MASTER'S THESIS

TELECOMMUNICATION ENGINEERING

Develop an algorithm to the flow allocation in asynchronous TSN network for the Industrial 4.0.

Multiple ATS instances

Author
Julia Caleya Sánchez

Supervisors
Pablo Ameigeiras Gutiérrez
Jonathan Prados Garzón

Higher Technical School of Information Technology and Telecommunications Engineering

—

Granada, 7th of September, 2023

Develop an algorithm to the flow allocation in asynchronous TSN network for the Industrial 4.0.

Multiple ATS instances.

Author
Julia Caleya Sánchez

Supervisors
Pablo Ameigeiras Gutiérrez
Jonathan Prados Garzón

Department of Signal Theory, Telematics and Communications
-
Granada, September of 2023

Develop an algorithm to the flow allocation in asynchronous TSN network for the Industrial 4.0.

Julia Caleya Sánchez

Keywords: TSN, prioritization, flows, delay, deterministic traffic, Asyncrhonous Traffic Shaper and Industria 4.0

Abstract

This document presents the research work carried out with the aim of designing, developing, and evaluating a Time Sensitive Networking (TSN) network scheduling solution based on Asynchronous Traffic Shaper (ATS), ensuring deterministic requirements for industrial networks. Industry 4.0 demands services with stringent quality of service, and only TSN enables their connectivity.

However, industrial IoT demands device mobility, which is impossible with TSN. Therefore, 5G networks are ideal as they offer low cost, robustness, and the interoperability of devices through ultra-reliable and low-latency wireless communications. Hence, the ideal scenario would be to integrate both technologies, a topic addressed by numerous research efforts, although only 3GPP has defined a proposal for integration. This integration model involves the use of synchronous TSN but presents challenges such as the need for a common time reference among network nodes and lower scalability compared to asynchronous TSN.

Therefore, this project focuses on the study of asynchronous TSN employing ATS. ATS is responsible for implementing flow routing in asynchronous TSN switches and consists of several queued stages for routing. This scheduling does not minimize the number of priorities used in each ATS, thus reducing the cost of the asynchronous network, as asynchronous networks directly depend on the priority levels available in ATS. Additionally, with a lower number of priorities in each ATS, configuring and operating an asynchronous TSN network becomes more straightforward.

Consequently, an algorithm has been defined in this project to minimize the number of priorities used by the ATSs in a TSN network while meeting the required delay for industrial services. This project formally formulates the problem of flow priority assignment in a asynchronous TSN network and demonstrates the optimization of our proposed algorithm. Furthermore, the proposed solution is generic, scalable, and has reduced complexity.

On the other hand, a simulator has been developed to implement this solution and verify the correct prioritization and delay distribution defined for an asynchronous TSN network. Three network topologies have also been implemented, and a study of the main characteristics of the existing service

types in the network has been conducted for a more realistic experimentation.

In the experimental tests, several simulations have been carried out in the test environment, testing the routing capacity and performance of ATSs against service types with critical delay requirements. Our solution has been compared with brute-force search to verify its optimality and correctness, resulting that the exection time of brute force is significantly higher than ours with exactly the same prioritization results. It has been observed that Flow Prioritization has higher utilization than PCP Prioritization, and the network topology does not affect the scalability of our algorithm. Furthermore, it has been deduced that utilization varies depending on the number of flows for services with strict delay requirements. Finally, it has been determined that the developed algorithm scales correctly with an increasing number of flows without excessive growth in execution time.

Desarrollo de un algoritmo para la asignación de flujos en redes TSN asíncronas para la Industria 4.0

Julia Caleya Sánchez

Palabras clave: TSN, priorización, flujos, requisito de retardo, tráfico determinista, Asyncrhonous Traffic Shaper y Industria 4.0

Resumen

En este documento se presenta el trabajo de investigación desarrollado que tiene como objetivo el diseño, desarrollo y evaluación de una solución de planificación de redes Time Sensitive Networking (TSN) basadas en Asyncrhonous Traffic Shaper (ATS), asegurando los requisitos deterministas de las redes industriales. La Industria 4.0 presenta servicios con calidades de servicios exigentes y solo TSN permite su conectividad. Sin embargo, el IoT industrial demanda la movilidad de los dispositivos siendo imposible con TSN. Por lo que las redes 5G son idóneas, ya que presentan un bajo coste, robustez y la interoperación de dispositivos mediante comunicaciones inalámbricas ultrafiables y de baja latencia. Por ende, lo ideal sería integrar ambas tecnologías, tema tratado por numerosas investigaciones aunque solo el 3GPP ha definido una propuesta de integración. Este modelo de integración contempla el uso de TSN síncrono mas presenta inconvenientes como la necesidad de referencia temporal común entre los nodos de la red y una menor escalabilidad que TSN asíncrono.

Por tanto, este proyecto se centra en el estudio de TSN asíncrono que emplean el ATS. El ATS es responsable de implementar el enrutamiento de flujo en los conmutadores TSN asíncronos, y cuenta con varias etapas encoladas para su enrutamiento. Esta planificación no minimiza el número de prioridades empleadas en cada ATS y por tanto, reducir el coste de la red asíncrona, ya que las redes asíncronas depende directamente de los niveles de prioridad disponibles en el ATS. Además, con un menor número de prioridades en cada ATS es más fácil configurar y operar una red TSN asíncrona.

En consecuencia, se ha definido un algoritmo que minimiza el número de prioridades que utiliza los ATSs de una red TSN mientras se cumple con el requisito de retardo demandado por los servicios industriales. En este proyecto se lleva a cabo una formulación formal del problema de asignación de prioridades de flujo en una red TSN asíncrona y se demuestra la optimización del algoritmo propuesto. Además, la solución propuesta es genérica, escalable y con complejidad reducida.

Por otro lado, se ha desarrollado un simulador que implementa esta solución para verificar el correcto funcionamiento de prioritización y dis-

tribución de retardos definidos para una red TSN asíncrona. También se han implementado tres topologías de red y realizado un estudio de las principales características que presentan los tipos de servicios existentes en la red para tener un experimentación más realista.

En las pruebas experimentales, se han realizado varias simulaciones en el entorno de pruebas, probando la capacidad y el rendimiento de enrutamiento de los ATSs frente a los tipos de servicios con requisitos de retardo críticos. Se ha comparado nuestra solución con la búsqueda de fuerza bruta para comprobar su optimalidad y correctitud, obteniendo que el tiempo de ejecución de la fuerza bruta es muy superior al nuestro con exactamente los mismos resultados de priorización. Se ha comprobado que la prioritización por flujos tiene una mayor utilización que por PCP y la topología de red no afecta a la escalabilidad del algoritmo. Por otro lado, se deduce que la utilización varía en función del número de flujos de los servicios con requisito de retardo estricto. Finalmente, se determina que el algoritmo desarrollado escala correctamente con el incremento del número de flujos sin un crecimiento excesivo del tiempo de ejecución.

D. **Pablo Ameigeiras Gutiérrez**, Profesor del Área de Telemática del Departamento Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

D. **Jonathan Prados Garzón**, Profesor del Área de Telemática del Departamento Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Develop an algorithm to the flow allocation in asynchronous TSN network for the Industrial 4.0.***, ha sido realizado bajo su supervisión por **Julia Caleya Sánchez**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 7 de septiembre de 2023.

Los directores:

Pablo Ameigeiras Gutiérrez **Jonathan Prados Garzón**

Acknowledgments

I would like to thank all those who have contributed to its completion, without them it would have been more difficult.

First of all, to my family, for being my unconditional support at all times from the first moment I decided to start my university studies, where distance has not been an inconvenience, but has allowed me to strengthen existing relationships. Especially, to my parents, Francisco and Rosario, for teaching me that everything can be achieved with hard work, effort and perseverance. Without all their love, guidance, encouragement and motivation, especially during these two years, this research would not have been possible.

To my sister, Alejandra, who has always been my example to follow, and to my brother, Paco, for teaching me to question myself and not to conform to anything in this life. Thanks to you, you have made me the person I am today.

To my supervisors, Pablo and Jonathan, for your total availability every time I needed it, guidance and constant support throughout this project. Your experience and knowledge were essential for the completion of this work. Especially, for showing me the world of research and wanting to continue in it. Also for your understanding and patience, and for mentoring me not only academically, but also personally and professionally.

I am also very grateful to the WiMuNet Lab group for their guidance, trust and support, and for allowing me to be a member of the research group.

Of course, I would like to thank my classmates, my friends from Fuente del Maestre, and above all, my friends from Granada for the fun, trust, and understanding you give me.

Thanks to all of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the coming of Fifth Generation (5G) wireless networks, progress has been made in the development of new services and new functionalities that were previously not considered due to the demands involved in their use. This evolution in wireless networks is driven by a number of key factors and reason such as [15, 16]:

- **Increasing Data Demand**: With the proliferation of smartphones, connected devices, and data-intensive applications, there has been a tremendous growth in data consumption. Users now expect fast and reliable connectivity to stream high-definition videos, engage in real-time communication, and access cloud services. The evolution to 5G is fueled by the need to accommodate this increasing data demand and provide enhanced network capacity.

- **Enhanced Mobile Broadband**: 5G aims to deliver significantly faster download and upload speeds compared to previous generations of wireless technology. By leveraging higher frequency bands, wider channel bandwidths, and advanced modulation techniques, 5G offers the potential for multi-gigabit-per-second data rates. This evolution addresses the demand for seamless streaming of high-resolution content and supports emerging applications like Virtual Reality (VR), Augmented Reality (AR), and immersive gaming.

- **Lower Latency**: Latency refers to the time it takes for data to travel between devices and the network. 5G technology targets ultra-low latency, reducing the delay to milliseconds. This is critical for real-time applications such as autonomous vehicles, remote surgery, and industrial automation, where instant responsiveness is crucial. The evolution to 5G enables the development of time-sensitive applications that rely on minimal delay.

- **Internet of Things (IoT) Connectivity**: The growth of IoT devices, which includes sensors, wearables, and connected machines, requires a robust and scalable wireless infrastructure. 5G is designed to provide massive Machine Type Communication (mMTC) capabilities, enabling the simultaneous connection of a large number of devices. This evolution supports the vision of a highly interconnected world with seamless IoT connectivity and enables applications in smart cities, smart homes, and industrial IoT.

- **Network Capacity and Efficiency**: 5G utilizes advanced technologies such as massive multiple-input multiple-output (MIMO), beamforming, and network slicing to optimize spectral efficiency and network capacity. These techniques enable better utilization of available frequency bands, improved spectrum efficiency, and increase of the entire network capacity. The evolution to 5G addresses the challenge of providing reliable connectivity in dense urban areas and crowded environments where multiple devices connect simultaneously.

- **Mission-Critical Communications**: Industries such as public safety, utilities, and transportation require highly reliable and resilient communication networks for mission-critical operations. 5G incorporates features like Ultra-Reliable and Low Latency Communication (URLLC) to ensure high reliability and availability. The evolution to 5G offers robust and secure wireless connectivity for applications that demand critical communication capabilities.

- **Technological Advancements**: The evolution of wireless technology leading to 5G is facilitated by advancements in hardware, signal processing, and network architecture. These advancements include higher-performance processors, improved antenna designs, more efficient modulation schemes, and Software-Defined Networking (SDN) approaches. These technological advancements contribute to the increased speeds, lower latency, and improved overall performance of 5G networks.

These reasons drive the evolution of wireless to 5G technology, with the goal of meeting the growing demand for faster speeds, lower latency, massive connectivity and enhanced network capacity. It enables a wide range of applications and use cases, empowering industries and consumers with transformative capabilities. 5G is envisioned to revolutionize the world of telecommunications, not only in Industry 4.0, but also in the mobile, medical, construction, logistics and even agriculture and livestock sectors [17]. In other words, this revolution has spread to all sectors of society, greatly facilitating the performance of tasks and the change of the workforce, creating new jobs. Likewise, the rise of automation in industries such as automotive, energy and transportation is driving the growth of the time-sensitive networking market. According to [18], the global time-sensitive networking market is estimated to be worth $200 million by 2023 and $1.7 billion by 2028, growing at a Compound Annual Growth Rate (CAGR) of 58.3% from 2023 to 2028.

However, the strict delay requirements and high reliability demanded by various industries, such as URLLC where delays of less than 1 ms and a packet reliability of at least 99.999% [19], have currently only been achieved with a layer 2 technology of the Open Systems Interconnection (OSI) model determined by the Institute of Electrical and Electronics Engineers (IEEE), called Time-Sensitive Networking (TSN). TSN is a set of standards developed to provide deterministic and time-critical communications over Ethernet networks. TSN ensures that critical data is transmitted with maximum accuracy and determinism by incorporating the precise timing and scheduling mechanisms used in Ethernet. It also allows time-critical and synchronized applications to coexist with non-time-sensitive traffic on a single network infrastructure, simplifying network management and reducing costs.

1.1 Context and motivation

Initially, TSN was designed for industrial automation and control systems to meet the Quality of Service (QoS) requirements demanded in terms of latency, jitter, reliability and packet loss. But today it plays a key role in industrial transformation and in various sectors such as:

- **Industrial Automation**: the real-time communication is essential for precise control and coordination of machinery and systems. TSN enables synchronized and deterministic communication between devices, facilitating tasks such as machine control, robotics, process automation, and Industrial Internet of Things (IIoT). The collaborative robots Cobots [20] work alongside human operators, requiring precise coordination and synchronization to ensure safety and efficiency that TSN can enabling seamless collaboration and enhancing the capabilities or human-robots team.

- **Automotive**: TSN is useful in Advanced Driver Assistance Systems (ADAS), autonomous driving, vehicle-to-vehicle (V2V) communication, and in-vehicle networks in the automobile industry. TSN can enable dependable, low-latency connectivity for crucial automobile functions, hence improving safety and performance.

- **Energy and Power Systems**: TSN can be applied to power generation, transmission, and distribution systems. It enables precise synchronization, control, and monitoring of devices and systems within the electrical grid. TSN can improve the efficiency, reliability, and responsiveness of smart grid components, such as substations, renewable energy sources, and energy management systems.

- **Aerospace and Defense**: TSN can be utilized in aerospace and defense applications that require real-time communication, coordination, and synchronization. It can support mission-critical systems, flight control, avionics, command and control systems, and communications between military vehicles or aircraft.

- **Audio/Video Streaming**: TSN can ensure low-latency and synchronized transmission of audio and video data. It is relevant for applications such as professional audio/video production, live broadcasting, multimedia streaming, and real-time video surveillance.

- **Healthcare**: TSN has potential applications in healthcare, particularly for networked medical devices and telemedicine. It can facilitate real-time monitoring, remote surgeries, and coordination of medical equipment in hospital settings. Also it is need real-time communication between various medical devices, such as patient monitoring system, infusion pump, and surgical robots.

- **Transportation and Logistics**: TSN can enhance communication and coordination in transportation and logistics systems. It can be used for real-time tracking, monitoring, and control of vehicles, logistics operations, and supply chain management.

- **Smart Cities**: TSN can contribute to the development of smart cities by enabling efficient and reliable communication in various domains such as traffic management, public transportation, infrastructure monitoring, and emergency response systems.

Specifically, TSN is a set of layer 2 standards under development that are specified as a series of amendments to the IEEE 802.1Q standard develop by the Time-Sensitive Networking task group of the IEEE 802.1 working group [21]. TSN ensures predictable traffic transmission by employing sophisticated and complicated schedulers for frame transmission on TSN bridge output ports. TSN standards define two types of schedulers: Asynchronous Traffic Shaper (ATS), is define in IEEE 802.1Qbv, and Time-Aware Shaper (TAS), as defined in IEEE 802.1Qcr. Currently, much of the TSN literature has focused on the synchronous version of TSN (802.1Qbv) essentially because it can provide deterministic transmissions. However, 802.1Qbv has notable issues or drawbacks. On the one hand, it requires synchronization, the performance of which depends on the accuracy of this synchronization. This causes it to scale inefficiently, as synchronization gradually degrades with the number of hops (devices traversed).On the other hand, it is not at all well adapted to scenarios involving transmissions with jitter as in the case of 5G or in Virtual Network Function (VNF). This makes 802.1Qbv flavor very difficult to integrate with 5G. Because of these problems, in this works we concentrate on asynchronous TSN networks, where a shared and exact time reference between the different ATS instances is not required to orchestrate the transmission of the output data.

There are several approaches to the configuration of ATS-based TSN networks [22, 23, 14] and various works related to the solutions proposed for the flow prioritization in asynchronous TSN networks [24, 13] specifically address the priority and flows assignment.

All of the proposed solutions have scalability problems, and it is vital to identify solutions that can handle and react to increased traffic smoothly and without sacrificing the QoS provided. As a result, scalability is a critical aspect in being able to correctly adapt to the growing evolution of new services with more rigorous QoS requirements. Furthermore, due to the nature of the proposed solutions, all of them are computationally complex to implement.

The main motivation for this work is that there is currently no asynchronous TSN prioritization and configuration setup that satisfies the QoS requirements required in Industry 4.0, for example, without having a combinatorial complexity and being easily scalable.

The personal motivation for the development of this work is due to the opportunity offered by the Wireless and Multimedia Networking Lab (TIC-235) (WIMUNET) to develop my professional practice around the professional practice in the field of time-sensitive networking research. From my point of view, this work is motivated by the fact that most TSN network configurations and designs employ the TAS scheduler instead of opting for the ATS scheduler that entails less configuration complexity. In addition, TSN networks based on ATS are more scalable and have a higher statistical multiplexing.

1.2 Objectives

The main objective of this Master's Thesis is the design, development and evaluation of a scheduling solution for TSN networks based on ATS instances, ensuring deterministic QoS requirements (delay, jitter and packet loss) adapted to the needs of industrial networks. The targeted solution must be generic, scalable and with reduced computational complexity. To this end, this objective can be decomposed into several sub-objectives:

1. **Study the operation of TSN and 5G technologies**: Initially, it is necessary to review the limitations of Industry 4.0 in order to determine the problems to be solved. The operation and mechanisms used by TSNs to satisfy quality of service requirements are studied, focusing on asynchronous networks. The planning mechanisms used by TSNs to carry out the prioritization of the flows to be transmitted are also analyzed. As well as the 5G architecture and proposals to integrate 5G and TSN. In other words, the tasks of this sub-objective are:

   1.1 Review of the constraints of Industry 4.0.

   1.2 Study of the 5G architecture.

   1.3 Study of TSN functionality.

   1.4 Study of the TSN scheduler.

   1.5 5G and TSN integration study.

   For its development, a search for information is carried out through the syllabus studied throughout the course, bibliographic documents and reference pages on the Internet as scientific publications related to this type of technology.

2. **Design of a flow priority allocation solution in a single ATS network**: After learning how TSN works, several task are carried out to develop this sub-objective:

   2.1 Review of existing works to date on scheduling in asynchronous networks.

   2.2 Design of a flow priority allocation solution for a network with a single ATS instance, reducing the number of priorities required in the ATS instance.

   2.3 Formal problem definition and proposed solution for the flows prioritization of single instance ATS.

   2.4 Development of an algorithm that implements the designed solution, allowing to verify that the delay requirements of each flow are met.

3. **Extension of the flow priority allocation algorithm in an asynchronous TSN network based on several ATSs**: After the design of the algorithm, it is necessary to know what is the distribution of the requirements of each industrial traffic to be transmitted in the network. That is, determine which flow requirements, mainly delay, in each of the ATS instances that make up the route to follow from an origin to a destination of a given traffic.

3.1 Review of delay distribution mechanisms in a TSN network.

3.2 Selection of the end-to-end (E2E) delay distribution mechanism between ATS instances.

3.3 Formal problem definition of the proposed delay distribution mechanism.

3.4 Development of the proposed delay distribution mechanism.

4. **Evaluation and validation of the proposed solution**: Several experiments are carried out to verify the correct operation of the proposed solution, verifying that the delay requirement is met in all flows. Simulation is considered as an evaluation method because there is no commercial device that implements asynchronous TSN.

4.1 Study of the features of industrial services.

4.2 Study of the topology used in the industrial environment.

4.3 Design of the experiments to be developed.

4.4 Analysis of the results obtained.

1.3 Methodology and Planning

The section 1.2 has detailed each of the secondary objectives established in order to achieve the main objective of developing a flow prioritization solution in a ATS-based TSN network, meeting the QoS requirements and reducing the level of priorities needed. A linear methodology is used, consisting of rigorously defining the problem, proposing solutions, developing an analytical model, validating this model or proposal and obtaining conclusions. This process allows to check the progress of the project in a clear and concise way, and to verify if the proposed objectives have been achieved.

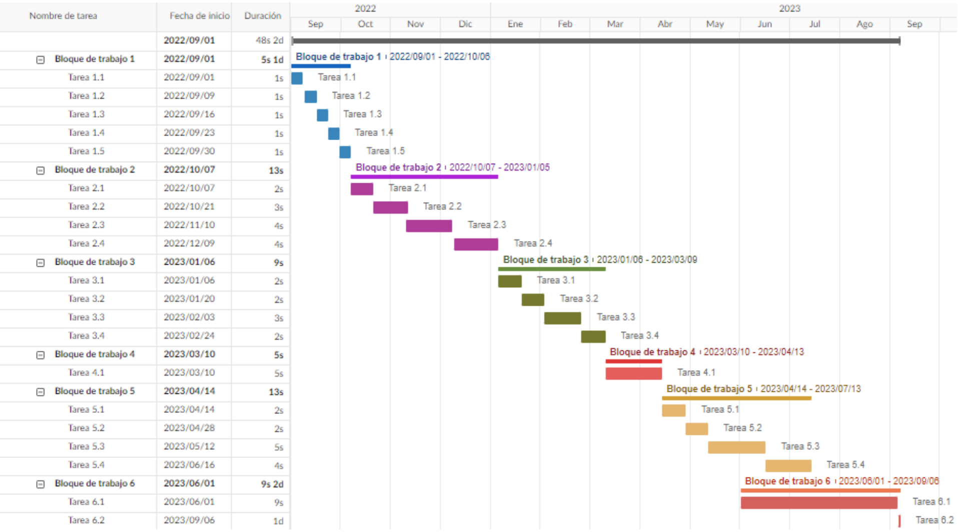
This methodology is subdivided into several works block according to each sub-objective detailed in the section 1.2 and new sub-objectives that are necessary to develop this project:

- **Study the operation of TSN and 5G technologies**
  - Task 1.1: Achievement of sub-objective 1.1 (1 weeks).
  - Task 1.2: Achievement of sub-objective 1.2 (1 weeks).
  - Task 1.3: Achievement of sub-objective 1.3 (1 weeks).
  - Task 1.4: Achievement of sub-objective 1.4 (1 weeks).
  - Task 1.5: Achievement of sub-objective 1.5 (1 weeks).

- **Design of a flow priority allocation solution in a single ATS network**.
  - Task 2.1: Achievement of sub-objective 2.1 (2 weeks).
  - Task 2.2: Achievement of sub-objective 2.2 (3 weeks).
  - Task 2.3: Achievement of sub-objective 2.3 (4 weeks).
  - Task 2.4: Achievement of sub-objective 2.4 (4 weeks).
- **Extension of the flow priority allocation algorithm in an asynchronous TSN network based on several ATSs**.
  - Task 3.1: Achievement of sub-objective 3.1 (2 weeks).
  - Task 3.2: Achievement of sub-objective 3.2 (2 weeks).
  - Task 3.3: Achievement of sub-objective 3.3 (3 weeks).
  - Task 3.4: Achievement of sub-objective 3.4 (2 weeks).
- **Implementation of an asynchronous TSN network**: It consists in the creation of an asynchronous TSN network with several ATS instances where the flow priority algorithm is implemented in each of them and the delay distribution method developed previously is applied.
  - Task 4.1: Implementation of an asynchronous TSN network with the flow priority allocation algorithm implemented in each of the ATS instances (5 weeks).
- **Validation and evaluation of the proposed solution**.
  - Task 5.1: Achievement of sub-objective 4.1 (2 weeks).
  - Task 5.2: Achievement of sub-objective 4.2 (2 weeks).
  - Task 5.3: Achievement of sub-objective 4.3 (5 weeks).
  - Task 5.4: Achievement of sub-objective 4.4 (4 weeks).
- **Memory writing**: all the information collected and the results obtained in each of the simulations for the development of this Master's thesis are written in detail.
  - Task 6.1: Report writing (9 weeks).
  - Task 6.2: Submit the documentation for the Master's thesis (1 day).

The total duration of the project has been 54 weeks. This equates to 378 days (including laborables and nonlaborables). The steps to follow throughout the project were clearly defined at the start of the project, and these may be seen previously.

The time planning is shown graphically in the Gantt chart in Figure 1.1. The figure clearly shows the work blocks and tasks, as well as the expected duration for each of them. It should be noted that the initial planning is strict and has been planned consciously so that the work will take enough time to be evaluated with the academic load of 30 credits.



Nombre de tarea	Fecha de inicio	Duración
	2022/09/01	48s 2d
⊟ **Bloque de trabajo 1**	**2022/09/01**	**5s 1d**
Tarea 1.1	2022/09/01	1s
Tarea 1.2	2022/09/09	1s
Tarea 1.3	2022/09/16	1s
Tarea 1.4	2022/09/23	1s
Tarea 1.5	2022/09/30	1s
⊟ **Bloque de trabajo 2**	**2022/10/07**	**13s**
Tarea 2.1	2022/10/07	2s
Tarea 2.2	2022/10/21	3s
Tarea 2.3	2022/11/10	4s
Tarea 2.4	2022/12/09	4s
⊟ **Bloque de trabajo 3**	**2023/01/06**	**9s**
Tarea 3.1	2023/01/06	2s
Tarea 3.2	2023/01/20	2s
Tarea 3.3	2023/02/03	3s
Tarea 3.4	2023/02/24	2s
⊟ **Bloque de trabajo 4**	**2023/03/10**	**5s**
Tarea 4.1	2023/03/10	5s
⊟ **Bloque de trabajo 5**	**2023/04/14**	**13s**
Tarea 5.1	2023/04/14	2s
Tarea 5.2	2023/04/28	2s
Tarea 5.3	2023/05/12	5s
Tarea 5.4	2023/06/16	4s
⊟ **Bloque de trabajo 6**	**2023/06/01**	**9s 2d**
Tarea 6.1	2023/06/01	9s
Tarea 6.2	2023/09/06	1d

2022
2023
Sep
Oct
Nov
Dic
Ene
Feb
Mar
Abr
May
Jun
Jul
Ago
Sep
Bloque de trabajo 1 ı 2022/09/01 - 2022/10/06
Tarea 1.1
Tarea 1.2
Tarea 1.3
Tarea 1.4
Tarea 1.5
Bloque de trabajo 2 ı 2022/10/07 - 2023/01/05
Tarea 2.1
Tarea 2.2
Tarea 2.3
Tarea 2.4
Bloque de trabajo 3 ı 2023/01/06 - 2023/03/09
Tarea 3.1
Tarea 3.2
Tarea 3.3
Tarea 3.4
Bloque de trabajo 4 ı 2023/03/10 - 2023/04/13
Tarea 4.1
Bloque de trabajo 5 ı 2023/04/14 - 2023/07/13
Tarea 5.1
Tarea 5.2
Tarea 5.3
Tarea 5.4
Bloque de trabajo 6 ı 2023/06/01 - 2023/09/06
Tarea 6.1
Tarea 6.2

Figure 1.1: Gantt diagram with the task planning

1.4 Cost estimation

This section details all the direct and indirect resources that have been used during the development of the project and finally, the estimated budget with the project costs.

The resources used have been classified into hardware, software and human resources. Each of them is detailed below.

1.4.1 Hardware resources

The hardware resources represent the technological components that have been implemented in the project. These elements are:

- **Personal Computer** [25]: it is the main tool, both technologically and administratively. An Asus Zenbook 14 has been used in the project with AMD Ryzen 7 4700U Central Processing Unit (CPU) at 2.00GHz with 8 cores and 16 GB of RAM. It has been used mainly for the research, design, implementation and documentation stages.

- **Server**: it is a tower computer with Intel(R) Core(TM) i7-6700K CPU at 4.00GHz with 4 cores and 32 GB of RAM. The capabilities of this computer exceed the capabilities of the personal computer, since it is used to carry out the simulation tests of this project.

1.4.2 Software resources

The software resources represent the programs and applications that have been used in the development of this project. The Table 1.1 shows the software that has been used:

Software program/ application	**Use Description**
SO Windows 10 Home 64 bit [26]	SO installed in the server
Matlab R2021b [27]	Implementation the new flow priority allocation algorithm, the delay distribution algorithm and the asynchronous TSN networks simulator
GanttPro [28]	Design of the task planning
Overleaf online LaTeX editor [29]	Document drafting

Table 1.1: Software resources description

1.4.3 Human resources

The human resources of this project take into account the people who have been involved in this project and the quantity of time that each one of them will dedicate to work on it. The Table 1.2 shows the time spent on this project by the student and her supervisor:

- **Julia Caleya Sánchez**: Student of M.Sc. Telecommunication Engineering of the University of Granada.
- **Pablo José Ameigeiras Gutiérrez**: Associate Professor of the Department of Signal Theory, Telematics and Communications of University of Granada.
- **Jonathan Prados Garzón**: Assistant Professor Doctor of the Department of Signal Theory, Telematics and Communications of University of Granada.

The hours spent by the student have calculated taking into account the sum of all the hours spent working on each of the tasks involved in the

Person	Hours worked
Julia Caleya Sánchez (student)	1620
Pablo Ameigeiras Gutiérrez (supervisor)	50
Jonathan Prados Garzón (supervisor)	50

Table 1.2: Human resources description

realization of the project. The student will invest approximately 6h/day. The hours invested by the supervisor include all the meetings to clarify all the doubts that may arise during the course of the project and for some orientation.

1.4.4 Project Budget

Finally, this section details the estimated budget for the project considering all the resources used as described above. Table 1.3 shows the estimated costs of each of these resources, as well as the total cost of the project.

Resource	Units	Unit cost	Subtotal cost
Laptop	1	800 €	800 €
Server	1	2000 €	2000 €
SO Windows 10 Home	1	145 €	145 €
MATLAB R2021b (anual license)	1	860 €	860 €
Pablo Ameigeiras Gutierréz (Supervisor Labor)	50 hours	50 €/hour	2500 €
Jonathan Prados Garzón (Supervisor Labor)	50 hours	50 €/hour	2500 €
Julia Caleya Sánchez (Student labor)	1620 hours	20 €/hour	32400 €
Total Cost			41.205 €

Table 1.3: Project budget

Please note that all software that is not open source has been included in the cost because a license is required for its use. The student's labour cost has been estimated at 20 € per hour as she is considered a junior engineer, while the director's labour cost has been estimated at 50 € per hour.

Consequently, it is concluded that the final budget for the final Master's Thesis is 41.205 €, forty one thousand two hundred five euros.

1.5 Project structure

This section describes the different parts that make up the report in order to provide the reader with an overview of the content of the chapters of this project. The following is a brief description of the contents of each chapter:

- **Chapter 1 - Introduction**: This first chapter attempts to give a general approach to the work. It consists of the following sections:

  - Context and motivation: consists of contextualizing the problem to be addressed in this work and the motivation for its development.
  - Objectives: consists of details the objectives to be met throughout the project.
  - Methodology and Planning: explains the steps to be followed and describes the temporary stages that will be developed throughout the project.
  - Cost Estimation: the hardware, software and human resources required for the project are indicated, as well as the total cost of the project.
  - Project Structure: the content of each of the chapters that make up the master's thesis is described.
  - Contributions and publications: in this section the publication made in a congress related to this work and the patent applied for are indicated.

- **Chapter 2 - State of Art**: This chapter summarizes the review of the state of the art and theoretical knowledge of the technologies used, divided into the following sections:

  - Industry 4.0: the limitations that exist in the current industrial networks are presented.
  - 5G: explains the fundamentals of fifth generation networks, in particular the integrations proposed to date with TSN.
  - TSN: explains the fundamentals of time-sensitive networks, specifically the mechanisms they implement to meet the quality of service requirements demanded.
  - State of the art of prioritization scheduler: bibliographic review of works similar to the one developed, through the compilation of scientific contributions obtained from Scopus [30], Google Scholar [31] and ResearchGate [32].

- **Chapter 3 - Problem Definition and solution design**: this chapter explains the problem to be solved and the formulation of the problem.

  - System Model: it determines the model of the system to be used in the solution of this problem.
  - Flow Allocation Problem: formally explains the problem to be solved and the partition of the problem into two simpler problems to be solved.

  - Delay requirement Distribution: details the formulation and proposed solution for the distribution of the delay requirement in an asynchronous TSN network.
  - Per ATS Flow Prioritization Algorithm: the proposed solution for the flow prioritization problem in a single instance ATS network is detailed and formally explained. The algorithm designed to solve the problem is also explained.

- **Chapter 4 - Experimental Evaluation**: the methodology, experimentation and results obtained are presented.
  - Methodology: The methodology followed for the experimental tests is detailed.
  - Experimental Setup: describes the design of experiments, i.e., the setup and the experimental data set.
  - Results: the results obtained in the experimentation carried out are analyzed and verified.

- **Chapter 5 - Conclusions and Future works**: The conclusions obtained at the end of the project are presented and the possible courses of action for the problems that remain to be addressed are detailed, as well as the future updating and improvement of the proposed solution.
  - Conclusions: the final conclusions of the project are presented.
  - Future works: the lines of improvement to be addressed and the next steps to be taken in relation to this project are described.

1.6 Contributions and publications

Finally, during the course of the project, several scientific contributions have been made, among them:

- A patent application titled *"Método de configuración de redes sensibles al retardo basadas en planificadores con conformación de tráfico asíncrono, y con calidad de servicio determinista"* from the flow prioritization algorithm in 2023. The application can be found in the Annex B.

- A paper titled *"Flow Prioritization for TSN Asynchronous Traffic Shapers"* in the TENSOR workshop of the International Federation for Information Processing (IFIP) Networking conference in June 2023 available in the IEEE Xplore database [33]. The paper is attached in the Annex A.

- The previous paper has been presented at this workshop.
- An IEEE Transactions is currently being prepared based on this Master's Thesis.

In addition, during the realization of this project a research initiation grant and the FPU grant have been obtained.

Chapter 2

State of the Art

The objective of this chapter is to provide an overview of the main constraints that exist in the industrial sector. It also explains the main functionalities of TSN and 5G and the different solutions proposed for the integration between 5G and TSN. Next, the planning functions used by TSN are shown. Finally, an overview of the TSN prioritization scheduler solutions proposed to date is given.

2.1 Industry 4.0

Industry 4.0 currently has a great relevance for the advancement of new technologies, because they are the sector where the greatest amount of capital is invested in order to implement new technological improvements in development to obtain greater benefits. The progress in the industry is clearly demonstrated throughout history, where it has suffered great revolutions to achieve progress and improve both the quality of workers and the opportunity to achieve greater economic benefits, as shown in Figure 2.1.

The first industrial revolution in 1760 introduced mechanical production equipment with the aim of eliminating the old handmade tools that were mainly used in agriculture. This new equipment was more productive because it was powered by water and steam energy. Territorial trade also benefited from the arrival of new means of transportation such as the locomotive or the steamboat. These new machines led to an increase in production, generating great fortunes and profits for the bourgeoisie. This increase in economic benefits caused the mentality of the bourgeoisie to be discouraged by the relentless pursuit of profitability instead of the care of their workers and technological evolution.

The arrival of the second industrial revolution at the end of the 19th century introduced chain processes using new energy sources, electricity and

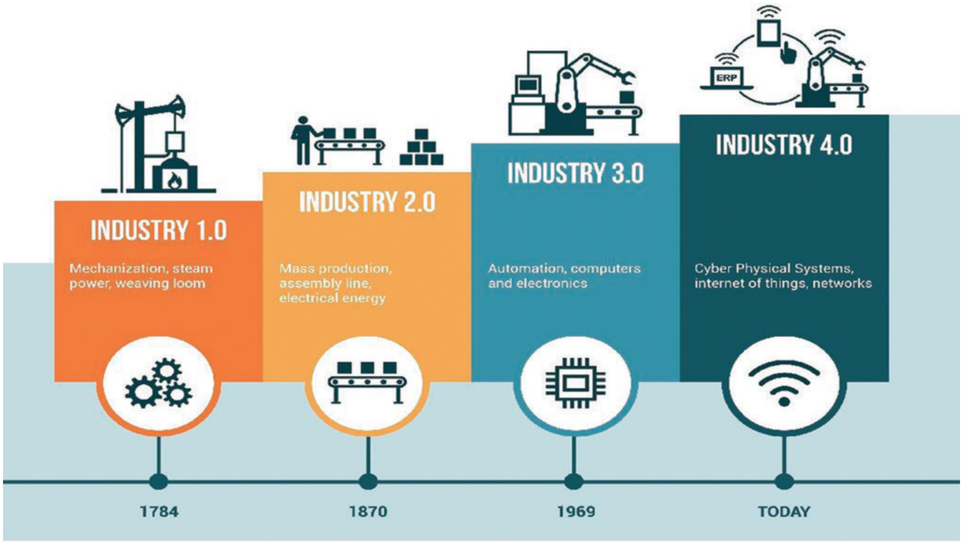

INDUSTRY 1.0
Mechanization, steam power, weaving loom
INDUSTRY 2.0
Mass production, assembly line, electrical energy
INDUSTRY 3.0
Automation, computers and electronics
INDUSTRY 4.0
Cyber Physical Systems, internet of things, networks
ERP
1784
1870
1969
TODAY

Figure 2.1: Progress of the industrial revolution stages [1]

gas, as substitutes for steam. With this change of energy source, mass production of products was achieved thanks to the creation of the conveyor belt. These transformations affected the organization and management of work in factories. It also brought about a change in international economic relations and communications with the appearance of new means of transport such as railways and automobiles.

The third industrial revolution took place at the beginning of the 21st century, transforming industry with the use of electronics and information technology to drive the automation of production and the digitization of information. These new technologies offered the possibility of information processing allowing the creation of new inventions such as computers or specific use devices. With the aim of facilitating data to entrepreneurs and the control of production with the use of robots, which allowed to simplify the manufacturing processes. In order to carry out automation, networks using bus technology were necessary, which are not only expensive but also difficult to interoperate with each other, mainly due to the huge range of different technologies that are in operation in an industry. In addition, each of these technologies provides connectivity to a certain service whose QoS are completely disparate from one another. There was also a transformation in the use of energy, with a greater use of renewable energies such as solar and wind power, as opposed to the use of nuclear power plants.

Finally, the fourth industrial revolution, known as Industry 4.0 [34], will begin in 2016. The trend of this revolution is the deterministic connectivity in the production chains of factories, i.e., the automation of production chains, with the aim of improving efficiency and productivity. For this purpose, use is made of the exchange of information between Cyber-Physical System (CPS) [35], which are connected to a Wireless Sensor Network (WSN). These systems are known as IIoT to enable cloud computing. In this way, it is achieved the creation of systems with high scalability, few failures and with a shorter waiting time between the different nodes to achieve greater interaction.

Therefore, with the advent of Industry 4.0, changes are taking place at the organizational, production and customer management levels to achieve greater efficiency and productivity among factories connected to each other through autonomous systems. These autonomous systems allow the identification of different patterns that humans would not be able to recognize in the short or long term. Moreover, it is not only manufacturing processes that are affected, but can influence the entire industry in general and society itself. Wireless connectivity enables communication with all the agents involved in a production process (suppliers, customers, investors, etc.). This increases the exchange of data reported between the multiple systems and the participants themselves. Also, communications between the components of the factories must be connected wirelessly to adjust and transmit configuration data in real time, avoiding possible failures or production problems. In this way, machinery must be intelligent enough to be able to generate, analyze and diagnose its own processes, without human intervention in the future.

This leads to a change of mindset, with the size of the company being insignificant. These companies will have to upgrade to avoid jeopardizing their business or face strong levels of competition. As could happen with Western factories, which have low production costs and their factories are highly digitized and connected to each other. In short, the aim is to unite the physical world with the digital world on the basis of a series of parameters so that the systems are capable of making decisions on their own.

The Figure 2.2 shows the main transformations in automation from Industry 3.0 to Industry 4.0, which has been derived from the ANSI/ISA-95 standard. It can be seen how the different levels that were originally hierarchical are now fully connected to each other. This evolution involves all devices used in industrial production, i.e. sensors that check the status of production and the process, Programmable Logic Controllers (PLCs) and actuators that control and manipulate the sensors, user monitoring tools, production process planners and even business strategies.

To drive the transformation of Industry 4.0, the Spanish Ministry of Industry, Trade and Tourism has launched the *"Industria Conectada 4.0"* program. This program aims to incorporate knowledge and new technologies for the digitization of the processes carried out in Spanish industrial companies. To this end, the Ministry provides free of charge an *Herramienta de*

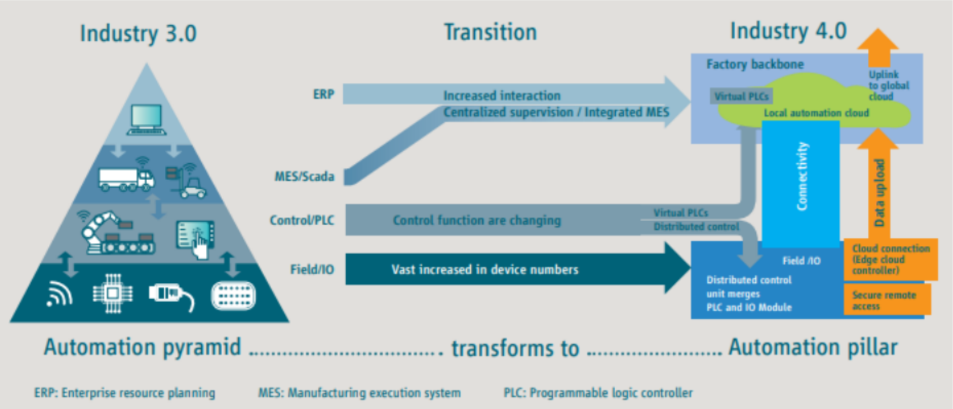

Industry 3.0
Transition
Industry 4.0
ERP
MES/Scada
Control/PLC
Field/IO
Increased interaction
Centralized supervision / Integrated MES
Control function are changing
Virtual PLCs
Distributed control
Vast increased in device numbers
Factory backbone
Virtual PLCs
Uplink to global cloud
Local automation cloud
Connectivity
Data upload
Field /IO
Distributed control unit merges PLC and IO Module
Cloud connection (Edge cloud controller)
Secure remote access
Automation pyramid .... transforms to .... Automation pillar
ERP: Enterprise resource planning
MES: Manufacturing execution system
PLC: Programmable logic controller

Figure 2.2: Industrial Transformation [2]

Autodiagnóstico Digital Avanzada (HADA) [36] that allows to evaluate the level of digital maturity and to compare the result with other companies. This tool provides a report that allows companies to plan and implement different actions to increase process actions to increase the productivity of production processes and the competitiveness between the external and internal market [37].

Industry 4.0 presents some necessary requirements to be able to comply with the new services that are emerging or will emerge. The main characteristics demanded by the industry are:

- **Low Latency**: Many industries rely on real-time communication for precise control, coordination, and decision-making. For example, the E2E packet must arrive immediately to avoid certain problem like the configuration of the determine machine.
- **Ultra Reliability**: Industries require robust and reliable systems that can function consistently under various conditions and loads. Downtime can lead to significant financial losses, so reliability is crucial to ensure uninterrupted operations. Also, the nodes that make up the network must ensure that the packets received have not suffered any failure or alteration in the transmission process. This is because it is impossible to retransmit the data continuously by the nodes, as they would suffer malfunctions in the quality of service of the service offered.
- **High Scalability**: Industrial operations often expand over time, and the systems in place should be scalable to accommodate increasing demands, whether in terms of data processing, network capacity, or production output. In addition, the industry demands a high number of connections to cover a large number of customers with strict QoS.

- **Mobility**: The IIoT demands a massive number of sensors and actuators, some of which require mobility in order to perform their functionality.
- **Security**: Industrial sectors handle sensitive data, valuable intellectual property, and critical infrastructure. Therefore, robust cybersecurity measures are essential to protect against cyber threats and unauthorized access.
- **Interoperability**: With the diversity of devices, equipment, and systems used in industries, interoperability is critical. Standardized communication protocols and open architectures allow different components to work together seamlessly.

The specific features demanded by industries can vary significantly depending on the sector, regulatory environment, and technological advancements. Meeting these demands requires collaboration between industry stakeholders, technology providers, and regulatory bodies to develop innovative solutions that address the unique challenges faced by each sector.

Currently, Industrial Ethernet (IE) is the dominant technology capable of meeting these quality of service requirements demanded to provide connectivity in mainly industrial environments. IE has essential features for industrial environments such as high resistance to weathering, high temperatures, vibrations and especially allows real-time device communication [38]. However, IE involves a set of standards where some of them are proprietary making them costly to implement and difficult to interoperate with each other. To solve this problem the IEEE has defined a layer 2 network technology of the OSI model.

TSN technology guarantees the QoS of traffic over a wired network. However, there is a clear preference for achieving device or client mobility through the use of wireless networks, such as 5G or Fourth Generation (4G). In addition, deployment costs are increased by the need to connect each of the devices via cabling. Furthermore, the existing technologies used do not guarantee that a certain task will be executed at the time, since, by using Ethernet, for example, they do not provide the determinism requirements needed for Industry 4.0. Due to the use of Carrier Sense Multiple Access (CSMA) as a mechanism for failure prevention in the transmission process, for example.

Therefore, thanks to 5G technology, the evolution to smart factories is enabled. Since 5G offers greater flexibility with lower cost and low latency, i.e. relatively shorter packet delivery times than previous generations. However, modern industrial networks present many use cases where synchronization with time, as is the case in closed-loop motion control, as well as the use of the cloud, is an important aspect. So, it must also be taken into account that industrial networks must be fault tolerant and with accurate time synchronization. In addition, low-power radio scheduling and distributed coordination within the network are necessary for proper time synchronization.

Consequently, one of the main problems with 5G in the areas where it is being deployed is its inability to deliver delay rates as low and with as much reliability as those required by Industry 4.0. This is not only on the radio side but also in the underlying E2E communications technologies. So it needs to incorporate other technologies to solve it. In this case, it is integrated with TSN, which allow to reduce the delay time, as well as the time synchronization between the devices that make up the network.

2.2 5G Network

5G networks are ideal candidates for future smart factories as they facilitate low-cost, highly robust interoperability between a high density of devices through reliable, low-latency wireless communications.

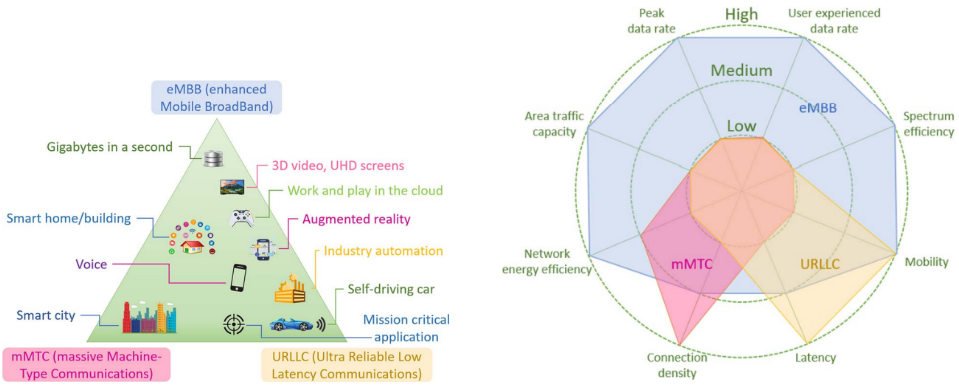
According to the expected requirements for 5G defined by International Telecommunications Union (ITU), three main types of services have been defined that provide the use of 5G:

- **enhanced Mobile Broadband (eMBB)**: is considered the evolution of the traditional connection by improving performance and user experience. The transfer rate requirements have been increased to about 20 Gbps on the downlink and 10 Gbps on the uplink with spectral efficiency improvements reaching up to 0.3 bps/Hz. Latencies are reduced to 4 ms and allow mobility between different base station coverage cells. In addition, it is considered that several deployment and service coverage scenarios can coexist, such as indoor/outdoor, urban and rural areas, offices and homes, local connectivity, etc. With these features, new types of services and more demanding applications such as high quality video, cloud services, augmented reality are enabled, for example.

- **massive Machine Type Communication (mMTC)**: service that supports scenarios that require high densities of interconnected devices, generating traffic with a lower volume of data that is not sensitive to delay. This type of service is closely linked to massive IoT, generalizing its use, with a wide range of devices with low cost, since they require few software and hardware requirements. In addition, they have low power consumption, allowing to host a large volume of traffic between them with high scalability, achieving up to 106 devices/km^2.

- **Ultra-Reliable and Low Latency Communication (URLLC)**: service that guarantee the very critical requirements in terms of end-to-end latency (below 1 ms), reliability and availability in the network (up to 99.999%). In other words, it welcomes those scenarios where low latency with high reliability and very high availability are required such as remote medical surgeries, remote driving vehicles, production processes in Industry 4.0, etc. For example, remote control processes for automation where 5 nines (99.999%) reliability is required with data rates of 100 Mbps and end-to-end delay of 50 ms. This is achieved through the *Edge Computing* or *Fog Computing* capabilities of 5G.

Figure 2.3a shows the different 5G services and their use cases, while Figure 2.3b shows the requirements for each type of service.



(a) Usage scenarios

(b) Requirements

Figure 2.3: Usage scenarios and requirements of 5G [3]

In order to carry out the implementation of these services, a series of 5G specifications have been developed by Third Generation Partnership Project (3GPP) [39]. 3GPP is the organization in charge of standardizing the radio part and the architecture of 5G systems. Its main objective is to ensure interoperability between network elements regardless of the manufacturer. In 2018, *3GPP Release 15* [5] was published, indicating the basic components that make up the 5G architecture and the characteristics of communications. *3GPP Release 16* includes new enhancements with the definition of URLLC communications and their integration with TSN for industry automation. In 2022 *Release 17* [40] will be published focusing on improvements for the reduction of the complexity of the radio part, augmented reality and better position accuracy, mainly. Currently, 3GPP is working on the development of *Release 18* focusing on the development of new capabilities for fixed/mobile broadband as well as industry verticals driven by artificial intelligence, machine learning and full duplex technologies based on a single platform [41]. This evolution can be seen in the Figure 2.4.

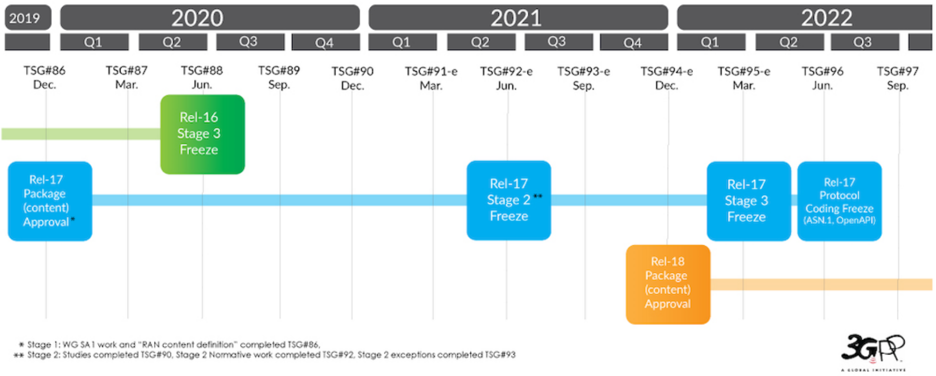

2019
2020
2021
2022
Q1
Q2
Q3
Q4
Q1
Q2
Q3
Q4
Q1
Q2
Q3
TSG#86 Dec.
TSG#87 Mar.
TSG#88 Jun.
TSG#89 Sep.
TSG#90 Dec.
TSG#91-e Mar.
TSG#92-e Jun.
TSG#93-e Sep.
TSG#94-e Dec.
TSG#95-e Mar.
TSG#96 Jun.
TSG#97 Sep.
Rel-16 Stage 3 Freeze
Rel-17 Package (content) Approval*
Rel-17 Stage 2** Freeze
Rel-17 Stage 3 Freeze
Rel-17 Protocol Coding Freeze (ASN.1, OpenAPI)
Rel-18 Package (content) Approval
* Stage 1: WG SA1 work and "RAN content definition" completed TSG#86,
** Stage 2: Studies completed TSG#90, Stage 2 Normative work completed TSG#92, Stage 2 exceptions completed TSG#93
3GPP
A GLOBAL INITIATIVE

Figure 2.4: Evolution of the versions published by 3GPP [4]

Currently, 5G networks have mainly two possible deployments, as shown in Figure 2.5. However, there are different deployment alternatives, as explained in [42]. The two deployments are:

- **5G *Non Standalone (NSA)* Network**: the 5G Radio Access Network (5G-RAN) and its New Radio (NR) interface are used in conjunction with the 4G core network, specifically in conjunction with the existing Evolved Packet Core (EPC). This makes the NR interface available without the need to modify the network infrastructure. The management of user traffic is shared between the two network nodes (5G and 4G) while the management of internal communication with the EPC is only performed by the Evolved Node B (eNB) node (4G) or the Next Generation Node B (gNB) node (5G).

- **5G *Stand Alone (SA)* Network**: The NR interface is connected to the 5G core, instead of the 4G core as in the previous case. This deployment is the only one that allows the incorporation of all the 5G services mentioned above.

Regardless of the deployment implemented the 5G network architecture is divided into two the Radio Access Network (RAN) and the 5G Core (5GC). Figure 2.6 shows the 5G network architecture.

Each of the main parts into which the 5G network is divided is detailed below:

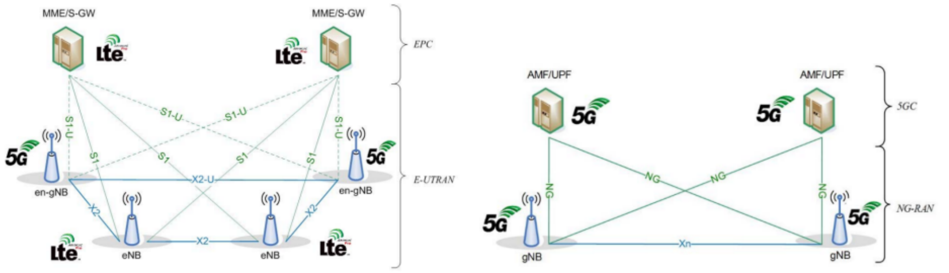


Figure 2.5: Deployment of the 5G NSA and SA network [5]

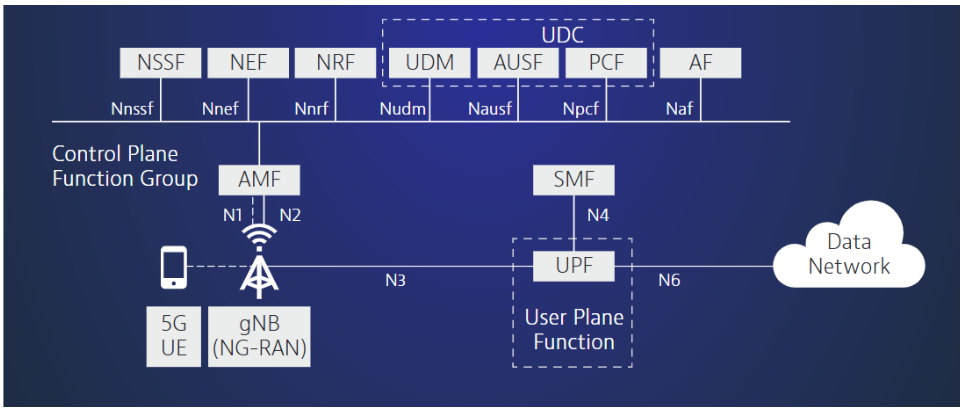


Figure 2.6: 5G Network [6]

- **Radio Access Network (RAN)**: also referred to as *fronthaul* or xHAUL networks, as they implement the concept of Network Functions Virtualisation (NFV). Thanks to this functionality, new features and technologies can be implemented. The RAN is mainly composed of a single gNB that connects to the 5G core through the NG interface and this to the CN interface. It connects to the User Equipment (UE) through the radio interface. On the other hand, it can be connected to another gNB through the Xn interface and/or to a eNB through the X2 interface. These interfaces and elements can be seen in Figure 2.7.

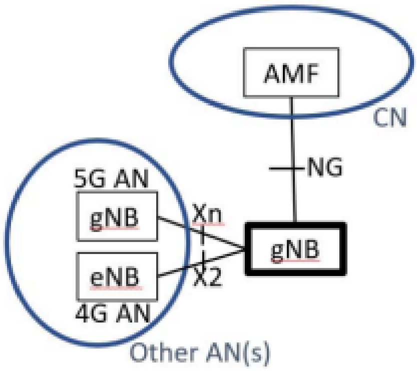


Figure 2.7: 5G RAN [5]

With the latest versions of the 5G standard, new technologies have emerged in the radio part such as massive Multiple Input Multiple Output (mMIMO). This technology consists of the use of multiple antennas in the gNB in order to increase the spectral efficiency and energy of the system. This functionality is possible thanks to the focusing of the different beams that make up the radio access signals by means of a beamformer. In this way, it is possible to reduce the transmission power by adding up all the signals arriving on the different channels with different delays and making an interferometric arrangement individually for each user in the corresponding uplink transmission time slot. With the use of this technique it is unnecessary to use the Cell-specific Reference Signal (CRS), characteristic of each of the cells for the estimation of the channel of the devices. This information is properly transmitted in 5G, reducing the power consumption of the estimation and consequently the interference, since no user data is transmitted. Also, these antennas are self-adaptive depending on the Doppler effect. That is, the beam directivity is increased as more antennas are added. In this way, lower power consumption is achieved by taking advantage of the propagation along a better path and the same resources between devices. This allows higher speeds, very useful for example for self-driving vehicles.

- **5G Core (5GC)**: This part of the 5G architecture performs user data processing and integration with the 5G network. It also performs the signaling in the network. The 5GC is made up of different nodes connected to each other through point-to-point interfaces to an architecture Service Based Architecture (SBA), based on services and storage. The core of the network is realized by means of NFVs. These functions are responsible for the separation between the data and control plane, achieving greater flexibility and simplicity to meet the needs of each application. The Figure 2.8 shows the composition of 5GC.

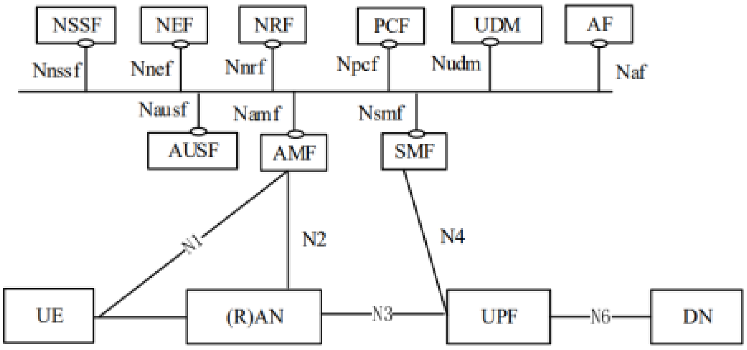


Figure 2.8: Elementary network functions in a 5G network [7]

The following is a brief description of each of the most relevant network functions of the SBA architecture:

- **Access and Mobility Management Function (AMF)**: This network function is responsible for the overall control of the network for signaling, i.e. between the RAN, where the UEs are located, and the 5G backbone. Among the functionalities is the registration in the network, authentication, mobility between cells, encryption, session establishment, location services, etc. Some of these functions are performed with the help of other network functions. Also, it supports *network slicing* and the selection of the corresponding Session Management Function (SMF). In short, this function acts as a boundary zone between the network core and the access network.
- **Session Management Function (SMF)**: is responsible for the provisioning of user sessions, in particular, the establishment,

modification and release of the various sessions between the network and the UE. It is also in charge of assigning the Internet Protocol (IP) addresses of the UEs that are connected to the network, managing the traffic that is sent to the User Plane Function (UPF) and controlling the application of the policies that have been applied and the QoS.

- **User Plane Function (UPF)**: is responsible for processing the user plane of both links and the forwarding of both links from the corresponding station to the backbone or external network. In other words, the UPF performs the forwarding and communication with the core network and the data network. This network function is controlled by the SMF. It also processes the data that has been forwarded to generate traffic reports, analyze the content of data packets and/or execute network or user policies.

- **Authentication Server Function (AUSF)**: This network function is the provider of the authentication service for devices connecting to the network. To do this, it is responsible for requesting connection processing and delivery of device credentials to the Authentication Credential Repository and Processing Function (ARPF) function.

- **Unified Data Management (UDM)**: is a database containing mobile subscriber data. It is responsible for the generation of authentication credentials that are used to allow devices to connect to the network. After verification, it authorizes the access of these devices to the information available in this database.

- **NF Repository Function (NRF)**: is responsible for the management of *Network Functions* services. Some of these actions are registration, authorization, discovery and deregistration.

- **Network Exposure Function (NEF)**: is responsible for handling externally sourced data. That is, all external applications must pass through this network function as if it were a secure Application Programming Interface (API) when accessing internal 5G network data. It also participates in routing and traffic policies.

- **Network Managment System (NMS)**: is a database used for network management. It contains all the necessary management information including 5G network configuration parameters, flow information, storage of Key Performance Indicators (KPIs), etc. Therefore, it is responsible for controlling all relevant information of the E2E connection establishment and allows implementing the network slicing functionality in the 5G network.

- **Policy Control Function (PCF)**: this network function applies the adopted policies in a unified way through a common framework. In order to verify correct compliance, it performs network behavior monitoring processes and guarantees QoS.
- **Network Slice Selection Function (NSSF)**: is responsible for selecting the network segments, coordinating with the AMF for their establishment. In this way, the UE can use different network segments simultaneously, differentiated by an identifier associated with each of these segments. Each of these network segments has linked selection policies.
- **Application Function (AF)**: is responsible for executing the functionality of the application server. That is, it interacts with the other network functions found in the control plane according to the type of service and the network properties. In this way it performs different operations such as interacting with the PCF for policy control, exposing services to end users, define the routing of application traffic through the Network Slice Subnet Instance (NSSI), etc. Likewise, it is the point of interconnection between the 5G network with other systems to be able to interact between them, as can happen with TSN, where this network function is in charge of exposing the available 5G network resources according to existing policies.

These virtualized network functions make it possible to implement different network slices to create E2E logical networks that can be adapted to different use cases. These networks are implemented on the same physical resources that are shared thanks to SDN and NFV technologies. The various network segments formed on the same structure are completely isolated from each other, with independent control and management. In addition, new segments can be created as needed, i.e. on demand. For example, Figure 2.9 shows a single physical infrastructure that implements several 5G network segments adapted and managed independently of each other. 5G network segments adapted and optimized for different use cases such as user applications (blue slice), eHealth (green slice) and machine-to-machine communications (red slice).

Just as there are modifications in the 5G network architecture, there are also new improvements in the physical layer, opting for more sophisticated solutions, mainly in the modulation used. In the uplink, DFT-S-OFDM is used, which is based on Orthogonal Frequency Division Multiplexing (OFDM) using Discrete Fourier Transform (DFT) precoding, and in the downlink Orthogonal Frequency Division Multiple Access (OFDMA) with a cyclic prefix is used.

In 5G networks for the physical layer, two frequency ranges have been

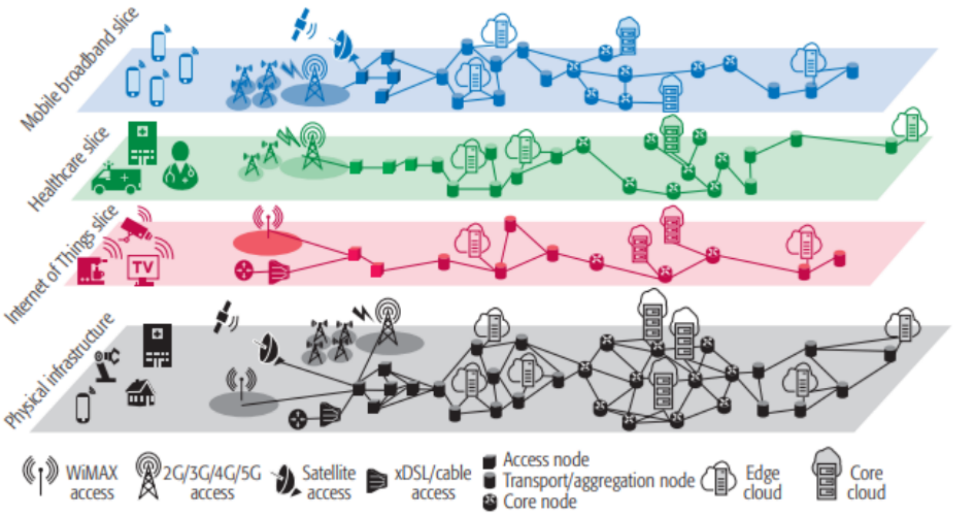


Figure 2.9: 5G network slices running on the same physical infrastructure [8]

specified: FR1 (sub-6-GHz) and FR2 (millimeter bands or mmWare). The FR1 band has a frequency range of 410 to 7125 MHz with carrier spacing of 5/10/15/20/25/ 30/40/50/60/80/90/100 MHz. While FR2 has a frequency range of 24250 to 52600 MHz with carrier spacing of 50/100/200/400 MHz [5].

Figure 2.10 shows a scheme with the two frequency bands used in 5G networks, as well as in 4G and Second Generation (2G)/Third Generation (3G) networks. It is observed that the FR2 or mmWare band are the ones that provide a higher channel bandwidth with lower latencies but with a lower coverage range. By increasing the frequency, it is observed that the bandwidth increases as the power consumed is reduced due to the fact that the interference with other devices or base stations disappears. However, increasing the frequency reduces the distance, causing it to be used only in enclosed and smaller spaces.

All multiple access mechanisms and procedures, physical channels, modulation, channel coding, etc., are defined in the 3GPP standard [5].

Other improvements that have been introduced with the standard is the use of a more flexible frame structure that presents different Subcarrier Spacings (SCS). The SCS is the distance between two consecutive subcarriers. Each subcarrier is made up of an OFDM symbol and a carrier configuring the most basic unit of radio resource considered in 5G, called Resource Element (RE). Each series of 12 subcarriers makes up a Resource Block (RB). Therefore, the bandwidth of a NR channel depends on the number of RBs, i.e., the greater the number of RBs the greater the bandwidth used. In the

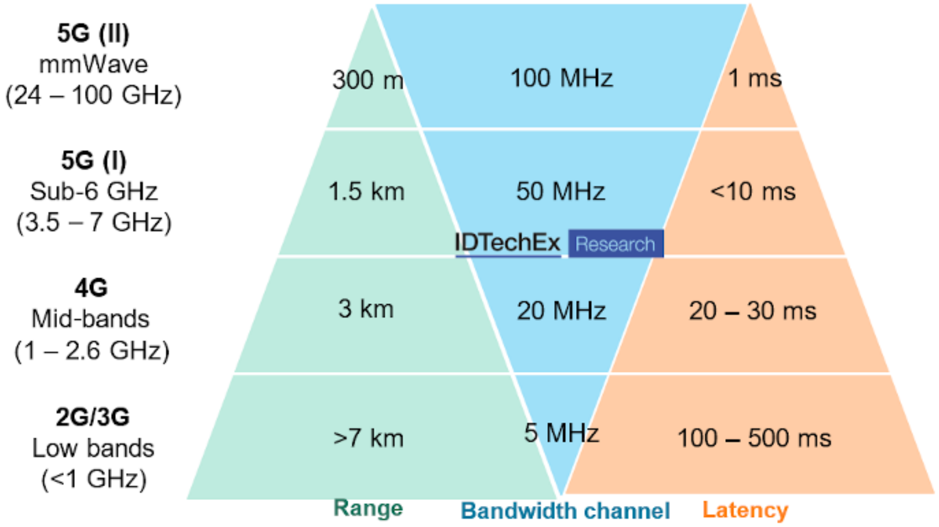


Figure 2.10: Frequency bands of 5G systems according to their use up to 40 GHz [9]

time domain, each complete frame, with a duration of 10 ms, is divided into 10 subframes. That is, each subframe has a duration of 1 ms. In turn, each of these subframes is divided into 1 time slot consisting of 14 OFDM symbols. The slot duration depends on the carrier transmission frequency. The described distribution is shown in Figure 2.11.

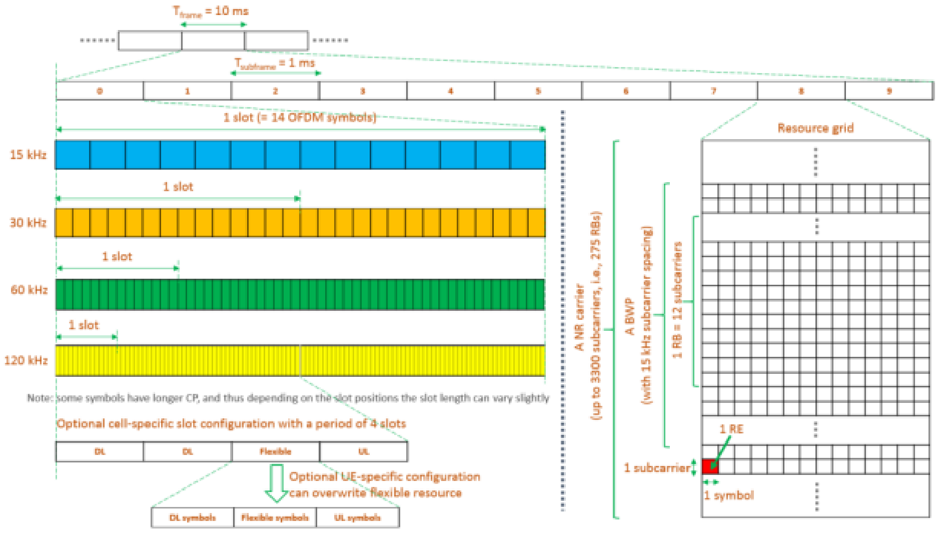

T_{frame} = 10 ms
$T_{subframe}$ = 1 ms
0
1
2
3
4
5
6
7
8
9
1 slot (= 14 OFDM symbols)
15 kHz
1 slot
30 kHz
1 slot
60 kHz
1 slot
120 kHz
Note: some symbols have longer CP, and thus depending on the slot positions the slot length can vary slightly
Optional cell-specific slot configuration with a period of 4 slots
DL
DL
Flexible
UL
Optional UE-specific configuration can overwrite flexible resource
DL symbols
Flexible symbols
UL symbols
A NR carrier (up to 3300 subcarriers, i.e., 275 RBs)
A BWP (with 15 kHz subcarrier spacing)
1 RB = 12 subcarriers
Resource grid
1 RE
1 subcarrier
1 symbol

Figure 2.11: Frame structure in 5G NR [10]

In the uplinks and downlinks to carry out the configuration of the existing channels between the gNB and the UE, a series of information is transmitted through these radio resources. For each link, uplink and downlink, the main channels are:

- **Uplink**:
  - *Physical Random Access Channel (PRACH)*: is used to request the connection request through random access by the UE. That is, to be able to establish a call or transmit a certain data burst.
  - *Physical Uplink Shared Channel (PUSCH)*: transmits the information from the UE by reserving a secure channel.
  - *Physical Uplink Control Channel (PUCCH)*: is used for upstream channel control information including Hybrid Automatic Repeat reQuest (HARQ), scheduling request and downstream channel status feedback information.
- **Downlink**:
  - *Physical Downlink Shared Channel (PDSCH)*: is used to transmit downlink user data over an authenticated secure channel.
  - *Physical Downlink Control Channel (PDCCH)*: transmits control information such as scheduling decisions between PDSCH and UE data via the PUSCH. The goal is to be able to adapt to variations that occur in the channel through the use of HARQ.
  - *Physical Broadcast Channel (PBCH)*: is used for the broadcast information systems that UEs need to be able to access the network.

2.3 Time Sensitive Networking (TSN)

TSN is a set of standards that are specified as a series of amendments to *IEEE 802.1* defined by the IEEE. In other words, it encompasses all the standards for Ethernet encapsulation (link layer or layer 2 of the OSI model) but for Virtual Local Area Network (VLAN) networks. Thanks to these standards, critical challenges in various sectors are solved by ensuring the deterministic transmission of flows with QoS in terms of strict requirements for latency, jitter, reliability and packet loss. This is because TSN focuses on minimizing the delay and jitter of packets being transmitted while the main objective of Ethernet networks is to increase bandwidth. Because of these capabilities, TSN technology is currently key to the development of deterministic networks, such as industrial networks or 5G networks.

TSN networks are made up of end devices, bridges, network-user interfaces and the TSN flows themselves. A brief description of each of these is given below:

- **End Devices**: the source and destination nodes of the flows. In other words, these are the nodes that run the applications and services that require deterministic transmissions.
- **Bridges**: are Ethernet switches that transmit the data frames of the TSN flows and receive them according to a defined time schedule.
- **User/Network Interface (UNI)**: is the connection between the user plane and the control plane of a TSN network. The user-side UNI is made up of the senders and receivers, while the network-side UNI is made up of the bridges that transmit the data frames from the sender to one or more receivers.
- **TSN Flows**: are unidirectional time-critical data frames. These frames are transmitted between end devices that require the timing to be deterministic. In addition, each stream has a unique identifier for each end device.

These components make it possible to form the control and data plane of the TSN network. One of the objectives of the TSN is to determine the flow requirements without the need for knowledge of the network. To this end, the flow requirements must be obtained from the network, as well as the network topology and the capabilities of the bridges present in order to configure these bridges to meet the demanded requirements. In the TSN standard, different control plane models have been defined for resource allocation, configuration, registration and management. So the discovery of network requirements is obtained differently depending on the control plane model. The three architectures detailed in the IEEE 802.1Qcc standard are: fully distributed, network centralized and user distributed and fully centralized, although the standard focuses mainly on the third model. The following is a brief explanation of the control plane models:

- **Fully Distributed Model**: user requirements are transmitted throughout the topology by using a distributed protocol. Specifically, each network component shares the properties required by the rest of the network nodes to establish TSN flows with QoS. In this model, the UNI interface is located between the end stations and the first or final TSN bridge in the network. The Figure 2.12 shows the representation of the fully distributed model of the TSN network.

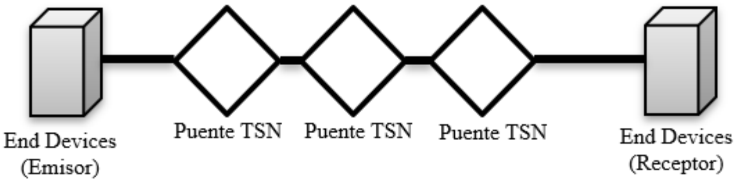


Figure 2.12: Fully distributed model [11]

- **Centralized Network and Distributed User Model**: a new component called Centralized Network Configuration (CNC) has been added, which contains a global view of the entire network topology and the frames to be transmitted. This component is responsible for the configuration of the TSN bridges, for the complex performance operations required by certain mechanisms such as schedules, and for the scheduling of the frames and routes to be followed by the flows on the bridges. In this case, the UNI is located between the bridge and the end station. User requirements are transmitted from the sender to the network edge bridge, a bridge connected to an end station, which communicates them to the CNC. A representation of the centralized network and distributed user model is shown in Figure 2.13.

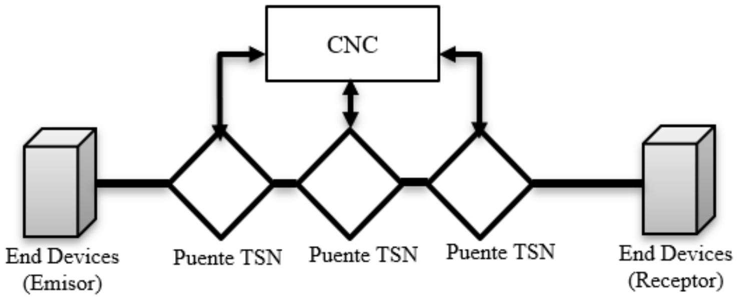


Figure 2.13: Centralized network and distributed user model [11]

- **Fully Centralized Model**: another new component called Centralized User Configuration (CUC) is added, which is in charge of discovering the end stations and receiving their characteristics and the requirements of the TSN flows. In this case, the exchange of user requirements is done between the CNC and the CUC, so the UNI is located between these two components. A representation of the fully centralized model is shown in Figure 2.14.

  Between the CUC and CNC there are a series of message exchanges to obtain the requirements. This information flow is:

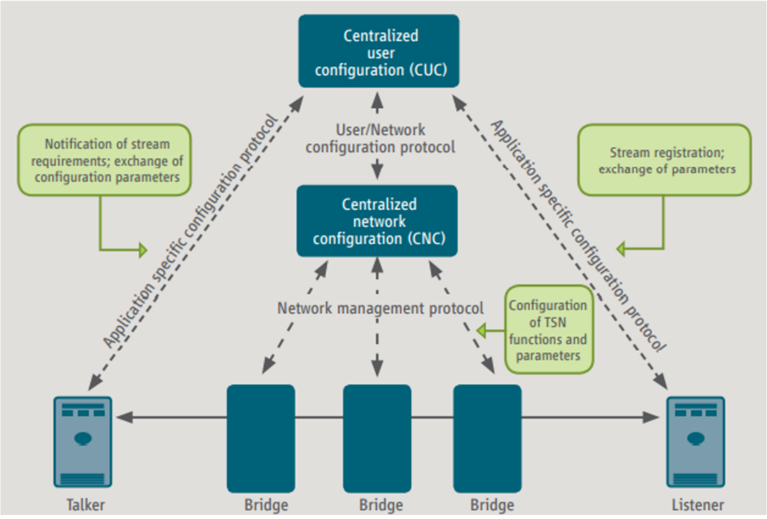

Centralized user configuration (CUC)
User/Network configuration protocol
Centralized network configuration (CNC)
Notification of stream requirements; exchange of configuration parameters
Stream registration; exchange of parameters
Application specific configuration protocol
Application specific configuration protocol
Network management protocol
Configuration of TSN functions and parameters
Talker
Bridge
Bridge
Bridge
Listener

Figure 2.14: Fully centralized model [2]

1. The end devices send their QoS requirements to the CUC. Some of these requirements are for example data rate, traffic classes, priorities, E2E latency, etc.
2. The CUC forwards this information to the CNC via UNI.
3. On the other hand, TSN bridges transmit their capabilities to the CNC. Some of these capabilities are for example bridge delays according to port and traffic class or scheduling delays according to port and priorities considered.
4. The CNC uses this information to determine the configuration of each bridge to meet the requirements of the TSN flows. It also defines the traffic transmission schedule according to the start times of the flows and the control times of the TSN schedulers.
5. After acquiring the configuration of the TSN bridges, the CNC transmits this information to the CUC and the CUC forwards it to the end devices.

Although three models of the control plane have been defined, the fully distributed model and the centralized network and distributed user cannot be implemented since they must use the Stream Reservation Protocol (SRP) protocol which does not meet the needs required by industrial networks. Therefore, the main model studied and used is the third model, i.e., completely centralized and the one on which this work focuses. In these networks, multiple CUC can coexist, but only one CNC entity. With the existence of the CNC and the CUC, the automation of network management is implemented through the SDN paradigm. In this way, it is possible to

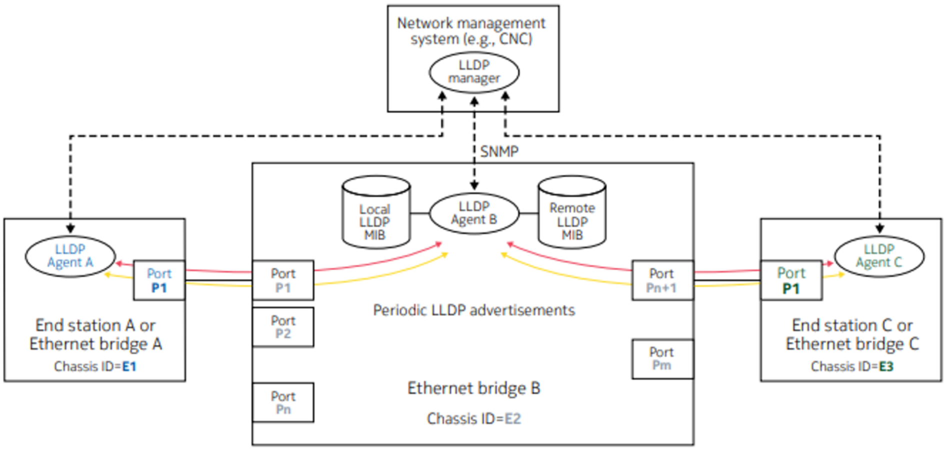


Figure 2.15: LLDP information exchange

know if the network meets the QoS requirements prior to the establishment of the network configuration.

The CNC to perform end-to-end route planning and optimization needs to know the complete network topology. For this discovery of the links between the bridges and the end stations of a network, a Link Layer Discovery Protocol (LLDP) is implemented. This protocol is defined in the IEEE 802.1AB standard and its functionality allows both the discovery of the entire network topology and the state of the network devices and their availability. This protocol deploys a set of LLDP agents at the end stations or bridges to perform the information exchange between neighboring nodes. LLDP frames are specific to each outgoing port in order to determine the network topology. These LLDP frames are received at the agent of the end stations or a bridge. This agent is in charge of verifying the frames and storing the information contained in a given remote Message Information Base (MIB). All the stored information can be retrieved by each bridge if management interfaces are used, such as those based on the Simple Network Management Protocol (SNMP) or MIB. The LLDP message exchange described above can be seen in Figure 2.15.

The SNMP is an IP-based protocol used to monitor and manage network bridges, networks, and end stations. A management interface is built between the bridges and the CNC in a centralized TSN network to receive information on the status and current configuration of the network nodes. Each of the current SNMP agents in the bridges or end stations transmits this information. Furthermore, these SNMP agents can change the state of network nodes and send notifications to the CNC, which is the network management system, to notify it of particular events. A MIB is used in all processes.

TSN, like Ethernet, are shared access technologies to the medium through the Time-Division Multiplexing Access (TDMA) technique in order to access the medium when the exact time of transmission is known. However, in the case of traffic where the exact time of transmission is not known, statistical multiplexing is used. The main difference between the two techniques is that with TDMA, time slots are pre-assigned to all existing communications, regardless of whether they are active or not. Whereas with statistical multiplexing, slots are only assigned when traffic is active. In this way, it is possible to increase channel utilization without wasting the temporary slots assigned to inactive communications.

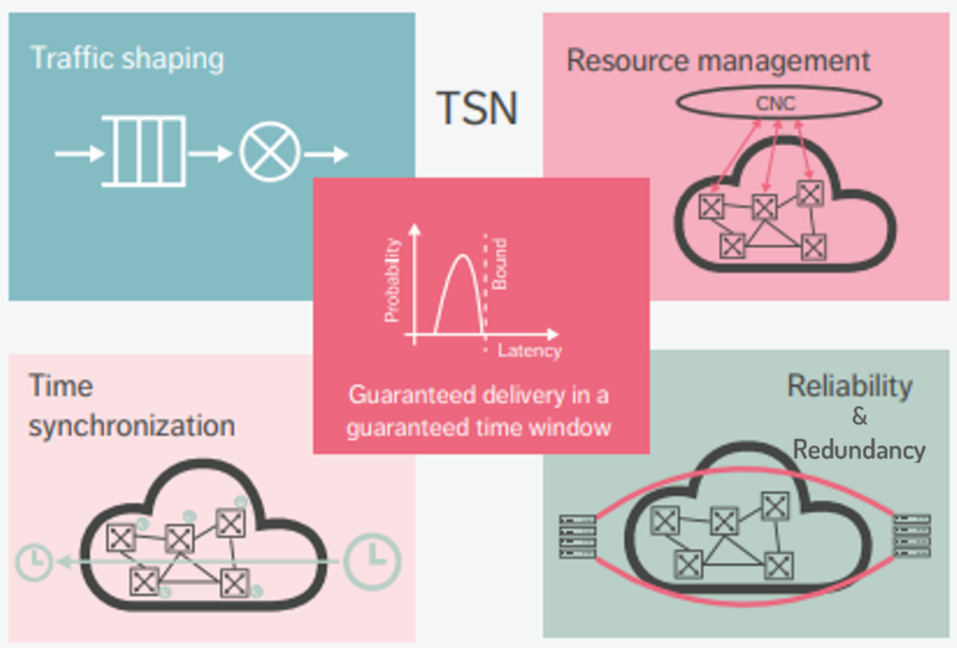


Figure 2.16: TSN features [12]

Figure 2.16 summarizes the main characteristics of TSN networks. As can be seen, they can be grouped into 4 blocks: traffic shaping, resource management, time synchronization and reliability. Each of these blocks groups together those IEEE 802.1Q standards that allow the characteristic to be developed. A brief description of each of the blocks is given below:

- **Traffic Shaping**: groups the standards that guarantee latency and jitter in a TSN network. For this purpose, TSN separates the traffic into traffic classes depending on the required QoS. Each traffic class groups the traffic of the services contemplated by the scheduler depending on their characteristics and requirements, being treated differently according to these. There are two types of schedulers defined in the standard: ATS and TAS. These two schedulers determine the configuration implemented in each of the outgoing traffic of the switch/bridge outgoing interface and filter the flows according to their QoS. The TAS, defined in the IEEE 802.1 Qbv standard, is used in synchronous TSN networks where it is necessary for all nodes to be temporally synchronized. While the ATS, defined in the IEEE 802.1Qcr standard, is used for asynchronous TSN networks where the nodes do not have to be temporally synchronized. Another mechanism also used in this block is *Ethernet frame preemption*, defined in the IEEE 802.3br and IEEE 802.1Qbu standards. This mechanism allows interrupting the transmission of lower priority frames in order to transmit high priority frames, reducing the transmission delay of critical frames.

- **Resource Management**: is realized differently depending on the TSN network configuration architecture. Resource management, regardless of the control plane architecture, enables dynamic discovery of the TSN network topology. It also provides configuration, network monitoring, allocation and registration of the resources needed to ensure the requirements of each flow, among other properties. For this purpose, the IEEE 802.1Qat and IEEE 802.1Qcc standards are defined, which are responsible for flow reservation and the IEEE 802.1CS standard for link-local reservation.

- **Reliability and Redundancy**: the IEEE 802.1CB standard is defined, which implements the Frame Replication and Elimination for Reliability (FRER) mechanism. This mechanism consists of transmitting several replicas of the same frame but over different paths of the network, avoiding the interruption of communication due to the fall of any link. It also implements path control and reservation techniques (IEEE 802.1Qca) and filtering techniques and control policies for each flow (IEEE 802.1Qci).

- **Time Synchoronization**: Each node in the TSN network has its own clock. These clocks initially have the same time reference but can suffer cumulative deviations and cause a malfunction of the network. To avoid this, all the clocks in the network must be temporarily synchronized. To do this, the generalized Precision Time Protocol (gPTP) mechanism, a profile of the Precision Time Protocol (PTP) standard defined in the IEEE 802.1AS standard, is implemented. This mechanism updates the time of each clock taking into account the possible deviations that may occur along the way. To do this, it first estimates the latency introduced into the network by exchanging a series of messages and then, this calculated latency is used to synchronize the clocks of the nodes.

As indicated in the main TSN features, TSN singles itself out for traffic conformation. For this, it divides the service traffic according to its QoS requirements, differentiating it by means of an identifier Priority Code Point (PCP) that is assigned to each traffic. This identifier is located in the VLAN tag of the Ethernet header of the frame and is assigned according to criticality and QoS. Criticality indicates the amount of risk posed by the data, specifically its level of system availability and the severity of system performance when a frame is lost. The PCP determines the priority of the frame, where a total of eight values are normally considered, from "0" to "7". Value "0" is usually assigned to traffic of type Best-Effort (BE), being the default class, while value "7" is assigned to traffic of higher priority [2].

The assignment of a given PCP value to traffic types is done through the strict priority mechanism, defined in the IEEE 802.1Q standard. The functionality of this mechanism is to perform the mapping between PCP values and Traffic Class (TC) depending on the number of TC supported on an egress port of the TSN bridge.

This identifier, the PCP, can be cataloged individually or as a group depending on the number of queues or classes that have been configured on the outgoing ports after the packet has been communicated. If each PCP value can be related to a different class then it can be treated uniquely according to the policies for that flow. However, there may be situations where the number of TC contemplated is less than the number of possible PCPs. For example, if the outgoing port of the bridge supports 8 TC, a PCP value will be associated with each TC, which will have its own queue waiting to be transmitted by the interface. But if on the other hand the number of TC is less than 8, then several PCPs must be merged into a single TC and this queue would have traffic with several PCP values. The latter case is described in Figure 2.17.

With scheduled traffic and TC differentiation, the interference suffered by TSN flows when transmitted over the network is considerably reduced. However, they still do not meet the demanding QoS requirements demanded by the industry for certain types of traffic. To achieve this, the frame prevention mechanism defined in the IEEE 802.3br and IEEE 802.1Qbu standards is implemented. This mechanism allows higher priority packets to preempt traffic with a lower priority to meet the latency requirement. This mechanism can cooperate with scheduled traffic to reduce latency.

Another mechanism implemented to achieve high reliability is the redundancy of the frames of the most critical services. FRER is defined in the IEEE 802.1CB standard and is used to increase the number of frames belonging to a given flow. This mechanism also ensures that the network is not unnecessarily overloaded as it is capable of detecting duplicate flows and eliminating them. In this way, redundancy transparency is guaranteed

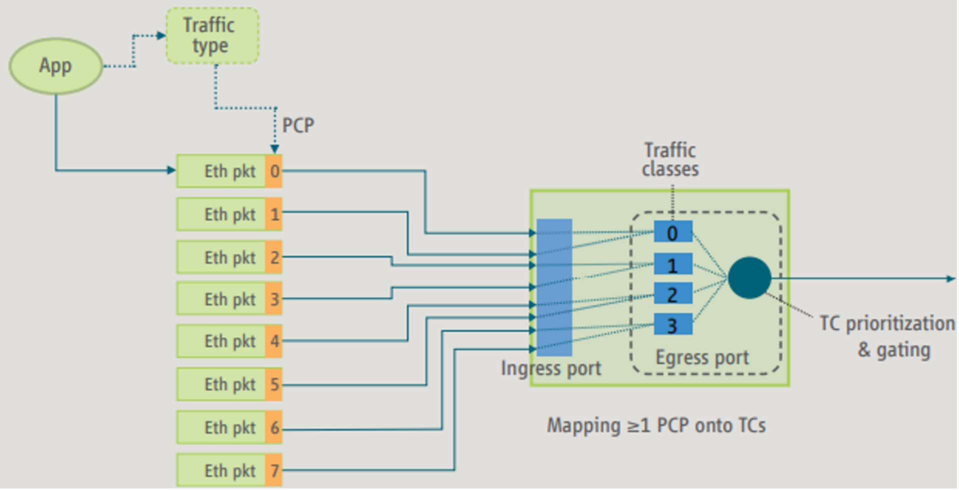


Figure 2.17: Mapping of PCP to traffic classes [2]

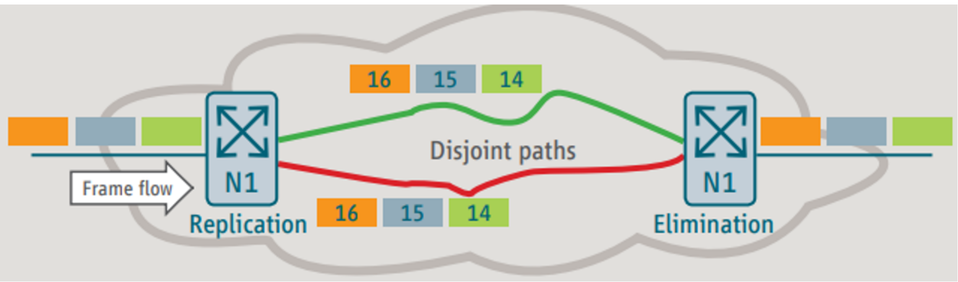


Figure 2.18: FRER mechanism [2]

for the application and its realization within the network. Thus, the FRER mechanism performs two main functions, as shown in Figure 2.18:

- **Replication Function**: is responsible for duplicating the frames and transmitting them over two or more disjoint routes. Each of the copies is assigned the same sequence number in order to facilitate its subsequent elimination.

- **Delete Function**: is responsible for the elimination of all duplicate frames received after a previous frame. To determine if the frame is a duplicate of the previous frame received, the sequence number associated with each frame is examined.

Also to ensure reliability and redundancy, the Per-Stream Filtering and Policing (PSFP) mechanism, defined in IEEE 802.1Qci, has been implemented. This mechanism allows the identification and management of non-comforming traffic, excessive bandwidth usage, whether intentional or un-intentional, and incorrect prioritization within a given time interval. To achieve these objectives, different control actions are implemented such as flow meters to provide data-driven surveillance or flow gates to provide time-based surveillance. A brief description of both monitoring is given below:

- **Data-based Surveillance**: uses flow meters that are applied to one or more TSN flows. These meters allow to provide the committed information rate and the information rate of the flows. The purpose of obtaining these measurements is to check whether the transmission information rate that is allowed to be transmitted exceeds the information rate.

- **Time-based Surveillance**: employ flow gates that must be temporally synchronized between the bridges and the end devices of the network. This monitoring mechanism is necessary mainly in synchronous networks since the scheduler gates are configured to open at certain time instants. If frames arrive outside this opening time slot, they are considered to be unwanted frames or interference. These frames are discarded since at that instant the gate is closed and does not allow access to the bridge.

In TSN networks there are two types of schedulers, depending on whether time synchronization is required or not, as previously mentioned. In synchronous TSN networks, their nodes must have the same time reference in order to guarantee the deterministic behavior of the network. For this purpose, the gPTP protocol is used for time synchronization and the TAS scheduler and the use of cyclic queuing, defined in the IEEE 802.1Qch standard, for the transmission of scheduled traffic. On the other hand, asynchronous TSN do not need time synchronization mechanisms to guarantee deterministic behavior. Therefore, in these networks their nodes do not need to have the same time reference to meet the QoS requirements demanded by the flows. Consequently, depending on the use of each type of TSN network, there are a series of advantages or disadvantages. One of the most important is that synchronous TSN are more expensive because of the need for time synchronization and the capacity utilization of their links is minimal due to the reservation of time slots. However, since asynchronous TSN networks do not have this time synchronization, there is an increase in the delay suffered by TSN flows, although they still comply with the required restrictions.

Table 2.1 provides a more detailed comparison of the advantages and disadvantages of synchronous and asynchronous TSNs.

Type of networks	**Advantages**	**Disadvantages**
Synchronous TSN network	Ideal for handling periodic deterministic traffic patterns. Features lower latency.	Not adaptable to aperiodic aperiodic deterministic. Involves higher configuration complexity. Uses more expensive technology. Exhibits lower link utilization. Scalability problems due to time synchronization.
Asyncrhonous TSN network	Well-suited for aperiodic (sporadic) deterministic traffic patterns. Exhibits lower configuration complexity. Provides higher link utilization. Offers higher scalability.	Incurs higher latency. Yields lower link utilization. Tends to be more expensive. Not suitable for periodic deterministic traffic patterns. Entails higher configuration complexity.

Table 2.1: Comparison between synchronous and asynchronous TSN networks [14]

With respect to planners there are also a number of disadvantages on the part of the TAS:

- TAS requieres synchronization temporal: The TAS uses a clock signal that degrades as the number of switches increases, as is the case with wireless links.
- TAS gating cyclic not synchronized with cyclic (non isochronous) applications: i.e., the TAS may receive non-cyclic traffic where the exact time of arrival of the packet is not known and therefore, such traffic is considered unwanted and is discarded according to the frame prevention mechanism and/or as it is traffic normally with high latency

requirements it cannot be transmitted until the next time slot causing its latency to increase.

- TAS requires complex scheduling process: i.e. the TAS must define several Gate Control List (GCL) in order to carry out data transmission taking into account the QoS requirements of the types of traffic arriving at the scheduler. It therefore involves a series of more complex mechanisms than those used by the ATS scheduler.

- TAS hardly compatible with VNF.

The selection between synchronous TSN networks and asynchronous TSN networks depends on specific network requirements, including the type of traffic, desired latency, scalability, and configuration complexity. It is crucial to consider these factors when deciding which network type is most suitable for a given application or use case. Although most of the use cases and articles studied employ the use of synchronous TSN networks, in this case we have chosen to study asynchronous TSN networks. That is, networks that implement the ATS scheduler, since it is considered to have great benefits that have not been studied in depth to date.

2.4 Asynchronous TSN

This paper focuses on the use of asynchronous TSN. In asynchronous TSN, neither time synchronization mechanisms (IEEE 802.1AS) nor synchronous TSN traffic scheduler algorithms (IEEE 802.1Qch and IEEE 802.1Qbv) are necessary. Because nodes in asynchronous TSN networks do not need the same time reference in their clocks to meet the QoS requirements demanded by the flows. In addition, synchronous TSN have major weaknesses due to the need for time coordination between their nodes, which hinders network scalability, and the use of reserved time slots for flows. This results in poor utilization of link capacity. Therefore, asynchronous TSN implement another scheduler that does not need a common time reference between all TSN bridges and improves the scalability and utilization of the links. The main difference is the use of another scheduler, since asynchronous TSN can implement the rest of the mechanisms explained in section 2.3. In the following subsections the scheduler used by asynchronous TSN is explained and a small review of the state of the art of existing articles and papers on prioritization schedulers is made.

2.4.1 ATS Scheduler

The scheduler employed by asynchronous TSN networks is Asynchronous Traffic Shaper (ATS) standardized in IEEE 802.1Qcr. The ATS develops an asynchronous method for processing TSN frames at the TSN bridge output ports. This mechanism is based on Urgency-Based Scheduler (UBS) proposed by researchers Specht and Samii [43]. UBS uses interleaved formed queues to regulate traffic and a strict priority queue to prioritize traffic. Specht and Samii consider a leaky bucket on asynchronous shapers for the of flow. The ATS can be a practical implementation of the UBS in 802.1Q standards [44]. In this paper, we adopt the nomenclature used in [43]. First, we explain how UBS works in order to subsequently understand the different stages of which the ATS scheduler is composed, since some of these stages develop processes similar to those of UBS.

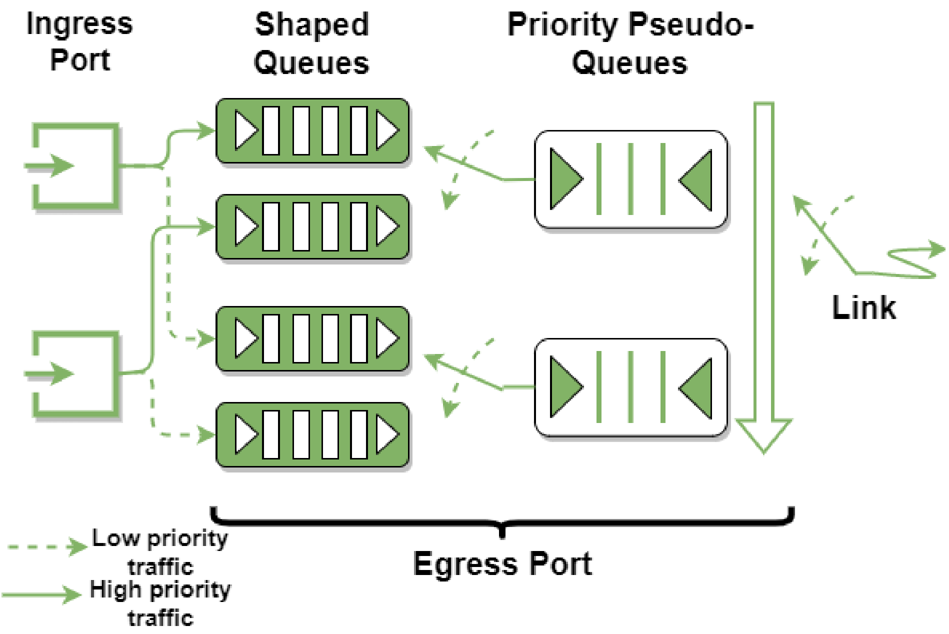


Figure 2.19: Architecture and operation of the UBS [13]

Figure 2.19 shows the UBS architecture and process. In this case, for simplicity of representation only one TSN bridge output port is shown but this same architecture is repeated for all TSN bridge output ports for UBS. The UBS conformer consists of two stages with a series of queues that follow a First Come, First Served (FCFS) discipline: a with queues shaped for interleaved configurations and another with queues depending on priority.

- **Interleaved shaping**: one or more shaped queues are used to regulate the traffic of a given set of flows with QoS requirements. These queues follow a First In, First Out (FIFO) discipline. In order to determine whether the frame can be transmitted or not, only the admission of the Head of Line (HOL) frame is checked. In other words, it is checked whether the HOL frame can be transmitted according to the regulatory restrictions of that flow. If transmission is possible, the frame is released for transmission to the next stage. In certain research, it has been determined that the use of interleaved shaping does not increase worst-case latency for UBS [44, 43]. In addition, UBS supports bucket conformer constraints with escape with the goal of imposing a committed data rate and burst size for each flow. That is, the streams being transmitted are constrained to a higher rate of the form:

$$A_f(t) \leq r_f * t + b_f \tag{2.1}$$

Where A_f corresponds to the amount of data that has been transmitted up to an instant of time t for the flow class f, while r_f is the committed data rate and b_f the committed burst size [44].

- **Priority Queues**: consists of a queue with FCFS discipline for each of the priority levels considered in the scheduler. Each of these queues of a given priority level is responsible for grouping all the outputs of the shaped queues associated with that priority level as indicated by the TC. Thus, a given priority queue P may have traffic coming from different shaped queues. In this case, a strict priority mechanism is used for the transmission of flows from each of the queues. This mechanism assigns a higher preference to those queues that have a higher level of priority so that they can access the shared medium first. In this way, these queues with higher preference access before the queues with a lower level of priority/preference. That is, even though a frame of priority P is ready to be transmitted, it must wait for the transmission of the frames in the queues with a higher priority than P.

A flow's assignment to an ATS scheduler is determined by two factors: the flows handled by the shaped queue and the priority levels associated with the priority queues. Although the selection of shaped queues is determined by three rules primarily [13]:

- Each shaped queue is associated with a single input port (rule *QAR1*).
- Each of the shaped queues is linked to a single priority queue of the previous bridge (rule *QAR2*).
- Each of the shaped queues in turn is linked to a single internal priority level (rule *QAR3*).

Thanks to the use of these rules, asynchronous TSN can achieve their benefits. In particular, QoS determinants are provided by the *QAR2* and *QAR3* rules, while the *QAR1* rule prevents non-conformed traffic from congesting the network, as it allows the isolation of network nodes. Also, the number of shaped queues that are necessary in the network to bind a queue/priority level can be calculated from these rules. For example, if an ATS with a number of ingress ports B is known and at each ingress port, determined as $b \in [1, B]$, traffic associated with a priority level, P_b, determined by the contiguous port, is received. Consequently, the shaped queues that are necessary for proper operation with P internal priority levels at the priority queuing stage and without having any priority assignment constraints are at least $S = P * \sum_{b=1}^{B} P_b$.

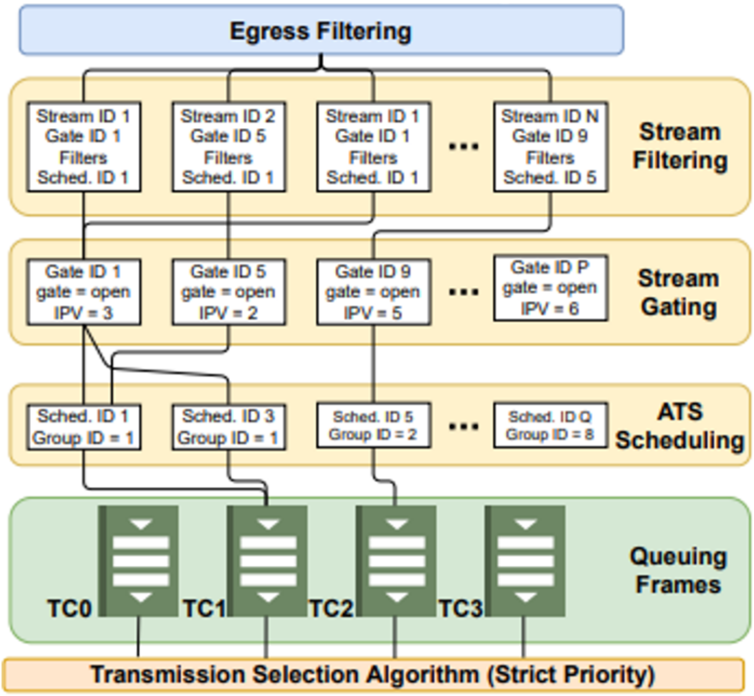

Egress Filtering
Stream ID 1 Gate ID 1 Filters Sched. ID 1
Stream ID 2 Gate ID 5 Filters Sched. ID 1
Stream ID 1 Gate ID 1 Filters Sched. ID 1
...
Stream ID N Gate ID 9 Filters Sched. ID 5
Stream Filtering
Gate ID 1 gate = open IPV = 3
Gate ID 5 gate = open IPV = 2
Gate ID 9 gate = open IPV = 5
...
Gate ID P gate = open IPV = 6
Stream Gating
Sched. ID 1 Group ID = 1
Sched. ID 3 Group ID = 1
Sched. ID 5 Group ID = 2
...
Sched. ID Q Group ID = 8
ATS Scheduling
TC0
TC1
TC2
TC3
Queuing Frames
Transmission Selection Algorithm (Strict Priority)

Figure 2.20: Architecture and operation of the ATS scheduler [13]

The global structure used by an ATS is detailed below. In Figure 2.20 this structure can be observed. In it, first the assignment of the flows to the correct ATS schedulers takes place and then the two stages of the UBS previously explained are implemented. Therefore, the procedure performed by the ATS scheduler consists of four phases, where the first three phases correspond to the first stage of the UBS and the last phase corresponds to the second stage of the UBS. A brief description of each of the phases is given below:

- **Stream filtering**: the classification and filtering of the incoming flow according to its QoS is carried out. Each of the incoming flows is assigned to a certain filter according to the priority level previously assigned according to its PCP and connection identifier. On certain occasions it may happen that some of the incoming frames exceed the maximum size of Service Data Unit (SDU), then the filters may block that frame to avoid overload problems. In addition, at this stage, each flow is also assigned the ATS scheduler associated with that filter and which gate the flow should take for its next stage.

- **Stream Gating**: the assignment of the Internal Priority Value (IPV) takes place in such a way that the real priority that this flow has as determined by its PCP for the gluing and transmission processes is cancelled. The objective of this assignment is to facilitate the fulfillment of the E2E requirements that are demanded by the flows. This is achieved by allowing a higher priority to be given to those flows that demand higher delay requirements.

- **ATS scheduling**: In this phase, the procedure described in the first stage of the UBS is executed. In other words, the flows are accumulated in the queues until the HOL frame can be transmitted in the next stage, complying with the flow regulation restrictions.

- **Queuing and Transmission**: In this stage, the distribution of the outgoing flows of each of the queues conformed to the buffers of this stage takes place. The allocation of the flows to each buffer depends on the associated TC. Therefore, in this stage the procedure described in the second stage of the UBS is carried out. To achieve this objective, it is necessary to have a table in each of the output ports that maps the priority levels of the flows to the traffic class in order to assign the flows to the correct TC queues. If the situation occurs where IPV mapping is enabled, then the mapping of flows to TC is done with this value instead of the actual frame priority level (PCP). Finally, once the flow is in the queues it is transmitted to the shared medium using a strict priority scheduler.

2.4.2 State of the art of prioritization scheduler

This section reviews existing work and projects to date on proposed solutions for flow prioritization in an asynchronous TSN network [22, 23, 14, 13, 24].

In [22], Specht and Sammii explore a Satisfiability Modulo Theories (SMT) solver to determine feasible configurations in ATS-based networks. To deal with the excessive complexity of the pure SMT solution, they suggest a Topology Rank Solver (TRS) heuristic. Nonetheless, TRS uses SMT for flow prioritization in at least one ATS instance.

Prados *et al.* present in [14] a solution that combines heuristic and convex optimization to find a long-term configuration of an ATS-based TSN network. Specifically, the approach in [14] addresses the problem outlined in [24], which seeks to reduce the likelihood of flow rejection. Although the cited works employ heuristic ways to deal with computational complexity, they employ an accurate optimization method to solve flow priority in the ATS instance, limiting their scalability as the number of flows increases.

In [23, 13], Prados *et al.* propose an online technique based on Deep Reinforcement Learning (DRL) to determine the configuration of each flow as it arrives at the network. The flow needs in this solution are uncertain and exhibit a low capability, i.e., they depend on the network topology and must be taught explicitly for each situation, resulting in a lengthy training period. Furthermore, these works lack a model of the flow allocation problem in asynchronous TSN networks.

2.5 Integration of 5G and TSN

Despite all the improvements that have been introduced in the development of 5G to meet the high reliability and delay requirements demanded by the three types of services defined in the standard, the requirements demanded by the URLLC service, mainly at the radio interface, are still not met. Above all, the delays required in Industry 4.0 and neither is delay control allowed in the same way as it is done with TSN. For this, it is necessary for 5G to be integrated with other technology to meet these requirements. Currently, TSN technology is the basic pillar for the main transformation. Therefore, the integration of both technologies, TSN and 5G, is a possible solution to meet all the requirements demanded by the industry.

3GPP in Release 16 [45] has proposed an integration solution between the 5G RAN and synchronous TSN. In this way, it enables support for industrial processes and automation, allowing greater flexibility, while reducing the cost of cabling. The 5G-ACIA in [2] clarifies the architecture proposed by 3GPP in Release 16. In this architecture, the 5G network is considered to act as a virtual switch of the TSN network.

Further updates to Release 16 [46] have included another integration proposal where the TSN acts as a transport network for the 5G network. Each of the proposed solutions is detailed below.

2.5.1 5G network as virtual switch

In this case the 5G system acts as a virtual bridge to the synchronous TSN network, providing connectivity at the Control Plane (CP) and the TSN ports at the User Plane (UP). In this way, the 5G system can connect to one or more synchronous TSN bridges with end devices transparently to the TSN network. That is, appearing to be another TSN bridge. This is achieved by incorporating TSN translators at the edge of the 5G system in order to enable the connection to the synchronous TSN network, the forwarding of data traffic in the UP and the configuration of the TSN bridges in the CP. Three TSN translators have been added to the network:

- *TSN Application Function (TSN AF)*: translator in the control plane that communicates the CNC of the TSN network with the AF application function of the 5G network. The objective of this communication is the establishment of the 5G logical bridge with the QoS requirements that are demanded by each flow and the configuration of the capabilities.
- *Device-side TT (DS-TT)*: translator that is located on the UE side. This translator can be deployed individually or as a set in the same UE.
- *Network-side TT (NW-TT)*: translator localized on the UPF side.

Figure 2.21 shows the detailed integration architecture of the 5G System (5GS) TSN logic bridge with TSN bridges both control plane and user plane.

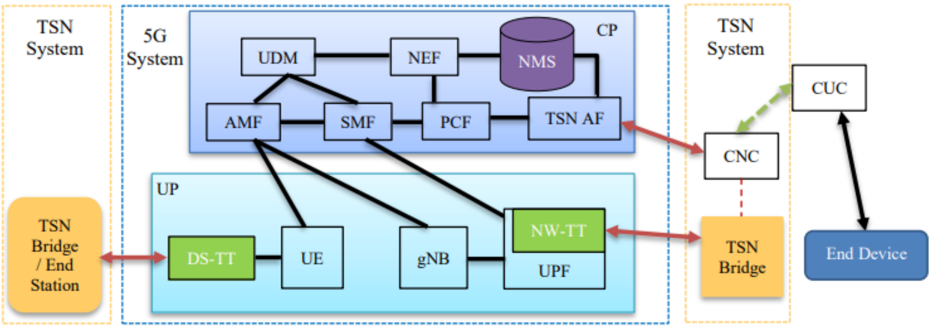


Figure 2.21: TSN and 5G integration architecture

In the UP, DS-TT and NW-TT act as input and output Ethernet ports on the 5GS TSN logic bridge. The functionality of these translators is to translate all the parameters needed to coordinate the communication between the 5G system and the TSN network. That is, they are responsible for mapping the requirements of the TSN flows to the requirements of the 5G system as set by the TSN AF. Each of the DS-TT ports has a 5G system-specific Packet Data Unit (PDU) session associated with it, while each of the NW-TT ports is associated with a physical port in the UPF. In addition, different TSN virtual bridges may be created within the same 5G system as shown in Figure 2.22.

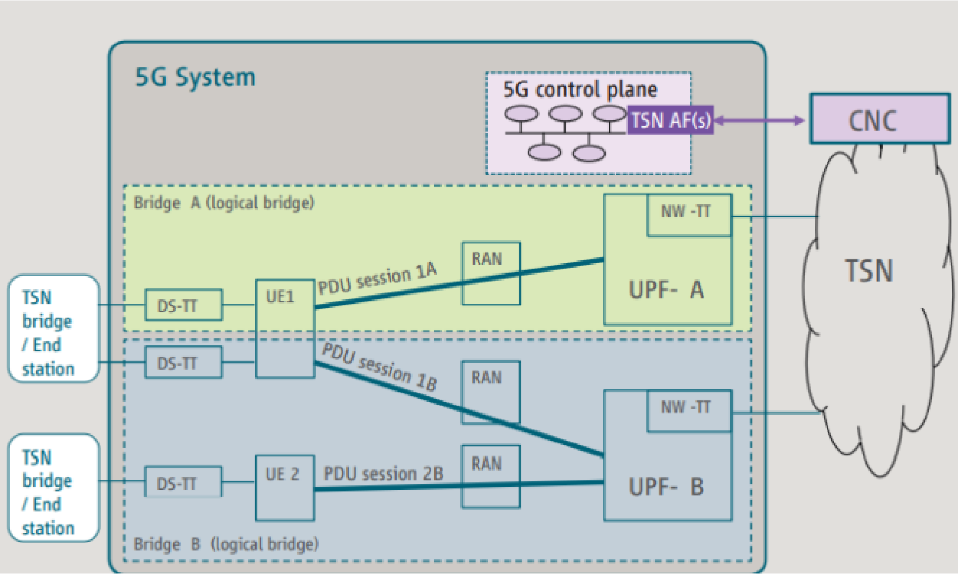


Figure 2.22: TSN-5G integration architecture with more than one TSN virtual bridge [12]

This is possible because a virtual bridge is mainly formed by a UPF and all the UEs connected to it, and in a 5G system since there can be several UPFs, several virtual bridges can be created. All PDU sessions that have been formed on DS-TT ports and are connected to a given UPF constitute a group and belong to the same virtual bridge. Therefore, multiple PDU sessions to different UPF can be created from one UE in order to create redundant transmissions or to isolate traffic. In this way, the UE presenting multiple PDU sessions to several UPF is shared among several virtual bridges, i.e., it would be shared with the same number of virtual bridges as different UPF to which each session is connected. Each of the DS-TT ports assigned to a PDU session belongs to a virtual bridge.

On the other hand, in the CP, the exchange of control data between the 5GS TSN logical bridge and the TSN bridges is established. This is because the CP is responsible for connection management, QoS policies, authentication and other management-related functions. The information exchanged between the different translators and the TSN AF allows to establish a correct configuration of the 5GS TSN logical bridge according to the QoS restrictions of the flows.

2.5.2 TSN network as 5G transport network

In this case, it is the TSN network that acts as the transport network for a 5G system. It can only be realized when the TSN network implements the fully centralized configuration model (IEEE 802.1Q [47]). In this case, the transport network Transport Network (TN) TSN is realized between the N3 interface, i.e., the interface connecting the RAN and the UPF, as seen in Figure 2.8. Therefore, the RAN and the UPF act in this case as end stations of a TSN TN network. This integration architecture can be seen in Figure 2.23.

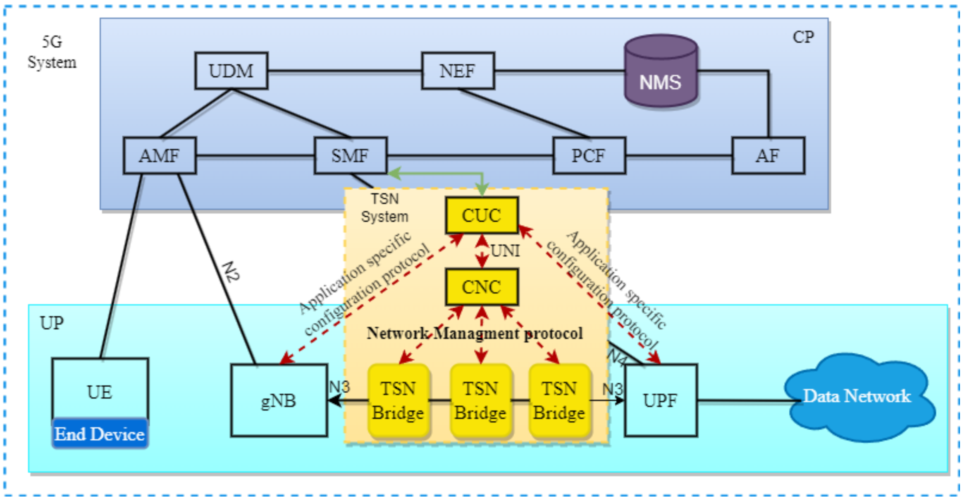


Figure 2.23: TSN architecture as TN of a 5G network

In this situation the CUC is co-located with the SMF, interacting with the CNC in Transport Network (TN CNC). The functionality of the SMF/ CUC is to provide the flow requirements according to the QoS of the flows, i.e. the translation of information from the Listener and Talker groups. This translated information is transmitted through the UNI to the TN CNC. The TN CNC receives this information and uses it to configure the routes and the planning of the transport network. Depending on the result obtained, the TN CNC returns to the SMF/CUC the status group containing the configuration of the communication to the end station. That is, information such as the stream identifier (*StreamID*), the accumulated latency (*AccumulatedLatency*), status information (*StatusInfo*) and interface configuration information (*InterfaceConfiguration*), the last one is optional.

If the Next Generation Radio Access Network (NG-RAN) and the UPF implement TSN Talker and Listener functionality (i.e. Access Network

Talker Listener (AN-TL) and Core Network Talker Listener (CN-TL), respectively) the SMF/CUC can communicate with them through a certain container (*TL-Container*). This message, *TL-Container*, carries between the SMF/CUC and AN-TL and CN-TL different data that have been defined in IEEE P802.1Qdj [48]. Some of these data are for example the *Status group* to the TN CNC containing the list of the path configuration interfaces, which is transmitted by the SMF/CUC. Also the *TL-Container* can be used by the AN-TL or *CN-TL* to transmit information about the list of interfaces it has associated with an identifier (*InterfaceID*), the capacities of the interfaces or the maximum supported buffer duration.

In addition, if the AN-TL and CN-TL functionality is implemented, respectively, they can perform the following functions:

- Retention and storage functionality: The data is stored in the AN-TL or CN-TL buffer if the Time-Sensitive Communication Assistance Information (TSCAI) indicates a certain Burst Arrival Time (BAT) in both the uplink and downlink at which time the burst should be transmitted.
- Flow transformation support: allows you to modify the flows for the respective exchange of information with the SMF/CUC.
- Transmission capabilities of interfaces and end station characteristics: transmits the interface capabilities (*InterfaceCapabilities*) and/or the characteristics of the interfaces connecting to the AN-TL or CN-TL end stations (*EndStationInterfaces*) to the SMF/CUC for their knowledge.
- Topology exchange functionality: allows network topology information to be transmitted to the SMF/CUC via LLDP on the TN.

On the other hand, with respect to flow requirements, the Core Network (CN) Packet Delay Budget (PDB) values (i.e., the maximum latency within the system in the control plane only) for critical flows was previously configured in the CNC in this case is currently configured in the SMF. Therefore, when a new QoS flow is configured, the SMF sends the dynamic value of the CN PDB and TSCAI for that flow to the NG-RAN.

In [13] they use synchronous TSN technology for the transport network instead of asynchronous TSN . However it has been found that the use of asynchronous TSN provides numerous benefits to the network. Some of the improvements are decreased network complexity and increased network scalability and flexibility. Therefore, the problem studied in this project is considered to be a potential solution for this integration architecture. This is possible since it eliminates a large part of the integration problem with synchronous TSN as occurs with the architecture explained in section 2.5.1. This occurs because the use of asynchronous TSN is considered, where the problem of time synchronization of each of the nodes of the network is eliminated and where the use of virtual functions is allowed, characteristics that cannot be taken into account in synchronous TSN.

Chapter 3

Problem definition and solution design

In this section, first we describe the system model for the context of the problem in section 3.1. Then, we formally formulate the flow allocation problem addressed in this work in section 3.2, where firstly we describe the formulation and the approach that is adapted to solve it. This approach consists in the decomposition of the main problem into two sub-problems. The first sub-problem consists of the distribution of the delay of a flow between the different ATSs instances detailed in section 3.3. The second sub-problem focuses on developing a flow prioritization algorithm in an ATS instance detailed in 3.4. For this second sub-problem we formally formulate the ATS flow prioritization in an ATS in subsection 3.4.1 and determine the design principles in subsection 3.4.2. Finally, we detail the implementation of the flow prioritization algorithm in subsection 3.4.3 and the correctness of the algorithm is explained in subsection 3.4.4.

3.1 System Model

Let us consider an asynchronous TSN network composed of a set of ATS-based TSN bridge. The network consists of Z TSN bridges that interconnect with each other through A simplex links. Each TSN bridge c contains I^c input ports and O^c output ports. An ATS is implemented on each of the output ports of the TSN bridge, the operation of the ATS has been explained in subsection 2.4.1. Therefore, each output port is responsible for handling the packet transmission at appropriate link. So, it is assumed that we have a link for each ATS instance in the network, so there is an A ATSs instances. Consequently, the network can be modeled as a directed graph (or digraph) denoted as $G = (\mathcal{Z}, \mathcal{A})$, where $\mathcal{Z}$ denotes the vertices (TSN bridges) and $\mathcal{A}$

denotes the edges (links or ATSs instances) of the graph. Therefore, $\mathcal{A}$ is the set of ATS instance or links while A is the number of ATSs instances ($A =\mid \mathcal{A} \mid$). Each edge $a \in \mathcal{A}$ has a weight C_a that represents the link capacity. The set of TSN switches directly connecting to the endpoints (sources) and receivers (destinations) is considered to be denoted as $\mathcal{V} \subseteq \mathcal{Z}$, i.e., the set of access bridges. It is considered that there exists a set of paths that are predifined, determined by $P_{s,d}$ interconnecting each source $s \in \mathcal{V}$ and destination $d \in \mathcal{V}$. Therefore, a path denoted as $w \in P_{s,d}$ encompasses a sequence of adjacent links, i.e., links that are connected by a TSN bridge. In turn, each path w is composed of $\mathcal{E}$ ATSs instances or links that belong to all TSN bridges ($\mathcal{E} \in \mathcal{Z}$). Each ATS instance e is one of the ATSs instances that conform the total set of ATSs instances ($\mathcal{E}$) of the selected path w. E indicates the number of ATSs instances that conform the set $\mathcal{E}$ of ATSs instances of the path w ($E =\mid \mathcal{E} \mid$).

In each ATS instance e is include P^e priority levels/queues and enough S^e shaped queues as to use all the priority levels regardless of the asynchronous TSN network configuration (e.g., number of ingress ports at the respective TSN bridge and prioritization considered in the previous hop). Without sacrificing generality, we suppose that each priority level is associated with an integer index p and lower indexes denote higher priority levels. So, the priority 1 is the level with the highest priority and the priority P is the level with the lowest priority. In this cases, we consider that the priority level 1 to $P-1$ are reserved to be accommodate the delay-sensitive traffic, whereas the priority level P is reserved to be accommodate the best-effort traffic, like remote access and maintenance in manufacturing [49]. Thanks to the definition of the IPV parameter in the ATS scheduler, as explained in section 2.4.1, it is possible to assign a priority level to each IPV in order to allow flow prioritization by overwriting the PCP priority assigned to each type of traffic.

In the network there are a set of delay-sensitive flows that must be conveyed, where each traffic flow, regardless of its traffic type, is constrained at the top by:

$$r \cdot t + b \ \ [50] \tag{3.1}$$

where r and b are the committed data rate and burst size (burstiness), respectively. This set of delay-sensitive flows $\mathcal{F}$ must be prioritized at each of the ATSs instances that conform the given path of each flow. We assume that there is a maximum E2E delay requirement, denoted as D_f, for each flow $f \in \mathcal{F}$ for all ATSs instances of path w.

The amount of flows, their associated traffic features r and b, and the E2E delay requisite are all known before the scheduling process. This is a regular occurrence in industrial networks, i.e. the characteristics of the flows are known before the operation of the network. For instance, we might have critical flows to communicate alarm events, control the motion of the operational technology devices, configure and diagnose problems with industrial devices, and control the industrial network.

Every traffic flow f_e in the network experiences a maximum delay experienced, D_{f_e}, when passing through an ATS instance e. This maximum delay experienced was derived by Specht and Samii in [43]. This delay D_{f_e} is given by:

$$D_{f_e} = \frac{\sum_{\forall f_e \in \mathcal{F}_{1_e} \cup ... \cup \mathcal{F}_{p_e}} b_{f_e} + \max_{\forall f_e \in \mathcal{F}_{p+1_e} \cup ... \cup \mathcal{F}_{8_e}} l_{f_e}}{C_e - \sum_{\forall f_e \in \mathcal{F}_{1_e} \cup ... \cup \mathcal{F}_{p-1_e}} r_{f_e}} + \frac{l_{f_e}}{C_e} \tag{3.2}$$

where r_{f_e} and b_{f_e} are the committed data rate and committed burst size (burstiness) by the set of flows, $\mathcal{F}_{1_e} \cup ... \cup \mathcal{F}_{p_e}$, with priority level higher or equal to flow f_e in the ATS instance $e \in \mathcal{E}$, respectively. The l_{f_e} is the maximum packet size of the flow f in the ATS instance e , $\max_{\forall f_e \in \mathcal{F}_{p+1_e} \cup ... \cup \mathcal{F}_{8_e}} l_{f_e}$ represent the maximum packet size for the set of flows with priority levels lower than flow f ($\mathcal{F}_{p+1_e} \cup ... \cup \mathcal{F}_{8_e}$) in the ATS instance $e \in \mathcal{E}$ and C_e is the capacity of the link e connecting to ATS instance e.

The maximum delay experienced, D_{f_e}, by each flow in the ATS instance e only considers the transmission delay and the queuing delay. However, it is necessary to consider the delay from the input port to the output port of the switch, i.e., the processing delay in the switch and the flow propagation delay. For its consideration, two variables have been included in formula 3.2: t_{proc} that determines the processing delay, and t_{prop} that determines the propagation delay. Both variables are constants. Therefore, the new formula for the maximum delay experienced, D_{f_e}, is:

$$D_{f_e} = \frac{\sum_{\forall f_e \in \mathcal{F}_{1_e} \cup ... \cup \mathcal{F}_{p_e}} b_{f_e} + \max_{\forall f_e \in \mathcal{F}_{p+1_e} \cup ... \cup \mathcal{F}_{8_e}} l_{f_e}}{C_e - \sum_{\forall f_e \in \mathcal{F}_{1_e} \cup ... \cup \mathcal{F}_{p-1_e}} r_{f_e}} + \frac{l_{f_e}}{C_e} + t_{proc} + t_{prop} \tag{3.3}$$

The sum of all the maximum delay experienced, D_{f_e} by a flow f when it traverses all the ATSs instances $e \in \mathcal{E}$ that conform the predefined path w determine the maximum delay E2E, D_f:

$$D_f \leq \sum_{\forall e \in \mathcal{E}} D_{f_e} \tag{3.4}$$

In this project, each flow f has a source (s_f) and a destination (d_f) specified, and there may exist a large number of possible paths (P_{s_f,d_f}) that connect the source to the destination. For our study, we assume that the path $w_f \in P_{s_f,d_f}$ is precomputed, as the solution we propose is agnostic to the path selection criteria. This selected path $w_f = w$ is considered to be predefined in advance.

Please, you can find the rest of the primary notations considered in the prioritization problem in Table 3.1.

3.2 Flow Allocation Problem

Based on the analysis showed in subsection 2.4.2, it has been realized that various options exist for configuring ATS-based TSN networks. However, all of these options exhibit scalability issues, necessitating the identification of solutions that can effectively handle and adapt to increased traffic without compromising the offered QoS. Therefore, scalability plays a crucial role in appropriately adapting to the growing evolution of new services, which come with more stringent QoS requirements. Furthermore, the proposed solutions, due to their inherent nature, entail significant computational complexity during implementation.

In this project, we propose a novel approach to address the aforementioned challenges by introducing a method for configuring ATS-based TSN networks. Our approach minimizes the number of priority levels in comparison to existing techniques while satisfying deterministic QoS requirements. The proposed solution aims to achieve a feasible prioritization of a set of traffic flows across multiple ATSs instances while meeting the specified delay requirements.

Our solution exhibits excellent scalability, efficiently accommodating an increasing number of flows. For a large volume of flows, the proposed solution endeavors to determine the feasible prioritization in an efficient manner. It is important to note that our solution assumes prior knowledge of the flow requirements, making it particularly suitable for industrial networks where traffic types are known in advance. Additionally, the algorithm used to implement our proposed solution offers a feasible and straight-forward approach, applicable to both online and offline scenarios.

The subsequent subsection provides a formal formulation of the overarching problem a TSN network featuring multiple ATSs instances and a determined number of flows to be resolved.

3.2.1 Overall Problem Formulation

This subsection details the problem to be solved, providing the formal formulation of the problem and the necessary considerations that must be known beforehand.

Notation	**Description**
$\mathcal{F}$	Set of flows to be prioritized in the all ATSs instances.
$\mathcal{F}_e$	Set of flows to be prioritized in the ATSs instance e.
$\mathcal{F}_{p_e}$	Set including all the flows currently allocated to priority level p in the ATS instance e .
E	Set of ATSs instances that conform the predefined path of the flow f.
C_e	Nominal link capacity of the link e connecting to ATS instance e.
r_f	Data rate of the flow f.
b_f	Burst size of the flow f.
l_f	Maximum frame size of the flow f.
D_f	E2E delay requisites for the flow f at the all ATSs instances.
D_{f_e}	Delay requisites for the flow f at the ATS instance e.
R_{f_e}	WCQD requisites for the flow f at the ATS instance e, being $R_{f_e} = D_{f_e} - \frac{l_f}{C_a}$.
Q_{p_e}	WCQD of priority level p at the ATS instance e.
Q_f	WCQD experienced by flow f at the all ATS instance.
Q_{f_e}	WCQD experienced by flow f at the ATS instance e.
$R_{\mathcal{F}_e}$	The most stringent WCQD requisites among all the flows $\mathcal{F}$ in the ATS instance e.
$R_{\mathcal{F}_p}$	The most stringent WCQD requisite among all the flows $\mathcal{F}_p$ for the priority level p.
$R_{\mathcal{C}_e}$	The most lenient WCQD requisites among all the flows $\mathcal{F}$ in the ATS instance e.
$R_{\mathcal{C}_p}$	The most lenient WCQD requisites among all the flows $\mathcal{F}_p$ for the priority level p.
P_e	Maximum number of priority levels (queues) available (implemented) in the ATS instance e.
t_{proc}	Proccessing delay.
t_{prop}	Propagation delay.

Table 3.1: Primary notation

The main problem addressed in this project is the prioritization of deterministic traffic in delay-sensitive networks. Therefore, the problem is to find a feasible or satisfiable prioritization for $\mathcal{F}$ flows in an asyncrhonous TSN network , i.e., the E2E delay requisite, D_f, $\forall f \in \mathcal{F}_e$ in all ATSs instances are met, to minimize the number of priority levels used in it. The following is the formal formulation of the stated problem:

$$\begin{aligned}
\mathit{minimize} \quad & \left\{ \max_{\forall f \in \mathcal{F}_e,\, \forall e \in \mathcal{E}} P_f \right\} \\
\text{s.t.} \quad & P_f \in [1, P_e - 1] \cap \mathbb{N} \;\; \forall f \in \mathcal{F}_e, \forall e \in E \; (C1); \\
& D_f \leq \sum_{\forall e \in E} D_{f_e} \; (C2); \\
& Q_f \leq R_f \;\; \forall f \in \mathcal{F}_e, \; \forall e \in E \; (C3); \\
& \sum_{\forall\, f \in \mathcal{F}_e} r_f \leq C_a \; \forall e \in E \; (C4).
\end{aligned} \tag{3.5}$$

where $\mathbb{N}$ is the set of natural numbers. P_f is the decision variable of the problem which denotes the priority level assigned to flow $f \in \mathcal{F}_e$ in all ATSs instances $e \in E$. This variable is integer and take values in the available levels for delay-sensitive traffic in the all ATSs instances, as specified in constraint *C1*. D_{f_e} is another decision variable that determines the delay requirement that the flow f can suffer at most in the ATS instance e of the selected path w_f, as specified in contraint *C2*.

Clarify that the prioritization is carried out by flow and by each ATS that makes up the asynchronous TSN network. The prioritization of the flows for each ATS instance is possible thanks to the IPV variable defined in the standard, allows solving the priority prioritization problem instead of the complete ATS network for each of the ATS that conform the TSN network.

This variable allows the prioritization problem of the complete ATS network to be reduced and resolved into a prioritization problem in each of the ATSs instances that make up this network. That is, in each of the ATSs instances, the real frame priority (PCP) level initially assigned to a new priority level can be modified to meet the delay requirement of each of the flows, as explained in the subsection 2.4.1.

The objective of the main problem (3.5) is to minimize the required number of flow priority levels that exist in a network for each of the ATS that conform the flow's predefined path. The motivation for choosing this optimization objective is that the cost of the asynchronous TSN network directly depends on the priority levels available in the ATSs instances. That is, the greater the number of priorities available each ATS, the greater the implementation costs (Capital Expenditure (CAPEX)) as the price of the ATS-based TSN bridge increases. Also, it is easier to configure and operate an asynchronous TSN network whose ATSs instances have fewer priority levels.

Regarding the main constraints, we must ensure that the aggregate rate of all flows through each ATS instance of all ATSs instances of is less than the nominal capacity of the links connecting the ATS instance (*C4*). In fact, this technological constraint is a primary assumption to derive (3.9) [43, 51]. On the other side, the delay requisites, Worst Case Queuing Delay (WCQD) requisites for all the flows in the all ATSs instances has to be met (*C3*).

3.2.2 Problem Design Partitioning

This section presents the approach that has been adapted to solve the problem of deterministic flow allocation in an asynchronous TSN network. But before we indicate the considerations that we have taken into account.

In the asynchronous TSN network, we consider that the different flows $\mathcal{F}$ are grouped into different type of services according to the QoS that are required. We assume that the characteristics of the flows are known a priori (e.g. data rate, burstiness, delay requisite and frame size). Knowledge of these characteristics is a common situation in many scenarios such as Industry 4.0.

One of these features is the E2E delay requirement, D_f, of each of the flows $f \in \mathcal{F}$. However, the flow must pass through different ATSs instances to reach the destination d_f of that flow. In order to carry out the flow scheduling in each of the ATSs instances that conform the TSN network considered, it is necessary to know what is the delay requirement of each of the flows passing through each of the ATSs instances. Therefore, it is necessary to determine a procedure to determine what is the delay requirement that the flow can suffer in each ATS instance (D_{f_E}) that conforms the predefined path w.

Therefore, as comment in the subsection 3.2.1, the two decision variables of the problem, D_{f_e} and P_f, are interrelated. Hence, finding the optimal solution to the optimization problem (3.5) requires jointly optimizing both decision variables. In order to solve the problem the approach that has been adapted is the division of the problem in two, since by separating the overall problem into two problems it is possible to solve it more easily and reach a feasible solution. This approach treated is independent of the formulation of the overall problem.

Each of the sub-problems addresses one of the essential aspects of the global problem. The question to be solved in each of the sub-problems is detailed below, both cases considering that the route to be followed by the different flows is predefined beforehand:

- **$1^{\underline{o}}$ Sub-Problem (*Delay requirement distribution*)**: This sub-problem addresses the distribution of the delay requirement of the flows among the different ATSs instances that make up the TSN network. That is, the first decision variable D_{f_e}, is solved while the second decision variable (the allocation of P_f) is omitted in this sub-problem. For this purpose, we consider that the delay requirements of E2E are known in advance and the route is precomputed. In addition, the traffic prioritization problem (P_f allocation) is considered to be solved.
- **$2^{\underline{o}}$ Sub-Problem (*Per ATS Flow Prioritization Algorithm*)**: This sub-problem addresses the problem of prioritizing of the flows present in a single ATS instance of a TSN network, i.e., the decision variable P_f is solved for each flow. Therefore, it is assumed that the variables D_{f_e} are fixed and equal to the result obtained in the first sub-problem. In addition, it is necessary to know the number of flows it is necessary to know the number of flows that pass through an ATS instance a priori.

Despite dealing first with sub-problem 1 on the distribution of the E2E delay requirement, the main contribution of this project originates from sub-problem 2, i.e., the per ATS flow prioritization algorithm.

The following sections will address the resolution of both sub-problems. First, section 3.3 solves the first sub-problem, i.e., the distribution of the maximum delay requirement among the ATSs instances that conform the predefined path of a flow. Finally, section 3.4 addresses the second sub-problem, i.e., the prioritization of deterministic traffic over a TSN as an industrial network.

3.3 Delay requirement distribution

This section addresses the first sub-problem, i.e., the distribution of the E2E delay requirement, D_{f_e}, of the various flows $\mathcal{F}$ among the e ATSs instances that conform the predefined path, $w \in P_{s,d}$, between a source s and recipient d. Next, the formal formulation of the $1^{\underline{o}}$ sub-problem is detailed and finally, the solution that has been addressed for its implementation is explained.

3.3.1 Problem Formulation

In this sub-problem it is considered that the prioritization of the ATSs instances is correct and it is not necessary to have to address their resolution.

Furthermore, it is considered that the characteristics of the different flows transmitted in the network are known in advance and the routing of all flows is predefined as explained in section 3.1. Some of these characteristics are for example the data rate (r_f), the burstiness (b_f), the frame size (l_f), the maximum E2E delay requirement (D_f) of the flow f.

The objective of this sub-problem is the distribution of the delay requirement E2E (D_f) of a given flow f among the ATSs instances $e \in E$ composing the predefined path $w \in P_{s,d}$ for flow f with source s and destination d. Below we present the formal formulation of the problem presented, i.e., we address the resolution of constraint *C2*. This upper bound of the maximum E2E delay experienced by any packet is [43, 44]. The optimization criterion, therefore, remains the same as the global problem for this sub-problem, i.e., the one indicated in section 3.2.1.

$$D_f \leq \sum_{e \in E} D_{f_e} \ (C2) \tag{3.6}$$

where D_{f_e} determines the maximum delay requirement that the flow f in the ATS instance e of the predefined path p can suffer. It must be ensured that the sum of all delay requirements for each ATS instance e is lower than the E2E delay requirement for all ATSs instances for flow f.

3.3.2 Solution Design

To be able to carry out the distribution of the delay among the different ATSs instances that conform the flow, the solution proposed in [14] has been chosen. That is, in Next Generation Transport Network Optimizer (NEPTUNO) addresses this problem and determines a method of allocating the delay requirement in each of the ATSs instances.

In their publication [14], Prados, Taleb, and Bagaa introduce NEPTUNO, an online solution designed to address flow allocation challenges in 5G backhaul networks. NEPTUNO combines optimization methods with data analytics to optimize the acceptance rate of flows within the network while ensuring the deterministic QoS requirements of critical flows. Although this work employs heuristic methods to cope with the computational complexity, it still uses an exact optimization method to address flow prioritization in the ATS instance, which limits its scalability with the number of flows, i.e., it causes the computational complexity to grow exponentially with the number of traffic flows to be accommodated. Unlike NEPTUNO, our proposal is scalable with the number of flows as indicated in the section 4.4. This shows that the computational time complexity is maintained and furthermore, it provides at the same time a feasible prioritization.

NEPTUNO also addresses the crucial issue of delay distribution among the different TSN bridges along the predefined path. To accomplish this, NEPTUNO explores three distinct approaches for distributing the E2E delay requirements:

- Approach based on link capacities: In this approach, the delay budget allocated to a specific TSN bridge is dependent on the link capacity associated with that particular bridge.
- Approach based on maximum expected link utilization: This approach takes into account the maximum expected aggregate rate of traffic to be served on a given TSN bridge. The total traffic load is distributed among the various TSN bridges accordingly.
- Approach based on maximum expected link utilization and capacities: This approach considers both the maximum expected utilization of the link and the capacities of the links when distributing the E2E delay requirements among the TSN bridges.

Within NEPTUNO, the first approach (based on link capacities) is employed as it yields the most favorable outcomes and benefits. This particular approach is found to be most effective in ensuring the efficient distribution of delay requirements among the TSN bridges along the predefined path. By leveraging this approach, NEPTUNO optimizes the allocation of E2E delay requirements in the network, contributing to improved performance and meeting the stringent QoS demands of time-sensitive applications in the TSN environment.

In this project, for the distribution of the delay requirement, the approximation used by NEPTUNO is used. That is, it is performed according to the capacities presented by the links (C_a) associated to each of the ATSs instances $e \in E$ that conform the predefined path w.

$$D_{f_e} = D_f \cdot \frac{\frac{1}{C_a}}{\sum_{a=1}^{|E|} C_a} \tag{3.7}$$

From the formula 3.7 the maximum delay requirement (D_{f_e}) allowed in each ATS instance e of the predefined path for each flow is obtained. In addition, the sum of all the delay requirements in each ATS instance does not exceed the E2E delay requirement of that flow. In this way, this subproblem is solved and allows to determine the delay requirement in each ATS instance to subsequently carry out the prioritization of the flows in each ATS instance.

3.4 Per ATS Flow Prioritization Algorithm

This section addresses the second sub-problem, i.e., the allocation of the set of flows $\mathcal{F}$ within a single ATS instance of a TSN network. To solve this sub-problem, a new solution has been proposed that by using an ATS-based TSN network configuration method. The method implements the objective of the main problem, i.e., to minimize the number of priority levels in a single ATS instance that meets the deterministic QoS requirements. For this, it is necessary to know the number of flows and their characteristics, as well as the delay requirement of each flow for that particular ATS instance, solved in the previous section 3.3.

For this solution, it is assumed that all flows $\mathcal{F}_e$ already possess a predefined path ($w_f \in P_{s,d}\, \forall f \in \mathcal{F}_e$) between a source TSN bridge s and a destination TSN bridge d, knowing the $\forall e \in E$ ATSs instances, which are located at each of the TSN bridge's output ports, through which the f flow traverses. Also, the delay requirement, D_{f_e}, that a given flow f may suffer at an ATS instance e is known beforehand. That is, this variable is computed in sub-problem 1 in the section 3.3.

We can define the WCQD requirement, R_{f_e}, for flow f at ATS instance e that conform its path w as:

$$R_{f_e} = D_{f_e} - \frac{l_{f_e}}{C_e} \tag{3.8}$$

where l_{f_e} is the maximum frame size of flow f in the ATS instance e and C_e the nominal capacity of ATS instance e. Without loss of generality, we assume that each $f \in \mathcal{F}$ for a given ATS instance e is associated with an integer index i, $R_{i_e} = R_{f_e}$. Specifically, lower indices mean more stringent WCQD requirements, i.e., $R_{i-1_e} \leq R_{i_e} \leq R_{i+1_e}$ with flows with indices 1 and $F =\mid \mathcal{F}_e \mid$ having the most stringent and most lenient WCQD requirements, respectively.

For example, it is considered that there is a set of flows to be transmitted in an ATS instance e, $f_{a_e} = f_a$ through $f_{z_e} = f_z$, as seen in Figure 3.1.

fa
fb
fc
fd
...
fw
fx
fy
fz

Figure 3.1: Set of flows to be prioritized in the ATS instance e

Each of these flows in the ATS instance e has a WCQD requisite determined by formula 3.8. These flows are ordered according to their WCQD requisite from the most stringent to the least. Therefore, in this case, flow $f_{a_e} = f_a$ has the most stringent WCQD requisite while flow $f_{z_e} = f_z$ has the most lenient WCQD requisite, as shown in the Figure 3.2.

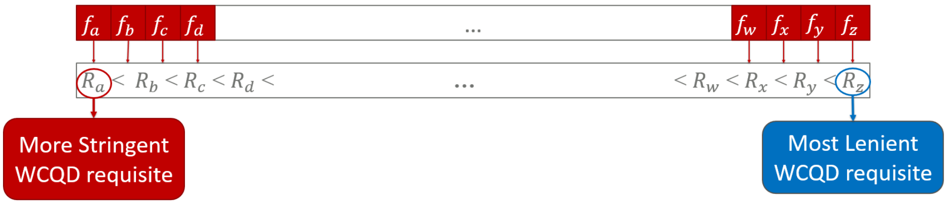

f_a f_b f_c f_d
...
f_w f_x f_y f_z
$R_a < R_b < R_c < R_d <$ ... $< R_w < R_x < R_y < R_z$
More Stringent WCQD requisite
Most Lenient WCQD requisite

Figure 3.2: Flows to be prioritized sorted by WCQD requisite in the ATS instance e

We consider that $f_{\mathcal{F}_p}$ is the flow with the most stringent requirement WCQD for priority level p while $f_{\mathcal{C}_p}$ is the flow with the most lenient requirement WCQD for priority level p.

Let f_{p_e} be the set of flows allocated to a priority level p at ATS instance e. The WCQD Q_{p_e} experienced by each flow assigned to a priority level p for an ATS instance e has the following upper bound [43, 51, 44]:

$$Q_{p_e} = \frac{\sum_{\forall f_e \in \mathcal{F}_{1_e} \cup \ldots \cup \mathcal{F}_{p_e}} b_{f_e} + \max_{\forall f_e \in \mathcal{F}_{p+1_e} \cup \ldots \cup \mathcal{F}_{8_e}} l_{f_e}}{C_e - \sum_{\forall f_e \in \mathcal{F}_{1_e} \cup \ldots \cup \mathcal{F}_{p-1_e}} r_{f_e}} \tag{3.9}$$

where r_{f_e} and b_{f_e} are the data rate and burst size (burstiness) for the flow f at ATS instance e, respectively. The l_{f_e} represents the maximum frame size for flow f at ATS instance e and C_e is the capacity of the link e connecting to ATS instance e.

Once the different flows are sorted the developed method is applied. Next, the formal formulation of the 2º sub-problem to be addressed is detailed, followed by the design principles to be considered, and the algorithm. Finally, the correctness and optimization of the proposed solution is provided.

3.4.1 Per ATS Problem Formulation

This subsection details the formal formulation of sub-problem 2, i.e., the prioritization of a set of flows under deterministic QoS requirement into a single ATS instance. The sub-problem 2 consists in finding a feasible or satisfiable prioritization for $\mathcal{F}_e$ at the respective ATS instance, i.e., the delay requisites $\forall f \in \mathcal{F}_e$ are met, to minimize the number of priority levels used in it. For simplicity, all variables used hereafter are considered for a single ATS, i.e., we simplify $\mathcal{F}_e = \mathcal{F}$. Below is the formal formulation of the stated problem:

$$\begin{aligned} \textit{minimize} \quad & \left\{ \max_{\forall f \in \mathcal{F}} P_f \right\} \\ \text{s.t.} \quad & P_f \in [1, P-1] \cap \mathbb{N} \; \forall f \in \mathcal{F} \, (C1); \\ & Q_p \leq R_f \; \forall \, f \in \mathcal{F}_p, \, p \in [1, P-1] \, (C2); \\ & \sum_{\forall \, f \in \mathcal{F}} r_f \leq C \, (C3). \end{aligned} \tag{3.10}$$

where $\mathbb{N}$ is the set of natural numbers. P_f is the decision variable of the problem which denotes the priority level assigned to flow $f \in \mathcal{F}$. This variable is integer and take values in the available levels for delay-sensitive traffic in the corresponding ATS instance, as specified in constraint *C1*. P is the maximum number of priority level available.

Regarding the primary constraints, we must ensure the aggregated rate traversing the ATS instance is lower than the nominal capacity (*C3*). In fact, this technological constraint is a primary assumption to derive 3.9 [43, 51]. On the other side, the WCQD requisites for all the flows has to be met (*C2*).

The objective of this sub-problem above is to minimize the required number of priority levels in an ATS instance. The motivation of choosing this optimization goal is because the cost of the asynchronous TSN network directly depends on the available priority levels in the ATSs instances. The higher the number of available priorities, indicating the maximum number of queues must have the ATS instance, is the higher the deployment cost (capital expenditures) as the ATS-based TSN bridge's price raises. Moreover, it is easier to configure and operate an asynchronous TSN network whose ATSs instances have lower number of priority levels.

3.4.2 Design Principles

Let us start introducing some relevant propositions that can be directly proven from 3.9 and are behind the rationale of the proposed algorithm. Also, these propositions are cornerstone for assisting the proof of the correctness and degree of optimality of our proposal.

Proposition 1. *The WCQD Q_1 for the first priority level (highest priority) is given by $Q_1 = \sum_{\forall \, f \in \mathcal{F}} b_f / C$ when there is a single priority level or $Q_1 = (\sum_{\forall \, f \in \mathcal{F}_1} b_f - max_{\forall f \in \mathcal{F} \setminus \mathcal{F}_1} l_f)/C$ when there are two or more priority levels. Moreover, Q_1 is the lowest WCQD in the ATS instance, i.e., $Q_1 < Q_p \; \forall \, p \in [2, P]$.*

Proof. Q_1 can be directly derived from 3.9. From 3.9, the aggregated burstiness of any priority level $p \in [2, Q]$ will include the aggregated burstiness of level 1 and by definition $l_f \leq b_f \; \forall f \in \mathcal{F}$. Also, the effective capacity of $p \in [2, Q]$ is reduced by the aggregated committed rate in level 1. Then, it always holds that $Q_1 < Q_p \; \forall p \in [2, P]$. □

Proposition 2. *Decreasing one priority level from p to $p+1$ of any flow f will increase its WCQD, but reduce or does not affect the WCQD of the rest of flows. Equivalently, increasing the priority level of any flow f will decrease its WCQD, but increase or does not affect the WCQD of the rest of flows.*

Proof. From 3.9, lowering the priority level of a flow f will reduce the aggregated burstiness of the priority level p it was originally accommodated by b_f. Since by definition $b_f \geq l_f^{(max)}$, the WCQD of p is reduced. For the new priority level $p+1$ of f, the aggregated burst size will remain the same, but its effective capacity $C - \sum_{k=1}^{p} r^{(k)}$ will increase by r_f, thus, decreasing the WCQD of $p+1$. For priority levels $k > p+1$ or $k < p$, the WCQD does not change. On the other hand, the maximum aggregated burst size seen by f remains the same, but its effective capacity is reduced by $\sum_{f \in \mathcal{F}_p} r_f$, thus increasing its WCQD. Last, increasing the priority level of a flow f is equivalent, in terms of the resulting WCQDs experienced by the flows, to keep the same priority level for f and decrease one priority level for the rest of the flows. □

Proposition 3. *If, in the highest priority level ($p = 1$), the most lenient WCQD requisite $R_{f_{F_1}}$, i.e., $R_{f_{F_1}} \geq R_f \; \forall f \in \mathcal{F}_1$, which is imposed by the flow $f_{F_1} = f_{|\mathcal{F}_1|}$, is not fulfilled, i.e., $Q_1 > R_{f_{F_1}}$, then, problem* (3.10) *has no satisfiable solution.*

Proof. From *Proposition 1*, $Q_1 < Q_p \forall p \in [2, P]$. From *Proposition 2*, decreasing the priority level of f_{F1} will increase its WCQD. On the other hand, decreasing the priority level of any flow $f \in \mathcal{F}_1 \; s.t. \; f \neq f_{F1}$ to reduce the Q_1 is neither possible because the WCQD of f will increase (*Proposition 2*) and from the premises of the proposition $R_{f_{F_1}} \geq R_f$, thus, R_f would not be met. □

Proposition 4. *If currently $Q_p \leq R_f \; \forall f \in \mathcal{F}_p$ and $\forall p \in [2, P]$, $\mathcal{F}_p == \emptyset \; \forall p \in [P, P+1]$, and we decrease 1 priority levels $\forall f \in \mathcal{F} \setminus \mathcal{F}_1$, then, we can freely distribute the flows originally allocated to $p = 1$ among the levels $p \in [1, 2]$ and the requisites of the rest of the flows will be still met, i.e., $Q_p \leq R_f \; \forall f \in \mathcal{F}_p$ and $\forall p \in [3, P+1]$.*

Proof. From 3.9, decreasing 1 priority levels for all the flows originally allocated in $p \in [2, P]$ will keep the same WCQD for them as $\mathcal{F}_p == \emptyset \; \forall p \in$

$[P, P+1]$. Then, no matter the prioritization we consider for the flows originally allocated in $p = 1$ among the levels $p \in [1, 2]$, also from 3.9, the WCQD will remain the same for all the flows originally allocated in $p \in [2, P]$. □

3.4.3 Algorithm

The proposed ATS flow prioritization algorithm is shown in Algorithm 1. The goal of the algorithm is to find a satisfiable prioritization, if at least one exists, for the set of flows $\mathcal{F}$ at a given ATS instance according to the optimization program (3.10). To that end, it iterates (*lines* $7 - 20$) until either a feasible solution is found, i.e., the delay requisites for all the flows are met while the link utilization is lower than 100% (*line* 10), or the problem infeasibility is determined (*line* 17). Please refer to *Propositions 1* and *3* for the rationale behind the latter algorithm exit condition.

At each iteration, first, the algorithm checks whether the WCQD requisites for all the flows allocated to the second priority level $\mathcal{F}_2$ are met (*line* 8). Please note that this condition is always met at the very first iteration as $\mathcal{F}_2$ initially equals the empty set. If the condition is met, which is verified using 3.9, the algorithm checks whether the WCQD requisites for all the flows allocated to $\mathcal{F}_1$ are met. If so, a feasible solution is found (*line* 10) and the algorithm finishes. Otherwise, the algorithm creates a new set $\mathcal{F}_k$ if needed and decreases the priority level for all the flows by one, leaving the set $\mathcal{F}_1$ empty (*lines* $12 - 13$). We refer hereinafter to this process as partition k or kth. The reason to follow the operation described above is that, once the algorithm finds a satisfiable prioritization for the flows allocated to the current priority levels 2 to k, the highest priority level can be further partitioned to find a feasible solution without affecting the WCQDs of the current priority levels 2 to k (*Proposition 4*).

If conditions in *line* 8 or *line* 9 are not met, then, the algorithm moves the flow f^* with the most stringent WCQD requisite currently in priority 2 to priority 1, i.e., it increases the priority of f^* (*lines* $15 - 16$). Nonetheless, if it turns out that f^* is the last flow in $\mathcal{F}_2$, the algorithm realizes that the problem has no solution (*line* 17).

Figure 3.3 shows the process followed by the algorithm to find a feasible prioritization solution by increasing the number of partitions. Specifically, it is observed how the number of partitions increases from 1 to k partitions and how it affects the increase of priorities and the assignment of flows to priority levels. As shown, the WCQD experienced at that priority level must be less than the minimum delay requirement experienced by the strictest flow at that priority level. If it is satisfied then as stated in *Proposition 4* it is not necessary to modify the flow allocation at that priority level and lower levels.

Algorithm 1 ATS Prioritization Algorithm

```
$Problem\_Solved = 1;$                                                      ▷ $BC\_O1$
$No\_Solution = 2;$                                                         ▷ $BC\_O2$
$Searching\_Solution = 3;$                                                  ▷ $BC\_O3$
$Initialize\ \mathcal{F}_1 \leftarrow \mathcal{F};\ \mathcal{F}_2 \leftarrow \emptyset;\ k = 1;$
function $PrioritizeFlows(\mathcal{F}_1,\ \mathcal{F}_2,\ k)$
    $prob\_status = Searching\_Solution;$
    while $prob\_status == Searching\_Solution$ do
        if $Q_2 \leq R_f\ \forall f \in \mathcal{F}_2$ then
            if $Q_1 \leq R_f\ \forall f \in \mathcal{F}_1$ then
                return $Problem\_Solved;$
            end if
            $k++;\ \mathcal{F}_k = \{\};$
            $\mathcal{F}_p \leftarrow \mathcal{F}_{p-1}\,\forall\, p = [2,k];\ \mathcal{F}_1 \leftarrow \emptyset;$
        end if
        $f^* \leftarrow \arg\min_{f \in \mathcal{F}_2} R_f;$
        $\mathcal{F}_2 \leftarrow \mathcal{F}_2 \setminus f^*;\ \mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{f^*\};$
        if $\mathcal{F}_2 == \emptyset$ then
            return $No\_Solution;$
        end if
    end while
end function
```

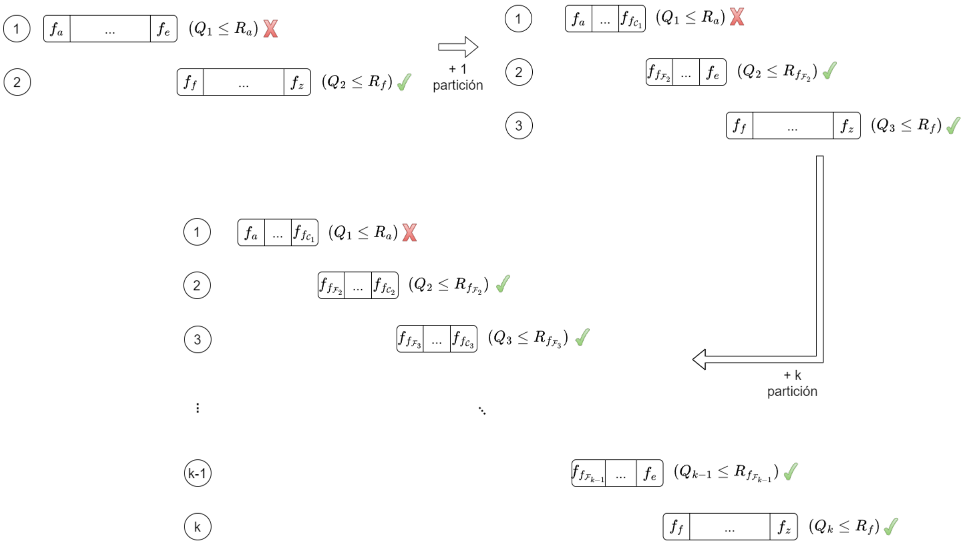

1
f_a ... f_e $(Q_1 \leq R_a)$ ✗
2
f_f ... f_z $(Q_2 \leq R_f)$ ✓
+ 1
partición
1
f_a ... f_{fc_1} $(Q_1 \leq R_a)$ ✗
2
$f_{f_{\mathcal{F}_2}}$ ... f_e $(Q_2 \leq R_{f_{\mathcal{F}_2}})$ ✓
3
f_f ... f_z $(Q_3 \leq R_f)$ ✓
+ k
partición
1
f_a ... f_{fc_1} $(Q_1 \leq R_a)$ ✗
2
$f_{f_{\mathcal{F}_2}}$ ... f_{fc_2} $(Q_2 \leq R_{f_{\mathcal{F}_2}})$ ✓
3
$f_{f_{\mathcal{F}_3}}$ ... f_{fc_3} $(Q_3 \leq R_{f_{\mathcal{F}_3}})$ ✓
k-1
$f_{f_{\mathcal{F}_{k-1}}}$ ... f_e $(Q_{k-1} \leq R_{f_{\mathcal{F}_{k-1}}})$ ✓
k
f_f ... f_z $(Q_k \leq R_f)$ ✓

Figure 3.3: Process from 1 partition to k partitions

3.4.4 Proof of algorithm correctness and optimality

This section includes the proof that Algorithm 1 for ATSs detailed in section 3.4.3 is correct and optimal. More precisely, we rely on the principle of mathematical induction to formally prove the theorem stated next.

Theorem 1. *Algorithm 1 finds always a satisfiable solution for the ATS prioritization problem* (3.10) *if any exists and the solution found is optimal for that problem, i.e., it minimizes the number of priority levels used in the ATS instance.*

Proof. The Algorithm 1 starts by checking *Proposition 1.* If the $Q_1 \leq R_{f_{\mathcal{F}_1}}$ solution is feasible for $P = 1$ (solution BC_O2), the main hypothesis is fulfilled. Otherwise the algorithm provides the solution BC_O3, where a partition is performed at the highest priority level. As previously defined a partition is the process carried out by Algorithm 1 to divide the current highest priority level into two. For the demonstration we will use the induction method based on the fact that at each iteration a partitioning of the flows with the highest priority level is performed. Let k denote the current partitioning index as in Algorithm 1 to find a solution to the prioritization of a set of TSN flows $\mathcal{F}$.

BASE CASE (P=2): In this case, a new partition is performed, i.e., the maximum number of priority levels is changed from $P = 1$ to $P = 2$. Let assume that the set of flows $\mathcal{F}$ are ordered according to their WCQD requirement, as shown in Figure 3.2.

Lemma 1. *The partitioning provided by Algorithm 1 guarantees that the number of flows at* $p = 2$ *is the maximum and meets its WCQD requirements and simultaneously the WCQD experienced at* $p = 1$ *is minimum.*

Proof of Lemma 1.

Proof. Let us assume that we have two cases, *case a* and *case b.* In both cases all flows of priority level $p = 2$ meet their WCQD requirements. But in *case b* the number of flows assigned to $p = 1$ is less than in *case a*, as shown in Figure 3.4.

The WCQD of *case a* for each priority level is calculated according to *Proposition 1* and formula 3.9:

$$Q_1^a = \frac{\sum_{\forall f\in\mathcal{F}_1} b_f + \max_{\forall f\in\mathcal{F}_2} l_f}{C} = \frac{\sum_{f=a}^{f} b_f + \max_{\forall f\in\mathcal{F}_2} l_f}{C}$$

$$Q_2^a = \frac{\sum_{\forall f\in\mathcal{F}_1\cup\mathcal{F}_2} b_f}{C - \sum_{\forall f\in\mathcal{F}_1} r_f} = \frac{\sum_{f=a}^{z} b_f}{C - \sum_{f=a}^{f} r_f}$$

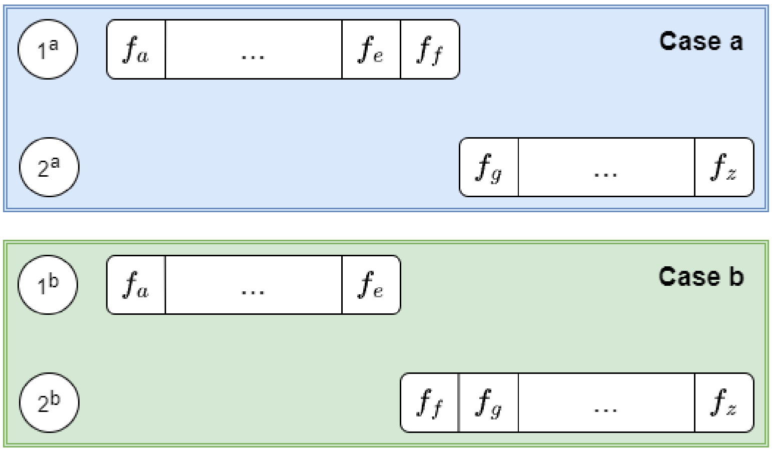


Figure 3.4: Flow allocation *case a* and *case b*

The WCQD of *case b* for each priority level is:

$$Q_1^b = \frac{\sum_{\forall f \in \mathcal{F}_1} b_f + \max_{\forall f \in \mathcal{F}_2} l_f}{C} = \frac{\sum_{f=a}^{e} b_f + \max_{\forall f \in \mathcal{F}_2} l_f}{C}$$
$$Q_2^b = \frac{\sum_{\forall f \in \mathcal{F}_1 \cup \mathcal{F}_2} b_f}{C - \sum_{\forall f \in \mathcal{F}_1} r_f} = \frac{\sum_{f=a}^{z} b_f}{C - \sum_{f=a}^{e} r_f}$$

Note that $Q_1^a > Q_1^b$ according to *Proposition 2* and therefore, partitioning allows the highest priority level to have the lowest possible WCQD and all flows at the lowest priority level ($p = 2$) to fulfill their WCQD requirements. Therefore, the algorithm assigns the partition with the lowest number of flows to level 1 whenever it is satisfied that all the flows of level 2 meet the WCQD requirement and thus it is satisfied that the WCQD experienced at $p = 1$ is minimum. So, *Lemma 1* has been demonstrated. ■

Once divided into two priority levels, three possible options can occur:

1. The priority level $p = 2$ does not find any flow f such that $Q_2 < R_f$ is satisfied. In this case, by *Proposition 3*, the Algorithm 1 has no solution.

2. Case 1 is not satisfied, and in addition the partition with $P = 2$ provides a solution that satisfies that $Q_1 \leq R_a$. In this case the Algorithm 1 has found a feasible solution with 2 priority levels, and since there was no solution with 1 priority level, then the algorithm has found the solution with the minimum number of priority levels.

3. Case 1 is not satisfied and in addition the partition with $P = 2$ provides a solution that satisfies that $Q_1 > R_a$. In this case there is no feasible solution with $P = 2$ because the level $p = 2$ cannot have more flows and the level $p = 1$ does not satisfy $Q_1 \leq R_a$.

After the base case, if the algorithm is in Case 3, there is no solution for $P = 2$, then further partitioning is necessary. That is, increase the number of priority levels to check if with a higher number of priority levels a solution can be found.

INDUCTION CASE (P=k+1): A new partition occurs when going from k to $k + 1$ and it is assumed that the partition for the case $P = k$, by the induction hypothesis, is satisfied. That is, the induction method allows us to assume that the hypothesis is satisfied for partition k where $P = k$, i.e., the $p \in [2, k]$ are feasible, fulfilling its WCQD requirement, and furthermore, at $p = 1$ the WCQD experienced is minimum (induction hypothesis).

Lemma 2. *A new partition is performed by going from k priority levels to $k + 1$ priority level.The partitioning provided by Algorithm 1 guarantees that the number of flows at $p \geq 2$ satisfies their WCQD requirement and simultaneously the WCQD experienced at $p = 1$ is minimum.*

Proof of Lemma 2.

Proof. Using *Proposition 4*, which states that all flows from levels $p = [2, k]$ to $p = [3, k+1]$ are moved, the analysis performed for the "BASE CASE" is also valid for the "INDUCTION CASE". That is, the Algorithm 1 continues to partition $p = 1$ in search of a feasible solution where the WCQD experienced at priority levels $p = [3, k + 1]$ are not affected by the assignment of flows $f \in \mathcal{F}_i, i = 1, 2$ or new partitions (see Figure 3.5).

For $k + 1$ partitions let us assume that we have two cases, *case a* and *case b*, for $p = [1, 2]$. In both cases all flows of priority level $p = [2, k + 1]$ meet their WCQD requirements. But in *case b* the number of flows assigned to $p = 1$ is less than in *case a*,as is is shown in Figure 3.6.

The WCQD of *case a* for each priority level is calculated according to *Proposition 1* and formula 3.9:

$$Q_1^a = \frac{\sum_{\forall f \in \mathcal{F}_1} b_f + \max_{\forall f \in \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_{k+1}} l_f}{C} = \frac{\sum_{f=a}^{fc_1} b_f + \max_{\forall f \in \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_{k+1}} l_f}{C}$$

$$Q_2^a = \frac{\sum_{\forall f \in \mathcal{F}_1 \cup \mathcal{F}_2} b_f}{C - \sum_{\forall f \in \mathcal{F}_1} r_f} = \frac{\sum_{f=a}^{fc_2} b_f}{C - \sum_{f=a}^{fc_1} r_f}$$

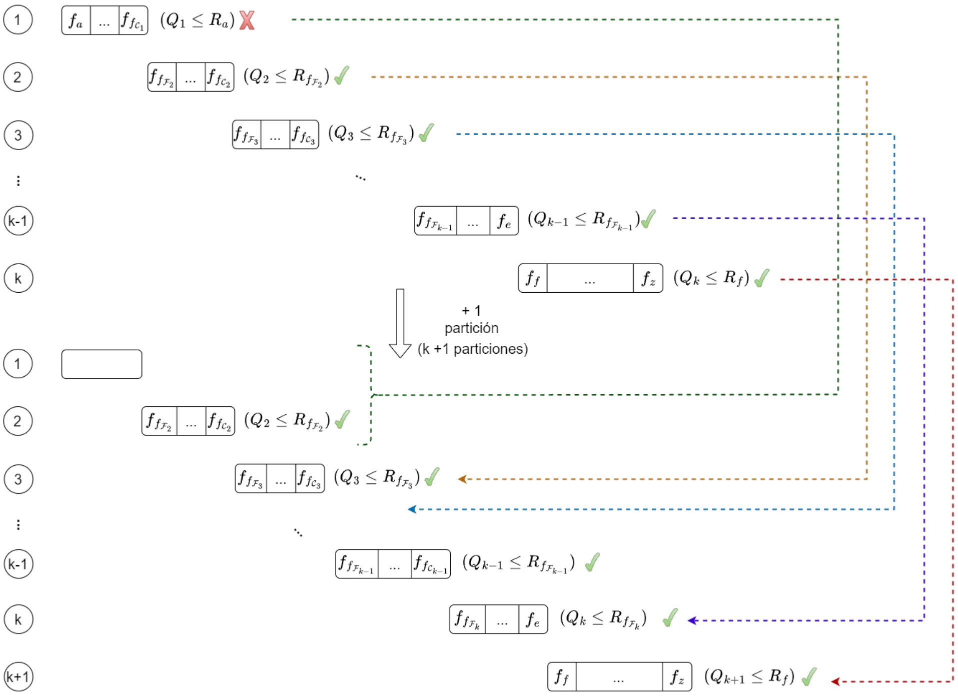

1
f_a ... f_{fc_1} $(Q_1 \le R_a)$
2
$f_{f_{\mathcal{F}_2}}$ ... f_{fc_2} $(Q_2 \le R_{f_{\mathcal{F}_2}})$
3
$f_{f_{\mathcal{F}_3}}$ ... f_{fc_3} $(Q_3 \le R_{f_{\mathcal{F}_3}})$
k-1
$f_{f_{\mathcal{F}_{k-1}}}$ ... f_e $(Q_{k-1} \le R_{f_{\mathcal{F}_{k-1}}})$
k
f_f ... f_z $(Q_k \le R_f)$
+ 1
partición
(k +1 particiones)
k+1
$f_{f_{\mathcal{F}_{k-1}}}$ ... $f_{fc_{k-1}}$ $(Q_{k-1} \le R_{f_{\mathcal{F}_{k-1}}})$
$f_{f_{\mathcal{F}_k}}$ ... f_e $(Q_k \le R_{f_{\mathcal{F}_k}})$
f_f ... f_z $(Q_{k+1} \le R_f)$

Figure 3.5: Process from k to $k+1$ partitions

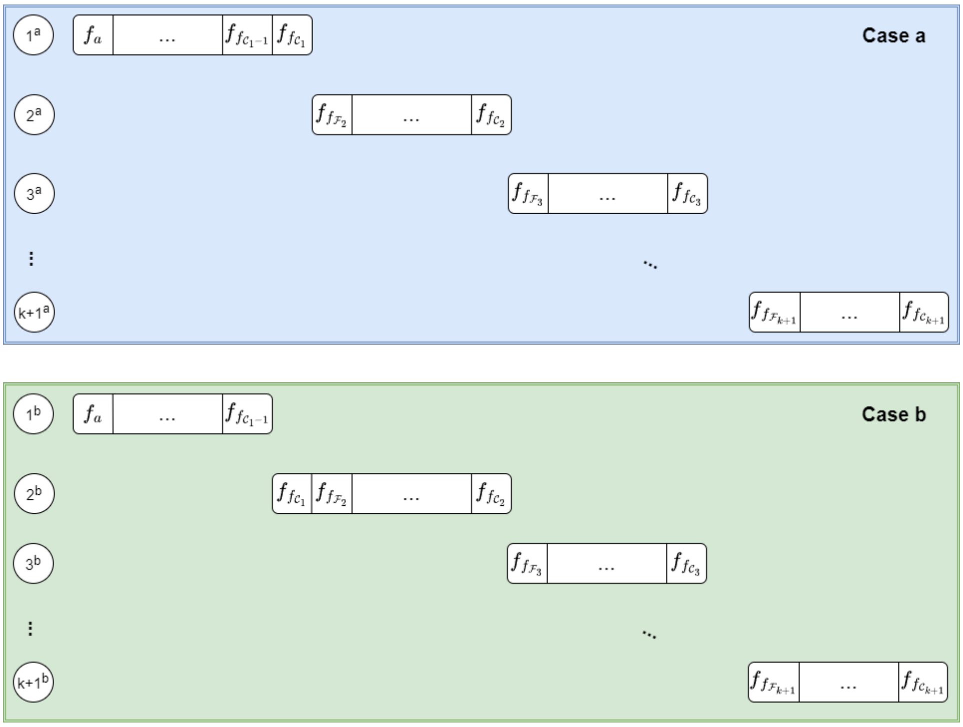

Case a
1^a
f_a ... f_{fc_1-1} f_{fc_1}
2^a
$f_{f_{\mathcal{F}_2}}$ ... f_{fc_2}
3^a
$f_{f_{\mathcal{F}_3}}$ ... f_{fc_3}
$k+1^a$
$f_{f_{\mathcal{F}_{k+1}}}$ ... $f_{fc_{k+1}}$
Case b
1^b
f_a ... f_{fc_1-1}
2^b
f_{fc_1} $f_{f_{\mathcal{F}_2}}$ ... f_{fc_2}
3^b
$f_{f_{\mathcal{F}_3}}$ ... f_{fc_3}
$k+1^b$
$f_{f_{\mathcal{F}_{k+1}}}$ ... $f_{fc_{k+1}}$

Figure 3.6: Flow allocation *case a* and *case b* for k partitions

The WCQD of *case b* for each priority level is:

$$Q_1^b = \frac{\sum_{\forall f \in \mathcal{F}_1} b_f + \max_{\forall f \in \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_{k+1}} l_f}{C} = \frac{\sum_{f=a}^{fc_1-1} b_f + \max_{\forall f \in \mathcal{F}_2 \cup \ldots \cup \mathcal{F}_{k+1}} l_f}{C}$$

$$Q_2^b = \frac{\sum_{\forall f \in \mathcal{F}_1 \cup \mathcal{F}_2} b_f}{C - \sum_{\forall f \in \mathcal{F}_1} r_f} = \frac{\sum_{f=a}^{fc_2} b_f}{C - \sum_{f=a}^{fc_1-1} r_f}$$

Note that $Q_1^a > Q_1^b$ according to *Proposition 2* and therefore, partitioning allows the highest priority level to have the lowest possible WCQD and all flows at the lowest priority level ($p = [2, k+1]$) to fulfill their WCQD requirements. Therefore, the algorithm assigns the partition with the lowest number of flows to level 1 whenever it is satisfied that all the flows of level 2 meet the WCQD requirement and thus it is satisfied that the WCQD experienced at $p = 1$ is minimum. So, *Lemma 2* has been demonstrated.

■

Once the partition is done, three possible options can occur:

1. The priority level $p = 2$ does not find any flow f such that $Q_2 < R_f$ is satisfied. In this case, by *Proposition 3*, the Algorithm 1 has no solution.

2. Case 1 is not satisfied, and in addition the partition with $P = k + 1$ provides a solution that satisfies that $Q_1 \leq R_a$. In this case the Algorithm 1 has found a feasible solution with $k + 1$ priority levels, and since there was no solution with a lower number of priority levels, then the algorithm has found the solution with the minimum number of priority levels.

3. Case 1 is not satisfied and in addition the partition with $P = k + 1$ provides a solution that satisfies that $Q_1 > R_a$. In this case there is no feasible solution with $P = k + 1$ because the level $p = 2$ cannot have more flows and the level $p = 1$ does not satisfy $Q_1 \leq R_a$.

Demonstration of Algorithm 1.

Since the Algorithm 1 iterates through all the steps from $P = 1$ onwards, then if a solution exists in each of these iterations, as demonstrated by *Lemmas 1* and *2*, then the Algorithm 1 finds it and it is the minimum in terms of priority levels.

Note that the algorithm continues to partition until priority level 1 has only a single flow, which is the end of the algorithm unless the algorithm previously indicated that there is no feasible solution by *Proposition 3*. □

Chapter 4

Experimental evaluation

This chapter describes the methodology used for the experimentation in the section 4.1 and the simulator created for the implementation of an asynchronous TSN network in the section 4.2. It also details the configuration proposed to carry out the different performance tests in section 4.3. Finally, the analysis of the performance measurements that have been carried out in the experimentation are presented in section 4.4.

4.1 Methodology

This project evaluates the new developed algorithm of prioritization of existing flows in a complete network. Due to the non-existence to date of commercial devices that implement asynchronous TSN and allow its modification, it is not possible to test the developed algorithm in a physically implemented network. Therefore, we have opted for experimental evaluation through simulations. In other words, the different ATS instances have been developed in a software application, as well as the creation of the flows and their transmission in the network.

In order to carry out the test, it is necessary to implement both the network topology used and the characteristics of the flows to be transmitted over the network. Therefore, the experimentation time is extended due to the creation of these functions in order to perform the evaluation in a network with several ATS instances where the developed algorithm and the delay distribution detailed in the chapter 3 are implemented.

Due to the short time frame of this project, the configuration and implementation of the wide variety of existing network topologies is not possible. Therefore, we have opted for the development of the main topologies used in the industrial environment. For this purpose, thanks to the study carried out in [24], three topologies have been chosen: daisy chain, star and ring. Since they are considered as the main network structures that are implemented in the current industry.

One of the main features to be taken into account in Industry 4.0, as mentioned above, is the scalability that the network must have as well as the ability to serve a large number of services. In other words, it must be considered that in the future industry there will be different types of services with different requirements among them and some of them with high bandwidth and reliability demands. Therefore, it has been decided to design different proofs of concept to demonstrate and observe the correct operation of the implemented algorithm in terms of stability and number of sensors (flows). In this project, four proofs of concepts have been designed to check if the above mentioned characteristics are verified. These four experiments are:

- **Optimality and correctness of flow prioritization algorithm per ATS instances**: In this experiment, it is verified that the prioritization algorithm developed is optimal and correct for a single ATS instance. Therefore, it is not necessary to implement any network topology. To do this, the developed algorithm is compared with the brute force search. Brute force searhc consists of checking all the possible prioritizations for the flows, and selecting that one requiring the minimum number of priority levels.

- **Comparison of prioritization mode**: In this experiment, prioritization is performed through two methods. In one of them, the different flows will be prioritized by creating a TC according to the PCP assigned to each flow. In other words, the flows with the same PCP value, those that have similar characteristics, are grouped into one type of TC. While the other other method is the prioritization by flows, i.e., the prioritization of all flows globally that pass through an ATS instance, without the need to group them according to their PCP. With this experiment the increase of sensors and services that can exist in the network, and its adaptation in the network.

- **Analysis of maximum utilization for different types of traffic**: In this experiment, the percentage of one of the types of services that exist in the network is modified, being one of those with the strictest delay requirements, to check how it influences network utilization.

- **Scalability**: In this experiment we compare the prioritization of the flows of two networks, where one of them the flows has to go through a greater number of nodes to reach the destination. In other words, two networks are created, one of them with a longer flow path than the other. In this way, we check how the distance affects the implemented algorithm and observe the reliability and the number of flows that the network is able to satisfy its requirements.

4.2 Simulator

This section details the different functions that have been necessary to develop in order to verify the correct operation of both the designed algorithm and the chosen delay distribution. In this way, it is possible to check the results of a network created in Matlab, obtaining the prioritization assigned to each of the nodes (ATSs instances) and the delay that will have each of the flows in each of the ATSs instances through which the flow passes.

The simulator created is based on a proof of concept and consists of 3 modules:

- **Network Generator**: implements the network topology to be simulated. In this case, the three main topologies used in Industry 4.0 have been generated, i.e. a ring, star and chain topology. This function also determines which are the source and destination nodes of the flows that are transmitted through the network.
- **Flow Generator**: it is responsible for generating the flows to be transmitted on the network. The flows that can be produced in Industry 4.0 have different QoS requirements, therefore, a table has been defined for each type of service in terms of the following parameters:
  - *Rate*: is the average traffic in bits per second (bit/s) generated by a flow. It is measured in Mega bits per second (Mbps).
  - *Burstiness*: is the maximum number of bits that a traffic flow can generate at a given instant. This parameter is measured in packets.
  - *Delay*: is the maximum E2E delay that flows can suffer in the asynchronous TSN network. This parameter is measured in milliseconds.
  - *Length*: is the maximum packet size of each flow. It may happen that in the study of the traffic type requirements there is no information on the packet size, we have chosen to consider that the maximum size is the Ethernet Maximum Transmission Unit (MTU), 1500 Bytes. This value is possible since TSN is an extension of Ethernet. This parameter is measured in KBytes.
  - *Duration*: is the average duration of the flows in the network. That is, this parameter indicates the average time that a user is

connected to the network generating traffic. This parameter is measured in seconds.

- **Prioritization solver**: This module is in charge of determining if a minimum assignment of the priorities of the flows that have been created in the flow generator is feasible in each of the ATS instances that conform the selected network. It performs three main functions:

    – It determines which is the optimal path for each of the flows according to their origin and destination. The origin and destination of each of the flows are randomly selected from the options that have been configured in the Network Generator module. In this case, the path is considered to be the one that has the least number of hops, i.e., the path that minimizes the number of ATS instances to be crossed is selected while does not exceed the capacity of the link connecting TSN bridges along the chosen path. It is essential to ensure that the selected path does not exceed the capacity of the links connecting the TSN bridges.

    – Once the path of each of the flows is determined according to its origin and destination, the delay distribution of each of the flows is performed by the ATS instances that traverse according to the path. In this case, the solution provided to sub-problem (3.6) in section 3.3 is implemented.

    – After the distribution of the maximum delays of the flows in each of the ATS instances of the network, the prioritization algorithm is executed in each ATS instance. In other words, the algorithm 1 detailed in subsection 3.4.3 is executed.

  After this procedure, a prioritization solution of the different flows is obtained for each of the ATS instances if there is a feasible solution lower than 8 priority levels, as considered in [7]. The priority level of each of the flows of each ATS instance and the delay suffered in that instance are also provided.

In the industrial network there is not only delay-critical traffic, but there are also best-effort services that can be transmitted in the same network without the need to fulfill a strict delay. In this way, all the remaining capacity of the links can be used to accommodate the services with best-effort traffic without modifying the priority assignment obtained from the critical flows. It is only necessary to add an additional minimum priority queue to accommodate the best-effort type of traffic. Specifically, in the simulator the only thing that needs to be changed is that all priority levels that accommodate flows with delay requirements perceive a maximum frame size associated with the non-preemptive operation of 1500 Bytes (Ethernet MTU). This takes into account the maximum packet size that the best-effort priority queue would have, and this is the only impact it would have on flows accommodated at higher levels.

These three modules have been implemented in the Matlab programming language. Matlab is a numerical computing system that offers an integrated development environment Integrated Development Environment (IDE) with its own programming language (M language). This tool is mainly used by engineers and scientists in order to analyze data, develop algorithms and create models. Matlab is used in many applications such as signal and image processing, control systems, wireless communications and robotics. We have chosen to use this language mainly because of its computational capacity and the ease of matrix interaction and arrays, which are necessary for the resolution of the problem addressed in this project.

4.3 Experimental Setup

In this section, the parameters and considerations configured for the experimental part of the project are detailed.

4.3.1 Topology

Firstly, the three network topologies considered have been configured in order to test the performance of the proposed flow prioritization solution in a complete network. As previously discussed in section 4.1, the Figure 4.1 shows the three topologies implemented. This figure shows the predefined route that interconnect a source-destination for the second experiment, and these routes are randomly assigned to each of the flows. That is, as can be seen there are four paths in each of the topologies, each of these paths have a number of flows randomly assigned from all flows that are transmitted in the network. In each topology it is considered that there are a total of 5 nodes. Specifically, the star topology has two sources of flows at nodes N2 and N4, and two destination nodes N3 and N5. In the daisy chain topology, the source nodes are N1 and N5 and the destination nodes are N3 and N4. Finally, in the ring topology the source nodes are N2 and N5 and the destination nodes are N3 and N4.

For the third experiment, we considered using only the daisy chain topology with the same routes and source and destination nodes. In the last experiment, there is only one route where one of the cases has a higher number of hops, i.e. ATSs instances to transmit while the other case has a lower number of hops. The capacity of each of the links is 1 Gbps.

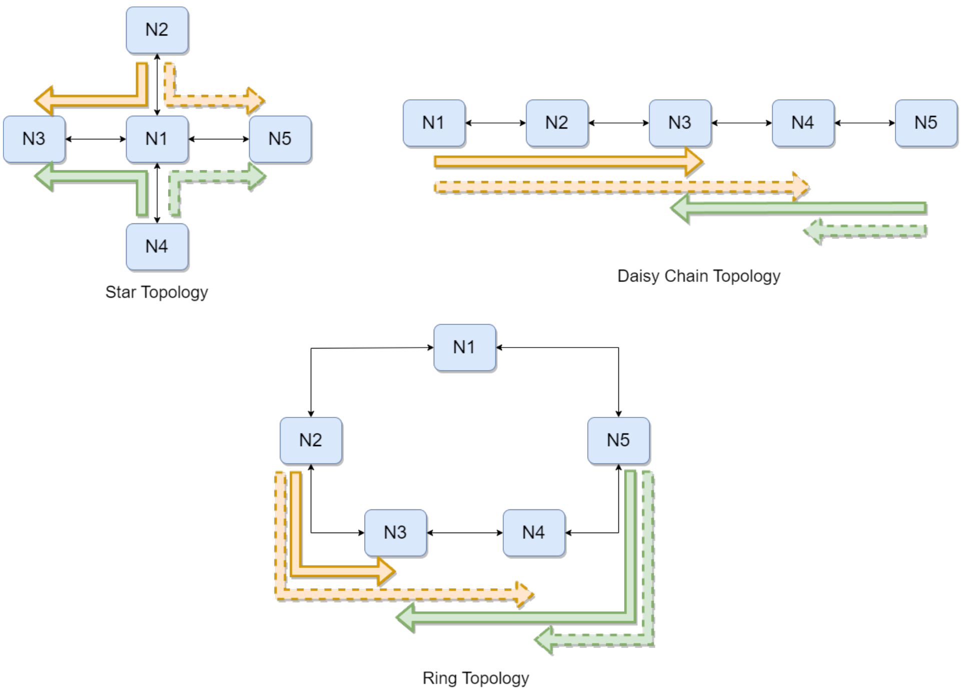


Figure 4.1: Industrial network topology

In this project the selected path w_f for each flow f will have the least number of hops, ATSs instances, through which the flow must pass to reach its destination among all the possible paths that are defined between the source and destination $s_f \in \mathcal{V}$ and $d_f \in \mathcal{V}$ as indicated in section 4.2. Thus, in Figure 4.1 it can be seen that the indicated source-destination path is the one with the lowest number of ATSs instances through which the flows must pass to reach the destination.

4.3.2 Flow characteristics

Due to the existence of a wide variety of services in the industrial network, a compound traffic model has been developed, summarized in Table 4.1 and based on [49, 52, 53, 54] with the objective to realistically capture in experiments the typical flow traffic demands and delay requirements in industrial scenarios. This table details the rate (r_q), burstiness (b_q), maximum E2E delay (D_q) and maximum packet size (l_q) for each of the traffic service types considered in a given range.

Specifically, seven types of services have been defined that can exist [54, 52]:

- **Cyclic with strict delay requisite**: this type of traffic is similar

Type of Services	r_q (Mbps)	b_q (packet)	D_q (ms)	l_q (KBytes)	**PCP**
Cyclic strict delay requisite	$0.8 - 8$	$1 - 4$	$0.5 - 1$	$0.05 - 1$	6
Mobile Robots	< 10	$1 - 4$	$1 - 500$	$0.04 - 0.25$	3
Cyclic lower delay requisite	$0.2 - 4e^{-3}$	$1 - 4$	$2 - 20$	$0.05 - 1$	5
Events: Control	> 12	$1 - 4$	$10 - 50$	$0.1 - 0.2$	4
Augmented Reality	$10 - 20$	$1 - 4$	10	$0.03 - 1.5$	2
Network Control	$4e^{-3} - 8e^{-3}$	$1 - 4$	$50 - 1000$	$0.05 - 0.5$	7
Config. & Diagnostics	2	$1 - 4$	$10 - 100$	$0.5 - 1.5$	1

Table 4.1: Per-service flow characteristics.

to motion control. This service controls the moving and/or rotating parts of the machines, where a series of messages are sent that must be performed in a cyclic and deterministic way.

- **Cyclic with lower delay requisite**: this type of traffic is similar to close-loop control. In these cases, latency and determinism are strict requirements, as well as service availability. The service area is larger than in motion control use cases and may not interact with the public network. These can be plant sensors that continuously perform measurements. Their data is transmitted to a controller that decides whether to modify the state and/or characteristics of the actuators or not.

- **Mobile Robots**: use of mobile robots with great functionality. They allow maximum flexibility in mobility, with certain autonomy and sensing capabilities (perceive and react to the environment). Mobile robots are supervised and controlled from a guidance control system, which obtains information from the process to avoid collisions, assign tasks and manage traffic.

- **Events: Control**: communication between different industrial controllers. Normally, there is no fixed configuration, and the control nodes that are in the network vary according to the state of the machines and the manufacturing plant, therefore, the connection of the control nodes is very important. It can be the control traffic coming from the motion subsystems that is transmitted between controlled.

- **Augmented Reality**: allows supervision of production processes and flows, step-by-step instructions for specific tasks (e.g., in manual assembly workplaces), and ad hoc support from a remote expert (e.g., for maintenance or service tasks). The traffic it generates is bidirectional between augmented reality devices and an image processing server, for example.

- **Network Control**: this type of traffic contains the control messages used in the network. It has a low volume but strict delivery requirements. Some of these control messages include for example clock synchronization (e.g. PTP), network redundancy (e.g. Multiple Spanning Tree Protocol (MSTP), Rapid Spanning Tree Protocol (RSTP)) and/or topology detection (e.g. LLDP).

- **Configuration & Diagnostics**: is responsible for the monitoring of processes and/or assets that may occur in industrial production. That is to say, the transport of data for the configuration of the devices and the diagnosis or update of the firmware. These data are usually transmitted via Transmission Control Protocol (TCP)/IP based protocols and do not have an instantaneous impact on the process. Therefore, they are not time-critical, but must eventually be delivered and therefore, the data acquisition process does not present latency requirements. Some services are for example:

  - Processing application information, such as order scheduling and production.
  - Devices have software/firmware that must be updated from time to time.
  - Diagnostic activities to monitor equipment status that create acyclic traffic.

In six column of Table 4.1, the value of the PCP assigned to each type of service according to [52] is shown.

All of this type of detailed traffic contains more or less critical delay requirements that need to be met. However, in the industry there is not only critical traffic but also best effort traffic that can be transmitted over the same network. Therefore, it has been considered in the experiments that there is an additional minimum priority queue where it is considered that there is best effort traffic in the background until the link utilization is completed.

4.3.3 Experimental Methodology

Each experiment consisted of finding the flow prioritization for an asynchronous TSN network with several ATS instances for $F = |\mathcal{F}|$ flows. F is the total number of flows existing in the network; however, each types of services presents a specific number of flows obtained from the traffic percentages deduced from the 5G-ACIA document [53]. Since this document includes other types of services different from those considered in this project, a mapping is performed between these types of services based on the similarities of their network parameters. The probabilities of each type of service are obtained by dividing the existing downlink rate of each type of service in the network by the total downlink rate. From these percentages, the number of flows for each type of service considered in this project is obtained for each of the simulations. The Table 4.2 summarizes these percentages assigned to each type of service. It should also be noted that the percentage of traffic is not the same as the percentage of flows, because each type of service generates a different amount of traffic.

Type of services	**Traffic Percentage**
Cyclic strict delay requisite	$0,6235$
Mobile Robots	$0,0301$
Cyclic lower delay requisite	$0,0805$
Events: Control	$0,1645$
Augmented reality	$0,077$
Network Control	$0,0245$
Config. & diagnostic	$2,68e^{-06}$

Table 4.2: Traffic percentage for each type of service

In each of the experiments, a certain sweep of number of flows F with jumps of X flows is performed. For each number of flows, 100 independent runs were conducted. In each run, flows are created until reaching the number F. The created flow is of one type of services or another based on the traffic percentage for each types of services from Table 4.2. Once the type of service is selected, the requirements (e.g., delay) are chosen within the specified range in Table 4.1 following a uniform distribution. After configuring the flow characteristics, an origin and destination are assigned to each of them following another uniform distribution. The capacity of each link in the network is 1 Gbps, and the processing delay t_{proc} and propagation delay t_{prog}, detailed in section 3.1, are considered null for simplicity as they are constant variables. It is assumed that the traffic conforms to the committed data rate and burstiness, then the ATS produces zero packet loss.

The four experiments were developed in Matlab and run on a server with Intel(R) Core(TM) i7-6700K CPU at 4.00GHz with 4 cores and 32 GB of RAM. For each experiment, the following statistics have been obtained: average execution time, priority assignment, utilization and the number of the realizations of each scenarios in which a feasible solution has been found. The reported execution time measurements represent the average across all runs for the same number of flows in all ATS instances. This execution time includes the computation time of the prioritization algorithm for the complete network and the distribution of the delay requirement among ATS instances. In other words, it does not include the time it takes for the simulator to create flows and the topology, and to save the obtained results. The priority assignment, utilization and the number of the realizations of each scenarios in which a feasible solution has been found are provided by the ATS instance with the highest utilization across the complete network, which means it has the highest number of flows.

In the first experiment, the prioritization problem for a single ATS is solved using the brute force algorithm and the one developed for comparison. Since problem (3.10) quickly becomes intractable as the scenario scale increases when solved using brute force, it has been decided to prioritize by TC according to IEEE 802.1Q, referred to in our case as “Prioritization by PCP”. This same Prioritization by PCP is also used for the second experiment. Specifically, each service in Table 4.1 was assigned to a TC and the corresponding PCP (see the sixth column in Table 4.1). For Prioritization by PCP, each TC considered is characterized by:

- The committed data rate and the burst size correspond to the sum of the committed data rate and the burst size of all flows, respectively

$$r_q = \sum_{\forall f \in q} r_f$$
$$b_q = \sum_{\forall f \in q} b_f$$

- The maximum frame size is determined as the maximum frame size among all its streams.

$$l_q = \max_{\forall f \in q} l_f$$

- The delay requirement is set as the most stringent delay requirement among all flows.

$$d_q = \min_{\forall f \in q} d_f$$

4.4 Results

This section details the results obtained for the four experiments that have been simulated in the simulator detailed in the section 4.2. First, we compare the priority allocation obtained for the prioritization by PCP solved through the brute force search algorithm and the algorithm developed for a single ATS instance. Next, the results obtained when comparing the prioritization by flow and by PCP are explained. Then, we show the results obtained when the traffic percentages for each types of services are modified. With this experiment, it is possible to check how the change in the percentage of a type of services in the network affects the utilization. Finally, the results obtained when the route of the flows is modified are analyzed in order to check the scalability of the implemented algorithm.

4.4.1 Optimality and correctness of flow prioritization algorithm per ATS instances

In this experiment, we verify whether the priority assignment of flows provided by our algorithm for a single ATS instance is correct and optimal. To do this, we compare this Prioritization by PCP result with the one obtained using the brute force algorithm.

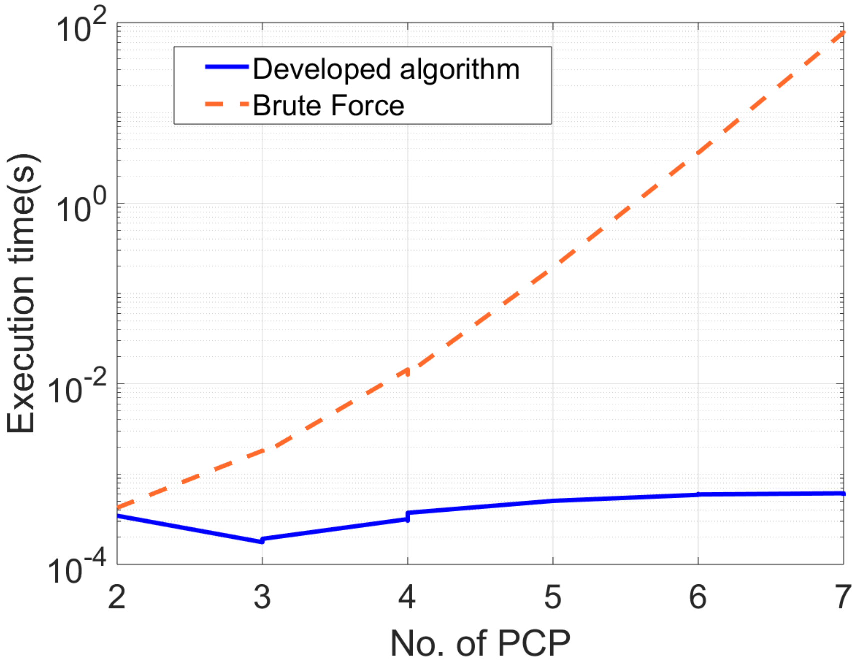


Figure 4.2: Algorithm execution time.

Figure 4.2 depicts the comparison of the average execution time exhibited by the brute force (labeled as “Brute Force”) and our heuristic-based (labeled as “Developed algorithm”) algorithms as a function of the number of TCs to be prioritized.

As observed, for the considered range of TCs, the results show that the execution time of the brute force algorithm exhibits an exponential growth whereas the our proposal scales well. For instance, for prioritizing seven TCs the brute force’s execution time is six orders of magnitude higher than our proposal. This makes unfeasible to use the brute force algorithm to carry out a per-flow prioritization in the ATS.

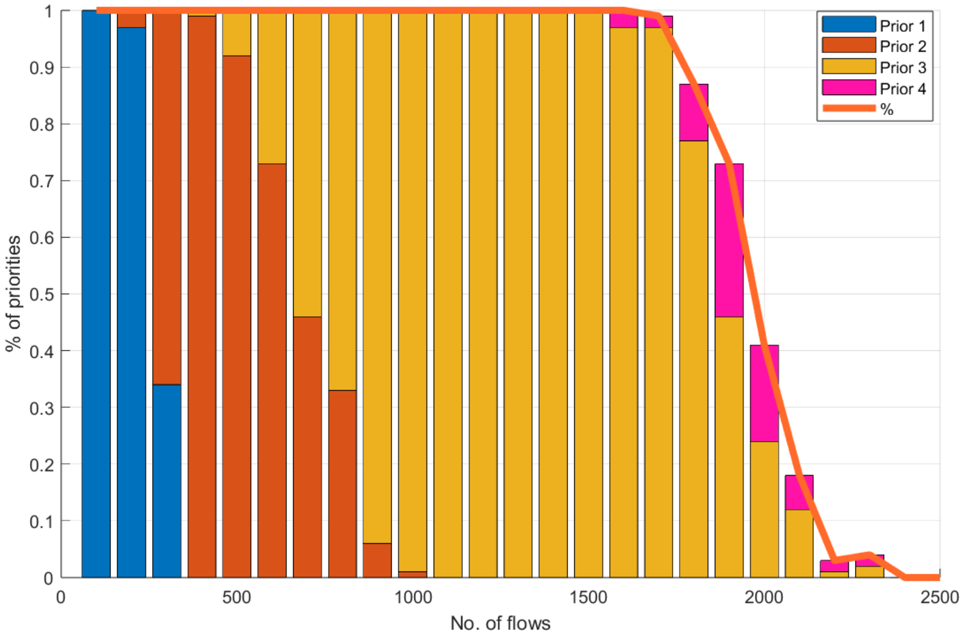


Figure 4.3: Priority assignment for different flows.

Figure 4.3 depicts the prioritization outputted by our solution for different scenarios. Each scenario includes a given number of flows (x-axis) grouped into TCs as explained in the previous subsection. For each value of F, 100 independent realizations were carried out, each sampling the flow features according to the ranges provided in Table 4.1 The line labeled as ’%’ represents the number of realizations of each scenario in which a feasible solution were found. Similarly, each bar, labeled as ’Prior P’, represents the percentage of realizations requiring at least P priority levels. As expected, the higher the number of flows in the scenario, which translates into higher utilization of the ATS link, the higher the probability of not finding a satisfiable solution, even if the number of priority levels is increased. Similarly, the minimum number of priority levels required increases with the traffic load.

The most remarkable result of this experiment is that both our proposal and the brute force algorithm outputted exactly the same prioritization for all the experiments, thus validating our proposal's operation and optimality. These results support that our solution finds a satisfiable solution if it exists and that solution requires the lowest number of priority levels as stated in section 3.4.

4.4.2 Comparison between prioritization by flows and by PCP

In this second experiment we compare the priority assignment obtained for the prioritization by flow and by PCP of the different traffic flows in the asynchronous TSN network. Prioritization by PCP has been explained in section 4.3. While the prioritization by flow consists of executing the algorithm designed for the number of flows considered in the experiment without the need to group the characteristics of the flows according to their PCP.

In this case, the number of flows is swept from 100 to 1300 flows in 10-flow steps. For each F value considered, 100 independent runs are performed in order to obtain better statistical data. In each of the runs the flow characteristics were randomly sampled according to the ranges provided in Table 4.1.

The following shows the prioritization results obtained and the average execution time for the ATS instance with the highest utilization depending on the topology followed.

Daisy chain Topology

In this case the daisy chain topology is simulated. Figure 4.4 shows the implemented topology and the capacity of each of the links.

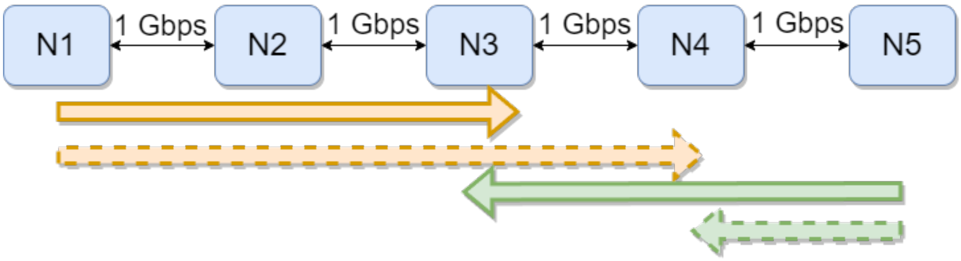


Figure 4.4: Daisy chain Topology

Figure 4.5 and 4.6 shows the average execution time exhibited by the prioritization by PCP (titled as “Execution Times by PCP”) and by the prioritization by flows (titled as “Execution Times by flows”) as a function of the number of flows to be prioritized, respectively.

As can be observed, the results show that the execution time for the prioritization by flows exhibits an exponential growth while for the prioritization by PCP it scales well. Specifically, it remains approximately constant at a value of 0.065s when all seven traffic classes are reached regardless of the number of flows. For example, for 300 flows the execution time is 0.064s for prioritization by PCP while for prioritization by flow it is twice the magnitude, 0.122s. However, as seen in Figure 4.5 the number of flows reaches a maximum of 300 flows, since this is the value at which prioritization by PCP stops providing a feasible solution. In contrast to the prioritization by flows which allows to obtain a feasible solution up to 1300 flows (see Figure 4.6).

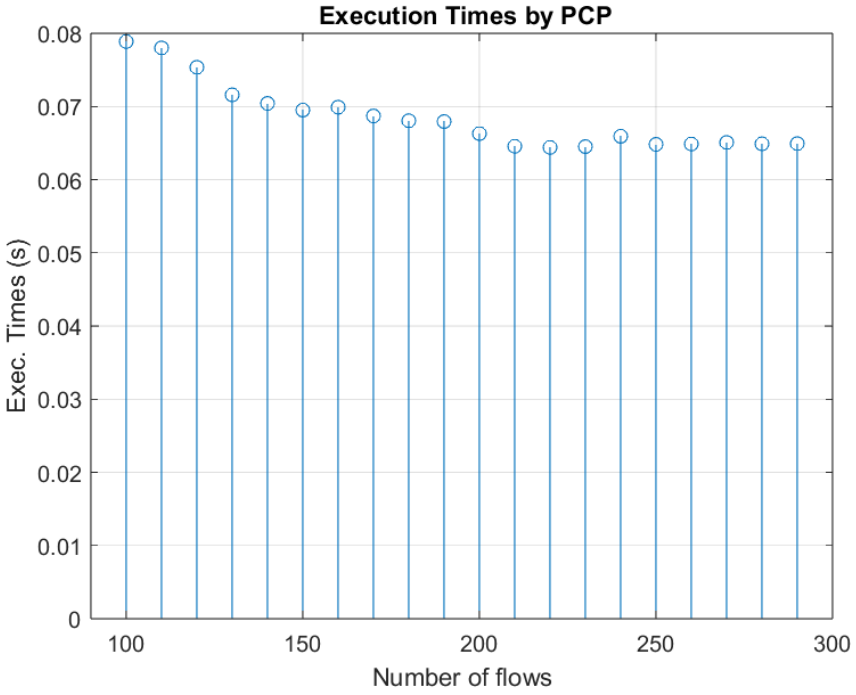


Figure 4.5: Daisy chain topology execution time for prioritization by PCP

Although the prioritization time per flow is exponential, the time obtained is not high, moreover, it allows to obtain a feasible solution for a larger number of flows. Figure 4.7 and 4.8 show the prioritization generated by PCP and flow prioritization for the different scenarios in the ATS instance with the highest utilization, respectively. The line labeled “Utilization” represents the maximum utilization obtained on the link with the highest utilization.

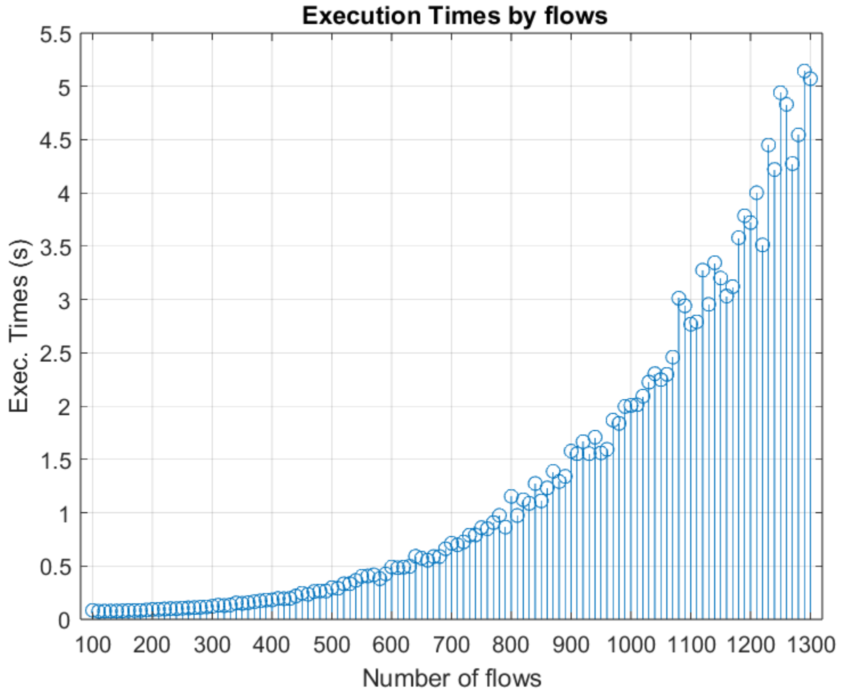


Figure 4.6: Daisy chain topology execution time for prioritization by flows

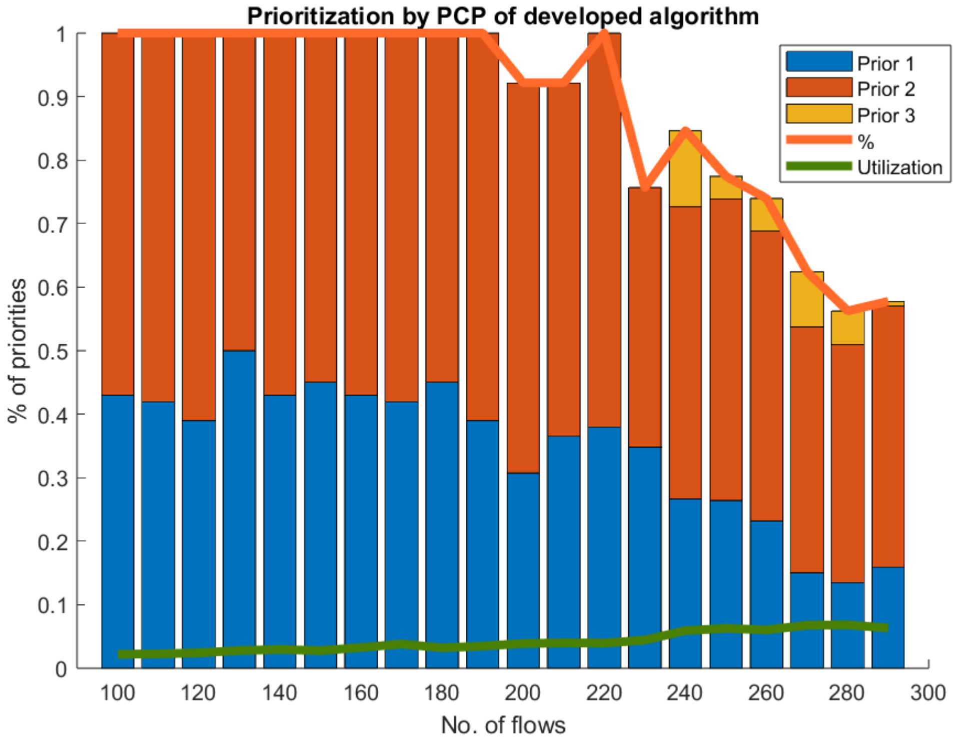


Figure 4.7: Priority assignment by PCP in the Daisy chain topology.

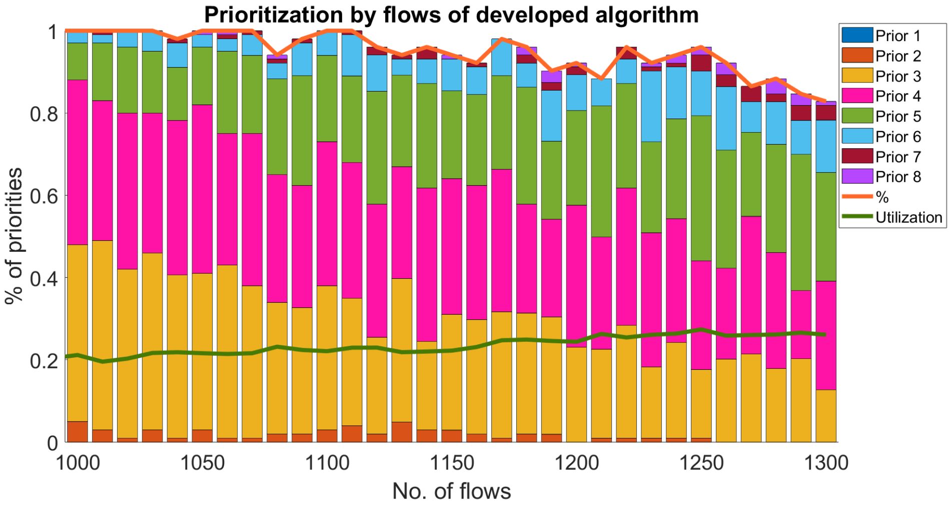


Figure 4.8: Priority assignment by flows in the Daisy chain topology.

If we compare both figures, 4.7 and 4.8, we can see that to obtain a percentage of feasible solutions around 80%, the PCP prioritization reaches a maximum value of 250 flows while the flow prioritization reaches a maximum value of 1300 flows. This causes that the maximum utilization reached with the prioritization by flows is double that obtained by the PCP prioritization, around 30%.

Therefore, as a conclusion of these simulations, the use of the algorithm of prioritization by flows in several ATS allows to obtain a feasible result for a greater number of flows as distinct from the prioritization by PCP. In addition, it allows to obtain a higher maximum utilization. However, it has the disadvantage of the execution time, which is higher in the case of flow prioritization. Nevertheless, the time obtained to obtain the priority assignment for 1300 flows in a complete network of 5 ATS instances is not excessively high and can be executed in an industrial network without major delay problems.

Star Topology

In this case the star topology is simulated. Figure 4.9 shows the implemented topology and the capacity of each of the links.

Figure 4.10 and 4.11 show the average execution time exhibited by PCP prioritization and per-flow prioritization as a function of the number of flows to prioritize. Again, the results show that the execution times for stream prioritization exhibits exponential growth while that for prioritization by PCP scales well. Specifically, it remains approximately constant at a value of 0.065s when all seven TCs are reached regardless of the number of flows.

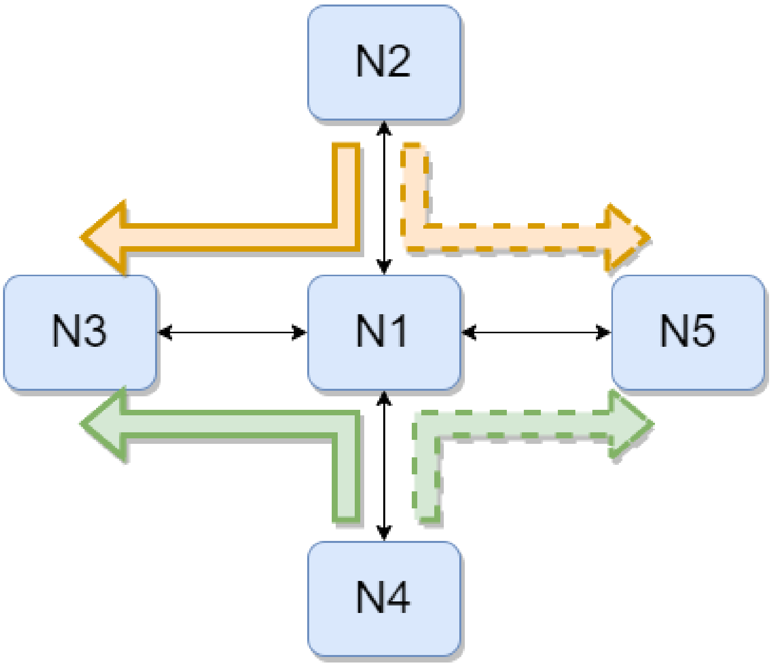


Figure 4.9: Star Topology

Similarly if compared for example for 300 flows the execution time is 0.064s for prioritization by PCP while it is twice as long for flow prioritization. However, for prioritization by PCP a feasible prioritization solution is obtained for a maximum number of 300 flows while for flow prioritization it is 1300 flows.

Therefore, although the prioritization time per flow is exponential, the time obtained is not high, and it allows to obtain a feasible solution for a larger number of flows. Figure 4.12 and 4.13 show the prioritization generated by PCP and prioritization by flow for the different scenarios in the ATS instance with the highest utilization, respectively. As for the daisy chain topology, the higher the number of flows in the scenario, which translates into a higher utilization of the ATS link, the higher the probability of not finding a satisfactory solution, even if the number of priority levels is increased.

Comparing both figures, 4.12 and 4.13, it is again observed that to obtain a feasible solution percentage around 80% the prioritization by PCP reaches a maximum value of 280 flows while for the prioritization by flows it is reached with a maximum value of 1250 flows. This causes that the maximum utilization reached with the prioritization by flows is double that obtained by the PCP prioritization, around 30%.

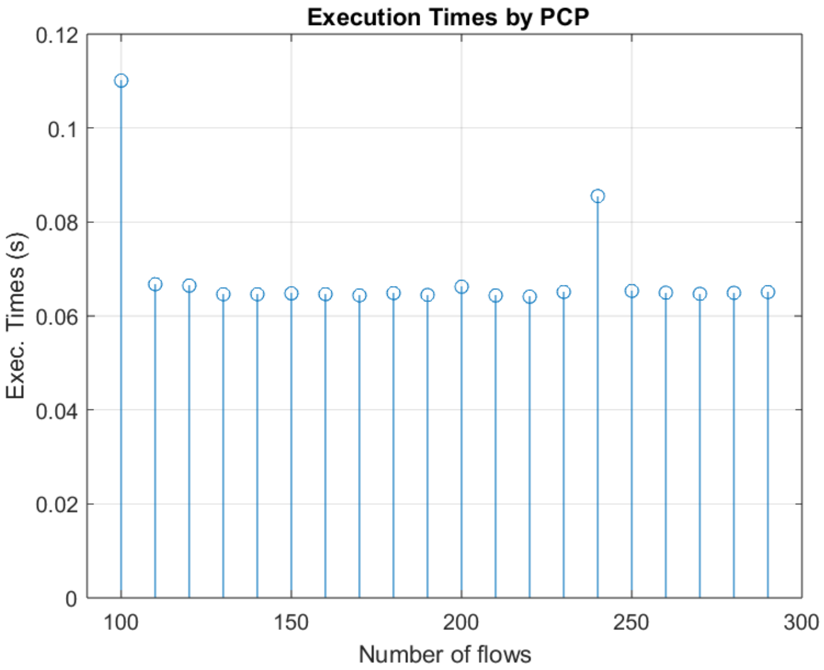


Figure 4.10: Star topology execution time for prioritization by PCP

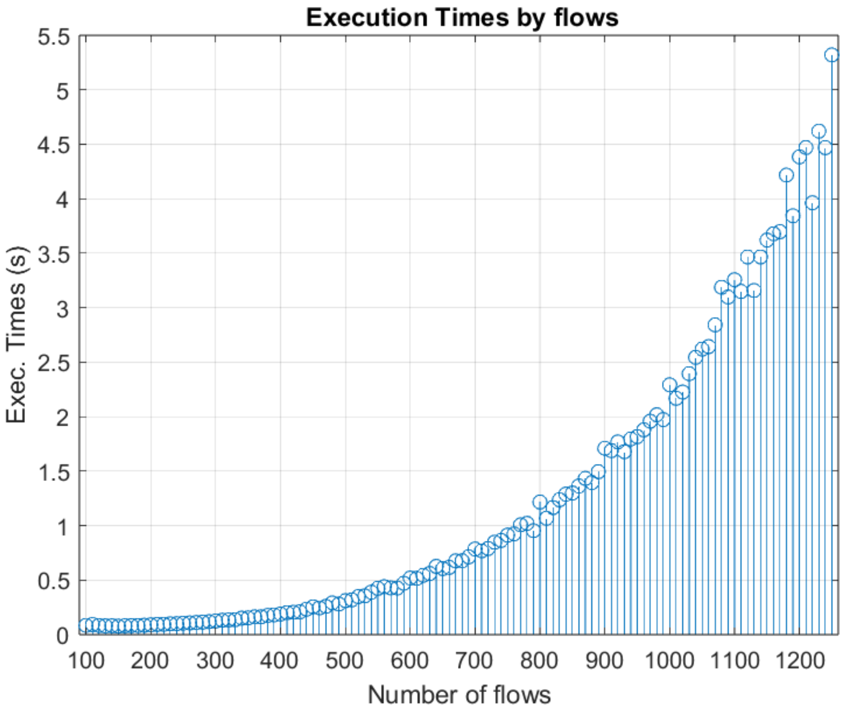


Figure 4.11: Star topology execution time for prioritization by flows

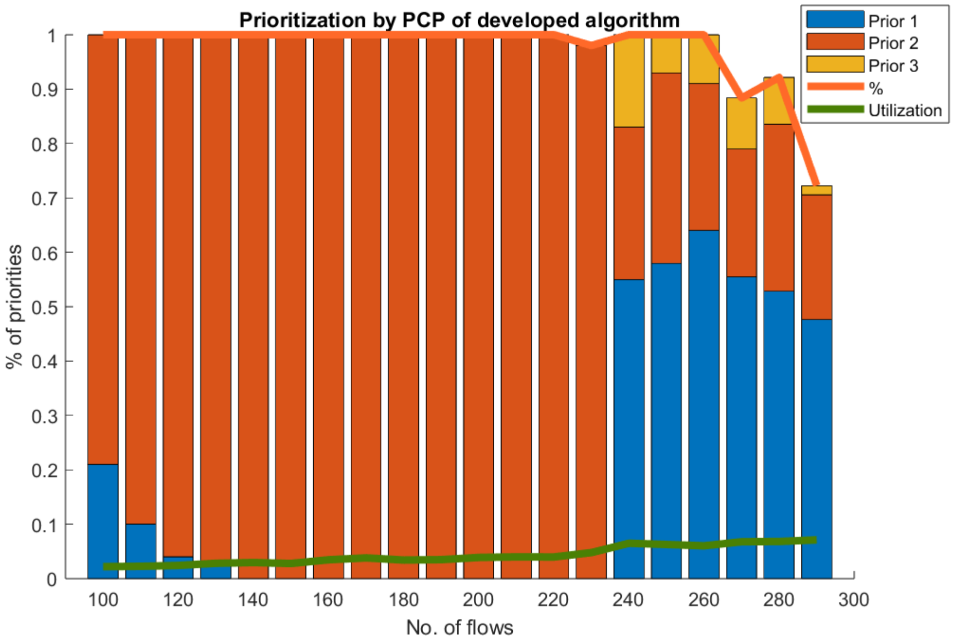


Figure 4.12: Priority assignment by PCP in the star topology.

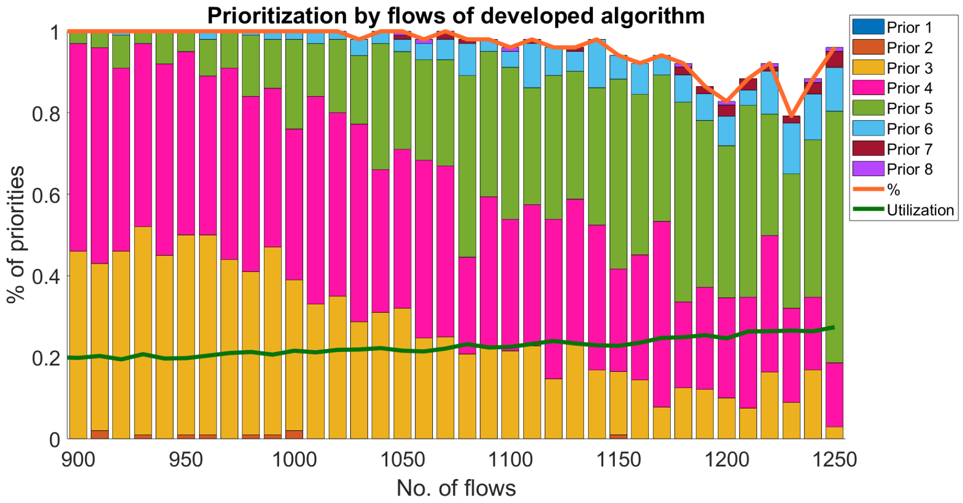


Figure 4.13: Priority assignment by flows in the star topology.

As a conclusion of these simulations, it is verified that the use of the algorithm of prioritization by flows in several ATS allows to obtain a feasible result for a greater number of flows as opposed to the prioritization by PCP. In addition, it allows to obtain a higher maximum utilization. However, it has the disadvantage of execution time, which is higher in the case of flow prioritization.

The variations that the star topology suffers with respect to the daisy chain topology can occur due to the number of hops in each network and because the characteristics of each of the flows are randomly assigned in each of the scenarios.

Ring Topology

In this case the ring topology is simulated. Figure 4.14 shows the implemented topology and the capacity of each of the links.

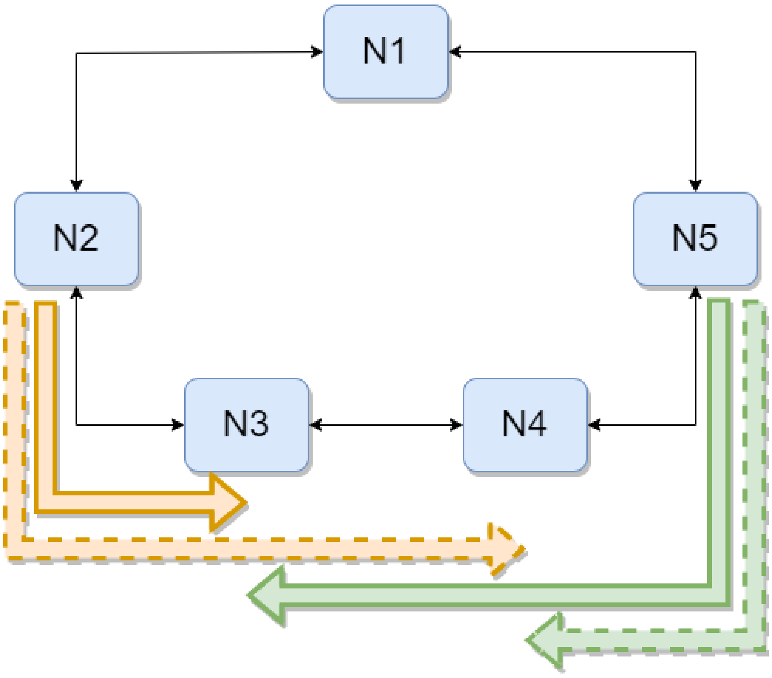


Figure 4.14: Ring Topology

Figure 4.15 and 4.16 show the average execution time exhibited by PCP prioritization and per-flow prioritization as a function of the number of flows to prioritize. As in the previous cases, the results show that the execution time for flow prioritization shows an exponential growth while for prioritization by PCP it scales well. Specifically, in this case it remains approximately constant at a value of 0.07s. For example, for 300 flows the execution time is 0.07s for prioritization by PCPwhile for prioritization by flow it is less than twice the magnitude, 0.122s. However, as seen in Figure 4.5 the number of flows reaches a maximum of 300 flows, since this is the value at which PCP prioritization ceases to provide a feasible solution. On the contrary, the prioritization by flows allows to obtain a feasible solution up to 1300 flows (see

Figure 4.6). It is also observed that for this topology the execution time is lower than that obtained in the rest of the topologies for the prioritization by flows.

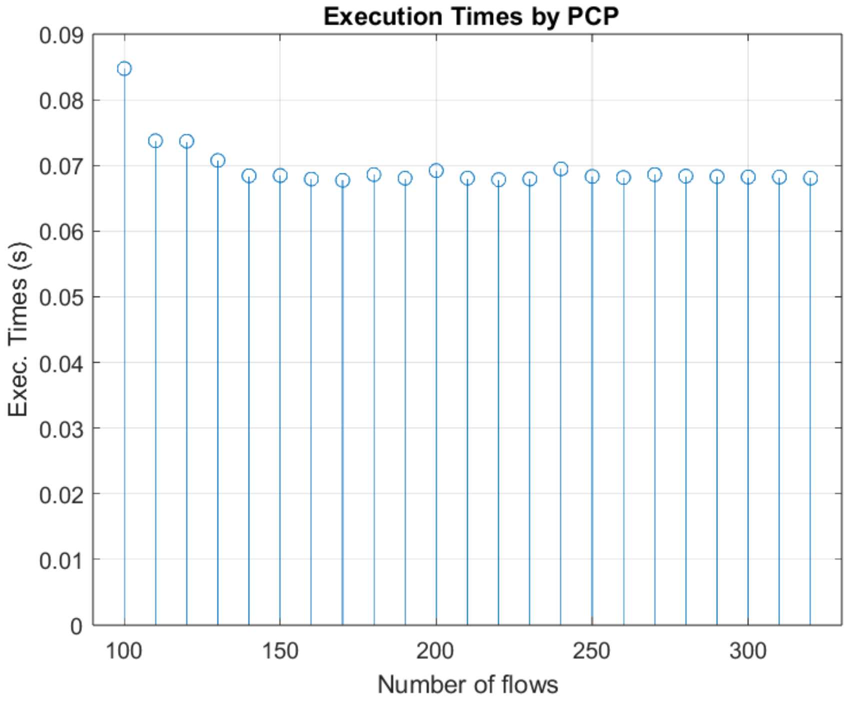


Figure 4.15: Ring topology execution time for prioritization by PCP

Figure 4.17 and 4.18 show the prioritization generated by PCP prioritization and flow prioritization for the different scenarios in the ATS instance with the highest utilization, respectively. As for the other topologies, the higher the number of flows in the scenario, which results in a higher utilization of the ATS link, the higher the probability of not finding a satisfactory solution, even if the number of priority levels is increased.

Comparing both figures, 4.17 and 4.18, it is again observed that to obtain a percentage of feasible solutions around 80% the prioritization by PCP reaches a maximum value of 310 flows while for the prioritization by flows it is reached with a maximum value of 1300 flows. Specifically, for 1300 flows, an operating range of 90% is obtained. Although the maximum utilization reached is still around 30% for flow prioritization, being twice as high as for PCP prioritization.

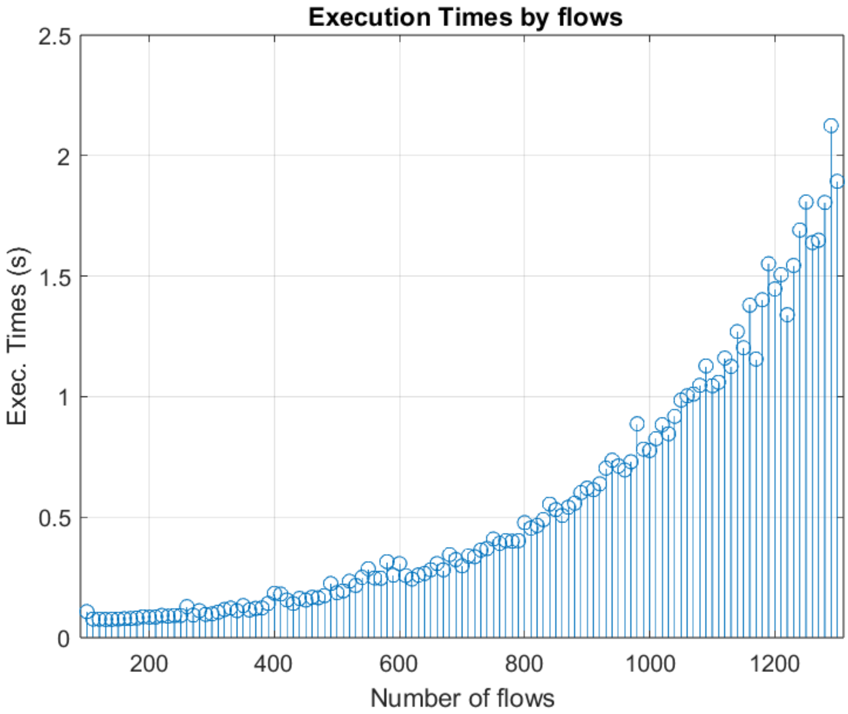


Figure 4.16: Ring topology execution time for prioritization by flows

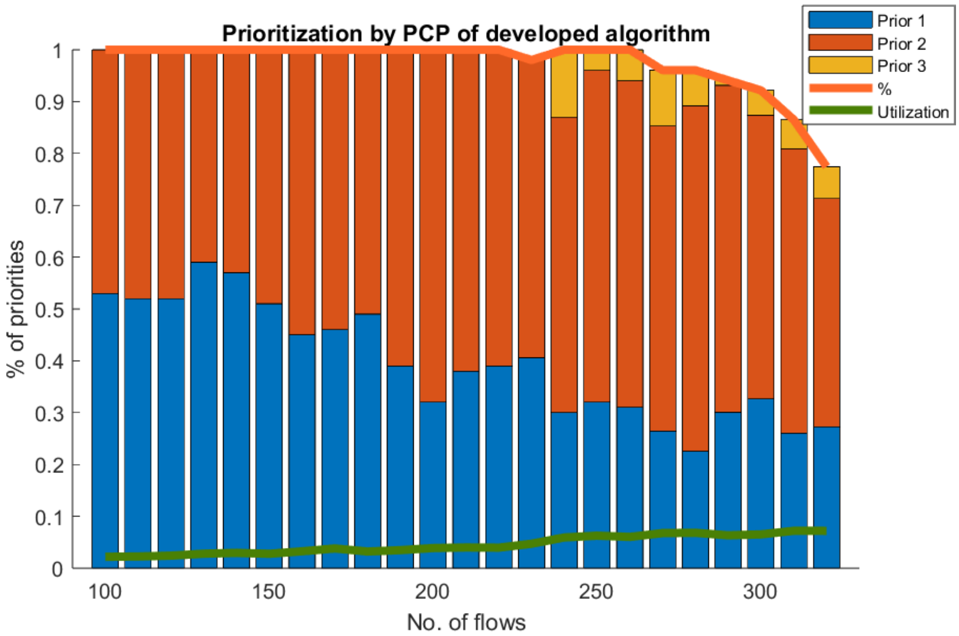


Figure 4.17: Priority assignment by PCP in the ring topology.

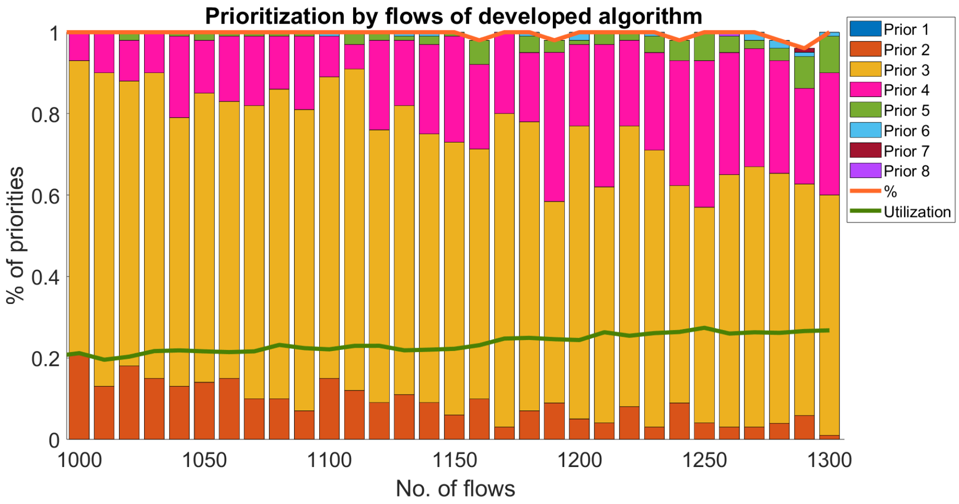


Figure 4.18: Priority assignment by flows in the ring topology.

Again, it is verified that the use of the algorithm of prioritization by flows in several ATS allows to obtain a feasible result for a greater number of flows as compared to the prioritization by PCP. Moreover, it allows to obtain a higher maximum utilization. Since the utilization of prioritization by PCP is around 5% which is unacceptably low while the utilization by flow is around 25% which is also a low but acceptable number. In addition, the execution time for flow prioritization is also acceptable. Furthermore, specifically for this particular topology the execution time is reduced.

The variations that occur between the three topologies are mainly due to the number of hops in each network and because the characteristics of each of the flows are randomly assigned in each of the scenarios.

4.4.3 Analysis of maximum utilization for different traffic class percentages

This experiment consists in the variation of the percentage of traffic of the different types of services used to calculate the number of flows assigned to each service.

Specifically, the variation of the percentage of traffic of the cyclic with strict delay requisite service is performed in order to analyze how the amount of flow assigned to a each critical service and therefore to a strict delay requirement affects the performance of the prioritization by flow algorithm. In this case, prioritization by flow has been chosen because, as it has been shown above, it allows to obtain a wider range of feasible solutions in terms of number of flows in a correct execution time. It also allows to study the maximum utilization that can be achieved for each of the fixed traffic percentages, as well as the maximum number of flows that can be transmitted in a network with a daisy chain topology (see Figure 4.4). In this case, the daisy chain topology has been chosen as it is the one that shows the average behavior of the three topologies studied.

Note that we are going to study 3 different cases of the original scenario for the cyclic service with strict delay requisite (62.35% indicated in Table 4.2), i.e., the original percentage of such service based on the 5G-ACIA [53]: 30%, 50% and 80% of the cyclic with strict delay requisite service. Exactly the percentage of cyclic with strict delay requisite service is not being modified, but the total rate obtained by this traffic is being varied to obtain the required percentage, maintaining the rates of the rest of the services obtained from the 5G-ACIA. In this way, the rest of the percentages can be obtained as the traffic rate of this service increases or decreases.

Again, the number of flows sweeps from 100 to 400 in steps of 100 flows for the different percentages of traffic and 100 runs are performed for each of the scenarios.

Tables 4.3, 4.4 and 4.5 show the percentages of traffic assigned to the different services when the percentage of traffic of the cyclic with strict delay requisite is set at 30%, 50% and 80%, respectively.

Type of services	**Traffic Percentage**
Cyclic strict delay requisite	$0,3005$
Mobile Robots	$0,0559$
Cyclic lower delay requisite	$0,1495$
Events: Control	$0,3056$
Augmented reality	$0,143$
Network Control	$0,0455$
Config. & diagnostic	$4,98e^{-06}$

Table 4.3: Traffic percentage for each type of service with 30% of Cyclic strict delay requisite services

Figure 4.19 shows the maximum utilization obtained as a function of percentage of traffic of the cyclic with strict delay requisite service when there is a percentage of 80% of realizations of each scenario in which a feasible solution was found.

As can be seen, for the first case (30%) the maximum utilization reached is 41.49% and is obtained for a maximum of 3500 flows while for the 36% case the maximum utilization is 35% and is obtained with 2200 flows, for the 62.35% case (original) the maximum utilization is 26.1% and is obtained with 1300, and for 80% the maximum utilization is 20.08% and is obtained with only 500 flows. This is due to the fact that the number of flows of cyclic with strict delay requisite service is much higher in the last case than

Type of services	**Traffic Percentage**
Cyclic strict delay requisite	$0,5$
Mobile Robots	$0,04$
Cyclic lower delay requisite	$0,1068$
Events: Control	$0,2184$
Augmented reality	$0,1022$
Network Control	$0,0325$
Config. & diagnostic	$3,56e^{-06}$

Table 4.4: Traffic percentage for each type of service with 50% of Cyclic strict delay requisite services

Type of services	**Traffic Percentage**
Cyclic strict delay requisite	$0,8$
Mobile Robots	$0,016$
Cyclic lower delay requisite	$0,0427$
Events: Control	$0,0874$
Augmented reality	$0,0409$
Network Control	$0,013$
Config. & diagnostic	$1,42e^{-06}$

Table 4.5: Traffic percentage for each type of service with 80% of cyclic strict delay requisite services

in the first, and therefore, the delay requirements to be met by the network are stricter, becoming unfeasible.

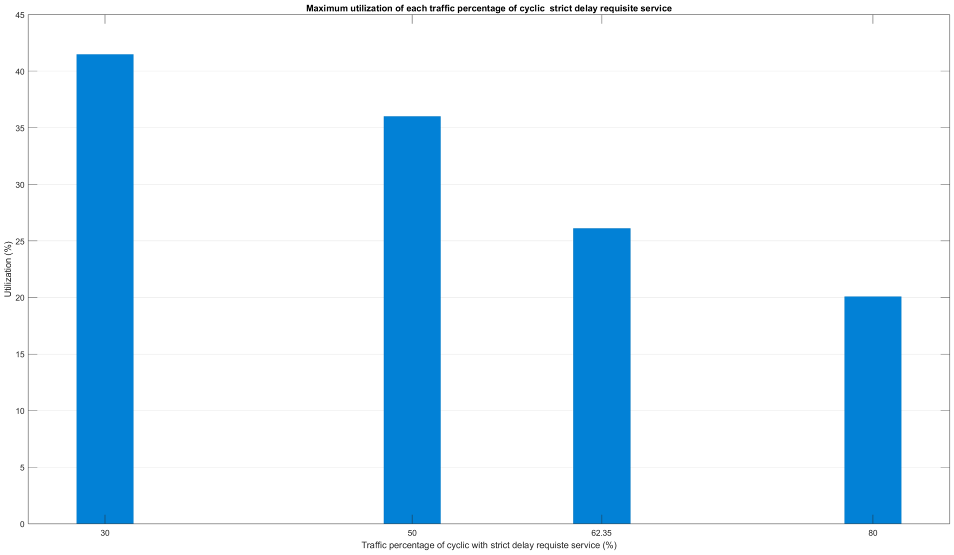


Figure 4.19: Utilization obtained according to the percentage of traffic

Therefore, we can deduce that as the amount of the cyclic with strict delay requisite service, critical in terms of delay requirement, increases, the maximum utilization obtained decreases. This is mainly due to the number of flows that the algorithm can find as a feasible solution is smaller and therefore the network utilization for critical industrial traffic decreases.

4.4.4 Scalability

Finally, the last experiment is detailed where the scalability of the proposed heuristic is tested. Specifically, the daisy chain topology has been implemented but with two routes, i.e., the same source node but different destination node. One of the routes has a larger number of ATSs instances that must pass the flows.

Note that having a larger network, i.e., with a greater number of ATSs, is different from flows passing through a greater number of ATS instances. Since if a flow passes through more ATSs instances, this implies that the delay requirement per ATS is reduced and therefore, the utilization is also reduced, since the delay requirement to be met in each ATS instance is stricter.

In this experiment again a sweep is performed on the number of flows from 100 to 1450 flows in 10-flow hops. In each of these scenarios the features of the flows are randomly selected within the range of Table 4.1 and 100 realizations of each scenario are run.

Shortest path

Figure 4.20 shows the daisy chain topology implemented for the simulation of this case. The source node of the flows is N1 while the destination node is N3, therefore, the flows must pass through 2 ATS instances to reach the destination.

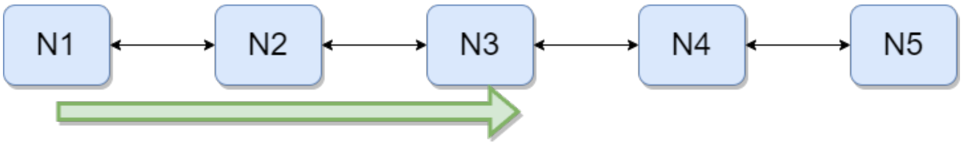


Figure 4.20: Daisy chain topology with the shortest path.

Figure 4.21 shows the prioritization generated by the prioritization by flows for the different scenarios. It is observed how the probability of obtaining a feasible solution decreases as the number of flows or link utilization increases. In this case, the maximum number of flows that allows to obtain an operating range of 80% of satisfactory solutions is reached around 1450 flows. In addition, it is obtained that the utilization of critical traffic has been increased to 30%.

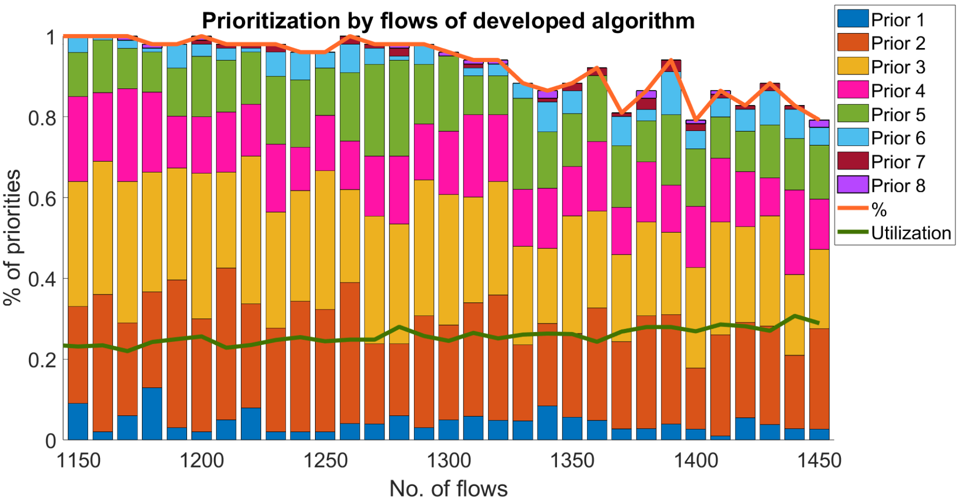


Figure 4.21: Priority assignment by flows in the daisy chain topology with the shortest path.

Figure 4.22 shows the average execution time exhibited by flow prioritization as a function of the number of flows to be prioritized for a shortest path. It is observed that the execution time increases exponentially with the number of flows but does not increase excessively.

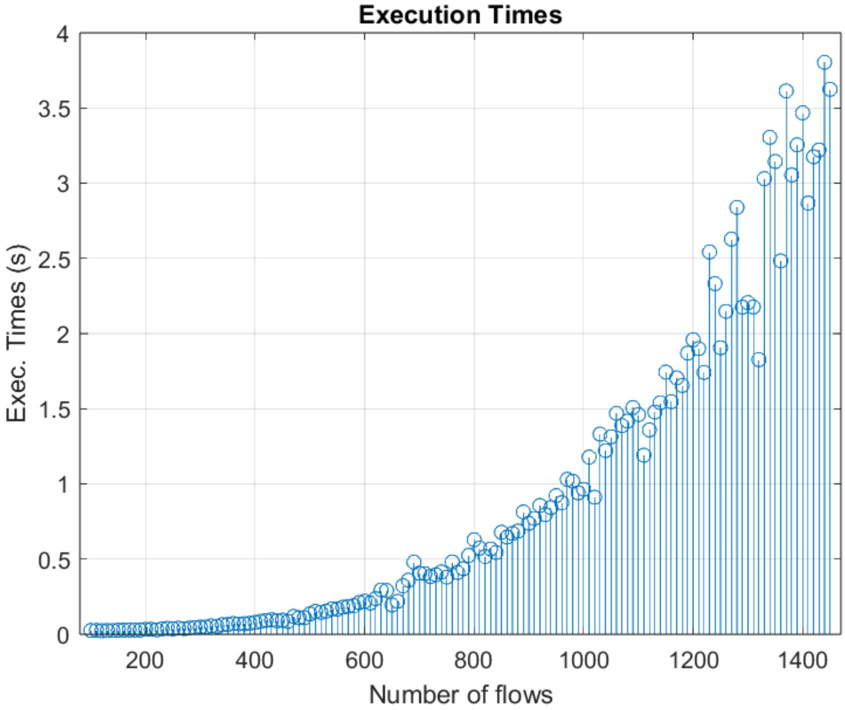


Figure 4.22: Daisy chain topology execution time with shortest path

Longest path

Figure 4.23 shows the daisy chain topology implemented in this simulation, where the source node of the flows is N1 and the destination node is N5. In total the flows must pass through 4 ATS instances to reach the destination. In this case, the flows must pass through two more ATS instances.

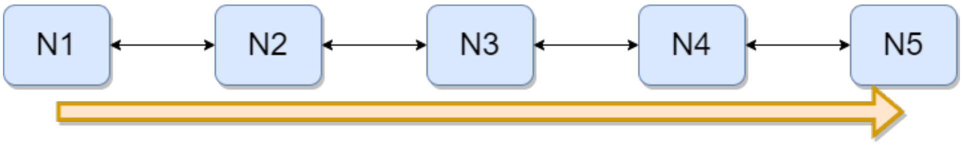


Figure 4.23: Daisy chain topology with the longest path

Figure 4.24 shows the prioritization generated by the flow prioritization algorithm for the different scenarios. As in the previous cases, it is observed that again the higher the number of flows in the scenario, i.e. the higher the utilization of the ATS link, the higher the probability of not finding a satisfactory solution, even if the number of priorities is increased. In contrast to the case of the shortest path, it is observed that for an operating range of 80% of satisfactory solutions the maximum possible number of flows in the scenario is 790 flows. Thus, the maximum utilization achieved is 20% of the link for critical traffic. The remaining link capacity can be used to transmit best effort traffic.

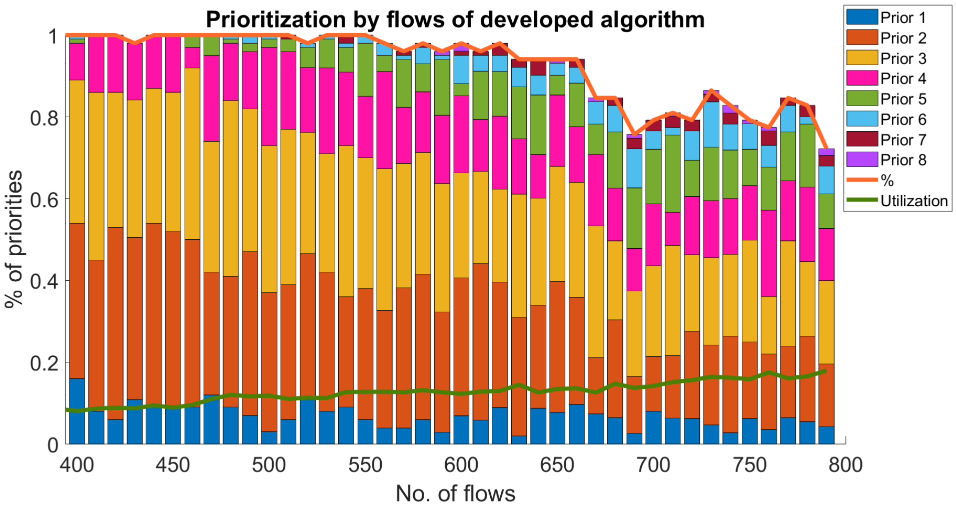


Figure 4.24: Priority assignment by flows in the daisy chain topology with the long path.

Figure 4.25 shows the average execution time exhibited by flow prioritization as a function of the number of flows to be prioritized for a longest path. As in the previous case, execution time increases exponentially with the number of flows but does not increase excessively.

Consequently, it follows that the case with a shorter route presents a higher number of flows to reach an 80% operating range due to the possibility of distributing the delay requirement among a smaller number of ATS instances, being more feasible to find a solution. Although the number of flows is approximately halved for the case of a longer path, it is still feasible and suitable for Industry 4.0. In addition, the execution time does not rise excessively when the number of flows in the network is increased for both a shorter and longer route. Therefore, it can be stated that the proposed solution scales correctly.

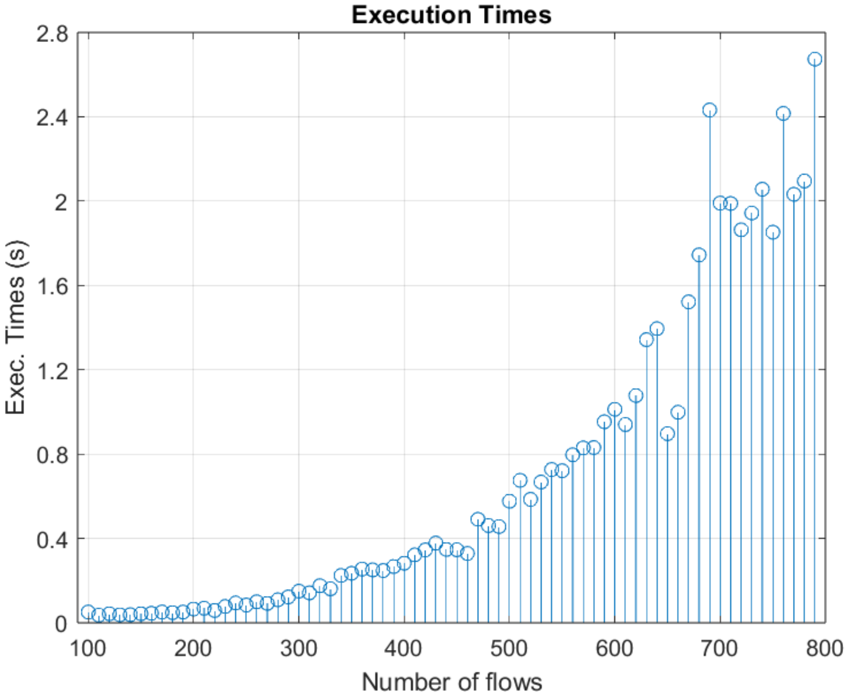


Figure 4.25: Daisy chain topology execution time with longest path

Chapter 5

Conclusions and future works

This chapter presents a summary of the objectives obtained and the conclusions reached after the analysis of the results. Finally, the lines of future work to be developed in this area for its improvement are mentioned.

5.1 Conclusions

To conclude, this paper has studied in depth the TSN protocol that guarantees deterministic traffic in the current industrial wired way, focusing on asynchronous TSN. We have also studied the performance and characteristics of 5G that makes it the preferred technology to meet the requirements of deterministic services demanded by Industry 4.0, thanks to the 5G URLLC service. Both TSN and 5G have been studied separately to check the advantages present in each of them in order to better understand the great trend for the integration of both to meet the requirements of the industry. Additionally, the current state of integration of both technologies for the industrial environment has been analyzed. This integration is mainly focused on the industrial environment as there is a need to automate processes to optimize their efficiency and achieve wireless flexibility of the devices allowing an increase in the volume of communications with a lower cost of deployment associated with the links between sensors.

Although the characteristics of 5G and the integration of 5G and TSN have been studied, this project has focused especially on analyzing the operation of asynchronous TSN networks and more specifically, the traffic scheduler used in asynchronous TSN (ATS) to understand the characteristics of these networks. The ATS is responsible for implementing flow routing in asynchronous TSN switches, which has several queued stages to carry out flow routing. However, to date, no mechanism has been studied that allows the optimization of the prioritization mechanism used or that obtains feasible results without great computational complexity, as can be seen in the various articles detailed in the section 2.4.2.

Therefore, this work has mainly focused on the investigation of the flow prioritization solutions proposed to date along with the proposal of a flow prioritization mechanism. This prioritization mechanism should minimize the number of priorities, meet the delay requirement demanded by the flows, scale correctly with the increase of the number of flows in the network and present a reduced complexity.

Likewise, we have studied the different existing mechanisms for delay distribution that have been developed to date. This delay distribution mechanism is necessary because the a priori known characteristics of the flows provide the maximum E2E delay that can be suffered, so it is necessary to distribute this delay among the different ATSs instances that the flow crosses. Among the possible solutions, the solution proposed in [14] has been chosen.

After defining the proposed prioritization heuristic and selecting the distribution mechanism, a simulator has been developed that implements both algorithms in order to verify the correct operation of prioritization and delay distribution defined in an asynchronous TSN network. For this purpose, several simulations have been performed in this test environment, testing the capacity and performance of the routing and scheduling of the ATS module against different types of services with critical delay requirements. From the performance of four experiments and the development of this project, the main conclusions that can be inferred from this project are as follows:

- The flow prioritization algorithm developed allows minimizing the number of priority levels of the network if a feasible solution exists. It has been verified both theoretically as detailed in section 3.4.4 and experimentally through the four experiments developed.

- The results of the first experiment demonstrate that our approach scales correctly, achieving a execution time that is six orders of magnitude less than that of brute force search and achieving the same prioritization results for both algorithms. Therefore, it follows that our algorithm is accurate and optimal.

- Prioritization by flow is able to find a feasible solution to a larger volume of traffic than prioritization by PCP. In other words, prioritization by flow has a higher utilization than by PCP.

- The prioritization results obtained by the three topologies are very similar among them, therefore, the network topology does not affect the scalability of the algorithm.

- The execution time obtained for a high volume of flows is relatively low, around 3s.
- The utilization varies depending on the amount of critical traffic with strict delay requirements, achieving a maximum value of 30% of the capacity of the ATS links when there is a lower number of cyclic with strict delay requisite service flows. The rest of the link capacity can be used to transmit best-effort traffic.
- The results provided by the last experiment reaffirm that the prioritization solution provided in this project scales well with increasing number of flows. That is, the proposed heuristic finds a feasible prioritization solution when increasing the number of flows with a high success rate for the different scenarios without excessive growth in execution time.

To date, the contributions of this project have been published in a conference paper titled “Flow Prioritization for TSN Asynchronous Traffic Shapers” [33] and a patent titled “Método de configuración de redes sensibles al retardo basadas en planificadores con conformación de tráfico asíncrono, y con calidad de servicio determinista”.

5.2 Future works

Despite the work developed to date is fully functional, there are still major challenges to be met. Some of the improvements to be developed to continue research in this field in order to achieve the desired integration between 5G and TSN to enable determinist traffic in Industry 4.0 are for example:

- Improve the simulator created by introducing new topologies and new types of traffics that have not been contemplated in this project.
- Develop a new delay distribution mechanism that takes into account the load that some ATSs instances may present and therefore affect the feasibility of meeting the delay requirement of the flows. That is, some flows must traverse a greater number of ATSs instances than others. Therefore, if some of these ATSs instances present a higher number of flows, the delay of the most critical traffic to be satisfied in these instances must be more lenient than in that ATS instance that presents a lower number of flows.
- Study the mapping between the different QoS of 5G traffic to the QoS present in the asynchronous TSN networks to allow the integration of TSN and 5G.

- Improve the utilization of critical traffic in order to obtain a greater amount of critical traffic.

- Integration of the asynchronous TSN network with the priority allocation algorithm developed as the transport network of the 5G network.

Appendix A

Conference Paper

Flow Prioritization for TSN Asynchronous Traffic Shapers

Julia Caleya-Sanchez*, Jonathan Prados-Garzon*, Lorena Chinchilla-Romero*, Pablo Muñoz *,
and Pablo Ameigeiras*
*Department of Signal Theory, Telematics and Communications (DTSTC), University of Granada, Spain
Emails: jcaleyas@ugr.es, jpg@ugr.es, lorenachinchilla@ugr.es, pabloml@ugr.es, pameigeiras@ugr.es

***Abstract*—Time-Sensitive Networking (TSN) technology is key to the development of current networks due to its capacity to provide a deterministic Quality of Service (QoS) mainly in terms of delay for different industrial traffic. It also simplifies management and improves the scalability of industrial networks. This articles focuses on Asynchronous Traffic Shape (ATS) for TSN. The key aspect of ATS is that by prioritizing flows delay requirement can be satisfied for each priority level. To that end, we formally formulate the problem of flow priority assignment in an network and we demonstrate the optimality of our proposed algorithm. We have compared our algorithm with the brute force search obtaining that the execution time of brute force is much higher than ours with exactly the same prioritization results.**

I. Introduction

Time-Sensitive Networking (TSN) is a set of layer 2 standards that are specified as a series of amendments to the IEEE 802.1Q standard. These standards solve critical challenges in various sectors by ensuring the deterministic transmission of flows with Quality of Service (QoS) in terms of strict requirements for latency, jitter, reliability and packet loss. Thanks to these capabilities, TSN technology is currently key to the development of deterministic networks, such as industrial or 5G networks.

TSN guarantees deterministic traffic transmission by the use of sophisticated and complex schedulers for the transmission of frames on the output ports of a TSN bridge. We can distinguish two types of schedulers defined in TSN standards: Asynchronous Traffic Shaper (ATS) and Time-Aware Shaper (TAS). In this case, we focus on asynchronous TSN networks, where a common and precise time reference is not necessary. The asynchronous TSN network uses the ATS, defined in IEEE 802.1Qcr. The ATS is based on the Urgency-Based Scheduler (UBS) proposed by Specht and Samii [1], which uses interleaved shaped queues to regulate traffic and a strict priority queue for traffic prioritization. In addition, ATS-based TSNs are more suitable for large-scale scenarios. Specifically, an ATS TSN is considered at each output port of the TSN bridge.

There are different alternatives that address the configuration of ATS-based TSN networks [1]–[3]. However, all of them present scalability problems, and it is necessary to find solutions that can handle and adapt correctly to the increase in traffic smoothly and without losing the QoS offered. Therefore, scalability is a key factor in order to be able to adapt correctly to the growing evolution of new services with more stringent requirements in terms of QoS. Additionally, due to the nature of the proposed solutions, these solutions are all computationally complex to implement.

In this work, a solution is proposed to solve the above problems, by a novel ATS-based TSN network configuration method, which minimizes the number of priority levels with respect to known techniques, under deterministic QoS requirements. The proposed solution attempts to find a feasible prioritization of a set of traffic flows in a single concrete ATS instance while fulfilling the delay requirements of the flows. This solution scales well with the increase in flows to be accommodated. That is, for a large number of flows, the proposed solution attempts to determine the feasible prioritization in an efficient way. The solution assumes that the requirements of the flows are previously known, being suitable, for example, for industrial networks where the types of traffic are known a priori. Additionally, the algorithm that develops the proposed solution presents a feasible and uncomplicated implementation and is valid for both online and offline solutions.

For evaluation purposes, we consider an industrial scenario with different types of traffic with different QoS requirements. An analysis of the degree of optimality and accuracy of the proposed algorithm is provided. Specifically, we demonstrate that proposed algorithm minimizes the number of priority levels required in an ATS instance, fulfilling the queuing delay requisites of the flows traversing the ATS instance. Furthermore, we compared the performance and flow prioritization by our algorithm with the exhaustive search of all possible configurations (brute force). The results show that our algorithm scales correctly, with execution times six orders of magnitude lower than brute force and with exactly equal prioritization results for both approaches.

The remainder of the paper is organized as follows: Section II review of the ATS description and existing work addressing the performance of an ATS-based network. Section III describes the system model and the prioritization problem and its formulation. Section IV defines the developed algorithm with its analysis and design principles. Section V provides the experimental results and section VI draws the conclusions.

This work has been partially funded by the H2020 research and innovation project 5G-CLARITY (Grant No. 871428), 6G-CHRONOS (TSI-063000-2021-28) and the Spanish Ministry of Universities (FPU Grant 21/04225).
ISBN 978-3-903176-57-7© 2023 IFIP

II. Background

A. ATS Description

The ATS defines an asynchronous method for handling frames on the TSN bridge output ports [4], [5]. The TSN standards [5], which specify the ATS, are based on the UBS proposed by Specht and Samii in [6]. In [6] is considered a leaky bucket in the asynchronous shapers for flow traffic regulation. The ATS can be a practical implementation of the UBS in 802.1Q standards [5]. In this work, we adopt the nomenclature used in [6].

The queuing model of the ATS is shown in Fig. 1 [3]. For simplicity, only one egress port is shown in Fig. 1, but there is an ATS instance for each egress port of the bridge. The ATS consists of two queuing stages: i) a set of shaped queues, which are First In, First Out (FIFO) queues with an interleaved regulator, for interleaved shaping and ii) a set of priority queues. In the first stage, interleaved shaping, to perform traffic control of a set of flows, each with its own requirements, the use of a single queue (shaped queue) can be employed. The use of shaped queues before strict priority queues avoids arbitrarily large worst-case delays, because the burstiness of the flows remains constant with each hop. However, it can lead to packet losses of flows. Remarkably, in [7] LeBoundec demonstrated that the Worst-Case Queuing Delay (WCQD) is not enhanced by the use of shaped queues in the ATS. That is, placing a minimal interleaved regulator after an arbitrary FIFO queue has no negative effect on the delay for the worst case combination. The second stage uses First Come, First Served (FCFS) queues with a strict priority transmission selection algorithm. In each of the queues, all the shaped queues with the same priority level are merged.

B. Related Works

This section overviews existing works related to the solutions proposed for the flow prioritization in asynchronous TSN networks [1]–[3], [8], [9].

In [1], Specht and Samii consider a Satisfiability Modulo Theories (SMT) solver to find a feasible configurations in ATS-based networks. They propose a Topology Rank Solver (TRS) heuristic to cope with the high complexity of the pure SMT solution. Nonetheless, TRS relies on SMT for flow prioritization in at least a single ATS instance. In [3], Prados *et al.* propose a solution combining heuristic and convex optimization to seek a long-term configuration of ATS-based TSN network. Specifically, the work in [3] addresses the problem formulated in [9] which aims to minimize the probability of flow rejection. Although the cited works are armed with heuristics methods to cope with computational complexity, they uses exact optimization method to address the flow prioritization in the ATS instance, which limits their scalability with the number of flows.

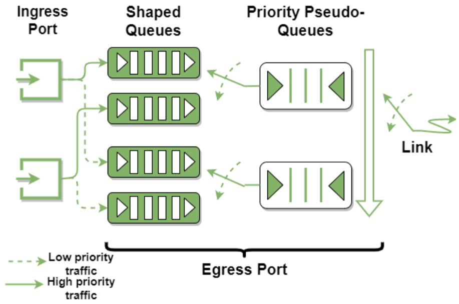


Fig. 1. ATS queuing model.

In [2] and [8], Prados *et al.* suggest an online approach based on Deep Reinforcement Learning (DRL) to determine the configuration of each flow as it arrives at the network. The requirements of the flows in this solutions are unknown and present a low capability, i.e., they depend on the network topology and have to be trained specifically for each scenario which leads to a large training time. Additionally, these works do not include a model of the flow allocation problem in asynchronous TSN networks. Moreover, all the solutions proposed are complex to implement.

In this work, unlike [2], [3], [8], [9], the proposed solution considers known flow characteristics and requirements, which is the common situation in many scenarios such as industrial networks. Furthermore, unlike the exact optimization methods considered in [1], [3], it is scalable as it allows to increase the number of flows in the scenario proving that the computational time complexity is maintained while providing a feasible prioritization. Last, remarkably, the proposed algorithm is easier to implement.

III. System Model and Problem Formulation

In this section, first we describe the considered system model. and then we formally formulate the ATS flow prioritization problem addressed in this work.

A. System Model

Let us consider an asynchronous TSN network comprising a set of ATS-based TSN bridges. There are a set of delay-sensitive flows to be conveyed through the network whose traffic is constrained by $r \cdot t + b$ [10], where r and b are the committed data rate and burst size (burstiness), respectively. The number of flows, their respective traffic features r and b, and their end-to-end (E2E) delay requisite are known beforehand. This is the common situation in industrial networks. For instance, we might have critical flows to communicate alarm events, control the motion of the operational technology devices, and steer the mobile robots through the factory floor. Each flow follows a specific path in the TSN network and its E2E delay requisite is somehow distributed among the different hops of the path.

Each TSN bridge includes an ATS instance at every egress port to handle the packets transmission at the link according to the operation described in the previous section. The ATS instances include P priority levels and enough shaped queues as to use all the priority levels regardless of the asynchronous TSN network configuration (e.g., number of ingress ports at the respective TSN bridge and prioritization considered in the previous hop). Without loss of generality, we assume that each priority level is associated with an integer index p and lower indexes mean higher priority levels. In this way, priority 1 is the level with the highest priority. The priority levels 1 to $P-1$ are reserved to accommodate delay-sensitive traffic, whereas the priority level P is destined for best-effort traffic (e.g., remote access and maintenance in manufacturing [11]).

Given the per-flow chosen paths and delay requisite distribution among hops, there is a set of delay-sensitive flows $\mathcal{F}$ to be prioritized at each ATS instance. We assume there is a worst-case delay requisite, denoted as D_f, for each flow $f \in \mathcal{F}$ at the respective ATS instance. Thus, we can define the WCQD requisite at the ATS instance for flow f as $R_f = D_f - l_f/C$, where l_f is the maximum frame size of the flow f and C is the nominal capacity of the link handled by the ATS instance. Without loss of generality, we assume that each flow $f \in \mathcal{F}$ is associated with an integer index i according to its WCQD requisite R_f, also denoted as R_i, $R_i = R_f$. Specifically, lower indexes mean more stringent WCQD constraints, i.e., $R_{i-1} \leq R_i \leq R_{i+1}$ being the flows with indexes 1 and $F = |\mathcal{F}|$ those with the most stringent and most lenient WCQD requisites, respectively.

Let $\mathcal{F}_p$ be the set of flows allocated to a priority level p. The WCQD Q_p experienced by every flow allocated to a priority level p is upper bounded as follows [6], [7], [12]:

$$Q_p = \frac{\sum_{\forall f \in \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_p} b_f + \max_{\forall f \in \mathcal{F}_{p+1} \cup \ldots \cup \mathcal{F}_8} l_f}{C - \sum_{\forall f \in \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_{p-1}} r_f} \qquad (1)$$

where r_f and b_f are the committed data rate and committed burst size (burstiness) for the flow f, respectively. Please find in Table I the primary notation considered in this work.

B. Problem Statement and Formulation

The problem addressed in this work consists in finding a feasible or satisfiable prioritization for $\mathcal{F}$ at the respective ATS instance, i.e., the delay requisites $\forall\, f \in \mathcal{F}$ are met, to minimize the number of priority levels used in it. Below is the formal formulation of the stated problem:

$$minimize \left\{ \begin{array}{l} \max\limits_{\forall f \in \mathcal{F}} P_f \,:\, P_f \in [1, 1-P] \cap \mathbb{N}\; \forall f \in \mathcal{F}\, (C1); \\ Q_p \leq R_f \;\forall\, f \in \mathcal{F}_p,\, p \in [1, P-1]\, (C2); \\ \sum\limits_{\forall\, f \in \mathcal{F}} r_f \leq C\, (C3). \end{array} \right\} \qquad (2)$$

where $\mathbb{N}$ is the set of natural numbers. The decision variables are P_f which denotes the priority level assigned to flow $f \in \mathcal{F}$. This variables are integer and take values in the available priority levels for delay-sensitive traffic in the corresponding ATS instance, as specified in constraint *C1*.

The objective of the problem above is to minimize the required number of priority levels in the ATS instance. The motivation of choosing this optimization goal is because the cost of the asynchronous TSN network directly depends on the available priority levels in the ATS instances. The higher the number of available priorities is, the higher the deployment costs (capital expenditures) as the ATS-based TSN bridge's price raises. Moreover, it is easier to configure and operate an asynchronous TSN network whose ATS instances have a lower number of priority levels.

TABLE I
PRIMARY NOTATION

Notation	Description
$\mathcal{F}$	Set of flows to be prioritized in the ATS instance.
$\mathcal{F}_p$	Set including all the flows currently allocated to priority level p in the target ATS instance.
C	Nominal link capacity of the target ATS instance.
r_f, b_f, and l_f	Committed data rate, committed burst size, and maximum frame size of the flow f, respectively.
R_f and D_f	WCQD and delay requisites for the flow f at the target ATS instance, being $R_f = D_f - l_f/C$.
Q_p and Q_f	WCQD of priority level p and experienced by flow f.
R_1 and R_F	The most stringent and most lenient WCQD requisites among all the flows $\mathcal{F}$.
P	Maximum number of priority levels (queues) available in the target ATS instance.

Regarding the primary constraints, we must ensure that the aggregated rate traversing the ATS instance is lower than the nominal capacity (*C3*). In fact, this technological constraint is a primary assumption to derive (1) [6], [7]. On the other side, the WCQD requisites for all the flows has to be met (*C2*).

IV. ATS Flow Prioritization Algorithm

A. Design Principles

Let us start introducing some relevant propositions that can be directly proven from (1) and are behind the rationale of the proposed algorithm. Also, these propositions are cornerstone for assisting the proof of the correctness and degree of optimality of our proposal.

Proposition 1: The WCQD Q_1 for the first priority level (highest priority) is given by $Q_1 = \sum_{\forall\, f \in \mathcal{F}} b_f/C$ when there is a single priority level or $Q_1 = (\sum_{\forall\, f \in \mathcal{F}_1} b_f - max_{\forall f \in \mathcal{F} \setminus \mathcal{F}_1} l_f)/C$ when there are two or more priority levels. Moreover, Q_1 is the lowest WCQD in the ATS instance, i.e., $Q_1 < Q_p \,\forall\, p \in [2, P]$.

Proof: Q_1 can be directly derived from (1). From (1), the aggregated burstiness of any priority level $p \in [2, Q]$ will include the aggregated burstiness of level 1 and by definition $l_f \leq b_f \forall f \in \mathcal{F}$. Also, the effective capacity of $p \in [2, Q]$ is reduced by the aggregated committed rate in level 1. Then, it always holds that $Q_1 < Q_p \,\forall\, p \in [2, P]$. ■

Proposition 2: Decreasing one priority level from p to $p+1$ of any flow f will increase its WCQD, but reduce or does not affect the WCQD of the rest of flows. Equivalently, increasing the priority level of any flow f will decrease its WCQD, but increase or does not affect the WCQD of the rest of flows.

Proof: From (1), lowering the priority level of a flow f will reduce the aggregated burstiness of the priority level p it was originally accommodated by b_f. Since by definition

$b_f \geq l_f^{(max)}$, the WCQD of p is reduced. For the new priority level $p+1$ of f, the aggregated burst size will remain the same, but its effective capacity $C - \sum_{k=1}^{p} r^{(k)}$ will increase by r_f, thus, decreasing the WCQD of $p+1$. For priority levels $k > p+1$ or $k < p$, the WCQD does not change. On the other hand, the maximum aggregated burst size seen by f remains the same, but its effective capacity is reduced by $\sum_{f \in \mathcal{F}_p} r_f$, thus increasing its WCQD. Last, increasing the priority level of a flow f is equivalent, in terms of the resulting WCQDs experienced by the flows, to keep the same priority level for f and decrease one priority level for the rest of the flows. ■

Proposition 3: If, in the highest priority level ($p = 1$), the most lenient WCQD requisite $R_{f_{F_1}}$, i.e., $R_{f_{F_1}} \geq R_f \, \forall \, f \in \mathcal{F}_1$, which is imposed by the flow $f_{F_1} = f_{|\mathcal{F}_1|}$, is not fulfilled, i.e., $Q_1 > R_{f_{F_1}}$, then, problem (2) has no satisfiable solution.

Proof: From *Proposition 1*, $Q_1 < Q_p \, \forall \, p \in [2, P]$. From *Proposition 2*, decreasing the priority level of f_{F1} will increase its WCQD. On the other hand, decreasing the priority level of any flow $f \in \mathcal{F}_1 \, s.t. \, f \neq f_{F1}$ to reduce the Q_1 is neither possible because the WCQD of f will increase (*Proposition 2*) and from the premises of the proposition $R_{f_{F_1}} \geq R_f$, thus, R_f would not be met. ■

Proposition 4: If currently $Q_p \leq R_f \, \forall \, f \in \mathcal{F}_p$ and $\forall \, p \in [2, P-M]$, $\mathcal{F}_p == \emptyset \, \forall \, p \in [P-M+1, P]$, and we decrease M priority levels $\forall \, f \in \mathcal{F} \setminus \mathcal{F}_1$, then, we can freely distribute the flows originally allocated to $p = 1$ among the levels $p \in [1, M+1]$ and the requisites of the rest of the flows will be still met, i.e., $Q_p \leq R_f \, \forall \, f \in \mathcal{F}_p$ and $\forall \, p \in [2+M, P]$.

Proof: From (1), decreasing M priority levels for all the flows originally allocated in $p \in [2, P-M]$ will keep the same WCQD for them as $\mathcal{F}_p == \emptyset \, \forall \, p \in [P-M+1, P]$. Then, no matter the prioritization we consider for the flows originally allocated in $p = 1$ among the levels $p \in [1, M+1]$, also from (1), the WCQD will remain the same for all the flows originally allocated in $p \in [2, P-M]$. ■

B. Algorithm

The proposed ATS flow prioritization algorithm is shown in Algorithm 1. The goal of the algorithm is to find a satisfiable prioritization, if at least one exists, for the set of flows $\mathcal{F}$ at a given ATS instance according to the optimization program (2). To that end, it iterates ($lines$ $7-20$) until either a feasible solution is found, i.e., the delay requisites for all the flows are met while the link utilization is lower than 100% ($line$ 10), or the problem infeasibility is determined ($line$ 17). Please refer to *Propositions 1* and *3* for the rationale behind the latter algorithm exit condition.

At each iteration, first, the algorithm checks whether the WCQD requisites for all the flows allocated to the second priority level $\mathcal{F}_2$ are met ($line$ 8). Please note that this condition is always met at the very first iteration as $\mathcal{F}_2$ initially equals the empty set. If the condition is met, which is verified using (1), the algorithm checks, again using (1), whether the WCQD requisites for all the flows allocated to $\mathcal{F}_1$ are met. If so, a feasible solution is found ($line$ 10) and the algorithm finishes. Otherwise, the algorithm creates a new set $\mathcal{F}_k$ if needed and decreases the priority level for all the flows by one, leaving the set $\mathcal{F}_1$ empty ($lines$ $12-13$). We refer hereinafter to this process as partition k or kth. The reason to follow the operation described above is that, once the algorithm finds a satisfiable prioritization for the flows allocated to the current priority levels 2 to k, the highest priority level can be further partitioned to find a feasible solution without affecting the WCQDs of the current priority levels 2 to k (*Proposition 4*).

If conditions in $line$ 8 or $line$ 9 are not met, then, the algorithm moves the flow f^* with the most stringent WCQD requisite currently in priority 2 to priority 1, i.e., it increases the priority of f^* ($lines$ $15-16$). Nonetheless, if it turns out that f^* is the last flow in $\mathcal{F}_2$, the algorithm realizes that the problem has no solution ($line$ 17).

C. Algorithm Analysis

This section includes the analysis of the proposed prioritization algorithm for ATSs detailed in Section IV-B. More precisely, we rely on the principle of mathematical induction to formally prove the theorem stated next.

Theorem 1: Algorithm 1 finds always a satisfiable solution for the ATS prioritization problem (2) if any exists and the solution found is optimal for that problem, i.e., it minimizes the number of priority levels used in the ATS instance.

Proof: Let k denote the current partition index as in Algorithm 1 to find a solution for prioritizing a set of TSN flows $\mathcal{F}$. As previously defined, a partition is the process carried out by Algorithm 1 for partitioning the current highest priority level into two. That is, decreasing the priority level of all the flows by one and, after, moving as many flows from $p = 2$ to $p = 1$ according to Algorithm 1 (see $lines$ $7-20$). Last, observe that for $k = 1$ Algorithm 1 just checks whether a satisfiable prioritization exists for a single priority level, i.e.,

Algorithm 1 ATS Prioritization Algorithm

```
 1: $Problem\_Solved = 1$;                                   ▷ $BC\_O1$
 2: $No\_Solution = 2$;                                      ▷ $BC\_O2$
 3: $Searching\_Solution = 3$;                               ▷ $BC\_O3$
 4: $Initialize\ \mathcal{F}_1 \leftarrow \mathcal{F}$; $\mathcal{F}_2 \leftarrow \emptyset$; $k = 1$;
 5: function $PrioritizeFlows(\mathcal{F}_1, \mathcal{F}_2, k)$
 6:     $prob\_status = Searching\_Solution$;
 7:     while $prob\_status == Searching\_Solution$ do
 8:         if $Q_2 \leq R_f \ \forall f \in \mathcal{F}_2$ then
 9:             if $Q_1 \leq R_f \ \forall f \in \mathcal{F}_1$ then
10:                 return $Problem\_Solved$;
11:             end if
12:             $k++$; $\mathcal{F}_k = \{\}$;
13:             $\mathcal{F}_p \leftarrow \mathcal{F}_{p-1} \, \forall \, p = [2, k]$; $\mathcal{F}_1 \leftarrow \emptyset$;
14:         end if
15:         $f^* \leftarrow \underset{f \in \mathcal{F}_2}{\arg\min} \ R_f$;
16:         $\mathcal{F}_2 \leftarrow \mathcal{F}_2 \setminus f^*$; $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{f^*\}$;
17:         if $\mathcal{F}_2 == \emptyset$ then
18:             return $No\_Solution$;
19:         end if
20:     end while
21: end function
```

$Q_{spl} \leq R_1$. Trivially, if Algorithm 1 finds a solution for $k = 1$, the main hypothesis holds true.

BASE CASE (k=2): For the base case Algorithm 1 may result in three possible outcomes: i) a satisfiable solution is found for two priority levels (BC_O1) ii) the prioritization problem has no solution (BC_O2), and iii) further partitions are required to find a potential feasible prioritization (BC_O3). For BC_O1, the hypothesis holds true as Algorithm 1 has previously explored the single priority level configuration and determined it is unfeasible. For $k = 2$, output BC_O2 is issued when the flow f_F verifying that $R_{f_F} \geq R_f \, \forall f \in \mathcal{F}$ cannot be even accommodated in a single priority level. That is because, in $k = 1$, the algorithm could not accommodate all the flows in a single priority level and in $k = 2$ the algorithm has move all the flows to $p = 1$ except f_F without fulfilling R_{f_F} (see in *lines 8* and *17*). Then, from *Proposition 3*, the problem (2) has no solution and the hypothesis still holds true.

It remains to show that for BC_O3 no satisfiable solution for two priority levels exists. For this output, the WCQD requisite of f_1 (the most stringent constraint), allocated to $p = 1$, is not fulfilled as $R_{f_1} \leq R_f \, \forall f \in \mathcal{F}$ and, in $k = 2$, BC_O3 happens when condition in *line 9* is unfulfilled. From *Proposition 2*, increasing the priority of any of the flows allocated to $p = 2$ only contributes to the nonfulfillment of R_1. On the other hand, if moving any of the flows $f \neq f_1$ in $p = 1$ to $p = 2$ would result in a feasible solution, then, from *Proposition 2*, decreasing the priority of any other flow s in $p = 1$ verifying that $R_s \geq R_f$ will also result in a feasible solution. Last, observe that Algorithm 1 accommodates as many flows with the most lenient requisites as possible in $p = 2$ at each partition, thus minimizing Q_1. Considering all above, we can conclude that no satisfiable solution exists for two priority levels if the Algorithm 1 status is BC_O3 at the end of $k = 2$. Then, the hypothesis still holds true.

INDUCTION CASE (k=n+1): For partition n, Algorithm 1 has distributed the flows with the most lenient requisites in $p = [2, n]$ and their requisites are met. Then, if the requisite of f_1, accommodated in $p = 1$, is not met further partitions are needed. The induction method allows us to assume that the hypothesis holds true for iteration n, i.e., the prioritization of the flows in $p \in [2, n]$ is satisfiable and it is the one requiring the minimum number of priority levels (induction hypothesis). For $k + 1$, similar to the "BASE CASE", Algorithm 1 will keep partitioning $p = 1$ in search of a feasible solution. From *Proposition 4*, the analysis carried out for the "BASE CASE" is also valid for the "INDUCTION CASE". Considering this fact together with the induction hypothesis, we can conclude that the main hypothesis holds true. Observe, after $k = n + 1$, Algorithm 1 might keep making partitions until letting f_1 alone in $p = 1$. At that point, the feasibility of the problem is easily checked from *Proposition 3*. ■

Last, note that Algorithm 1 might output a feasible prioritization that requires a number of priority levels higher than the one supported by the respective ATS instance. In this case, the solution would be unfeasible due to the aforementioned constraint. However, that does not affect the analysis.

V. Results

In this section we provide the experimental results to show the scalability, optimality, and correctness of our proposal.

A. Experimental Setup

First, we devised a compound traffic model, summarized in Table II, based on [11], [13], [14] and references therein to realistically capture in our experiments the typical per-flow traffic demands and delay requisites in industrial scenarios. Each performed experiment consisted in finding the flow prioritization in an ATS instance for $F = |\mathcal{F}|$ flows. The prioritization problem was solved using the brute force algorithm and Algorithm 1 for comparison. Brute force consists of checking all the possible prioritizations for the flows in $\mathcal{F}$, and selecting that one requiring the minimum number of priority levels. Both algorithms were developed in Matlab and run in a server with Intel(R) Core(TM) i7-6700K Central Processing Unit (CPU) at 4.00GHz with 4 cores and 32 GB of RAM. Since problem (2) fast becomes intractable with the scenario scale when it is solved using brute force, we decided to do a per IEEE 802.1Q Traffic Class (TC) prioritization. Specifically, each service in Table II was mapped onto a TC and the respective Priority Code Point (PCP) (see sixth column in Table II). For each experiment, the number of flows $F^{(q)}$ considered for each TC q was proportional to its per-flow expected committed data rate (second column in Table II), i.e., $F^{(q)} = E[r_q]/(\sum_{k=1}^{7} E[r_k]) \cdot F$. Last, the traffic characteristics and delay requisite of each individual flow were uniformly sampled from the ranges provided in Table II according to the TC it belongs to. Please observe that the procedure described above results in seven TCs to be prioritized, each characterized by: i) a committed data rate and a burst size that correspond to the sum of the committed data rate and burstiness of all of its flows, respectively; ii) a maximum frame size which is determined as the maximum frame size among all of its flows; and iii) a delay requisite established as the most stringent delay requirement among all the flows. Assuming the traffic conforms to the committed data rate and burstiness the ATS produces zero packet losses. For

TABLE II
PER-SERVICE FLOW CHARACTERISTICS.

Services	r_q (Mbps)	b_q (packet)	D_q (ms)	l_Q (KBytes)	**PCP**
Cyclic-Synchronous	$8-0.8$	$1-4$	$1-0.5$	$1-0.05$	6
Mobile Robots	< 10	$1-4$	$500-1$	$0.25-0.04$	3
Cyclic-Asynchronous	$0.2 - 4e^{-3}$	$1-4$	$20-2$	$1-0.05$	5
Events: Control	> 12	$1-4$	$50-10$	$0.2-0.1$	4
Augmented Reality	$20 - 10$	$1-4$	10	$1.5-0.03$	2
Network Control	$8e^{-3}-4e^{-3}$	$1-4$	$1000 - 50$	$0.5-0.05$	7
Config. & Diagnostics	2	$1-4$	$100-10$	$1.5-0.5$	1

each value of F considered, we executed 100 independent realizations with different flow characteristics in each realizations. The execution times measurements reported are the average of all those runs resulting in the same number of TCs.

B. *Performance Evaluation*

Fig. 2 depicts the comparison of the average execution time exhibited by the brute force (labeled as ‘Brute Force’) and our heuristic-based (labeled as ‘Developed algorithm’) algorithms as a function of the number of TCs to be prioritized.

As observed, for the considered range of TCs, the results show that the execution time of the brute force algorithm exhibits an exponential growth whereas the our proposal scales well. For instance, for prioritizing seven TCs the brute force’s execution time is six orders of magnitude higher than our proposal. This makes unfeasible to use the brute force algorithm to carry out a per-flow prioritization in the ATS.

Fig. 3 depicts the prioritization outputted by our solution for different scenarios. Each scenario includes a given number of flows (x-axis) grouped into TCs as explained in the previous subsection. For each value of F, 100 independent realizations were carried out, each sampling the flow features according to the ranges provided in Table II. The line labeled as ’%’ represents the number of realizations of each scenario in which a feasible solution were found. Similarly, each bar, labeled as ’Prior P’, represents the percentage of realizations requiring at least P priority levels. As expected, the higher the number of flows in the scenario, which translates into higher utilization of the ATS link, the higher the probability of not finding a satisfiable solution, even if the number of priority levels is increased. Similarly, the minimum number of priority levels required increases with the traffic load. The most remarkable result of this experiment is that both our proposal and the brute force algorithm outputted exactly the same prioritization for all the experiments, thus validating our proposal’s operation and optimality. These results support that our solution finds a satisfiable solution if it exists and that solution requires the lowest number of priority levels as stated in Section IV.

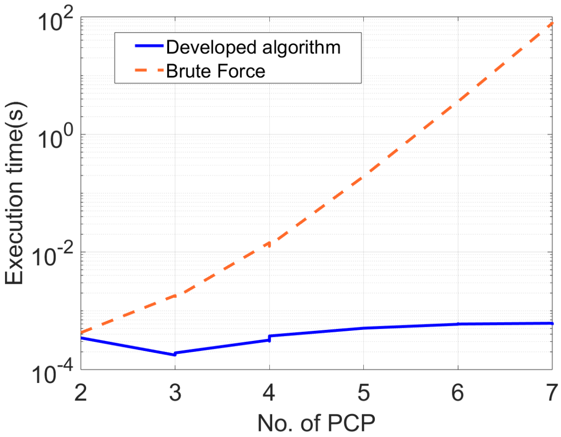


Fig. 2. Algorithm execution time.

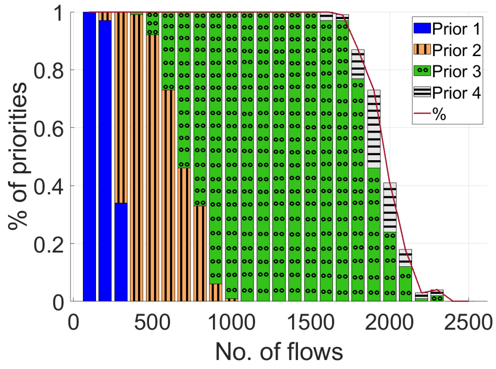


Fig. 3. Priority assignment for different flows.

VI. Conclusion

In this paper, the ability of the developed prioritization algorithm to obtain a feasible solution with the lowest WCQD in TSN has been evaluated. We have formally formulated the prioritization problem and the resolution through our proposal. Our algorithm and brute force have been compared with industrial traffic. The outcomes demonstrate that our approach scales correctly, achieving an execution time that is six orders of magnitude less than that of brute force and achieving the same prioritizing outcomes for both algorithms.

References

[1] J. Specht and S. Samii, “Synthesis of queue and priority assignment for asynchronous traffic shaping in switched ethernet,” in *2017 IEEE Real-Time Systems Symposium (RTSS)*, 2017, pp. 178–187.

[2] J. Prados-Garzon, T. Taleb, and M. Bagaa, “Learnet: Reinforcement learning based flow scheduling for asynchronous deterministic networks,” in *2020 IEEE Int. Conf. on Commun. (ICC)*, 2020, pp. 1–6.

[3] J. Prados-Garzon, T. Taleb, and M. Bagaa, “Optimization of flow allocation in asynchronous deterministic 5g transport networks by leveraging data analytics,” *IEEE Trans. Mob. Comput.*, pp. 1–1, 2021.

[4] A. Nasrallah *et al.*, “Ultra-low latency (ull) networks: The ieee tsn and ietf detnet standards and related 5g ull research,” *IEEE Commun. Surveys Tutorials*, vol. 21, no. 1, pp. 88–145, Firstquarter 2019.

[5] “Ieee draft standard for local and metropolitan area networks–media access control (mac) bridges and virtual bridged local area networks amendment: Asynchronous traffic shaping,” *IEEE P802. 1Qcr/D2.1*, pp. 1–152, Feb. 2020.

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

[7] J.-Y. Le Boudec, “A theory of traffic regulators for deterministic networks with application to interleaved regulators,” *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2721–2733, Dec. 2018.

[8] J. Prados-Garzon and T. Taleb, “Asynchronous time-sensitive networking for 5g backhauling,” *IEEE Netw.*, vol. 35, no. 2, pp. 144–151, 2021.

[9] J. Prados-Garzon, L. Chinchilla-Romero, P. Ameigeiras, P. Muñoz, and J. M. Lopez-Soler, “Asynchronous time-sensitive networking for industrial networks,” in *2021 Joint European Conf. on Netw. and Commun. & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 130–135.

[10] J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet*. Springer, 2001.

[11] 3GPP TS22.104 V17.4.0. (2020) Service Requirements for Cyber-Physical Control Applications in Vertical Domains.

[12] “IEEE Draft Standard for Local and metropolitan area networks–Bridges and Bridged Networks Amendment: Asynchronous Traffic Shaping,” *IEEE P802.1Qcr/D2.1, Feb. 2020*, pp. 1–152, 2020.

[13] “Integration of 5g with time sensitive networking for industrial communications,” White Paper, 5G ACIA, Feb. 2021.

[14] “A 5g traffic model for industrial use cases,” White Paper, 5G ACIA, Nov. 2019.

Appendix B

Patent

Bibliography

[1] UpKeep. ¿cuál es la diferencia entre Industria 3.0 e Industria 4.0? [Online]. Available: https://www.upkeep.com/es/learning/industry-3-0-vs-industry-4-0/

[2] 5G-ACIA, "Integration of 5G with time-sensitive networking for industrial communications," 5G Alliance for Connected Industries and Automation (5G-ACIA), Tech. Rep., 2020.

[3] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 905–929, 2020.

[4] Observatorio Nacional 5G. El 3GPP fija las prioridades de 5G advanced, previéndose aprobar las especificaciones en 2024. [Online]. Available: https://on5g.es/el-3gpp-fija-las-prioridades-de-5g-advanced-previendose-aprobar-las-especificaciones-en-2024/

[5] 3GPP TR 21.915 V15.0.0. (2019) Digital cellular telecommunications system (phase 2+) (GSM); universal mobile telecommunications system (UMTS); LTE; 5G; Release 15. [Online]. Available: https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

[6] (2019, 5) ¿Qué es la arquitectura de la tecnología 5G? [Online]. Available: https://www.viavisolutions.com/es-mx/5g-architecture

[7] 3GPP TS 23.501 V16.6.0. (2020) System architecture for the 5G system (5GS) (Release 16). [Online]. Available: https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

[8] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, pp. 80–87, 5 2017.

[9] Conetronica. Despliegue a gran escala de la tecnología 5G mmwave. [Online]. Available: https://www.conectronica.com/noticias/despliegue-a-gran-escala-de-la-tecnologia-5g-mmwave

[10] X. Lin, J. Li, R. Baldemair, J.-F. T. Cheng, S. Parkvall, D. C. Larsson, H. Koorapaty, M. Frenne, S. Falahati, A. Grovlen, and K. Werner, "5g new radio: Unveiling the essentials of the next generation wireless access technology," *IEEE Communications Standards Magazine*, vol. 3, pp. 30–37, 2019.

[11] J. Caleya-Sanchez, "Integración de 5G y TSN en redes privadas industriales," Master's thesis, Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada, Granada, 09 2021.

[12] J. Farkas, B. Varga, G. Miklós, and J. Sachs, "5G-TSN integration meets networking requirements for industrial automation," Ericsson, Tech. Rep., 2019.

[13] J. Prados-Garzon and T. Taleb, "Asynchronous time-sensitive networking for 5G backhauling," *IEEE Network*, vol. 35, pp. 144–151, 2021.

[14] J. Prados-Garzon, T. Taleb, and M. Bagaa, "Optimization of flow allocation in asynchronous deterministic 5G transport networks by leveraging data analytics," *IEEE Transactions on Mobile Computing*, vol. 22, pp. 1672–1687, 3 2023.

[15] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5G evolution: A view on 5G cellular technology beyond 3GPP release 15," *IEEE Access*, vol. 7, pp. 127 639–127 651, 2019.

[16] T. Kolding, P. Andres, K. Niemela, T. Jacobsen, S. Ganesan, R. Abreu, B. Bertenyi, and D. Chandramouli, "5G time service white paper," Nokia Bell Labs, Tech. Rep., 2021.

[17] Y. Wei, "The role of 5G in private networks for industrial IoT," Qualcomm, Tech. Rep., 2019.

[18] MarketsAndMarkets, "Time-sensitive networking market by type (IEEE 802.1 AS, IEEE 802.1 Qbv, IEEE 802.1 CB, IEEE 802.1 Qbu), component (switches, hubs routers & gateways, controllers & processors, isolators & convertors), end user, region - global forecast to 2028," MarketsAndMarkets, Tech. Rep., 2022.

[19] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh, "5G URLLC: Design challenges and system concepts," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1–6.

[20] M. Peshkin and J. E. Colgate, "Cobots," *Industrial Robot: An International Journal*, vol. 26, pp. 335–341, 1 1999. [Online]. Available: https://doi.org/10.1108/01439919910283722

[21] Time-sensitive networking (TSN) task group. [Online]. Available: https://1.ieee802.org/tsn/

[22] J. Specht and S. Samii, "Synthesis of queue and priority assignment for asynchronous traffic shaping in switched ethernet," in *2017 IEEE Real-Time Systems Symposium (RTSS)*, 6 2017, pp. 178–187.

[23] J. Prados-Garzon, T. Taleb, and M. Bagaa, "Learnet: Reinforcement learning based flow scheduling for asynchronous deterministic networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 6 2020, pp. 1–6.

[24] J. Prados-Garzon, L. Chinchilla-Romero, P. Ameigeiras, P. Muñoz, and J. M. Lopez-Soler, "Asynchronous time-sensitive networking for industrial networks," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 130–135.

[25] Asus zenbook. [Online]. Available: https://www.asus.com/es/laptops/for-home/zenbook/zenbook-14-um425/techspec/

[26] Windows 10. [Online]. Available: https://www.microsoft.com/es-es/software-download/windows10

[27] Matlab. [Online]. Available: https://es.mathworks.com/pricing-licensing.html?prodcode=ML&intendeduse=comm

[28] Ganttpro website. [Online]. Available: https://ganttpro.com/es/?_ga=2.174420514.1676861969.1688046507-144406673.1688046507

[29] Overleaf website. [Online]. Available: https://es.overleaf.com/project

[30] Scopus website. [Online]. Available: https://www.scopus.com/home.uri

[31] Google scholar website. [Online]. Available: https://scholar.google.es/

[32] Reseach gate website. [Online]. Available: https://www.researchgate.net/

[33] J. Caleya-Sanchez, J. Prados-Garzon, L. Chinchilla-Romero, P. Muñoz, and P. Ameigeiras, "Flow prioritization for TSN asynchronous traffic shapers," in *2023 IFIP Networking Conference (IFIP Networking)*, 2023, pp. 1–6.

[34] B. B. Sánchez, R. P. A. Garrido, D. de Rivera, and Álvaro Sánchez Picot, "Enhancing process control in industry 4.0 scenarios using cyber-physical systems," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 7, pp. 41–64, 2016. [Online]. Available: http://isyou.info/jowua/papers/jowua-v7n4-3.pdf

[35] i-SCOOP. Industry 4.0 and the fourth industrial revolutioon explained. [Online]. Available: https://www.i-scoop.eu/industry-4-0/

[36] C. y turismo Ministerio de Industria. Herramienta de autodiagnÓstico digital avanzada. [Online]. Available: https://hada.industriaconectada40.gob.es/hada/auth/login

[37] Dekalabs. Evolución de la industria tradicional al modelo industria 4.0. [Online]. Available: https://dekalabs.com/deka-akademy/evolucion-de-la-industria-tradicional-al-modelo-industria-4

[38] Profibus&Profinet. ¿Qué es ethernet industrial? [Online]. Available: https://profibus.com.ar/ethernet-industrial/

[39] S. Baek, D. Kim, M. Tesanovic, and A. Agiwal, "3GPP new radio Release 16: Evolution of 5G for industrial internet of things," *IEEE Communications Magazine*, vol. 59, pp. 41–47, 2021.

[40] 3GPP TR 121 917V17.0.1. (2023) Digital cellular telecommunications system (phase 2+) (GSM); universal mobile telecommunications system (UMTS); LTE; 5G; (Release 17). [Online]. Available: https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

[41] G. Americas, "Becoming 5G advanced the 3GPP 2025 roadmap indesign," 2022. [Online]. Available: https://www.5gamericas.org/becoming-5g-advanced-the-3gpp-roadmap/

[42] J. P. G. J. M. L. S. P. A. Gutiérrez, "Dynamic provisioning of softwarized 5G mobile core networks," Ph.D. dissertation, 2018. [Online]. Available: http://hdl.handle.net/10481/53865

[43] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched ethernet networks," in *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, vol. 2016-August. Institute of Electrical and Electronics Engineers Inc., 8 2016, pp. 75–85.

[44] (2020, 2) IEEE standard for local and metropolitan area networks–bridges and bridged networks amendment: Asynchronous traffic shaping.

[45] 3GPP Technical Specification (TS) 23.501. (2020, 8) System architecture for the 5G system (5GS). [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/

[46] 3GPP Technical specification (TS) 23.501. (2023, 6) System architecture for the 5G system (5GS). [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/

[47] IEEE Std 802.1Q-2022. (2022) IEEE standard for local and metropolitan area networks-bridges and bridged networks. [Online]. Available: https://standards.ieee.org/ieee/802.1Q/10323/#:~:text=1Q-,Standard%20for%20Local%20and%20Metropolitan%20Area%20Networks%2D%2DBridges%20and,provide%20Bridged%20Networks%20and%20VLANs.

[48] IEEE Std P802.1Qdj-d1.0. IEEE draft standard for local and metropolitan area networks - bridges and bridged networks - amendment xx: Configuration enhancements for time-sensitive networking. [Online]. Available: https://1.ieee802.org/tsn/802-1qdj/

[49] 3GPP TS22.104 V17.4.0, "Service requirements for cyber-physical control applications in vertical domains; (Release 19)," 3GPP, Tech. Rep., 2023. [Online]. Available: http://www.3gpp.org

[50] J.-Y. Boudec and P. Thiran, Eds., *Network Calculus: A theory of deterministic queuing systems for the Intenrnet.* Springer, 7 2001.

[51] J.-Y. L. Boudec, "A theory of traffic regulators for deterministic networks with application to interleaved regulators," *IEEE/ACM Transactions on Networking*, vol. 26, pp. 2721–2733, 12 2018.

[52] 5G-ACIA, "Integration of 5G with time-sensitive networking for industrial communications," 5G Alliance for Connected Industries and Automation (5G-ACIA), Tech. Rep., 2021. [Online]. Available: https://5g-acia.org/wp-content/uploads/2021/04/5G-ACIA_IntegrationOf5GWithTime-SensitiveNetworkingForIndustrialCommunications.pdf

[53] 5G-ACIA, "A 5G traffic model for industrial use cases," 5G Alliance for Connected Industries and Automation (5G-ACIA), Tech. Rep., 11 2019. [Online]. Available: https://5g-acia.org/media/publications/

[54] M. Linehan, "Time sensitive networks for flexible manufacturing testbed characterization and mapping of converged traffic types," Industrial Internet Consortium, Tech. Rep., 2019.

Acronyms

2G Second Generation

3G Third Generation

3GPP Third Generation Partnership Project

5G Fifth Generation

5GC 5G Core

5G-RAN 5G Radio Access Network

5GS 5G System

4G Fourth Generation

ADAS Advanced Driver Assistance Systems

AF Application Function

ATS Asynchronous Traffic Shaper

AMF Access and Mobility Management Function

AN-TL Access Network Talker Listener

AR Augmented Reality

ARPF Authentication Credential Repository and Processing Function

API Application Programming Interface

AUSF Authentication Server Function

BAT Burst Arrival Time

BE Best-Effort

CAGR Compound Annual Growth Rate

CAPEX Capital Expenditure

CN Core Network

CNC Centralized Network Configuration

CN-TL Core Network Talker Listener

CP Control Plane

CPS Cyber-Physical System

CPU Central Processing Unit

CRS Cell-specific Reference Signal

CSMA Carrier Sense Multiple Access

CUC Centralized User Configuration

DFT Discrete Fourier Transform

DRL Deep Reinforcement Learning

DS-TT Device-side TT

E2E end-to-end

eNB Evolved Node B

eMBB enhanced Mobile Broadband

EPC Evolved Packet Core

FCFS First Come, First Served

FIFO First In, First Out

FRER Frame Replication and Elimination for Reliability

GCL Gate Control List

gNB Next Generation Node B

gPTP generalized Precision Time Protocol

HADA Herramienta de Autodiagnóstico Digital Avanzada

HARQ Hybrid Automatic Repeat reQuest

HOL Head of Line

IDE Integrated Development Environment

IE Industrial Ethernet

IEEE Institute of Electrical and Electronics Engineers

IIoT Industrial Internet of Things

IoT Internet of Things

IP Internet Protocol

IPV Internal Priority Value

ITU International Telecommunications Union

KPI Key Performance Indicator

LLDP Link Layer Discovery Protocol

MIB Message Information Base

MIMO multiple-input multiple-output

mMIMO massive Multiple Input Multiple Output

mMTC massive Machine Type Communication

MSTP Multiple Spanning Tree Protocol

MTU Maximum Transmission Unit

NEPTUNO Next Generation Transport Network Optimizer

NEF Network Exposure Function

NFV Network Functions Virtualisation

NG-RAN Next Generation Radio Access Network

NSA Non Standalone

NMS Network Managment System

NSSI Network Slice Subnet Instance

NSSF Network Slice Selection Function

NR New Radio

NRF NF Repository Function

NW-TT Network-side TT

OFDM Orthogonal Frequency Division Multiplexing

OFDMA Orthogonal Frequency Division Multiple Access

OSI Open Systems Interconnection

PBCH Physical Broadcast Channel

PCP Priority Code Point

PCF Policy Control Function

PDB Packet Delay Budget

PDCCH Physical Downlink Control Channel

PDSCH Physical Downlink Shared Channel

PDU Packet Data Unit

PLC Programmable Logic Controller

PRACH Physical Random Access Channel

PSFP Per-Stream Filtering and Policing

PTP Precision Time Protocol

PUCCH Physical Uplink Control Channel

PUSCH Physical Uplink Shared Channel

QoS Quality of Service

RAN Radio Access Network

RB Resource Block

RE Resource Element

RSTP Rapid Spanning Tree Protocol

SA Stand Alone

SBA Service Based Architecture

SCS Subcarrier Spacings

SDN Software-Defined Networking

SDU Service Data Unit

SMF Session Management Function

SMT Satisfiability Modulo Theories

SNMP Simple Network Management Protocol

SRP Stream Reservation Protocol

TAS Time-Aware Shaper

TC Traffic Class

TCP Transmission Control Protocol

TDMA Time-Division Multiplexing Access

TN Transport Network

TN CNC CNC in Transport Network

TSCAI Time-Sensitive Communication Assistance Information

TSN Time-Sensitive Networking

TSN AF TSN Application Function

TRS Topology Rank Solver

UBS Urgency-Based Scheduler

UDM Unified Data Management

UE User Equipment

UNI User/Network Interface

URLLC Ultra-Reliable and Low Latency Communication

UP User Plane

UPF User Plane Function

V2V vehicle-to-vehicle

VLAN Virtual Local Area Network

VNF Virtual Network Function

VR Virtual Reality

WCQD Worst Case Queuing Delay

WSN Wireless Sensor Network