware [9] and the *OMNET++* [2] simulator with the *CoRE4INET* [1] model suite. The transmission of a TT message of a virtual link vl_k on a link l_i is defined by the TT transmission window formed by the window start time $s_{k,i}$ and the window end time $e_{k,i}$.

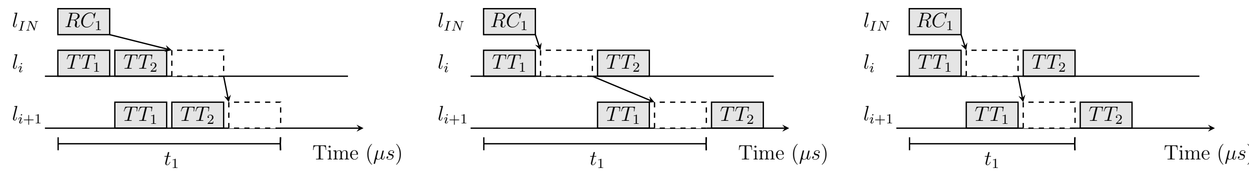
First Bit Strict (FBS): "First Bit Strict" TT transmission windows require the first bit of the TT message of a virtual link vl_k to be transmitted on link l_i no sooner than the transmission window start time $s_{k,i}$ and no later than the window end time $e_{k,i}$. This transmission window implementation is suitable for the shuffling integration strategy and requires a transmission window size that is large enough to accommodate a maximum sized RC message and the first bit of the TT message. During run-time, an RC message arriving just before the start time of a TT transmission window can be transmitted fully, without compromising the validity of the scheduled TT message transmission.

Last Bit Strict (LBS): "Last Bit Strict" TT transmission windows require the last bit of a TT message of a virtual link vl_k to be transmitted on link l_i no sooner than the transmission window start time $s_{k,i}$ and no later than the transmission window end time $e_{k,i}$. This transmission window implementation is suitable for the timely block and preemption integration strategies. For example, in *OMNET++*, the node initiates the TT message transmission at the transmission window start time $s_{k,i}$ and an appropriate transmission window size $\text{len}(w)$ ensures that the last bit will arrive before the transmission window end time $e_{k,i}$.

The benefits and drawbacks of these two transmission window implementations are similar to the integration strategies they can be used with. While FBS transmission windows allow for more flexibility and better bandwidth utilization by avoiding idle links or message preemptions, LBS transmission windows result in lower transmission jitter for TT messages since are not delayed by contending RC messages.

5. WINDOW PLACEMENT STRATEGIES

Although related work analyzes how TT schedules can affect delay and jitter of RC messages, none of these works do so at the TT transmission window level on multiple consecutive links. In this section, we discuss two placement strategies to show that suboptimal placement of TT transmission



(a) Pipelined ASAP (PA) (b) Aligned Distributed (AD) (c) Link-Based Offsets (LBO)

Figure 3: Window-Level placement strategies

windows on consecutive links, henceforth called "path segment", can negatively impact the end-to-end delay and jitter of RC messages which share the same path segment. Figure 3 illustrates the placement strategies. An RC message of the virtual link RC_1 arrives on l_{IN} and shares the path segment $[l_i, l_{i+1}]$ with the TT virtual links TT_1 and TT_2.

Pipelined ASAP (PA) A naïve strategy to place the TT transmission windows of multiple TT virtual links on the same path segment is illustrated in Figure 3a. TT transmission windows for the virtual links TT_1 and TT_2 are placed on l_i as early as possible. Furthermore, transmission windows of a TT virtual link are placed back to back on l_i and l_{i+1} to reduce the queuing times of TT_1 and TT_2. This is problematic if RC messages of virtual link RC_1, which share the path segment $[l_i, l_{i+1}]$ with the TT virtual links, arrive on l_{IN} during the early part of the schedule where the TT transmission windows are placed. Due to the lack of gaps between the TT transmission windows, these RC messages can be delayed for a long time before they can traverse l_i. RC messages may also arrive during a later time where no TT transmission windows are placed (after TT_2 in Figure 3a). These RC messages can traverse the path segment unhindered which results in a very low transmission delay. The combination of these long and short transmission delays results in a large jitter for the RC messages.

Note that this only happens when the timely block or preemption strategies are used. With shuffling, RC_1 would be transmitted after TT_1 because TT_2 is "shuffled" to a later point. As a result, TT_2 will also be transmitted after RC_1 on l_{i+1} so RC_1 will not be delayed on this link.

Aligned Distributed (AD) In order to reduce the continuous blocking of RC messages of RC_1, even under timely block and preemption strategies, the TT transmission windows can be distributed evenly over the physical link schedule. Steiner, in [8], describes this as "increasing the porosity" of the TT schedule. This allows RC messages to be transmitted during the vacant intervals in the schedule. However, if the TT transmission windows are aligned as shown in Figure 3b, AD can still result in a large amount of delay for messages of the virtual link RC_1. Here, messages of RC_1 arriving on l_{IN} suffer less delay on l_i than in the PA strategy, but are delayed by a TT transmission window again on l_{i+1}. In the worst case, RC_1 messages are delayed by messages of TT_1 or TT_2 on l_i. In the best case, the RC messages arrive during a vacant interval on l_i and are only delayed on l_{i+1}. While this results in a lower jitter for messages of RC_1 than the PA strategy, the best-case delay is much worse.

Again, this problem is less apparent under the shuffling strategy, where RC_1, depending on the exact arrival sequence, may begin transmission earlier and cause messages of TT_1 or TT_2 to be shuffled to a later point and, thus, will not suffer as much delay.

6. LINK-BASED OFFSETS (LBO)

To remedy the problems outlined with schedules using PA and AD placement strategies we propose a new strategy named "link-based offsets" (LBO). As a starting point for LBO, a schedule according to the AD placement strategy is used. LBO then determines an integer value o_l for each physical link $l \in \mathcal{L}$, such that, for two consecutive physical links $l_i, l_{i+1} \in \mathcal{L}$, the following equation is satisfied:

$$\forall i : |o_i - o_{i+1}| \mod 2 = 1 \tag{1}$$

Equation 1 requires that the difference between offsets on two consecutive links is odd. Note that this also implies that the offsets on two consecutive links cannot have the same value.

After the offsets o_l have been determined, they are applied to their respective physical links by shifting all TT transmission windows in the link schedule by $o_l \times \text{len}(w)$. The resulting schedule is a hybrid between the PA and AD schedules, with sufficient "porosity" [8] to avoid long blocking intervals for RC messages, while also avoiding the repeated delaying of RC messages on consecutive links as seen in AD schedules.

Figure 3c presents an LBO schedule. A message of RC_1, arriving on l_{IN} is, at worst, delayed by TT_1 or TT_2 once on l_i. After the RC_1 message has been transmitted on l_i it can continue its transmission on link l_{i+1} since l_{i+1} is either already idle at this point or will soon become idle after the transmission of TT_1 or TT_2 is completed. The resulting worst-case delay for the RC message on the shared path segment is much lower compared to PA and AD schedules. The worst-case happens if the message of RC_1 is delayed by messages of either TT_1 or TT_2 on l_i. In the best case, the RC message arrives during a vacant interval of l_i. The jitter of the messages on RC_1 is thus similar to the AD strategy and much better than the PA strategy.

When an AD schedule with identical lengths for all TT transmission windows is used as a base schedule, a single integer offset per link o_l achieves the desired effect. Other base schedules with e.g. non-identical window lengths may require more complex algorithms, non-integer offsets and/or multiple offsets applied to the TT transmission windows of separate TT virtual links. We defer this investigation to future work.

Link-based offsets do not require modifications to existing TTEthernet hardware, as the computation and application of the offsets o_l can be implemented in the offline TTEthernet scheduler and the resulting schedules do not break the TTEthernet specification [7].

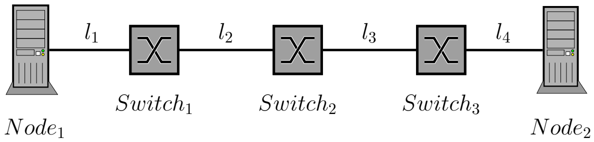


Figure 4: Simulated Network Topology

7. EVALUATION

To verify the benefits of LBO, we perform a TTEthernet network simulation using the *OMNET++* [2] simulator with the *CoRE4INET* [1] model suite.

7.1 Simulation Parameters

The topology of our network is depicted in Figure 4. It consists of two endpoint nodes and three switches connected via four 100 Mbit/s links (l_1 to l_4). $Node_1$ transmits messages on five TT virtual links and one RC virtual link. All six virtual links share a common destination, $Node_2$, and a common path $[l_1, l_2, l_3, l_4]$.

The period of the TT messages and BAG of the RC message are set to $1ms$ to simplify the manual creation of the schedules. The transmission window length of all transmission windows in the schedules is set to $100\mu s$. *OMNET++* uses LBS transmission windows and timely block, so we aim to completely utilize the TT transmission windows by using the largest possible TT payload length at which no transmission window violations occur. As a result, all TT messages have a payload length of 1218 Bytes. The resulting Ethernet frame on the physical layer is 1244 Bytes long[1]. Two transmission windows on the same physical link are separated by an *Inter-Frame-Gap* safety-margin $\tau_{sm,IFG}$ of $1\mu s$ to account for Ethernet inter-frame gaps ($0.96\mu s$ for 100 Mbit/s). Similarly, transmission windows of the same virtual link on two consecutive physical links are separated by a *Hardware* safety-margin $\tau_{sm,HW}$ of $100ns$ to account for the time the message needs to propagate from the input port of a node to the output port. The payload length of RC messages is set to 1000 Bytes to allow RC messages to fit into the vacant intervals of the schedules without requiring a precise time of arrival. To obtain data for different release time phases of RC messages, the inter-arrival time of two consecutive RC messages is distributed uniformly between $1ms$ and $2ms$. This results in RC messages being released at all possible phases in the schedule, which allows to examine the impact of schedules which do not distribute the TT transmission windows evenly.

We extend the RC traffic source and traffic sink implementations to store timestamps immediately before sending and immediately after receiving the message to obtain the end-to-end delay. Each simulation is run for 60 simulated seconds, resulting 40000 data points for each simulation run.

7.2 Schedules

We investigate three TT schedules with different window placement strategies:

[1]In theory, a $100\mu s$ transmission window should be able to accommodate frames with a length of 1250 Bytes, however, we encountered transmission errors where frames arrived too late for frame lengths above 1244. We suspect that the simulator cannot simulate ideal, zero-length hardware delays which requires lowering the payload length to avoid invalid message arrival times.

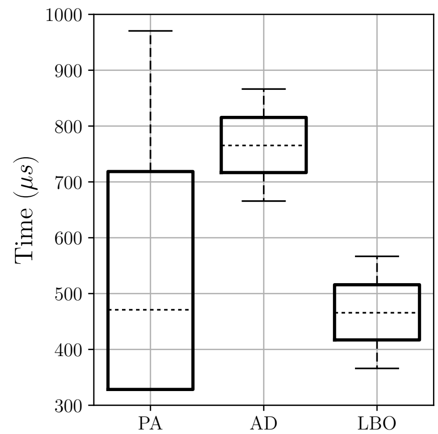


Figure 5: End-to-end delay of RC frames

Pipelined ASAP Under the PA strategy all TT messages are scheduled as soon as possible. As a result, a continuous interval of $5 \times (\text{len}(w) + \tau_{sm,IFG})$ is reserved for TT traffic on each link schedule. All transmission windows are shifted by $\text{len}(w) + \tau_{sm,HW}$ on each consecutive link. This results in minimal transmission time for all TT messages, but results in a long interval on each physical link where no RC message can be transmitted due to the timely block integration strategy of *OMNET++*.

Aligned Distributed The AD strategy distributes transmission windows evenly across the schedule for a given link. As a result, a gap of one TT transmission window length $\text{len}(w)$ is inserted between all transmission windows on the same link. The start and end times of the TT transmission windows are aligned on all physical links, i.e. each link contains a TT transmission window starting at 0 μs, $200\mu s$ and so on. The safety-margins $\tau_{sm,IFG}$ and $\tau_{sm,HW}$ are not necessary due to the artificial gaps between TT transmission windows, both on the same physical link as well as on two consecutive physical links.

Link-Based Offsets The link-based offsets schedule implements our strategy detailed in Section 6. The TT transmission windows are distributed across each physical link schedule similar to the AD placement strategy. However, offsets o_l are applied to the physical links such that Equation 1 described in Section 6 holds. Specifically, with the AD schedule as starting point, an offset of $1 \times \text{len}(w)$ is applied to l_2 and l_4. After applying the offsets, hardware safety-margins $\tau_{sm,HW}$ are inserted where necessary.

7.3 Results

The resulting end-to-end delays of the three simulation runs are depicted in Figure 5. The three box-plots show the distribution of end-to-end delays for the RC messages under the three transmission window placement strategies.

7.3.1 Worst case delay

Figure 5 shows that PA results in the highest worst case delay, while AD results in slightly lower, and LBO in the lowest worst case delay. For PA, RC messages, which arrive during the time where TT transmission windows are placed in the schedule, are delayed by up to five consecutive window lengths. Under AD, while the continuous blocking is avoided through increased porosity, the alignment of the TT transmission windows causes large delays of RC messages while they traverse the path from source to destination. RC messages in the LBO schedule only experience minimal delay, i.e. when a RC message finishes transmission on a link, it merely has to wait for the currently active TT transmission window to end and can then begin transmission on the following physical link immediately. The theoretical worst-case delays of the three strategies in our topology are as follows:

$$\begin{aligned} wc(PA) =& 2 \times \tau_{RC} + 5 \times (\text{len}(w) + \tau_{sm,IFG}) + \\ & 3 \times (\text{len}(w) + \tau_{sm,HW}) + \tau_{sm,IFG} = 972.3\mu s \\ wc(AD) =& 2 \times (\tau_{RC} + \tau_{sm,IFG}) + 7 \times \text{len}(w) = 866\mu s \\ wc(LBO) =& 2 \times (\tau_{RC} + \tau_{sm,IFG}) + 4 \times \text{len}(w) + \\ & 3 \times \tau_{sm,HW} = 566.3\mu s \end{aligned}$$

7.3.2 Best case delay

The best case delays for messages under the PA and LBO strategies are about 200 μs lower than their worst case. This is due to the fact that, in the worst case an RC message arrives on l_1 just too late to be transmitted resulting in additional blocking time on l_1. In the best case, the RC message is not delayed on l_1 at all because it arrives during a time where l_1 will be idle long enough for the RC message to be transmitted. Under PA, the best case also occurs if the RC message arrives when l_1 will be idle long enough for the RC message to be transmitted. The RC message will then be able to pass through its path with zero additional delay, resulting in the lowest best case transmission time of all three strategies. The theoretical best case delays of the three strategies in our topology are as follows:

$$\begin{aligned} bc(PA) =& 4 \times (\tau_{RC} + \tau_{sm,HW}) = 328.4\mu s \\ bc(AD) =& 2 \times (\tau_{RC} + \tau_{sm,IFG}) + 5 \times \text{len}(w) = 666\mu s \\ bc(LBO) =& 2 \times (\tau_{RC} + \tau_{sm,IFG}) + 2 \times \text{len}(w) + \\ & 3 \times \tau_{sm,HW} = 366.3\mu s \end{aligned}$$

7.3.3 Summary

In summary, the simulation corroborates the fact that the link-based offsets strategy can reduce the impact of TT transmission windows on end-to-end delay and jitter of RC messages. The LBO schedule results in jitter values similar to the AD placement strategy, whilst providing a lower best-, worst- and average case delay. LBO also produces best and average case delays similar to the PA strategy, whilst having significantly less jitter and a much lower worst case delay. Furthermore, the results obtained from the simulation closely match the theoretical best and worst case delays of RC messages under the evaluated schedules.

8. CONCLUSION AND FUTURE WORK

In this work, we analyzed the impact of TT transmission window placement in consecutive physical link schedules on end-to-end delay and jitter of RC messages. We presented two placement strategies and identified their issues. In order to improve delay and jitter of RC traffic, the placement of TT transmission windows should be considered whilst scheduling TTEthernet networks. To reduce the impact of TT transmission windows on RC messages, we proposed the link-based offsets placement strategy. Link-based offsets can reduce both the end-to-end delay and the jitter of RC messages, can be implemented in TTEthernet schedulers and require no modifications to existing hardware. Finally, we evaluated the performance of link-based offsets against the other placement strategies through simulation, which confirmed the improvements for delay and jitter of RC messages when LBO is used.

The use case in the simulation was intentionally kept simple for this initial presentation. In future work, we will investigate the benefits of link-based offsets in industrial scale use cases with larger topologies. Our evaluation also considered a single RC virtual link, whereas in realistic scenarios, multiple RC virtual links will compete for the same (partial) network path. Finally, this work assumed simplifications w.r.t. the parameters of the TT and RC virtual links, such as equal message and TT transmission window sizes, periods and virtual link paths. In future work, we want to relax these limitations, to provide a more generic analysis of the link-based offsets strategy. Finally, we want to investigate the impact of link-based offsets on TT messages as well.

9. REFERENCES

[1] CoRE4INET framework for OMNET++. http://core4inet.core-rg.de. (Accessed on 03/30/2017).

[2] OMNeT++ Discrete Event Simulator. https://omnetpp.org/. (Accessed on 04/04/2017).

[3] M. Abuteir and R. Obermaisser. Scheduling of rate-constrained and time-triggered traffic in multi-cluster TTEthernet systems. In *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, July 2015.

[4] M. Boyer, H. Daigmorte, N. Navet, and J. Migge. Performance impact of the interactions between time-triggered and rate-constrained transmissions in TTEthernet. In *8th European Congress on Embedded Real Time Software and Systems* , Jan. 2016.

[5] S. S. Craciunas and R. S. Oliver. SMT-based Task- and Network-level Static Schedule Generation for Time-Triggered Networked Systems. In *22nd International Conference on Real-Time Networks and Systems*, 2014.

[6] F. Heilmann, A. Syed, and G. Fohler. Mode-changes in cots time-triggered network hardware without online reconfiguration. In *14th Workshop on Real-Time Networks (RTN'16) at ECRTS'16*, July 2016.

[7] SAE. Time-Triggered Ethernet, A6802, 2011.

[8] W. Steiner. Synthesis of static communication schedules for mixed-criticality systems. In *2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, March 2011.

[9] W. Steiner, G. Bauer, B. Hall, M. Paulitsch, and S. Varadarajan. TTEthernet dataflow concept. In *Proceedings of the 2009 Eighth IEEE International Symposium on Network Computing and Applications*.

[10] E. Suethanuwong. Message fragmentation of event-triggered traffic in TTEthernet systems using the timely block method. In *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, March 2016.

[11] D. TamasSelicean, P. Pop, and W. Steiner. Timing analysis of rate constrained traffic for the TTEthernet communication protocol. In *2015 IEEE 18th International Symposium on Real-Time Distributed Computing*.

[12] L. Zhao, H. Xiong, Z. Zheng, and Q. Li. Improving worst-case latency analysis for rate-constrained traffic in the time-triggered ethernet network. *IEEE Communications Letters*, pages 1927–1930, Nov 2014.

[13] X. Zhou, F. He, and T. Wang. Using network calculus on worst-case latency analysis for TTEthernet in preemption transmission mode. In *2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec 2016.