Technical Report on Bridge-Local Guaranteed Latency with Strict Priority Scheduling

Alexej Grigorjew
University of Würzburg, Germany
alexej.grigorjew@uni-wuerzburg.de

Florian Metzger
University of Würzburg, Germany
florian.metzger@uni-wuerzburg.de

Tobias Hoßfeld
University of Würzburg, Germany
tobias.hossfeld@uni-wuerzburg.de

Johannes Specht
University of Duisburg-Essen, Germany
johannes.specht@uni-due.de

Franz-Josef Götz
Siemens AG, Germany
franz-josef.goetz@siemens.com

Jürgen Schmitt
Siemens AG, Germany
juergen.jues.schmitt@siemens.com

Feng Chen
Siemens AG, Germany
chen.feng@siemens.com

***Abstract*—Bridge-local latency computation is often regarded with caution, as previous efforts with the Credit-Based Shaper (CBS) showed that CBS requires network wide information for tight bounds. Recently, new shaping mechanisms and timed gates were applied to achieve such guarantees nonetheless, but they require support for these new mechanisms in the forwarding devices.**

This document presents a per-hop latency bound for individual streams in a class-based network that applies the IEEE 802.1Q strict priority transmission selection algorithm. It is based on self-pacing talkers and uses the accumulated latency fields during the reservation process to provide upper bounds with bridge-local information. The presented delay bound is proven mathematically and then evaluated with respect to its accuracy. It indicates the required information that must be provided for admission control, e.g., implemented by a resource reservation protocol such as IEEE 802.1Qdd. Further, it hints at potential improvements regarding new mechanisms and higher accuracy, given more information.

I. Introduction

When the Audio Video Bridging task group [1] first specified mechanisms for deterministic latency bounds, their initial efforts were focused on the Credit-Based Shaper (CBS). CBS [2, IEEE 802.1Qav] is used to re-shape traffic on a per-class basis such that it does not exceed its assigned bandwidth share at every hop. Complementary, the Stream Reservation Protocol (SRP) [3, IEEE 802.1Qat] was designed to provide the necessary information to compute deterministic latency bounds for CBS-shaped traffic. It specifies an admission control system that provides every bridge with a traffic specification (TSpec) from every stream that passes through that bridge. The standard IEEE 802.1BA [4] was intended to provide usage-specific profiles for this admission control system. It suggests two stream reservation classes A and B with a maximum end-to-end latency target of $2\,\mathrm{ms}$ and $50\,\mathrm{ms}$ respectively, along with example worst case calculations based on the bridge-local information from the SRP.

However, the *Deggendorf use case* [5] showed that the latency targets can be exceeded under certain conditions. The worst case delay in bridge h_n depends on the stream configuration of the earlier hops $\{h_1,...,h_{n-1}\}$, which includes those streams that do not pass through h_n and share their TSpec. The upper bound in 802.1BA did not account for such cases and is not generally applicable. The task group then specified new mechanisms and was later renamed into Time-Sensitive Networking (TSN) [6] to account for the broader range of use cases.

The most prominent mechanism is specified in the Enhancements for Scheduled Traffic (EST) [7, IEEE 802.1Qbv], also referred to as *timed gates*. It is based on a common sense of synchronized clock time, and timed gate control lists in each bridge. However, switching hardware must support this new mechanism, and there is no distributed, dynamic admission control system specified for timed gates yet. Later, Asynchronous Traffic Shaping (ATS) was specified in IEEE P802.1Qcr [8], [9]. It applies per-stream shaping and allows for per-hop latency bounds for each stream with bridge-local information, but also requires support for this new transmission selection algorithm in the switch fabric.

Contribution: This work presents a formally proven per-hop latency bound for the IEEE 802.1Q Strict Priority transmission selection algorithm [10, Section 8.6.8] which is commonly available in current bridges. It is generally applicable with self-pacing talkers and does not rely on per-hop shaping or timed gates. Unlike previous works, it focuses on a distributed admission control system with low complexity and bridge-local information rather than tight end-to-end bounds. This information can be included in the reservation process of upcoming admission control systems such as the Resource Allocation Protocol (RAP) [11].

The remainder of this document is structured as follows. Section II presents background information regarding the used symbols, the assumed delay model, the assumed talker

Table I
USED VARIABLES, SYMBOLS, AND ABBREVIATIONS IN THIS DOCUMENT

Variable	Description
TQ	Transmission Queue
SF, S & F	Store and Forward
Proc	Refers to **Proc**essing Delay (Switch Fabric)
Prop	Refers to **Prop**agation Delay
$\mathcal{S}$	Set of all streams $i \in \mathcal{S}$ in the system
f_i	A particular frame of stream i
$\mathcal{F}_i$	Set of all possible frames f_i from stream i
p_i	Traffic class (priority) of frames from stream i
$\delta_p^{h_k}$	Per-hop latency guarantee of traffic class p at the hop h_k
δ_p	Per-hop latency guarantee of class p at the current hop (the bridge that computes the delay bound)
b_i	Burst size (typically $\hat{\ell}_i$) of stream i in bit
ℓ_{f_i}	Length of frame f_i from stream i in bit
$\hat{\ell}_i$	Maximum length of frames from stream i: $\hat{\ell}_i = \max_{f_i \in \mathcal{F}_i}\{\ell_{f_i}\}$
τ_i	Burst interval: time duration during which the talker of stream i transmits no more than one burst b_i
$y_{i,x}$	Number of bursts from a higher class stream x that interferes with stream i
z_x	Number of bursts from a stream x with the same traffic class that interferes with stream i
r	Link speed (e.g., $1\,\mathrm{Gbit/s}$)
$accMaxD_i^{h_k}$	Accumulated maximum latency during the reservation of stream i at hop h_k: $accMaxD_i^{h_k} = \sum_{q=1}^{k} \delta_{p_i}^{h_q}$
$accMinD_i^{h_k}$	Accumulated minimum latency during the reservation of stream i at hop h_k: $accMinD_i^{h_k} = k \cdot \check{d}_i^{SF}$
$d_{f_i}^{TQ}$	Delay caused on frame f_i from stream i by interfering frames in the transmission queue
$d_{f_i}^{SF}$	Delay caused on frame f_i from stream i by the store and forward (S & F) operation; $d_{f_i}^{SF} = \ell_{f_i}/r$
d_i^{TQ}, d_i^{SF}	Possible delays of stream i based on its frames $f_i \in \mathcal{F}_i$
$\check{d}_i^{SF}$	Smallest transmission time of stream i: $\check{d}_i^{SF} = \min(d_i^{SF})$
$d_i^{TQ,SF}$	Sum of both delays: $d_i^{TQ} + d_i^{SF}$
$d_i^{h_1..h_k}$	Sum of delays of stream i on its path from h_1 to h_k
d_i^{e2e}	Sum of all delays of stream i on its path (end-to-end)

behavior, and the assumed admission control process. In Section III, the per-hop latency upper bound is presented, and evaluated with respect to accuracy in Section IV. Related publications and technologies are discussed in Section V. Finally, Section VI concludes the paper.

II. BACKGROUNDS

All used symbols in each figure and equation in this document are explained in Table I. Thereby, sets are always uppercase calligraphic letters ($\mathcal{S}$ and $\mathcal{F}_i$). Small letters represent an element of the set (frame $f_i \in \mathcal{F}_i$) or individual values (link speed r). The delay bound presented in this document refers to all frames f_i of an observed stream i. Other streams that contribute to the delay of f_i are denoted with the index x, e.g., the burst b_x or frame size $\hat{\ell}_x$.

A. Switch Delay Model

The underlying delay model used for the latency calculation is presented in Figure 1. Thereby, the per-hop delay d_{f_i} of frame f_i is comprised of the processing delay $d_{f_i}^{Proc}$, the

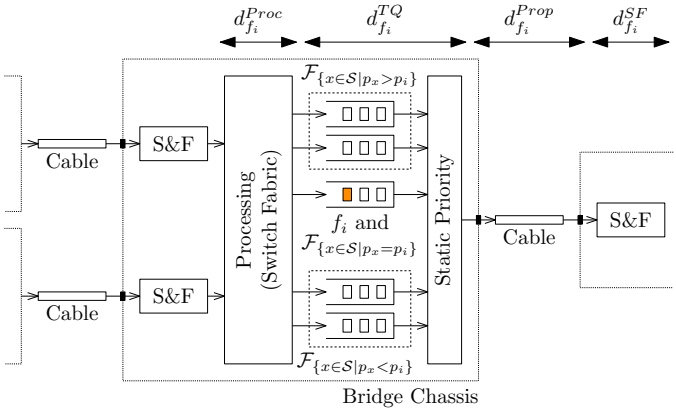


Figure 1. Illustration of per-hop delay components and different traffic priority classes.

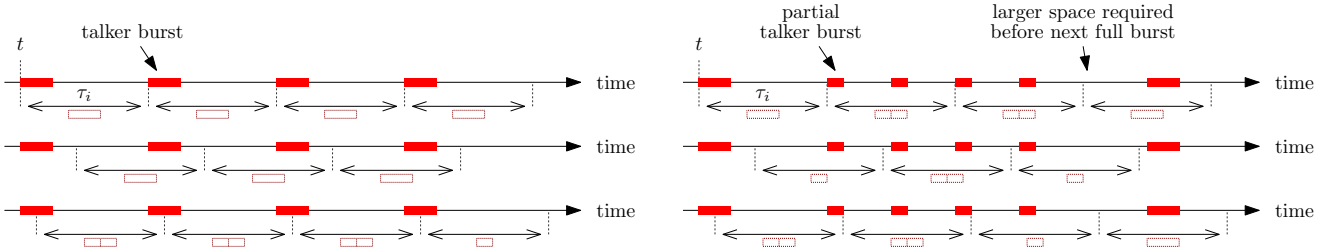
queuing delay $d_{f_i}^{TQ}$, the propagation delay $d_{f_i}^{Prop}$, and the transmission delay (store and forward) $d_{f_i}^{SF}$: $d_{f_i} = d_{f_i}^{Proc} + d_{f_i}^{TQ} + d_{f_i}^{Prop} + d_{f_i}^{SF}$.

In this document, the processing delay is not considered further as it highly depends on the used switch implementation. It is assumed that an upper bound for the per-frame processing delay exists that can simply be added here. This processing delay contains all steps performed by the switch fabric before a frame hits one of the transmission queues at the egress port, such as filters, mac address table lookups, and meters.

The queuing delay represents the time that f_i spends in the transmission queue (TQ) of its respective traffic class. This document assumes a class based forwarding process as per IEEE 802.1Q [10, Section 8.6.8], i.e., there is one queue for each supported traffic class. This queue is shared with all frames $f_x \in \mathcal{F}_x$ from all streams x that have the same class: $\{x \in \mathcal{S} \mid p_x = p_i\}$. The other queues of the same egress port are occupied by frames f_x of higher traffic classes $\{x \in \mathcal{S} \mid p_x > p_i\}$ or lower classes $\{x \in \mathcal{S} \mid p_x < p_i\}$. Frames of higher traffic class queues may pass f_i as soon as they are available for transmission. Ongoing lower class transmissions are completed even when f_i arrives in the queue (i.e., no frame preemption is used), but no further lower class transmissions are started before f_i and every other higher classed frame are completely transmitted. This property allows the coexistence of reserved, high priority real-time streams and low priority best effort streams in the same network.

The propagation delay is determined by the physical distance that the frame travels in the network and the respective medium's propagation speed (e.g., $2 \times 10^8\,\mathrm{m/s}$ for copper wire). The maximum propagation delay of Ethernet links with copper wires measures roughly $100\,\mathrm{m}/(2 \times 10^8\,\mathrm{m/s}) = 0.5\,\mu\mathrm{s}$, which is negligible in most use cases ($r = 1\,\mathrm{Gbit/s}$) or can be considered as a fixed upper bound in addition to the processing time and Equation 2. Finally, the transmission delay (store and forward) is determined by the link speed r and the frame size ℓ_{f_i}: $d_{f_i}^{SF} = \ell_{f_i}/r$.

Due to the existence of traffic independent upper bounds for the processing delay $d_{f_i}^{Proc}$ and the propagation delay $d_{f_i}^{Prop}$,



(a) Full-burst frames sent in the interval τ_i.

(b) Bursts are divided into multiple smaller frames.

Figure 2. Illustration of the burst interval τ_i from Constraint 4. The time lines show frames from a single stream at a single hop.

only the queuing and transmission delay is considered in this document. The sum of both delays is referred to as $d_{f_i}^{TQ,SF} = d_{f_i}^{TQ} + d_{f_i}^{SF}$.

B. Constraints for Streams and Talkers

It is assumed that all talkers and bridges in the network adhere to the following constraints. In case of misbehaving devices (e.g., babbling idiot), bridges may use meters and filters to prevent further damage in the network. The presented constraints are utilized to obtain an upper bound for the per-hop latency in Section III-A.

(1) IEEE 802.1Q [10, Section 8.6.8] priority transmission selection is used, i.e., frames within a higher traffic class are always selected for transmission before lower traffic classes when more than one frame is eligible for transmission.

(2) Bridges use a FIFO transmission selection algorithm for all frames within the same queue (traffic class). This is based on the strict priority algorithm as per 802.1Q [10, Section 8.6.8.1], but the order of all frames with a given VID and priority is preserved, independently of their flow hash, destination, or source address.

(3) Talker transmissions do not exceed a previously communicated burst size b_i and maximum frame size $\hat{\ell}_i$ for each stream i. (cf. Section II-C)

(4) All talkers pace their traffic according to their burst interval τ_i (self-pacing talkers). For any point t in time, the traffic sent by the talker for stream i in the time interval $[t, t+\tau_i]$ may not exceed its burst size b_i.

(5) No further shaping mechanisms (apart from Constraints 3 and 4) are used in any considered traffic class. The earliest frame of each considered traffic class is always regarded eligible for transmission.

(6) For every bridge h and every traffic class p there is a predefined per-hop delay guarantee δ_p^h. Admission control prevents the deployment of streams in the network that would cause delay violations for any already accepted stream, i.e., it always holds at every hop h for every stream $i \in \mathcal{S}$:

$$d_i^{TQ,SF} \leq \delta_{p_i}^h \tag{1}$$

The pacing from Constraint 4 can be enforced by a token bucket or leaky bucket algorithm [12], [13] at the talker. Figure 2 illustrates the intended property. Red bars indicate frame transmissions at the talker. The arrows below indicate the burst interval τ_i at different positions t in the time line. The dashed bars below the arrow indicate the amount of traffic that was transmitted within the interval, possibly comprised of multiple frames. For any point t in time, this amount may not exceed the burst size b_i. This is similar to the Class Measurement Interval in IEEE standard 802.1Q [10, Section 34.6.1] with variable interval lengths.

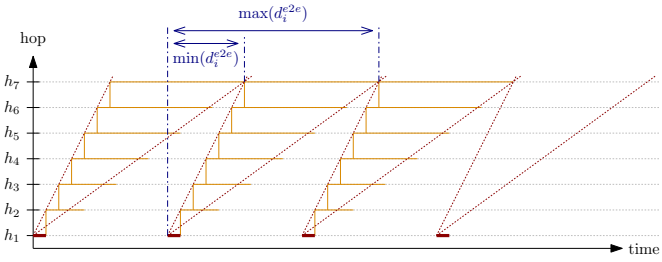


Figure 3. Illustration of Constraint 4 and frame residence times.

Constraint 4 is equivalent to burst-rate envelopes commonly used in Network Calculus and in the Asynchronous Traffic Shaper [9]. If a talker is pacing its frames with the burst interval τ_i and burst size b_i, it adheres to a traffic envelope with burst b_i and rate $r_i = b_i/\tau_i$. Similarly, if a stream is bounded by an envelope with burst b_i and rate r_i, it is paced with the burst interval $\tau_i = b_i/r_i$.

Frame residence times. The above properties can be used to compute the possible residence time of each frame at each hop on its path. Figure 3 illustrates a worst case estimate of possible residence times of multiple bursts at each hop. Each line on the Y-axis represents an individual hop on the stream's path. The red bars at hop h_1 indicate new bursts at the talker. Each horizontal orange bar above h_1 represents the possible residence time of the respective burst at each hop h_k, from the earliest possible time of arrival ($\min(d_i^{h_1..h_{k-1}})$) to the latest possible time of departure ($\max(d_i^{h_1..h_k})$). These can be calculated based on their transmission time $\check{d}_i^{SF} = \min(d_i^{SF})$ and their per-hop latency bound $\delta_{p_i}^{h_k}$. At the last hop h_n on the path, it holds $\min(d_i^{e2e}) = (n-1)\check{d}_i^{SF}$ and $\max(d_i^{e2e}) = \sum_{q=1}^{n} \delta_{p_i}^{h_q}$. These values are accumulated and passed along to each hop during the reservation process, as explained in Section II-C.

The example in Figure 3 shows the isolation of bursts when $\tau_i > \max(d_i^{e2e}) - \min(d_i^{e2e})$ holds for the burst interval τ_i.

In this case, the residence times of individual bursts do not overlap at any hop on the path. Only a single burst of this stream can be in the transmission queue at the same time. A similar illustration is later used in Figure 4 to prove the latency bound for the general case where residence times may actually overlap.

C. Resource Reservation

In order to compute a latency bound, some information on the involved streams must be communicated towards the admission control instance. This may either be a central controlling entity in the network, or can be implemented in a distributed way in each forwarding device. Depending on the exact mechanism (transmission selection algorithm, constraints) and the desired latency bound, different information may be required. The remainder of this document assumes a *distributed* resource reservation without a dedicated controller. In addition, a frame's priority does not change on its path through the network.

In this case, the latency bound can be calculated on a *per-hop* basis, using only bridge-local information regarding the streams that traverse the respective hop. Therefore, the presented guarantee is suited for a distributed admission control approach, such as developed by the TSN standardization group in P802.1Qdd [11] (RAP, Resource Allocation Protocol). The required per-stream information to apply Equation 2 includes:

- the traffic class (priority) p_i of each stream i,
- the maximum frame length $\hat{\ell}_i$ and minimum frame length $\check{\ell}_i$ of each stream i (e.g., $1542\,\mathrm{B}$ and $64\,\mathrm{B}$),
- the committed burst size b_i of each stream i (possibly comprised of multiple frames),
- the minimum time interval τ_i between two burst transmissions at the talker,
- the accumulated maximum latency $accMaxD_i^{h_k}$ and minimum latency $accMinD_i^{h_k}$.

Most of this information can be communicated by the talker in a traffic specification (TSpec) inside a *talker advertise attribute* during the stream reservation process. It is carried along the network across all required bridges towards a potential listener, which responds with a *listener join attribute* if it attempts to subscribe for the respective stream.

The bridges are pre-configured with a maximum allowed per-hop delay $\delta_{p_i}^{h_k}$ for each supported traffic class p_i. In the talker advertise attribute, each bridge h_k on the path adds its per-hop delay guarantee $\delta_{p_j}^{h_k}$ to the reservation's accumulated maximum latency attribute ($accMaxD_i^{h_k}$), and the minimum transmission delay $\check{d}_j^{SF}$ to the accumulated minimum latency ($accMinD_i^{h_k}$). These attributes can be used to derive $\max(d_j^{h_1..h_n})$ and $\min(d_j^{h_1..h_{n-1}})$ at each hop h_n. At the end of the path (listener), they provide a worst case estimate for the end-to-end delay of the new stream.

During the listener join procedure, Equation 2 is used at each hop h_k to verify that the current per-hop delay bound d_i of each already existing stream i does not exceed the upper bound $\delta_{p_i}^{h_k}$ of its traffic class. If each stream is still bounded below its guarantee, i.e., if $d_i \leq \delta_{p_i}^{h_k}$ for each stream $i \in \mathcal{S}$, the new reservation of stream j can be accepted by the current hop h_k.

Note the difference between the current delay bound d_i (result of Eq. 2) and the guaranteed upper bound $\delta_{p_i}^{h_k}$ that is pre-configured for each traffic class p_i. The latter $\delta_{p_i}^{h_k}$ is used to obtain a maximum end-to-end delay guarantee during the reservation process. The formula for d_i is only used to verify that the pre-defined bound $\delta_{p_i}^{h_k}$ still holds at each new reservation.

III. Latency Bound

Based on the delay model and constraints from Section II, the worst case queuing and transmission delay of all frames from a stream i at hop h_k is bounded by Equation 2:

$$d_i^{TQ,SF} \leq \sum_{\{x \in \mathcal{S} | p_x > p_i\}} y_{i,x} b_x / r \; + \sum_{\{x \in \mathcal{S} | p_x = p_i\}} z_x b_x / r + \max_{\{x \in \mathcal{S} | p_x < p_i\}} \hat{\ell}_x / r \tag{2}$$

Thereby, $y_{i,x}$ and z_x represent the maximum number of bursts from higher and same class streams x that can interfere with the observed stream i. A lower bound for the current hop h_k is given by Equation 3 and 4:

$$y_{i,x} \geq \left\lceil \frac{accMaxD_x^{h_k} - accMinD_x^{h_{k-1}} + \delta_{p_i}}{\tau_x} \right\rceil \tag{3}$$

$$z_x \geq \left\lceil \frac{accMaxD_x^{h_k} - accMinD_x^{h_{k-1}}}{\tau_x} \right\rceil \tag{4}$$

Note that Equation 2 is split into three parts representing higher traffic classes $\{x \in \mathcal{S} \mid p_x > p_i\}$, interference from the same class $\{x \in \mathcal{S} \mid p_x = p_i\}$, and lower classes $\{x \in \mathcal{S} \mid p_x < p_i\}$. The transmission delay of stream i is explicitly included in these parts, as the second subset $\{x \in \mathcal{S} \mid p_x = p_i\}$ includes i itself. In addition, note that the delay bound is equal for all streams with the same traffic class p_i. It is sufficient to compute the bound only once for each supported traffic class of the bridge.

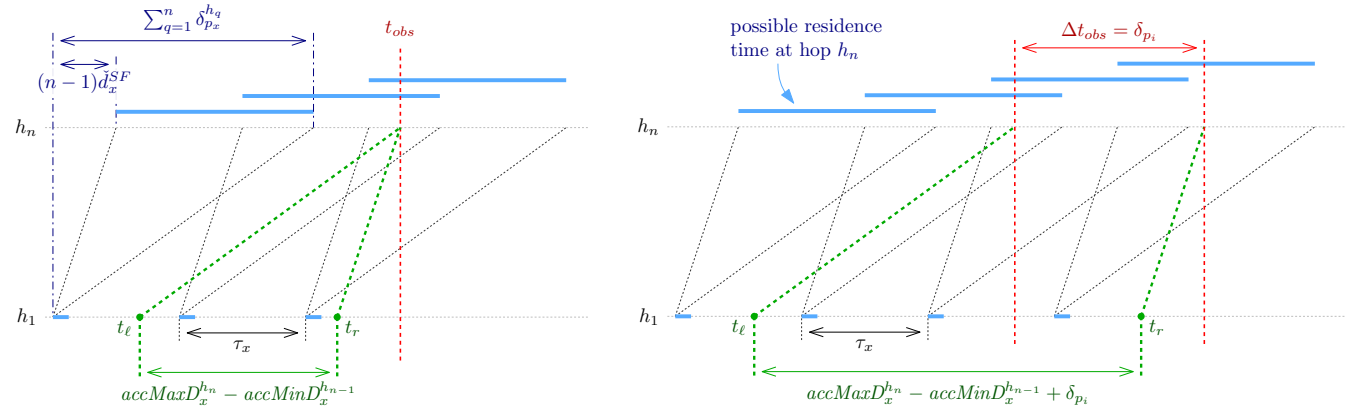
A. Proof

In this proof, the bounds from the different classes are considered individually, as each frame only belongs to a single class. The sum of these individual delay bounds represents an upper bound for the general delay $d_i^{TQ,SF}$, as the union of the three subsets represents the entire set of interfering streams $s \in \mathcal{S}$.

a) Same traffic class: Assume that the delay caused by the same class as stream i is larger than $\sum_{\{x \in \mathcal{S} | p_x = p_i\}} z_x b_x / r$. This implies that:

(1) At least one stream x exceeds its burst size b_x. This is a contradiction to Constraint 3.

(2) There was at least one stream x that had more than z_x bursts ($b_{x,1}, ..., b_{x,m}$, $m = z_x + 1$) in the same queue in the moment when f_i arrived in the bridge h_n, since the bridge is using a FIFO transmission selection algorithm as



(a) Number of bursts b_x that can be in the queue at the same time.

(b) Number of bursts b_x that can appear during the time duration δ_{p_i}.

Figure 4. Relevant time duration at hop h_1 to estimate the number of bursts from stream x in a different hop h_n.

by Constraint 2. This observed moment is referred to as t_{obs} and the situation is illustrated in Figure 4a. In order to see m frames in the queue of h_n at the moment t_{obs}, their possible residence times (i.e., the interval between their earliest moment of arrival and their latest moment of departure) must overlap at this point. This is illustrated by the two horizontal blue bars that intersect t_{obs}.

Any frame whose residence time intersects t_{obs} at h_n must have been transmitted from h_1 after the moment $t_\ell = t_{obs} - accMaxD_x^{h_n}$, otherwise it would violate Constraint 6 as the frame took longer than $\delta_{p_x}^{h_q}$ at one of the n hops. Similarly, these frames must have been transmitted before the moment $t_r = t_{obs} - accMinD_x^{h_{n-1}}$, as the minimum time needed to move from hop h_1 to h_n is $accMinD_x^{h_{n-1}} = (n-1)\check{d}_x^{SF}$.

During the interval $t_r - t_\ell = accMaxD_x^{h_k} - accMinD_x^{h_{k-1}}$, $m = z_x + 1$ bursts must have been transmitted at h_1, which means that the burst interval τ_x must have passed more than $m - 1 = z_x$ times: $(accMaxD_x^{h_k} - accMinD_x^{h_{k-1}})/\tau_x > z_x$. This is a contradiction to Equation 4, therefore the assumption must be false.

b) Higher traffic classes: Assume that the queuing delay caused by frames from higher classes is larger than $\sum_{\{x \in \mathcal{S} | p_x > p_i\}} y_{i,x} b_x / r$. This is only possible if:

(1) At least one stream x exceeds its burst size b_x. This is a contradiction to Constraint 3.

(2) There was at least one stream x of which more than $y_{i,x}$ bursts ($b_{x,1}, ..., b_{x,m}$, $m = y_{i,x} + 1$) arrived at the bridge while the observed frame f_i from stream i was in the queue. Frame f_i spent at most $\Delta t_{obs} = \delta_{p_i}$ time units in the queue, otherwise, it would violate Constraint 6. Similarly to the previous observation for the same-class interference, this time interval at hop h_n can be projected to the interval $t_r - t_\ell = accMaxD_x^{h_n} - accMinD_x^{h_{n-1}} + \delta_{p_i}$ at hop h_1. This is illustrated in Figure 4b. Any frames that arrive before t_ℓ or after t_r in h_1 cannot be in the queue of h_n during Δt_{obs}, otherwise they would violate Constraint 6 or their minimum transmission time $\check{d}_x^{SF}$.

In order to see $m = y_{i,x} + 1$ bursts during the interval $accMaxD_x^{h_n} - accMinD_x^{h_{n-1}} + \delta_{p_i}$, the time period τ_x must have passed more than $m - 1 = y_{i,x}$ times: $(accMaxD_x^{h_k} - accMinD_x^{h_{k-1}} + \delta_{p_i})/\tau_x > y_{i,x}$. This is a contradiction to Equation 3, therefore the assumption must be false.

c) Lower traffic classes: Assume that the delay caused by frames from lower classes is larger than $\max_{\{x \in \mathcal{S} | p_x < p_i\}} \hat{\ell}_x / r$. This implies that:

(1) At least one stream x exceeds its maximum frame size $\hat{\ell}_x$. This is a contradiction to Constraint 3.

(2) More than one frame from lower classes has been transmitted after f_i arrived in the queue. This is a contradiction to Constraint 1, therefore the assumption must be false.

IV. Evaluation

In this section, a brief evaluation of the accuracy of Equation 2 is presented. Section IV-A discusses potential improvements and sources of inaccuracy in the worst case estimation, while Section IV-B shows the actual difference between a simulated worst case scenario and the computed upper bound.

A. Sources of Overestimation

Equation 2 aims to represent the worst case scenario as closely as possible, but it must remain computationally feasible and work with only the available bridge-local information. Therefore, there are two main sources of inaccuracy in the formula which are discussed in the following.

a) Higher traffic classes: Equation 3 defines $y_{i,x}$ as a lower bound for the number of frames from a higher prioritized class (stream x) that can cause additional latency for frames of stream i. The rationale behind this bound is to assess the number of bursts from stream x that can appear during the entire residence time δ_{p_i}. Thereby, δ_{p_i} is used as an upper bound for the time that f_i spends in the transmission queue (d_i^{TQ}). In reality, this time may be much shorter: the pre-defined guarantee δ_{p_i} is often larger than the currently maximum achievable queuing delay d_i^{TQ}, especially in low load situations.

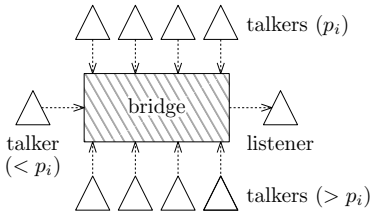


Figure 5. Evaluation topology for delay comparisons. A single bridge with a variable number of talkers and streams is considered. Every link has a data rate of 1 Gbit/s.

This upper bound is used anyway: an accurate assessment of the queuing delay d_i^{TQ} results in a recursive relation. The delay d_i^{TQ} depends on the number of possible higher class frames, which in turn depends on d_i^{TQ} itself, and resolving this equation becomes even more complex when other higher traffic class streams are considered as they depend on each other. Therefore, δ_{p_i} is used as a simple upper bound to derive a closed form for Equation 3. Future work may attempt to improve this assessment by using a different bound, or by adjusting the constraints in Section II-B.

b) Frame locality: In Equation 2, the interference of the same traffic class $\sum_{\{x \in \mathcal{S} | p_x = p_i\}} z_x b_x / r$ assumes that, in the worst case, all bursts b_x arrive at the same time, immediately before the observed frame f_i arrives. This is only possible if all bursts arrive from different input ports. In general, the arrival times are not entirely independent of each other. For example, imagine a single talker sending multiple bursts of different streams in succession. These frames would locally arrive from the same input port and have very little impact on each other. They would mainly be delayed by streams from other input ports, from a different part of the network. Depending on the topology, the assumed worst case scenario may not be able to occur.

These dependencies may be considered in the latency bound, but this requires a significant rework of the constraints, the available information from admission control, and a new approach to proof this bound. An improved bound is left for future investigations.

B. Quantitative Comparison with Worst Case Scenarios

In this section, the inaccuracies discussed earlier are quantitatively investigated. Therefore, the presented bound is compared to an artificial worst-case scenario with Strict Priority Transmission Selection (SP) in a single switch. This scenario is evaluated in a frame-level discrete event simulation, which reports the resulting queuing and transmission delays. In addition, the presented SP bound is also compared to the theoretical upper bound when using Asynchronous Traffic Shaping (ATS) instead of SP. The applied formula for the ATS bound is based on [9, Eq. 21]. Note that the simulation has only been applied with the SP algorithm, the actually achievable worst case latencies when using ATS are not reported.

a) Evaluation setup: The general scenario is depicted in Figure 5. A single talker is used to produce frames from lower traffic classes (classes $< p_i$), as only one frame from lower classes is transmitted in this scenario. In addition, a variable number of talkers contain streams with the same traffic class (p_i) and higher classes ($> p_i$), respectively. In this evaluation, only the number of streams from each class is varied, the remaining parameters (burst size b_i, burst interval τ_i) remain constant. The respective values are presented in Table II. Only a single hop is considered, and the per-hop delay bound δ_{p_i} is set to $\tau_i = 1\,\text{ms}$.

Table II
BURST AND INTERVAL PARAMETERS FOR THE THREE CONSIDERED CLASSES IN THE WORST-CASE SCENARIO.

Traffic class p_x	**Burst** b_x	**Interval** τ_x	**Per-hop delay** δ_{p_x}
lower ($< p_i$)	1500 B	100 ms	100 ms
same (p_i)	256 B	1 ms	1 ms
higher ($> p_i$)	64 B	250 µs	250 µs

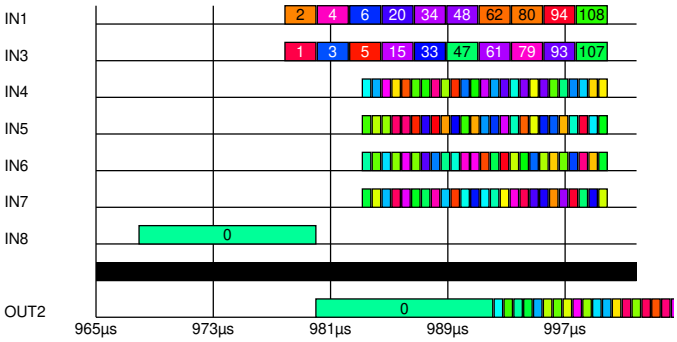


Figure 6. Example worst case scenario. Frame 108 is observed. One lower-class frame (IN8), 19 same-class frames (IN1 and IN3), and 100 higher-class frames (IN4–IN7) are transmitted before frame 108.

In the worst case simulation, all streams are automatically scheduled for the maximum possible interference with the observed frame f_i. Therefore, the *last* frame of each other talker arrives immediately before f_i, with all the other frames from the same ports arriving directly before them. The lower traffic class is scheduled to arrive first, such that its transmission begins 1 ns before the first transmission from any other talker would begin.

An exemplary worst case setup is illustrated in Figure 6. The y-axis shows different input ports (IN1 to IN8) as well as the output port OUT2. The x-axis shows an example time line where colored blocks represent the transmission time of frames from different streams. The frame with label 0 is the only low-priority frame, it arrives first and is transmitted on OUT2 immediately after that. The observed frame f_i is the last frame on port IN1 and has the label 108, it is transmitted last on OUT2 (beyond the visible extract). Frames on the ports IN1 and IN3 have the same traffic class p_i as the observed frame. The smaller frames on the ports IN4–IN7 higher class frames (100 in this figure). They are transmitted right after frame 0 in the order they arrived.

b) Higher traffic classes: First, the difference between the bounds and the simulated worst case are compared de-

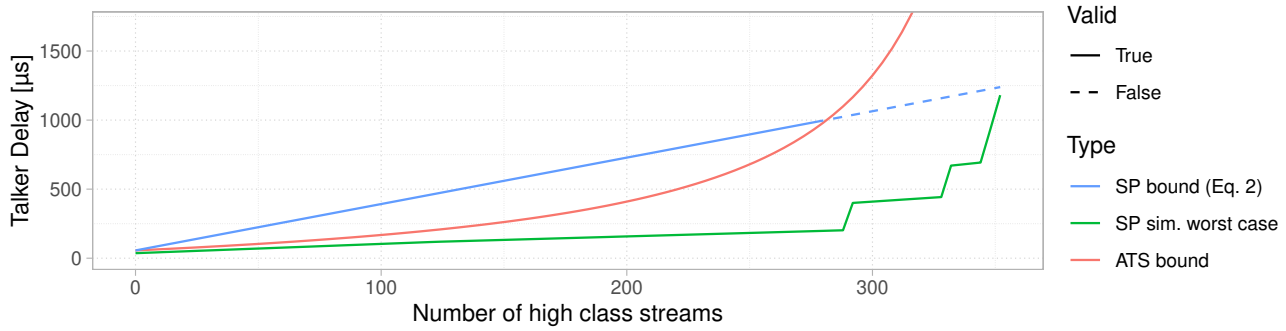


Figure 7. Comparison of the SP upper bound (Eq. 2), the ATS bound, and the simulated SP worst case latency with a varying number of high class streams.

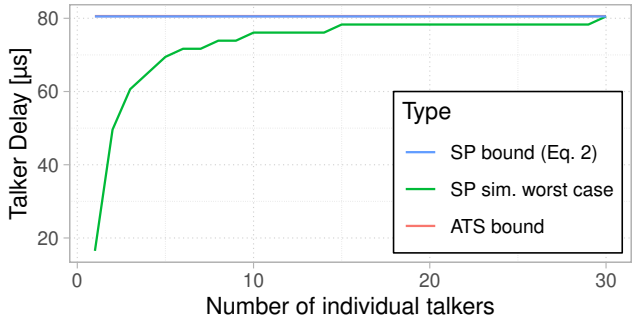


Figure 8. Comparison of the upper bound (Eq. 2) and the simulated worst case latency with a varying number of talkers (fixed total number of streams).

pending on the number of streams from higher traffic classes. Therefore, the amount of streams with the *same* traffic class is fixed at 19, whereat 9 streams arrive from the same input port as f_i, while the other 10 streams arrive from a different port. The number of *higher* traffic class streams is varied from 0 to 352. These streams are equally split on four individual input ports.

Figure 7 displays the delay values on the y-axis: SP upper bound (Eq. 2) in blue, SP worst case simulation in green, and ATS upper bound in red. The x-axis shows the number of higher traffic class streams in the network (input ports `IN4`–`IN7` in Fig. 6). As all streams are configured equally, the SP bound (blue) shows a linear increase. It always assumes that each higher class stream interferes with f_i with the maximum number of bursts $y_{i,x}$. In this scenario, Equation 3 returns $y_{i,x} = \lceil 1250/250 \rceil = 5$ higher class bursts in the worst case. However, throughout most scenario configurations (until approx. 280 high class streams), the simulated worst case curve (green) shows a much slower increase.

This effect has been discussed in Section IV-A. The queuing delay (d_i^{TQ}) is not sufficient to delay f_i long enough to meet all of the 5 estimated higher class frames during its time in the queue. Only after the number of higher class streams reaches roughly 280, d_i^{TQ} becomes large enough (almost 250 µs) to encounter the 2nd generation of high class frames, all of which are being transmitted first, which results in the steep increase of the green curve at 280 streams. Increasing the number of streams further (to almost 500 µs delay), the 3rd generation of higher class streams is encountered, resulting in another increase. Finally, at 352 higher class streams, all 5 estimated generations from each high class stream are encountered by f_i, raising its delay close to 1250 µs.

Note that the delay values at which the steps of the green curve occur are not *exactly* multiples of $\tau_x = 250$ µs. This is partly due to the other inaccuracies mentioned before. It should also be noted that, in this particular scenario, the reported delay values above $d_i = \delta_{p_i} = 1000$ µs are only considered theoretically. They are marked as *invalid* and dashed in the figure. Admission control would prevent streams from deployment beyond that point as per Constraint 6. They are included to show that, once the actual delay reaches far enough to encounter all frames predicted by Equation 3 ($y_{i,x}$), the delays of the worst case simulation reach very close to the upper bound from Equation 2.

The *buffer* created by the overestimation at lower load levels is not necessarily a bad thing: it prevents the switch from accepting too many lower class stream reservations. The lower classes may otherwise occupy a larger portion of the available bandwidth and cause the steep increase of the green curve to appear much sooner. This *linear* occupancy of the resource "delay" aids in a balanced usage of network resources and of achieved real-time throughput in a dynamic environment.

The ATS bound in red shows a much slower increase in the valid section of the SP bound. As it does not rely on the latency guarantee δ_{p_i} and can be computed independently of the admission control system, in theory, the bound remains valid beyond 1000 µs. Reshaping the traffic at every hop shows a benefit, but the gap between the two bounds closes as the number of streams increases. The non-linear increase of the ATS bound is due to the bandwidth-aware impact of higher traffic classes. The remaining bandwidth is part of the denominator, therefore the more bandwidth is consumed by higher classes, the higher the delay becomes.

c) Frame locality: In this paragraph, the influence of multiple frames arriving from the same talker (and the same ingress port) is discussed. Therefore, Equation 2 is used in a scenario with 30 other streams from the same traffic class (31 streams total). These 30 streams are first transmitted from a single ingress port (1 talker), and later equally distributed

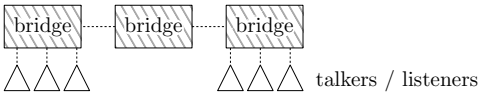


Figure 9. Evaluation topology for capacity comparisons. Talker and listener of each stream is chosen at random. Every link has a data rate of $1\,\mathrm{Gbit/s}$.

Table III
STREAM PARAMETERS FOR THE CAPACITY COMPARISON.

Traffic class p_x	**Burst** $b_x = \hat{\ell}_x$	**Burst interval** τ_x
3 (high)	128 B	250 µs
3 (high)	256 B	500 µs
3 (high)	512 B	1000 µs
2 (low)	1024 B	2000 µs
2 (low)	1522 B	4000 µs

between 2–30 talkers. In the last considered scenario, all streams arrive from a different ingress port, theoretically resulting in the maximum possible interference.

Figure 8 presents the comparison of latencies depending on frame locality. As the number of streams in this scenario does not change, Equation 2 (blue) and the ATS bound (red) always return the same latency (approx. 80 µs). Shaping has no impact here due to the absence of higher class streams, and because the number of interfering bursts from the same class is always $z_x = 1$.

As expected, the actual worst case latency from the simulation (green) is lower if more than one frame is transmitted from the same ingress port. Adding more talkers and distributing the 30 streams equally among them causes the frames to arrive more concurrently, resulting in more unfinished work when f_i enters the transmission queue. Finally, when sending each stream via its own ingress port (individual talker), the expected worst case delay of Equation 2 is met.

As mentioned earlier, including these effects in the latency bound (and its proof) requires additional information in the reservation process and is to be discussed in future work.

C. Comparison of Network Capacities

In this section, the delay bounds of the two approaches Strict Priority Transmission Selection (SP, Eq. 2) and Asynchronous Traffic Shaping (ATS) [9, Eq. 21] are not compared directly, but their impact on the network capacity when used in an admission control system is investigated. Therefore, stream reservations are simulated in the network in Figure 9. In this simulation, a varying number of streams is deployed with random properties: talker node, listener node, and one of the five stream classes from Table III are selected randomly for each stream. These streams attempt to reserve network resources (delay and bandwidth) successively. At each reservation, each of the hops compares the current delay bound d_i of each stream i with its delay guarantee δ_{p_i}. If $d_i \leq \delta_{p_i}$ still holds for the new stream and every already deployed stream i, the new reservation is accepted by the current hop. If each hop accepts the new reservation, the stream is deployed in the network, otherwise it is discarded and the next stream reservation is tested.

This evaluation reports the number of successful reservations under varying parameters. As two traffic classes are considered in Table III (3 and 2), four different configurations are tested for both delay guarantees δ_3 and δ_2, ranging from 100 µs to 8000 µs. In addition, four different numbers of total stream reservations are attempted for each delay guarantee, ranging from 100 to 2000 streams. For each such configuration, 20 repetitions with random streams are performed. The resulting mean values for the number of successfully deployed streams, as well as their 99.5% confidence intervals, are presented in Figure 10.

In the first two delay configurations ($\delta_{3,2} = (100\,\mu\mathrm{s}, 250\,\mu\mathrm{s})$ and $\delta_{3,2} = (200\,\mu\mathrm{s}, 500\,\mu\mathrm{s})$), the network capacities with SP and ATS show no significant difference, independently of the number of requested streams. This is in line with the results from Figure 7, as both bounds reached the critical delay $d_i = \delta_{p_i}$ at roughly the same time. This is the case when the delay guarantee δ_{p_x} is small enough such that $accMaxD_x^{h_k} - accMinD_x^{h_{k-1}} < \tau_x$ holds. In such cases, the number of interfering same class bursts is $z_x = 1$, the shaper shows no benefit here. However, if the delay guarantee δ_{p_i} is large enough to allow multiple bursts of the same stream in flight ($z_x > 1$), the shaper significantly outperforms the regular SP configuration when a sufficient number of streams is deployed, accepting up to 70% more streams in the last scenario ($\delta_{3,2} = (2000\,\mu\mathrm{s}, 8000\,\mu\mathrm{s})$) before the delay guarantee is hit.

V. RELATED WORK

This work is embedded in the context of IEEE 802.1Q switched Ethernet networks and relies on a priority transmission selection for coexistence with best effort streams, and on an admission control system with a distributed stream reservation procedure that resigns from central control units. However, a multitude of other switched Ethernet technologies for real-time communications were developed concurrently. The authors of [14] provide a comprehensive overview of technologies applied in automotive, aviation, and industrial networks. The most similar technique compared to strict priority transmission and self-pacing talkers is Avionics Full DupleX Switched Ethernet (AFDX). It applies a two-priority first come first served scheduling mechanism, as well as end host shaping with minimum inter-frame intervals (bandwidth allocation gap shaping) [15].

Independently of the applied technology, the computation of latency bounds in such networks is discussed on a fundamental level, and different approaches have emerged. In [16], the authors present a holistic worst case analysis of the Credit-Based Shaper and argue that the Strict Priority mechanism is a special case where the idle slope equals the link speed. [17] provides a holistic delay analysis of AFDX streams. The trajectory approach has been applied extensively to provide tighter end-to-end delays in AVB switched Ethernet networks [18], [19]. Most prominently, Deterministic Network Calculus

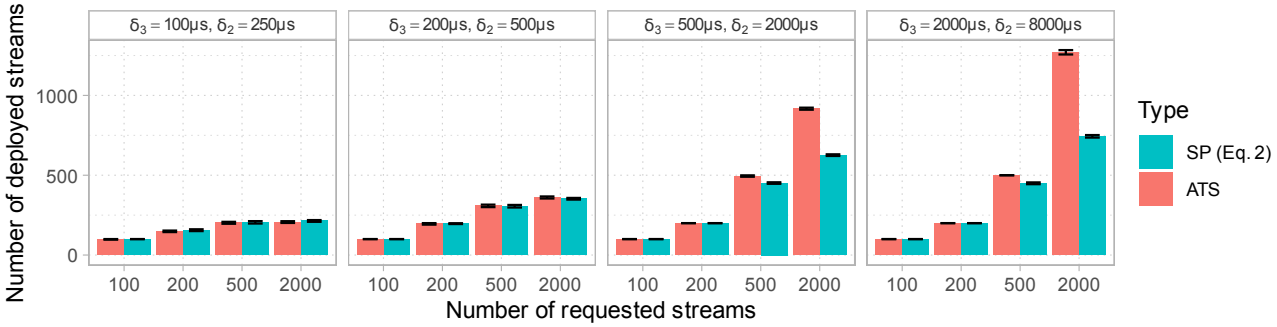


Figure 10. Comparison of network capacity with the SP bound and the ATS bound with a varying number of streams and different delay guarantees δ_{p_i}. Bars show mean values from 20 repetitions, whiskers show 99.5% confidence intervals.

(DNC) is used to compute tight end-to-end delay bounds with an increasing accuracy. These works have specialized in *separate flow analysis* [20]–[22], application of the *pay multiplexing only once* principle [23]–[25], and more sophisticated approaches to provide more accurate bounds with lower complexity [26], [27].

One key difference of these works to our proposed latency bound is the complexity and the required information. Most formal latency analyses focus on end-to-end delays. They assume network-wide knowledge of all deployed streams and their aim is to compute all delays as tightly as possible. Their high cost often renders them unfeasible [27] and while some systems are designed for dynamic workloads [28], the computation of network-wide end-to-end delays still requires a central processing unit with global information. This work presents a latency bound designed for a distributed admission control system that provides proven per-hop delay bounds with information from only bridge-local streams that does not rely on (re)shaping or timed gates.

VI. Conclusion

This paper presents a per-hop latency bound with bridge-local information for the queuing and transmission delay in an IEEE 802.1Q strict priority switch. It provides deterministic latency without the need for network-wide information or sophisticated transmission selection features in the forwarding devices. It relies on self-pacing talkers, priority transmission selection, and an admission control system that communicates accumulated minimum and maximum latencies. A formal proof shows that there are no edge cases similar to the Deggendorf use case [5] that would allow frames to exceed their bounds.

A comparison of the latency bound with simulations shows that under regular usage, the reported bound is very close to the actual worst case situation. Only under skewed traffic mixes with large numbers of higher priority streams, or if a large number of streams originates from the same input port, the latency bound is more pessimistic. Even in such situations, it performs comparable to a sophisticated per-hop shaping mechanism, and in scenarios with small delay guarantees, it can even be used without penalty.

The presented delay model can be further extended and improved by future investigations. A similar methodology can be applied to derive much lower per-hop latency bounds with the use of timed gates (802.1Q [10, Section 8.6.8.4]), given additional constraints in the network. Including more information during the reservation may allow to take frame locality into account to address the inaccuracies caused by multiple frames arriving in succession. Finally, future work may derive a thorough comparison for different bounds and their requirements, including Asynchronous Traffic Shaping (802.1Qcr) [8], [9], the Credit-Based Shaper [2], and timed gates to determine which mechanism is best suited for which situation.

References

[1] IEEE, *Audio Video Bridging Task Group*, 2012. [Online]. Available: http://ieee802.org/1/pages/avbridges.html (visited on 01/13/2020).

[2] "IEEE Standard for Local and Metropolitan Area Networks – Virtual Bridged Local Area Networks – Amendment: Forwarding and Queuing Enhancements for Time-Sensitive Streams", *IEEE Std 802.1Qav-2010*, 2010.

[3] "IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 14: Stream Reservation Protocol (SRP)", *IEEE Std 802.1Qat*, 2010.

[4] "IEEE Standard for Local and metropolitan area networks – Audio Video Bridging (AVB) Systems", *IEEE Std 802.1BA-2011*, pp. 1–45, 2011.

[5] C. Boiger, "Class A bridge latency calculations", in *IEEE 802 November Plenary Meeting*, 2010.

[6] IEEE, *Time-Sensitive Networking (TSN) Task Group*, 2020. [Online]. Available: https://1.ieee802.org/tsn/ (visited on 01/13/2020).

[7] "IEEE Standard for Local and Metropolitan Area Network – Bridges and Bridged Networks – Amendment 25: Enhancements for Scheduled Traffic", *IEEE Std 802.1Qbv-2015*, 2015.

[8] "Draft Standard for Local and Metropolitan Area Networks — Bridges and Bridged Networks — Amendment: Asynchronous Traffic Shaping", *IEEE P802.1Qcr/D0.5*, 2018.

[9] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched ethernet networks", in *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, Jul. 2016, pp. 75–85.

[10] "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.

[11] "IEEE Standard for Local and Metropolitan Area Networks – Bridges and Bridged Networks – Amendment: Resource Allocation Protocol (RAP)", *IEEE Std P802.1Qdd*, 2019.

[12] J. Turner, "New directions in communications (or which way to the information age?)", *IEEE communications Magazine*, vol. 24, no. 10, pp. 8–15, 1986.

[13] A. S. Tanenbaum, "Computer networks, fourth edition", *Prentice Hall PTR*, 2003.

[14] X. Li and L. George, "A survey of switched ethernet solutions for real-time audio/video communications", in *Building Wireless Sensor Networks*, Elsevier, 2017, pp. 1–30.

[15] F. He, L. Zhao, and E. Li, "Impact analysis of flow shaping in ethernet-avb/tsn and afdx from network calculus and simulation perspective", *Sensors*, vol. 17, no. 5, p. 1181, 2017.

[16] J. Diemer, D. Thiele, and R. Ernst, "Formal worst-case timing analysis of ethernet topologies with strict-priority and avb switching", in *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*, IEEE, 2012, pp. 1–10.

[17] J. J. Gutiérrez, J. C. Palencia, and M. G. Harbour, "Holistic schedulability analysis for multipacket messages in afdx networks", *Real-Time Systems*, vol. 50, no. 2, pp. 230–269, 2014.

[18] S. Martin, P. Minet, and L. George, "The trajectory approach for the end-to-end response times with non-preemptive fp/edf", in *International Conference on Software Engineering Research and Applications*, Springer, 2004, pp. 229–247.

[19] X. Li and L. George, "Deterministic delay analysis of avb switched ethernet networks using an extended trajectory approach", *Real-Time Systems*, vol. 53, no. 1, pp. 121–186, 2017.

[20] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer Science & Business Media, 2001, vol. 2050.

[21] K. Lampka, S. Bondorf, and J. Schmitt, "Achieving efficiency without sacrificing model accuracy: Network calculus on compact domains", in *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, IEEE, 2016, pp. 313–318.

[22] A. Bouillard, "Algorithms and efficiency of network calculus", *École Normale Supérieure*, 2014, Habilitation thesis.

[23] M. Fidler, "Extending the network calculus pay bursts only once principle to aggregate scheduling", in *International Workshop on Quality of Service in Multiservice IP Networks*, Springer, 2003, pp. 19–34.

[24] H. Schiøler, J. J. Jessen, J. D. Nielsen, and K. G. Larsen, "Network calculus for real time analysis of embedded systems with cyclic task dependencies.", in *Computers and Their Applications*, Citeseer, 2005, pp. 326–332.

[25] J. B. Schmitt, F. A. Zdarsky, and I. Martinovic, "Improving performance bounds in feed-forward networks by paying multiplexing only once", in *14th GI/ITG Conference-Measurement, Modelling and Evalutation of Computer and Communication Systems*, VDE, 2008, pp. 1–15.

[26] A. Bouillard and É. Thierry, "Tight performance bounds in the worst-case analysis of feed-forward networks", *Discrete Event Dynamic Systems*, vol. 26, no. 3, pp. 383–411, 2016.

[27] S. Bondorf, P. Nikolaus, and J. B. Schmitt, "Quality and cost of deterministic network calculus: Design and evaluation of an accurate and fast analysis", *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, p. 16, 2017.

[28] K. Jang, J. Sherry, H. Ballani, and T. Moncaster, "Silo: Predictable message latency in the cloud", *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 435–448, 2015.