Journal of Systems Architecture 116 (2021) 102037

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A novel frame preemption model in TSN networks

Mohammad Ashjaei [*], Mikael Sjödin, Saad Mubeen

Mälardalen University, Västerås, Sweden

ARTICLE INFO

Keywords:
Time-sensitive networking
TSN
Performance analysis
Response times
Frame preemption
Automotive networks

ABSTRACT

This paper identifies a limitation in the frame preemption model in the TSN standard (IEEE 802.1Q-2018), due to which high priority frames can experience significantly long blocking delays, thereby exacerbating their worst-case response times. This limitation can have a considerable impact on the design, analysis and performance of TSN-based systems. To address this limitation, the paper presents a novel and more efficient frame preemption model in the TSN standard that allows over 90% reduction in the maximum blocking delay leading to lower worst-case response times of high priority frames compared to the frame preemption model used in the existing works. The paper also shows that the improvement becomes even more significant in multi-switch TSN networks. In order to evaluate the effects of preemption, the paper performs simulations by enabling and disabling preemptions as well as enabling and disabling the Hold/Release mechanism supported by TSN. Furthermore, the paper performs a comparative evaluation of the two models of frame preemption in TSN using simulations. The evaluation shows that the maximum response times of high priority frames can be significantly reduced with very small impact on the response times of lower priority frames. The paper also shows the improvement in the maximum response times of higher priority frames using an automotive industrial use case that employs a multi-hop TSN network for on-board communication.

1. Introduction

The IEEE Time Sensitive Networking (TSN) task group [1] was established in 2012 to develop a set of high-bandwidth and low-latency real-time network communication protocols based on switched Ethernet. The previous generation of these protocols, developed by the Audio Video Bridging task group [2], were unable to support many novel features provided by the TSN protocols, for example, offline Scheduled Traffic (ST), among others. These features in TSN enable the support for hard real-time traffic. However, it becomes a daunting challenge for the network designers to use various combinations of these features when designing TSN-based applications. The research community has taken several initiatives to address this challenge, e.g., developing timing analysis considering various features in the TSN standards [3–5], configuration and optimization of TSN networks [6], and utilization of TSN in various application domains [7–10].

In this paper we study one such feature, namely the frame preemption, described in the IEEE 802.1Qbu amendment and is now rolled into the IEEE 802.1Q-2018 base standard [11]. We identify a limitation in the frame preemption model in TSN, which is also followed by all the existing works in the area. The limitation manifests from the way the preemption model is described that can result in potentially long blocking delays for high priority frames (see Section 4.1 for details). Consequently, these blocking delays can increase the worst-case response times of high priority frames, which may result in violation of their timing requirements. Furthermore, the limitation in the frame preemption model combined with the other well-defined features in TSN, including the ST traffic, gate mechanism, and credit-based shapers, can have a considerable impact on the design and network performance in TSN-based applications. To address the aforementioned limitation, we propose a novel and more efficient frame preemption model that conforms to the TSN standards. The proposed model allows significant reduction in the blocking delays of high priority frames resulting in their lower worst-case response times compared to the case when the classical model of frame preemption is used. Moreover, we perform a comprehensive comparative evaluation of two models of frame preemption considering the effect of combining various features in TSN in both single- and multi-hop networks. **The main contributions** in this paper are as follows.

- We identify a limitation in the frame preemption model in the TSN standard (IEEE 802.1Q-2018), due to which high priority frames can experience significantly long blocking delays, thereby exacerbating their worst-case response times.
- We propose a novel and more efficient frame preemption model in TSN. We show that the proposed model results in lower maximum

[*] Corresponding author.
E-mail addresses: mohammad.ashjaei@mdh.se (M. Ashjaei), mikael.sjodin@mdh.se (M. Sjödin), saad.mubeen@mdh.se (S. Mubeen).

https://doi.org/10.1016/j.sysarc.2021.102037
Received 29 June 2020; Received in revised form 23 December 2020; Accepted 25 January 2021
Available online 27 January 2021

blocking delays (over 90% reduction in the maximum blocking delay) for high priority frames leading to their lower worst-case response times compared to the classical frame preemption model in TSN that is used in the existing works.

- We quantify the improvement in the response times of high priority frames due to the proposed frame preemption model (complying with the TSN standard) in comparison to the classical frame preemption model. We also show that the proposed model provides significantly better response times for high priority frames in multi-switch TSN networks with very small impact on the response times of low priority frames.
- We perform simulations to evaluate the effects of enabling and disabling preemptions, enabling and disabling the Hold/Release mechanism together with preemption.
- We perform comparative evaluation of the two models of frame preemption in TSN using simulations. In addition, we evaluate the two models by using an automotive industrial use case.

2. Scheduled traffic and preemption in TSN

The frame preemption is updated with respect to the previous standard (IEEE 802.3br [12]) to allow preemption of an ongoing frame transmission in favor of an urgent traffic by means of the "Express/Preemptable" property associated to each traffic class. This section presents some of the features in TSN that are tightly related to frame preemption.

2.1. Queuing and forwarding mechanism

The queuing and forwarding mechanism supported by the TSN standards distinguishes between critical and non-critical traffic classes by applying strict priority scheduling between them. The critical traffic classes are the Stream Reservation (SR) classes A and B, while non-critical traffic is the best-effort (BE) class. Class A is the highest priority traffic class, whereas class B is the second highest priority class. A credit-based shaper (CBS) algorithm is defined to control the SR transmission. According to this algorithm, every SR class gets a certain credit to allow transmission of its frames. A frame in each SR class can be transmitted only if the credit is zero or positive. The credit decreases with a constant rate, known as the *sendSlope*, during the frame transmission. Whereas, the credit replenishes with a constant rate, known as the *idleSlope*, when there is a frame pending for transmission but the port is busy in transmitting frames belonging to other classes. The idleSlope is configured in the switch for each SR traffic class. The sendSlope is equal to the difference between the port rate and the idleSlope. The summation of idleSlope values in all SR classes should not exceed the value of the port rate.

2.2. Enhancement for scheduled traffic

The enhancement for Scheduled Traffic (ST) in the TSN standards provides temporal isolation for the urgent traffic. The ST traffic is scheduled for transmission according to a static schedule (created offline). In particular, *transmission gates* are associated to each queue of a switch port, and transmission from a queue is allowed only if the relevant gate is open. The gates operation follows a list, called the Gate Control List (GCL) that is periodically repeated. The gate mechanism allows to temporally isolate the scheduled traffic from the other traffic classes. That is, if a frame belonging to a non-ST class becomes ready for transmission close to a gate opening event in the GCL, the non-ST frame is transmitted only if its transmission can be completed before opening of the gate for the ST traffic. According to Clause 8.6.8.2 in the IEEE 802.1Qbv standard, the credit for the SR classes is accumulated only when the transmission gates for the corresponding queues are open, and the idleSlope for SR traffic is scaled up compared to the one defined in the CBS algorithm. The credit is "frozen" during the time when the gates are closed for the SR classes.

2.3. Preemption support

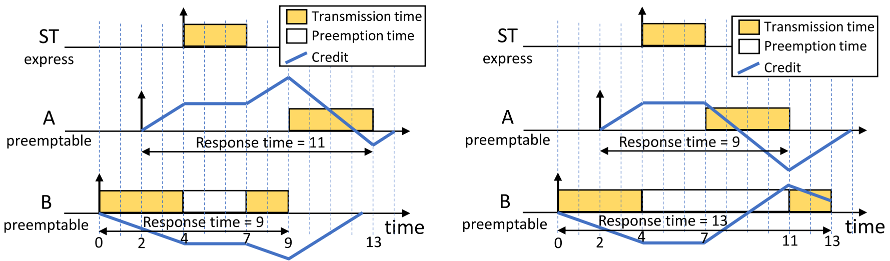
The TSN standard defines support for frame preemption on a transmission port of a switch. The preemption support introduces *express* and *preemptable* modes for each traffic class. There is a corresponding Medium Access Control (MAC) interface to provide such a support. The express traffic can preempt the preemptable traffic, but it cannot be preempted. Preemption support can be combined with the CBS and gate mechanisms discussed in Sections 2.1 and 2.2. For example, the ST class can be set as express, while the rest of the classes are set as preemptable.

3. Related work

The main contributions in the TSN standards with respect to the previous generation of protocols were to introduce the time-triggered transmission on top of other traffic classes. Some of the major enhancements include the support for ST traffic, frame preemption support and clock synchronization. Many works have investigated different aspects of TSN, including different time-aware shapers [13], scheduling policies [14], load balancing in TSN networks [15], fault tolerance [16] and holistic modeling of TSN communication [17]. Moreover, applicability of using TSN for vehicular on-board communication has also been studied recently in few works, e.g., [7] and [8]. In order to guarantee the timeliness of traffic in TSN networks, some works addressed the analysis and simulation of TSN networks. Few works have addressed the schedulability analysis of traffic from classes A and B. An analysis is given in [3] to compute the worst-case delay of frames in TSN considering the time-aware shaper for single-switch networks. Moreover, the work presented in [18] proposed an analysis for time-aware shaper and peristaltic shaper, while a very recent work in [4] presented an analysis based on network calculus. Moreover, the work in [19] presents a timing analysis considering frame preemption in TSN networks. The work in [20] complements the previous analysis by considering various modes in combination with the Hold/Release mechanism as well as the delay analysis for ST frames.

Besides the timing analysis, several works have addressed different performance analyses for the TSN networks. The work in [21] proposed a new mechanism within the preemption to reduce the amount of overhead. The work in [22] presented an analysis in utilizing TSN in industrial automation systems. The analysis is based on the network calculus to evaluate the frame delays. Moreover, the performance analysis of various traffic allocations, including stream-based and class-based, is studied in [23]. In order to design an optimum bandwidth allocation, the work in [24] proposed a method to derive bandwidth on each port of a TSN network. Moreover, the work in [25] proposed a technique to compute the worst-case backlog per class that allows a better network design. The effect of guard band while using the gate mechanism in combination with the CBS algorithm is presented in [26]. The paper provides evidence that the standard proposal on freezing the credit can lead to overflow in some corner examples. A recent work in [27] indicates that considering the express mode for a class of traffic may impose higher latency for other high-priority preemptable traffic classes that have firm timing constraints but do not have the express mode enabled. In order to improve the preemption method, the work proposed a modification in the MAC layer to obtain a multi-level preemption in which higher priority preemptable class may preempt lower priority preemptable class. Compared to the work in [27], we consider a different system model. The main differences are: (i) we consider the scheduled traffic and credit-based shaper together with preemption, (ii) our model considers the highest priority traffic to be express, while other traffic to be preemptable, and (iii) we define the changes only in the pMAC without adding any other modules to the MAC layer, whereas the work in [27] suggests to add a tpMAC module.

To sum up, several attempts have been carried out (and are ongoing) to study the performance of TSN networks from different angles in



(a) Classical preemption model

(b) Novel and efficient preemption model

Fig. 1. Two models of frame preemption in TSN.

various application domains. In this paper, we propose a novel frame preemption model that complies with the TSN standard. We show the impact of the proposed model on the network performance. Moreover, the proposed model is extensively evaluated with respect to the traditional frame preemption model used in the existing works. To the best of knowledge, none of the previous works have addressed the studied issues in this paper.

4. Investigation of the preemption effects

This section investigates and presents the effects of enabling preemption on the performance of TSN networks. First, we identify that there can be multiple models of frame preemption in TSN. Thereafter, we study the effect of these models on the response times of frames belonging to various traffic classes.

4.1. Classical model of frame preemption in TSN

One of the complex configurations in a TSN network is when the preemption is enabled in combination with the gate mechanism and credit based shaper. According to Clause 8.6.8.2 in the IEEE 802.1Q-2018 standard, the transmission of a frame in a preemptable class can be interrupted by a frame in an express class. In this case, the interrupted frame waits in the MAC, which only happens when the preemption mechanism is used.

> *According to the existing frame preemption model in the TSN standard, if a frame belonging to a preemptable class (say Frame 1) is preempted by another frame belonging to an express class (say Frame 2), Frame 1 resumes its transmission once Frame 2 has been transmitted even if a higher priority frame belonging to another preemptable class (say Frame 3) is ready for transmission with non-negative credit at the time when the transmission of Frame 2 is completed, thereby causing a blocking delay to the higher priority Frame 3.*

An example of the above scenario is depicted in Fig. 1(a), where three frames belonging to three different classes are transmitted on one port. The ST class is set to express, while classes A and B are set to preemptable. Class B frame is ready at time 0 and since it has non-negative credit, it can be transmitted. Although class A frame becomes ready at time 2, it has to wait because it is set to preemptable and there is an ongoing transmission. On the other hand, ST frame can preempt the transmission of other classes as it is set to express. Therefore, the ST frame preempts the transmission of class B frame from time 4 to time 7. After the transmission of the ST frame is completed at time 7 the class B frame resumes its transmission, despite class A frame being ready to transmit with non-zero credit. At time 9, the transmission of the class B frame is completed, at which time the class A frame can start its transmission. The response times of the class A and B frames are 11 and 9 time units respectively.

4.2. Novel and efficient model of frame preemption in TSN

This section presents a novel model for the frame preemption mechanism in the TSN standard.

> *According to this model of frame preemption, if a frame belonging to a preemptable class (say Frame 1) is preempted by another frame belonging to an express class (say Frame 2), and if a frame belonging to another preemptable class (say Frame 3) with priority higher than the priority of Frame 1 is ready for transmission at the time when the transmission of Frame 2 is completed then Frame 3 will be transmitted before resuming the transmission of Frame 1 as long as the traffic class of Frame 3 has a non-negative credit.*

In the new model, we allow the MAC to store the preempted frame if there is a higher priority frame ready for transmission. Fig. 1(b) depicts the same example in Fig. 1(a) with the new model of frame preemption in TSN. In this example, after transmission of the ST frame at time 7, since the credit for class A frame is positive as well as the priority of class A frame is higher than the priority of the preempted class B frame, the class A frame is transmitted while keeping the class B frame in the MAC. As it can be seen, the response time of class A frame decreases to 9 time units from 11 time units compared to Fig. 1(a). On the other hand, the response time of class B frame increases to 13 time units from 9 time units compared to Fig. 1(a). Basically, the new model does not allow any blocking for the higher priority frames from the lower priority frames using the opportunity of interrupting the lower priority transmission. In the rest of the paper, we represent the classical implicit preemption definition by *blocking preemption definition* and the newly proposed model by *non-blocking preemption definition*.

It can be seen from the two scenarios in Fig. 1 that the response times of higher priority frames belonging to preemptable traffic class can be improved (reduced) by using the non-blocking preemption definition. The improvement in the response times is attributed to reduction in the maximum blocking delay, which is discussed in the next subsection.

We further describe the novel frame preemption model in TSN in a generic form as follows. An express frame can preempt any lower priority preemptable frame. After the transmission of the express frame is completed, the MAC selects the next frame for transmission by first checking whether there is a higher priority frame pending with zero or positive credit. The preemption point by an express frame can be seen as a *system tick* where the scheduler checks for the next frame for transmission based on the priority and availability of credit (with the same order of importance), instead of resuming any pending frame. This means that the buffer that is holding a preempted frame may need to keep more than one frame that is originally foreseen in the standard preemption model. An example of this case can be seen in Fig. 2. At time 0 a class B frame starts its transmission as there is no other frame pending and the credit is 0. Although at time 2 a class A frame is ready for transmission, the transmission of class B frame continues as both classes are preemptable. At time 4 an express frame arrives that can preempt other classes and its transmission finishes

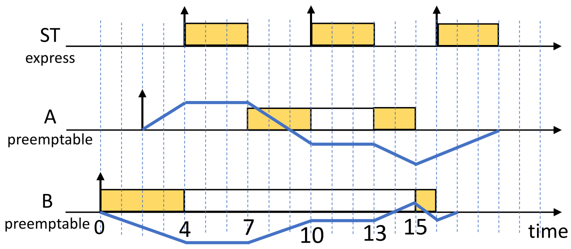


Fig. 2. Nested preemption in TSN.

at time 7. At this point, because of the preemption point, the MAC can select among all preemptable frames. Since there is a pending frame from class A with higher priority than the pending class B, the MAC selects it for transmission. However, at time 10 another express frame preempts the transmission of other classes. At this time we have two frames preempted in the MAC. At time 13 the MAC should select among pending frames. Therefore, the higher priority frame is selected for transmission. At time 15 the pending frame in class B resumes its transmission. This situation is called *nested preemption* in real-time systems which prevents any priority inversion in the system. This requires an update in the MAC which will be discussed in the next section.

Another consequence of the non-blocking preemption is on the credit behavior. According to the TSN standard, the credit is frozen during the preemption by an express frame. We also follow this definition in case of the non-blocking preemption. Therefore, during a preemption that is imposed by a high priority non-express frame the credit is not frozen as the preemption did not occur due to an express frame. Looking at the example in Fig. 2, at time 7 until time 10 the class A frame is being transmitted as it got a chance after the ST preemption. During this time, however, the credit of class B frame increases following the rules of credit in the standard. Again, at time 10 the credit is frozen until time 13 because of preemption by an ST frame.

4.3. Improvement in the maximum blocking time

Let us first consider the blocking definition of preemption and calculate the maximum blocking time experienced by a high-priority frame (f_i) belonging to a preemptable class due to lower priority frames belonging to another preemptable class. We denote this blocking time experienced by frame f_i by B_i^{BP}, the value of which is equal to the maximum transmission time in the set of frames with priorities lower than f_i.

$$B_i^{BP} = \max_{\forall j \in lp(f_i)} C_j \tag{1}$$

In this context, the worst-case scenario for f_i occurs when it becomes ready for transmission while a low-priority frame belonging to another preemptable class that already started its transmission is preempted by the highest-priority express frame. Note that the transmission time (C_j) of a TSN frame is computed using the following equation [3].

$$C_j = \frac{FrameSize_j * 8}{PortRate} = \frac{(Payload_j + Header_j) * 8}{PortRate} \tag{2}$$

where $PortRate$ represents the network speed on the port. According to [28], the maximum blocking time for a frame is computed as the maximum frame size among all lower priority classes. The maximum size of a TSN frame contains 1500 Bytes of payload and 42 Bytes of frame overhead. Assume that the $PortRate$ is equal to 100 Mbit/s and using Eq. (2) considering the blocking preemption definition, the maximum blocking time experienced by the high priority frame belonging to a preemptable class is 123.36 μs. That is,

$$B_i^{BP} = (\frac{(1500 + 42) \times 8}{100}) = 123.36 \text{ μs}$$

Now consider the case of the non-blocking preemption definition. The blocking time experienced by f_i in this case is denoted by B_i^{NBP}. The blocking term B_i^{NBP} is defined as the blocking time experienced by f_i due to the lower priority traffic transmission under non-blocking preemption. This blocking time can be much lower compared to the case of the blocking preemption definition. According to Annex R.2 in the IEEE 802.1Q-2018 standard, a frame with less than or equal to 123 Bytes (including the frame overhead) cannot be preempted. Considering the port rate of 100 Mbit/s, the blocking time experienced by f_i can become 9.84 μs, according to Eq. (2).

$$B_i^{NBP} = (\frac{123 \times 8}{100}) = 9.84 \text{ μs}$$

Hence, **the maximum improvement in the blocking delay** experienced by the high-priority frame using the non-blocking preemption definition can be **113.52** μs (123.36 μs - 9.84 μs), which is over 90% improvement in the blocking delay.

4.4. Effect of multiple blocking on the response time under non-blocking preemption

According to [29], a frame in TSN network may face multiple blocking from low-priority frames due to the credit behavior. We refer the interested readers to the formal proofs in [29]. Here, we show that the effect of multiple blocking does not lead to more improvement in the response time of a high-priority frame under the non-blocking preemption definition.

We first consider the blocking preemption definition. Fig. 3 shows an example that contains three different traffic classes, including an ST class (set to express) and classes A and B (both are set to preemptable). There are two frames in Class A that are activated at time 2. The other classes have different activation times per frame as shown by the vertical arrows in Fig. 3. Note that the frame of interest is the class A frame and is highlighted with green color. The first frame in class B is preempted by the first ST frame between time 4 and 7. The class B frame resumes at time 7 and finishes its transmission at time 9. The first frame in Class A (identified as the yellow frame) is transmitted between time 9 and 13. At time 13, the frame of interest (second frame in Class A identified with green color) is ready but cannot start its transmission because the credit for Class A is negative. The second frame in class B is activated at time 14. Since the credit for class B is zero and the credit for class A is still negative (a little less than zero), the second frame of class B initiates its transmission. Similar preemption is caused by another ST frame which makes the frame of interest to start and complete its transmission at times 23 and 26 respectively, resulting in its response time of 24 time units. Using the blocking preemption definition, the frame under analysis experiences blocking by two lower priority frames because of the credit behavior.

Now we consider the case of the non-blocking preemption.

Lemma 1. *Using the non-blocking preemption definition, response time of a frame will be improved only by the size of one blocking frame and not by multiple blocking frames on a link.*

Proof. The only scenario in which the response time of the frame of interest can be improved (while keeping the rest of the parameters the same) is when the previous frame in the same class can be transmitted earlier. In comparison to the blocking preemption, the non-blocking preemption will actually lead to earlier transmission of the frame of interest. As shown in Fig. 4, the non-blocking preemption will allow the first frame in Class A to start its transmission at time 7 and complete its transmission at time 11 instead of finishing at time 13 in the case of the blocking preemption depicted in Fig. 3. In order for the frame of interest to be transmitted earlier, the credit should become (at least) zero earlier. However, in this proof, we show that in both definitions of preemption the credit becomes zero at the same time. This means that

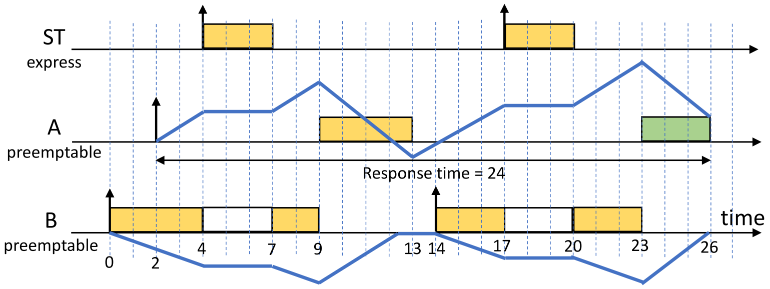


Fig. 3. Blocking preemption. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

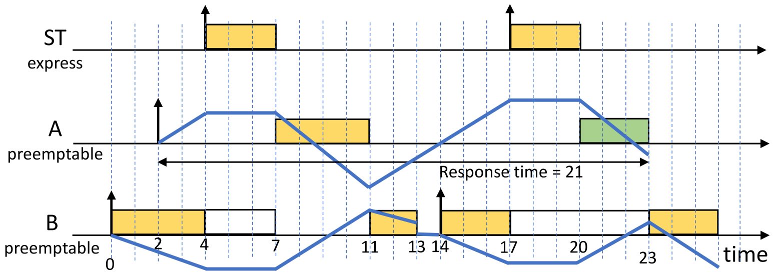


Fig. 4. Non-blocking preemption.

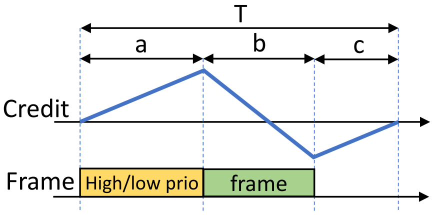


Fig. 5. Case I: interference at the beginning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

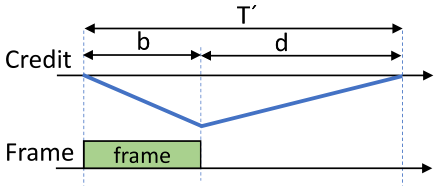


Fig. 6. Case II: no interference. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

even if the first frame in class A can be transmitted earlier in Fig. 4 compared to Fig. 3, in both cases the credit becomes zero at time 14.

There are three possible cases that can occur for a credit in an SR class as shown in Figs. 5, 6 and 7. The interval between the time when the credit is zero and the time when the credit becomes zero again after the frame transmission is denoted by T in Case I, T' in Case II and T'' in Case III. The frame transmission time is the same in all cases and is equal to b, i.e., the time interval during which the credit decreases because of the frame transmission. The frame transmission is highlighted with green color in the figures.

Case I: This case defines a scenario in which the frame of interest is ready for transmission but a frame from another higher-priority or low-priority class is already under transmission. During the transmission of the other frame, the credit for the Class of the frame of interest will be replenished. This interval is indicated by 'a' in Fig. 5. The frame of interest starts its transmission when the port becomes free. After transmission of the frame, the credit replenishes to zero as indicated by the interval 'c' in Fig. 5. Hence,

$$a + b + c = T \tag{3}$$

The rate of increasing and decreasing the credit are defined by α^+ and α^-, respectively. Therefore, for Case I:

$$a.\alpha^+ + b.\alpha^- + c.\alpha^+ = 0 \tag{4}$$

Case II: In this case, we assume that the frame of interest does not receive any interference in the beginning. The frame starts its transmission immediately when the credit for its class is zero. After the transmission, the credit replenishes to zero again as indicated by the interval 'd' in Fig. 6. Therefore, for Case II:

$$b.\alpha^- + d.\alpha^+ = 0 \tag{5}$$

Solving Eqs. (4) and (5), we derive:

$$a.\alpha^+ + b.\alpha^- + c.\alpha^+ = b.\alpha^- + d.\alpha^+ \tag{6}$$

Eq. (6) can be simplified as follows:

$$\begin{aligned} a.\alpha^+ + c.\alpha^+ = d.\alpha^+ &=> a + c = d \\ a + c + b &= b + d \\ T &= T' \end{aligned} \tag{7}$$

Therefore, in both cases the duration of credit changing from zero and replenishing to zero again is equal.

Case III: In this case, we assume that the interference (both by low-priority and high-priority frames) received by the frame of interest when it is ready for its transmission is large leading to a large increase in the credit as shown by the interval 'e' in Fig. 7. The difference between this case and the case in Fig. 5 is the behavior of the credit. In Case III, the credit will not become negative during the frame transmission and has to become zero immediately after full transmission of the frame. Whereas, in Case I the credit becomes negative and it has to be replenished to zero after the frame transmission. Therefore, in Case III the interference at the beginning should be large enough to increase the credit such that it cannot become negative during the frame transmission. According to the credit-based shaper, when

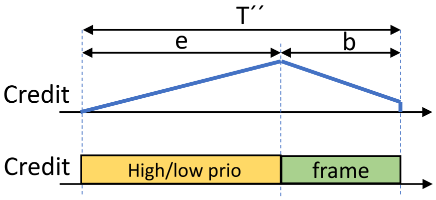


Fig. 7. Case III: large interference in the beginning. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

there is no frame pending for transmission and the credit is positive it becomes zero immediately. Hence, after the transmission is completed, the remaining credit for the class of the frame under analysis becomes zero. Similar to the previous cases, we can write:

$$e.\alpha^{+} + b.\alpha^{-} = 0 \tag{8}$$

Solving Eqs. (4) and (8), we can derive:

$$a.\alpha^{+} + b.\alpha^{-} + c.\alpha^{+} = e.\alpha^{+} + b.\alpha^{-} \tag{9}$$

Eq. (9) can be simplified as follows:

$$\begin{aligned} a.\alpha^{+} + c.\alpha^{+} = e.\alpha^{+} &=> a + c = e \\ a + c + b &= e + b \\ T &= T'' \end{aligned} \tag{10}$$

We can conclude that in all the possible cases, the time duration of using the credit for the frame transmission is equal, i.e., $T = T' = T''$, regardless of any interference to the frame transmission at any time after its activation. Therefore, the earlier transmission of previous frames in a class because of the non-blocking preemption cannot lead to earlier transmission of the frame in the same class. This is because the credit replenishment to zero at a certain time happens regardless of when the frame transmission occurs during the interval when the credit changes from zero and back to zero again. **Hence, the non-blocking definition of preemption improves the response time of a frame only by the size of one blocking frame over a link.** □

Lemma 1 showed that the response time of a frame will be improved by the size of one blocking frame under non-blocking preemption over one link. Note that according to [20], the response time of a frame on multiple links is the summation of the response times on each link on the path of the frame. Therefore, we can extend the lemma to state that the response time of a frame will be improved by the size of one blocking frame per link over the path of the frame under non-blocking preemption.

4.5. Non-blocking preemption and multi-hop networks

In the previous subsections, we have investigated the effect of the non-blocking preemption only in one port. The investigation results indicate that the response time of a frame can be improved by 113.52 μs in a 100 Mb/s network, thanks to the reduced blocking delay. This improvement can be increased manifolds in the industrial systems that employ the TSN network with multiple switches. Although the improvement on one port is limited, here we show that the response time improvement can be significant when a frame crosses multiple hops. To show this, we present an example in Fig. 8 by considering the blocking preemption. In this example, we consider a frame from class A as the frame of interest, identified as the green frame in Fig. 8. The green frame is transmitted through two ports, Port 1 to Port 2. The frame is activated at time 2, but the port is busy in transmitting an ST frame. The green frame starts its transmission after the frame from class B has completed its transmission. The green frame arrives at Port 2 at time 10. However, another frame in class A was activated slightly before the arrival of the green frame. Therefore, the green frame has to wait for the credit replenishment, and is fully transmitted at time 29. The response time of the green frame in this example is 27 time units.

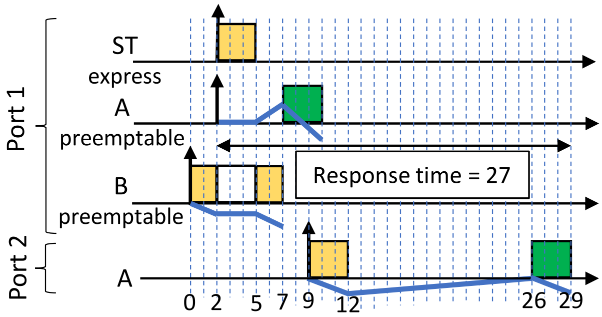


Fig. 8. Blocking definition in multi-hop TSN networks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

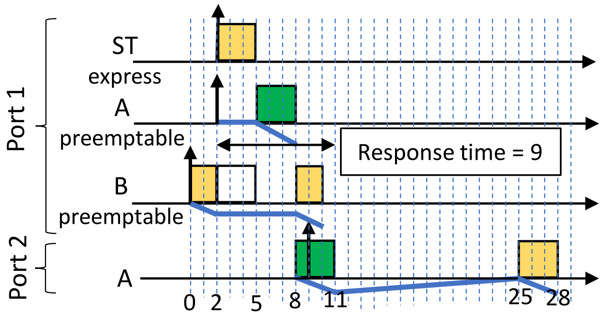


Fig. 9. Non-blocking definition in multi-hop TSN networks.

Now consider the non-blocking definition in the same example shown in Fig. 9. Since the transmission of the green frame on Port 1 initiated earlier, it can be transmitted on Port 2 earlier than the frame belonging to the same class, resulting in the much lower response time of 9 time units for the green frame. As it can be seen in this example that the improvement in the response time of the frame occurs by avoiding the blocking from the lower priority frames as well as by avoiding the same priority interference and from the effect of the credit behavior on multi-hop networks. This improvement can be significant depending on the credit allocation on each port.

4.6. Hold and release mechanism

According to Annex S of the IEEE 802.1Q-2018 standard, when the preemption mechanism is used in combination with the scheduled traffic two modes can be implemented, being “with” or “without” Hold/Release mechanism. In most of the TSN performance studies, including the timing analysis of the network, (e.g., [4,5,19,30]), this mode is considered explicitly as using Hold/Release. The very recent work in [20] considers both modes in the worst-case delay analysis of classes A and B. Nevertheless, selection of this mode can have a considerable effect on the performance of both the SR and ST traffic. First, we consider the working principle of this mode. The Hold/Release mechanism in TSN allows to implement an explicit “guard band” before a transmission window for ST traffic. As it is mentioned before, a frame with size less than or equal to 123 Bytes cannot be preempted. In order to protect the transmission of ST frames, the transmission of any frame belonging to other non-ST classes is prevented during the time interval corresponding to the transmission of 123 Bytes before the scheduled transmission of an ST frame. This mechanism ensures that the ST transmission will have zero jitter. In the mode without Hold/Release, the preemptable traffic is allowed to continue transmitting up to 123

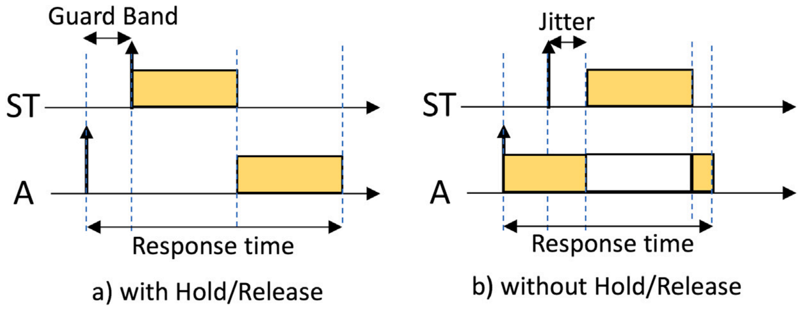


Fig. 10. Hold/Release mechanism in TSN.

bytes even *after* the transmission gate of the express traffic has opened. In this mode, there is no guard band required and in fact the ST traffic may be delayed due to not fully protecting the interference. Whereas, in the case of using the Hold/Release mechanism, the ST traffic does not experience any jitter, however, at the cost of larger response times of non-ST frames.

Fig. 10 shows the two modes in combination with the preemption mechanism. As it can be seen in this example, with Hold/Release mechanism there is zero jitter for the ST transmission at the cost of longer response times of the frames belonging to the other classes. On the other hand, the ST traffic will experience a maximum jitter of up to the transmission time of 123 Bytes without Hold/Release mechanism, while the other classes may get shorter response times down to the transmission time of 123 Bytes. In Section 6, we show the effects of the two mentioned mechanisms based on running extensive random cases using simulations.

5. Implications on the H/W support and analysis tools

The proposed novel model of non-blocking preemption in TSN comes with several implications on the hardware support and timing guarantees. In this section we discuss and present the modifications on the hardware and analysis tools according to the proposed non-blocking preemption in TSN.

5.1. Hardware support

According to the IEEE 802.3br (and consequently the IEEE 802.1Qbu) standard an extension is needed to support preemption on the MAC layer. The block diagram of the preemption support is shown in Fig. 11. The implementation requires two sub-MAC layers, known as eMAC for handling express traffic and pMAC for handling preemptable traffic. MAC Merge sub-layer is the main component in the preemption support, which receives a request (`MM_CTL.request`) from MAC Client to specify hold or release transmission of preemptable traffic with two values of `HOLD` and `RELEASE` for the request. In the case where preemption is enabled, the MAC Merge layer sends MAC Merge Packet (known as mPacket) to Reconciliation layer, which is an interface to the physical layer (PHY). Note that when the preemption is disabled, the MAC Merge layer will send packets instead of mPackets. An mPacket contains one of the following: (i) a complete express packet, (ii) a complete preemptable packet, (iii) an initial fragment of a preemptable packet, or (iv) a continuation fragment of a preemptable packet. An mPacket contains a 1-byte SMD (Start mPacket Delimiter) that indicates whether the packet is holding an express or preemptable packet. SMD can also indicate whether the mPacket is an initial or a continuation fragment of a preemptable packet. In addition, the mPacket contains a fragmentation count (`FRAG_COUNT`) that increments for each continuation fragment of the preemptable packet to prevent reassembling an invalid packet in case any fragment is lost. An mPacket has a Cyclic Redundancy Check (CRC) field that contains an mCRC value. The value is calculated based on the first byte of the initial mPacket up to the last byte of the transmitted mPacket.

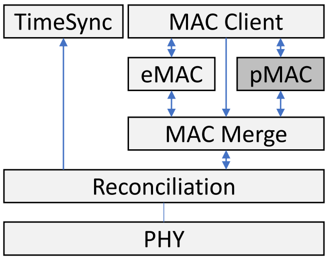


Fig. 11. The block diagram of preemption support.

The transmission and reception processes in the case of preemption enabled are described in Section 99.4.4 and Section 99.4.5 of the IEEE 802.3br standard, respectively. The transmission process starts by receiving a packet from eMAC or pMAC and preempts the coming packet if the MAC Client sends the hold request. The transmission of the packet from pMAC resumes after full transmission of the mPacket from eMAC, by sending preamble followed by the SMD before continuing transmission of the preempted packet. The transmission process generates an mCRC for each mPacket as specified above. In the receiving node, the reception process checks the SMD of the incoming packets. Based on the SMD, the process can indicate if the incoming packet is an express mPacket or a partial fragment of a preemptable packet. Consequently, the receiving packet will be forwarded to eMAC or pMAC. In the case of receiving a preempted packet, the frame count is checked to identify which part of the preempted packet is being received. Moreover, the reception process checks whether the last four bytes of the mPacket match the mCRC value. If they match then that means the packet was preempted and the process will wait until it receives the next mPacket. Note that the reconciliation interface is synchronized by the TimeSync block in the communication stack.

The new model of frame preemption in TSN requires storing more than one packet in the pMAC due to the nesting preemption cases. When the MAC Client sends a request to hold, the MAC Merge will preempt the packet and transmit the mPacket instead. In addition to this change, few changes should be done in the transmission and reception processes. After transmission of the mPacket, the transmission process should select among several preempted packets in the pMAC. The selection should be based on (i) priority of the packet that should be checked in the 802.1Q tag, and (ii) the credit availability in the queue. According to Section 8.6.8.2 of the IEEE 802.1Qbu standard, when the credit based shaper algorithm is combined with preemption then the credit and transmission will be checked by the MAC layer. Depending on if the mPacket is an initial or continuation fragment mPacket, different SMD values will be used, which are known as SMD-S for the initial fragment mPacket and SMD-C for the continuation fragment mPacket. The details are described in Section 99.3.3 of the IEEE 802.3br standard. In order to support the proposed preemption model the SMD values should be extended to encode the packet identifiers. This is important to ensure that the packets that are being preempted are correctly reassembled in the receiver. The reception process stays the same with an exception that the receiver node may receive different preempted frames that need to be checked by the frame count in the preempted frame structure. The reception process first checks the SMD value of any received mPacket. If it is an SMD-S then the mPacket is an initial fragment, while if it is an SMD-C it is a continuation fragment. The reception process then should check if there is an ongoing preemption. If there is no ongoing preemption, then receiving an SMD-C is an error which should be reported to the pMAC layer. If there is an ongoing preemption, it should check if the packet identifier encoded in the SMD field matches the one that is preempted. This indicates that the same packet being preempted is resumed for transmission. However, if an SMD-S or an SMD-C with different packet identifier than the ongoing preempted mPacket is received, it indicates that another packet is being transmitted and the preempted packet is still on hold in preemption. The reception process should be able to save more than one mPackets before sending it to the pMAC layer due to the cases where a nesting preemption occurs. The process also checks the mCRC value for the correctness of the received mPacket for the packet that is being resumed.

5.2. Timing analysis support

The new model will affect the worst-case timing analysis of traffic in TSN networks. We have developed a timing analysis based on the worst-case response time analysis in [20] where the analysis considers preemption, credit based shaper and scheduled traffic with various preemption modes, i.e., with and without hold and release mechanism. In the proposed analysis we considered the traditional blocking definition of frame preemption in TSN. Therefore, in order to provide upper bounds on the response times of frames, the existing analysis needs to be adapted to consider the impact of the non-blocking definition of preemption in TSN. In this subsection we provide a brief description of modifications for different classes of traffic, while the detailed analysis is left for the future work.

We consider four classes of traffic, including ST (scheduled traffic), A, B and best-effort (BE). Class ST has the highest priority in the system, followed by the priorities of classes A and B. Class BE is the non-real-time traffic with no timing constraints. In this model, class ST is express, whereas other traffic classes are preemptable. This means that class ST can preempt the transmission of other traffic classes. In order to compute the worst-case response time for a traffic we should identify all factors that can interfere or block the transmission of the traffic. Further, we present the overall analysis description for classes A and B separately.

For the response-time analysis of frames in class A three elements have to be considered: (i) interference from ST class as the higher priority class, (ii) blocking by the lower priority class, and (iii) interference by the same priority class due to FIFO queue. Fig. 12 shows two scenarios in which a class A frame receives interference from the three elements mentioned above. As it can be seen in Fig. 12-(a) the frame in class A can initiate while another frame from class B is already being sent on the port. Since class A is not express, it has to wait for the frame in class B to be transmitted first even if it has zero or positive credit. Therefore, class A frame can experience a blocking by the frame in class B as the lower priority blocking. Note that the class A frame can receive a similar blocking from the BE traffic. As the ST class is express, it can preempt the transmission of other traffic classes. Hence, we can see that the class A frame receives interference due to preemption by the ST frame. According to the non-blocking preemption, class A can be executed after the first ST frame has completed its transmission as long as the credit for class A is zero or positive as shown in Fig. 12-(a). We have deliberately skipped the interference that class A frame may receive from same priority frames within the same class as it would complicate the presentation of delays in Fig. 12-(a).

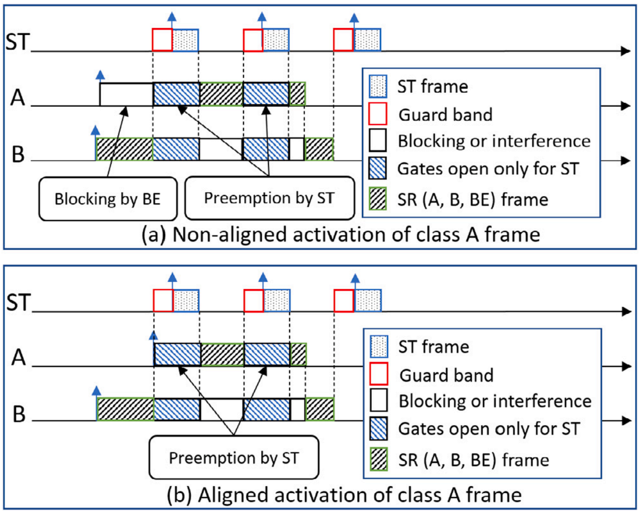


Fig. 12. Effect of non-blocking preemption on the response times of class A frames.

In the second scenario, the class A frame is activated just before opening of the gate for ST transmission, i.e., just before the start of the guard band for the ST frame. In this scenario, although the class B frame started its transmission earlier, it cannot block the class A frame. The reason is that the class A frame will be transmitted right after the ST preemption without being blocked by the class B frame thanks to the non-blocking definition of frame preemption in TSN. In fact, when the activation time of the class A frame is aligned with the ST frame activation or during the ST transmission, the class A frame will not experience any blocking by the low priority traffic. Other elements, including the ST interference and the same priority interference, still contribute to the delay for the class A frame. To sum up, in the response-time analysis for class A frame we need to determine the its activation time. If the class A frame is activated either at the beginning of the ST guard band or during the ST transmission, then the blocking is zero. Otherwise, the blocking in the worst-case is the largest frame size among the lower priority classes as shown in [20].

For the response time analysis of class B frames, we need to consider interference from three elements in a similar fashion. The elements are the same as compared to the analysis for class A frames, i.e., (i) high priority interference by class A and ST, (ii) blocking by the lower priority class, (iii) the same priority interference due to FIFO queue. The main difference between the analysis for class B and class A is the computation of higher priority interference. According to the model, class B frame is interfered by classes A and ST with different behavior. Class ST interferes preemptively, while class A interferes non-preemptively. This makes a special case, which has been extensively discussed in [20] with a 2-phase calculation solution. In the case of non-blocking preemption definition, this behavior is slightly different. The class A frame acts as a non-preemptive high priority interference until the preemption occurs by an ST frame. Reaching to the preemption point by an ST frame allows the class A frame to change the behavior to a preemptive high priority interference. Analyzing this behavior requires a deep investigation to provide a non-pessimistic solution. On the other hand, the other elements, including the same priority interference and the blocking by the lower priority class still contribute delays to the transmission of class B frame (see Fig. 13).

6. Evaluation

This section evaluates the performance of TSN networks with respect to various effects of preemption discussed in the previous section.

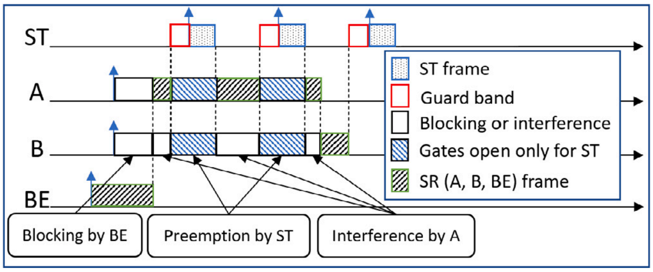


Fig. 13. Effect of non-blocking preemption on the response times of class B frames.

First, we perform simulations to evaluate the effects of disabling and enabling preemption, while considering only the blocking definition of preemption. Second, we perform simulations to evaluate the difference between the two frame-preemption models using a multi-hop TSN network architecture. Third, we perform simulations to evaluate the effects of enabling and disabling the Hold/Release mechanism together with the blocking definition as well as the non-blocking definition of preemption. Finally, we perform a comparative evaluation between the blocking and non-blocking definitions of preemption using an automotive industrial use case.

6.1. Simulation setup and assumptions

Currently, there are two main simulation tools for TSN networks, namely NeSTiNg [31] and CoRE4INET [32]. Both of these tools are based on OMNeT+simulation platform. However, the existing simulation tools do not support the non-blocking preemption and Hold/Release mechanism with combination of traffic shapers in TSN. Therefore, we develop a simulation platform with the above mentioned features. The simulation platform is written in C/C++, which makes it compatible to be easily integrated into the other simulation platforms based on OMNeT++. We consider the following configuration setup in all the simulation experiments on a single network link. The port rate (port bandwidth) is set to 100 Mbit/s. We consider four different traffic classes that include ST, A, B and BE. The ST class is set to express, while the remaining classes are set to preemptable. The credits for classes A and B are set according to the IEEE 802.1Q-2018 standard using the utilization of frames. We generate 1000 sets of frames, where each set consists of 32 frames. Hence there are 32 000 frames in each experiment. The frame periods are uniformly selected in the range [10, 20] ms with the step size of 1 ms. The size of the payload in each frame is selected uniformly in the range [500, 1500] Bytes with the step size of 500 Bytes. The distribution of frames according to the type of traffic in each frame set is as follows: 10, 10, 10, 2 for classes ST, A, B, and BE respectively. The simulation in each experiment is run for 5 seconds. Note that the experiment presented in Section 6.4 has a different configuration setup, which is discussed in the corresponding subsection.

6.2. Preemption vs. non-preemption

In order to show the effect of enabling preemption in the TSN network, we generate the frame sets according to the configuration described in Section 6.1 and measure the maximum response times of the frames using simulations. Fig. 14 shows two histograms depicting the maximum response times of all generated frames in class A in the cases where preemption in class A is disabled and enabled respectively. In the case when preemption is enabled, we consider the blocking definition of preemption. As it can be seen in Fig. 14 the histograms follow a normal distribution which is expected due to the uniform random generation of the frames. The peak of the distribution (i.e., largest frequencies of the frame response times) is more towards the left-hand side when preemption is enabled. The minimum response times observed in the case of disabling and enabling preemption are 2380 μs and 166 μs respectively. Similarly, the maximum response times observed in the case of disabling and enabling preemption are 17 705 μs and 13 405 μs respectively.

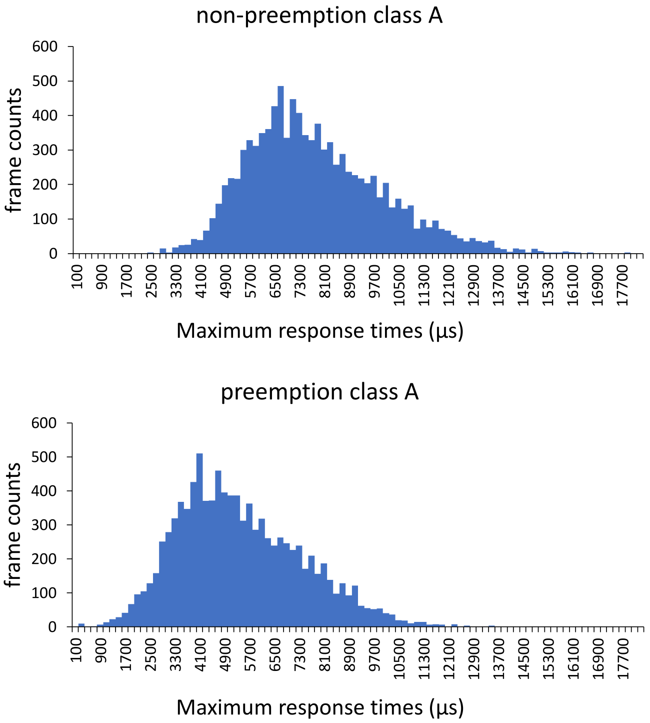


Fig. 14. Histograms of the maximum response times of the frames in class A: preemption vs non-preemption.

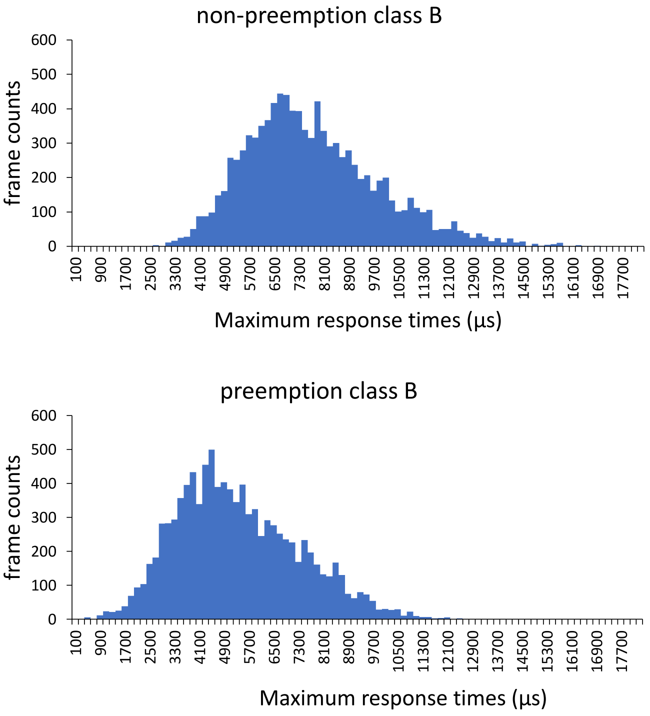


Fig. 15. Histograms of the maximum response times of the frames in class B: preemption vs non-preemption.

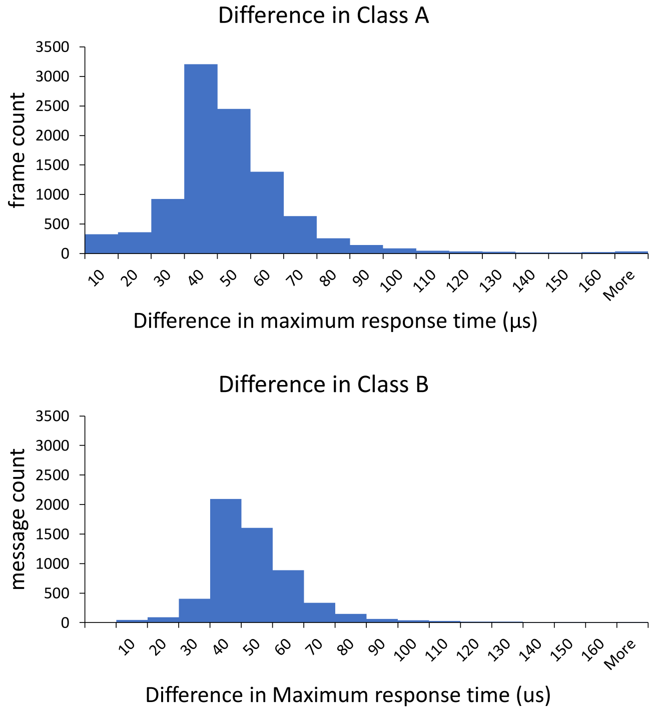


Fig. 16. Histograms of the difference between the maximum response times of the frames with the Hold/Release mechanism enabled and disabled in blocking preemption.

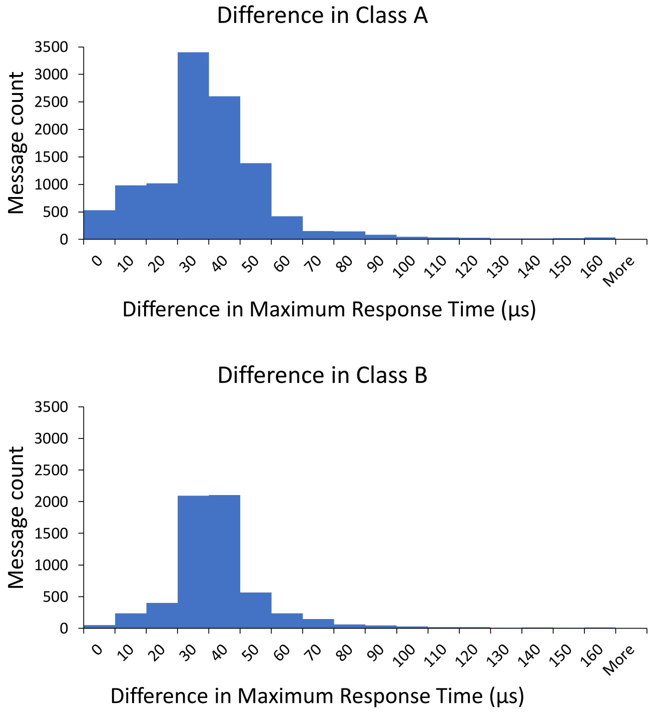


Fig. 17. Histograms of the difference between the maximum response times of the frames with the Hold/Release mechanism enabled and disabled in non-blocking preemption.

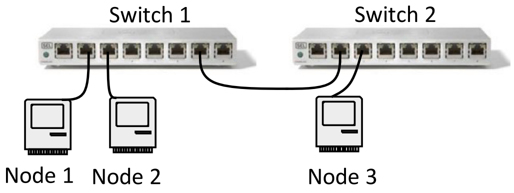


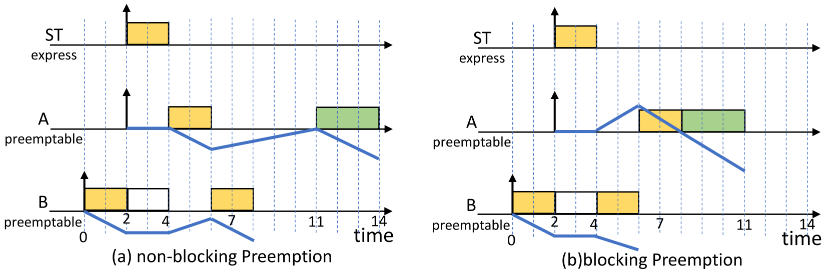
Fig. 18. Network topology used for the evaluation of blocking and non-blocking models of preemption in TSN.

Similar observation can be seen for class B frames in Fig. 15 when preemption is enabled and disabled respectively. The minimum response times of class B frames observed in the case of disabling and enabling preemption are 2562 µs and 332 µs respectively. Similarly, the maximum response times observed for class B frames in the case of disabling and enabling preemption are 16 884 µs and 12 894 µs respectively.

Based on this experiment, we can conclude that enabling preemption in a TSN network can significantly affect the maximum response times of the frames in both classes A and B. The main reason for better response times in both classes is that depending on coming ST frame, the transmission of other classes should be on hold far in advance to prevent any interference with the ST transmission. This leads to longer queuing delays for the class A and B frames. By enabling the preemption, part of the frames in classes A and B can be transmitted which can lead to shorter delays. In this experiment the number of ST frames are 10 with frequent generation that can lead to multiple preemption points, hence longer queuing occurs for the other classes. As it can be seen that the majority of the measured maximum response times in Fig. 14 in the case of non-preemption are in the range $[5700, 9700]$ µs. Whereas, majority of the measured maximum response times with preemption enabled are in the range $[3300, 7300]$ µs. Similarly, for class B shown in Fig. 15 the majority of the measured maximum response times with non-preemption are in the range $[4900, 9700]$ µs, while with preemption are in the range $[2500, 7300]$ µs.

6.3. Hold/release mechanism

In this subsection we investigate the effect of enabling and disabling the Hold/Release mechanism together with the blocking definition of preemption as well as the non-blocking definition of preemption. First, we simulate 1000 generated frame sets by enabling the Hold/ Release mechanism and measure the maximum response times of the frames. Thereafter, we simulate the same frame sets by disabling the Hold/Release mechanism and measure the corresponding maximum response times of the frames. Note that we perform the simulations in two modes of blocking and non-blocking preemption. We observe that the maximum response times of the frames with the Hold/Release mechanism enabled are always higher than the corresponding maximum response times of the frames with the Hold/Release mechanism disabled in both cases of blocking and non-blocking preemption, i.e., the difference between the two is always positive. The histograms in Fig. 16 show the difference between maximum response times of the frames with the Hold/Release mechanism enabled and disabled for class A and B using the blocking preemption. Whereas, Fig. 17 depicts the histograms showing the difference between maximum response times of the frames with the Hold/Release mechanism enabled and disabled for class A and B using the non-blocking preemption. In the case of blocking preemption, for both classes A and B, the largest difference in the maximum response times of the frames is around 160 µs, while a large majority of the difference in the response times is in the range $[40, 70]$ µs. Moreover, in the case of non-blocking preemption, the majority of differences between maximum response times of the frames with Hold/Release mechanism enabled and disabled are around 40 µs, while a large number of differences in the maximum response times of the frames are still in the range $[30, 80]$ µs. Note that the simulation results shown in Figs. 16 and 17 are for only one port. Many industrial systems comprise multiple switch architectures [30,33–35], in which case the difference between the maximum response times with the Hold/Release mechanism enabled and disabled can be significantly



(a) non-blocking Preemption

(b)blocking Preemption

Fig. 19. A corner case where the response time of a frame in class A has higher response time under non-blocking preemption. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

higher due to the frames crossing multiple hops through the TSN network.

As discussed in Section 4.6, the ST traffic is expected to experience some jitter when the Hold/Release mechanism is not used. In this experiment, we have also measured the jitter that ST frames experience during the simulation time. The minimum, average and maximum measured jitter in this experiment are 0 μs, 15.1 μs and 52 μs, respectively.

6.4. Blocking vs. non-blocking

In order to demonstrate the effects of the two models of frame preemption in TSN, we consider a TSN network with fixed topology consisting of more than three links. There are three nodes that are connected to two TSN switches as shown in Fig. 18. Each of Node 1 and Node 2 send two ST, one class A and one class B frames to Node 3. It can be seen in Fig. 18 that all of these frames traverse through three links between source and destination. These 8 frames are regarded as one frame set and we generate 500 of these frame sets. Hence, the total frame count in the system is 4000, out of which there are 1000 frames for each class A and B. The ST class is express, while other classes are preemptable. The credits for classes A and B are set according to the IEEE 802.1Q-2018 standard using the utilization of frames. The frame periods are uniformly selected within the range $[2.5, 5]$ ms with the steps of 0.5 ms for ST class. Similarly, the frame periods for classes A and B are uniformly selected within the range and $[10, 15]$ ms with the steps of 1 ms. Each ST frame is initiated with an offset, that is randomly selected from the range $[10, 100]$ μs with the step of 10 μs. The size of payload for all frames is 1500 Bytes. The port rate is set to 100 Mbit/s and each simulation is run for 5 s.

First, we simulate each set of frames using the blocking definition of preemption. Thereafter, we simulate the same set of frames with the non-blocking definition of preemption. At the end of the simulations, we measure the maximum response time of the frames. We calculate the normalized difference between the maximum response times of the frames under the two different definitions of preemption to magnify the difference using the following equation:

$$NormalizedDif = (RT_B - RT_{NB})/RT_B \times 100 \tag{11}$$

where, RT_B and RT_{NB} are the maximum measured response times of a frame under the blocking and non-blocking definitions, respectively.

The normalized difference between the maximum response times of the class A frames under the two definitions of preemption is shown in Fig. 20. In this figure, the horizontal axis (x-axis) shows the normalized difference in percentage, while the vertical axis (y-axis) shows the number of frames in the simulations. As it can be seen in Fig. 20, just above 350 frames out of 1000 did not show any improvement in their maximum response times as the normalized difference between the maximum measured response time of frames in blocking and non-blocking preemption is 0. However, we can see a significant number of frames have a level of improvement in the range of [2, 20]% and around 60 frames with 92% normalized difference in the maximum response time. On the other hand, we can see several frames which have negative normalized difference, i.e., their maximum response times have increased under the non-blocking definition of preemption (see Eq. (11)). We believe that the increase in the maximum response times of class A frames under the non-blocking preemption can occur in a corner case that we observe in this evaluation. We explain the corner case using an example in Fig. 19. Basically, this case can occur due to a situation where two frames in class A are ready for transmission at the same time in the FIFO queue. Looking at the scenario in Fig. 19-(a), consider that both frames in class A, represented by the yellow and green boxes on the second timeline in Fig. 19-(a), are ready for transmission at time 2 while a frame in class B is already under transmission. A frame in class ST preempts the transmission of the class B frame between time 2 and 4. The ST frame completes its transmission at time 4. At this time, the first (yellow) frame in the class A queue will be sent under the non-blocking preemption. After the transmission of this frame is completed at time 6, the preempted frame in class B is resumed as the credit for class A is negative. After replenishing the credit in class A, the second (green) frame can be transmitted. The same example in Fig. 19-(b) shows the scenario under the blocking preemption. In this case, the frame in class B resumes its transmission after the ST frame transmission is completed. During this time the credit of class A frame increases such that it gains credit that is enough to transmit both frames in class A. Thanks to the increase in the credit of class A, the second (green) frame can be transmitted right after the transmission of the previous (yellow) frame. This allows the frame under analysis (green frame) in class A to be transmitted faster compared to the case under the non-blocking preemption. Note that this case can occur only if both frames are ready in the FIFO queue and the credit of class A can increase (during the transmission of the low priority frame) to a value that is enough to transmit both frames in the class A queue without any further delay.

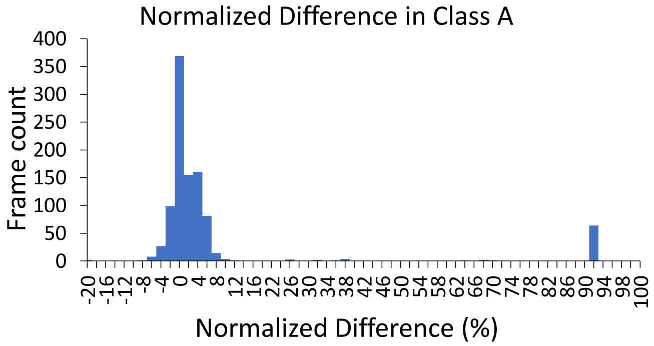


Fig. 20. Histograms of the difference between the maximum response times of the class A frames with blocking and non-blocking preemption.

The normalized difference for class B is illustrated in Fig. 21. As it can be seen in the figure, the normalized difference between the maximum response times of 901 frames out of 1000 is 0%. This means that

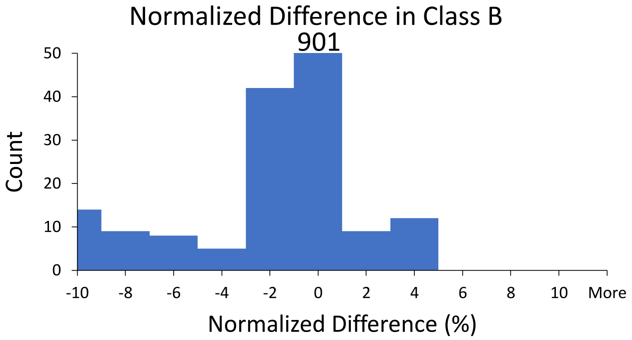


Fig. 21. Histograms of the difference between the maximum response times of the class B frames with blocking and non-blocking preemption.

there is no change in the maximum response times of the frames under the two different definitions of preemption. We also see that several frames are affected negatively by using the non-blocking definition. This is an expected result because as we also show in Section 4 the improvement in the response time of class A frames comes at the cost of impact on the response time of class B frames. As it can be seen in Fig. 1 and Fig. 2 when applying the non-blocking preemption, the lower priority frame might stay in the MAC buffer for longer time compared to the standard blocking definition of frame preemption. This can lead to larger response times for the lower priority frames in several cases. In this experiment, the generated frames are random, hence the effects of negative differences come from this fact that the frames in class B suffered by staying longer in the queues in favor of the frames in class A. To summarize, based on this experiment we can conclude that the non-blocking definition of preemption in TSN can bring significant improvement for class A traffic, specially if the system is designed carefully, and at the same time no large impact will appear on class B traffic.

6.5. Automotive application use case

In order to show the effect of the blocking and non-blocking definitions of preemption in TSN, we consider a use case inspired by an industrial prototype system. The prototype system, developed by the BWM group [33], has been used for validating the results in several existing works [30,34,35], among others. The network topology in the use case is depicted in Fig. 22. The use case consists of three TSN switches, where Switch 1 and Switch 3 are mounted in the front and back of the vehicle respectively. We consider two cameras (CAM1 and CAM2) for driver assistance that send video frames to the main computer that is denoted by Head Unit (HU). Three control nodes represented by (Control 1–3) send control signals to the Head Unit. The Bulk node facilitates the support for the vehicle's communication to the internet. Moreover, we consider an A/V node that manages the vehicle's audio/video infotainment system. The main purpose of the use case demonstration is to show the worst-case scenario in which a frame can miss its deadline if we use blocking preemption, while it meets its deadline by using the non-blocking preemption.

There are 7 frames in the system: 3 belong to ST class, 2 belong to class A and the remaining 2 belong to class B. All frames are assumed to carry the maximum payload permitted by Ethernet, i.e., 1500 Bytes of payload. The network speed is set to 100 Mbit/s. Hence, the transmission time of each frame is 123 μs according to Eq. (2). Various properties of each frame in the use case, including the multi-hop path through which the frame traverses from its source to its destination, are presented in Table 1. In this example we assume that the frame under analysis has a period of 10 ms, while the periods of other frames in classes A and B are set to 20 ms. Moreover, each ST frame is transmitted with a period of 9 ms. In order to create the worst-case scenario for the frame under analysis, two of the ST frames are scheduled with offsets of 492 μs each as shown in Table 1. It should be noted that the specified values in Table 1 are selected such that the frame under analysis experiences worst-case interference from the other frames. Note that we assume implicit deadlines for the frames, i.e., the deadline of each frame is equal to its period. The ST class is set to "Express", whereas both classes A and B are set to "Preemptable". Also, the Hold/Release mechanisms is enabled. Given the schedule of the ST class is available, we investigate the effects of different definitions of preemption on class A and B and demonstrate them in Fig. 23. The schedule in Fig. 23 is provided for up to 10 ms and each traffic class is identified with a different color for clarity. All frames are assumed to be ready for transmission at time 0, except for the class B frames (Bulk and A/V) that are activated sporadically. For the sake of simplicity, we neglect the overhead of adding an extra header when resuming the transmission of a preempted frame.

Table 1
Frame properties considered in the use case.

ID	Multi-hop path (source to destination)	Period	Class	Offset
CAM	CAM2→SW3→SW2→SW1→HU	10 ms	A	NA
1	Control2→SW3→SW2→SW1→HU	9 ms	ST	0
2	Control3→SW2→SW1→HU	9 ms	ST	492 μs
3	CAM1→SW2→SW1→HU	20 ms	A	NA
4	Bulk→SW2→SW1→A/V	20 ms	B	NA
5	Control1→SW1→HU	9 ms	ST	492 μs
6	A/V→SW1→HU	20 ms	B	NA

The frame of interest in this use case is CAM that belongs to class A. Note that the transmission pattern of CAM along its subpath (CAM2→SW3→SW2) is the same in the two cases of the blocking and non-blocking definitions of preemption because no preemption occurs along this subpath. Hence, we present only one timeline for the two cases in Fig. 23. On the other hand, the CAM frame has different transmission patterns along its remaining subpath (SW2→SW1→HU) in the two cases due to the occurrence of preemption. Therefore, we present separate timelines for the two cases in Fig. 23.

According to the presented schedule, the CAM frame is sent at time 0 from CAM2 node to Switch SW3. The frame arrives at SW3 at time 123 μs. As the ST frame 1 is scheduled for transmission at time 123 μs in the output port of SW3, the CAM frame starts its transmission from SW3 after transmission of frame 1 is completed at time 246 μs. CAM is received at SW2 at time 369 μs. However, CAM cannot be immediately transmitted from SW2 to SW1 because frame 3 belonging to class A and destined for SW1 was already in the queue at SW2. The frames in a queue that belong to the same class are transmitted using the first-in-first-out policy. Since both CAM and frame 3 belong to class A, the credit of class A becomes negative after transmission of frame 3 from SW2. Therefore, the CAM frame must wait for the credit replenishment before it can be transmitted from SW2 to SW1. For the sake of simplicity, we only show the credit of class A on links SW2→SW1 and SW1→HU. The idleSlope for class A on the two links is set to 1.37 bits/μs. As the sendSlope is 1.37 - 100 = −98.63 bits/μs and the frame size is 123 μs, the credit becomes −98.63 × 123 = −12131.49 bits at time 246 μs. The credit for class A on link SW2→SW1 becomes zero at 9347 μs considering the idleSlope of 1.37bits/μs and given that it is constant during the transmission of frames 1 and 2. However, the sporadic frame 4 belonging to class B starts its transmission from SW2 to SW1 at 9236 μs and is preempted by frame 1 after 10 μs from the start of its transmission. Note that preemption of a frame after 10 μs from the start of its transmission is allowed. As discussed in Section 4, a frame with less than or equal to 123 Bytes (or 9.84 μs of transmission time) cannot be preempted, according to Annex R.2 in the IEEE 802.1Q-2018 standard. The transmission of frame 1 from SW2 to SW1 is completed at time 9369 μs. From this point onward, the blocking and non-blocking preemptions have different communication patterns as shown in Fig. 23 by separate timelines for SW2→SW1 and SW1→HU.

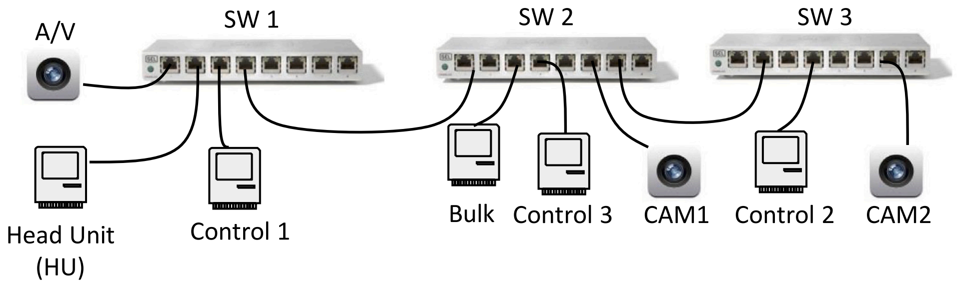


Fig. 22. Network topology in the automotive use case.

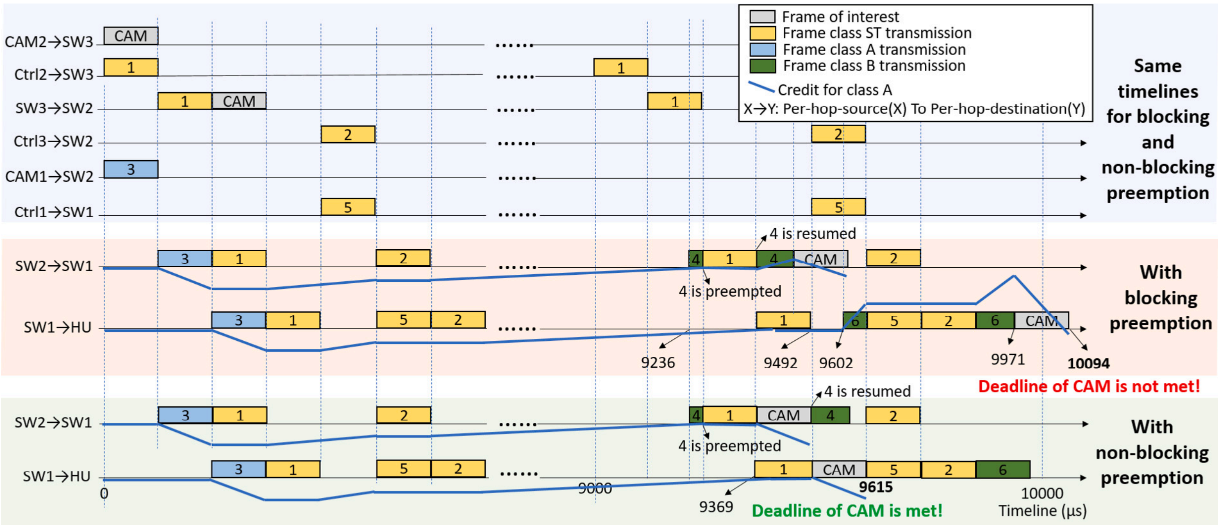


Fig. 23. Schedule of the frame set in the use case considering the blocking and non-blocking definitions of preemptions separately. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In the case of the blocking definition of preemption, the preempted frame 4 (belonging to class B) resumes its transmission at time 9369 μs instead of starting the transmission of the CAM frame despite having a non-negative credit for class A. This is due to the blocking definition of preemption. The CAM frame starts its transmission from SW2 at time 9482 μs and is received at SW1 at time 9605 μs. Note that frame 2 has a period of 9000 μs with an offset of 492 μs. This means that frame 2 starts at time 9492 μs from its node, hence it will not preempt CAM frame. Upon arrival at SW1, the CAM frame cannot start its transmission towards HU because frame 6 from A/V node already started its transmission slightly before the arrival of CAM at 9602 μs. Since, class A is not set to Express, the CAM frame cannot preempt frame 6. However, frame 6 is preempted by the ST frames 5 and 2 as they are transmitted from SW1 at time 9615 μs and 9738 μs, respectively. After frame 6 completes its transmission at 9971 μs, the CAM frame is transmitted and is completely received at the destination node HU at time 10 094 μs. Hence, with the blocking definition of preemption, the CAM frame misses its deadline of 10 ms.

In the case of the non-blocking definition of preemption, also shown in Fig. 23, CAM is transmitted from SW2 to SW1 at time 9369 μs after the transmission of ST frame 1 is completed. Upon arrival at SW1, CAM is immediately transmitted and is completely received at the destination node HU at time 9615 μs. Therefore, with the non-blocking definition of preemption, the CAM frame meets its deadline of 10 ms.

7. Conclusions and future work

In this paper we have identified a limitation in the frame preemption model supported by the TSN standard IEEE 802.1Q-2018 and used by the existing works. This model incurs an extra blocking time for higher priority frames resulting in their larger worst-case response times. This blocking time can become significant in the case of multi-switch TSN networks. We have showed that this limitation can have a considerable impact on the design, analysis and performance of TSN-based systems. We have proposed a novel and more efficient model of preemption conforming to the standard, which allows over 90% reduction in the maximum blocking delay leading to lower worst-case response times for higher priority frames in comparison to the frame preemption model used in the existing works. We have evaluated the effects of preemption by performing several experimental simulations, while enabling and disabling preemptions and the Hold/Release mechanism supported by TSN. Finally, a comparative evaluation of the two models of preemption in TSN is performed using simulations as well as by means of an automotive industrial use case of a distributed embedded system that uses a multi-hop TSN network for real-time network communication. The use case demonstrates that a frame can miss its deadline due to extra amount of blocking that is inherent to the classical model of frame preemption. Whereas, the same frame meets its deadline when the proposed model of frame preemption is used due to significantly lower amount of blocking delay experienced by the higher priority frames.

As the future work we are currently aiming at various directions. Firstly, we are planning to setup hardware experiments based on two to three different vendor TSN switches and investigate different implementations for the switches and their impact. In addition, we plan to investigate more complicated combination of TSN features, e.g., setting low priority traffic to express mode while high priority traffic is in preemptable mode and see the effects of different models of frame preemption. As a continuation of this work, we are working on developing a schedulability analysis for the non-blocking preemption to be able to analytically compare the two models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work in this paper is supported by the Swedish Government Agency for Innovation Systems (VINNOVA) via the DESTINE and PROVIDENT projects, and the Swedish Knowledge Foundation (KKS) via the FIESTA, HERO and DPAC projects.

References

[1] IEEE 802.1 Time-Sensitive Networking (TSN) Task Group, https://1.ieee802.org/tsn.

[2] IEEE AVB Task Group, http://www.ieee802.org/1/pages/avbridges.html.

[3] D. Maxim, Y.-Q. Song, Delay analysis of AVB traffic in time-sensitive networks (TSN), in: Proceedings of the 25th International Conference on Real-Time Networks and Systems, RTNS, 2017, http://dx.doi.org/10.1145/3139258.3139283.

[4] L. Zhao, P. Pop, Z. Zheng, Q. Li, Timing analysis of AVB traffic in TSN networks using network calculus, in: Real-Time and Embedded Technology and Applications Symposium, 2018, pp. 25–36.

[5] D. Thiele, R. Ernst, Formal worst-case timing analysis of Ethernet TSN's burst-limiting shaper, in: Design, Automation Test in Europe Conference Exhibition, DATE, 2016, pp. 187–192.

[6] P. Pop, M.L. Raagaard, S.S. Craciunas, W. Steiner, Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks, IET Cyber-Phys. Syst. Theory Appl. 1 (1) (2016) 86–94, http://dx.doi.org/10.1049/iet-cps.2016.0021.

[7] S. Brunner, J. Roder, M. Kucera, T. Waas, Automotive E/E-architecture enhancements by usage of Ethernet TSN, in: 13th Workshop on Intelligent Solutions in Embedded Systems, 2017.

[8] M.H. Farzaneh, A. Knoll, Time-sensitive networking (TSN): An experimental setup, in: IEEE Vehicular Networking Conference, 2017.

[9] L. Lo Bello, W. Steiner, A perspective on IEEE time-sensitive networking for industrial communication and automation systems, Proc. IEEE (2019).

[10] L. Lo Bello, R. Mariani, S. Mubeen, S. Saponara, Recent advances and trends in on-board embedded and networked automotive systems, IEEE Trans. Ind. Inf. (2019).

[11] IEEE, IEEE standard for local and metropolitan area network–bridges and bridged networks, IEEE Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014), 2018, pp. 1–1993, http://dx.doi.org/10.1109/IEEESTD.2018.8403927.

[12] IEEE 802.3br-2016 - IEEE Standard for Ethernet Amendment 5: Specification and management parameters for interspersing express traffic. Amendment to IEEE Std 802.3-2015, October, 2016, Available: https://standards.ieee.org/standard/802_3br-2016.html.

[13] G. Alderisi, G. Patti, L. Lo Bello, Introducing support for scheduled traffic over IEEE audio video bridging networks, in: Conference on Emerging Technologies Factory Automation, 2013.

[14] S. Kehrer, O. Kleineberg, D. Heffernan, A comparison of fault-tolerance concepts for IEEE 802.1 Time Sensitive Networks (TSN), in: Conference on Emerging Technology and Factory Automation, 2014.

[15] F.A.R. Arif, T.S. Atia, Load balancing routing in Time-Sensitive Networks, in: 3rd International Scientific-Practical Conference Problems of Infocommunications Science and Technology, 2016.

[16] K.S. Umadevi, R.K. Sridharan, Multilevel ingress scheduling policy for Time Sensitive Networks, in: International Conference on Microelectronic Devices, Circuits and Systems, 2017.

[17] S. Mubeen, M. Ashjaei, M. Sjödin, Holistic modeling of time sensitive networking in component-based vehicular embedded systems, in: 45th Euromicro Conference on Software Engineering and Advanced Applications, SEAA, 2019.

[18] D. Thiele, R. Ernst, J. Diemer, Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers, in: Vehicular Networking Conference, 2015.

[19] D. Thiele, R. Ernst, Formal worst-case performance analysis of time-sensitive Ethernet with frame preemption, in: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation, ETFA, 2016, pp. 1–9.

[20] L. Lo Bello, M. Ashjaei, G. Patti, M. Behnam, Schedulability analysis of time-sensitive networks with scheduled traffic and preemption support, J. Parallel Distrib. Comput. 144 (2020) 153–171, http://dx.doi.org/10.1016/j.jpdc.2020.06.001.

[21] H. Lee, J. Lee, C. Park, S. Park, Time-aware preemption to enhance the performance of Audio/Video Bridging (AVB) in IEEE 802.1 TSN, in: 2016 First IEEE International Conference on Computer Communication and the Internet, 2016, pp. 80–84.

[22] J. Zhang, L. Chen, T. Wang, X. Wang, Analysis of TSN for industrial automation based on network calculus, in: 24th IEEE International Conference on Emerging Technologies and Factory Automation, 2019.

[23] D. Hellmanns, J. Falk, A. Glavackij, R. Hummen, S. Kehrer, F. Dürr, On the performance of stream-based, class-based time-aware shaping and frame preemption in TSN, in: 2020 IEEE International Conference on Industrial Technology, 2020, pp. 298–303.

[24] J. Cao, M. Ashjaei, P.J. Cuijpers, R.J. Bril, J. Lukkien, An independent yet efficient analysis of bandwidth reservation for credit-based shaping, in: International Workshop on Factory Communication Systems, 2018.

[25] E. Mohammadpour, E. Stai, M. Mohiuddin, J. Le Boudec, Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping, in: 30th International Teletraffic Congress, ITC 30, Vol. 02, 2018, pp. 1–6.

[26] M. Boyer, H. Daigmorte, Impact on credit freeze before gate closing in CBS and GCL integration into TSN, in: 27th International Conference on Real-Time Networks and Systems, 2019, pp. 80–89.

[27] M.A. Ojewale, P.M. Yomsi, B. Nikolić, Multi-level preemption in TSN: Feasibility and requirements analysis, in: 2020 IEEE 23rd International Symposium on Real-Time Distributed Computing, 2020.

[28] M. Ashjaei, G. Patti, M. Behnam, T. Nolte, G. Alderisi, L. Lo Bello, Schedulability analysis of ethernet audio video bridging networks with scheduled traffic support, Real-Time Syst. 53 (2017).

[29] U.D. Bordoloi, A. Aminifar, P. Eles, Z. Peng, Schedulability analysis of Ethernet AVB switches, in: 20th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2014.

[30] P. Axer, D. Thiele, R. Ernst, Formal timing analysis of automatic repeat request for switched real-time networks, in: Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems, 2014.

[31] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, K. Rothermel, NeSTiNg: Simulating IEEE time-sensitive networking (TSN) in OMNeT++, 2019, http://dx.doi.org/10.1109/NetSys.2019.8854500.

[32] T. Steinbach, H.D. Kenfack, F. Korf, T.C. Schmidt, An extension of the OMNeT++ INET framework for simulating real-time ethernet with high accuracy, in: Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 2011, pp. 375–382.

[33] H.-T. Lim, K. Weckemann, D. Herrscher, Performance study of an in-car switched ethernet network without prioritization, in: Communication Technologies for Vehicles, 2011.

[34] G. Alderisi, A. Caltabiano, G. Vasta, G. Iannizzotto, T. Steinbach, L.L. Bello, Simulative assessments of IEEE 802.1 Ethernet AVB and time-triggered ethernet for advanced driver assistance systems and in-car infotainment, in: IEEE Vehicular Networking Conference, 2012.

[35] F. Reimann, S. Graf, F. Streit, M. Glaß, J. Teich, Timing analysis of Ethernet AVB-based automotive E/E architectures, in: 18th IEEE Conference on Emerging Technologies Factory Automation, 2013.

Ph.D. Mohammad Ashjaei is a senior lecturer in the Complex Real-Time Systems (CORE) and the Heterogeneous Systems - Hardware Software Co-design (HERO) research groups at Mälardalen University in Sweden. Mohammad has received his Ph.D. degree in Computer Science in November 2016 from Mälardalen University. His main research interests include real-time systems, real-time distributed systems, scheduling algorithms on networks and processors, schedulability analysis techniques, resource reservation and reconfiguration mechanisms for real-time networks. He is also giving lectures on various topics related to embedded systems and data communication networks.

Prof. Mikael Sjödin is since 2006 a professor of real-time system at Mälardalen University, Sweden. He is leading the research group Model-Based Engineering of Embedded Systems that focus on development of methods and tools for model-based engineering of embedded systems. Including: models for architectural and behavioral descriptions of system and requirements for systems, techniques for analyzing and transforming models, and runtime architectures for resource efficient, predictable embedded systems. Since 2012 he is the research director for Embedded Systems, a research environment with 200 active researchers at Mälardalen University. Between 2000 and 2006 Mikael held numerous

industrial positions and worked as software architect and project manager. Mikael received his PhD and MSc in computer science from Uppsala University in 2000 and 1995 respectivly.

Ph.D. Saad Mubeen is an Associate Professor at Mälardalen University, Sweden. He has previously worked in the vehicle industry as a Senior Software Engineer at Arcticus Systems and as a Consultant for Volvo Construction Equipment, Sweden. He is a Senior Member of IEEE and a Co-chair of the Subcommittee on In-vehicle Embedded Systems within the IEEE IES Technical Committee on Factory Automation. His research focus is on model- and component-based development of predictable embedded software, modeling and timing analysis of in-vehicle communication, and end-to-end timing analysis of distributed embedded systems. Within this context, he has co-authored over 135 publications in peer-reviewed international journals, conferences and workshops. He has received several awards, including the IEEE Software Best Paper Award in 2017. He is a PC member and referee for several international conferences and journals respectively. He is a guest editor of IEEE Transactions on Industrial Informatics (TII), Elsevier's Journal of Systems Architecture and Microprocessors and Microsystems, ACM SIGBED Review, and Springer's Computing journal. He has organized and chaired several special sessions and workshops at the international conferences such as IEEE's IECON, ICIT and ETFA. For more information see http://www.es.mdh.se/staff/280-Saad_Mubeen.