

Title	Time-Sensitive Networking for industrial IoT: integration, analysis, and performance evaluation
Authors	Seliem, Mohamed
Publication date	2024
Original Citation	Seliem, M. 2024. Time-Sensitive Networking for industrial IoT: integration, analysis, and performance evaluation. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2024, Mohamed Seliem. - https://creativecommons.org/licenses/by/4.0/
Download date	2026-10-10 14:55:26
Item downloaded from	https://hdl.handle.net/10468/16388

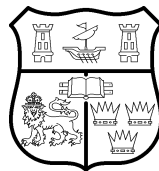
Time-Sensitive Networking for Industrial IoT: Integration, Analysis, and Performance Evaluation

Mohamed Seliem

BSc, MSc

orcid.org/0000-0002-2132-9249

**Thesis submitted for the degree of
Doctor of Philosophy**



NATIONAL UNIVERSITY OF IRELAND, CORK

COLLEGE OF SCIENCE, ENGINEERING AND FOOD SCIENCE
SCHOOL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

August, 2024

Head of Department: Prof. Utz Roedig

Supervisors: Prof. Dirk Pesch, Dr. Ahmed Zahran

Research supported by Science Foundation Ireland

Contents

List of Figures	iv
List of Tables	vii
Abstract	ix
Acknowledgements	xi
1 Introduction	1
1.1 Motivation	1
1.2 Thesis statement and contributions	3
1.2.1 Thesis statement	3
1.2.2 Key contributions	3
1.3 Thesis organisation	4
1.4 List of publications	5
1.5 Use of AI Tools	6
2 Background	7
2.1 Time Sensitive Networking	7
2.1.1 Key Principles of TSN	8
2.1.2 Traffic Shaping and Scheduling	9
2.1.3 Redundancy and Fault Tolerance	10
2.1.4 Industrial Profile of TSN	10
2.2 Network Calculus Principles	11
2.2.1 Overview of Network Calculus	12
2.2.2 Application of Network Calculus in TSN	15
2.2.3 Deterministic Bounds for Real-Time Traffic	16
2.2.4 Traffic Shaping and Resource Allocation	18
2.2.5 Evaluation of Network Performance	20
2.3 Relevant Related Work	22
2.3.1 Evaluating Real-time Ethernet Solutions:	23
2.3.2 Performance Assessment of TSN Amendment	25
2.3.3 Modelling Time Sensitive Networking through Simulation	27
2.3.4 Worst Case Analysis of Industrial Time Sensitive Networks	30
3 Traffic Requirements for Industrial Applications	32
3.1 TSN industrial classes of traffic	33
3.1.1 Scheduled Traffic (ST)	34
3.1.2 Audio-Video Bridging Traffic	36
3.1.3 Best-Effort Traffic	37
3.2 Essential aspects of traffic requirements for industrial applications	38
3.2.1 Deterministic Communication for Industrial Applications:	39
3.2.2 Bandwidth Requirements for Industrial Applications	40
3.2.3 Latency and Jitter Constraints in Industrial Applications	41
3.2.4 Quality of Service (QoS) Considerations	42
3.3 Industrial use cases in smart manufacturing	44
3.3.1 Control Loops	45
3.3.2 Video surveillance	46

3.3.3	Sequence of Events	48
3.3.4	Network monitoring and Diagnostics	50
3.4	Quality Control and Classification After Production	52
3.5	Collaborative Robot based Industrial Automation	55
4	Timing Analysis of TSN Based Industrial Networks using Network Calculus	58
4.1	System Overview	59
4.2	Worst-case delay analysis for multiple classes of traffic.	62
4.2.1	Arrival and Service curves of scheduled traffic	62
4.2.2	Arrival and Service curves of AVB traffic	63
4.2.3	Arrival and Service curves of BE traffic	65
4.3	Worst Case Delay bounds per single hop	65
5	Improved Timing Analysis of TSN based Redundant Industrial Networks	68
5.1	System Architecture Model	69
5.2	Arrival curves for different industrial traffic	72
5.3	Service curves for different industrial traffic	76
5.4	Industrial data traffic analysis	81
5.4.1	Tight-bounds for Industrial data traffic	81
5.4.2	Worst case end-to-end delay and backlog analysis	82
6	Evaluation and Validation of TSN Industrial Networks	84
6.1	Performance Evaluation of TSN based industrial network using simulation models	86
6.1.1	TSN network structure and design	86
6.1.2	Traffic models characterisation	87
6.1.3	Simulation model implementation	91
6.1.4	Results and discussion	93
6.2	Validation and Performance Evaluation of the TSN based industrial networks timing analysis Framework	100
6.2.1	Network Model	100
6.2.2	Model Implementation	102
6.2.3	Evaluation and Results	103
6.3	Experimental evaluation of the improved timing analysis of TSN based industrial networks	108
6.3.1	Experimental Evaluation Setup:	108
6.3.2	Model Validation	109
6.3.3	Impact of Routing	110
6.3.4	Impact of increasing traffic loads with different priorities	112
6.3.4.1	Impact of credit accumulation rate	114
6.3.4.2	Redundancy and load distribution	115
6.3.5	Industrial use-case:	117
7	Integrating TSN with SDN and Wi-Fi: Enhanced Time-Critical Communication Solutions	124

7.1	Software-Defined Time Sensitive Networks for Industrial Automation	125
7.1.1	Software defined TSN related work	127
7.1.2	Factory Automation Model And System Architecture . .	128
7.1.3	Scheduling and Routing in Software Defined TSN	132
7.1.3.1	Scheduling approach in SD-TSN:	132
7.1.3.2	Routes calculation in SD-TSN	134
7.1.4	Validation and Discussion	135
7.1.4.1	SD-TSN test environment setup	135
7.1.4.2	SD-TSN emulation condition	136
7.1.4.3	SD-TSN performance evaluation	137
7.2	Wireless TSN Extension to Enable Deterministic Connectivity .	140
7.2.1	Time sensitive use cases with wireless connectivity requirements	141
7.2.2	WiFi integration with TSN:	143
7.2.3	Simulation setup and discussion	146
7.3	Conclusion	150
8	Conclusion and Future Directions	153
8.1	Conclusion	153
8.2	Future directions	157

List of Figures

2.1	Key Principles of Time-Sensitive Networking.	9
2.2	Standard Arrival Curve	13
2.3	Network Calculus rules. Adapted from [MHG20].	14
3.1	TSN switch interface with traffic shaping, frame pre-emption, and strict priority.	33
3.2	Smart manufacturing vertical model	38
3.3	Basic Control Loop	45
3.4	Plant floor video surveillance use case	47
3.5	Network monitoring and Diagnostics use case	51
3.6	Quality Control and Classification After Production (QCAP)	53
3.7	Collaborative Robot based Industrial Automation	57
4.1	Single hop model of QCAP TSN Node.	60
4.2	TSN device interface processing different data flows arriving si- multaneously	61
5.1	Production System Architecture	71
5.2	Arrival model for the isochronous frames with pre-emption over- heads	73
5.3	eMAC transmission with: (a) preemptable traffic ($\leq minPKT$) (b) preemptable traffic with Hold and Release mode ($\leq$ $minPKT$).	78
5.4	pMAC transmission with: (a) pre-emptive traffic ($> minPKT$), (b) pre-emptive traffic with Hold and Release mode ($>$ $minPKT$), (c) non-pre-emptive traffic with guard band, (d) non- pre-emptive traffic with different priorities.	78
5.5	Credit allocation: (a) credit accumulation continues in the guard-band slot (b) credit accumulation stops in the guard-band slot.	79
6.1	Network structure for QCAP in OMNET++.	87
6.2	Queue Length distribution for Worst Case with infinite queue size configuration (Deterministic traffic)	94
6.3	Queuing Time distribution for Worst Case with infinite queue size configuration (Deterministic traffic)	95
6.4	End-to-end delay distribution for Worst Case with infinite queue size configuration (Deterministic traffic)	95
6.5	Delivery ratio distribution for Worst Case with infinite queue size configuration (Deterministic traffic)	96
6.6	Queue Length distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)	97
6.7	Queuing Time distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)	97

6.8	End-to-end delay distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)	98
6.9	Delivery ratio distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)	98
6.10	Queue Length distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)	99
6.11	Queuing Time distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)	100
6.12	End-to-end delay distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)	100
6.13	Delivery ratio distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)	101
6.14	Representative QCAP network topology	101
6.15	Comparison between worst-case delay bounds for QCAP data flows	104
6.16	Comparison between worst-case delay bounds for different flow rate	105
6.17	The effect of transmission offset and increasing the bandwidth of critical interfaces on the WCD of scheduled traffic.	106
6.18	Comparison between different idle slopes on the WCD of different traffic types.	107
6.19	Synthetic test-cases.	110
6.20	Comparison between WCD bounds for flow types using different NC based frameworks.	111
6.21	Traffic grouping and interface selection impact on the worst case delay for different traffic classes.	112
6.22	Delay bound for different number of flows for different traffic classes.	113
6.23	Different credit accumulation rates impact on the overall traffic classes delay.	114
6.24	Worst case delay of traffic classes for varying offered load.	115
6.25	Improvement of the Worst case delay bounds for traffic classes.	116
6.26	Industrial use case (Connection of Rings).	118
6.27	Detection	119
6.28	Actuation	121
6.29	Inspection and H/M Interface	121
6.30	Ethernet	122
7.1	Multi-Service Control Layer of a Factory Automation Model	128
7.2	Proposed SD-TSN Architecture	129
7.3	TSN switch internal components	131
7.4	SD-TSN testing use case.	136
7.5	End to end latency versus number of real time transmitted packets.	138
7.6	End to end latency for different types of traffics with different priorities.	138

7.7	Average scheduling run time per TSN flow versus number of scheduled flows across the network.	139
7.8	TSN-based applications with wireless connectivity requirements	142
7.9	WiFi-TSN network to support collaborative robots based industrial automation	147
7.10	Industrial automation using collaborative robots: Average latency of CDT traffic flows.	150
7.11	Industrial automation using collaborative robots: Average latency of VST traffic flows.	150
7.12	Industrial automation using collaborative robots: Average latency of SDT traffic flows.	151
7.13	Industrial automation using collaborative robots: Average latency of NCT traffic flows.	151

List of Tables

3.1	isochronous and non-isochronous cyclic real-time traffic type properties	35
3.2	Collaborative Robots based Industrial Automation Application Requirements	57
6.1	QCAP Data Flow Characterisations.	88
6.2	Traffic model characterisations (CDT, Class A, Class B).	88
6.3	Traffic model characterisation for web traffic.	88
6.4	Network and simulation parameters	93
6.5	The average end-to-end delay and delivery ratio of QCAP simulation results (Standard Ethernet [FIFO]).	96
6.6	The average end-to-end delay and delivery ratio of QCAP simulation results (Standard Ethernet [PRIO]).	97
6.7	The average end-to-end delay and delivery ratio of QCAP simulation results (TSN [TAS]).	98
6.8	The average end-to-end delay and delivery ratio of QCAP simulation results (TSN [TAS+CBS]).	99
6.9	Data flow characterisations	102
6.10	Data flows description.	109
6.11	Data flows characterisations	120
7.1	WiFi-TSN configuration and simulation parameters	148

I, Mohamed Seliem, certify that this thesis is my own work and has not been submitted for another degree at University College Cork or elsewhere.

Mohamed Seliem

Abstract

Industrial automation networks demand precise timing, minimal latency, and negligible packet loss for efficient real-time data exchange. Time-Sensitive Networking (TSN) emerges as a crucial technology for future automation, promising enhanced timing accuracy, reduced packet delay, and improved networking determinism. This thesis explores and innovates within TSN functionalities to address key aspects of industrial networking and related technologies.

The critical need for reliable real-time data exchange across industries is examined, introducing TSN principles such as time synchronisation, deterministic communication, traffic shaping, and Quality of Service (QoS) assurances. Through simulation, typical industrial use cases and traffic requirements are evaluated, focusing on priority queuing, Time Aware shaping (TAS), and Credit Based Shaping (CBS) to meet latency constraints. The findings demonstrate TSN's ability to orchestrate network traffic while adhering to strict timing requirements, highlighting its practical relevance in industrial automation.

In smart manufacturing environments, the optimisation of industrial networks for Quality Control and Classification After Production (QCAP) is emphasised. By leveraging TSN standards, diverse QoS requirements are addressed to enhance efficiency and reliability. Fault tolerance in Industrial Internet of Things (IIoT) applications is also investigated using network calculus principles to analyse worst-case latency, providing insights into network performance and stability.

The research integrates TSN with Software Defined Networks (SDN) to manage network configurations, focusing on traffic scheduling in industrial applications. Network-device contracts are proposed for traffic schedule computation and distribution, demonstrating scalability through Mininet emulation. Additionally, Wi-Fi is explored as a complementary technology for IoT applications, evaluating its potential to reduce latency and enhance industrial automation.

This thesis offers a comprehensive analysis of TSN performance across various scenarios and its integration with complementary technologies, providing valuable insights for advancing industrial automation and connectivity within the industry 4.0 paradigm.

This one goes to you, Mom.

Acknowledgements

I would like to take this opportunity to express my heartfelt gratitude to the individuals and organisations that have played a significant role in the completion of my doctoral thesis.

First and foremost, I am deeply thankful to my supervisor, Dirk Pesch, for their unwavering guidance, mentorship, and dedication throughout this research journey. Their profound knowledge, invaluable insights, and continuous encouragement have been instrumental in shaping the outcome of this thesis.

I am also indebted to my thesis co-supervisor, Ahmed Zahran, for their expert advice, critical feedback, and valuable suggestions that have enriched the quality of this research.

My heartfelt appreciation goes out to the staff and faculty of the Computer Science and IT department at University College Cork, who provided a stimulating academic environment, access to resources, and administrative support. I am grateful to Science Foundation Ireland for their financial support under Grant number 16/RC/3918, which enabled me to carry out this research.

I would like to extend my thanks to my fellow graduate students and colleagues who have been a source of inspiration, camaraderie, and intellectual exchange throughout my doctoral journey. Your support and friendship have made this experience all the more rewarding.

I owe a debt of gratitude to my family for their unwavering support, love, and encouragement. My parents have been a constant source of motivation, and their belief in my abilities has been a driving force in my pursuit of higher education.

Last but not least, I want to express my profound appreciation to my friends and loved ones who have provided emotional support and understanding during the ups and downs of this academic endeavour.

This thesis would not have been possible without the collective support and encouragement of these individuals and institutions. I am truly grateful for their contributions to my academic and personal growth.

Mohamed Seliem
August 2024

Acronyms

CBS Credit Based Shaping. ix, 86

FIFO First In First Out. 86

GCL Gates Control List. 132

IIoT Industrial Internet of Things. ix

ITS Intelligent Transportation Systems. 142

NC Network calculus. 59

OVSDB Open vSwitch Database Management Protocol. 130

PLC Programmable Logic Controller. 70

PRIQ Priority Queuing. 86

PTP Precision Time Protocol. 144

QCAP Quality Control and Classification After Production. ix, 59

QoS Quality of Service. ix

SD-TSN Software Defined-Time Sensitive Networking. 132

SDN Software Defined Networks. ix

TAS Time Aware shaping. ix, 86

TSF Time Synchronisation Function. 144

TSN Time-Sensitive Networking. ix, 7, 59

UDP User Datagram Protocol. 91

V2I Vehicle-to-Infrastructure. 142

V2V Vehicle-to-Vehicle. 142

WiFi Wireless Fidelity. iii, vi, vii, 141, 143–148, 152

Chapter 1

Introduction

1.1 Motivation

Industrial communications rely on robust wired networks for efficient data exchange among devices and components. Protocols like Ethernet, Profinet, and Modbus are essential due to their high communication rates, lower installation costs, and ease of integration with existing systems [BSB⁺19]. However, traditional protocols often fall short in meeting the precise timing and real-time requirements of modern industrial applications. For instance, Profibus can experience jitter affecting data delivery times, and Modbus, particularly in its serial form, lacks guaranteed fixed transmission times. These issues can lead to unacceptable delays and inconsistencies in hard real-time systems and can also become bottlenecks in high-throughput applications.

To overcome these limitations, the IEEE 802.1 working group introduced the Time-Sensitive Networking (TSN) framework, designed to provide performance guarantees for critical traffic across networks that support both Information Technology (IT) and Operational Technology (OT).

This thesis investigates the integration of TSN functionalities into industrial applications, evaluating their benefits and constraints. Various methodologies are used for assessing TSN performance, including network simulation tools [Var01], traffic generation and analysis tools [LW04], network emulation platforms [Tea12], TSN testbeds, and model-based design tools [Mat12]. The research aims for a comprehensive performance evaluation of TSN functionalities in industrial environments using these tools.

The thesis is motivated by the following key objectives:

1. A Evaluating TSN Queuing Mechanisms: Optimising network infrastructure in industrial settings requires precise timing, bounded latency, minimal jitter, and zero packet loss. This research assesses TSN queuing mechanisms to identify the most effective methods for reliable communication. By evaluating various queuing strategies, the study aims to enhance overall network performance through prioritisation of time-sensitive traffic and efficient management of non-critical data.

2. Analysing TSN-Based Converged Networks: Investigating TSN in converged networks—which integrate diverse data traffic types—aims to evaluate the effectiveness of deterministic communication mechanisms within unified infrastructures. This analysis contributes to standardisation and best practices for deploying deterministic networking technologies.

3. Timing Analysis of Traffic Shaping Methods in the Industrial TSN Profile: The Industrial TSN Profile includes several shaping methods, such as Time-Aware Shaping (TAS), Credit-Based Shaping (CBS), and Frame Pre-emption. Analysing their timing aspects helps understand their impact on industrial networking applications, focusing on worst-case latency, jitter, and packet loss under various conditions.

4. Assessing the TSN based Networks Redundancy to Eliminate Losses:

Redundancy strategies are crucial for minimising packet losses and ensuring continuous operation in industrial and real-time environments. This research explores different redundancy mechanisms like link and path redundancy to enhance network reliability and fault tolerance.

5. Enhancing TSN Network Management: Integrating Software-Defined Networking (SDN) with TSN aims to improve network management through centralised control and programmability. This includes optimising Quality of Service (QoS) and adapting to dynamic network conditions to enhance overall efficiency.

6. Extending TSN to Wireless Applications: With growing demand for wireless connectivity, extending TSN capabilities to support wireless communication is crucial. This research explores mechanisms for integrating TSN with wireless technologies while maintaining determinism and reliability for time-sensitive applications.

1.2 Thesis statement and contributions

1.2.1 Thesis statement

This thesis provides a comprehensive exploration of TSN in industrial networks, offering novel insights and solutions. Contributions include developing and evaluating simulation models for TSN queuing mechanisms, introducing a Network Calculus-based framework for latency analysis, analysing the TSN industrial profile, proposing a novel TSN network management architecture leveraging SDN, and extending TSN functionalities to wireless networks. These findings enhance understanding and improve communication, reliability, and performance in industrial automation scenarios.

1.2.2 Key contributions

To enhance the understanding, analysis, and application of TSN in industrial networks, providing valuable insights and solutions for the industry, the following key contributions have been made:

1. Simulation Models for TSN Queuing Mechanisms: The INET framework in OMNeT++ has been extended to incorporate various queuing models tailored for industrial applications. The study includes a performance analysis of Time-Aware Shaping (TAS) and Credit-Based Shaping (CBS) compared with priority queuing and standard Ethernet FIFO. Results demonstrate the impact of optimal network configuration and queuing methods on performance.

2. Network Calculus-Based Framework for Analysing TSN-Based Networks: A timing analysis framework based on network calculus evaluates worst-case end-to-end latency for various traffic classes, integrating frame prioritisation and preemption. This framework shows tighter delay bounds compared to existing methods.

3. Comprehensive Analysis of the TSN Industrial Profile: This analysis focuses on eight data traffic classes specified in the TSN industrial profile, addressing constraints from isochronous and non-isochronous traffic. It provides precise latency limits for AVB data flows and evaluates the impact of network configuration and load distribution on end-to-end delay.

4. Novel TSN Network Management Architecture: The thesis proposes an SDN-based architecture for managing TSN network configurations, including

queuing and scheduling. Implementing this architecture using Mininet assesses how well end-devices adhere to transmission schedules and the scalability of SDN for real-time requirements.

5. Extending TSN to Wireless Networks: The research evaluates integrating TSN with Wi-Fi, addressing implementation choices and configuration for TSN tools compatible with IEEE 802.11. This aims to enhance performance for wireless industrial applications and create new opportunities for use cases requiring both TSN's reliability and wireless communication's mobility.

In conclusion, the research has practical implications for enhancing communication, reliability, and performance within industrial automation. A primary focus is on applying these findings to real-world use cases and industrial scenarios.

1.3 Thesis organisation

The remainder of the thesis is organised as follows:

Chapter 2 discusses TSN's core principles, including time synchronisation, deterministic communication, and traffic shaping. It introduces the Industrial TSN Profile and network calculus concepts, setting the stage for the thesis research.

Chapter 3 explores TSN's role in addressing diverse traffic needs in industrial environments, including its impact on smart manufacturing and various industrial use cases.

Chapter 4 applies Network Calculus to assess worst-case delays in TSN-based networks, focusing on single-hop models and the challenges of managing diverse data traffic.

Chapter 5 analyses TSN-based industrial communication systems, discussing traffic classes, network architecture, and performance metrics like upper bounds on data traffic and worst-case delays.

Chapter 6 evaluates TSN performance through simulation and model validation, comparing different routing strategies and assessing optimal strategies for minimising delay.

Chapter 7 examines the integration of SDN with TSN and Wi-Fi, detailing the SD-TSN architecture and evaluating its impact on time-critical communication and network management.

Finally, **Chapter 8** concludes the work by summarising key findings and discussing future research directions. It highlights the main insights gained throughout the study and outlines potential areas for further exploration and development.

1.4 List of publications

The research detailed in this dissertation has yielded numerous peer-reviewed publications, listed below. Notably, the study, highlighted in the **TSN-based industrial network performance analysis** [SZP22b] publication, was honoured with the Best Paper Award at the first Internet of Time-Critical Things (IoTime) workshop. Additionally, the investigation outlined in the **Delay analysis of TSN-based industrial networks with preemptive traffic using network calculus** [SZP23] earned the prestigious Computer Science and IT Department's Best Paper Award in 2023. Furthermore, the findings presented in this dissertation have been actively disseminated through various industry and public outreach initiatives.

1. **Mohamed Seliem**, Ahmed Zahran, and Dirk Pesch. Tsn-based industrial network performance analysis. In 2022 IEEE 8th World Forum on Internet of Things (WF-IoT), pages 1–7, 2022.
2. **Mohamed Seliem**, Ahmed Zahran, and Dirk Pesch. Quality checks after production: Tsn-based industrial network performance evaluation. In 2022 4th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE), pages 1–7, 2022.
3. **Mohamed Seliem** and Dirk Pesch. Software-Defined Time Sensitive Networks (SD-TSN) for industrial automation. In 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN), pages 1–7, 2022.
4. **Mohamed Seliem**, Ahmed Zahran, and Dirk Pesch. Delay analysis of TSN based industrial networks with preemptive traffic using network calculus. In 2023 IFIP Networking Conference (IFIP Networking), pages 1–9, 2023.
5. **Mohamed Seliem**, Ahmed Zahran, and Dirk Pesch. Delay analysis of redundant TSN-based industrial networks using network calculus. In 2023 IEEE 29th International Symposium on Local and Metropolitan Area Networks (LANMAN), pages 1–3, 2023.

6. **Mohamed Seliem**, Ahmed Zahran, and Dirk Pesch. Wireless TSN Extension to Enable Deterministic Connectivity: Implementation and Performance Evaluation. In 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), pages 1–7, 2023.
7. **Mohamed Seliem**, Ahmed Zahran, and Dirk Pesch. TSN based Industrial Automation: Worst Case Analysis using Network Calculus. **Journal Paper(Under Review)**.

1.5 Use of AI Tools

In conducting the research for this thesis, AI tools such as ChatGPT were utilised to assist in various aspects of the research process. ChatGPT was employed to help with generating initial drafts, refining language, and providing suggestions for content development. While AI tools were instrumental in enhancing the clarity and coherence of the document, it is important to note that the final content, analysis, and interpretations are the result of rigorous human review and critical thinking.

The use of ChatGPT was guided by the principle of augmenting human insight rather than substituting it. All AI-generated content was carefully evaluated and revised to ensure it met the academic standards and integrity required for this thesis. This approach aimed to leverage AI's capabilities while maintaining a high level of scholarly rigour and originality.

Chapter 2

Background

Time-Sensitive Networking (TSN) has garnered significant attention, not only in research but also within various industries, reflecting a growing interest in its potential to provide bounded latency and jitter. TSN, with its diverse profiles, holds promise for a wide array of applications beyond the industrial sector, including automotive, multimedia, and healthcare. This chapter focuses specifically on the critical requirements of real-time industrial applications, aiming to contribute to the understanding and improvement of TSN performance within this context.

Research in this area has led to the development of TSN protocols and standards that cater to specific timing requirements. Key standards within the IEEE 802.1 series, such as IEEE 802.1Qbv for Time-Aware Shaping (TAS), IEEE 802.1Qbu for Pre-emption, and others within the IEEE 802.1 family, play a crucial role in shaping the landscape of TSN. These standards are being recognised beyond the IEEE realm, with adoption by organisations like ETSI for integration into emerging technologies such as 5G.

2.1 Time Sensitive Networking

Reliable and real-time data exchange is essential across various sectors, including industrial automation, healthcare, and automotive [DSA⁺14][LXW20]. Time-Sensitive Networking (TSN) emerges as a transformative technology designed to address unique challenges, ensuring precision and deterministic communication [BS19]. For instance, TSN is critical in coordinating motion control systems in manufacturing or ensuring low-latency communication in healthcare

applications. TSN is a set of standards and protocols that enhance various networking technologies, aiming to be MAC layer agnostic to deliver deterministic communication and enable seamless integration [NTA⁺18].

Various industrial applications, such as manufacturing, process automation, robotics, power generation, and transportation, rely heavily on interconnected systems to optimise efficiency, productivity, and safety. Traditional Ethernet networks may struggle to guarantee precise timing, low-latency, and synchronization required for time-sensitive industrial processes [VZS19]. TSN addresses these issues by providing deterministic communication, which ensures data is transmitted within known time bounds and minimises uncertainties.

2.1.1 Key Principles of TSN

TSN's capabilities are rooted in several key principles:

- **Time Synchronisation:** TSN uses the IEEE 1588 Precision Time Protocol (PTP) to ensure precise time alignment across devices [syn20][Sta18][IEE21a]. This synchronization enables coordinated actions and precise event time stamping.
- **Deterministic Communication:** TSN ensures data transmission within known and predictable time bounds, eliminating uncertainties. This is crucial for applications requiring precise timing.
- **Traffic Shaping and Scheduling:** TSN employs mechanisms such as Time-Aware Shaping (TAS) and Credit-Based Shaping (CBS) to allocate network bandwidth effectively. TAS regulates traffic based on precise time slots, while CBS uses credits to manage bandwidth allocation [IEE18][IEE10].
- **Quality of Service (QoS):** TSN introduces differentiated QoS to cater to various traffic classes, ensuring that time-critical data receives priority while accommodating less time-sensitive data.
- **Redundancy and Fault Tolerance:** TSN incorporates redundancy mechanisms like Parallel Redundancy Protocol (PRP) [IEE17] and High-availability Seamless Redundancy (HSR) [HR21] to enhance network reliability and fault tolerance.
- **Resource Reservation:** TSN enables precise bandwidth and latency reservation for time-sensitive applications, ensuring consistent and predictable

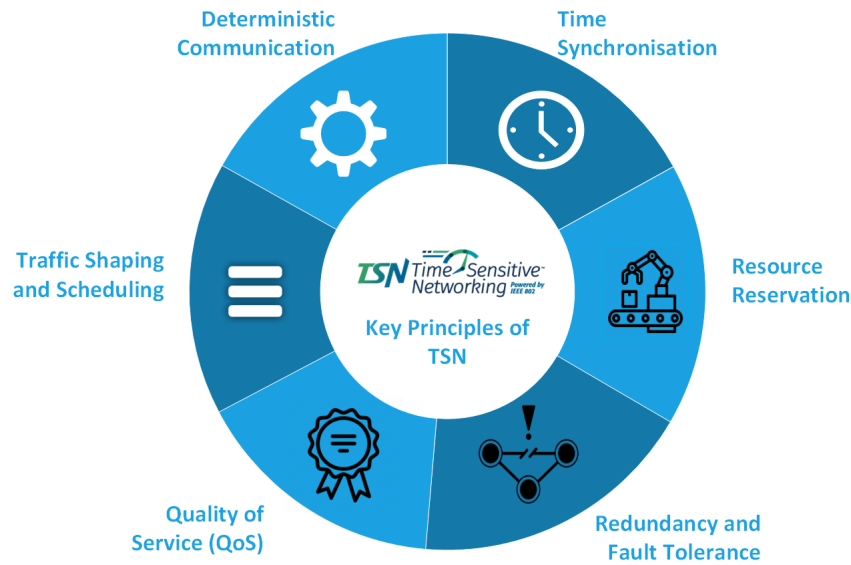


Figure 2.1: Key Principles of Time-Sensitive Networking.

performance.

By embracing these principles, TSN contributes to improvements in industrial communication, facilitating enhanced reliability, real-time capabilities, and coordinated data transmission.

2.1.2 Traffic Shaping and Scheduling

Time-Sensitive Networking introduces sophisticated traffic shaping and scheduling mechanisms to manage network resources and prioritise critical data:

- **Traffic Shaping:** TSN uses Time-Aware Shaping (TAS) to regulate data transmission rates and Credit-Based Shaping (CBS) to allocate bandwidth efficiently. TAS prevents excessive burstiness by adhering to predefined time slots, while CBS uses credit mechanisms to manage network load [IEE18][IEE10].
- **Scheduling:** TSN employs time-triggered and time-aware scheduling to manage data transmission. Time-triggered scheduling follows a strict timeline for data frames, while time-aware scheduling allows dynamic adjustments based on timing requirements [MSKS13].
- **Coordinated Traffic Flow:** The synergy between traffic shaping and scheduling ensures that critical traffic classes receive the necessary re-

sources and priority, while non-critical traffic adapts to available bandwidth.

The combination of these mechanisms enables precise control, deterministic communication, and low-latency data transmission, facilitating advanced automation and Industry 4.0.

2.1.3 Redundancy and Fault Tolerance

Reliability and fault tolerance are crucial in industrial automation. TSN employs various redundancy mechanisms:

- **Parallel Redundancy Protocol (PRP):** PRP creates two independent data paths for transmission, ensuring zero downtime in case of a failure [IEE17].
- **High-availability Seamless Redundancy (HSR):** HSR provides sub-millisecond failover times by creating a redundant ring topology [HR21].
- **Fault Detection and Recovery:** TSN includes mechanisms for fault detection and recovery [GZPS17]. Network devices monitor link health and activate redundant paths as needed.
- **Seamless Integration with Industrial Protocols:** TSN integrates with legacy Ethernet and industrial protocols, allowing gradual adoption and leveraging existing infrastructure [KKH14].

These mechanisms ensure continuous operation and minimise the impact of network failures, contributing to improved reliability and performance in industrial applications.

2.1.4 Industrial Profile of TSN

The Industrial Profile of TSN [G⁺21] extends TSN standards to meet the unique requirements of industrial applications:

- **Time-Triggered Traffic:** Emphasises precise scheduling for synchronised operations in industrial environments.
- **Support for Multiple Real-Time Traffic Types:** Includes support for various real-time traffic types like audio-video bridging and time-sensitive network bridging.

- **Deterministic Forwarding:** Ensures real-time data is forwarded with strict adherence to timing requirements.
- **Seamless Integration with Legacy Ethernet:** Allows for the integration of TSN into existing networks without disruption.
- **Configuration and Diagnostics:** Introduces specialised traffic classes for configuration and diagnostics, enhancing network manageability.

The Industrial Profile of TSN aims to enhance productivity, security, and connectivity in manufacturing and industrial automation through its focus on time-triggered traffic, real-time support, and seamless integration.

2.2 Network Calculus Principles

Network calculus [LBT01][BARW11][LB96][VBK16] is a mathematical framework designed to analyse and optimise the performance of communication networks, especially where real-time communication and strict performance guarantees are essential. It offers a systematic approach to understanding data flows within networks, enabling effective resource allocation and addressing time-sensitive needs. This section explores network calculus principles and their application in evaluating and optimising Time-Sensitive Networking (TSN) standards and protocols.

Structure of the Section:

- **Overview of Network Calculus:** We will provide a concise overview of the key concepts and principles of network calculus, including fundamental theorems and mathematical tools used for analysing data flow in communication networks.
- **Application of Network Calculus in TSN:** This subsection will explore how network calculus is applied within the context of Time-Sensitive Networking, focusing on its role in analysing TSN performance and ensuring timely delivery of critical data.
- **Deterministic Bounds for Real-Time Traffic:** We will discuss how network calculus helps derive deterministic bounds on packet delays and buffer requirements for real-time traffic, and how this analysis supports the design of TSN to meet strict timing constraints.
- **Traffic Shaping and Resource Allocation:** This subsection will highlight

how network calculus assists in analysing traffic shaping and resource allocation. We will examine how service curves facilitate bandwidth reservation and allocation, providing insights that inform TSN design decisions.

- **Evaluation of Network Performance:** The final part will focus on applying network calculus to evaluate the performance of time-sensitive networks. We will review performance measurement approaches and how well TSN deployments achieve their objectives.

By examining network calculus principles and their application to TSN, this section demonstrates how this mathematical framework enables efficient, predictable, and reliable communication within modern network infrastructures.

2.2.1 Overview of Network Calculus

Network calculus provides a framework for understanding and optimising the deterministic performance bounds of communication networks, rooted in queuing theory [Jia09]. It allows engineers to model and analyse network performance under certain conditions, providing insights into traffic behaviour and service guarantees. However, it is important to acknowledge that network calculus may not fully capture the stochastic nature of real-world networks, and its deterministic insights should be considered alongside the inherent variability in network performance.

The core concepts of network calculus involve analysing data flow within network nodes using two main types of curves:

Arrival Curve α : The arrival curve characterises the pattern of data arrival into a network node. It quantifies the amount of data arriving over a specific period. For example, a constant bit rate traffic source can be described by a linear arrival curve, indicating a steady data arrival rate, as shown in Figure 2.2.

Service Curve β : The service curve represents the output behaviour of a network node, defining the maximum rate at which the node can process and transmit data. A strict service curve ensures that data exits the node no faster than the specified rate.

Network calculus uses these curves to analyse the interaction between data arrival and service at network nodes, deriving fundamental performance metrics.

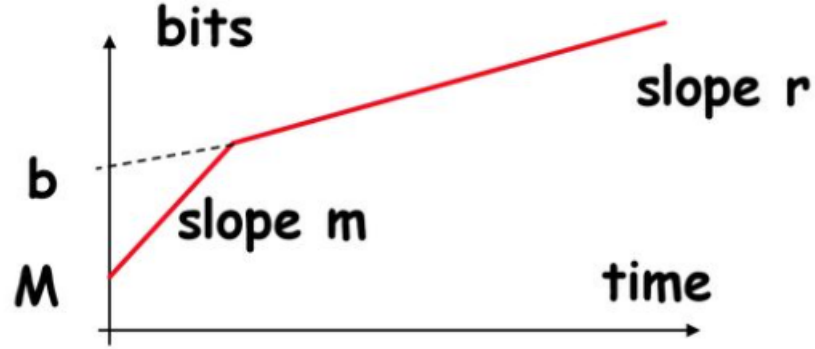


Figure 2.2: Standard Arrival Curve

A key result in network calculus is the Convolution Theorem [LBT01], which facilitates mathematical analysis of network data flows.

Convolution Theorem: This theorem states that the output envelope of a network node is the convolution of the arrival curve and the service curve. Mathematically, it is expressed as the minimum convolution of these curves. This convolution operation allows engineers to predict worst-case network behaviour and estimate crucial metrics, such as delay and backlog experienced by data packets.

The most commonly used formulations are for the convolution and deconvolution operations, which are defined by [LBT01] as follows:

$$f \otimes g(t) = \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\} \quad (2.1)$$

$$f \oslash g(t) = \sup_{0 \leq s} \{f(t+s) - g(s)\} \quad (2.2)$$

2.1, min-plus convolution, has the same proprieties as usual convolution, e.g. commutative and associative. It can be easily calculated by placing one of the curves $g(t)$ at each point of the other curve $f(t)$. The resulting curve $f \otimes g(t)$ connects all the minimum points of the placed instances of $g(t)$ as depicted in Figure 2.3.c. Moreover, the deconvolution in 2.2, depicted in 2.3.d, is obtained by shifting the curve $g(t)$ by $t_0 \forall t_0 \in [0, t]$ and increasing the curve $f(t)$ by the maximum vertical deviation between $f(t)$ and $g(t - t_0)$.

One of the key applications of network calculus is its ability to analyse and assess the potential for providing Quality of Service (QoS) guarantees for different classes of traffic in communication networks. By carefully designing and bounding arrival and service curves, engineers can gain valuable insights into

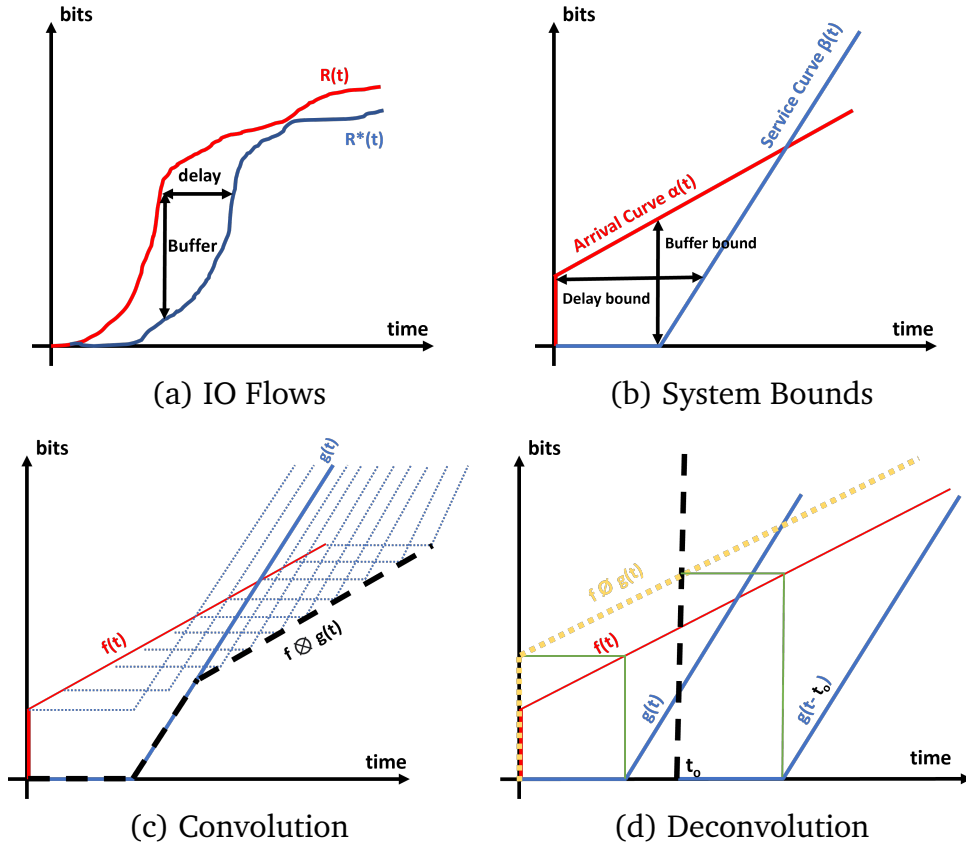


Figure 2.3: Network Calculus rules. Adapted from [MHG20].

the network's performance and its capacity to meet QoS requirements. Implementing QoS guarantees involves specific resource management algorithms that leverage the insights from network calculus analysis to optimise the treatment of both time-critical and non-critical data streams.

Network calculus has significantly contributed to the development of real-time communication protocols, such as the IEEE Time-Sensitive Networking (TSN) standard. It has been instrumental in analysing TSN to ensure the timely delivery of critical data and establishing synchronization and time-awareness across the network [Mes18][DXL⁺22].

In summary, network calculus offers a robust and adaptable mathematical framework for evaluating, constructing, and enhancing communication networks. By leveraging network calculus (NC) as a methodology, engineers can systematically build efficient and reliable networks tailored to the stringent requirements of time-sensitive applications. The use of concepts such as envelopes and the Convolution Theorem in network calculus not only facilitates insightful analysis but also serves as a valuable tool in the design process.

This dual role of network calculus is transformative in modern networking paradigms [LBT01][KS22]. The following subsections will delve deeper into the practical applications of network calculus in Time-Sensitive Networking and its role in ensuring the successful deployment of real-time communication systems.

2.2.2 Application of Network Calculus in TSN

Time-Sensitive Networking represents a paradigm shift in communication networks, designed to meet the demands of latency-sensitive and time-critical applications. By integrating network calculus principles into the TSN architecture, engineers can ensure reliable, predictable, and deterministic data transmission, making TSN a robust solution for industrial automation, automotive communication, and other mission-critical applications.

Guaranteed Timely Data Delivery: In TSN, certain applications necessitate that data be delivered within stringent time bounds to maintain system stability and safety. Network calculus enables engineers to analyze the network's behaviour under varying traffic loads and derive deterministic bounds on data packet delays. This capability enhances the understanding of data flow characteristics and supports the effective management of critical data packets within predefined time constraints. While network calculus provides valuable insights, implementing mechanisms to ensure timely delivery involves specific design considerations and resource management strategies aimed at preventing system failures or data loss.

Quality of Service Assurances: Network calculus aids in providing QoS assurances in TSN by informing the design and analysis of scheduling techniques that enforce differentiated treatment of traffic classes. By defining appropriate arrival and service curves for different traffic streams, engineers can allocate resources based on application priority levels. This approach allows TSN networks to implement bandwidth reservation and priority-based scheduling, ensuring that time-sensitive data receives higher priority compared to non-critical traffic.

Traffic Scheduling and Shaping: TSN networks often encounter variations in traffic patterns due to the coexistence of diverse applications with differing bandwidth requirements. Network calculus supports the development of efficient traffic scheduling and shaping algorithms to manage these variations. By

modelling and analysing traffic using arrival and service curves, engineers can regulate data flow to prevent congestion and prioritise critical traffic during network contention.

Resource Allocation and Admission Control: Admission control is crucial in TSN for determining whether to accept or reject new traffic streams based on available network resources. Network calculus enables engineers to calculate the maximum allowable arrival rates of new traffic streams without compromising the performance guarantees of existing streams. This aids in effective resource allocation and capacity planning to accommodate the dynamic nature of time-sensitive applications.

Bandwidth and Buffer Management: Bandwidth reservation and buffer management are vital for ensuring predictable delivery of critical data in TSN networks. Network calculus helps engineers calculate the required buffer sizes to store time-sensitive packets and prevent buffer overflows. This approach minimises jitter and enhances data transmission predictability, essential for time-critical systems.

Validation of TSN Configurations: Network calculus provides a structured approach to validate and verify the performance of TSN configurations. Engineers can simulate various traffic scenarios, evaluate network behaviour under different loads, and ensure that the deployed TSN infrastructure meets the desired QoS requirements.

In summary, the application of network calculus in Time-Sensitive Networking greatly enhances the reliability and predictability of communication systems. By leveraging network calculus principles, TSN can guarantee timely delivery of critical data, provide differentiated QoS assurances, and effectively manage network resources. Integrating network calculus into TSN holds promise for improving the performance of time-sensitive applications and supporting industries that rely on reliable, real-time communication and control systems. Detailed discussions on this integration and its contributions are covered in later chapters.

2.2.3 Deterministic Bounds for Real-Time Traffic

Network calculus provides a valuable foundation for assessing a network's capability to deliver time-sensitive data reliably, which is crucial for various networking environments, including Time-Sensitive Networking. Its unique abil-

ity to derive deterministic bounds on various network performance metrics for real-time traffic is instrumental in meeting the stringent requirements of critical applications.

Max-Plus Algebra and Convolution Theorem: At the core of network calculus [VBK16] are Max-Plus algebra and the Convolution Theorem, which are essential for calculating deterministic bounds for real-time traffic streams. Max-Plus algebra allows for the modelling of packet transmission times and service capabilities of network elements in a unified framework, enabling the analysis of worst-case scenarios.

Maximum Delay Calculation: By utilising Max-Plus algebra and the Convolution Theorem, network calculus can accurately determine the maximum delay experienced by real-time data packets as they traverse the network. Engineers can predict upper bounds on packet delays with high precision, facilitating the design of TSN networks that adhere to strict deadlines essential for mission-critical applications.

Buffer Requirements Analysis: Deterministic bounds on buffer requirements are crucial in real-time systems to prevent packet loss and ensure smooth data flow. Network calculus helps engineers calculate the necessary buffer sizes at different network nodes, ensuring efficient storage and transmission of time-sensitive packets without causing congestion.

Enhancing Reliability in TSN Applications: The derived bounds on delay and buffer requirements provide essential guidelines for improving the reliability of TSN applications. Incorporating these insights into TSN network design and configuration ensures reliable data transmission, even under demanding conditions.

Avoiding Packet Loss: Packet loss can severely impact time-sensitive environments, leading to compromised performance or safety hazards. Network calculus enables engineers to calculate buffer sizes needed to avoid packet loss, thereby ensuring uninterrupted data delivery and overall system stability.

Optimization of Network Performance: The ability to calculate deterministic bounds through network calculus is crucial for optimising TSN network performance. Network calculus allows engineers to model and evaluate network behaviour analytically, helping them make informed decisions on adjusting parameters and resource allocations. Studies such as [LBC12] illustrate how this analytical approach aids in balancing QoS requirements with efficient resource

utilization in TSN networks.

Validation of TSN Design: Deterministic bounds derived from network calculus facilitate thorough validation of TSN designs. By comparing calculated bounds with performance targets, engineers can assess the feasibility and effectiveness of their TSN implementations.

2.2.4 Traffic Shaping and Resource Allocation

In Time-Sensitive Networking, traffic shaping and efficient resource allocation are crucial for ensuring predictable and reliable data transmission. Network calculus provides a powerful framework for designing traffic shaping algorithms and optimising resource allocation to meet the diverse service requirements of different traffic classes.

Traffic Shaping Algorithms: Traffic shaping algorithms are essential in TSN for regulating data flow to meet specified service requirements. Network calculus supports the design of effective traffic shaping mechanisms by providing analytical models to predict network behaviour. For instance, network calculus can help design a shaping mechanism that smooths out bursty traffic, aligning data transmission rates with network capacity and preventing congestion.

By analysing arrival and service curves of various traffic streams, engineers can design shaping policies that mitigate excessive data bursts, reduce packet jitter, and ensure a controlled and predictable flow of time-sensitive data. Well-designed shaping algorithms are critical for meeting QoS guarantees and avoiding congestion in TSN networks.

Service Curves for Bandwidth Allocation: A key concept in network calculus is service curves [BJT10], which define the maximum rate at which a network element can process and transmit data. Engineers can use service curves to allocate bandwidth and prioritise traffic based on its criticality.

By assigning appropriate service curves to different traffic classes, such as real-time control data, video streams, and best-effort traffic, engineers ensure that time-sensitive applications receive the necessary bandwidth to meet their performance requirements. This dynamic resource allocation, based on service curves, allows TSN networks to adapt to changing traffic conditions and prioritise mission-critical data during network congestion or contention. Translating service curves into switch configurations involves network management mechanisms that utilise insights gained through network calculus analysis.

Guaranteeing Resources for Critical Applications: In TSN, certain applications require stringent resource guarantees to ensure timely and reliable operation. Network calculus facilitates the reservation and allocation of resources for critical applications, providing them with the bandwidth necessary to meet specific demands.

Engineers can use network calculus to analyse the behaviour of critical traffic and calculate the minimum bandwidth required to maintain desired QoS guarantees. This approach ensures that time-sensitive data packets receive preferential treatment, even when non-critical traffic saturates the network.

Optimising Resource Utilization: Efficient resource allocation is crucial in any network, including TSN. Network calculus enables engineers to optimise resource utilization by fine-tuning bandwidth and buffer space allocation among different traffic classes. For example, network calculus analysis might reveal that allocating more bandwidth to time-critical traffic while allowing less critical traffic to use the remaining capacity can enhance overall performance. By leveraging network calculus insights, engineers can strategically allocate resources to prioritise critical applications, ensuring efficient utilization and meeting QoS requirements.

Considering network capacity, service curves, and traffic demands, engineers can balance providing necessary resources for critical applications with maximising overall network efficiency. This optimization leads to improved network performance and better resource utilization, making TSN networks more robust and capable of handling diverse traffic scenarios.

Enabling Multi-Service Networks: TSN often supports a variety of applications with different service requirements. Network calculus enables the design of multi-service networks, where different traffic classes coexist and receive appropriate resources and prioritization. For example, network calculus analysis can guide the design of a TSN network that accommodates both time-critical traffic from industrial automation systems, which requires low latency, and less critical traffic from monitoring applications, allowing for flexible and adaptive resource allocation.

By iteratively refining traffic shaping algorithms and resource allocation strategies based on insights from service curve analysis, engineers can create TSN networks that meet the needs of diverse applications, including real-time control, audio, video, and best-effort data. This versatility allows TSN to be a

comprehensive and flexible solution for time-critical applications in industrial automation, automotive, and other industries.

Traffic shaping and efficient resource allocation, empowered by network calculus principles, are essential for achieving optimal network performance in Time-Sensitive Networking. By employing traffic shaping algorithms, leveraging service curves for bandwidth allocation, and guaranteeing resources for critical applications, TSN networks can deliver reliable, predictable, and high-performance data transmission, ensuring seamless operation of time-critical applications in dynamic environments.

2.2.5 Evaluation of Network Performance

Evaluating network performance is essential for validating the effectiveness of Time-Sensitive Networking (TSN) implementations. Network calculus provides a rigorous framework for assessing key performance metrics and analysing network behaviour under various traffic scenarios. By leveraging network calculus, engineers can gain insights into the network's capabilities, identify potential bottlenecks, and fine-tune the system to optimise its performance.

Several key metrics are particularly important when evaluating TSN network performance:

Delay: Network calculus allows engineers to analyse and calculate the maximum packet delay for different traffic classes and network paths, providing valuable insights into worst-case scenarios. For average delay calculations, additional methodologies such as queuing theory may be employed. This analysis is crucial for ensuring that critical applications meet their timing requirements and maintain reliable communication.

Jitter: Jitter, or the variation in packet arrival times, can significantly impact the synchronization of time-sensitive applications. Network calculus facilitates the calculation of jitter bounds, enabling engineers to assess the stability and predictability of data delivery. Managing jitter is essential for applications requiring precise timing and coordination.

Packet Loss: Packet loss can lead to data retransmissions and compromise the reliability of time-sensitive systems. Network calculus helps evaluate the packet loss rate under various conditions, providing insights into the network's robustness and identifying potential areas for improvement. Understanding

packet loss rates is vital for maintaining system reliability and ensuring data integrity.

Backlog: Backlog refers to the amount of data awaiting transmission at network nodes. By analysing backlog using network calculus, engineers can assess the adequacy of buffer sizes and the effectiveness of traffic shaping mechanisms. This analysis ensures that the network can handle data without excessive delays or congestion.

Network calculus offers several analysis techniques to evaluate TSN network performance:

Worst-Case Scenario Analysis: Engineers use worst-case scenario analysis by considering the convolution of arrival and service curves to identify scenarios where network performance is most challenging. This technique helps validate the network's ability to meet stringent performance guarantees by examining the network's behaviour under extreme conditions.

Stability Analysis: Stability is critical for ensuring consistent and predictable performance in TSN networks. Network calculus facilitates stability analysis by examining data arrival rates and the network's capacity to handle traffic. Stability definitions and conceptualizations in this context are discussed in studies such as [NTA⁺18] and [ZHLL18], which contribute to a comprehensive understanding of system behaviour.

Sensitivity Analysis: Sensitivity analysis investigates how changes in network parameters or traffic patterns impact performance. Network calculus enables engineers to perform sensitivity analysis, helping them understand the effects of different configurations on network behaviour. This analysis is crucial for optimising network performance under varying conditions.

The application of network calculus in evaluating TSN designs is a critical validation step. By comparing mathematical formulations and defined performance metrics with predefined targets, engineers can verify whether the network meets its performance goals. This validation ensures that the network is capable of delivering the required QoS for time-sensitive applications.

Performance evaluation also helps identify potential areas for optimization. By analysing performance metrics, engineers can fine-tune network configurations, traffic shaping algorithms, and resource allocations to enhance overall efficiency and reliability. This iterative process of optimization leads to a more

robust and effective TSN implementation.

Furthermore, performance evaluation provides real-world insights into the network's behaviour under dynamic conditions. Engineers can use the knowledge gained from this process to make informed decisions about network scaling, capacity planning, and future enhancements.

In conclusion, evaluating network performance using network calculus is a crucial step in assessing the effectiveness of Time-Sensitive Networking implementations. By analysing key performance metrics and employing various analysis techniques, engineers can validate network performance, optimise behaviour, and gain valuable insights to ensure the successful deployment of time-critical applications across diverse industries.

2.3 Relevant Related Work

The intersection of timing analysis, Time-Sensitive Networking, and multi-class traffic management has emerged as a pivotal area of research in the context of industrial real-time applications. Industrial environments demand precise and deterministic communication to ensure safe and efficient operations. This necessitates a comprehensive exploration of existing literature to understand the advancements, challenges, and methodologies that characterise the field. The complexity of industrial environments is compounded by the coexistence of diverse traffic classes, each characterised by distinct timing constraints and QoS requirements. These classes often include critical control traffic, real-time data flows, and less time-sensitive technical or business operations traffic. A nuanced examination of research addressing the management of these multi-class traffic scenarios is vital to appreciate the strategies devised to allocate network resources efficiently while respecting individual class requirements. Within this context, timing analysis plays a crucial role in evaluating the performance of TSN-based industrial systems. The analysis encompasses methods for estimating worst-case communication latencies, assessing the impact of network congestion, and validating compliance with specified timing constraints. The literature encapsulates various approaches, including analytical modelling, simulation-based techniques, and empirical studies. By surveying these methods, we gain insights into their applicability, accuracy, and limitations in the context of TSN networks catering to multi-class traffic. The scientific foundation established in prior research not only informs our present study but also de-

lineates the contours of unexplored territory. By reviewing the developments in timing analysis techniques, investigating the practical challenges faced during experimental validation, and identifying gaps in the current body of knowledge, we motivate our contributions. As we progress through the following sections, we conduct an in-depth exploration of the pertinent literature. This review not only orients our research within the broader topic of timing analysis for TSN-based systems but also positions us to make informed strides toward addressing the intricate challenges, such as optimising latency and ensuring Quality of Service, presented by multiple classes of traffic. By synthesising these insights, we intend to synthesise a coherent framework for our research, capitalising on the current state of the art while advancing knowledge in the understanding of timing analysis for TSN-based industrial real-time applications with diverse classes of traffic.

As this thesis encompasses several interconnected research domains, we organise the review of related work into distinct subsections. To commence, it is imperative to recognise that TSN does not stand as the exclusive solution for real-time Ethernet. Within subsection 2.3.1, we provide an inclusive survey of existing alternatives to real-time Ethernet solutions, delineating their standout features, merits, and drawbacks.

Subsequently, in subsection 2.3.2, we offer a condensed overview of the extensive body of research dedicated to TSN. Within this realm, research papers exhibit a spectrum of approaches, some scrutinising the constituent elements of TSN in isolation, while others furnish an evaluation of TSN as a communication system realising its full potential with the inclusion of all associated standards. A myriad of evaluation methods has been proposed, encompassing simulations of TSN, alongside analytical models derived through the NC methodology.

Moving forward to subsection 2.3.3, we present an inventory of TSN simulators that have been developed using a variety of simulation tools. Subsection 2.3.4 is devoted to the presentation of worst-case end-to-end delay analytical models, formulated through the application of the network calculus, and these models are configured to account for different TSN functionalities.

2.3.1 Evaluating Real-time Ethernet Solutions:

Ethernet offers a range of advantageous qualities such as broad applicability, compatibility, and cost-effectiveness. Nevertheless, its non-deterministic be-

haviour resulting from the CSMA/CD arbitration mechanism renders it unsuitable for real-time communication. Various strategies have been proposed to address this challenge, as detailed in the following references.

In [Doy04a], P. Doyle elucidates the inherent characteristics of conventional Ethernet, highlighting its limitations in non-deterministic applications. Collision avoidance can be achieved through the implementation of full-duplex links and the integration of switches into the network topology. The concept of Switched Ethernet can be merged with real-time protocols to ensure determinism.

In a subsequent publication [Doy04b], an exploration of several existing real-time solutions that leverage these concepts is undertaken. The analysis encompasses EtherNet/IP, PROFINet, Ether-CAT, and ETHERNET Powerlink, accompanied by an assessment of the respective advantages and disadvantages of each approach. Moreover, the publication offers insights into the typical network topologies and frame formats associated with these solutions. This pioneering work represents one of the earliest research endeavours in the realm of real-time Ethernet and has served as a foundational reference for subsequent research papers. Additionally, it introduces the IEEE 1588 standard, which guarantees sub-microsecond synchronisation accuracy of distributed clocks.

In their white paper [APA5], Kingstar conducts an extensive examination of five distinct protocols designed for real-time Ethernet applications. The protocols under comparative scrutiny encompass EtherCAT, EtherNet/IP, Ethernet Powerlink, PROFINET IRT, and SERCOS III. This evaluation ultimately favours EtherCAT as the most promising solution in terms of both performance and cost-effectiveness. Importantly, all the protocols reviewed adhere to open standards, obviating the need for exclusive vendor technologies. The paper serves as a valuable resource for understanding the current state of real-time Ethernet practices. However, it is worth noting that the paper's approach may not align with the expectations of the academic research community, as it lacks a detailed explanation of experimental design, rendering the results non-reproducible.

Meanwhile, in the work by T. Steinbach and colleagues [SKS11], the focus shifts to exploring the feasibility of implementing switched Ethernet in in-car networking scenarios. Traditional in-car communication solutions like Controller Area Network (CAN) and FlexRay exhibit limitations in scalability, which becomes increasingly pertinent as the number of network nodes continues to grow. The integration of switched Ethernet with real-time protocols is investi-

gated as a potential solution to meet temporal constraints. Within this study, TTEthernet is evaluated as a prospective backbone for in-car networks, offering three distinct classes of traffic: time-triggered, rate-constrained, and best-effort. The analysis demonstrates that time constraints can be met, albeit with some limitations in the best-effort class's jitter. However, it is essential to note that the paper lacks the provision of simulation tools and experiment configurations, thus precluding the reproducibility of the reported findings.

In their research documented in [AML⁺17], M. Ashjaei and collaborators delve into an investigation of the performance of a switched Ethernet network for vehicular applications, making use of the AVB protocol. AVB, which precedes TSN (Time-Sensitive Networking), comprises four fundamental standards: Audio Video Bridging Systems (IEEE 802.1BA), Timing and Synchronisation for Time-Sensitive Applications (IEEE 802.1AS), Stream Reservation Protocol (IEEE 802.1Qat), and Forwarding and Queuing for Time-Sensitive Streams (IEEE 802.1Qav). AVB defines two distinct classes of traffic, namely high-priority class A and low-priority class B. The standard also introduces the Credit-Based Shaping algorithm, designed to mitigate traffic bursts.

Notably, the use-case model adopted in this analysis mirrors the one previously detailed in a separate study [LWH11] conducted by the BMW Group Research and Technology. The results obtained from this investigation affirm the schedulability of the task set when employing AVB. It is important to underscore that this thesis heavily draws upon the work undertaken by M. Ashjaei et al., employing an identical use-case and model configuration.

2.3.2 Performance Assessment of TSN Amendment

Evaluating the performance of the TSN standard necessitates the examination of all its constituent amendments. While a comprehensive analysis of AVB has already been undertaken, other aspects of TSN, e.g. scheduled traffic (IEEE 802.1Qbv) and frame pre-emption (IEEE 802.1Qbu), should be considered as these are integral to TSN's extension of the AVB standard.

In their work, Luxi Zhao et al. [ZPC18] conduct an extensive worst-case analysis of IEEE 802.1Qbv for TSN using network calculus. Their focus is exclusively on scheduled traffic, omitting frame pre-emption due to its potential complexity. The paper begins by elucidating the functionality of an IEEE 802.1Qbv-capable switch, highlighting its routing table revisions, traffic forwarding, priority filter-

ing based on PCP, and queue management through Gate Control Lists (GCLs). The worst-case latency analysis of scheduled traffic in this context assumes unrestricted GCLs, meaning that scheduled traffic windows do not overlap, enabling multiple scheduled traffic classes with different priorities. The accuracy of their analysis is validated through two test cases, one synthetic with six nodes and two switches, and another more realistic with 31 nodes and 15 switches. Both cases demonstrate compliance with real-time constraints for scheduled traffic. The provided analysis not only ensures the real-time capabilities of TSN, but also facilitates the synthesis of GCLs, thereby enhancing the scalability of TSN.

The utilisation of IEEE 802.1Qbv raises the challenge of devising a valid schedule for time-triggered traffic. Silviu S. Craciunas et al. [COCS16] address this issue by identifying functional parameters of the amendment, delineating its constraints, and articulating deterministic Ethernet constraints. They propose an offline scheduler based on these constraints, guaranteeing low and deterministic latency for time-triggered traffic. Key parameters include device capabilities and queue configuration, with a focus on deterministic networks where both switches and end-systems are scheduled. The number of queues, gate operation modes, and various scheduling constraints are also outlined. Satisfiability Modulo Theories (SMT) are employed to solve the scheduling problem. Their experiments reveal that the scheduling complexity increases with network utilisation, prompting the exploration of several optimisation avenues.

W. Jia et al. [JLC13] undertake the performance evaluation of IEEE 802.1Qbu, addressing the challenge posed by the need for deterministic end-to-end latency in time-critical traffic (TCT) without frame pre-emption. They introduce a frame pre-emption mechanism to resolve this issue. In cases of non-time-critical traffic (NTCT) transmission, TCT could experience MTU transmission delay. IEEE 802.1Qbu allows TCT to pre-empt NTCT upon arrival at the frame pre-emption switch, resuming NTCT transmission after TCT completion. The MAC layer of a switch identifies frame priority through a 3-bit priority field in the standard Ethernet header. Control symbols, namely HOLD and RETRIEVAL, are utilised to inform receivers about frame pre-emption, enabling smooth transitions between traffic streams. Simulation experiments employ three scheduling schemes: FIFO, non-pre-emptive priority scheduling (NPPS), and pre-emptive priority scheduling (PPS), with results demonstrating significant reductions in average jitter and delay when PPS is employed. The paper effectively highlights the advantageous characteristics of IEEE 802.1Qbu, showcasing significant im-

provements in real-time communication within Ethernet networks through this amendment.

Lastly, Lin Zhao et al. [ZHLL18] conduct a comparative analysis of TSN and TT Ethernet, focusing on their similarities and differences. Both technologies employ Time-Division Multiple Access (TDMA) to avoid traffic collisions by precisely dividing transmission time. The authors introduce both technologies, elucidating their respective network architectures. In terms of bandwidth allocation, TSN surpasses TT Ethernet, as the latter may lead to low-priority flows starving in the presence of large high-priority traffic transmission. TSN, conversely, employs the Stream Reservation Protocol (SRP) to ensure fairness in bandwidth allocation for low-priority traffic. The authors also compare TSN and TT Ethernet in terms of delay analysis, highlighting TSN's superior performance due to mechanisms such as CBS and frame pre-emption that allow for high bandwidth and minimal end-to-end delay. TT Ethernet's capability to combine Time-Triggered (TT) and Rate-Constrained (RC) traffic in various ways is discussed, as is its redundancy approach, which falls short compared to TSN's ability to avoid frame loss by utilising multiple paths. In summary, TSN emerges as the more flexible and adaptable solution, capable of accommodating modifications to network topology, whereas TT Ethernet requires an entirely new schedule to accommodate changes.

2.3.3 Modelling Time Sensitive Networking through Simulation

In the paper [JLH⁺18], the authors introduce a TSN simulation model that they developed using OMNeT++. Within the array of frameworks that extend the functionalities of OMNeT++, they leverage the INET framework and Communication over Real-time Ethernet (CoRE4INET) extension to construct their TSN simulator. Alongside these frameworks, they also implement amendments of IEEE 802.1Qbv and IEEE 802.1QAS. The core objective of this simulation effort is to verify the capability of TSN in ensuring deterministic end-to-end latency. The paper elaborates on the development process of the TSN simulation model, which comprises two primary components.

The first part involves the creation of a TSN switch with prioritisation and gating capabilities. Traffic is directed to specific queues based on its priority, and the transmission is controlled through Gates Control Lists. In the second part, the authors establish a network topology to validate the functionalities of the

switch. The paper also offers detailed guidance on how to derive GCLs for various network topologies and their associated parameters. GCLs are employed to facilitate diverse transmission modes, including protected transmission of time-triggered messages, unprotected transmission of other messages, and the use of a guard band. By ensuring that the guard band accommodates the worst-case scenario, where it equals the maximum frame transmission time, the authors guarantee that unprotected messages do not interfere with protected ones. The simulation results conclusively demonstrate that TSN, along with its two amendments, can indeed ensure deterministic and low-end latency for protected traffic. Notably, the best-case, average-case, and worst-case response times for protected traffic remain identical, while the latency of unprotected traffic increases over time, rendering it non-deterministic.

P. Heise et al. introduce the TSimNet simulation framework, which encompasses all non-time-based TSN standards in [HGO16]. Acknowledging that time synchronisation is unsuitable for avionic networks due to certification constraints, its implementation was deemed unnecessary. TSimNet is implemented using the OMNeT++ simulation tool and its extension, the INET framework. Frame pre-emption, frame replication and recovery, and per-stream filtering represent the amendments integrated into the existing framework. The authors embark on a performance evaluation of TSN, commencing with a simple topology to assess the impact of frame pre-emption. They compare minimum, maximum, and average end-to-end latencies with and without frame pre-emption, conclusively revealing that frame pre-emption significantly reduces latency. In the subsequent evaluation, an industrial topology is employed, and the results underscore the dependency of latency reduction on the network's configuration.

The thesis report authored by H. Laine [Lai15] offers a succinct overview of TSN amendments, with a specific focus on three: Time-aware Shaper, Credit-based Shaper, and Frame pre-emption. The author devises a TSN simulator implemented in Java, albeit lacking a graphical user interface. Consequently, simulating various system configurations necessitates the provision of files defining messages, specifying topology, virtual links, and simulation configurations. The simulation results affirm that time-critical frames adhere to the intended transmission schedule with bounded latency. Additionally, it is observed that the amendments related to frame pre-emption and Credit-based Shaper exhibit preferences for Audio-Video frames and Best Effort traffic.

J. Falk and his team introduce the NeSTiNg Network Simulator for Time-

Sensitive Networking [FHC⁺19], which is constructed with OMNeT++. NeSTiNg harnesses numerous components and modules from the INET framework, augmenting them with three pivotal TSN features: the Time-aware shaper (IEEE Std 802.1Qbv), Credit-based shaper (IEEE Std 802.1Qav), and Frame pre-emption (IEEE Std 802.1Qbu). At the heart of NeSTiNg lies a TSN switch, predominantly implemented using pre-existing modules from INET. The configuration of switch ports is adaptable, with each port comprising three core modules: *eth*, *processingDelay*, and *relayUnit*.

Upon receipt of an incoming frame, it initiates its journey through the *eth* module. As an incoming frame, it is promptly directed from the mac module through the express queue. The *processingDelay* component introduces a measured delay to the frame, while the *relayUnit* module meticulously routes it to its designated output port. Subsequently, the frame traverses the *processingDelay* component residing within the output port. When cast as an outgoing frame, it advances through the *queuingFrames* component, which judiciously assesses the frame's priority based on the PCP field, subsequently mapping it to the appropriate queue. The *transmissionSelection* component takes charge of selecting the subsequent frame for transmission by evaluating queue priorities and their respective states. Gate management for the queues is overseen by the *gateController* component, which maintains a gate control list. The frame is then dispatched in accordance with either a strict priority or credit-based algorithm. The *tsAlgorithms* component is adjustable by the user for each queue. Ultimately, the frame reaches its destination at the mac module, via either the preemptable or express queue.

NeSTiNg has been made accessible to the research community, with the authors' notable contribution lying in an extensive evaluation of the TSN functionalities embedded within this network simulator. The results presented in the paper are predominantly focused on the network simulator itself, encompassing metrics such as average simulation run-time and memory consumption. However, it is worth noting that the paper does not delve into the performance evaluation of TSN-based applications; its scope remains centred on the simulator's performance characteristics.

2.3.4 Worst Case Analysis of Industrial Time Sensitive Networks

In the realm of industrial networks, where complex data traffic specifications co-exist with a plethora of design choices, the utilisation of TSN standards necessitates careful consideration. These considerations span a wide spectrum, encompassing topology selection, synchronisation of network devices, determination of the number of data traffic classes, selection of shaping techniques, and the configuration thereof, among others. As a result, informed and well-structured design decisions become paramount. Given the intricacies of this decision-making process, network calculus has emerged as a valuable tool within the literature [MHG20], providing a robust framework to derive actual worst-case results within TSN-based networks. In the subsequent sections, we provide a succinct overview of noteworthy contributions that leverage Network Calculus (NC) for the analysis of TSN shaping techniques.

In a noteworthy study, researchers in [ZPZL18] applied network calculus to dissect audio/video traffic when influenced by time-triggered traffic. Their analysis considered both pre-emption and non-pre-emption modes, resulting in precise upper bounds for worst-case delays associated with Class A and Class B traffic. Expanding on this foundation, another study detailed in [ZPZ⁺20] encompassed not only two but multiple classes of AVB traffic. Authors in [GZRP18] undertook a comprehensive analysis of credit bounds for AVB traffic, while also scrutinising guard bands and the ramifications of scheduled traffic. Moreover, they introduced tighter latency bounds for AVB traffic by limiting the output of the CBS and accounting for link transmission rates. [ZPC18] delved deeper into time-aware shaping, investigating the influence of GCLs on traffic, albeit limited to non-pre-emption mode. Meanwhile, [MSMLB18] explored asynchronous time shaping within TSN, affecting traffic before encountering other shapers like CBS. This study provided a stringent per-flow bound for AVB traffic under regulation and derived overall end-to-end latency bounds, which proved to be lower than the sum of per-switch delay bounds. Building upon this, the same authors made enhancements [MSLB19b] by incorporating rate-latency service curves, where packet transmissions aligned with physical link rates, making the derived bounds packet size-dependent. In [MSLB19a], they introduced superior upper bounds for AVB classes, applicable across various TSN configurations, facilitating the computation of end-to-end delay and network buffer sizes.

The aforementioned research contributions have collectively contributed to the

analysis of diverse TSN shaping techniques. Building upon this foundation, our work extends the discourse by exploring the impact of the pre-emption mode on credit evolution, particularly within AVB traffic when coexisting with time-critical traffic. Additionally, we investigate the influence of higher-priority traffic—subjected to CBS and TAS—on lower-priority but mission-critical data traffic, spanning network control, events, configuration, and diagnostics. Our contribution includes producing simulation results, allowing us to compare and contrast them with the analytical results derived from Network Calculus. This comparison, encompassing worst-case scenarios and typical stochastic system evaluations, enhances the depth of our knowledge and offers valuable insights into the performance of the proposed system under various conditions. Our work is grounded in the TSN industrial profile, but its adaptability allows for extensions to cater to other TSN profiles and applications.

Chapter 3

Traffic Requirements for Industrial Applications

The advancements in industrial technology have led to increased efficiency, automation, and improved connectivity. Industries such as manufacturing, process automation, transportation, and power generation are currently experiencing notable transformations. In these dynamic environments, the paramount importance of dependable, real-time communication cannot be overstated.

This section explains into the intricacies of TSN-based real-time industrial applications, with a special focus on understanding the vital role of traffic requirements. As industries increasingly adopt automation, robotics, and interconnected systems, the need for precise and deterministic communication intensifies. Thus, grasping the intricacies of traffic demands within industrial applications becomes a linchpin for harnessing TSN's potential to revolutionise modern industries.

Our exploration commences with an examination of the diverse range of industrial applications positioned to gain advantages from Time-Sensitive Networking. This exploration spans sectors encompassing manufacturing, process automation, robotics, power generation, and transportation. We underscore the profound significance of real-time communication in sculpting the future of these industries. In each sector, efficient and dependable communication systems are pivotal, enabling precise coordination, boosting productivity, and upholding unwavering safety standards.

Furthermore, we venture into TSN's pivotal role in meeting the unique traffic requirements of these industrial applications. TSN's strengths in delivering

deterministic communication, ultra-low latency, and synchronised data transmission play a pivotal role in facilitating seamless collaboration across diverse industrial processes and components. Empowered by precise control, synchronisation, and coordination, industries can push the boundaries of innovation, ushering in highly sophisticated systems that redefine the realm of possibilities.

As we continue our exploration, we navigate the classification of traffic within the TSN domain, recognising the distinct attributes and time-critical demands exhibited by various traffic classes. Appreciating this classification is pivotal in tailoring TSN solutions to cater to the specific and varied needs of diverse industrial use cases. We traverse the landscape of bandwidth requirements, latency constraints, and the unyielding demands for reliability, all of which collectively govern the traffic requirements for each application.

Ultimately, this section seeks to illuminate the indispensable role of real-time communication in reshaping modern industries through the lens of TSN. Our comprehensive exploration aims to equip readers with a profound understanding, fostering insights that contribute to the ever-expanding domain of knowledge at the intersection of TSN and industrial applications.

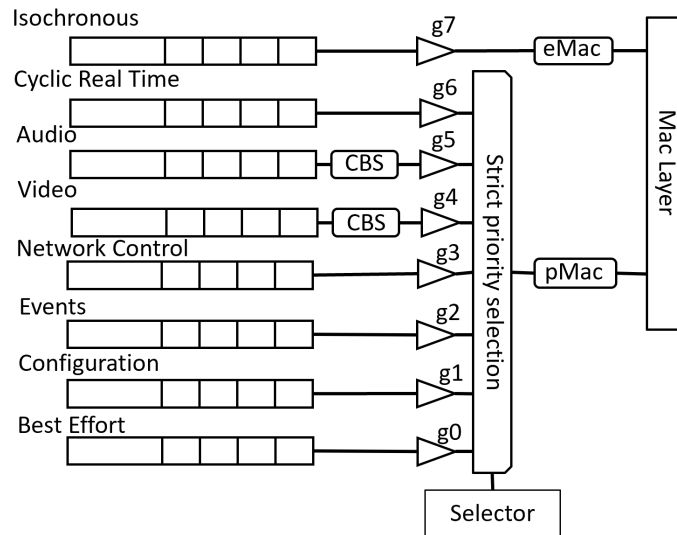


Figure 3.1: TSN switch interface with traffic shaping, frame pre-emption, and strict priority.

3.1 TSN industrial classes of traffic

Ethernet frames, particularly in Ethernet-based fieldbus systems, play a crucial role in facilitating communication among field devices within industrial appli-

cations. These systems exchange data using data frames between field devices. To determine frame priority, various methods are used, including binary classification based on time criticality (Time-critical or Non-critical) and classification into traffic patterns like Time-Triggered traffic, Audio-Video Bridging traffic, and Best Effort traffic. The latter results in four traffic classes, with AVB traffic containing Class A and Class B.

However, the TSN industrial profile expands this classification, defining eight traffic classes: isochronous cyclic real-time, cyclic real-time, network control, audio, video, alarms/events, configuration/diagnostics, and best effort. Network devices, such as TSN switches, process these traffic classes based on assigned priority code point (PCP) per the industrial TSN profile. Time-critical traffic receives the highest priority, followed by AVB traffic, and then best-effort-based traffic.

Figure 3.1 illustrates a TSN switch interface supporting the industrial TSN queuing mechanism and various TSN shaping methods (explained in section 2.1). This interface efficiently manages the diverse traffic classes, adhering to the prioritisation specified by the TSN industrial profile.

By embracing the TSN framework and its comprehensive traffic classification, industrial applications achieve enhanced performance, predictability, and reliability in communication systems. Critical real-time data is transmitted promptly and accurately, contributing to seamless operations and safety. AVB traffic ensures synchronised multimedia data streams, catering to audio and video applications' quality of service. Meanwhile, the best effort classes handle non-critical data reliably.

As Time-Sensitive Networking becomes increasingly adopted in industrial settings, the potential for innovation and efficiency in communication is evident. Engineers and network administrators have the opportunity to design robust, time-critical systems that contribute to productivity, safety standards, and the evolution of industrial automation.

3.1.1 Scheduled Traffic (ST)

The scheduled/Time-triggered traffic in Time-Sensitive Networking (TSN) encompasses two priority classes: isochronous cyclic real-time and non-isochronous cyclic real-time, denoted as Priority Code Point (PCP) 7 and 6, respectively. These classes are of utmost importance among all traffic types

Table 3.1: isochronous and non-isochronous cyclic real-time traffic type properties

Property	Description
Transmission Scheme	Periodic or Event-driven.
Constraints	Necessary data transmission restrictions for proper operation of traffic type, which is a single or combination of the following: • Latency: Bounded or Guaranteed. • Bandwidth • No constraints
Periodicity	Depends on data transmission constraints: • Bounded latency: Data deadline period, • other constraints: Data transmission period,
Synchronisation with network cycle	Defines if sender TSN nodes' data transmission is synced with the active network cycle.
Synchronised to network access	Defines Whether the industrial application using this traffic pattern is synced with the network access.
Jitter tolerance	The amount of jitter that may occur and that the receiving destination(s) must be able to handle is indicated for transmitted data with latency constraints.
Frame loss ratio	Defines the traffic type's tolerance to frame losses, which influence the network design to provide high redundancy.
Payload	Defines payload data type and size to be transmitted: • Fixed: Transmitted data always of the same size. • Bounded: Transmitted data may vary in size, but it should not exceed a given maximum size, e.g. Ethernet frame size of 1500 bytes.

due to their periodic data transmission schemes, requiring almost negligible fail-over time. To gain deeper insights into these traffic classes in industrial use cases, Table 3.1 presents essential properties for identification (see Section 3.3).

To ensure determinism, minimal delay, and ultra-low jitter (approximately in the microsecond range), TSN devices implement shaping methods to support critical scheduled traffic transmissions. For instance, isochronous flows are labelled as *express frames*, allowing them to pre-empt other frames transmitted at the same moment. Additionally, cyclic real-time flows are always preceded by a guard band to prevent correlation with less priority traffic transmissions. This is achieved by defining Time-Aware Shaping (TAS) schedules to operate the Gates Control List of the TSN interfaces.

Overall, the prioritisation and shaping of scheduled/Time-triggered traffic classes in TSN facilitate seamless and precise communication for time-critical industrial applications. These mechanisms enable efficient transmission of periodic data with minimal disruptions, thus empowering industries to achieve unprecedented levels of determinism and reliability. By strategically implementing shaping methods and strict scheduling, TSN networks ensure the timely delivery of crucial data, advancing the capabilities of modern industrial systems.

3.1.2 Audio-Video Bridging Traffic

In the traditional model of analogue audio video (AV) equipment, audio and video streams were transmitted over one-way, single-purpose, point-to-point connections. This approach posed significant challenges, particularly in professional applications, where the abundance of wiring was cumbersome. Additionally, AVB traffic, which carries audio and video streams, demanded high synchronisation precision and low transmission latency for each stream to prevent issues like dropped video frames or unwanted audio glitches such as pops or silences. Synchronisation between video and related audio streams was essential to maintain seamless playback.

In professional setups, preserving phase coherence among multiple digitally connected speakers was equally vital. To address these challenges and enhance AVB performance, the TSN profile defined distinct classes for AVB traffic. Class A traffic caters to audio streams, while class B traffic is dedicated to video streams. To ensure precise timing and latency guarantees, TSN devices assign

high priority to AVB frames, setting the PCP to 5 and 4.

Moreover, a leaky bucket-based credit shaping method is employed to control AVB traffic bursts effectively. By doing so, the quality of service requirements for both lower priority flows and AVB flows are meticulously met.

Through an examination of TSN's sophisticated traffic management mechanisms, our research explores the potential for enhanced reliability and precision in AVB traffic within industrial settings. Seamless synchronisation between audio and video streams, coupled with efficient handling of traffic bursts, results in an elevated audiovisual experience. These advancements empower professional applications with streamlined wiring, seamless playback, and enhanced performance, revolutionising the realm of audio-video communication in industrial contexts.

3.1.3 Best-Effort Traffic

IT traffic in TSN encompasses flows that can be carried over standard IT networks. While belonging to non-critical classes of traffic, they still demand specific services to support various functionalities. For instance, network control traffic is vital for proper network management, necessitating prioritisation. Likewise, diagnostics, configurations, events, and alarms classes of traffic require high reliability to facilitate resource monitoring and management, as detailed in Section 3.3.4.

Furthermore, web traffic, such as browsing and file transfers, falls under the category of best effort traffic. Although non-critical, these flows are assigned the lowest priority compared to the aforementioned traffic classes.

To optimise and manage the transmission of flows, TSN devices employ a combination of shaping methods. For instance, they utilise Time-Aware Scheduler gates to allocate slots for IT traffic transmission while holding AVB traffic. Moreover, AVB queues accumulate credit only during the transmission of best effort frames when TAS gates are open. This strategic approach ensures the coexistence of both IT and Operational Technology traffic, enabling each traffic class to meet its specific timing and throughput requirements. The Time-Aware Scheduler gates play a crucial role in orchestrating the synchronised transmission of IT and OT traffic, ensuring efficient and reliable communication across the network.

Through effective shaping and prioritisation, TSN enables seamless integration

of IT traffic within industrial settings, bolstering network performance, and delivering reliable communication. By providing tailored services for each traffic class, TSN facilitates the simultaneous transmission of critical Operational Technology data and less critical Information Technology data, transforming the landscape of industrial communication and network management.

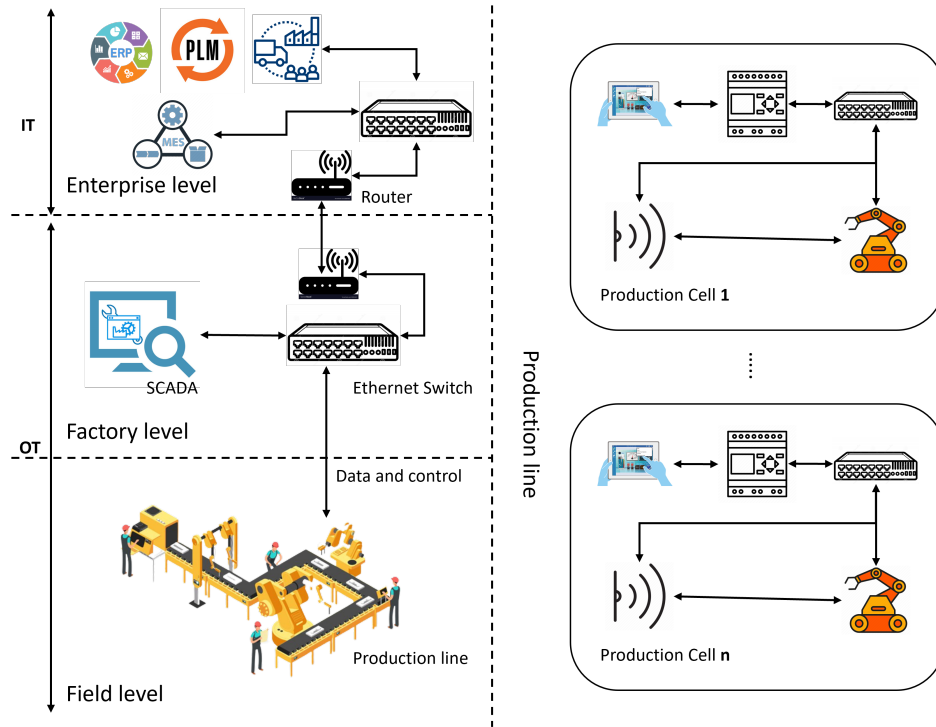


Figure 3.2: Smart manufacturing vertical model

3.2 Essential aspects of traffic requirements for industrial applications

In Time-Sensitive Networking, addressing the unique traffic requirements of industrial applications is pivotal for achieving seamless and efficient communication. This section presents into the essential aspects of traffic requirements for industrial applications, focusing on deterministic communication, bandwidth requirements, latency and jitter constraints, redundancy and reliability, and Quality of Service considerations. The discussion encompasses the traffic classes of time-triggered, audio-video bridging, and information technology, exploring how each class demands specific handling to meet the stringent demands of modern industrial processes.

3.2.1 Deterministic Communication for Industrial Applications:

In the ever-evolving landscape of industrial automation, the need for deterministic communication has emerged as a pivotal requirement. Industrial settings demand precise and timely data transmission, particularly for real-time control and seamless coordination. Deterministic communication ensures the delivery of messages within well-defined timeframes, reducing uncertainties and facilitating synchronised operations among interconnected components.

Time-Sensitive Networking (TSN) is a solution specifically engineered to fulfil the demands of deterministic communication in industrial settings. Within TSN's traffic classes, the time-triggered traffic class stands out as a robust mechanism for meeting the stringent requirements of real-time control and coordination. Through the provision of fixed and predictable transmission schedules, the time-triggered traffic class enables industrial systems to execute precise control loops and achieve flawless synchronisation among interconnected devices.

The time-triggered traffic class operates within the TSN network on a meticulously choreographed schedule. This deterministic approach ensures that data is sent and received at precisely defined intervals, introducing a level of reliability and predictability crucial for safety-critical applications. For instance, in the realm of industrial robotics, where split-second decisions carry significant consequences, deterministic communication provided by the time-triggered traffic class guarantees the execution of control commands with absolute precision, mitigating potential risks and optimising overall productivity.

Beyond its attributes of reliability, predictability, and safety, the time-triggered traffic class also facilitates the creation of a time-aware network infrastructure. Adherence to fixed schedules enables efficient resource allocation within the network, minimising contention and ensuring that critical messages reach their destinations punctually. This reduction in the likelihood of collisions and data loss, coupled with the assurance of prioritised and expedited message delivery, enhances system responsiveness and reliability.

In dynamic industrial environments, where multiple components must function in tight coordination, the synchronisation of data transmission becomes a pivotal factor for successful automation. The time-triggered traffic class enables this synchronisation by aligning the activities of diverse devices to a common time reference. Messages are delivered in accordance with a synchronised

schedule, enabling coordinated actions, preventing data drift, and preserving temporal integrity across the network.

Moreover, the time-triggered traffic class paves the way for the development of distributed control systems capable of autonomous operation with minimal central coordination. This decentralised approach enhances fault tolerance, reduces reliance on centralised control units, and bolsters system scalability and resilience.

In conclusion, deterministic communication, exemplified by the time-triggered traffic class in TSN, assumes a central role in enabling efficient and secure industrial automation. With its fixed transmission schedules, synchronisation capabilities, and unwavering reliability, this traffic class empowers industries to embrace automation and interconnectedness, propelling them toward unparalleled levels of efficiency and productivity in the modern industrial landscape.

3.2.2 Bandwidth Requirements for Industrial Applications

Industrial applications are characterised by a wide array of data types, each with its distinct bandwidth requirements. Time-triggered traffic, as a critical component of real-time control, demands deterministic bandwidth allocation to ensure timely and reliable transmission of crucial data. On the other hand, audio-video bridging traffic necessitates sufficient bandwidth to support uninterrupted transmission of high-quality audio and video streams, essential for seamless multimedia applications in industrial settings.

Conversely, information technology traffic, encompassing standard Ethernet data, may have relatively lower bandwidth demands compared to real-time traffic classes. However, even though IT traffic is non-critical, it still requires efficient utilisation of available bandwidth to coexist harmoniously with real-time traffic classes. Efficient bandwidth management becomes indispensable to strike a balance between different traffic types and guarantee smooth communication across the network.

In this multifaceted landscape, TSN's advanced capabilities come to the forefront, offering the means to prioritise and allocate bandwidth according to the specific demands of each traffic class. By intelligently managing bandwidth resources, TSN enables industrial applications to optimise data transmission and achieve seamless coordination between real-time and non-real-time data, fostering a harmonious and efficient communication ecosystem.

3.2.3 Latency and Jitter Constraints in Industrial Applications

In industrial applications, where real-time responsiveness stands as a paramount requirement, the significance of low latency and minimal jitter cannot be overstated. These factors are pivotal in defining the effectiveness and dependability of communication systems. Industrial processes often necessitate swift and consistent data delivery to attain precise coordination and synchronisation among interconnected devices and components. As such, the ability to ensure minimal delays and fluctuations in data transmission assumes a central role in facilitating seamless operations across diverse industrial domains.

Within Time-Sensitive Networking (TSN), the time-triggered traffic class exemplifies the integration of low latency and minimal jitter into the fabric of industrial communication. By adhering to a meticulously defined and predictable transmission schedule, the time-triggered traffic class mitigates uncertainties associated with data delivery, guaranteeing that critical messages reach their destinations with utmost precision and reliability. This deterministic quality plays a vital role in enabling precise control loops and tightly synchronised actions, rendering it especially well-suited for safety-critical applications within industrial automation.

In parallel, audio-video bridging (AVB) traffic, which caters to high-quality audio and video streams, underscores the importance of low latency to maintain seamless synchronisation between audio and video components. In applications like video surveillance, manufacturing line monitoring, or collaborative robotics, where real-time multimedia streams are integral, low latency becomes indispensable in averting issues such as audio-video drift or disruption, which could jeopardise data analysis and decision-making integrity.

Information technology traffic, encompassing standard Ethernet data, may not impose the same stringent latency requirements as real-time traffic classes. Nevertheless, even within the context of IT traffic, timely data delivery remains critical for efficient information exchange and network responsiveness. For instance, IT traffic may encompass data pertinent to remote configuration, diagnostics, and event reporting, where prompt updates are vital for overall system health and expediting decision-making.

As industries progressively embrace interconnected systems and rely on data-driven decision-making, effective management of latency and jitter emerges

as a focal point in the design of robust industrial communication networks. TSN, equipped with inherent capabilities for low-latency communication and advanced traffic management, provides an ideal framework to address these demands.

Through the implementation of mechanisms for traffic prioritisation, scheduling, and bandwidth allocation, TSN ensures that time-critical messages are accorded preferential treatment within the network. The management of data transmission empowers industrial applications to optimise communication, striking a delicate balance between real-time and non-real-time data streams. Such orchestration is indispensable in forestalling contention and data collisions, thereby reducing jitter and latency spikes, while simultaneously ensuring harmonious coexistence among various traffic classes.

In conclusion, the significance of low latency and minimal jitter in industrial applications cannot be overstated. From precise control loops and synchronised multimedia streams to efficient information exchange, these elements underpin the efficiency, safety, and reliability of industrial communication systems. TSN's capabilities in latency and jitter management, coupled with its support for diverse traffic classes, position it as a pivotal technology in facilitating seamless, real-time communication across multifarious industrial domains.

3.2.4 Quality of Service (QoS) Considerations

In industrial communication, the ability to distinguish between various Quality of Service levels is of utmost significance. This capability enables critical traffic classes to be accorded higher priority over less time-sensitive data, ensuring that essential communication requirements are met. Time-Sensitive Networking offers a versatile array of QoS mechanisms, empowering industrial applications to customise their communication infrastructure to cater to the specific needs of different traffic classes. By adeptly managing QoS, TSN ensures that the diverse requirements of real-time and non-real-time traffic are addressed, ultimately optimising overall network performance and facilitating the seamless coexistence of various data streams.

A pivotal aspect of QoS management in industrial applications lies in the prioritisation of traffic classes. Time-triggered traffic, serving real-time control and coordination, is frequently allocated the highest priority. By affording time-triggered traffic this elevated status, TSN guarantees the swift and dependable

delivery of critical messages associated with precise control loops and synchronisation. This prioritisation assumes heightened importance in safety-critical applications, where mere milliseconds can significantly influence the prevention of potential hazards or disruptions.

Conversely, audio-video bridging traffic, designed to support high-quality audio and video streams, is typically assigned a lower but deterministic priority. AVB traffic necessitates stable and predictable transmission times to maintain synchronisation between audio and video components. This deterministic prioritisation ensures that AVB streams experience consistent and dependable transmission, effectively mitigating audio-video drift and disruptions that could detrimentally affect industrial multimedia applications.

Meanwhile, IT traffic can be designated as best-effort priority, facilitating harmonious coexistence with real-time traffic classes. While IT traffic may not demand the stringent guarantees of real-time data, it nonetheless benefits from the efficient utilisation of available network resources, ensuring that non-critical data exchanges are conducted with reasonable efficiency.

TSN's adaptable QoS management empowers industrial applications to fine-tune their communication networks in accordance with specific requirements. The ability to differentiate QoS guarantees that real-time control and time-sensitive data take precedence, thereby preserving system stability and safety, while non-real-time traffic harmoniously coexists without hindering critical application performance.

Furthermore, TSN's QoS mechanisms encompass more than mere prioritisation. TSN can provide guaranteed bandwidth allocation for particular traffic classes, assuring that time-sensitive applications have the requisite resources to meet their demands. Additionally, TSN supports shaping and policing mechanisms to govern and enforce bandwidth constraints, averting excessive data traffic from monopolising network resources and ensuring equitable and efficient data distribution across the network.

In summation, the capability to differentiate QoS levels in industrial communication holds significant importance in enabling efficient and dependable data exchange. TSN's comprehensive suite of QoS mechanisms empowers industrial applications to tailor their network infrastructure to accommodate the diverse requirements of time-triggered, AVB, and IT traffic classes. By offering prioritisation, bandwidth allocation, shaping, and policing capabilities, TSN empow-

ers industries to devise communication networks that maximise performance, safety, and efficiency, while concurrently ensuring the seamless coexistence of real-time and non-real-time data streams.

3.3 Industrial use cases in smart manufacturing

Smart manufacturing and industrial automation have paved the way for the seamless coexistence of both Informational Technology (IT) and Operational Technology (OT). The OT traffic is divided into two levels of abstraction: the field level, where multiple production lines operate with distinct objectives to fulfil various assigned tasks, and the factory level, which captures real-time OT data to enable automation of industrial processes running at the field level. Figure 3.2 illustrates a vertical model of a smart factory, where the OT is covered by these two levels.

In contrast, IT is abstracted as an enterprise and management level, where enterprise resource planning takes place to manage real-time business processes. This system can be modelled as a set of applications hosted over numerous yet single-interface devices, each featuring a single MAC address and IP address for every end system (ES). For instance, consider an industrial automation model built on top of a TSN-based network designed to monitor, configure, and control field devices. This model incorporates a multiservice control layer with a mainstream process comprising diverse technologies and equipment, including SCADA, PLC, and real-time devices. The communication between these devices generates the aforementioned traffic types.

In general, industrial applications encompass various discrete and diverse production processes. To gain a comprehensive understanding of these applications, it is helpful to classify them into broader use cases, such as Control loops, Video surveillance, Sequence of events, and Network monitoring and diagnostics. This classification aids in discerning the distinct aspects of each industrial application, facilitating optimised design and operation.

With the integration of IT and OT in smart manufacturing, industries can leverage real-time data and intelligent decision-making to enhance productivity and efficiency. TSN-based networks provide the necessary determinism, low latency, and reliability required for critical industrial communication, fostering a robust ecosystem for the seamless coexistence of IT and OT traffic.

3.3.1 Control Loops

Control loops represent a critical use case in the realm of industrial automation, where precision and efficiency are paramount. They serve as the backbone of process control in industrial automation, aiming to maintain essential process variables within predetermined ranges. These variables can encompass temperature, pressure, flow rate, level, and more, depending on the specific application. The fundamental principle behind control loops revolves around achieving and sustaining a desired set point by continuously adjusting the process variables.

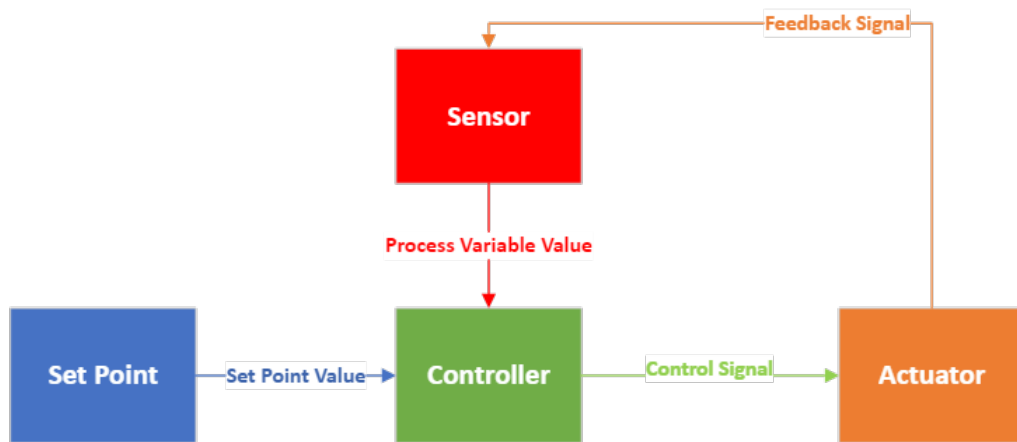


Figure 3.3: Basic Control Loop

A control loop, depicted in Figure 3.3, is composed of several key components that work in concert to ensure precise regulation:

- **Process Variable (PV):** The process variable is the real-time measurement of the parameter being controlled, obtained through sensors. It serves as the feedback to the control loop.
- **Setpoint (SP):** The setpoint is the target value for the process variable, which is predetermined based on the operational requirements. The control loop endeavours to maintain the process variable close to this set point.
- **Controller:** At the heart of the control loop, the controller compares the actual process variable (PV) with the desired setpoint (SP) and computes the error signal, which is the difference between the two.
- **Actuator:** The actuator is responsible for effecting changes in the process to bring the process variable closer to the setpoint. It can be a valve, motor, or other devices capable of manipulating the process.

- **Feedback:** The controller utilises the feedback provided by the process variable (PV) to iteratively adjust the actuator. This continuous feedback loop allows the control system to maintain the process variable at the desired set point accurately.

Control loops play a pivotal role in various industries, including manufacturing, chemical processing, energy generation, and more. By ensuring stable and precise control of process variables, control loops optimise production processes, enhance product quality, and contribute to the overall safety and efficiency of industrial operations. In the context of Time-Sensitive Networking (TSN), control loops can leverage the deterministic and low-latency communication capabilities of TSN to enhance their performance further. Reliable and precise communication between sensors, controllers, and actuators is paramount for rapid and accurate control responses. By integrating TSN into control loops, industries can realise faster and more predictable control actions, resulting in improved process stability and responsiveness. In conclusion, control loops represent a foundational use case in industrial automation, facilitating the precise regulation of process variables. Leveraging the capabilities of TSN, control loops can achieve unprecedented levels of determinism and responsiveness, unlocking new dimensions of efficiency and productivity in the industrial landscape.

3.3.2 Video surveillance

Video surveillance is a pivotal use case in the domain of industrial automation, e.g. Figure 3.4, offering a comprehensive solution for enhancing safety, security, and operational efficiency within industrial facilities. It entails the strategic deployment of video cameras and monitoring systems across various areas within an industrial facility. It involves capturing and analysing video footage to enable real-time monitoring, incident investigation, process optimisation, and more. The utilisation of video surveillance in industrial automation is driven by the desire to safeguard assets, ensure safety compliance, and streamline operations.

Video surveillance plays a significant role in many industrial applications, such as:

- **Security and Safety Monitoring:** Video surveillance systems act as vigilant sentinels, closely monitoring critical areas, access points, and vulnerable

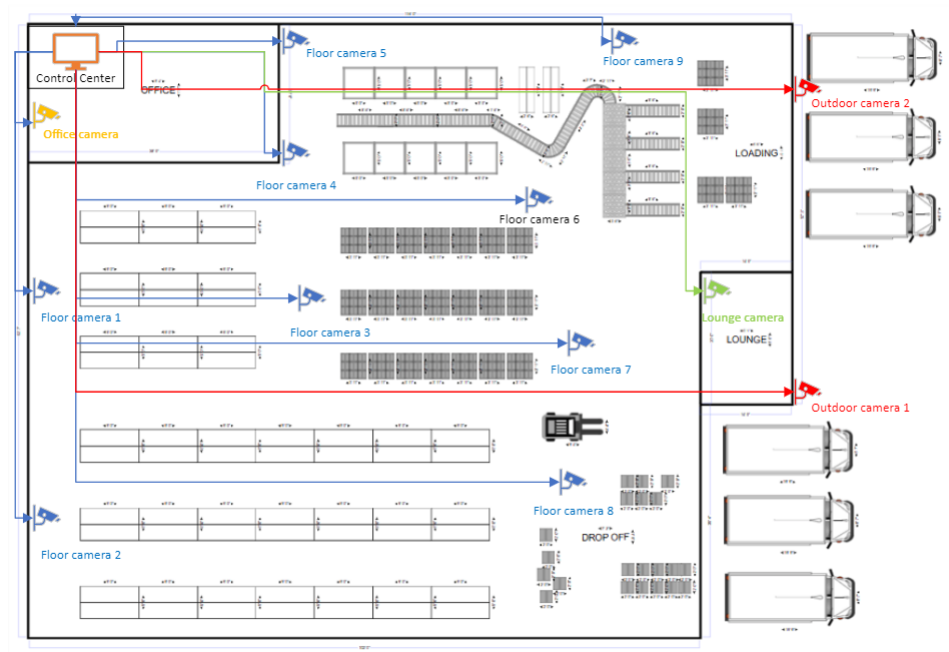


Figure 3.4: Plant floor video surveillance use case

equipment. By providing real-time surveillance, they deter unauthorised access, prevent theft, vandalism, and other security breaches. Moreover, video surveillance contributes to enhancing workplace safety by identifying potential hazards and promptly responding to safety incidents.

- **Process Monitoring and Control:** Industrial processes and equipment are continuously monitored through strategically positioned video cameras. This real-time observation allows operators and engineers to ensure smooth and efficient operations. Any anomalies or deviations from standard procedures can be quickly identified and addressed.
- **Incident Investigation:** In the unfortunate event of accidents, incidents, or process deviations, video surveillance footage serves as invaluable evidence for conducting thorough investigations. The recorded video aids in determining the root cause of the issue, enabling the implementation of corrective actions and preventing future occurrences.
- **Remote Monitoring and Maintenance:** Video surveillance systems enable remote monitoring of industrial sites, reducing the need for physical presence on-site. Operators and maintenance personnel can remotely observe equipment and processes, leading to faster response times and facilitating predictive maintenance practices.

- **Environmental Monitoring:** Beyond security and process monitoring, video surveillance can extend to environmental monitoring. Cameras can be used to track environmental conditions in sensitive areas, such as temperature, humidity, and air quality, ensuring adherence to environmental regulations and protecting equipment integrity.
- **Real-Time Alerts and Notifications:** Video surveillance systems are increasingly equipped with advanced analytics and artificial intelligence algorithms. These capabilities enable real-time alerts and notifications for specific events or anomalies, such as unauthorised access or abnormal activities, enabling swift and targeted responses.
- **Integration with Other Systems:** Video surveillance is often integrated with other industrial automation systems, such as SCADA and access control. This integration creates a synergistic security and monitoring infrastructure, reinforcing safety protocols and optimising resource allocation.

In conclusion, video surveillance is a powerful tool that goes beyond traditional security applications, becoming an indispensable asset in modern industrial automation. By fortifying safety, streamlining processes, and providing real-time intelligence, video surveillance helps industries navigate the challenges of an increasingly complex and interconnected industrial landscape.

3.3.3 Sequence of Events

In industrial automation, the Sequence of Events (SOE) use case plays a pivotal role in ensuring precise event logging, timestamping, and sequential recording of occurrences within industrial processes or equipment. This functionality is crucial for real-time monitoring, troubleshooting, and post-event analysis, offering valuable insights into the sequence and timing of critical events. Industrial applications often involve a rapid and simultaneous occurrence of various events, such as alarms, status changes, process events, and equipment trips. The SOE functionality adeptly captures and records these events in the order of their occurrence, providing accurate timestamps for each event. This chronological record is invaluable for comprehending process dynamics, system behaviour, and identifying potential issues during normal operation or abnormal situations.

The SOE use case is particularly significant for:

- **Post-Event Analysis:** After an event or abnormal situation, the SOE func-

tionality allows operators and engineers to perform detailed post-event analysis. The precise timestamps facilitate event correlation, root cause analysis, and the identification of contributing factors.

- **Forensic Investigations:** In case of incidents or accidents, the SOE data becomes a valuable resource for forensic investigations. The sequential record with accurate timestamps helps reconstruct the events leading to the incident and aids in determining the contributing factors.
- **Process Optimisation:** Analysing the SOE data allows operators to identify patterns and inefficiencies in the industrial process. By understanding the sequence of events during normal operation, opportunities for process optimisation and performance improvements can be identified.
- **Safety and Reliability:** For safety-critical systems, accurate event logging and timestamping are crucial. The SOE functionality enhances the safety and reliability of industrial processes by enabling prompt response to critical events and facilitating preventive maintenance.

The SOE use case imposes specific traffic requirements on the communication network to ensure accurate and reliable event logging and timestamping. These requirements include:

- **Deterministic Communication:** To achieve precise and sequential event recording, the communication network must provide deterministic communication with low latency and minimal jitter. This ensures that event data is transmitted promptly and reliably, preserving the correct sequence of events.
- **High Reliability:** The communication network should exhibit high reliability to prevent data loss during event recording. Redundancy mechanisms, such as Parallel Redundancy Protocol (PRP) or High-availability Seamless Redundancy (HSR), can be employed to maintain continuous operation and data integrity.
- **Synchronisation:** Accurate event timestamping requires time synchronisation across all devices in the industrial automation system. The communication network should support IEEE 1588 Precision Time Protocol (PTP) or similar time synchronisation mechanisms to achieve precise time alignment.
- **Bandwidth Management:** The SOE use case may generate bursts of event

data during critical situations. Effective bandwidth management and shaping mechanisms are essential to prioritise SOE traffic and prevent congestion, ensuring timely delivery of event information.

The Sequence of Events use case in industrial automation is a vital functionality that ensures accurate event recording and timestamping. By providing a chronological record of events with precise timestamps, SOE facilitates post-event analysis, forensic investigations, process optimisation, and enhances the safety and reliability of industrial processes. Meeting the traffic requirements, including deterministic communication, high reliability, synchronisation, and effective bandwidth management, is critical to ensure the success and effectiveness of the SOE use case in industrial automation systems.

3.3.4 Network monitoring and Diagnostics

In industrial automation, the Network Monitoring and Diagnostics use case plays a critical role in ensuring the optimal performance and reliability of communication networks. As the backbone of interconnected industrial systems, the network serves as the lifeline that enables seamless data exchange, control, and coordination among various devices and components. To maintain uninterrupted and efficient industrial processes, it is essential to monitor and diagnose the network continuously. Network monitoring involves real-time surveillance and data collection of key network performance parameters. Monitoring tools and software are deployed to track critical metrics such as network traffic, bandwidth utilisation, latency, jitter, packet loss, and error rates. This continuous monitoring provides operators and engineers with invaluable insights into the health and efficiency of the industrial network. These insights enable proactive measures to be taken, preventing potential issues and ensuring the network operates optimally to support critical industrial operations.

By monitoring network parameters, industrial operators can:

- Detect abnormal patterns or deviations in network traffic, which could indicate potential network congestion or performance issues.
- Identify bandwidth bottlenecks that may impact real-time data transmission or critical control processes.
- Monitor latency and jitter to ensure timely delivery of time-sensitive data and synchronisation across interconnected devices.

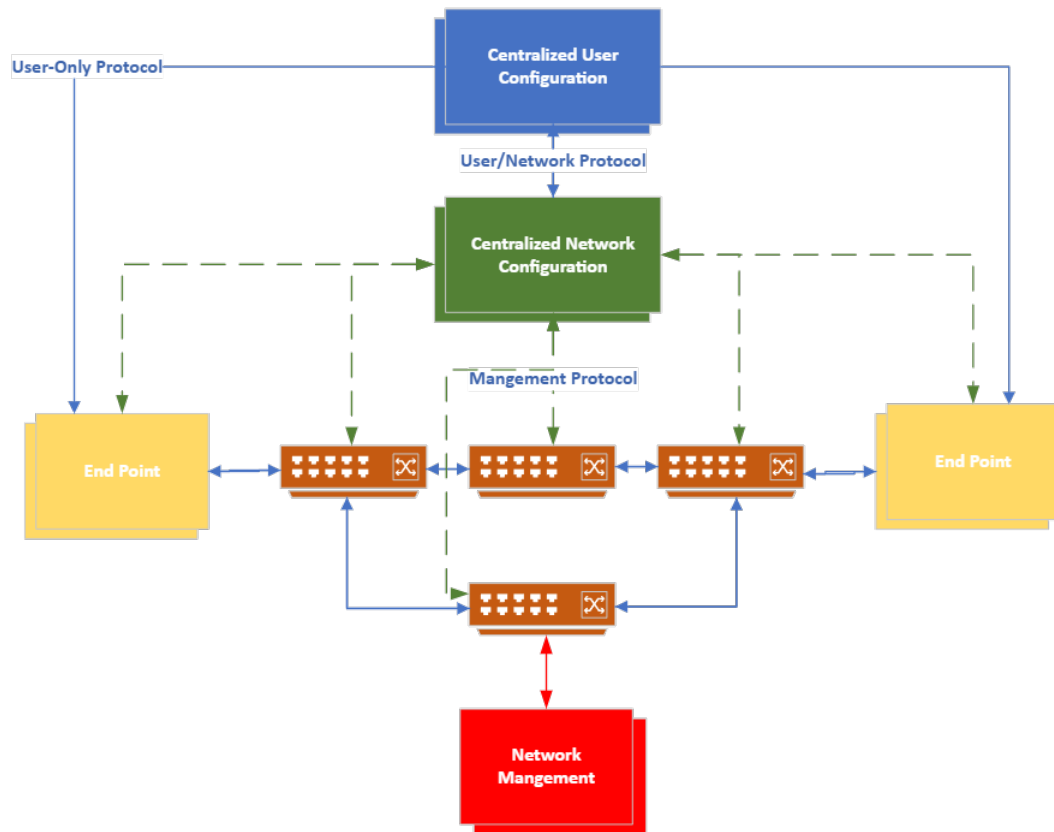


Figure 3.5: Network monitoring and Diagnostics use case

Network diagnostics complement network monitoring by investigating and resolving network issues or anomalies. When monitoring tools detect deviations or abnormalities, diagnostics tools and techniques are employed to identify the root causes of problems and isolate faulty components. The key aspects of network diagnostics in industrial automation include:

- Proactively identifying potential network failures or weak points and implementing corrective measures to prevent downtime.
- Troubleshooting and resolving network-related problems, such as hardware malfunctions, misconfigurations, or security breaches, in a timely manner.
- Performing post-event analysis to understand the root cause of network disruptions and applying preventive strategies to avoid recurrence.

The Network Monitoring and Diagnostics use case impose specific traffic requirements on the industrial communication network to ensure effective data collection and analysis. These requirements include:

Real-Time Data Collection: Network monitoring tools rely on real-time data

to provide up-to-date network performance insights. The communication network should offer low-latency and deterministic data delivery to ensure timely transmission of network metrics.

Bandwidth Management: Network monitoring generates additional traffic for data collection and analysis. To prioritise monitoring traffic and prevent interference with critical control and real-time data, effective bandwidth management mechanisms should be in place.

Network Security: As monitoring and diagnostics tools access sensitive network data, ensuring the security of these tools and their data transmissions is crucial. Robust security measures, such as authentication, encryption, and access controls, must be implemented to safeguard network monitoring and diagnostics.

The Network Monitoring and Diagnostics use case in industrial automation is pivotal for ensuring the smooth and reliable operation of communication networks. By continuously monitoring network performance and promptly diagnosing issues, industrial operators and engineers can optimise network efficiency, minimise downtime, and maintain the highest level of reliability in industrial processes. Meeting the traffic requirements, including real-time data collection, effective bandwidth management, and robust network security, is fundamental to enable successful network monitoring and diagnostics in industrial automation systems.

3.4 Quality Control and Classification After Production

In smart manufacturing, the assurance of product quality after the production process is of paramount importance. Our innovative industrial application, meticulously tailored for quality control and classification after production, strategically employs cutting-edge methodologies. A cornerstone of this approach lies in the deployment of high-definition cameras, diligently capturing visual data of inspected products. This dynamic solution seamlessly integrates four pivotal use cases—control loops, video surveillance, sequence of events, and network monitoring and diagnostics—into an intricately layered architecture, as depicted in Figure 3.6.

The architecture finds its strength in three interconnected layers:

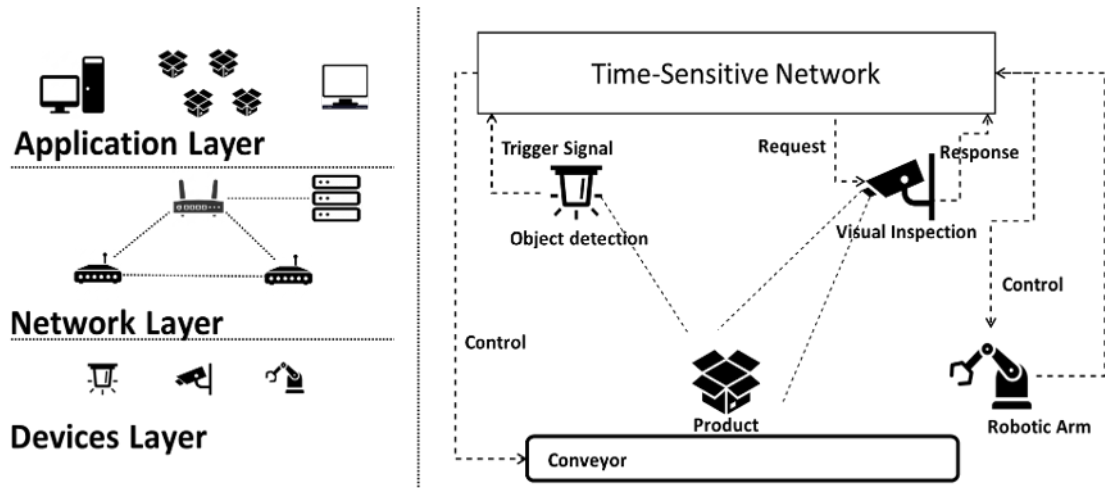


Figure 3.6: Quality Control and Classification After Production (QCAP)

Devices Layer: This foundational layer houses a multitude of devices, sensors, and actuators, working in unison to gather data and enact actions guided by the configurations from the application layer.

Network/Communication Layer: Encompassing a suite of essential network devices—switches, routers, and management servers—this layer forms the bedrock for uninterrupted communication, connecting the device layer with the application layer.

Application Layer: Serving as the nerve centre of the framework, this layer encompasses diverse components such as product patches, monitoring applications, and system configuration tools. It serves as the orchestrator for executing quality checks across multiple production lines.

The Key Quality Control Operations are:

- **Object Detection:** Anchored by control loops, each production line integrates an object detection unit. This unit efficiently triggers quality checks upon the arrival of new products. The timeliness and precision of these trigger messages are of paramount importance, necessitating minimal latency.
- **Object Inspection:** Incorporating video surveillance capabilities, the visual inspection unit meticulously monitors products by capturing images or video clips. This data flow's inherently bursty nature may impact both available bandwidth and overall quality of service.
- **Object Classification:** Post-inspection data analysis prompts a control

signal, effectively guiding a robotic arm's classification decision for each product. This operation requires strict determinism to synchronise robotic arm actions seamlessly with the conveyor belt's movement.

Additionally, Within the sequence of events use case, QCAP application meticulously records and analyses the chronological flow of production activities. This detailed production timeline serves as a valuable resource for post-production analysis, facilitating root cause investigation and continuous improvement strategies. Moreover, QCAP establishes a robust network infrastructure. Network devices like switches, routers, and management servers operate cohesively to ensure the seamless communication between device layer components and the application layer. Any deviations from expected norms trigger alerts, allowing swift interventions to maintain consistent quality output.

QCAP yields distinct data flows, each carrying its own unique attributes:

- **Control Data Traffic (CDT):** Elevated in priority and endowed with stringent low-latency requirements, this critical data traffic undergoes meticulous processing. It is uniquely tagged with the highest priority code point (PCP).
- **Class A Traffic:** A product of our inspection units, this high-priority traffic serves as the conduit for vital data pertaining to inspected products. It demands exceptional reliability, guaranteed upper bounds on end-to-end delay, and minimal jitter to facilitate real-time classification outcomes.
- **Class B Traffic:** Leveraging video surveillance capabilities, this traffic efficiently monitors the production floor through streamed video data. Resource allocation becomes paramount, particularly given the high-resolution nature of the video frames and the application of MJPEG compression.
- **Best Effort Traffic:** Exemplifying efficient network resource utilisation, web traffic, encompassing activities such as HTTP 1.0-based web browsing and file transfers, engages in a dynamic exchange. This traffic's randomised request intervals and response sizes introduce adaptability to the network load.

QCAP as an industrial application function as a test case for simulations and evaluations. Within our ongoing endeavours, this application emerges as a pivotal cornerstone for rigorous simulations, meticulously evaluating the intricate

interplay of data flows, priorities, and network dynamics. By emulating real-world scenarios, we meticulously refine our network management strategies, enhancing their efficiency and reliability. This provides a robust environment for validating the practicality and effectiveness of our TSN network management innovations. In essence, this industrial application—dedicated to quality control and classification after production—unites advanced camera technology with a multi-layered architectural model, orchestrating seamless data traffic management. Through an analytical lens, we discern operations and categorise data flows, culminating in the precision, efficiency, and reliability synonymous with contemporary industrial manufacturing. Notably, this comprehensive approach not only ensures product excellence but also accelerates the advancement of network simulations and evaluations. This unique amalgamation encapsulates the synergy between practical application and theoretical testing.

3.5 Collaborative Robot based Industrial Automation

A compelling application that can leverage the integration of Wi-Fi 802.11 with TSN involves a time-critical industrial automation scenario where collaborative robots collaborate, as illustrated in Figure 3.7. In this specific use case, TSN serves as the linchpin for achieving precise synchronisation and deterministic communication between the robots and their control systems, while Wi-Fi adds the invaluable element of wireless connectivity. The data traffic models within this application encompass:

Control Data Traffic: This category encompasses the vital exchange of commands and feedback signals between the central controller and the robotic units. TSN brings forth its deterministic prowess to ensure that control messages traverse the Wi-Fi network with exacting precision, guaranteeing low-latency and highly reliable delivery of these time-critical instructions and responses. To maintain the exact coordination required between the robots and their control systems, this traffic model necessitates the implementation of rigorous scheduling and synchronisation mechanisms.

Sensor Data Traffic: Within this context, collaborative robots may be equipped with an array of sensors, including cameras or force sensors, generating data that must be promptly transmitted to the central controller for real-time analy-

sis and decision-making. Wi-Fi, empowered by TSN, steps in to facilitate the deterministic transmission of sensor data, thereby enabling synchronised and punctual delivery of this vital sensor information. Quality of Service mechanisms come into play to prioritise sensor data traffic, minimising latency to cater to the demands of time-sensitive data processing.

Safety Traffic: In the domain of industrial automation, safety reigns supreme. Signals pertaining to safety, such as emergency stop commands or safety status updates, demand the highest priority and utmost reliability. With TSN-driven Wi-Fi, the system ensures the timely and deterministic conveyance of safety traffic, leaving no room for compromise. This guarantees that critical safety directives are communicated instantly, and safety-related information is perpetually refreshed and exchanged.

Non-Time-Critical Traffic: Besides the time-sensitive data, there may be non-time-critical traffic in the wireless network, such as software updates, diagnostics, or non-real-time monitoring. These data streams can utilise the remaining available bandwidth in the Wi-Fi network without affecting the determinism and real-time performance of the time-sensitive traffic. QoS mechanisms can be applied to allocate bandwidth and prioritise different traffic types based on their criticality.

Video Surveillance and Network Monitoring traffic: Collaborative robots assume the role of astute observers, harnessing cameras and sensors to glean real-time insights from their surroundings. This dynamic stream of data, imperative for instantaneous decision-making, traverses the network with the precision of TSN and the adaptability of Wi-Fi. In parallel, Network Monitoring assumes the role of a vigilant guardian, tirelessly identifying anomalies and irregularities within the intricate digital landscape. TSN takes charge to ensure that monitoring data is transmitted promptly and deterministically, reinforcing the ecosystem's resilience.

To realise seamless communication among collaborative robots within a wireless network, while upholding meticulous synchronisation, determinism, and real-time communication, we have outlined the general application requirements in Table 3.2. These requisites ensure that time-critical data traffic, including control commands, sensor data, and safety signals, is consistently delivered with unwavering reliability and minimal latency.

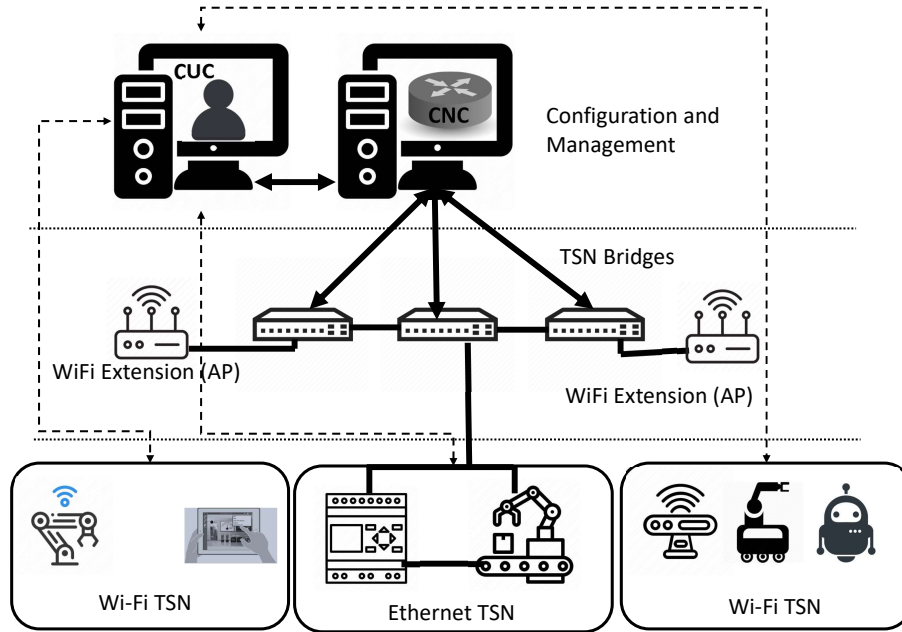


Figure 3.7: Collaborative Robot based Industrial Automation

Table 3.2: Collaborative Robots based Industrial Automation Application Requirements

Requirement	Description/value
Traffic Profiles	Isochronous flows Asynchronous messages Audio/Video Best effort
Throughput	Low (isoc./async.) Low to High (video)
Latency	100 - 0.1 ms (isoc.) 100 - 10 ms (async and video)
Reliability	99.99999 %
Mobility	Low mobility (mostly indoor)
Power	Typically, mains powered for high availability

Chapter 4

Timing Analysis of TSN Based Industrial Networks using Network Calculus

Meeting the traffic requirements, detailed in the previous chapter, is a key measure on the performance of the supporting network. For example, in industrial applications, the supporting network has to provide deterministic and reliable communication. Thus, it ensures successful operation of the application processes.

There exist few methods to verify a network performance:

1. using actual network deployment, then measure the performance parameters in a real operational site. However, this becomes inefficient and not cost-effective solution without prior planning or testing.
2. using emulators or simulators to assess a modelled version of the network and therefore drive a good understanding of the network design and operation. However, in some cases, no implementation of the target network functionality exists in the available tools.
3. using theoretical frameworks, like Network Calculus, to derive a set of equation to derive the bounds on a network functionality. Then, build a framework implementation to evaluate the worst-case performance analysis of this network functionality.

The core contribution of this chapter is to leverage network calculus principles in describing the behaviour of different classes of traffic, i.e. scheduled, au-

audio/video, and best effort, while traversing an industrial network that supports different TSN shaping methods including: TAS, CBS, and frame pre-emption. Initially, the arrival curve followed by the service curve, for each class of traffic, have been presented. The arrival curves describe the traffic nature, while the service curve describes how the TSN scheduling and shaping methods will serve each traffic class.

Using the definition of arrival and service curves, this chapter proceeds to derive bounds on the worst-case delay, for each flow belonging to different traffic classes. The worst case occurs when multiple flows arrive at the same interface and are ready to be transmitted. For example, a critical frame, of a flow marked with high priority and e-mac, will delay the other frames with less priority and might also pre-empt the ongoing frame transmission. Finally, end-to end delay for a flow, traversing a multiple hops path within Time-Sensitive Networking (TSN)-based network, could be computed by aggregating the worst-case per hop delay along that path.

The outcome, of the analysis and derivations, is a network calculus-based framework, which builds a robust understanding of worst-case delay analysis that could be used to gain insights into TSN-based networks

4.1 System Overview

Using a Network calculus (NC)-based approach gives us a crucial tool to determine the worst-case delay bound. However, the complexity increases when different types of data traffic are combined, making it difficult to accurately compute this bound. This is especially true for data flows that are bursty at times, which affects other flows traversing the same path, leading to increased delays along the path. Such delays accumulate as a flow passes through multiple hops. Initially, the analysis focuses on a single-hop model that represents a Quality Control and Classification After Production (QCAP) node. This model is explained in detail in Section 3.4 and illustrated in Figure 4.1, serves as a key framework. The analysis of this model focus on defining detailed arrival and service curves, which are essential for understanding the behaviour of each traffic class and form the foundation to build NC-based framework.

In this model, a dynamic and carefully structured TSN-enabled network is considered that serves as the foundation for developing a precise worst-case delay bound, as detailed in Section 2.2.3. Each network TSN device has a set of inter-

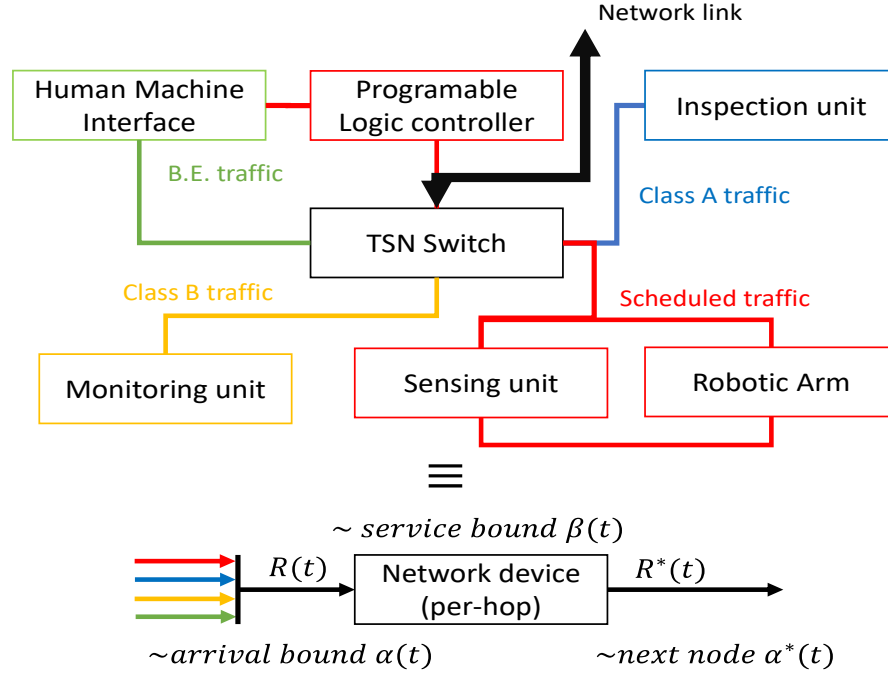


Figure 4.1: Single hop model of QCAP TSN Node.

faces, where each interface implements four distinct queues to process scheduled traffic, Class A, Class B, and best-effort traffic. The interaction between these queues, along with the unique behaviour of each traffic class, provides a comprehensive view of worst-case delay scenarios. Figure 4.2 depicts the complete operation of a TSN interface.

In this network model, the following assumptions, of the TSN interface behaviour when processing frames of different traffic classes, are considered:

Scheduled traffic frames are given the highest priority. To ensure these frames experience minimal delay, a carefully designed time-aware shaping scheduler is implemented. This scheduler allocates a dedicated transmission window exclusively for the highest-priority traffic. When a scheduled traffic frame arrives, the control gate for its queue opens immediately, allowing the frame to pass through without delay, while the gates for all other queues remain closed. A key aspect of this analysis is the integration of the TSN Hold/Release preemption mode, as described in [Pre16]. This integration creates a protective guard band—a kind of buffer—before the start of the scheduled traffic’s transmission window. This guard band includes the minimum preemptable segment of a lower-priority frame, ensuring that the scheduled traffic’s window remains protected from interference. Additionally, each frame categorised as sched-

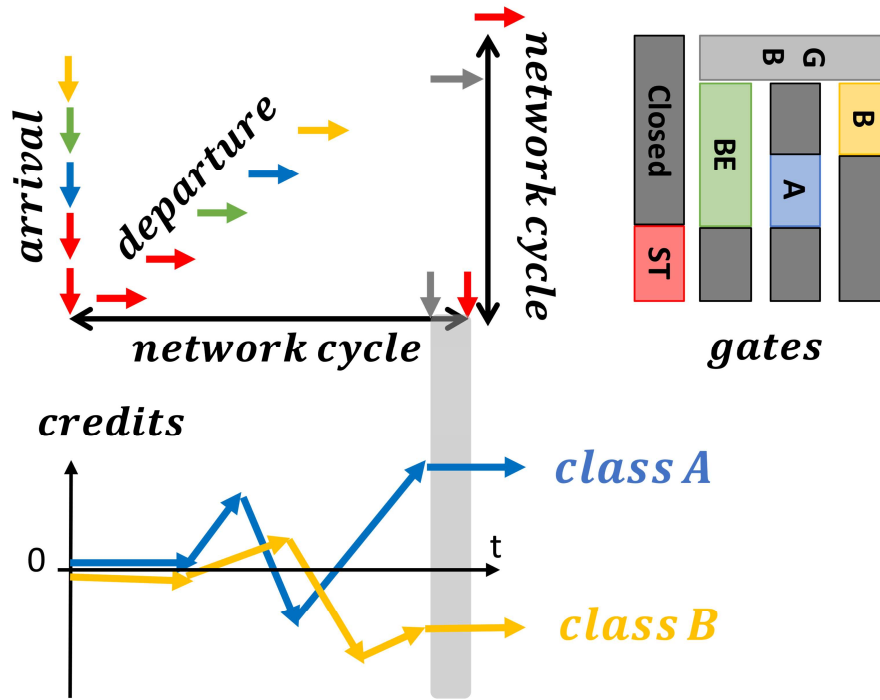


Figure 4.2: TSN device interface processing different data flows arriving simultaneously

uled traffic is treated as a single transmission entity. This means that all bits within the frame are synchronised, ensuring the frame is transmitted as a unified whole.

The behaviour of **Class A** and **Class B** traffic frames is characterized by bursts of activity, forming a dynamic bit stream. To manage this behaviour, credit-based shaping is employed as an effective strategy. This approach regulates the burstiness of the traffic before it is transmitted. Credits accumulate only during idle periods of **Class A/B** traffic, specifically when the traffic is outside the scheduled transmission window and the guard band buffer interval. Transmission is controlled by the opening and closing of control gates for **Class A** and **Class B** traffic queues, following a carefully planned schedule. These two traffic classes share the transmission window that remains after the scheduled traffic has passed, but each class is given its own distinct transmission opportunities to ensure that their control gates do not open simultaneously. This separation is crucial for maintaining a smooth transmission process. Interestingly, the control gate for best-effort traffic can align with those of the Class A and B queues, leading to better bandwidth utilisation. Transmission preemption adds another layer of complexity—if a frame's transmission is interrupted, the remaining part of the frame waits for the next available transmission window. When different

traffic frames are queued for transmission, the priority system takes over, determining the order in which frames are sent.

Best effort traffic is characterised by its sporadic bit stream, with bursts of activity. Often, these frames overlap with Class A and Class B traffic in the shared transmission window. In this complex interaction, a particular scenario arises when a frame is preempted at the guard band. If higher-priority frames arrive at the same time, the preempted frame must wait its turn to be transmitted. An important protocol governs how frame transmission continues during preemption. Specifically, if a frame has already started transmitting before a higher-priority, preemptable frame (like Class A or Class B) arrives, the ongoing transmission is allowed to continue without interruption. This approach aims to maintain a smooth and uninterrupted flow of transmissions, even in the case of dynamic frame interactions.

4.2 Worst-case delay analysis for multiple classes of traffic.

Based on the aforementioned assumptions and considering the scenario of a single-hop model within a QCAP node. the Arrival and Service curves are driven for multiple classes of traffic to provide insights into this analysis.

4.2.1 Arrival and Service curves of scheduled traffic

Packet train arrival curve is used to model scheduled flows where packets arrive in real-time as bursts rather than continuous flow. The arrival curve α_{ST} is defined as follows:

$$\alpha_{ST}(t) = \begin{cases} \sigma; & 0 \leq t \leq T_{ST} \\ R.(t - T_{ST}) + \sigma; & T_{ST} \leq t \end{cases} \quad (4.1)$$

Where σ represent the largest burst size of a scheduled flow, T_{ST} is the minimum time between the end of one burst and the start of the next, and R is the average arrival rate of packets. Moreover, a general driven form for describing the arrival curve, based on the aforementioned TSN interface model considering the periodic arrival of scheduled frames with:

$$\alpha_{ST}(t) = \sum_i L_i \times \frac{t}{T_i}, \quad (4.2)$$

where L_i represents the i^{th} frame/burst length and T_i represents the arrival period of the i^{th} scheduled traffic flow, respectively.

The service curve (β) of the scheduled traffic has a rate-latency form that can be defined by

$$\beta_{ST}(t) = c \times (t - T_{ST}), \quad (4.3)$$

where c represents the link speed and T_{ST} represents the maximum encountered delay for a scheduled traffic frame. T_{ST} can be expressed as

$$T_{ST} = \frac{L_{ST,k} - L_{minST,k}}{c}, \quad (4.4)$$

where $L_{ST,k}$ represents the total length of the allocated window for all scheduled traffic frames at hop k and $L_{minST,k}$ represents the minimum frame length of the scheduled traffic at hop k .

4.2.2 Arrival and Service curves of AVB traffic

The arrival curves of class A and class B traffic are constrained by a token bucket, defined by

$$\alpha_{class}(t) = r_{class} \times t + \sigma_{class}, \quad (4.5)$$

where r_{class} represents the sum of the traffic class data rate and σ_{class} represents the sum of the traffic class burst.

While the service curves are based on rate-latency equations that apply to both classes A and B traffic. For class A traffic, the service curve is given by

$$\beta_A(t) = R_A \times (t - T_A), \quad (4.6)$$

where R_A represents the effective rate of class A traffic and T_A represents the maximum delay encountered by a bit arriving at an empty queue. R_A can be expressed as

$$R_A = \frac{I_A \times [c - r_{ST,k}]}{c} - \frac{I_A \times r_{GB}}{c - I_A}, \quad (4.7)$$

where I_A represents the idle credit accumulation slope (bits/s) for AVB Class A traffic, $r_{ST,k}$ represents the sum of all scheduled traffic rates going through

hop k , i.e., $r_{ST,k} = \sum_i (\frac{L_i}{T_i})$ per k , and r_{GB} represents the guard band rate that is calculated as $r_{GB} = \frac{L_{minPkt}}{T_{minPkt}}$, where L_{minPkt} represents the maximum length of the minimum non-preemptable portion of a low priority frame and T_{minPkt} represents the guard band duration. The first term in eq. (4.7) captures the transmission slope/rate of class A traffic outside the scheduled traffic transmission window, while the second term represents the guard band duration where class A transmissions are not allowed. T_A can be expressed as

$$T_A = \frac{L_{ST,k}}{r_{ST,k}} + \frac{L_{minPkt,k} + L_{BE}}{c - r_{ST,k}}, \quad (4.8)$$

where L_{BE} represents the maximum frame length of the best effort traffic. In eq. (4.8) the first term represents the worst case which is when a guard band occurs and class A has negative credit. In this case, a class A frame has to wait for the guard band duration, the transmissions of all scheduled traffic, and the best effort traffic queued until it has accumulated enough positive credit to be allowed to be transmitted.

Similarly, the service curve of class B traffic can be defined as

$$\beta_B(t) = R_B \times (t - T_B), \quad (4.9)$$

where R_B represents the effective rate of class A traffic and T_B represents the maximum delay encountered by a bit arriving at an empty class B queue. R_B and T_B can be expressed as

$$R_B = \frac{I_B \times [c - r_{ST}]}{c} - \frac{I_B \times r_{GB}}{c - I_B}, \quad (4.10)$$

$$T_B = \frac{L_{ST,k}}{r_{ST,k}} + \frac{L_{minPkt,k} + L_A + L_{BE}}{c - r_{ST,k}}, \quad (4.11)$$

where I_B represents the idle slope of AVB Class B traffic and L_A represents the maximum frame length of class A traffic. Eq. (4.10) is analogues to eq. (4.7). Eq. (4.11) corresponds to the worst case when a guard band occurs and class B has negative credit. Therefore, a class B frame must wait for the guard band duration, transmissions of all scheduled traffic, the class A transmission window, and best effort traffic queued given that class B still needs to accumulate enough positive credit.

4.2.3 Arrival and Service curves of BE traffic

The arrival curve for best effort traffic represents the maximum amount of data that within a designated time interval (T), without any specific guarantees on the rate or timing of the traffic. Best effort traffic is typically subject to fluctuations and exhibits the following characteristics: unpredictable arrival patterns, no guaranteed delivery, and variable rates.

To model the BE traffic arrival, considering peak rates over shorter intervals, the following formula could be used:

$$\alpha_{BE}(t) = \begin{cases} R_{Peak} \cdot t, & t \leq T_{Peak} \\ R_{avg} \cdot t + \sigma, & t > T_{Peak} \end{cases} \quad (4.12)$$

Here, R_{Peak} represents peak arrival time, R_{avg} denotes the average arrival rate, σ represents the burst size, and T_{peak} is a threshold time interval over which peak rates are observed.

The BE service curve is also defined by rate-latency

$$\beta_{BE}(t) = R_{BE} \times (t - T_{BE}) \quad (4.13)$$

$$R_{BE} = c - r_{ST} - \min(r_{AVB}), \quad (4.14)$$

$$T_{BE} = \sigma_{BE} + \frac{L_{ST,k}}{r_{ST,k}} + \frac{L_{minPkt} + L_A + L_B}{c - r_{ST}} \quad (4.15)$$

where σ_{BE} is the burst period of the BE traffic.

4.3 Worst Case Delay bounds per single hop

Within the aforementioned defined system scope, scheduled traffic seeks guaranteed transmission times, AVB traffic vies for time-sensitive multimedia delivery, and BE traffic thrives on flexibility and efficient bandwidth utilisation. The symphony of these classes creates a complex interplay that reverberates across the network, impacting the worst-case delay scenarios experienced by each flow. Therefore, in accordance with the network calculus framework where the arrival curves characterise the upper limit of arriving bits within a specific

time frame, while the service curves dictate the minimum number of bits transmitted in that period. These curves form the cornerstone of this exploration, offering profound insights into network intricacies.

Hence, the delay bounds are defined for a given flow (f), traversing a single hop (k), and belonging to a specific traffic class (cls), as follows:

$$D_{f,k} = T_{cls} + \frac{\sigma_{cls} - L_{min,f}}{R_{cls}} + \frac{L_{min,f}}{c}, \quad (4.16)$$

Here, T_{cls} represents the maximum delay encountered by a frame of, cls , traffic class, σ_{cls} is the burst size, $L_{min,f}$ is the minimum frame length, R_{cls} is the traffic rate for this class, and C represents the link speed.

Equation 4.16 includes the different elements that derive the worst-case per-hop delay for each frame traversing the TSN-based network. However, for end-to-end delays, it is crucial to consider a multi-hop model that uses accumulation delay, where the total end-to-end delay is the sum of per-hop delays. For example, it could be used to analyse each hop separately to determine its delay contribution, considering both its own traffic and the competing traffic. Then, the total delay is the sum of delays at each hop, considering the effects of competing traffic at each hop. Furthermore, it is worthy to note that other factors like network topology and traffic patterns play crucial roles in determining delays. Therefore, those are investigated and analysed further in the upcoming chapters.

Other than worst case-delay analysis, the backlog/queues length is an important parameter for network performance evaluation, that allows managing buffer requirements effectively. The backlog $B(t)$ at time can be expressed as the difference between the amount of data that has arrived and the amount of data that has been transmitted. It represents the amount of data waiting to be processed. Thus, it mathematically can be expressed as:

$$B(t) = \max(0, A(t) - C \cdot t) \quad (4.17)$$

Backlog cannot be negative because it represents the amount of data waiting to be processed:

- When $A(t) \leq C \cdot t$: the arrival rate is less than or equal to the transmission rate. The backlog $B(t)$ is zero, indicating that the buffer is empty, or no

data is waiting to be processed.

- When $A(t) > C \cdot t$: the arrival rate exceeds the transmission rate. The backlog $B(t)$ is positive, indicating that data is accumulating in the buffer because it is arriving faster than it can be transmitted.

In conclusion, this analysis provides a guide to test the feasibility of network configurations against the network capacity to accommodate critical and non-critical traffic flows for industrial use cases.

Chapter 5

Improved Timing Analysis of TSN based Redundant Industrial Networks

This chapter builds up on the previous analysis and further improves the timing analysis for TSN-based industrial networks, aiming to offer a thorough understanding of the temporal characteristics and performance of industrial communication systems using TSN. The analysis considers the complex interactions between various traffic classes, their timing parameters, and the underlying network architecture. This exploration is crucial for ensuring the robustness of the communication infrastructure and optimizing its efficiency to support a wide range of industrial applications.

The chapter is structured as follows:

System Architecture Model In this section, a comprehensive application model and architecture for an industrial automation system are introduced. This clear representation of the system's components, interconnections, and communication requirements forms the foundation for the analysis that follows.

Arrival Curves for Different Industrial Traffic A key aspect of this study is the detailed characterization of the various traffic classes common in industrial automation TSN profiles. By defining the arrival curves for each class, a quantifiable representation of data flow is created, providing deeper insights into the network's temporal dynamics.

Service Curves for Different Industrial Traffic Building on the arrival curve

framework, this section presents the analysis of service curves. The focus is on how higher-priority traffic affects the service delivery of lower-priority streams. It explores various mechanisms, including preemption mode, gate shaping, and priority selection, to shed light on their impact on traffic scheduling and timing guarantees.

Industrial Data Traffic Analysis This section is further divided into two integral parts:

1. **Tight-Bounds for Industrial Data Traffic:** derive strict upper bounds for industrial data traffic, taking into account different service scenarios and network conditions. These bounds are crucial for evaluating how well the network can meet timing requirements under various circumstances.
2. **Worst Case End-to-End Delay and Backlog Analysis:** The main goal of this analysis is to estimate the worst-case end-to-end delays for each traffic class. By closely examining how traffic arrival patterns interact with service behaviours, a thorough assessment of the network's performance in terms of delay and data backlog is presented.

In conclusion, this chapter offers a systematic exploration of TSN-based industrial networks to enhance the understanding of their timing characteristics and performance details. The insights gained are intended to significantly aid in the design, deployment, and optimization of robust and efficient industrial communication systems.

5.1 System Architecture Model

The Industrial Internet of Things (IIoT) requires the integration of both Information Technology (IT) and Operational Technology (OT). For instance, in smart manufacturing and industrial automation, OT operates at two levels:

Field Level: this involves multiple production lines, each with distinct goals to complete various tasks.

Factory Level: this level collects OT data in real-time, enabling the automation of processes occurring at the field level. On the IT side, the focus is on the Enterprise and Management Level, where enterprise resource planning (ERP) manages real-time business processes.

Such a system must meet several requirements: minimal network failures, low

installation and maintenance costs, support for a limited number of devices, and a single interface assignment for each device.

To model such systems, a set of applications are considered, which are hosted over many yet single-interface devices, i.e. single MAC address and single IP for each end system (*ES*). The communication between those devices results in eight different types of traffic classes. The behaviour and characteristics, e.g. packet size, periodicity... etc., of each traffic class are given by the system designer. The set of all flows that belong to the different types of traffic are defined as TC , 5.1 and 5.2, where i represents the traffic type and priority and m represents the flow count.

$$TC = \bigcup_{i=0}^{i=7} TC_i \quad (5.1)$$

$$TC_i = \sum_{\forall m} f_{m,\{V_s,V_d\}}^i \quad (5.2)$$

Traffic priority is defined using 3-bits Priority Code Point (PCP), i.e. the higher the value of i , the higher the priority of the traffic, e.g. TC_7 is the set of all flows that is characterised as isochronous cyclic real-time. A flow $f_{m,\{V_s,V_d\}}^i$ belongs to TC_i is transmitted by a source node V_s and directed to a destination node V_d . The path computations to route the flows is an important problem and affects the traffic latency. However, this problem could be addressed separately as in [GRK16], therefore all the routing tables are assumable predefined.

The system architecture of a smart factory field level consists of a set of production lines that are connected by switches (**SW**) and managed by a Central Network Configuration (CNC) entity. A single production line is a set of connected production cells (**C**), where each cell is formed by connecting a group of devices (**D**), e.g. sensor, actuator, Programmable Logic Controller (PLC), human machine interface (HMI) and network bridges (**B**)¹.

This system architecture defines a TSN domain as a group of TSN hosts that are connected by TSN network devices. A TSN domain could join other TSN domain to form a larger domain. An example is represented in Figure 5.1, where production cells, each cell is as a TSN domain, are joined to form a larger TSN domain, i.e. production line. Using such architecture, It becomes convenient

¹Bridges are constrained devices, e.g. 2 or 4 ports, therefore are used to connect the devices of a production cell. In contrast, switches are more capable, therefore are used to connect production cells of a production line.

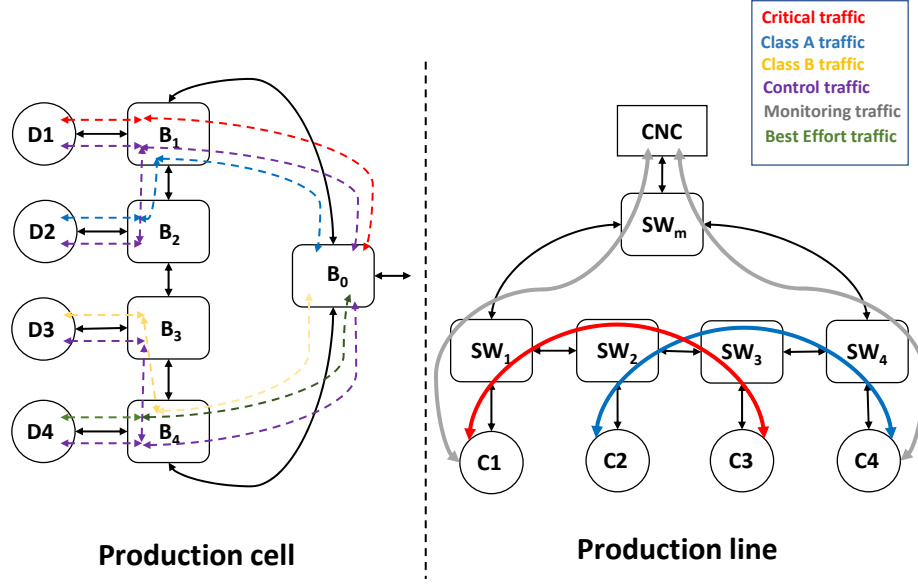


Figure 5.1: Production System Architecture

to study the industrial network using both macro perspective/cell level or micro perspective/device level, which could benefit from an industrial network symmetry and the rules of network calculus in analysing it as discussed later in the following section. A TSN domain topology is modelled as an undirected graph $G(V, L)$, where V represents the graph vertices and L represents the set of physical full-duplex links that connect the graph vertices. The definition of V depends on the domain scope, 5.3.

$$V = \begin{cases} C \cup SW, & \text{MACRO.} \\ D \cup B, & \text{MICRO.} \end{cases} \quad (5.3)$$

For Figure 5.1, a ring topology is used to connect a group of devices to form a production cell, where $V = D \cup B$ is a set of devices and bridges. In addition, a group of switches is used to connect the production cells, $V = C \cup SW$, forming an architecture of a production line. A physical link, L_{ab} , connects two vertices, $[V_a, V_b]$, and has capacity of $L_{ab} \cdot C$ Mbps. Links are depicted as solid black lines with double arrows. For each data flow, an optimal path $f_{\{V_s, V_d\}}^i \cdot P$ is predefined, which is an order sequence of the data links connecting V_s and V_d . For example, a flow belongs to an event/alarm class $f_{\{V_{D2}, V_{B0}\}}^2$ happening at D_2 to be reported through B_0 , has the following path sequence: $\{[D_2, B_2], [B_2, B_1], [B_1, B_0]\}$.

5.2 Arrival curves for different industrial traffic

The industrial automation profile presents a dynamic set of traffic types, each with its unique attributes. The main point lies in unveiling the arrival curve for each traffic class, allowing us to decipher the individual characteristics that define traffic behaviour within the industrial automation TSN framework.

isochronous cyclic real-time: End devices, which are synchronised to the network working clock, generate the isochronous data traffic. Each flow, of this traffic class, has a period τ_f that is equal to or a multiple of the network cycle period $T_{network}$. i.e. $T_{network} = \text{MIN}(T_1, T_2, \dots, T_n)$ where n is the number of flows of this traffic class. Therefore, the traffic behaviour repeats every $T_{schedule}$. i.e. $T_{schedule} = \text{LCM}(T_1, T_2, \dots, T_n)$. The arrival curve of each flow is represented in Eq. 5.4, where l_i is the packet i^{th} size, which is greater than or equal to the flow minimum packet size L_f , T_i is a multiple of the flow period T_f . For example, Figure 5.2, three flows, $n = 1, 2, 3$, with burst size L_n , and an interval between successive bursts T_n for each flow. The arrival curve for each flow, $\alpha_i(t)$, is covered by Eq. 5.4. Then, the aggregate arrival curve for all three flows, $\alpha_{tot}(t)$, is driven by Eq. 5.5. where $T_{network} = \text{MIN}(T_1, T_2, T_3) = T_1$ and $T_{schedule} = \text{LCM}(T_1, T_2, T_3) = 8T_1 = 4T_2 = 2T_3$. Then to count for the overhead resulting from the preemption, $\alpha_{oh}(t)$, the arrival curve is represented by Eq. 5.6.

$$\alpha_f(t) = \sum_{\forall i \in f} l_i \times \left\lceil \frac{t}{\tau_i} \right\rceil, i \in Z_+^* = \{1, 2, \dots\} \quad (5.4)$$

Using the aggregation property of network calculus, the aggregate arrival curve, of n isochronous flows is a periodic stream with period $T_{schedule}$, and is presented in 5.5. Where x_f represents the reduction ratio of a flow period, i.e. $x_f = \frac{T_f}{T_{network}}$.

$$\alpha_{iso}(t) = \frac{1}{T_{network}} \sum_{f=1}^n L_f \times \left\lceil \frac{t}{x_f} \right\rceil \quad (5.5)$$

Isochronous frames are serviced by the network devices as express frames, i.e. frames belong to this traffic class pre-empt other traffic, which adds overhead to those pre-empted traffic. In the worst case, for each set of Isochronous frames transmissions, there will be a pre-emption for a lower priority frame. Since the overhead only appears immediately after isochronous frame transmission,

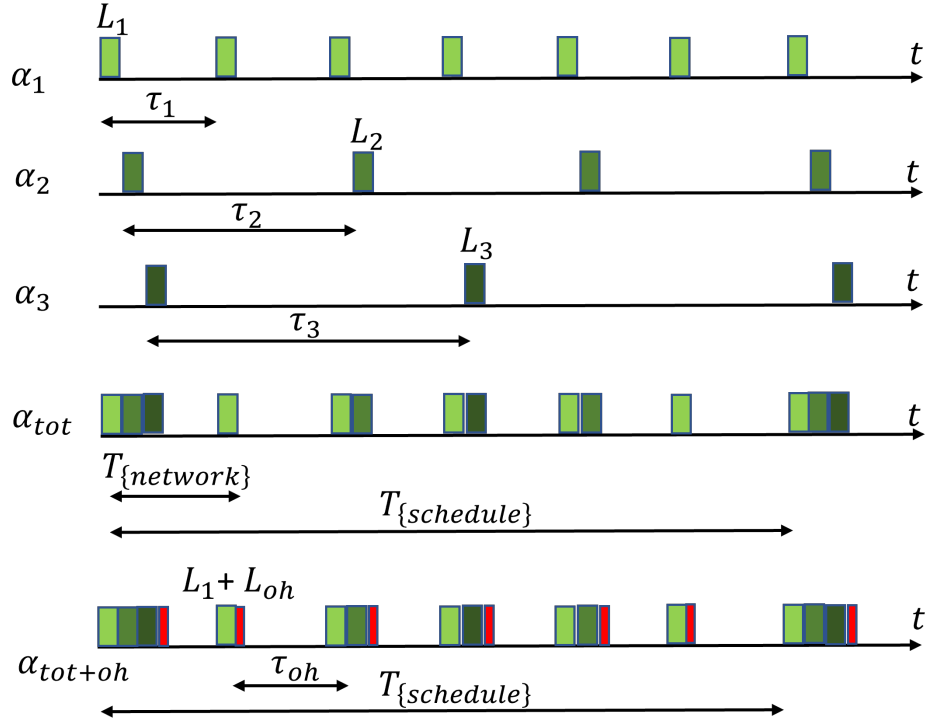


Figure 5.2: Arrival model for the isochronous frames with pre-emption overheads

the arrival curve of overheads can be given by 5.6. Where L_{oh} represents the overhead size in bits and τ_{oh} is the time period for a new overhead arrival. For worst case analysis, τ_{oh} is equal to the period of the maximum isochronous flow rate, i.e. $\tau_{oh} = \min(\tau_1, \tau_2, \dots, \tau_n)$.

$$\alpha_{oh}(t) = L_{oh} \times \left\lceil \frac{t}{\tau_{oh}} \right\rceil \quad (5.6)$$

non-isochronous cyclic real-time: frames that belong to this traffic class are the result of applications, which are not synchronised to the network working clock. However, the data transmissions of the end devices, running those applications, are synchronised to a working clock or local timescale. A guard band is established before non-isochronous frames transmissions to omit the lower priority traffic transmissions. Thus, to ensure that real-time frames are not being hold when they arrive, especially the non-isochronous frames are preemptable frames, i.e. frames belong to this traffic class can't pre-empt other traffic. In the worst case scenario, a guard band is present before non-isochronous frames transmission, which has a size L_{GB} that is equal to the minimum of maximum lower priority traffic frame size $L_{max}^{PCP<6}$ or the amount of bits that could be transmitted till the next non-isochronous frame transmission $L_{idle}^{PCP=6}$.

i.e., $L_{GB} = \min(L_{max}^{PCP<6}, L_{idle}^{PCP=6})$. The arrival curve of the aggregate non-isochronous flows is given by:

$$\alpha_{non_iso}(t) = \sum_{f=1}^n (L_f \times \left\lceil \frac{t}{\tau_f} \right\rceil) + (L_{GB} \times \left\lceil \frac{t}{\tau_{GB}} \right\rceil) \quad (5.7)$$

Eq. 5.7 could be simplified, by considering the guard band as traffic flow of variable bits size $L_0 = L_{GB}$ and period $\tau_0 = \tau_{GB}$, to be:

$$\alpha_{non_iso}(t) = \sum_{f=0}^n (L_f \times \left\lceil \frac{t}{\tau_f} \right\rceil) \quad (5.8)$$

Audio (class A): leaky bucket model is used to generally specify the audio traffic behaviour. The transmissions of audio frames depend on the application transmission rate r_{audio} [bits/s] and the value of the initial burst b_{audio} [bits]. In addition, audio transmissions are limited by the device/port transmission rate r_{device} [bits/s]. This model captures the system ability to handle a sudden influx of data up to a certain limit, which is relevant for audio traffic due to the following:

- **Codec Delays:** When an audio stream starts, there might be an initial burst of packets due to the codec's buffering and processing. Codecs often buffer a small amount of audio before starting transmission to ensure smooth playback.
- **Initial Synchronisation:** The initial burst can also include packets needed for synchronization between sender and receiver, such as establishing jitter buffers and handling any initial control messages for setting up the stream.
- **Variability in Network Conditions:** Even if the source generates audio packets at a constant rate, variations in network conditions can lead to momentary bursts of packets arriving at the receiver. The leaky bucket model can accommodate these variations by allowing a small burst.
- **Packet Reordering and Retransmissions:** In some cases, network issues can cause packets to arrive out of order or require retransmissions, leading to temporary bursts.

The arrival curve of an audio application is defined as:

$$\alpha_{audio}(t) = \min\{r_{device} \times t, r_{audio} \times t + b_{audio}\} \quad (5.9)$$

The aggregate arrival curve for a set of audio flows m , with different flow rates r_a , bursts b_a and arriving from in ingress ports, is upper bounded by:

$$\alpha_{audio}(t) = \min\left\{\sum_{\forall in} r_{in} \times t, \sum_{a=1}^m r_a \times t + b_a\right\} \quad (5.10)$$

Video (class B): similar to audio streams, the aggregate arrival curve for video streams is defined by:

$$\alpha_{video}(t) = \min\left\{\sum_{\forall in} r_{in} \times t, \sum_{v=1}^m r_v \times t + b_v\right\} \quad (5.11)$$

Network Control: periodic traffic produced by the network controller to update network devices with forwarding tables and schedules. This traffic class is of a high priority for the industrial network to function as expected, e.g. supporting topology based redundant networks. The control traffic packet size varies, therefore a packet train, with left continuity, traffic model is used to describe its behaviour, which captures the bursty nature of packet arrivals and ensures that the packet arrival process is continuous from the left, meaning that there are no sudden jumps or gaps in the traffic flow. Considering that the industrial network contains multiple controllers k , the aggregate arrival curve could be specified as follows:

$$\alpha_{ncl} = \sum_{\forall k} \left\{ \left(\sum_{n \geq 1} L_n \right)_k \times \left\lceil \frac{t}{\tau_k} \right\rceil \right\} \quad (5.12)$$

The lower priority traffic including alarms, events, configuration, diagnostics and best effort are all of sporadic nature with a variable packet sizes. A fluid model is used to describe the arrival curve behaviour. This is to consider the effect of transmitting those traffic classes on the credit accumulation of the audio and video traffic. In addition, a study of the co-existence of both IT and OT traffic on the same converged network while meeting each class of quality of service requirements. The behaviour of each class is specified by the sub-

additive closure $\bar{\alpha}$ of the arrival curve α , which are specified as follows:

$$\bar{\alpha}_{cls} = \inf\{\delta_0, \alpha_{cls}, \alpha_{cls} \otimes \alpha_{cls}, \alpha_{cls} \otimes \alpha_{cls} \otimes \alpha_{cls}, \dots\};$$

with $\delta_0(0) = 0$ and $\delta_0(t) = +\infty$ for $t > 0$.

, where

$$\alpha_{cls} = r_{cls} \times t + b_{cls} \quad (5.13)$$

5.3 Service curves for different industrial traffic

This section presents the service curves derivation for lower priority traffic when impacted by the higher priority traffic, e.g. pre-emption mode, gate shaping or priority selection. In addition, it discusses the impact of the lower priority traffic arrival on higher priority traffic service, e.g. credit accumulation for AVB traffic and non-pre-emption mode.

Isochronous cyclic real-time: A TSN switch serves the highest priority traffic once it arrives and a rate-latency service curve, 5.14, defines the service provided by a TSN interface; where C is the link rate and T_{iso} is the maximum wait period for an iso frame to be transmitted. The worst-case transmission scenario happens when a lower priority traffic is being transmitted, and a size of less than minimum packet size $minPKT$ or 123 octets is left to transmit [FFTB17], i.e. the remaining part can't be pre-empted. Thus, an isochronous packet has to wait, then T_{iso} is defined by 5.15.a. Another approach is to use Hold and Release mac merge service interface where a guard band is used before eMAC frames to trigger Hold request (stop pMAC transmissions), while Release request is triggered upon transmitting all the queued isochronous frame(s) as depicted in Figure 5.3. Then, T_{iso} is defined by 5.15.b. Thus, it ensures minimal latency (cut-through).

$$\beta_{iso}(t) = C \times (t - T_{iso})^+. \quad (5.14)$$

$$T_{iso} = \frac{1}{C} \times \begin{cases} \sum_{\forall f \in n} L_f + minPKT - L_i & ; 5.3.a \\ \sum_{\forall f \in n} L_f - L_i & ; 5.3.b \end{cases} \quad (5.15)$$

Non-isochronous cyclic real-time: Similar latency-rate service curve as [FBM⁺19], which is similar to 5.14, but without pre-emption. Therefore, Non-isochronous has to wait for a period of T_{LP} , which is the maximum time re-

quired to complete a lower priority traffic frame transmission that has started transmitting already. A guard band could be used to avoid the aforementioned case, which improves the overall non-isochronous traffic latency. The guard band has to be considered while deriving the service curve as follows:

$$\beta_{non-iso}(t) = C \times (t - T_{LP})^+. \quad (5.16)$$

, where

$$T_{LP} = \frac{1}{C} \times \begin{cases} \sum_{f \in iso} L_f + \sum_{m} L_m + \\ \max(L_{LP}) - L_i & \text{without } GB. \\ \sum_{f \in iso} L_f + \sum_{m} L_m - \\ L_i & \text{with } GB. \end{cases}$$

Audio (Class A): A TSN switch classifies audio frames as a high priority data traffic. However, this traffic is timely shaped to ensure the deterministic transmissions of a higher priority traffic (e.g. real time traffic.). In addition, It is controlled by a credit-based shaping algorithm to limit the bursts and ensure the transmissions of lower priority traffic. In Figure 5.4, a highlight of a higher priority traffic effect on the audio frames transmission. First, Figure 5.4.a depicts the effect of an express (*eMac*) isochronous frame arrival on a preamble (*pMac*) audio frame. Where, the queued audio frames may only resume transmissions after the express frames being sent, given that there is still available credit and controlled gates are open. An overhead occurs due to re-transmitting the pre-empted portion of the transmitted audio frame. In addition, Figure 5.4.b depicts the use of a guard band to delay the audio frames transmission till a higher priority non-isochronous traffic needs to be transmitted. Finally, Figure 5.4.c depicts the impact of using hold/release pre-emption mode on the transmission of audio frames. Moreover, T_{HP} refers to the non-service period of the audio frames due to the presence of a higher priority scheduled traffic (either an express or a higher priority preamble frames).

Credit accumulation and spending are another aspects that affect the audio frames transmission. T_{CR} refers to the time period needed for the audio traffic queue to accumulate enough credit to be permitted a transmission opportunity. There exist two approaches for credit allocation: *the first approach*, Figure 5.5.a,

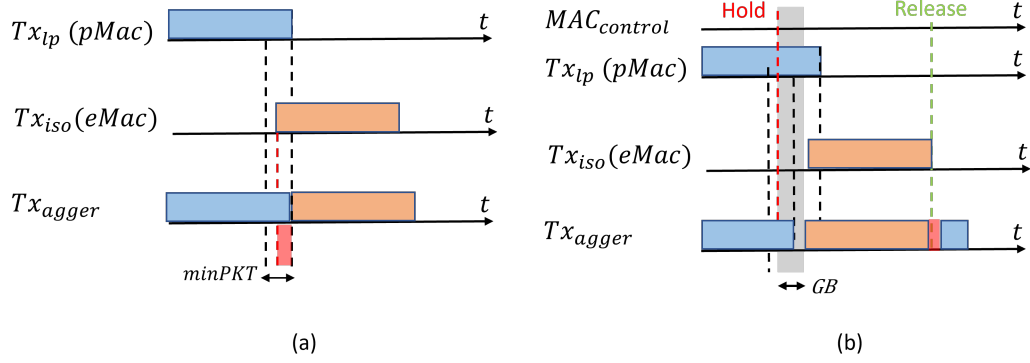


Figure 5.3: eMAC transmission with: (a) preemptable traffic ($\leq minPKT$) (b) preemptable traffic with Hold and Release mode ($\leq minPKT$).

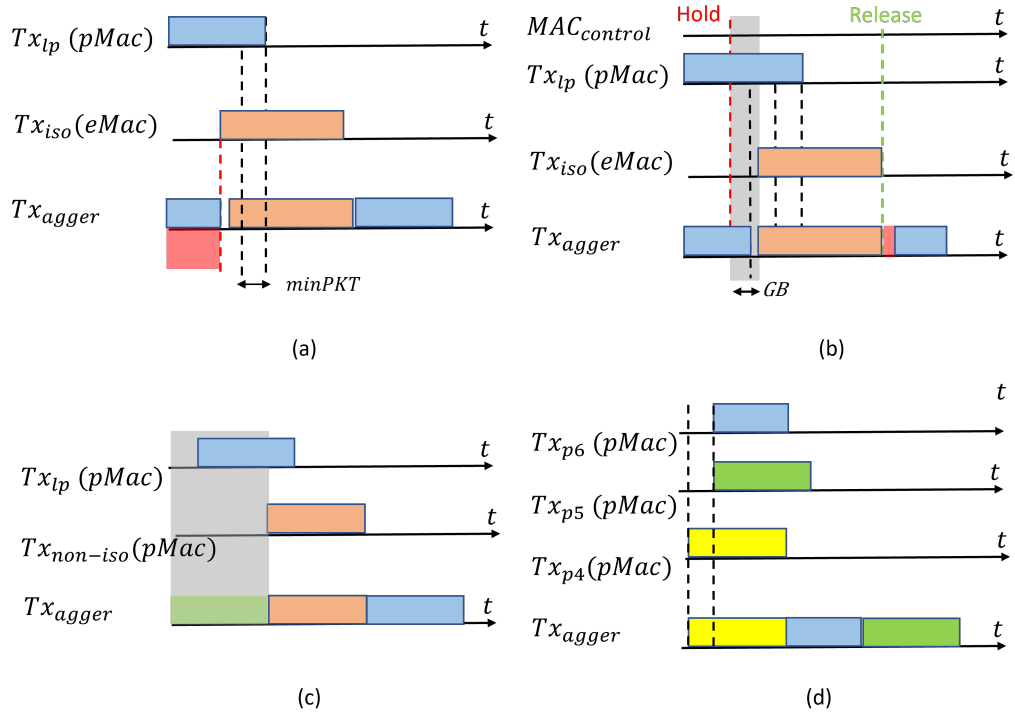


Figure 5.4: pMAC transmission with: (a) pre-emptive traffic ($> minPKT$), (b) pre-emptive traffic with Hold and Release mode ($> minPKT$), (c) non-pre-emptive traffic with guard band, (d) non-pre-emptive traffic with different priorities.

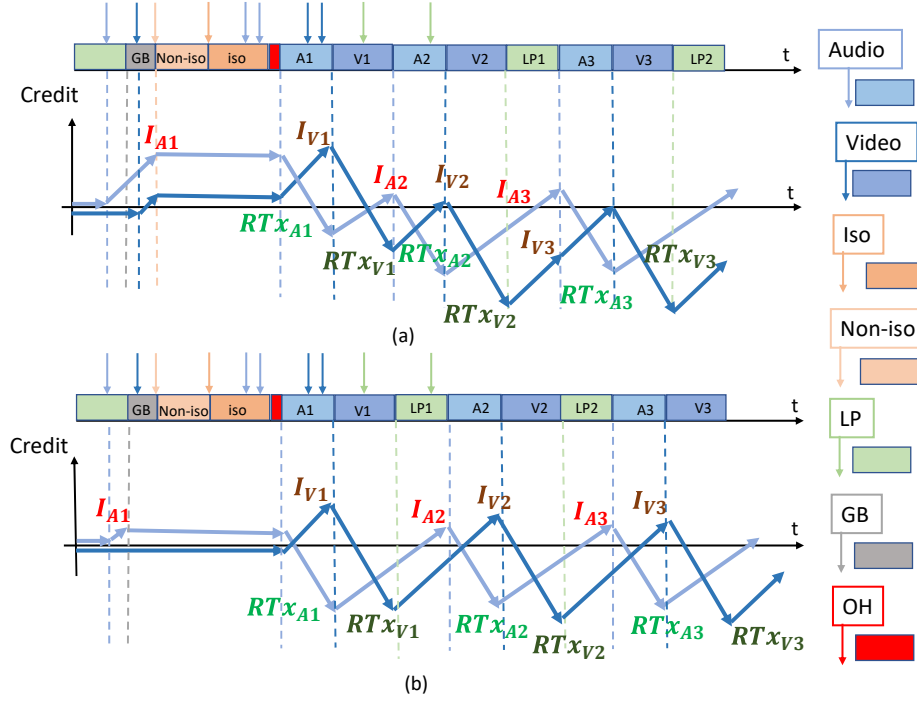


Figure 5.5: Credit allocation: (a) credit accumulation continues in the guard-band slot (b) credit accumulation stops in the guard-band slot.

follows the credit behaviour highlighted in the TSN standards, where credit still accumulates in a presence of a guard-band, i.e. controlled gates are close. *Second approach*, Figure 5.5.a, credit only accumulates while controlled gates are open. The work in [ZPZ⁺20] argue the fairness of the first approach and suggests an update of the standard to consider the second approach. Moreover, T_{LP} defines the non-service time due to ongoing transmission of a lower priority traffic. This case occurs when audio traffic does not have enough credit yet to transmit. It only obtains permission to transmit after a lower priority traffic started transmitting, therefore, it has to wait till the ongoing transmission to be completed considering that the controlled gates are open. Accordingly, the service curve of the audio traffic is defined as follows:

$$\beta_{audio}(t) = R_{audio} \times (t - T_{audio})^+. \quad (5.17)$$

where

$$R_{audio} = I_{audio} \times \left[\frac{(C - r_{HP} - r_{oh})}{C} - \frac{r_{GB}}{C - I_{audio}} \right]$$

$$T_{audio} = T_{HP} + T_{CR} + T_{LP}$$

$$T_{HP} = \frac{1}{C} \times (L_{oh} + \sum_{\forall f \in HP} L_f) + \frac{1}{C - I_{audio}} \times L_{GB}$$

$$T_{CR} = \begin{cases} \frac{1}{C-I_{video}} \times L_{video} - \frac{1}{C-I_{audio}} \times (L_{audio} - L_{LP} + L_{GB}) & \text{approach 1.} \\ \frac{1}{C-I_{video}} \times L_{video} - \frac{1}{C-I_{audio}} \times (L_{audio} - L_{LP}) & \text{approach 2.} \end{cases}$$

$$T_{LP} = \frac{1}{C - I_{audio}} (L_{LP}).$$

Video (Class B): Similar analysis, to the audio traffic, could be considered for this type of traffic. A TSN interface provide a latency service curve considering the three periods where no service could be provided, which are: scheduled traffic is being transmitted, either isochronous or non-isochronous, accumulating credit while other credit based shaped traffic is being transmitted, and lower priority traffic is being transmitted right before accumulating enough traffic to be able to transmit. In addition, the two approach of credit accumulations are considered. Therefore, the service curve for the video data traffic is concluded as follows:

$$\beta_{video}(t) = R_{video} \times (t - T_{video})^+. \quad (5.18)$$

, where

$$R_{video} = I_{video} \times \left[\frac{(C - r_{HP} - r_{oh})}{C} - \frac{r_{GB}}{C - I_{video}} \right]$$

$$T_{video} = T_{HP} + T_{CR} + T_{LP}$$

$$T_{HP} = \frac{1}{C} \times (L_{oh} + \sum_{\forall f \in HP} L_f) + \frac{1}{C - I_{video}} \times L_{GB}$$

$$+ \frac{1}{C - I_{audio}} \times L_{audio}$$

$$T_{CR} = \begin{cases} \frac{1}{C-I_{audio}} \times L_{audio} - \frac{1}{C-I_{video}} \times (L_{video} - L_{LP} + L_{GB}) & \text{approach 1.} \\ \frac{1}{C-I_{audio}} \times L_{audio} - \frac{1}{C-I_{video}} \times (L_{video} - L_{LP}) & \text{approach 2.} \end{cases}$$

$$T_{LP} = \frac{1}{C - I_{video}} L_{LP}.$$

Network control: due to the importance of this traffic class, it holds the highest priority among the lower priority traffic. Considering that the higher priority data traffic are constrained by the defined arrival curves in Section IV.A, the service curve for the network control traffic is defined by 5.19. Where ρ_{ncl} and σ_{ncl} represent the aggregate transmission rates and the aggregate bursts of the preceding higher priority data traffic.

$$\beta_{ncl}(t) = [(C - \rho_{ncl}) \times t - \sigma_{ncl}]^+. \quad (5.19)$$

The remaining data traffic including alarms, events, configuration, diagnostics and best effort are all have similar service curve $\beta_{class}(t)$. The main difference is the priority of the data traffic class, which will impact the definition of ρ_{class} and σ_{class} to include the accumulated data rates and bursts from the preceding in priority data traffic.

5.4 Industrial data traffic analysis

This section outlines the analysis, using the deterministic bounds, introduced in 2.2.3, for the eight data traffic classes of the industrial TSN profile. Then, the obtained bounds could be used to calculate the worst case delay for each traffic class.

5.4.1 Tight-bounds for Industrial data traffic

By modelling the incoming flow $R(t)$ and outgoing flow $R^*(t)$ of a system, It could be possible to assess the time delay and the backlog offered by this system, Figure 2.3.a. A system could be a network device or group of devices that are connected together to form a designated topology. Network calculus provides three tight bounds, given that a flow is constrained by arrival curve $\alpha(t)$ and served in a system by service curve $\beta(t)$ as in Figure 2.3.b, which are:

1- Maximum Buffer size/ Buffer bound ($\lceil B \rceil$): the maximum vertical deviation between the arrival curve and the service curve. It can be calculated using

$$\lceil B \rceil \leq \sup_t \{\alpha(t) - \beta(t)\} \quad (5.20)$$

2- Maximum delay/ delay bound $\lceil D \rceil$: the maximum horizontal deviation between the arrival curve and the service curve. It can be calculated using

$$\lceil D \rceil = \sup_s \{ \inf_t \{ \alpha(s) \leq \beta(s+t) \} \} \quad (5.21)$$

3- Arrival curve of the system output: the resulting output from a system experience variable delay, i.e. increase in burstiness due to the nondeterminism of the system elements. Therefore, network calculus captures the upper bound arrival curve of the system output $\alpha^*(t)$ using the following formula:

$$\alpha^*(t) = \sup_{0 \leq s} \{ \alpha(t+s) - \beta(s) \} \quad (5.22)$$

The three tight bounds could be shortened and hence evaluated using Min-Plus algebra as previously explained in section 2.2.1.

5.4.2 Worst case end-to-end delay and backlog analysis

Worst case end-to-end delay: D_{class}^h defines the worst case queuing delay for the traffic class flows traversing a single hop, i.e. TSN interface

$$D_{class}^h = h(\alpha_{class}^h(t), \beta_{class}^h(t)) \quad (5.23)$$

where $\alpha_{class}^h(t)$ is the aggregate arrival curve of all the traffic class flows, as described in section 5.2, and $\beta_{class}^h(t)$ is the service curve, introduced in section 5.3, that is offered by node h . Then for each traffic flow the worst case per hop overall delay can be obtained by

$$D_f^h = T_{class} + \frac{\sigma_{class} - L_{min,f}}{R_{class}} + \frac{L_{min,f}}{C} \quad (5.24)$$

Moreover, the worst case end-to-end delay is obtained by adding up the per-hop delays along the flow path.

Flow shape at intermediate hop: α_f^{h*} defines the arrival curve of a flow f , which is an output of the current hop arriving at the next hop h^* along its path

$$\alpha_f^{h*}(t) = \alpha_f^h(t) \odot \delta_{D_f^h}(t) \quad (5.25)$$

$$\delta_{D_f^h}(t) = \begin{cases} 0 & t \leq D_f^h \\ \infty & otherwise \end{cases}$$

Network queues configurations: Number of active queues for a TSN interface can be obtained by analysing the types of flows traversing a node at any instance of the network run time. In addition, the size of each queue, for this interface, could be obtained by calculating the maximum vertical deviation between the queued flows aggregate arrival rate and the interface service rate.

Chapter 6

Evaluation and Validation of TSN Industrial Networks

In this chapter, a comprehensive evaluation of advanced network configurations is presented and their impact on the Worst-Case Delays (WCDs) for industrial traffic classes. The conducted analysis builds upon existing methodologies by introducing novel approaches both in simulation techniques and in deriving actionable insights from these simulations. The following highlights the new contributions of the conducted simulations:

- **Enhanced Simulation Framework:** an implementation of impNC/TSN_industrial23 framework is utilised to provide an in-depth analysis of redundant industrial networks. This framework offers improved granularity and accuracy in simulating network performance, particularly for Time-Sensitive Networking (TSN) and Audio Video Bridging (AVB) traffic classes. Unlike previous frameworks, impNC/TSN_industrial23 integrates various traffic models and network configurations, allowing me to evaluate a broader range of scenarios and optimise configurations more effectively.
- **Advanced Traffic Regrouping Techniques:** the performed simulation introduces a novel approach to traffic regrouping. It evaluates the impact of redistributing traffic flows across different paths and optimising offered loads to minimize WCDs. This technique represents a significant advancement over traditional methods by dynamically adjusting traffic distribution in real-time, rather than relying on static configurations.
- **Offset Strategies for Delay Reduction:** another approach is introduced to explore the implementation of time offsets for traffic flows to address

queuing delays. This approach provides a new perspective on managing traffic flows in a way that significantly reduces overall WCDs, enhancing the efficiency and reliability of network operations.

- **Optimal Accumulation Rate Selection:** selecting the optimal accumulation rate for AVB classes without affecting lower-priority traffic represents a new approach for balancing performance across different traffic classes. This method ensures that high-priority AVB traffic is managed effectively without compromising the quality of service for other traffic types.

Moreover, the following points summarise the new insights, compared to other related work, resulting from the research through this thesis:

- **Detailed Traffic Class Analysis:** The main findings provide a detailed understanding of how different network configurations affect various traffic classes, such as isochronous and non-isochronous flows. This level of detail helps identify specific configuration changes needed to meet strict deadlines for critical traffic, offering insights that were less clear in previous studies.
- **Practical Feasibility and Configuration Validation:** We assess the feasibility of network configurations through practical evaluation rather than theoretical models. By testing default settings, optimising traffic distribution, and adjusting link bandwidth, a more accurate picture of how these configurations impact real-world performance is offered.
- **Operational Dependencies and Network Unit Interactions:** the conducted analysis addresses the interdependencies between network units, which previous studies might have overlooked. By considering how units interact within the entire network, the potential operational challenges and failure points that could affect overall performance, could be highlighted.

This chapter highlights the significance of using simulation techniques and practical insights, resulting from my research, to analyse network optimisation in industrial. The integration of advanced traffic management strategies and detailed feasibility assessments represents a significant step forward in optimising network configurations for industrial applications

6.1 Performance Evaluation of TSN based industrial network using simulation models

6.1.1 TSN network structure and design

To evaluate the performance of different queuing mechanisms, while handling both critical and non-critical traffic; a network setup, of five switches and 19 end devices, has been implemented to form a hybrid, tree and star, topology. A tree topology ensures the aggregation of all traffic flow traversing upward towards the root, also ensures the disaggregation of traffic flows traversing downward towards the leaves. Additionally, a star topology provides a single-hop network with multiple sources/destinations of traffic. Thus, a mix of critical and non-critical traffic will traverse the network, offering different loads across different interfaces. Every switch is configurable to operate a single, or mix, of the following queuing, scheduling or shaping techniques:

First In First Out (FIFO): An outgoing frame, through an egress port, is stored in a single queue. All queued frames are served in order of their arrival.

Priority Queuing (PRIO): An outgoing frame, through an egress port, is classified to be stored in a designated queue based on its priority. The Priority Code Point (PCP) is a 3-bit field in the VLAN (Virtual Local Area Network) tag, which is used to decide on the priority of the frame. The higher the PCP value, the higher the priority of a frame.

Time Aware shaping (TAS): Outgoing frames, stored in multiple queues, are controlled by a set of gates. Each gate is either ON, to allow a frame to be transmitted, or OFF to block a transmission from that queue. Gates are controlled by a schedule called the Gate Control Lists.

Credit Based Shaping (CBS): Outgoing frames, of a single queue, are controlled by the queue credit. Transmitting frames leads to spending credit with a *transmission slope* rate, while a queue obtains credit, with an *idle slope* rate, by being idle and waiting for other queues to finish their transmissions. A queue stops transmitting when it has negative credit and may start transmitting again when it has zero or more credit.

End systems are used to run the implemented application models. Each end system generates different types of traffic to perform the quality checks as previously stated in chapter 2. We classify the end systems as follows:

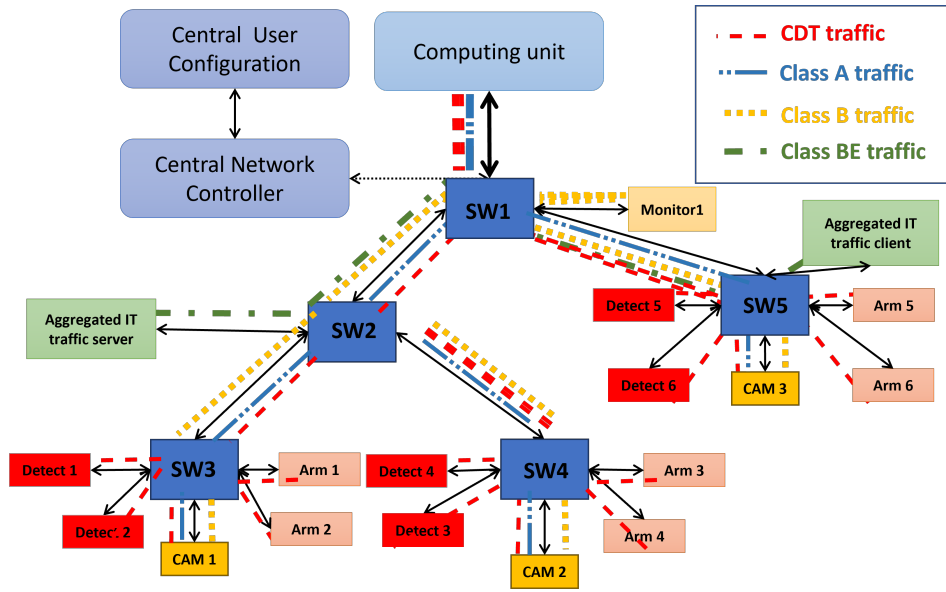


Figure 6.1: Network structure for QCAP in OMNET++.

1. Detection sensors: six units are deployed and used to perform the detection part of QCAP.
2. Cameras: three units are deployed and used to perform the inspection part of QCAP. In addition, they perform live production floor monitoring.
3. Edge unit: single central unit that commands the inspection and classification units to perform their QCAP operations.
4. Monitoring: single unit that display all the streamed data flows from the monitoring cameras.
5. Robotic arms: six units are deployed to execute an action received from an Edge unit.
6. Web source/sink: Client/Server web traffic over the network.

The data flows generated by those end systems are summarised in table 6.1. d_{max} represents the maximum allowed end-to-end delay that the frames of these flows can sustain, [LNS⁺15].

6.1.2 Traffic models characterisation

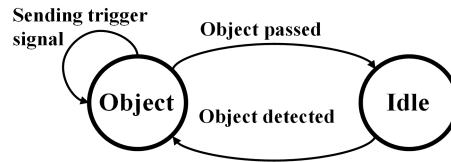
The network hosts produce four types of traffic (see Tables 6.2 and 6.3), which are characterised as follows:

Control Data Traffic: Product detection signals are critical traffic. It requires

Table 6.1: QCAP Data Flow Characterisations.

flow	type	Source	Sink	d_{max}
$f_1 \dots f_6$	CDT	Sensors	Edge unit	$100 \mu s$
f_7, f_8, f_9	Class A	Cameras	Edge unit	60 ms
f_{10}, f_{11}, f_{12}	Class B	Cameras	Monitoring	70 ms
f_{13}	Class BE.	Web Source	Web Sink	-
$f_{14} \dots f_{19}$	CDT	Edge unit	R.Arms	$100 \mu s$
f_{20}, f_{21}, f_{22}	Class A	Edge unit	Cameras	60 ms
f_{23}, f_{24}, f_{25}	Class B	Monitoring	Cameras	70 ms
f_{26}	Class BE.	Web Source	Web Sink	-

determinism as well as the least delay ($\sim 50 - 100 \mu s$), failing to do so will lead to a failure in the overall process of the product quality checking. The following finite state machine is used to model this traffic:



A production line with an output rate of n units per hour, each unit is of a length L . The production line conveyor is of a speed S . The detection unit remains in idle state till a new product arrives, every $3600/n$ second, then it changes to object state where it notifies the edge unit before it goes back to idle state once the object has passed. The traffic parameters are as follows: 1) Packet size = 174 bytes (128 payloads + 46 headers), 2) Packet interval = $500 \mu s$. Thus, the data rate is 2.8 Mbps while in object state and drops to zero in the idle state. $t_{detect} = \frac{L}{S}$, $t_{idle} = \frac{3600 \cdot S - L \cdot n}{n \cdot S}$ define the transmission and idle periods. Additionally, the commands issued by the edge unit to instigate the visual inspection or to control the robotic arms are critical. All the critical

Table 6.2: Traffic model characterisations (CDT, Class A, Class B).

	PCP	Packet Size (Bytes)	Packet Interval (ms)	[Idle, Active]	Max. data rate (Mbps)	Max. delay
CDT	6	174	0.5	$[t_{idle}, t_{detect}]$	2.8	$100 \mu s$
Class A	5	$\alpha = 1.1, L = 180, H = 558$	$\alpha = 1.1, L = 0.125, H = 0.5$	$[(3600 - 2n)/n, 2]$	35.7	60 ms
Class B	3	$\alpha = 1.2, L = 180, H = 1246$	$\alpha = 1.2, L = 0.5, H = 5$	Always active	19.94	70 ms

Table 6.3: Traffic model characterisation for web traffic.

		Class Best Effort	
		Browsing	File Transfer
Total Size	Main object size Embedded object size Embedded objects per page	$\mu = 8.37, \sigma = 1.37, a = 100 \text{ B}, b = 2 \text{ MB}$ $\mu = 6.17, \sigma = 2.36, a = 50 \text{ B}, b = 2 \text{ MB}$ $\alpha = 1.1, L = 2, H = 55$	$\mu = 14.45, \sigma = 0.35, a = 100 \text{ B}, b = 5 \text{ MB}$ - -
Timing	Reading time Parsing time	$\lambda = 0.033$ $\lambda = 7.69$	$\lambda = 0.006$ -

traffic has the highest Priority Code Point value 6, which is included in the VLAN tag of the transmitted frames from the critical traffic sources. Therefore, when configured, the network devices identify those and moreover process them with the highest priority.

Class A Traffic: inspection recordings of the targeted product, which are transmitted when the inspection unit receives a trigger from the edge unit. This type of traffic requires high bandwidth, guarantees on delivery as well as deterministic latency. It is important for the edge unit to complete the product quality checking at a specific time. Thus, the robotic arm can classify the product based on the check results. To model this traffic, dependency on the event of product arrival is considered, which leads to burstiness in the outgoing traffic from the inspection unit. The following parameters define the traffic nature:

1. inter-arrival time between frames = 50 ms,
2. number of packets per frame = 8,
3. packet size = Truncated Pareto distribution with $\alpha = 1.1$, $L = 180$ bytes, $H = 558$ bytes,
4. packet interval = Truncated Pareto distribution: $\alpha = 1.1$, $L = 125\mu s$, $H = 500\mu s$, and
5. burst duration = 2.0 seconds.
6. high priority traffic (PCP = 5).

Class B Traffic: video streaming service of the floor using the cameras, which are embedded in the inspection units. All the class B traffic has the same destination that is the monitor unit. Each camera produces 24 frames per second with 1280 x 720 resolution, MJPEG compressed format, and triple stream support (RTP/RTSP, Unicast RTP, Multicast RTP). To model this traffic, we consider that a consistent great image quality and robustness are required. While such characteristics requires high bandwidth and storage demands, it will be a realistic stress test for the backbone network performance. The following parameters define the traffic nature:

1. inter-arrival time between frames = 40 ms,
2. number of packets per frame = 8,
3. packet size = Truncated Pareto distribution with $\alpha = 1.2$, $L = 20$ bytes, $H = 1246$ bytes,

4. packet interval = Truncated Pareto distribution with $\alpha = 1.1$, $L = 500\mu s$, $H = 5ms$, and
5. priority traffic (PCP = 3).

Class Best Effort Traffic: information technology traffic, non-critical flows, that coexist with the critical industrial traffic. BE traffic has priority code point set to zero (PCP = 0), which will be transmitted in case a free slot exists. In addition, when a credit-based shaper is applied, the shaped traffic gain credits upon successfully transmitting BE traffic. Two sources of data flows are considered to model this traffic:

Web browsing: Client unit issues n requests per session to model HTTP 1.1 with keep alive behaviour, where n is a random variable that has exponential distribution with $\lambda = 5$. For each request, the HTTP server provides a reply that has a total size equal to the main size, main object size, plus the total size of the embedded objects in a requested page, equals to embedded object size times embedded objects per page. Those parameters [NORDS⁺20] are random variables with the following distribution characteristic:

1. main object size follows a truncated log normal distribution with mean of 11 KB, standard deviation of 26 KB, minimum size of 100 bytes, and maximum size of 2 MB.
2. embedded object size follows a truncated log normal distribution with mean of 7.75 KB, standard deviation of 125.2 KB, minimum size of 50 bytes, and maximum size of 2 MB.
3. embedded objects per page follows a truncated Pareto distribution: $\alpha = 1.1$, $L = 2$, $H = 55$. In addition, reading time, which is time gap between requests, and, parsing time, which is time gap between sessions, are random variables with exponential distribution function ($\lambda = 0.033$ $\lambda = 7.69$ respectively).

File transfer: Client unit issues an FTP request to fetch a data file from the FTP server with the following parameters:

1. File size (main object size) follows a truncated log normal distribution with a mean of 2 MB, standard deviation of 0.724 MB, minimum size of 100 bytes, and maximum size of 5 MB.
2. reading time is exponentially distributed, with an expected rate of occurrences = 0.006.

6.1.3 Simulation model implementation

We implement six applications to model the behaviour of QCAP, which are as follows:

UDP burst source: modelled by a two-state machine, with states *sleep* and *burst*. While in *burst* state, the application generates a network packet every transmission interval t_{send} that is of a size equal to pkt_{size} . The application switches to *sleep* state after the burst duration ends, and reverts back to *burst* after time t_{sleep} . Pseudocode 1 depicts the main procedures to implement this application. The generated packets are sent through a UDP socket, with respective IP address and port numbers of a destination.

UDP sink: models a User Datagram Protocol (UDP) server that listens all the time to configured ports. Once a packet is received, it processes its content and initiates the appropriate procedure.

Video streaming client: models UDP client. It transmits a single request to a unique destination address and port number, and it expects to receive the streamed data from the unique address. Request re-transmission occurs when the server is not receiving data for a period of t_{idle} .

Video streaming server: implements UDP server process. The UDP server receives a request for data streaming and replies with a stream of packets of a configured file size according to the Class B traffic model in Section II.

TCP client: models the client process of a generic TCP request/response service using sessions. It opens a TCP connection to a configured destination address, then transmits several requests. The connection remains open until the complete data arrives before sending a new request.

TCP server: models the server process of a generic TCP request/response service using sessions. It listens to a configured port and expects to receive connection requests. After processing a request, it replies with the same message but with a different length according to the request. Replies could be delayed by a configured constant time.

Each end system of the network, in Figure 6.1, runs single or multiple instances of those applications, which results in the data flows described in Table 6.1.

Pseudocode 1 UDP burst source

Input: start time = t_{start} , stop time = t_{stop} , burst duration = t_{burst} , sleep duration = t_{sleep} , transmission interval = t_{send} , destination address = $addr_{dst}$, destination port = $port_{dst}$.

Output: packet send = pkt_{send} , Number of sent packets = $nPKT_{sent}$

Require: $t_{start}, t_{stop}, t_{send}, t_{burst}, t_{sleep} \geq 0$
& $t_{stop} \geq t_{start}$

procedure START()

$nxtSleep = simTime()$;

$nxtBurst = simTime()$;

$nxtPkt = simTime()$;

$actvBurst = True$;

$timer \leftarrow SEND$

 GenerateBurst();

end procedure

procedure GENERATE BURST($t_{stop}, t_{send}, t_{burst}, t_{sleep}$)

if $simTime() \leq t_{stop}$ **then**

$now \leftarrow simTime()$;

if $nxtPkt < now$ **then**

$nxtPkt \leftarrow now$;

end if

$nxtPkt += t_{send}$

if $actvBurst \ \& \ nxtBurst \leq now$ **then**

if $t_{burst} == 0$ **then**

$actvBurst = false$;

else

$nxtSleep = now + t_{burst}$;

$nxtBurst = nxtSleep + t_{sleep}$;

end if

end if

$pkt_{send}(addr_{dst}, port_{dst})$;

$nPKT_{sent} ++$;

if $actvBurst \ \& \ nxtPkt \geq t_{stop}$ **then**

$nxtPkt = nxtBurst$;

end if

if $t_{stop} \geq 0 \ \& \ nxtPkt \geq t_{stop}$ **then**

$timer \leftarrow STOP$

$nxtPkt = t_{stop}$;

end if

 ScheduleAt($nxtPkt, timer$);

end if

end procedure

6.1.4 Results and discussion

The section evaluates the proposed framework and simulations by running and investigating different network and traffic configurations. The parameters listed in Table 6.4 are used for configurations.

Table 6.4: Network and simulation parameters

Parameter	value
Simulation time	90 s
Processing delay of a switch	1 μ s
Network link speed	100 Mbps
Propagation delay	0.05 μ s

To thoroughly evaluate the different network configurations and assess which is the optimal one, we simulate three different test cases. First, the worst case scenario is considered where all flow sources are transmitting with the maximum data rate. Therefore, the state of an overloaded network could be assessed. We consider two scenarios here, one with infinite queue sizes, and the other with limited queue size of maximum 1000 packets. Both cases consider deterministic traffic, which ease the computation of transmission schedules. For the third test case, a stochastic traffic models are considered. The network performance is assessed in terms of end-to-end delay and delivery ratio for different network switch configurations. In addition, this evaluation includes the behavioural analysis of the overloaded network bottleneck link, which is the link between the root switch and the left switch as most of the data flows pass through it. We investigate the lengths of the queues and the queuing times for this link.

Figures 6.2 and 6.3 illustrate the effects of different queue management strategies on network traffic. Under the FIFO (First-In, First-Out) configuration, all types of traffic are processed in the order they arrive, regardless of their priority. This results in all traffic types sharing the same queue, leading to uniform queuing delays. Therefore, high-priority and low-priority packets experience the same delay because they are handled in a single, undifferentiated queue.

In contrast, the PRIO (Priority) configuration uses multiple queues, each assigned a different priority level. High-priority traffic, such as real-time communications or critical data, is placed in higher-priority queues, which are processed more quickly. As a result, these high-priority packets experience shorter queue lengths and reduced queuing times. Conversely, lower-priority traffic, like monitoring data, is placed in lower-priority queues. These queues can become congested if high-priority traffic floods the network, leading to longer

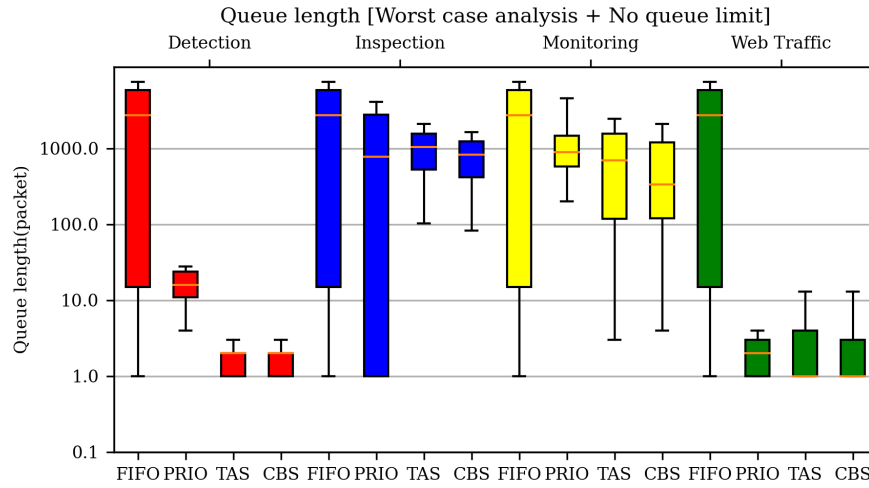


Figure 6.2: Queue Length distribution for Worst Case with infinite queue size configuration (Deterministic traffic)

queue lengths and increased delays for lower-priority packets.

The figures also reveal that when Class A traffic overwhelms the network, it can severely impact the performance of Class B traffic. Class A traffic's excessive volume fills up the network's capacity, causing substantial delays and queue build-up for Class B traffic. For instance, if Class A traffic is too high, it blocks or delays the delivery of Class B traffic, which may include web responses. This congestion can prevent timely processing of web requests and responses, leading to a decrease in the number of requests made by web clients. Thus, while PRIO configuration improves handling of high-priority traffic, it cannot completely mitigate the negative effects of excessive lower-priority traffic, highlighting the challenges of balancing different traffic types effectively.

Next is the end-to-end delay assessment, Figure 6.4, for the different types of data flows, against the different network configurations. As expected, all different configurations, except FIFO, provide within range acceptable delay for the detection system. However, the other types of traffic experience higher delays than the maximum allowed. The delay becomes significantly high for monitoring traffic, Class B, when PRIO configurations are used. Finally, Figure 6.5 shows the measured number of delivered packets. The different network configurations manage to achieve 100% delivery ratio for the critical, class A, and class B traffic. However, the number of delivered packets of the web traffic, class BE, is of interest, as it indicates the capability of the network to handle the coexistence of both operational technology and information technology traffics. The FIFO configuration delivers the highest number of best effort packets, but

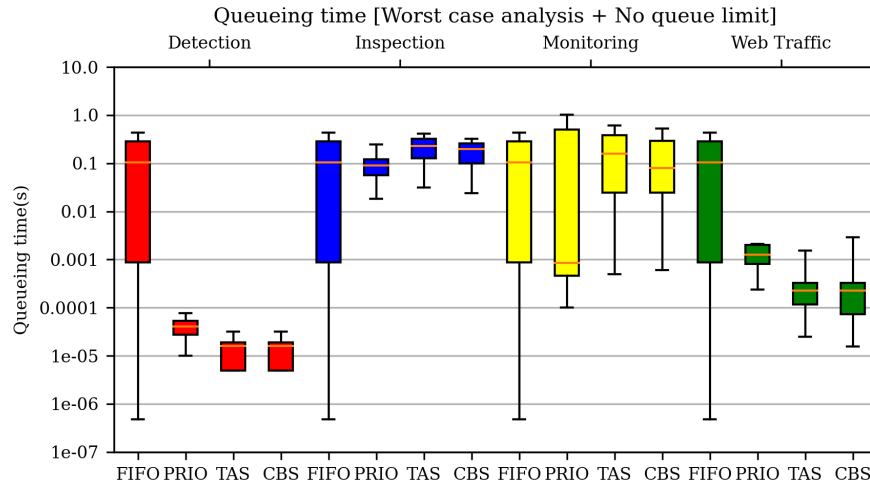


Figure 6.3: Queueing Time distribution for Worst Case with infinite queue size configuration (Deterministic traffic)

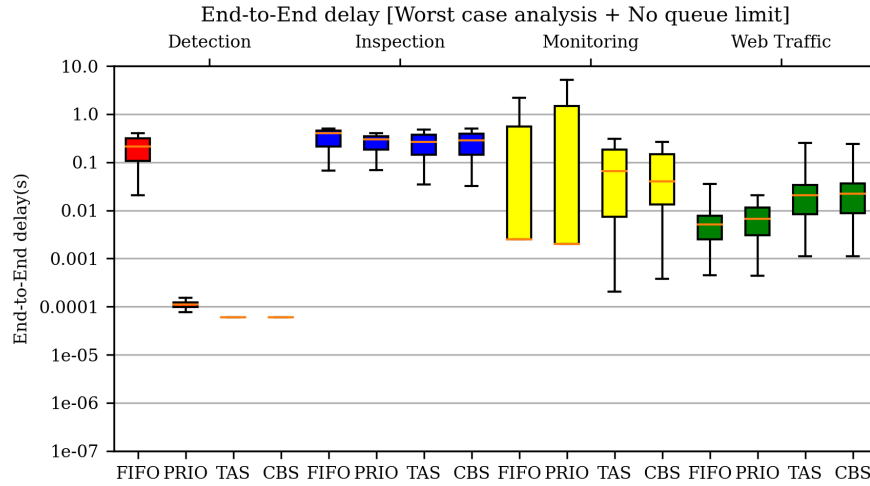


Figure 6.4: End-to-end delay distribution for Worst Case with infinite queue size configuration (Deterministic traffic)

at the expense of end-to-end delay. In contrast, PRIO and TAS configurations deliver 90.6% of packets, which happens due to oppressing or delaying web client requests by higher priority traffic or by controlling gates with a priority transmission schedule. The CBS configuration performs better by successfully delivering 96% packets, compared to FIFO, of the web traffic. This is due to the credit shaping where class A and class B have to gain credit before transmitting, which allows class BE traffic to be successfully transmitted.

In the second test case, a limited queue size for the egress ports of all the network switches is configured. Queues are configured with a drop tail mechanism. First, Figures 6.6 and 6.7 depict the distribution of the queue length and

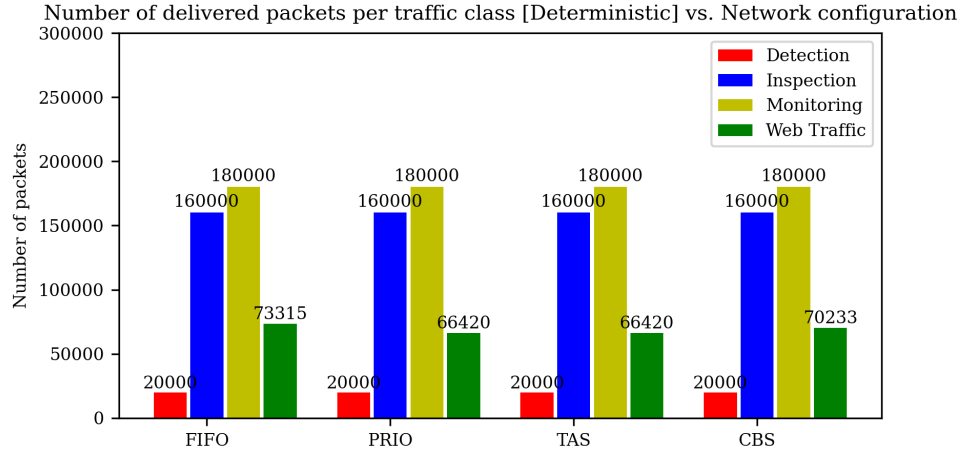


Figure 6.5: Delivery ratio distribution for Worst Case with infinite queue size configuration (Deterministic traffic)

Table 6.5: The average end-to-end delay and delivery ratio of QCAP simulation results (Standard Ethernet [FIFO]).

	Standard Ethernet [FIFO]	
	Avg. end-to-end delay (s)	Delivery ratio (%)
CDT	75×10^{-3}	73
Class A	79×10^{-3}	82.8
Class B	13×10^{-3}	99.1
Class BE	78×10^{-3}	92.1

queuing time over the simulated period. The PRIO configuration significantly decreases the queuing time for the critical traffic, same for TAS and CBS, when compared to FIFO, which is a good indication that they meet the QCAP's deadlines requirements. However, PRIO performance drops compared to TAS and CBS while handling lower priority traffic. We assess the overall performance with end-to-end delay and delivery ratio measurements as depicted in Figs. 6.8 and 6.9 and summarised in Tables 6.5, 6.6, 6.7, and 6.8. While PRIO prioritises the critical traffic, it could not meet the deadline requirements. In contrast, both TSN configurations provide deterministic upper bounds that are below the QCAP deadlines. Moreover, using only the PRIO configuration, for the monitoring system, a higher end-to-end delay is shown. This is due to the bursts in class A traffic, yet TAS and CBS configurations provide better control over those and therefore demonstrate overall better end-to-end delay for all the subsystems. In addition, CBS provides the highest delivery ratios for all the subsystems and the web traffic.

For the third test case, the system is assessed with realistic stochastic traffic

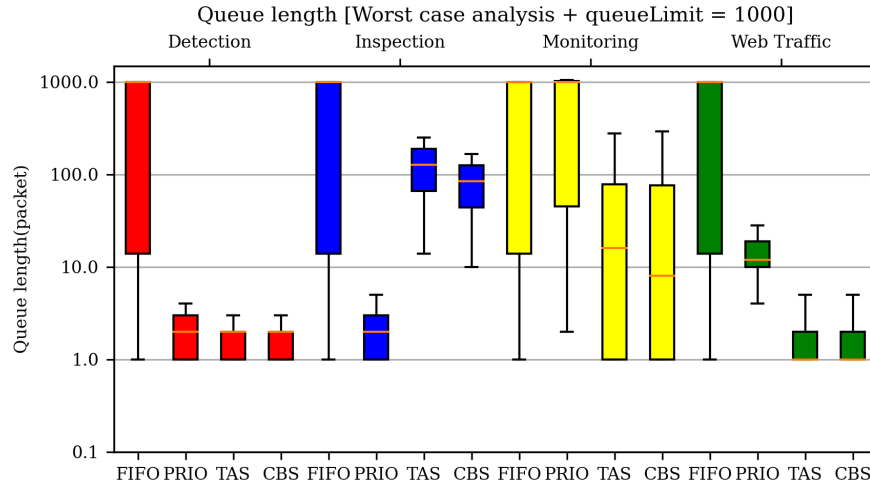


Figure 6.6: Queue Length distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)

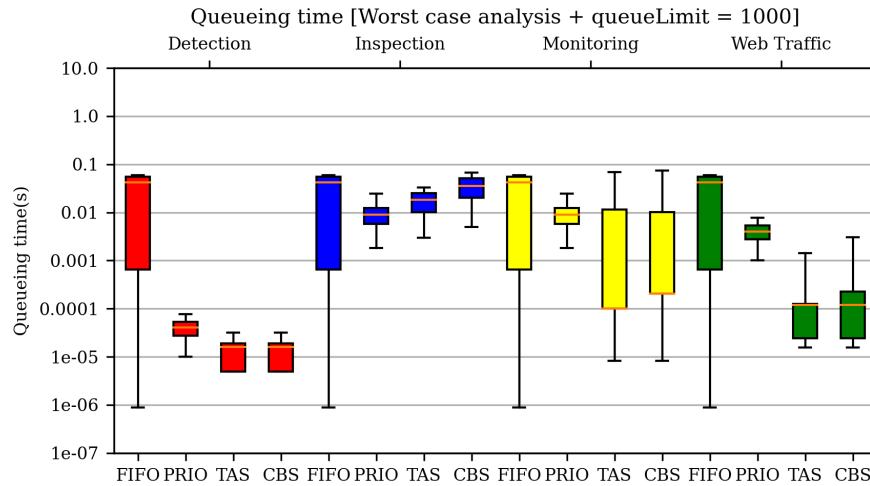


Figure 6.7: Queuing Time distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)

Table 6.6: The average end-to-end delay and delivery ratio of QCAP simulation results (Standard Ethernet [PRIO]).

	Standard Ethernet [PRIO]	
	Avg. end-to-end delay (s)	Delivery ratio (%)
CDT	111×10^{-6}	100
Class A	49×10^{-3}	88.7
Class B	61×10^{-3}	92.6
Class BE	10×10^{-3}	90.6

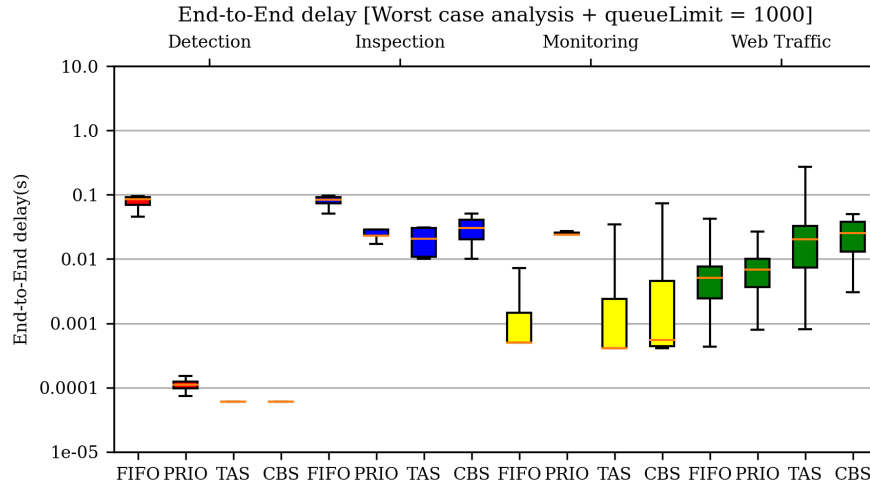


Figure 6.8: End-to-end delay distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)

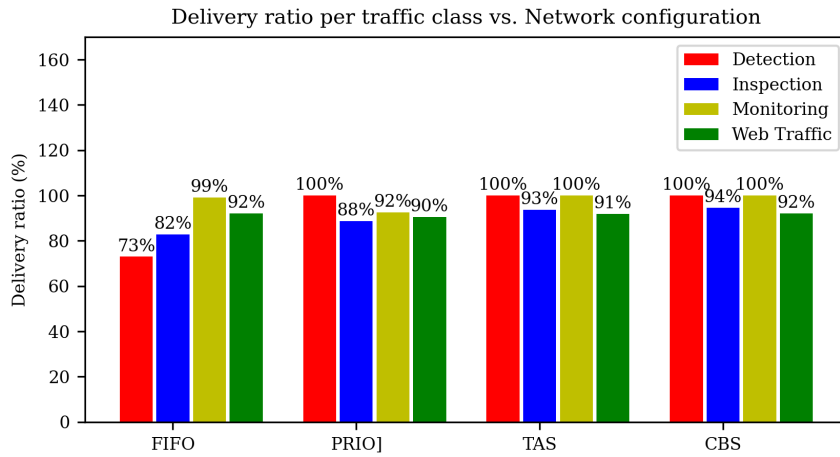


Figure 6.9: Delivery ratio distribution for Worst Case with limited queue size (deterministic traffic, queue size = 1000)

Table 6.7: The average end-to-end delay and delivery ratio of QCAP simulation results (TSN [TAS]).

	TSN [TAS]	
	Avg. end-to-end delay (s)	Delivery ratio (%)
CDT	61×10^{-6}	100
Class A	35×10^{-3}	93.7
Class B	5.7×10^{-3}	100
Class BE	24.5×10^{-3}	91.9

Table 6.8: The average end-to-end delay and delivery ratio of QCAP simulation results (TSN [TAS+CBS]).

	TSN [TAS+CBS]	
	Avg. end-to-end delay (s)	Delivery ratio (%)
CDT	61×10^{-6}	100
Class A	48×10^{-3}	94.6
Class B	6.5×10^{-3}	100
Class BE	24×10^{-3}	92.1

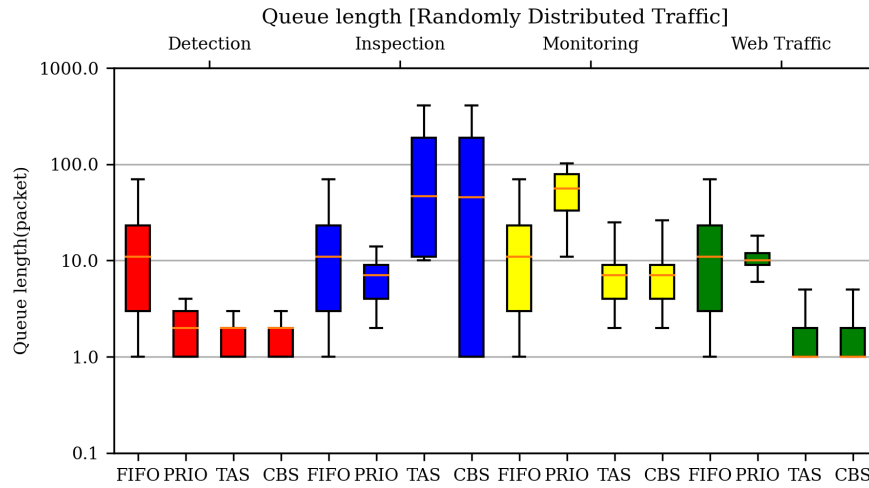


Figure 6.10: Queue Length distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)

models to measure the resilience of the TAS and CBS configurations against the random changes in the data flows behaviours. Figure 6.10 depicts the queue length distribution for the TAS and CBS. We can see that TAS and CBS outperform the other configurations, except for the inspection process. This is due to the bursty nature of class A traffic. Figure 6.11 shows the queuing times at the congested link for the different system processes. For TAS and CBS, detection traffic is below $100\mu s$, inspection, monitoring and web traffics are below 100ms. Finally, Figures 6.12 and 6.13 present the two main measures, which concludes that using TAS and CBS guarantee reliable delivery for the critical traffic with a deterministic end-to end latency. In addition, they provide better utilisation for the network resources by allowing web traffic to coexist without affecting the QoS of the critical traffic.

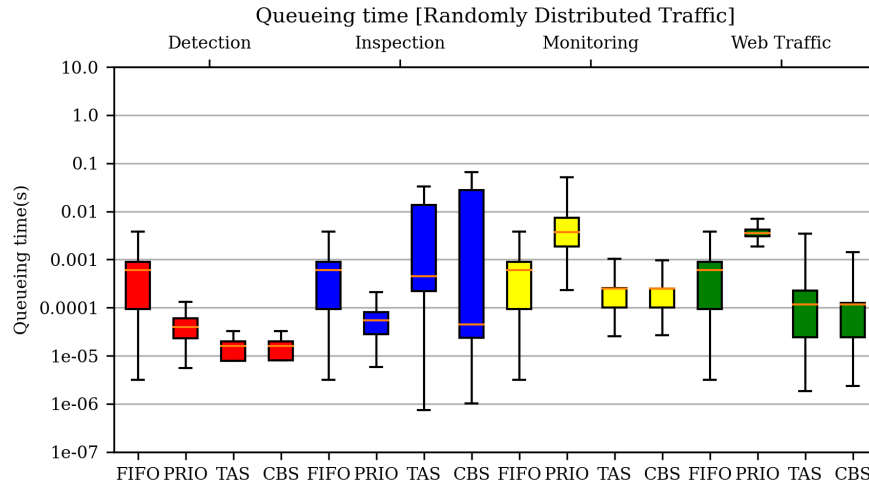


Figure 6.11: Queueing Time distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)

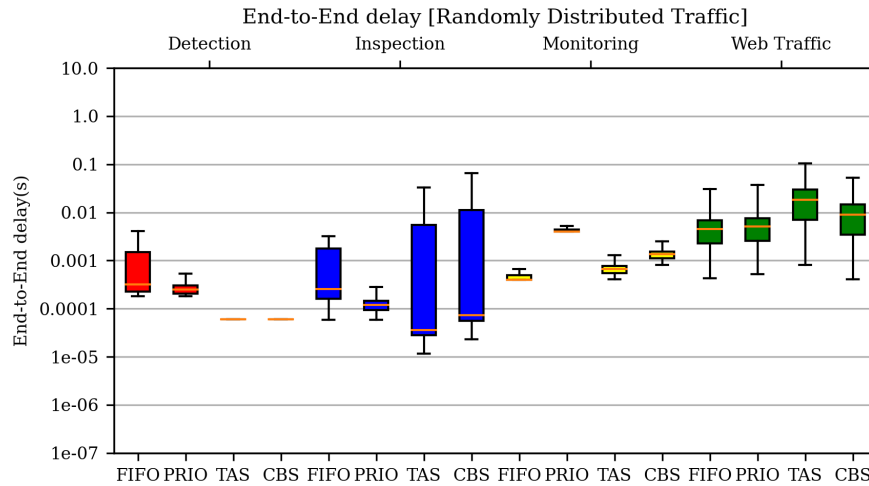


Figure 6.12: End-to-end delay distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)

6.2 Validation and Performance Evaluation of the TSN based industrial networks timing analysis Framework

6.2.1 Network Model

We consider a two-level tree topology of the single hop model of a QCAP node, depicted in Figure 4.1. The QCAP topology consists of three QCAP nodes, five switches, and four Ethernet devices as represented in Figure 6.14. The breakdown of QCAP node components is:

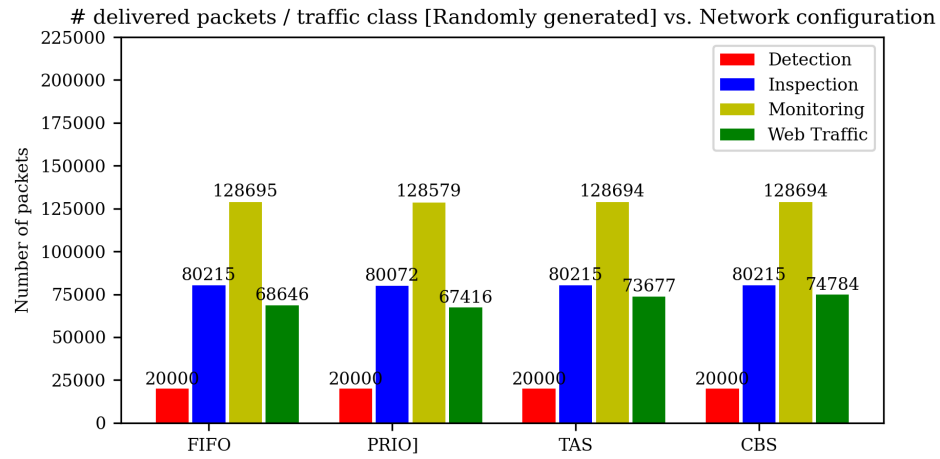


Figure 6.13: Delivery ratio distribution for Worst Case with limited queue size (randomly distributed traffic, queue size = 1000)

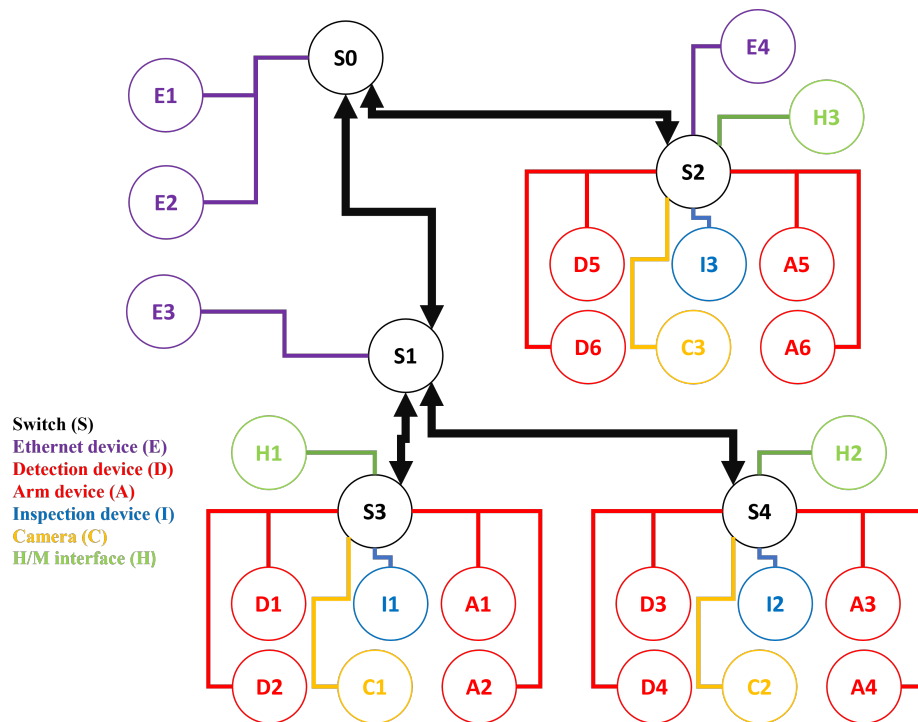


Figure 6.14: Representative QCAP network topology

Table 6.9: Data flow characterisations

Flow: source, sink	Type	Priority (PCP) ¹	Deadline
$f_1 \dots f_6 : D_i, E_1$ $f_7 \dots f_{12} : E_1, A_i$	CDT/ST	7	$100\mu s$
$f_{13} \dots f_{15} : I_i, E_1$ $f_{16} \dots f_{18} : E_1, I_i$	class A	5	$60ms$
$f_{19} \dots f_{21} : C_i, E_2$ $f_{22} \dots f_{24} : E_2, C_i$	class B	3	$80ms$
$f_{25} \dots f_{27} : H_i, E_2$ $f_{28} \dots f_{30} : E_2, H_i$ $f_{31} : E_3, E_4$ $f_{32} : E_4, E_3$	B.E.	0	-

- Detection device (D): a sensing unit to detect the arrival of a new product for quality checking.
- Inspection device (I): a visual inspection unit to capture images of a detected product and submit those to make a decision on the quality of the product.
- Arm device (A): robotic arm used to remove a product not meeting the quality check from the production line.
- Camera (C): visual unit to real time monitor the floor area.
- H/M interface (H): Human machine interface to monitor and configure the operating devices of a QCAP node.
- Ethernet device (E): An Ethernet connected computing device that works as controller, computing unit, or monitoring interface for the industrial application.

The characterisation of the data traffic flows are summarised in Table 6.9. Generally, data exchanged as part of the control loop, e.g. direct communication between sensors, controllers and actuators, holds the highest priority, i.e. scheduled data traffic. The inspection unit produces class A traffic and the cameras produce class B traffic as detailed in Chapter 4.

6.2.2 Model Implementation

For this evaluation, an implementation of the aforementioned models were introduced using two approaches, OMNeT++ based discrete event simulation and Real-Time Calculus (RTC) toolbox based implementation of the developed

network calculus models. First, the worst-case end-to-end delay bounds are obtained using the following implementations:

1. OMNeT++ based simulation framework for QCAP use case, as introduced in 6.1.
2. RTC [WT06] based implementation of the presented industrial TSN/NC based WCD analysis, Section 4.
3. RTC based implementation of Credit Based Shaping (CBS) + Strict Priority (SP) for industrial automation using NC [ZCWW19].

Second, the proposed industrial TSN/NC-based implementation is used to evaluate whether the QCAP network performance meets the different data flow QoS requirements. For this evaluation, a typical network configurations and data traffic parameters for industrial applications are used, those could be summarised as follows:

- Network link capacity (C): 100 Mbps or 1 Gbps.
- Maximum processing delay of a switch: $1 \mu s$
- Maximum buffer size of a switch: 1000 frames.
- Propagation delay: $1 \mu s / 100m$
- Maximum frame size: Scheduled (L_i) = 128 B, class A (L_A) = 1500 B, class B (L_B) = 1500 B.
- Minimum cycle time/duration between frames: $T_{ST} = 500 \mu s$, $T_A = 1 ms$, $T_B = 500 \mu s$.
- Maximum data rate: r_{ST} : 2.05 Mbps, $r_A = 12 Mbps$, $r_B = 24 Mbps$.
- Idle slope: $I_A = 40\%$, $I_B = 20\%$ of the link capacity.
- Maximum burst duration: $\sigma_A = 2 sec$, $\sigma_B = 4 sec$.

6.2.3 Evaluation and Results

The evaluation focuses on obtaining and comparing the maximum value (i.e. the worst case) of the end-to end delay for each flow using the three approaches described above. Firstly, we simulate the behaviour of the QCAP network model using OMNeT++. The model includes 32 flows, as highlighted in Table 6.9. Next, implementations of both the developed industrial TSN/NC model and

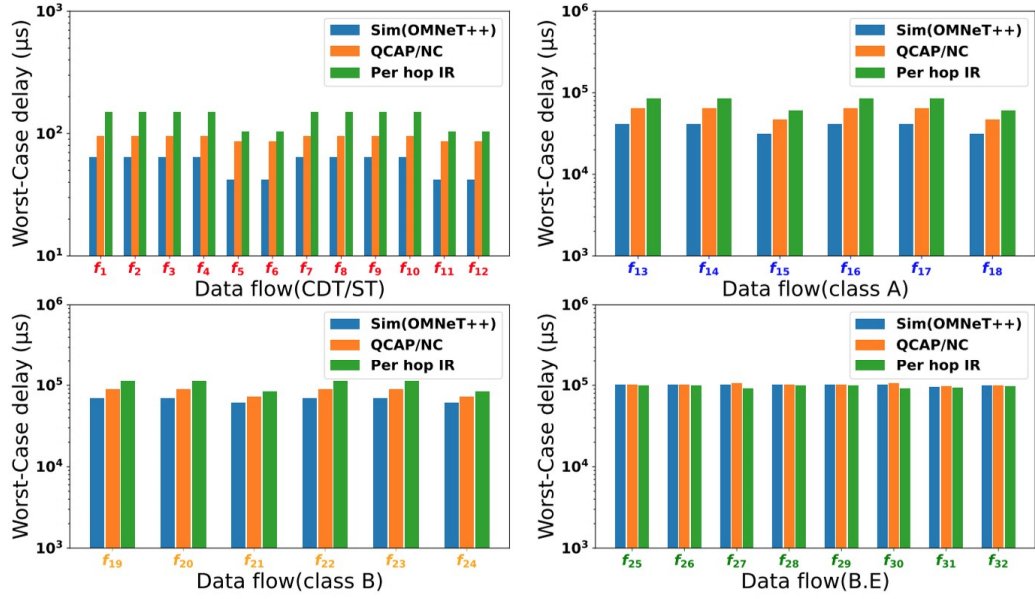


Figure 6.15: Comparison between worst-case delay bounds for QCAP data flows

the per hop Interleaved Regulator (IR) NC-based model [ZCWW19] are used for comparison.

Figure 6.15 compares the delay bounds obtained by the three aforementioned methods. Figure 6.15 has four sub-plots with classified data flows, one plot for each data traffic class. Our NC model outperforms the other NC model by providing tighter delay bounds. This is expected, considering that the per hop IR model does not consider the pre-emption feature of the scheduled traffic. Additionally, it does not consider the relation between the best effort traffic and the credit accumulation of both class A and class B traffic. While the developed industrial TSN/NC framework exceeds the worst-case delay results when compared to the simulation model, its run-time is 10x faster than the network simulator with statistically safe results. Therefore, the developed TSN/NC framework can be used as a first stage assessment for the design of TSN-based industrial automation applications such as the QCAP use case.

Next, focusing on measuring the QCAP network performance for different configurations using the developed TSN/NC framework. For that, we start with relaxing the network configurations by choosing larger bandwidth (1 Gbps) links between switches. In addition, the transmission rates were doubled for

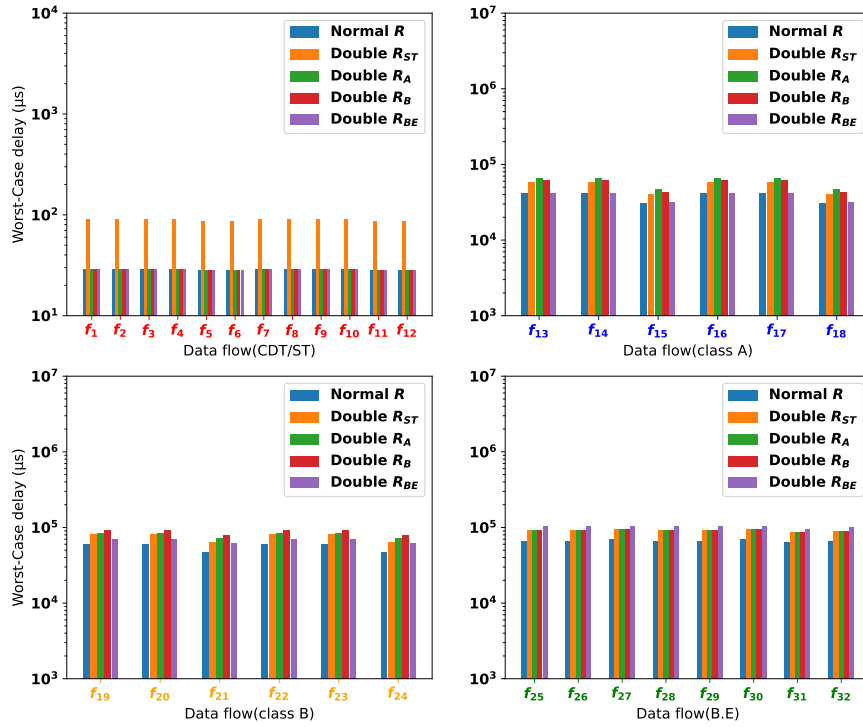


Figure 6.16: Comparison between worst-case delay bounds for different flow rate

each set of flows and then observe the impact of this increase on the Worst Case Delay for all data flows. Figure 6.16 depicts the WCD bounds for the four different sets of flows of this experiment, where we observe the following:

- For scheduled flows (CDT/ST traffic), the WCD bound remains almost the same when doubling the other data flow rates. However, it only slightly changes when doubling the rates of the CDT/ST traffic itself. This is due to the high priority of flows belonging to this type of traffic, as well as the pre-emption mode, which allows these data flows to be processed faster when they collide with lower priority traffic.
- For AVB traffic (class A and class B), the WCD bound increases progressively when data rates of the same type of traffic or rates of traffic with higher priority increase/double. Furthermore, the WCD bound becomes progressively less when low priority data rates increase. This is due to the credit based and strict priority shaping functionality offered by TSN switches.

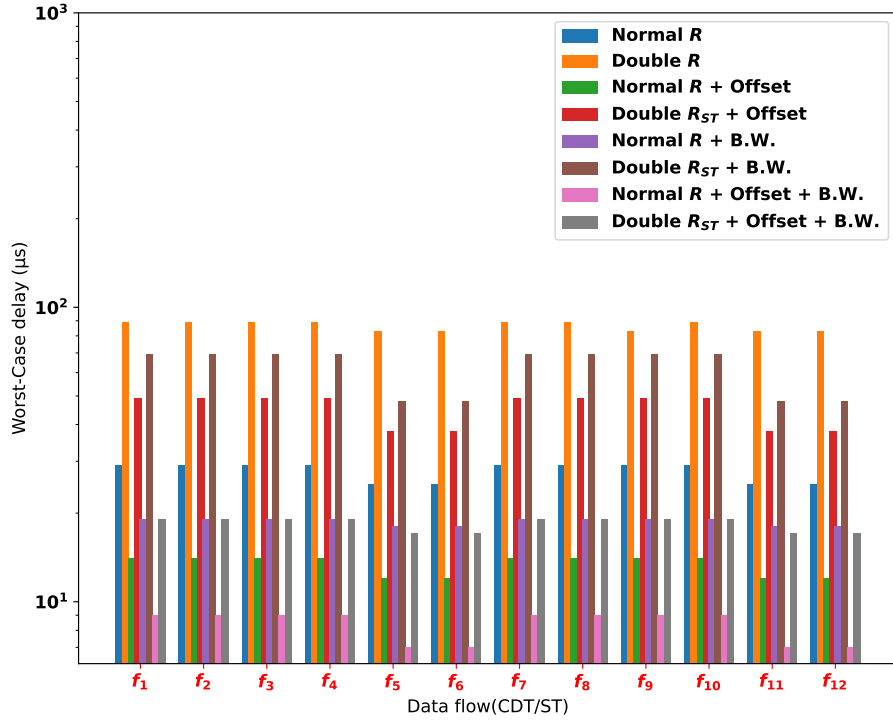


Figure 6.17: The effect of transmission offset and increasing the bandwidth of critical interfaces on the WCD of scheduled traffic.

- For best effort traffic, increasing the ST traffic rate leaves less bandwidth to be shared among the other three types of traffic, which increases the delay bound on the lowest priority traffic, e.g. BE class. However, increasing the data rates of AVB traffic creates more transmission opportunities for BE traffic while AVB traffic accumulates credit.

The previous experiments could be used to identify the WCD bounds for all traffic flows traversing the QCAP network. This analysis demonstrates the successful operation for each process of the QCAP use case can be guaranteed. Moreover, the proposed TSN/NC framework demonstrates lower WCD bounds on critical data traffic, allowing to accommodate more data flows if necessary. Figure 6.17 provides a study of two properties that impact the delay bounds of the critical traffic:

1. adding an offset period to the start of ST flow transmissions. This ensures that frames belonging to this traffic will arrive at different time instances, resulting in less queuing time.

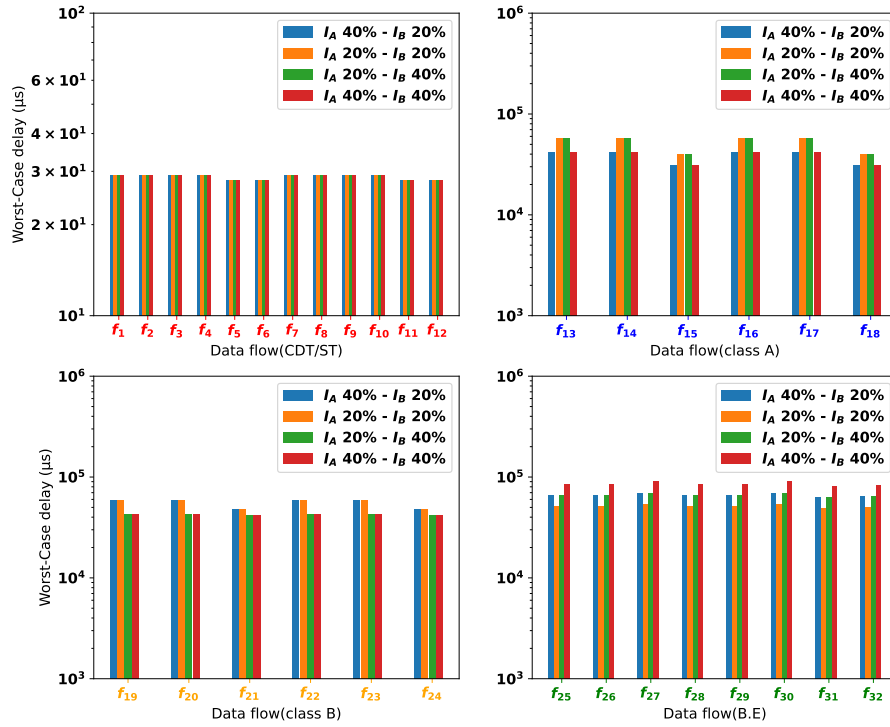


Figure 6.18: Comparison between different idle slopes on the WCD of different traffic types.

2. identifying critical links to be configured with larger bandwidth, hence reducing the transmission and propagation delay. Therefore, the overall WCD for ST flows is reduced.

In this context, critical links do not imply links between switches only, where large number of flows are expected, but also interfaces where multiple ST flows are waiting for transmission. For example, in the QCAP topology, the link between TSN switch one (S1) and TSN switch zero (S0) is a critical link as many flows are traversing this link. However, the link that connects S0 to the Computing unit/Ethernet device 1 (E1) is the interface where all ST flows are transmitted through as part of the detection and inspection QCAP processes. In summary, setting a different offset for the ST critical flows and increasing the bandwidth of critical links can result in a 91% reduction in the worst case end-to-end delay of these flows, as depicted in Figure 6.17.

Our final experiment targets the influence of AVB traffic classes idle slopes on the WCD bounds of the different data flows. We expect that the larger the

idle slope of an AVB class, the lower the WCD bound for this class of traffic. Moreover, the higher the WCD bound for the classes sharing bandwidth with this class of traffic. Figure 6.18 depicts the effect of varying idle slopes for class A and class B traffic. Starting with the normal case where $I_A = 40\%$ and $I_B = 20\%$, the idle slope of class A is set to 20% . As expected, the ST flows' WCD is not affected. However, class A flows' WCD bound increases by a factor of 2 and the same applies to class B flows due to the dependency between the two types of AVB flows. Moreover, reducing the idle rates leads to reducing the WCD bound on the best effort data flows, which can be expected as more BE frames can be transmitted while AVB flows are accumulating credit to transmit. On the contrary, increasing idle slopes of AVB classes to $I_A = 40\%$ and $I_B = 40\%$ leads to faster credit accumulation for both class A and B, which reflects on the reduced WCD bound of the flows that belong to these classes. In addition, increasing idle slopes of AVB classes utilises most of the available bandwidth, which leaves no room for best effort transmissions and hence increases its end-to-end-delay.

6.3 Experimental evaluation of the improved timing analysis of TSN based industrial networks

This section presents the validation of the derived delay bound and investigate the impact of routing, increasing traffic loads with different transmission priorities, credit accumulation rates, and redundancy and load distribution on the delay performance of TSN. Additionally, the proposed framework is used to analyse the performance of (*use case 07*) of a redundant industrial networks [G⁺21].

6.3.1 Experimental Evaluation Setup:

In this evaluation, we consider different network topologies. The carried traffic is split into eight different types of data traffic as described in Section 5.

The network parameters and configurations are summarised as follows:

- Links are full duplex, with a bandwidth $C = 1$ Gbps.
- Switch output port has 8 priority queues.
- Maximum processing delay of a switch: $t_{pr} = 1\mu s$.

Table 6.10: Data flows description.

Traffic class	Flow	Src.	Dest.	Flow count	Max. size (B)	Min. cycle / Min. frame duration (ms)	Max. rate ($Mbps$)	Max. burst duration (s)
Isochronous	$f_1, \dots, f_{10}$	C1	C4	10	128	0.5	2.05	n.a.
Non-isochronous	$f_1, \dots, f_{20}$	C1	C4	10	1000	2	4	n.a.
Class A	$f_{21}, \dots, f_{25}$	C2	C5	5	1500	1	12	4
Class B	$f_{26}, \dots, f_{30}$	C2	C5	5	1500	0.8	15	2
Network control	$f_{31}, \dots, f_{40}$	C3	C6	10	500	50	0.08	n.a.
Events	$f_{41}, \dots, f_{45}$	C3	C6	5	1500	n.a.	2	1
Configurations	$f_{46}, \dots, f_{50}$	C3	C6	5	1500	n.a.	2	1
Best effort	f_{51}, f_{52}	C3	C6	2	1500	n.a.	100	n.a.

- Maximum buffer size of a switch: 1000 packets.
- Propagation delay: 100 ns .
- Idle slope I_A : 20% of Link capacity
- Idle slope I_B : 10% of Link capacity

To ensure deterministic and real-time communication, the guidelines for computing the TSN network schedule, for industrial automation, are as follows:

- Isochronous and Non-isochronous flows are time-sensitive and require guaranteed real-time performance. Additionally, traffic characteristics of each identified flow are defined in Table 6.10.
- TAS is configured to adhere to the traffic profiles of each flow. Moreover, Stream Reservation Protocol is configured to ensure that essential traffic is allocated the necessary bandwidth.
- Parallel Redundancy Protocol is used to configure backup paths for critical flows to ensure network reliability.

6.3.2 Model Validation

The simulation worst case delay represent the reference for validating the derived bound. We first validate the tightness of the bounds using a basic network topology of two TSN switches S , as depicted in Figure 6.19. The traffic sources $C_1 - C_3$ generate n_i data flows for each traffic class i . The details of the traffic classes and data flows description is presented in Table 6.10.

We compare delay bound for different TSN traffic classes as derived by different NC frameworks [ZCWW19, MS17, ZPZL18, ZPZ⁺20]. Figure 6.20 plots the delay bound derived by these frameworks together with the worst case delay from the simulated network. The WCD bounds, calculated by the developed framework, denoted as impNC/TSN_industrial23, are tighter compared to the

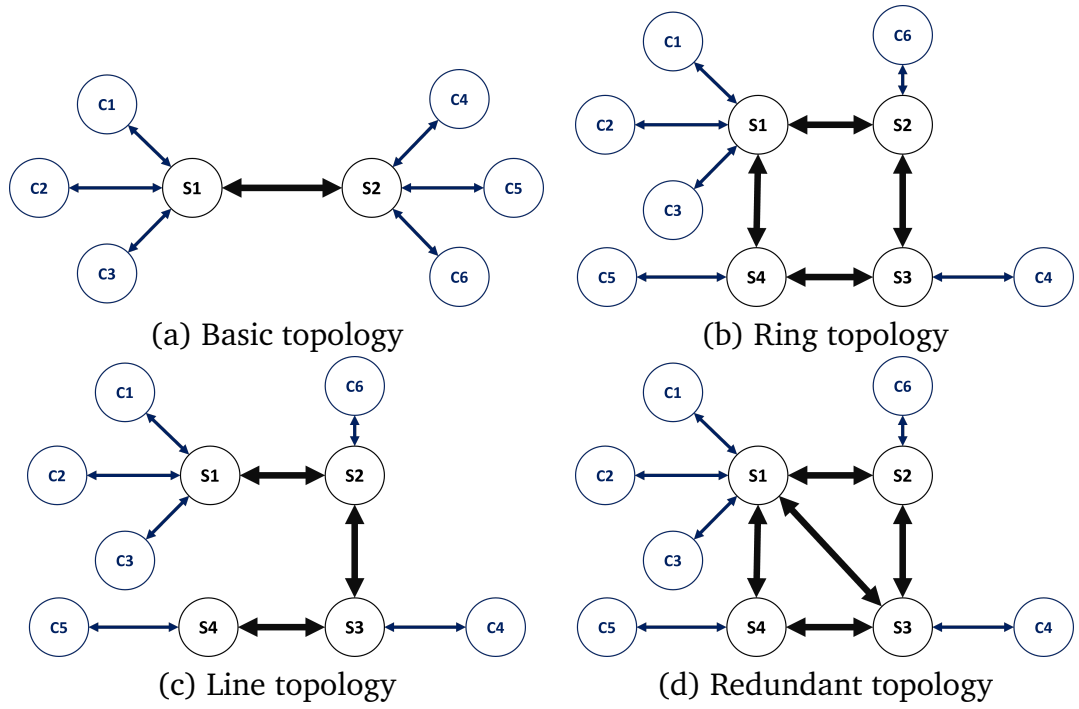


Figure 6.19: Synthetic test-cases.

other frameworks with respect to the obtained simulator results. NC/TSN17 [MS17] provides the loosest delay bound for all traffic classes, as it does not consider the cyclic real-time traffic properties and mainly focuses on the AVB traffic. NC/TSN_industrial19 [ZCWW19] offers tighter bounds as it considers the impact of credit and priority shaping for the co-existing industrial traffic. NC/TSN18 [ZPZL18] and its improved version impNC/TSN21 [ZPZ⁺20] offer a reduced WCD through introducing physical link and credit base shaping while considering the cyclic real-time traffic window. However, they do not consider the industrial profile of TSN traffic properties, therefore do not provide tighter WCD bounds for lower priority traffic compared to NC/TSN_industrial19. In contrast, the proposed NC based framework considers the industrial profile aspects, shaping methods, e.g. priority, credit and time, and additionally different integration modes, e.g. pre-emption. Therefore, It becomes more useful and deterministic for industrial application with safety guarantees.

6.3.3 Impact of Routing

We consider the impact of routing different traffic classes in the ring topology illustrated in Figure 6.19b. on the end-to-end WCD. We consider a ring topology, Figure 6.19b, with the same network configurations and parameters stated in Table 6.10. Flows of different traffic classes, entering through $S1$, can be

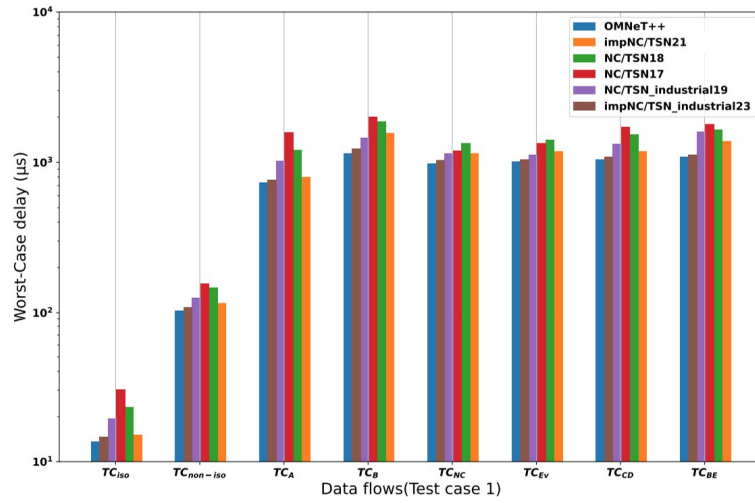


Figure 6.20: Comparison between WCD bounds for flow types using different NC based frameworks.

grouped as follows:

- Scheduled, i.e. iso and non-iso, and AVB traffic share the same interface, while BE traffic solely use the other interface.
- Scheduled and BE traffic share the same interface, while AVB traffic solely use the other interface.
- iso and AVB traffic share the same interface, while non-iso and BE share the other interface.
- non-iso and AVB traffic share the same interface, while iso and BE share the other interface.

Selecting an interface is based on achieving the shortest path for each flow. For iso and non-iso, the shortest path includes three hops, regardless of the selected interface. However, for other traffic classes, AVB and BE, the shortest path includes two hops. Figure 6.21 depicts the outcome of this setup. As shown, the WCD for the highest priority traffic class, iso flows, is not impacted regardless of the other traffic classes it is grouped with, which is expected. However, the WCD for non-iso traffic class slightly drops when grouped with any other traffic classes but the iso traffic. Therefore, distributing scheduled traffic across different paths yields lower WCD for the traffic flows. WCD for other traffic classes, AVB and BE, vary depending on the other grouped traffic class. Our analysis show that grouping iso traffic with BE traffic and grouping non-iso traffic with

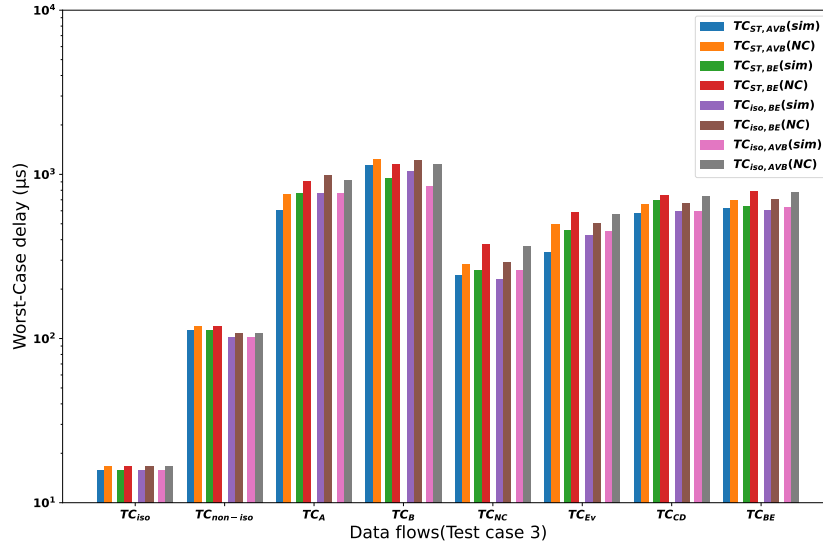


Figure 6.21: Traffic grouping and interface selection impact on the worst case delay for different traffic classes.

AVB achieve the optimised WCD bounds for all traffic classes. As transmitting iso frames is deterministic and synchronised with the network cycle, it won't much increase the WCD for BE frames transmitted across the same path. Moreover, the guard band of the non-iso traffic guarantees the minimum WCD for the non-iso data flows, while having the smallest impact on the AVB traffic traversing the same path. AVB frame that pulled out of the queue before the arrival of a guard, can be transmitted and the other AVB queue may accumulate credit, which ensures better utilisation and lower the WCD bound for the AVB traffic.

6.3.4 Impact of increasing traffic loads with different priorities

First, the impact of increasing the number of flows per traffic class on the end-to-end WCD of all traffic classes is analysed. For example, doubling the amount of cyclic real-time flows, iso and non-iso, and measure the WCD for all traffic classes. The same applies to doubling the AVB and BE traffic. For this experiment, a similar flows descriptions as in Table 6.10 is used. Also, a line topology is used, Figure 6.19.c, to avoid different paths' selection. This to ensure that all traffic classes traverse the same longest possible network path. Figure 6.22 depicts the results of this experiment. Increasing the number of the highest priority flows, i.e. iso traffic, results in a higher WCD bounds for all traffic classes.

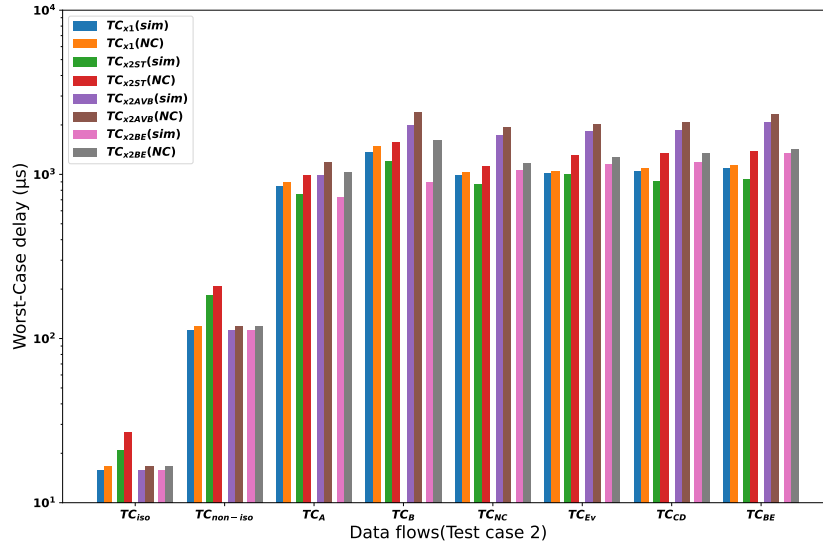


Figure 6.22: Delay bound for different number of flows for different traffic classes.

Frames belong to different traffic classes have to wait for an iso frame to be transmitted, including iso frames of other flows that share the same interface queue. To obtain an optimised bound for iso frames, a time offset will be added to ensure that the minimum number of iso frames exist in an interface queue at any instance, i.e. schedule optimisation. Similar behaviour for non-iso traffic is noticed, with a slight difference where increasing the number of non-iso flows won't impact iso flows WCD bound, this is due to pre-emption property of the highest priority traffic. For AVB traffic classes, it is noticed that increasing number of iso and non-iso flows slightly increase the WCD bound. In comparison, increasing the number of AVB flows has a large impact on the WCD of AVB and lower priority traffic classes. This is due to the different frame sizes, which relates to the consumed bandwidth as well as the transmission delay of the frames. Moreover, the processing speed of iso and non-iso frames with an objective of no-queuing while AVB flows experience longer queuing time. In addition, transmitting an AVB frame requires enough accumulated credit, leading to longer processing time and therefore larger delay. At last, increasing the number of BE flows not only increases WCD bounds for lower priority traffic but also increases WCD bounds for AVB traffic. This is due to the pessimism aspect of assuming that BE frame gets transmitted right before an AVB queue accumulates enough credit to transmit, leading to larger queuing delay. However, it is noticed, for AVB traffic classes, that increasing BE flows number results in

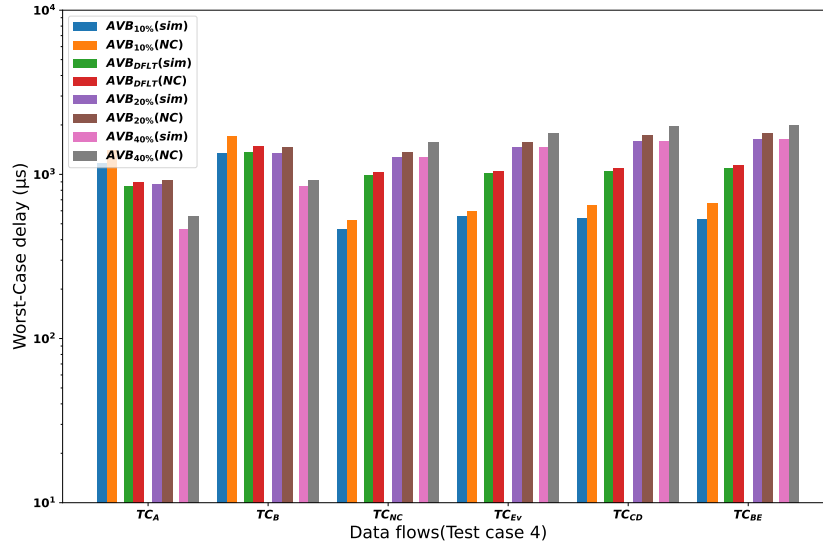


Figure 6.23: Different credit accumulation rates impact on the overall traffic classes delay.

a lower WCD bound in comparison to increasing AVB flows number due to the following reasons: AVB accumulates credit while BE getting transmitted, which guarantees transmission right after BE transmission completion given that AVB class window still valid. However, AVB spends credit while AVB getting transmitted, which results in a waiting period for credit accumulation as previously discussed.

6.3.4.1 Impact of credit accumulation rate

We also analyse the impact of credit accumulation rate on the WCD of the different classes of industrial traffic. Since both iso and non-iso have high priority, changes in scheduling lower priority traffic would not impact them. Therefore, the focus of the results on the remaining six traffic classes, as shown in Figure 6.23. The accumulation rate for AVB classes is defined as the proportion of the link capacity dedicated for AVB traffic. Hence, increasing this proportion will increase the probability of having positive credit and priority of AVB transmission when compared to lower priority traffic. Figure 6.23 shows that the end-to-end WCD of class A traffic experiences around 60% reduction in when accumulation rate increases four times, $I_A = 10\%$ to 40% . For class B traffic, the WCD reduction is around 50% for $4x$ the increase in the accumulation rate, $I_B = 10\%$ to 40% . While both classes experience a high reduction percentage, increasing

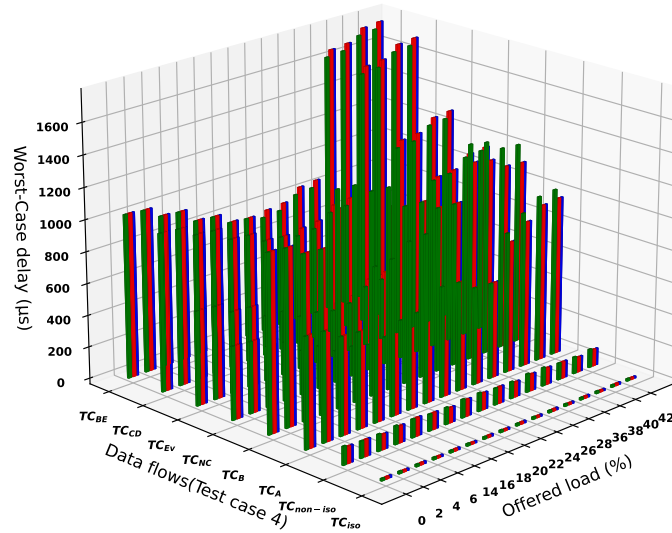


Figure 6.24: Worst case delay of traffic classes for varying offered load.

class A accumulation rate gives larger reduction as expected due to the priority selection aspect. On the contrary, the remaining four classes of traffic, i.e. BE traffic, experience higher WCD trend when accumulation rates increase. This happens due to AVB traffic consumption of the available bandwidth, left after transmitting iso and non-iso, while BE frames are being hold by the priority selection shaping. The proposed framework could be used to assess and decide on optimal accumulation rates, which is an important aspect for the network design. Therefore, it could be ensured that all traffic classes QoS requirements are met.

6.3.4.2 Redundancy and load distribution

Redundant networks are a common design in critical industrial applications to ensure fault tolerance. This experiment considers a redundant network topology, depicted in Figure 6.19.d. Our objective is to compute the WCD for all traffic classes while considering different load distributions across the different network paths. For example, routing all incoming flows through one link $L(S1, S2)$, which leaves other links $L(S1, S3)$ and $L(S1, S4)$ with no loads. Fig-

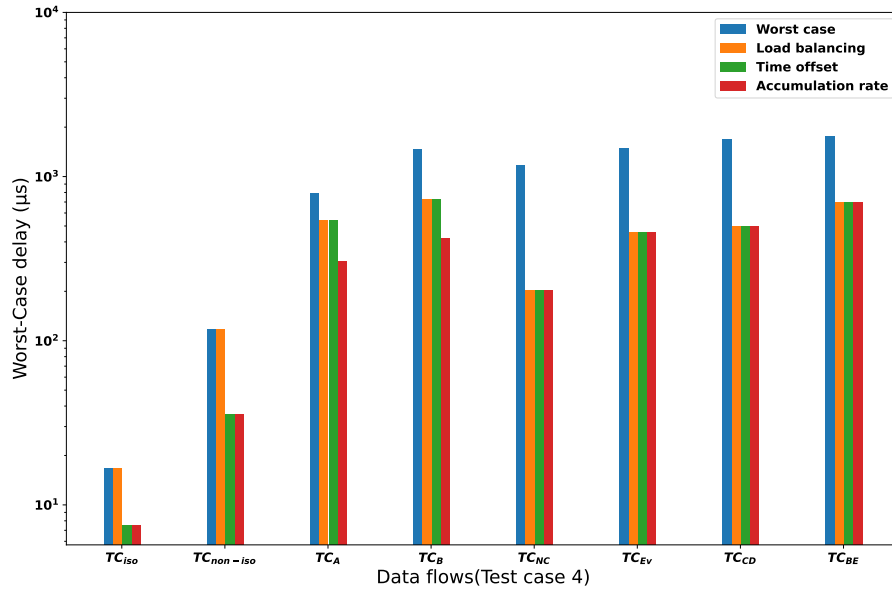


Figure 6.25: Improvement of the Worst case delay bounds for traffic classes.

Figure 6.24 depicts the computed WCD for varied loads over the aforementioned three links. Using this setup, iso traffic WCD is only affected by number of hops across the path, regardless of the offered load from other traffic classes. However, for the rest of classes we notice that the optimal value for offered loads, over different paths, can be obtained by computing where minimum WCDs of different traffic classes are included. In addition, the upper bound on the WCDs in case of failure could be obtained, e.g. all frames traverse the same path, by calculating the surface curve where maximum WCDs are included.

We conclude this synthetic evaluation by highlighting the main findings on obtaining lower WCDs for industrial traffic classes. In Figure 6.25, the worst case was assumed where all flows are transmitted at the same instance, traversing the same path, and with minimum accumulation rate for AVB traffic. First, by recomputing the WCD after regrouping the traffic to traverse different paths with optimal offered loads, a lower WCDs are obtained for all traffic classes except the iso and non-iso traffic. Next, by setting a considered time offset for each flow, this could lead to a minimised queuing delay and therefore overall lower WCDs for all traffic classes. Finally, selecting an optimal accumulation rate for AVB classes without impacting the WCDs for the lower priority traffic classes.

6.3.5 Industrial use-case:

In this section, the proposed impNC/TSN_industrial23 framework analyse the use case of a redundant industrial networks, based on [G⁺21] (*use case 07*) and depicted in Figure 6.26. A redundant network topology of four TSN domains, described in Section 5 is considered. TSN domains, two to four, implement Quality Control cells proposed in [SZP22a]. The breakdown of the network components is:

- Detection unit (D): sensing unit, part of the control loop, that detects new items' arrival. Applications running over this unit produce iso, non-iso, events and diagnostics data traffic.
- Actuation unit (A): control unit that performs actions on items after being checked. Same data traffic is produced as detection unit.
- Inspection unit (I): recording unit that perform visual inspection for quality checked items. Applications running over this unit produce class A, class B, network control, events, alarms, configurations and diagnostics data traffic.
- Ethernet interface (E): communication interface to connect different purpose units such as logic controllers, computing units, or monitoring and diagnostics clients of the industrial use case. Applications running over this unit generate and handle all traffic classes.
- Human/Machine interface (H): embedded system to monitor and configure the operating machines of the industrial use case. This system produces events, alarms, the best effort, configurations and diagnostics data traffic.

This analysis considers a typical network configurations and data traffic parameters for industrial applications as follows:

- Network link capacity (C): 100 Mbps or 1 Gbps.
- Maximum processing delay of a switch: $1 \mu s$
- Maximum buffer size of a switch: 1000 frames.
- Propagation delay: 100 ns.
- Maximum frame size: Isochronous (L_{iso}) = 128B, non-Isochronous ($L_{non-iso}$) = 1000B, network control (L_{ncl}) = 500B, for other traffic

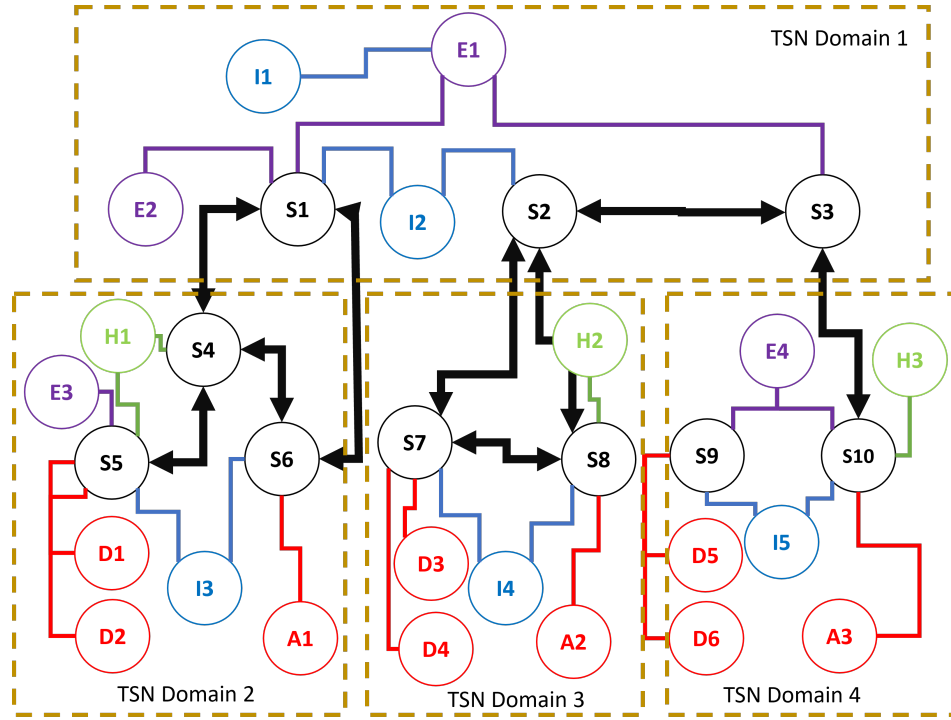


Figure 6.26: Industrial use case (Connection of Rings).

classes: (L_{class}) = 1500B.

- Minimum cycle time: Isochronous (τ_{iso}) = $250\mu s$, non-Isochronous ($\tau_{non-iso}$) = $2ms$, network control (τ_{ncl}) = $50ms$.
- Maximum data rate: class A (r_A) = 12Mbps, class B (r_B) = 15Mbps, best effort (r_{be}) = 20Mbps.
- Idle slope: I_A = 40%, I_B = 20% of the link capacity.
- Maximum burst duration: class A (σ_A) = 4s, class B (σ_B) = 2s, Events (σ_{ev}) = 1s, Configurations (σ_{cfg}) = 1s, best effort (σ_{be}) = 2s.

The characterisation of the traffic flows per network component, traversing the four TSN domains, are summarised in Table 6.11. Each TSN domain has **64** flows of different traffic classes. In addition, **14** data flows are exchanged between Ethernet devices across all domains, **36** flows are exchanged between TSN domain 1 and different units in other TSN domains, and **14** data flows exchanged between E_1, E_2 and I_1, I_2 . The total count for this industrial use case is **256** data flows.

Our objective is to compute the maximum value (i.e. the worst case) of the end-to-end delay for traffic flows per QCAP unit. Then, verify that the network

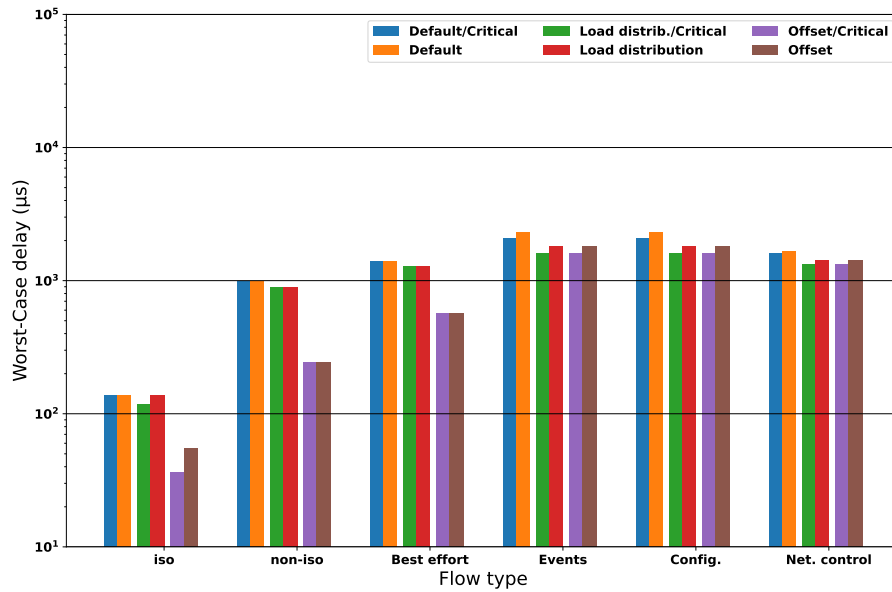


Figure 6.27: Detection

configurations are feasible, i.e. satisfy quality of service requirements for all traffic classes.

For that, a consistent network configurations (Default) (DFLT) were considered by choosing the same bandwidth (100 Mbps) for all links. Additionally, the case where all traffic sources are transmitting at the same time is considered, i.e. without an offset, and traffic flows are traversing the shortest path without considering the offered load of this path. Next, the traffic flows are regrouped to be transmitted across different paths with optimal load distribution. Then, this study considers the case where traffic transmissions has an offset to reduce the queuing component in the overall delay. Finally, the study relaxes the network configurations by choosing larger bandwidth (1Gbps) for what is identified as critical links, e.g. links between switches and links that carry large number of iso and non-iso traffic flows.

Figures 6.27, 6.28, 6.29, and 6.30 depict the WCD bounds for the traffic flows of the different QCAP units in Table 6.11, where the following points are observed:

- The WCD bounds, of the Detection unit depicted in Figure 6.27, exceeds the deadline for the iso traffic flows ($100\mu s$) unless adding an offset period to the start of iso flow transmissions and distributing the traffic flows across different paths. Moreover, configuring the critical links with larger

Table 6.11: Data flows characterisations

Unit	Flow: source, sink	type	deadline
Detection	$f_{i,1} \dots f_{i,5} : D_i, A_j$	iso	$100\mu s$
	$f_{i,6} \dots f_{i,10} : D_i, I_k$	non-iso	$1ms$
	$f_{i,11} : D_i, H_j$	BE	n.a.
	$f_{i,12} : D_i, E_2$	events	$10ms$
	$f_{i,13} : D_i, E_2$	config.	$10ms$
	$f_{i,14} : D_i, E_1$	net. control	$10ms$
Actuation	$f_{j,1} \dots f_{j,10} : A_j, D_i$	iso	$10\mu s$
	$f_{j,11} \dots f_{j,15} : A_j, I_k$	non-iso	$1ms$
	$f_{j,16} : A_j, H_j$	BE	n.a.
	$f_{j,17} : A_j, E_2$	events	$10ms$
	$f_{j,18} : A_j, E_2$	config.	$10ms$
	$f_{j,19} : A_j, E_1$	net. control	$10ms$
Inspection	$f_{k,1} \dots f_{k,10} : I_k, D_i$	non-iso	$1ms$
	$f_{k,11} \dots f_{k,15} : I_k, A_j$	non-iso	$1ms$
	$f_{k,16} : I_k, E_k$	class A	$15ms$
	$f_{k,17} : I_k, E_k$	class B	$20ms$
	$f_{k,18} : I_k, E_k$	BE	n.a.
	$f_{k,19} : I_k, E_2$	events	$10ms$
	$f_{k,20} : I_k, E_2$	config.	$10ms$
	$f_{k,21} : I_k, E_1$	net. control	$10ms$
Ethernet	$f_{k,1} : E_k, H_k$	BE	n.a.
	$f_{k,2} \dots f_{k,3} : E_k, I_k$	BE	n.a.
	$f_{k,4} : E_k, E_1$	net. control	$10ms$
	$f_{k,5} : E_k, E_2$	events	$10ms$
	$f_{1,4} \dots f_{1,9} : E_1, D_i$	net. control	$10ms$
	$f_{1,10} \dots f_{1,12} : E_1, A_j$	net. control	$10ms$
	$f_{1,13} \dots f_{1,15} : E_1, E_k$	net. control	$10ms$
	$f_{2,4} \dots f_{2,9} : E_2, D_i$	events	$10ms$
	$f_{2,10} \dots f_{2,15} : E_2, D_i$	config.	$10ms$
	$f_{2,16} \dots f_{2,18} : E_2, A_j$	events	$10ms$
	$f_{2,19} \dots f_{2,21} : E_2, A_j$	config.	$10ms$
	$f_{2,22} : E_2, E_k$	events	$10ms$
	$f_{2,23} : E_2, E_k$	config.	$10ms$
H/M interface	$f_{j,1} : H_j, E_k$	events	$10ms$
	$f_{j,2} \dots f_{j,3} : H_j, D_i$	config.	$10ms$
	$f_{j,4} : H_j, A_j$	config.	$10ms$
	$f_{j,5} : H_j, I_k$	config.	$10ms$

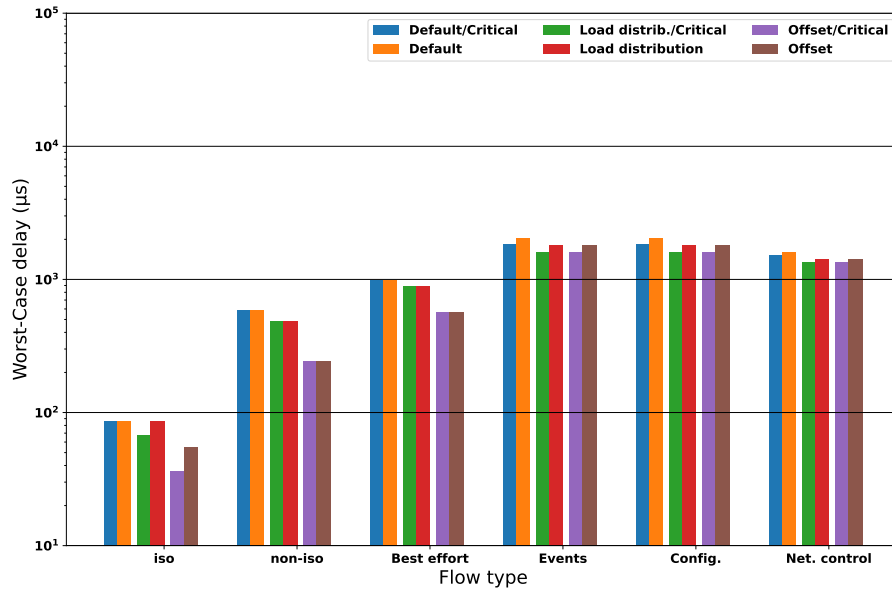


Figure 6.28: Actuation

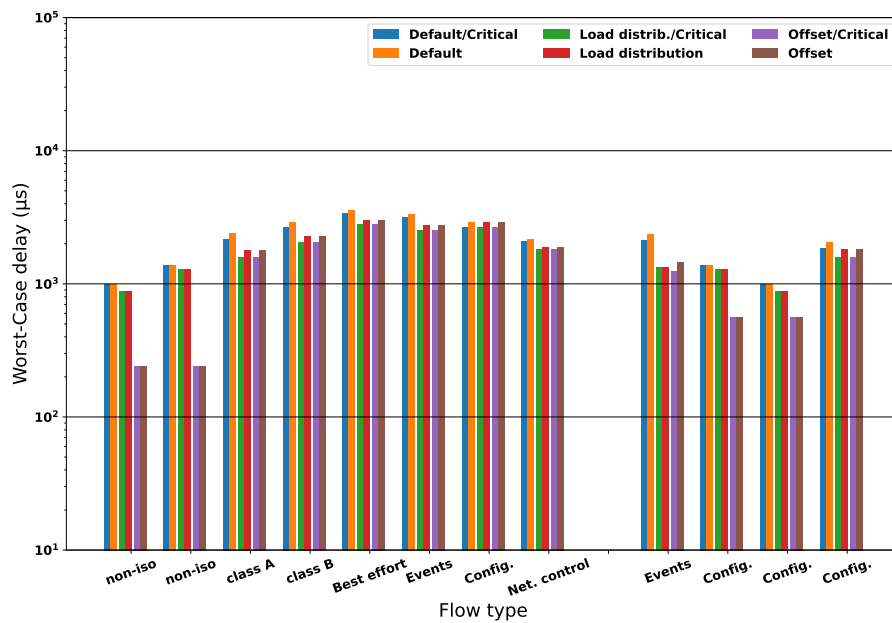


Figure 6.29: Inspection and H/M Interface

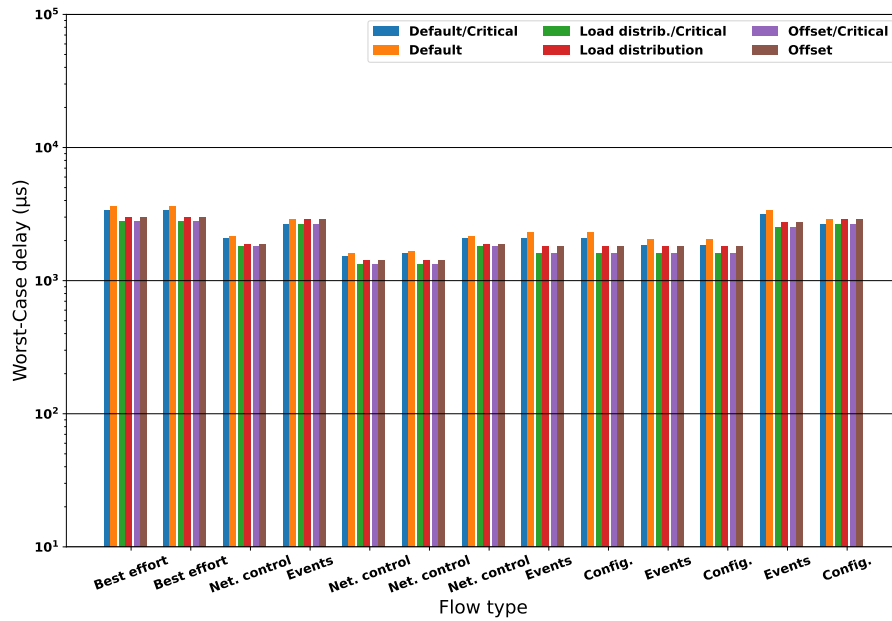


Figure 6.30: Ethernet

bandwidth reduces the transmission delay. Therefore, the overall WCD for the iso flows is reduced.

- The WCD bounds, of the Inspection unit depicted in Figure 6.29, exceeds the deadline for the non-iso traffic flows (1ms). This is expected as non-iso flows of the Inspection units are less prior to the iso flows of the detection and actuation units. Regrouping the traffic flows will reduce the WCD, allowing the non-iso set of flows, sent to the detection unit, ($f_{k,1} \dots f_{k,10}$) to not exceed the deadline. However, the other non-iso set of flows, sent to the actuation unit, ($f_{k,11} \dots f_{k,15}$) requires adding an offset period to the start of the non-iso flow transmissions as well to satisfy the deadline requirement.
- For other units (Actuation, Ethernet, Human/Machine interface), the WCD bounds are within the defined deadline, even without traffic regrouping, load distribution or adding an offset. However, in practical setup all units operations are dependent, therefore the aforementioned units operation will fail even though solely calculating the WCD bounds for a unit shows otherwise.

This previous analysis shows that it is possible to confirm the success or failure of each QCAP unit operation. Additionally, it illustrates how tuning network

configurations exhibits lower WCD bounds on traffic flows, which helps in assessing the feasibility of, and optimising, those configurations, also offering the possibility of adding more traffic flows if necessary.

Chapter 7

Integrating TSN with SDN and Wi-Fi: Enhanced Time-Critical Communication Solutions

This chapter investigates the relationship between Time-Sensitive Networking (TSN) and two transformative technological paradigms: Software-Defined Networking (SDN) and Wi-Fi. While TSN has already proven its ability to enhance real-time communication within Ethernet networks, integrating it with SDN and Wi-Fi takes these capabilities to new levels. The combination of TSN with the programmability of SDN and the widespread accessibility of Wi-Fi expands the possibilities for time-sensitive applications.

The discussion begins by examining the dynamic interaction between TSN and SDN, where SDN's centralized management and flexibility complement TSN's deterministic features, creating a powerful framework for optimizing and controlling network traffic in real-time. Additionally, the integration of TSN with Wi-Fi introduces the benefits of wireless connectivity, enabling mobile and versatile applications while maintaining the strict timing requirements necessary for time-sensitive data.

Together, TSN, SDN, and Wi-Fi form a robust alliance capable of addressing the unique challenges across various industries, including industrial automation, healthcare, smart cities, and transportation. The following sections explore the fundamental principles, technical complexities, and relevant use cases that highlight the transformative potential of merging TSN with SDN and Wi-Fi. The practical implications of these innovations are also discussed, underscoring

their impact on a wide range of industries and heralding a new era of enhanced time-critical communication solutions.

7.1 Software-Defined Time Sensitive Networks for Industrial Automation

Industry 4.0 [WSJ17] revolves around the deployment of Cyber-Physical Systems (CPS), which consist of a network of distributed digital devices, including sensors and actuators, designed to measure, monitor, and control physical processes. These devices form the device layer, where they communicate with each other and with devices from other CPSs. Data collected by these sensing devices must be transmitted to the application layer, while commands to control actuators originate from the same layer. As a result, the effectiveness of CPSs is heavily reliant on the communication and network layer, which is responsible for transmitting data and control signals both within their own systems and between different systems.

The application and device layers impose specific limitations, constraints, and requirements on the communication and network layers to handle various types of data traffic, each with different quality of service needs. For instance, industrial applications often demand strict timing and reliability criteria, necessitating the design of industrial networks with different considerations compared to traditional networks [Sau10] [GH12]. Non-deterministic network delays and jitter can significantly impact the control quality of a CPS. In systems like isochronous motion control, maintaining extremely tight jitter bounds (approximately 1 to 10 microseconds) is essential for system stability [KK15].

Traditionally, Real-Time Ethernet (RTE) has played a critical role in industrial networks, often supplemented by additional protocols to meet the specific requirements of industrial applications, such as determinism, reliability, and efficient communication with well-bounded delay and jitter. RTE achieves these goals by modifying the Medium Access Control (MAC) layer, allowing for exceptionally low latency and deterministic responses. Key Industrial Ethernet protocols in this domain include EtherCAT (Beckhoff), EtherNet/IP (Rockwell), PROFINET (Siemens and GE), POWERLINK (B&R), and Sercos III. However, implementing RTE [Fel05] faces significant challenges when transitioning existing systems to adopt new protocols. Real-time performance can be compromised by application protocols based on TCP, and the efficiency of data

transmission is often reduced due to protocol overhead, particularly since the minimum Ethernet frame size is 64 bytes, whereas typical industrial communication data sizes are much smaller, usually in the range of 1 to 8 bytes. To address these challenges, the IETF DetNet Working Group [FTVF19] and the IEEE 802.1 Time-Sensitive Networking Task Group have developed a set of standards that define mechanisms for time-sensitive data transmission across deterministic Ethernet networks, ensuring robust real-time capabilities. Thus, TSN emerges as a crucial technology for addressing the challenges of real-time automation and industrial networks.

Numerous challenges remain in the implementation of Time-Sensitive Networking (TSN) for real-time data transmission over wired networks, driving the research discussed in this section. One key challenge is the need for time-aware shaping configurations, which are crucial for ensuring an upper bound on network delay while achieving optimal performance across various use cases. In this approach, Software-Defined Time-Sensitive Networking (SD-TSN) leverages the software-defined network (SDN) paradigm to simplify network configurations and facilitate the distribution of schedules. SDN enables the deployment of network applications that execute centralized algorithms, providing forwarding decisions, configuring resources to manage TSN queues and shaping parameters, and running resource utilization algorithms to monitor network performance.

To demonstrate the effectiveness of this design, experiments were conducted within an emulated environment using Mininet [Tea12]. The contributions of this work are detailed as follows:

- The concept of traffic scheduling in industrial applications is introduced, highlighting the necessity for network-device contracts to compute and distribute traffic schedules, ensuring that critical traffic maintains an upper bound on latency.
- A SDN-based solution is implemented on the Mininet network emulation platform to assess the performance of end-devices in adhering to calculated transmission schedules.
- The performance of this implementation is evaluated, demonstrating that it effectively meets real-time network requirements for practical network sizes.

This section is organized as follows: It begins with a discussion of related work

in the field. Next, the system model is presented, illustrating the integration of SDN and TSN to support a Smart Factory use case. This is followed by an examination of the process involved in computing and distributing transmission schedules, with a focus on end devices and controller contracts. The section concludes with a detailed setup of the Mininet environment, an evaluation of the work, conclusions drawn from the findings, and identification of areas for future research and potential contributions.

7.1.1 Software defined TSN related work

Industrial automation [DSA⁺14] relies on a robust network layer that connects complex systems, enabling the supervision, configuration, and control of field devices. This intricate network architecture presents several challenges, such as the need for timely and accurate decision-making, synchronization of multicore systems with varying clock rates, deterministic data frame transmission, reduction of processing time for queued frames in outgoing buffers, and the imperative of sending frames at or before a specific time, T .

Standards like IEEE 802.1 TSN and IETF DetNet have been established to support time-synchronized, low-latency streaming over Layer 2 bridged networks and to create deterministic data pathways with defined bounds on packet latency, loss, and jitter over Layer 3 routed networks. Despite these standards, configuring real-time networks remains a complex issue [Kál]. Several studies have explored how SDN could enhance the configuration and management of industrial networks.

For instance, research presented in [HDSD16] demonstrates a proof of concept for applying SDN management principles to Real-Time Ethernet. This work integrates the OpenFlow protocol with Powerlink real-time Ethernet to address data loss due to link failures. Additionally, [STB19] proposes a comprehensive SDN solution that leverages TSN features not only for data path management but also for precise time synchronisation.

However, these studies did not address the complex problem of traffic scheduling, which is recognized as NP-hard (similar to bin-packing) [NDR16]. The study in [Ste10] tackles this issue by presenting various Integer Linear Programming (ILP) formulations for computing routes and transmission schedules for time-triggered flows. The authors also demonstrate a proof of concept for their Time-Sensitive Software-Defined Networking (TSSDN) solution, which

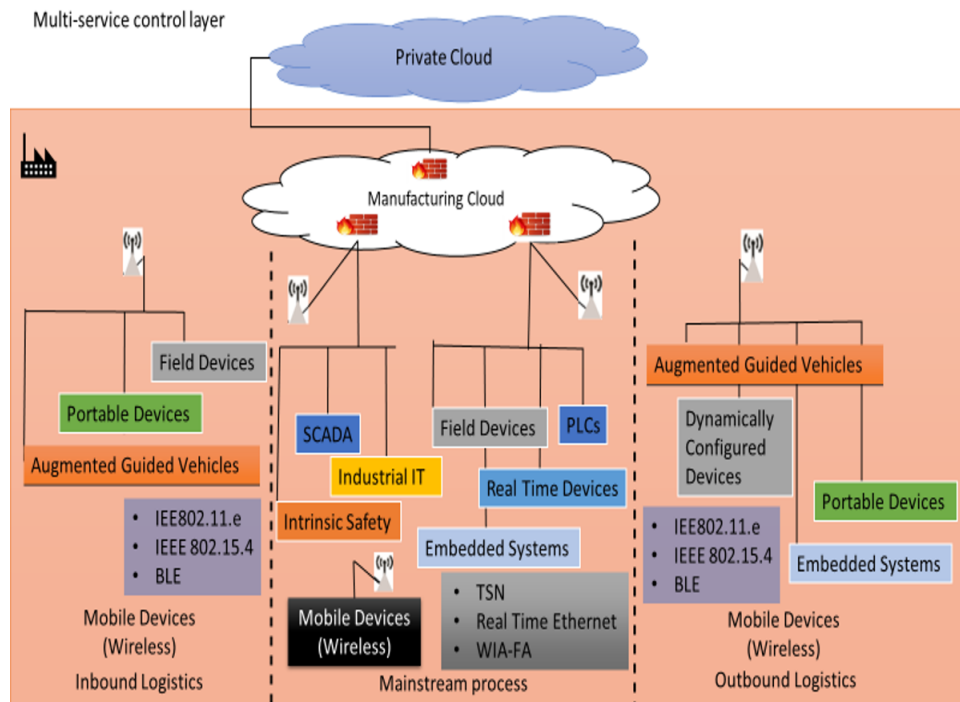


Figure 7.1: Multi-Service Control Layer of a Factory Automation Model

achieves virtually constant end-to-end latency (with a standard deviation of $\leq 1 \mu s$) and a worst-case jitter of $\leq 7 \mu s$.

In contrast to these studies, the SD-TSN proposal focuses not only on automating the configuration of time-sensitive networks but also on enabling interaction between TSN and the network operating system through contracts, simplifying the scheduling process. This approach also enhances network utilization by incorporating traffic flow-related information that influences scheduling decisions. Moreover, the SD-TSN proof of concept is being evaluated in the Mininet network emulator, providing a thorough and accurate assessment of its capabilities.

7.1.2 Factory Automation Model And System Architecture

In a typical industrial network environment, a diverse array of nodes collaborates to fulfil various control and protection functions, facilitating the transmission and reception of different data streams and enabling network-wide monitoring and data visualization. Figure 7.1 illustrates a multiservice control layer within a factory automation model, encompassing a wide range of technologies and equipment, including SCADA systems, Programmable Logic Controllers (PLCs), and field and real-time devices. The figure also highlights

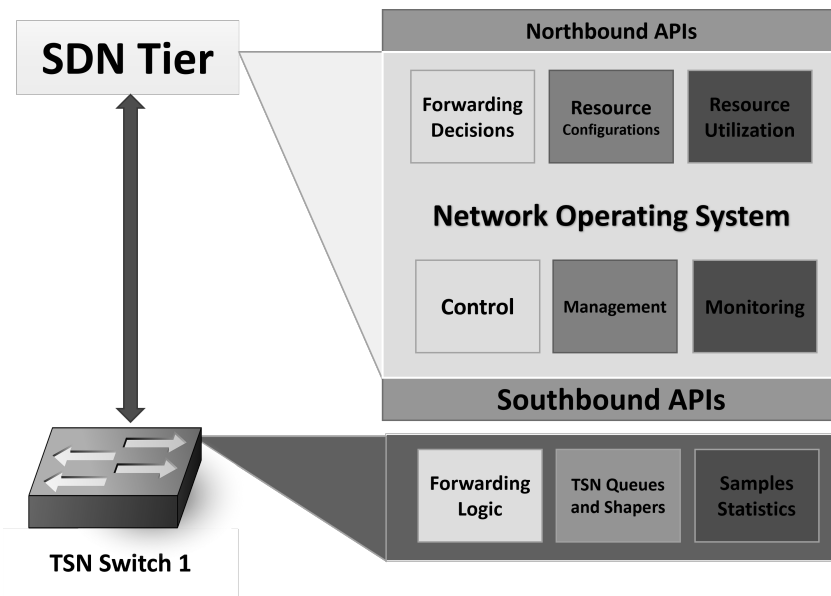


Figure 7.2: Proposed SD-TSN Architecture

Automated Guided Vehicles (AGVs), wireless communication links, and a variety of other devices such as personal computers, printers, and cameras. This factory control layer is interconnected with a manufacturing cloud, enabling data exchange with external networks, leveraging significant data storage capacity, harnessing cloud processing power, and enhancing overall security.

Industrial networks, as described, are characterized by complex architectures with numerous integrated devices. Even relatively small industrial networks may exhibit a hierarchical structure of three or four levels, often organized through a combination of topologies like rings and stars. The diverse nature of these integrated devices, each with unique communication characteristics and sometimes lacking interoperability, often results in the presence of vendor-specific management systems. Moreover, variations in bandwidth or network load directly influence data transfer performance. Consequently, diligent resource monitoring is essential to sustain system performance. Effective resource monitoring relies on using network state information as a key driver for control functions, enabling dynamic decision-making to adapt to changing conditions.

The proposed architecture, depicted in Figure 7.2, addresses both Layer 3 and Layer 2 components within the system model. Layer 3, commonly known as the SDN tier, is built upon a Network Operating System (NOS) [CABI⁺15] and primarily comprises three key functionalities:

Control Plane: The control plane is responsible for managing the network's core operations and includes an OpenFlow [MAB⁺08] network controller that

establishes forwarding rules. It also features a TSN application controller that manages independent reservation decisions for each application's end devices and TSN switches, utilizing selected protocols. The control plane consists of the following subcomponents:

- **Discovery Service:** This service uses the Link Layer Discovery Protocol (LLDP) to periodically broadcast discovery packets to all TSN switches and awaits response packets, enabling the identification of the network's topology. End devices are initially identified during the establishment of contracts for reserving traffic paths, or later through the inspection of injected packet-in messages.
- **Path Computation:** The network controller determines network routes to manage traffic efficiently and prevent the occurrence of loops within the network.
- **Dynamic Control of Traffic Reservations:** The TSN application controller directs end devices to establish contracts and make reservations. Alternatively, it may consult the network controller for reservations on behalf of the end devices. TSN switches defer reservation decisions to the network controller, following the IEEE P802.1Qcc standard.

Resource Management The Open vSwitch Database Management Protocol (OVSDb) handles management and configuration operations on network devices. This component allows for the creation, configuration, and deletion of ports, queues, and switch tunnels. The management protocol governs various network characteristics, including:

- Selection of topology control for both non-TSN and TSN traffic, ensuring optimal network performance across different types of traffic.
- Maximum bandwidth allocation for TSN flows across each physical link, adhering to the standards outlined in IEEE Std 802.1Q and/or IEEE Std 802.1Qca, to maintain efficient data flow and avoid congestion.
- Timing and synchronization protocols and parameters based on IEEE Std 802.1AS-2011, which are crucial for ensuring that all network components remain synchronized, particularly in time-sensitive applications.

Resource Monitoring: is managed by distributed sFlow agents integrated into TSN switches. These agents continuously monitor network traffic and report statistical samples, known as sFlow datagrams, to the sFlow collector. sFlow

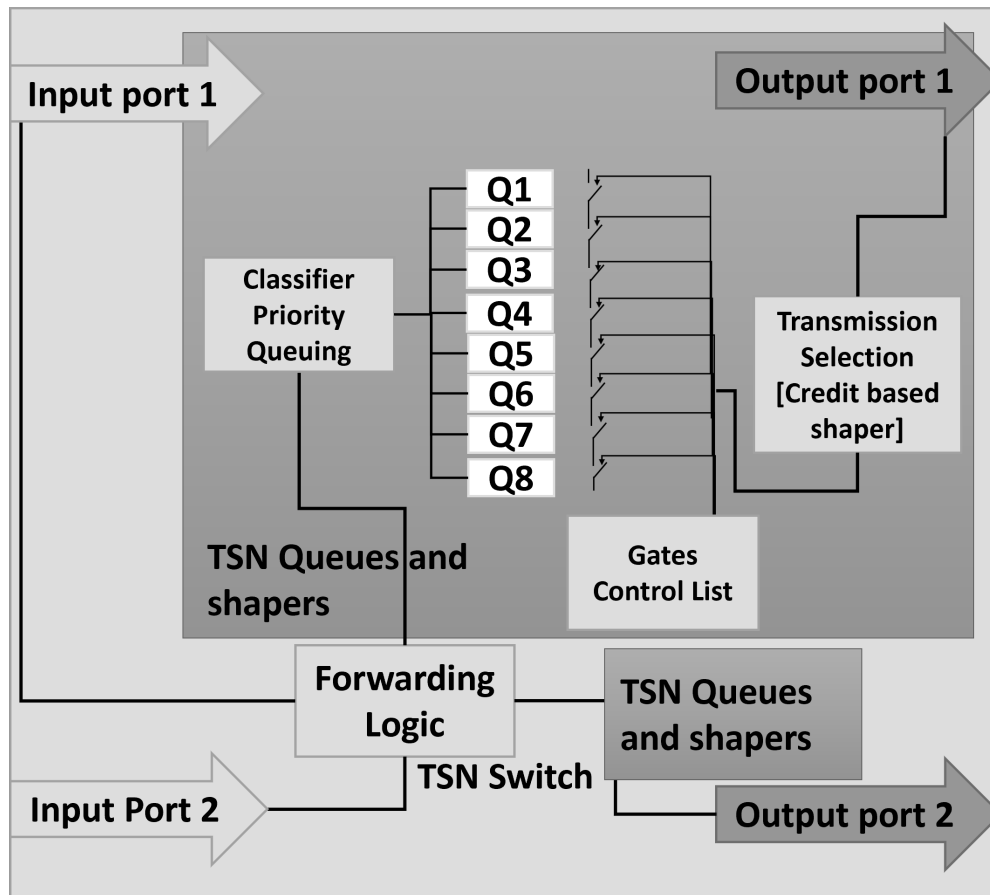


Figure 7.3: TSN switch internal components

employs both packet-based sampling and time-based sampling methods to create a detailed profile of network traffic, providing comprehensive visibility and allowing for real-time monitoring and management of network resources. This ensures that the network can efficiently handle the demands of time-sensitive and critical industrial applications.

Layer 2, depicted in Figure 7.3, is structured around Layer 2 TSN switches, which are critical for ensuring time-sensitive networking performance. These TSN switches are equipped with forwarding logic, sFlow agents, TSN priority queues, and traffic shapers, all of which contribute to managing and optimizing network traffic. The key components of these switches include:

- **Classifier:** Utilizes a three-bit Priority Code Point (PCP) to categorize each packet into a specific traffic class, such as TSN traffic or non-TSN traffic, ensuring that packets are handled according to their priority level.
- **Priority Queues:** Separate queues are assigned for different types of traffic, with real-time traffic being given the highest priority. Lower-priority

traffic is placed in other queues, ensuring that time-sensitive data is transmitted with minimal delay.

- **Gates and Gates Control List (GCL):** These gates control the transmission of packets from the queues. The GCL contains entries that manage the state of each gate, with each entry specifying a timestamp for when the gate should open or close. This ensures precise timing and synchronization for packet transmission.
- **Transmission Selection:** When multiple gates are open at the same time, this component refers to a secondary scheduling algorithm, such as credit-based shaping, to determine the sequence in which packets are transmitted. This ensures that the network can handle simultaneous traffic flows efficiently, maintaining the quality of service for time-sensitive applications.

Together, these components form the backbone of Layer 2 in the TSN architecture, providing the necessary infrastructure to manage traffic effectively and meet the stringent requirements of industrial networks.

7.1.3 Scheduling and Routing in Software Defined TSN

In our Software Defined-Time Sensitive Networking (SD-TSN) approach, both coarse-grained schedules is merged, which specify transmission times at the end devices, and fine-grained link schedules, which specify transmission times at the network core. Below, the primary approach is outlined for computing transmission schedules and routing in SD-TSN. This approach revolves around the swift and incremental scheduling of TSN flows through the use of network and application contracts. This not only reduces execution times but also preserves the quality of schedules, ensuring a high number of accommodated flows.

7.1.3.1 Scheduling approach in SD-TSN:

For each TSN flow, the end devices (or hosts) establish a contract with the TSN controller, providing critical details about the flow, including the maximum packet size and the maximum number of packets per time interval. In return, the TSN controller reserves dedicated resources for this TSN flow, such as bandwidth, buffering, and scheduling resources. These contracts are crucial for the following reasons:

- **Buffer Allocation:** The contracts ensure that adequate buffer space is allocated for the TSN flow, reducing the risk of congestion loss.
- **Flow Pacing:** The contracts regulate the pacing of TSN flows across the network, ensuring that packets are delivered precisely when needed rather than as quickly as possible. This approach enhances network determinism, critical for time-sensitive applications.
- **Bandwidth Utilization:** The contracts allow non-TSN traffic to utilize any bandwidth not reserved by TSN flows, ensuring efficient use of network resources.

These contracts are also flexible, allowing for the creation of new contracts and the removal of old ones as necessary

Transmission schedules are organized in a cyclic manner, with each cycle divided into N slots. The duration of each cycle is based on the base time interval, t_{base} , and the duration of each slot, t_{slot} , constrained by the longest network path latency, $t_{longest_path}$.

- **Slot Calculation:** the number of slots, N , is determined is determined by the following formula:

$$N = INT\left[\frac{t_{base}}{t_{slot}}\right], t_{slot} \leq t_{longest_path} \quad (7.1)$$

Here, t_{base} and t_{slot} are system parameters calculated by the network controller.

- **Slot Allocation:** the TSN controller assigns a specific slot to each contracted TSN flow and creates GCLs that are distributed to network devices, including TSN switches. Each switch then allocates the appropriate time slot to the sources of TSN flows.
- **Synchronisation:** All devices are equipped with synchronized clocks (according to IEEE 802.1AS), ensuring that TSN switches and end devices (hosts) adhere to the same cycle base time for their schedules. The hosts calculate the exact transmission times using the base period, slot length, transmission period, and their allocated time slot.
- **Transmission Timing:** Sources transmit TSN packets precisely at the allocated time slot. If there are no TSN packets to send during a slot, sources can either remain idle or transmit non-TSN packets, such as web traffic.

This approach maximizes network utilization by allowing the transmission of best-effort traffic while waiting for TSN packets to arrive during their designated slots. Moreover, TSN switches can be equipped with frame pre-emption mechanisms, as specified in IEEE 802.1Qbu, enabling the interruption of best-effort frames to prioritize the transmission of higher-priority frames. This ensures that critical TSN traffic is delivered on time, maintaining the deterministic nature of the network.

7.1.3.2 Routes calculation in SD-TSN

The network controller calculates a specific set of shortest paths to route TSN flows through. These paths are determined when end devices sign contracts to reserve paths for their traffic. As a result, each TSN flow is constrained to be routed through one of the paths within this set, avoiding the exploration of random paths. In formulating this problem, several assumptions are made:

Set of Free Time Slots: There exists a set of available time slots that can be assigned to a path, denoted as T . $T \equiv \{t_l\}$, where $l \in [1, \dots, L]$, and L represents the number of available slots for path assignment.

Set of TSN Flows: A set of TSN flows originating from an end device is considered, labelled as F . $F \equiv \{f_i\}$, where $i \in [1, \dots, NF]$, and NF represents the number of flows to be scheduled.

Set of Candidate Paths: For each TSN flow, a set of potential candidate paths between source and destination is defined, denoted as CP . $CP \equiv \{cp_k\}$, where $k \in [1, \dots, NCP]$, and NCP represents the number of candidate paths for each flow $f_i \in F$.

Mapping of Flows to Candidate Paths: Each TSN flow $f_i \in F$ is mapped to a specific candidate path $cp_k \in CP$, forming the set F_{CP} . $F_{CP} \equiv \{f_{cp}^{i,k}\}; \forall i \in NF, \forall k \in NCP$, where $f_{cp}^{i,k}$ signifies the feasibility of routing flow i , f_i , over a candidate path k , cp_k .

Mapping of Candidate Paths to Time Slots: Each candidate path $cp_k \in CP$ is linked to a specific time slot $t_l \in TL$ to create the set CP_T . $CP_T \equiv cp_t^{k,l}; \forall k \in NCP, \forall l \in L$, where $cp_t^{k,l}$ denotes the assigned free slot for a particular path, ensuring the transmission of the TSN flow.

These defined sets and mappings are fundamental to the formulation of the path allocation problem for TSN flows. The objective function aims to maximise

the number of flows that can be routed through candidate paths with allocated time slots. It can be expressed as follows:

$$\text{maximise } \sum_{\forall i \in NF} [\sum_{\forall k \in NCP} [f_{cp}^{i,k} * \sum_{\forall l \in L} [cp_t^{k,l}]]] \quad (7.2)$$

Subject to the following constraints:

- At most, one time slot is allocated to a candidate path.
- Candidate paths that share links must not be assigned the same time slot.

Ultimately, the network controller allocates time slots to candidate paths based on flow rate and timing requirements, ensuring that the end-to-end latency is less than $t_{longest_path}$. This allocation strategy enables the routing of TSN flow packets through their designated candidate paths, meeting the specified latency criteria.

7.1.4 Validation and Discussion

This section provides a comprehensive overview of the evaluation process and testing conducted to assess the performance of SD-TSN. It details the test environment setup, experimental procedures, and the results obtained.

7.1.4.1 SD-TSN test environment setup

The test environment for evaluating SD-TSN was established using a combination of open-source software tools, outlined as follows:

Mininet: was used to emulate the network infrastructure required for the evaluation. This tool enables the creation of virtual networks on a single machine, each running an actual kernel. By emulating end devices and network switches within Mininet, various scenarios based on the SD-TSN architecture were simulated to evaluate performance under different conditions.

TSN Software Switch: A software-based Open vSwitch (OVS) was configured to function as a TSN switch. This switch supports both the sFlow protocol for monitoring network traffic and the OVSDB protocol for switch configuration. Kernel modifications were made to enable MultiQueue (MQ) and configure the Queueing Discipline (qdisc) to use the TAsynchronous Priority (taprio) scheduler. This setup allows for the implementation of GCLs for managing the output ports of the switch.

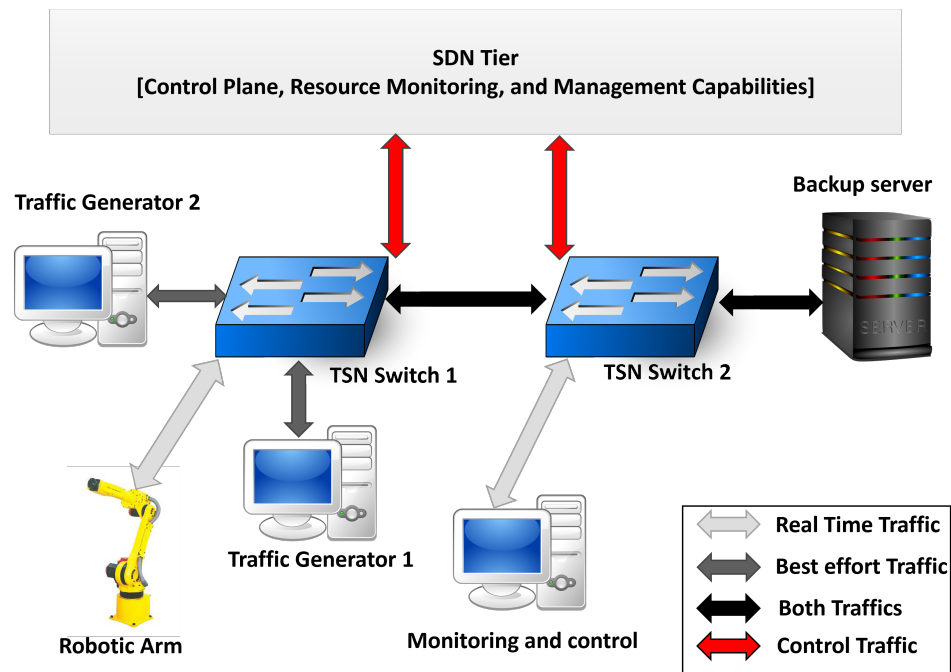


Figure 7.4: SD-TSN testing use case.

sFlow Collector: was employed to gather and analyze network traffic data. It continuously receives sFlow datagrams from all network devices and converts these into metrics that provide insights into network flows, interfaces, and device statuses.

Network Controllers: Two controllers were used in the setup: the OpenFlow network controller and the TSN controller. The OpenFlow controller manages the network's data plane, while the TSN controller handles the control plane functions, including traffic scheduling and resource allocation.

A software implementation for the objective function and solver was developed to maximize the number of routed TSN flows while maintaining predetermined latency bounds. This software was crucial for evaluating the SD-TSN solution.

7.1.4.2 SD-TSN emulation condition

Figure 7.4 illustrates a testing use case for a network implementing the SD-TSN architecture. This use case encompasses different network namespaces representing network devices (TSN switches, TSN controller, and Network controller), as well as host namespaces simulating the behaviour of real-time and best-effort traffic sources (Traffic generators, Robotic Arm, Monitoring and control unit, and Back-up server). Additionally, virtual Ethernet pairs enable communication between namespaces and with the main host namespace.

To configure the emulation parameters and replicate various link capabilities and network conditions, Linux Traffic Control and NetEm tools were used. This allowed us to perform tasks such as:

- Configuring link parameters, including packet queues on interfaces and bandwidth constraints.
- Emulating packet loss rates to assess path performance degradation.

To generate traffic, tools like ping, iperf, or Netperf, were used to create ICMP, TCP, and UDP flows. These flows were categorised as follows:

Real-time traffic: Representing TSN flows that need to be sent across the network and delivered at specific times.

Best-effort traffic: Simulating non-TSN flows, such as web traffic or data streams traversing the network.

Network Control traffic: Symbolising the traffic flow between network devices responsible for configuring and maintaining network operations.

The emulation process allowed us to verify the correct operation of the network design and validate various aspects of traffic scheduling. Furthermore, Mininet's Python API enabled the definition of different scenarios tailored to the specific features and metrics under evaluation.

7.1.4.3 SD-TSN performance evaluation

SD-TSN operates in accordance with the computed schedules, as described in Section 7.1.3.2. In contrast, non-constrained TSN (NC-TSN) requires the network controller to explore the entire solution space for available paths. Consequently, SD-TSN enhances network determinism by adhering to predefined schedules. To properly evaluate SD-TSN's performance compared to the basic NC-TSN, our experiments primarily focus on measuring the end-to-end latency for TSN flows under various scenarios. This approach allows us to assess SD-TSN's effectiveness in delivering hard real-time guarantees, a critical requirement for supporting industrial environments. Additionally, the SD-TSN control plane's capacity is evaluated to compute precise, accurate, and scalable transmission schedules.

Figure 7.5 shows the result from the assessment of end-to-end latency for 10,000 packets of real-time traffic sent across the network with a period of 20 ms. The results for NC-TSN show latency variations ranging from 27 to

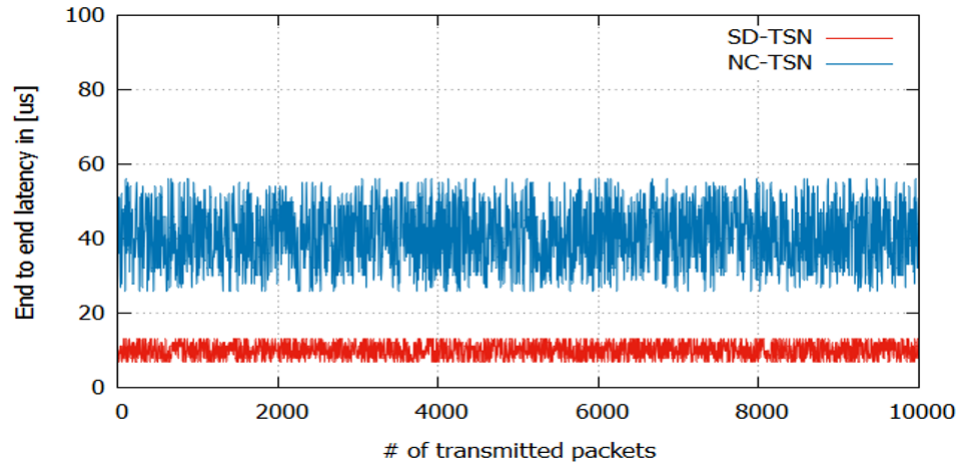


Figure 7.5: End to end latency versus number of real time transmitted packets.

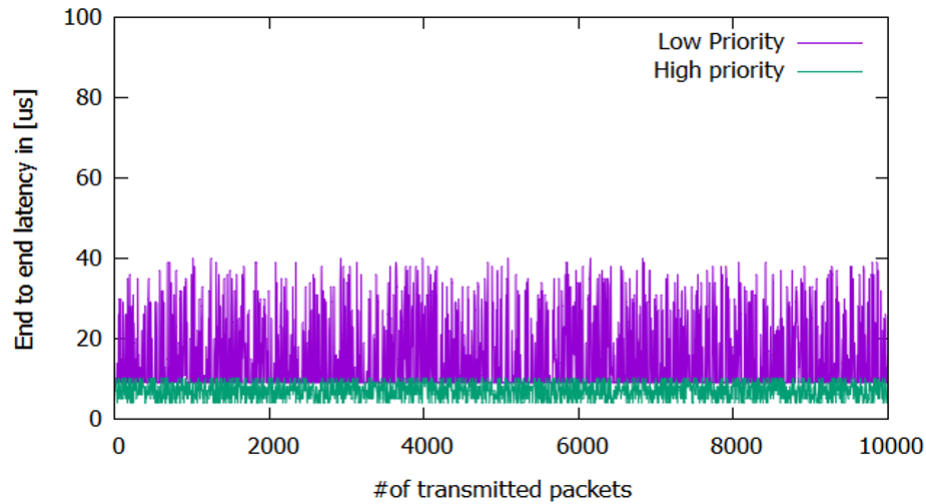


Figure 7.6: End to end latency for different types of traffics with different priorities.

56 μs , with an average latency of 43.46 μs and a standard deviation of 5.32 μs . This variability is attributed to the variable delays (ranging from 20 to 100 μs) that packets experience while traversing the network stacks. Additionally, some packets are not precisely assigned to a network interface, leading to non-deterministic delays. On the other hand, for SD-TSN, the results indicate latency variations between 7 and 13 μs , with an average latency of 9.64 μs and a standard deviation of 2.32 μs . These low end-to-end latency values signify that packets in SD-TSN are accurately placed on network interfaces with minimal deterministic delays. This outcome underscores SD-TSN's ability to adhere to transmission schedules with high precision while providing stringent bounds on the delays encountered in end systems.

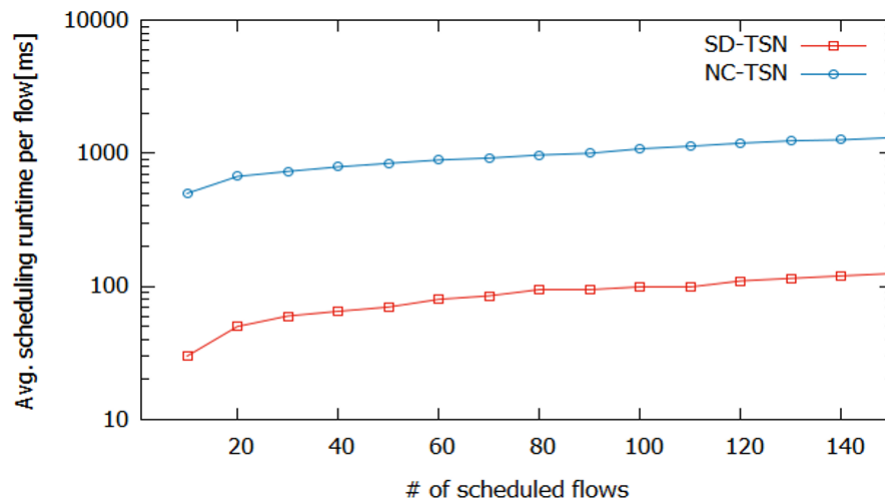


Figure 7.7: Average scheduling run time per TSN flow versus number of scheduled flows across the network.

In another experiment, assessment of the end-to-end latency for 10,000 packets of mixed traffic (real-time and best-effort) sent across the network with a period of 20 ms. The TSN real-time traffic was assigned high priority to meet its real-time requirements, while the best-effort traffic was assigned lower priority. Figure 7.6 illustrates the end-to-end latency for the TSN traffic, which remained consistent between 7 and 13 μs . This stability is due to TSN switches supporting frame pre-emption, ensuring that higher-priority frames are not affected by lower-priority best-effort traffic.

While the best-effort traffic does experience some impact due to prioritisation, it still achieves acceptable latency, ranging from 9 to 41 μs . This outcome is beneficial for enhancing network utilisation, as it ensures that real-time traffic maintains its stringent latency requirements while accommodating best-effort traffic with minimal disruption.

To assess the scalability of SD-TSN compared to the basic NC-TSN, a conducted experiment where the number of scheduled TSN flows are increased across the network. This experiment models a tree topology consisting of 4 TSN switches, 20 end devices, and both the network and TSN controllers. The objective was to maximise the number of flows that could be routed with allocated time slots. The runtime is measured for computing accurate schedules while varying the number of flows from 10 to 160.

Figure 7.7 demonstrates that the runtime for computing schedules using SD-TSN is at least an order of magnitude lower than that for NC-TSN. Specifically,

SD-TSN took 10 seconds (100 ms per flow * 100 flows) to compute schedules for over 100 flows. In contrast, NC-TSN required approximately 2 minutes to compute schedules for the same number of flows. In summary, the evaluation results highlight the following key findings:

1. SD-TSN ensures an end-to-end latency of less than 13 μ s for scheduled TSN flows.
2. SD-TSN effectively supports best-effort traffic without impacting the latency of real-time traffic, leading to improved network utilisation.
3. SD-TSN exhibits strong scalability, capable of supporting more than 100 flows in approximately 10 seconds for a medium-sized network. This demonstrates its efficiency in managing larger traffic loads.

7.2 Wireless TSN Extension to Enable Deterministic Connectivity

Wireless communication plays a crucial role in achieving the goals of the Industrial Internet of Things (IIoT). Recent advancements in Wi-Fi technology, particularly through iterations of the IEEE 802.11 standard, have addressed the increasing demands for speed, capacity, and reliability. The introduction of IEEE 802.11ac (Wi-Fi 5) and IEEE 802.11ax (Wi-Fi 6) has significantly improved performance metrics, making Wi-Fi a viable option for industrial applications [IEE13][IEE21b].

Wi-Fi offers several advantages in industrial automation, including flexibility, scalability, and enhanced reliability. The ability to easily install and reconfigure devices allows for adaptability in dynamic production environments [ABC⁺23]. Modern Wi-Fi technologies have incorporated features such as Multi-User Multiple Input Multiple Output (MU-MIMO), beam forming, and advanced interference management, all of which contribute to greater reliability and Quality of Service (QoS) [MTM22][LLW⁺17]. These improvements are crucial for maintaining consistent performance in industrial settings where connectivity demands are high.

TSN is designed to enable deterministic communication within Ethernet networks, and its integration with Wi-Fi can provide significant benefits for industrial automation. TSN introduces mechanisms for low latency, synchro-

nized, and reliable communication, which are essential for applications requiring precise timing and coordination, such as robotics and motion control [SHM⁺21][SPS⁺23]. By combining Wi-Fi with TSN, it is possible to achieve deterministic communication over wireless networks, addressing the limitations traditionally associated with wireless technology [Kin21].

Efforts to integrate TSN with Wi-Fi are ongoing, and several initiatives aim to standardize and enhance this integration. The IEEE 802.11 TSN Task Group is actively working on standards that will enable deterministic communication within Wi-Fi networks [AMS⁺21]. The Wi-Fi Alliance has also been developing specifications to support TSN integration, which will improve interoperability and network performance [All23]. Additionally, the Industrial Internet Consortium has established test beds to explore the practical aspects of integrating TSN with various networking technologies, including Wi-Fi [LHP⁺21].

Recent studies highlight the potential of wireless TSN extensions in industrial automation. The combination of Wi-Fi and TSN is anticipated to address the need for reliable and deterministic wireless communication, enhancing connectivity, efficiency, and productivity in industrial environments [SIV21]. This integration is expected to play a critical role in advancing industrial automation and supporting the evolving requirements of the IIoT.

This section outlines the structure and content of the upcoming sections. It provides a roadmap for the reader, indicating the topics that will be covered in this section, which discuss time-sensitive use cases that require wireless connectivity, highlight the extension of TSN functionalities for WiFi, introduce a simulation model for a WiFi TSN network supporting collaborative robots, introduced in Section 3.5, and mention the analysis of results obtained from simulations.

7.2.1 Time sensitive use cases with wireless connectivity requirements

Time-sensitive use cases that demand wireless connectivity with deterministic and time-critical capabilities are prevalent across various industries. These use cases illustrate the importance of integrating TSN with wireless technology to meet specific requirements. Here is a summary of these use cases:

Industrial Automation: The combination of TSN and wireless connectivity is highly valuable in industrial automation. It facilitates real-time control, synchronisation, and communication within wireless industrial networks. This en-

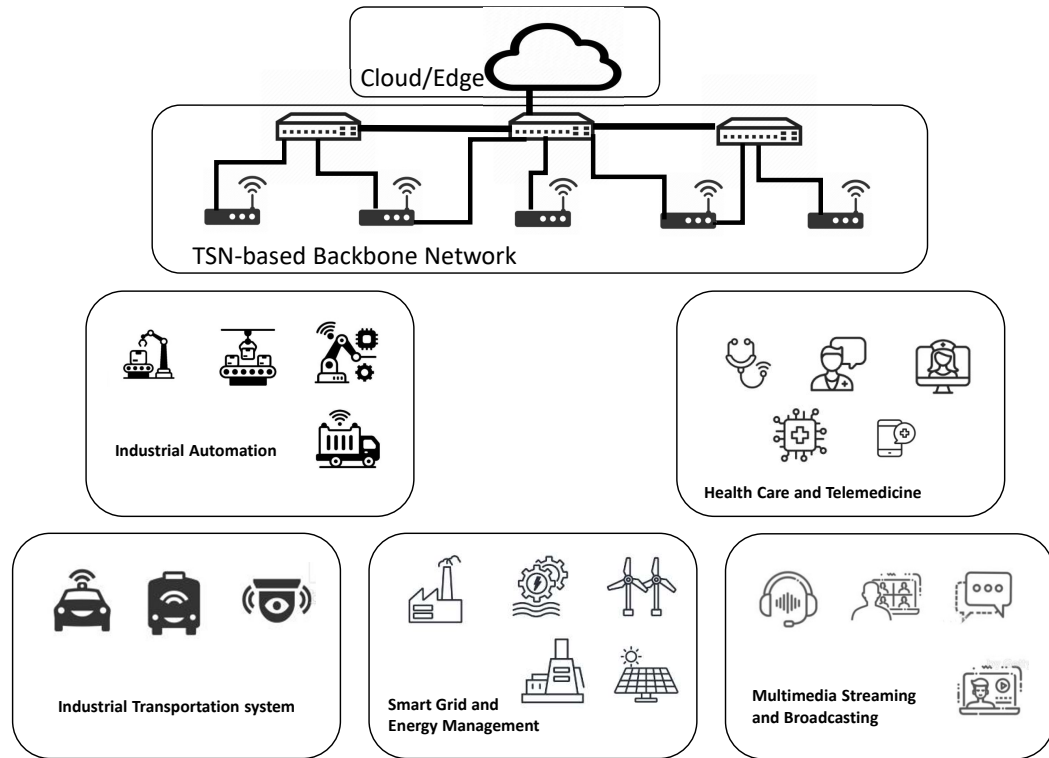


Figure 7.8: TSN-based applications with wireless connectivity requirements

ables precise coordination of machinery, robots, and sensors, supporting applications like synchronised motion control, distributed control systems, and collaborative robotics. TSN ensures the reliable and deterministic transmission of critical control and monitoring data over wireless links, crucial for time-sensitive industrial processes.

Intelligent Transportation Systems (ITS): TSN with wireless connectivity plays a vital role in ITS, where precise timing and communication are essential for safety and efficiency. Wireless TSN enables synchronised communication and coordination among vehicles, roadside infrastructure, and traffic management systems. It supports time-critical applications such as Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, adaptive traffic control, and collision avoidance. TSN-based wireless networks enhance transportation safety and optimise traffic flow.

Smart Grids and Energy Management: TSN-based wireless networks find applications in smart grids and energy management systems. They facilitate real-time monitoring, control, and synchronisation of energy generation, distribution, and consumption processes. TSN allows precise coordination and communication between power generation sources, grid infrastructure, and smart

devices, supporting applications like demand response, energy optimisation, and grid stability. Wireless TSN ensures the reliable and deterministic transmission of critical energy data in these systems.

Multimedia Streaming and Broadcasting: TSN with wireless connectivity also benefits multimedia streaming and broadcasting applications. By combining TSN with wireless links, deterministic and time-sensitive multimedia traffic, including audio and video streaming, can be transmitted reliably and synchronised over wireless networks. This ensures the high-quality and synchronised delivery of media content for applications such as live events, real-time multimedia production, and immersive media experiences.

Healthcare and Telemedicine: TSN-based wireless networks offer potential applications in healthcare and telemedicine. They enable real-time and synchronised communication between medical devices, wearable sensors, and healthcare systems. TSN ensures precise timing and coordination for remote patient monitoring, telemedicine consultations, and the integration of medical devices. Wireless TSN networks facilitate the reliable and timely transmission of vital health data, supporting improved patient care and remote medical services.

These use cases demonstrate the versatility and advantages of TSN-based wireless networks across various industries. The integration of TSN with wireless connectivity provides the determinism and time-critical capabilities needed for real-time control, synchronisation, and communication in industrial automation, intelligent transportation systems, smart grids, multimedia streaming, healthcare, and other domains.

7.2.2 WiFi integration with TSN:

The integration of WiFi/IEEE 802.11 with Time-Sensitive Networking involves merging TSN capabilities and functions with WiFi networks to enable accurate and time-critical communication. In this section, we consider the examination of viable implementation options, with a particular emphasis on handling configuration and management tasks for TSN tools that are compatible with both IEEE 802.11 and the most recent WiFi 6/6E devices.

synchronisation WiFi networks inherently lack determinism and exhibit variable latency in comparison to wired Ethernet networks. Consequently, extending TSN synchronisation to WiFi networks can be a complex yet essential under-

taking, particularly for applications demanding precise timing and low-latency communication in wireless networks. Clock synchronisation plays a pivotal role in maintaining accurate timing for WiFi devices. While WiFi networks can implement synchronisation protocols like the IEEE 802.11v Time Synchronisation Function (TSF), which offers a foundational level of clock synchronisation among WiFi devices, TSN necessitates a higher degree of precision in time synchronisation. To achieve this level of accuracy, techniques such as Precision Time Protocol (PTP) can be harnessed to establish highly accurate clock synchronisation among TSN-capable devices operating over WiFi. Synchronising the clocks of WiFi devices through these methods enables the attainment of the level of synchronisation required for TSN applications.

Scheduling and prioritisation: Several approaches can be employed to enable TSN scheduling functions over WiFi networks. One such approach involves leveraging existing TSN scheduling mechanisms and adapting them to work seamlessly with WiFi networks. This adaptation requires modifying TSN protocols to account for the distinct characteristics of WiFi, including variable latency and interference. The adjusted TSN scheduling algorithms factor in wireless channel conditions and make dynamic adjustments to scheduling parameters to ensure the reliability and determinism of communication. Furthermore, techniques like frame aggregation and prioritisation can be implemented to enhance the efficiency of TSN transmission over WiFi.

Frame aggregation entails the consolidation of multiple TSN frames into a single WiFi frame, reducing overhead and enhancing overall throughput. Prioritisation mechanisms play a pivotal role in giving precedence to TSN traffic over other types of WiFi traffic, thereby ensuring that TSN packets experience reduced latency and are delivered with higher reliability. Additionally, ongoing advancements in WiFi technology, such as the introduction of IEEE 802.11be (WiFi 7) standards, are aimed at providing robust support for real-time and deterministic communication. These standards encompass features like improved channel access mechanisms, enhanced QoS support, and reduced latency, all of which contribute to the extension of TSN scheduling capabilities across WiFi networks.

Quality of service: Extending Time-Sensitive Networking Quality of Service over WiFi networks requires the implementation of mechanisms to prioritise TSN traffic and maintain the necessary service levels, even in the face of challenges posed by wireless communication. Several common approaches have

been proposed for this purpose:

- **WiFi Multimedia (WMM):** WMM is a QoS mechanism that implements features based on the IEEE 802.11e standard. It categorizes Wi-Fi traffic into four Access Categories (ACs) according to priority levels: voice, video, the best effort, and background. While WMM is not a direct implementation of all IEEE 802.11e features, it incorporates the traffic prioritisation and QoS concepts defined by 802.11e. Traffic associated with Time-Sensitive Networking (TSN) can be mapped to either the voice or video ACs, which receive higher priority. This prioritization ensures that TSN packets are given preferential treatment, resulting in reduced latency compared to other traffic categories.
- **Traffic Differentiation:** Effectively distinguishing TSN traffic from other WiFi traffic is a key aspect of ensuring quality of service. Techniques like Differentiated Services Code Point (DSCP) marking or VLAN tagging can be applied to label TSN packets with specific identifiers. WiFi access points and routers can then be configured to recognise these markings and prioritise TSN traffic accordingly.
- **Enhanced QoS Mechanisms:** Enhancements to the existing QoS mechanisms in WiFi networks are necessary to better support TSN requirements. These improvements may encompass adjustments to contention-based channel access methods, frame aggregation techniques, and traffic scheduling algorithms. By customising these mechanisms to prioritise TSN traffic, it becomes possible to minimise latency and jitter, ultimately achieving deterministic communication.
- **Admission Control and Resource Reservation:** To ensure that TSN traffic consistently receives the desired level of service, the use of admission control and resource reservation mechanisms is pivotal. Prior to allowing TSN traffic into the network, admission control checks can be conducted to ascertain the availability of adequate resources that meet the QoS requirements. Reservation protocols, such as RSVP (Resource Reservation Protocol), can be employed to allocate dedicated bandwidth or resources exclusively for TSN traffic, effectively guaranteeing its performance.

Compatibility with Ethernet-based TSN networks: Both WiFi TSN and Ethernet-based TSN networks adhere to the same IEEE TSN standards. WiFi TSN devices can employ identical frame formats and protocols as their Ethernet-

based TSN counterparts, ensuring complete compatibility between the two. TSN relies on Ethernet-based protocols, including IEEE 802.3 Ethernet and IEEE 802.1 Ethernet Bridging, for frame transmission and network communication. In the case of WiFi TSN devices, they can encapsulate TSN frames within WiFi frames by utilising appropriate headers. This approach allows for the seamless transmission of TSN traffic over WiFi networks while maintaining compatibility with Ethernet-based TSN devices.

Moreover, the network management tools and configuration interfaces originally designed for Ethernet-based TSN networks can be extended to include WiFi TSN devices. This extension enables network administrators to efficiently manage and configure WiFi TSN devices within the existing TSN network infrastructure. Consequently, a unified management framework is established, ensuring compatibility and smooth integration with Ethernet-based TSN networks.

7.2.3 Simulation setup and discussion

To assess the advantages of integrating WiFi-TSN, we created a sample setup for a collaborative robot application, illustrated in Figure 7.9. We used MATLAB's WLAN Toolbox to evaluate this scenario based on the parameters outlined in Table 7.1. Our analysis focuses on an industrial floor environment where a single Access Point (AP) with a communication range of $R = 30m$ is deployed. This AP serves as a gateway, facilitating traffic routing between the wired TSN, WiFi-TSN, the central network, and user configuration units. It connects to a backbone network through a TSN switch.

Within this industrial floor, a variety of mobile and fixed collaborative robots, along with wireless inspection units, are deployed. These robots and inspection units are randomly distributed within the communication range of the AP. Additionally, the floor accommodates logic controllers, human-machine interfaces, and Ethernet devices, all interconnected with the wired TSN backbone. This setup allows us to evaluate the impact and benefits of integrating WiFi-TSN in this industrial context.

In the network, traffic flows are prioritised based on their assigned priority levels, regardless of whether they originate from wired or wireless end devices. These traffic types are divided into various classes, which are defined by their respective priority levels, as described in 3.5. Every network device, including

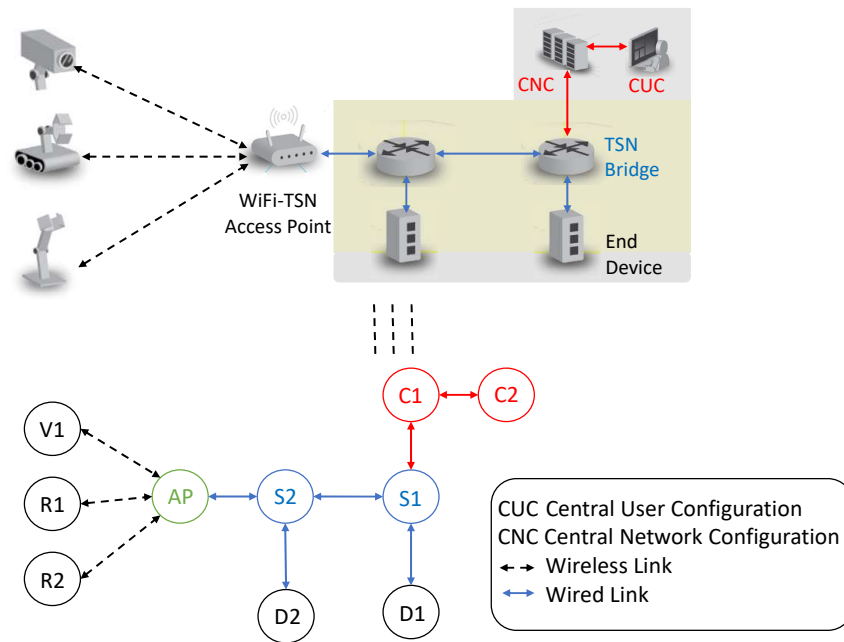


Figure 7.9: WiFi-TSN network to support collaborative robots based industrial automation

Access Points (APs) and TSN switches, supports four access categories: Critical Data Traffic (CDT), Video Streaming Traffic (VST), Safety Data Traffic (SDT), and Non-Critical Traffic (NCT). Additionally, these devices implement two distinct traffic shaping policies:

- Without Shaping and prioritisation: In this policy, packets are transmitted strictly in the order they arrive, without considering their type or priority level.
- Pre-emptive or Credit-Based with prioritisation: This policy prioritises the transmission of both CDT and VST packets within the network. CDT packets are granted the highest priority, causing interruptions and suspensions of other traffic types until the CDT transmission is completed, and its queue is emptied. However, this approach introduces additional delays due to the overheads associated with the fragments (N_f fragments) of the pre-empted packet. Similarly, VST packets are transmitted ahead of SDT and NCT traffic types, but are contingent on the accumulation of sufficient credit. When enough credit has accumulated, VST packets receive priority; otherwise, they must wait until lower-priority traffic transmissions have accumulated adequate credit. This credit-based mechanism ensures

Table 7.1: WiFi-TSN configuration and simulation parameters

Deployment Parameters	Description	Value
R	Environment communication range radius	30 m
N_{AP}	Number of Access Points	1
N_{SW}	Number of TSN switches	2
N_{CDT}	Number of critical data flows traversing the network	10
N_V	Number of Video streams traversing the network	[2, 4, 6, 8, 10]
N_{SDT}	Number of event based safety data traffic flows traversing the network	[5, 10]
N_{NCT}	Number of Non-time critical flows traversing the network	[5, 10]
P_{AP}	Access Point position	Fixed
P_{ES}	End devices positions	Random
PHY and MAC Parameters	Description	Value
f_C	Operating frequency	5 GHz
BW	Channel Bandwidth	40 MHz
P_t	Transmission power of the Access Point	20 dBm
$PL_{TMB}(d)$	TMB path loss model	See [ACB19]
T_{oh}	Preemption Overhead time	20 μ s
N_f	Number of fragments of a preemptable packet	Variable
Traffic Parameters	Description	Value
L_{CDT}	Critical data traffic packet length	128 Bytes
L_V	Video streams packet length	1500 bytes
L_{SDT}	Safety data traffic packet length	500 bytes
L_{NCT}	Non-time critical packet length	1500 bytes
-	Packet Arrival process	Pareto
R_{CDT}	Critical traffic data rate	2 Mbps
R_V	Video stream data rate	8 Mbps
R_{SDT}	Safety traffic data rate	variable
R_{NCT}	Non-time critical traffic data rate	10Mbps
Application Parameters	Description	Value
t_{CDT}	deadline for control data flow packets	10 ms
t_V	Guaranteed end to end latency for video streams	$\sim$ 60 ms
t_{SDT}	Safety traffic maximum delay	$\sim$ 50 ms
t_{NCT}	Non-Critical traffic maximum delay	$\leq$ 100 ms

efficient utilisation of network resources while granting preference to VST packets when appropriate.

To thoroughly examine the treatment of traffic flows from end devices, we conducted an extensive analysis comprising twenty different network configurations. These configurations encompassed various combinations of traffic flows and were defined based on the following parameters: $N_V = \{2, 4, 6, 8, 10\}$, $N_{SDT} = \{5, 10\}$, $N_{NCT} = \{5, 10\}$, and $N_{CDT} = 10$.

To ensure the robustness and accuracy of our evaluation, each configuration underwent 100 simulation runs. In each run, the locations of mobile robots were dynamically altered. We employed the TMB (Three-slope Multi-wall and Body) path loss model designed for 5 GHz indoor scenarios [ACB19], along with the MCS (Modulation and Coding Scheme) table from the IEEE 802.11ax standard. It's important to note that IEEE 802.11be was expected to adopt the MCS table from IEEE 802.11ax and enhance it with additional modes derived from its PHY layer improvements. These enhancements included support for up to 16 spatial streams, higher order modulation (4096-QAM), and wider channels (320 MHz). These advancements played a significant role in automatically

computing the data rate for each user in the specified configurations.

The performance evaluation is focused on analysing the average latencies of CDT, VST, SDT, and NCT flows under different network configurations and traffic shaping policies. The results, illustrated in the following figures, are summarized as follows:

Figure 7.10 shows the average latency of Critical Data Traffic (CDT) under the two traffic shaping policies. The following observations are made:

- With Prioritisation: Latencies are consistently around 10 ms, with low variability, indicating that the prioritisation policy effectively meets the 10 ms deadline for CDT flows.
- Without Prioritisation: Higher median latencies and greater dispersion are observed, with some latencies exceeding the 10 ms target, indicating delays and potential deadline misses.

Figure 7.11 displays the average latency for Video Streaming Traffic (VST). As depicted,

- With Prioritisation: The median latency is lower and more stable, with fewer outliers, reflecting more predictable and efficient handling of VST flows.
- Without Prioritisation: Higher median latency and increased variability are observed, particularly in higher load scenarios, indicating delays and less predictable performance

Moreover, figure 7.12 shows the average latency for Safety Data Traffic (SDT). It shows that:

- With Prioritisation: Latency remains consistently below the 50 ms deadline, with narrow interquartile ranges and few outliers, demonstrating effective management within the required QoS parameters.
- Without Prioritisation: Higher median latencies, with many exceeding the 50 ms deadline, particularly in high-load scenarios, showing that prioritisation significantly improves performance for safety-critical traffic.

Finally, figure 7.13 illustrates the average latency for Non-Critical Traffic (NCT), where the following remarks are made:

- With Prioritisation: Median latency is lower with reduced variability,

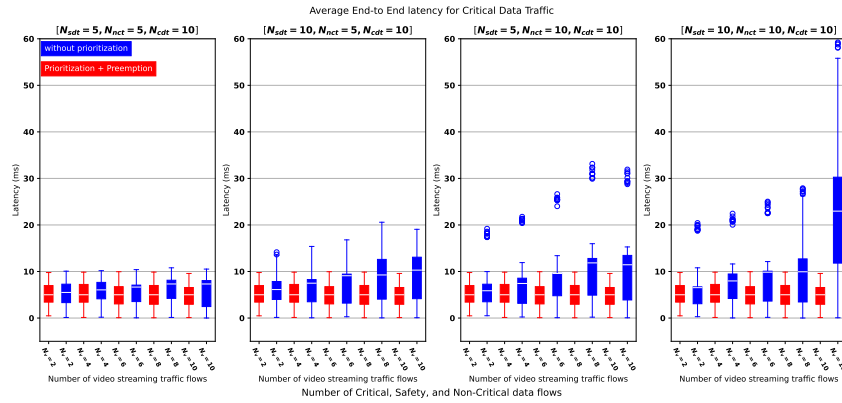


Figure 7.10: Industrial automation using collaborative robots: Average latency of CDT traffic flows.

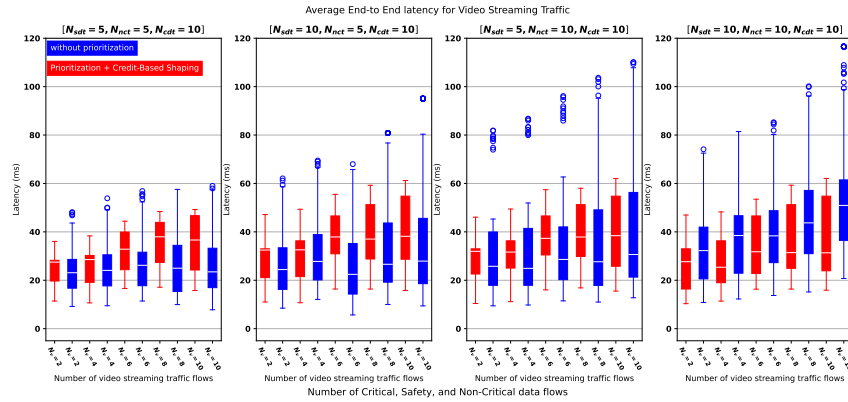


Figure 7.11: Industrial automation using collaborative robots: Average latency of VST traffic flows.

showing improved regulation and performance.

- Without Prioritisation: Higher median latency and greater dispersion are observed, indicating less predictable performance and increased delays for non-critical traffic.

Overall, those figures demonstrate that prioritisation using access levels and credit-based shaping policies substantially improve network performance by reducing latency and enhancing predictability for both critical and non-critical traffic types. In contrast, the absence of prioritisation leads to increased latency and variability, adversely affecting time-sensitive communications.

7.3 Conclusion

In this section, we've covered the TSN integration with both software-defined networks and IEEE 802.11 technologies. Through rigorous exploration and ex-

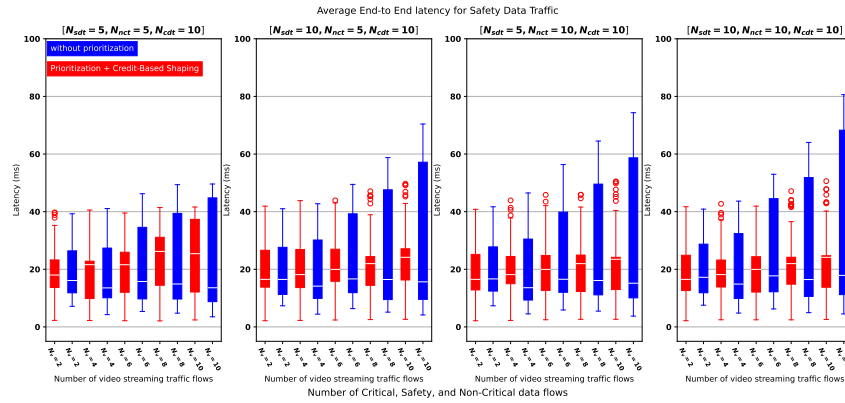


Figure 7.12: Industrial automation using collaborative robots: Average latency of SDT traffic flows.

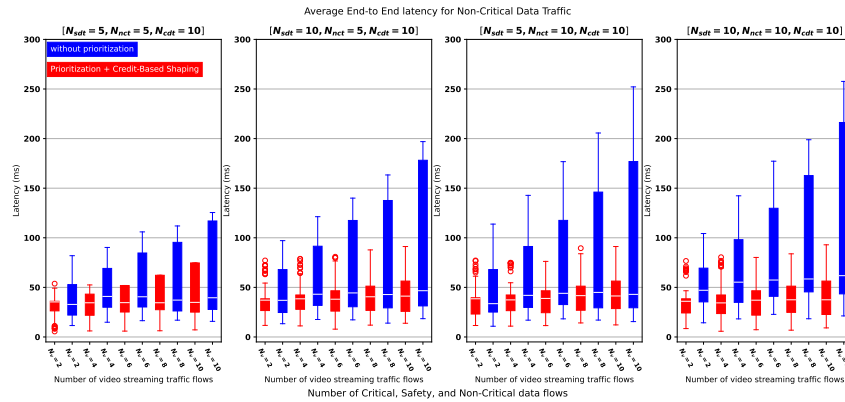


Figure 7.13: Industrial automation using collaborative robots: Average latency of NCT traffic flows.

perimentation, we've uncovered the immense potential that TSN holds for a wide array of industries and use cases, particularly within the context of industrial automation, intelligent transportation systems, and beyond.

In the first part, we introduced the concept of Software-Defined Time-Sensitive Networks, a powerful approach designed to maximise network resource management and ensure Quality of Service in time-critical industrial network systems. By leveraging software-defined network concepts and carefully designed scheduling algorithms, SD-TSN not only guarantees real-time performance for TSN flows but also optimises network utilisation by accommodating other traffic types. Our prototype implementation, evaluated through various scenarios, has proven the efficiency of SD-TSN in computing precise transmission schedules, ensuring bounded latency for TSN flows, and scaling gracefully to handle many flows within minimal run-time. Our future work aims to extend this model to include wireless devices and incorporate accurate end device and channel

models for even more reliable results.

The second part of our exploration focused on the integration of TSN with IEEE 802.11 technology, with a special emphasis on IEEE 802.11be and its potential to revolutionise low-latency communications. By offering a well-defined time-sensitive operation mode, TSN-integrated WiFi opens up new possibilities across various IoT sectors, promising real-time capabilities and low-latency communications. While WiFi may not ensure absolute determinism, we discussed strategies to mitigate latency, such as dynamic spectrum access, cooperative access point approaches, and prioritisation mechanisms. Our investigation highlighted the importance of meticulous planning, configuration, and management in successfully integrating TSN with WiFi. While this combination is ideal for scenarios requiring a hybrid network with real-time and general-purpose connectivity, it may not suit ultra-critical applications demanding absolute determinism. Collaboration among standardisation bodies, industry stakeholders, and researchers is critical for refining this integration and overcoming challenges such as latency and reliability.

To unlock the full potential of TSN-integrated WiFi, a holistic approach is essential, ensuring seamless coexistence and optimising performance. This requires continued research and innovation, as the integration of TSN and wireless technology holds great promise for industries ranging from industrial automation to healthcare. As these technologies evolve and intersect, they will shape the future of the Internet of Things, enabling a wide range of applications that depend on both real-time performance and reliable connectivity.

Chapter 8

Conclusion and Future Directions

8.1 Conclusion

This thesis has made significant progress in understanding, analysing, and applying TSN (Time-Sensitive Networking) in industrial networks. Through a series of detailed studies and analyses, it offers valuable insights and solutions to optimise communication, reliability, and performance in industrial automation settings. Reflecting on this research, several key conclusions and insights stand out::

1.Optimising Industrial Networks with TSN Queuing Mechanisms: The exploration of TSN (Time-Sensitive Networking) queuing mechanisms highlights their essential role in optimizing industrial networks. By extending the INET framework and developing a TSN-based network architecture for the Quality Control After Production use case, including an in-depth performance analysis of various queuing models. This work helps pinpoint the most effective queuing strategies tailored to specific applications and configurations, thereby enhancing communication reliability and overall network performance.

The findings indicate that while a FIFO (First-In-First-Out) configuration may be adequate for basic traffic management, more advanced setups like PRIO (Priority), TAS (Time-Aware Shaping), and CBS (Credit-Based Shaping) offer substantial benefits. These benefits include better deadline adherence, delay control, and reliable delivery of critical traffic. Although PRIO prioritizes critical traffic, it can struggle with lower-priority traffic and may not consistently meet strict deadlines. On the other hand, TAS and CBS configurations show superior performance by providing deterministic delay bounds and maintain-

ing high delivery ratios for all traffic types, including critical, monitoring, and web traffic. Additionally, TAS and CBS configurations prove resilient under realistic stochastic traffic models, managing random traffic behaviour changes effectively. They ensure reliable delivery for critical traffic while accommodating other traffic types without compromising service quality.

In summary, the assessment emphasizes the need for carefully selecting network configurations based on specific traffic requirements and operational needs. TAS and CBS configurations stand out as promising solutions for achieving robust and efficient network performance, particularly in critical infrastructure environments where meeting deadlines and ensuring reliable delivery are crucial. These insights can inform future network design and management strategies, helping implement optimized configurations to address the evolving demands of modern communication systems.

2. Advancing Converged Networks with TSN:

Through the performance analysis of TSN-based converged networks, this study have demonstrated the feasibility and effectiveness of deterministic communication mechanisms within unified network infrastructures. This research contributes to the ongoing efforts in interoperability, by providing valuable guidance for deploying TSN in complex network environments that manage various types of data traffic.

3. Enhancing Timing Analysis for Industrial TSN Profiles: The timing analysis of various traffic shaping methods within the Industrial TSN Profile provides crucial insights for optimizing industrial networking applications. By utilising Network Calculus, this study carefully assessed the performance and effectiveness of these shaping techniques. This analysis has allowed us to establish more accurate latency bounds for different data traffic classes, taking into account factors such as traffic class grouping, interface selection, and redundant network architectures.

The study includes a thorough evaluation of end-to-end delay performance in a QCAP (Quality Control After Production) network scenario, using various simulation techniques and models. It demonstrates that the TSN/NC framework offers tighter delay bounds compared to traditional per-hop IR NC models. By addressing key factors like preemption, traffic prioritization, and network configurations, this study shows how the framework improves performance guarantees for critical data traffic in industrial automation scenarios like QCAP.

Moreover, the experiments highlight the importance of strategic network design choices, such as adjusting offset periods for scheduled traffic transmissions and enhancing critical links with greater bandwidth. These optimizations significantly reduce worst-case delay bounds, ensuring the efficient operation of essential processes within the QCAP use case. Additionally, our analysis of AVB (Audio Video Bridging) traffic idle slopes reveals the complex interactions between different traffic classes and their effects on delay bounds, offering valuable guidance for network design and optimization in real-world industrial settings. Overall, this study advances TSN-based network design methodologies for industrial automation, providing practical strategies to enhance performance and reliability in demanding applications like QCAP.

4. Improving Network Resilience with TSN-based Redundancy: The evaluation of TSN-based network redundancy has revealed effective strategies for minimizing packet losses and ensuring continuous operation in industrial environments. By examining various redundancy mechanisms, it offers valuable insights for improving network resilience against failures and disruptions, meeting the high availability requirements of industrial applications.

Key observations from our findings include:

1. **Isochronous Traffic:** The Worst Case Delay (WCD) is mainly influenced by the number of hops along the path, regardless of other traffic loads.
2. **Non-Isochronous, AVB, and Best Effort Traffic:** Optimal load distribution across different paths can significantly reduce WCDs. This to identify the minimum WCDs for various traffic classes and determined the best load distribution strategy for achieving lower WCD bounds.
3. **Failure Scenarios:** WCD upper bounds are explored in failure scenarios, such as when all frames travel the same path. Then, the maximum WCDs in these situations could be identified.

In the synthetic evaluation, effective strategies are highlighted for achieving lower WCDs for industrial traffic. These strategies include:

- **Traffic Rearrangement:** Directing traffic along different paths with optimal loads.
- **Time Offsets:** Setting offsets for each flow to reduce queuing delays.
- **Accumulation Rates:** Choosing optimal accumulation rates for AVB traffic while ensuring lower-priority traffic is not compromised.

These methods demonstrate practical ways to optimise network configurations and achieve lower WCD bounds across various traffic classes.

5. Optimising Network Management with SDN and TSN Integration: Integrating SDN (Software-Defined Networking) principles with TSN represents a major shift, offering centralized control and programmability to enhance Quality of Service and traffic management in industrial communication environments. The proposed TSN network management architecture, built on SDN principles, introduces innovative methods for traffic scheduling and network configuration. By implementing and evaluating this approach using the Mininet network emulation platform, it is concluded that managing TSN industrial networks with SDN can effectively meet real-time requirements.

SD-TSN (SDN-enabled TSN) proves to be a robust solution for achieving deterministic and low-latency network performance, particularly in industrial settings where real-time guarantees are critical. It maintains strict adherence to schedules, ensuring precise packet placement and minimal deterministic delays, resulting in very low end-to-end latency—averaging $9.64 \mu s$ with a standard deviation of $2.32 \mu s$. Additionally, SD-TSN efficiently supports mixed traffic types, upholding strict latency requirements for real-time traffic while accommodating best-effort traffic with reasonable latency levels.

The scalability assessment further highlights SD-TSN's effectiveness in handling larger traffic loads. Scheduling runtimes for over 100 flows average around 10 seconds in a medium-sized network, in stark contrast to NC-TSN (Network Calculus-based TSN), which experiences longer scheduling times and greater latency variations due to its need to explore the entire solution space for path availability. Thus, SD-TSN not only delivers consistent, low-latency performance but also scales efficiently, making it a strong candidate for modern industrial networks where both performance and flexibility are essential. Overall, these findings confirm that SD-TSN is well-suited to meet the demanding requirements of industrial environments, positioning it as a promising solution for next-generation time-sensitive networking applications.

6. Extending TSN to Support Wireless Applications:

The investigation into extending TSN (Time-Sensitive Networking) to wireless applications addresses the growing need for wireless connectivity across various industries. By developing mechanisms and protocols to integrate TSN with wireless technologies, this is to enhance flexibility and mobility in industrial

communication environments. This integration allows for dynamic reconfiguration of network topologies and application deployment, improving adaptability and responsiveness in industrial settings.

The findings highlight the critical importance of prioritizing essential traffic flows, such as control and safety data, to ensure timely and reliable communication in dynamic industrial environments. Additionally, it is observed that a positive correlation between video streaming latency and other traffic flows, emphasizing the need for efficient traffic management strategies to reduce latency variations and maintain consistent performance. The integration of WiFi-TSN not only ensures seamless communication between wired and wireless devices but also optimizes network resources to meet diverse traffic demands effectively.

Overall, this study advances the understanding of WiFi-TSN integration in industrial automation and underscores the importance of traffic prioritization and shaping policies to achieve predictable and bounded latency for time-sensitive applications. Future research and development in this area will be crucial for enhancing network performance, scalability, and robustness, supporting the progress of Industry 4.0 and smart manufacturing initiatives.

In conclusion, this thesis provides practical insights for optimising industrial network infrastructures. Simulating TSN queuing and shaping mechanisms reveals optimal configurations to support both critical and non-critical traffic, ensuring efficient and reliable communication. Utilising the network calculus framework offers a robust methodology for analysing industrial traffic and determining the worst-case end-to-end delay, which is crucial for maintaining system performance and reliability. Furthermore, integrating SDN and Wi-Fi with TSN enhances network management, flexibility, and scalability. These findings can be applied to improve network performance, reliability, and overall system efficiency in industrial automation settings.

8.2 Future directions

As this thesis concludes, it opens up several exciting avenues for future research that can significantly advance TSN, industrial automation, and the integration of complementary technologies. Here are key directions for further exploration:

1. Flows organisation and prioritisation: Scheduling flows is a critical chal-

lenge, especially with high traffic volumes, dynamic connections, and long-distance transmissions. Developing scalable and robust traffic scheduling solutions, such as those based on Credit-Based Fair Queuing (CBFQ) methods, is essential. Further efforts are needed to tackle specific issues like time interval configurations, traffic granularity adjustments, and optimization of queuing mechanisms. The TSN analytic framework provides valuable support for designing and evaluating TSN traffic schedulers, helping to identify and resolve performance bottlenecks and optimization opportunities.

2. Service Level Assurance: Ensuring service level assurance is crucial for meeting the diverse needs of various industrial applications. Research into resource allocation and reservation is fundamental for optimizing network configurations to support TSN flows with distinct Quality of Service (QoS) demands. Efficient resource management and bandwidth reservation techniques enable network administrators to prioritize critical traffic, reduce latency, and establish reliable communication channels for applications in industrial automation, automotive systems, and real-time control. Future research could explore dynamic resource allocation algorithms, efficient routing protocols, and advanced traffic shaping mechanisms to enhance network performance and QoS.

3. Comprehensive models for the credit evolution and TSN pre-emption mode: There is a need for further research into how credits are allocated and used within TSN networks, particularly in systems using various shaping techniques. Studying the dynamic evolution of credits, including replenishment rates, consumption patterns, and the impact of network conditions on credit availability, can help improve credit-based scheduling mechanisms. Additionally, investigating TSN pre-emption mode, especially in meeting strict QoS requirements and prioritizing certain data flows, is crucial. Research should focus on pre-emption thresholds, policies, and their effects on network performance to develop effective pre-emptive scheduling strategies.

4. Integration with other technologies: Developing integration solutions that ensure TSN guarantees within heterogeneous networks is a significant research focus. For example, deterministic wireless networks are vital for mission-critical applications requiring reliable communication across wireless channels. Incorporating TSN capabilities into 5G networks is also essential for enabling deterministic communication, real-time data delivery, and dynamic resource allocation. Research into time-sensitive packet forwarding, traffic shaping, and network slicing will be crucial for advancing deterministic communication and

ensuring predictable performance for critical applications in industrial automation, automotive systems, and multimedia streaming.

These future research directions highlight the evolving nature of TSN, industrial automation, and related technologies. Addressing these areas will contribute to ongoing innovation and growth in the field, furthering the vision of precision, reliability, and connectivity in industrial environments.

Nomenclature

Abbreviations

AGV	Automated Guided Vehicles
AVB	Audio Video Bridging
BE	Best Effort
CAN	Controller Area Network
CBR	Constant Bit Rate
CBS	Credit Based Shaping
CDT	Control Data Traffic
CPS	Cyber-Physical Systems
eMac	Express Frame
FIFO	First In First Out
GCL	Gates Control List
HMI	Human Machine Interface
HSR	High-availability Seamless Redundancy
IIoT	Industrial Internet of Things
ILP	Integer Linear Program
IT	Information Technology
LCM	Least Common Multiple
MAC	Media Access Control
NC	Network calculus

NOS	Network Operating System
NPPS	Non-Preemptive Priority Scheduling
NTCT	Non-Time-Critical Traffic
OT	Operational Technology
PCP	Priority Code Point
PLC	Programmable Logic Controller
pMac	Preemptable Frame
PPS	Preemptive Priority Scheduling
PRIO	Priority Queuing
PRP	Parallel Redundancy Protocol
PTP	Precision Time Protocol
QCAP	Quality Checks After Production
QCAP	Quality Control and Classification After Production
QoS	Quality of Service
RC	Rate-Constrained
RSVP	Resource Reservation Protocol
RTC	Real-Time Calculus
RTE	Real Time Ethernet
RTP	Real-time Transport Protocol
SDN	Software Defined Networks
SMT	Satisfiability Modulo Theories
SoE	Sequence of Events
SRP	Stream Reservation Protocol
ST	Scheduled Traffic
TAS	Time Aware shaping
TCT	Time-Critical Traffic
TSF	Time Synchronization Function

TSN Time-Sensitive Networking

TT Time-Triggered

UDP User Datagram Protocol

WCD Worst Case Delay

WCD Worst-Case Delay

WiFi Wireless Fidelity

Variables

H_{credit} Maximum amount of possible accumulated credits.

I_n Accumulation rate.(in kilobits/second), $n \in (A, B)$

L_{credit} Minimum amount of credits that can be reached.

R_{port} Port transmission rate.

R_{Tx} Depletion rate.(in kilobits/second), $n \in (A, B)$

S_i Maximum size of any burst of traffic that can delay available for transmission frame.

S_{max} Maximum frame size. (in bytes)

T_{min} Minimum inter-frame intervals. (in milliseconds)

TC One of the eight industrial TSN profile traffic classes.

References

- [ABC⁺23] Eneko Artetxe, Oscar Barambones, Isidro Calvo, Pablo Fernández-Bustamante, Imanol Martin, and Jokin Uralde. Wireless technologies for industry 4.0 applications. *Energies*, 16(3), 2023.
- [ACB19] Toni Adame, Marc Carrascosa, and Boris Bellalta. The tmb path loss model for 5 ghz indoor wifi scenarios: On the empirical relationship between rssi, mcs, and spatial streams. In *2019 Wireless Days (WD)*, pages 1–8. IEEE, 2019.
- [All23] Wi-Fi Alliance. Wi-fi certified 6: Capacity, efficiency, and performance for advanced connectivity. Technical report, Wi-Fi Alliance, 2023.
- [AML⁺17] Mohammad Ashjaei, Saad Mubeen, John Lundbäck, Mattias Gålnder, Kurt-Lennart Lundbäck, and Thomas Nolte. Modeling and timing analysis of vehicle functions distributed over switched ethernet. In *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 8419–8424. IEEE, 2017.
- [AMS⁺21] Mahin K Atiq, Raheeb Muzaffar, Óscar Seijo, Iñaki Val, and Hans-Peter Bernhard. When ieee 802.11 and 5g meet time-sensitive networking. *IEEE Open Journal of the Industrial Electronics Society*, 3:14–36, 2021.
- [APA5] WHICH STANDARD STANDS APART. 5 real-time, ethernet-based fieldbuses compared. 5.
- [BARW11] Andrzej Borys, Mariusz Aleksiewicz, Dariusz Rybarczyk, and Katarzyna Wasielewska. Some principles of network calculus revisited. *International Journal of Electronics and Telecommunications*, 57(3):279–284, 2011.

- [BJT10] Anne Bouillard, Laurent Jouhet, and Éric Thierry. Comparison of different classes of service curves in network calculus. *IFAC Proceedings Volumes*, 43(12):306–311, 2010.
- [BS19] Lucia Lo Bello and Wilfried Steiner. A perspective on iee time-sensitive networking for industrial communication and automation systems. *Proceedings of the IEEE*, 107(6):1094–1120, 2019.
- [BSB⁺19] Dietmar Bruckner, Marius-Petru Stănică, Richard Blair, Sebastian Schriegel, Stephan Kehrer, Maik Seewald, and Thilo Sauter. An introduction to opc ua tsn for industrial communication systems. *Proceedings of the IEEE*, 107(6):1121–1131, 2019.
- [CABI⁺15] Martin Casado, Keith Eric Amidon, Peter J Balland III, Natasha Gude, Justin Pettit, Benjamin Levy Pfaff, Scott J Shenker, and Daniel J Wendlandt. Network operating system for managing and securing networks, July 14 2015. US Patent 9,083,609.
- [COCS16] Silviu S Craciunas, Ramon Serna Oliver, Martin Chmelfk, and Wilfried Steiner. Scheduling real-time communication in iee 802.1 qbv time sensitive networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, pages 183–192, 2016.
- [Doy04a] Paula Doyle. Introduction to real-time ethernet i. *Contemporary Control Systems*, 5(4), 2004.
- [Doy04b] Paula Doyle. Introduction to real-time ethernet ii. 2004.
- [DSA⁺14] Peter Danielis, Jan Skodzik, Vlado Altmann, Eike Bjoern Schweissguth, Frank Golasowski, Dirk Timmermann, and Joerg Schacht. Survey on real-time communication via ethernet in industrial automation environments. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–8. IEEE, 2014.
- [DXL⁺22] Libing Deng, Guoqi Xie, Hong Liu, Yunbo Han, Renfa Li, and Keqin Li. A survey of real-time ethernet modeling and design methodologies: From avb to tsn. *ACM Computing Surveys (CSUR)*, 55(2):1–36, 2022.

- [FBM⁺19] N Finn, JYL Boudec, E Mohammadpour, J Zhang, B Varga, and J Farkas. Detnet bounded latency. *IETF draft work-in-progress*, 2019.
- [Fel05] Max Felser. Real-time ethernet-industry prospective. *Proceedings of the IEEE*, 93(6):1118–1129, 2005.
- [FFTB17] János Farkas, Norman Finn, Patricia Thaler, and Ericsson Huawei Broadcom. Ieee 802.1 time-sensitive networking (tsn). Technical report, Tech. Rep, 2017.
- [FHC⁺19] Jonathan Falk, David Hellmanns, Ben Carabelli, Naresh Nayak, Frank Dürr, Stephan Kehrer, and Kurt Rothermel. Nesting: Simulating ieee time-sensitive networking (tsn) in omnet++. In *2019 International Conference on Networked Systems (NetSys)*, pages 1–8. IEEE, 2019.
- [FTVF19] N Finn, P Thubert, B Varga, and J Farkas. Rfc 8655: Deterministic networking architecture, 2019.
- [G⁺21] IEEE 802.1 Working Group et al. Iec/ieee 60802 tsn profile for industrial automation, 2021.
- [GH12] Brendan Galloway and Gerhard P Hancke. Introduction to industrial control networks. *IEEE Communications surveys & tutorials*, 15(2):860–880, 2012.
- [GRK16] Jochen W. Guck, Martin Reisslein, and Wolfgang Kellerer. Function split between delay-constrained routing and resource allocation for centrally managed qos in industrial networks. *IEEE Transactions on Industrial Informatics*, 12(6):2050–2061, 2016.
- [GZPS17] Voica Gavrilit, Bahram Zarrin, Paul Pop, and Soheil Samii. Fault-tolerant topology and routing synthesis for ieee time-sensitive networking. In *Proceedings of the 25th International Conference on Real-Time Networks and Systems*, pages 267–276, 2017.
- [GZRP18] Voica Gavrilit, 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.
- [HDSD16] Matthias Herlich, Jia Lei Du, Fabian Schörghofer, and Peter Dorfinger. Proof-of-concept for a software-defined real-time eth-

- ernet. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4. IEEE, 2016.
- [HGO16] Peter Heise, Fabien Geyer, and Roman Obermaisser. Tsimnet: An industrial time sensitive networking simulation framework based on omnet++. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE, 2016.
- [HR21] Duc NM Hoang and Jong Myung Rhee. Comparative analysis of iec 62439–3 (hsr) and ieee 802.1 cb (frer) standards. In *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 231–235. IEEE, 2021.
- [IEE10] 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.
- [IEE13] Ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements–part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications–amendment 4: Enhancements for very high throughput for operation in bands below 6 ghz. *IEEE Standards Association*, 2013.
- [IEE17] Ieee standard for local and metropolitan area networks–frame replication and elimination for reliability. *IEEE Std 802.1CB-2017*, pages 1–102, 2017.
- [IEE18] Iso/iec/ieee international standard – information technology – telecommunications and information exchange between systems – local and metropolitan area networks – specific requirements – part 1q: Bridges and bridged networks amendment 3: Enhancements for scheduled traffic. *ISO/IEC/IEEE 8802-1Q:2016/Amd.3:2017(E)*, pages 1–62, 2018.
- [IEE21a] Iec/ieee international standard - precision clock synchroniza-

- tion protocol for networked measurement and control systems. *IEC/IEEE 61588-2021*, pages 1–504, 2021.
- [IEE21b] Ieee 802.11ax-2021: Ieee standard for information technology—telecommunications and information exchange between systems—local and metropolitan area networks—specific requirements—part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications—amendment 6: Enhancements for high efficiency wireless. *IEEE Standards Association*, 2021.
- [Jia09] Yuming Jiang. Network calculus and queueing theory: Two sides of one coin: Invited paper. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, Brussels, BEL, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).
- [JLC13] Wen-Kang Jia, Gen-Hen Liu, and Yaw-Chung Chen. Performance evaluation of ieee 802.1 qbu: Experimental and simulation results. In *38th Annual IEEE Conference on Local Computer Networks*, pages 659–662. IEEE, 2013.
- [JLH⁺18] Junhui Jiang, Yuting Li, Seung Ho Hong, Aidong Xu, and Kai Wang. A time-sensitive networking (tsn) simulation model based on omnet++ . In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 643–648. IEEE, 2018.
- [Kál] György Kálmán. Mass configuration of network devices in industrial environments. In *The Twelfth International Conference on Networks (ICN 2013)*, pages 107–111.
- [Kin21] Arnold Baraka Doste Kinabo. Time sensitive networking for wi-fi based wireless industrial environments. 2021.
- [KK15] Ikhwan Kim and Taehyoun Kim. Guaranteeing isochronous control of networked motion control systems using phase offset adjustment. *Sensors*, 15(6):13945–13965, 2015.
- [KKH14] Stephan Kehrer, Oliver Kleineberg, and Donal Heffernan. A comparison of fault-tolerance concepts for ieee 802.1 time sensitive

- networks (tsn). In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–8. IEEE, 2014.
- [KS22] Hermann Kopetz and Wilfried Steiner. Real-time communication. In *Real-time systems: Design principles for distributed embedded applications*, pages 177–200. Springer, 2022.
- [Lai15] Harri Laine. Simulating time-sensitive networking. *Technical University of Denmark, Lyngby, Denmark*, 2015.
- [LB96] Jean-Yves Le Boudec. Network calculus made easy. Technical report, EPFL Departement d’Informatique (DI), 1996.
- [LBC12] Jörg Liebeherr, Almut Burchard, and Florin Ciucu. Delay bounds in communication networks with heavy-tailed and self-similar traffic. *IEEE Transactions on Information Theory*, 58(2):1010–1024, 2012.
- [LBT01] Jean-Yves Le Boudec and Patrick Thiran. *Network calculus: a theory of deterministic queuing systems for the internet*. Springer, 2001.
- [LHP⁺21] David Lou, Jan Holler, Dhruvin Patel, Ulrich Graf, and Matthew Gillmore. The industrial internet of things networking framework. *Industrial IoT Consortium*, 2021.
- [LLW⁺17] Xiaomin Li, Di Li, Jiafu Wan, Athanasios V Vasilakos, Chin-Feng Lai, and Shiyong Wang. A review of industrial wireless networks in the context of industry 4.0. *Wireless networks*, 23:23–41, 2017.
- [LNS⁺15] M Laner, N Nikaein, P Svoboda, M Popovic, D Drajić, and S Krco. Traffic models for machine-to-machine (m2m) communications: Types and applications. In *Machine-to-machine (M2M) Communications*, pages 133–154. Elsevier, 2015.
- [LW04] Ulf Lamping and Ed Warnicke. Wireshark user’s guide. *Interface*, 4(6):1, 2004.
- [LWH11] Hyung-Taek Lim, Kay Weckemann, and Daniel Herrscher. Performance study of an in-car switched ethernet network without prioritization. In *Communication Technologies for Vehicles: Third International Workshop, Nets4Cars/Nets4Trains 2011, Oberpfaffen-*

- hofen, Germany, March 23-24, 2011. *Proceedings 3*, pages 165–175. Springer, 2011.
- [LXW20] Yuqian Lu, Xun Xu, and Lihui Wang. Smart manufacturing process and system automation—a critical review of the standards and envisioned scenarios. *Journal of Manufacturing Systems*, 56:312–325, 2020.
- [MAB⁺08] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, mar 2008.
- [Mat12] Starting Matlab. Matlab. *The MathWorks, Natick, MA*, 2012.
- [Mes18] John L Messenger. Time-sensitive networking: An introduction. *IEEE Communications Standards Magazine*, 2(2):29–33, 2018.
- [MHG20] Lisa Maile, Kai-Steffen Hielscher, and Reinhard German. Network calculus results for tsn: An introduction. In *2020 Information Communication Technologies Conference (ICTC)*, pages 131–140. IEEE, 2020.
- [MS17] Dorin Maxim and Ye-Qiong Song. Delay analysis of avb traffic in time-sensitive networks (tsn). In *Proceedings of the 25th International Conference on Real-Time Networks and Systems*, pages 18–27, 2017.
- [MSKS13] Philipp Meyer, Till Steinbach, Franz Korf, and Thomas 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. IEEE, 2013.
- [MSLB19a] Ehsan Mohammadpour, Eleni Stai, and Jean-Yves Le Boudec. Improved credit bounds for the credit-based shaper in time-sensitive networking. *IEEE Networking Letters*, 1(3):136–139, 2019.
- [MSLB19b] Ehsan Mohammadpour, Eleni Stai, and Jean-Yves Le Boudec. Improved delay bound for a service curve element with known transmission rate. *IEEE Networking Letters*, 1(4):156–159, 2019.

- [MSMLB18] Ehsan Mohammadpour, Eleni Stai, Maaz Mohiuddin, and Jean-Yves Le Boudec. Latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping. In *2018 30th International Teletraffic Congress (ITC 30)*, volume 2, pages 1–6. IEEE, 2018.
- [MTM22] Erfan Mozaffariahrar, Fabrice Theoleyre, and Michael Menth. A survey of wi-fi 6: Technologies, advances, and challenges. *Future Internet*, 14(10), 2022.
- [NDR16] Naresh Ganesh Nayak, Frank Dürr, and Kurt Rothermel. Time-sensitive software-defined network (tssdn) for real-time applications. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, pages 193–202, 2016.
- [NORDS⁺20] Jorge Navarro-Ortiz, Pablo Romero-Diaz, Sandra Sendra, Pablo Ameigeiras, Juan J Ramos-Munoz, and Juan M Lopez-Soler. A survey on 5g usage scenarios and traffic models. *IEEE Communications Surveys & Tutorials*, 22(2):905–929, 2020.
- [NTA⁺18] Ahmed Nasrallah, Akhilesh S Thyagaturu, Ziyad Alharbi, Cuixiang Wang, Xing Shao, Martin Reisslein, and Hesham ElBakoury. Ultra-low latency (ull) networks: The ieee tsn and ietf detnet standards and related 5g ull research. *IEEE Communications Surveys & Tutorials*, 21(1):88–145, 2018.
- [Pre16] IEEE TSN Amendment 26: Frame Preemption. 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)*, pages 1–52, 2016.
- [Sau10] Thilo Sauter. The three generations of field-level networks—evolution and compatibility issues. *IEEE Transactions on Industrial Electronics*, 57(11):3585–3595, 2010.
- [SHM⁺21] Youhwan Seol, Doyeon Hyeon, Junhong Min, Moonbeom Kim, and Jeongyeup Paek. Timely survey of time-sensitive networking: Past and future directions. *Ieee Access*, 9:142506–142527, 2021.
- [SIV21] 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*, 18(10):7361–7372, 2021.
- [SKS11] Till Steinbach, Franz Korf, and Thomas C Schmidt. Real-time ethernet for automotive applications: A solution for future in-car networks. In *2011 IEEE International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, pages 216–220. IEEE, 2011.
- [SPS⁺23] Gourav Prateek Sharma, Dhruvin Patel, Joachim Sachs, Marilet De Andrade, Janos Farkas, Janos Harmatos, Balazs Varga, Hans-Peter Bernhard, Raheeb Muzaffar, Mahin K Atiq, et al. Towards deterministic communications in 6g networks: State of the art, open challenges and the way forward. *IEEE Access*, 2023.
- [Sta18] Kevin B. Stanton. Distributing deterministic, accurate time for tightly coordinated network and software applications: Ieee 802.1as, the tsn profile of ptp. *IEEE Communications Standards Magazine*, 2(2):34–40, 2018.
- [STB19] 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.
- [Ste10] Wilfried Steiner. An evaluation of smt-based schedule synthesis for time-triggered multi-hop networks. In *2010 31st IEEE Real-Time Systems Symposium*, pages 375–384. IEEE, 2010.
- [syn20] 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.
- [SZP22a] Mohamed Seliem, Ahmed Zahran, and Dirk Pesch. Quality checks after production: Tsn-based industrial network performance evaluation. In *2022 4th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, pages 1–7, 2022.
- [SZP22b] Mohamed Seliem, Ahmed Zahran, and Dirk Pesch. Tsn-based industrial network performance analysis. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pages 1–7, 2022.

- [SZP23] Mohamed Seliem, Ahmed Zahran, and Dirk Pesch. Delay analysis of tsn based industrial networks with preemptive traffic using network calculus. In *2023 IFIP Networking Conference (IFIP Networking)*, pages 1–9, 2023.
- [Tea12] Mininet Team. Mininet an instant virtual network on your laptop (or other pc), 2012.
- [Var01] András Varga. Discrete event simulation system. In *Proc. of the European Simulation Multiconference (ESM’2001)*, pages 1–7, 2001.
- [VBK16] Amaury Van Bemten and Wolfgang Kellerer. Network calculus: A comprehensive guide. 2016.
- [VZS19] Stefano Vitturi, Claudio Zunino, and Thilo Sauter. Industrial communication systems and their future challenges: Next-generation ethernet, iiot, and 5g. *Proceedings of the IEEE*, 107(6):944–961, 2019.
- [WSJ17] Martin Wollschlaeger, Thilo Sauter, and Juergen 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.
- [WT06] Ernesto Wandeler and Lothar Thiele. Real-time calculus (rtc) toolbox, 2006.
- [ZCWW19] Jiayi Zhang, Lihao Chen, Tongtong Wang, and Xinyuan Wang. Analysis of tsn for industrial automation based on network calculus. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 240–247. IEEE, 2019.
- [ZHLL18] Lin Zhao, Feng He, Ershuai Li, and Jun Lu. Comparison of time sensitive networking (tsn) and ttethernet. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, pages 1–7. IEEE, 2018.
- [ZPC18] Luxi Zhao, Paul Pop, and Silviu S Craciunas. Worst-case latency analysis for ieee 802.1 qbv time sensitive networks using network calculus. *Ieee Access*, 6:41803–41815, 2018.

- [ZPZ⁺20] Luxi Zhao, Paul Pop, Zhong Zheng, Hugo Daigmore, and Marc Boyer. Latency analysis of multiple classes of avb traffic in tsn with standard credit behavior using network calculus. *IEEE Transactions on Industrial Electronics*, 68(10):10291–10302, 2020.
- [ZPZL18] Luxi Zhao, Paul Pop, Zhong Zheng, and Qiao Li. Timing analysis of avb traffic in tsn networks using network calculus. In *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 25–36. IEEE, 2018.