Mälardalen University Licentiate Thesis 338

Enhancing TSN Adoption by Industry:

Tools to Support Migrating Ethernet-Based Legacy Networks into TSN

Daniel Bujosa Mateu

Mälardalen University

Mälardalen University Press Licentiate Theses
No. 338

ENHANCING TSN ADOPTION BY INDUSTRY:

TOOLS TO SUPPORT MIGRATING ETHERNET-BASED LEGACY NETWORKS INTO TSN

Daniel Bujosa Mateu

2023

School of Innovation, Design and Engineering

ISBN 978-91-7485-586-9
ISSN 1651-9256
Printed by E-Print AB, Stockholm, Sweden

“The improvement of understanding is for two ends:
first, our own increase of knowledge;
secondly, to enable us to deliver that knowledge to others."
John Locke

Acknowledgements

I am deeply grateful to my supervisors Mohammad, Julián, Alessandro, and Thomas for providing me with the opportunity to grow and develop as a researcher. Their guidance, support, and encouragement throughout the Licentiate have been invaluable, and I appreciate the effort they invested in making the journey enjoyable and productive.

I also want to express my heartfelt thanks to my family and friends, particularly my parents and brother, for their constant support and love, despite the distance between us. Their unwavering belief in me has been a driving force throughout my academic journey.

Last, but no least, I am deeply thankful to my fiancée for her invaluable support and understanding. She has been a constant source of motivation and has made this journey an unforgettable experience.

This research is supported by the Swedish Knowledge Foundation (KKS) under the FIESTA project and the Swedish Governmental Agency for Innovation Systems (VINNOVA) under the DESTINE and PROVIDENT projects. Julián Proenza was supported by Grant pid2021-124348ob-i00, funded by MCIN/AEI/ 10.13039/501100011033 / ERDF, EU.

Daniel Bujosa Mateu
April 2023
Västerås, Sweden

Abstract

New technologies present opportunities and challenges for industries. One major challenge is the ease, or even feasibility, of its adoption. The Time-Sensitive Networking (TSN) standards offer a range of features relevant to various applications and are key for the transition to Industry 4.0. These features include deterministic zero-jitter, low-latency data transmission, transmission of traffic with various levels of time-criticality on the same network, fault tolerance mechanisms, and advanced network management allowing dynamic reconfiguration.

This thesis aims to develop tools and mechanisms that enable the industry to adopt TSN easily and efficiently. Specifically, we facilitate the migration of legacy networks to TSN, enabling the preservation of most of the legacy networks and solutions while reducing costs and adoption time. Firstly, we introduce LETRA (Legacy Ethernet-based Traffic Mapping Tool), a tool for mapping Ethernet-based legacy traffic to the new TSN traffic classes. Secondly, we develop HERMES (Heuristic Multi-queue Scheduler), a heuristic Time-Triggered (TT) traffic scheduler that can meet the characteristics of legacy networks and provide quick results suitable for reconfiguration. Thirdly, we develop TALESS (TSN with Legacy End-Stations Synchronization), a mechanism to avoid adverse consequences caused by the lack of synchronization between legacy networks and TSN. Finally, we improve the Stream Reservation Protocol (SRP) to enhance Audio-Video Bridging (AVB) traffic configuration in terms of termination and consistency.

Sammanfattning

Uppfinningen av ångmaskinen i slutet av 1700-talet markerade början på en kontinuerlig och snabb process av automatisering och förbättring inom industrin, som tog ytterligare fart i och med införandet av datorstyrda maskiner och robotik i mitten av 1900-talet. Moderna fabriker och deras produkter är beroende av hundratals specialiserade processorer, inklusive sensorer, ställdon och styrenheter, som samarbetar för att utföra uppgifter och tillhandahålla tjänster. Dessa processorer är beroende av kommunikations-subsystem för att samordna och dela resurser. Dessa system, som vanligtvis kallas distribuerade system, omger oss och används av de flesta människor hela tiden. Exempel på distribuerade system finns i en mängd olika exempel, från moderna bilar till fabriker som producerar olika varor. I en bil finns det till exempel många olika sensorer och ställdon som hastighetsmätare, positionssensorer, bränsleinsprutare och tändspolar, som alla arbetar tillsammans för att se till att bilen fungerar säkert och effektivt, medan det i fabriker används robotarmar, transportband och andra anordningar för att automatisera produktionsprocessen och öka effektiviteten.

Den snabba utvecklingen av tekniken kan dock göra det svårt för företag att hålla jämna steg med de senaste verktygen och systemen, eftersom kostnaden för att införa tekniken kanske inte är kostnadseffektiv, inte bara på grund av att tekniken måste förvärvas, utan också på grund av de förändringar som införandet kräver i andra system som samarbetar. Till exempel skulle införandet av ett nytt kommunikations-subsystem kräva att alla enheter som använder det anpassas. Dessutom kräver uppgraderingen till nyare teknik ofta betydande resurser, inte bara ekonomiska utan även naturresurser. Detta kan leda till ökat avfall och ökade koldioxidutsläpp, vilket utgör en risk för miljön. Dessutom kan följderna av teknikuppgraderingar, till exempel bortskaffande av föråldrad utrustning och produktion av e-avfall, ha ytterligare miljöpåverkan.

I den här avhandlingen fokuserar vi på Time Sensitive Networking (TSN), en ny kommunikationsstandard med betydande fördelar för den framväxande tekniken. Även om TSN-tekniken ger många fördelar, bland annat högre kom-

munikationshastighet och lägre latenstider, saknar många nuvarande industrisystem mjuk- och hårdvarukraven för att stödja denna teknik. Målet med vår forskning är därför tvåfaldigt: för det första att förbättra TSN's mekanismer för att göra den mer attraktiv för industrin och för det andra att utveckla verktyg som möjliggör en sömlös migration och integration av äldre system till TSN, så att slutstationerna kan utnyttja fördelarna med TSN utan att behöva byta ut eller uppgradera större delen av systemet. Detta tillvägagångssätt sparar värdefull tid och resurser och minskar det avfall som uppstår under processen.

List of Publications

Papers included in this thesis[1]

Paper A: Daniel Bujosa, Mohammad Ashjaei, Alessandro V. Papadopoulos, Julián Proenza, Thomas Nolte. "*LETRA: Mapping Legacy Ethernet-Based Traffic into TSN Traffic Classes.*" In the 26^{th} IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2021).

Paper B: Daniel Bujosa, Mohammad Ashjaei, Alessandro V. Papadopoulos, Julián Proenza, Thomas Nolte. "*HERMES: Heuristic Multi-queue Scheduler for TSN Time-Triggered Traffic with Zero Reception Jitter Capabilities.*" In Proceedings of the 30^{th} International Conference on Real-Time Networks and Systems (RTNS 2022).

Paper C: Daniel Bujosa, Daniel Hallmans, Mohammad Ashjaei, Alessandro V. Papadopoulos, Julián Proenza, Thomas Nolte. "*Clock Synchronization in Integrated TSN-EtherCAT Networks.*" In the 25^{th} IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2020).

Paper D: Daniel Bujosa, Mohammad Ashjaei, Alessandro V. Papadopoulos, Julián Proenza, Thomas Nolte. "*Improved Clock Synchronization in TSN Networks with Legacy End-Stations.*" Technical report at Mälardalen University, Sweden, pending for submission to a journal.

Paper E: Daniel Bujosa, Inés Álvarez, Julián Proenza. "*CSRP: An Enhanced Protocol for Consistent Reservation of Resources in AVB/TSN.*" In the IEEE Transactions on Industrial Informatics 17 (TII 2020).

[1]The included papers have been reformatted to comply with the thesis layout.

Other Relevant publications[2]

Paper F: Inés Álvarez, Luis Moutinho, Paulo Pedreiras, Daniel Bujosa, Julián Proenza, Luis Almeida. "*Comparing admission control architectures for real-time Ethernet.*" In the IEEE Access, 2020, vol. 8.

Paper G: Daniel Bujosa, Mohammad Ashjaei, Alessandro V. Papadopoulos, Julián Proenza, Thomas Nolte. "*Work-in-Progress: The Effects of Clock Synchronization in TSN Networks with Legacy End-Stations.*" In the 27^{th} IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2022).

[2]Not included in this thesis.

Contents

I

Thesis

Chapter 1

Introduction

The emergence of novel technologies presents potential solutions and enhancements to industries that can confer a competitive edge by minimizing costs, improving products, or promoting environmental sustainability. By leveraging such technologies, businesses can increase performance, optimize resource utilization, and reduce harmful emissions. Nevertheless, embracing these opportunities is not without obstacles. The challenges often stem from the practicality and feasibility of integrating the new technologies into the industry.

Time-Sensitive Networking (TSN) is a new technology that has the potential to reshape industrial communications and facilitate the transition to Industry 4.0. The development of TSN can be traced back to the creation of the IEEE Audio-Video Bridging (AVB) Task Group (TG) in 2005. This group focused on adding real-time capabilities to Ethernet for audio and video streaming. The AVB TG developed three projects: IEEE Std 802.1AS [5] for clock synchronization, IEEE Std 802.1Qav [1] for Credit-Based Shaping (CBS), and IEEE Std 802.1Qat [2] for the Stream Reservation Protocol (SRP). To ensure minimum Quality of Service (QoS) when using the aforementioned standards, the AVB TG also created a set of rules called IEEE Std 802.1BA-2011: Audio Video Bridging Systems [3]. Together, these standards are known as the AVB standards.

As interest in AVB technology grew beyond audio and video streaming, the AVB TG was renamed to TSN TG in 2012[1]. The TSN standards are an expansion of the AVB standards to meet the needs of additional applications, such as automotive[49], automation[59], and energy distribution[48]. TSN offers a variety of compelling features, including support for mixed hard and soft real-time communications, flexibility of traffic requirements, and fault tolerance mechanisms. These features enable TSN to provide new solutions within

[1] https://1.ieee802.org/tsn/

modern industrial systems, such as increased bandwidth, improved real-time behavior, improved fault tolerance, and even the integration of multiple legacy networks into a single TSN network.

However, most Ethernet-based communication networks nowadays utilize a variety of devices and protocols that cannot be seamlessly integrated with TSN. Additionally, due to hardware limitations, several of these networks would not be able to support TSN integration. Therefore, the adoption of TSN by the industry requires significant investments in both software and hardware upgrades. Such modifications often result in time and resource costs that may not be cost-effective. Furthermore, the upgrade process leads to the generation of a substantial amount of technology waste, consisting of relatively good-condition equipment that could have been reused.

In this work, our objective is to develop tools and mechanisms to facilitate the migration and integration of legacy networks with TSN. This integration can be done in different ways, such as through the use of gateways. However, this would not allow the legacy networks to take advantage of other TSN features such as higher bandwidth or low jitter. Thus, we propose to directly replace the communication subsystem of the legacy network, i.e., all devices exclusively responsible for communication (excluding the end-stations), with TSN. This integration technique ensures that legacy end-stations can keep their communication behavior and protocols, agnostic of the change, leading to improved integration of various legacy networks and enabling them to leverage the advantages of TSN potential features.

In order to achieve the desired migration and integration of legacy networks with TSN, several challenges should be tackled. Firstly, legacy traffic needs to be identified and defined accurately. Secondly, the legacy traffic should be mapped into the different TSN traffic types, based on its specific requirements. Finally, the traffic mapped as TSN TT traffic should be scheduled accordingly. Furthermore, if scheduling is required, it is necessary to have a global view of time, and thus proper synchronization between the legacy network and TSN must be ensured.

In this thesis we address the above challenges by proposing novel and efficient solutions. Firstly, we identify Ethernet-based trafic parameters relevant for mapping legacy traffic into TSN. Secondly, we propose a mapping algorithm that can efficiently map Ethernet-based traffic into TSN trafic classes by taking into account their timing requirements. Finally, we develop a TT traffic scheduling algorithm that conforms to the requirements of legacy networks and a synchronization mechanism that enables the communication of legacy scheduled traffic even when the legacy end-stations and TSN are not synchronized.

Outline. The Licentiate thesis is organized as follows. Chapter 2 provides the necessary background for the understanding of this work. Chapter 3 presents the problems identified and addressed in the thesis. Chapter 4 presents the related work. Chapter 5 describes the research method used in the work. Chapter 6 lists the challenges and contributions covered in this thesis and, finally, Chapter 7 concludes the paper and presents future directions.

Chapter 2

Background

This chapter summarizes the main mechanisms involved in the work and the background necessary to facilitate its understanding.

2.1 TSN traffic classes

TSN end-stations communicate by transmitting Ethernet frames through routes consisting of links and TSN switches. TSN end-stations and switches are nodes that support clock synchronization and traffic shaping. TSN devices feature output ports that support up to eight First-In-First-Out (FIFO) queues, each associated with a specific priority level. TSN supports three traffic classes: Time-Triggered (TT) traffic, Audio-Video Bridging (AVB) traffic, and Best-Effort (BE) traffic. Each priority level is assigned to one of these traffic classes depending on the shaping mechanisms it applies. However, it is common that TT traffic has the highest priority, while BE traffic has the lowest priority. It is important to note that multiple queues can cover the same traffic class. For example, AVB traffic can consist of classes A, B, and C, each associated with a distinct priority level.

Figure 2.1 shows an example of a TSN device output port with four queues configured as TT traffic with the highest priority, AVB classes A and B traffic with the second and third highest priority, and a BE traffic class as the lowest priority. This is determined by the mechanisms applied to each queue, including Time-Aware Shaper (TAS) and CBS, which will be described in depth below.

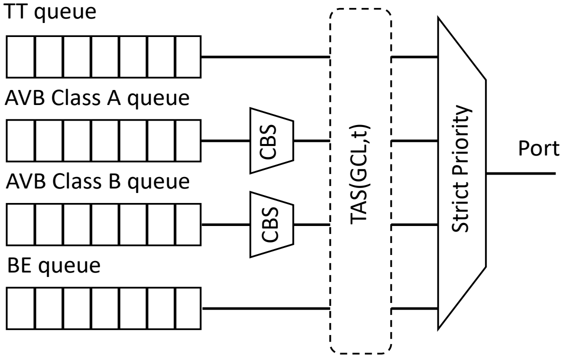

Figure 2.1: A TSN output port with four FIFO queues: one TT queue, two AVB queues, and one BE queue.

2.1.1 Time-Triggered Traffic

TT traffic is transmitted in accordance with a fixed offline schedule that specifies the exact time slot each TT frame is transmitted. To prevent interference between frames, TT traffic uses the TAS mechanism shown in Figure 2.1 as defined in IEEE 802.1Qbv [30]. The TAS mechanism associates each queue with a gate that can be opened or closed. Frames in a queue can be transmitted only when the gate is open; otherwise, they are blocked. The Gate Control List (GCL) is responsible for managing the gates, specifying precisely when they should open and close. The GCL is a cyclic list that repeats the schedule, with each entry in the list specifying the precise time at the nanosecond level when the gate should be open or closed. The term *transmission window* refers to the time interval during which a gate is open.

Figure 2.2 depicts an example of how the TAS operates for two TT queues with different priorities (6 and 7). The example assumes the transmission of three TT frames with a period of 4 time units through a switch port. One of the frames is set to the highest priority 7 (blue), while the other two frames are set to priority 6 (red and green). The figure shows the gates for the two queues and their states at different time slots based on the GCL, which specifies when the gates should be open or closed. During the first time slot (T0 to T1), the gate for the priority 6 queue is open (1 in GCL), while the gate for priority 7 is closed (0 in GCL), allowing the transmission of the red frame. In the second time slot (T1 to T2), the gate for priority 7 is open, allowing the transmission of the blue frame. Both gates are closed in the third time slot (T2 to T3), preventing any transmission. Finally, in the last time slot, the

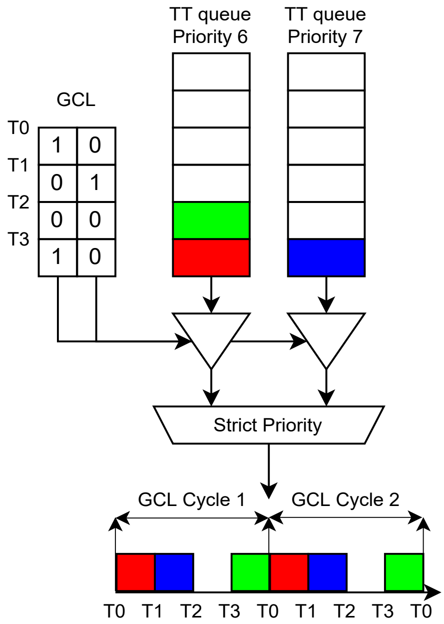

Figure 2.2: TSN TAS gate mechanism.

gate for the priority 6 queue is open, allowing the transmission of the green frame. The figure also shows two cycles of frame transmission at the bottom, demonstrating how the GCL repeats the schedule.

2.1.2 Audio-Video Bridging Traffic

The AVB TG [1] presented the CBS that applies credits to AVB queues. Credits are consumed when a frame in that queue is transmitted, and are replenished when there is a pending frame in the queue or if the credit is negative, even if no frame is waiting in the queue. These consumption and replenishment ratios, which are configured in the CBS, are constant and determine the bandwidth reserved for each AVB queue. To transmit data, AVB queues need to have a positive or zero credit, and their gate must be open based on the TAS and GCL. CBS usually separates classes into A and B, and allows lower-priority traffic transmission even if higher-priority traffic is waiting, based on the credits. This leads to improved QoS for lower-priority traffic and reduced buffering. Despite the unknown activation time of AVB traffic due to potential blocking from other AVB classes or ST queues, there are techniques to estimate its worst-case response time [11].

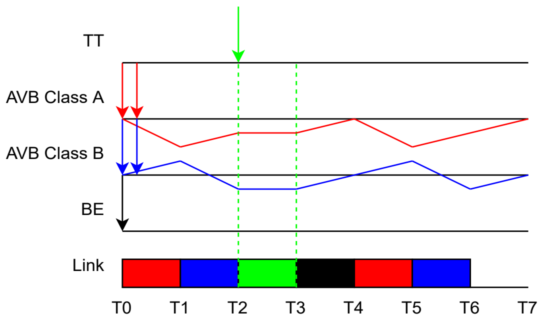

Figure 2.3: TSN CBS mechanism.

The operations of CBS for two AVB queues (Classes A and B) interacting with one higher priority TT queue and one lower priority BE queue are illustrated in Figure 2.3. The downward arrows in the figure indicate when a frame arrives at its respective queue of an output port, the vertical dashed lines indicate the TT traffic window, and the slopes indicate the credit evolution. The figure demonstrates that at T0, two frames of each AVB class and one BE frame are ready to be transmitted. As the credit of both AVB queues is zero, class A begins transmitting the first frame because it has higher priority. This consumes its credit while increasing the credit of AVB class B. By T1, the credit of AVB queue class A becomes negative, causing class B to begin transmitting, even though a higher priority frame is waiting. At T2 a frame arrives at the TT queue as scheduled, hence GCL closes all gates but the corresponding to the TT queue to ensure its transmission without interruptions. Note that, during the transmission of TT traffic, the credits are frozen, i.e. not changing. After transmitting the TT frame at T3, the AVB traffic credit remains negative, allowing the BE frame to be transmitted even though two higher priority frames are waiting. Finally, the credits are replenished during the transmission of the BE frame. Therefore, at T4 and T5, AVB frames of classes A and B are transmitted as in T0 and T1.

2.1.3 Best-Effort traffic

The lowest priority traffic type, BE, does not offer any real-time guarantees. A queue that carries BE traffic is not shaped by CBS and can only transmit frames if its gate is open, and all other AVB queues have negative credit or if there is no AVB traffic available for transmission. This behavior is illustrated in Figure 2.3 at time T3 to T4 where the credits for both classes A and B are negative.

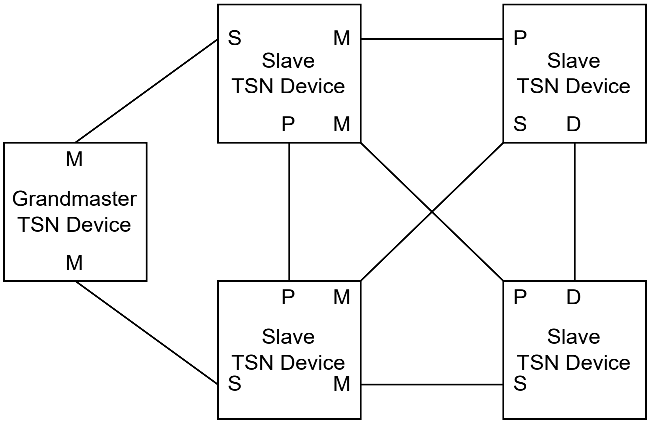

Figure 2.4: Example of TSN time-synchronization spanning tree.

2.2 TSN Clock Synchronization

The IEEE 802.1AS standard [5] describes the gPTP mechanism that provides TSN clock synchronization, which is composed of three key parts: the Best Master Clock Algorithm (BMCA), the Propagation Delay Measurement (PDM) mechanism, and the Transport of Time-synchronization Information (TTI). The BMCA determines the grandmaster clock, which acts as the reference clock in the TSN network, and establishes the hierarchy between TSN devices (TSN end-stations and switches). After establishing the hierarchy, the PDM mechanism is used to measure the propagation delay between TSN devices. Finally, the TTI mechanism is used to disseminate the grandmaster time to synchronize the other TSN devices. The following subsections provide a detailed description of each of the three mechanisms.

2.2.1 BMCA

The BMCA algorithm constructs a spanning tree for time synchronization, with the grandmaster TSN device as the root. An example of this, is illustrated in Figure 2.4. In this tree, each TSN device can act as either a grandmaster or a slave, and each port can be categorized as a Master port (M), Slave port (S), Passive port (P), or Disabled port (D). To determine these behaviors, each system periodically broadcasts a special message called `announce message`. This message contains various parameters, but we focus on two parameters, the `systemIdentity` and the `stepsRemoved`. The `systemIdentity` parameter indicates the accuracy of the sender's

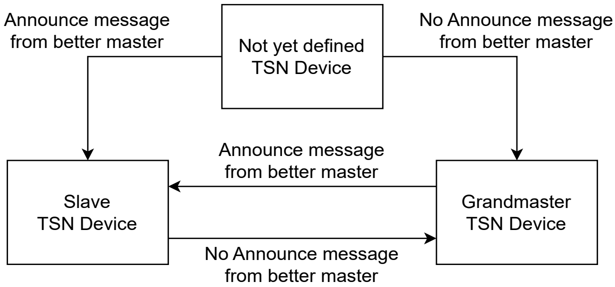

Figure 2.5: Time-aware system BMCA evolution.

clock, while `stepsRemoved` parameter denotes the distance between the transmitter and the receiver. Specifically, the `stepsRemoved` value is incremented each time the `announce message` is forwarded. For instance, consider a line topology consisting of three TSN devices, where the first device sends its `announce message`. The second device receives this message with the `stepsRemoved` parameter value of 0, but before forwarding it to the next device, it increments the `stepsRemoved` value. Thus, the last TSN device receives the `announce message` sent by the first device with a `stepsRemoved` parameter value of 1.

The diagram in Figure 2.5 illustrates how a TSN device can either act as a slave or grandmaster clock if it does not have an assigned role. A TSN device becomes a slave if it receives an `announce message` from a better clock, i.e., a message with a greater `systemIdentity` parameter. Conversely, if the TSN device does not receive any `announce message` from a better clock within a defined period (defined by the periodicity of the `announce message` transmission), it becomes the grandmaster clock. Similarly, a grandmaster or a slave TSN device can switch roles based on the reception of an `announce message` from a better clock or lack of it. On the other hand, the proximity to the grandmaster determines the roles of ports in a TSN device, which can be determined using the `stepsRemoved` parameter. The port closest to the grandmaster becomes the slave port, and only one port in the TSN device can have this role. The port closest to the grandmaster clock in a link becomes the master port. Disabled ports are those that are explicitly disabled, while ports that are neither master, slave, nor disabled are passive ports.

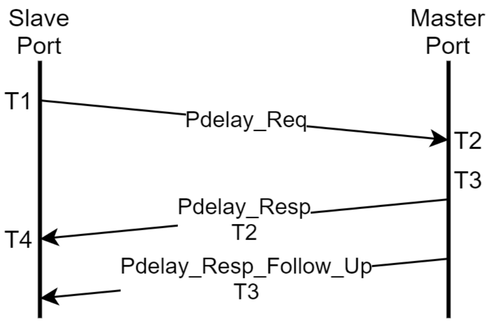

Figure 2.6: PDM diagram.

2.2.2 PDM

Once the spanning tree has been created with the grandmaster as the root, PDM is used by the slaves to calculate the propagation delays between its slave port and the master port of the TSN device connected to it. The process is illustrated in Figure 2.6. The PDM process begins with one slave transmitting a delay request message `Pdelay_Req` via its slave port to another TSN device, which may be the grandmaster or another slave. The initiating slave records the time at which the message is sent ($T1$). The receiving TSN device receives the message through its master port, records the time at which the message is received ($T2$), and transmits $T2$ back to the initiating slave while recording the transmission time ($T3$). The initiating slave receives $T2$ and records the time at which it is received ($T4$). Finally, the receiving TSN device transmits $T3$ to the initiating slave, allowing the latter to calculate the delay as $Delay = \frac{(T4-T1)-(T3-T2)}{2}$.

2.2.3 TTI

After the creation of the spanning tree with the grandmaster as the root and the measurement of the delays by the slave TSN devices, TTI is initiated. TTI involves the TSN devices transmitting their local time via their respective master ports to the slave TSN devices connected to them. The TSN devices, which receive the message via their slave ports, add the previously measured delay and update their local time accordingly.

2.3 Stream Reservation Protocol

The TSN TG relies on SRP in many of its projects, as it plays a crucial role in verifying resource availability within the network and reserving these resources, which enables a bounded end-to-end delay and prevents packet loss caused by the buffer overflow. Furthermore, the flexibility of SRP allows for dynamic traffic requirement modifications during run-time. While there exist three different SRP architectures, this work will focus solely on the distributed version. For additional information regarding the other architectures, please refer to [4].

SRP follows the publisher-subscriber paradigm to enable real-time data communications through streams. In this paradigm, the publisher, known as a *talker*, transmits data to subscribers, known as *listeners*. Each stream in SRP is a logical communication channel that carries traffic defined by a set of parameters such as the frame size and period. For example, if a temperature sensor acts as a talker and wishes to transmit its measurements with a 10 ms period and a 1-byte payload to other end-station (the listeners), the network must first verify that sufficient resources are available. If that is the case, the network will then create a stream with the specified period and payload to transmit the sensor data.

It should be emphasized that decisions regarding the reservation of resources are solely made based on local information. However, there is crucial information related to the reservations that need to be distributed throughout the network. For instance, this information may include the amount of resources required for a stream or whether a particular switch has sufficient resources available. Such information is transmitted via specialized messages referred to as talker and listener attributes.

An example of the SRP mechanism in a linear network topology is described in Figure 2.7. The network comprises a talker (T), a listener (L), and two switches (S1 and S2). Before transmitting frames, the talker must create a stream by broadcasting a Talker Advertise (TA) message. This message includes stream identification and resource requirements which are used in the Admission Control (AC) by the rest of the devices of the network to check whether there are enough resources for the stream to be created. Switches receiving the message check output ports for resource availability. If an output port has insufficient resources, it sends a Talker Failed (TF) message with the reason for failure. If the output port has enough resources, it forwards the TA message to the next device.

End-stations that are not interested in the stream do not take any further action on receiving the TA or TF messages. However, if a end-station wants

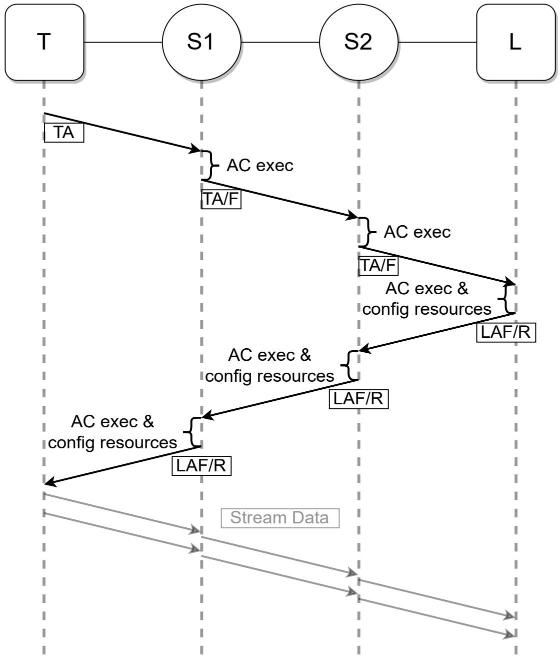

Figure 2.7: Time diagram of the resource reservation mechanism in a network with a linear topology.

to become a listener, it checks its resources and sends a Listener Ready (LR) message to the switch if it has enough resources. If it does not have enough resources, it sends a Listener Asking Failed (LAF) message. The switch receiving the LR or LAF messages checks its resources again and forwards a combined response to the talker: (i) if all ports receive LR messages, the switch transmits an LR message to the talker; (ii) if all ports receive LAF messages, it transmits another LAF message to the talker; and (iii) if there is a mix of LR and LAF messages, it transmits a Listener Ready Failed (LRF) message to the talker.

Finally, the talker waits for an LR or LRF message before starting data transmission. Additionally, it can delete the stream at any time using the unadvertise stream mechanism, which is broadcast to all devices.

Chapter 3

Problem Formulation

In this work, we aim to facilitate the adoption of TSN by the industry via developing tools and mechanisms to enable the migration and integration of legacy networks with TSN. While there are several approaches to adopt TSN, including complete replacement of the existing network with TSN software and hardware or using TSN as a backbone while connecting all legacy networks through gateways, these options have limitations. The former is a resource-intensive process, whereas the latter does not allow the legacy networks to benefit from TSN features. Therefore, we propose replacing the communication subsystems of legacy networks, i.e., the set of devices exclusively responsible for communication excluding the end-stations, with a single TSN network, while ensuring that the legacy end-stations, applications, communication protocols, and traffic with different timing requirements continue to operate as effectively as before, if not better. This approach allows the legacy end-stations to take advantage of the benefits offered by TSN, such as high bandwidth, support for mixed hard and soft real-time communications, flexibility of traffic requirements, and fault tolerance mechanisms. Our proposed method enables the migration and integration of legacy networks with TSN, thereby facilitating a smooth transition to this technology.

To enable the industry to adopt TSN solutions, and achieve the desired integration, a proper migration and integration methodology of the legacy traffic to TSN must be designed. This poses several challenges, which are addressed in this work. Regarding the traffic migration, the legacy traffic must be classified into the three previously mentioned TSN traffic classes. Such mapping must meet the timing requirements of the legacy traffic. On the other hand, regarding the legacy networks integration, the traffic classified as TT due to its high time requirements must be scheduled, considering its legacy characteristics. In addition, traffic scheduling must be scalable to allow for the migration of large networks and must be fast in case new or legacy reconfiguration mechanisms need to be supported. Finally, the TSN schedule must be aligned with the transmission of the frames in the legacy network even when synchronization between the legacy end-stations and the TSN network is not possible.

Finally, a previous work [15] showed that SRP lacks termination and consistency, which can result in the inefficient utilization of bandwidth. This can limit the network's ability to integrate with and support legacy networks, thereby hindering its overall scalability. We will next take a closer look at each of the problems to be solved.

3.1 Legacy Ethernet-based Traffic Mapping

The mapping consists of clustering the legacy traffic based on its characteristics into the three types of TSN traffic, including TT, AVB, and BE traffic. In this thesis, we will focus on Ethernet-based legacy traffic. One of the main features of TSN, and the reason behind the growing interest in adopting it by the industry, is its traffic flexibility, i.e., its ability to combine several types of traffic on the same network. Thanks to this feature, TSN seems to be key to advancing the industry to the incipient Industry 4.0 paradigm but also makes it possible for TSN to integrate different legacy networks in the same TSN network. However, each legacy network has particular characteristics that are not directly linked to the different types of TSN traffic; hence a proper mapping methodology is key to the proper migration and integration of legacy networks into TSN. To achieve this, we must identify the characteristics of the legacy traffic that are relevant to defining its behavior in the TSN network and, based on these characteristics, split the traffic among the different types of TSN traffic classes.

3.2 TT Scheduling

As mentioned before, in TSN, a GCL is defined for each queue in each TSN device output port, in such a way that the GCL identifies the moments in which the gate of each queue will be open. The scheduling of TT traffic, and its synthesis in GCLs, is known to be an NP-complete problem [44]. Several solutions are proposed in the literature to schedule TT traffic in TSN networks that are mainly based on Integer Linear Programming (ILP) and Constrained Programming (CP) [10]. These solutions are known to have high time complexity, i.e., they require a long time to schedule large networks, thus they are not generally scalable. In addition, these solutions are not suitable for networks that require dynamic reconfigurations as the new configuration should be created relatively fast. A few heuristic schedulers are also proposed, e.g., [40], whose performance is not properly compared with the ILP and CP solutions. In this work, we seek to develop a heuristic scheduler capable of synthesizing the GCLs of the legacy traffic mapped as TT traffic with acceptable performance and low scheduling times enabling the migration of large legacy networks.

3.3 Legacy Networks Synchronization

TSN TT traffic requires the network to be fully synchronized. Otherwise, different devices could exhibit clock drift, which would cause the transmission and reception of frames to not properly match the TSN schedule. For example, Figure 3.1 shows the effects of transmitting TT traffic between two legacy end-stations that are synchronized via a legacy clock synchronization, through a TSN network that is not synchronized with them. As we can see, the sender transmits frames faster than the TSN network forwards them in its transmission windows. This causes frames to arrive at the receiver increasingly later than their legacy scheduled time (dashed envelopes). Furthermore, since the transmission of frames by the TSN network to the listener is slower than the transmission by the talker to the TSN network, the frames stack up in the buffers. However, the buffers are not infinite, hence frames that arrive once the buffer is full are discarded. Similar phenomenon occurs when the TSN clock is faster than the legacy network one. A more detailed description of the causes and consequences of the effects of the lack of synchronization between legacy end-stations and TSN can be found in paper [16].

Despite the importance of synchronization, in many cases, legacy networks may not be able to implement TSN's synchronization protocols. Therefore, it is necessary to analyze the effects of the lack of synchronization between legacy networks and TSN in heterogeneous networks, i.e. networks combining TSN and legacy end-stations, and to develop mechanisms to avoid such adverse effects.

3.4 Distributed SRP

As mentioned before, paper [15] demonstrated the lack of termination and consistency of the distributed SRP at both the application and infrastructure levels. On the one hand, termination issues are mainly due to the fact that SRP listeners do not inform the bridges nor the talkers when they are not interested in binding to a stream. On the other hand, consistency issues are mainly due to

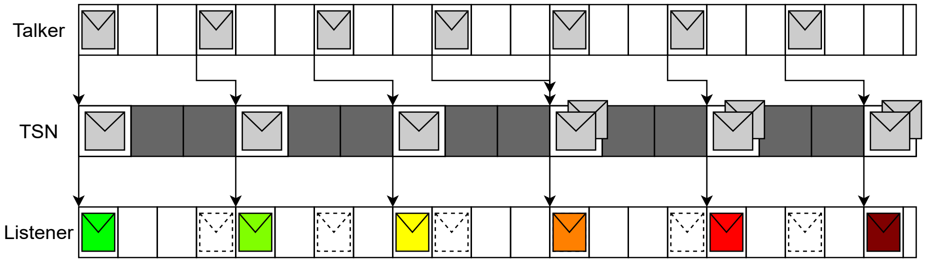

Figure 3.1: Example of positive drift synchronization issue.

the fact that information related to the reservations is propagated in a single direction. That is, the talker attribute transmitted by a talker is forwarded always towards the listeners; while, when listeners and switches reply to a stream declaration, the information is only forwarded towards the talker. Thus, not all the devices involved in the reservation of a stream receive the same information.

This does not directly affect the migration and integration of legacy traffic, but may lead to a waste of bandwidth and resources that limit both the addition of new traffic and the adoption of legacy traffic. In this regard, we need to improve the distributed SRP in order to provide network devices with a consistent view of the reservation of resources so that they can make rather complex decisions within a bounded time avoiding waste of resources..

Chapter 4

State of the Art Review

The TSN TG's work since 2012 has been highly relevant, resulting in the community dedicating a significant amount of research to the study, application, and improvement of TSN. For instance, the work in [6] studied the effects of the TAS, the work in [31] analyzed the fault tolerance issues, while the work in [7] proposed time redundancy to tolerate temporary faults. In addition, the work in [58] studied the scheduling policies, the works [61] and [35] analyzed schedulability of traffic with different TSN features and the load balancing was studied in [8]. Moreover, the work in [10] provided an up-to-date comprehensive survey of the TSN-related research.

There are also a few works on reconfiguration and integration in dynamic system networks. For example, on the one hand, work in [47] analyzes the ability of SDN to accelerate the Time-to-Integrate process in evolving topologies from a synchronization point of view. On the other hand, the works in [27] and [28] aim to provide auto-configuration to TSN by introducing a Configuration Agent into the network, an entity that continuously monitors the network for changes and automatically updates the configuration to adapt to those changes while maintaining the desired quality of service.

Other works on integrating legacy networks into TSN networks are the works in [38] and [51]. The former integrated a few of the TSN standards into Sercos III, which is a closed system that allows standard Ethernet devices to be plugged in, to improve its performance; while the later proposed an integration methodology of wireless TSN (802.11). However, as [18] shows in their experimental setup, there are still many challenges to enable a proper integration between TSN and Non-TSN devices. Therefore, in this work we focus on analyzing the works that address the problems mentioned above.

4.1 Mapping

Regarding traffic mapping, a meta-heuristic method is proposed in [23] that maps mixed-criticality applications into the TSN traffic classes. Although the aim is similar to ours, the proposed method does not cover all cases that are studied and exist in industrial applications. The method, thus, becomes suitable for cases where only very few mixed-criticality levels are assumed in the network with no extensive timing information. Other papers such as [21] only map a specific type of traffic while papers such as [19] analyze the characteristics of different types of TSN traffic and provide guidance on how to perform the mapping. However, to the best of our knowledge, there are no automated tools for mapping legacy traffic based on its characteristics into TSN.

4.2 Scheduling

Within the context of TT traffic scheduling in TSN networks, the works in [50] and [39] present a joint routing and scheduling algorithm formalized as an ILP and as a meta-heuristic scheduling approach based on a Genetic Algorithm (GA) approach, respectively. The work in [20] presents an SMT-based scheduler capable of scheduling networks with several TT queues. The work in [24] proposes a GCL synthesis approach based on Greedy Randomized Adaptive Search Procedure (GRASP) meta-heuristic [45], which takes AVB traffic into consideration, whereas the work in [25] proposes a joint routing and scheduling approach for TT and AVB traffic by means of an integrated heuristic and meta-heuristic strategy. In the latter work, the K-Shortest Path (KSP) method [60] is utilized for routing, and GRASP is used to schedule both TT and AVB at the same time. Moreover, the work in [12] synthesizes a network topology that supports seamless redundant transmission for TT traffic by proposing a greedy heuristic algorithm for joint topology, routing, and scheduling synthesis. Finally, paper [56] provides a comprehensive survey of scheduling techniques and algorithms used in TSN to support time critical applications.

The above-mentioned solutions are mostly based on ILP or constraint programming, while some of them exploit the use of meta-heuristics, e.g., GA. However, these solutions normally are highly time-complex, which makes them not scalable. Few works target heuristic solutions with lower time complexity. For instance, the work in [40] proposes a heuristic routing and scheduling algorithm called Heuristic List Scheduler (HLS) that is limited to a single TT queue, while the work in [57] compares 4 heuristic algorithms combining routing and scheduling (Modified Most Loaded Heuristic (MML), Bottleneck

Heuristic (BN), Coefficient of Variation Heuristic (CV) [9][33], and Modified Dot Product Heuristic (MDP) [41]), all with scheduling times greater than 100 ms and unable to handle multiple queues.

Despite the large amount of work done on scheduling in TSN, most works are not a priori compatible with the specific characteristics of the legacy traffic, such as offsets, drifts, or specific reception jitters. Moreover, solutions are normally time-complex or very limited in terms of schedulability.

4.3 Synchronization

One of the key features of TSN is clock synchronization. Most of the works in the literature are focused on integrating TSN with wireless or 5G networks. For example, in work [29] authors implement a low-overhead beacon-based time synchronization mechanism to provide highly accurate synchronization to the wireless networks, thus they can be used in the context of high determinism TSN networks. Moreover, works [13] and [46] extend IEEE 802.1AS and IEEE 802.11 respectively with the intention of integrating TSN with wireless networks while [52] discusses the integration challenges of Wired TSN and Wireless Local Area Network (WLAN) technologies and proposes a Hybrid TSN device architecture. On the other hand, [26] introduces the integration of TSN time synchronization (IEEE 802.1AS) conform with 5G while [17] proposes a cross domain clock synchronization method based on data packet relay to solve the end-to-end cross domain clock synchronization problems caused by the different 5G-TSN integrated network clock domains. Finally, other works such as [54], [53] and [55] has evaluated the performance of 5G-TSN networks.

According to our state of the art review, and to the best of our knowledge, there is no work addressing the challenges of synchronizing legacy devices onto a TSN network.

4.4 AVB SRP

There are many works related to the study of AVB's efficiency. For example, in [43] the authors apply network calculus to evaluate the real-time performance of Ethernet AVB in automotive networks and in [37] the authors provide insights into the performance of AVB and TSN in automotive Ethernet networks, concluding that both protocols can provide reliable and deterministic communication but their performance depends on network configuration and traffic characteristics. Moreover, paper [36] proposes an extension to the

IEEE 802.1 AVB protocol to allow for the coexistence of synchronous and asynchronous traffic and paper [34] addresses the challenges of designing an IP/Ethernet-based in-car network for real-time applications, suggesting solutions such as network segmentation and quality of service mechanisms for which AVB may be relevant. On the other hand, in the work presented in [42] the authors detect a drawback in the resource reservation de-registration specification, which leads to the waste of the network resources, and proposed some solutions. Moreover, some works present solutions to provide fault tolerance against permanent faults using SRP [32].

Nevertheless, in a previous work [15] we detected the lack of termination and consistency properties in the SRP and how this could cause bandwidth losses or loss of new requests due to overbuffering. Although we proposed some solutions, these were not implemented or evaluated in any way before.

Chapter 5

Research Methodology

The objective of this thesis is the development of methodologies capable of overcoming the challenges presented in this work and their corresponding implementation in tools and mechanisms, either independent or integrated, that facilitate the adoption of TSN by the industry. For this purpose, we followed the hypothetico-deductive [22] research method. Figure 5.1 shows the process of the research.

- **Start**: This project aims to facilitate the industry to adopt TSN both by providing tools and mechanisms for the migration and integration of legacy networks with TSN as well as by eliminating defects in existing mechanisms that could limit such migration and integration.

- **Literature Review**: Once the objective is defined, we perform an extensive state-of-the-art review on Ethernet traffic mapping, TT traffic scheduling and synchronization, and AVB traffic configuration. This review includes papers and standards with the intention of understanding the technology in depth and looking for possible points of conflict for the integration of legacy networks, as well as deficiencies in the existing protocols and mechanisms.

- **Identification of Limitations**: With the knowledge gained through the literature review, we identify the limitations of TSN in terms of migration and integration of legacy networks as well as the performance of its mechanisms. In cases where the review is not sufficient to fully identify the limitations, we raise hypotheses that we verify by running experiments and/or simulations with models.

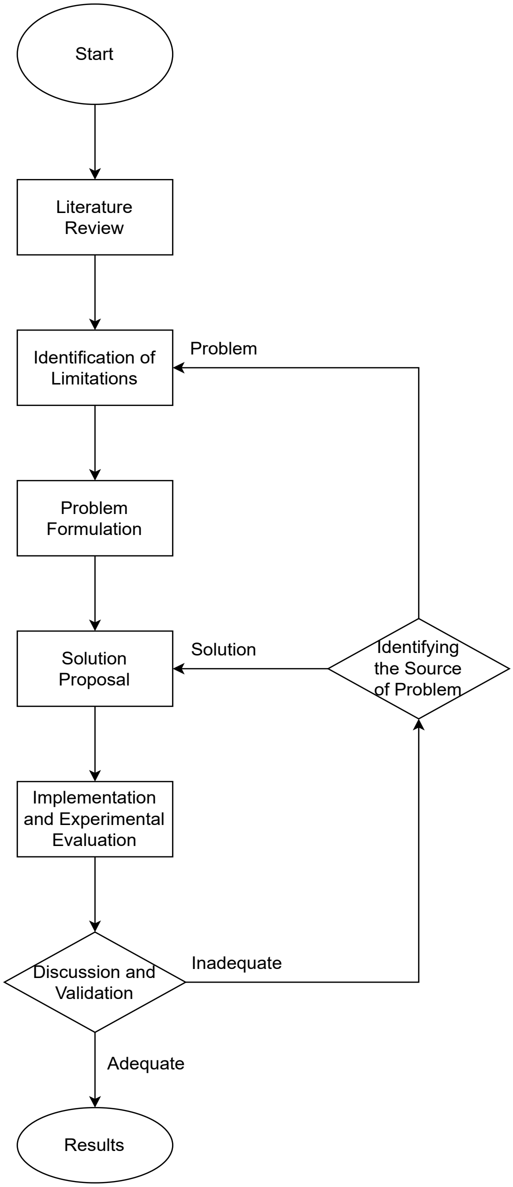

Figure 5.1: Research Methodology

- **Problem Formulation**: In this stage, we identify the problems to be solved and the objectives to address them based on the results obtained during the literature review and the identification of limitations.

- **Solution Proposal**: Once the problems to be solved have been identified and defined, novel solutions which address such problems are discussed and proposed.
- **Implementation and Validation**: The solutions from the previous stage are implemented as as prototypes, tools or models and are evaluated. The effectiveness of the solution is inspected by a feedback loop. If the evaluation output is judged inadequate, we trace back to identify the source of the problem. If the inadequate output is due to the problem formulation, the process is returned and performed from the beginning. Otherwise, the proposed solution is reassessed by refining the current solution or defining a new one. Finally, if the results of the evaluation are deemed adequate, i.e. the limitation identified through the literature review or models and/or experiments in the "identification of limitations" phase has been successfully resolved with a sufficient degree of certainty, the process ends with the publication of the proposed solution. Regarding validation, we conduct experiments for the evaluation of the tools and implemented solutions and formal verification for the validation of the models.

Chapter 6

Thesis Goals and Contributions

The overall goal of this thesis is to enable and facilitate TSN adoption by the industry. To this end, it is essential to develop tools and mechanisms that enable the migration and integration of legacy networks to TSN; otherwise, it would be necessary to replace almost the entire network, which would involve high costs and time.

6.1 Research Gaps and Industry Needs

There has been a growing interest in TSN by the industry in the last few years. This is due to its characteristics which include high bandwidth combined with real-time capabilities, traffic flexibility, and fault tolerance mechanisms, to name a few. These features are key for the integration and interoperability of different levels of operation at the industrial level which seems key for the evolution to an Industry 4.0 paradigm as well as for new products and increasingly large and complex solutions. However, implementing TSN in factories that are already established and in operation or in products may not be cost-effective as it would require changing all their existing devices, networks, and solutions. Therefore, in this work we analyze the limitations of TSN to manage the migration and integration of legacy networks and propose solutions. Looking at the current state of the art we have identified a set of key research gaps given the industry needs towards achieving the overall goal. The limitations identified through the literature review and the experiments and models during the "identification of limitations" phase were synthesized into the following research gaps:

- **RGap**$_1$: In TSN the traffic is divided into 3 types of traffic (TT, AVB, and BE) which in turn can have different priority levels (up to a maximum of 8 priority levels in total). Each of the traffic types has unique characteristics that do not need to have a direct correspondence with the legacy traffic to be migrated or integrated into the new TSN network. Inadequate traffic mapping in the new network may result in not meeting the requirements of the legacy networks.
- **RGap**$_2$: The scheduling of TT traffic, and its synthesis in GCLs, is known to be an NP-complete problem [44]. This leads to either unscalable or low-performance schedulers that, in most cases, do not take into account the specific requirements of legacy traffic.
- **RGap**$_3$: The lack of synchronization is also a problem since drift between clocks causes variations between reserved and required bandwidth. This is especially problematic in the case of TT traffic where such drift can lead to loss of frames or additional delays of up to an entire period.
- **RGap**$_4$: Finally, during the literature review and the search for limitations, we detected termination and consistency issues in the distributed version of the SRP which may lead to a waste of resources that can limit the migration and integration of legacy traffic.

6.2 Research Goals

Given the set of research gaps, we have identified 4 specific research goals of this thesis, i.e., the research goals required to solve the aforementioned problems are:

- **RG**$_1$: Develop a mapping methodology to categorize Ethernet-based traffic into the three types of TSN traffic, including TT, AVB, and BE traffic, with the goal of maximizing its schedulability.
- **RG**$_2$: Develop a scalable heuristic TSN TT traffic scheduler that can handle large volumes of traffic within a reasonable time while maintaining high schedulability using multiple TT queues.
- **RG**$_3$: Develop a synchronization mechanism that eliminates the drift between the TSN network and the legacy network schedule caused by the lack of synchronization. The mechanism should adapt the non-TT transmission windows to adjust transmission and reception ratios, thus

preventing the adverse effects caused by the drift and ensuring seamless integration in terms of synchronization.

- **RG**$_4$: Improve the distributed SRP configuration protocol to provide it with termination and consistency to ensure proper configuration of the TSN AVB traffic avoiding waste of resources.

6.3 Research Contributions

The research contributions in the thesis to address the research goals are as follows:

- **RC**$_1$: We develop a Legacy Ethernet-based Traffic model that can characterize traffic of any Ethernet-based communication protocol. The traffic model will help us to defines the behavior of the legacy traffic by means of a set of parameters that will later be used to map it. This model will partially address RG$_1$.

- **RC**$_2$: We develop a mapping methodology that can map the legacy Ethernet-based frames characterized by the model proposed in RC$_1$ into different TSN traffic classes. We implement the mapping method as a tool, named Legacy Ethernet-based Traffic Mapping Tool or LETRA. We integrated LETRA with TSN traffic scheduling to perform evaluations on different synthetic networks. The results show that the proposed mapping method obtains up to 90% improvement in the schedulability ratio of the traffic compared to an intuitive mapping method on a multi-switch network architecture. Through this contribution we obtain RG$_1$.

- **RC**$_3$: We propose a heuristic scheduler for TT traffic in TSN networks, called Heuristic Multi-queue Scheduler (HERMES), that takes advantage of multiple queues for TT traffic to provide high schedulability with very low scheduling times. Frames in HERMES can be configured to be scheduled in two modes of zero or relaxed reception jitter, which provides better control for users. Through a set of experiments, we show that HERMES can perform better than CP-based solutions, i.e., it results in more schedulable networks, by allowing it to use multiple queues, and at the same time, it provides the results within 17 to 800 times faster. Through this contribution we obtain our goal defined in RG$_2$.

- **RC**$_4$: We formulate the problem of having inconsistent clock synchronization mechanisms in an integrated EtherCAT-TSN network and we describe the effects of this inconsistency in the network behavior. Then, we propose a solution to integrate the clock synchronization mechanisms described by the two network technologies, i.e., EtherCAT and TSN, to obtain a precise synchronization. Finally, we model our proposed clock synchronization solution to verify its correctness. Through this contribution we obtain a partial solution to our goal defined in RG_3. However, through obtaining this contribution we also realize that clock integration requires specific solutions for each protocol to be integrated. This hinders adoption due to the time needed to design and implement each solution, and makes compatibility between solutions difficult. For example, if there are two legacy networks with two different synchronization protocols (P1 and P2) and we want to integrate them with TSN, we would need a specific solution for the integration of P1 with TSN and another one for P2 with TSN that might not even be compatible with each other. For this reason, we determine that it would be more efficient to palliate the problems generated by the lack of synchronization in a general way from the TSN network without involving the legacy networks.

- **RC_5**: We develop a mechanism called TALESS (TSN with Legacy End-Stations Synchronization) to address the adverse effects caused by the lack of synchronization identified through experiments on a network prototype. This solution is general and transparent for any legacy Ethernet-based network communicating through TSN. TALESS is modeled and validated through simulations with realistic network values. On the other hand, the mechanism is implemented in a network prototype to experimentally demonstrate its effectiveness. Finally, we compare the results of the experiment with the simulation model to verify both implementations. Through this contribution we obtain a complete solution to our goal defined in RG_3.

- **RC_6**: We use the **UPPAAL** model checker [14] to build a model of the SRP protocol. The model allow us to verify that SRP does not provide termination nor consistency, and to identify the scenarios in which these problems occur. We discuss the consequences derived from the absence of termination and consistency, and propose several modifications to the protocol to address these issues. Finally, we select the best one and develop a new protocol called CSRP (Consistent Stream Reservation Protocol). To ensure the correctness of our design, we create a **UPPAAL** model for CSRP and validate it through verification testing. Through this contribution we obtain our goal defined in RG_4.

6.4 Included Papers

The research contributions are proposed in the form of published papers in conferences and journals. The order of the papers is in accordance with the contributions. Five papers are included in the licentiate thesis: Paper A, B, C and E have already been published while paper D is pending for submission.

6.4.1 Paper A

Title:

LETRA: Mapping Legacy Ethernet-Based Traffic into TSN Traffic Classes

Authors:

Daniel Bujosa, Mohammad Ashjaei, Alessandro V Papadopoulos, Julian Proenza, Thomas Nolte.

Status:

Published in the 26^{th} IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2021.

Abstract:

This paper proposes a method to efficiently map the legacy Ethernet-based traffic into Time Sensitive Networking (TSN) traffic classes considering different traffic characteristics. Traffic mapping is one of the essential steps for industries to gradually move towards TSN, which in turn significantly mitigates the management complexity of industrial communication systems. In this paper, we first identify the legacy Ethernet traffic characteristics and properties. Based on the legacy traffic characteristics we presented a mapping methodology to map them into different TSN traffic classes. We implemented the mapping method as a tool, named Legacy Ethernet-based Traffic Mapping Tool or LETRA, together with a TSN traffic scheduling and performed a set of evaluations on different synthetic networks. The results show that the proposed mapping method obtains up to 90% improvement in the schedulability ratio of the traffic compared to an intuitive mapping method on a multi-switch network architecture.

Authors' Contributions:

I was the main driver of the work under the supervision of the co-authors. The plan for the paper was formed in joint discussions with the co-authors. I performed the tool implementation and evaluations and wrote the draft of the paper. The co-authors have reviewed the paper, after which I have improved it.

6.4.2 Paper B

Title:

HERMES: Heuristic Multi-queue Scheduler for TSN Time-Triggered Traffic with Zero Reception Jitter Capabilities

Authors:

Daniel Bujosa, Mohammad Ashjaei, Alessandro V Papadopoulos, Julian Proenza, Thomas Nolte.

Status:

Published in the 30^{th} International Conference on Real-Time Networks and Systems (RTNS), 2022.

Abstract:

The Time-Sensitive Networking (TSN) standards provide a toolbox of features to be utilized in various application domains.The core TSN features include deterministic zero-jitter and low-latency data transmission and transmitting traffic with various levels of time-criticality on the same network. To achieve a deterministic transmission, the TSN standards define a time-aware shaper that coordinates transmission of Time-Triggered (TT) traffic. In this paper, we tackle the challenge of scheduling the TT traffic and we propose a heuristic algorithm, called HERMES. Unlike the existing scheduling solutions, HERMES results in a significantly faster algorithm run-time and a high number of schedulable networks. HERMES can be configured in two modes of zero or relaxed reception jitter while using multiple TT queues to improve the schedulability. We compare HERMES with a constraint programming (CP)-based solution and we show that HERMES performs better than the CP-based solution if multiple TT queues are used, both with respect to algorithm run-time and schedulability of the networks.

Authors' Contributions:

I was the main driver of the work under the supervision of the co-authors. The plan for the paper was formed in joint discussions with the co-authors. I performed the tool implementation and evaluations and wrote the draft of the paper. The co-authors have reviewed the paper, after which I have improved it.

6.4.3 Paper C

Title:

Clock Synchronization in Integrated TSN-EtherCAT Networks

Authors:

Daniel Bujosa, Daniel Hallmans, Mohammad Ashjaei, Alessandro V Papadopoulos, Julian Proenza, Thomas Nolte.

Status:

Published in the 25^{th} IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), 2020.

Abstract:

Moving towards new technologies, such as Time Sensitive Networking (TSN), in industries should be gradual with a proper integration process instead of replacing the existing ones to make it beneficial in terms of cost and performance. Within this context, this paper identifies the challenges of integrating a legacy EtherCAT network, as a commonly used technology in the automation domain, into a TSN network. We show that clock synchronization plays an essential role when it comes to EtherCAT-TSN network integration with important requirements. We propose a clock synchronization mechanism based on the TSN standards to obtain a precise synchronization among EtherCAT nodes, resulting to an efficient data transmission. Based on a formal verification framework using UPPAAL tool we show that the integrated EtherCAT-TSN network with the proposed clock synchronization mechanism achieves at least 3 times higher synchronization precision compared to not using any synchronization.

Authors' Contributions:

I and Daniel Hallmans were the main drivers of the work under the supervision of the co-authors. The plan for the paper was formed in joint discussions with the co-authors. I performed the tool implementation and evaluations, and I wrote the draft of the paper in collaboration with Daniel Hallmans. The co-authors reviewed the paper, after which I improved it.

6.4.4 Paper D

Title:

TALESS: TSN with Legacy End-Stations Synchronization

Authors:

Daniel Bujosa, Mohammad Ashjaei, Alessandro V Papadopoulos, Julian Proenza, Thomas Nolte.

Status:

Pending for submission to the IEEE Transactions on Industrial Informatics.

Abstract:

Time Sensitive Networks (TSN) have become one of the most important communications standards in many industrial sectors. However, TSN requires specific hardware and software. This makes it difficult for established companies to adopt TSN, as it would imply changing most legacy hardware and software which may not be cost-effective in most cases. In order to enable the adoption of TSN by the industry and be more environmentally sustainable, it is necessary to develop tools to integrate legacy systems with TSN. In this paper, we propose a solution for the coexistence of different time domains from different legacy subsystems with their corresponding synchronization protocols in a single TSN network. To this end, we experimentally identified the effects of replacing the communications subsystem of a legacy Ethernet-based network with TSN in terms of synchronization. Based on the results, we propose a solution called TSN with Legacy End-Stations Synchronization (TALESS). TALESS is able to identify the drift between the TSN communications subsystem and the legacy devices and modify the TSN schedule to adapt to the time domains of the integrated legacy systems in order to avoid the effects of the lack of synchronization between them. We validate TALESS through both simulations and experiments on a prototype. Thereby we demonstrate that, thanks to TALESS, legacy systems are able to synchronize through TSN and even improve features such as their reception jitter or their integrability with other legacy systems.

Authors' Contributions:

I was the main driver of the work under the supervision of the co-authors. The plan for the paper was formed in joint discussions with the co-authors. I performed the tool implementation and evaluations and wrote the draft of the paper. The co-authors have reviewed the paper, after which I have improved it.

6.4.5 Paper E

Title:
CSRP: An Enhanced Protocol for Consistent Reservation of Resources in AVB/TSN
Authors:
Daniel Bujosa, Ines Álvarez, Julian Proenza.
Status:
Published in IEEE Transactions on Industrial Informatics, 2020.
Abstract:
The IEEE Audio Video Bridging (AVB) Task Group (TG) was created to provide Ethernet with soft real-time guarantees. Later on, the TG was renamed to Time-Sensitive Networking (TSN) and its scope broadened to support hard real-time and critical applications. The Stream Reservation Protocol (SRP) is a key work of the TGs as it allows reserving resources in the network, guaranteeing the required quality of service (QoS). AVB's SRP is based on a distributed architecture, while TSN's is based on centralized ones. The distributed version of SRP is supported and used in TSN. Nevertheless, it was not designed to provide properties that are important for critical applications. In this work we model SRP using UPPAAL and we study the termination and consistency. We verify that SRP does not provide such properties. Furthermore, we propose an improved protocol called Consistent Stream Reservation Protocol (CSRP) and we formally verify its correctness using UPPAAL.
Authors' Contributions:
I was the main driver of the work under the supervision of the co-authors. The plan for the paper was formed in joint discussions with the co-authors. I performed the tool implementation and evaluations and wrote the draft of the paper. The co-authors have reviewed the paper, after which I have improved it.

6.4.6 Mapping the Included Papers with the Research Contributions

The mapping of the aforementioned thesis contribution into published and planned publications, that are included in the thesis, is shown in Table 6.1.

Given the overall set of contributions towards the research goals of this thesis we believe that in summary the thesis make a significant step towards achieving the overall goal of the thesis.

Table 6.1: The mapping of the research goals to the papers included in the thesis.

	Paper A	**Paper B**	**Paper C**	**Paper D**	**Paper E**
$\mathbf{RC}_1$	✓				
$\mathbf{RC}_2$	✓				
$\mathbf{RC}_3$		✓			
$\mathbf{RC}_4$			✓		
$\mathbf{RC}_5$				✓	
$\mathbf{RC}_6$					✓

Chapter 7

Summary and Future Work

7.1 Summary

In this thesis we have developed tools and mechanisms for the migration and integration of legacy networks into TSN as well as improved some of its mechanisms. In this way, we seek to facilitate the adoption of TSN avoiding the time, effort, and resources that would be required to replace a functional network with a new TSN network. Furthermore, thanks to the proposed solutions, legacy networks can not only communicate through TSN normally, which improves their integration with other legacy networks and reduces the overall complexity thanks to the use of a single communication protocol, but they are also able to benefit from the TSN enhancements providing a better service.

We have improved TSN's SRP to avoid wasting resources (RC_4) and we have created three tools aimed at migrating and integrating legacy networks with TSN. These tools include: firstly, LETRA, a tool for mapping legacy traffic into the different types of TSN traffic (RC_1); secondly, HERMES, which is a heuristic scheduling algorithm implemented as a fast and scalable tool capable of scheduling the traffic mapped by LETRA (RC_2); and finally, TALESS, a synchronization tool capable of eliminating the adverse effects caused by the lack of synchronization between legacy networks and TSN (RC_3).

7.2 Future Work

As future work, we seek to develop a methodology for traffic identification in legacy networks. This will be able to, by sampling traffic from an existing network or by analyzing the network specification, parameterize the legacy traffic in terms of relevant parameters for the migration of such traffic to TSN.

In addition, we will develop a TSN frame forging mechanisms for Ethernet frames. It will be able to modify the Ethernet frame fields of legacy traffic dynamically to conform to the TSN format, e.g., by assigning TSN priorities. This will enable the transmission of legacy traffic as a specific TSN traffic type transparently to the legacy end-stations.

Finally, these tools will be integrated with those already presented in this work. This will provide a set of tools capable of identifying traffic, mapping it, scheduling it, and synchronizing it all in one. Such set of tools would considerably reduce migration and integration times and allow adoption of TSN with minimal costs.

Bibliography

[1] 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)*, pages C1–72, 2010.

[2] IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP). *IEEE Std 802.1Qat-2010 (Revision of IEEE Std 802.1Q-2005)*, 2010.

[3] IEEE Standard for Local and Metropolitan Area Networks–Audio Video Bridging (AVB) Systems. *IEEE Std 802.1BA-2011*, pages 1–45, 2011.

[4] IEEE Standard for Local and Metropolitan Area Networks–Bridges and Bridged Networks – Amendment 31: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements. *IEEE Std 802.1Qcc-2018 (Amendment to IEEE Std 802.1Q-2018 as amended by IEEE Std 802.1Qcp-2018)*, pages 1–208, Oct 2018.

[5] 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)*, pages 1–421, 2020.

[6] G. Alderisi, G. Patti, and L. Lo Bello. Introducing support for Scheduled traffic over IEEE Audio Video Bridging networks. In *Conf. Emerging Technologies Factory Automation*, 2013.

[7] Inés Álvarez, Ignasi Furió, Julián Proenza, and Manuel Barranco. Design and Experimental Evaluation of the Proactive Transmission of Replicated Frames Mechanism over Time-Sensitive Networking. *Sensors*, 21(3):756, 2021.

[8] F. A. R. Arif and T. S. Atia. Load balancing routing in Time-Sensitive Networks. In *Int. Scientific-Practical Conference Problems of Infocommunications Science and Technology*, 2016.

[9] Emmanuel Arzuaga and David R Kaeli. Quantifying load imbalance on virtualized enterprise servers. In *Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*, pages 235–242, 2010.

[10] Mohammad Ashjaei, Lucia Lo Bello, Masoud Daneshtalab, Gaetano Patti, Sergio Saponara, and Saad Mubeen. Time-Sensitive Networking in Automotive Embedded Systems: State of the Art and Research Opportunities. *Journal of Systems Architecture*, 110:1–47, September 2021.

[11] Mohammad Ashjaei, Gaetano Patti, Moris Behnam, Thomas Nolte, Giuliana Alderisi, and Lucia Lo Bello. Schedulability analysis of Ethernet Audio Video Bridging networks with scheduled traffic support. *Real-Time Systems*, 53(4):526–577, 2017.

[12] Ayman A Atallah, Ghaith Bany Hamad, and Otmane Ait Mohamed. Fault-resilient topology planning and traffic configuration for IEEE 802.1 Qbv TSN networks. In *2018 IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*, pages 151–156. IEEE, 2018.

[13] Haytham Baniabdelghany, Roman Obermaisser, et al. Extended synchronization protocol based on IEEE802. 1AS for improved precision in dynamic and asymmetric TSN hybrid networks. In *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–8. IEEE, 2020.

[14] Gerd Behrmann, Alexandre David, and Kim G. Larsen. *A Tutorial on Uppaal*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[15] D. Bujosa, I. Alvarez, D. Čavka, and J. Proenza. Analysing Termination and Consistency in the AVB's Stream Reservation Protocol. In *Proceedings of the IEEE 24th International Conference on Emerging Technologies and Factory Automation (ETFA 2019)*, October 2019.

[16] Daniel Bujosa, Andreas Johansson, Mohammad Ashjaei, Alessandro V Papadopoulos, Julian Proenza, and Thomas Nolte. The Effects of Clock Synchronization in TSN Networks with Legacy End-Stations. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2022.

[17] Zichao Chai, Wei Liu, Mao Li, and Jing Lei. Cross Domain Clock Synchronization Based on Data Packet Relay in 5G-TSN Integrated Network. In *2021 IEEE 4th International Conference on Electronics and Communication Engineering (ICECE)*, pages 141–145. IEEE, 2021.

[18] Sameer Chouksey, Hariram Selvamurugan Satheesh, and Johan Åkerberg. An experimental study of TSN-nonTSN coexistence. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1577–1584. IEEE, 2021.

[19] Industrial Internet Consortium et al. Time sensitive networks for flexible manufacturing testbed characterization and mapping of converged traffic types, 2019.

[20] Silviu S Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. Scheduling Real-Time Communication in IEEE 802.1 Qbv Time Sensitive Networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, pages 183–192, 2016.

[21] Théo Docquier, Ye-Qiong Song, Vincent Chevrier, Ludovic Pontnau, and Abdelaziz Ahmed-Nacer. Iec 61850 over tsn: Traffic mapping and delay analysis of goose traffic. In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 246–253. IEEE, 2020.

[22] Gordana Dodig-Crnkovic. Scientific Methods in Computer Science. In *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges*, 2002.

[23] Voica Gavriluţ and Paul Pop. Traffic-Type Assignment for TSN-Based Mixed-Criticality Cyber-Physical Systems. *ACM Trans. Cyber-Phys. Syst.*, 4(2), 2020.

[24] Voica Gavriluţ and Paul Pop. Scheduling in Time Sensitive Networks (TSN) for mixed-criticality industrial applications. In *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, pages 1–4, 2018.

[25] Voica Gavriluţ, Luxi Zhao, Michael L Raagaard, and Paul Pop. AVB-aware Routing and Scheduling of Time-Triggered traffic for TSN. *IEEE Access*, 6:75229–75243, 2018.

[26] Michael Gundall, Christopher Huber, Peter Rost, Rüdiger Halfmann, and Hans D Schotten. Integration of 5G with TSN as prerequisite for a highly flexible future industrial automation: Time synchronization based on IEEE 802.1 AS. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pages 3823–3830. IEEE, 2020.

[27] Marina Gutiérrez, Astrit Ademaj, Wilfried Steiner, Radu Dobrin, and Sasikumar Punnekkat. Self-configuration of IEEE 802.1 TSN networks. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8. IEEE, 2017.

[28] Marina Gutiérrez, Wilfried Steiner, Radu Dobrin, and Sasikumar Punnekkat. A Configuration Agent based on the Time-Triggered paradigm for Real-Time Networks. In *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, pages 1–4. IEEE, 2015.

[29] Jetmir Haxhibeqiri, Xianjun Jiao, Muhammad Aslam, Ingrid Moerman, and Jeroen Hoebeke. Enabling TSN over IEEE 802.11: Low-overhead time synchronization for wi-fi clients. In *2021 22nd IEEE international conference on industrial technology (ICIT)*, volume 1, pages 1068–1073. IEEE, 2021.

[30] IEEE. Ieee standard for local and metropolitan area networks–bridges and bridged networks, 2018. Includes TSN Time-Aware Shaper (TAS) definition, among other TSN-related features and protocols.

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

[32] O. Kleineberg, P. Fröhlich, and D. Heffernan. Fault-Tolerant Ethernet Networks with Audio and Video Bridging. In *ETFA2011*, pages 1–8, Sept 2011.

[33] Leonard Kleinrock. Queueing systems, volume i: Theory, vol. I, 1975.

[34] H. Lim, L. Völker, and D. Herrscher. Challenges in a Future IP/Ethernet-based in-car Network for Real-Time Applications. In *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 7–12, June 2011.

[35] Lucia Lo Bello, Mohammad Ashjaei, Gaetano Patti, and Moris Behnam. Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support. *Journal of Parallel and Distributed Computing,*, 144,, 2020.

[36] P. Meyer, T. Steinbach, F. Korf, and T. C. Schmidt. Extending IEEE 802.1 AVB with Time-Triggered Scheduling: A Simulation Study of the Coexistence of Synchronous and Asynchronous Traffic. In *2013 IEEE Vehicular Networking Conference*, pages 47–54, Dec 2013.

[37] Jörn Migge, Josetxo Villanueva, Nicolas Navet, and Marc Boyer. Insights on the Performance and Configuration of AVB and TSN in Automotive Ethernet Networks. In *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*, Toulouse, France, January 2018.

[38] Seifeddine Nsaibi, Ludwig Leurs, and Hans D Schotten. Formal and simulation-based timing analysis of Industrial-Ethernet sercos III over TSN. In *2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pages 1–8, 2017.

[39] Maryam Pahlevan and Roman Obermaisser. Genetic Algorithm for scheduling Time-Triggered traffic in Time-Sensitive Networks. In *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, volume 1, pages 337–344. IEEE, 2018.

[40] Maryam Pahlevan, Nadra Tabassam, and Roman Obermaisser. Heuristic list scheduler for Time Triggered traffic in Time Sensitive Networks. *ACM Sigbed Review*, 16(1):15–20, 2019.

[41] Rina Panigrahy, Kunal Talwar, Lincoln Uyeda, and Udi Wieder. Heuristics for vector bin packing. *research.microsoft.com*, 2011.

[42] D. Park, J. Lee, C. Park, and S. Park. New Automatic De-Registration Method Utilizing a Timer in the IEEE802.1 TSN. In *2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI)*, pages 47–51, Oct 2016.

[43] R. Queck. Analysis of Ethernet AVB for Automotive Networks using Network Calculus. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, pages 61–67, July 2012.

[44] Michael Lander Raagaard and Paul Pop. Optimization algorithms for the scheduling of IEEE 802.1 Time-Sensitive Networking (TSN). *Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep*, 2017.

[45] Mauricio GC Resende and Celso C Ribeiro. GRASP: Greedy Randomized Adaptive Search Procedures. In *Search methodologies*, pages 287–312. Springer, 2014.

[46] Alexey M Romanov, Francesco Gringoli, and Axel Sikora. A precise synchronization method for future wireless TSN networks. *IEEE Transactions on Industrial Informatics*, 17(5):3682–3692, 2020.

[47] Siwar Ben Hadj Said, Quang Huy Truong, and Michael Boc. SDN-based configuration solution for IEEE 802.1 Time Sensitive Networking (TSN). *ACM SIGBED Review*, 16(1):27–32, 2019.

[48] R. Salazar, T. Godfrey, L. Winkel, N. Finn, C. Powell, B. Rolfe, and M. Seewald. Utility Applications of Time Sensitive Networking White Paper (D3). Technical report, IEEE, 2018.

[49] S. Samii and H. Zinner. Level 5 by Layer 2: Time-Sensitive Networking for Autonomous Vehicles. *IEEE Communications Standards Magazine*, 2(2):62–68, 2018.

[50] Eike Schweissguth, Dirk Timmermann, Helge Parzyjegla, Peter Danielis, and Gero Mühl. ILP-based routing and scheduling of multicast realtime traffic in Time-Sensitive Networks. In *2020 IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 1–11. IEEE, 2020.

[51] Oscar Seijo, Zaloa Fernández, Iñaki Val, and Jesús A López-Fernández. SHARP: towards the integration of Time-Sensitive communications in legacy LAN/WLAN. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, 2018.

[52] Oscar Seijo, Xabier Iturbe, and Inaki Val. Tackling the Challenges of the Integration of Wired and Wireless TSN with a Technology Proof-of-Concept. *IEEE Transactions on Industrial Informatics*, 2021.

[53] Haochuan Shi, Adnan Aijaz, and Nan Jiang. Evaluating the Performance of Over-the-Air Time Synchronization for 5G and TSN Integration. In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–6. IEEE, 2021.

[54] Jiajia Song, Makoto Kubomi, Jeffrey Zhao, and Daisuke Takita. Time synchronization performance analysis considering the frequency offset inside 5G-TSN network. In *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6. IEEE, 2021.

[55] Tobias Striffler and Hans D Schotten. The 5G Transparent Clock: Synchronization Errors in Integrated 5G-TSN Industrial Networks. In *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*, pages 1–6. IEEE, 2021.

[56] Thomas Stüber, Lukas Osswald, Steffen Lindner, and Michael Menth. A Survey of Scheduling in Time-Sensitive Networking (TSN). *arXiv preprint arXiv:2211.10954*, 2022.

[57] Ammad Ali Syed, Serkan Ayaz, Tim Leinmüller, and Madhu Chandra. Dynamic Scheduling and Routing for TSN based in-vehicle networks. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE, 2021.

[58] K. S. Umadevi and R. K. Sridharan. Multilevel ingress scheduling policy for Time Sensitive Networks. In *Int. Conf. Microelectronic Devices, Circuits and Systems*, 2017.

[59] M. Wollschlaeger, T. Sauter, and J. Jasperneite. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1):17–27, 2017.

[60] Jin Y Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

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