Schedulability Analysis of Best-Effort Traffic in TSN Networks

Bahar Houtan*, Mohammad Ashjaei*, Masoud Daneshtalab*, Mikael Sjödin*, Sara Afshar†, Saad Mubeen*
*Mälardalen University, Västerås, Sweden
†Volvo Construction Equipment, Eskilstuna, Sweden

2021 IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA) | 978-1-7281-2989-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ETFA45728.2021.9613511

***Abstract*—This paper presents a schedulability analysis for the Best-Effort (BE) traffic class within Time Sensitive Networking (TSN) networks. The presented analysis considers several features in the TSN standards, including the Credit-Based Shaper (CBS), the Time-Aware Shaper (TAS) and the frame preemption. Although the BE class in TSN is primarily used for the traffic with no strict timing requirements, some industrial applications prefer to utilize this class for the non-hard real-time traffic instead of classes that use the CBS. The reason mainly lies in the fact that the complexity of TSN configuration becomes significantly high when the time-triggered traffic via the TAS and other classes via the CBS are used altogether. We demonstrate the applicability of the presented analysis on a vehicular application use case. We show that a network designer can get information on the schedulability of the BE traffic, based on which the network configuration can be further refined with respect to the application requirements.**

I. Introduction

Today's vehicular embedded systems deal with many challenges, such as scalability in data communication, computational complexity and guaranteeing determinism for hard real-time traffic. These challenges are due to increased functionality of automotive systems, which demands coexistence of diverse applications within the same network. Diversity of applications can be in terms of priority, timing constraints or bandwidth constraints. Therefore in recent years, IEEE Time-Sensitive Networking (TSN) task group has been formed to provide features that can support such diversity in communication and act as a backbone communication-network for some industrial domains; in particular for the vehicular on-board communication-systems [1], [2].

TSN standards are toolboxes where various features depending on the application can be used to improve the performance of the communication in the application. In particular, TSN standards allow temporal isolation of the Scheduled Traffic (ST), which is sent according to a static schedule created offline. The temporal isolation is realized by the Time-Aware Shaper (TAS) mechanism using gates operation on ports of a TSN switch. This guarantees timing determinism for ST. In addition to the TAS mechanism, TSN standards define the Credit-Based Shaper (CBS) mechanism that allows reservation of the bandwidth by reserving links' capacity based on credits associated to each class. The shapers define different traffic classes in TSN networks, being classes A, B and Best-Effort (BE). Classes A and B use the CBS mechanism, while class BE does not use the CBS mechanism. Class A has a higher priority than class B, and class BE has the lowest priority in the network. Hence, BE traffic will only be sent when no higher priority traffic is contending for the link.

One of the main industrial domains, where using TSN is gaining significant momentum is the vehicular domain due to their advancements in functionalities, smart devices, and high-bandwidth sensors. However, redesigning and replacing the existing communication systems is not reasonable, mainly because of extra costs imposed by the new design. Therefore, the vehicular applications consider the TSN network as a backbone network supporting high-bandwidth between several Electronic Control Units (ECU). One of the challenging and non-trivial tasks is to support the legacy traffic and map them into the TSN traffic classes. Due to the complexity of applying a combination of TSN features, there is a high tendency in vehicular industry to use only the ST class for hard real-time strictly periodic traffic. For example, using a combination of the TAS and CBS mechanisms in practice is not realized easily, due to the complexity of the CBS configuration. Also, it is common to use the BE class for the legacy traffic, that has no strict deadlines but a minimum quality-of-service (QoS) demand. Since, the BE class is simple to use and configure, obtaining a level of QoS via schedulability analysis is useful for a large class of BE traffic.

To this end, none of the existing works support the schedulability analysis of the BE class, when the TAS, CBS and frame preemption are used in the network. The main contribution in this paper is to develop a schedulability analysis that can verify the worst-case response time of each individual BE message in the network when CBS, TAS and frame preemption are used. Furthermore, the paper shows the applicability of the analysis on a use case from the vehicular domain.

II. Background and Related Work

In general, there are four main analysis techniques for TSN networks: (1) RTA, (2) network calculus, (3) use of eligible intervals, and (4) use of machine learning. Fig. 1 shows a timeline of the existing schedulability analysis approaches for TSN. According to [2], the focus of the first works in the existing schedulability analysis has evolved from only supporting CBS to more sophisticated models, which include the combination of CBS with time-triggered traffic and frame preemption support. There are several recent works that provide schedulability analysis of TSN networks considering the frame preemption [3], [4], [5]. Among the above mentioned techniques, this paper focuses on the worst-case RTA. More specifically, we extend the work in [3] to provide schedulability analysis for BE class in TSN networks. The following sections present a summary of the most significant research in the area of schedulability analysis for TSN.

A. *Response Time Analysis (RTA) for TSN*

The work in [6] is one of the first works aiming at the utilization of AVB for in-vehicle communication. The

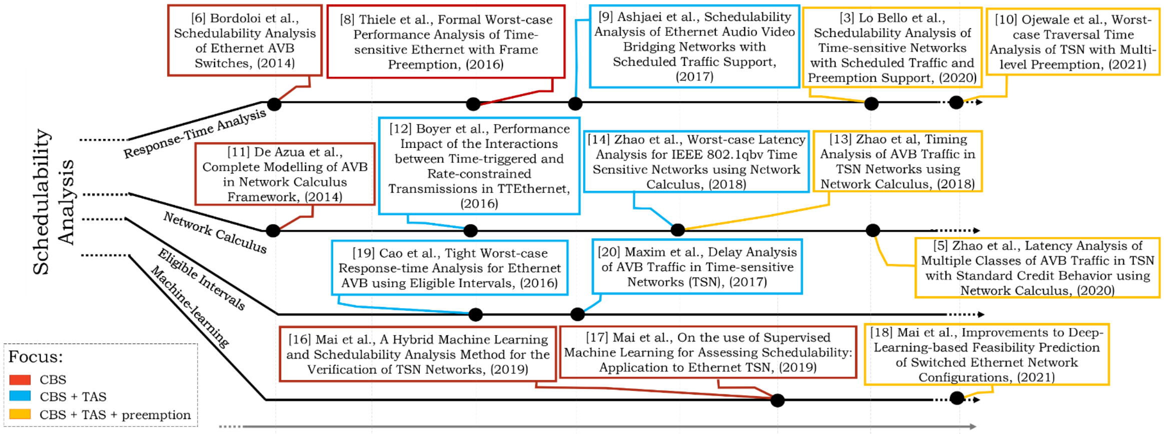
[6] Bordoloi et al., Schedulability Analysis of Ethernet AVB Switches, (2014)
[8] Thiele et al., Formal Worst-case Performance Analysis of Time-sensitive Ethernet with Frame Preemption, (2016)
[9] Ashjaei et al., Schedulability Analysis of Ethernet Audio Video Bridging Networks with Scheduled Traffic Support, (2017)
[3] Lo Bello et al., Schedulability Analysis of Time-sensitive Networks with Scheduled Traffic and Preemption Support, (2020)
[10] Ojewale et al., Worst-case Traversal Time Analysis of TSN with Multi-level Preemption, (2021)
Schedulability Analysis
Response-Time Analysis
[11] De Azua et al., Complete Modelling of AVB in Network Calculus Framework, (2014)
[12] Boyer et al., Performance Impact of the Interactions between Time-triggered and Rate-constrained Transmissions in TTEthernet, (2016)
[14] Zhao et al., Worst-case Latency Analysis for IEEE 802.1qbv Time Sensitive Networks using Network Calculus, (2018)
[13] Zhao et al, Timing Analysis of AVB Traffic in TSN Networks using Network Calculus, (2018)
Network Calculus
[19] Cao et al., Tight Worst-case Response-time Analysis for Ethernet AVB using Eligible Intervals, (2016)
[20] Maxim et al., Delay Analysis of AVB Traffic in Time-sensitive Networks (TSN), (2017)
[5] Zhao et al., Latency Analysis of Multiple Classes of AVB Traffic in TSN with Standard Credit Behavior using Network Calculus, (2020)
Eligible Intervals
Machine-learning
[16] Mai et al., A Hybrid Machine Learning and Schedulability Analysis Method for the Verification of TSN Networks, (2019)
[17] Mai et al., On the use of Supervised Machine Learning for Assessing Schedulability: Application to Ethernet TSN, (2019)
[18] Mai et al., Improvements to Deep-Learning-based Feasibility Prediction of Switched Ethernet Network Configurations, (2021)
Focus:
🟥 CBS
🟦 CBS + TAS
🟨 CBS + TAS + preemption
2014
2015
2016
2017
2018
2019
2020
2021

Fig. 1: Timeline of schedulability analysis techniques for TSN since 2014.

proposed RTA only supports CBS and the class A traffic is assumed to be the highest-priority traffic. Hence, the work in [6] does not consider any interference by the ST class. The RTA was further developed in several works [7] to support the hard real-time time-triggered traffic in TSN, which needs to be shaped by deterministic traffic shaper mechanisms, such as the TAS or the peristaltic shaper. A comparative evaluation of these two shapers discussed in [7] shows that shaping the hard real-time traffic based on the peristaltic shaper can cause larger blocking times to the message under analysis. Hence, the work in [8] extended the previous work to provide the worst-case RTA for TSN supporting the TAS and preemption. The work in [9] considered the CBS and TAS mechanisms in combination, where the TAS mechanism was a variation of the TSN standard. Furthermore, the work in [3] proposed the worst-case RTA that considers the TAS, CBS and different variations of frame preemption support (i.e., with and without Hold and Release mechanisms according to the standard). Finally, the work in [10] introduced a worst-case traversal analysis for TSN, which is based on a system model that allows multiple preemption levels [4].

B. Analysis for TSN based on Network Calculus

Network calculus is a well-known technique to calculate the worst-case delays in networks. A network calculus analysis for AVB was introduced in [11]. Later, the work in [12] provided an approach to integrate the timing analysis for the periodic time-triggered traffic and the rate-constrained sporadic traffic in TTEthernet. Though TTEthernet has several similarities to TSN, it does not feature the CBS mechanism.

Within the context of TSN, the works in [13] and [14] considered the influence of the scheduled traffic on the AVB traffic in preemption mode. The main focus of this work is to employ network calculus to find tighter worst-case delay bounds for classes A and B. The work was further developed in [5] to support multiple AVB classes in the system model and to include the effects of traffic classes such as ST, A and B. Finally, the work in [15] presents the state of the art in network calculus-based TSN methods.

C. Analysis for TSN based on Machine Learning

An interesting approach for verification of TSN networks and their schedulability analysis using machine learning techniques was followed in the works presented in [16] and [17]. This approach mainly focuses on verifying the feasibility of TSN configurations by combining a schedulability analysis and a machine learning technique. The work in [18] further improved the approach by applying deep learning-based techniques. However, these techniques are not primarily proposed to provide predictability guarantees for TSN networks, yet are interesting for verifying the TSN network configurations.

D. Analysis for TSN based on Eligible Intervals

Besides the aforementioned works, a technique based on the concept of eligible intervals was proposed in the context of AVB network considering solely the CBS mechanism in [19]. Further, the work in [20] used the same concept to analyze the delays of classes A and B in the presence of the ST class.

To sum up, the previous schedulability analysis techniques have addressed several features in TSN, but are still limited to the analysis of classes A and B. This seems intuitive as the BE class was not supposed to be used for traffic with timing requirements. However, in order to avoid the complexity of the CBS traffic, the designers of TSN applications often prefer to assign the legacy traffic with timing requirements to the BE class instead of the CBS traffic (classes A and B). Hence, schedulability analysis of the BE traffic class is needed. Provisioning of the RTA for the BE traffic class in TSN networks is the main focus of this paper.

III. System Model

The system model consists of two parts. First, the network model defines the physical attributes of the network. Second, the traffic model describes the message attributes that are communicated between the source and destination end stations.

A. Network Model

In the network model, we assume TSN switches that support CBS, TAS, and clock synchronization. The network topology is modelled through a set of links shown by $\mathcal{L}$ that contains all network links. A typical link in the network is indicated by l, where $l \in \mathcal{L}$. Links are two-directional connections linking an end station to a switch and one switch to another switch. R specifies the network bandwidth per link, i.e., the maximum allowed load on the links. We assume that all the links have the same bandwidth. Moreover, the parameters $\alpha^{+}_{z,l}$ and $\alpha^{-}_{z,l}$ are the idleSlope and sendSlope values associated with the link l, which are only applicable when traffic from credit-based class z uses the link. The idleSlope is the rate of increasing the credit for a class of traffic when there is a pending message for transmission in the class. The sendSlope is the rate of credit consumption when the message is being transmitted. The sum of idleSlope and sendSlope is equal to R. Finally, the delay due to hardware design of the switch is assumed to be bounded by γ.

B. Traffic Model

The traffic is modelled by a message, m_i. A message contains a stream of data to be transmitted from the source end station to the destination end station. An instance of a message is called a frame. We assume that traffic preemption is enabled. The traffic class ST is configured as an express class, while other classes are preemptable (not express). Therefore, ST messages can preempt messages from the other classes, but messages from the other classes cannot preempt the lower-priority messages. The fragments of a preempted message obtain an individual header; therefore, preemptions cause extra overhead due to the extra transmission time required for the headers of the fragments. The parameter v is the transmission time of an Ethernet frame header. The attributes of the message are represented by the tuple $\left\langle C_i, T_i, D_i, P_i, \mathcal{O}_i^l, \mathcal{L}_i \right\rangle$. Where C_i is the transmission time of m_i. The transmission time is calculated based on R and the size of the message. A guard band is a reserved slot added before the ST schedule slots in case the preemption by the ST class is enabled to prevent any interference by the lower-priority messages that cannot be preempted. The duration of the guard band is represented by λ. According to the IEEE 802.3br standard, a message fragment less than 123 $Bytes$ cannot be preempted. T_i is the message m_i's period and D_i specifies the end-to-end deadline of m_i. We consider implicit deadlines, i.e., the deadline of each message is assumed to be equal to its period. The message priority is represented by P_i. We assume that the possible values of P_i can be assigned from the set of class IDs in the set $\{ST, A, B, BE\}$, and each TSN class is associated with a queue. Moreover we consider only two AVB classes, but it's worth noting that up to 8 AVB classes can be supported by the shaper mechanism. In case the message is from the ST class, deterministic schedules on each of the links need to be specified at the Gate Control List (GCL) of the TSN switches. Therefore, the parameter $\mathcal{O}_i^l$ represents the set of offsets of the ST message m_i on link l within $\mathcal{L}_i$, where $\mathcal{L}_i$ is the set of links in the route of m_i from the source end station to the destination end station, and $\mathcal{L}_i \in \mathcal{L}$. We assume that the ST schedules (i.e., the offsets for the ST messages) are given using a scheduling technique, e.g., [21]. J_i^l is the jitter on the link l, which is the delay variations (bound) in the transmission of the message m_i on the link l.

IV. Revisiting RTA for TSN without the BE Traffic

The worst-case response time of a message consists of three components: (1) interference from the messages belonging to the higher-priority traffic (hp), (2) interference from the messages belonging to the same priority traffic (sp), and (3) blocking from the messages belonging to the lower-priority traffic (lp). The RTA for BE traffic in TSN, presented in this paper, is built upon the existing RTA [3], which is revisited in this section. The following subsections present the worst-case RTA for the messages belonging to classes A and B.

A. RTA for Class A Messages

In order to explain various factors contributing to the response time of a class A message, the transmission trace of an example is presented in Fig. 2. A message could be routed through several links and switches; hence the worst-case RTA must be performed per link. The trace in Fig. 2 shows a shared link between messages from three different priority classes in TSN, namely classes ST, A, and B. For example, there are four periodic frames of an ST message that are activated at every 5 time units. The transmission time of the ST message is 1 time unit. Class A contains a message A_1 with the period of 20 time units and transmission time of 3 time units. The period and transmission time of the class B message are 20 and 3 time units respectively. Fig. 2 shows the transmission trace until the hyperperiod (20 time units) of all the messages. Moreover, the upward arrows indicate the activation times of the messages in the hyperpeiod.

We are interested in the response time of message A_1, where we need to find a critical instant candidate which can lead to the worst-case scenario. The worst-case scenario for A_1 occurs when the higher and the same priority messages are activated at the same time as the activation of A_1, i.e., at time 0 in Fig. 2. For simplicity of the trace, we excluded the same priority interference, and the credits of classes A and B are not shown. However, we assumed that the credits are 0 at the time of the message's activation. Moreover, the worst-case scenario also considers that there is an ongoing transmission of a lower-priority message (B_1) on the port when the message under analysis is activated. Since the higher priority class with respect to class A is set to express, these express messages are handled by the TAS allowing the express messages to preempt the lower-priority messages. In this case, preemption overhead due to adding an Ethernet frame header to each preempted fragment of the messages needs to be taken into account. This interference on A_1 can be seen at time 5 in Fig. 2.

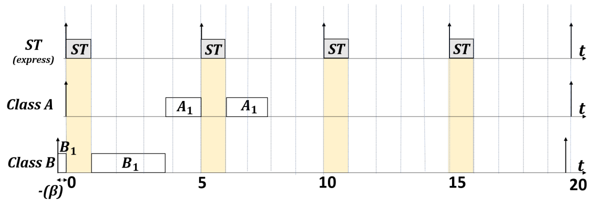

Fig. 2: Trace of a message set: class A message under analysis.

Another component of the worst-case scenario is when the lower-priority message from class B is activated at a slightly earlier time than the message under analysis from class A, at time $-\beta$ in Fig. 2. The message B_1 starts its transmission at $-\beta$ as there are no higher priority messages that are queued for transmission at the same time. Although at time 0 the frame A_1 is activated, it cannot be preempted by B_1 because it is set to preemptable. However, the ST frames can preempt both B_1 and A_1, as can be seen in Fig. 2. After full transmission of B_1, the frame A_1 can be transmitted, although it may still be preempted by the ST frames.

In summary, the worst-case response time for the class A message consists of: (i) interference from the messages belonging to the ST class; (ii) ST preemption overhead; (iii) blocking due to lower-priority messages belonging to the B and BE classes; and (iv) interference from other same priority messages from class A. The following subsections explain each of the elements contributing to the worst-case response time of a class A message.

1) Interference from higher-priority messages: Since the ST class is scheduled offline, an offset per link is assigned to each ST message. In order to consider the offsets of the ST messages in the analysis, the offset-based analysis in [22] is adapted in which a transaction is defined to contain several tasks. The adaptation of the transactional model into the ST messages is as follows. Since in the system model, we assumed one ST class using one queue of each port, then all ST messages form one transaction and its period is the least common multiple (LCM) of all ST periods. Fig. 3 represents an example of two ST messages in a transaction, namely message x and message w, with the periods T_x and T_w, respectively. The transaction period is indicated by T, which is the LCM of T_x and T_w. The parameter k represents an instance of a message within the transaction's period. That is, $k = [1, n]$, where n is equal to the total number of instances of the message within the transaction's period. For example, n for the message w within T is 2 as there are two frames of message w during T. Similarly, n for x within T equals to 1.

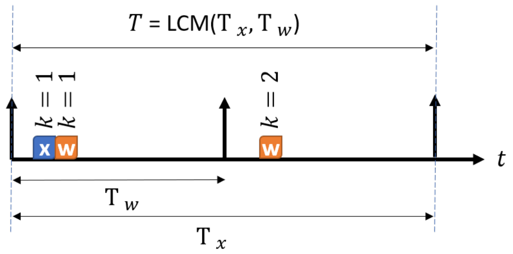

Fig. 3: An example of a transaction.

Since the activation time of the message under analysis from class A is unknown, each activation of an ST frame within the hyperperiod can be a critical-instant candidate for the class A message. The largest response time caused by the ST class interference to the message under analysis can be found by generating critical-instant candidate combinations from the ST messages. Therefore, it is assumed that the message under analysis is activated at time 0. Then, the phase of the ST transaction is shifted such that the activation of one of the ST messages coincides with the activation of the message under analysis at each critical-instant combination. Since multiple frames of a message can appear in the hyperperiod, i.e., multiple frames in the ST transaction (Fig. 3), the phase should be derived for all frames within the transaction. In order to derive the offsets for all frames in the transaction, we use Eq. (1), where I^l_j is an array containing the offsets of all frames of ST message m_j on link l during the transaction period T_{lcm} (hyperperiod of all ST messages).

$$\mathrm{I}^l_j = \{(k-1)T_j + O^l_j - \lambda : k = 1..n, n = \tfrac{T_{lcm}}{T_j}\} \quad (1)$$

The phase between ST message m_j and the k^{th} frame of the critical-instant candidate ST message m_c on link l is denoted by $\Phi^l_{jc[k]}$, that is calculated by Eq. (2).

$$\Phi^l_{jc[k]} = (O^l_j - I^l_c[k]) \; mod \; T_{lcm} \quad (2)$$

O^l_j is the offset of ST message m_j on link l, whereas $I^l_c[k]$ is the offset of k^{th} frame of m_c on l. The worst-case interference by ST messages is calculated by Eq. (3), when candidate ST message m_c coincides with the critical instant [3].

$$\mathrm{W}^l_{c[k]}(t) = \textstyle\sum_{\forall j \in ST \wedge l \in \mathcal{L}_j} \left(\left\lfloor \frac{\Phi^l_{jc[k]}}{T_{lcm}} \right\rfloor + \left\lceil \frac{t-\Phi^l_{jc[k]}}{T_{lcm}} \right\rceil \right) C_j \quad (3)$$

As it is mentioned earlier, in case of any preemption, the fragmented frame will again obtain a frame header. Therefore, the frame becomes larger in size with each preemption. The impact of this preemption overhead is calculated by Eq. (4).

$$\mathrm{V}^l_{c[k]}(t) = \textstyle\sum_{\forall j \in ST \wedge l \in \mathcal{L}_j} \left(\left\lfloor \frac{\Phi^l_{jc[k]}}{T_{lcm}} \right\rfloor + \left\lceil \frac{t-\Phi^l_{jc[k]}}{T_{lcm}} \right\rceil \right) v \quad (4)$$

2) Blocking by lower priority messages: Since the CBS controls the transmission of class A traffic, the blocking of class A message under analysis can occur under two circumstances. Firstly, when the lower-priority messages are activated slightly earlier than the activation of the class A message. Secondly, a lower-priority message takes the port for transmission while the class A message is activated but the credit for class A is negative. It has been shown in [6] that several lower-priority messages can block the class A message. However, if the same priority interference is inflated as in Eq. (6) then it is safe to consider only one blocking message from the lower-priority traffic as shown in Eq. (5).

$$\mathrm{BB}^l_i = \max_{\substack{\forall m_j \in lp(m_i) \\ \wedge\, l \in \mathcal{L}_j}} \{C_j\} \quad (5)$$

3) Interference from same priority messages: To cover the need for accounting the blocking due to the mentioned circumstances, the same priority interference should be inflated by $\left(1 + \frac{\alpha^-_{A,l}}{\alpha^+_{A,l}}\right)$. Therefore, Eq. (6) calculates the same priority interference for class A message under analysis m_i. For more information and the proofs please refer to [6] (and subsequently in [3]).

$$\mathrm{ISA}^l_i = \sum_{\substack{\forall m_j \in sp(m_i), i \neq j \\ \wedge\, l \in \mathcal{L}_j}} \left(1 + \frac{\alpha^-_{A,l}}{\alpha^+_{A,l}}\right) C_j \quad (6)$$

In the above equation, $\alpha^+_{A,l}$ and $\alpha^-_{A,l}$ are the idleSlope and sendSlope values of class A on the link l, respectively.

4) Response-time calculations: The response-time analysis iteratively considers the blocking and interference on the message under analysis (m_i) on the link l in time intervals (t). The iterative process is continued until the values of the computed response times are equal in two consecutive iterations or the value of the response time exceeds the corresponding deadline. Consequently, in a deadline-constrained model, we call a message schedulable if the response time of the message is less than or equal to the message's deadline. The response time of a message in class A is calculated by Eq. (7).

$$\mathrm{RT}^{l,(x)}_{ic[k]} = W^l_{c[k]}(RT^{l,(x-1)}_{ic[k]}) + V^l_{c[k]}(RT^{l,(x-1)}_{ic[k]}) + BB^l_i + ISA^l_i + C_i \quad (7)$$

Where the index of the current and the previous iterations are specified by (x) and $(x-1)$, respectively. The worst-case response time on link l is the maximum value among all calculated response times based on all critical instant candidates' combinations of m_c, as shown in Eq. (8).

$$\mathrm{RT}^l_i = \max_{\forall m_c \& \forall k} \{RT^l_{ic[k]}\} \quad (8)$$

B. RTA for Class B Messages

Class B is non-express similar to class A. Hence, the four elements that influence the worst-case response time of a class B message include: (i) interference from the higher-priority ST class and class A messages; (ii) class ST preemption overhead; (iii) blocking by class BE messages; (iv) interference from other same priority messages from class B. Fig. 4 shows an example with four classes of ST, A, B and BE. In this example, we are interested in one of the critical instant candidates of the class B message, i.e., B_1. Similar to the previous example in Fig. 2, the worst-case scenario considers that a lower-priority message BE_1 has started slightly before the activation of B_1. Since class B is not express, B_1 cannot preempt the transmission of BE_1. When the transmission of BE_1 is completed, a higher-priority class A message, A_1, is activated. Assuming that the credit for class A is zero or positive, A_1 will be transmitted first. Afterwards, B_1 can be transmitted. As shown in Fig. 2, the ST messages can preempt all lower-priority messages, including B_1. An interesting observation in this example is that the higher-priority interference on class B messages can be preemptive (via class ST messages) or non-preemptive (via class A messages) at the same time.

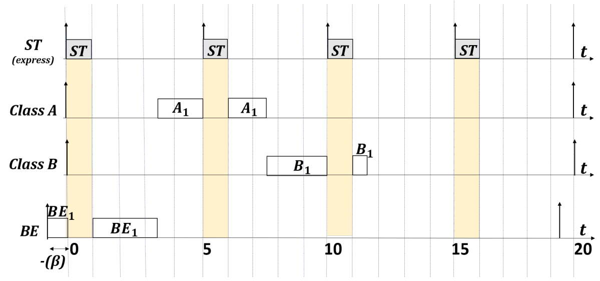

Fig. 4: Trace of a message set: class B message under analysis.

1) Interference from higher-priority messages: Interference from the ST messages as well as the preemption overhead due to the ST messages is calculated using Eq. (3) and Eq. (4), respectively. The class A traffic is not express in contrast to the ST class. Therefore, class A messages cannot preempt any of the lower-priority messages. As a result, the interference of class A follows a non-preemptive high-priority interference model. When the link is already occupied by other messages, the current frame of the message under analysis must be queued until the link is idle. Eq. (9) calculates the high-priority interference of class A messages on the class B message, where the queuing delay is denoted by $\omega^l_{ic[k]}(q)$.

$$\mathrm{IA}^l_i(\omega^l_{ic[k]}(q)) = \sum_{\substack{\forall m_j \in classA, i \neq j \\ \wedge\, l \in \mathcal{L}_j}} \left\lfloor \frac{\omega^l_{ic[k]}(q) + J^l_j}{T_j} + 1 \right\rfloor C_j \quad (9)$$

Above, $\omega^l_{ic[k]}(q)$ is the queuing delay on the link l when the k^{th} frame of the critical instant candidate (m_c) coincides with the critical instant. Moreover, it has been shown in [3] that any frame of a message under analysis (m_i) may lead to the worst-case situation (not necessarily the first frame of the message). Thus, q frames should be evaluated for the worst-case analysis. J^l_j is the jitter of each frame of class A message m_j on link l.

2) Blocking by the lower-priority messages: BE class is the only lower priority traffic than the message under analysis (m_i) from class B. The blocking by this class is presented as BBE^l_i, which is calculated the same as BB^l_i in Eq. (5).

3) Interference from same priority messages: The same priority interference of class B messages on the message under analysis (m_i) is calculated by Eq. (10), which takes into account the inflation factor of class B. Besides, the equation considers that the worst-case scenario can be caused by any of the q instances from the message under analysis (m_i). For more information and the proofs please refer to [6] (and subsequently in [3])

$$\mathrm{ISB}^l_i = \sum_{\substack{\forall m_j \in sp(m_i), i \neq j \\ \wedge l \in \mathcal{L}_j}} \left(1 + \frac{\alpha^-_{B,l}}{\alpha^+_{B,l}}\right) C_j \left\lfloor \frac{(q-1)T_i}{T_j} + 1 \right\rfloor \quad (10)$$

4) Response-time calculations: The response time calculations of class B are performed in two phases. In the first phase, the queuing delay from all sources of interference and blocking that have an influence on the q^{th} frame of the message under analysis (m_i) is iteratively calculated using Eq. (11).

$$\begin{aligned}\omega^l_{ic[k]}(q) = W^l_{c[k]}(\omega^l_{ic[k]}(q)) + V^l_{c[k]}(\omega^l_{ic[k]}(q)) + \\ IA^l_i(\omega^l_{ic[k]}(q)) + BBE^l_i + ISB^l_i + (q-1)C_i\end{aligned} \quad (11)$$

In the second phase, the message m_i can only be interfered by ST messages preemptively and not by other classes as their interference is already accounted in the first phase. Therefore, the response time of m_i when m_c from ST class coincides the critical instant for the q^{th} frame is calculated by Eq. (12). Note that the second phase starts at time $\omega^l_{ic[k]}(q)$, i.e., after the calculation of the busy period.

$$\begin{aligned}RT^{l,(x)}_{ic[k]}(q) = \omega^l_{ic[k]}(q) + W^l_{c[k]}(RT^{l,(x-1)}_{ic[k]}(q)) + \\ V^l_{c[k]}(RT^{l,(x-1)}_{ic[k]}(q)) + C_i - (q-1)T_i\end{aligned} \quad (12)$$

Finally, Eq. (18) calculates the maximum value of response time, which is the maximum value of $RT^l_{ic[k]}$ calculated for the q^{th} frame of the message under analysis (m_i).

$$RT^l_{ic[k]} = \max_{q=1..q_{max}} \{RT^l_{ic[k]}(q)\} \quad (13)$$

V. Proposed RTA for the BE traffic in TSN

This section extends the existing RTA for TSN to support the class BE messages. Class BE does not utilize the express mode. Moreover, the traffic passing from the class BE queue is subjected to interference by the CBS and ST classes. Since the system model does not contain any lower-priority classes than the class BE, unlike other classes, the class BE messages do not experience any blocking delay. Fig. 5 shows an example of a BE frame's transmission trace and the interference by the higher-priority messages. There is a message belonging to each of these classes. There are four periodic frames of an ST message that are activated every 5 time units. The transmission time of the ST message is 1 time unit. Each message in classes A, B and BE has a period of 20 time units and a transmission time of 3 time units.

A critical instant of the class BE message occurs when it is activated at the same time with the higher-priority messages. Firstly, the class ST message that is in the express mode, preempts the class BE message and its higher-priority messages from classes A and B. The message belonging to class A has subsequently higher priority than the messages of classes B and BE. Therefore, the class A message (in case of positive credit) is transmitted as soon as the link is freed by the class ST message. Similarly, the class B message is transmitted after the transmission of class A message is completed. The class BE message is also fragmented due to the preemption caused by the ST frames. Note that the class BE message is interfered by the class ST messages preemptively and by classes A and B messages non-preemptively. The elements that influence the RTA of class BE frames, include: (i) interference from class ST, classes A and B as the higher-priority classes; (ii) ST preemption overhead; (iii) interference from other same priority messages from class BE.

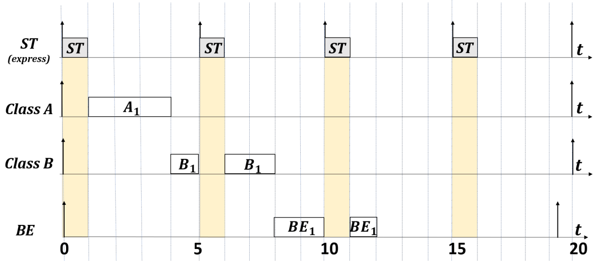

Fig. 5: Trace of a message set: class BE message under analysis.

A. Interference from Higher-priority Messages

The set of higher-priority messages with respect to the class BE message includes messages from classes ST, A and B. The class ST messages interfere with the BE message according to the TAS schedules. Whereas, the classes A and B are shaped based on the corresponding class' credit. Consequently, we can expect the same effect of interference and preemption overhead due to the ST schedules as presented for RTA of class A and class B in Section IV-A1. Eq. (14) calculates the interference by classes A and B based on the queuing delays in a similar way to the existing non-preemptive interference.

$$\mathrm{IAB}_i^l(\omega_{ic[k]}^l(q)) = \sum_{\substack{\forall m_j \in classA \cup classB, i \neq j \\ \wedge\ l \in \mathcal{L}_j}} \left\lfloor \frac{\omega_{ic[k]}^l(q) + J_j^l}{T_j} + 1 \right\rfloor C_j (14)$$

where, J_j^l is the jitter of higher-priority messages from classes A and B that cause the queuing delay to the q^{th} frame of the message under analysis (m_i). The calculation of J_j^l for both classes A and B will be presented at the end of this section.

B. Interference from the Same Priority Messages

In a deadline-constrained model, only one frame of the same priority message can be queued before the queuing of the message under analysis. Eq. (15) calculates the same priority interference for m_i in class BE.

$$\mathrm{ISBE}_i^l = \sum_{\substack{\forall m_j \in sp(m_i), i \neq j \\ \wedge l \in \mathcal{L}_j}} C_j \left\lfloor \frac{(q-1)T_i}{T_j} + 1 \right\rfloor \quad (15)$$

Note that unlike the same priority interference for classes A and B, the same priority interference for class BE is not inflated. The reason is that the inflation factor is needed to cover the case of multiple blocking by lower-priority messages. As in this model, the BE class is the lowest-priority class, the concept of inflation factor becomes irrelevant.

C. Response-time Calculations

The RTA for class BE messages is performed in two phases similar to the analysis for class B (in Section IV-B4). Therefore, in the first phase of the analysis, the queuing delay from all sources of interference that have an influence on the q^{th} frame of the message under analysis (m_i) from class BE is calculated by Eq. (16).

$$\begin{aligned}&\omega_{ic[k]}^l(q) = W_{c[k]}^l(\omega_{ic[k]}^l(q)) + V_{c[k]}^l(\omega_{ic[k]}^l(q)) + \\ &IAB_i^l(\omega_{ic[k]}^l(q)) + ISBE_i^l + (q-1)C_i\end{aligned} \quad (16)$$

where, the queuing delay $\omega_{ic[k]}^l(q)$ includes: (i) preemptions by the ST class $W_{c[k]}^l(\omega_{ic[k]}^l(q))$; (ii) ST preemption overhead $V_{c[k]}^l(\omega_{ic[k]}^l(q))$; (iii) credit-shaped high-priority traffic interference by classes A and B ($IAB_i^l(\omega_{ic[k]}^l(q))$); and (iv) the transmission time of all the q frames of the message under analysis that are waiting in the queue.

In the second phase, the queuing delay of the BE message under analysis which is derived in the previous stage is utilized to compute the response time of the message under analysis (m_i). In this phase, m_i can only be interfered by the ST messages preemptively. The response-time calculations in the second phase are presented by Eq. (12).

$$\begin{aligned}&RT_{ic[k]}^{l,(x)}(q) = \omega_{ic[k]}^l(q) + W_{c[k]}^l(RT_{ic[k]}^{l,(x-1)}(q)) + \\ &V_{c[k]}^l(RT_{ic[k]}^{l,(x-1)}(q)) + C_i - (q-1)T_i\end{aligned} \quad (17)$$

The worst-case response time of m_i from class BE is the maximum $RT_{ic[k]}^l$ of the k^{th} critical instant candidate combination (m_c) as shown in Eq. (18).

$$\mathrm{RT}_{ic[k]}^l = \max_{q=1..q_{max}}\{RT_{ic[k]}^l(q)\} \quad (18)$$

Where q_{max} represents the maximum number of frames of the message under analysis that are queued for transmission during its maximum busy period. The value of q_{max} is derived as the smallest positive integer value that satisfies the inequality (19).

$$\begin{aligned}&W_{c[k]}^l(\omega_{ic[k]}^l(q)) + V_{c[k]}^l(\omega_{ic[k]}^l(q)) + IAB_i^l + qC_i + \\ &\sum_{\substack{\forall m_j \in classA \cup classB, i \neq j \\ \wedge\ l \in \mathcal{L}_j}} \left\lceil \frac{\omega_{ic[k]}^l(q) + J_j^l}{T_j} \right\rceil C_j \leq q.T_i\end{aligned} \quad (19)$$

The worst-case response time of the class BE message (m_i) on link l is equal to the maximum value of the response times with respect to all critical-instant candidates as shown in Eq. (20).

$$RT_i^l = \max_{\forall m_c \& \forall k} \{RT_{ic[k]}^l\} \tag{20}$$

D. Calculations for Queuing Jitter of Classes A and B

The arrival of a message from class A or B on a link that is shared with the class BE message (m_i) can vary due to traversal of the class A and B messages through multiple links in the network. This variation results in the queuing jitter of the A and B messages. This jitter may have a significant effect on the response time of m_i as shown in the calculations for higher-priority interference in Eq. 14 and Eq. 19. To calculate the jitter on link l, we should find the worst-case and best-case delays from the sender end station to the link l where it is shared by m_i. The worst-case RTA of classes A and B messages are already presented in Section IV. The best-case response time of classes A and B can occur by assuming that the corresponding credits are always available for the messages upon their arrival for transmission. Thus their best-case response times will be equal to their corresponding transmission times. Eq. (21) shows the calculations for release jitter of class A and B messages (m_j) on link l.

$$J_j^l = \sum_{L=1..l} RT_j^L - \sum_{L=1..l} BT_j^L$$
$$\sum_{L=1..l} BT_j^L = l.C_j \tag{21}$$

E. Response Time over Multiple Links

As messages across multiple links are buffered in the queues of each switch, the worst-case response time of a message crossing multiple links is the sum of the per-link worst-case response times. Eq. (22) shows the worst-case response time of m_i in class BE crossing multiple links from its source till its destination end station. Note that the switch hardware latency γ is added for each link.

$$\text{RT}_i = \sum_{l=1..|\mathcal{L}_i|} (RT_i^l) + (|\mathcal{L}_i| - 1).\gamma \tag{22}$$

VI. Evaluation

This section presents the evaluation of the presented response time analysis on a vehicular application use case. The main intention of the evaluation is to show that a network designer can evaluate the number of deadline misses in the BE traffic using the proposed analysis. Based on this information, the overall network configuration can be adjusted and refined.

A. Vehicular Application Use Case

The use case is inspired from a most-commonly used industrial case study on a modern car developed in [23]. Based on the presented use case, we define a TSN network that consists of 14 end stations connected through two TSN switches, as shown in Fig. 6. In this use case, we assume that the total network bandwidth is 10Mbps and the fabrication delay in all switches is assumed to be $5\mu s$.

In this use case, there are three cameras: CAM1, CAM2, and CAM3. These cameras are mounted on different sides of the car and send video streams using classes A and B. The receiver of the video streams is the Head Unit, which is the main (central) ECU in the use case. Moreover, FCAM is the main front camera of the car that sends the video streams to the PU_FCAM for further processing. Similarly, the Aud/Vid node transmits infotainment streams to the AVSink node. Both FCAM and Aud/Vid use classes A and B for the data transmission. In addition, we consider three control nodes (e.g., engine control) that send control signals to the Head Unit. The control end stations are denoted by CTRL1-CTRL3. The control signals are allocated to the ST class as we want to obtain full deterministic behaviour for them. Finally, we considered three end stations, being Diagnosis, Logging and Backup that transmit diagnostics information, stored data from other ECUs, and software update information. The data from these end stations are sent to the Head Unit. Table I shows the traffic characteristics.

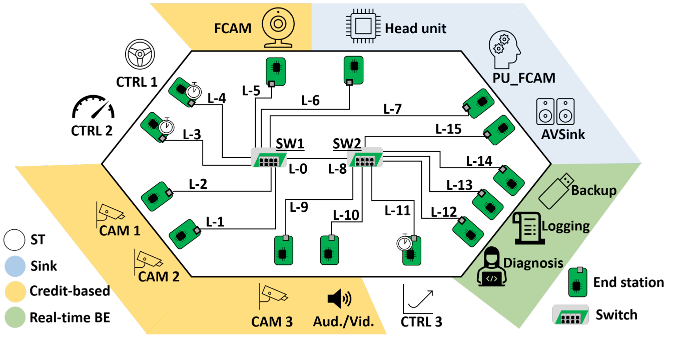

Fig. 6: Vehicular application use case topology.

TABLE I: Various traffic in the vehicular application use case.

ID	Sender	Receiver	$T = D$ (μs)	Payload ($Bytes$)	$C(\mu s)$	Class
M-0	Head unit	CTRL1	20000	20	49.6	ST
M-1	Head unit	CTRL2	20000	20	49.6	ST
M-2	Head unit	CTRL3	20000	20	49.6	ST
M-3	Head unit	CAM1	10000	786	662.4	A
M-4	Head unit	CAM2	10000	786	662.4	A
M-5	Head unit	CAM3	10000	786	662.4	A
M-6	PUFCAM	FCAM	5000	786	662.4	A
M-7	Head unit	CAM1	10000	786	662.4	B
M-8	Head unit	CAM2	10000	786	662.4	B
M-9	Head unit	CAM3	10000	786	662.4	B
M-10	AVSink	Aud./Vid.	5000	1472	1211.1	B
M-11	Head unit	Backup	10000	500	433.6	BE
M-12	Head unit	Logging	10000	1500	833.5	BE
M-13	Head unit	Diagnosis	10000	2000	1233.5	BE

As there are several class A and B messages, the values of the idleSlope for these two classes need to be allocated on every link through which these messages traverse. In order to calculate the idleSlope values, we use recommendation from the IEEE 802.1Q-2018 standard. Therefore, the idleSlope is equal to the utilization of the traffic with scaling up by considering the ST traffic. Table II shows the idleSlope values assigned to each link, where the link IDs are shown in Fig. 6. Note that zero credit means there are no messages from the associated CBS class on the link.

B. Analysis Results

We implemented the proposed analysis as an in-house analysis engine. The configuration and message set in the vehicular application use case were fed as input to the implemented

TABLE II: idleSlope of class A and class B per link.

idleSlope ($Mbps$)	l_1	l_2	l_5	l_6	l_7	l_8	l_9	l_{10}	l_{15}
Class A	1.32	1.32	2.64	3.97	2.64	1.32	1.32	0	0
Class B	1.32	1.32	3.97	3.97	0	1.32	1.32	4.84	4.84

analysis. The analysis took 15 milliseconds to run on an HP Elitebook with an Intel Core i5 processor and a 16-Gigabyte RAM. Table III shows the worst-case response times of all messages calculated by the implemented analysis. As it can be seen, the messages from classes ST, A and B meet their deadlines. However, two of the BE messages (M-12 and M-13) miss their deadlines as their response times ($10674 \mu s$ and $11074 \mu s$) exceed their deadlines ($10000 \mu s$). Note that the presented analysis calculates the response times of messages per link. Hence, the analysis can identify the most congested links that may become bottleneck in the TSN network. The network designer can use this information to check whether it is acceptable for the class BE messages to miss their deadlines with a small margin, otherwise a new configuration should be setup to refine the application by considering the bottlenecks indicated by the proposed analysis.

TABLE III: Calculated response times (RT) in the use case.

ID	$T = D(\mu s)$	$RT(\mu s)$	Schedulable
M-0	20000	104.2	Yes
M-1	20000	104.2	Yes
M-2	20000	158.8	Yes
M-3	10000	3808.7	Yes
M-4	10000	3808.7	Yes
M-5	10000	5792.9	Yes
M-6	5000	1329.8	Yes
M-7	5000	2427.4	Yes
M-8	10000	5843.9	Yes
M-9	10000	5843.9	Yes
M-10	10000	8506.5	Yes
M-11	10000	9474.0	Yes
M-12	10000	10674.0	No
M-13	10000	11074.0	No

VII. Conclusion and future work

In this paper, we argued that it is simpler to use BE class in TSN networks within a vehicle instead of classes A and B for the traffic with no hard real-time timing requirements. This is due to the complexity of the CBS configuration when it is combined with the TAS and frame preemption mechanisms in TSN. However, this requires us to provide a level of QoS for the BE traffic, thus a schedulability analysis for the BE class becomes essential. This paper presented the worst-case response-time analysis for the BE traffic in TSN while considering the effects of classes ST, A and B via the TAS and CBS, and the frame preemption support. To the best of our knowledge, this is the first schedulability analysis for the BE class in TSN networks. We used the presented analysis on a vehicular application use case to show how the analysis can provide essential information to the network designer. The future work entails using the analysis for BE traffic to improve the offline schedules for the ST traffic in TSN networks.

Acknowledgements

The work presented in this paper was supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the DESTINE, PROVIDENT and INTERCONNECT projects and the Swedish Knowledge Foundation via the DPAC and HERO projects.

References

[1] L. Lo Bello, R. Mariani, S. Mubeen, and S. Saponara, "Recent Advances and Trends in On-Board Embedded and Networked Automotive Systems," *IEEE Transactions on Industrial Informatics*, 2019.

[2] M. Ashjaei, L. L. Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, "Time-Sensitive Networking in Automotive Embedded Systems: State-of-the-Art and Research Opportunities," *Journal of Systems Architecture*, 2021.

[3] L. Lo Bello, M. Ashjaei, G. Patti, and M. Behnam, "Schedulability Analysis of Time-Sensitive Networks with Scheduled Traffic and Preemption Support," *Journal of Parallel and Distributed Computing*, 2020.

[4] M. A. Ojewale, P. M. Yomsi, and B. Nikolić, "Multi-level Preemption in TSN: Feasibility and Requirements Analysis," in *IEEE International Symposium on Real-Time Distributed Computing*, 2020.

[5] L. Zhao, P. Pop, Z. Zheng, H. Daigmorte, and M. Boyer, "Latency Analysis of Multiple Classes of AVB Traffic in TSN with Standard Credit Behavior using Network Calculus," *IEEE Transactions on Industrial Electronics*, 2020.

[6] U. D. Bordoloi, A. Aminifar, P. Eles, and Z. Peng, "Schedulability Analysis of Ethernet AVB Switches," in *International Conference on Embedded and Real-Time Computing Systems and Applications*, 2014.

[7] D. Thiele, R. Ernst, and J. Diemer, "Formal Worst-Case Timing Analysis of Ethernet TSN's Time-Aware and Peristaltic shapers," in *IEEE Vehicular Networking Conference*, 2015.

[8] D. Thiele and R. Ernst, "Formal Worst-Case Performance Analysis of Time-Sensitive Ethernet with Frame Preemption," in *IEEE International Conference on Emerging Technologies and Factory Automation*, 2016.

[9] M. Ashjaei, G. Patti, M. Behnam, T. Nolte, G. Alderisi, and L. L. Bello, "Schedulability Analysis of Ethernet Audio Video Bridging Networks with Scheduled Traffic Support," *Real-Time Systems*, 2017.

[10] M. A. Ojewale, P. M. Yomsi, and B. Nikolić, "Worst-Case Traversal Time Analysis of TSN with Multi-level Preemption," *Journal of Systems Architecture*, 2021.

[11] J. A. R. De Azua and M. Boyer, "Complete Modelling of AVB in Network Calculus Framework," in *Proceedings of the International Conference on Real-Time Networks and Systems*, 2014.

[12] M. Boyer, H. Daigmorte, N. Navet, and J. Migge, "Performance Impact of the Interactions between Time-triggered and Rate-constrained Transmissions in TTEthernet," in *European Congress on Embedded Real Time Software and Systems* , 2016.

[13] L. Zhao, P. Pop, Z. Zheng, and Q. Li, "Timing Analysis of AVB Traffic in TSN Networks using Network Calculus," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2018.

[14] L. Zhao, P. Pop, and S. S. Craciunas, "Worst-Case Latency Analysis for IEEE 802.1 Qbv Time Sensitive Networks using Network Calculus," *IEEE Access*, 2018.

[15] L. Maile, K. Hielscher, and R. German, "Network Calculus Results for TSN: An Introduction," in *Information Communication Technologies Conference*, 2020.

[16] T. L. Mai, N. Navet, J. Migge, "A Hybrid Machine Learning and Schedulability Analysis Method for the Verification of TSN Networks," in *International Workshop on Factory Communication Systems*, 2019.

[17] ——, "On the Use of Supervised Machine Learning for Assessing Schedulability: Application to Ethernet TSN," in *Proceedings of the International Conference on Real-Time Networks and Systems*, 2019.

[18] T. L. Mai and N. Navet, "Improvements to Deep-learning-based Feasibility Prediction of Switched Ethernet Network Configurations," in *International Conference on Real-Time Networks and Systems*, 2021.

[19] J. Cao, P. J. L. Cuijpers, R. J. Bril, and J. J. Lukkien, "Tight Worst-case Response-time Analysis for Ethernet AVB using Eligible Intervals," in *IEEE World Conference on Factory Communication Systems*, 2016.

[20] D. Maxim and Y.-Q. Song, "Delay Analysis of AVB Traffic in Time-Sensitive Networks (TSN)," in *Proceedings of the International Conference on Real-Time Networks and Systems*, 2017.

[21] B. Houtan, M. Ashjaei, M. Daneshtalab, M. Sjödin, and S. Mubeen, "Synthesising Schedules to Improve QoS of Best-effort Traffic in TSN Networks," in *International Conference on Real-Time Networks and Systems*, 2021.

[22] J. Maki-Turja and M. Nolin, "Fast and Tight Response-Times for Tasks with Offsets," in *Euromicro Conference on Real-Time Systems*, 2005.

[23] H. Lim, K. Weckemann, and D. Herrscher, "Performance Study of an In-car Switched Ethernet Network without Prioritization," in *Communication Technologies for Vehicles*, 2011.