Size-Based Queuing: An Approach to Improve Bandwidth Utilization in TSN Networks

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

ABSTRACT

Ethernet has become one of the most researched network technologies for modern, safety-critical, distributed systems due to its maturity and low cost. Standard Ethernet, however, is unsuitable for safety-critical applications due to its lack of delivery guarantees and lack of delay bounds. The Time-Sensitive Networking (TSN) task group proposes several extensions to Standard Ethernet to address these shortcomings. One of the proposed extensions is the Time-Aware Shaper (TAS, IEEE 802.1Qbv-2015) which introduces a gate mechanism that enables scheduled transmissions and minimal jitter for high-priority messages. In order to achieve low jitter for these messages, guard bands are inserted into the schedule of the other traffic classes to ensure that the outgoing link is idle when the transmission of the high-priority message is scheduled to occur. This can result in wasted bandwidth.

In this work, we propose Size-Based Queuing (SBQ) — an approach to improve bandwidth utilization in TSN networks that utilize TAS and guard bands. After a detailed description of SBQ, we provide a method to schedule an SBQ-enabled TSN network and conduct a qualitative analysis of the impact of SBQ on the transmission time of messages and the improvements on bandwidth utilization. We show that SBQ can reduce the amount of wasted bandwidth in a network, without impacting the transmission of high-priority messages or requiring extensive changes of the TSN network.

CCS CONCEPTS

• **Networks → Packet classification**; **Network experimentation**; *Bridges and switches*;

KEYWORDS

Time-Sensitive Networking, Time-Aware Shaper, Guard-Bands, Priority-Queuing, Bandwidth Utilization

1 INTRODUCTION

In modern distributed systems, existing networks struggle to meet the increasing demand in network bandwidth introduced by new applications such as image recognition in industrial plants or Advanced Driver Assistance Systems (ADAS) in the automotive domain. As a result, Ethernet is now at the center of attention, due to its high bandwidth, maturity and low cost. Standard Ethernet is, however, unsuitable for safety-critical applications, due to its lack of delivery guarantees and lack of delay bounds. To address these issues, several extensions to Standard Ethernet have been proposed. Aside from Avionics Full-DupleX Switched Ethernet (ARINC 664 p7 [1]) and Time-Triggered Ethernet (AS6802 [12]), Time-Sensitive Networking (TSN [8]) is one such extension. TSN is a continuation of the Audio-Video Bridging amendments to IEEE802.1Q to further improve the accomplishments made by AVB and apply them to new domains such as the industrial domain.

One of the amendments is the Time-Aware Shaper (TAS, described in Section 4). TAS enables scheduled message transmission with low jitter by adding gates controlled by a gate schedule to the FIFO queues of the output ports of TSN network switches. A transmission selection algorithm selects messages from the queues for transmission in FIFO order, based on the priority of the messages and the gate status of the respective queue. If the gate of a queue is open, a pending message inside this queue may be selected by the transmission selection algorithm and, thus, egress through the output port. If the gate is closed, a pending message can't egress even if the outgoing network link is idle. If a message is not preempted during egress once the gate of its respective queue closes, higher-priority messages may experience blocking caused by lower priority messages.

In this work, we use the terms TAS message and TAS queue for high-priority, scheduled messages and their respective high-priority queues. In order to enable scheduled transmissions with minimal jitter for TAS messages, the gates of the non-TAS queues close a certain amount of time before the gate of the TAS queue opens. The resulting time intervals, where all gates are closed, are called "guard bands". Because the presence of guard bands may lead to time intervals where no message can egress, even though messages are ready inside their respective queues, the available network bandwidth during these time intervals is wasted.

In order to reduce the amount of wasted bandwidth, existing guard bands in the gate schedule have to be shrunk, or the amount of guard bands in the schedule has to be reduced. However, since guard bands are necessary to ensure low jitter of TAS message transmissions, arbitrarily removing or shrinking the guard bands increases the impact of non-TAS messages on TAS message transmissions. In order to maintain the guarantees for TAS messages and reduce the amount of wasted bandwidth in a TSN network, new approaches are required.

In this work, we propose a new approach to reduce the amount of wasted bandwidth called Size-Based-Queuing (SBQ). Under SBQ, we assign the messages entering a TSN network switch to the priority queues of the output port not only according to their priority, but also according to their size. This is achieved by adding additional queues to separate the incoming messages. We modify the gate

RTN'18, July 2018, Barcelona, Spain

schedule to accommodate the increased number of queues, and to provide sufficient output bandwidth to each of them. Since SBQ allows to control the maximum size of messages inside the queues, the guard bands in the gate schedule can be shrunk accordingly, resulting in an increase of output link bandwidth utilization without impacting the TAS transmissions.

The remainder of this paper is structured as follows: Section 2 discusses related work while Section 3 presents our system model. Section 4 introduces the Time-Aware Shaper and Section 5 describes the problem we address. Section 6 describes the SBQ approach in detail. We provide a qualitative analysis of SBQ in Section 7 before concluding this work in Section 8.

2 RELATED WORK

In this Section we discuss related works that target Ethernet, TSN and TAS.

The timing aspects of the Time-Aware Shaper [3] have been analyzed by Thiele et al. in [16] and Thangamuthu et al. in [15]. The former work uses the compositional performance analysis framework (CPA) while the latter simulates a particular automotive setup to benchmark TAS against other TSN shapers. Thiele et al. acknowledge the effect of guard bands on the timing analysis of non-TAS traffic, but don't discuss approaches to mitigate this effect. Thangamuthu et al. omit the topic of guard bands in the context of TAS entirely.

Scheduling of TSN networks with TAS is performed by Craciunas et al. in [5], Duerr et al. in [6] and Hisano et al. in [11]. Craciunas et al. describe a scheduler for TAS which is capable of scheduling TSN networks with an arbitrary number of TAS and non-TAS queues. Minimizing the amount of wasted bandwidth due to guard bands is not considered. Duerr et al. propose a schedule compression algorithm in [6] to group TAS transmissions together to reduce the amount of gate-opening events for the TAS traffic queue, and thus the amount of required guard bands. Their approach differs from ours, as their aim is reduce the number of guard bands, while our approach focuses on shrinking the size of existing guard bands. While the work of Hisano et al. is also aimed at reducing the amount of wasted bandwidth in TAS-enabled TSN fronthaul networks, they focus on the bandwidth lost due to underutilized gate-open intervals of the TAS traffic class, whereas our work focuses on the bandwidth lost due to guard bands.

Alternative queuing strategies for Ethernet have been proposed, for example, by Zhang et al. and Shreedar et al. [13, 17] (Deficit Round Robin) and Goerges et al. [7] (Weighted Round Robin, Fair Queuing), however, these approaches predate TSN and TAS and, thus, don't target these new technologies.

Specht et al. explore the possibility of adding additional queues and sub-shapers in front of the output priority queues in [10, 14] in order to perform additional shaping of incoming data streams. While this approach bears similarity to ours because it takes additional metrics of incoming messages into account, it is implemented as an additional stage before the priority queues and TAS and thus does neither consider guard bands and their implications nor do the authors offer mitigation strategies for wasted bandwidth.

The preemption mechanisms of TSN as defined in IEEE 802.1-Qbu [2] can also be used to minimize bandwidth wastage due to guard bands. Ongoing transmissions of non-TAS messages are preempted in favor of TAS traffic. This, however, requires all involved devices to support frame preemption, whereas our approach allows devices that support SBQ to be mixed with legacy TSN devices. In this work, we assume a non-preemptive TSN network.

In summary, the bandwidth wastage caused by the presence of guard bands in TAS schedules has not been extensively studied. Duerr et al. [6] aim to reduce the amount of guard bands, but other related works merely acknowledge the existence of guard bands or ignore them altogether. To our knowledge, this work is the first to analyze the possibility of shrinking existing guard bands to reclaim wasted bandwidth in a non-preemptive network.

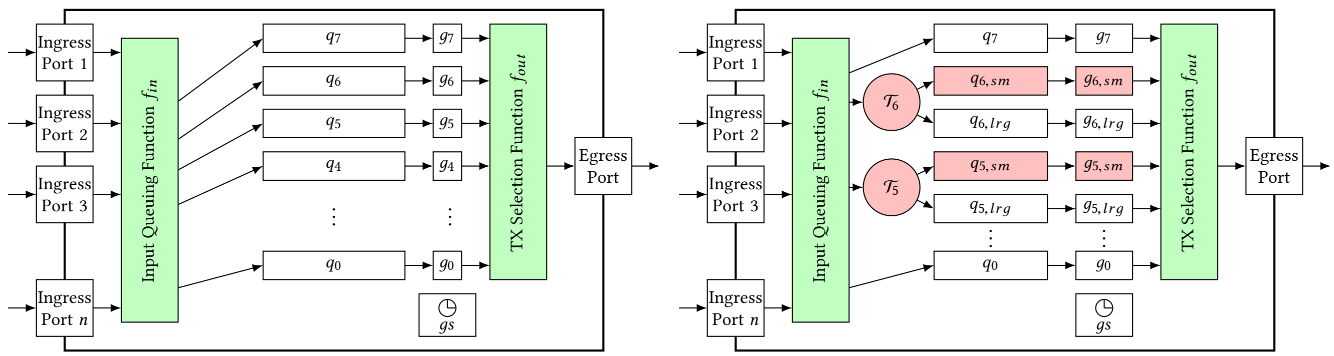
3 SYSTEM MODEL

In this work, we focus on a TSN-enabled network where devices synchronize their local clocks (e.g. via IEEE802.1AS-rev [9]) to establish a global timebase and enable cooperative scheduled transmissions in the network. The network consists of a set of nodes $\mathcal{N}$ connected by bi-directional Ethernet links $\mathcal{L}$. The set of nodes contains endpoint nodes $\mathcal{ES} \subset \mathcal{N}$, responsible for the production and consumption of all network messages $\mathcal{M}$, and switches $\mathcal{SW} \subset \mathcal{N}$ responsible for forwarding messages from their source to their destination. We focus on the switches $\mathcal{SW}$ of the network and consider two classes of messages: TAS messages, which have the highest priority and require scheduled transmissions with low jitter, and non-TAS messages carrying medium and low priority messages with slightly relaxed timing requirements (e.g. AVB traffic) and no timing requirements (e.g. Best Effort), respectively.

For the sake of simplicity, and without loss of generality, we assume unicast transmissions, as multicast or broadcast messages can be easily modeled using multiple copies of a message. Each message $m_\star \in \mathcal{M}$ has a size $64\text{Bytes} \leq len(m_\star) \leq 1522\text{Bytes}$ which remains constant during the message life-cycle.

In this work, we use the term *input* for any possible source of messages which egress through port p. Messages can arrive via the input in any particular order, even simultaneously. SBQ does not depend on the exact arrival sequence of messages, or whether messages ingress the switch on the same port or from different ports, so we omit this information. Once a message $m_\star$ ingresses a switch via any of the switch ports, it is immediately processed by the input queuing function $f_{in,p}$ and stored in one of the queues $q_{i,p} \in \mathcal{Q}_p$ of the designated egress port p where i denotes the priority of the queue and the message. The current backlog of queue $q_{i,p}$ is expressed by $bl_{i,p}(t)$. To describe the position of $m_\star$ in the queue, we define $pre_{m_\star}(t)$ as the set of messages that are in front of $m_\star$ in $q_{i,p}$ at time t. If $pre_{m_\star}(t) = \emptyset$, $m_\star$ is at the front of $q_{i,p}$ at time t.

The network uses TAS, so each queue $q_{i,p}$ has an associated gate $g_{i,p} \in \mathcal{G}_p$ which is controlled by a gate schedule gs_p. If $g_{i,p}$ is open as per gs_p and the network becomes idle, messages in $q_{i,p}$ may be selected for egress by the transmission selection function $f_{out,p}$. If $g_{i,p}$ is closed, messages in $q_{i,p}$ can't be selected by $f_{out,p}$. We define the term *output* as the egress port and the respective physical link which connects the egress port to the next node on the message path. Only one message can egress through the egress port at a time. Messages are transmitted non-preemptively.



(a) TSN Switch without SBQ

(b) TSN Switch with SBQ applied to priorities 6 and 5

Figure 1: Simplified overview of a TSN switch with and without SBQ

4 TIME-AWARE SHAPER

In this Section, we describe the Time-Aware Shaper (TAS [3]) and provide definitions to describe the traversal time of a message through a TSN switch. For the sake of clarity, the port index p has been omitted in this chapter and associated figures. Figure 1a) depicts a simplified overview of the components of a TSN switch responsible for routing TAS and non-TAS messages from the ingress ports to their shared egress port. The input queue function f_{in} assigns each message incoming $m_\star$ to one of the available queues q_i of the egress port according to the message priority i. This priority is determined by reading the *Priority Code Point* field from the 802.1Q header of the message. As this field is three bits long, up to eight priorities can be used for each port of a switch. Each queue q_i has an associated gate g_i controlled by the gate schedule gs to control whether a message $m_\star$ can be selected by f_{out} to egress the switch via the egress port. In Figures 1a) and 1b), q_7 is the TAS message queue, while the other queues are used for non-TAS messages.

Remember that guard bands in the gate schedules of the non-TAS queues ensure that a pending TAS message can immediately start its egress once gate g_7 opens.

To analyze the traversal time of any message $m_\star$ which ingresses via the input at time $t = 0$ and egresses via the egress port, we define the following time-dependent state variables:

Definition 4.1. $\mathcal{I}_{out}(t)$ is a boolean value which is equal to *true* if the output link *out* connected to the egress port of a switch is *idle* at time t, and *false* otherwise.

Definition 4.2. $\mathcal{V}_{m_\star}(t)$ (message $m_\star$ is *visible* at time t) is a boolean value which is equal to *true* if the message $m_\star$ is at the front of its respective queue q_i at time t, i.e. $pre_{m_\star}(t) = \emptyset$, and *false* otherwise

Definition 4.3. $\mathcal{A}_i(t)$ (priority i is *active* at time t) is a boolean value which is equal to *true* if the gate g_i (of queue q_i) is open at time t, and q_i has pending messages, i.e. $g_i(t) = open \wedge bl_i(t) > 0$, and *false* otherwise.

Equation 1 describes the required condition for a message $m_\star$ arriving at the input at $t = 0$ to begin egress through the output port at time $t = x$ expressed using definitions 4.1, 4.2 and 4.3.

$$\mathcal{I}_{out}(x) \wedge \mathcal{V}_{m_\star}(x) \wedge \mathcal{A}_i(x) \wedge \bigwedge_{j \in hp(\mathrm{i})} \overline{\mathcal{A}_j(x)} = true \qquad (1)$$

After $m_\star$ has been assigned to a queue q_i all predecessors of $m_\star$ have to leave the queue first, before $m_\star$ becomes visible to the transmission selection function f_{out} (i.e. $\mathcal{V}_{m_\star}(x) = true$). After $m_\star$ reaches the front of its respective queue q_i, that queue has to become active, i.e. its corresponding gate g_i has to be open (i.e. $\mathcal{A}_i(x) = true$). In addition the output has to be idle (i.e. $\mathcal{I}_{out}(x) = true$) and all queues of higher priority than q_i have to be inactive (i.e. $\mathcal{A}_j(x) = \text{false}\ \forall j \in hp(i)$). If all of these conditions hold at $t = x$, $m_\star$ will start egressing via the egress port.

5 PROBLEM DESCRIPTION

In this section we describe the problem which we address in this work. Figure 2 illustrates the bandwidth wastage that can occur in a TSN switch which uses TAS and a gate schedule which contains guard bands. The figure depicts the input *in* to a an input queuing function $f_{in,p}$ of a TSN switch port p, the backlogs of the queues for TAS messages bl_s and non-TAS messages bl_u, the gate schedules of the gates for TAS traffic g_s and non-TAS traffic g_u and, finally, the output sequence *out* via the egress port p.

in shows the ingress sequence of incoming messages. Messages can arrive in any particular order and even simultaneously, as described in Section 3. In this example, three non-TAS messages $\{u_1, u_2, u_3\}$ and two TAS messages $\{s_1, s_2\}$ ingress via *in*. The length of all messages is equivalent to 2 "time units" except u_3 which only has a length equivalent to 1 time unit. The backlogs bl_s and bl_u increase as messages ingress via *in* and are stored in the respective queues, and decrease as messages egress via *out*. A value of *o* for the gate states g_s and g_u represents an open gate, while a value of *c* denotes a closed gate. Notice the two guard bands, b_1 and b_2 in the gate state graph of g_u, during which g_u is closed to ensure an idle output link for the scheduled transmissions of the TAS messages.

The two crosshatched segments in the output sequence *out* — w_1 and w_2 — represent the time intervals where bandwidth is wasted in this scenario. During w_1, the network is idle, even though message u_2 is ready for transmission. However, g_u is closed due to the guard

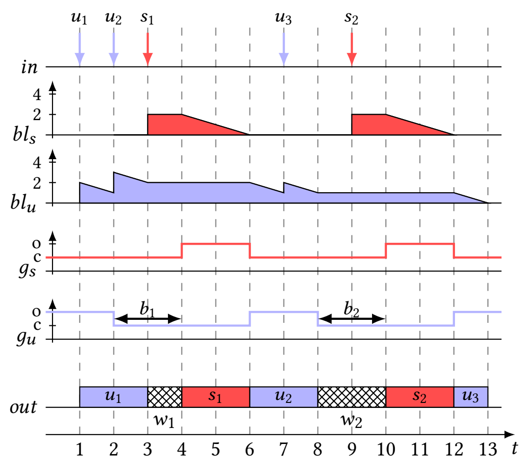


Figure 2: Bandwidth usage inefficiencies caused by guard bands: $len(u_1) = len(u_2) = len(s_1) = len(s_2) = 2; len(u_3) = 1$

band b_1, so u_2 can't start to egress via port p. Similarly, during w_2, message u_3 is ready for transmission. In this case, u_3 could actually finish its transmission without interfering with the egress of s_2. However, since the size of the guard bands is set according to the largest possible message size in q_u, the egress of u_3 is deferred to time point 12. Note how the two TAS messages s_1 and s_2 experience no jitter in this example, as both spend the same amount of time inside q_s before they egress.

This example shows that the usage of TAS with guard bands can lead to bandwidth wastage in a TSN network. The bandwidth wastage depends not only on the placement of the guard bands and gate openings of the scheduled traffic, but also on the size and order of the messages entering the unscheduled queue (e.g. if u_2 were only one time unit long), u_3 could egress before g_u closes, leading to no bandwidth wastage (no message ready to transmit during time interval [8,10]).

6 SIZE-BASED QUEUING

Section 6.1 describes how to modify a TSN switch in order to implement Size-Based Queuing (SBQ) while Section 6.2 provides an approach on generating schedules for an SBQ enabled TSN network.

6.1 Implementation of SBQ

SBQ requires the following modifications:

- For any number of non-TAS priorities, add n queues with associated gates to the existing queue, forming a queue-set $q_{i,p,n}$ with cardinality $c = 1 + n$.
- For each queue-set, belonging to a priority i of egress port p, define a set of $c - 1$ thresholds $\tau_{i,p,k} \in \mathcal{T}_{i,p}$ with $\forall k, \tau_{i,p,k} < \tau_{i,p,k+1}$. These thresholds are used by the input queuing function $f_{in,p}$ to assign incoming non-TAS messages to a queue-set according to their priority i and to a queue within each queue-set according to their size.
- Modify the gate-schedule gs_p to accommodate the new gates created by SBQ. These gates require their own opening and closing events, like the existing gates, so new columns have to be added in the gate-schedule.
- Schedule the new queues such that all queues of a queue-set receive sufficient portions of the available output bandwidth via the egress port (our approach is provided in Section 6.2).
- Shrink the guard bands in the gate-schedule of each individual queue of the queue-set to reflect the largest possible size of messages in the respective queue.
- Modify $f_{in,p}$ and $f_{out,p}$ to handle the new queues and implement the distribution of messages to queues according to their size.

Note that these modifications don't necessarily have to be made to all non-TAS priorities of all egress ports of all switches in a network. Instead, SBQ can be applied individually, leaving individual priorities, egress ports or entire TSN switches in the network unchanged.

Figure 1b) shows an exemplary implementation of SBQ in the TSN switch in Figure 1a). Once more, the port index p has been omitted to improve the clarity. In this example, for priorities $i = 5$ and $i = 6$, one queue has been added to the existing one, forming a queue-set containing two queues ($c = 2$) for each of the two priorities. Priorities are assigned to the new queues such that $prio(q_{6,sm}) > prio(q_{6,lrg}) > prio(q_{5,sm}) > prio(q_{5,lrg})$. This allows us to use the existing f_{out} by simply adding the additional priorities.

One threshold value τ_i for priorities 5 and 6 is used to assign the incoming large messages ($len(m_\star) \geq \tau_i$) to $q_{6,lrg}$ and $q_{5,lrg}$, and small messages ($len(m_\star) \leq \tau_i$) to $q_{6,sm}$, $q_{5,sm}$. A corresponding gate for each new queue is added, as well as an updated gate-schedule gs and input queueing function f_{in} to accommodate the additional queues. The remaining priorities $i \in \{0, \ldots, 4\}$ are not modified.

6.2 Scheduling multiple queues per priority

Applying SBQ to a TSN switch affects the gate-schedule generation of the network. Even though the newly added queues may contain messages with equal priority, the transmission selection function has to be aware of the additional queues.

In this work, we propose to treat each queue in each queue-set with a separate priority, and adjust the schedule of the respective gates to open and close such that the bandwidth is shared in a fair way between the queues. One such example is provided in Figure 3. In this figure, a single non-TAS queue q_i is split into a queue-set containing two queues q_{lrg} and q_{sm}, as shown for priorities 5 and 6 in Figure 1b). We assume a maximum message size $len(m_\star) < 2$ and choose a threshold $\tau = 1$. This assures that $1 \leq len_{q_{lrg}}(m_\star) < 2$ and $len_{q_{sm}}(m_\star) < 1$. q_{sm} is assigned a higher priority than q_{lrg}. This allows us to use the existing transmission selection function $f_{out,p}$ by simply increasing the amount of possible priorities. For this example, g_{lrg} inherits the schedule of g_i. In order to allow messages from both queues to win arbitration via $f_{out,p}$, g_{sm} is only periodically opened and left open for at least 2 time units to allow a maximum sized message from q_{lrg} to egress and a message from q_{sm} to start transmission. Closing g_{sm} is necessary to avoid starvation of q_{lrg} in case that q_{sm} is never empty. q_{sm} is given a higher priority than q_{lrg} to minimize the required amount of time where the queue with higher priority has to be closed (q_{lrg} would have to be closed for 2 time units in order for one large message to

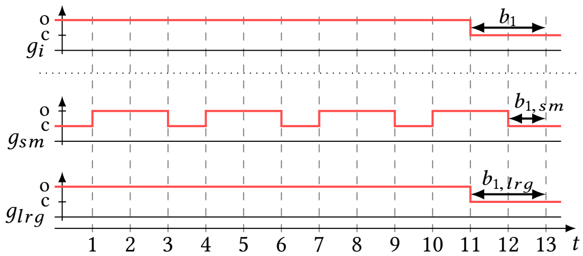


Figure 3: Scheduling approach for SBQ queue-sets

finish transmission and a small message to win arbitration afterwards). The resulting distribution of egress bandwidth for q_{sm} and q_{lrg} can be adjusted by, i.e., adjusting the size and frequency of the open gate intervals of q_{sm}.

Also note the guard band b_1 in the schedule of g_i also present in g_{lrg} (named $b_{1,lrg}$). Due to the limited maximum message size in q_{sm}, the corresponding guard band, $b_{1,sm}$, can be reduced to half its size, which allows for messages from q_{sm} to start transmission until time point 12.

7 ANALYSIS

As simulation tools capable of simulating SBQ are not yet available, we will perform a qualitative analysis of SBQ in this section.

Section 7.1 illustrates how SBQ can improve the bandwidth utilization of the egress port. Section 7.2 will discuss the impact of SBQ on the traversal time of messages in an SBQ enabled TSN network. Sections 7.3 and 7.4 discuss the implementation cost of SBQ, as well as the issue of message overtaking under SBQ, respectively.

7.1 SBQ impact on bandwidth utilization

In order to show how SBQ can improve the bandwidth utilization of a TSN network, we apply SBQ to the example illustrated in Figure 2. The updated scenario is illustrated in Figure 4. The original non-TAS queue q_u has been split into a queue for messages of size $len(m_\star) > 1$, q_{lrg} and a small queue q_{sm} for messages of size $len(m_\star) \leq 1$. This will cause u_1 and u_2 to be put into q_{lrg} and u_3 to be put into q_{sm}. Because q_{sm} can only contain messages smaller or equal to 1, the two guard bands $b_{1,sm}$ and $b_{2,sm}$ can be reduced to one time unit in the respective gate schedule for g_{sm}. This, in turn, allows u_3 to start transmission at time 8, where it previously was not eligible for transmission. SBQ has, thus, reduced bandwidth wastage without impacting the transmission of the TAS messages s_1 and s_2.

7.2 SBQ impact on message traversal time

We can use Definitions 4.1, 4.2 and 4.3 from Section 4 to describe the impact of SBQ on the traversal time of a message through a TSN switch, where the queue belonging to the priority of the message has been split using SBQ. For the following description, assume a message $m_\star$ of priority i. We focus on the traversal time of this message through a TSN switch sw where, for the egress port p, queue $q_{i,p}$ has been split into n queues $q_{i,p,n}$ using a set of $k = n-1$ thresholds $\tau_{i,p,k}$. Distinct priorities have been assigned to each of the the queues $q_{i,p,n}$ and the schedule gs_p has been extended to schedule the new queues, for example, by using an approach as described in Section 6.2. We assume that $m_\star$ is stored in $q_{i,p,j}$. The traversal time of $m_\star$ through sw depends on the following factors:

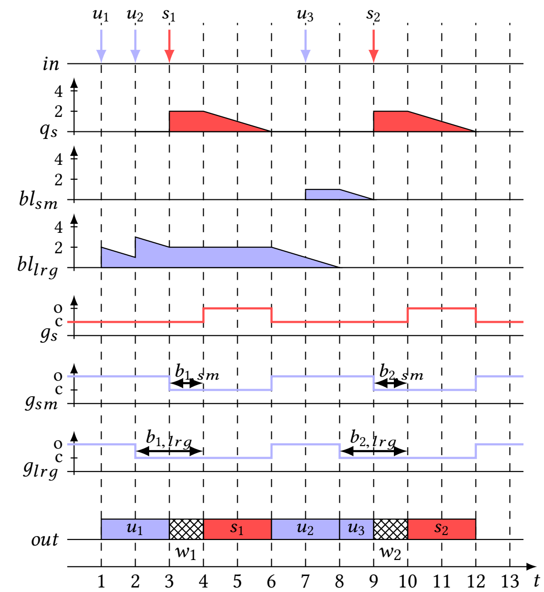


Figure 4: Bandwidth usage improvement with SBQ: $len(u_1) = len(u_2) = len(s_1) = len(s_2) = 2; len(u_3) = 1$

(1) How often queue $q_{i,p,j}$ becomes active.
(2) How likely it is that $q_{i,p,j}$ wins arbitration against the other queues of the egress port
(3) How many messages $m_\star$ competes against inside queue $q_{i,p,j}$.

Item 1 is affected by the gate-schedule gs_p for the gate $g_{i,p,j}$, as priority i,j only becomes active if the respective gate $g_{i,p,j}$ is open. As shown in Section 6.2, the gates of higher-priority queues have to be closed at some point to allow lower-priority queues to win arbitration. As a result, when using SBQ, the sub-priority i,j will be active less often than priority i from the scenario without SBQ (i.e. $\mathcal{A}_{i,j}(t)$ will be true less often than $\mathcal{A}_i(t)$). Because of this the traversal time of message $m_\star$ through the switch will increase under SBQ.

Item 2 depends on the priority assignment of the queue-set $q_{i,p,n}$. The lower the priority of $q_{i,p,j}$ the more queues win arbitration if they are active at the same time. The traversal time of message $m_\star$ is, therefore, related to the priority of queue $q_{i,p,j}$. Under SBQ, if $q_{i,p,j}$ has a low priority within the queue-set, the traversal time of message $m_\star$ through the switch will increase

Item 3 depends on the thresholds $\tau_{i,p,k}$ and the distribution of message sizes on the input. If SBQ is used, message $m_\star$ will have a less or equal amount of predecessors $pre_{m_\star}(t)$ at any given time, since the incoming messages are now distributed between multiple queues. $\mathcal{V}_{m_\star}$ can, thus, become true at an earlier point, reducing the traversal time of $m_\star$ under SBQ.

In summary, the impact of SBQ on the traversal time of a message $m_\star$ depends on the parameters used to implement SBQ (thresholds, gate-schedule and priority assignment). While some aspects of SBQ can reduce the traversal time, other aspects can increase it. The detailed analysis is left as future work.

Also note that SBQ does not affect the transmission of TAS messages, as guard bands are still inserted such that non-TAS messages can't interfere with TAS transmissions. The usage of SBQ impacts the traversal time of non-TAS messages, as the amount of backlog in the queue-set formed by SBQ at a given time may differ from the backlog of the original queue.

7.3 Implementation cost of SBQ

In order to implement SBQ, several components are added to a TSN switch. When a queue is split into a queue-set, multiple queues are added, depending on the number of thresholds. This may require additional memory in the switch. For each added queue, a new gate, as well as additional space in the gate schedule are required as well. This, again, increases the memory requirements of a TSN switch with SBQ enabled. Finally, the input queuing function $f_{in,p}$ has to be extended, in order to be able to parse the length of each incoming message and assign it to the correct queue within a queue-set. Similarly, the transmission selection function $f_{out,p}$ has to be extended to be able to handle the larger amount of queues.

Implementing SBQ does not break existing TSN standards or protocols, as packet formats and other TSN functionalities are not changed by SBQ. This allows to mix SBQ enabled devices with standard TSN devices and puts SBQ at an advantage over IEEE802.1Qbu [2] where frame preemption capabilities can only be used if all involved devices support it.

7.4 Message overtaking

One issue that arises with the usage of SBQ is message overtaking. If the size of the messages of a given data stream varies, the individual messages of the stream can be assigned to different queues within a queue-set. Depending on the backlog and schedule of the queues in the queue set, the transmission sequence of messages may change. Message overtaking requires additional logic to be implemented in the receiver to reconstruct the original message sequence. If messages of a particular stream of data have constant size, message overtaking is not an issue, as all messages of the stream will be put into the same queues on their paths and, thus, will be transmitted in order.

This issue is, however, also present in existing TSN standards, in particular IEEE 802.1CB [4] where messages are replicated across multiple paths in the network to achieve redundancy. Here, if the transmission time of the redundant paths differs, in the presence of message losses, messages on the fast path may overtake messages on the slow path. The standard does not outline counter-measures for this issue. Message overtaking, thus, has to be considered during design time or mitigated on higher layers of the network stack, both for IEEE 802.1CB as well as SBQ.

8 CONCLUSION AND FUTURE WORK

In this work, we introduced Size-Based Queuing (SBQ), an approach to reduce the amount of wasted bandwidth in a non-preemptive TSN network which uses the Time-Aware Shaper. After a description of the approach, we performed a qualitative analysis. This analysis has shown how the usage of SBQ affects the traversal time of a non-TAS message through a TSN switch. Most importantly, SBQ does not affect high-priority TAS messages. We also concluded that the impact of SBQ greatly depends on the parameters of the approach. At the same time, we have shown that SBQ can reduce the amount of wasted bandwidth. We also discussed the issue of message overtaking, which has to be considered during network design or mitigated at runtime.

For future work, we want to address the parameters which are introduced by SBQ, namely the set of thresholds and the gate-schedule for the newly added queues. As these parameters influence the impact of SBQ on message traversal time, analyzing them and providing guidance on how to set them will allow us to quantify the benefits of SBQ more precisely. We also want to explore the possibility of using other policies to arbitrate between the queues of a queue-set formed by SBQ. Other policies, such as Round Robin, may offer better performance, while, at the same time, simplifying the schedule generation for SBQ-enabled networks.

REFERENCES

[1] 2009. Aircraft Data Network Part 7. Avionics Full-Duplex Switched Ethernet Network. ARINC Specification 664 P7-1. (September 2009).

[2] 2016. IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption. *IEEE Std 802.1Qbu-2016 (Amendment to IEEE Std 802.1Q-2014)* (2016).

[3] 2016. IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic. *IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q-2014/Cor 1-2015)* (2016).

[4] 2017. IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability. *IEEE Std 802.1CB-2017* (2017).

[5] Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. 2016. Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS '16)*.

[6] Frank Dürr and Naresh Ganesh Nayak. 2016. No-wait Packet Scheduling for IEEE Time-sensitive Networks (TSN). In *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS '16)*.

[7] J. P. Georges, T. Divoux, and E. Rondeau. 2005. Strict Priority versus Weighted Fair Queueing in Switched Ethernet Networks for Time Critical Applications. In *19th IEEE International Parallel and Distributed Processing Symposium*.

[8] IEEE 802.1 Time-Sensitive Networking Task Group. 2018. IEEE 802.1 Time-Sensitive Networking. http://www.ieee802.org/1/pages/tsn.html. (2018). (Accessed on 04/19/2018).

[9] IEEE 802.1 Time-Sensitive Networking Task Group. 2018. P802.1AS-Rev - Timing and Synchronization for Time-Sensitive Applications. https://1.ieee802.org/tsn/802-1as-rev/. (2018). (Accessed on 04/19/2018).

[10] IEEE 802.1 Time-Sensitive Networking Task Group. 2018. P802.1Qcr - Bridges and Bridged Networks Amendment: Asynchronous Traffic Shaping. https://1.ieee802.org/tsn/802-1qcr/. (2018). (Accessed on 04/19/2018).

[11] D. Hisano, Y. Nakayama, T. Kubo, T. Shimizu, H. Nakamura, J. Terada, and A. Otaka. 2017. Gate-Shrunk Time Aware Shaper: Low-Latency Converged Network for 5G Fronthaul and M2M Services. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*.

[12] SAE. 2011. Time-Triggered Ethernet, A6802. (2011).

[13] M. Shreedhar and G. Varghese. 1996. Efficient fair queuing using Deficit Round-Robin. *IEEE/ACM Transactions on Networking* (1996).

[14] J. Specht and S. Samii. 2016. Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks. In *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*.

[15] S. Thangamuthu, N. Concer, P. J. L. Cuijpers, and J. J. Lukkien. 2015. Analysis of Ethernet-switch traffic shapers for in-vehicle networking applications. In *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*.

[16] D. Thiele, R. Ernst, and J. Diemer. 2015. Formal worst-case timing analysis of Ethernet TSN's time-aware and peristaltic shapers. In *2015 IEEE Vehicular Networking Conference (VNC)*.

[17] Xiao Zhang and L. N. Bhuyan. 2003. Deficit round-robin scheduling for input-queued switches. *IEEE Journal on Selected Areas in Communications* (2003).