Analyzing and Modeling the Latency and Jitter Behavior of Mixed Industrial TSN and DetNet Networks

Lukas Wüsteney
Esslingen University; Belden Inc.

David Hellmanns
University of Stuttgart; Belden Inc.

Markus Schramm
Esslingen University

Lukas Osswald
University of Tübingen

René Hummen
Belden Inc.

Michael Menth
University of Tübingen

Tobias Heer
Esslingen University

ABSTRACT

Today, industrial real-time communication is commonly designed based on two key principles to satisfy the challenging Quality of Service (QoS) requirements of industrial applications: a) local communication and b) purpose-built networks. IEEE Time-Sensitive Networking (TSN) and IETF Deterministic Networking (DetNet) promise to lift these two limitations. This facilitates the transformation of previously loosely integrated automation network parts from isolated, purpose-built real-time networks to more tightly integrated, open, multi-purpose networks of networks. With TSN and DetNet, each of these interconnected networks, e.g., machine or backbone networks, can and will be fined-tuned for optimal performance regarding the different real-time applications located inside them. The resulting patchwork of DetNet-connected TSN networks, however, creates a challenge for cross-network real-time communication: predicting QoS properties, such as the end-to-end latency. To address this challenge, we propose a model that allows calculating best-case and worst-case latencies for time-critical communication across different DetNet-connected TSN networks. This enables validating end-to-end communication requirements in open, multi-purpose industrial networks. Our evaluation with real industrial hardware shows the applicability of our proposed model.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

TSN, DetNet, Worst-Case Model, Latency Model, Real Time

ACM Reference Format:
Lukas Wüsteney, David Hellmanns, Markus Schramm, Lukas Osswald, René Hummen, Michael Menth, and Tobias Heer. 2022. Analyzing and Modeling the Latency and Jitter Behavior of Mixed Industrial TSN and DetNet Networks. In *The 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '22), December 6–9, 2022, Roma, Italy.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/3555050.3569138

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CoNEXT '22, December 6–9, 2022, Roma, Italy
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9508-3/22/12…$15.00
https://doi.org/10.1145/3555050.3569138

1 INTRODUCTION

In the past, time-critical communication in industrial applications mainly used restricted and well-defined special-purpose networks for single machines or production lines. To achieve the challenging real-time communication requirements of Industrial Control Systems (ICS), machine or production line networks used one of many vendor-specific Ethernet dialects, e.g., PROFINET, SERCOS III, EtherCAT, or CC-Link [24]. With the introduction of Time-Sensitive Networking (TSN) [4], IEEE Ethernet became a vendor-independent real-time communication standard promising more compatibility and less fragmented networks for real-time traffic in factories.

TSN standardizes a variety of Quality of Service (QoS) mechanisms, providing different levels of guarantees from which the vendors of production lines and machine builders can select to provide deterministic communication latency bounds required for the respective industrial application. The TSN mechanisms range from traffic-class-based bandwidth reservation to per-stream QoS policies, including a Time Division Multiple Access (TDMA) scheme. With an appropriate selection of mechanisms, a single TSN network can be fine-tuned to the right level of real-time requirements down to the order of microsecond guarantees. To operate properly, most of these mechanisms require that the switches are similarly configured and tightly coordinated. Such a coordinated TSN network or network segment, in which a specific configuration exists, including a single time source and a common set of mechanisms, is called a *TSN domain*.

Typically, industrial networks consist of components from multiple parties: component vendors supply TSN-capable components, machine builders combine these components to machines or cell networks, and the factory backbone network connects all machine and cell networks in the factory. Each party strives to choose the best QoS mechanisms for the task at hand, e.g., building a set of synchronized machines as part of a production line or implementing a local safety feedback loop. This leads to different requirements for the connected TSN domains, which in turn leads to heterogeneous coexisting network configuration schemes as well as different employed TSN mechanisms. The TSN standard does not enforce uniform network configurations, and the Deterministic Networking (DetNet) IETF standards [17] even facilitate the connection of different TSN domains. However, applications using such an ICS network require the worst-case latency to be within certain bounds. Yet, to the best of our knowledge, there are no models to predict the end-to-end latency bounds for a stream that traverses multiple different and potentially not well-coordinated TSN domains.

©ACM 2022. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *The 18th International Conference on emerging Networking EXperiments and Technologies*, Roma, Italy, Dec. 6–9, 2022 The original publication is available at ACM Digital Library: https://doi.acm.org/10.1145/3555050.3569138

The goal of this work is to study the effects of the combination of different TSN mechanisms on traffic traversing different TSN domains. Based on our observations, we build a model that allows for determining best-case and worst-case latency bounds for inter-domain TSN streams. We use this model to calculate end-to-end delays for sample networks and compare them to delays measured on a testbed consisting of industrial TSN switches.

The model is useful to assess the QoS properties of interdomain TSN traffic in future ICS, to detect potential bottlenecks, and to choose appropriate QoS mechanisms. We open-source the code of our model as a tool for researchers and ICS network designers.[1]

This paper is structured as follows: First, we highlight how TSN and DetNet are used in industrial networks and provide background information on TSN and DetNet in Section 2 before we discuss related work in Section 3. Section 4 summarizes the problem statement. In Section 5 and 6, we introduce a delay and bandwidth model for a single TSN node and extend it to paths of multiple nodes as well as to networks with interfering TSN streams. Finally, Section 7, shows our evaluation and the applicability of the model before Section 8 concludes the paper.

2 TSN AND DETNET IN ICS

Concepts like Industry 4.0 and Smart Factory change the way industrial applications are designed. More and more components that used to be located in close proximity to the controlled processes on the factory floor are being virtualized and centralized in local data centers. Even controllers that control fast-motion applications and monitor process parameters can now be hosted as virtual instances, so-called virtual Programmable Logic Controllers (vPLCs), which save costs, improve scalability, and increase the reliability of critical control functions. Due to the relocation of the control function to a centralized location, real-time communication between vPLCs and field devices, e.g., sensors and actuators, must be possible across the entire factory network. Hence, future industrial networks must support local control loops within a machine as well as real-time control loops that traverse large parts of the factory network on the way to and from the local data center.

This section gives an overview of future industrial networks based on DetNet-connected TSN networks. We then provide technical details on traffic in industrial networks and the TSN mechanisms that are relevant to our work.

2.1 Future Industrial Networks

Industrial control systems follow a hierarchical structure. Figure 1 shows an example of such an ICS with one isolated machine and two multi-purpose machines based on DetNet-connected TSN networks. Today, machine networks are isolated, purpose-built, and require vendor-specific Ethernet dialects for their time-critical communication (cf. Fig. 1 Isolated Machine). The backbone does not carry any time-critical communication today. In the future, with TSN and DetNet, machine networks are no longer dedicated to one single machine purpose, and backbone networks can serve time-critical traffic. TSN replaces the Ethernet dialects in machine networks (cf. Fig. 1 TSN Machines A and B). However, the specific

[1]Code is available at https://github.com/hs-esslingen-it-security/hses-detnet-tsn-latency-jitter-model/

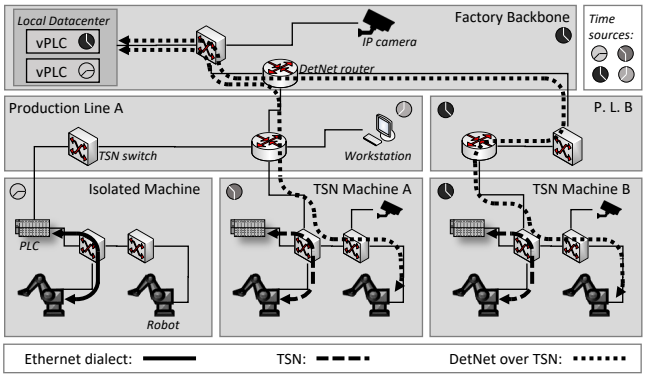


Figure 1: Example of a next-generation Industrial Control System (ICS) using TSN and DetNet.

and individual QoS requirements in the machines result in specific parameterizations of the TSN networks. The backbone networks profit from the standardized next-generation Ethernet with TSN to benefit time-critical traffic. The DetNet architecture, defined by the IETF [17], enables the expression of QoS requirements between independent TSN networks. Hence, communication with end-to-end QoS requirements is possible with DetNet routers connecting independent TSN networks (cf. Fig. 1 TSN Machine B, Production Line B, and Factory Backbone).

2.2 Traffic Types in Industrial Networks

Traffic in industrial environments is differentiated based on their periodicity or the QoS requirements of the applications. For example, motion control applications require periodic traffic and have time-bound QoS requirements. The periodic traffic specifies a fixed pattern of frame sizes and the number of frames per period. In contrast, non-periodic traffic does not have time-bound QoS requirements or fixed traffic patterns. An example of non-periodic traffic is the transfer of a firmware update to a PLC.

To address the different needs of the different types of traffic in a network, IEEE Ethernet supports eight priorities ranging from 0 to 7. Typically, an Ethernet switch has a dedicated egress queue per Layer-2 priority to enable non-blocking buffer structures and efficiently serve high-priority frames before low-priority frames are served. These Layer-2 priorities are encoded in the PCP value of the VLAN tag and, therefore, only present within one Layer-2 network. To enable end-to-end QoS across multiple TSN domains, *DetNet over TSN* [33] defines the routing of high-priority traffic, preserving its Layer-2 priority values between TSN networks. In the following section, we introduce the different TSN QoS mechanisms and how they prioritize the frames of a stream based on this priority value.

2.3 TSN Mechanisms and Parameters

TSN is a family of standards developed in the TSN Task Group of the IEEE 802.1 group. TSN defines several mechanisms that can be used in different application scenarios. In the following, we briefly present the TSN mechanisms that are part of the industrial TSN profile IEC/IEEE 60802 [6] and, thus, are relevant for ICSs.

2.3.1 Strict Priority. Strict Priority (SP), as specified in IEEE 802.1Q [4], is the default transmission selection algorithm in Ethernet networks. At each egress port, the Ethernet switch selects the next frame for transmission based on its priority. However, each transmission is finished first before the next frame is selected for transmission. This leads to undesired delays for high-priority frames if the transmission of a low-priority frame is already in progress. The transmission duration, and therefore the delay for high-priority frames, depends on the frame size of the low-priority frames and the link speed.

2.3.2 Frame Preemption. Frame Preemption (FP), as standardized in IEEE 802.1Qbu [2] and IEEE 802.3br [7], enables the preemption of low-priority transmissions to prefer high-priority frames. The preempted transmission continues once the high-priority transmission is complete. To determine which frames can preempt other frames, the network administrator assigns each of the eight Ethernet priorities to the express or preemptible class. Frames with a priority in the express class can preempt frames with a priority in the preemptible class and are served with a higher priority than all frames with preemptible Ethernet priorities.

Whether or not preemptible frames can be preempted also depends on the remaining bytes to be sent, i.e., frames with a remaining size of less than 124 B cannot be preempted. Compared to SP, this reduces the worst-case queuing time of an express frame waiting for the transmission duration of a preemptible frame from 1,522 B to 123 B, i.e., to less than a tenth.

2.3.3 Time-Aware Shaper. The Time-Aware Shaper (TAS) mechanism, as specified in IEEE 802.1Qbv [3], adds a Time Division Multiple Access (TDMA) scheme to each egress queue. The TAS mechanism implements a gate per egress queue, which can be either open or closed, to allow or disallow transmission from a queue. Thereby, the Gate Control List (GCL) is a list of tuples storing the gate states, i.e., the open and close state for each gate and the duration for this gate state. The TAS executes the GCL in a cyclic manner, i.e., it starts with the first entry and continues the execution of the list until the configured cycle time is reached. In the following, we refer to the period in which the gate for a certain priority is open as the *TAS window* for this priority. As the TAS mechanism is an optional add-on for TSN networks, all priorities within a TAS window are prioritized through SP or FP. The benefit of the TAS mechanism is exclusive access to the medium. However, if a frame arrives while the gate of its priority is closed, it must wait, even if no other transmission is in progress.

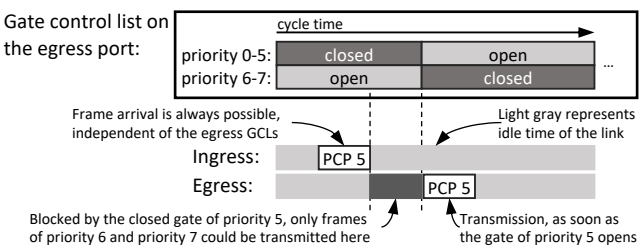


Figure 2: Operation of the Time-Aware Shaper (TAS).

Figure 2 shows the operation of the TAS. For each of the eight egress queues, gates open and close over time. In the figure, a frame with PCP5 arrives and meets an empty queue, but it cannot be sent immediately due to a closed gate for priority 5.

2.3.4 Time Synchronization. Time synchronization between two devices enables the synchronization of their TAS mechanism and their applications, i.e., control loops. A TSN domain, as defined by the IEC/IEEE 60802 [6], requires time synchronization with IEEE 802.1AS [5] between all of the devices belonging to this domain. The synchronization between TSN domains is optional and not common. Typically, TSN domains are designed to be self-contained, i.e., they have an independent time source to ensure a reliable operation of the control tasks without external influences. This paper is the first to model the interaction between unsynchronized domains, as this directly influences the calculation of end-to-end latencies in heterogeneous network deployments.

3 RELATED WORK

This section reviews related work regarding schedule optimization, modeling approaches for time-critical networks, and Network Calculus (NC).

3.1 Schedule Optimization

The research community on calculating TAS or TDMA schedules for networks, in general, is very active. We distinguish their work in pure schedule calculation [14, 15, 19–21, 23, 30, 32] and combined calculation of scheduling and routing [16, 22, 31]. These approaches calculate new schedules for existing topologies. In contrast, we assume that both the network topology as well as the schedules are defined beforehand and cannot be changed. Hence, they are applied while the network configuration can still change while our work becomes relevant when the networks and configurations are already in place and operational.

3.2 Modeling Approaches

Different approaches exist to verify that a given network configuration, e.g., a calculated schedule, achieves the QoS requirements.

This work is relevant for domains with strict verification requirements, such as the aviation industry. There are many approaches and optimizations for Aviation Full-Duplex (AFDX) Ethernet networks and their configurations. The components of an aircraft are usually specifically built or customized to be used in a particular combination and configuration. Thus, AFDX networks are comparable to single TSN machine networks, as they are self-contained and uniform, which distinguishes them from industrial networks. Therefore, the modeling of AFDX networks assumes aligned configuration and time synchronization [8, 9]. Adnan et al. [8, 9] perform worst-case delay analysis using timed automata for that domain. Charara et al. [13] conduct a comparison of network calculus, queuing network simulation, and model checking for AFDX networks.

In contrast, for TSN networks, configuration verification is not applied at a large scale yet. Different approaches, like response-time analysis and machine learning, enable the modeling of TSN QoS mechanisms on single nodes. The related work based on response-time analysis ranges from analyzing AVB networks using the Credit-based Shaper [12] to scheduled traffic with multi-level

frame preemption [11, 29]. Mai et al. analyze the feasibility of TSN network configurations with machine learning [27] and improve their method in subsequent work using graph neural network [26]. However, these approaches do not consider mixed deployments, as required for our model.

We discuss the modeling approach Network Calculus, often used for the delay analysis of TSN networks, in the next section.

3.3 Network Calculus (NC)

NC is a mathematical framework based on the min-plus algebra to compute tight upper delay bounds in packet-based networks. To achieve those tight bounds, NC requires models that are specific to combinations of forwarding mechanisms. This requires a high effort when various models should be investigated. Only a few such combinations are investigated. As a concrete example, Zhao et al. [34] provide an NC model for the TAS configured on the highest priority and the Credit-Based Shaper (IEEE 802.1Qav [1]) for other priorities. In a comprehensive literature review, Maile et al. [28] showed that, e.g., a combination of TAS and frame preemption is still missing, which is covered by our approach. Therefore, current NC models are not sufficient to investigate all combinations of forwarding mechanisms we consider in our study. A drawback of our approach is that the delay bounds are less tight than those of NC, which is acceptable for studying technical design options. We expect that NC research will extend its set of covered combinations of forwarding mechanisms in the future, but each of them requires substantial effort.

4 PROBLEM STATEMENT

Due to the individual specialized purposes of different TSN domains as well as the hierarchical structure of an ICS, an aligned TSN configuration cannot be expected. Moreover, the different networks are configured by different parties and are set up at different times. This limits the possibility of changing the configurations of these networks to achieve specific end-to-end QoS properties for the industrial applications using them. However, without the possibility of changing the configuration, predicting the QoS capabilities of the network is of paramount importance to ensure the safe operation of the application. Yet, approaches to model and predict QoS properties of TSN streams that cross multiple differently configured TSN domains are missing. In the following, we provide two examples to illustrate the challenges for predicting QoS parameters, i.e., latency and jitter, when traffic traverses different TSN domains.

As a first example, we consider an FP-based domain that transmits data into a TAS domain. The time a stream arrives at the second domain depends on the interference with other express streams and the number of preemptions. Without a model that considers these effects and calculates the interference between the streams, it is difficult to tell if an express stream frame will hit or miss the next TAS window. Missing the gate creates significant jitter and may produce undesirable effects in industrial applications.

A second example of difficult-to-predict behavior is two neighboring TAS domains that use different time sources, e.g., one time source for the factory backbone network and one provided as part of a machine. Even if the configuration of the neighboring domains is well aligned, i.e., the TAS windows are planned to open at the same time, clock drift of the time sources may lead to situations that are difficult to foresee. To illustrate the gravity of this problem, we refer to the oscillators of our evaluation switches [10], which have a precision of ±2.5 ppm, i.e., a worst-case clock drift of 2.5 µs per second. For a cycle time of 1 ms, this precision results in a theoretical drift of one cycle time in only 6.7 minutes. In our lab, we observed this effect every 4.5 hours. From the perspective of the first switch, the TAS window of the second switch may open at any time in the cycle time. The application experiences high jitter every 4.5 hours when the gates are just aligned or misaligned. Predicting such behavior is important because sporadic errors are extremely difficult to debug in complex industrial applications.

To address the issues highlighted in both examples, we develop a model to predict all relevant effects for streams traversing multiple TSN domains. The model must cover all relevant combinations of different TSN mechanisms, mechanism parametrization, and different time sources. In addition, the model must consider all periodic traffic streams and their interactions.

5 NODE MODEL

This section introduces a model to compute best-case and worst-case delay bounds for a stream s passing through a single TSN switch or DetNet router. As both devices are required to have an upper-bound processing delay and can implement the same TSN mechanisms, the model does not differentiate between these two and is applicable to both. The model calculates the best-case and worst-case delays for a *stream of interest* s based on the bandwidth requirements of the applications. First, we explain the assumptions of the model, defining the scope for the applicability. Afterward, we introduce the different delay components of a single forwarding node for the stream of interest. We conclude this section with a complete forwarding delay model for s on a single TSN node v. In Section 6, we extend this model from a single node to a sequence of nodes so that we can cover the whole path of a stream, including TSN switches and DetNet routers using unaligned configurations.

5.1 Model Assumptions

The different TSN mechanisms enable various configuration options, potentially leading to very complex configurations. We reduce the degree of complexity of our model by making some limiting assumptions. In the following, we explain our four basic assumptions and highlight why they pose no major limitations for the applicability of the model to industrial use cases.

A1) Knowledge of High-Priority Streams: The model assumes that all high-priority streams with the same or higher priority as the stream of interest are known. Yet, the model does not require knowledge of streams with lower priority or of streams belonging to the best-effort class. This assumption does not limit the applicability of the model because industrial applications using TSN are engineered, and all high-priority streams are typically known, while only best-effort traffic may occur spontaneously. Hence, this assumption matches the reality of industrial TSN networks well.

A2) One TAS Window per Cycle Time: Our model assumes that during one cycle time only one TAS window is available for the stream of interest. This limits the possible GCL configurations for the priority of the stream of interest to periodic TAS windows

Table 1: Nomenclature

Variable	Unit	Description
$\mathcal{V}$	-	Vertices of the graph
v	-	$v \in \mathcal{V}$ (switch, router, end device)
$\mathcal{E}$	-	Unidirectional edges of the graph
e	-	$e \in \mathcal{E}$ (link between devices)
c_e	B/s	Bandwidth of link e
$\mathcal{P}^e_{\text{express}}$	-	Priorities in the express class on link e
$\mathcal{S}$	-	All streams in the network
$\mathcal{S}^e$	-	All streams on link e
$\mathcal{S}^{e,s}_X$	-	All streams on e that fulfill X with respect to s
$Path_s$	$[(v,e)]$	Path of s from sender to receiver
b_s	B	Layer-2 frame size of s
p_s	-	Priority of s
CT^s_{app}	ns	Application cycle time
$CT^{e,s}_{\text{GCL}}$	ns	TAS cycle time
$w^{v,s}_{\text{trans}}$	ns	Transmission window of s on node v
$w^{v,s}_{\text{arriv}}$	ns	Arrival window of s on node v
t_X	ns	Point of time at which event X occurs
d^v_Y, $d^{e,s}_Y$	ns	Delay caused by Y, specific to v, e, and s
j^v_Y, $j^{e,s}_Y$	ns	Jitter for delay d_Y, specific to v, e, and s

with equisized durations. While this prevents the modeling of non-periodic, non-equidistant, or non-equisized TAS window configurations, such configurations are of limited use for priority traffic in an industrial real-time network because control traffic is periodic in most cases. Hence, we expect this limitation to only marginally affect the applicability of the model in industrial networks.

A3) Processing within one Cycle Time: The model assumes that a switch processes each stream within one network cycle time for TAS or one application cycle time if the TSN domain does not use the TAS mechanism. This means that we assume that no frames will be buffered for transmission in the next cycle time. Although this assumption seems quite limiting at first, commercially available industrial switches do not have sufficient buffer space to buffer across cycle times. For example, industrial switches have buffer sizes to store traffic for 25 µs to 200 µs at 1 Gbit/s per port [18]. Hence, designing a TSN network to rely on buffering by switches would lead to pathologic effects like packet loss as well as implementation- and vendor-dependent failures. Therefore, this assumption does not limit the applicability of the model for well-planned networks.

A4) Interference of Streams: To simplify the handling of interfering streams, the model assumes that all streams within a TAS window or on an SP and FP link can interfere with each other in the worst case. This simplification leads to a pessimistic prediction of the worst-case delays between two synchronized TSN domains in which streams are timed to not interfere with each other. However, we assume most independent TSN domains in industrial networks are unsynchronized or use different time sources and, therefore, are not affected by this simplification. In comparison to A1) - A3), this assumption does not limit the applicability of the model but simplifies the calculation at the cost of higher worst-case bounds in tightly synchronized cases.

5.2 Delay Model of a Forwarding Node

Figure 3 illustrates the inner workings of a single switch, including all relevant delays and events for our model. In the following, we describe the calculation of these delays individually, based on the nomenclature introduced in Table 1. We start with the hardware-dependent delays: processing, propagation, and transmission delay. Afterward, we introduce the delays that depend on the TSN configuration and other traffic: interference, blocking, and gate delay. Next, we describe the combination of these three delays, the queuing delay. We conclude this section with a definition of the impact of time synchronization. Section 5.3 covers the calculation of the total best-case and worst-case delay for one TSN node.

Table 2: Example transmission delays.

Size	100 Mbit/s	1 Gbit/s	2.5 Gbit/s	10 Gbit/s
64 B	6.7 µs	672 ns	269 ns	67 ns
123 B	11.8 µs	1.2 µs	470 ns	118 ns
1,522 B	123.4 µs	12.3 µs	4.9 µs	1.2 µs

5.2.1 Processing Delay d^v_{proc}. The processing delay d^v_{proc} is the time v needs to process a frame. It is node-specific and depends on the hardware and software implementation. We measured the processing delays for two industrial Layer-2 switches and obtained 1050 ns and 1550 ns, respectively. We use the first switch for the evaluation in Section 7.

The processing delay is not constant and is subject to a Gaussian distribution. Therefore, processing jitter j^v_{proc} is symmetric in the range of 50 ns and 190 ns for the two switches, respectively. Thus, the worst-case observed processing delay is $d^v_{\text{proc}} + j^v_{\text{proc}}$, whereas the best-case processing delay is $d^v_{\text{proc}} - j^v_{\text{proc}}$.

5.2.2 Propagation Delay d^e_{prop}. The propagation delay d^e_{prop} is the duration a signal travels on the wire. It is proportional to the cable length and medium-specific, e.g., for copper, d^e_{prop} is 5 ns/m.

5.2.3 Transmission Delay $d^{e,s}_{trans}$. The transmission delay $d^{e,s}_{\text{trans}}$ describes the time to send the content of a frame on the wire. Thus, the delay is link-speed- and frame-size-dependent. We express it as $d^{e,s}_{\text{trans}} = (20\text{ B} + b_s)/c_e$. Thereby, we also consider the Layer-1 overhead of 20 B for the inter-frame gap, preamble, and start frame delimiter. For reference, Table 2 lists the transmission delays for selected frame sizes and bandwidths. The smallest possible Layer-2 frame size is 64 B. The minimum non-preemptible frame size is 123 B. The maximum Ethernet frame size is 1,522 B.

5.2.4 Interference Delay $d^{e,s}_{ifr}$. The interference delay is caused by all other streams on link e with the same or higher priority than s. Formally, we express the set of interfering streams $\mathcal{S}^{e,s}_{\text{ifr}}$ as

$$\mathcal{S}^{e,s}_{\text{ifr}} = \{g | g \in \mathcal{S}^e \wedge g \neq s \wedge p_g \geq p_s\} \quad (1)$$

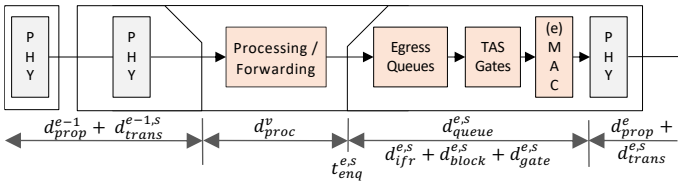


Figure 3: Delay model of a forwarding node with one ingress and one egress port.

Additionally, if frame preemption is enabled and $s \in \mathcal{P}^e_{\text{express}}$, we can reduce this set to:

$$\mathcal{S}^{e,s}_{\text{ifr}} = \{g | g \in \mathcal{S}^e \wedge g \neq s \wedge p_g \geq p_s \wedge p_g \in \mathcal{P}^e_{\text{express}}\} \quad (2)$$

For the TAS mechanism, we reduce the set of interfering streams to the streams allowed within the TAS window ($\mathcal{S}^{e,s}_{\text{window}}$):

$$\mathcal{S}^{e,s}_{\text{ifr}} = \{g | g \in \mathcal{S}^e \wedge g \neq s \wedge p_g \geq p_s \wedge g \in \mathcal{S}^{e,s}_{\text{window}}\} \quad (3)$$

As the different TSN mechanisms can be used in combination, the reductions applied in Equations 1-3 may apply in combination, producing a subset of potentially interfering streams which we denote as $\mathcal{S}^{e,s}_{\text{ifr}}$.

In the worst case, the switch v transmits all of these other streams first, and stream of interest s has to wait for $d^{e,s}_{\text{ifr}}$. As discussed in Section 5.1, all streams influence the worst-case required bandwidth for s based on their frame size and cycle time. Therefore, we model the interference delay as follows:

$$d^{e,s}_{\text{ifr}} = \sum_{g}^{\mathcal{S}^{e,s}_{\text{ifr}}} \lceil CT^s_{\text{app}} / CT^g_{\text{app}} \rceil \cdot d^{e,g}_{\text{trans}} \quad (4)$$

Typically, inter-domain streams share segments of their path, e.g., they travel along the same path in a transit domain. These streams do not interfere on all hops but only when they first meet on the path since their transmission is serialized after the first common hop. For e, these interfering streams following the same *path segment* can be defined with the help of the stream set on the previous edge $e-1$:

$$\mathcal{S}^{e,s}_{\text{ifrpath}} = \{g | g \in \mathcal{S}^{e,s}_{\text{ifr}} \wedge g \neq s \wedge g \in \mathcal{S}^{e-1,s}_{\text{ifr}}\} \quad (5)$$

Similarly, we can define the set of interfering streams that arrive at different ingress ports and share the same egress port as follows:

$$\mathcal{S}^{e,s}_{\text{ifrcross}} = \{g | g \in \mathcal{S}^{e,s}_{\text{ifr}} \wedge g \neq s \wedge g \notin \mathcal{S}^{e-1,s}_{\text{ifr}}\} \quad (6)$$

For both interference subsets $\mathcal{S}^{e,s}_{\text{ifrcross}}$ and $\mathcal{S}^{e,s}_{\text{ifrpath}}$, we define $d^{e,s}_{\text{ifrcross}}$ and $d^{e,s}_{\text{ifrpath}}$ in analogy to $d^{e,s}_{\text{ifr}}$ (cf. Equation 4).

5.2.5 Blocking Delay $d^{e,s}_{block}$. The worst-case blocking delay $d^{e,s}_{\text{block}}$ is the time the stream of interest s waits for its transmission on link e while a lower-priority transmission blocks the link. In a TAS window, only a subset of priorities is eligible for transmission. If there are no streams with a lower priority than p_s in $\mathcal{S}^{e,s}_{\text{window}}$ and in the best case, the blocking delay for s is zero, and only interference delay is relevant (cf. 5.2.4). When using FP, the blocking delay for the lowest priority in the express category is, at most, the transmission delay of one 123 B frame. In all other cases, the blocking delay for s is the transmission delay of a 1,522 B frame.

5.2.6 Gate Delay $d^{e,s}_{gate}$. The gate delay is the time the stream of interest s waits for the TAS window to open. For the calculation of $d^{e,s}_{\text{gate}}$, we distinguish between the unsynchronized reception of s, i.e., anytime in the cycle, and the synchronized reception of s, i.e., at a known offset in the cycle. Without TAS, $d^{e,s}_{\text{gate}}$ is zero because the gate is always open.

In the unsynchronized worst-case scenario, the largest stream g in $\mathcal{S}^{e,s}_{\text{ifr}}$ arrives just ahead of s, and both do not fit into the TAS window anymore. We describe this additional delay effect with:

$$d^{e,s}_{\text{dwell}} = d^{e,g}_{\text{trans}} + d^{e,g}_{\text{block}} + d^{e,s}_{\text{trans}} \quad (7)$$

Hence, s waits until the next open TAS window, and we conclude:

$$d^{e,s}_{\text{gate}} = CT^{e,s}_{\text{GCL}} - d^{e,s}_{\text{window}} + d^{e,s}_{\text{dwell}} \quad (8)$$

In the unsynchronized scenario, the best case of $d^{e,s}_{\text{gate}}$ is zero.

In the synchronized scenario, $d^{e,s}_{\text{gate}}$ depends on when a frame is ready for transmission, i.e., after the processing. We define the enqueuing time of s as absolute time $t^{e,s}_{\text{enq}}$. As the gate open and close times are relative to the cycle, we use $t^{e,s}_{\text{enqCT}} = t^{e,s}_{\text{enq}} \mod CT^{e,s}_{\text{GCL}}$ to describe the relative enqueuing offset within the cycle. We distinguish three cases to calculate the gate delay $d^{e,s}_{\text{gate}}$. First, s is enqueued before the gate opens at $t^{e,s}_{\text{open}}$ (cf. case C1 in Equation 9). Second, s is enqueued early enough, so it still fits into the TAS window before it closes at $t^{e,s}_{\text{close}}$, considering potential interference (cf. case C2 in Equation 10). Third, s does not fit into the gate anymore or is received after the gate is closed. In particular, if s waits at the closed gate, potentially other streams in $\mathcal{S}^{e,s}_{\text{ifrpath}}$ arrive earlier, but also at the closed gate, and interfere when the gate opens. Therefore, we define $d^{e,s}_{\text{gate}}$ as:

$$\text{case C1: } t^{e,s}_{\text{enqCT}} < t^{e,s}_{\text{open}} \quad (9)$$

$$\text{case C2: } t^{e,s}_{\text{enqCT}} + d^{e,s}_{\text{dwell}} + d^{e,s}_{\text{ifrcross}} < t^{e,s}_{\text{close}} \quad (10)$$

$$d^{e,s}_{\text{gate}} = \begin{cases} t^{e,s}_{\text{open}} - t^{e,s}_{\text{enqCT}} + d^{e,s}_{\text{ifrpath}} & \text{case C1} \\ 0 & \text{case C2} \\ CT^{e,s}_{\text{GCL}} - t^{e,s}_{\text{enqCT}} + t^{e,s}_{\text{open}} + d^{e,s}_{\text{ifrpath}} & \text{otherwise} \end{cases} \quad (11)$$

In the synchronized scenario, $d^{e,s}_{\text{gate}}$ can have different values for the best case and worst case, depending on the predicted $t^{e,s}_{\text{enqCT}}$ values. The result of Equation 11 is limited to the result of Equation 8.

5.2.7 Queuing delay $d^{e,s}_{queue}$. The queuing delay is a combination of three other delay components: $d^{e,s}_{\text{block}}$, $d^{e,s}_{\text{ifr}}$, and $d^{e,s}_{\text{gate}}$ (cf. Figure 3). However, if the TAS mechanism is configured on a node and the frame needs to wait for the gate to open, no frame of a lower priority can cause a blocking delay $d^{e,s}_{\text{block}}$. Therefore, we use the maximum of $d^{e,s}_{\text{gate}}$ and $d^{e,s}_{\text{block}}$ in this case:

$$d^{e,s}_{\text{queue}} = max(d^{e,s}_{\text{gate}}, d^{e,s}_{\text{block}}) + d^{e,s}_{\text{ifrcross}} \quad (12)$$

5.2.8 Time Synchronization. Our model expects time-synchronization to be implemented in accordance with IEEE 802.1AS as required by the industrial TSN profile IEC/IEEE 60802 [6]. We measured a symmetric synchronization jitter for the devices in our lab resulting in an achieved accuracy of 30 ns with IEEE 802.1AS using default settings. However, this accuracy and synchronization jitter is implementation-specific. Hence, the clock jitter j^v_{sync} is a node property and denotes the symmetric maximum deviation from a perfect synchronization of two neighboring nodes.

5.3 Complete Single Node Model

In conclusion, we define the single node forwarding delay $d^{v,e,s}_{\text{forward}}$ of s on the forwarding node v and the edge e as a sum of all delays

presented in Figure 3. Additionally, Equation 13 considers the store-and-forward accordion effect created by the largest stream g in $\mathcal{S}^{e,s}_{\text{ifrpath}}$, i.e., a long $d^{e,g}_{\text{trans}}$ limits forwarding of s on every hop for the duration of the frame size difference between s and g.

$$d^{v,e,s}_{\text{forward}} = d^{e}_{\text{prop}} + d^{e,s}_{\text{trans}} + d^{v}_{\text{proc}} + d^{e,s}_{\text{queue}} + max(0, d^{e,g}_{\text{trans}} - d^{e,s}_{\text{trans}}) \quad (13)$$

Finally, we calculate the best-case and worst-case delay for a single node. The transmission window $w^{v,s}_{\text{trans}}$ and arrival window $w^{v,s}_{\text{arriv}}$ denote the difference between the best-case and worst-case latency at the transmission and arrival time for s on the switch v. We discuss both cases (best case and worst case) individually:

5.3.1 Best-case scenario. In the best case, stream s does not interfere with any other stream nor is subject to blocking by other traffic, i.e., the full bandwidth is available for s. Therefore, we set $d^{e,s}_{\text{ifr}}$ and $d^{e,s}_{\text{block}}$ to zero in the best-case calculation. Similarly, if there is no TAS configured or there is no time synchronization, we set $d^{e,s}_{\text{gate}}$ to zero. For a stream that was received with synchronization, we apply Equation 11 for the transmission. We subtract j^{v}_{proc} for the different delay components in the best-case scenario. With this correction, we assume a best-case behavior of all components on the forwarding path. Additionally, we assume that the gate delay $d^{e,s}_{\text{gate}}$ is reduced by j^{v}_{sync}, i.e., the gate opens as early as possible.

5.3.2 Worst-case scenario. To calculate the worst-case behavior, we need to apply Equation 4 and $d^{e,s}_{\text{block}}$ based on the knowledge of maximum frame sizes, TSN mechanisms configured, and other time-critical streams. Similar to the best case, we need to calculate the gate delay $d^{e,s}_{\text{gate}}$ but this time with the worst-case assumptions for $d^{e,s}_{\text{ifr}}$ and $d^{e,s}_{\text{block}}$. However, we add the j^{v}_{proc} and increase $d^{e,s}_{\text{gate}}$ by j^{v}_{sync}, i.e., the gate opens as late as possible.

6 MULTI-NODE MODEL

We extend the single-node model to the path of a stream along multiple TSN nodes. Since the stream might traverse multiple TSN domains on its path, the nodes might use different TSN mechanisms and configurations. This multi-node model has two main goals: first, determining the best-case and worst-case latency bounds for each stream in the network, and second, determining the required and available bandwidth for the streams.

6.1 Latencies of a Stream

To calculate the best-case and worst-case end-to-end latency bounds of a stream, we repeatedly use the single-node model for each node of the path and sum up the best-case and worst-case delays along the path. The calculation starts with the sender and iterates the tuples in $Path_s$.

6.1.1 Latency for Traversing Adjacent TSN Nodes. For each stream, we determine a transmission window $w^{v,s}_{\text{trans}}$ on the egress port of the switch v using the best-case and worst-case scenarios. This transmission window equals the arrival window $w^{v+1,s}_{\text{arriv}}$ on the ingress port of the next switch $v+1$ on the path. Depending on the upper and lower bounds of the arrival window, a frame can be immediately forwarded, delayed due to closed gates, or interfere with the traffic of the same or higher priority, influencing the transmission window $w^{v+1,s}_{\text{trans}}$. Afterward, we follow the path of a frame

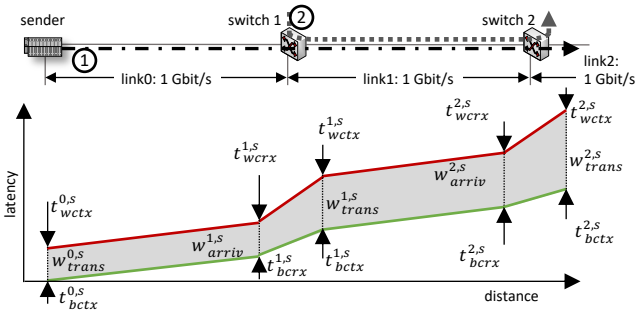


Figure 4: Visualization of the arrival window calculated with the introduced model for a fictive setup

along its transmission from node to node and calculate the total best-case and worst-case delays.

We begin with the periodic transmission of the frame at the sender, e.g., a vPLC or a sensor. This transmission event can either happen (i) at a specific point in time (offset) in the cycle, (ii) during a transmission window, or (iii) at an arbitrary moment within the cycle. Cases (i) and (ii) require high-precision or medium-precision, respectively, of time-synchronization to enable TSN-domain-wide coordination of transmission times and windows. Case (iii) does not require time synchronization at all. For a precise and accurate offset in case (i) and transmission window in case (ii), we can directly derive the best case and worst case as arrival bounds on the first switch of the path of the stream. In the unsynchronized case (iii), we can use the start and end of the cycle as best-case and worst-case bounds because these times are the most and least favorable transmission instances regarding latency. In Figure 4, the sender transmits stream 1 within the transmission window $w^{0,s}_{\text{trans}}$. The gray area shows the range of the expected latency. The lower edge shows the best-case and the upper edge the worst-case latency. The resulting best- and worst-case latency values are the bounds of the first arrival window $w^{1,s}_{\text{arriv}}$. This arrival window starts at the best case $t^{1,s}_{bcrx}$ and has a duration of $t^{1,s}_{wcrx} - t^{1,s}_{bcrx}$.

Next, for the first TSN switch, we use the boundaries of the arrival window $w^{1,s}_{\text{arriv}}$ to calculate the individual best-case and worst-case latency of the stream, depending on the TSN configuration of the switch, considering all other streams that traverse the switch. For example, in Figure 4, stream 2 is received on a different port and can interfere with stream 1 on the following path to switch 2. This leads to an increased transmission window $w^{1,s}_{\text{trans}}$, compared to the arrival window $w^{1,s}_{\text{arriv}}$. The stream traverses additional switches on the path $Path_s$ throughout the network. At each step, the bounds of the transmission window become the bounds of the arrival window of the next switch on the path. Finally, at the receiver of the stream, we can compare the expected best-case and worst-case latency to the required QoS requirements of the application.

Based on Equation 13 for the forwarding of a single node, the following sum shows the complete end-to-end latency d_{e2e} for s:

$$d^{s}_{e2e} = \sum_{(v,e)=(sender,link0)}^{Path_s} d^{v,e,s}_{\text{forward}} \quad (14)$$

We denote each hop of s on its path as a tuple of a node and an egress link (v, e). The initial value of the tuple is the combination of the

sender and the first link to which the sender transmits the frame. To obtain best- and worst-case bounds, we calculate Equation 14 with best-case and worst-case assumptions for $d^{v,e,s}_{\text{forward}}$, respectively.

So far, we did not include the influence of different link speeds on the latency. We discuss this impact in the following section.

6.1.2 Impact of Link Speeds. In factory networks, a mixture of link speeds is typical. For example, the machine networks typically operate at 100 Mbit/s, whereas backbone networks run at up to 10 Gbit/s. The link speed significantly affects the forwarding process in terms of transmission, interference, and blocking delays. Table 2 gives an overview of the relation between frame size, link speed, and transmission delay. As the table indicates, the lower the link speed, the higher the delay of the stream of interest s caused by interfering streams. The model calculates the best-case and worst-case latencies based on the available bandwidth. Hence, the link speed for each link in the network is included in $d^{e,s}_{\text{trans}}$. However, if frames arrive at high link speed and are sent on a port with low link speed, congestion can occur. To model this behavior, we assume the worst-case congestion of all streams that potentially arrive before the stream of interest when the link speed changes from high to low on a node. Therefore, $d^{e,s}_{\text{queue}}$ increases by $d^{e,s}_{\text{ifrpath}}$ at the first slower edge. If interfering frames leave a switch faster than they arrived, the streams ahead of s do not influence s.

6.2 Bandwidth Requirements

Our model translates the QoS requirements of streams into the required bandwidth. As introduced in Section 5, we compare these bandwidth requirements to the available bandwidth. Only if sufficient bandwidth is available, the model can calculate best-case and worst-case latencies. At the sender, the required bandwidth is defined by frame size and application cycle time. In the network, the theoretical bandwidth requirement per network cycle can change. In this section, we introduce two cases in which this theoretical bandwidth requirement increases, as multiple frames of one stream arrive in the same network cycle. For both cases, we derive a factor that we use in Section 6.2.3 to cope with these effects.

6.2.1 Impact of Different Cycle Times. The cycle time between two nodes can either be higher or lower. If the cycle time on the second node is higher, multiple cycles on the first node are processed, while only one cycle on the second node is processed. Hence, multiple streams from different cycles can end up in the same cycle at the second node. If the cycle time on the second node is lower, the second node processes the cycles faster than the first node. With assumption A3) in Section 5.1, i.e., all streams need to be processed in one cycle, this results in cycles that process all frames from the first and slower node and some cycles with no frames at all.

The effect can occur either between the original sender and the first TSN domain with the TAS configured or between two TSN domains with different cycle times. For each of these occurrences, we compute a factor, which we use in Section 6.2.3 to update the worst-case required bandwidth in total. Equation 15 presents the difference between the sender and the first TSN domain with the TSN switch v. Equation 16 describes the factor for the difference between two TAS-based TSN domains.

$$f_{app} = \lceil CT^{e,s}_{\text{GCL}} / CT^{s}_{\text{app}} \rceil \tag{15}$$

$$f_{CT} = \lceil CT^{e,s}_{\text{GCL}} / CT^{e-1,s}_{\text{GCL}} \rceil \tag{16}$$

6.2.2 Impact of Large Arrival Windows. Similar to different cycle times, a large arrival window $w^{v,s}_{\text{arriv}}$ can result in multiple frames per cycle. Specifically, if $w^{v,s}_{\text{arriv}}$ is larger than the TAS cycle $CT^{e,s}_{\text{GCL}}$. For example, if one iteration of a specific stream s is transmitted with the worst-case latency, and the next stream is transmitted with the best-case latency, they end up in the same cycle in the network. Equation 17 describes the calculation of the resulting factor.

$$f_{arriv} = \lceil w^{v,s}_{\text{arriv}} / CT^{e,s}_{\text{GCL}} \rceil \tag{17}$$

6.2.3 Calculation of Required Bandwidth. With these factors, the model scales the incoming bandwidth to the theoretical bandwidth required at transmission. Hence, we calculate this theoretical bandwidth requirement with the Layer-2 frame size on the previous node b^{v-1}_s. We start with the required bandwidth of the first switch based on the size of the frame at the sender b^{sender}_s:

$$b_{s,v} = b^{sender}_s \cdot f_{app} \tag{18}$$

On each of the following hops, the model calculates the expected bandwidth as follows:

$$b_{s,v} = b^{v-1}_s \cdot f_{arriv} \cdot f_{CT} \tag{19}$$

With Equations 18 to 19, the model is able to determine the worst-case required bandwidth per network cycle for each stream on every hop. Hence, the model is able to compare the available bandwidth (defined by link speed and TAS configuration) to the required bandwidth. This enables the calculation of the best-case and worst-case latencies and the prediction of resource utilization.

7 EVALUATION

We evaluate our model with an experimental series of measurements using real hardware because the model includes physical effects like processing and time-synchronization jitter. First, we evaluate the accuracy of the model to predict the correct forwarding delay for a single node. Second, the evaluation covers the analysis of the gate delay prediction. Finally, we execute a series of measurements on a topology consisting of three TSN domains. Specifically, we generated 196 different configurations for this topology, with individual configurations per TSN domain. With these configurations, we analyze the applicability of the model regarding end-to-end latency prediction, resource utilization prediction, and the identification of high-latency and high-jitter links.

7.1 Setup and Methodology

For our evaluation, we use a line topology with three off-the-shelf industrial TSN switches (RSPE35 from Belden Inc. [10]). If not stated otherwise, each link has a bandwidth of 1 Gbit/s. The stream of interest s is transmitted through each switch. It has a Layer-2 frame size of 256 B and uses the highest priority. We use each switch as an independent TSN domain, e.g., potentially different TSN mechanisms, different parameterization, and different time sources. Hence, this topology models a scenario with three TSN domains as it exists in industrial environments. Additionally, we can inject traffic on three links on each switch to emulate adjacent network segments and their influences. We use two of these links for streams with high priority and one link to saturate the network with best-effort

traffic with a frame size of 1,522 B. In our measurements, we use the *IXIA XGS12* [25] traffic generator to ensure precise transmission of interfering high-priority traffic and best-effort traffic.

We use timestamping of Ethernet frames to precisely determine delays and path latency. To this end, each switch and the sender write their current time into the frame when receiving and transmitting the frame. The receiving timestamp represents the arrival time of the first received bit. The switch also inserts the transmission timestamp when it transmits the first bit. This timestamp represents the instance at the end of the queuing without the propagation delay. The difference between the reception and the transmission timestamp includes the transmission, processing, and queuing delay. In contrast, the difference between the transmission and the reception timestamp between two adjacent switches denotes the propagation delay, including the time-synchronization jitter. The propagation delay for the cables in the setup is 5 ns, while the time synchronization jitter j^v_{sync} for the devices in the setup is 30 ns. Hence, the accuracy of our measurements is only within j^v_{sync}.

We evaluate our model in three different evaluation scenarios. In the first two scenarios (Sections 7.2 and 7.3), we focus on analyzing the real and predicted single node forwarding delay $d^{v,e,s}_{\text{forward}}$. For each of these scenarios, we use 1,000 measurements with 3,000 transmission cycles per measurement. In the third scenario (Section 7.4), we show the applicability of the model in heterogeneous TSN domains. In this scenario, we evaluated 196 network configurations and evaluated each configuration with 50 measurement runs and 3,000 transmission cycles per measurement.

To show the relevance of the worst-case predictions, we must ensure that even rare worst-case behavior can be observed in our real-world measurements. Therefore, we selected a relatively short application cycle time CT^s_{app} of 100 µs to make collisions of priority frames more likely. In reality, these worst cases can also occur with longer cycle times, however, within weeks or even months of operation. This may pose even more problems for the reliable and safe operation of an industrial application. Similarly, the total worst-case latency in a complete network is less likely for larger network deployments. Hence, in our single-node evaluations, we can observe the predicted worst case with a higher probability than in the evaluation with three TSN domains. The predicted best case is more likely to occur and is present in most evaluations.

7.2 Single Node Queuing Delay

We begin the evaluation with the analysis of a single TSN switch. The goal of this section is to evaluate if the model describes the behavior of a single node correctly and sufficiently. Specifically, the goal of the model is to provide reliable upper and lower latency bounds, i.e., never underestimate the measured worst-case latency and never overestimate the measured best-case latency. Additionally, the predictions should be close to the measured behavior to be meaningful for analyzing real-life deployments. To analyze the different delay components in the forwarding delay $d^{v,e,s}_{\text{forward}}$, we evaluate six different TSN settings. Table 3 summarizes the six settings with the configuration of the TSN switches. We send the frame of the stream of interest at the beginning of the cycle time. For each of the six settings, we execute the evaluation with and without interference.

Table 3: Single Node Evaluation Settings

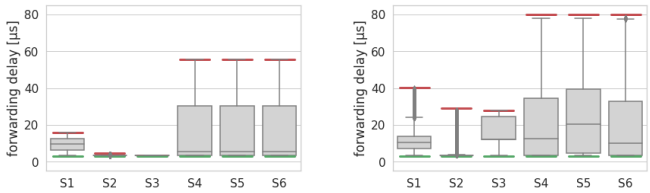
Setting	Description
S1	Strict Priority
S2	FP with priority 7 in the express category
S3	synchronized TAS with prio. 7 in the TAS window
S4	unsynchronized TAS with prio. 7; CT^s_{app} = 100µs
S5	unsynchronized TAS with prio. 7; CT^s_{app} = 45µs
S6	unsynchronized TAS with prio. 7; CT^s_{app} = 196µs

In the evaluation run without interference, we evaluate the prediction of the processing delay d^v_{proc}, blocking delay $d^{e,s}_{\text{block}}$, and gate delay $d^{e,s}_{\text{gate}}$, depending on the TSN configuration. In the evaluation with interference, we evaluate the interference delay prediction $d^{e,s}_{\text{ifr}}$. For the interfering traffic, we send two high-priority streams. Each has a cycle time of 100 µs and a frame size of 1,522 B.

Figure 5 provides the comparison of the prediction and measurement for the six settings S1-S6 without interfering traffic in Figure 5a and with interfering traffic in Figure 5b. For each setting, the bottom horizontal line green line presents the predicted best-case, and the top horizontal line red line the worst-case delay of the single TSN switch. In between, the gray box plot shows the distribution of the measured results, i.e., the box represents the 25th and 75th percentiles, and the whiskers present the 1.5 IQR. In summary, the prediction approximates the measured values in all evaluation runs. As expected for a single node, the difference between the observed best-case and worst-case and the predicted delays is small. In the following sections, we discuss the details of these measurements.

7.2.1 Best-Case Prediction. In every setting, the predicted best-case latency of the stream implies direct forwarding without queuing delay $d^{e,s}_{\text{queue}}$. The predicted best-case latency is 3.17 µs. It consists of 1,050 ns d^v_{proc}, 5 ns d^e_{prop}, and 2,200 ns $d^{e,s}_{\text{trans}}$, reduced by the jitter of 80 ns, i.e., 50 ns j^v_{proc} and 30 ns j^v_{sync}. Throughout all measurements in all six settings, with and without interference, the observed best case was 3.20 µs. Setting S3 without interference is a good reference for a best-case situation since using TAS prevents any external influences. The difference between prediction and measurement across all evaluation settings was 30 ns. This difference is negligible, as it is in the range of j^v_{sync}.

7.2.2 Worst-Case Prediction. In this section, we evaluate the worst-case prediction for each setting, with and without interference. Compared to the best-case prediction, the pure forwarding delay includes d^v_{proc}, d^e_{prop}, $d^{e,s}_{\text{trans}}$ and adds the jitter components. The model predicts the sum of these delays with 3.34 µs. Additionally, the worst-case prediction includes all delay components of $d^{e,s}_{\text{queue}}$,



(a) no interference **(b) interference**

Figure 5: Single node queuing delay evaluation

i.e., $d^{e,s}_{\text{block}}$, $d^{e,s}_{\text{ifr}}$, and $d^{e,s}_{\text{gate}}$, depending on the configured TSN mechanisms. Across all evaluation runs, the predicted worst-case latency is always larger than the measured latency. Specifically, the difference between the measurement and prediction is always below 0.2 µs without interference and 2.1 µs for the settings with interference, which is less than 2.8% of the window between best-case and worst-case delay. As the predicted values vary between the settings, we discuss the details in the following.

Strict Priority. Setting S1 represents the Strict Priority configuration. As described in Section 5.2.5, critical traffic can be blocked by a 1,522 B frame, increasing the blocking delay $d^{e,s}_{\text{block}}$ to 12.3 µs. This results in a predicted worst-case latency of 15.67 µs, where we measured 15.56 µs. With the additional interference delay $d^{e,s}_{\text{ifrcross}}$ of 24.6 µs, we predict 40.23 µs and measured a worst case of 39.98 µs.

Frame Preemption. Setting S2 additionally uses FP. The measurement stream is in the express category, and the background traffic is in the preemptible category. With these settings, the blocking delay is reduced to 1.2 µs (duration of a 123 B fragment). For S2 without interference, we predict a worst case of 4.51 µs and measure a worst case of 3.85 µs. With interference, the predicted $d^{e,s}_{\text{ifrcross}}$ increases to 24.6 µs, leading to a total predicted worst-case latency of 29.2 µs. The measured worst case for the frame preemption with interference on a single node is 27.79 µs.

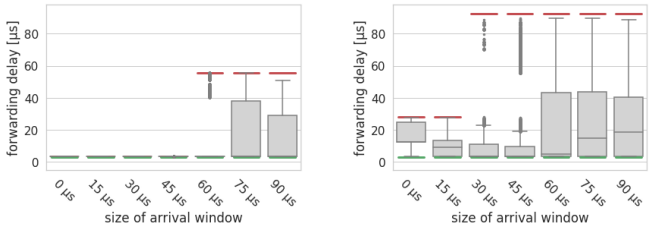
Synchronized TAS. In setting S3, the TAS does not allow best-effort traffic within the same window as the measurement traffic. Therefore, the predicted blocking delay $d^{e,s}_{\text{block}}$ is reduced to 0 µs. The time-synchronization between the sender and switch results in a predicted gate delay $d^{e,s}_{\text{gate}}$ of 0 µs. For S3 without interference, we predict a worst-case latency of only 3.34 µs (the pure worst-case forwarding delay, without queuing delay) and measure 3.33 µs. Hence, the difference between the best case and worst case is only influenced by the jitter. For S3 with interference, the model predicts 27.93 µs, and we measured 27.92 µs. This difference between prediction and measurement is both cases is 10 ns and, therefore, the model describes the behavior adequately.

Unsynchronized TAS. In this section, we discuss the three different unsynchronized settings S4, S5, and S6. Unsynchronized means that the stream can arrive at any time in the cycle. Each of these settings has a different application cycle time, i.e., S4 has 100 µs, S5 has 45 µs, and S6 has 196 µs. We selected these different cycle times to generate cycles containing no frames of the measurement stream and up to three frames of the measurement stream.

The model calculates the gate delay based on Equation 8, which does not include the periodicity of s. Also, the forwarding delay $d^{v,e,s}_{\text{forward}}$ is independent of the periodicity of s. Hence, the model predicts for all three settings, without interference, a worst-case latency of 55.57 µs. With interference, the model predicts a worst-case latency of 79.96 µs. We measured a maximum latency of 55.36 µs without interference and 78.1 µs with interference. Therefore, we conclude that the model approximates the behavior adequately.

7.3 Single Node Gate Delay

Following Section 6.1, frames may arrive in a window, the arrival window. This window is defined by the best-case and worst-case latency along the previous path. When forwarding a frame, TAS can cause an additional gate delay, depending on the time at which the frame arrives. This part of the evaluation shows that we model the gate delay accurately for different TAS settings.



(a) no interference **(b) interference**

Figure 6: Single node gate delay evaluation

We use setting S3 and vary the size of the arrival window from 0 µs to 90 µs in steps of 15 µs to predict and measure the gate delay. In all measurements, the switch was configured with a 50 µs TAS window in a 100 µs TAS cycle. Figure 6a shows the results without interference and 6b with interference. The calculated interference delay is the transmission time of two 1,522 B frames, i.e., $d^{e,s}_{\text{ifr}}$ = 24.6µs. For all measured settings, the prediction approximates the measured behavior with an accuracy of 0.8% without interference and 2% with interference.

In the experiment without interference (Figure 6a), all frames received within an arrival window smaller than 50 µs fit into the TAS window and, therefore, have a gate delay of 0 µs. Once the arrival window exceeds 50 µs, some frames arrive while the TAS window is closed and must wait for it to open. This is visible in Figure 6a for both the measurements and the predictions with an arrival window of 60 µs, 75 µs, and 90 µs. For example, for an arrival window of 60 µs, the measured delay is 55.1 µs while the worst-case prediction is 55.58 µs.

In the experiment with interference, the interfering streams from other ingress links also affect the growth of the transmission window (cf. Equation 12). As a result, the stream does not fit into the TAS window starting already at an arrival window of 30 µs (cf. Equation 10) with a predicted gate delay of 64.5 µs. With Equation 13, this results in a worst-case delay prediction of 92.58 µs and a measured worst-case delay of 88.31 µs. Again, the model approximates the measured behavior adequately.

7.4 Evaluation on Three TSN Domains

The model is designed to predict latency bounds for combinations of TSN domains with different configurations. Therefore, we evaluate the model on a three-domain topology as presented in Figure 7. This setup approximates a factory network with a machine network, a backbone network, and a local data center network.

To cover a vast set of different domain combinations, we generated 196 different TSN parameter settings for a path traversing three domains. These settings include different combinations of SP, FP, and different TAS configurations, i.e., cycle times and TAS windows. For completeness, Figure 10 in the appendix presents the overview of all measurements for the settings. Across all evaluation runs, our model always predicted the best-case and worst-case latencies correctly, i.e., it never predicted higher best-case nor lower worst-case latencies than measured in our real TSN testbed.

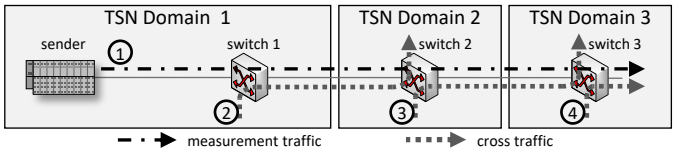


Figure 7: Evaluation topology; Stream 1: 256 B, Stream 2: 1,024 B, Stream 3: 1,024 B, Stream 4: 1,024 B

The 196 settings vary in the selection and parameterization of the TSN mechanisms, cycle times, and link speed. Additionally, we enabled and disabled the time synchronization between the domains. For each setting, we performed 50 measurement runs with 3,000 cycles per measurement. The details and results for each setting are shown in Table 5 and Section A.1 in the Appendix.

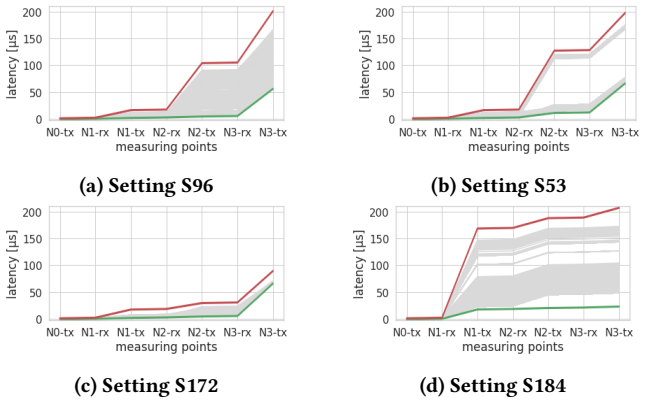
Figure 7 illustrates the evaluation topology. The stream of interest s is stream 1. Its cycle time is 100 µs, and the frame size is 256 B. Throughout the measurements, we inject additional high-priority streams on each switch (streams 2, 3, and 4) to emulate traffic from the adjacent network infrastructure. Specifically, each of these additional streams has a cycle time of 100 µs and a frame size of 1,024 B. The network is saturated to 95% load with best-effort traffic using additional links on each switch.

Our model aims to enable the deployment of time-critical applications in multi-domain topologies, i.e., to analyze the end-to-end latency, predict resource utilization, and identify high-latency and high-jitter links. In the following, we analyze the applicability of our model for each of these use cases using selected scenarios.

7.4.1 Analysis of End-to-End Latency. In this section. we present four examples out of the 196 evaluated settings in more detail. The selection of these measurements focuses on four possible configuration differences between TSN domains: 1) time synchronization, 2) different TAS configurations, 3) a combination of different TSN mechanisms, and 4) different link speeds.

The diagrams in Figure 8 show the measuring points in the network on the x-axis and the latency on the y-axis, similar to Figure 4. For the three-domain evaluation, the measuring points are: sender (N0-tx), switch 1 reception (N1-rx) and transmission (N1-tx), switch 2 reception (N2-rx) and transmission (N2-tx), and switch 3 reception (N3-rx) and transmission (N3-tx). The light gray lines show the latencies of a frame progressing through the cascade of switches and measuring points. The area of all gray lines shows the range of observed behavior. The lower thick green line indicates the predicted best-case latency, and the upper thick red line indicates the predicted worst-case latency.

In setting S96, each of the three TSN domains is configured with the TAS mechanism. The sender (N0-tx) transmits traffic in a synchronized way, i.e., with a precise offset, into the first TSN domain. However, the second domain is not synchronized to the first domain, which leads to an increased predicted worst-case latency between N2-rx and N2-tx. This is because the gate delay $d^{e,s}_{\text{gate}}$ is increased due to missed open gates. Domain 2 and domain 3 are synchronized to each other but have different TAS windows configured, so that switch 3 opens the gate later in the cycle. This leads again to increased latency in the best-case and worst-case prediction and in the measured latency. Figure 8a shows that the prediction of $w^{v,s}_{\text{arriv}}$ and $w^{v,s}_{\text{trans}}$ along the path is close to the measured latency.



(a) Setting S96 **(b) Setting S53**

(c) Setting S172 **(d) Setting S184**

Figure 8: Analysis of end-to-end latency

In setting S53, each TSN domain is configured with the TAS mechanism, and all TSN domains are synchronized with each other. However, this time the second TSN domain does not use a cycle time of 100 µs but uses 200 µs. The measured trace clearly shows this effect, as half of the streams have an offset of 100 µs. The model predicts this behavior correctly and shows the same offset in the worst-case latency (cf. Figure 8b).

Figure 8c shows the measurement trace and prediction for setting S172 representing two sequential FP-based TSN domains (switch 1 and 2) communicating into one synchronized TAS-based TSN domain (switch 3). The model predicts the worst-case latencies for the FP-based TSN domains based on the knowledge of the express priorities correctly. The synchronized TAS domain has a static offset of the TAS window to the transmission time at N0-tx, causing the frames to arrive at a closed gate at N3-rx. Therefore, all traffic is enqueued before the gate opens and is sent out directly when the gate opens. Both the model and the measurements show the different behaviors of FP and a closed TAS window.

Figure 8d shows the measurement trace and prediction for S184, consisting of three SP-based TSN domains. In contrast to the previously described measurements, this measurement includes a 100 Mbit/s link between N1-tx and N2-rx. The lower link speed increases the transmission delay of all traffic. This effect is clearly shown by the increased latency at N1-tx for the measurement and prediction in the best case and worst case. On the subsequent links, the link speed is 1 Gbit/s, which decreases the growth of the predicted and the measured latency. As visible in the figure, no measured frames met the predicted best-case and worst-case latency because it is unlikely that a frame is either never blocked or blocked on all hops. For long run times, which are typical for industrial applications, such unlikely worst-case events must still be expected.

In summary, the four examples cover different possible variations for heterogeneous TSN domains. In combination with the results listed in Appendix A, they show the capability of the model to predict meaningful end-to-end latencies for a wide range of settings.

7.4.2 Analysis of Resource Utilization. As introduced in Section 5.1, the available bandwidth of a link is defined by the cycle time, link speed, and TAS window. Based on the knowledge of all high-priority streams and their paths, the model calculates their required bandwidth. In combination, these two values represent the utilization of

Table 4: Analysis of resource utilization

Setting	Link 0	Link 1	Link 2	Link 3	Packet loss
S74	1.7%	18.0%	12.4%	99.2%	False
S55	1.7%	18.0%	14.6%	116.8%	True
S63	1.7%	18.0%	16.8%	134.4%	True

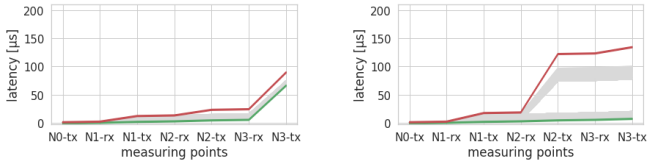
the network resources. We use percentages to express the amount of utilization, with 100% as the full utilization of a link. Above 100% utilization, streams need to be buffered for more than one cycle in the worst case, and packet loss might occur because of insufficient buffer space. Across all of our measurements, we did not observe packet loss or exceed the predicted worst-case latencies for scenarios with a predicted maximum resource utilization of less than 100%. In contrast, we observed packet loss in all scenarios with a predicted utilization above 100%, besides for settings S56, S57, S60, and S124. For these four settings, we conducted a manual parameter analysis, which showed that higher latencies and packet losses are possible with these settings for a longer test duration.

In the following, we discuss three settings, for which one shows 99.2% utilization (S74) on a link and two related settings that show overutilization at 116.8% (S55) and 134.4% (S63) (cf. Table 4). For all cases, the model correctly predicts the feasibility and infeasibility of the settings. All three settings are configured with TAS in each of the domains. The cycle time in the middle domain is set to 100 µs (S74), 200 µs (S55), and 300 µs (S63), whereas the first and the last TSN domain on the path have a cycle time of 100 µs. The TAS window duration is the same across the settings. Table 4 presents the calculated resource utilization for the priority of the stream of interest per link. The utilization values show the relation between the predicted required and the available bandwidth within the open TAS window. Hence, resource utilization refers to the bandwidth available to a specific Ethernet priority instead of the full bandwidth of the Ethernet link.

In summary, our model predicted overutilization for 32 of the 196 settings. For 28 of these settings, we observed signs of overutilization (losses and additional delays) in our measurements. For 164 settings, no overutilization was predicted. Our measurements confirm these predictions since no losses or prolonged latencies were observed in these settings.

7.4.3 Identifying High-Latency and High-Jitter Links. In addition to determining best-case and worst-case latencies, the model is also useful to find the specific links or hops on the path that influence latency and jitter the most. This is achieved by analyzing the change of the worst-case and best-case latency in the reception and transmission window from hop to hop. In Figure 9, a hop with a steeply increasing worst-case prediction visually indicates such points with a negative impact on latency. Likewise, an increase in the distance between the best case and worst case shows an increase in jitter.

Figure 9 shows two settings and their resulting predicted best-case and worst-case latencies. Figure 9a shows an example of a link that strongly increases the latency because of a closed TAS window at the time the frames of s arrive. Hence, the frames must wait before they are forwarded at the last hop. Improving the alignment of the last TAS window can solve this problem.



(a) Setting S31 **(b) Setting S196**

Figure 9: Identifying high-latency and high-jitter links

Figure 9b shows a combination of an FP-domain, a TAS-domain, and an FP-domain, which leads to a slightly more complex behavior. In this example, the interference and blocking delays in the first TSN domain enlarge $w^{v,s}_{\text{arriv}}$ and $w^{v,s}_{\text{trans}}$, such that not every stream fits into the TAS window in the second domain. All streams that arrive after the gate is closed need to wait for the next cycle, which drastically increases the jitter. This inefficiency can be solved by moving the TAS window towards the end of the cycle so that the worst case still fits into the window.

While such problems are easy to foresee manually for small networks with few streams, larger and more complex networks with hundreds of streams prevent such manual analysis. In such situations, a model with the ability to determine problematic domain interconnections can help to identify slow links or inefficient TAS configurations before application errors are observed.

8 CONCLUSION

IEEE TSN and IETF DetNet offer a promising set of mechanisms to create a common deterministic factory network. However, the use of different TSN mechanisms in TSN domains makes determining QoS guarantees for TSN inter-domain traffic difficult. Previous work has avoided this problem by assuming a homogeneous set of mechanisms and a unified schedule and time source for all domains. For example, all nodes would use a common time synchronization or perfectly aligned TAS schedules would be configured. However, in practice, this cannot be assumed for many cases because of the combination of industrial machines and production lines that are assembled and configured individually.

In this work, we present a model to calculate the best-case and worst-case latencies for heterogeneous industrial networks based on TSN. In particular, our model is capable to cope with existing static TSN configurations of individual machine networks and missing time synchronization between them. We also consider the jitter caused by the network infrastructure, which is often neglected.

We show the applicability and fidelity of our model in a real testbed with actual industrial-grade TSN hardware. Our evaluation shows that the model can predict the behavior of real industrial TSN network devices accurately. Hence, with the help of the model, it becomes feasible to calculate the achievable QoS guarantees across different TSN domains in order to determine if they are sufficient for running specific industrial applications reliably.

ACKNOWLEDGMENTS

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under support code 16KIS1161 (Collaborative Project KITOS). The authors alone are responsible for the content of the paper.

REFERENCES

[1] IEEE 802.1. 2010. IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12: Forwarding and Queuing Enhancements for Time-Sensitive Streams. *IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005)* (2010), C1–72. https://doi.org/10.1109/IEEESTD.2009.5375704

[2] IEEE 802.1. 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), 1–52. https://doi.org/10.1109/IEEESTD.2016.7553415

[3] IEEE 802.1. 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), 1–57. https://doi.org/10.1109/IEEESTD.2016.8613095

[4] IEEE 802.1. 2018. 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), 1–1993. https://doi.org/10.1109/IEEESTD.2018.8403927

[5] IEEE 802.1. 2020. IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications. *IEEE Std 802.1AS-2020 (Revision of IEEE Std 802.1AS-2011)* (2020), 1–421. https://doi.org/10.1109/IEEESTD.2020.9121845

[6] IEEE 802.1. 2022. IEC/IEEE 60802 TSN Profile for Industrial Automation. *IEC/IEEE 60802 D1.4* (2022). https://www.ieee802.org/1/files/private/60802-drafts/d1/60802-d1-4.pdf

[7] IEEE 802.3. 2016. IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic. *IEEE Std 802.3br-2016 (Amendment to IEEE Std 802.3-2015 as amended by IEEE Std 802.3bw-2015, IEEE Std 802.3by-2016, IEEE Std 802.3bq-2016, and IEEE Std 802.3bp-2016)* (2016), 1–58. https://doi.org/10.1109/IEEESTD.2016.7592835

[8] Muhammad Adnan, Jean-Luc Scharbarg, Jérôme Ermont, and Christian Fraboul. 2010. Model for worst case delay analysis of an AFDX network using timed automata. In *Proceedings of 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010, September 13-16, 2010, Bilbao, Spain*. IEEE, 1–4. https://doi.org/10.1109/ETFA.2010.5641124

[9] Muhammad Adnan, Jean-Luc Scharbarg, Jérôme Ermont, and Christian Fraboul. 2012. An improved timed automata approach for computing exact worst-case delays of AFDX sporadic flows. In *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation, ETFA 2012, Krakow, Poland, September 17-21, 2012*. IEEE, 1–8. https://doi.org/10.1109/ETFA.2012.6489576

[10] Belden Inc. 2022. RSPE switches from Hirschmann. *Managed RSP Switches* (2022). https://hirschmann.com/en/Hirschmann_Produkte/Industrial_Ethernet/managed_rsp_switches/rspe_expandable/index.phtml

[11] Lucia Lo Bello, Mohammad Ashjaei, Gaetano Patti, and Moris Behnam. 2020. Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support. *J. Parallel and Distrib. Comput.* 144 (2020), 153–171.

[12] Unmesh D. Bordoloi, Amir Aminifar, Petru Eles, and Zebo Peng. 2014. Schedulability analysis of Ethernet AVB switches. In *2014 IEEE 20th International Conference on Embedded and Real-Time Computing Systems and Applications*. 1–10. https://doi.org/10.1109/RTCSA.2014.6910530

[13] Hussein Charara, Jean-Luc Scharbarg, Jérôme Ermont, and Christian Fraboul. 2006. Methods for bounding end-to-end delays on an AFDX network. In *18th Euromicro Conference on Real-Time Systems, ECRTS'06, 5-7 July 2006, Dresden, Germany, Proceedings*. IEEE Computer Society, 193–202. https://doi.org/10.1109/ECRTS.2006.15

[14] 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 2016, Brest, France, October 19-21, 2016*, Alain Plantec, Frank Singhoff, Sébastien Faucou, and Luís Miguel Pinho (Eds.). ACM, 183–192. https://doi.org/10.1145/2997465.2997470

[15] 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 2016, Brest, France, October 19-21, 2016*, Alain Plantec, Frank Singhoff, Sébastien Faucou, and Luís Miguel Pinho (Eds.). ACM, 203–212. https://doi.org/10.1145/2997465.2997494

[16] Jonathan Falk, Frank Dürr, and Kurt Rothermel. 2018. Exploring Practical Limitations of Joint Routing and Scheduling for TSN with ILP. In *24th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2018, Hakodate, Japan, August 28-31, 2018*. IEEE Computer Society, 136–146. https://doi.org/10.1109/RTCSA.2018.00025

[17] Norman Finn, Pascal Thubert, Balazs Varga, and János Farkas. 2019. Deterministic Networking Architecture. RFC 8655. https://doi.org/10.17487/RFC8655

[18] Friesen, A and Schriegel, S and Biendarra, A. 2021. PROFINET over TSN Guideline Version 1.21. *Profibus International (PI), Profibus International (PI)* (2021).

[19] Zdenek Hanzálek, Pavel Burget, and Premysl Sucha. 2009. Profinet IO IRT Message Scheduling. In *Proceedings of the 21st Euromicro Conference on Real-Time Systems, ECRTS 2009, Dublin, Ireland, July 1-3, 2009*. IEEE Computer Society, 57–65. https://doi.org/10.1109/ECRTS.2009.18

[20] Zdenek Hanzálek and Jan Dvorák. 2022. Incremental Scheduling of the Time-triggered Traffic on TTEthernet Network. In *Proceedings of the 11th International Conference on Operations Research and Enterprise Systems, ICORES 2022, Online Streaming, February 3-5, 2022*, Federico Liberatore, Greg H. Parlier, and Marc Demange (Eds.). SCITEPRESS, 302–313. https://doi.org/10.5220/0010953700003117

[21] David Hellmanns, Alexander Glavackij, Jonathan Falk, René Hummen, Stephan Kehrer, and Frank Dürr. 2020. Scaling TSN Scheduling for Factory Automation Networks. In *16th IEEE International Conference on Factory Communication Systems, WFCS 2020, Porto, Portugal, April 27-29, 2020*. IEEE, 1–8. https://doi.org/10.1109/WFCS47810.2020.9114415

[22] David Hellmanns, Lucas Haug, Moritz Hildebrand, Frank Dürr, Stephan Kehrer, and René Hummen. 2021. How to Optimize Joint Routing and Scheduling Models for TSN Using Integer Linear Programming. In *RTNS'2021: 29th International Conference on Real-Time Networks and Systems, Nantes, France, April 7-9, 2021*, Audrey Queudet, Iain Bate, and Giuseppe Lipari (Eds.). ACM, 100–111. https://doi.org/10.1145/3453417.3453421

[23] Bahar Houtan, Mohammad Ashjaei, Masoud Daneshtalab, Mikael Sjödin, and Saad Mubeen. 2021. Synthesising Schedules to Improve QoS of Best-effort Traffic in TSN Networks. In *RTNS'2021: 29th International Conference on Real-Time Networks and Systems, Nantes, France, April 7-9, 2021*, Audrey Queudet, Iain Bate, and Giuseppe Lipari (Eds.). ACM, 68–77. https://doi.org/10.1145/3453417.3453423

[24] INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC). 2019. Industrial communication networks - Profiles - Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC/IEEE 8802-3. *IEC 61784-2:2019* (2019).

[25] Keysight Technologies. 2022. XGS12 Chassis Platform. *Products and Services* (2022). https://www.keysight.com/us/en/products/network-test/network-test-hardware/xgs12-chassis-platform.html

[26] Tieu Long Mai and Nicolas Navet. 2021. Improvements to deep-learning-based feasibility prediction of switched ethernet network configurations. In *29th International Conference on Real-Time Networks and Systems*. 89–99.

[27] Tieu Long Mai, Nicolas Navet, and Jörn Migge. 2019. A hybrid machine learning and schedulability analysis method for the verification of TSN networks. In *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*. IEEE, 1–8.

[28] Lisa Maile, Kai-Steffen Hielscher, and Reinhard German. 2020. Network calculus results for TSN: An introduction. In *2020 Information Communication Technologies Conference (ICTC)*. IEEE, 131–140.

[29] Mubarak Adetunji Ojewale, Patrick Meumeu Yomsi, and Borislav Nikolić. 2021. Worst-case traversal time analysis of tsn with multi-level preemption. *Journal of Systems Architecture* 116 (2021), 102079.

[30] Francisco Pozo, Guillermo Rodríguez-Navas, Hans Hansson, and Wilfried Steiner. 2015. SMT-based synthesis of TTEthernet schedules: A performance study. In *10th IEEE International Symposium on Industrial Embedded Systems, SIES 2015, Siegen, Germany, June 8-10, 2015*. IEEE, 162–165. https://doi.org/10.1109/SIES.2015.7185055

[31] Eike Bjoern Schweissguth, Peter Danielis, Dirk Timmermann, Helge Parzyjegla, and Gero Mühl. 2017. ILP-based joint routing and scheduling for time-triggered networks. In *Proceedings of the 25th International Conference on Real-Time Networks and Systems, RTNS 2017, Grenoble, France, October 04 - 06, 2017*, Enrico Bini and Claire Pagetti (Eds.). ACM, 8–17. https://doi.org/10.1145/3139258.3139289

[32] Wilfried Steiner. 2010. An Evaluation of SMT-Based Schedule Synthesis for Time-Triggered Multi-hop Networks. In *Proceedings of the 31st IEEE Real-Time Systems Symposium, RTSS 2010, San Diego, California, USA, November 30 - December 3, 2010*. IEEE Computer Society, 375–384. https://doi.org/10.1109/RTSS.2010.25

[33] Balazs Varga, János Farkas, Andrew G. Malis, and Stewart Bryant. 2021. Deterministic Networking (DetNet) Data Plane: IP over IEEE 802.1 Time-Sensitive Networking (TSN). RFC 9023. https://doi.org/10.17487/RFC9023

[34] Luxi Zhao, Paul Pop, Zhong Zheng, and Qiao Li. 2018. Timing analysis of AVB traffic in TSN networks using network calculus. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 25–36.

A EVALUATION DETAILS

This section presents additional details on the evaluation run with 196 generated settings for three TSN domains. First, we present the details of the different TSN domain configurations. Second, we summarize the results of the evaluation runs. Last, we close this section with the detailed results for each configuration setting.

A.1 Configuration Definitions

This section summarizes the configurations used in our evaluation. The available source code[2] provides all details to initialize the model for each setting. The default link speed is 1 Gbit/s.

Domain A.

TAS 1 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 10 µs
- Gate Duration: 55 µs

TAS 2 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 30 µs
- Gate Duration: 55 µs

TAS 3 TAS configuration:
- Cycle-Time: 50 µs
- Gate Offset: 10 µs
- Gate Duration: 20 µs

FP 1 Express Priorities: 6 and 7
FP 2 100 Mbit/s and Express Priorities: 6 and 7
SP 1 Only Strict Priority
SP 2 100 Mbit/s and Strict Priority

Domain B.

TAS 1 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 5 µs
- Gate Duration: 25 µs

TAS 2 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 25 µs
- Gate Duration: 55 µs

TAS 3 TAS configuration:
- Cycle-Time: 200 µs
- Gate Offset: 25 µs
- Gate Duration: 80 µs

TAS 4 TAS configuration:
- Cycle-Time: 300 µs
- Gate Offset: 25 µs
- Gate Duration: 80 µs

TAS 5 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 5 µs
- Gate Duration: 80 µs

FP 1 Express Priorities: 6 and 7
FP 2 100 Mbit/s and Express Priorities: 6 and 7
SP 1 Only Strict Priority
SP 2 100 Mbit/s and Strict Priority

[2] https://github.com/hs-esslingen-it-security/hses-detnet-tsn-latency-jitter-model/

Domain C.

TAS 1 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 80 µs
- Gate Duration: 15 µs

TAS 2 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 10 µs
- Gate Duration: 45 µs

TAS 3 TAS configuration:
- Cycle-Time: 75 µs
- Gate Offset: 10 µs
- Gate Duration: 30 µs

TAS 4 TAS configuration:
- Cycle-Time: 200 µs
- Gate Offset: 80 µs
- Gate Duration: 10 µs

TAS 5 TAS configuration:
- Cycle-Time: 100 µs
- Gate Offset: 80 µs
- Gate Duration: 10 µs

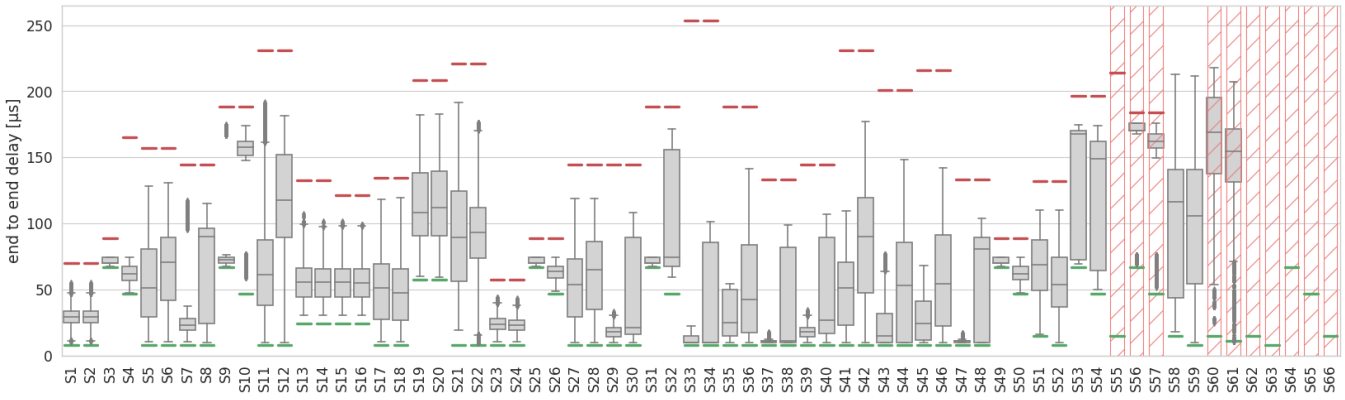
FP 1 Express Priorities: 6 and 7
FP 2 100 Mbit/s and Express Priorities: 6 and 7
SP 1 Only Strict Priority
SP 2 100 Mbit/s and Strict Priority

In each configuration, the stream of interest has a frame size of 256 B and a periodicity of 100 µs. Additionally, in each TSN domain, we inject interfering traffic of size 1,024 B and periodicity of 100 µs. We saturate the network with 95% of best-effort traffic. For the measurements with 100 Mbit/s, we reduced the periodicity of all streams to 1 ms and the frame size of interfering streams to 300 B.

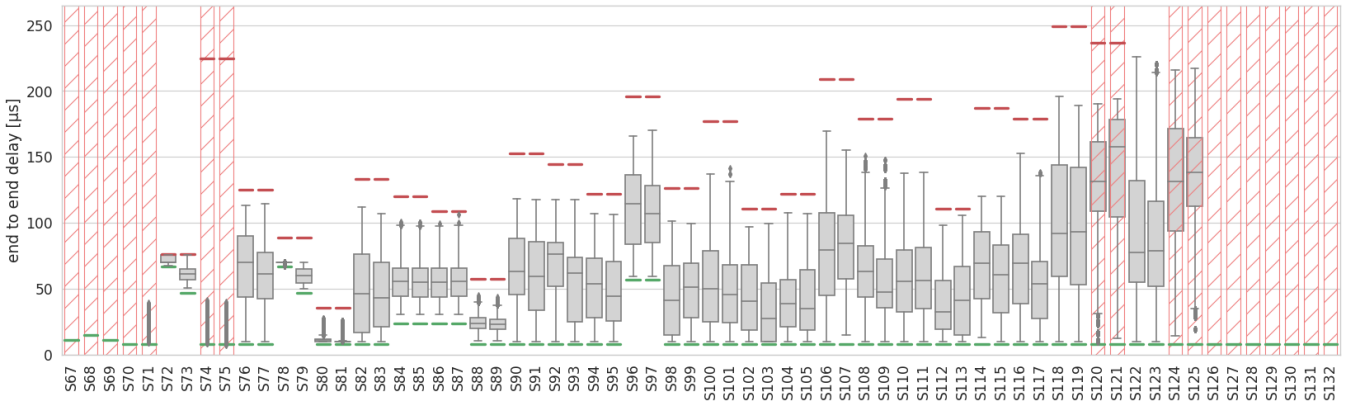
A.2 Evaluation Summary

Throughout the evaluation, we executed measurements on 196 differently configured TSN domains. For each configuration, we executed 50 measurement runs with 3,000 cycles in all measurement runs. Additionally, we used our model to predict each of the presented settings. Figure 10 presents all measurements with gray box plots. The predicted best-case and worst-case latencies are shown with the bottom (green) line and the top (red) line. We added red hatches to all settings for which the model predicted a resource utilization of more than 100%. Table 5 show the details of every measurement. In the following, we present an explanation for the shortened headlines:

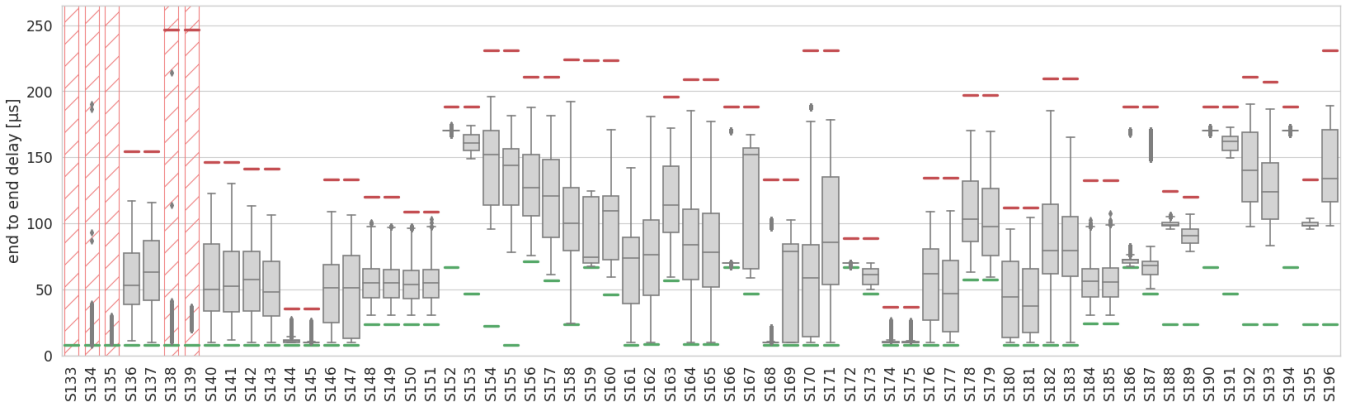
Setting Identification of a measurement configuration
W Transmission window size of sender in µs
D1 Configuration of TSN domain 1
Sync1 Synchronization between TSN domain 1 and 2
D2 Configuration of TSN domain 2
Sync2 Synchronization between TSN domain 2 and 3
D3 Configuration of TSN domain 3
P. BC Predicted best-case latency in µs
M. BC Measured best-case latency in µs
M. BC Measured worst-case latency in µs
P. WC Predicted worst-case latency in µs
P. RU Predicted resource utilization above 100%



(a) Settings S1-S66



(b) Settings S67-S132



(c) Settings S133-S196

Figure 10: Evaluation of end-to-end latencies in 196 settings

Table 5: Evaluation Summary

Setting	W [µs]	D1	Sync1	D2	Sync2	D3	P. BC [µs]	M. BC [µs]	M. WC [µs]	P. WC [µs]	P. RU
S1	0	SP 1	True	SP 1	True	SP 1	8.04	10.62	54.85	70.01	False
S2	20	SP 1	True	SP 1	True	SP 1	8.04	10.84	54.59	70.01	False
S3	0	SP 1	True	SP 1	True	TAS 1	67.02	67.37	74.3	88.48	False
S4	20	SP 1	True	SP 1	True	TAS 1	47.02	47.65	74.29	164.96	False
S5	0	SP 1	True	SP 1	False	TAS 1	8.04	10.37	128.52	156.77	False
S6	20	SP 1	True	SP 1	False	TAS 1	8.04	10.18	130.63	156.77	False
S7	0	SP 1	True	TAS 1	True	SP 1	8.01	10.08	117.17	144.48	False
S8	20	SP 1	True	TAS 1	True	SP 1	8.01	10.07	114.86	144.48	False
S9	0	SP 1	True	TAS 1	True	TAS 1	67.02	67.37	174.29	188.48	False
S10	20	SP 1	True	TAS 1	True	TAS 1	47.02	59.37	174.28	188.48	False
S11	0	SP 1	True	TAS 1	False	TAS 1	8.01	9.97	191.68	231.24	False
S12	20	SP 1	True	TAS 1	False	TAS 1	8.01	10.07	181.37	231.24	False
S13	0	SP 1	True	SP 2	True	SP 1	23.88	30.74	106.92	132.55	False
S14	20	SP 1	True	SP 2	True	SP 1	23.88	30.72	101.24	132.55	False
S15	0	SP 1	True	FP 2	True	FP 1	23.88	30.74	101.09	121.36	False
S16	20	SP 1	True	FP 2	True	FP 1	23.88	30.72	98.89	121.36	False
S17	0	SP 1	False	TAS 1	True	SP 1	8.04	10.11	120.65	134.43	False
S18	20	SP 1	False	TAS 1	True	SP 1	8.04	10.14	119.42	134.43	False
S19	0	SP 1	False	TAS 1	True	TAS 1	57.09	59.66	182.29	208.38	False
S20	20	SP 1	False	TAS 1	True	TAS 1	57.09	59.21	183.01	208.38	False
S21	0	SP 1	False	TAS 1	False	TAS 1	8.04	19.05	191.46	221.19	False
S22	20	SP 1	False	TAS 1	False	TAS 1	8.04	9.97	176.09	221.19	False
S23	0	TAS 1	True	SP 1	True	SP 1	8.01	10.09	43.56	57.7	False
S24	20	TAS 1	True	SP 1	True	SP 1	8.01	10.07	42.69	57.7	False
S25	0	TAS 1	True	SP 1	True	TAS 1	67.02	67.37	74.28	88.48	False
S26	20	TAS 1	True	SP 1	True	TAS 1	47.02	48.6	74.28	88.48	False
S27	0	TAS 1	True	SP 1	False	TAS 1	8.01	9.98	118.93	144.46	False
S28	20	TAS 1	True	SP 1	False	TAS 1	8.01	9.98	119.09	144.46	False
S29	0	TAS 1	True	TAS 1	True	SP 1	7.98	9.95	32.97	144.48	False
S30	20	TAS 1	True	TAS 1	True	SP 1	7.98	9.97	108.2	144.48	False
S31	0	TAS 1	True	TAS 1	True	TAS 1	67.02	67.37	74.3	188.48	False
S32	20	TAS 1	True	TAS 1	True	TAS 1	47.02	58.99	171.3	188.48	False
S33	0	TAS 1	True	TAS 1	True	TAS 2	7.95	9.86	22.43	253.48	False
S34	20	TAS 1	True	TAS 1	True	TAS 2	7.95	9.87	100.97	253.48	False
S35	0	TAS 1	True	TAS 1	True	TAS 3	7.95	9.87	54.28	188.48	False
S36	20	TAS 1	True	TAS 1	True	TAS 3	7.95	9.87	141.14	188.48	False
S37	0	TAS 1	True	TAS 1	True	FP 1	7.98	9.87	17.29	133.28	False
S38	20	TAS 1	True	TAS 1	True	FP 1	7.98	9.87	98.8	133.28	False
S39	0	TAS 1	True	TAS 1	False	SP 1	7.98	9.94	33.93	144.48	False
S40	20	TAS 1	True	TAS 1	False	SP 1	7.98	9.96	106.64	144.48	False
S41	0	TAS 1	True	TAS 1	False	TAS 1	7.98	9.89	109.55	231.24	False
S42	20	TAS 1	True	TAS 1	False	TAS 1	7.98	9.88	177.03	231.24	False
S43	0	TAS 1	True	TAS 1	False	TAS 2	7.98	9.87	76.44	201.24	False
S44	20	TAS 1	True	TAS 1	False	TAS 2	7.98	9.88	148.15	201.24	False
S45	0	TAS 1	True	TAS 1	False	TAS 3	7.98	9.86	67.78	216.24	False
S46	20	TAS 1	True	TAS 1	False	TAS 3	7.98	9.87	143.03	216.24	False
S47	0	TAS 1	True	TAS 1	False	FP 1	7.98	9.86	16.91	133.28	False
S48	20	TAS 1	True	TAS 1	False	FP 1	7.98	9.86	103.97	133.28	False
S49	0	TAS 1	True	TAS 2	True	TAS 1	67.02	67.37	74.29	88.48	False

Evaluation Summary (continued)

Setting	W [µs]	D1	Sync1	D2	Sync2	D3	P. BC [µs]	M. BC [µs]	M. WC [µs]	P. WC [µs]	P. RU
S50	20	TAS 1	True	TAS 2	True	TAS 1	47.02	47.64	74.28	88.48	False
S51	0	TAS 1	True	TAS 2	False	TAS 1	14.7	15.75	110.24	132.16	False
S52	20	TAS 1	True	TAS 2	False	TAS 1	7.98	9.91	109.85	132.16	False
S53	0	TAS 1	True	TAS 3	True	TAS 1	67.02	69.58	174.29	196.64	False
S54	20	TAS 1	True	TAS 3	True	TAS 1	47.02	50.06	174.29	196.64	False
S55	0	TAS 1	True	TAS 3	True	TAS 4	14.67	3274.24	3374.3	214.3	True
S56	0	TAS 1	True	TAS 3	True	TAS 5	67.02	69.99	175.8	184.3	True
S57	20	TAS 1	True	TAS 3	True	TAS 5	47.02	51.85	175.79	184.3	True
S58	0	TAS 1	True	TAS 3	False	TAS 1	14.7	17.92	213.12	340.32	False
S59	20	TAS 1	True	TAS 3	False	TAS 1	7.98	9.88	211.27	340.32	False
S60	0	TAS 1	True	TAS 3	False	TAS 5	14.7	24.67	217.78	332.98	True
S61	20	TAS 1	True	TAS 3	False	TAS 5	11.26	10.78	207.28	332.98	True
S62	0	TAS 1	True	TAS 4	True	TAS 4	14.67	2374.42	3574.29	422.46	True
S63	20	TAS 1	True	TAS 4	True	TAS 4	7.95	2752.85	3174.26	422.46	True
S64	0	TAS 1	True	TAS 4	True	TAS 5	67.02	1969.7	2175.78	292.46	True
S65	20	TAS 1	True	TAS 4	True	TAS 5	47.02	1950.51	2173.87	292.46	True
S66	0	TAS 1	True	TAS 4	False	TAS 4	14.7	2336.5	3606.38	541.14	True
S67	20	TAS 1	True	TAS 4	False	TAS 4	11.26	2411.38	3519.26	541.14	True
S68	0	TAS 1	True	TAS 4	False	TAS 5	14.7	1625.73	2431.51	541.14	True
S69	20	TAS 1	True	TAS 4	False	TAS 5	11.26	1603.47	2318.03	541.14	True
S70	0	TAS 1	True	TAS 5	True	TAS 4	7.95	2274.26	3374.29	306.14	True
S71	20	TAS 1	True	TAS 5	True	TAS 4	7.95	9.37	3374.28	306.14	True
S72	0	TAS 1	True	TAS 5	True	TAS 5	67.02	67.56	75.79	76.14	False
S73	20	TAS 1	True	TAS 5	True	TAS 5	47.02	50.28	75.78	76.14	False
S74	0	TAS 1	True	TAS 5	False	TAS 4	7.98	8.38	3409.65	224.82	True
S75	20	TAS 1	True	TAS 5	False	TAS 4	7.98	7.36	3405.34	224.82	True
S76	0	TAS 1	True	TAS 5	False	TAS 5	7.98	9.9	113.15	124.82	False
S77	20	TAS 1	True	TAS 5	False	TAS 5	7.98	9.87	114.44	124.82	False
S78	0	TAS 1	True	FP 1	True	TAS 1	67.02	67.68	70.02	88.48	False
S79	20	TAS 1	True	FP 1	True	TAS 1	47.02	49.99	70.02	88.48	False
S80	0	TAS 1	True	FP 1	True	FP 1	8.01	9.87	27.07	35.32	False
S81	20	TAS 1	True	FP 1	True	FP 1	8.01	9.86	25.38	35.32	False
S82	0	TAS 1	True	FP 1	False	TAS 1	8.01	9.88	111.8	133.27	False
S83	20	TAS 1	True	FP 1	False	TAS 1	8.01	9.87	107.14	133.27	False
S84	0	TAS 1	True	SP 2	True	SP 1	23.85	30.73	100.63	120.25	False
S85	20	TAS 1	True	SP 2	True	SP 1	23.85	30.75	99.95	120.25	False
S86	0	TAS 1	True	FP 2	True	FP 1	23.85	30.72	99.24	109.05	False
S87	20	TAS 1	True	FP 2	True	FP 1	23.85	30.71	106.42	109.05	False
S88	0	TAS 1	False	SP 1	True	SP 1	8.01	10.13	44.72	57.7	False
S89	20	TAS 1	False	SP 1	True	SP 1	8.01	10.07	43.58	57.7	False
S90	0	TAS 1	False	SP 1	True	TAS 1	7.98	9.96	118.04	152.65	False
S91	20	TAS 1	False	SP 1	True	TAS 1	7.98	9.97	117.34	152.65	False
S92	0	TAS 1	False	SP 1	False	TAS 1	8.01	9.97	117.39	144.46	False
S93	20	TAS 1	False	SP 1	False	TAS 1	8.01	9.97	117.6	144.46	False
S94	0	TAS 1	False	TAS 1	True	SP 1	8.01	9.95	106.7	122.13	False
S95	20	TAS 1	False	TAS 1	True	SP 1	8.01	9.97	106.06	122.13	False
S96	0	TAS 1	False	TAS 1	True	TAS 1	57.06	59.01	165.87	196.08	False
S97	20	TAS 1	False	TAS 1	True	TAS 1	57.06	58.98	170.03	196.08	False
S98	0	TAS 1	False	TAS 1	True	TAS 2	7.98	9.86	101.37	126.08	False
S99	20	TAS 1	False	TAS 1	True	TAS 2	7.98	9.88	99.67	126.08	False

Evaluation Summary (continued)

Setting	W [µs]	D1	Sync1	D2	Sync2	D3	P. BC [µs]	M. BC [µs]	M. WC [µs]	P. WC [µs]	P. RU
S100	0	TAS 1	False	TAS 1	True	TAS 3	7.98	9.88	136.69	177.08	False
S101	20	TAS 1	False	TAS 1	True	TAS 3	7.98	9.87	141.49	177.08	False
S102	0	TAS 1	False	TAS 1	True	FP 1	8.01	9.87	96.77	110.93	False
S103	20	TAS 1	False	TAS 1	True	FP 1	8.01	9.87	99.25	110.93	False
S104	0	TAS 1	False	TAS 1	False	SP 1	8.01	9.95	107.29	122.13	False
S105	20	TAS 1	False	TAS 1	False	SP 1	8.01	9.95	107.21	122.13	False
S106	0	TAS 1	False	TAS 1	False	TAS 1	8.01	9.9	169.6	208.89	False
S107	20	TAS 1	False	TAS 1	False	TAS 1	8.01	15.0	154.98	208.89	False
S108	0	TAS 1	False	TAS 1	False	TAS 2	8.01	9.88	150.81	178.89	False
S109	20	TAS 1	False	TAS 1	False	TAS 2	8.01	9.88	147.52	178.89	False
S110	0	TAS 1	False	TAS 1	False	TAS 3	8.01	9.87	137.5	193.89	False
S111	20	TAS 1	False	TAS 1	False	TAS 3	8.01	9.87	138.07	193.89	False
S112	0	TAS 1	False	TAS 1	False	FP 1	8.01	9.87	98.03	110.93	False
S113	20	TAS 1	False	TAS 1	False	FP 1	8.01	9.87	105.62	110.93	False
S114	0	TAS 1	False	TAS 2	True	TAS 1	7.98	12.84	120.35	187.08	False
S115	20	TAS 1	False	TAS 2	True	TAS 1	7.98	9.95	119.84	187.08	False
S116	0	TAS 1	False	TAS 2	False	TAS 1	8.01	9.9	152.49	178.89	False
S117	20	TAS 1	False	TAS 2	False	TAS 1	8.01	9.88	137.96	178.89	False
S118	0	TAS 1	False	TAS 3	True	TAS 1	7.98	9.86	196.04	249.24	False
S119	20	TAS 1	False	TAS 3	True	TAS 1	7.98	9.88	194.33	249.24	False
S120	0	TAS 1	False	TAS 3	True	TAS 5	7.98	9.88	1791.23	236.9	True
S121	20	TAS 1	False	TAS 3	True	TAS 5	7.98	12.04	1792.98	236.9	True
S122	0	TAS 1	False	TAS 3	False	TAS 1	8.01	9.88	225.98	362.05	False
S123	20	TAS 1	False	TAS 3	False	TAS 1	8.01	9.88	220.12	362.05	False
S124	0	TAS 1	False	TAS 3	False	TAS 5	8.01	14.08	215.76	354.71	True
S125	20	TAS 1	False	TAS 3	False	TAS 5	8.01	19.39	1350.86	354.71	True
S126	0	TAS 1	False	TAS 4	True	TAS 4	7.98	2291.71	3595.84	455.06	True
S127	20	TAS 1	False	TAS 4	True	TAS 4	7.98	2200.74	3594.52	455.06	True
S128	0	TAS 1	False	TAS 4	True	TAS 5	7.98	1595.78	2404.99	345.06	True
S129	20	TAS 1	False	TAS 4	True	TAS 5	7.98	1597.26	2509.72	345.06	True
S130	0	TAS 1	False	TAS 4	False	TAS 4	8.01	2196.32	3636.4	562.87	True
S131	20	TAS 1	False	TAS 4	False	TAS 4	8.01	2164.62	3556.41	562.87	True
S132	0	TAS 1	False	TAS 4	False	TAS 5	8.01	1549.26	2389.7	562.87	True
S133	20	TAS 1	False	TAS 4	False	TAS 5	8.01	1558.86	2594.31	562.87	True
S134	0	TAS 1	False	TAS 5	True	TAS 4	7.98	7.73	3414.71	454.74	True
S135	20	TAS 1	False	TAS 5	True	TAS 4	7.98	9.48	3415.03	454.74	True
S136	0	TAS 1	False	TAS 5	True	TAS 5	7.98	10.77	117.18	154.74	False
S137	20	TAS 1	False	TAS 5	True	TAS 5	7.98	9.91	115.59	154.74	False
S138	0	TAS 1	False	TAS 5	False	TAS 4	8.01	10.17	3433.56	246.55	True
S139	20	TAS 1	False	TAS 5	False	TAS 4	8.01	19.57	3428.31	246.55	True
S140	0	TAS 1	False	TAS 5	False	TAS 5	8.01	9.88	122.4	146.55	False
S141	20	TAS 1	False	TAS 5	False	TAS 5	8.01	11.4	129.96	146.55	False
S142	0	TAS 1	False	FP 1	True	TAS 1	7.98	9.88	112.87	141.46	False
S143	20	TAS 1	False	FP 1	True	TAS 1	7.98	9.88	106.12	141.46	False
S144	0	TAS 1	False	FP 1	True	FP 1	8.01	9.87	27.25	35.32	False
S145	20	TAS 1	False	FP 1	True	FP 1	8.01	9.87	26.35	35.32	False
S146	0	TAS 1	False	FP 1	False	TAS 1	8.01	9.87	109.0	133.27	False
S147	20	TAS 1	False	FP 1	False	TAS 1	8.01	9.88	106.35	133.27	False
S148	0	TAS 1	False	SP 2	True	SP 1	23.85	30.74	100.56	120.25	False
S149	20	TAS 1	False	SP 2	True	SP 1	23.85	30.73	97.81	120.25	False

Evaluation Summary (continued)

Setting	W [µs]	D1	Sync1	D2	Sync2	D3	P. BC [µs]	M. BC [µs]	M. WC [µs]	P. WC [µs]	P. RU
S150	0	TAS 1	False	FP 2	True	FP 1	23.85	30.74	97.56	109.05	False
S151	20	TAS 1	False	FP 2	True	FP 1	23.85	30.71	102.99	109.05	False
S152	0	TAS 2	True	TAS 1	True	TAS 1	67.02	167.37	174.29	188.48	False
S153	20	TAS 2	True	TAS 1	True	TAS 1	47.02	148.73	174.28	188.48	False
S154	0	TAS 2	True	TAS 1	False	TAS 1	22.35	95.7	196.02	231.24	False
S155	20	TAS 2	True	TAS 1	False	TAS 1	7.98	78.24	181.59	231.24	False
S156	0	TAS 2	False	TAS 1	True	TAS 1	71.43	75.73	187.63	211.19	False
S157	20	TAS 2	False	TAS 1	True	TAS 1	57.06	61.22	181.58	211.19	False
S158	0	TAS 2	False	TAS 1	False	TAS 1	23.38	24.04	191.91	224.0	False
S159	0	TAS 3	True	TAS 1	True	TAS 1	67.02	67.38	124.28	223.48	False
S160	20	TAS 3	True	TAS 1	True	TAS 1	46.39	59.0	170.63	223.48	False
S161	0	TAS 3	True	TAS 1	False	TAS 1	7.98	9.9	141.79	281.24	False
S162	20	TAS 3	True	TAS 1	False	TAS 1	8.35	9.89	181.17	281.24	False
S163	0	TAS 3	False	TAS 1	True	TAS 1	57.06	59.0	172.01	196.19	False
S164	0	TAS 3	False	TAS 1	False	TAS 1	8.38	9.9	185.01	209.0	False
S165	20	TAS 3	False	TAS 1	False	TAS 1	8.38	9.88	176.82	208.89	False
S166	0	FP 1	True	TAS 1	True	TAS 1	67.02	67.36	170.02	188.48	False
S167	20	FP 1	True	TAS 1	True	TAS 1	47.02	58.95	167.11	188.48	False
S168	0	FP 1	True	TAS 1	True	FP 1	8.01	9.86	102.51	133.28	False
S169	20	FP 1	True	TAS 1	True	FP 1	8.01	9.87	102.32	133.28	False
S170	0	FP 1	True	TAS 1	False	TAS 1	8.01	9.87	188.09	231.24	False
S171	20	FP 1	True	TAS 1	False	TAS 1	8.01	9.88	178.53	231.24	False
S172	0	FP 1	True	FP 1	True	TAS 1	67.02	67.56	70.04	88.48	False
S173	20	FP 1	True	FP 1	True	TAS 1	47.02	49.67	70.02	88.48	False
S174	0	FP 1	True	FP 1	True	FP 1	8.04	9.87	26.4	36.43	False
S175	20	FP 1	True	FP 1	True	FP 1	8.04	9.87	26.15	36.43	False
S176	0	FP 1	True	FP 1	False	TAS 1	8.04	9.87	108.56	134.38	False
S177	20	FP 1	True	FP 1	False	TAS 1	8.04	9.86	109.49	134.38	False
S178	0	FP 1	False	TAS 1	True	TAS 1	57.09	62.92	170.02	197.19	False
S179	20	FP 1	False	TAS 1	True	TAS 1	57.09	59.46	169.86	197.19	False
S180	0	FP 1	False	TAS 1	True	FP 1	8.04	9.87	95.35	112.05	False
S181	20	FP 1	False	TAS 1	True	FP 1	8.04	9.87	104.31	112.05	False
S182	0	FP 1	False	TAS 1	False	TAS 1	8.04	9.88	184.95	210.0	False
S183	20	FP 1	False	TAS 1	False	TAS 1	8.04	9.9	165.36	210.0	False
S184	0	SP 2	True	SP 1	True	SP 1	23.88	30.74	102.62	132.55	False
S185	20	SP 2	True	SP 1	True	SP 1	23.88	30.74	107.26	132.55	False
S186	0	SP 2	True	SP 1	True	TAS 1	67.02	67.72	170.02	188.48	False
S187	20	SP 2	True	SP 1	True	TAS 1	47.02	50.47	170.02	188.48	False
S188	0	SP 2	True	TAS 1	True	SP 1	23.85	95.88	106.26	124.48	False
S189	20	SP 2	True	TAS 1	True	SP 1	23.85	78.64	106.67	120.25	False
S190	0	SP 2	True	TAS 1	True	TAS 1	67.02	167.49	172.58	188.48	False
S191	20	SP 2	True	TAS 1	True	TAS 1	47.02	149.54	172.42	188.48	False
S192	0	SP 2	True	TAS 1	False	TAS 1	23.85	97.6	190.26	211.24	False
S193	20	SP 2	True	TAS 1	False	TAS 1	23.85	83.03	186.54	207.01	False
S194	0	FP 2	True	TAS 1	True	TAS 1	67.02	167.4	172.59	188.48	False
S195	0	FP 2	True	TAS 1	True	FP 1	23.85	95.86	103.52	133.28	False
S196	0	FP 2	True	TAS 1	False	TAS 1	23.85	98.3	189.24	231.24	False