Time Sensitive Networking in Multimedia and Industrial Control Applications

by

Ahmed H. A. A. Nasrallah

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2022 by the
Graduate Supervisory Committee:

Martin Reisslein, Chair
Violet R. Syrotiuk
Robert LiKamWa
Akhilesh Thyagaturu

ARIZONA STATE UNIVERSITY

May 2022

ABSTRACT

Ethernet based technologies are emerging as the ubiquitous de facto form of communication due to their interoperability, capacity, cost, and reliability. Traditional Ethernet is designed with the goal of delivering best effort services. However, several real time and control applications require more precise deterministic requirements and Ultra Low Latency (ULL), that Ethernet cannot be used for. Current Industrial Automation and Control Systems (IACS) applications use semi-proprietary technologies that provide deterministic communication behavior for sporadic and periodic traffic, but can lead to closed systems that do not interoperate effectively. The convergence between the informational and operational technologies in modern industrial control networks cannot be achieved using traditional Ethernet.

Time Sensitive Networking (TSN) is a suite of IEEE standards designed by augmenting traditional Ethernet with real time deterministic properties ideal for Digital Signal Processing (DSP) applications. Similarly, Deterministic Networking (DetNet) is a Internet Engineering Task Force (IETF) standardization that enhances the network layer with the required deterministic properties needed for IACS applications. This dissertation provides an in-depth survey and literature review on both standards/research and 5G related material on ULL. Recognizing the limitations of several features of the standards, this dissertation provides an empirical evaluation of these approaches and presents novel enhancements to the shapers and schedulers involved in TSN. More specifically, this dissertation investigates Time Aware Shaper (TAS), Asynchronous Traffic Shaper (ATS), and Cyclic Queuing and Forwarding (CQF) schedulers. Moreover, the IEEE 802.1Qcc, centralized management and control, and the IEEE 802.1Qbv can be used to manage and control scheduled traffic streams with periodic properties along with best-effort traffic on the same network infrastructure. Both the centralized network/distributed user model (hybrid model)

and the fully-distributed (decentralized) IEEE 802.1Qcc model are examined on a typical industrial control network with the goal of maximizing scheduled traffic streams. Finally, since industrial applications and cyber-physical systems require timely delivery, any channel or node faults can cause severe disruption to the operational continuity of the application. Therefore, the IEEE 802.1CB, Frame Replication and Elimination for Reliability (FRER), is examined and tested using machine learning models to predict faulty scenarios and issue remedies seamlessly.

I dedicate this dissertation to my loving parents, who always stood by me and supported all my decisions, and especially to my late father and in his memory, who would have been very proud to see me graduate with a Ph.D in his Alma Mater. Finally to my close and very special friends, who selflessly showed unconditional love and support.

ACKNOWLEDGMENTS

My deepest regards and thanks to my advisor Prof. Martin Reisslein, without whom this would have not been possible. I am immensely thankful to the committee members, Prof. Violet R. Syrotiuk, Prof. Robert LiKamWa, and Prof. Akhilesh Thyagaturu, for their guidance and continued support. My heartfelt thanks to my friends, lab-mates, colleagues and mentors, for their sincere love and support. In addition, I am also very grateful to Hesham ElBakoury, Chief Architect, Huawei for his insightful discussion as a part of our collaboration project with Huawei and for generous funding support, which made my research possible. Lastly, but certainly not the least, I would like to thank Kuwait University for giving me the opportunity to continue and complete this program.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Traditional networks which provide end-to-end connectivity to the users have only been successful in reducing the operating end-to-end latencies to the order of tens of milliseconds. However, present and future applications demand Ultra-Low Latency (ULL). For instance, the end-to-end latencies should be on the order of a few microseconds to a few milliseconds for industrial applications [432], around 1 millisecond for the tactile Internet [166, 281], and on the order of 100 microseconds for the one-way fronthaul in wireless cellular networks. For example, critical healthcare applications, e.g., for tele-surgery, and transportation applications [172] require near real-time connectivity. Throughput requirements largely depends on the application needs, which may vary widely from small amounts of IoT data to large exchanges of media data transfers to and from the cloud (or the fog to reduce latency) [379]. Additionally, autonomous automotive vehicles [373], augmented and virtual reality (AR/VR), as well as robotic applications, which are essential for Industrial IoT (IIoT), may require both high data rates as well as ULL [133, 134, 192, 358]. High speed data rates may be required for transporting video feeds from camera sensors that are used to control actuators in vehicles and robots [67]. Therefore, in such heterogeneous environments and applications, a dedicated mechanism to universally accommodate a diverse range of ULL requirements would be very helpful [180]. More broadly, industrial and commercial processes in the era of Cyber-Physical Systems (CPS), Industrial Internet of Things (IIoT), Industry 4.0 need interoperable communication stacks between devices, systems, and protocol layers.

1.1 Latency Terminology

Generally, latency refers to the total end-to-end packet delay from the instant of the beginning of transmission by the sender (talker) to the complete reception by the receiver (listener). The term ultra-low latency (ULL) commonly refers to latencies that are very short, e.g., on the order of a few milliseconds or less than one millisecond. ULL applications often require deterministic latency, i.e., all frames of a given application traffic flow (connection) must not exceed a prescribed bound [359], e.g., to ensure the proper functioning of industrial automation systems. It is also possible that applications may require probabilistic latency, i.e., a prescribed delay bound should be met with high probability, e.g., for multimedia streaming systems [57, 415], where rare delay bound violations have negligible impact of the perceived quality of the multimedia.

Latency jitter, or jitter for short, refers to the packet latency variations. Often ULL systems require very low jitter. Latency and jitter are the two main quality of service (QoS) metrics for ULL networking. We note that there are a wide range of ULL applications with vastly different QoS requirements, see Table 1.1. For instance, some industrial control applications have very tight delay bounds, e.g., only a few microseconds, while other industrial control applications have more relaxed delay bounds up to a millisecond.

1.2 Time Sensitive Networking Application Needs and Related Traffic Specification

While Table. 1.1 provides end to end QoS requirements for several applications, there are a fixed number of traffic classes (in addition to Table 1.4) and their traffic characteristics that are specifically used in industrial automation and control systems.

Table 1.1: End-to-end latency and jitter requirements for typical ULL applications

Area	Application	QoS Requirements Latencies	Jitter
Medical [114, 443, 455]	Tele-Surgery, Haptic Feedback	3–10 ms	< 2 ms
Industry [29]	Indust. Automation, Control Syst.	0.2 μs–0.5 ms for netw. with 1 Gbit/s link speeds	meet lat. req.
		25 μs–2 ms for netw. with 100 Mbit/s link speeds	meet lat. req.
	Power Grid Sys.	approx. 8ms	few μs
Banking [302]	High-Freq. Trading	< 1 ms	few μs
Avionics [377]	AFDX Variants	1–128ms	few μs
Automotive [210, 230, 336, 401]	Adv. Driver. Assist. Sys. (ADAS)	100–250 μs	few μs
	Power Train, Chassis Control	$< 10\mu$s	few μs
	Traffic Efficiency & Safety	< 5 ms	few μs
Infotainment [100]	Augmented Reality	7–20 ms	few μs
	Prof. Audio/Video	2–50 ms	< 100 μs

A number of these have been investigated in the IEC/IEEE 60802 TSN profile [73] for industrial automation and summarized in Table 1.2 and Table 1.3.

Examples of Isochronous traffic types include printing machines with synchronized drives where devises synchronously poll inputs and apply synced outputs by exchanging data at a predefined bit rate. Therefore, for Isochronous data where applications are synced to common clock, transmission jitter needs to be minimal with zero loss and interference from other traffic. Examples of Cyclic traffic include pick & place and sorting in a commercial conveyor belts where devices sample inputs and apply outputs cyclically (depending on the data transmission period). Examples of Events traffic type include event-based control and alarm/warning and operator commands where devices need to generate messages based on event triggers, i.e., changes in the

Table 1.2: TSN traffic type criteria

Traffic Type	**Description**
Data transmission periodicity	Cyclic or periodic, acyclic or sporadic
Typical period	Planned data transmission interval
Synchronized to the network	Indication whether application is synchronized to network time
Data delivery guarantee	Deadline, latency, and Bandwidth
Tolerance to interference	Application's tolerance to certain latency variation (jitter)
Tolerance to loss	Application's tolerance to certain amount of packet loss
Typical application data size	Application payload (fixed or variable)
Criticality	Criticality of the application's data for operation of critical parts of the system

operating environment that require attention. Such operation must be able to handle bursts of data without loss (or up to a certain tolerance of loss). Examples of Configuration and Diagnostic traffic include network and system wide management and configuration (e.g., SNMP, RESTCONF/NETCONF, etc.) where devices create peak data ready for transmission intermittently that can tolerate impact to latency and loss. Examples of Network Control traffic include clock synchronization (e.g., PTP or gPTP), network redundancy (e.g., RSTP, PCR), and topology detection changes (e.g., LLDP) where devices generate traffic used for network control operation that are usually in low volume but have critical delivery requirements due to higher layer protocols' usage. Examples of Best Effort include large bulk data back-up systems which can occur sporadically or periodically. Generally, Best Effort traffic is provided no guarantees and can suffer data loss when high priority traffic uses all the bandwidth reserved. Finally, examples of Audio and Video (i.e., multimedia traffic) are voice over IP and video surveillance traffic where the data is usually consumed by human operators and require no vision based control application traffic and packet/frame loss may lead to degraded quality of experience but not necessarily failure of the application.

Table 1.3: TSN traffic types (P - Periodic, S - Sporadic, D - Deadline, L - Latency, B - Bandwidth, H - High, M - Medium, L - Low)

Type	Periodicity	Typical period	Synchronized to network	Data delivery guarantee	Tolerance to interference	Tolerance to loss	Typical application payload size	Criticality
Isochronous	P	<2 ms	Yes	D	No	No	Fixed 30-100B	H
Cyclic	P	2 ms-20 ms	No	L	$\leq$ latency	1-4 frames	Fixed 50-1000B	H
Events	S	N/A	No	L	N/A	Yes	Variable 100-1500B	H
Network Control	P	50 ms-1 s	No	B	Yes	Yes	Variable 50-500B	H
Config. & Diagnostics	S	N/A	No	B	N/A	Yes	Variable 500-1500B	M
Best Effort	S	N/A	No	N/A	N/A	Yes	Variable 30-1500B	L
Audio	P	Frame Rate	No	L	N/A	Yes	Variable 1000-1500B	L
Video	P	Sampling Rate	No	L	N/A	Yes	Variable 1000-1500B	L

1.3 Background

1.3.1 IEEE 802.1 Overview

Before we delve into the standardization efforts of the IEEE Time-Sensitive Network (TSN) Task Group (TG), we briefly explain the organizational structure of the IEEE 802.1 Working Group (WG). The 802.1 WG is chartered to develop and maintain standards and recommended practices in the following areas: 1) 802 LAN/MAN architecture, 2) internetworking among 802 LANs, MANs, and other wide area networks, 3) security, 4) 802 overall network management, and 5) protocol layers above the MAC and LLC layers. Currently, there are four active task groups in this WG: 1) Time Sensitive Networking, 2) Security, 3) Data Center Bridging, and 4) OmniRAN.

The main IEEE 802.1 standard that has been continuously revised and updated over the years is IEEE 802.1Q-2014 [19], formally known as the IEEE 802.1D standard. That is, IEEE 802.1Q-2014, which we abbreviate to IEEE 802.1Q, is the main Bridges and Bridged Networks standard that has incorporated all 802.1Qxx amendments, where "xx" indicates the amendment to the previous version of 802.1Q.

IEEE 802.1 Bridge

IEEE 802.1Q extensively utilizes the terminology "IEEE 802.1 bridge", which we abbreviate to "bridge". A bridge is defined as any network entity within an 802.1 enabled network that conforms to the mandatory or optional/recommended specifications outlined in the standard, i.e., any network node that supports the IEEE 802.1Q functionalities. IEEE 802.1Q details specifications for VLAN-aware bridges and bridged LAN networks. More specifically, IEEE 802.1Q specifies the architectures and protocols for the communications between interconnected bridges (L2 nodes), and the inter-process communication between the layers and sublayers adjacent to the main 802.1 layer (L2).

802.1Q Traffic Classes

The IEEE 802.1Q standard specifies traffic classes with corresponding priority values that characterize the traffic class-based forwarding behavior, i.e, the Class of Service (CoS) [19, Annex I]. Eight traffic classes are specified in the 802.1Q standard, whereby

Table 1.4: IEEE 802.1 Traffic Classes

Priority	**Traffic Class**
0	Background
1	Best effort
2	Excellent effort
3	Critical application
4	"Video" $<$ 100 ms latency and jitter
5	"Voice" $<$ 10 ms latency and jitter
6	Internetwork control, e.g., OSPF, RIP
7	Control Data Traffic (CDT), e.g., from IACSs

the priority level ranges from lowest priority (0) to highest priority (7), as summarized in Table 1.4.

1.4 General Development Steps from Ethernet Towards Time Sensitive Networking

Ethernet has been widely adopted as a common mode of networking connectivity due to very simple connection mechanisms and protocol operations. Since its inception in the 1970s [295, 296] and first standardization in the IEEE 802.3 standard in 1983 [434], Ethernet has kept up with the "speed race" and today's Ethernet definitions support connections up to 400 Gbps. Due to the ever increasing demands, there is an ongoing effort to advance Ethernet connectivity technologies to reach speeds up to 1 Tbps. The best-effort Ethernet service reduces the network complexity and keeps protocol operations simple, while driving down the product costs of Ethernet units. Despite the enormous successes and wide-spread adoption of Ethernet, the Ethernet definitions fundamentally lack deterministic quality of service (QoS) properties of end-to-end flows. Prior to the development of the TSN standards, ULL applications, e.g., industrial communications, deployed point-to-point communication and circuit switching or specialized/semi-proprietary specifications, such as, fieldbus communication, e.g., IEEE 1394 (FireWire), Process Field Network (Profinet), or Ethernet for Controlled Automation Technology (EtherCAT). In general, the Ethernet definitions lack the following aspects for supporting ULL applications:

i) Lack of QoS mechanisms to deliver packets in real time for demanding applications, such as real time audio and video delivery.

ii) Lack of global timing information and synchronization in network elements.

iii) Lack of network management mechanisms, such as bandwidth reservation mechanisms.

iv) Lack of policy enforcement mechanisms, such as packet filtering to ensure a guaranteed QoS level for an end-user.

Motivated by these Ethernet shortcomings, the Institute of Electronics and Electrical Engineers (IEEE) and the Internet Engineering Task Force (IETF) have proposed new definitions to introduce deterministic network packet flow concepts. The IEEE has pursued the Time Sensitive Networking (TSN) standardization [158] focusing mainly on the physical layer (layer one, L1) and link layer (layer two, L2) techniques within the TSN task group in the IEEE 802.1 working group (WG). The IETF has formed the DETerministic NETwork (DETNET) working group focusing on the network layer (L3) and higher layer techniques which is out of scope for this dissertation.

1.5 Organization of this Dissertation

This dissertation is organized as follows. Chapter 2 comprehensively surveys TSN related standardization and literature, 5G application domain related to TSN and outlines the potential limitations and future work directions. Chapter 3 highlights the limitations of the standard Time Aware Shaper (TAS) and presents enhancements to TAS. A performance evaluation is conducted and a comparison is shown between the enhanced TAS (or adaptive TAS), standard TAS, and Asynchronous Traffic Shaper (ATS) which is an event triggered shaper. Since the findings in Chapter 3 show a lack of admission control and resource reservation, Chapter 4 designs and implements these mechanisms to augment TAS with specifications related to IEEE 802.1Qcc (centralized management). An empirical evaluation is conducted with multiple scenarios to show the efficacy of the proposed framework. Since most mechanisms that implement TSN require time synchronization (e.g., coordinated queuing operations), large scale deterministic networks is investigated in Chapter 5. More specifically, the CQF mechanism is studies and compared against the Paternoster algorithm with varying link distances. Deviating from scheduling/shaping and management/configuration operations, fault tolerance and frame reliability is investigated in Chapter 6. A machine learning proof of concept is shown and evaluated with preliminary results against different approaches that implement Frame Replication and Elimination (FRER). Finally, Chapter 7 concludes the dissertation.

Chapter 2

ULTRA-LOW LATENCY (ULL) NETWORKS: THE IEEE TSN AND RELATED 5G ULL RESEARCH

2.1 Introduction

This chapter provides a comprehensive up-to-date survey of standards and research studies addressing networking mechanisms for ULL applications. Section 2.2 covers the IEEE TSN standards that have grown out of the AVB standards and focus primarily on the link layer, while Section 2.3 covers the ULL research studies related to TSN. The sequence of the section on standards followed by the section on related research studies is inspired by the temporal sequence of the development of the ULL field, where standard development has typically preceded research studies.

A large portion of the ULL applications will likely involve wireless communications, whereby the fifth generation (5G) wireless systems will play a prominent role. In particular, the emerging tactile Internet paradigm with end-to-end target latencies below 1 ms is tightly coupled to the ongoing 5G developments [38, 39, 58, 135, 379]. The support of 5G wireless ULL communications services will likely heavily rely on the TSN standards and research results. On the other hand, due to prevalence and importance of wireless communications in today's society, the particular 5G wireless context and requirements will likely influence the future development of ULL standards development and research. We believe that for a thorough understanding of the complete ULL research area it is vital to comprehensively consider the ULL standards, namely TSN, as well as a main "application domain" of ULL standards and research results. We anticipate that 5G wireless communications will emerge as a highly important application domain of ULL standards and research results and we therefore survey ULL related standards and research studies for 5G wireless systems in Section 2.4.

Section 2.5 identifies the main gaps and limitations of the existing TSN standards as well as ULL related 5G standards and research studies and outlines future research directions to address these gaps and limitations.

2.1.1 Related Literature

While to the best of our knowledge there is no prior survey on time sensitive networking (TSN), there are prior surveys on topics that relate to TSN. We proceed to review these related surveys and differentiate them from our survey.

A survey on general techniques for reducing latencies in Internet Protocol (IP) packet networks has been presented in [87]. The survey [87] gave a broad overview of the sources of latencies in IP networks and techniques for latency reduction that have appeared in publications up to August 2014. The range of considered latency sources included the network structure, e.g., aspects of content placement and service architectures, the end point interactions, e.g., aspects of transport initialization and secure session initialization, as well as the delays inside end hosts, e.g., operating system delays. In contrast, we provide an up-to-date survey of the IEEE Time Sensitive Networking (TSN) standard for the link layer and related research studies. Thus, in brief, whereas the survey [87] broadly covered all latency sources up to 2014, we comprehensively cover the link and network layer latency reduction standards and studies up to July 2018.

A few surveys have examined specific protocol aspects that relate to latency, e.g., time synchronization protocols have been surveyed in [256, 273], routing protocols have been surveyed in [126, 191, 249], while congestion control protocols have been covered in [271, 440]. Several surveys have covered latency reduction through mobile edge and fog computing, e.g., see [278, 307, 323, 402]. Also, the impact of wireless protocols on latency has been covered in a few surveys [40, 77, 142, 176, 342, 344, 400], while smart grid communication has been covered in [153]. Low-latency packet processing has been surveyed in [95], while coding schemes have been surveyed in [68, 143]. A comprehensive guide to stochastic network calculus, which can be employed to analyze network delays has appeared in [167].

Several surveys have covered the Tactile Internet paradigm [39, 166, 281, 390], which strives for latencies on the order of one millisecond. The AVB standard, which is a predecessor to the IEEE TSN standards development was surveyed in [154, 406]. In contrast to these existing surveys we provide a comprehensive up-to-date survey of the IEEE TSN standards development and the related research studies.

2.2 IEEE TSN Standardization

This section surveys the standardization efforts of the IEEE 802.1 TSN Task Group. IEEE 802.1 TSN Group standards and protocols extend the traditional Ethernet data-link layer standards to guarantee data transmission with bounded ultra-low latency, low delay variation (jitter), and extremely low loss, which is ideal for industrial control and automotive applications [169, 293]. TSN can be deployed over Ethernet to achieve the infrastructure and protocol capabilities for supporting real-time Industrial Automation and Control System (IACS) applications.

In order to give a comprehensive survey of the current state of the art of TSN standardization, we categorize the standardization efforts for the network infrastructure supporting ULL applications. We have adopted a classification centered around the notion of the TSN flow, which is defined as follows. An end-to-end unicast or multicast network connection from one end station (talker, sender) to another end

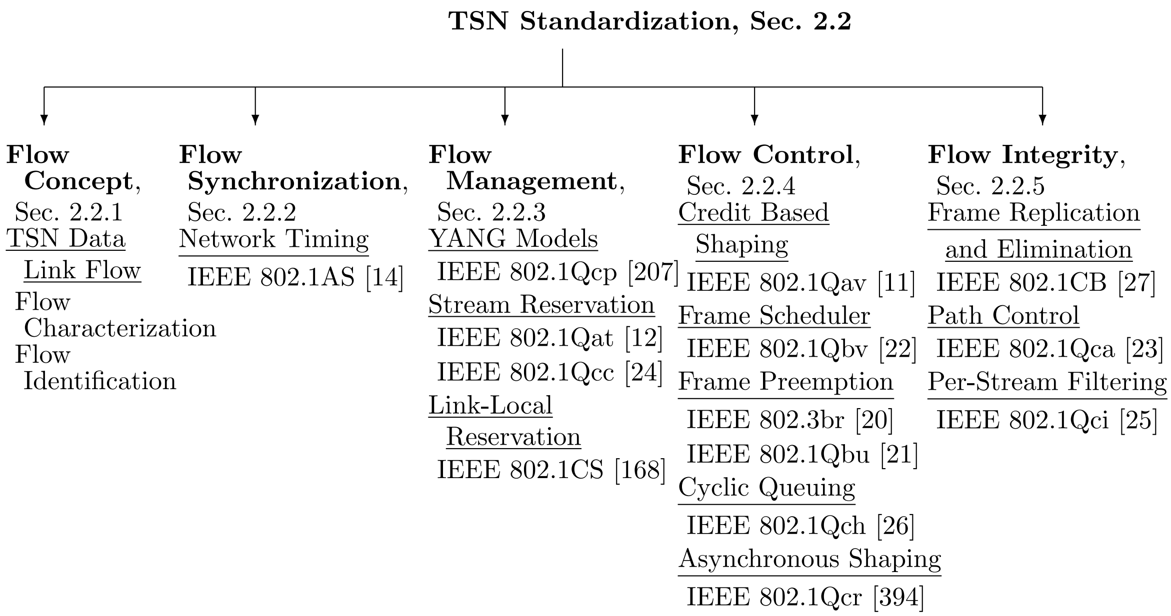


Figure 2.1: Classification taxonomy of Time Sensitive Networking (TSN) standardization.

station(s) (listener(s), receiver(s)) through the time-sensitive capable network is defined as a *TSN flow*, which we often abbreviate to "flow" and some publications refer to as "stream". We have organized our survey of the standardized TSN mechanisms and principles in terms of the TSN flow properties, as illustrated in Fig.2.1. Complementary to the taxonomy in Fig.2.1, Fig. 2.2 provides a historical perspective of the TSN standards and the ongoing derivatives and revisions.

2.2.1 Flow Concept: PCP and VLAN ID Flow Identification

A TSN flow (data link flow) is characterized by the QoS properties (e.g., bandwidth and latency) defined for the traffic class to which the flow belongs. In particular, a TSN flow is defined by the priority code point (PCP) field and VLAN ID (VID) within the 802.1Q VLAN tag in the Ethernet header. The PCP field and VID are assigned based on the application associated with the flow. Fig. 2.3 outlines the general QoS characteristics of the traffic classes related to the Informational Technology

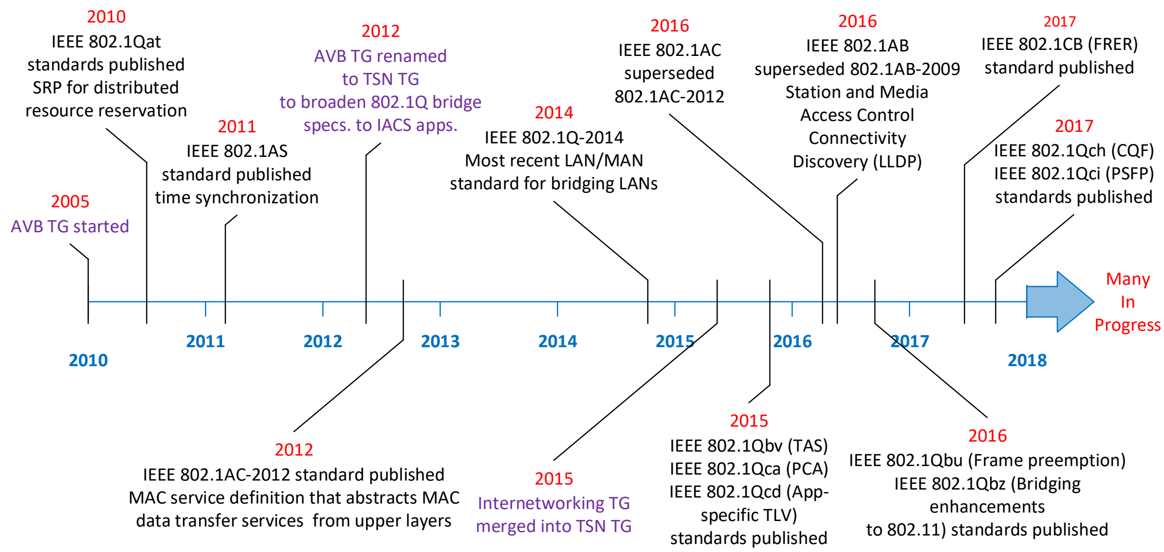


Figure 2.2: Timeline of IEEE TSN task group (TG), highlighting significant milestones and depicting the shift from Audio Video Bridging (AVB) to TSN.

(IT) and Operational Technology (OT) domains. Furthermore, Fig. 2.3 provides the main features for each block, including typical applications used. As IT and OT establish a converged interconnected heterogeneous network, the delay bottleneck must be diminished to tolerable levels for IACS applications, i.e., the machine and control floor networks.

2.2.2 Flow Synchronization

IEEE 802.1AS Time Synchronization for Time-Sensitive Applications

Many TSN standards are based on a network-wide precise time synchronization, i.e., an established common time reference that is shared by all TSN network entities. The time synchronization is, for instance, employed to determine opportune data and control signaling scheduling. Time synchronization is accomplished through the IEEE 802.1AS stand-alone standard [14, 397], which uses a specialized profile (selection of features/configuration) of IEEE 1588-2008 (1588v2) [9], the generic Precision Time

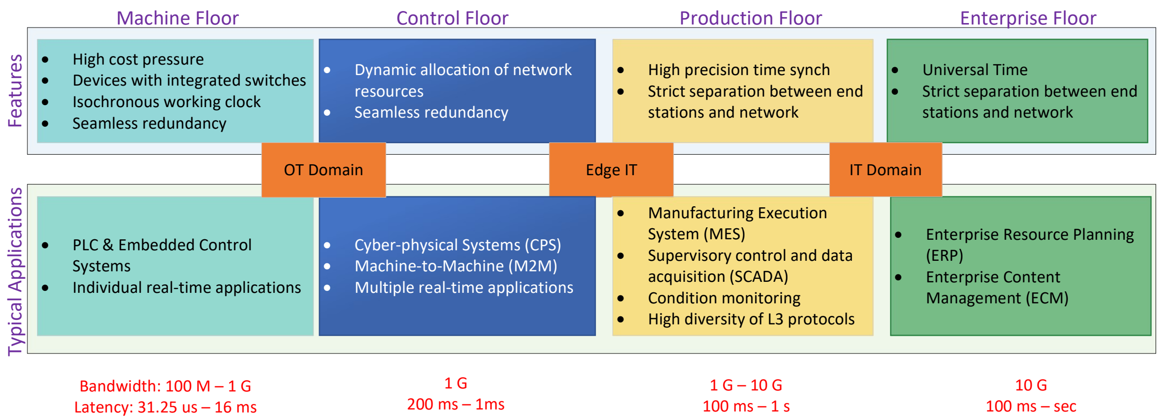


Figure 2.3: Illustration of the broad range of QoS requirements according to the network setting (floor), whereby the machine floor requires the highest level of determinism and the lowest latency. Traditional networking is deployed on the enterprise floor. The top row summarizes the features required at each floor, while the bottom row illustrates typical example applications. TSN can, in principle, be deployed everywhere, but typically, TSN is most attractive for the real-time systems in the OT Domain, i.e., the machine and control floors.

Protocol (gPTP). The gPTP synchronizes clocks between network devices by passing relevant time event messages [256]. The message passing between the Clock Master (CM) and the Clock Slaves (CSs) forms a time-aware network, also referred to as gPTP domain, as illustrated in Fig. 2.4. The time-aware network utilizes the peer-path delay mechanism to compute both the residence time, i.e., the ingress-to-egress processing, queuing, and transmission time within a bridge, and the link latency, i.e., the single hop propagation delay between adjacent bridges within the time-aware network hierarchy with reference to the GrandMaster (GM) clock at the root of the hierarchy [14, Section 11]. The GM clock is defined as the bridge with the most accurate clock source selected by the Best Master Clock Algorithm (BMCA) [9].

For example, in Fig. 2.4, the bottom left-most 802.1AS end point receives time information from the upstream CM which includes the cumulative time from the GM to the upstream CM. For full-duplex Ethernet LANs, the path delay measurement between the local CS and the direct CM peer is calculated and used to correct the

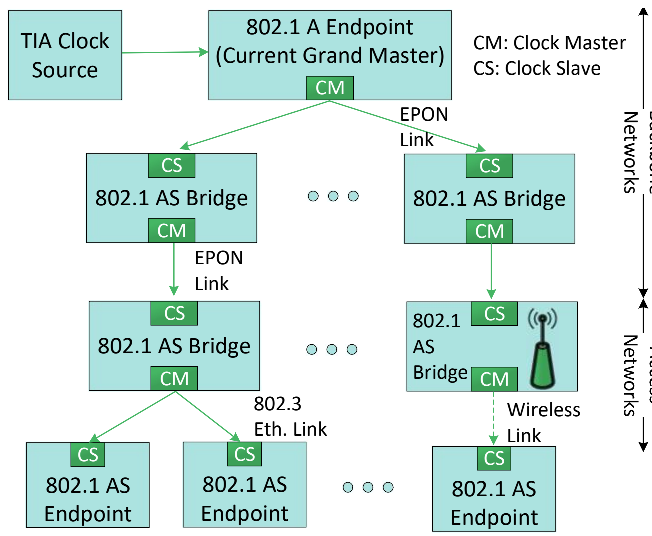


Figure 2.4: Illustration of a typical gPTP domain operation and time sharing where the selected GM source distributes timing information to all downstream 802.1AS bridges. Each bridge corrects the delay and propagates the timing information on all downstream ports, eventually reaching the 802.1AS end points (end stations). The International Atomic Time (TIA) is the GM's source for timing information.

received time. Upon adjusting (correcting) the received time, the local clock should be synchronized to the gPTP domain's GM clock.

In general, gPTP systems consist of distributed and interconnected gPTP and non-gPTP devices. Time-aware bridges and end points are gPTP devices, while non-gPTP devices include passive and active devices that do not contribute to time synchronization in the distributed network. gPTP is a distributed protocol that uses a master-slave architecture to synchronize real-time clocks in all devices in the gPTP domain with the root reference (GM) clock. Synchronization is accomplished through a two-phase process: The gPTP devices 1) establish a master-slave hierarchy, and 2) apply clock synchronization operations. In particular, gPTP establishes a master-slave hierarchy using the BMCA [9], which consists of two separate algorithms, namely data set comparison and state decision. Each gPTP device operates a gPTP engine, i.e., a gPTP state machine, and employs several gPTP UDP IPv4 or IPv6 multicast and unicast messages to establish the appropriate hierarchy and to correctly synchronize time [14]. Any non-time aware bridge that cannot relay or synchronize timing messages does not participate in the BMCA clock spanning tree protocol.

The time synchronization accuracy depends mainly on the accuracy of the residence time and link delay measurements. In order to achieve high accuracy, 802.1AS time-aware systems correct the received upstream neighbor master clock's timing information through the GM's frequency ratio, this process is called logical syntonization in the standard. In the synchronization context, frequency refers to the clock oscillator frequency. The frequency ratio is the ratio of the local clock frequency to the frequency of the time-aware system at the other end of an attached link. 802.1AS achieves proper synchronization between time-aware bridges and end systems using both the frequency ratio of the GM relative to the local clock to compute the synchronized time, and the frequency ratio of the neighbor CM relative to the local CS to correct any propagation time measurements.

IEEE802.1AS-REV introduces new features needed for time-sensitive applications. These features include the ability to support multiple time domains to allow rapid switchover should a GM clock fail, and improved time measurement accuracy.

Summary and Lessons Learned

IEEE 802.1AS provides reliable accurate network wide time synchronization. All gPTP systems compute both the residence time and the link latency (propagation delay) and exchange messages along a hierarchical structure centered around the selected GM clock to accurately synchronize time. Flow control and management components, e.g., IEEE 802.1Qbv and 802.1Qcc (see Sections 2.2.4 and 2.2.3), can utilize the 802.1AS timing synchronization to provide accurate bounded latency and extremely low loss and delay variation for TSN applications.

An open aspect of time synchronization is that the frequent periodic exchange of timing information between the individual network entities can stress and induce backpressure on the control plane. The control plane load due to the time synchronization can ultimately impact ULL applications. A centralized time synchronization system, e.g., based on a design similar to software defined networking (SDN) [52, 329], with message exchanges only between a central synchronization controller and individual network entities could help mitigate the control plane overhead. However, such a centralized synchronization approach may create a single-point of failure in the time synchronization process. The detailed quantitative study of these tradeoffs is an interesting direction for future research.

2.2.3 Flow Management

Flow management enables users or operators to dynamically discover, configure, monitor, and report bridge and end station capabilities.

IEEE 802.1Qcp YANG Data Model

The TSN TG has proposed the IEEE 802.1Qcp TSN Configuration YANG model standard to achieve a truly universal Plug-and-Play (uPnP) model. The IEEE 802.1Qcp standard utilizes the Unified Modeling Language (UML), specifically the YANG [82, 85] data model. The YANG data model provides a framework for periodic status reporting as well as for configuring 802.1 bridges and bridge components, including Media Access Control (MAC) Bridges, Two-Port MAC Relays (TPMRs), Customer Virtual Local Area Network (VLAN) Bridges, and Provider Bridges [207]. Additionally, IEEE 802.1Qcp is used to support other TSN standard specifications, such as the Security and Datacenter Bridging TG standards 802.1AX and 802.1X.

YANG [82, 85] is a data modeling language for configuration data, state data, remote procedure calls, and notifications for network management protocols, e.g., NET-

CONF and RESTCONF. NETCONF is the Network Configuration Protocol [151] that provides mechanisms to install, manage, and delete the configurations of network devices. The industry wide adoption of the YANG formalized data modeling language, e.g., by the IETF and the Metro Ethernet Forum (MEF), is an important motivation for integrating, automating, and providing support for YANG data modeling in 802.1 bridges and related services for upper layer components.

IEEE 802.1Qat Stream Reservation Protocol (SRP) and IEEE 802.1Qcc Enhancements to SRP and Centralization Management

The IEEE 802.1Qat Stream Reservation Protocol (SRP) [12], which has been merged into 802.1Q, provides a fundamental part of TSN. In particular, IEEE 802.1Qat specifies the admission control framework for admitting or rejecting flows based on flow resource requirements and the available network resources. Moreover, IEEE 802.1Qat specifies the framework for reserving network resources and advertising streams in packet switched networks over full-duplex Ethernet links. Most of the standards that use priorities, frame scheduling, and traffic shaping protocols depend on SRP [12], since these protocols work correctly only if the network resources are available along the entire path from the sender (talker) to the receivers (listeners). IEEE 802.1Qat is a distributed protocol that was introduced by the AVB TG to ensure that the AVB talker is guaranteed adequate network resources along its transmission path to the listener(s). This is accomplished using the Multiple Registration Protocol (MRP) [19, Section 10], where the traffic streams are identified and registered using a 48-bit Extended Unique Identifier (EUI-48). The EUI-48 is usually the MAC source address concatenated with a 16-bit handle to differentiate different streams from the same source and is also referred to as *StreamID*. The SRP reserves resources for a stream based on the bandwidth requirement and the latency traffic class using three signaling protocols, namely 1) the Multiple MAC Registration Protocol (MMRP), 2) the Multiple VLAN Registration Protocol (MVRP), and 3) the Multiple Stream Registration Protocol (MSRP) [12, 19, Section 35].

MMRP and MVRP control the group registration propagation and the VLAN membership (MAC address information [19, Sections 10 and 11]), while MSRP conducts the distributed network resource reservation across bridges and end stations. MSRP registers and advertises data stream characteristics and reserves bridge resources to provide the appropriate QoS guarantees according to the talker's declared propagation attributes, which include the SRP parameters that are sent by the end station in MSRP PDUs (MSRPDUs). A station (talker) sends a reservation request with the MRP, i.e., the general MRP application which registers the stream resource reservation. The 802.1 TSN TG has developed the MRP Attribute Declaration (MAD) for describing the request based on the stream characteristics. All participants in the stream have an MSRP application and MAD specification and each bridge within the same SRP domain can map, allocate, and forward the stream with the necessary resources using the MRP attribute propagation (MAP) [12]. Fig. 2.5 illustrates the MRP architecture.

In essence, the SRP protocol ensures QoS constraints for each stream through the following steps:

1. Advertise the stream.

2. Register the paths of the stream.

3. Calculate worst-case latency of the stream.

4. Establish an AVB domain.

5. Reserve the bandwidth for the stream.

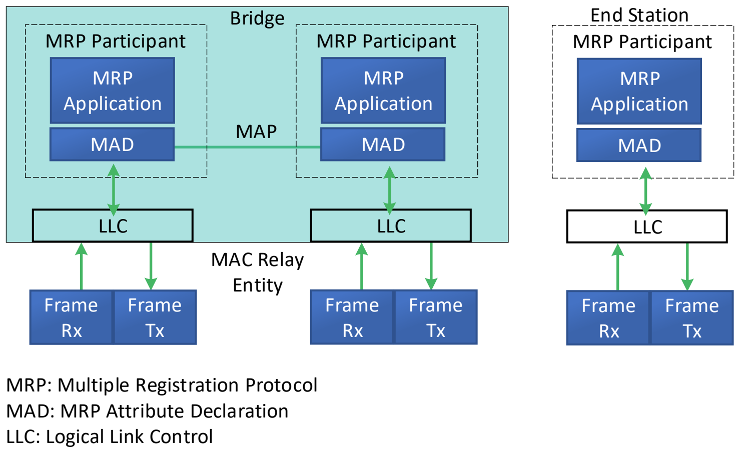


Figure 2.5: Illustration of Multiple Registration Protocol (MRP) architecture: Each end station (illustrated on the right) declares the propagation attributes using the MRP Attribute Declaration (MAD) and the MRP Applications encapsulated as an MRP participant which gives end stations the ability to register resources. The MRP participant entry is stored in bridges and mapped between all required ports using MRP Attribute Propagation (MAP). A bridge mapping between two different interfaces in the LAN is illustrated on the left.

Since the existing IEEE 802.1Qat (802.1Q Section 35) SRP features a decentralized registration and reservation procedure, any changes or new requests for registrations or de-registrations can overwhelm the network and result in intolerable delays for critical traffic classes. Therefore, the TSN TG has introduced the IEEE 802.1Qcc standard to improve the existing SRP by reducing the size and frequency of reservation messages, i.e., relaxing timers so that updates are only triggered by link state or reservation changes.

Additionally, IEEE 802.1Qcc [24] provides a set of tools to manage and control the network globally. In particular, IEEE 802.1Qcc enhances the existing SRP with a User Network Interface (UNI) which is supplemented by a Centralized Network Configuration (CNC) node, as shown in Fig. 2.6. The UNI provides a common method of requesting layer 2 services. Furthermore, the CNC interacts with the UNI to

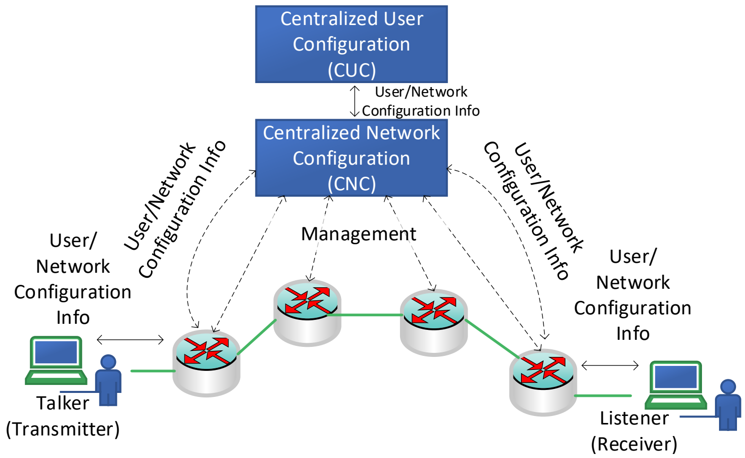


Figure 2.6: Illustration of Centralized Network Configuration (CNC): End stations interact with the network entities via the User-Network Interface (UNI). The CNC receives the requests, e.g., flow reservation requests, and provides corresponding management functions. An optional CUC provides delay-optimized configuration, e.g., for closed-loop IACS applications. The solid arrows represent the protocol, e.g., YANG or TLV, that is used as the UNI for exchanging configuration information between Talkers/Listeners (users) and Bridges (network). The dashed arrows represent the protocol, e.g., YANG or TLV, that transfers configuration information between edge bridges and the CNC.

provide a centralized means for performing resource reservation, scheduling, and other types of configuration via a remote management protocol, such as NETCONF [151] or RESTCONF [83]; hence, 802.1Qcc is compatible with the IETF YANG/NETCONF data modeling language.

For a fully centralized network, an optional Centralized User Configuration (CUC) node communicates with the CNC via a standard Application Programming Interface (API), and can be used to discover end stations, retrieve end station capabilities and user requirements, and configure delay-optimized TSN features in end stations (mainly for closed-loop IACS applications). The interactions with higher level reservation protocols, e.g., RSVP, are seamless, similar to how the AVB Transport Protocol IEEE 1722.1 [16] leverages the existing SRP.

802.1Qcc [24] still supports the fully distributed configuration model of the original SRP protocol, i.e., allows for centrally managed systems to coexist with decentralized ad-hoc systems. In addition, 802.1Qcc supports a "hybrid" configuration model, allowing a migration path for legacy AVB devices. This hybrid configuration management scheme when coupled with IEEE 802.1Qca Path Control and Reservation (PCR) (see Section 2.2.5) and the TSN shapers can provide deterministic end-to-end delay and zero congestion loss.

IEEE 802.1CS Link-Local Reservation Protocol (LRP)

To effectively achieve tight bounds on latency and zero congestion loss, traffic streams need to utilize effective admission control policies and secure resource registration mechanisms, such as the SRP [12] and the SRP enhancements and management standard [24]. While the MRP [19, Section 10] provides efficient methods for registering streams; the database holding the stream state information, is limited to about 1500 bytes. As more traffic streams coexist and the network scale increases, MRP slows significantly as the database increases proportionally which results in frequent cyclic exchanges through the MAD between all bridge neighbors.

The Link-Local Reservation Protocol (LRP) [168] has been introduced by the 802.1 TSN TG to efficiently replicate an MRP database between two ends of a point-to-point link and to incrementally replicate changes as bridges report new network developments or conditions. Additionally, the LRP provides a purging process that deletes replicated databases when the source of such databases remains unresponsive or the data gets stale. Furthermore, the LRP is optimized to efficiently handle databases on the order of 1 Mbyte.

While MRP is considered application specific, i.e., the MRP operations are defined by each registered application, LRP is an application neutral transport protocol. Fig 2.7 illustrates the LRP protocol architecture operating within bridges or end points.

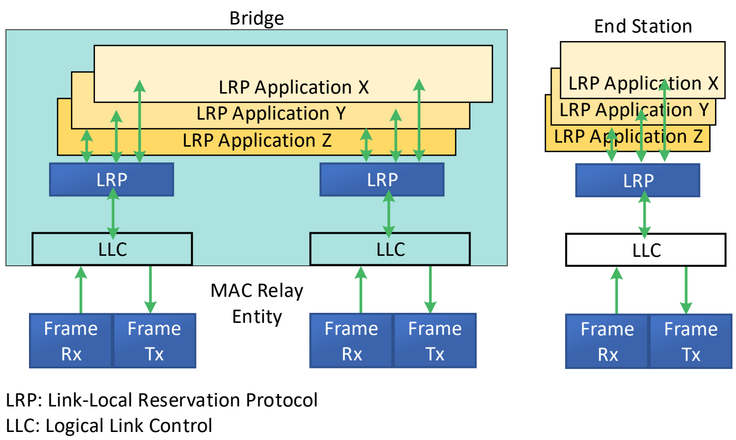


Figure 2.7: Illustration of LRP Architecture: A Link-Local Reservation Protocol (LRP) instance (illustrated by the blue LRP box) interacts with each application and provides a generic transport service for multiple registered LRP applications, which are represented by yellow colored boxes near the top of the illustration.

Resource Allocation Protocol (RAP)—Towards a Distributed TSN Control Model

Although the SRP and the related MSRP (MSRPv1 [24]) were designed for distributed stream configuration (including registration, reservation, and provisioning), SRP is generally restricted to A/V applications with a limited number of Stream Reservation (SR) classes, e.g., classes A and B for the Credit Based Shaper (CBS), see Section 2.2.4. SRP guarantees the QoS characterized by each stream through the reservation in conjunction with shaper mechanisms, see Section 2.2.4. IEEE 802.1Qcc pushed for more centralized configuration models, where all the newly established TSN features, e.g., shaping, preemption, and redundancy, are supported through the CNC configuration model. Any distributed model is currently restricted to CBS.

The Resource Allocation Protocol (RAP) [102] leverages the LRP to propagate TSN stream configuration frames that include resource reservation and registration information in a manner similar to MSRP. The MSRP (and MSRPv1) is geared towards AVB systems, while RAP is defined for TSN enabled systems for distributed stream configuration. The RAP promises to improve scalability (through LRP), to support all TSN features, to improve performance under high utilization, and to enhance diagnostic capabilities.

Summary and Lessons Learned

Flow management allows distributed (legacy SRP and RAP) as well as centralized (802.1Qcc and 802.1CS) provisioning and management of network resources, effectively creating protected channels over shared heterogeneous networks. Moreover, flow management offers users and administrators Operations, Administration, Maintenance (OAM) functions to monitor, report, and configure (802.1Qcp and 802.1Qcc) network conditions. This allows for fine-grained support of network services while enforcing long term allocations of network resources with flexible resource control through adaptive and automatic reconfigurations.

However, both centralized and distributed flow management models have specific deployment advantages and disadvantages. For example, a centralized entity presents a single point of failure, whereas, distributed schemes incur extensive control plane overhead. A centralized scheme can benefit from SDN implementation and management but could result in new infrastructure cost for the operators. Nevertheless, the choice of deployments can be based on the relative performance levels among centralized and distributed nodes, as well as the use of existing infrastructure and the deployment of new infrastructure. Future research needs to thoroughly examine these trade-offs.

Another important future research direction is to examine predictive models that estimate the resource reservation requirements in bridges. Estimations may help in effectively managing queues and scheduling while efficiently utilizing the network resources.

2.2.4 Flow Control

Flow control specifies how frames belonging to a prescribed traffic class are handled within TSN enabled bridges.

IEEE 802.1Qav Forwarding and Queuing of Time-Sensitive Streams

IEEE 802.1Qav specifies Forwarding and Queuing of Time Sensitive Streams (FQTSS), which has been incorporated into 802.1Q. IEEE 802.1Qav serves as a major enhancement to the forwarding and queuing operation in traditional Ethernet networks. IEEE 802.1Qav specifies bridge operations that provide guarantees for time-sensitive (i.e., bounded latency and jitter), lossless real-time audio/video (A/V) traffic [11]. The IEEE 802.1Qav standard [11, 19, Section 34], details flow control operations, such as per priority ingress metering and timing-aware queue draining algorithms.

IEEE 802.1Qav was developed to limit the amount of A/V traffic buffering at the downstream receiving bridges and/or end stations. Increasing proportions of bursty multimedia traffic can lead to extensive buffering of multimedia traffic, potentially resulting in buffer overflows and packet drops. Packet drops may trigger retransmissions, which increase delays, rendering the re-transmitted packets obsolete and diminishing the Quality of Experience (QoE).

IEEE 802.1Qav limits the amount of buffering required in the receiving station through the Stream Reservation Protocol (SRP) [12] in conjunction with a credit-based shaper (CBS). The CBS spaces out the A/V frames to reduce bursting and bunching. This spacing out of A/V frames protects best-effort traffic as the maximum AVB stream burst is limited. The spacing out of A/V frames also protects the AVB traffic by limiting the back-to-back AVB stream bursts, which can interfere and cause congestion in the downstream bridge.

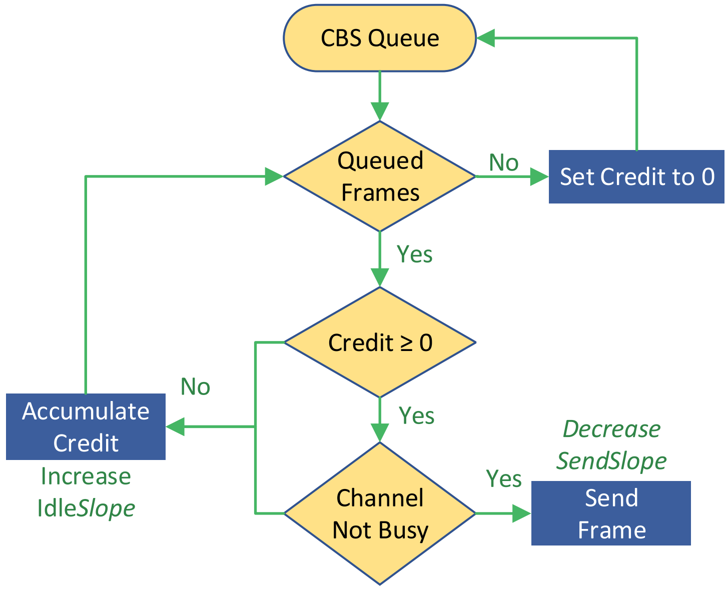


Figure 2.8: Flow-chart illustration of the Credit-Based Shaper (CBS) operation for a given queue. A queue is permitted to transmit if both credits are greater than or equal to zero, and the channel is vacant.

The CBS shaper separates a queue into two traffic classes, class A (tight delay bound) and class B (loose delay bound). Each class queue operates according to the throttling mechanism illustrated in Fig. 2.8. When no frame is available in the queue, the credit for the queue is set to zero. A queue is eligible for transmission if the credit is non-negative. The credit is increased by *idleSlope* when there is at least one frame in the queue, and decreased by *sendSlope* when a frame is transmitted. The *idleSlope* is the actual bandwidth reserved (in bits per second) for the specific queue and traffic class within a bridge [19, Section 34], while the *sendSlope* is the port transmit rate (in bits per second) that the underlying MAC service supports. Furthermore, two key limiting parameters are defined: *i*) *hiCredit* and *ii*) *loCredit*, which are functions of the maximum frame size (in the case of *loCredit*) and maximum interference size (in the case of *hiCredit*), the *idleSlope*/*sendSlope* (respectively), and the maximum port transmit rate. Further details can be found in [11, Annex L]. The CBS throttles each shaped traffic class to not exceed their preconfigured bandwidth limits (75%

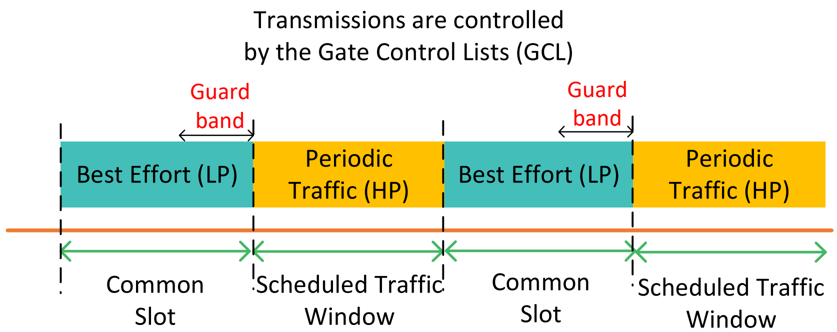


Figure 2.9: IEEE 802.1Qbv Time-Aware Shaper (TAS) [22]: Scheduled traffic is sent over synchronized Time-Division Multiplexing "windows" within the Ethernet traffic. Yellow marked frames are time-sensitive high priority (HP) traffic that have guaranteed reserved resources across the network, while the blue frames correspond to best-effort low priority (LP) traffic.

of maximum bandwidth due to bandwidth intensive applications, e.g., audio and video [11, Section 34.3.1]). The CBS in combination with the SRP is intended to bound delays to under 250 μs per bridge [12]. Overall, the IEEE 802.1Qav Ethernet AVB standard guarantees worst-case latencies under 2 ms for class A and under 50 ms for class B up to seven network hops [11].

However, some key CBS disadvantages are that the average delay is increased and that the delay can be up to 250 μs per hop, which may be too high for industrial control applications [395]. Also, CBS struggles to maintain delay guarantees at high link utilizations.

In order to address the CBS shortcomings, the TSN TG has introduced other standards, e.g., IEEE 802.1Qbv, 802.1Qch, and 802.1Qcr, which are reviewed in the following subsections. Also, addressing the CBS shortcomings is an active research area, see Section 2.3.3.

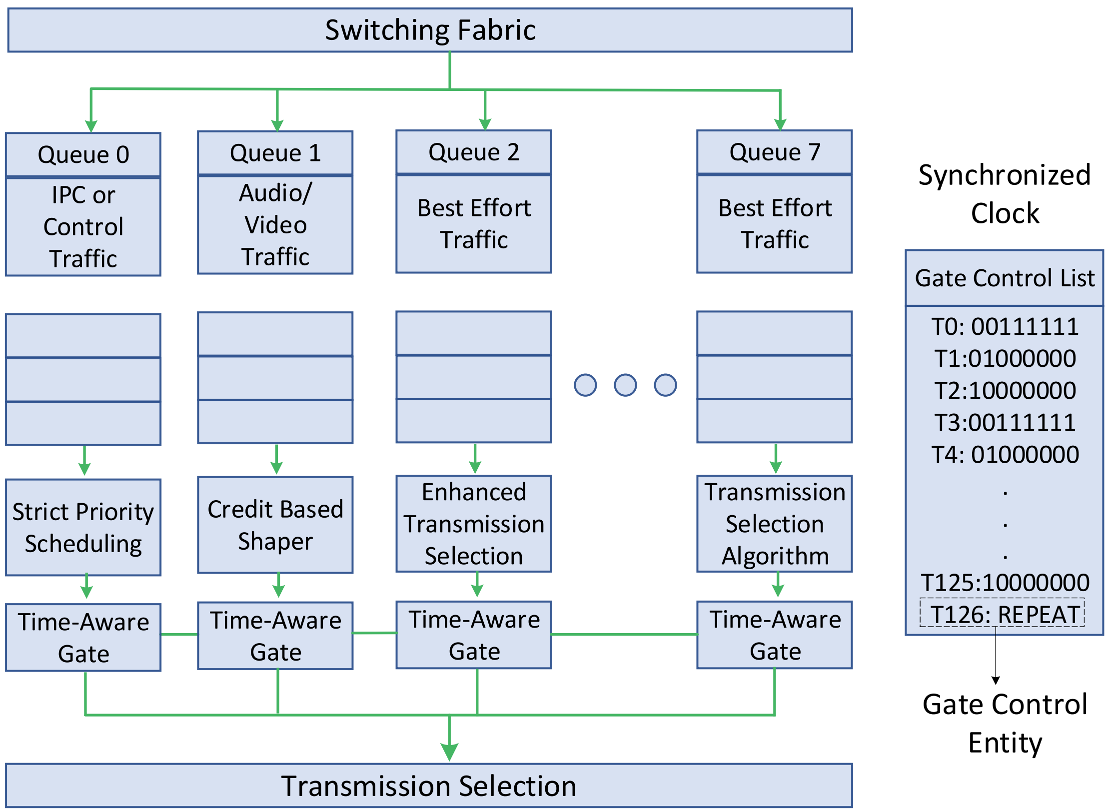


Figure 2.10: IEEE 802.1Qbv: Illustration of egress hardware queue with 8 software queues, each with its unique transmission selection algorithm. The transmissions are controlled by the Gate Controlled List (GCL) with multiple Gate Control Entries (GCEs) that determine which software queues are open. For instance, in time interval T0, the gates for queues 2 through 7 are open, and the transmission selection at the bottom arbitrates access to the medium [19, Section 8.6.8]. In time interval T1, the gate opens for AV traffic from Queue 1, and a credit based shaper (CBS) regulates the frame transmissions from Queue 1. In time interval T2, the gate opens for Queue 0 and strict priority scheduling selects the frames to transmit from Queue 0.

IEEE 802.1Qbv Enhancements to Traffic Scheduling: Time-Aware Shaper (TAS)

As a response to the IEEE 802.1Qav shortcomings, the TSN task group proposed a new traffic shaper, namely the IEEE 802.1Qbv Time-Aware Shaper (TAS) [22] along with the IEEE 802.1Qbu frame preemption technique [21] to provide fine-grained QoS [389]. The TAS and frame preemption mechanisms are suitable for traffic with deterministic end-to-end ULL requirements, e.g., for critical control or Inter-Process

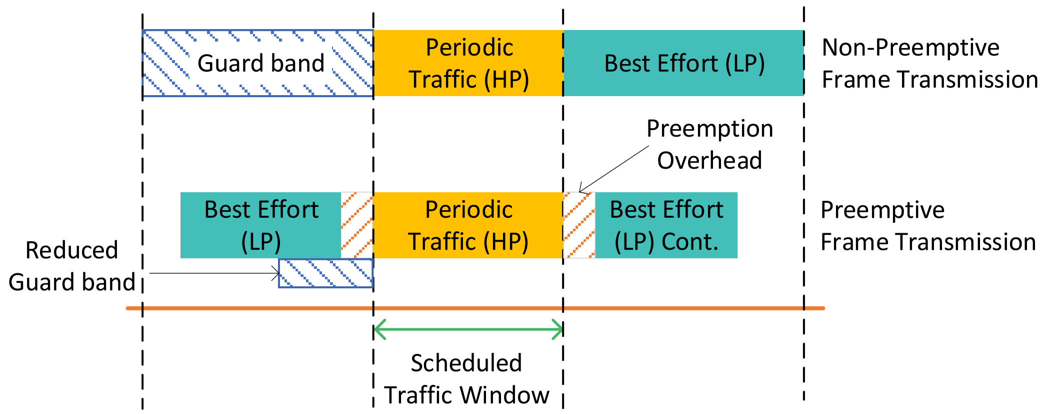


Figure 2.11: The IEEE 802.1Qbv transmission selection prevents low priority (best effort) frames from starting transmission if the transmission cannot be completed by the start of the scheduled traffic window. This transmission selection essentially enforces a guard band (sized as a maximum size frame) to protect the scheduled traffic window. With preemption (IEEE 802.3br, IEEE 802.1Qbu) the guard band can be reduced to the smallest Ethernet frame fragment.

Communication (IPC) traffic, with sub-microsecond latency requirements. In particular, the TAS schedules critical traffic streams in time-triggered windows, which are also referred to as protected traffic windows or as time-aware traffic windows. Thus, TAS follows the TDMA paradigm, similar to Flexible Time-Triggered Ethernet (FTT-E) [297, 347], whereby each window has an allotted transmission time as shown in Fig. 2.9. In order to prevent lower priority traffic, e.g., best effort traffic, from interfering with the scheduled traffic transmissions, scheduled traffic windows are preceded by a so-called guard band.

TAS is applicable for ULL requirements but needs to have all time-triggered windows synchronized, i.e., all bridges from sender to receiver must be synchronized in time. TAS utilizes a gate driver mechanism that opens/closes according to a known and agreed upon time schedule, as shown in Fig. 2.10, for each port in a bridge. In particular, the Gate Control List (GCL) in Fig. 2.10 represents Gate Control Entries (GCEs), i.e., a 1 or 0 for open or closed for each queue, respectively. The frames of a given egress queue are eligible for transmission according to the GCL, which is synchronized in time through the 802.1AS time reference. The GCL is executed in periodically repeating cycle times, whereby the each cycle time contains one GCL execution. Within a cycle time, the time period during which a gate is open is referred to as the time-aware traffic window. Frames are transmitted according to the GCL and transmission selection decisions, as illustrated in Fig. 2.10. Each individual software queue has its own transmission selection algorithm, e.g., strict priority queuing (which is the default). Overall, the IEEE 802.1Qbv transmission selection at the bottom of Fig. 2.10 transmits a frame from a given queue with an open gate if: (*i*) The queue contains a frame ready for transmission, (*ii*) higher priority traffic class queues with an open gate do *not* have a frame to transmit, and (*iii*) the frame transmission can be completed before the gate closes for the given queue. Note that these transmission selection conditions ensure that low priority traffic is allowed to *start* transmission only if the transmission will be completed by the start of the scheduled traffic window for high priority traffic. Thus, this transmission selection effectively enforces a "guard band" to prevent low priority traffic from interfering with high priority traffic, as illustrated in Fig 2.11.

One critical TAS shortcoming is that some delay is incurred due to additional sampling delay, i.e., due the waiting time until the next time-triggered window commences. This sampling delay arises when unsynchronized data is passed from an end-point to the network. Task and message scheduling in end-nodes would need to be coupled with the TAS gate scheduling in the networks in order to achieve the lowest latencies. Moreover, synchronizing TSN bridges, frame selections, and transmission times across the network is nontrivial in moderately sized networks, and requires a fully managed network. Also, the efficient use of bandwidth with TAS needs to be thoroughly examined. Overall, TAS has high configuration complexity. Future research needs to carefully examine the scalability to large networks, runtime

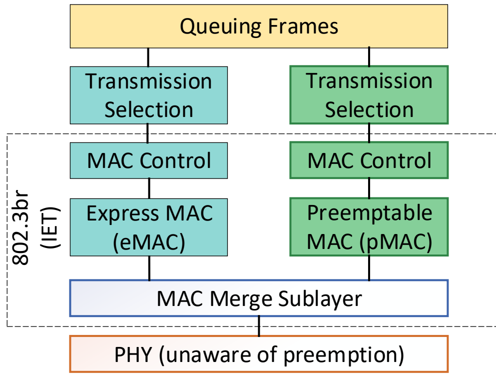


Figure 2.12: Illustration of the layering for the Ethernet MAC Merge Sublayer: The MAC Merge Sublayer provides a Reconciliation Sublayer (RS) service for pMAC and eMAC frames. The RS service supports two main ways to hold the transmission of a pMAC frame in the presence of an eMAC frame: By preempting (interrupting) the pMAC frame transmission, or by preventing the start of the pMAC frame transmission.

reconfiguration, and the integration of independently developed sub-systems.

IEEE 802.3br and 802.1Qbu Interspersing Express Traffic (IET) and Frame Preemption

To address the ULL latency requirements and the inverted priority problem, i.e., the problem that an ongoing transmission of a low priority frame prevents the transmission of high priority frames, the 802.1 TG along with the 802.3 TG introduced frame preemption (802.1Qbu and 802.3br) [20, 21]. Frame preemption separates a given bridge egress port into two MAC service interfaces, namely preemptable MAC (pMAC) and express MAC (eMAC), as illustrated in Fig. 2.12. A frame preemption status table maps frames to either pMAC or eMAC; by default all frames are mapped to eMAC. Preemptable frames that are in transit, i.e., they are holding on to the resource (transmission medium), can be preempted by express frames. After the transmission of an express frame has completed, the transmission of the preempted frame can resume.

With preemption, the guard band in Fig. 2.9 can be reduced to the transmission time of the shortest low priority frame fragment. Thus, in the worst case, the transmission of the low priority frame fragment can be completed before starting the transmission of the next high priority frame. The transmission of the leftover fragmented frame can then be resumed to completion. Note that this preemption occurs only at the link-level, and any fragmented frame is reassembled at the MAC interfaces. Hence the switches process only complete frames internally. That is, any frame fragments transmitted over a physical link to the next bridge are re-assembled in the link layer interface; specifically, the MAC merge sublayer (see Fig. 2.12) in the link layer of the next bridge, and the next bridge then only processes complete frames. Each preemption operation causes some computational overhead due to the encapsulation processing by the bridge to suspend the current fragment and to transition the operational context to the express traffic frame and vice versa, which is illustrated in Fig. 2.11. Note that this overhead occurs only in layer 2 in the link interface.

IEEE 802.1Qch Cyclic Queuing and Forwarding (CQF)

While the IEEE 802.1Qav FQTSS with CBS works well for soft real-time constraints, e.g., A/V traffic, the existing FQTSS has still several shortcomings, including, *i*) pathological topologies can result in increased delay, and *ii*) worst-case delays are topology dependent, and not only hop count dependent, thus buffer requirements in switches are topology dependent. The TSN TG introduced Cyclic Queuing and Forwarding (CQF) [26], also known as the Peristaltic Shaper (PS), as a method to synchronize enqueue and dequeue operations. The synchronized operations effectively allow LAN bridges to synchronize their frame transmissions in a cyclic manner, achieving zero congestion loss and bounded latency, independently of the network topology.

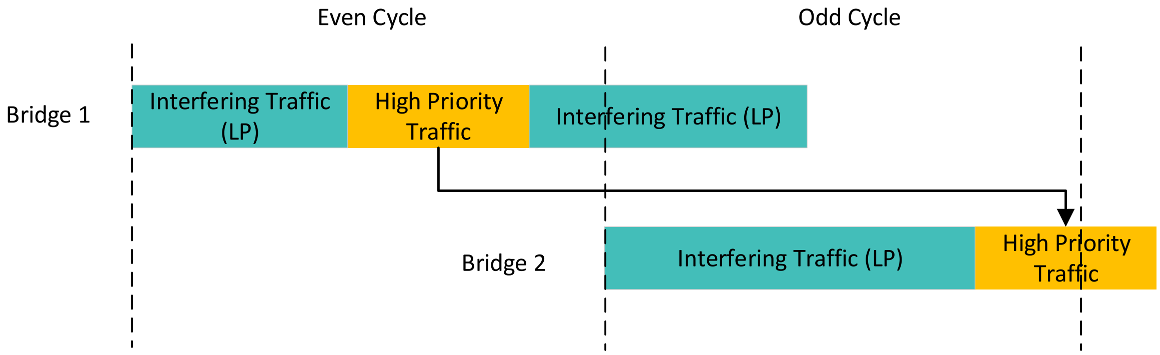


Figure 2.13: Illustration of Cyclic Queuing and Forwarding (CQF) without preemption for a linear network: Each High Priority (HP) traffic frame scheduled on a cycle (even or odd) is scheduled to be received at the next bridge in the next cycle, whereby the worst-case HP frame delay can be two cycle times. In the illustrated example, the HP traffic is delayed due to low priority interfering traffic, but still meets the two cycle time delay bound.

Suppose that all bridges have synchronized time, i.e., all bridges are 802.1AS enabled bridges, and suppose for simplicity of the discussions that wire lengths and propagation times are negligible. Then, time sensitive streams are scheduled (enqueued and dequeued) at each time interval or cycle time with a worst-case deterministic delay of two times the cycle time between the sender (talker) and the downstream intermediate receiver, as illustrated in Fig. 2.13. In essence, the network transit latency of a frame is completely characterized by the cycle time and the number of hops. Therefore, the frame latency is completely independent of the topology parameters and other non-TSN traffic.

CQF can be combined with frame preemption specified in IEEE 802.3Qbu, to reduce the cycle time from the transmission time of a full size frame to the transmission time of a minimum size frame fragment (plus all the TSN traffic), as illustrated in Fig. 2.14. Note however that for CQF to work correctly, all frames must be kept to their allotted cycles, i.e., all transmitted frames must be received during the expected cycle at the receiving downstream intermediate bridge [26]. Therefore, the

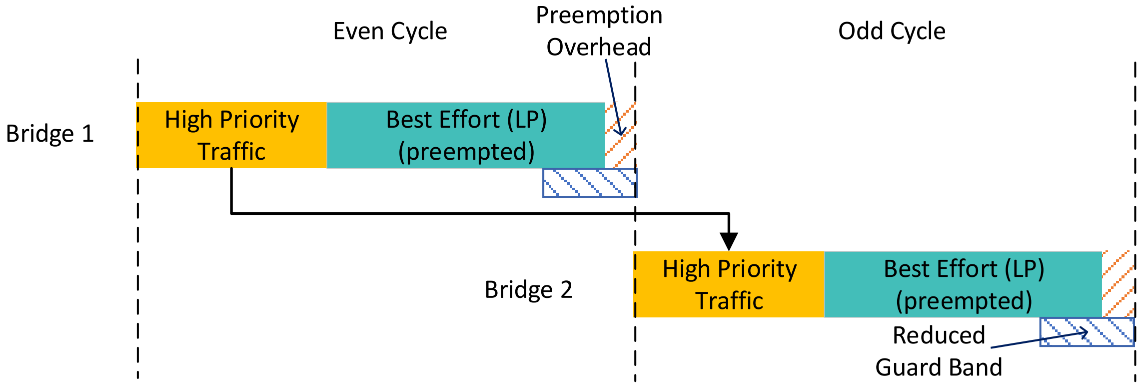


Figure 2.14: Illustration of CQF with preemption for a linear network: A Guard Band (GB) before the start of the cycle prevents any interfering (LP) traffic from affecting the High Priority (HP) traffic. The CQF without preemption in Fig. 2.13 did not prevent the LP traffic from interfering with HP traffic, while the CQF with preemption prevented the LP traffic from interfering with HP traffic. Thus, preemption can improve the performance for HP traffic.

cycle times, the alignment of the cycle times among the bridges in the network, and the timing of the first and last transmissions within a cycle need to be carefully considered in order to ensure that the desired latency bounds are achieved. To this end, CQF in conjunction with IEEE 802.1Qci ingress policing and the IEEE 802.1Qbv TAS ensures that all frames are kept within a deterministic delay and guaranteed to be transmitted within their allotted cycle time.

IEEE 802.1Qcr Asynchronous Traffic Shaping (ATS)

While CQF and TAS provide ULL for critical traffic, they depend on network-wide coordinated time and, importantly, due to the enforced packet transmission at forced periodic cycles, they utilize network bandwidth inefficiently [395]. To overcome these shortcomings, the TSN TG has proposed the IEEE 802.1Qcr Asynchronous Traffic Shaper (ATS) [394], which is based on the urgency-based scheduler (UBS) [395, 396]. The ATS aims to smoothen traffic patterns by reshaping TSN streams per hop, implementing per-flow queues, and prioritizing urgent traffic over relaxed traffic. The

ATS operates asynchronously, i.e., bridges and end points need not be synchronized in time. Thus, ATS can utilize the bandwidth efficiently even when operating under high link utilization with mixed traffic loads, i.e., both periodic and sporadic traffic.

The UBS is based on the Rate-Controlled Service Disciplines (RCSDs) [452]. RCSDs are a non-work conserving class of packet service disciplines which includes Rate-Controlled Static Priority [451] and Rate-Controlled Earliest Deadline First [184]. The RCSD packet scheduling consist of two components: the rate controller implements the rate-control policies, and the scheduler implements the packet scheduling according to some scheduling policy, e.g., Static-Priority, First-Come-First-Serve, or Earliest Due-Date First. By separating the rate controller and scheduler, the RCSD effectively decouples the bandwidth for each stream from its delay bound, i.e., allocating a prescribed amount of bandwidth to an individual stream is independent of the delay bound. Hence, RCSD can support low delay and low bandwidth streams.

UBS adds a few improvements to RCSDs [452], namely: 1) UBS provides low and predictable worst-case delays even at high link utilization, 2) low implementation complexity due to the separation of per-flow queues from per-flow states where flow state information, such as Head-of-Queue frame and time stamp, is stored, and 3) independence from the global reference time synchronization; specifically, individual flow delays are analyzed at each hop, i.e., per-hop delay calculation, and end-to-end delays are calculated based on the network topology and by the closed-form composition of the per-hop delays calculated initially.

The fundamental aim of the RCSD is to individually control frame selection and transmission at each hop between the transmitter and receiver, i.e., per hop shaping. As pointed out by Specht et al. [395], the RCSD has multiple scalability problems, including dynamic reordering of packets within separate queues according to the packets' eligibility times, i.e., priority queue implementation with non-constant complex data structures, such as heaps. Specialized calender queues have been proposed to achieve constant complexity [395]. However, calender queues require RCSD capable switches to have large memory pools, are difficult to control as the network size scales up, and are ideal only for some specific applications with special properties. Therefore, Specht et al. [395] utilize the RCSD concept with the outlined improvements and have proposed a novel UBS solution as the core of the ATS standard.

Summary and Lessons Learned

Flow control mainly enforces rules to efficiently forward and appropriately queue frames according to their associated traffic classes. All existing flow controls follow similar principles, namely, certain privileges are associated with TSN flows while non-TSN flows are delayed. Nearly all existing schedulers and shapers enforce fair transmission opportunities according to each flow's traffic class. The transmission selection algorithm selects the appropriate stream within a given traffic class according to the network and traffic conditions. Flow control collaborates with flow management, see Section 2.2.3, and flow integrity, see Section 2.2.5, to ensure adequate resources are available for TSN streams.

Overall, we can classify real-time TSN systems into event-triggered systems and time-triggered systems. For example, IEEE 802.1Qbv is a time-triggered shaper, while IEEE 802.1Qcr is an event-triggered shaper. An interesting future research direction is to explore whether both types of shapers can be combined. That is, would it be efficient to dynamically change a flow's priority, individually or collectively, and to reshape flows based on neighbor network conditions while each flow is shaped by a centralized computed schedule incorporating time slots at each egress's port? For example, a stream initially sent with a certain high priority can be downgraded to low priority based on downstream network conditions while adhering to each bridge's time-aware scheduler and gating mechanism.

Also, it will be interesting to investigate whether IEEE 802.1Qbv can be replaced with an event-triggered shaper that guarantees an upper bound on latency, but not generally a deterministic latency. Changing TAS into an event-triggered shaper can lead to more flexible and easily computed schedules since certain events, e.g., incoming frames or network changes, can require schedule changes at runtime.

2.2.5 Flow Integrity

To accomplish the goals of deterministic ultra-low latency, jitter, and packet loss, TSN streams need to deliver their frames regardless of the dynamic network conditions, including physical breakage and link failures. Several techniques have been standardized to enable flow integrity.

IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER)

IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER) [27], is a stand-alone standard that ensures robust and reliable communication using proactive measures for applications that are intolerant to packet losses, such as control applications. 802.1CB FRER minimizes the impact of congestion and faults, such as cable breakages, by sending duplicate copies of critical traffic across disjoint network paths, as shown in Fig. 2.15. If both frames reach their destination, the duplicate copy is eliminated. If one copy fails to reach its destination, the duplicate message can still be received, effectively providing seamless proactive redundancy at the cost of additional network resources.

In order to minimize network congestion, the packet replication can be selected based on traffic class and the path information acquired through the TSN stream identification (*stream_handle*), plus a sequence generation function. The sequence

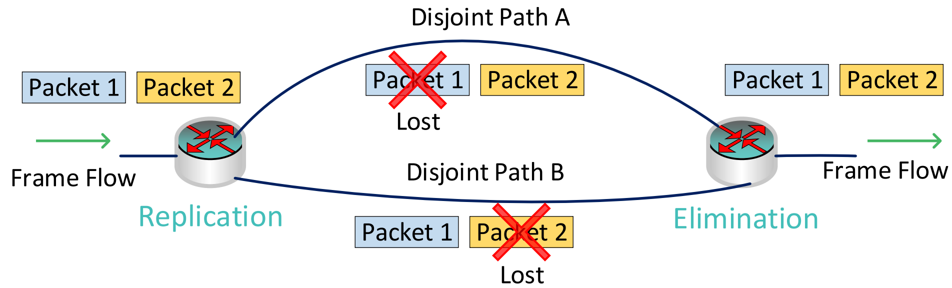


Figure 2.15: Illustration of FRER operation: The first bridge replicates the frame and transmits the duplicated frames on two disjoint paths. The FRER operation can be started and ended at any bridge between the sender and receiver.

generation function generates identification numbers for replicated frames to determine which frames to discard and which frames to pass on so as to ensure correct frame recovery and merging. The frame redundancy information is carried in a Redundancy Tag [27]. Frame sequence numbers and timing information are also needed to limit the memory needed for duplicate frame detection and elimination. For example, FRER may only be employed for critical traffic, while best effort and other loss-tolerant traffic is transmitted normally. FRER is compatible with industrial fault-tolerance architectures, e.g., High Availability and Seamless Redundancy (HSR) [232] and the Parallel Redundancy Protocol (PRP) [231]. We note that frame duplication, routing, and elimination are non-trivial tasks that will likely require centralized management. Hence, such protocols can be combined with other standards, e.g., 802.1Qcc and 802.1Qca, to ensure seamless redundancy and fast recovery in time-sensitive networks.

IEEE 802.1Qca Path Control and Reservation (PCR)

IEEE 802.1Qca Path Control and Reservation (PCR) is based on and specifies TLV extensions to the IETF Link State Protocol (LSP), the Intermediate Station to Intermediate Station (IS-IS) protocol [335]. IEEE 802.1Qca allows the IS-IS protocol to control bridged networks beyond the capabilities of shortest path routing (ISIS-SPB) [19, 418, Section 28], configuring multiple paths through the network [23, 159]. IEEE 802.1Qca PCR aims to integrate control protocols required to provide explicit forwarding path control, i.e., predefined protected paths set-up in advance for each stream, bandwidth reservation, data flow redundancy (both protection and restoration), and distribution of control parameters for flow synchronization and flow control messages [23].

In general, 802.1Qca specifies bridging on explicit paths (EPs) for unicast and multicast frame transmission, and protocols to determine multiple active topologies, e.g., Shortest Path, Equal Cost Tree (ECT), Internal Spanning Tree (IST), Multiple Spanning Tree Instance (MSTI), and Explicit Tree (ET), in a bridged network. Explicit forwarding paths, as opposed to hop-by-hop forwarding, mitigate disruptions caused by the reconfiguration of bridging protocols. PCR has similar goals and evolved from spanning tree protocols, e.g., the Rapid Spanning Tree Protocol (RSTP) [19, Section 13.4], the Multiple Spanning Tree Protocol (MSTP) [19, Section 13.5], and the Shortest Path Bridging (SPB) [19, Section 27].

The IEEE 802.1Qca standard is based on Shortest Path Bridging (SPB) [19, Section 27] and incorporates a Software Defined Networking (SDN) hybrid approach [23]. In the hybrid approach, the IS-IS protocol in the data plane handles basic functions, e.g., topology discovery and default path computation, while the SDN controller [70] in the control plane manages the Explicit Paths (EPs), as shown in Fig. 2.16. In particular, the controller utilizes dedicated path computation server nodes called Path Computation Elements (PCEs) [421], defined by the IETF PCE WG [421], to manage the EPs. A PCE interacts with the IS-IS protocol to handle and install requests for the network and can interact with the SRP protocol, see Section 2.2.3, to reserve re-

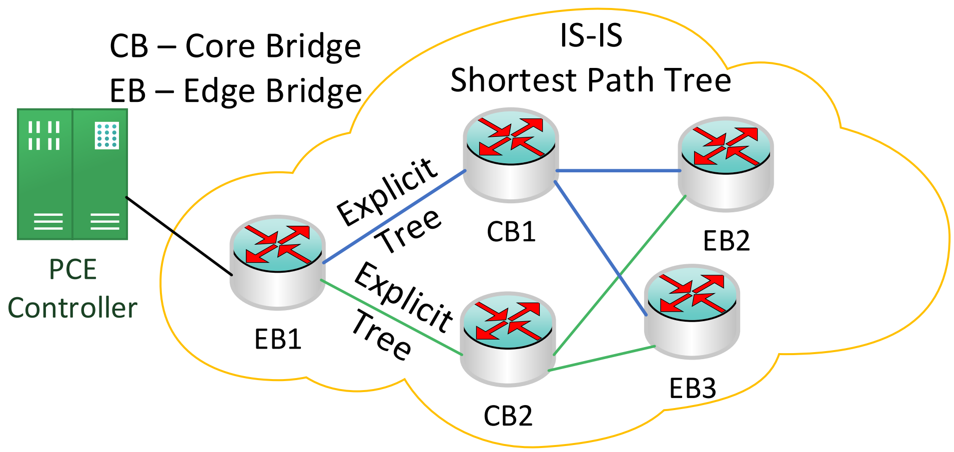


Figure 2.16: Illustration of Explicit Paths (EPs): A control plane PCE SDN controller installs computed Explicit Tree (ET) paths via the IS-IS data plane. Two computed ET paths are shown represented by the green and blue lines.

sources along the EPs. Additionally, the PCEs can manage redundancy on the EPs, thus providing protection on top of the EPs by utilizing alternate paths, e.g., Loop Free Alternates (LFAs) [23], that reroute in a few milliseconds.

IEEE 802.1Qci Per-Stream Filtering and Policing (PSFP)

The IEEE 802.1Qci per-stream filtering and policing (PSFP) standard [25], also known as ingress policing/gating standard, filters and polices individual traffic streams based on rule matching. IEEE 802.1Qci prevents traffic overload conditions, that are caused, for instance, by erroneous delivery due to equipment malfunction and Denial of Service (DoS) attacks, from affecting intermediate bridge ports and the receiving end station, i.e., improves network robustness. IEEE 802.1Qci may be used to protect against software bugs on end points or bridges, but also against hostile devices and attacks. IEEE 802.1Qci specifies filtering on a per flow (stream) basis by identifying individual streams with a *StreamID*, which utilizes the 802.1CB stream handler method [27]. The identified individual streams can then be aggregated, processed, and finally queued to an input gate. As illustrated in Fig. 2.17, each gate performs

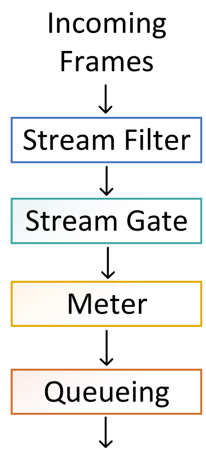


Figure 2.17: Illustration of PSFP flow: The flow is first filtered according to per-flow policies. Then, a gating mechanism regulates the flow. Finally, flow metering ensures bandwidth limitations before a frame is queued for forwarding.

three functions.

The PSFP stream filter performs per-flow filtering by matching frames with permitted stream IDs and priority levels, and then applies policy actions. The PSFP stream gate coordinates all streams such that all frames proceed in an orderly and deterministic fashion, i.e., similar to the 802.1Qch signaling process, see Section 2.2.4. The PSFP flow metering enforces predefined bandwidth profiles for streams. The metering may, for instance, enforce prescribed maximum information rates and burst sizes.

Summary and Lessons Learned

Flow integrity provides path redundancy, multi-path selection, as well as queue filtering and policing. Flow integrity also prevents unauthorized or mismanaged and rogue streams on bridged LANs.

In general, as network devices improve in terms of hardware performance, they can be equipped with more state information within the core network. The increased state information allows for fine grain QoS management at the expense of control messages for efficient control dissemination in the network. Future research needs to carefully examine the trade-offs between disseminating more extensive control messages and the resulting QoS management improvements.

2.2.6 Discussion on TSN Standardization

The IEEE TSN TG has standardized deterministic networking for Layer 2 Ethernet based bridging LANs. These standards have been revised and continue to be updated to reflect the convergence of the industrial and consumer markets. Overall, the TSN standards guarantee the required QoS requirements for data transmission and provide sufficient measures to enable end-to-end functional communication safety in the network. Essentially, the TSN standardization provides the recommended practices for enabling low latency, jitter, and data loss, as well as redundancy and reservation. In addition, the TSN standardization provides mechanisms for bandwidth limitation, dynamic reconfiguration, centralized management, and strict timing features.

Timing measurement and sub-microsecond time synchronization as basis for TSN standard mechanisms can be achieved with IEEE 802.1AS and the updated revised version 802.1AS-REV. Essentially, all gPTP network entities contribute to distributing and correcting delay measurement timing information based on the source GM. 802.1AS-REV provides, among others, GM redundancy for fast convergence.

Several flow management standards, including IEEE 802.1CB (FRER), 802.1Qca (PCR), 802.1Qci (PSFP), 802.1Qcc (Enhanced SRP and centralized Management), 802.1CS (LRP) and RAP have been published or are in progress to enable redundancy, path reservation, bandwidth limitation, dynamic reconfiguration, as well as overall flow integrity and management. Although standard Ethernet provides redundancy features, e.g., through spanning tree protocols, the convergence time in the event of a failure is too slow for real-time IACS applications. Therefore, FRER is used to proactively enable seamless data redundancy at the cost of additional bandwidth consumption. Moreover, PCR in combination with FRER and 802.1Qcc enables fast recovery, efficient path redundancy, and dynamic runtime flow management. Furthermore, PSFP manages, controls, and prevents rogue flows from deteriorating the network performance. Since SRP and the related signaling protocols are fully distributed mechanisms targeted towards AVB applications, the SRP and MRP protocols are not scalable to large networks with real-time IACS applications due to a limited state information database for the registered flows, see Section 2.2.3. Therefore, LRP in conjunction with RAP as the signaling protocol features a decentralized approach to support resource reservations for scalable TSN enabled networks.

To achieve low latency, several flow control standards have been released, including IEEE 802.1Qbv (TAS), 802.1Qch (CQF), and IEEE 802.1Qcr (ATS). For TAS, IEEE 802.1Qbu frame preemption can ensure that the transmission channel is free for the next express traffic transmission. CQF can coordinate ingress and egress operations to reduce the TAS configuration complexity, albeit at the expense of higher delays. Finally, ATS has been proposed to provide deterministic operations independently of the reference time synchronization and low delays for high link utilization. The efficient dynamic configuration of these flow control standard mechanisms, including IEEE 802.1Qbv, is an open challenge that requires extensive future standardization and research efforts.

The TSN mechanisms do not explicitly define mechanisms to specifically reduce packet jitter. The various TSN mechanisms for ensuring very short deterministic packet delays implicitly achieve very low packet jitter. Moreover, resource reservation and admission control can further reduce end-to-end jitter by limiting interfering traffic, which is typically the main cause of jitter. Additionally, CQF can coordinate ingress and egress operations, which can cause jitter, to reduce delays to sub-microsecond levels or to bound delays to within a few microseconds, effectively eliminating jitter caused by the physical properties of links and switching fabrics [321]. However, while it is very unlikely that high jitters occur in a TSN network, in the event of high jitter, the TSN standards do not actively delay or throttle flows to compensate for the high jitter condition. Such specific jitter control operations are an open issue for potential future TSN standards development.

The TSN standardization has so far excluded the specific consideration of security and privacy. The IEEE 802.1 Security TG has addressed security and privacy in general IEEE 802.1 networks, i.e., functionalities to support secure communication between network entities, i.e., end stations and bridges. The TG has detailed a number of standards and amendments, including 802.1X Port-based Network Access Control (PNAC) [13, 18] , 802.1AE MAC Security (MACsec) [8, 15, 17, 28], and 802.1AR Security Device Identity (DevID) [10], that focus on providing authentication, authorization, data integrity, and confidentiality. Specifically, PNAC utilizes industry standard authentication and authorization protocols enabling robust network access control and the establishment of a secure infrastructure. Furthermore, PNAC specifies the MACsec Key Agreement (MKA) [18] protocol. MACsec specifies the use of cryptographic cipher suites, e.g., Galois/Counter Mode of Advanced Encryption Standard cipher with 128-bit key (GCM-AES-128), that allow for connectionless user data confidentiality, frame data integrity, and data origin authentication, essentially providing a set of protocols that ensures protection for data traversing Ethernet LANs. For instance, DevID is a unique per-device identifier that cryptographically binds a device to the DevID. Thus, 802.1 LAN devices can be authenticated and appropriate policies for transmission and reception of data and control protocols to and from devices can be applied. The IEEE 802.1 Security TG is working on amendments to address privacy concerns and to include a YANG model allowing configuration and status reporting for PNAC in 802.1 LANs. The integration of the security protocols and standards with TSN enabled networks needs to be addressed in future research and standardization. For instance, the impact of the security stack overhead on TSN flows and the impact of the security overhead on OT related applications running over Ethernet LANs need to be investigated. Thus, there are ample research opportunities for testing and benchmarking to ensure the efficient integration of legacy security protocols with TSN.

The important area of networks for industrial applications often employs cut-through switching techniques. An interesting future research direction is to investigate how networking with cut-through switching compares with networking based on the TSN standards (tool sets). More broadly, even though many standards and recommended practices addressing deterministic networking have been published, significant testing and benchmarking is needed to provide assurances to the industry and consumer markets.

2.3 TSN Research Studies

This section surveys the existing research studies towards achieving ULL in the context of the TSN standards. The TSN standards provide tool sets to enable TSN characteristics, such as flow synchronization and flow control (see Sec. 2.2), in conventional networks. Based on the application requirements, various TSN standard tools can be independently and selectively adopted on network segments to enable TSN characteristics. Similar to the organization of the review of TSN standards in Fig. 2.1, we organize the survey of TSN related research studies in Fig. 2.18 according to the same classification as the TSN standards in Fig. 2.1. To date there have been no specific research studies on the TSN flow concept; therefore, we omit the flow

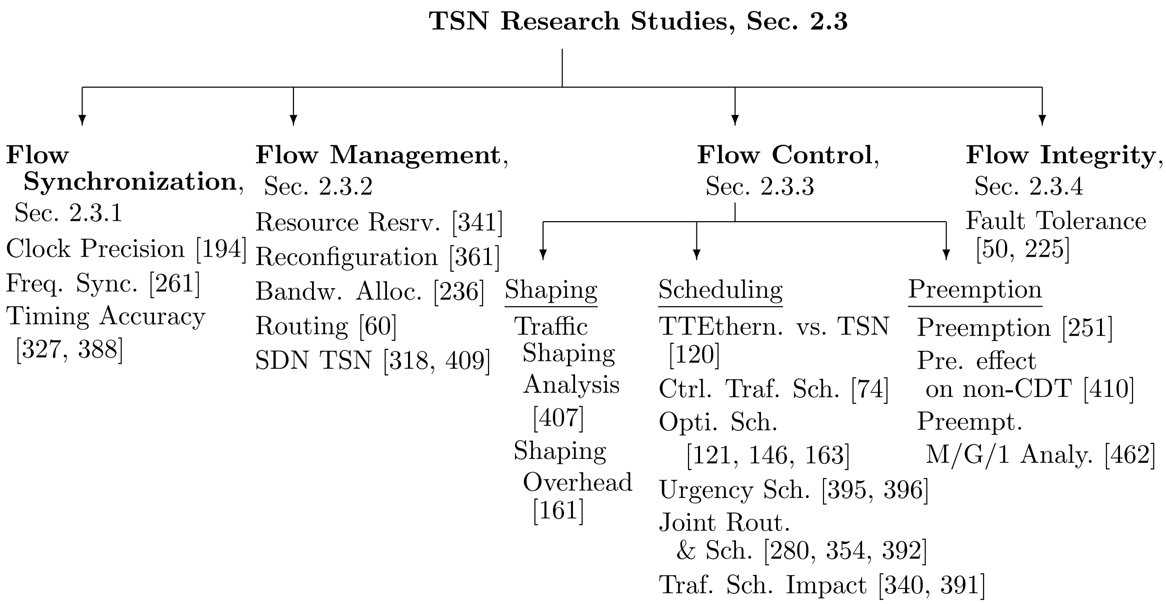


Figure 2.18: Classification of TSN research studies.

concept category in Fig. 2.18.

2.3.1 Flow Synchronization

Clock Precision

Most existing time synchronization implementations are limited to clock precision on the order of sub-microseconds [393]. The global sharing of the timing information across the network elements allows the clocks in the network elements to be precisely synchronized relative to each other (see Section 2.2.2). The challenges associated with network wide clock synchronization are not limited to one particular network attribute. Rather, a wide set of network attributes, including hardware capabilities, such as clock stability, and isolation from environmental impacts e.g., temperature, and software implementations, e.g., for designing an effective closed-loop system to track and correct the timing drifts, influence the synchronization quality in the network as a whole. As a result, most current deployments rely on sub-microsecond clock precision techniques. However, future trends in network applications require a tighter clock synchronization on to the order of sub-nanoseconds in Ethernet networks. For instance, the control system of the CERN Large Hadron Collider (LHC) communication network has to operate with sub-nanosecond precision to share timing and perform time-trigger actions [124].

Gutierrez et al. [194] have analytically evaluated the synchronization process and the quality of the timing error estimation in large scale networks based on the IEEE 802.1AS TSN synchronization standard. In particular, Gutierrez et al. focused on the clock synchronization quality with a small margin of error between each node for a large network consisting of a few thousand nodes with maximum distances between the grandmaster clock and synchronized node clocks spanning up to 100 hops. The study of the protocol behavior included various network aspects, such as clock granularity, network topology, PHY jitter, and clock drift. The results from probabilistic analytical modeling and simulation evaluations indicate that implementation specific aspects, such as PHY jitter and clock granularity, have a significant impact on the clock precision with deviations reaching 0.625 μs in the TSN synchronization process. Therefore, it is critical to ensure that the physical properties of the clock within each node are accurate so as to ensure the overall quality of the synchronization process in TSN networks that adopt IEEE 802.1AS.

Frequency Synchronization

Li et al. [261] have introduced a novel networking device architecture that provides ULL switching and routing based on synchronization. Their design integrates a state-of-the-art FPGA with a standard x86-64 processor (which supports both 32 and 64 bit operation) to support TSN functions. The system provides frequency synchronization over standard Ethernet to the entire network. Frequency synchronization enables distribution of timing information with low-jitter across the network. In the frequency

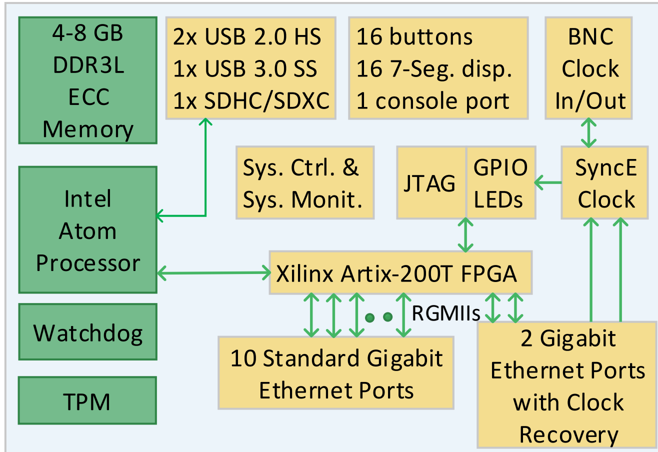


Figure 2.19: Illustration of frequency synchronization design supporting TSN with clock recovery and network wide synchronization [261].

synchronization design illustrated in Fig. 2.19, datapaths are enabled with one or more synchronous modules supported by clock synchronization. These datapaths are allocated resources in terms of bit rate and packet rate based on the worst-case traffic load. This design exploiting hardware synchronization capabilities achieves cut-through latencies of 2 to 2.5 μs for twelve Gigabit Ethernet ports at full line rate packet processing [261]. The constituents of the observed latency were identified as pipeline delay, arbitration delay, aggregation delay, backpressure cycles, cross-clock domain synchronization cycles, datapath width adaptation cycles, and head-of-line blocking cycles. Emphasizing the importance of the hardware implementation of the frequency synchronization process, Li et al. [261] suggest that their novel hardware implementation and timing distribution process based on frequency synchronization across networks can be easily extended to other custom designs.

Timing Accuracy

Although TSN protocols offer very accurate timing information for the inter clock alignment, the validity and accuracy of the received timing information can still be uncertain. That is, typically the timing information received from the grand master is blindly followed by the clock alignment process, which can potentially result in out-of-sync clocks if the received timing information is not accurate. The detection of erroneous timing information by the receiving node can potentially help time critical network applications to re-trigger the verification, calibration, and re-synchronization process. Moreover, nodes can use this information to alert network applications to request a new path or to terminate critical operations that require timing precision. Therefore, timing accuracy is an essential aspect in TSN networks.

The time-error is the relative clock difference between the slave and the grand master. The time-error can still exist even if the slave node applies the timing corrections based on timing error estimation. The timing accuracy represents the overall quality of the timing distribution throughout the network. The timing accuracy at a node can be estimated in two ways: i) by receiving the timing information from another source and periodically comparing to check the accuracy, and ii) keeping track of the node's self error and (ingress and egress) port latencies to predict the inaccuracy in the received timing. Noseworthy et al. [327] have specifically addressed the timing inaccuracy of a Precision-Time Protocol (PTP) node with the help of an auxiliary node. The proposed network-based system monitors and measures the timing errors and port latencies to track the self errors independently of the PTP protocol and network application. Such a system can share the information with other nodes so that the other nodes can estimate the timing errors. In addition to the timing error of a PTP node, the ingress and egress delays in the PTP nodes for a specific TSN flow have been estimated and used in the process of clock reference maintenance. A PTP extension to wireless networks has been investigated in [388] while related measurement techniques have been examined in [241, 242].

Summary and Lessons Learned

An important aspect of timing and synchronization in TSN networks is to estimate the relative timing difference between two nodes. Timing differences may arise because of clock errors, synchronization errors, as well as tracking and estimation errors [256]. Clock errors are caused by the timing drifts resulting from hardware imperfections. Synchronization errors are caused by false timing information and wrong interpretation of timing information. Tracking and estimation errors can, for instance, arise due to sleep states for power savings. In deep-sleep states, only a minimal set of sub-systems is kept alive. Moreover, the clock system is typically switched from high resolution and high precision to low resolution and low precision, which may incur large clock drifts. The repeated switching of the clocking system may accumulate significant synchronizing errors that need to be corrected by external sources. In order to achieve high-order precision in the clock implementation for TSN applications, all aspects of the clock errors must be considered to mitigate the effects arising from incorrect local timing.

The clock synchronization in the network requires significant bandwidth, i.e., imposes a significant overhead in the network. The synchronization data needs to be propagated throughout the network in a deterministic fashion. Hence, the synchronization traffic interferes with the scheduled and regular traffic. Therefore, the design of TSN networks requires careful consideration of the overhead resulting from the synchronization process and efforts to reduce the overhead. On the other hand, the effectiveness of the protocol that facilitates the synchronization process is limited by the node capability to preserve a synchronized local clock. If the local clock skew is high compared to the frequency of the synchronization process, then the local clock will often have the wrong timing. Therefore, the future design of synchronization

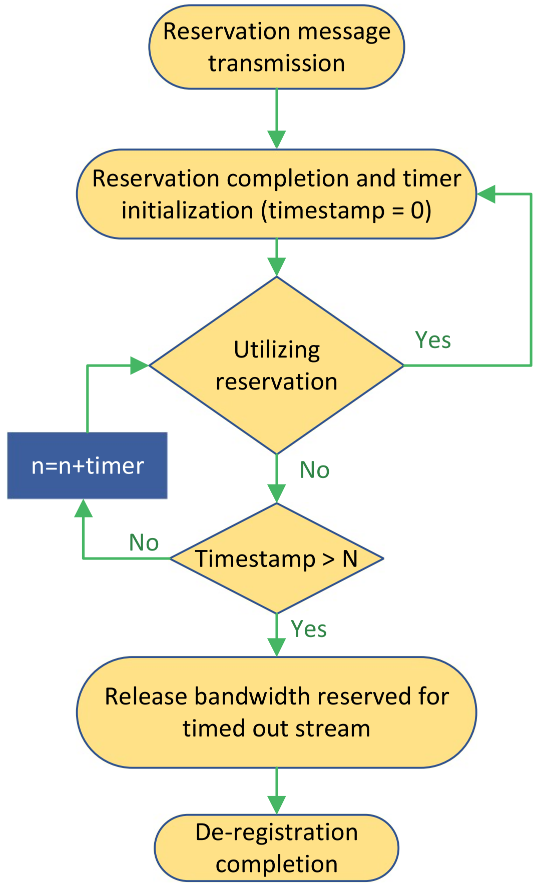


Figure 2.20: The automatic flow de-registration process monitors the network for transmission activity and removes the resource reservations when a flow is idle for more than a threshold duration [341].

protocols and the frequency of synchronization should be based on the node characteristics.

2.3.2 Flow Management

Resource Reservation

A resource reservation process is typically applied across the network elements so as to ensure that there are sufficient resources for processing TSN flow frames with priority. The TSN IEEE 802.1Qat protocol defines the resource reservation mechanism in TSN networks, see Section 2.2.3. Park et al. [341] revealed that the TSN

IEEE 802.1Qat standard lacks effective procedures for terminating reserved resources. The existing standardized resource release mechanism involves signaling among TSN nodes to establish a distributed management process, such that the connection reservations are torn-down and the resources released when the TSN flow is no longer needed. Similarly, when there is a renewed need for the TSN flow, the connection with its resource reservation is re-initiated based on the flow's traffic requirements. For networks with a few nodes and short end-to-end delays, the management process has relatively low signaling complexity and does not significantly impact the TSN flows. However, Park et al. [341] found that the numbers of nodes that are typical for in-vehicle networks result in a pronounced increase of the overall control message exchanges for the tear-down and re-initiation of connections.

Therefore, Park et al. [341] have proposed an automatic de-registration to tear down reservations. All participating nodes run the algorithm to de-register the reserved resources in a synchronized manner across the entire network based on the network wide synchronization capability in TSN networks. Figure 2.20 presents the flow chart of the automatic de-registration process: A timer is initialized to track the idle times for a specific TSN flow. Once the timer meets a predefined threshold, the resource reservations of the flow are automatically torn-down by all the participating nodes. The de-registration process is simultaneously performed throughout the network based on the synchronized timers. The downside of such an automatic de-registration process is the overhead for the re-activation process of the resource reservation for TSN flows which were deactivated due to a short period of inactivity. Thus, for highly bursty traffic, the automatic de-registration process may negatively impact the overall network performance since the idle times between traffic bursts may trigger the automatic de-registration.

Raagaard et al. [361] have examined GCL reconfiguration in the context of CNC and CUC (see Section 2.2.3). The actual underlying scheduling mechanism is an elementary greedy earliest deadline first heuristic. That is, flows with earlier deadlines are scheduled first. A weakness of the approach appears to be the long reconfiguration time. Despite the algorithmic simplicity, reconfigurations take between several seconds to up to a minute. Dynamic runtime management and reconfiguration of the IEEE 802.1Qbv GCL schedules thus continue to pose a significant challenge and are an important topic for future research [79, 123, 181, 193, 200, 333, 355, 376].

Bandwidth Allocation

Bandwidth allocation reserves the physical transmission resources required to meet the delay requirements of an end-to-end flow. A specific bandwidth allocation challenge in TSN arises from the multiple traffic classes, such as the different priority levels for scheduled traffic and best-effort non-scheduled traffic.

Ko et al. [236] have developed a simulation model to study the impact of the Maximum Transmission Unit (MTU) size of TSN traffic packets on the performance for scheduled traffic within a specific bandwidth allocation framework. Specifically, Ko et al. have examined bandwidth allocations for the scheduled traffic based on TSN definitions. Ko et al. assume that 75% of the bandwidth is allocated to the different QoS traffic classes, while the remaining 25% of the bandwidth is allocated to best-effort traffic. In particular, two classes of QoS traffic were considered, namely scheduled traffic and audio/video traffic. Bandwidth is allocated such that the total bandwidth allocated to scheduled and audio/video is always 75%, i.e., the allocation ratio between QoS traffic and best-effort traffic is maintained constant (75% to 25%). The study varies the bandwidth ratio between the scheduled traffic and the audio/video traffic. The bandwidth allocation for the scheduled traffic was varied by varying its MTU size. The simulations for a specific in-vehicle network scenario found that an MTU size of 109 bytes (corresponding to a bandwidth allocation of 7% to scheduled traffic), optimally allocated bandwidth to the scheduled traffic, which achieved an average end-to-end latency of 97.6 μs.

Routing

In contrast to routing mechanisms in conventional networks, Arif et al. [60] have proposed a computationally efficient optimization method to evaluate the routing paths for a TSN end-to-end connection. The proposed solution considers an optimality criterion that minimizes the routing path delays which effectively reduces the end-the-end latency of the TSN flows across the network. The proposed approach also considers multipath jitter, as well as the probability of loop occurrence while evaluating the end-to-end routing path of the TSN flow. The main purpose of the routing is to load balance the TSN flows in the network nodes and thus to reduce the routing path delays.

Software Defined Networking for TSN

The centralized computation and management of routing of an end-to-end TSN flow follows similar principles as the central control in the SDN paradigm. A formal adoption of the SDN paradigm in TSN networks has been presented by Nayak et al. [318]. Nayak et al. employed SDN principles to evaluate the routing of TSN flows and to apply the evaluated routes to the network nodes. As shown in Fig. 2.21, the proposed SDN controller implements four main management functions, namely monitor, analyze, plan, and execute to establish and control the TSN flows. Nayak et al. have conducted delay and flexibility simulation evaluations of several routing mechanisms with the SDN approach and without the SDN approach to quantify the benefits offered by SDN. Based on simulations, Nayak et al. have proposed the

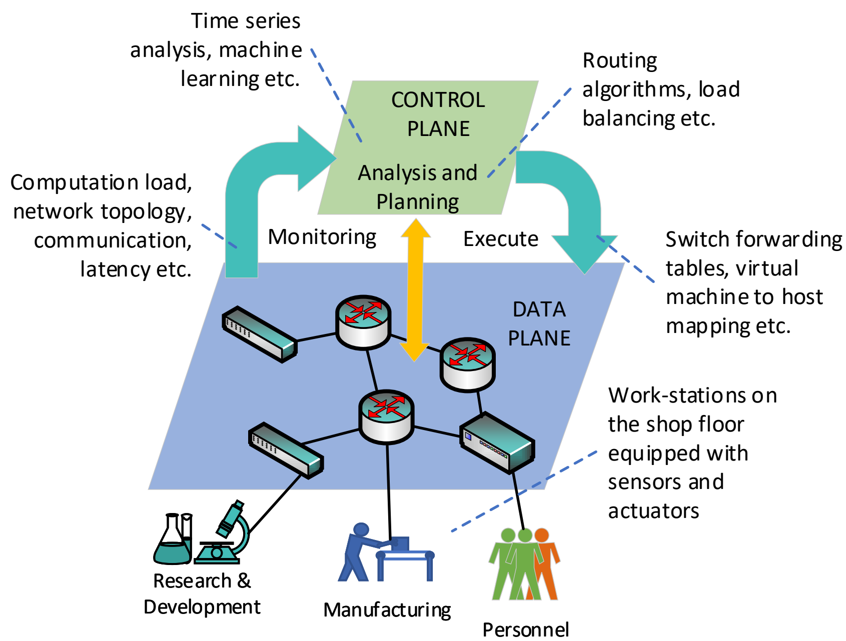


Figure 2.21: Software Defined Networking (SDN) based Time Sensitive Networking (TSN) in industrial network setting: Monitoring sensors from various factory locations deliver information to the centralized controller. The centralized controller applies the time sensitive networking rules across the industrial networks to support critical connectivity paths [318].

adoption of SDN to existing processes for the network management of time-sensitive applications.

While SDN inherently provides management flexibility [54, 211, 416], the actual deployment characteristics of SDN for TSN still need to be carefully characterized. Towards this goal, Thiele et al. [409] have presented the challenges in adapting SDN for TSN networks. Specifically, Thiele et al. have performed a timing analysis of an end-to-end TSN flow in the SDN framework to verify the limitations of SDN, such as overhead, scalability, and control plane delay in meeting the TSN requirements for in-vehicle networks. Thiele et al. used a compositional performance analysis framework to model the SDN network performance for TSN. The SDN deployment requires a centralized controller for the global management of the TSN network from flow establishment to tear-down. The placement of the controller among the TSN nodes can be challenging since the control signalling communication between nodes and controller can span across the entire network. Each TSN flow establishment process requires the exchange of control messages between a TSN node and the controller. As the numbers of TSN nodes and flows increase, the overhead due to control message exchanges could increase, affecting the overall TSN performance. Moreover, the flow setup process requires the TSN node to request the flow rules from the SDN controller which can increase the flow setup time as compared to a static non-SDN scenario. Therefore, to determine the feasibility of SDN for in-vehicle TSN networks an analytical formulation was verified through simulations. The simulation results demonstrate that the worst-case SDN network configuration delay is 50 ms, which is typically tolerable for admission control and fault recovery in conventional Ethernet networks. A related SDN based control plane architecture has recently been proposed in [378].

Summary and Lessons Learned

In addition to dynamic flow establishment based on current network characteristics, flow management ensures that TSN networks preserve the time-sensitive characteristics, such as low end-to-end delay, when the network characteristics, such as topology and number of nodes, change. The adaptability of the network to changes in network characteristics is an important network design aspect that needs to be examined in detail in future research. This future research needs to address the control plane as well as the data plane.

Currently, IEEE 802.1Qcc has centralized management, but does not preclude distributed management. The TSN TG has started the process of chartering a project to standardize RAP, see Section 2.2.3, which uses distributed management. Generally, centralized management can reduce the traffic overhead and reduce the management complexity. The detailed investigation of the tradeoffs between centralized and distributed management is an important direction for future research.

The static allocation of link resources to a TSN flow can result in low network efficiency. Dynamic link resource allocation provides more efficiency and flexibility. More specifically, a flow management technique can be implemented to statistically multiplex several flows sharing common network resources, while the worst-case flow performance is still bounded by a maximum prescribed value. A pitfall that needs to be carefully addressed is the network complexity in developing and deploying flow management techniques in actual networks. SDN may be a promising technology for the management of dynamic resource allocation in TSN networks. SDN also provides an inherent platform to design advanced TSN flows management mechanisms, such as admission control and security mechanisms.

2.3.3 Flow Control

The overall temporal characteristics of a TSN flow are dictated by the flow control mechanisms that are applied in the intermediate nodes. The flow control mechanisms implemented at each TSN node directly impact the process of frame traversal through each node that a particular flow is defined to pass through. A variety of flow control mechanisms are employed in the intermediate nodes before an enqueued frame is scheduled for transmission over the physical link. The most critical flow control mechanisms in TSN nodes are traffic shaping as well as scheduling and preemption.

Traffic shaping limits the traffic rate to a maximum allowed rate, whereby all traffic exceeding the maximum allowed rate is buffered and scheduled for transmission at an available opportunity. (In contrast, traffic policing simply drops the exceeding traffic.) The downside of traffic shaping is queuing delay, while the downside of policing is that excess frame dropping can affect the TCP transmission windows at the sender, reducing the overall network throughput.

Traffic Shaping

Control-Data Traffic (CDT) is the TSN traffic class for transmissions of control traffic with the shortest possible delay. In addition to the CDT class, TSN distinguishes traffic class A and class B. Collectively, these traffic classes are shaped by the traffic shapers in the TSN nodes to meet the delay requirements. The traffic shapers ensure that *i*) the CDT is allocated resources with strict priority, *ii*) the TSN traffic is isolated from the regular traffic, and *iii*) the wait times for enqueued frames are bounded. Towards these goals, various traffic shaping methods have been standardized, see Section 2.2.4, in order to satisfy the requirements of the flows based on their traffic classes.

Shaping Analysis: Thangamuthu et al. [407] have conducted a detailed comparison of the standard TSN traffic shaping methods. In particular, Thangamuthu et al. have compared the burst limiting shaper (BLS, a variation of CBS, which was considered in research but not incorporated into the TSN standard), the time aware shaper (TAS), and the peristaltic shaper (PS), see Section 2.2.4. The simulations show that for typical 100 Mbps Ethernet network deployments the in-vehicle delay requirements are met for most applications, except for applications with strict delay requirements. Therefore, additional ULL mechanisms are recommended, in addition to the traffic shaping, to satisfy strict application requirements. Thiele et al. [299, 411, 412] have conducted a complementary formal timing analysis and worst-case latency analysis of the different shapers for an automotive Ethernet topology, while an avionics context has been considered in [197]. Moreover, general latency and backlog bounds have recently been derived in [90, 173, 174, 216, 304, 458, 459]. As alternatives to CBS and TAS shaping, a pre-shaping approach at the senders has been explored in [315]. A complementary analysis of the ATS shaper has bee conducted in [461]. Pre-shaping has been found to be effective for a low number of hops. However, the pre-shaping effectiveness decreases with increasing hop count. Also, pre-shaping does not protect the shaped traffic flows from other unshaped or misbehaving flows in the network. The wireless fronthaul context, see Section 2.4.1, has been considered in [426].

Traffic Shaping Overhead: Traffic shaping, in particular the TAS can significantly impact the configuration overhead throughout the network, especially for temporary (short lived) TSN flows, such as those that originate from plug-and-play devices. The transmission schedule for TAS gate control must be evaluated and maintained at each traversed TSN node corresponding to each temporary flow. The schedule information at each node is generated and managed as a network configuration. These network configurations must be applied across the network to establish an end-to-end TSN flow. The temporary TSN flows resulting from plug-and-play connections can create a deluge of management traffic overhead.

To address this overhead issue, Farzaneh et al. [161] have presented an ontology based automatic configuration mechanism. Application management service and TSN management service entities coordinate the connection establishment and teardown procedures, managing the control plane actions for the TSN network. A TSN knowledge database is implemented to track and manage new, existing, and previous connections. For each connection, QoS requirements, assignments, and source details, such as port and devices are identified and analyzed to build an ontology of TSN flows corresponding to an application and device. Thus, whenever the plug-and-play event for a specific device occurs in the network, the TSN configurations are automatically retrieved and applied, lowering the overhead compared to the conventional connection management scheme. Although the automatic configuration mechanism is similar to the principles of SDN, Farzaneh et al. have discussed the process based automatic configuration mechanism independently of SDN. Nevertheless, the ontology based automatic configuration mechanism can be easily adapted to SDN by implementing the proposed application management service and TSN management functions as an SDN application.

Scheduling

TTEthernet vs. TSN: Craciunas et al. [120] have presented an overview of scheduling mechanisms for Time-Triggered Ethernet (TTEthernet) [119, 239, 398] and TSN. In the TTEthernet switch, the incoming frames for an outgoing egress port are temporarily stored in a buffer, and wait for the scheduler to assign a transmission-slot based on the precomputed schedule. In contrast, the incoming frames in TSN are directly inserted into priority queues, and these priority queues are served based on prescribed schedules. The fundamental difference between TTEthernet and TSN is the scheduling procedure, whereby the TTEthernet buffer is served based on global static scheduling information, i.e., a *tt-network-schedule* assigned to meet the end-to-end delay requirements. In contrast, TSN employs a dynamic schedule local to each node for control frame transmissions from priority queues. TSN switches may be synchronized to network timing and can preempt an ongoing lower priority transmission, which is not possible in a TTEthernet switch. Thus, the deployment of TSN switches as opposed to TTEthernet switches can improve support for delay critical applications. However, the implementation cost and complexity (due to synchronization) of TSN is typically higher than for TTEthernet.

Control Traffic Scheduling: Bello et al. [74] have presented an overview of TSN standards and examined the scheduling of control traffic flows in intra-vehicular Ethernet networks. More specifically, Bello et al. focused on the IEEE 802.1Qbv standard for scheduled traffic. Bello et al. have implemented the scheduled traffic mechanism for automotive connectivity applications by utilizing the time-sensitive properties of TSN. In particular, flow prioritization has been used to prioritize the control traffic flows over regular data flows. The traffic flows are separated into multiple priority queues and scheduling procedures are applied across the queues. Bello et al. [74] developed a simulation model for an automotive network to study the behaviors of TSN supported network modules. The evaluation in simulation indicated significant latency reductions by up to 50% for the control traffic flows, i.e., the scheduled traffic flows, compared to non-scheduled traffic. A limitation of the Bello et al. [74] study is that it considered only the automotive network domain and did not consider the wider applicability and potential of TSN.

Optimization Based Scheduling: An important shortcoming of the IEEE 802.1Qbv standard, which defines the transmission of scheduled traffic in TSN, is that there are no specific definitions of algorithms to determine the transmission schedule of frames on a link. In addition, the IEEE 802.1Qbv standard enforces a time spacing, i.e., guard bands, between the scheduled traffic types. The guard bands isolate scheduled traffic belonging to a specific class from other traffic classes, including the best-effort traffic class. A critical pitfall in the IEEE 802.1Qbv standard is that as the number of traffic classes increases, there can potentially be a large number of guard band occurrences during the traffic transmissions over the link. Traffic schedules with frequent guard bands waste bandwidth and can contribute to latency increases. Hence, an important direction for future work is to develop traffic transmission schedules with reduced numbers of guard band occurrences in order to prevent wasted bandwidth and to keep latencies low.

Dürr et al. [146] have modeled TSN scheduling as a no-wait job-shop scheduling problem [444]. Dürr et al. then have adapted the Tabu search algorithm [72, 186, 277] to efficiently compute optimal TSN transmission schedules while reducing the occurrences of guard bands. The evaluation in simulations indicate that the proposed algorithm can compute the near-optimal schedules for more than 1500 flows on contemporary computing systems while reducing the guard band occurrences by 24% and reducing the overall end-to-end latency for TSN flows. With the minimal duration of guard bands, see Section 2.3.3, the receivers have to be actively synchronized for the correct reception of TSN frames. The existence of guard bands in the traffic flows provides an inherent secondary synchronization for the receivers. However, it should be noted that the implementation of such optimization algorithms can increase the network node complexity as well as protocol operations, increasing the overall operational cost of the device. These scheduling principles have been further developed in [321] towards the incremental addition of new flows.

Craciunas et al. [121] have examined the scheduling of real-time traffic, whereby the transmission schedules are computed through optimization methods. The constraints for the optimization problem formulation are based on the generalized TSN network configuration in terms of the characteristics of the Ethernet frames, physical links, frame transmissions, end-to-end requirements, and flow isolation. While considering a comprehensive set of parameters, the optimization problem is modeled to compute transmission schedules in an online fashion (i.e., frame arrival event driven) to achieve low latency and bounded jitter. While a complex optimization problem can provide a near optimal solution, it is also important to consider the required computation times. Addressing the complexity aspect, Craciunas et al. have proposed several extensions to the optimization process and outlined the implications for the computation time. Craciunas et al. [121] have conducted an evaluation in simulation for various network loads and configurations. The simulation results indicate that an optimization process can be scalable while achieving the desired level of scheduling benefits, i.e., bounded latency and jitter for an end-to-end connection carrying real-time traffic. Craciunas et al. have further developed this optimal scheduling problem in [122, 399] A related scheduling approach based on a graphical model has recently been examined by Farzaneh et al. [163], while a recent study by Kentis et al. [227] has examined the impact of port congestion on the scheduling.

Joint Routing and Scheduling: TSN frame transmissions out of the queues can be controlled through gating (see Section 2.2.4), whereby a predefined event triggers the gate to transmit a frame from a queue according to a prescribed scheduling policy. With event triggering, the frame transmissions follow the predefined time triggered pattern, resulting in so-called time triggered traffic [148, 239, 297, 369]. Pop et al. [354] have designed a joint routing and scheduling optimization that evaluates the time trigger events to minimize the worst-case end-to-end frame delay. The time trigger schedule is based on an optimization problem formulated with integer linear programming. The proposed optimization problem comprehensively considers the network topology as well as time trigger flows and AVB flows. The time trigger flows follow the shortest route, while AVB flows follow a greedy randomized adaptive search approach. Simulation evaluations indicate that the compute time to evaluate the time triggered scheduling and AVB routing optimization is acceptable as compared to the timing of the frame flows. A limitation of the approach by Pop et al. is that the optimizations are not scalable and flexible when there are changes in the properties of the network infrastructure, e.g., topology changes. When there are such network infrastructure changes, then the entire optimization process must be reconfigured. The recent related study by Smirnov et al. [392] has focused on mixed criticality levels while the study by Mahfouzi et al. [280] has focused on the stability aspects of joint routing and scheduling.

Impact of Traffic Scheduling: Although TSN networks provide a pathway to achieve ULL through enhancements to the existing Ethernet standards, the benefits are limited to TSN flows as opposed to best-effort traffic. That is, in case of mixed transmissions, where the TSN defined transmissions are multiplexed with non-scheduled best effort traffic transmissions, there are no guarantees for the effective behavior of the non-scheduled best-effort traffic. If there are requirements for the non-scheduled traffic, such as a hard deadline for frame delivery in an end-to-end connection, the application can be severely affected due to the interference from the scheduled TSN traffic. The behavior characterization of non-scheduled traffic can be challenging and unpredictable due to the interference from scheduled TSN traffic. Therefore, Smirnov et al. [391] have provided a timing analysis to study the uncertainty of critical non-scheduled traffic in presence of scheduled TSN traffic interference. The challenge in the characterization of scheduled interference is to consider all possible traffic scenarios, such as all possible scheduling types, resulting in long computation times. Smirnov et al. propose an approach to integrate the analysis of worst-case scheduled interference with traditional end-to-end timing analysis approaches to reduce the computation times. Such an integrated approach can estimate an upper bound on the scheduled interference for various scheduling types, and the evaluations show significant computation time reductions.

A complementary study by Park et al. [340] has investigated the performance of scheduled traffic as opposed to the non-scheduled traffic. Park et al. performed extensive simulations focusing on TSN to verify whether the end-to-end flow requirements are impacted by increasing numbers of TSN nodes in the presence of non-scheduled traffic. The simulations employed the general network wide synchronous event-triggered method for frame transmissions in TSN networks. The simulations for an in-vehicle network based on the event triggered scheduling for various traffic types show that the delay requirements of control traffic can be successfully met for up to three hops. However, the scheduled traffic needs to be transferred within at most five hops to meet the typical 100 μs delay requirement for critical control data in in-vehicle networks.

At a given TSN node, the events to trigger an action that is then utilized for traffic scheduling can either be generated by a processing unit within the TSN node or by an external control entity. With the development and proliferation of SDN, future research can develop various event generation techniques based on the centralized SDN control and management. The generated events can trigger various TSN specified actions, such as frame transmissions, frame dropping, or frame preemption, enabling new applications for SDN control and management. To the best of our knowledge, event triggering methods based on SDN have not yet been investigated in detail, presenting an interesting direction for future research. However, SDN based management of TSN has already proposed and we discuss the applicability of SDN for managing TSN flows in Sec. 2.3.2.

While scheduled TSN transmissions provide low latency for prioritized traffic, lower-priority traffic which is also TSN scheduled can be significantly affected by higher priority traffic. In order to advance the understanding of the behaviors of traffic shapers on low-priority TSN traffic, Maixum et al. [288] have analyzed the delay of Ethernet frames that are scheduled according to a hierarchical CBS or TAS in TSN switches. The evaluations by Maixum et al. indicate that the traffic scheduling for higher priority TSN flows can potentially result in traffic burstiness for lower priority TSN flows, increasing the overall delay for the lower priority traffic. This is because, long bursts of higher priority traffic starve the scheduling opportunities for lower priority frames, leading to the accumulation of low priority traffic. In addition to the static scheduling order, Maixum et al. have also studied the effects of changing the scheduling orders in terms of end-to-end delay for both higher and lower priority levels. The formal worst-case delay analysis and simulation results indicate that low priority traffic is severely affected by the scheduled higher priority traffic. Simulations of an automotive use-case indicate a worst-case delay for the prioritized traffic of 261 μs, while the worst-case delay for low priority traffic is 358 μs.

Preemption

Preemption Mechanism: Lee et al. [251] have examined the preemption mechanism (see Section 2.2.4) in conjunction with the TSN timing and synchronization characteristics to estimate the transmission properties of CDT and non-CDT frames. In particular, Lee et al. have proposed to insert a special preemption buffer into the transmission selection module that operates across all the different queues at the bottom in Fig. 2.10 to aid with the preemption mechanism. Lee et al. have then analyzed the timing dynamics of the preemption. They note that in actual deployments there are likely timing synchronization errors which impact the frame boundary calculations. Therefore, a minimum safety margin that avoids collisions should be maintained while implementing the preemption mechanism. Lee et al. [251] advocate for a safety margin size of 20 bytes, accounting 5 bytes for an error margin and 15 bytes for synchronization errors. The simulation evaluations justify the impact of the synchronization errors on the safety margin duration and end-to-end delay. Related preliminary preemption analyses have been conducted in [214].

Preemption Effect on Non-CDT: Preemption prioritizes CDT frame transmissions over the transmission of regular Ethernet frames. Thus, preemption of non-CDT traffic can negatively impact the end-to-end characteristics of non-CDT traffic. In addition, low priority CDT frames can be preempted by high priority CDT frames. Hence, the preemption process can impact the end-to-end delay differently for the different priority levels even within the CDT traffic. Thiele et al. [410] have formulated an analytical model to investigate the implications of preemption on the end-to-end delay characteristics of CDT and non-CDT traffic. Thiele et al. have compared standard Ethernet with preemption (IEEE 802.1Q + IEEE 802.3br) and TSN Ethernet with time triggered scheduling and preemption (IEEE 802.1Qbv + IEEE 802.3br) with the baseline of standard Ethernet (IEEE 802.1Q) without preemption. The worst-case end-to-end latency of CDT with preemption was on average 60% lower than for 802.1Q without preemption. Due to the CDT prioritization, the worst-case latency of non-CDT traffic increased up to 6% as compared to the baseline (802.1Q) due to the overhead resulting from the preemption process. Hence, the impact of preemption of non-CDT traffic is relatively minor as compared to the performance improvements for CDT traffic. Additionally, the latency performance of standard Ethernet with preemption is comparable to that of Ethernet TSN with preemption. Therefore, Thiele et al. [410] suggest that standard Ethernet with preemption could be an alternative to TSN for CDT traffic. Standard Ethernet would be much easier to deploy and manage than TSN, as TSN requires the design and maintenance of the IEEE 802.Qbv gate scheduling processes along with time synchronization across the network.

Preemption Analysis and Hardware Implementation: Zhou et al. [462] conducted a performance analysis of frame transmission preemption. In particular, Zhou et al. adapted a standard M/G/1 queueing model to estimate the long run average delay of preemptable and non-preemptable frame traffic and evaluated the frame traffic through simulations. The numerical results from the adapted M/G/1 queueing model and the simulations indicate that preemption is very effective in reducing the frame delays for express non-preemptable traffic relative to preemptable traffic; the average frame delays of the express traffic are one to over three orders of magnitude shorter than for preemptable traffic. Zhou et al. have also provided the VHDL design layout of the transmit unit and receive unit for frame preemption for an FPGA based hardware implementation.

Summary and Lessons Learned

Flow control mechanisms ensure that intermediate nodes support the end-to-end behavior of a TSN flow. Traffic shaping controls the frame transmission over the egress port in a TSN switch. Each traffic shaper strives to transmit a frame from a priority queue within the shortest possible deadline while minimizing the impact on the transmissions from other queues. A finer resolution of priority levels, i.e., a higher number of priority levels provides increasingly fine control over frame transmissions from multiple queues. As a limiting scenario, an independent queue can be implemented for each individual flow in a TSN node. However, such fine-grained prioritization would require extensive computation and memory resources in each TSN node. To overcome this, virtual queues can be implemented by marking the frames in a single queue, eliminating the need for a number of queues equal to the number of TSN flows. Each marked frame can be scheduled based on the marking value. As low priority flows can potentially face long delays due to resource starvation from the scheduling of high priority flows, dynamic (i.e., changeable) priority values can be assigned to virtual queues. Dynamic priorities can prevent prolonged delays for flows that were initially assigned low priority. The priority levels can be dynamically changed based on the wait time or the total transit delay of a frame compared to a predefined threshold. Advanced dynamic priority techniques, such as priority inversion, could be implemented such that the worst-case delay of low priority traffic is kept within prescribed limits.

2.3.4 Flow Integrity

Fault Tolerance

The AVB task group was mainly introduced to add real-time capabilities to the best effort Ethernet service. Industrial control networks expect more reliable and stricter QoS services as compared to best effort Ethernet network service. Fault tolerance is a critical part of industrial networks. The general principle for enabling fault tolerance in a network is to introduce redundancy.

Following this general principle, TSN provides fault tolerance through redundancy mechanisms, such as frame replication and elimination as well as path control and reservations, see Section 2.2.5. Kehrer et al. [225] have conducted research on possible fault-tolerance techniques for TSN networks. The main challenges associated with fault tolerance mechanisms in TSN networks are the restoration processes for the end-to-end link failures while preserving the network topology, i.e., without causing any significant break in continuous network connectivity. To address this, Kehrer et al. have compared two approaches: *i*) decoupled stream reservation and redundancy [234], and *ii*) harmonized stream reservation and redundancy (which corresponds to IEEE 802.1CB).

In the decoupling approach, the stream reservation protocol registers and reserves the streams independently of the redundancy requirements. This decoupled approach allows for arbitrary redundancy protocols to be utilized. In contrast, the harmonized approach integrates establishment of the reservation and the redundancy requirements. More specifically, the IEEE 802.1Qca stream reservation protocol is coupled with the IEEE 802.1CB frame duplication.

The main pitfall to avoid is to understand the application requirements in terms of flexibility before choosing the redundancy approach. Specific industrial automation networks may have peculiar reliability requirements that may be more flexibly met with the decoupled approach. On the other hand, the decoupled approach has a higher protocol overhead and requires more network bandwidth due to the distributed and independent mechanisms along with the lack of coordination between stream reservation and redundancy, as opposed to the integrated approach. A related fault tolerance approach based on redundant packet transmissions has been examined in [50] while a mixing of temporal and spatial redundancy has been proposed in [48].

Summary and Lessons Learned

Failure recovery and fault tolerance are key aspects of reliable network design. However, to date there has been only scant research to address the critical challenges of resource reservation for fault tolerance while considering ULL requirements. Future research has to investigate the wide range of tradeoffs and optimizations that arise with reliability through frame replication. For instance, high priority flows could have reservations of dedicated resources, while low priority flows could share a common reserved resource. The dedicated resources would enable the instantaneous recovery of the high priority TSN flows; albeit, at the expense of a slight reduction of the overall network efficiency due to the redundancy. In the event of failure for a low priority traffic flow, the connection could be reestablished with a new flow path considering that the flows can tolerate delays on the order of the connection reestablishment time. Centralized SDN management can also provide the flexibility of dynamic path computation and resource reallocation in the event of failures. Therefore, the area of flow integrity requires immediate research attention to design and evaluate the performance of efficient recovery processes based on priority levels.

2.3.5 General TSN Research Studies

TSN is being widely adopted in critical small-scale closed automotive and industrial networks to establish reliable ULL end-to-end connections. However, a key TSN limitation is exactly this focus on closed networks, e.g., in-vehicle networks and small-scale robotic networks. The network applications running in robots and in in-vehicle networks often involve significant interactions with external non-TSN networks. Robotic and vehicular network applications require a tight integration with mobility handling procedures by the external network. If advanced network features, such as mobility, are not properly supported in the external network, then the TSN benefits are fundamentally limited to small-scale closed networks. Therefore, smooth interoperability between TSN and different external networks is essential for TSN operation in heterogeneous network scenarios. Ideally, the connectivity between TSN and non-TSN networks should be able to accommodate similar characteristics as TSN to ensure the overall end-to-end connection requirements in heterogeneous deployments.

V2X Communication

Juho et al. [252] have proposed iTSN, a new methodology for interconnecting multiple TSN networks for large-scale applications. The iTSN methodology utilizes wireless protocols, such as IEEE 802.11p, for the inter-networking between different TSN networks. In particular, the sharing of global timing and synchronization information across the interconnected network is important for establishing a common timing platform to support TSN characteristics in the external networks. The iTSN network uses the IEEE 802.11p WAVE short message protocol to share the timing information between different TSN networks. Critical rapid alert messages can be prioritized not only within a given TSN network, but also across multiple interconnecting networks. Thus, the iTSN methodology enables, for instance, vehicular networks to transmit safety critical messages to control nodes, e.g., Road Side Units (RSUs) [279], with delays on the order of microseconds in a heterogeneous deployment. Through the adoption of such reliable inter-connectivity techniques, the vehicle braking safety distance can be achieved in much shorter (microseconds) time spans than the currently feasible range of milliseconds. Overall, TSN and an interconnecting technique, such as iTSN, can create a communication platform for safe autonomous driving systems.

Network Modeling

Although TSN standards have received significant attention in networks for automotive driving, a major challenge in network deployment is managing the complexity. As automotive driving technology progresses, more requirements are imposed on the existing in-vehicle network infrastructure. As the number of sensors increase in an in-vehicle network, the increasing connectivities and bandwidth requirements of the sensors should be correspondingly accommodated in the network planning. However, the dynamic changes in the network requirements for an in-vehicle control system could require a more extensive network infrastructure, resulting in higher expenditures. Considering the complexities of automotive networks, Farzaneh et al. [164] have proposed a framework to analyze the impact of adding new sensors to an existing infrastructure that supports critical applications. In particular, the network configuration that fulfills all the requirements, including newly added sensors, must be dynamically evaluated and implemented. Towards this end, the Farzaneh et al. [164] framework involves a design and verification tool based on a Logic Programming (LP) method to support the reconfiguration and design verification processes for an in-vehicle TSN network. The proposed framework consists of comprehensive logical facts and rules from which a user can query the database with the requirements to obtain configurations that satisfy the requirements. A key characteristic of the proposed approach is that the network modeling process considers the most accurate logical facts and rules of the TSN applications and requirements to obtain an efficient configuration and verification process.

TSN Simulation Framework

Heise et al. [199] have presented the TSimNet simulation framework to facilitate the development and verification of TSN networks. TSimNet was primarily implemented to verify industrial use-cases in TSN networks. The simulation framework is based on OMNeT++, whereby the non time-based features, such as policy enforcement and preemption are implemented in a modular fashion to increase the flexibility of designing new network mechanisms suitable for industrial networks. For instance, the initial evaluation of the simulation framework for frame preemption mechanisms indicates that the end-to-end latency can be increased if the network is not configured in an optimized way for critical functions, such as scheduling and traffic shaping. Heise et al. have evaluated the computational cost of the TSimNet framework for various network function simulations, such as policing, recovery, and preemption in terms of CPU and memory requirements. The simulation framework also features Application Programming Interfaces (APIs) for TSN mechanisms that do not require time synchronization, such as stream forwarding, per-stream filtering, as well as frame replication and recovery. APIs can be invoked by the simulation framework through a profile notification. The basic framework modules also include the TSimNet Switch Model, which can identify streams based on MAC, VLAN, and/or IP addresses, while the TSimNet Host Model implements complex functions, such as ingress and egress policy, as well as traffic shaping. Related simulation evaluations with OMNeT++ have been reported in [328], while a TSN simulation model based the OPNET simulation framework has been presented in [338].

Hardware and Software Design

Hardware and software component designs to support TSN functions, such as scheduling, preemption, and time-triggered event generation in TSN nodes require significant engineering and development efforts. Hardware implementations are highly efficient in terms of computational resource utilization and execution latency but result in rigid architectures that are difficult to adapt to new application requirements. On the other hand, software implementations can flexibly adapt to new application requirements, but can overload CPUs due to the softwarization of network functions, such as time-triggered scheduling and hardware virtualization.

Gross et al. [189] have presented a TSN node architecture design where the time-sensitive and computationally intensive network functions are implemented in dedicated hardware modules to reduce the CPU load. The proposed hardware/software co-design approach flexibly allocates network function to be executed completely in hardware, completely in software, or in both hardware and software based on the dynamic load. The flexible allocation is limited to network functions that independently scale with the timing requirements, such as the synchronization protocol. More specifically, Gross et al. have considered time-triggered transmissions, frame reception and timestamping, and clock synchronization. The hardware modules can produce the time-triggered events nearly jitter free, implement frame reception and time-stamping in real-time, and synchronize clocks with a high degree of precision. Thus, the hardware modules improve the overall TSN node performance compared to a software-only implementation. The performance evaluations from a prototype implementation based on a Virtex-6 FPGA showed a significant reduction in the CPU load compared to a software-only implementation. Additionally, the precision of the time-triggered event generation in the hardware implementation was improved by a factor of ten compared to software triggered events.

TSN Testbeds

Generally, testbeds are ideal platforms used for testing and proof of concept investigations to check the viability of novel solutions in the real-world (as opposed to software emulation and simulation tests). In TSN, several organization (with private backing) have attempted to build and deploy TSN testbeds. One of the most prevalent testbed for TSN is deployed by the Industrial Internet Consortium (IIC). It is uses the Open Platform Communication (OPC) Unified Architecture (UA) publish-subscribe model and the specification of the TSN standards to enable deterministic behavior for a converged Ethernet layer supporting both critical and best effort traffic. Another testbed that continues to address and test TSN specification is the Labs Network Industry (LNI) 4.0 TSN testbed, though information is not available for the testbed. While these testbeds provide a means to test and evaluate TSN standards and research proposals in real production systems, the cost involved is high and flexibility for open researcher access is difficult. Therefore, an emulation environment (e.g., OMNeT++) with several plug-ins (e.g., INET, SUMO/Veins, SimuLTE, etc.) provide an easier method of integrating TSN specification on a virtual TCP/IP stack and test/evaluate such models seamlessly.

Summary and Lessons Learned

The general aspects of TSN that determine the overall success of TSN designs and implementations are the inter-interoperability with heterogeneous network architectures, such as LANs, WANs, and core networks. Most of the research on TSN to date has focused on in-vehicle networks which are independent and isolated from external networks. Another limitation of the TSN research field is the lack of a simulation framework that encompasses large-scale heterogeneous network architectures. Valid use cases that include both localized and external network interactions, such as automotive driving, should be created and considered in benchmark evaluations. Currently, the general use-case in most TSN research studies is an in-vehicle network supporting on-board sensor connectivity and audio/video transmission for infotainment. Future custom TSN simulation frameworks should be based on networks that support next-generation applications with localized and external network interactions, such as automotive driving. Similarly, the SDN based TSN management could exploit hierarchical controller designs to extend the management from localized networks, such as in-vehicle networks, to external networks, such as vehicle-to-any (V2X) networks.

2.3.6 Discussion on TSN Research Studies

The TSN network infrastructure and protocols have to support bounded end-to-end delay and reliability, to support basic features related to critical applications of IoT, medical, automotive driving, and smart homes. TSN based solutions for addressing the requirements of these applications result in complex network infrastructures supporting various protocols. Hence, simplified TSN network management mechanisms are essential to reduce the complexity while achieving the critical needs of the

ULL applications.

The deterministic TSN network behavior has so far been generally applied to a closed network, i.e., a network spanning only the scope of a particular application, for instance, in-vehicle networks. However, the connectivity to external networks, such as cellular and WLAN networks, enhances the capabilities of TSN networks. For instance, in automotive driving, the application requirements can be controlled by weather data from the cloud or by sharing information with a neighboring TSN in-vehicle network. Therefore, reliable, secure, and low-latency communication between multiple TSN networks is essential to support a wide range of future applications. The lack of TSN standards for connecting and communicating with external TSN and non-TSN networks is impeding the research activities in inter-operating networks and needs to be urgently addressed. In summary, we identify the following main future design requirements for TSN research:

i) Support for a wide range of applications spanning from time-sensitive to delay tolerant applications with flow level scheduling capabilities.

ii) Connectivity between multiple closed TSN architectures.

iii) Flexible and dynamic priority allocations to ensure bounded end-to-end latency for lower priority traffic.

iv) Adoption of SDN for the centralized management of TSN functions with a global network perspective.

v) Efficient timing information sharing and accurate clock design through self-estimation and correction of local clock skewness.

vi) Computationally efficient hardware and software designs.

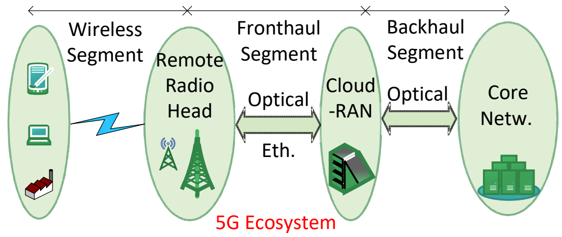


Figure 2.22: The main network segments that constitute the 5G ecosystem are the wireless segment, the fronthaul segment, as well as backhaul segment with corresponding and core network. In addition to various research efforts on the wireless segment, a variety of research efforts have been conducted on the fronthaul as well as the backhaul and corresponding core network. In this chapter we focus mainly on the ULL techniques in the fronthaul and backhaul network segments.

Generally, TDM can enforce a deterministic (100%) latency bound, but the TDM average delay is typically somewhat higher than the statistical multiplexing average delay (and TDM has low utilization for bursty traffic). With proper admission control, statistical multiplexing can provide statistical guarantees for latency bounds [108], e.g., the probability for exceeding the delay bound can be very low, e.g., less than 10^{-4} probability that the delay bound is violated. These rare occurrences of violating the delay bound "buy" usually much higher utilization (throughput) than TDM and lower average delay (for bursty data traffic) [235, 260, 365, 366, 457]. An interesting future research direction is to examine the tradeoffs between deterministic and probabilistic delay bound assurances in detail for ULL traffic served with TSN mechanisms.

2.4 5G Ultra-Low Latency (ULL)

5th Generation (5G) cellular technology is a paradigm shift in the network connectivity as 5G is expected to comprehensively overhaul the network infrastructure by establishing an end-to-end ultra-reliable and ultra-low latency connection [281, 390]. 5G is also expected to improve the network efficiency in terms of network utilization, control plane overhead, and energy savings.

As illustrated in Fig. 2.22, the overall 5G ecosystem can be classified in terms of wireless access, fronthaul, as well as backhaul segment with corresponding and core network. The wireless access is responsible for the wireless connectivity between the devices and the radio nodes. The fronthaul connects the radio nodes to the radio baseband processing units, while the backhaul connects the radio baseband processing units to core networks. The core network interconnects with the Internet at large, including data centers, to provide end-to-end services to devices. A large number of 5G research efforts have been conducted in the wireless access domain; additionally, many articles have presented overviews of the 5G advancements [34, 56, 78, 80, 144, 147, 209, 220, 274, 303, 348, 352, 387].

The recent survey on low latency characteristics in 5G by Parvez et al. [344] focuses on waveform designs, wireless protocol optimizations, microwave backhaul architectures, SDN architectures for backhaul and core networks, and content caching mechanism for 5G. To the best of our knowledge, there is no prior survey that comprehensively covers the ULL aspects across the 5G network segments from the fronthaul to the core networks focusing on the transport mechanisms of the user data and the control plane signalling. We fill this gap by providing a comprehensive survey of ULL techniques across the 5G wireless access, fronthaul, as well as backhaul and core networks in this section.

5G ULL mechanisms are motivated by applications that require ultra low end-to-end latency. As discussed by Lema et al. [254], the business use cases for low latency 5G networks include health-care and medical applications, driving and transport, entertainment, and industry automation. Remote health-care and medical interventions, including robotic tele-surgery, require reliable communication with ultra-low latency. Assisted and automatic driving require high data rates for sensor data processing as well as low latency to ensure quick responses to changing road conditions. Immersive and integrated media applications, such as Augmented Reality (AR) and

Table 2.1: Latency comparison at multiple components of network connectivity over 3G (High Speed Packet Access (HSPA)), 4G (LTE), 4.9G (pre 5G), and 5G [5].

Delay Comp. (ms)	**3G**	**4G**	**4.9G**	**5G**
DL Trans.	2	1	0.14	0.125
UL Trans.	2	1	0.14	0.125
Frame alig.	2	1	0.14	0.125
Scheduling	1.3	0–18	Pre-sch.	Pre-sch.
UE proc.	8	4	0.5	0.250
eNB proc.	3	2	0.5	0.250
Trans.+Core	2	1	0.1	0.1
Total Delay (ms)	20	10–28	1.5	1

Virtual Reality (VR) for gaming and entertainment require high data rates for video transmissions and extremely low latency to avoid jitter in the video and audio. With these demanding business needs and application requirements, 5G is expected to continuously evolve to support ultra and extremely-low latency end-to-end connectivity.

2.4.1 5G ULL Standardization

In this section, we identify the key components in 5G standards for supporting ULL mechanisms. Various standardization organizations contribute to the development of 5G standards, including the IEEE and IETF, as well as the Third Generation Partnership Project (3GPP), and the European Telecommunications Standards Institute (ETSI). We first discuss the standards related to the 5G fronthaul interface, and subsequently we present the 5G architecture components which include the backhaul. The fundamental latency limits of 5G standards are summarized in Table 2.1. The 4.9G corresponds to the optimization efforts for LTE towards 5G, where a drastic more than 10 fold reduction in the latency is achieved. The current standardization efforts have targeted the total delay for 5G to be 1 ms or lower.

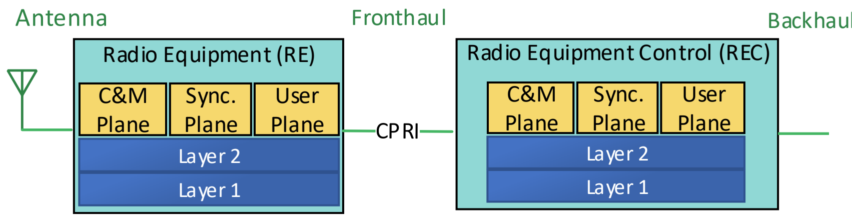


Figure 2.23: Common Public Radio Interface (CPRI) system overview [3]: The Radio Equipment Control (REC) connects to the Radio Equipment (RE) via the CPRI interface. The REC is part of the Base Band Unit (BBU) and the RE is part of the Remote Radio Head (RRH) in the Cloud-RAN architecture.

Common Public Radio Interface (CPRI and eCPRI)

CPRI: The Common Public Radio Interface (CPRI) [130] is a digital interface for transporting information between Radio Equipment (RE) and Radio Equipment Control (REC). The RE resides at the Remote Radio Head (RRH) and is responsible for the transmission of radio signals while the baseband signal processing is conducted at the BaseBand Unit (BBU) which implements the REC. In particular, CPRI provides the specifications for packing and transporting baseband time domain In-phase/Quadrature (I/Q) samples. Figure 2.23 illustrates the connectivity of BBU and REC with the RRH and RE using the CPRI. CPRI mandates the physical layer (L1) to be optical Ethernet transmissions over fiber, while the MAC layer can include control and management, synchronization, and user data. CPRI has been widely adapted for LTE and 4G deployments due to the protocol simplicity and readily available dark fiber owned by cellular operators [97].

5G is expected to support high bandwidth connections up to several Gbps, resulting in very high effective I/Q CPRI data rates. For instance, a massive MIMO connectivity with 64 antennas for both transmission and reception would require more than 100 Gbps [419]. Additionally, the CPRI Service Level Agreements (SLAs) require delays below 75 μs. Therefore, CPRI poses severe scalability issues as the required data rate increases drastically with the number of antennas for massive MIMO which

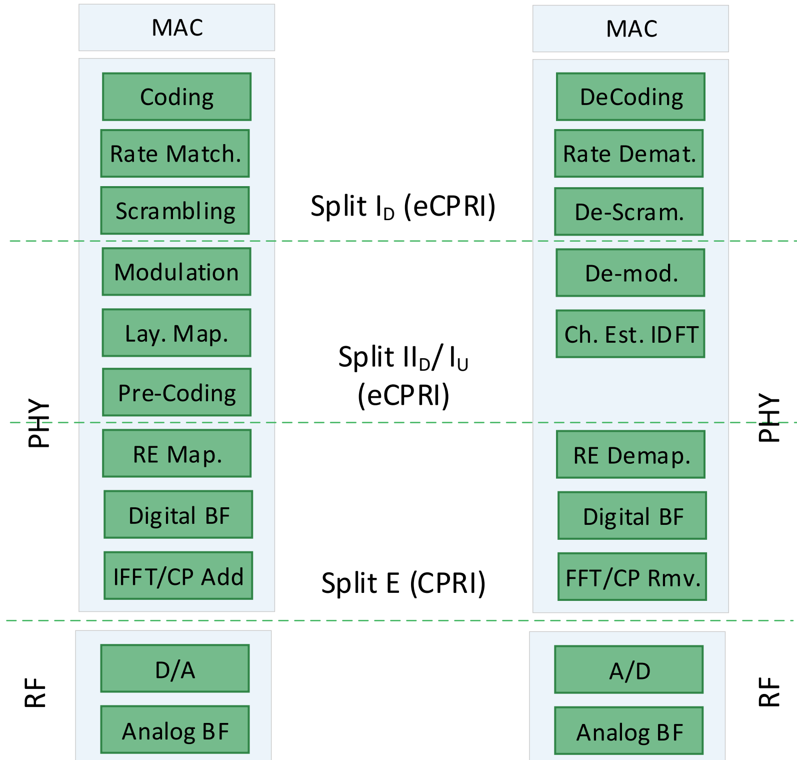


Figure 2.24: Split options defined by eCPRI the steps above the horizontal dashed line are processed at the BBU and the steps below the dashed line are processed at the RRH: Split E corresponds to the CPRI data, split I$_D$ corresponds to the eCPRI downlink data after scrambling, split II$_D$ corresponds to the eCPRI downlink data after pre-coding, and split I$_U$ corresponds to the eCPRI uplink data after RE-demap [130].

are widely considered for 5G networks [419]. Dense Wavelength Division Multiplexing (DWDM) and Optical Transport Networks (OTNs) can support the stringent CPRI SLA requirements. However, dense deployments of 5G radio nodes due to the short mmWave range require fiber connectivity to large numbers of radio nodes. Therefore, eCPRI, an enhanced version of CPRI, has been proposed to address the scalability issues of CPRI [306]. The 5G fronthaul enabled by eCPRI will not only reduce the required fronthaul bandwidths, but also relax latency requirements compared to CPRI.

Table 2.2: CPRI Split E as well as eCPRI splits I_D, II_D (downlink), and I_U (uplink) one-way packet delay and packet loss requirements [7].

CoS name	**Example use**	**One-way max. packet delay**	**One-way pkt. loss ratio**
High	User Plane	100 μs	10^{-7}
Medium	User Plane (slow), C&M Plane (fast)	1 ms	10^{-7}
Low	C&M Plane	100 ms	10^{-6}

eCPRI: eCPRI reduces the effective data rate carried over the L1 connection between RE and REC. eCPRI also removes the mandatory L1 requirements, thus allowing operators to implement low-cost Ethernet links. More specifically, the data rate reduction is achieved by various functional split options as shown in Fig. 2.24. The split option defines the allocation of the RF and PHY processing steps to the RRH and BBU. The steps above the split indicated by a horizontal dashed line in Fig. 2.24 are conducted at the BBU, while the steps below the split are conducted at the RRH. Accordingly the split option governs the type of signal (and its corresponding QoS requirements) that has to be transmitted over the fronthaul network. eCPRI primarily defines two split options in the downlink. The I_D split performs PHY layer bit scrambling at the BBU, while RF transmissions are modulated at the RRH. Similarly, the II_D split conducts pre-coding, Resource Element (RE) mapping, digital Bandpass Filter (BF), and IFFT/FFT and Cyclic Prefix (CP) at the BBU. In contrast to the downlink, eCPRI defines only one split option in the uplink I_U, whereby the PHY layer functions, from the channel estimation to the decoding, are conducted at the BBU, while RE demapping to RF transmissions are processed at the RRH.

In contrast to eCPRI, CPRI only carries the output from the IFFT/FFT and Cyclic Prefix (CP) at the BBU to the RF Digital to Analog (D/A) converter at the RRH. The delay requirements for the various Classes of Service (CoS) for the I_D and

II_D splits (eCPRI) and the E split (CPRI) are summarized in Table 2.2. The high CoS corresponding to split E (CPRI) requires the one way maximum packet delay to be on the order of 100 μs. The split E transports the I/Q data and in-band Control and Management (C&M) information. The medium CoS, which supports both the user and C&M plane data, requires 1 ms delay. The low CoS for the uplink eCPRI I_U split requires 100 ms delay.

The eCPRI services include:

i) User plane I/Q data transport between BBU and RRH, user plane control and management (C&M), and support services, such as remote reset.

ii) Time synchronization between BBU and RRH.

iii) Operation and management (OAM), including eCPRI connection setup, maintenance, and tear-down.

eCPRI supports various message formats to transport I/Q data according to the adopted split option. The protocol stack description of eCPRI services over IP and Ethernet is illustrated in Fig. 2.25. The eCPRI specific protocol layer transports the time domain I/Q data for split E, or frequency domain I/Q data for splits I_D and I_U. eCPRI messages are transmitted as UDP/IP packets whereby the eCPRI header and data constitute the UDP packet payload. The UDP packet headers contain both the source and destination IP addresses of the eCPRI nodes. Various message types control the overall operation of eCPRI over Ethernet links, including one-way delay measurement, remote reset, and event indication.

Unlike CPRI, which requires point-to-point and point-to-multipoint operation in a master-to-slave configuration, eCPRI is agnostic to the network topology which may encompass local area networks, as well as public routers and switches. The logical topologies that are possible with eCPRI include:

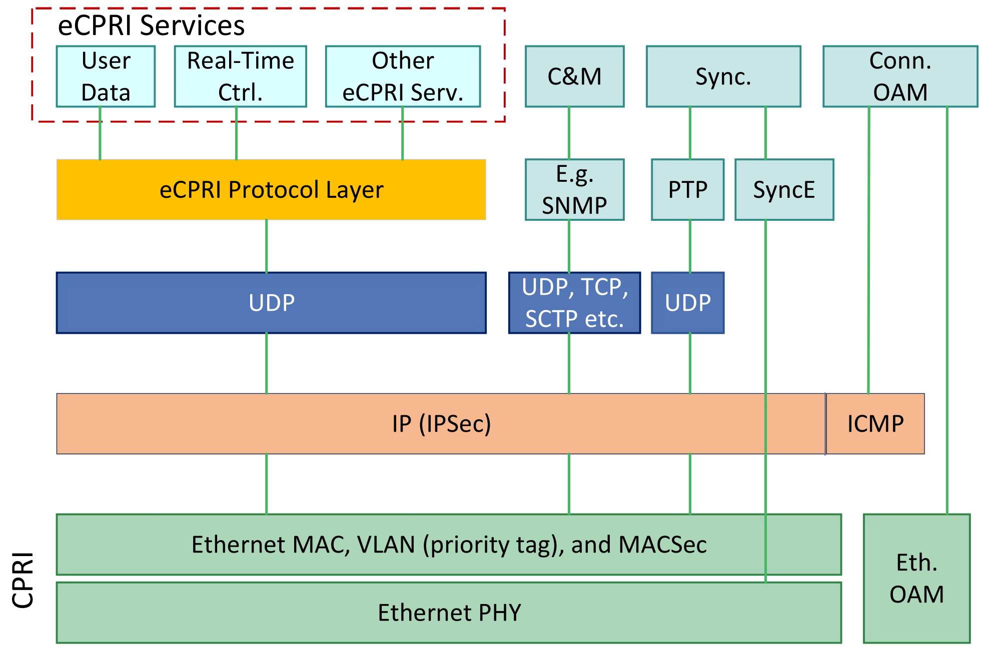


Figure 2.25: The eCPRI protocol stack consists of the eCPRI protocol layer, which transports the data from various split options over generic UDP and IP protocol layers. The lower layers, PHY and MAC, are equivalent to the CPRI protocol. The eCPRI services as well as the eCPRI control and management data along with synchronization are supported by the eCPRI protocol stack [7].

- Point-to-point, i.e., one BBU to one RRH which is similar to CPRI.
- Point-to-multi-point, i.e., one BBU to multiple RRHs (supported in CPRI as well).
- Multi-point-to-multi-point, i.e., multiple BBUs to multiple RRHs (mesh configuration), unique to eCPRI.

In a generalized Ethernet network carrying multiple traffic types (including best effort traffic), the user plane I/Q data and the real time O&M data require high priority transmissions. TSN mechanisms, see Sec. 2.2, can enable Ethernet networks to meet the eCPRI delay requirements. eCPRI management messages and user plane data can be regarded as Control Data Traffic (CDT) that is transmitted with high priority scheduling over the TSN network. Traffic requirements for user plane data vary for the different split options, which can be assigned different TSN priority levels. For instance, the C&M data is typically not as delay sensitive as user plane data; hence, a lower priority can assigned to C&M traffic. However for critical C&M data, such as remote reset while troubleshooting a Remote Equipment (RE) problem may require higher priority levels than user data. Therefore, two priority levels can be assigned to C&M traffic, i.e., a priority level higher than user data and another priority level lower than user data. These priority levels can be readily supported by TSN networks, which accommodate eight independent priority queues.

Summary and Lessons Learned: 5G technology supports diverse applications with a wide range of data rates and latency requirements, which directly translate to requirement for a flexible and scalable fronthaul. CPRI and eCPRI provide standardized protocols for inter-operating with existing cellular infrastructures. CPRI may not be suitable for supporting massive broadband services due to the very high required I/Q data rates. Also, the CPRI latency requirements need to be carefully considered and may require the judicious use of the scheduled traffic concept [426]. eCPRI overcomes the data rate issue through functional splits but increases the complexity of remote radio nodes. Another shortcoming of eCPRI is that the system considers asymmetrical OFDM in the downlink and uplink, i.e., single-carrier OFDM (SC-OFDM) in the uplink. Symmetrical OFDM systems are being investigated for increased spectral uplink efficiency [419]. However, there is no specific split defined for symmetrical OFDM systems in eCPRI. Remote spectrum analysis for troubleshooting RF issues is possible in CPRI; whereas, eCPRI does not provide such remote RF evaluation capabilities, although splits I_U and I_D allow for remote RF management. Hence, mechanisms for the transmission of sampled time domain I/Q samples from the RRH back to the BBU must be developed for advanced troubleshooting.

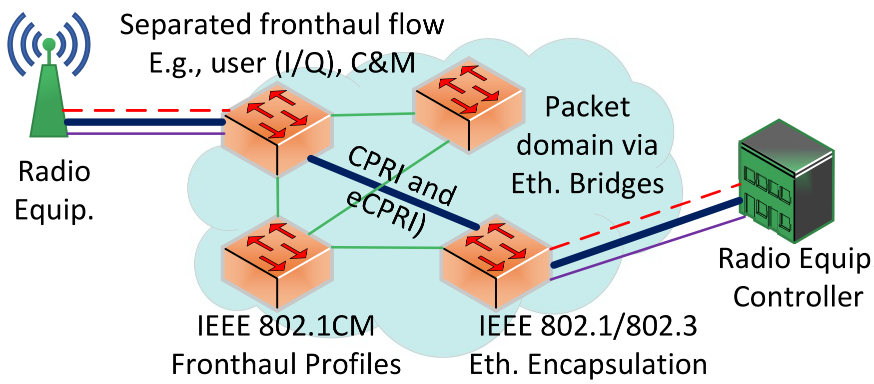


Figure 2.26: IEEE 802.1CM defines the support for Ethernet-based fronthaul in a bridged network. Flows are separated into different classes and a specific fronthaul profile is applied to each class to transport the flows over the Ethernet bridges based on the flow requirements [4].

IEEE 802.1CM: Time-Sensitive Networking for Fronthaul

The IEEE 802.1CM standard [4] is a CPRI-IEEE 802.1 collaboration to provide bridged Ethernet connectivity for fronthaul networks, as illustrated in Fig. 2.26. An 802.1CM bridge must support a data rate of 1 Gbps or higher on each port. The IEEE 802.1CM requirements are derived from CPRI and eCPRI so as to support various splits, such as splits at the FFT, demapping, and scrambling radio functions. IEEE 802.1CM defines mechanisms for end stations, bridges, and LANs to establish Ethernet networks that can support the time sensitive transmissions of fronthaul streams. In current cellular network deployments, the separation between RRH and BBU requires connectivity with stringent latency and capacity requirements. These fronthaul connection requirements could not be readily provided by today's bridged Ethernet networks.

IEEE 802.1CM provides specific mechanisms, such as scheduling, preemption and synchronization mechanisms, to satisfy the fronthaul requirements. With IEEE 802.1CM, mobile operators can utilize large segments of existing bridged networks to support 5G fronthaul networks, reducing capital expenditures. Moreover, centralized management mechanisms can be employed for automatic network reconfigurations, reducing the operational expenditures compared to manual network configuration. IEEE 802.1CM distinguishes Class 1 traffic for CPRI and Class 2 traffic for eCPRI. In terms of network synchronization, the IEEE 802.1 CM standard specifies two mechanisms: *i*) packet timing using protocols, such as the Precision Time Protocol (PTP) for point-to-point synchronization distribution from a remote common master, and *ii*) co-located common master for both BBU and RRH.

Latency Components of a Bridge: A bridge supporting fronthaul network functionalities needs to tightly control the latency and synchronize its functions. The latency for a single hop in a bridge network is the time duration from the arrival of the last bit of a given frame at a given bridge port to the arrival of the last bit of the same frame at a particular port at the next hop bridge. The main delay components are:

i) Store and forward delay t_{SF} due to all the elements responsible for the internal frame forwarding from ingress to egress port.

ii) Queueing (interference) delay t_{Queuing} due to ongoing transmissions of higher priority frames.

iii) Self queuing delay $t_{\text{Self\_Queuing}}$ due to frames of the same class that arrive across multiple ports and need to be sequentially queued.

iv) Periodic Constant Bit Rate (CBR) high priority data flow delay $t_{\text{MaxGoldFrameSize}+Pre+SFD+IPG}$. IQ data flows are referred to as gold flows in IEEE 802.1 CM. The CBR data delay $t_{\text{MaxGoldFrameSize+Pre+SFD+IPG}}$ of a gold frame corresponds to an IQ data frame with maximum frame size with Preamble (Pre), Start Frame Delimiter (SFD), and Inter Packet Gap (IPG).

The total worst-case self-queuing delay in a bridge can be evaluated based on the number N_p of ingress ports that can receive interfering gold frames which need to be transmitted over egress port p, and the total number of flows $F_{i,p}$ supported between ingress port i and egress p. Let $G_k^{i,p}$ denote the maximum number of frames belonging to a gold flow k traversing from ingress port i to egress port p that can be grouped into a single time window before the reception of frames at the ingress edge port of the bridge network. The resulting worst-case self-queuing delay at port j can be evaluated as

$$\begin{aligned} t^{j,p}_{\text{Self\_Queueing}} &= t_{\text{MaxGoldFrameSize+Pre+SFD+IPG}} \\ &\times \sum_{i=1, i\neq j}^{N_P} \sum_{k=1}^{F_{i,p}} G_k^{i,p}. \end{aligned} \tag{2.1}$$

Without preemption, the maximum queuing delay t_{Queuing} incurred by gold flows depends on the maximum size of the low priority frame along with preamble (Pre), Start Frame Delimiter (SFD), and the Inter Packet Gap (IPG), which results in $t_{\text{Queuing}} = t_{\text{MaxLowFrameSize+Pre+SFD+IPG}}$. However with preemption, a high priority frame is transmitted right after the transmission of the fragment of the preemptable frame, which includes the Cyclic Redundancy Check (CRC) and Inter Frame Gap (IFG). Therefore, the total worst-case delay $t_{\text{MaxBridge}}$ for gold flows in a bridge can be evaluated as

$$\begin{aligned} t_{\text{MaxBridge}} &= t_{\text{MaxGoldFrameSize+Pre+SFD+IPG}} \\ &+ t_{\text{SF}} + t_{\text{Queuing}} + t_{\text{SelfQueuing}}. \end{aligned} \tag{2.2}$$

Fronthaul Profiles: In general, the fronthaul flows in a bridged network are classified into High Priority Fronthaul (HPF), Medium Priority Fronthaul (MPF), and

Low Priority Fronthaul (LPF) flows. The HPF corresponds to class 1 I/Q data and class 2 user plane data with the requirement of 100 μs end-to-end one-way latency. Similarly, the MPF corresponds to the class 2 user plane (slow) data and class 2 C&M (fast) data. The LPF could include the C&M data of class 1 and 2 traffic. IEEE 802.1 CM defines two profiles, namely profiles A and B, to service different fronthaul technologies supporting both class 1 and 2. The MPF data is typically assigned a priority level immediately below the HPF; whereas, the LPF data is assigned a priority immediately below the MPF data. In contrast to the traffic classes which are designed based on the relative priorities, the profiles are designed based on the worst-case end-to-end delay within a given traffic class.

Profile A: The goal of profile A is to simplify the deployments and support only strict priority, focusing on the transport of I/Q user data as high priority traffic and C&M data with lower priority. The maximum fame size for all traffic is 2000 octets.

Profile B: Profile B adopts advanced TSN features, including frame preemption, as defined in IEEE 802.3br and 802.1Qbu, as well as strict priorities to carry I/Q user data as high priority traffic and C&M data as low priority preemptable traffic. The maximum frame size for user data is 2000 octets, while all other traffic can have variable maximum frame sizes.

Summary and Lessons Learned: IEEE 802.1CM primarily supports CPRI and eCPRI connectivity over bridged networks. IEEE 802.1CM enables cellular operators to use the existing Ethernet infrastructure reducing the capital and operational expenditures. However, the lack of support for generalized fronthaul networks limits the applicability of the IEEE 802.1CM standard to a wider set of 5G applications, such as crosshaul [131]. The relative performance of the low priority C&M traffic as compared to the high priority I/Q user data traffic (i.e., the ULL traffic) still needs to be thoroughly investigated to understand the behavior of traffic classes when operating at high load levels that approach the link capacities.

Although the delay and synchronization aspects have been specified in the standards, the security and reliability issues have not yet been considered in detail. Hence, security and reliability present a wide scope for future research and standards development. These security and reliability issues should be investigated by the fronthaul task force which is responsible for the IEEE 802.1 CM standards development.

We note that a cellular operator may choose to change priority levels as desired. A potential pitfall is that regular (non-fronthaul) traffic could be assigned higher priority than fronthaul user data or C&M traffic. Such a priority assignment would increase the self-queuing and queuing delays for the fronthaul traffic. Thus, the relative priority levels of the different traffic priority classes need to be carefully considered in the network resource allocation.

Next Generation Fronthaul Interface (NGFI)

Overview: Although the IEEE 802.1 CM, CPRI, and eCPRI fronthaul protocols provide implementation directions for fronthaul networks, the lack of fronthaul architectural standardizations has prompted the IEEE standards group to commission the IEEE 1914 Working Group (WG) [6] to define the standards for packet-based Fronthaul Transport Networks (FTN). In particular, the IEEE 1914 WG has defined two standards: *i*) IEEE P1914.1 focusing on architectural concepts related to both data and management fronthaul traffic in an Ethernet based mobile FTN networks, and *ii*) IEEE P1914.3 focusing on the encapsulation of I/Q data for Radio Over Ethernet (RoE). In comparison to IEEE 1914.3, the latency impact on the fronthaul deployment is mainly influenced by IEEE P1914.1. Hence, we primarily focus on architectural concepts, protocol operations, traffic management, and requirements as

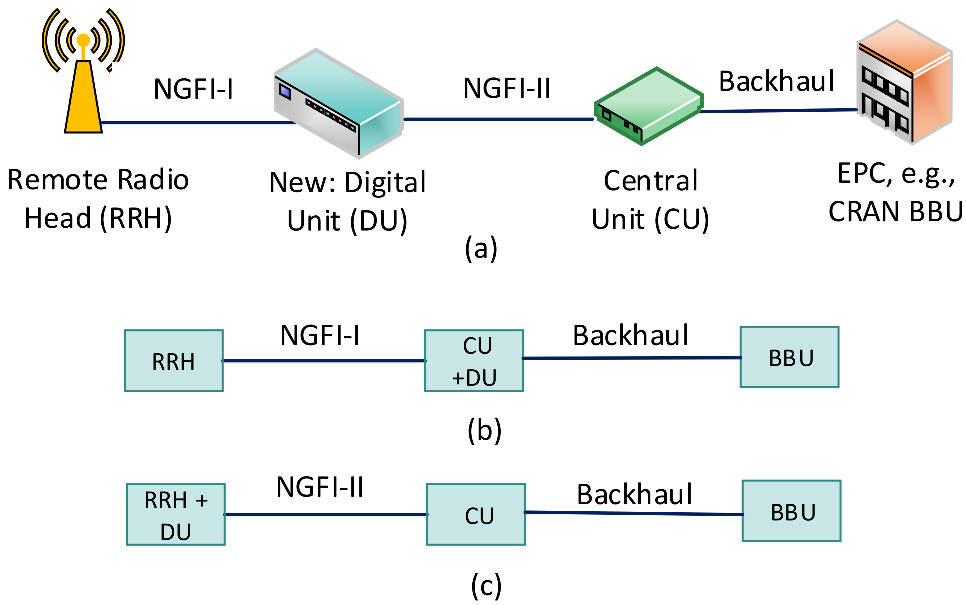


Figure 2.27: Illustration of two-level architecture options for next-generation fronthaul transport network: (a) RRH is connected via NGFI-I fronthaul interface to Digital Unit (DU) and DU is connected via NGFI-II interface to Central Unit, (b) RRH is connected via NGFI-I interface to integrated CU and DU, and (c) DU is integrated with RRH and connected via NGFI-II to CU [6].

well as the definitions for fronthaul links as defined by IEEE P1914.1. The goals of IEEE P1914.1 are to support 5G critical use cases, such as massive broadband services and to design a simplified fronthaul architecture that can utilize the existing standard Ethernet deployments of cellular operators. However, IEEE 1914.1 does not define the functional split aspects of the fronthaul, while aligning with 3GPP to support functional splits suitable for 5G.

Two-Level Fronthaul Architecture: IEEE P1914.1 defines a two-level fronthaul architecture that separates the traditional RRU to BBU connectivity in the CRAN architecture into two levels, namely levels I and II. Level I connects the RRH via a Next Generation Fronthaul Interface-I (NGFI-I) to a new network element, the so-called Digital Unit (DU). Level II connects the DU via an NGFI-II interface to the newly introduced Central Unit (CU), as shown in Fig. 2.27(a). Figs. 2.27(b) and (c) show different deployment options with integrated RRH and DU, and with integrated CU and BBU, respectively. The purpose of the two-level architecture is to distribute (split) the radio node (i.e., eNB/base station) protocol functions between CU and DU such that latencies are relaxed, giving more deployment flexibilities. In general, NGFI-I is targeted to interface with the lower layers of the function split which have stringent delay and data rate requirements. In contrast, NFGI-II is targeted to interface with the higher layers of the function split relative to NGFI-I, relaxing the requirements for the fronthaul link.

The NGFI is designed to mainly address:

i) Scalability: To enable C-RANs and Virtual-RANs that are functional split and traffic independent.

ii) Resource Utilization: To achieve statistical multiplexing by supporting variable MIMO and Coordinated Multipoint (CoMP) for 5G.

iii) Flexibility: To operate in a radio technology agnostic manner while supporting SDN controlled dynamic reconfigurations.

iv) Cost Effective: To utilize existing cellular network infrastructure.

Additionally, NGFI supports connectivity to Heterogeneous Networks (HetNets) by decoupling the transport requirements from the radio technologies. Thus, multiple traffic classes, as summarized in Table 2.3, can be transported by the NGFI network, mainly to support latencies according to the application demands. The C&M class supports low-latency control plane data for radio node signalling. Data plane latencies vary according to the different subclasses 0–4 to support multiple technologies and deployment versions with multiple split options. Subclass 0 requires the highest priority with 50 μs of maximum allowed latency, while subclass 4 has the lowest

Table 2.3: NGFI Transport Classes of Service; Low split, Med. split, and High split are relative to the positioning of the split in Fig. 2.24, whereby the low split is closer to the bottom of Fig. 2.24.

Class	**Sub Class**	**Max. Lat.**	**Pri.**	**App.**
C&M	Sync.	TBD	TBD	
	Low Lat. RAN ctrl. plane	100 μs	2	
Data Plane	Subclass 0	50 μs	0	ULL data
	Subclass 1	100 μs	1	Low split.
	Subclass 2	1 ms	2	Med. split
	Subclass 3	3 ms	3	High split
	Subclass 4	10 ms	4	Legacy Backhaul
Trans. Net. C&M	Trans. Net. ctrl. plane	1 ms	2	

priority and a 10 ms maximum delay bound. Subclass 4 can, for instance, be used for the legacy backhaul over the NGFI interface. The traffic of each subclass is independently transported between the end points without any mutual interference while achieving statical multiplexing gain among the subclasses.

Summary and Lessons Learned: The NGFI primarily addresses the scalability and cost issues with the current fronthaul solutions, such as CPRI. With NGFI, connections between DU and CU can be directly connected by an Ethernet link supporting IEEE P1914.1 specifications. The NFGI L2 subclass 0 transport service can readily accommodate the requirements of the existing CPRI deployments without any changes to the infrastructure deployments. Thus, NGFI is expected to play a significant role in the unification of heterogeneous radio technologies at the transport level and support converged fronthaul and backhaul networks for interconnected and coexisting 4G and 5G technologies. An important aspect to investigate in future research is the tradeoff between link utilization and multiplexing gain for the standard Ethernet networks while adopting these new fronthaul support architectures and protocols.

Backhaul Networks

Overview: The backhaul networks consisting of core network elements play a critical role in setting up the end-to-end flows. Core networks control the user data scheduling in both uplink and downlink. The control signalling of the radio technology, e.g., LTE, can contribute to flow latency when user devices transition among various states, e.g., idle to active (connected) and vice versa [106, 290, 417]. For scenarios with intermittent data activity, devices typically implement a state transitioning mechanism from active to idle to conserve computing and wireless resources. For instance, if the inter packet delay is more than 40 ms, the device can pro-actively change the radio control state to idle. Thus, within a single ULL flow session, there can be multiple user device state transitions between idle and active. The core network manages the control plane signalling of the radio technology whereby advanced methods can be implemented to reduce the state transition overhead during flow setup, thereby reducing the latency. For ULL flows, irrespective of whether the traffic is intermittent or has a constant bit rate, the end-to-end latency should be minimized for both flow setup and steady state traffic flow.

An efficient backhaul network design can reduce control plane signaling for both initial ULL flow setup and steady state traffic. We give brief overviews of the two standardization efforts that efficiently implement the 5G core network functions for setting up and supporting ULL flows, namely Control and User Plane Separation (CUPS) of EPC and Next Generation (NG) Core.

Control and User Plane Separation (CUPS) of EPC: The SDN paradigm of separating the control and data plane functions while centralizing the overall control plane has provided substantial advantages in traditional networks. The 3GPP has proposed Control and User Plane Separation (CUPS) [2] for the Evolved Packet Core

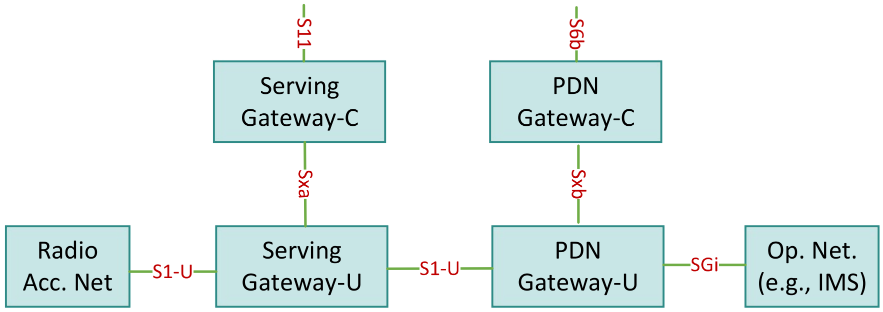


Figure 2.28: Illustration of Control and User Plane Separation (CUPS) for the EPC as proposed by the 3GPP. The Serving-GW (S-GW) functions and the PDN-Gateway (P-GW) functions in the EPC are split between S-GW-C (i.e., control), S-GW-U, and P-GW-U (i.e., user) to increase the flexibility of existing EPC networks [2].

(EPC) backhaul of the LTE radio technology, see Fig. 2.28, to adapt SDN principles in the cellular backhaul core networks to achieve similar benefits. Current network deployments are facing increased capital and operational expenditures when scaling the infrastructures to meet the capacity demands from the users. This infrastructure scaling problem is exacerbated by the integrated control and user plane functions in the existing backhaul networks. CUPS targets *i*) flexible deployments in both distributed and centralized control plane, and *ii*) independent scaling of control and user plane functions.

CUPS plays an important role in reducing the overall end-to-end latency through the cellular operator networks by selecting the user plane nodes that are close to the RAN node. In particular, the data is transported without having to interact with the control plane nodes for the path setup, which is especially beneficial for user mobility scenarios. That is, the flow paths of user plane nodes are dynamically adapted according to the requirements and mobility, without having to negotiate with control plane entities, such as SGW-C and PGW-C. This capability will greatly increase the backhaul flexibility of the existing LTE radio technology deployments. New interfaces, namely Sxa, Sxb, and Sxc, see Fig. 2.28, have been introduced to communicate between the control and user planes of the Serving-GW (S-GW). The main advantages of CPUS in comparison to the existing EPC are:

i) Removal of GPRS Tunneling Protocol (GTP) and session management between control plane entities.

ii) A cross connection interface between control and user plane, such that any control function can interact with any user function.

iii) A UE is served by a single control plane, but the data flow path may traverse multiple data plane functions.

iv) A control plane function is responsible for creating, managing, and terminating a flow over the user plane functions. All 3GPP control functions, such as PCC, charging, and admission control are supported within control plane function, while the user plane is completely agnostic to the 3GPP control functions.

v A legacy EPC consisting of S-GW and PDN-GW can be replaced with new user plane and control plane split nodes without any impact on existing implementations.

Summary and Lessons Learned: CUPS provides a mechanism to adapt advanced resource management functions, such as SDN, to existing networks while improving the flexibility. The reduction of data plane and control plane overhead, particularly the removal of GTP tunneling, allows user data to be transported without encapsulation and without GTP sessions. Moreover, the user device state transitions trigger control plane activities in the core networks. Therefore, the separation of control and data plane not only increases the flexibility, but also reduces the radio control signalling to support ULL flows. Thus, cellular operators can incrementally upgrade towards 5G deployments. For distributed deployments, future research needs

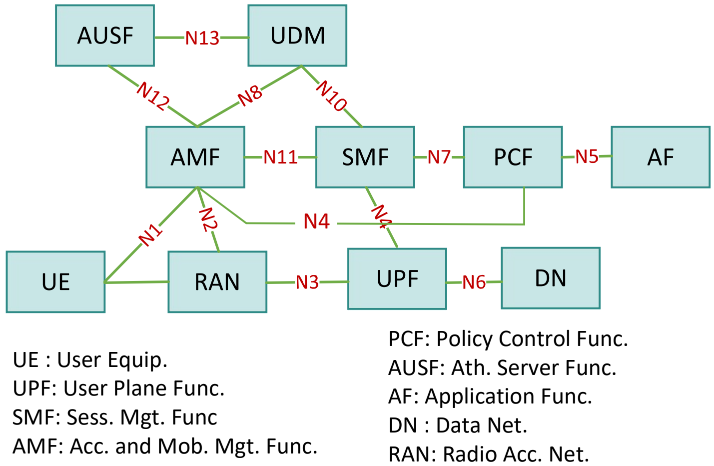


Figure 2.29: Illustration of 3GPP Next Generation (NG) Core: Point-to-point reference architecture based on service functions to support 5G radio nodes [229].

to thoroughly examine the placement and implementation of control and user plane entities without impacting the overall EPC system behavior.

Next Generation (NG) Core

Overview of NG Core Architecture: The 3GPP Next Generation (NG) core [229] is equivalent to the LTE Evolved Packet Core (EPC). However, the NG core network has been redesigned to separate and isolate the network nodes based on service functions, i.e., functions related to the radio service, such as user authentication and session management. While the EPC core provides the network functionality for the LTE backhaul, the NG core specifically provides the backhaul for the standalone 5G New Radio (NR) technology [1]. A non-standalone 5G would operate in coexistence with EPC and LTE support.

The existing EPC core collectively implements the LTE radio service functions in a combined fashion within the backhaul network gateways, such as S-GW and P-GW. In contrast, the NG core separates the service functions at the network nodes level. The service function concept is akin to Network Function Virtualization (NFV) in that multiple virtualized network functions are needed to implement a single service function.

NG Core Elements: The point-to-point NG core architecture is based on service functions supporting the 5G radio nodes, as show in Fig. 2.29. The fundamental motivation of the NG core is to support advanced network implementations and network management schemes, such as network slicing, NFV, network service function chaining, and SDN to address the scalability and flexibility of the core network. Each NG core element is connected to other elements through Nx interfaces. Critical NG core elements include:

i) The Access and Mobility Function (AMF) implements the access control and mobility aspects of the user context.

ii) The Session Management Function (SMF) is responsible for the data path setup and tracking and terminating based on the policy function.

iii) The User Plane Function (UPF) defines the data path characteristics based on the users requirements and policy.

iv) The Policy Control Function (PCF) controls the user policy, such as roaming and network resource allocations, for network management, including network slicing.

v) The Unified Data Management (UDM) manages the subscriber information which is used for admission control and for defining the data path policies.

vi) The Network Repository Function (NRF) maintains the registry of service functions distributed throughout the network.

Summary and Lessons Learned: The NG core decouples the network service functions from the gateway nodes, allowing the core network to implement the network nodes based on service functions, which enhances the deployment flexibility. As a result, operators have more freedom in transitioning from an existing core network to the NG core by separating the core network elements based on the service functions. However, future research needs to thoroughly examine the overhead of the control plane management, e.g., virtualization [195]. For instance, the overhead directly influences power consumption, and network efficiency for the ULL flow setup in the core network data path, which must be carefully evaluated. Therefore, performance, resource utilization, and overhead must be considered while designing the optimal infrastructure deployment.

Discussion on 5G ULL Standardization

In this section we have provided a brief overview of key components in the 5G standardization efforts that contribute to ULL connectivity. Several wireless connectivity and signalling optimizations have reduced the latency overhead in the data and control planes of the wireless air interface. Also, the new Radio Resource Control (RRC) inactive state reduces the signalling for the RRC inactive to active state transition (compared to the conventional LTE RRC idle to RRC active transition). A wide variety of options, e.g., functional splits of CPRI and NGFI for the fronthaul, exist for meeting the requirements of 5G components. Therefore, the design of an end-to-end 5G supported system requires a comprehensive latency analysis across all segments to select the right candidate set of transport mechanisms, protocols, and architectural solutions.

Broadly speaking, the improvements that the TSN standards bring to bridged networks can feed into novel standard developments for Ultra-Reliable and Low Latency

Communications (URLLC) in cellular networks in two main areas: *i*) backhaul network, and *ii*) fronthaul network. In traditional cellular networks, the various backhaul network nodes, such as the Home Subscriber Service (HSS) and the Radio Network Controller (RNC), are typically interconnected by bridged networks. The adoption of TSN improves the capabilities and enhances the performance of the bridged networks that interconnect the backhaul nodes. In contrast, fronthaul nodes, such as the Remote Radio Head (RRH) and the Cloud-RAN (C-RAN), were typically interconnected by point-to-point optical links (as opposed to the bridged networks) as the fronthaul interconnections have very strict latency and throughput requirements. The introduction of TSN enables bridged networks to provide the strict latency and throughput requirements needed for the fronthaul. Thus, TSN can enable the end-to-end URLLC support across both the fronthaul and the backhaul for cellular networks.

Overall, the adaptability of each solution for 5G deployment could impact the end-to-end ULL flow latency. Flexibility could improve the scalability and network utilization, but the control plane separation requires careful consideration of control plane overhead and latency. Similarly, deployments of new architectures, such as NG core, could result in efficient backhaul management to support ULL mechanism with minimal overhead, but may require high expenditures for cellular operators. Nevertheless, as deployment options vary widely based on the implementation, relative performance evaluation based on distances between different nodes, interfaces, protocol overhead, transport mechanisms, and architectural consideration need to be conducted in future research as ground work towards optimal 5G system design.

2.4.2 5G ULL Research Studies

This section surveys the research studies on 5G ULL mechanisms following the classification in Fig. 2.30. In particular, we first give a brief overview of the main

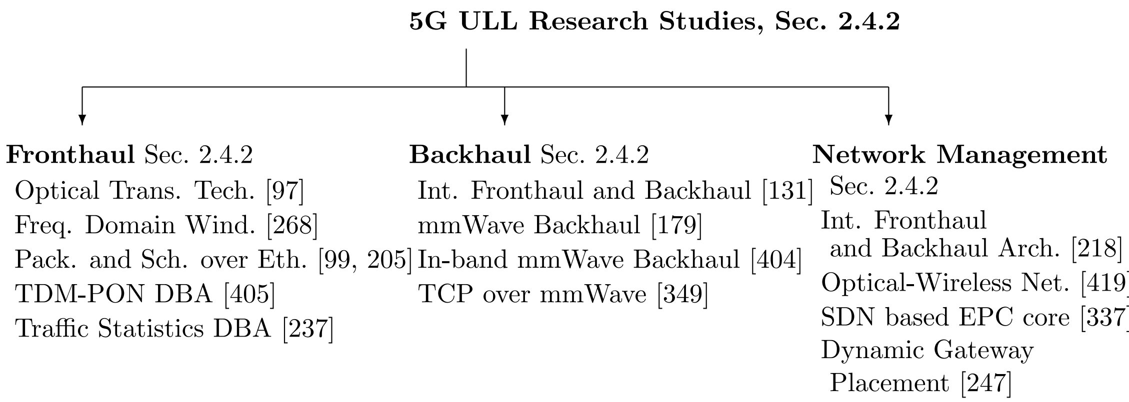


Figure 2.30: Classification of 5G Research Studies.

ULL research directions in the 5G wireless access segment and refer to the extensive 5G wireless access literature for more details [34, 89, 127, 139, 344, 400]. Then we survey in detail the research studies addressing ULL in the fronthaul, backhaul, and network management of fronthaul and backhaul.

5G Wireless Access ULL Research Studies

In this section we give a brief overview of the main research directions on ULL techniques in the 5G wireless access segment. Efforts to reduce the latency in the wireless access segment have been mainly focused on two aspects: *i*) shortening of the Transmission Time Interval (TTI), and *ii*) reduced processing time for each TTI [308]. The TTI is the fundamental time unit for the protocol operations, e.g., transmissions, in a given wireless technology, e.g., LTE. A shorter TII contributes to an overall reduced Round-Trip Time (RTT) due to shorter cycles. For example, in LTE, the number of OFDM symbols in one TTI can be reduced from 7 to 2 or 3 OFDM symbol to reduce the latency [35]. In contrast to LTE which uses only Orthogonal Frequency Division Multiplexing (OFDM) based waveforms, the New Radio (NR) access technology [343] for 5G provides a platform to design and implement more flexible waveforms based on both OFDM and non-OFDM over a wide range of spectrum resources, including

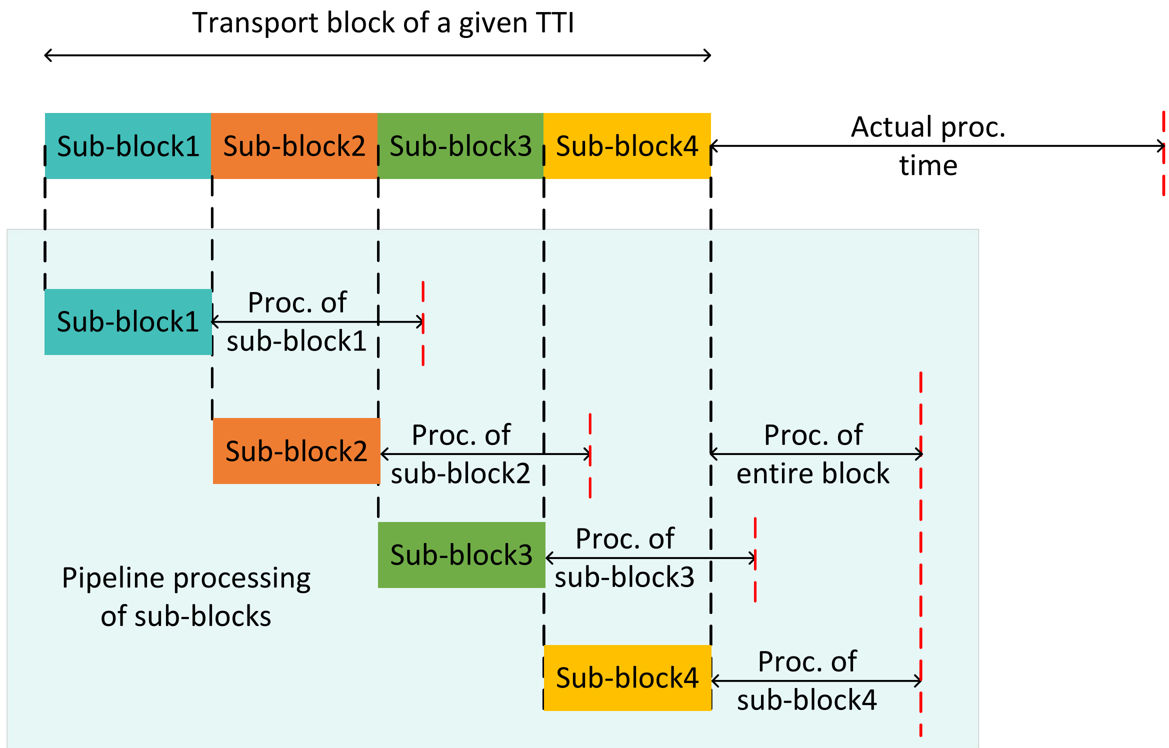


Figure 2.31: A given frame can be divided into multiple sub-blocks. Each sub-block is independently processed without having to wait for the entire frame to arrive to start the processing, reducing the overall latency [308].

microwave and mmWave [88].

In terms of reducing the TTI processing time, if a given TTI is divided into multiple sub-blocks, and each block is independently processed in a pipelined fashion, the overall processing time can be reduced [308], as illustrated in Fig. 2.31. The independent processing of sub-blocks incurs an overhead in terms of both the physical wireless resources (i.e., Resource Element (RE)) mapping and the processing overhead for demapping. The mapping and demapping operations mainly involve table lookups and minimal arithmetic computations. Thus, current hardware implementation can readily accommodate this mapping and demapping processing overhead. Without pipelined processing, the radio node has to wait for the entire TTI frame to arrive before starting to process the symbol, incurring the delay.

Alternatively, the OFDM sub carrier spacing in the frequency domain can be increased, thus inherently reducing the TTI duration in the time domain, i.e., reducing the OFDM symbol duration. However, such techniques require increased guard bands in both the frequency and time domains to protect from inter-carrier and inter-symbol interferences as well as increased hardware complexity in terms of tight synchronization and sensitive receiver designs.

The next generation Node B radio node in the context of 5G is often referred to as gNB; this gNB is equivalent to the eNB in 4G LTE. For simplicity, we follow the common eNB terminology to refer to the radio node in both legacy and 5G technology. The wireless link latency in 5G networks can typically be attributed to two sources: *i*) user plane latency when the User Equipment (UE) is in the CONNECTED state (i.e., active radio link is established between UE and radio node (eNB/gNB)), and ii) control plane latency when device is in idle state (i.e., no active radio link connectivity exists). The user plane latency in the uplink consists of the delays for the scheduling, and the UE to eNB transport, including the packet processing. The wireless control plane latency consists mainly of the delays for the state change from IDLE to CONNECTED through a signaling process, such as PAGING and Random Access CHannel (RACH). With increasing numbers of devices connecting to 5G networks, robust scheduling mechanisms are essential to preserve the fairness among all the devices in terms of latency and data rate. Intermittent data generation, e.g., in IoT, increases the control plane signaling due to the IDLE to CONNECTED transitions [291]. Furthermore, in small cell environments, the device mobility, e.g., for automotive and industrial robot applications, can result in additional data and control plane delays. The additional data plane delay in mobility scenarios is associated with the wireless link discontinuity during the handover process. Whereas, the control plane delay in the mobility scenarios is associated with the signaling over the core network due to device transitions between eNBs.

Robotic systems in industrial networks require ULL for control system loops. As compared to unlicensed wireless access (e.g., WiFi), the licensed LTE and 5G technologies not only provide ultra reliable and low latency connectivity for a closed ecosystem of industrial networks, but also support seamless mobility for robotic systems [61]. The scheduling of data from the devices is a MAC layer procedure which incurs significant delays in 5G wireless networks. To address the scheduling delay, pro-active granting, similar to Semi-Persistent Scheduling (SPS) [384], i.e., periodic grants to device for transmission, can be employed. However, pro-active granting could reduce the overall link utilization due to over-provisioning of scheduling resources. In LTE with 1 ms TTI, the Round Trip Time (RTT) for a Scheduling Request (SR) and GRANT is at least 4 ms, resulting in data transmission delays of 8 ms or more. Proactive granting can reduce the packet delay to less than 4 ms by eliminating the SR and GRANT procedures.

Fronthaul

The fronthaul segment connects the radio nodes, i.e., radio transmission nodes, to the radio processing nodes, i.e., radio signal processing [351]. Typically, radio nodes are referred to as Remote Radio Units (RRUs) and radio processing nodes are referred to as Base Band Units (BBUs). Cloud-RAN (CRAN) technology [101] centralizes and virtualizes the BBU functions such that a given BBU can connect to and serve several RRHs. Initial CRAN designs entirely virtualized the BBU functions and transported only time domain In-Phase/Quadrature (I/Q) samples to RRHs. However, the time domain I/Q transport technology was limited by strict delay and bandwidth requirements that hampered the scalability of deployments. Recent CRAN designs feature flexible BBU function separation between CRAN and RRU to meet scalability and latency demands [71, 437]. While there exist extensive discussions on fronthaul challenges and future designs [45, 84, 98, 368], we focus on the key aspects of fronthaul techniques supporting ULL connectivity.

Optical Transport Techniques: The Common Public Radio Interface (CPRI) [130], see Section 2.4.1, imposes an overall fronthaul link delay limit of 5 ms, excluding the propagation delay [130]. Typically, the distance between BBU and RRU is 20 km with a delay tolerance of 100 μs and a frequency accuracy within 2 ppm (parts per million). In addition to the CPRI requirements, the deployment consideration should also consider the availability of fiber, cost efficiency, CPRI propagation delay, as well as administration and management, since fiber providers are typically different from mobile network providers. The main topology consideration for deployments are the point-to-point, daisy chain, multi-path ring, and mesh topologies. Point-to-point links provide dedicated fiber resources for the fronthaul connectivity, but can be expensive. The daisy chain topology allows the fiber resources to be shared among multiple RRUs; however, a link failure can impact all the connected RRUs. Multipath ring and mesh topologies provide generally a better balance between fiber availability, cost, and resilience to link failures. Fronthaul data can be transported through several optical transport techniques [97]:

Optical Transport Network (OTN): The OTN uses a TDM approach over a single wavelength which can be extended to multiple wavelengths through dense wavelength division multiplexing (DWDM). OTN has relatively high power consumption, as OTN equipment requires power for the optical transmissions at both receiver and transmitter.

Passive Optical Network (PON): PONs may provide a cost-effective option for fiber deployments, if PONs are already deployed for fiber to the home connections.

Recent PON developments [81, 96, 222, 289, 300, 442] support both high bit rates and low latencies to meet the fronthaul requirements. PON technology is also power efficient as compared to the OTN.

Point-to-Point with CWDM: Point-to-point links with a wavelength multiplexer for Coarse Wavelength Division Multiplexing (CWDM) are generally cheaper than an OTN with DWDM. Motivated by diverse optical transport options, Chanclou et al. [97] have proposed a WDM optical network solution to meet the data rates and latency requirements of the CRAN fronthaul. Automatic wavelength assignment is enabled by passively monitoring the RRUs through a self-seeded approach [372] that considers the bit rates, latencies, jitter and synchronization, as well as fiber availability of the CPRI links.

Frequency Domain Windowing: The general 5G end-to-end latency guideline is 1 ms, while the total fronthaul link (propagation) delay budget is 200 μs [266]. Consider a 20 km fronthaul link, then the processing delay (for CPRI signal and protocol processing) would need to be significantly lower than the link (propagation) delay, i.e., on the order of a few μs. The general consideration for the processing delay in the fronthaul is 5 μs. In an effort to further reduce the processing delay of 5 μs, Liu et al. [268] have designed an optical transport system supporting the CPRI-equivalent rate of 59 Gbps. 48 LTE RF signals of 20 MHz each were transmitted through a single WDM channel with an effective RF bandwidth of 1.5 GHz. The processing delay was reduced through a Frequency Domain Windowing (FDW) technique that reduces the overall FFT/IFFT size in the process of channel aggregation and de-aggregation. FDW is applied to each N-point IFFT corresponding to every aggregated channel. The FDW technique attenuates the high-frequency components such that the inter-channel crosstalk is reduced. As a result, the effective FFT/IFFT size can be reduced, thereby reducing the overall processing latency. The experimental results for the fronthaul distance of 5 km have shown an overall fronthaul delay reduction from 5 μs to 2 μs.

Packetization and Scheduling over Ethernet: Similar to optical transport of I/Q data from BBU to RRH, I/Q data can be digitized and packetized for the transmissions over Ethernet. Radio over Ethernet (RoE) [41, 42, 62, 63] defines the process of converting radio signal I/Q data to packets which can be transported over Ethernet. The main issues associated with the packetization process while encapsulating the I/Q data over the fronthaul link are: *i*) overhead, *ii*) packetization latency, and *iii*) scheduling delay. The packetization overhead results from the frame and packet headers. Therefore, to reduce the overhead, each frame must be created with the maximum I/Q data possible such that the overall number of packets and Ethernet frames is minimized. However, a large frame size adds wait time for the data filling up the maximum frame size. Hence, reducing the latency requires the transmission of short frames.

The scheduling of Ethernet frames can provide multiplexing gain through resource sharing, however, the scheduling can incur queuing delays. Therefore, to achieve low latency the overhead, packetization latency, and scheduling delay must be carefully considered. Chang et al. [99] have evaluated the CRAN performance in terms of packetization and scheduling on the Ethernet fronthaul. For functional splits along layer boundaries, for instance when the complete PHY layer is implemented in the RRH, or the complete MAC and PHY layers are implemented in the RRH, an RRH Ethernet gateway has been introduced to perform the scheduling, aggregate the traffic from RRH nodes, and discard the packets which are past their deadlines. For instance, look-ahead depth packetization packs channel estimation I/Q data such that the channel estimation data precedes regular payload data in the demodulation. That is, demodulation does not wait for all the frame I/Q data to process the I/Q data related to channel estimation.

In contrast, the prefetch method [99] waits uniformly over all the I/Q data for the packetization to receive the Reference Signal (RS) symbols consisting of I/Q for channel estimation. More specifically, the packetization process is performed for transporting the I/Q data to the base band processing module only when all the required I/Q symbols corresponding to the RS within the look-ahead depth buffer have been received. Thus, transporting the I/Q data needed for the channel estimation has priority as compared to regular I/Q data. Various scheduling policies were applied to study the impact of the packetization process based on first-come, first-served (FCFS), shortest processing time (SPT), least remaining bit (LRB), earliest due date (EDD), and least slack time (LST). The performance analysis evaluated the maximum number of RRHs supported over the RRH link for a given Ethernet link capacity, packet size, scheduling policy, and functional split. The simulation results showed that packetization techniques (e.g., look-ahead depth and prefetch) while employing the LRB scheduling policy with packet discarding provided a significant multiplexing gain and supported the maximum number of RRHs. In a related research effort, Hisano et al. [205] have adapted the gating mechanism (see Section 2.2.4) to support low-latency 5G fronthaul.

TDM-PON Dynamic Bandwidth Allocation: In a PON system, distributed Optical Network Units (ONUs) connects to a central Optical Line Terminal (OLT) via a shared optical fiber. The transmissions from the ONUs to the OLT are controlled by a scheduler implemented at the OLT. In a TDM-PON system, the OLT coordinates the transmissions from multiple ONUs such that there are no collisions on the shared

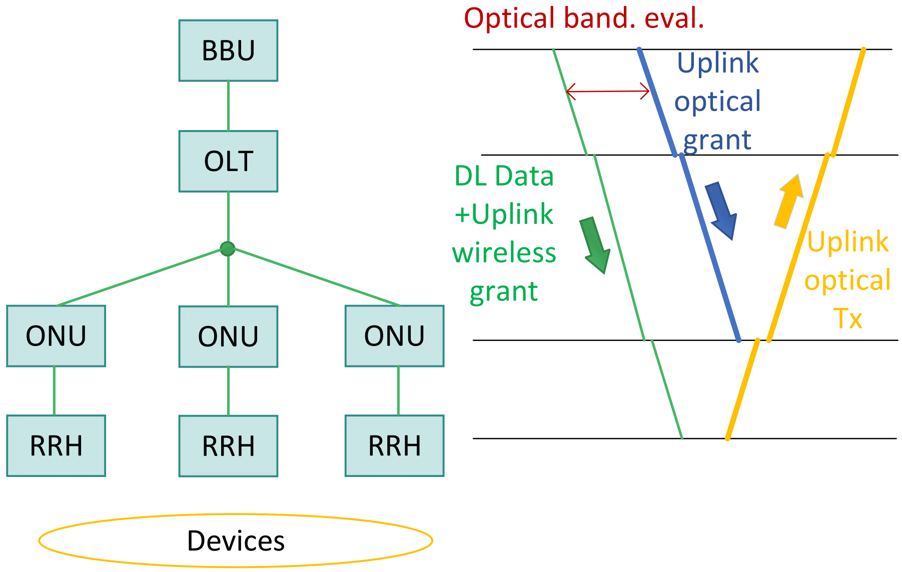


Figure 2.32: DBA scheme optimizing latency: Grants for the optical transmissions are evaluated in advance and sent to ONUs based on the wireless uplink information which is known to the BBU [405].

fiber. The Dynamic Bandwidth Allocation (DBA) mechanism assigns the transmission resources to ONUs based on the QoS deadlines. For each DBA polling cycle, each ONUs transmits a REPORT message indicating the queue size to the OLT. The OLT processes the REPORT messages from all ONUs to determine the transmission schedule. The transmission schedule is then sent to all the ONUs with GRANT messages indicating the exact transmission details for each specific ONU. This polling DBA mechanism consists of reporting the demands and waiting for the grants from centralized scheduler; therefore, typically, the total end-to-end PON delay is on the order of milliseconds [66, 292], i.e., much higher than the fronthaul requirements of a few micro seconds. A PON system in the CRAN framework connects the RRHs to ONUs, and the BBU to the OLT. Thus, the BBU can schedule transmissions from the RRHs. Due to the PON delay characteristics, the PON system is not readily suitable for fronthaul application.

To address the PON delay, Tashiro et al. [405] have presented a novel DBA mechanism specifically for fronthaul applications. As the BBU assigns the grants for wireless upstream transmissions of the devices attached to an RRH (i.e., ONU), the RRH upstream bandwidth requirements are known at the BBU (i.e., OLT) ahead in time. In wireless LTE systems, the request reporting to grant reception (related to wireless scheduling) is separated by 4 ms in the protocol operations, similarly the grant reception to RF transmissions is separated by 4 ms. Hence, the total protocol delay from request to transmission is 8 ms. As illustrated in Fig. 2.32, concurrent to the grant evaluation for wireless transmissions, grants for the optical transmissions of the RRHs (i.e., ONUs) can also be evaluated and transmitted to the RRHs ahead of time, eliminating the report and grant cycle between ONUs and OLT. The experimental evaluation of a TDM PON system with advance scheduling has demonstrated average end-to-end latencies of less than 40 μs, and packet jitters of less than 25 μs for fronthaul distances up to 20 km.

Traffic Statistics Based Bandwidth Allocation: Fixed Bandwidth Allocation (FBA) can address the overhead and scheduling delay incurred by the DBA mechanism, but fixed bandwidth allocations may waste resources due to over provisioning. For variable traffic, statistical multiplexing can be employed to increasing the bandwidth and resource utilization. Based on this principle, Kobayashi et al. [237] have proposed a TDM-PON bandwidth allocation scheme based on the traffic statistics of the variable fronthaul traffic. The proposed scheme considers the long term traffic characteristics on the order of several hours. The allocated bandwidth is then adapted based on the estimated long term mean and variance, which can, for instance, be obtained through monitoring the packet traffic with software defined networking based techniques [188, 264, 447], the bandwidth allocation requests [272, 463], or monitoring the optical signal levels [204]. The estimated bandwidth allocation is applied over the subsequent time period, and a new bandwidth allocation is estimated for each time period. The experimental results demonstrated end-to-end fronthaul latencies of 35 μs, while the effective link bandwidth utilization was increased by 58% compared to FBA.

However, one of the shortcomings of the proposed bandwidth allocation based on traffic statistics is that it does not consider the specific fronthaul split option. For a traditional CRAN, where the RF I/Q samples are transported from RRH to BBU, a constant bit rate is required at all times; thus the FBA can efficiently meet the fronthaul requirements. Traffic variations according to varying user activity occur only for higher order functional CRAN splits. Therefore, traffic statistics based bandwidth allocation is limited to higher functional split fronthauls with a split position towards the upper end of Fig. 2.24.

Summary and Lessons Learned: In a typical CRAN deployment where the RF I/Q is transported from RRH to BBU, the fronthaul traffic is independent of the user data which results in a constant bit rate over fronthaul links at all times to support the normal operations of BBU and RRH. Hence, there can be significant power consumption overhead for the CRAN deployment [92, 403]. Therefore, the new designs of fronthaul solutions should consider the overall energy consumption in addition to the end-to-end latency [435]. Several advanced physical layer techniques, such as, modulation, detection, and DSP (e.g., I/Q compression) for fiber transmissions have been proposed as part of energy efficient designs [267, 322, 413]. While the higher order functional splits provide statistical multiplexing gains, the worst-case delay must be analyzed to ensure that latency is within the delay budget of the fronthaul link. The fronthaul infrastructure is typically non-flexible and must support the deployments of future 5G networks [187]. Therefore, the fronthaul designs, such as bandwidth allocation and resource sharing mechanism designs, should be able to readily accommodate new developments in the 5G technology. Although several techniques exists to mitigate the delay in fronthaul networks, there has been no research yet to address the synchronization of RRH and BBU to a universal timing. Flexible fronthaul techniques can be developed based on reconfigurable network functions and physical layer entities, such as modulators and transparent spectral converters, in the framework of Software Defined Optical Networks (SDON) [284]. For instance, Cvijetic et al. [125] have proposed an SDN based topology-reconfigurable optical fronthaul architecture. The dynamic reconfiguration of fronthaul can support low latency inter BS communications necessary for bidirectional Coordinated MultiPoint (CoMP) for inter-cell interference cancellation and inter-cell D2D.

Backhaul

Integrated Fronthaul and Backhaul: The backhaul connects Radio Access Networks (RANs) to core networks, e.g., the LTE backhaul connects the RAN eNB node (base station) to the Evolved Packet Core (EPC) core network. Typically, in CRAN technology, the RRH only implements a split part of the eNB functions, for instance, the eNB PHY layer is implemented at the RRH, while the MAC and higher layers are implemented at the BBU. Thus, the RRH, the BBU, and the fronthaul connecting them, jointly constitute an eNB. Thus, if the endpoints of a link in a 5G network are the RRH and BBU, then the link operates as a fronthaul. On the other hand, if the endpoints are the eNB and EPC, then the link operates as a backhaul. With the centralization of the computing in the core network, such as in a CRAN, the BBU and EPC can be implemented at a single physical location which enables the deployment of a common infrastructure in an architecture to support both eNBs and RRHs over a common platform.

The crosshaul (Xhaul) architecture [131] provides a common platform to support both fronthaul and backhaul using an Xhaul transport network. In the SDN framework, the Xhaul transport network provides reconfigurability while operating over heterogeneous switches and links, such as microwave, mmWave, optical, and high speed Ethernet. In an effort to ensure the ULL capability of configurable integrated fronthaul and backhaul networks, Li et al. [258] have proposed an X-Ethernet based on Flexible Ethernet [212] technology for the Xhaul architecture. The experimental demonstration of X-Ethernet has demonstrated an average latency of 640 ns as compared to 30–50 μs in a traditional Ethernet switch, indicating that X-Ethernet can be deployed as a part of the Xhaul data plane. As the control plane latency of X-Ethernet for reconfigurations has not been identified, the overall suitability of X-Ethernet for Xhaul needs further investigations.

MillimeterWave (mmWave) Backhaul: MillimeterWave (mmWave) radio technology for wireless communications operates in the spectrum between 30 and 300 GHz [88, 298, 429]. mmWaves have relatively short wavelengths and thus suffer pronounced signal attenuation with propagation distance and due to obstacles. Also, mmWaves exhibit high directionality. Therefore, mmWave technology exploits beamforming by focusing the signal energy in a narrow spatial beam to support longer propagation distances. Nevertheless, the typical operational range of mmWave links is in the range of several hundred feet. Longer distances require several intermediate repeaters which increase the latency. On the positive side, the high attenuation property of mmWave signals facilitates geographical frequency reuse; thus saving the operators spectrum resources by avoiding co-channel interference.

The availability of high bandwidths in the mmWave spectrum can provide high capacity links which are potentially suitable for both fronthaul and backhaul. To date, mmWave research in the context of 5G networks has mainly focused on the backhaul [132, 404] and we survey the mmWave based techniques that specifically target ULL transport. Generally, the latency requirements in the backhaul are relaxed compared to the very strict latency requirements for the I/Q user data transport in the fronthaul. Thus, mmWave transport with its required repeaters for covering distances beyond a few hundred feet is generally better suited for backhaul. Future research may examine whether it is possible to exploit the high capacity mmWave transport for fronthaul. Also, mmWave transport may be suitable for particular 5G connectivity scenarios, e.g., for connecting a Customer Premises Equipment (CPE) home gateway to an external serving gateway, e.g., a 5G base station (gNB).

Gao et al. [179] have presented a mmWave based backhaul for 5G using massive-MIMO to support a high number of radio nodes, i.e., Base Stations (BSs). The proposed approach exploits Beam Division Multiplexing (BDM) whereby an independent beam is dedicated to a BS, thus creating a backhaul link through spatial multiplexing. Each mmWave beam supports a high capacity link, hence, a Time Division Multiplexing (TDM) scheduling can be employed to share the resources within a single beam, supporting multiple BSs over a single link. However, the scheduling of BDM resources with TDM can incur significant end-to-end latency as compared to BDM without TDM, and therefore must be carefully evaluated specific to the backhaul latency requirements.

In-band mmWave Backhaul: The in-band mmWave technique shares the spectrum resources with the wireless access (i.e., BS to device), and backhaul (i.e., BS to BS and BS to core network). Since the wireless access and backhaul resources compete for the same spectrum resources in the in-band communication, there can be significant overhead in terms of capacity and latency. To analyze the in-band mmWave communications in terms of capacity, Taori et al. [404] have conducted a feasibility study and showed that 25% of the resources of the mmWave link is sufficient to support the user data rates over the wireless link up to 0.8 Gbps. Typically, in the in-band backhaul connectivity, the resources are shared in TDM fashion between wireless and backhaul applications impacting both wireless and backhaul end-to-end connectivity during congestion. Although the suitability of in-band communication is justified in terms of capacity, the implications of in-band communication on the latency has not been characterized, and hence can compromise the performance of the entire end-to-end connectivity if not carefully considered. Taori et al. [404] have also proposed a point-to-multipoint transmission for BS to BS (inter-BS) communication based on in-band mmWave backhaul connectivity. Inter-BS communication is necessary to support mobility features, such as handover and redirection, as well as advanced radio features, such as inter cell interference cancellation using Coordinated MultiPoint (CoMP) and self organizing networks. As the deployments of BS increase to meet the capacity demands through small cells, the demand for coordination among neighboring BSs will increase. Hence, inter-BS communication is an important aspect of 5G that needs be addressed in a flexible, simple and cost effective manner. In-band mmWave connectivity provides a cost effective solution for inter-BS connectivity along with flexibility due to a wireless connection, as compared to the physical deployment of optical fiber infrastructure. Point-to-multipoint mmWave connectivity results in a simpler and cost effective solution through a dynamic reconfiguration of mmWave links based on the requirements.

TCP over 5G mmWave Links: mmWave links have typically high bandwidths, but are prone to outages as they require Line-of-Sight (LoS). Thus, there are high chances for temporary link disruptions, which can result in temporary congestion.

TCP congestion control could negatively impact the overall capacity and the latency when a link is temporarily interrupted as a result of buffer bloating. Active Queue Management (AQM) can be applied to adaptively drop packets from the queue such that the queue size is contained for a particular flow to keep the end-to-end delay on average below a threshold. Control Delay (CoDel) [324] is an AQM technique which ensures short packet sojourn delays, i.e., short packet delays from ingress to egress. Each packet is time-stamped at the ingress and elapsed time is evaluated for the packet drop decision. Building on the well-known non-linear relationship between drop rate and throughput in TCP [287], the time interval between packet drops is reduced inversely proportional to the square root of the number of drops so as to linearly vary the throughput in relation to the drop count [324].

To investigate the impact of temporary 5G mmWave link disruptions on end-to-end network connections, Pieska et al. [349] have evaluated the TCP performance tradeoff between capacity and latency. The evaluation indicated that the disruption duration and frequency directly impact the TCP performance in addition to the aggressiveness of the TCP variant, such as TCP Reno, TCP Illinois, TCP Cubic, and TCP Scalable. Although CoDel is a promising technique in curtailing the buffer bloat in regular TCP networks, Non-LoS (NLOS) occurrences of a mmWave link can result in significant throughput loss of TCP over mmWave links due to extensive CoDel packet dropping, especially for a single flow of the TCP Reno variant. However, the evaluations indicated that CoDel can achieve low latency and fast recovery for flows with short RTTs and disruption durations. Nevertheless, to avoid the implications of buffer bloat, new TCP designs should be able to accommodate short link disruptions, specifically for 5G mmWave connectivity for access, fronthaul, and backhaul.

Summary and Lessons Learned: Small cells where the devices are close to the radio nodes are widely adopted to save power and to offload the burden on the macro wireless cells [441]. However, the small cell traffic needs to be eventually aggregated at the backhaul, resulting in demanding requirements for the small cell connectivity with the core networks. The connectivity can be provided through fiber backhaul links that can be shared through FiWi techniques among multiple wireless nodes [78]. mmWave technology is another promising technology for meeting the high bandwidth and ULL requirements for next generation connectivity, such as, small cell backhaul supporting 5G, and fronthaul and backhaul sharing [246]. mmWave wireless links support *i*) high throughput with short symbol and frame durations, and *ii*) high user numbers at a given radio node. Thus, mmWave backhaul can increase the overall capacity of cellular networks in terms of supported flows with low-latency QoS. As compared to the power consumption of optical communications, the power consumption of mmWave links is typically significantly higher due to the scattering of wireless transmissions as compared to the guided optical waves in a fiber. Therefore, mmWave requires new energy efficient methods in resource management and shared backhaul and fronthaul for 5G applications.

In contrast, optical wireless communication [219] utilizes the visible light with similar characteristics as mmWave. In addition to the directionality (LoS) and spatial multiplexing properties, optical wireless communication suffers from interference due to ambient light sources. Similar to mmWave designs, the system design should be robust to accommodate disruptions due to temporary link obstructions. Future designs should also ensure synchronization on the order of 65 ns [3, 109, 426] while supporting the shared fronthaul and backhaul.

Network Management

ULL mechanisms are closely related to network management for meeting the flow demands in terms of resource allocation, reliability, congestion control, and end-to-end QoS. The increasing number of protocols that support the fronthaul and backhaul connectivity in a single end-to-end path creates a heterogeneous environment. The comprehensive (end-to-end) management of this heterogeneous network environment can be complex without the support of an inter-operative mechanism. Management mechanisms based on Software Defined Network (SDN) could provide a single platform for the coordination of a multitude of protocols [165, 213, 276, 414].

Integrated Fronthaul and Backhaul Architecture: Both Distributed-RAN (DRAN) and CRAN offer unique deployment options for cellular operators to enable cellular connectivity to the users. DRAN conducts the baseband signal processing at the remote Base Station (BS). As a result, the BS to core network (backhaul) connectivity has relaxed QoS requirements and thus can be leased in the access network domain. On the other hand, CRANs require dedicated fiber links (typically owned by the cellular operator) for connecting the radio nodes to the core networks. Therefore, 5G networks are expected to uniformly support DRAN and CRAN architectures for enabling cellular connectivity to the users.

Jungnickel et al. [218] have proposed an integrated fronthaul and backhaul based on SDN to commonly support DRAN and CRAN deployments for cellular operators. Traditional Ethernet deployment strategies [113], such as the E-tree, can be adapted for the CRAN, and the E-LAN for D-RAN based on their topology support. To utilize the existing fiber, independent wavelengths can be used to meet the latency and capacity requirements of the fronthaul and backhaul. For example, the backhaul can use TDM within a single wavelength that is shared among multiple radio nodes, and

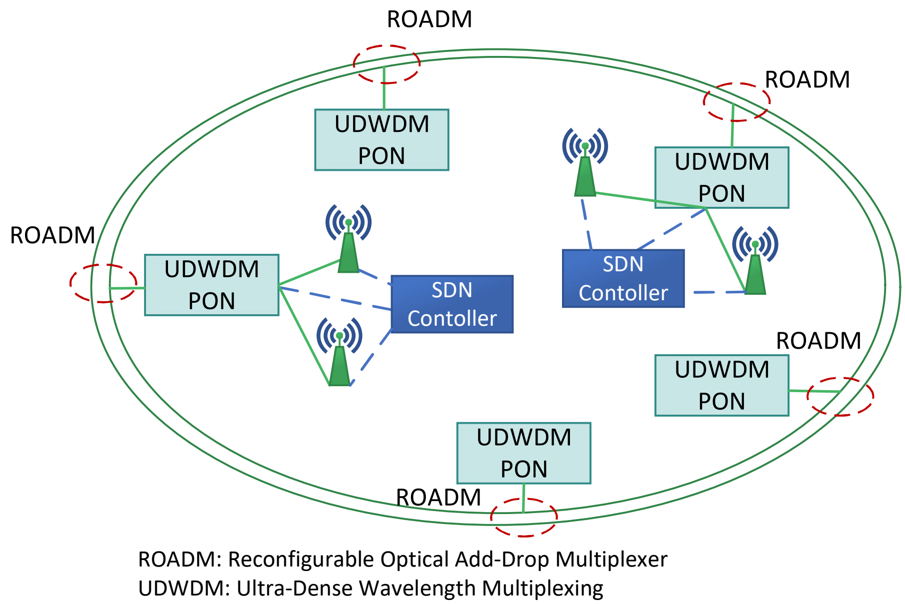


Figure 2.33: Simplified version of ULL optical wireless architecture where WDM ring connects to wireless nodes and SDN controller through PON framework [419].

the fronthaul requires a dedicated wavelength between radio node and CRAN. However, the sharing of traditional access networks in E-Tree and E-LAN mode can cause security issues. Nevertheless, SDN provides both flexibility of statistical multiplexing in both the optical and electrical domains, and security through the virtualization of the network infrastructure. In a similar study, Ameigeiras et al. [53] have proposed a hierarchical SDN architecture based on virtualization, as well as Ethernet and IPv6 technologies focusing on low latency.

Optical Wireless Networking: The inter-working of optical and wireless technologies has been explored in FiWi networks [65, 105, 263] and in the general context of optical-wireless integration in access and metro networks [37, 185, 282, 375, 445]. As next-generation applications demand ULL and high reliability, there is a great need to integrate optical and wireless technologies with minimal impact on the traditional cellular infrastructures, such as 4G LTE. Towards this end, the 5G STEP-FWD project [419] has been funded by the European Commission to develop novel networking solutions that closely integrate the optical and wireless technologies within the 5G framework.

Vardakas et al. [419] have proposed a high capacity and low latency 5G backhaul architecture as illustrated in Fig. 2.33. Network densification is supported by small cells which are connected to macro BSs through PONs, mainly: *i*) Optical Line Terminals (OLTs) connected through fiber links, *ii*) point-to-point dedicated links, and *iii*) local Optical Network Unit (ONU) connections through a fiber protection ring offered by dark fiber. The dark fiber utilization provides a cost effective solution as the infrastructure already exists. The wireless access by the small cells and backhaul connectivity supported by PONs are controlled by a unified SDN management framework. mmWave-UDWDM technology effectively utilizes the wavelength and space division multiplexing, while PONs provide effective backhaul connectivity. The SDN management can support dynamic reconfigurability to support advanced network features, such as self-organization and self-healing for ultra-reliable infrastructure networks.

SDN Based Evolved Packet Core (EPC) Networks: Pagé et al. [337] have presented an SDN architecture for the LTE Evolved Packet Core (EPC) to support low-latency towards an evolutionary 5G core network. OpenFlow technology has been integrated into the switching nodes that connect the BSs (i.e., eNBs) to the EPC. The advantages of SDN based switching include reduced need for protocol based transport services, such as GTP, elimination of the Serving-Gateway (S-GW) which conventionally provides flow based services, such as buffering and connection management. In contrast to the conventional LTE backhaul connectivity, where the S-GW anchors the connections of the eNBs to the P-GW, the SDN based EPC is managed by an SDN controller, which replaces the S-GW control plane functions. The S-GW data plane functions are replaced by the SDN supported switching nodes. Thus, the SDN architecture eliminates the data and control bearer based connectivity [117] by replacing the large GTP messages with small SDN control messages. Additionally, the SDN based switching nodes can assist in attach and mobility procedures to reduce the overall load on the EPC core. As a result, the overall end-to-end latency can be reduced by reducing the data plane and control plane latency introduced by the intermediate nodes in the EPC core.

Dynamic Gateway Placement: Lakkakorpi et al. [247] have proposed a low latency technique in an SDN based backhaul network architecture that is fully reconfigurable. The gateway functions and queue management are configured to achieve low latency by minimizing the flow reestablishment procedures. The SDN controller dynamically programs the switching nodes to implement the network functions based on the flow characteristics. More specifically, an anchor switching node is dynamically selected to implement the gateway functions and AQM based on the flow mobility characteristics. For instance, in case of frequent handovers, the flow path must often be reconfigured to pass from one gateway function node to another. Therefore, the gateway functions can be implemented deeper in the core networks for the specific flows with frequent handovers, such that only the path routing is updated during handovers. This implementation of the gateway functions in the core networks also distributes the gateway functions across the switching nodes, reducing the overall burden on the core network.

Summary and Lessons Learned: In addition to the optimization of handover latencies in the wireless access, the backhaul architecture should support lower handover latencies. Chen et al. [104] have discussed the need for efficient backhaul architecture to support ultra-short handover latencies. However, the discussions are limited to DBA mechanisms in PONs for optimizing the LTE X2 and S1 interfaces.

In 5G technology, handovers can cause temporary disruptions to large data flows which can result in buffer-bloat problems across the network. New congestion control mechanisms must be adapted to address the short and temporary disruptions due to handovers during large data transfers. SDN based strategies can help to address these challenging handover problems [448]. However, existing studies have not considered the control plane latency and complexity, which may significantly impact the overall end-to-end latency. Therefore while ensuring the flexibility and reliability in 5G networks, it is also important to consider the end-to-end latency, through infrastructure based solutions, such as, dense wavelength-division multiplexed (DWDM) optical ring transport networks [433] using dark fiber, which is both energy and cost efficient.

Discussion on ULL 5G Research Studies

There have been numerous research efforts in the wireless access segment of 5G networks. However, there is still a need for research to solve compelling technical challenges [257] in enabling ultra-reliable ULL communication. These research challenges include infrastructure reuse, as well as cost and power efficiency. Throughout, the implications of wireless access techniques on ULL services should be carefully considered. For instance, the emerging 5G New Radio (NR) platform proposes new waveform designs. The symbol and frame durations as well as the guard band durations (e.g., cyclic prefixes in the OFDM symbol) in these new waveforms would directly impact ULL services. Increasingly complex waveforms would require longer symbols and longer frames, not only because of limited receiver processing capabilities, but also to maintain the synchronous delay between uplink and downlink messages. Thus, increasing the waveform complexity would tend to increase the wireless round trip delay. Moreover, the channel characteristics, such as the maximum (mobility) speed of 5G user devices and the cell size influence the guard band duration. For example, a high speed train scenario requires a relatively long doppler correction. Similarly, rural deployments require large cells. In both situations, a long guard band (cyclic prefix) is preferred such that the inter symbol interference can be minimized. A long guard band (cyclic prefix) would imply relatively long symbols and frames which could negatively affect ULL services. Thus, the new waveform designs in the 5G platform should carefully consider the impact on ULL services throughout the development process.

With the radio node densification, user mobility between radio nodes is expected to increase dramatically, which can significantly increase the control plane complexity in terms of user context updates in the core networks. Therefore, a light weight (i.e., reduced user context) user information set must be managed by the core networks, as opposed to intense policy and security mechanisms that contribute to control plane complexity. End-to-end security can reduce the burden of security measures by the core network. Similarly, user activities can be tracked by the radio node to enforce the policy and QoS measures across the network.

SDN plays an important role not only for managing fronthaul, backhaul, and core networks, but also for reducing the network complexity by reducing the network function implementation in dedicated entities, such as policy enforcement and user authentication. SDN can also integrate the heterogeneous protocol operations through dynamic packet header manipulation such that the protocol overheads are minimized.

Content caching in edge nodes has been widely discussed for reducing the delivery latency in fog-RAN and edge computing domains [221, 262, 362, 383]. SDN provides a platform for caching content across the entire network as well as based on user demands, optimizing both content caching and latency. Although 5G technology is primarily focused on power optimization of user devices and wireless radio nodes [301, 334, 370], the overall energy consumption of the network responsible for the end-to-end packet delivery should also be considered in future designs.

2.5 Future Work Directions

In this section we discuss the main open TSN research problems and outline directions for future research efforts in TSN.

2.5.1 Time Sensitive Networks (TSN)

Inter-Scheduler Coordination

Time aware shapers implement local scheduling principles specific to each TSN node. The end-to-end time sensitive characteristics of a flow are established under the assumption that each TSN node in the flow path guarantees the time sensitive characteristics. However, if an intermediate TSN node fails to enforce the TSN characteristics due to overload, or due to scheduler or timing inaccuracies, the overall end-to-end flow characteristics can be compromised. This situation may be more likely for TSN nodes that are positioned where multiple flows can aggregate as opposed to the edge nodes (that are traversed by only few flows).

To address this shortcoming, future research should develop a robust inter-scheduler coordination mechanism. The coordination mechanism should facilitate interactions between the time aware shapers in the TSN nodes in a flow path to ensure the overall end-to-end time sensitive characteristics of the flow. For instance, upon frame reception at the destination, the overall end-to-end latency can be estimated and the information can be fed back to the nodes. The TSN nodes can then establish a self performance profile. The interactions of the time aware shapers would enable inter-scheduler coordination such that each TSN node can guarantee the time sensitive scheduling relative to the end-to-end behavior of the flow path similar to time-triggered scheduling [297].

However, time-triggered scheduling depends on time synchronization to synchronously trigger the scheduling over the entire flow path. In contrast, the inter-scheduler coordination enables dynamic changes of the scheduler policies, such as timing adjustments of frame transmissions (i.e., to delay or advance the transmissions in the scheduled time slots) correcting the synchronization inaccuracy. Thus, the time aware scheduler depends not only on the time synchronization, but also on the end-to-end flow characteristics. The inter-scheduler coordination can be enabled through a centralized mechanism. For instance, an SDN based control can monitor the end-to-end characteristics of the flows, and configure the timing advances and corrections of the time aware schedulers at specific TSN nodes as required.

In-band Control Plane Overhead

Control plane data in TSN network corresponds to the data generated from the control functions, e.g., for setting up connections, synchronizing nodes, managing flows, and tearing down connections. The impact of control plane data in TSN networks has been largely ignored to date in research and standardization. Control plane traffic could be transported with the in-band connectivity of the high priority Control Data Traffic (CDT) class, which carries time critical information from data sources, such as sensors. However, the control plane traffic would then compete with the CDT traffic.

Resource reservations in TSN networks to enable the deterministic time-sensitive properties are typically estimated based on CDT traffic requirements. Since the control plane traffic rates are generally significantly lower than the CDT traffic, the in-band control plane traffic is generally ignored in the system design and resource reservations. However, new use TSN cases, such as robotics and automated drones, may require the establishment of short lived TSN flows with commensurate frequent triggering of control plane activities. Thus, new use cases may significantly increase control plane data traffic. Therefore, new resource reservations designs, especially for the in-band control plane data transport should consider both the control plane data traffic as well as the CDT traffic in evaluating the resource reservation requirements. We anticipate that it will be particularly challenging to ensure the requirements of the varying and dynamic control plane data as compared to the steady CDT traffic.

Low Priority Deadline Traffic

TSN nodes preempt an ongoing low priority frame transmission for transmitting an incoming high priority frame to guarantee the absolute minimum TSN node transit delay of the high priority frame. Depending on the intensity of the high priority traffic, a low priority frame can be preempted several times. As a result, the end-to-end delay characteristics of the low priority traffic cannot be guaranteed as the preemption occurrences depend directly on the high priority traffic intensity. If the high priority traffic intensity is significantly higher than the low priority traffic intensity, then the end-to-end delay of the low priority traffic can be greatly increased. Generally, low priority traffic carries delay sensitive data, that is less critical than high priority traffic data, but still should be delivered within a worst-case deadline. In the current state of the art, there exists no mechanism in research nor standards to ensure the worst-case end-to-end delay of low priority traffic under preemption.

Therefore, future research needs to develop new mechanisms to ensure a bounded worst-case delay for low priority traffic in TSN networks. A key challenge in designing a bounded worst-case delay for low priority traffic is to not degrade the performance of high priority traffic. Rather, the new mechanisms should opportunistically accommodate low priority traffic transmissions to meet a worst-case deadline.

Impact of Synchronization Inaccuracy

Several techniques for improving the synchronization accuracy while minimizing the synchronization errors have been developed for TSN networks. However, there are a lack of studies that quantify the implications of synchronization inaccuracies on the TSN network performance in terms of end-to-end delay and throughout. For low cost devices which are typically employed in large scale networks and for remote applications in IoT scenarios, the synchronization may not be as accurate as for industrial and robotic applications. Due to synchronization errors in TSN nodes, the transmissions scheduled by the time-aware shaper over a particular time slot, can extend or advance to adjacent time slots, which can impact the overall scheduling mechanism in a TSN node. For instance, in a time-triggered network, where all the TSN nodes schedule a flow based on synchronized timing information, synchronization errors can offset the time-triggers which can miss the schedule of a very short frame depending on the timing offset duration. Therefore, the performance impact due to synchronization errors for multiple priority traffic classes, frame sizes, and timing offset durations requires a close investigation.

Ingress and Egress Nodes for TSN

TSN networks are typically implemented in closed environments, such as in-vehicular and industrial control environments. However, most use cases require external connectivity to inter-operate with other networks. So far, no mechanism exists for establishing a common platform for the inter-operation of TSN networks with external non-TSN networks. We envision the inter-operation of TSN networks with non-TSN networks in two ways: *i*) centralized SDN management, and *ii*) ingress and egress based management for the TSN network. In case of the centralized SDN, a TSN flow outside the TSN network can be distinguished and apply for resource reservations to ensure the delay sensitive characteristics. In case of ingress and egress based management, an outside flow that requires TSN properties while traversing through a TSN network can be identified and configured over the entire flow path such that the end-to-end flow integrity is preserved.

TSN Performance for 5G Fronthaul Applications

Fronthaul networks transport the highly delay sensitive In-phase/Quadrature (I/Q) symbol information between the central base band processing units and the remote radio heads. Therefore, typical deployments prefer optical fiber to establish high capacity and low latency links. Although traditional Ethernet can meet the capacity requirements, delay requirements are challenging to achieve with Ethernet networks. However, due to time sensitive properties, TSN Ethernet is being considered as a potential candidate L2 protocol for 5G fronthaul applications as an alternative to the Common Public Radio Interface (CPRI) and eCPRI [130] protocols. The adoption of TSN for existing Ethernet infrastructures could result in significant capital and operating expenditures for new fiber deployments. But, the actual performance of TSN networks for fronthaul applications has not yet been investigated for the various fronthaul splits [98]. The PHY and sub-PHY splits require strict deadlines on the order of sub-microseconds. On the other hand, function splits in the MAC, Radio Resource Control (RRC), and higher cellular protocol layers relax the delay requirements to the order of milliseconds. A comprehensive performance evaluation considering the full range of aspects of fronthaul applications, such as relative performance between Ethernet Passive Optical Networks (EPONs) and TSN Ethernet, packetization, functional split, and fronthaul distances for a Cloud Radio Access

Network (CRAN) system could provide deep insight towards deployment considerations for mobile operator networks. The ULL requirements of a wide range of 5G wireless network applications and services have been extensively documented, see e.g., [30, 36, 61, 78, 93, 107, 110, 116, 144, 145, 147, 157, 250, 254, 257, 262, 269, 274, 281, 291, 301, 303, 308, 345, 348, 350, 352, 367, 371, 387, 390, 425, 431, 436, 453]. Thus, there is an extensive need to research latency reductions for 5G wireless networks. Investigating the combined impacts of the various latency reduction techniques developed in future 5G wireless network studies in conjunction with TSN based fronthaul is an important direction for future research.

TSN Applied to Wide Area Networks

The time-sensitive protocol mechanisms that are applied to micro-environments, such as automotive networks, can also be applied to macro-environments, such as Wide Area Networks (WANs). In most situations, the end-to-end network delay is dominated by the wait time in the queues (buffers) of intermediate forwarding nodes. With the TSN rules applied to nodes, the overall end-to-end delay of a flow over a WAN network can be significantly reduced. However, WAN networks typically handle large numbers of flows and operate at very large capacities, making the TSN flow management very challenging. Despite these challenges, WAN networks should, in principle, be capable of supporting TSN characteristics for specific flows that require strict end-to-end latency bounds, such as remote surgery in health-care applications, where a doctor could operate on a patient across a WAN network. One possible approach to handle the challenging flow management could be through SDN based control. The large geographical WAN area would likely require an SDN control hierarchy consisting of multiple control plane entities, such as, local and root controllers, as well an orchestrator.

2.5.2 5G Networks

Seamless Networks Access

Although 5G is envisioned to support ULL and high data rates in both the wireless air interface and the core networks, the seamless network access across multiple operators and connectivity technologies, such as cable and DSL networks, is still an open issue in terms of inter-networking functions. The inter-networking functions across multiple networks and technology domains must be able to negotiate the same set of services while the devices are operating in the 5G domain.

Network Session Migration

The current network connectivity technology trends, including the 5G technology trends, enumerate several network interfaces that concurrently connect a user device to different networks, such as WiFi, LTE, 3G, and Ethernet. However, the actual network characteristics of each interface change over time. For instance, in cellular communications the transmit power is proportional to the distance from the base stations. Hence due to device mobility, the transmit power varies based on the relative distance between base station and device. While there exists a static way of choosing the network interface based on application requirement [91], a dynamic selection based on the network interface characteristics in real time remains an open research challenge. Additionally, once a session is established over an interface, any changes in the network characteristics that impede the connection quality would negatively impact the end-to-end latency. To maintain low latencies, an active session should be handed over to a different interface without interrupting the session.

2.6 Conclusion

This chapter has comprehensively covered networks supporting ultra-low latency (ULL) applications. Providing ULL support requires specialized network protocol mechanisms that have been standardized for the link layer in the IEEE Time Sensitive Networking (TSN) set of standards. In addition, extensive research studies have begun to investigate in detail the performance characteristics and limitations of these link and network layer ULL mechanisms. Aside from this link and network layer perspective, extensive standardization and research efforts have approached ULL support from the perspective of the common wireless device-to-core network communication chain. In particular, the emerging fifth generation (5G) wireless systems provide extensive support mechanisms for ULL applications.

The survey has revealed numerous gaps and limitations of the existing ULL networking mechanisms that present a wide range of avenues for future research. Aside from addressing the limitations of the individual ULL support mechanisms, there is an urgent need to comprehensively evaluate the cooperation of the various developed ULL mechanisms. Judicious configuration and cooperation of the various ULL mechanisms will likely be critical for providing effective ULL services to the end users. More Specifically, mechanisms for flow control namely the Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS) are the proposed enhancements to the traditional Credit-Based Shaper (CBS) which require further investigations. The next chapter presents the limitations of TAS and proposes enhancements which are showcased in a comprehensive empirical evaluations that compares TAS, the enhanced TAS and ATS.

Chapter 3

PERFORMANCE COMPARISON OF IEEE 802.1 TSN TIME AWARE SHAPER (TAS) AND ASYNCHRONOUS TRAFFIC SHAPER (ATS)

3.1 Introduction

A wide range of communication network applications, including industrial network applications [208, 223, 446] and wireless network applications [223, 253, 255, 265, 353, 357, 363, 430], require ultra-low latency (ULL) on the order of milliseconds or less [44, 224, 311, 364]. Ethernet technology has been widely adopted as link-layer connectivity standard in communication networks as a result of Ethernet's openness and cost effectiveness. However, traditional Ethernet has been designed to provide high link utilization so as to achieve maximum end-to-end throughput for best effort services [215, 326]. While best effort services provide high link utilization, and simple implementation, the end-to-end delay cannot be guaranteed. Hence, Ethernet is not well suited for applications that require deterministic end-to-end delay, such as industrial and automation control, professional audio/video production, and automotive control. The deterministic and low latencies required by these applications resulted in the development of semi-proprietary technologies, such as Time-Triggered Ethernet (TTEthernet), EtherCAT, and FlexRay [283] that limit interoperability and interconnectivity.

To address deterministic latency requirements and ULL requirements, the IEEE 802.1 Time-Sensitive Networking (TSN) Task Group (TG) has defined a suite of standards that extend Ethernet technology. The IEEE 802.1 TSN standardization extends the standard Ethernet services with additional features that provide deterministic guarantees, robustness, as well as integrated diagnostics and management services. These standards outline and define new mechanisms that enable distributed synchronized real-time systems using standard Ethernet technologies, allowing the convergence of high-priority low-latency scheduled traffic (ST) and standard best effort (BE) Ethernet traffic on the same network. In this chapter, we focus on the IEEE 802.1Qbv [22] Enhancements to Scheduled Traffic, i.e., the Time-Aware Shaper (TAS), and the IEEE 802.1Qcr [394] Asynchronous Traffic Shaper (ATS). TAS defines a mechanism for time-driven control and scheduling of data frames, whereas, ATS defines a mechanism for assigning eligibility times to received frames according to the ATS state machine.

3.1.1 Contributions

We comprehensively evaluate the performance of TAS and ATS in terms of mean and maximum packet delays and packet loss ratios. In particular, we make the following contributions:

i) We conduct a rigorous evaluation of the TAS standard mechanisms in OMNet++ simulation environment with respect to a comprehensive set of parameters to assess performance characteristics and to reveal limitations. More specifically, we investigate the impact of the TAS shaper parameters, e.g., the ST to BE traffic gating proportions and traffic loads.

ii) We design and evaluate a novel TAS adaptive bandwidth sharing (ABS) mechanism to enhance link utilization. Moreover, we design and evaluate an adaptive control of the traffic gating proportions, namely the adaptive slotted window (ASW) mechanism, to provide low delays for scheduled traffic even for high network traffic loads. We examine the impact of the combination of these two adaptive mechanisms on ST and BE traffic.

iii) We compare the TAS and ATS shapers to evaluate whether ATS is capable of achieving similar performance as TAS in industrial networks with periodic and sporadic data transfer tasks. Based on the traffic and network models, we infer several characteristics that determine the advantages of using ATS and/or TAS.

3.1.2 Organization

This chapter is organized as follows. Section 3.1.3 contrasts related work from our study. Section 3.2 provides background on the 802.1 TSN standardization. Section 3.3 explains the timing analysis and algorithms for TAS, adaptive TAS, and ATS, including their respective state machines. Section 3.3.1 introduces the novel adaptive bandwidth sharing (ABS) and adaptive slotted window (ASW) mechanisms designed to overcome the limitation of standard TAS. Section 3.4.1 introduces the simulation network model and traffic models. Section 3.4.2 presents the evaluation results for the TSN standard and adaptive TAS, while Section 3.4.3 presents the ATS results. Section 3.5 concludes the chapter.

3.1.3 Related Work

Generally, two main approaches are undertaken when designing real-time Ethernet based packet switched networks: *i*) synchronous time-triggered based medium access control (e.g., TAS), and *ii*) asynchronous event-triggered approach (e.g., ATS).

The TSN standardization, and in particular the realization of the time-driven TAS, involves strict synchronized time requirements [388], similar to Time Triggered Ethernet (TTEthernet) [239], to ensure accurate deterministic service for each traffic flow. Although TTEthernet shares Time Division Multiplexing (TDM) similarities with TAS, TAS operates on predefined traffic classes and not on individual frames as

TTEthernet does. Additionally, TAS, as part of the IEEE 802.1 TSN standardization, can be coupled with several TSN standards, e.g., IEEE 802.1Qci [25] (PSFP), IEEE 802.1Qch [26] (CQF), to add fine-grained control and management to ST flows so as to provide highly precise guaranteed deterministic behaviors.

FTT-Ethernet [346, 347] is a master/multi-slave (MS) protocol that leverages the design principles of switched Ethernet and provides time-triggered scheduling to guarantee timeliness. FTT-Ethernet enforces dynamic quality of service (QoS) and admission control to ensure guaranteed bandwidth and bounded network induced latencies, specializing in Controller Area Network (CAN) field networks. However, the FTT-Ethernet protocol has not gained widespread adoption due to lack of standardizations/certifications and timeliness guarantees. Meyer et al. [297] have presented a network scenario to analyze the impact of time-triggered scheduling on AVB class A traffic in automotive scale networks. Specifically, Meyer et al. [297] analyzed the impact of time-triggered communication on competing traffic on the same infrastructure.

The IEEE TSN TG has published a series of standards that govern stream/flow shapers within 802.1 bridges/switches, and in particular the TAS mechanism. A first analysis of TAS and related TSN shapers has been presented by Thangamuthu et al. [407]. Thangamuthu et al. performed a comprehensive timing analysis on the TSN shapers with emphasis on the end-to-end delays for high-priority Control Data Traffic (CDT). Thangamuthu et al. argued that TAS achieves the lowest latencies and jitter. Similar to Meyer et al.'s automotive topology, Thangamuthu et al. evaluated three proposed traffic shapers, namely TAS, Peristaltic Shaper [26], and Burst Limiting Shaper, on a small automotive topology. In contrast, in this study, we examine industrial control networks with varied periodic and sporadic traffic for synchronous (TAS) and asynchronous (ATS) shapers and introduce novel adaptation mechanisms for TAS.

Thiele et al. [412] have compared TAS against the TSN Peristaltic Shaper [26] and standard 802.1Q Ethernet, i.e., scheduling purely based on the frame priority. Thiele et al. provided significant insight into the operation and timing analysis of switched Ethernet using TAS (and the Peristaltic Shaper), including delay analysis for several blocking effects and delay analysis of TSN and non-TSN streams. Park et al. [340], Farzaneh et al. [162], and Maxim et al. [288] have evaluated the performance of TAS in automotive in-vehicle networks focusing on automotive use-cases. Nsaibi et al. [328] have evaluated TAS in the specific context of a Sercos III network. In contrast, we perform a comprehensive evaluation of TAS in the context of general industrial control networks and compare TAS with the emerging Asynchronous Traffic Shaper (ATS).

For completeness, we note that several studies have examined a range of specific aspects of operating TSN networks. A theoretical worst-case delay analysis of TAS has been conducted in [458]. Dürr et al. [146] have derived an offline scheduler that given periodic time-triggered Ethernet frames can optimally schedule and reduce gate-driver entries with minimized end-to-end delays. Craciunas et al. [121, 122] presented a scheduling method to compute static schedules for TAS using Satisfiability Modulo Theory (SMT) and Optimization Modulo Theory (OMT) solvers. Craciunas et al. identified key functional constraints affecting the behavior of TSN networks which are used to set a generalized configuration of parameters for real-time ST streams. Raagaard et al. [361] have developed a heuristic algorithm that reconfigures TAS switches according to runtime network conditions, while related routing and scheduling schemes have been further studied in [183]. Feasible schedules are produced and forwarded using a configuration agent (composed of a Centralized User Configuration (CUC) and Centralized Network Configuration (CNC)). Raagaard et al's model places emphasis on appearing and disappearing flows in a fog computing platform that takes into account the flows' properties and possible routes.

Nayak et al. [316] have explored how routing impacts the TAS scheduling. Furthermore, Nayak et al. [320] have employed Software Defined Networking (SDN) to incrementally add new TAS flows while preserving the QoS of existing TAS flows. Moreover, several recent studies, e.g. [41–43, 63, 205, 426, 427], have explored time-triggered scheduling for 5G fronthaul networks. Furthermore, in-car Ethernet communications have gained significant traction, specifically for automotive applications, such as multimedia/infotainment and Advanced Driver Assistance Systems [74].

The ATS standard is still in the draft state. At this point, only few studies have examined ATS in the TSN context. Zhou et al. [461] have examined two ATS alternatives, namely *i*) the Urgency Based Scheduler (UBS), and *ii*) Paternoster policing and scheduling. For UBS, two main interleaved algorithms have been presented, in particular, frame-by-frame leaky bucket and token based leaky bucket. In this chapter, we focus mainly on token based ATS since the ATS draft standard follows a token bucket based algorithm. The simulations in Zhou et al. [461] consider only one-hop transmission of sporadic traffic, with emphasis on the comparison of Paternoster and UBS. In contrast, this study considers a more realistic networking scenario with multi-hop transmissions of sporadic and periodic traffic. Emphasizing the need to prevent burstiness cascades in TSN and extending the interleaved shapers/regulators analyzed in [86], Mohammadpour et al. [305] have mathematically analyzed credit based shaper (CBS) and ATS service curves for audio video bridging (AVB) traffic classes as well as bounds on the CBS and ATS response times and traffic backlogs.

3.2 Background: IEEE 802.1 Time Sensitive Networking (TSN)

3.2.1 IEEE 802.1Qbv: Time Aware Shaper (TAS)

TAS operates in a similar manner as TTEthernet [239], i.e., based on time-triggered scheduling. While TTEthernet implements time-based scheduling of individual frames or flows, the TAS schedule is based on predefined traffic classes. Scheduling based on traffic classes, as opposed to individual frames or flows, provides better scalability. Since traffic classes are defined according to the priority code point (PCP) values of the Virtual Local Area Network (VLAN) ID (VID) tag of 802.1Q frames, only applying 802.1Qbv TAS cannot enable fined-grained identification and control on the level of individual streams or flows. Additional mechanisms, such as Per-Stream Filtering and Policing (PSFP) [25] and Frame Replication and Elimination for Reliability (FRER) [27], allowing identification of frames based on the Stream ID and overriding of the traffic class encoded in the PCP code, would be necessary to achieve the same level of per-flow QoS as in TTEthernet [120].

TAS schedules high-priority critical traffic streams in reserved time-triggered windows. In order to prevent lower priority traffic, e.g., best effort (BE) traffic, from interfering with the scheduled traffic (ST) transmissions, ST windows are preceded by a so-called guard band. TAS needs to have all time-triggered windows synchronized, i.e., all bridges from sender to receiver must be synchronized in time, usually through the 802.1AS time reference [22, Sections 8.6.8.4 and 8.6.8.4.10]. TAS utilizes a gate driver mechanism that opens and closes according to a known and agreed upon time schedule for each port in a bridge. In particular, the Gate Control List (GCL) contains Gate Control Entries (GCEs), i.e., a sequence of 1's or 0's that represent whether a queue is eligible to transmit. The frames of a given egress queue are eligible for transmission according to the GCL, which is synchronized in time.

Frames are transmitted according to the GCEs in the GCL and transmission selection decisions. Each individual software queue has its own transmission selection algorithm, e.g., strict priority queuing. Overall, the IEEE 802.1Qbv transmission selection transmits a frame from a given queue with an open gate if: (*i*) The queue contains a frame ready for transmission, (*ii*) higher priority traffic class queues with an open gate do *not* have a frame to transmit, and (*iii*) the frame transmission can be completed before the gate closes for the given queue. Note that these transmission selection conditions ensure that low priority BE traffic is allowed to *start* transmission only if the transmission will *be completed* by the start of the window for high-priority ST traffic. Thus, this transmission selection effectively enforces a "guard band" to prevent low priority traffic from interfering with high-priority traffic.

3.2.2 IEEE 802.1Qcr: Asynchronous Traffic Shaper (ATS)

While TAS performs well in imposing traffic determinism, the stringent timing requirements, in particular the high required precision levels of the timing synchronization across the TSN domain, increase complexity and threaten the reliability of the TSN network domain if any timing misalignment occurs. Furthermore, several synchronization challenges, e.g., skew or drift in timing signal frames, clock inaccuracy, and lost timing frames, can lead to inaccuracies downstream from the synchronized master clock in the TSN domain. As the network scale increases, so does the synchronization complexity.

As an alternative to TAS, the Asynchronous Traffic Shaping (ATS) (IEEE 802.1Qcr [394]) imposes similar traffic determinism without the need for strict timing synchronization. Initially, ATS was proposed based on research by Specht et al. [395] on an Urgency Based Scheduler (UBS) that operates according to two approaches: *i*) Length-Rate Quotient (LRQ) and *ii*) Token Based Emulation (TBE). In this chapter, we focus on the token bucket based approach due to the draft standard's similar approach to ATS.

The token bucket based ATS approach achieves QoS through asynchronously operating sub-shapers. Talkers (transmitters) can decide when to send as long as they comply with the prescribed rate limits. The ATS sub-shapers regulate the traffic at every hop at the granularity of an individual stream (or flow) or an aggregate of multiple streams (or flows). Each switch operating UBS includes a number of shaped queue instances associated with a number of shaper groups that are controlled internally through the use of Internal Priority Values (IPVs) and interleaved regulators, whereby each group is scheduled using an independent local clock. UBS thus effectively implements hierarchical per-hop shaping.

3.3 Timing Mechanisms and Transmission Algorithms

This section provides more detail on the TAS and ATS mechanisms, specifies the main TAS and ATS parameters and state variables, and gives the main TAS and ATS algorithms. Figs. 3.1 and 3.2 illustrate the finite state machine operation, including the main operating states and event transitions, of TAS and ATS, respectively.

3.3.1 TAS

Time Aware Scheduling

TAS considers two main traffic types, namely high-priority scheduled traffic (ST) and low-priority best effort traffic (BE). ST is buffered in the ST queue, and BE is buffered in the BE queue within switches, i.e., TAS implements frame priority isolation by traffic class. TAS divides up the transmission opportunities so as to deterministically satisfy the ST QoS bounds. TAS ensures that the ST delay is bounded and protects ST from any cross-traffic interference. TAS switches shift the

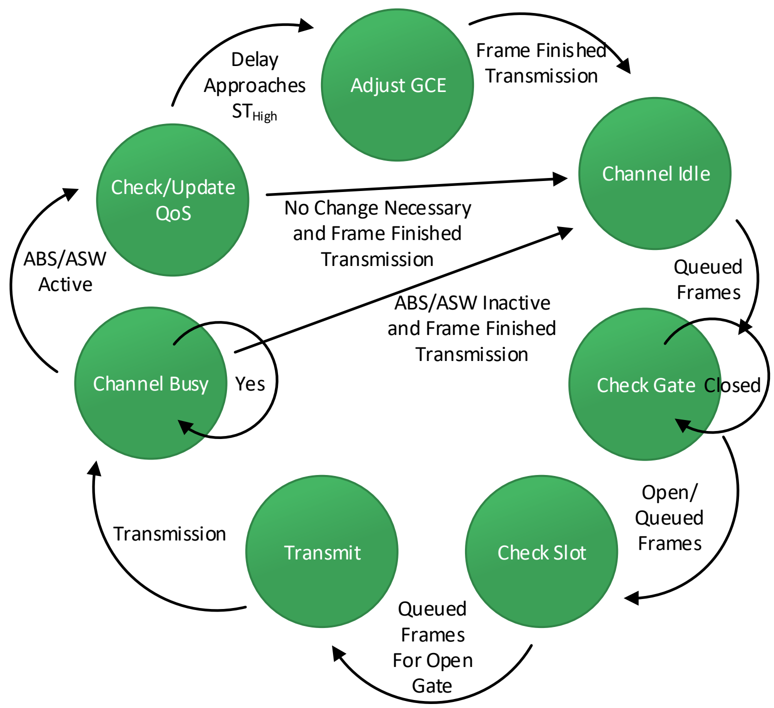


Figure 3.1: Time Aware Shaper (TAS) state machine illustrating the states and transitional operations for both standard and adaptive TAS.

gate status to open/close depending on the cyclic GCL composed of several GCEs. The GCL is programmed to follow a strict TDMA approach that depends on a global synchronized time (in our simulation, that time is *simTime()*).

Fig. 3.3 shows a timing diagram where the guard band (although not explicitly present) is inherent in the TAS implementation. Since the size of each packet is known, the transmission delay can be calculated. If the current ST packet that is awaiting transmission on a idle output channel has a total transmission time less than the time duration until the start of the BE slot time, then TAS will send the ST packet. Otherwise, the ST packet is scheduled for the next GCL cycle. Therefore, if an ST packet arrives at the beginning of a BE slot, then it has to wait at least until the beginning of the next ST slot. While the ST to BE proportions are fixed in Fig. 3.3, we consider dynamically changing the proportion ratio according to the current runtime delay experienced at the receiver in Section 3.3.1.

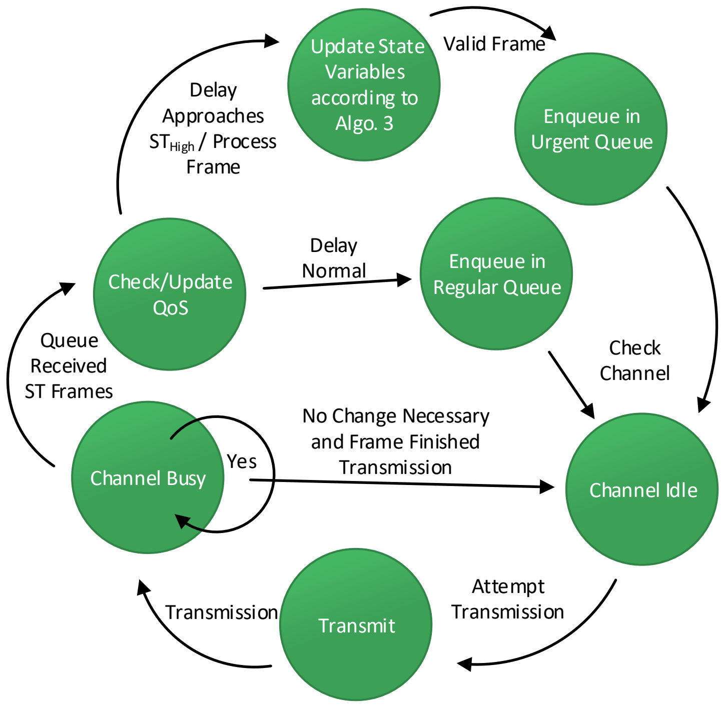


Figure 3.2: Asynchronous Traffic Shaping (ATS) state machine illustrating the states and transitional operations.

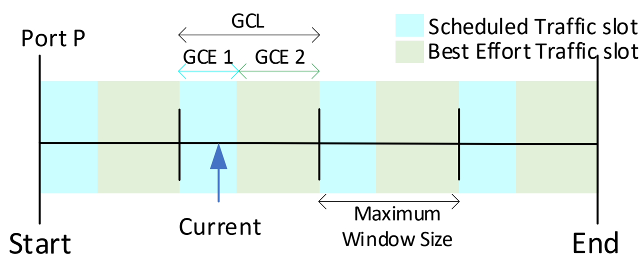


Figure 3.3: Illustration of Time Aware Shaper (TAS) time line: Each GCL cycle has a duration equal to the maximum window time, i.e., the cycle time (CT). Gates are opened and closed according to the GCEs within a give GCL cycle.

Scheduling Algorithm

1. *Maximum Window Time [Cycle Time (CT)]*: The Maximum Window Time, which is also referred to as cycle time (CT), specifies the basic time period for the GCL to repeat. The total gate times (windows) that can be allocated within one CT to the various traffic classes must sum to less than or equal to the CT.

2. *Slotted Windows*: Slotted windows define the timed transmission windows given

Algorithm 1 Time Aware Shaper (TAS) algorithm applied to each switch output (egress) port for two traffic classes, namely ST and BE. The TAS standard specifies fixed static bandwidth allocations and windows. We introduce novel adaptive bandwidth sharing (ABS) and adaptive slotted windows (ASW), see Section 3.3.1.

$Port_P$: Considered Egress Port
ST^R: Given ST Gating Proportion for GCE
CT: Given Cycle Time (maximum window time) for GCL
procedure TAS($Port_P$)
    if $Port_P$ is idle **then**
        $S_i \leftarrow simTime_{current}/CT$
        $ST_{slot} \leftarrow floor(S_i) * CT + ST^R * CT$
        $BE_{slot} \leftarrow ceil(S_i) * CT$
        if $simTime_{current} \leq ST_{slot}$ **then**
            if !isEmpty(ST_Q) **then**
                if $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq ST_{slot}$ **then**
                    send($Packet_{ST}$, $Port_P$)
                else
                    Schedule(ST_{slot})
                end if
            end if
        else
            if !isEmpty(BE_Q) **then**
                if $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq BE_{slot}$ **then**
                    send($Packet_{BE}$, $Port_P$)
                else
                    Schedule(BE_{slot})
                end if
            end if
        end if
    else
        Schedule($simTime_{current} + \frac{Pkt_{length}^{transmitted}}{Channel_{capacity}}$)
    end if
    return
end procedure

to each traffic class on a given switch. Initially, these values are statically predefined. However, with a centralized management method and accurate traffic characterization, it is possible to dynamically define the slotted window times at runtime, i.e., achieve TAS reconfiguration. In our implementation, the Priority Ratio ST^R indicates the proportion of the cycle time that is allocated to the slotted window for the ST class. The BE class is allocated the leftover proportion of the CT.

3. *Gating Mechanism*: The gating mechanism is the primary mechanism that given a time-based signal (timer) blocks or unblocks a queue from transmission. In our implementation, we calculate the current window slot based on the global simulation time in OMNeT++ (Lines 6–8 in Algorithm 1) and according to the calculated slot, check if the current simulation time belongs to the ST or BE slot (Lines 9 and 17), effectively blocking any transmission from consideration depending on the current global synchronized simulation time.

4. *Queue Management*: Queue management determines the removal of frames within queues as well as the internal queue structure and management. For our implementation, each queue (total of two traffic class queues for each egress port) is implemented as a *cPacketQueue* container class using the strict priority structure. Packets are inserted into the back of the ST or BE queue according to the packet traffic class (ST or BE). FIFO is considered within a given queue.

Our initial simulations evaluate TAS without bandwidth sharing or changing the gating ratio at each switch port with OMNeT++ according to Algorithm 1, while TAS with adaptive bandwidth sharing (ABS) and adaptive slotted windows (ASW) are implemented using Algorithm 2. In Algorithm 2, the ST^R is updated whenever an update message is received at a specific egress port on the switch. Our ASW implementation (see Section 3.3.1) follows a distributed approach of propagating an end-point message to allow the ST slot to expand or shrink depending on the perceived runtime delay at the end-point (e.g., sink). The message is propagated in the reverse path of the affected flows.

Motivation and Specification for an Adaptive TAS

Evaluations of the standard TAS (which are presented in detail in Section 3.4) indicated an overall shortcoming of standard TAS in that the gating ratio between ST and BE traffic was a limiter that can increase or decrease the ST delay while inversely impacting the BE delay. Additionally, link utilization had been observed to be very low due to strict slotted windows that when not configured correctly can deteriorate the ST and BE QoS. Therefore, we propose to mitigate such limitations by introducing a controllable adaptive bandwidth sharing (ABS) and a dynamic adaptive slotted window (ASW) mechanism. ABS and ASW are ST centric in the sense that they favor ST traffic over BE traffic following Algorithm 2.

Adaptive Bandwidth Sharing (ABS) Specification: To enable high utilization, we propose adaptive bandwidth sharing (ABS) of the ST and BE slot times to blocked queues if a given transmission opportunity would otherwise go unused. That is, ABS temporarily shares the bandwidth of the current slot time. ABS is specified in Algorithm 2, specifically in Lines 26 and 36.

As shown in Fig. 3.4, the transmission opportunity reserved for BE (due to the BE slot timer) can be temporarily shared with ST traffic. When the timer is set to BE or ST, the non-empty (ST or BE) waiting queue can be selected for transmission if the reserved (BE or ST) queue is empty. While this operation may contradict the TAS operation of preventing cross-interference, the idea behind ABS is to further

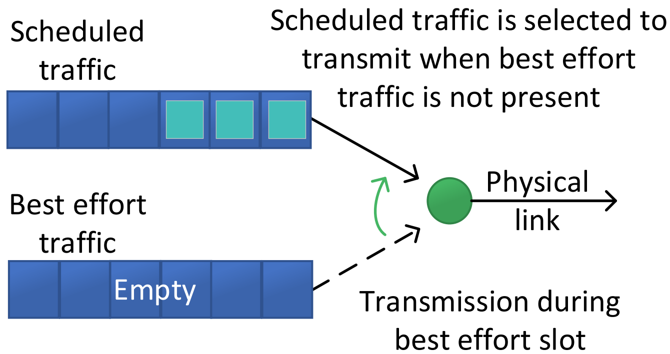


Figure 3.4: Time Aware Shaper (TAS) Adaptive Bandwidth Sharing (ABS) illustration.

reduce the ST delays while keeping BE delays relatively low, effectively mitigating BE traffic starvation.

Adaptive Slotted Windows (ASW) Specification: We propose an Adaptive Slotted Window (ASW) mechanism that shifts the ST to BE gating ratio according to network runtime statistics. This adaptive approach should be able to cope with bursts of ST traffic while temporarily sacrificing the BE QoS. For the ASW modification of the standard TAS mechanism, we feed back receiver network runtime statistics to the upstream switches. The upstream statistics reporting can be implemented over listener (receiving node) to multiple talkers (sending nodes) trees, analogous to the recently proposed listener microstream-interleaving [103] in the context of TSN cyclic queueing and forwarding (CQF) [26] through updates of the traffic specifications.

As illustrated in Fig. 3.5, the upstream switch determines whether it is necessary to expand or reduce the slotted window according to a predefined delay threshold for a particular traffic class. If the current runtime delay approaches the threshold, then we expand the slot for the traffic class in question, and vice versa.

Specifically, we update the gating ratio through a step size of $\Delta = 0.1$. The TSN standardization and recent literature typically set the maximum ST traffic delay bound to $ST^R_{\max} = 0.1$ ms for a maximum of 5 switch hops. For the lower threshold,

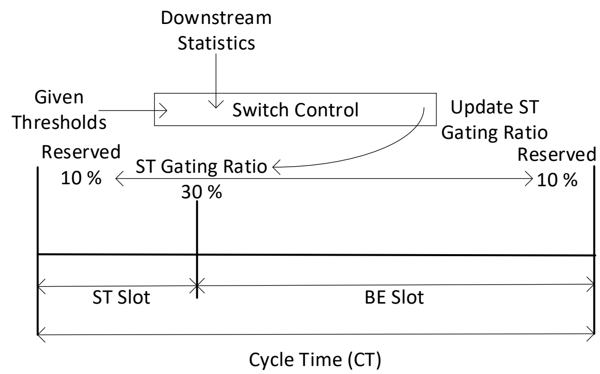


Figure 3.5: Time Aware Shaper (TAS) Adaptive Slotted Windows (ASW) illustration.

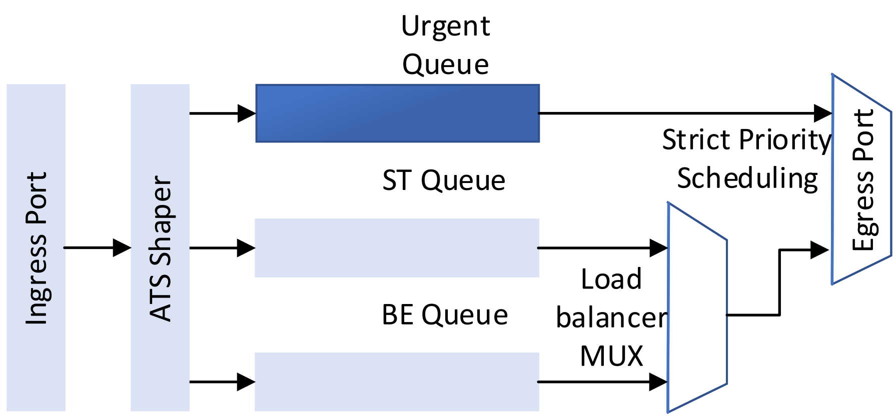


Figure 3.6: Illustration of Asynchronous Traffic Shaper (ATS) bridge operation: The ATS shaper at the ingress determines whether to regulate/shape ST traffic flows utilizing the urgent queue. The traditional ST and BE queues follow a fair multiplexed transmission scheme allocating fair transmission times to both ST and BE.

we select half, i.e., $ST^R_{\min} = 0.05$ ms, to avoid frequent updates of the ST gating ratio. Additionally, we reserve 10% of the CT for each traffic class to avoid severe congestion when the network is highly loaded, as specified in Algorithm 2, specifically in Lines 7–13. Future research could examine the impact of the granularity of the step size Δ.

3.3.2 ATS

ATS shapers assign eligibility times to frames belonging to specific streams which are then used for traffic regulation by the ATS transmission selection algorithm [394,

Algorithm 2 Adaptive Time Aware Shaper (Adaptive TAS) algorithm applied to each output port of the switch for two traffic classes, namely ST and BE. The Adaptive Bandwidth Sharing (ABS) mechanism is specified in Lines 26 and 36. The Adaptive Slotted Window (ASW) mechanism is specified in Lines 7–13.

1: $Port_P$: Considered Egress Port
2: ST_R: ST Gating Ratio initialized to some value
3: ST_{High} Given ST maximum threshold
4: ST_{Low} Given ST minimum threshold
5: Δ Given gating ratio step size
6: CT: Given Cycle Time (maximum window time) for GCL
7: **procedure** CALCULATE $ST^R(ST_R)$
8:     **if** $Delay_{Sink}^{current} \geq ST_{High}$ **AND** $ST^R + \Delta \leq 0.9$ **then**
9:         $ST^R = ST^R + \Delta$
10:     **else if** $Delay_{Sink}^{current} \leq ST_{Low}$ **AND** $ST^R - \Delta \geq 0.1$ **then**
11:         $ST^R = ST^R - \Delta$
12:     **end if**
13: **end procedure**
14: **procedure** TAS($Port_P$)
15:     **if** $Port_P$ is idle **then**
16:         $S_i \leftarrow simTime_{current}/CT$
17:         $ST_{slot} \leftarrow floor(S_i) * CT + ST^R * CT$
18:         $BE_{slot} \leftarrow ceil(S_i) * CT$
19:         **if** $simTime_{current} \leq ST_{slot}$ **then**
20:             **if** !isEmpty(ST_Q) **then**
21:                 **if** $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq ST_{slot}$ **then**
22:                     send($Packet_{ST}$, $Port_P$)
23:                 **else**
24:                     Schedule(ST_{slot})
25:                 **end if**
26:             **else if** !isEm.(BE_Q) **AND** $\frac{Pkt_{len.}^{cur.}}{Ch._{cap.}} \leq ST_{sl.}$ **then**
27:                 send($Packet_{BE}$, $Port_P$)
28:             **end if**
29:         **else**
30:             **if** !isEmpty(BE_Q) **then**
31:                 **if** $\frac{Pkt_{length}^{current}}{Channel_{capacity}} \leq BE_{slot}$ **then**
32:                     send($Packet_{BE}$, $Port_P$)
33:                 **else**
34:                     Schedule(BE_{slot})
35:                 **end if**
36:             **else if** !isEm.(ST_Q) **AND** $\frac{Pkt_{len.}^{cur.}}{Ch._{cap.}} \leq BE_{sl.}$ **then**
37:                 send($Packet_{ST}$, $Port_P$)
38:             **end if**
39:         **end if**
40:     **else**
41:         Schedule($simTime_{current} + \frac{Pkt_{length}^{transmitted}}{Channel_{capacity}}$)
42:     **end if**
43:     **return**
44: **end procedure**

Section 8.6.8.5] without adhering to global network time synchronization. According to the draft standard [394], each bridge provides an ATS Shaper Instance Table with parameters and variables of up to *MaxShaperInstances* independent ATS shaper instances, an ATS Shaper Group Instance Table with parameters and variables of up to *MaxShaperGroupInstances* independent ATS shaper group instances, and an ATS Port Parameter Table with parameters and variables shared by all ATS shaper instances associated with a reception port. To evaluate pure ATS without added complexity or confounding protocol mechanisms, we simulated ATS in isolation without Per-Stream Filtering and Policing (PSFP) [25].

Each ATS shaper instance assigns eligibility times to the associated frames, and discards frames in rare situations. The underlying operations are performed by an ATS shaper state machine [394, Section 8.6.11] associated with an ATS shaper instance which is built following the token bucket algorithm. This state machine updates the associated bucket empty time and group eligibility time state variables based on the $TokenRate_{size}$ parameter, the $TokenBurst_{size}$ parameter, the $Max_{ResidenceTime}$ parameter, the frame arrival times, and the total frame length. If an ATS shaper instance discards a frame, the discard frame counter of the associated port is increased. For bridges with TAS support, and without support for Enhanced Scheduled Traffic (i.e., TAS) and PSFP, the ATS traffic stream gates are permanently open and only used for Internal Priority Value (IPV) assignment, i.e., traditional queue arbitration and management. Fig. 3.6 illustrates the high level overview of the ATS model implemented in OMNeT++. All received frames need to pass the ATS shaper module before being admitted to the queues. The ATS shaper directs some traffic to the urgent queue according to the ATS state variables, eligibility time (of a given considered frame), and the current QoS experienced at the current hop, i.e., per-hop shaping. After shaping is performed, strict-priority scheduling is used to grant channel access to frames at the egress queue. The regular ST and BE queues are multiplexed at the egress to allow fair sharing of the channel, i.e., prevent starvation of BE traffic. All queues are FIFO queues.

Algorithm 3 The Asynchronous Traffic Shaper (ATS) defined in IEEE 802.1Qcr Draft 0.5 [394] computes the ST frame Eligibility Time (ET), assigns the ET to each ST frame, and updates the ATS state machine variables in each ingress port.

$TokenRate_{size}$ Given token information rate
$TokenBurst_{size}$ Given token burst size
$Max_{ResidenceTime}$ Given max residence time
$Bucket_{EmptyToFullDuration} = \frac{TokenBurst_{size}}{TokenRate_{size}}$ Given total bucket recovery duration
procedure ATS_ProcessFrame($Frame_{Received}$)
    $Bucket_{LengthRecoveryDuration} = \frac{Frame_{size}}{TokenRate_{size}}$
    $ATS_{ShaperEligibilityTime} = Bucket_{EmptyTime} + Bucket_{LengthRecoveryDuration}$
    $Bucket_{FullTime} = Bucket_{EmptyTime} + Bucket_{EmptyToFullDuration}$
    $Frame_{EligibilityTime} = MAX(Frame^{ArrivalTime}_{Received}, ATS_{GroupEligibilityTime}, ATS_{ShaperEligibilityTime})$
    if $Frame_{EligibilityTime} \leq Frame^{ArrivalTime}_{Received} + Max_{ResidenceTime}$ **then**
        $ATS_{GroupEligibilityTime} = Frame_{EligibilityTime}$
        if $Frame_{EligibilityTime} < Bucket_{FullTime}$ **then**
            $Bucket_{EmptyTime} = ATS_{ShaperEligibilityTime}$
        else
            $Bucket_{EmptyTime} = ATS_{ShaperEligibilityTime} + Frame_{EligibilityTime} - Bucket_{FullTime}$
        end if
        $Queue_{Urgent} \leftarrow Frame_{Received}$
    else
        Discard_Frame($Frame_{Received}$)
    end if
end procedure

Algorithm 3 specifies the general concept for implementing the ATS shaper state machine in an ingress bridge port. A local clock *ATS_ShaperClock* is used to determine arrival times of frames at a given ingress port. Upon the arrival of an ST frame to the switch ingress port, the current time stamp is tagged to the incoming ST frame. Similarly, the departure times from the egress ports are tagged to the ST frames so that the elapsed times in the various switches on the path to the destination sink can be tracked [394, Section 8.6.11.2.1.]. The accumulated elapsed time is compared against the ST_{high} threshold [394]. If the elapsed time is less than $0.8 \times ST_{high}$, then the frame is enqueued in the ST queue. If the elapsed time is greater than

$0.8 \times ST_{high}$, then the ATS_PROCESSFRAME procedure is invoked for the ST frame. The ATS_PROCESSFRAME procedure returns the assigned eligibility time to the ST frame and queues the ST frame to the urgent queue, or designates the frame for discard. In our evaluations, ST frames that would be discarded by the standard Alg. 3, Line 19, are enqueued in the ST queue. Received BE frames are directly enqueued in the BE queue.

The urgent queue follows a token bucket algorithm to regulate the traffic according to the following state variables used in Alg. 3.

1. $TokenBurst_{size}$ – The maximum token capacity of the token bucket, in bits, for an ATS shaper instance.

2. $TokenRate_{size}$ – The rate at which the token bucket is refilled with tokens until the maximum token capacity $TokenBurst_{size}$ is reached, in bits per second.

3. $Bucket_{EmptyTime}$ – A state variable that contains the most recent time instant when the token bucket of the ATS shaper instance was empty, in seconds. At initialization, the number of tokens in the token bucket is set to the $TokenBurst_{size}$.

4. $Bucket_{EmptyToFullDuration}$ – The time duration required to accumulate a number of tokens equivalent to the $TokenBurst_{size}$, in seconds.

5. $Bucket_{FullTime}$ – The most recent time instant when the number of tokens in the token bucket is equivalent to the $TokenBurst_{size}$, in seconds.

6. $Bucket_{LengthRecoveryDuration}$ – The duration required to accumulate a number of tokens equivalent to the frame length, i.e., the length of the currently considered frame, in seconds.

7. $Frame_{EligibilityTime}$ – The eligibility time (time to send) of the frame, without considering the device-internal forwarding processing delays.

8. $Max_{ResidenceTime}$ – This parameter limits the duration for which frames can reside in a bridge, in nanoseconds.

9. $ATS_{GroupEligibilityTime}$ – A state variable that contains the most recent $Frame$ $_{EligibilityTime}$ from the previous frame, as processed by any ATS shaper instance in the same ATS shaper group, in seconds.

10. $ATS_{ShaperEligibilityTime}$ – The earliest time when there are enough tokens in the bucket to transmit the packet.

The assigned eligibility time (used by the ATS transmission selection algorithm) is calculated by measuring the variation (offset) between the local ATS shaper clock and the ATS transmission selection clock, and the forwarding processing delay within a switch. Our evaluations assume that the processing delay and the time offset between the ATS shaper local clock and the ATS transmission clock are negligible. Therefore, the assigned eligibility time is governed by the token bucket process in the ATS state machine (Lines 13 and 15 in Alg. 3). The assigned *EligibilityTime* calculated by the ATS_ProcessFrame procedure is agnostic to device internal parameters and the link characteristics.

A frame is eligible for transmission if the assigned eligibility time [394, Section 8.6.11.2.2] is less than or equal to the current time. The current time is determined by the Transmission Selection Clock, which is a local system clock. The Transmission Selection Clock determines the selectability time per frame, which is the time at which the arrival frame is queued [394, Section 8.6.6] and available for transmission selection. All frames that reach their selectability time are selected for

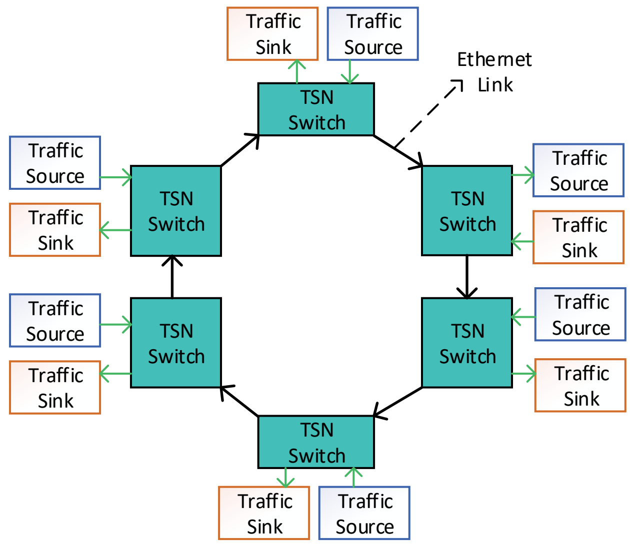


Figure 3.7: Industrial control loop topology [190]: Each source sends data in the clockwise direction traversing a varying number of hops around the ring.

Table 3.1: Traffic ratio scenarios (ST and BE traffic proportions relative to the total traffic load ρ_L) for sporadic (Poisson) traffic model, and corresponding TAS ST gating ratios ST^R, whereby the cycle time is set to $CT = 50$ μs.

	Traffic		**Gate**	
	ST	**BE**	**ST**	**BE**
Scenario 1	20%	80%	20%	80%
Scenario 2	20%	80%	30%	70%

transmission in ascending order of the assigned eligibility times. Frames with identical assigned eligibility times are selected according to the ordering requirement specified in [394, Section 8.6.6].

3.4 Performance Evaluation

3.4.1 System Overview and Simulation Setup

This section explains the simulation setup, i.e., presents the considered industrial network topology and simulation scenarios. Throughout, we employ the OMNeT++ [420] simulation environment.

Network Model

We consider a ring network topology. The ring topology is ubiquitous for industrial networks, which typically require ultra-low delay service. In particular, we consider a ring consisting of six switches, as shown in Fig. 3.7. The switch-to-switch links operate as full-duplex Ethernet links with a capacity (transmission bitrate) of $R = 1$ Gbps. One traffic source and one traffic sink are directly attached to each switch. The distance between two successive switches around the ring is 100 m, corresponding to a propagation delay of 0.5 μs. The switch egress port buffer size is set to 512 Kbyte (KB) for each traffic type, i.e., each switch egress port has a 512 KB buffer for ST traffic and a 512 KB buffer for BE traffic.

Traffic Model

According to the International Electrotechnical Commission (IEC) and the IEEE TSN task group there is wide range of use cases for TSN [73]. These use cases can generate a wide variety of traffic, including isochronous (periodic) traffic, e.g., for control loops, as well as random (sporadic) traffic, e.g., from alarms or event monitors. In order to conduct a comprehensive evaluation, we consider sporadic traffic as well as periodic packet traffic.

We generate sporadic packet traffic according to independent Poisson processes with the same packet generation rate at each traffic source. Each traffic source independently randomly generates data packets of size 580 bytes. For the sporadic traffic model, both scheduled traffic (ST) and best effort traffic (BE) are generated according to independent Poisson processes. We remark that we employ the terminology "scheduled traffic (ST)" to indicate high-priority traffic that is to be transmitted in the high-priority ST traffic slots of the GCL cycles. In our sporadic traffic scenario, the scheduled traffic (ST) is generated at random times (i.e., asynchronously) in the traffic sources (i.e., it is not scheduled in advance); nevertheless, for consistence with the common TSN terminology, we refer to this sporadic high-priority traffic as "scheduled traffic (ST)". In particular, ST and BE traffic are generated as per the traffic proportions in Table 3.1, i.e., 20% of the generated packets are high-priority ST packets, and 80% of the generated packets are low-priority BE packets. The traffic intensity is characterized through the relative traffic load, i.e., the traffic intensity relative to the $R = 1$ Gbps link transmission bitrate. For instance, a load of $\rho_L = 0.5$ corresponds to a total bitrate of 0.5×1 Gbps injected into the network across the six source nodes, whereby each source node uniformly contributes one sixth of the total load.

We also consider a periodic (pre-planned) traffic model, with periodic high-priority ST and sporadic Poisson low-priority BE traffic. Each ST traffic source has a periodic traffic generation module that is synchronized to the cycle time structure of the switches. Thus, each traffic source generates a prescribed number of ST packets and injects them into the network right at the instant when the ST traffic slot starts at the switch that the traffic source is attached to. For periodic ST traffic, we consider 64 byte packets (which are typical for control data traffic (CDT)). Each traffic source injects a prescribed number of π, $\pi = 1, 2, 4$, or 8 CDT traffic packets (of 64 bytes each) at the beginning of each ST slot (i.e., once per cycle). Thus, the periodic traffic contributes a fixed ST bitrate of $\pi \times 64 \times 8$ bit $\times 6$ source nodes $/CT$. The corresponding ST intensity is obtained by dividing the ST bitrate by the link transmission bitrate R. In addition to the ST intensity, the network is loaded with a BE traffic intensity of ρ_L. For clarity, we report the delay performance results separately for different values of the number of periodic ST traffic packets π. That is, we show separate curves for $\pi = 1, 2, 4$, and 8. Each plot shows the performance as a function of the BE (background) traffic intensity ρ_L.

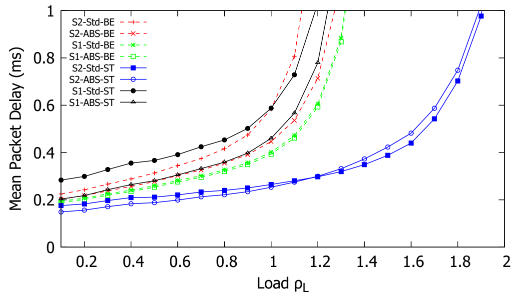
Each packet independently randomly travels a hop distance of one, two, three, four, or five switch-to-switch hops in the clockwise direction around the ring with probabilities 0.1, 0.1, 0.1, 0.3, and 0.4, respectively. Note that for the uniform load of multi-hop packet traffic injected at each source node in conjunction with the traffic routing in one direction along the ring, the ring nodes experience uniform steady-state packet traffic loading. Thus, as the load level is increased, all ring nodes experience the same high steady-state traffic loading level, i.e., all ring nodes become essentially bottlenecks. Therefore, the performance at a bottleneck node with a prescribed high loading level in a different topology, e.g., in a mesh topology, is essentially equivalent to that of a ring bottleneck node with the same high loading level.

We load the network with 20% high-priority ST traffic and 80% low-priority BE traffic as listed in Table 3.1. For a given evaluation scenario, we simulate 100 seconds of network operation, which corresponds to over 30 Million packets for a load of 2.0.

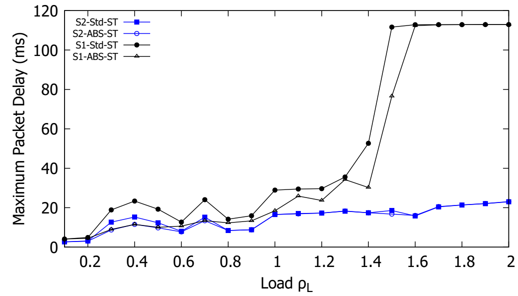
TAS and ATS Settings

In this chapter, we report TAS results for a cycle time $CT = 50\ \mu$s, which is commonly considered for TSN studies. In additional evaluations that are not included, we conducted evaluations for $CT = 100\ \mu$s and found similar results as for $CT = 50\ \mu$s. Two ST to BE gate ratios are considered, see Table 3.1. The scenario 1 (S1) gate ratio matches the ST to BE traffic ratio. Scenario 2 (S2) gives ST traffic a 30% proportion of the gate times while the ST traffic is still only 20% of the total traffic.

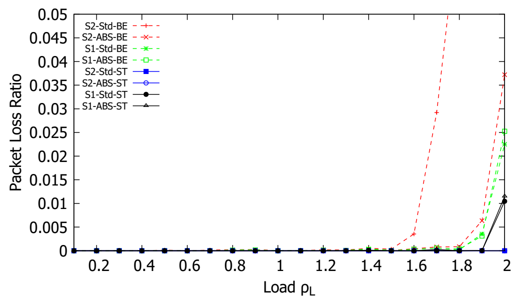
The ATS parameters were set to $ST_{High} = 0.1$ ms and $ST_{Low} = 0.05$ ms, as well as $TokenRate_{size} = 128$ KB/s, $TokenBurst_{size} = 512$ KB, and $Max_{ResidenceTime} = 20\ \mu$s based on extensive empirical trials that sought to achieve low ST packet delays, while providing a reasonable overall packet traffic service.



(a) Mean packet delay



(b) Maximum packet delay



(c) Packet loss ratio

Figure 3.8: TAS with Adaptive Bandwidth Sharing (ABS) compared to standard TAS for sporadic ST sources.

3.4.2 TAS Evaluation

Sporadic ST Sources

We first compare standard TAS with the proposed adaptive TAS.

Adaptive Bandwidth Sharing (ABS): Fig 3.8(a) shows the average end-to-end packet delay for both standard TAS (Std) and adaptive TAS with ABS. We observe that for scenario 1 (S1) with equal ST/BE traffic and gating ratios, the standard TAS approach gives slightly higher delays for ST than for BE. This is mainly due to the relatively small ST windows for the considered ST/BE ratio of 20/80. ABS with its dynamic window utilization mitigates the effects of that small ST windows and reduces the mean ST delays to close to the BE delays. For the more typical

TAS operating scenario 2 (S2) with a slightly higher ST/BE gating ratio than the ST/BE traffic ratio, we observe from Fig 3.8(a) that the ST mean packets delays are significantly lower than the BE delays. We also observe that ABS leaves the ST delay unchanged, while significantly reducing the BE delays.

Fig. 3.8(b) shows the maximum ST packet delays. We observe that for S2, the maximum packet delays are significantly lower than for S1. Nevertheless, these maximum ST packet delays are significantly higher than the typical ST delay targets on the order of a millisecond. This result indicates that sporadic (random) traffic can experience worst-case delays of ten or more milliseconds with standard TAS and TAS with ABS.

Fig. 3.8(c) shows the total packet loss ratios. We observe that ST and BE traffic experience no loss at low to moderate loads. We observed from Fig. 3.8(a) that in the S1 scenarios, the mean ST delays were higher than the corresponding mean BE delays. Now, we observe from Fig. 3.8(c) that for the S1 scenarios, the ST packet losses are lower than the BE traffic losses. ST traffic has smaller losses since ST traffic has the same buffer space (512 KB) available as BE traffic, but ST traffic has a four times smaller traffic volume than BE traffic (see Table 3.1). We also observe that for S2, the ST packet loss is consistently zero (for both standard TAS and for TAS with ABS); this is due to the overprovisioning of the gating ratio in favor of ST. Moreover, we observe for S2 that ABS reduces the BE packet loss compared to standard TAS.

Adaptive Slotted Windows (ASW): Fig. 3.9 shows the end-to-end average ST and BE packet delays for TAS with ASW compared to the standard TAS. We observe for S1 with the initially equal ST/BE traffic ratios and gating ratios that ASW achieves substantial ST delay reductions compared to standard TAS; whereby the delay reductions are most pronounced at high loads. We also observe that the S1-BE

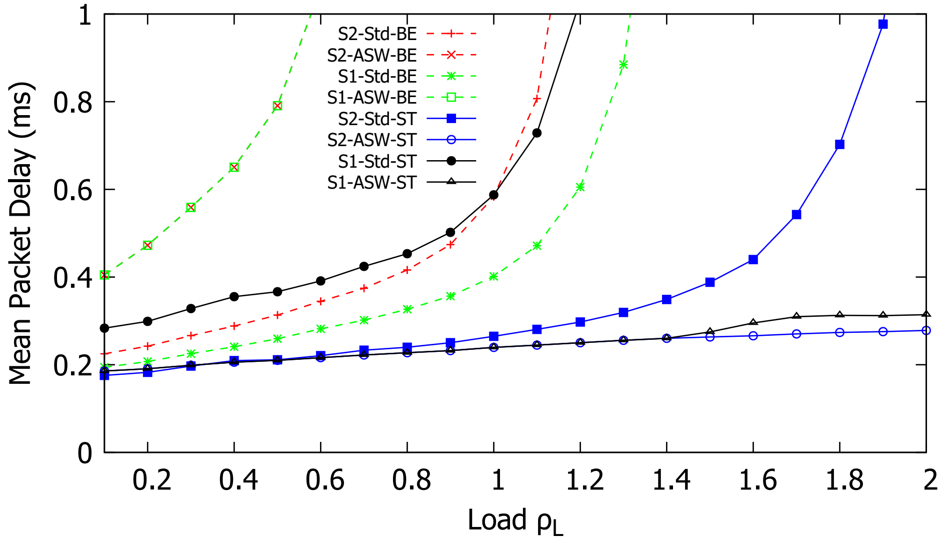


Figure 3.9: Mean packet delay for TAS with Adaptive Slotted Windows (ASW) compared to standard TAS for sporadic ST sources. (The curves for S1-ASW-BE and S2-ASW-BE are overlapping.)

delays are correspondingly increased by ASW. These delay results indicate that ASW effectively expands the ST window to consistently ensure low ST delays, even when the initial ST/BE gating ratio setting does not favor ST traffic. In particular, we observe from Fig. 3.9 that Scenarios 1 and 2, which differ in the initial gating ratio settings give essentially equivalent ASW delays. These equivalent delays are due to the continuous gating ratio updates of the proposed ASW mechanism, i.e., with ASW, the packet delays are over the long run independent of the initial ST/BE gating ratio since the ASW mechanism dynamically adapts the ST/BE gating ratio.

ABS & ASW Combined: Fig. 3.10 shows the mean ST and BE packet delays for TAS with the combined ABS and ASW in comparison to the standard TAS. We observe that similar to the TAS with ASW-only delays in Fig. 3.9, the combined ABS+ASW achieves substantial ST delay reductions (0.1 ms constant delay for S1 and S2 cases) compared to the standard TAS. Further comparisons of Figs. 3.9 and 3.10

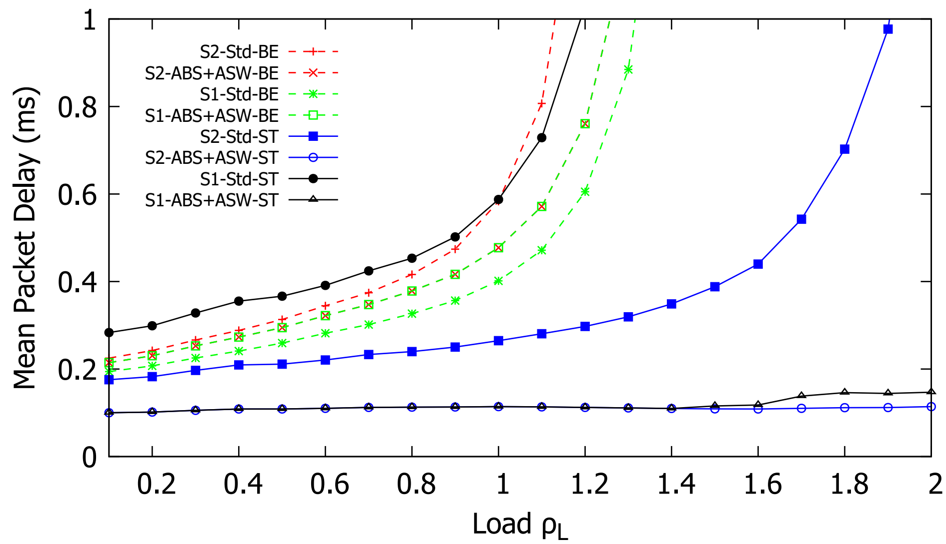


Figure 3.10: Mean packet delay for TAS with combined ASW and ABS compared to standard TAS for sporadic ST sources.(The S1-ABS+ASW-BE and S2-ABS+ASW-BE curves are overlapping.)

indicate that the combined ABS+ASW reduces the mean delays from slightly above 0.2 ms for ASW only to around 0.1 ms for ABS+ASW. Moreover, we observe from the comparison of Figs. 3.9 and 3.10 that the combined ABS+ASW substantially reduces the BE delays (from 0.4 ms to 0.2 ms at low loads). For instance, for S1 with a load of $\rho_L = 0.4$, ASW-only gives a mean BE packet delay of approximately 0.65 ms in Fig. 3.9; whereas the combined ABS+ASW gives a corresponding mean BE packet delay of 0.27 ms in Fig. 3.10. Thus, the ABS+ASW combination can extract substantial additional delay reductions for both ST and BE packets through the dynamic ABS sharing across the ST and BE gating windows on top of the underlying ASW dynamic adaptation of the gating ratios.

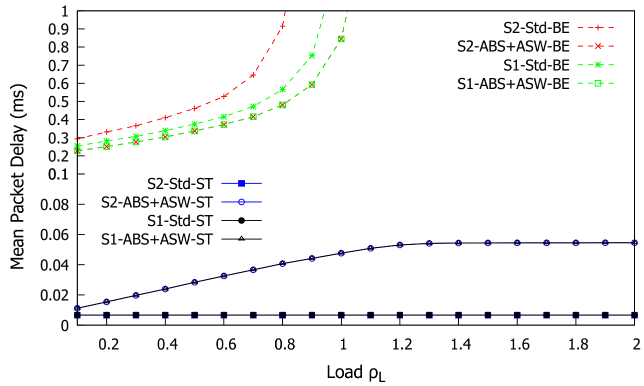
In additional evaluations, we found that the S2 maximum ST frame delays for the combined ABS & ASW are below 4 ms for all load levels. Thus, the combined ABS & ASW achieves substantial reductions from the maximum ST frame delays of up to around 20 ms for standard TAS and TAS with ABS in Fig. 3.8. An interesting future work direction is to add frame preemption [21] to TAS with the combined ABS & ASW in order to further reduce the maximum (worst-case) ST packet delays.

Periodic ST Sources

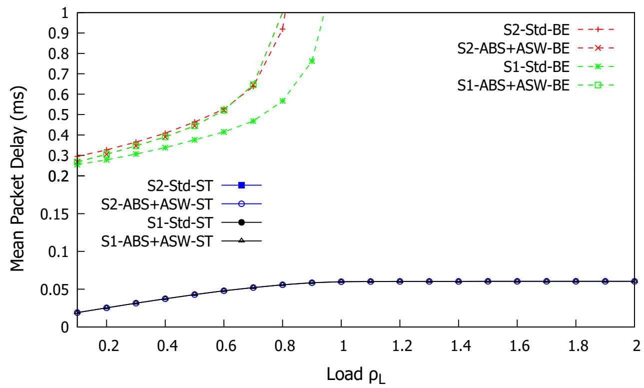
Similar to the evaluation for sporadic (Poisson) ST traffic sources, we have compared the proposed adaptive TAS mechanisms to standard TAS for periodic ST traffic sources, as specified in Section 3.4.1. We have considered the periodic ST traffic injection rates $\pi = 1, 2, 4$, and 8 ST packets per CT. We present only plots for $\pi = 4$ and 8; the plots for $\pi = 1$ and 2 are very similar to the plots for $\pi = 4$. Also, for brevity, we only present results for TAS with combined ABS & ASW.

Fig. 3.11 shows the average end-to-end ST and BE packet delays for $\pi = 4$ and 8 ST packets per CT. We observe from Fig. 3.11(a) that for $\pi = 4$, standard TAS consistently achieves very low mean ST delays below 0.01 ms (for both S1 and S2) for the entire range of BE traffic loads. In contrast, the mean ST delays for TAS with ABS & ASW (both for S1 and S2) increase nearly linearly with increasing BE traffic load until a load of around $\rho_L = 1.2$ and then flatten out around 0.06 ms. Turning to Fig. 3.11(b) for $\pi = 8$, we observe that the delays for TAS with ABS & ASW increase at a slightly steeper slope but flatten out at around the same level as for $\pi = 4$. On the other hand, the delays of standard TAS for $\pi = 8$ are 81 ms and 54 ms for S1 and S2 for the entire range of BE traffic loads. These results indicate that TAS with ABS & ASW can provide robust low-delay service to ST traffic, even at relatively high loads of periodic ST traffic. Standard TAS with fixed parameter settings would require manual intervention to adjust to such high ST traffic loads. TAS with ABS & ASW automatically adjusts to high ST traffic loads and reduces the BE gate allocations so as to keep prioritizing ST traffic.

The mean packet delay results in Fig. 3.11 confirm that the ABS+ASW approach



(a) $\pi = 4$ ST packets/CT



(b) $\pi = 8$ ST packets/CT; the standard TAS ST delays are above the plotted range.

Figure 3.11: Mean packet delay for TAS with combined ASW and ABS compared to standard TAS for periodic ST sources that inject π ST packets per CT. (The S1-ABS+ASW-BE and S2-ABS+ASW-BE curves are overlapping.)

is independent of the initial setting of the gating ratio (S1 or S2 from Table 3.1 were considered). This result is expected since ASW reactively adapts the gating ratio; thus, the initial gating ratio setting becomes irrelevant. We proceed to consider ABS+ASW with only one initial setting in subsequent evaluations.

We also evaluated the maximum packet delays for $\pi = 4$ and 8 ST packets per CT. We found that the maximum ST packet delays were below 0.25 ms for $\pi = 4$. For $\pi = 8$ packets/CT, standard TAS S1 and S2 maximum delays were 108 ms and 71 ms, respectively, for all BE traffic loads ρ_L. In contrast, we observed that TAS

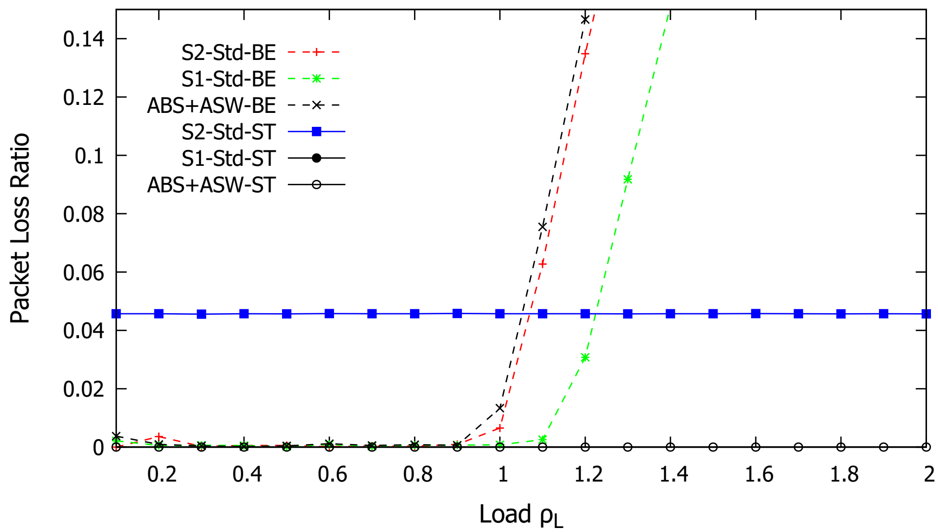


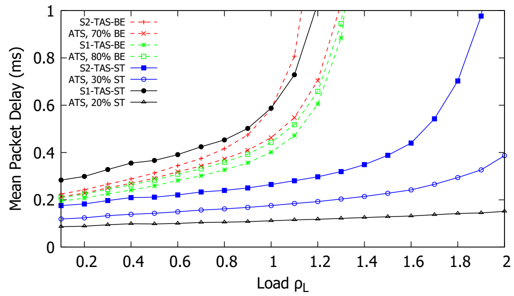
Figure 3.12: Packet loss ratio for TAS with combined ASW and ABS compared to standard TAS for periodic ST sources injecting $\pi = 8$ packet/CT. For standard TAS, the ST packet loss for S1 is consistently at 0.37 (37%).

with combined ABS & ASW consistently achieved maximum packet delays on the order of 0.2 ms.

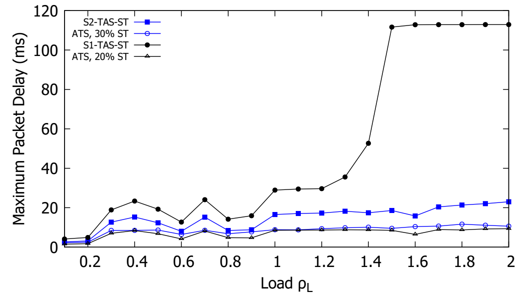
Fig. 3.12 shows the packet losses for $\pi = 8$ ST packets/CT as a function of the BE load. We observe that TAS with ABS+AWS achieves zero ST packet losses throughout. In contrast, standard TAS gives substantial ST packet losses, even for very low BE loads. The combined ABS+ASW drops BE traffic at approximately the same rate as standard TAS in S2.

Summary of Packet Delay Variation Results

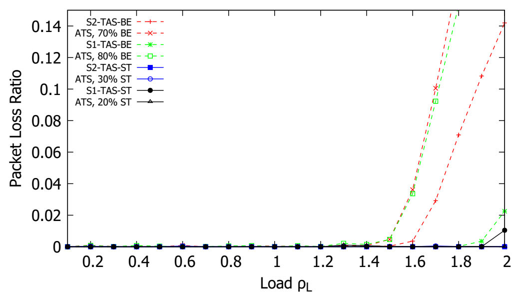
While the presented performance evaluation has focused on mean and maximum packet delays and losses, we have also evaluated packet delay variations (jitter). Generally, TAS strives for very short packet delays, accordingly, packet delay variations are expected to be small. In summary, we found for sporadic traffic that the ST delay variations (represented by the standard deviation of the packet delays) were on



(a) Mean packet delay



(b) Maximum packet delay



(c) Packet loss ratio

Figure 3.13: ATS compared with standard TAS for sporadic ST sources.

the order of 0.1 ms or less with adaptive TAS, while BE packets experienced delay variations up to 10 ms at high loads. Similarly, for the periodic traffic scenario, the ST packets had significantly smaller delay variations (on the order of 0.1 ms of less) than the BE packets (1 ms or higher for moderate to high loads. Overall, we also found that adaptive TAS gave lower packet delay variations than standard TAS.

3.4.3 ATS Evaluation

For the evaluation of ATS (which does not have the concept of gating that TAS has), we consider the traffic proportions 20% of ST with 80% BE, as well as 30% ST with 70% BE.

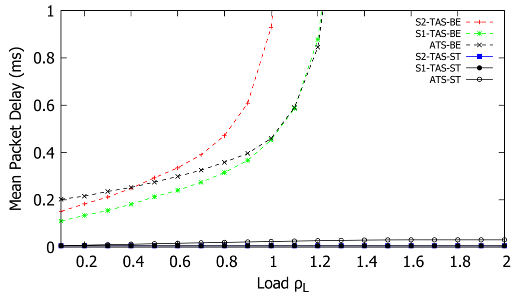
Sporadic ST Sources

Fig. 3.13 shows the mean and maximum ST and BE end-to-end delays as well as the packet loss ratio. Generally, we observe from Fig. 3.13 that ATS performs significantly better than standard TAS for sporadic traffic sources. In particular, we observe from Fig. 3.13(a) that for 20% ST traffic, ATS gives substantially lower ST delays than standard TAS; whereby the delay reduction with ATS is particularly pronounced compared to TAS S1.

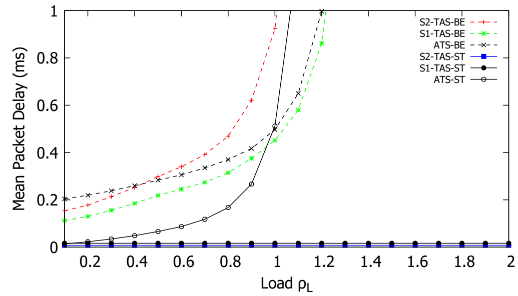
We observe from Fig. 3.13(b) that ATS provides the same short maximum packet delays for both 20% and 30% ST traffic. In contrast, standard TAS gives relatively short maximum ST packet delays for S2, while the maximum ST packet delays for S1 shoot up to around 100 ms for moderately high loads. With ATS, each switch ingress queues the ST packets in the urgent queue if the runtime delay is close to the threshold (whereby we set the "close" parameter to 0.8 times the ST threshold). Within the urgent queue, the ATS algorithm follows the leaky bucket policy, ensuring consistent packet service.

We observe from Fig. 3.13(c) that ATS achieves nearly zero ST packet losses; the ATS ST losses are lower than the S1 TAS losses at very high loads. We also observe that ATS suffers from higher BE packet losses than TAS at high loads. This increased ATS BE packet loss is mainly due to increased delays at the BE queue and correspondingly higher probabilities of packet losses at the BE queue.

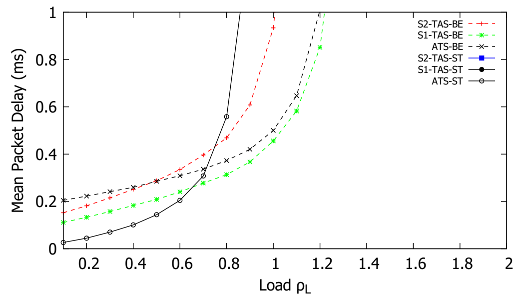
While the ATS simulation produced better results than TAS, the difficultly was mainly in setting and adjusting the configuration parameters for the ATS state variables such that the QoS for ST were guaranteed and the BE traffic was not starved. Shifting fixed static configuration management to dynamic variable configuration is needed to further enhance the granularity of ATS with respect to the number of



(a) $\pi = 2$ ST packets/CT



(b) $\pi = 4$ ST packets/CT



(c) $\pi = 8$ ST packets/CT; S1 and S2 TAS

ST delays are 71 ms and 53 ms, constant

Figure 3.14: Mean packet delay for ATS compared with standard TAS for periodic ST sources.

flows and queue management schemes. Additionally, resource allocation and dropping rogue flows are needed to stop floods of large frames into the TSN domain to allow timely flows to proceed within the contract agreement.

Periodic ST Sources

Following the TAS evaluation for periodic ST sources, we similarly evaluate ATS for periodic ST sources. The ST traffic injection rates are set to $\pi = 2,\ 4$, and 8 ST packets per 50 μs CT. Fig. 3.14 shows the mean end-to-end packet delays for both ST and BE packets. We observe from Fig. 3.14(a) that for the low injection rate $\pi = 2$ ST packets/CT, the ST packet delays were minuscule for both ATS and TAS. We

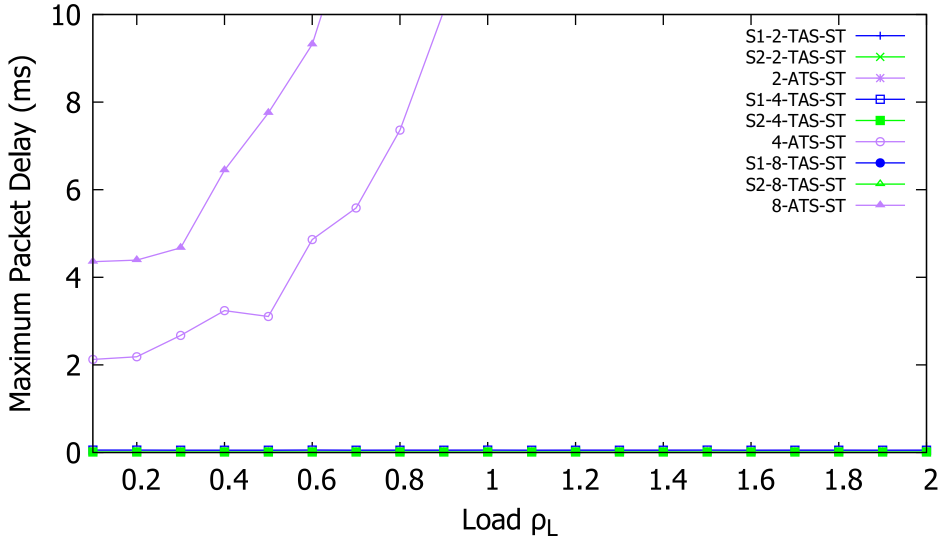


Figure 3.15: Maximum packet delay for ATS compared to standard TAS for periodic ST sources. The S1 and S2 TAS ST packet delays for $\pi = 8$ are 108 ms and 71 ms.

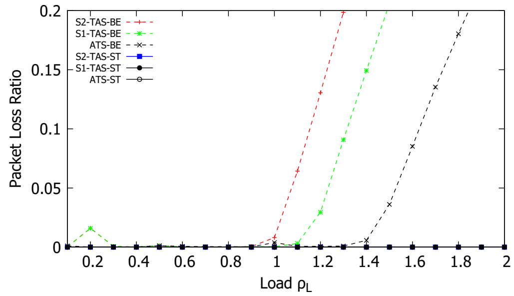
observe from Fig. 3.14(b) that for the higher $\pi = 4$ ST packets/CT injection rate, TAS continues to ensure very low ST packet delays across the entire load range. In contrast, we observe that the ATS ST packet delay increases exponentially with the load; ATS provides mean ST packet delays below 1 ms only for loads below $\rho_L = 1$. With ATS, the resource allocation is fixed and the ATS shapers and urgent queues become saturated when the ST injection rate is $\pi = 4$. We further observe from Fig. 3.14(c) that for the high ST injection rate $\pi = 8$, ATS provides sub-millisecond mean packet delays up to a load around $\rho_L = 0.8$; whereas, TAS gives delays above 50 ms consistently for all load levels. Intuitively, the inherently asynchronous ATS struggles with moderately high to high ($\pi = 4$ and 8) periodic (synchronous) ST packet traffic as the asynchronous ATS prioritization mechanisms do not work in lock-step with the traffic sources. Thus, the ATS delay performance degrades gradually as the ST and BE traffic loads increase. On the other hand, the inherently synchronous TAS can either consistently provide sub-millisecond ST packet delays ($\pi = 2$ and

4), or completely fails ($\pi = 8$), even for low loads of competing BE traffic. This abrupt failure of TAS is due to the prescribed gating ratio that is synchronized to the underlying cycle time; the ST traffic either fits into the ST portion of the cycle (or not) and thus either meets real-time requirements (or not). In contrast, the asynchronous ATS degrades gradually as the competing BE traffic load increases.

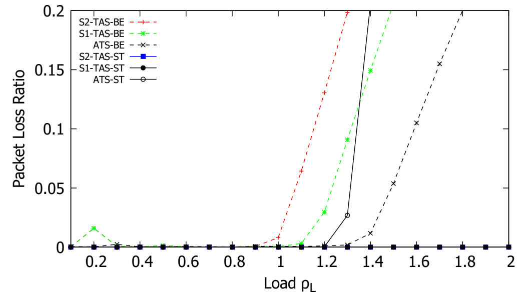
Fig. 3.15 shows the maximum packet delay experienced within the network. The results generally mirror the mean delay results in Fig. 3.14 in that for $\pi = 2$, all schemes give minuscule maximum packet delays well below one millisecond. For $\pi = 4$, the maximum ATS delay increases with the BE traffic load, reaching 10 ms for a BE traffic load around $\rho_L = 0.9$, while TAS continues for provide minuscule maximum delays. For $\pi = 8$, TAS gives very high delays on the order of 100 ms, while ATS gives around 10 ms maximum delay for a load of $\rho = 0.6$. Essentially, these results are again due to the ATS state machine becoming gradually overwhelmed as the load increases, whereas TAS either fits the ST traffic into the ST gate window or not.

Fig. 3.16 shows the packet loss ratios for both BE and ST traffic with ATS compared to standard TAS. We observe that similar to the mean delay behaviors in Fig. 3.14, (*i*) the ATS and TAS ST packet loss ratios are zero for the low ST traffic rate $\pi = 2$, (*ii*) for the moderate ST traffic rate $\pi = 4$, ATS starts to drop ST packets at a moderately high BE traffic load while TAS still achieves consistently zero losses, and (*iii*) for the high $\pi = 8$ ST traffic rate, S1 TAS gives 0.37 (37%) loss while S2 TAS gives a loss ratio close to 0.05 consistently for all BE load levels; in contrast, ATS gives zero ST losses for low BE loads and then increasing ST losses for moderately high BE loads. We also observe from Fig. 3.16 that ATS gives lower BE packet losses than TAS for all considered scenarios.

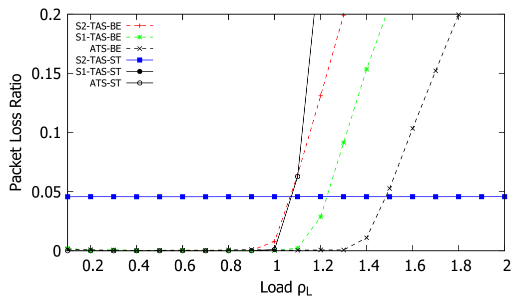
The explanation for these loss ratio behaviors is similar to the explanation of the



(a) $\pi = 2$ ST packets/CT



(b) $\pi = 4$ ST packets/CT



(c) $\pi = 8$ ST packets/CT

Figure 3.16: Packet loss ratio for ATS compared with standard TAS.

mean delay behaviors. Essentially, the asynchronously operating ATS gradually degrades with increasing BE traffic load when the synchronous ST traffic is moderately high to high. In other words, each ATS switch operates individually in complete isolation from the other switches and the sources In contrast, all the TAS switches are synchronized to the common cycle time that is the underlying time basis for the period ST packet transmissions by the sources.

Summary of Packet Delay Variation Results

Compared to the packet delay variations of TAS (see Section 3.4.2), ATS gave generally higher packet delay variations. With ATS, sporadic ST packets experienced delay variations (standard deviations of packet delays) ranging from 0.1 ms for low loads to over 0.6 ms for moderate to high loads. Periodic ST packets experienced delay variations ranging from 0.4 ms to 50 ms for ST packet injection rates of $\pi = 2$ and 4 with moderate to high BE loads.

3.5 Conclusions and Future Work

This chapter has examined the Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS) mechanisms of the 802.1 Time Sensitive Networking (TSN) standard. In order to address the TAS shortcomings, we have proposed two novel mechanisms, namely Adaptive Bandwidth Sharing (ABS) and an Adaptive Slot Window (ASW) mechanisms. We have evaluated the TSN standard mechanisms and the novel mechanisms through extensive simulations with both sporadic Poisson traffic and periodic traffic. We found that standard TAS generally achieves ultra-short latencies if the gating ratio for high-priority traffic is sufficiently large to accommodate the high-priority traffic volume. We also observed that the introduced ABS mechanism enhances the quality of service provided to low-priority best effort traffic, while maintaining the ultra-short latencies for high-priority traffic. The introduced ASW mechanism dynamically adjusts the gating ratio for high-priority traffic so as to ensure ultra-low latencies for high-priority traffic, even for fluctuating background traffic loads of low-priority traffic.

We also found that the asynchronous ATS performs generally well compared to TAS for sporadic (asynchronous traffic). However, for periodic ST traffic with moderately high rates, ATS gives increasing ST packet delays for increasing loads of competing BE traffic. In contrast, TAS with time synchronization to the underlying period of the ST traffic sources either provides very short (order of milliseconds or less) delays irrespective of the BE traffic load, or has consistently high delays (on the order of 100 ms) when the configured gate ratio is too small for the ST traffic.

There are numerous other opportunities for future TSN research that we proceed to outline. Combining the Adaptive Bandwidth Sharing (ABS) and the Adaptive Slotted Window (ASW) TAS enhancements, i.e., Adaptive TAS, with frame/packet preemption will likely achieve further delay reductions. The delay reductions will be particularly relevant for low priority inversion blockage scenarios, i.e., to reduce the delay of low priority traffic that would be blocked due to slot allocations to ST traffic. Additionally, modifying the ST gating ratio through updates ranging from course-grained increments/decrements to fine-grained values relative to network conditions may achieve further performance enhancements. The present study examined static networking scenarios in that a given evaluation run considered a fixed prescribed traffic load. Future research may consider transient TSN network scenarios with varying traffic loads, e.g., scenarios that connect new and remove old periodic and sporadic ST sources during runtime.

We believe that another important future research direction will be the integration of the TSN network control with the emerging universal SDN control of communication networks [76, 118, 226, 456]. It will be critical to define standardized interfaces that facilitate SDN control down to the TSN TAS gating level. The gate operation should still be tied to the time synchronization (which can run independently of the SDN control). However, the specific actions and quantities of slot durations that follow upon time synchronization points should be under SDN control. For instance, SDN control should be able to obtain the utilization level of the various slots, and then be able to adjust gating ratios.

Chapter 4

RECONFIGURATION ALGORITHMS FOR HIGH PRECISION COMMUNICATIONS IN TIME SENSITIVE NETWORKS

4.1 Introduction

IEEE 802.1 Time Sensitive Networking (TSN) provides a standardized framework of tools for providing deterministic Ultra-Low Latency (ULL), e.g., for industrial control applications, automotive networking, smart grid applications, and avionics communication systems [75, 141, 170, 182, 285, 313, 423]. In particular, the IEEE 802.1Qbv Time Aware Shaper (TAS) has received extensive attention as a key tool for achieving a deterministic ULL network service. The TAS operation requires careful planning of the synchronized time cycles [380, 408] and the gate times that are allocated to the Scheduled Traffic (ST) and the unscheduled Best-Effort traffic (BE). The TAS parameter settings specifying the timing characteristics (cycle time, gate slot allocations) are also commonly referred to as the Qbv schedule or the TAS schedule. For a given static networking scenario, the TAS operation with a properly configured Qbv schedule can ensure the deterministic ULL required by demanding industrial and automotive applications [59, 156, 202, 294, 312, 360, 399].

Modern network scenarios often involve dynamic changes with varied use cases, such as changes in the network nodes and network topology, or the traffic pattern. For instance, nodes or links may be dynamically added or removed. Or, nodes may inject additional traffic flows or traffic flows may terminate, or the latency requirements of flows may change dynamically. Such dynamic changes have been included in the use cases defined by the IEC/IEEE 802.1 TSN TG [73, 424]. In a typical industrial environment, sensors that periodically or sometimes sporadically send ambient measurements to a local gateway require certain Quality of Service (QoS) guarantees [33, 111, 175, 206, 314]. In such a volatile and dynamic environment, new machinery that requires prioritized execution (e.g., emergency cooling procedures or maintenance tasks for network traffic tests) may be brought onto the factory floor. To deal with such scenarios, the Time-Aware Shaper (TAS) Gate Control Lists (GCLs) in coordination with the Network Management Entities (NMEs), e.g., Centralized Network Configuration (CNC), have to adapt to changing environment conditions by judiciously applying reconfiguration such that stream deadlines and QoS are satisfied.

Generally, in such dynamic networking scenarios, applying only admission control will clearly guarantee (in accordance with a traffic shaper) the QoS metrics of the admitted flows. However, for a given static network configuration, the total number of admissible streams may be well below the number of streams that seek network service. Therefore, adding a dynamic reconfiguration strategy to manage and configure the network appears to be a plausible and attractive solution that intuitively should lower capital and operational expenditures as it mitigates the over-provisioning of network resources. The general idea for such an allocation scheme is to control network access in a timely and orderly fashion such that a maximum number of streams can be effectively serviced.

Our objective therefore is to maximize the number of admitted flows (i.e., tasks or streams) in such a dynamically changing and volatile environment while keeping the TSN QoS metric guarantees. In this paper, we focus on the IEEE 802.1Qbv [22] enhancements and design a reconfiguration framework taking inspiration from the IEEE 802.1Qcc [24] standards for managing, configuring, and reconfiguring a TSN network.

In IEEE 802.1Qbv, a TAS time slot (corresponding to a GCE and also referred to as slot time) is defined as the portion of the cycle time (CT, which corresponds to the GCL); TAS time slots are allocated to high-priority ST traffic. In our model, the switch/controller computes the TAS time slot for all admitted streams as follows. Essentially, as streams get registered, we keep track of the available remaining capacity, which we set initially to the maximum available capacity on each egress port until the load (which depends on the ST slot size and the cycle time) is negative, i.e., oversubscribed link. Such a link over-subscription invokes a procedure call that increases the slot time (by a step size of 1%, or more fine-grained increments) until the remaining load is positive. This procedure is iteratively called until all registered streams and the new stream are appropriately registered with a sufficient ST slot time to transmit all frames during a single appropriately sized CT.

Our proposed TAS configuration/reconfiguration is designed for the centralized (hybrid) model and for the fully-distributed configuration model. In the "hybrid" model, the CNC is utilized for configuration exchanges and network side management, as explained in more detail in Section 4.3. In the distributed approach, the GCE slot parameters are configured in a distributed manner by the switches as per the distributed algorithm/procedure explained in Section 4.4. For brevity we refer to the centralized network/distributed user model (hybrid model) also as the centralized model or the centralized topology. We refer to the fully-distributed (decentralized) model also as the decentralized model or the decentralized topology.

4.1.1 Related Work

We first note that general performance evaluation strategies for TAS have been explored in [201, 245, 332] and we follow these strategies in our study. Raagaard et al. [356, 361] have presented a heuristic scheduling algorithm that reconfigures TAS switches according to runtime network conditions. Feasible schedules are computed and forwarded using a configuration agent (composed of a Centralized User Configuration (CUC) and Centralized Network Configuration (CNC)). Raagaard et al's model places emphasis on the schedule computation complexity for appearing and disappearing synthetic flows in a fog computing platform. Complementary to this approach, we develop comprehensive centralized and distributed reconfiguration frameworks based on firm bandwidth computation strategies that execute at run-time. Further, we conduct a comprehensive performance evaluation of our two frameworks considering common packet flow QoS metrics for both high-priority ST and low-priority BE traffic.

The proposed approach by Nayak et al. [319] exploits the logical centralization paradigm of SDN with real-time traffic to achieve optimal scheduling and routing. Integer Linear Programming (ILP) formulations were used to solve the combined problem of routing and scheduling time triggered traffic. Two main proposals for routing are given, namely *i*) scheduling and path-sets routing, and *ii*) scheduling and fixed-path routing whereby the ILP formulations are used to find near optimal flow to time-slot allocations. However, the ILP does not scale well with the number of flows, does not provide schedules at runtime speeds, and does not work well with dynamic flow configuration (or reconfiguration). To enhance the architecture proposed by Nayak et al. [319], an augmentation is proposed in [320] that incrementally adds time sensitive flows to the scheduler making the proposed approach reconfiguration capable. Additionally, Nayak et al. [316, 317] provide an analysis and evaluation to the problem of flow-span and routing protocol (Equal Cost Multi-Path, and Shorted Path) on transmission scheduling. Further routing refinements have been studied in [69, 243, 244, 317, 331].

Focusing on in-vehicular networks, Hackel et al. [196] have proposed a SDN based TSN framework that performs reconfiguration using the Stream Reservation Protocol

(SRP) as a means to register and allocate resources for TSN streams. The TSN with SDN is evaluated with two TSN switches and two clients (a sources and sink). In contrast, we provide extensive evaluation for larger network topologies and sources. Using OpenFlow and openPowerLink, Herlich et al. [203] have provided a proof-of-concept model that highlights the advantages of TSN with SDN and real-time Ethernet protocol. While the model shows promising advantages in theory, only a coarse-grained evaluation was presented that, in contrast to our evaluation, does not examine stream admission rates and TSN QoS. Focusing on remote monitoring and telemetry, Kobzan et al. [238] have presented a solution concept and implementation of an SDN based TSN architecture using IEEE 802.1Qcc. However, the concept is provided without any empirical evaluation. To the best of our knowledge, there are no prior detailed studies on a fluctuating volatile source or a dynamic stream resource allocation and admission control policy in conjunction with a network reconfiguration policy being executed while flows are carried in a TAS time scheduled network. We provide a comprehensive design and evaluation of an SDN based TSN model that bases the specification on the standardization given by the IEEE 802.1Q standard.

Vlk et al. [422] have proposed a simple hardware enhancement of a switch along with a relaxed scheduling constraint that increases schedulability and throughput of the time-triggered traffic but maintains the deterministic nature and timeliness guarantees in a TSN network. Several related scheduling refinements that are orthogonal to the reconfiguration studied in this chapter have been examined in [155, 202, 217, 259, 333, 450]. We note for completeness that multicast for TSN has been studied in [381, 449], while our focus is on unicast traffic. This chapter provides the full operational details as well as comprehensive performance evaluations.

4.1.2 Contributions

We comprehensively evaluate the performance of TAS for reconfigurations in the hybrid and fully distributed models with respect to network deployment parameters, such as the time period for the Gate Control List (GCL) to repeat (whereby the duration of one GCL repetition corresponds to the CT), the gating ratio proportion, i.e., Gate Control Entry (GCE) proportion, to control the delay perceived at the receiving end, the signaling impact on ST and BE classes, and the packet loss rate experienced at the receiving end. In particular, we make the following contributions:

i) We design a CNC interface for a TSN network to globally manage and configure TSN streams, including admission control and resource reservation.

ii) We integrate the CNC in the control plane with TAS in the data plane to centrally manage and shape traffic using the CNC as the central processing entity for flow schedules as more flows are added.

iii) We modify and test the model to operate in a distributed fashion, i.e., the signaling is conducted in-band and the control plane processing is conducted at the individual distributed switches.

iv) We evaluate each design approach for a range of numbers of streams with different TAS parameters. We show results for admission ratios, network signaling overhead, and QoS metrics.

4.1.3 Organization

This chapter is organized as follows. Section 4.2 provides background information and an overview of related work on the 802.1 TSN standardization, focusing on the enhancements to ST as well as centralized management and configuration.

Section 4.3 shows the complete top-down design of the CNC (hybrid model) and the main components that achieve ultra-low latencies and guaranteed QoS for a multitude of ST streams. Similarly, Section 4.4 shows the approach used in implementing the decentralized (fully distributed) TAS reconfiguration model. The simulation setup as well as main parameters and assumptions are given in Section 4.5 and results are presented in Section 4.5.2 and Section 4.5.3. Finally conclusions and future work are outlined in Section 4.6.

4.2 Background: IEEE 802.1 Time Sensitive Networking

4.2.1 IEEE 802.1Qbv: Time Aware Shaper (TAS)

TAS's main operation is to schedule critical traffic streams in reserved time-triggered windows. In order to prevent lower priority traffic, e.g., BE traffic, from interfering with the ST transmissions, ST windows are preceded by a so-called guard band. TAS is applicable for Ultra-Low Latency (ULL) requirements but needs to have all time-triggered windows synchronized, i.e., all bridges from sender to receiver must be synchronized in time [380, 408]. TAS utilizes a gate driver mechanism that opens/closes according to a known and agreed upon time schedule for each port in a bridge. In particular, the Gate Control List (GCL) represents Gate Control Entries (GCEs), i.e., a sequence of on and off time periods that represent whether a queue is eligible to transmit or not.

The frames of a given egress queue are eligible for transmission according to the GCL, which is synchronized in time through the 802.1AS time reference. Frames are transmitted according to the GCL/GCE and transmission selection decisions. Each individual software queue has its own transmission selection algorithm, e.g., strict priority queuing. Whereby, a software queue is the queue before the NIC hardware queue takes ownership of the currently forwarded frame in an 802.1 switch. Overall, the IEEE 802.1Qbv transmission selection transmits a frame from a given queue with an open gate if: (*i*) The queue contains a frame ready for transmission, (*ii*) higher priority traffic class queues with an open gate do *not* have a frame to transmit, and (*iii*) the frame transmission can be completed before the gate closes for the given queue. Note that these transmission selection conditions ensure that low-priority traffic is allowed to *start* transmission only if the transmission will *be completed* by the start of the ST window for high-priority traffic. Thus, this transmission selection effectively enforces a "guard band" to prevent low-priority traffic from interfering with high-priority traffic [170].

4.2.2 IEEE 802.1Qcc: Centralized Management and Configuration

IEEE 802.1Qcc [24] provides a set of tools to globally manage and control the network. In particular, IEEE 802.1Qcc enhances the existing Stream Reservation Protocol (SRP) with a User Network Interface (UNI) which is supplemented by a Centralized Network Configuration (CNC) node. The UNI provides a common method of requesting layer 2 services. Furthermore, the CNC interacts with the switch UNI to provide a centralized means for performing resource reservation, scheduling, and other types of configuration via a remote management protocol, such as NETCONF [151] or RESTCONF [83]; hence, 802.1Qcc is compatible with the IETF YANG/NETCONF data modeling language.

The IEEE 802.1Qcc standard specifies three models for configuring the Time-Aware Shaper (TAS) gating schedules (GCL/GCE timing): a fully-centralized model, a centralized network/distributed user model (hybrid model), and a fully-distributed configuration model. The centralized model greatly eases control and configuration messages sent across the network and can precisely configure TAS schedules due to having the complete knowledge of the network and the full capabilities of each bridge. However the centralized model suffers from common disadvantages, such as a single-point of failure, relatively large capital/operational (CapEx/OpEx) expenditures (as the centralized control may be superfluous in a small-scale network [102]), and adding unnecessary complexity to a small-scale network.

Compared to the centralized network/distributed user model (hybrid model), the fully centralized model does not add any benefits for the reconfiguration approach towards enhancing the resource allocation and QoS nor does it allow better deterministic forwarding. The main usage for the CUC is to take into account the application's complex timing and computation requirements for industrial applications which is out of scope for our evaluation. Rather, our focus is on the reconfiguration for proper resource allocation. Therefore, we focus on the centralized network/distributed user model (hybrid model) form of the centralized model in this study.

A fully-distributed configuration model (e.g., SRP over MRP or RAP over LRP) may be attractive for some networks. The fully-distributed configuration model avoids the added complexity and single point of failure of a centralized management entity. Moreover, Chen et al. [102] have argued that the centralized configuration models can be an over-design for real-time applications with relaxed latency requirements (order of magnitude of milliseconds). Chen et al. have also argued that the distributed model is more scalable. (However, studies of the fully distributed model with RAP over LRP targeted typically applications with relatively relaxed latency requirements.)

In the absence of a Centralized Network Configuration (CNC) node, the TSN Task Group (TG) specifies the IEEE 802.1CS (Link-Local Registration Protocol, LRP) [168] standard for registration and distribution of application configuration parameters over point-to-point links targeting newly published TSN features. A legacy protocol, such as the Stream Reservation Protocol (SRP) [12] which is primarily used for Audio-Video Bridging (AVB) applications, is intended to serve as the main resource reservation and admission control protocol. However, extending and porting the SRP to be utilized for bridges that support TAS will not suffice since bandwidth reservation cannot directly apply TAS's time slot reservation natively. Therefore, the Resource Allocation Protocol, IEEE 802.1Qdd (RAP) [102], has been proposed to apply a distributed resource reservation that can exchange TSN features.

4.3 Hybrid Model Design and Framework Considerations

This section presents our design methodology and main signaling framework for the centralized network/distributed user model (hybrid model). Our main goals behind designing the CNC are given by the following constraints. Additionally, the CNC can be logically or physically connected to the data plane with in-band or out-of-band management links. With in-band communication under the hybrid model, only one switch is physically connected to the CNC; thus, signaling packets between the switches and CNC affect data traffic similar to the distributed approach, but the CNC still functions as the centralized configuration. For the hybrid model evaluations in this study, we consider out-of-band communication, i.e., all switches are physically attached to the CNC.

1. Our focus is mainly on stream based network adaptation. By this technique, fluctuating streams (already registered streams and new incoming streams) and their requirements can be accommodated by the network dynamically based on a single remote procedure call to the CNC.

2. We identify and execute flow requirements by populating the registration table. The control plane resource orchestration is purely carried out by monitoring existing flows which have been satisfied.

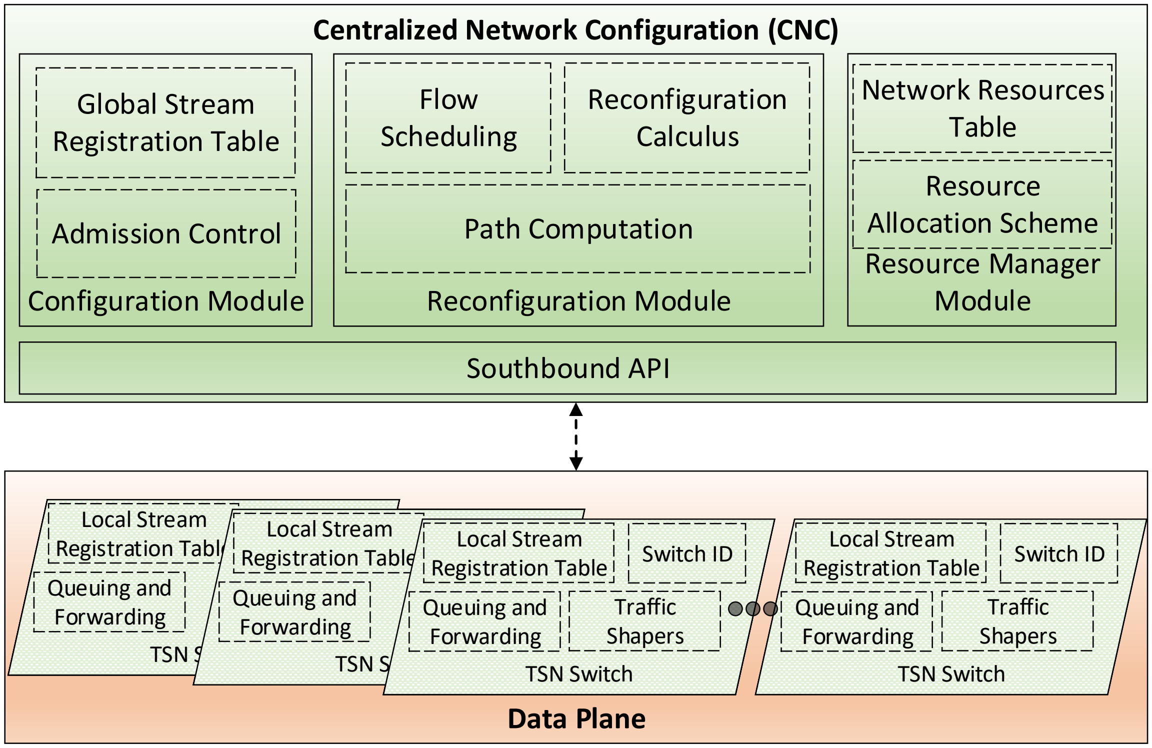


Figure 4.1: Network Management Entity Framework for TSN Switches: Centralized Network Configuration (CNC) is used to send and receive Control Data Traffic (CDT), which we define as the signaling traffic, e.g., the UNI information to and from the CNC and switches/sources or LLDP discovery packets, to configure routing segments and network resources.

3. We conduct resource allocation based on the stream network resource utilization.

Our main assumption to accurately apply admission control and, consequently, reconfiguration, is that each source must define a flow in terms of total resources needed (governed by the bandwidth requirements) and the total time needed for the resource to be used (which in our traffic model is termed the resource utilization time). Essentially, the CNC uses this information (which is tagged in the Ethernet frame header) to determine whether a stream (flow) is admitted or rejected.

4.3.1 Core Components

Our design is split into two layers, Control Plane and Data Plane, following the decoupling SDN paradigm, thereby inheriting the benefits of the orthogonality of the two planes, as shown in Fig. 4.1.

Configuration Module

The configuration module is the main component that interacts with the registered flows and network components. It includes the global stream registration table which records all approved streams transmitting in the network (i.e., currently utilizing network resources (bandwidth)), and the admission control element that encapsulates and decapsulates CDT headers and forwards the information to the necessary module/element.

Global Stream Registration Table: The source streams (devices/users) make a Remote Procedural Call (RPC) via the stream registration interface for providing information that can be mapped as a unique tuple structure identification $< FlowID, BridgeGateway >$. Upon receiving the registration packet, i.e., Control Data Traffic (CDT), the CNC determines whether the new stream can be accepted in its stream table. To guarantee the QoS for all registered streams, admission control principles are applied to all streams according to the stream's path, required network resources, and available resources. Fig. 4.2 shows an example where the source sends a CDT stream request to the gateway switch, which is then forwarded to the CNC for admission control and resource reservation.

Admission Control: The admission control element is the first element that the new streams interacts with. The admission control element in the configuration mod-

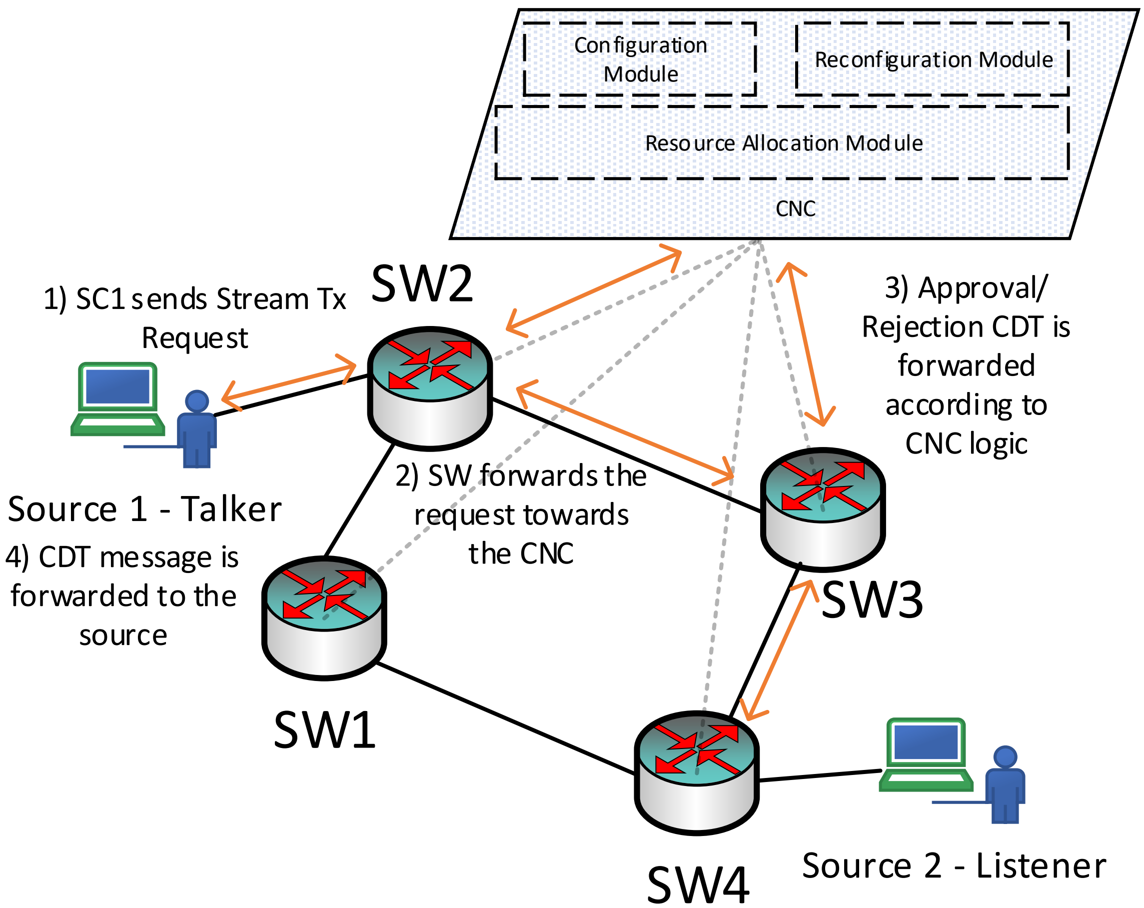


Figure 4.2: Centralized Model Example: Source 1 sends a CDT stream request to its gateway. The gateway forwards it to its governing CNC. The CNC decides if the stream will be serviced according to the source UNI. All switches in the explicit path for the stream are notified if the stream is accepted. Otherwise, the gateway is alerted of the rejection. Lastly, the gateway forwards it back to the source which prompts the source to start sending data ST traffic in the next available cycle (if approved).

ule globally manages all streams transmitting in the TSN domain governed by the CNC. The admission control element extracts the necessary information from the CDT packet and forwards the information according to the CDT type. The CNC applies several steps to decide whether to accept or reject the stream transmission request.

1. The CNC checks the destination address(es) of the stream and consults its

resource manager module for network resources available on the new stream's path, which is computed based on the path computation element within the CNC.

2. According to the bandwidth required for the new stream (calculated at the bridge gateway for the new stream as the stream packet rate multiplied by the packet size and divided by the ST slot time), all links on the path are checked to see if enough bandwidth is available for the new stream.

3. In the event that not enough resources are available, the CNC applies the TAS reconfiguration module to identify the bottleneck link(s) and to check whether the gating ratio can be increased for that specific traffic class whose current resource utilization will not exhaust the resources by being added to the TAS slot reservation.

Reconfiguration Module

The reconfiguration module includes the flow scheduling element (for our network model, the Time-Aware Shaper (TAS) is used in the data plane), the reconfiguration calculus element which optimizes flow registration according to each stream's total resource utilization and flow deadlines, and finally the path computation element which defines the path for all streams according to the QoS constraint.

Flow Scheduling: The flow scheduling element currently takes the Time-Aware Shaper into consideration. Due to the TAS's requirements on time synchronization between network components (switches, hosts, etc.), the CNC follows the same principle of scheduling flows according to a known timescale (initially set to be 50 μs in our network model). The CNC then passes on this time synchronization information to the TSN enabled switches within its domain. Any approved streams will transmit frames according to the time scale specified by the flow scheduler in the CNC.

Reconfiguration Calculus: In addition to centrally managing resources and providing admission control policies to the network, the CNC can invoke the TAS reconfiguration strategy with the goal of borrowing BE time slots for pending ST traffic streams. This element consults the resource manager module on the bottleneck link and checks whether the added stream will oversubscribe the link. The TAS reconfiguration incrementally (1% of total CT) increases the traffic class slot time and reserves it for the new stream.

Path Computation: For large scale and complex LAN/MAN topologies, it is often required to supplement streams with equal cost paths in the event of a path disruption (e.g., link failure, stream saturation, and explicit congestion). The CNC's path computation element is tasked with finding such paths as a fail-over approach to avoid any violations to any stream's QoS. Presently, our model has a rudimentary application of path computation, i.e., it is defined statically for all core network components (shortest path), since our main emphasis in this study is on reconfiguration based on stream characteristics as defined by the source.

Resource Manager Module

The resource manager module centrally manages all network resources within the CNC's domain. It includes the network resource table that records all streams' usage of resources, and the resource allocation scheme element to which we delegate the task of calculating the required network resources for a given stream according to an allocation scheme.

Network Resource Table: To remove certain overheads on the configuration module, the network resource table operates in tandem with the global stream registration table to accurately determine the required network resources (mainly bandwidth for our traffic model). It classifies streams based on periodic stream properties. Any stream that has been approved by the CNC has a record attached to it in the network resource table.

Resource Allocation Scheme: Several allocation schemes can be implemented for all traffic classes defined in the network. For periodic streams, the time slot given by the flow scheduler (according to the TAS Cycle Time and number of traffic classes) and the data rate defined by the source is used to calculate the required bandwidth for each link on the path to the destination (i.e., sink).

Data Plane

The data plane contains all core switches. Any TSN switch interfaced by the CNC is given a switch ID and has a local stream registration table. The remaining switch elements compose the forwarding and queuing operation with several traffic shapers (802.1Qbv TAS in our network model).

Local Stream Registration Table: This data plane registry contains the subset of source streams that are established for the corresponding bridge gateway and attached sources to each port. The CNC delegates some control to the bridge gateway to instruct and alert sources of any new network conditions and explicit changes.

Traffic Shaper — Time-Aware Shaper (TAS): The TAS is the main shaping and scheduling mechanism that controls the gating schedules for all the traffic classes within the TSN domain (which is considered to be equivalent to the CNC domain).

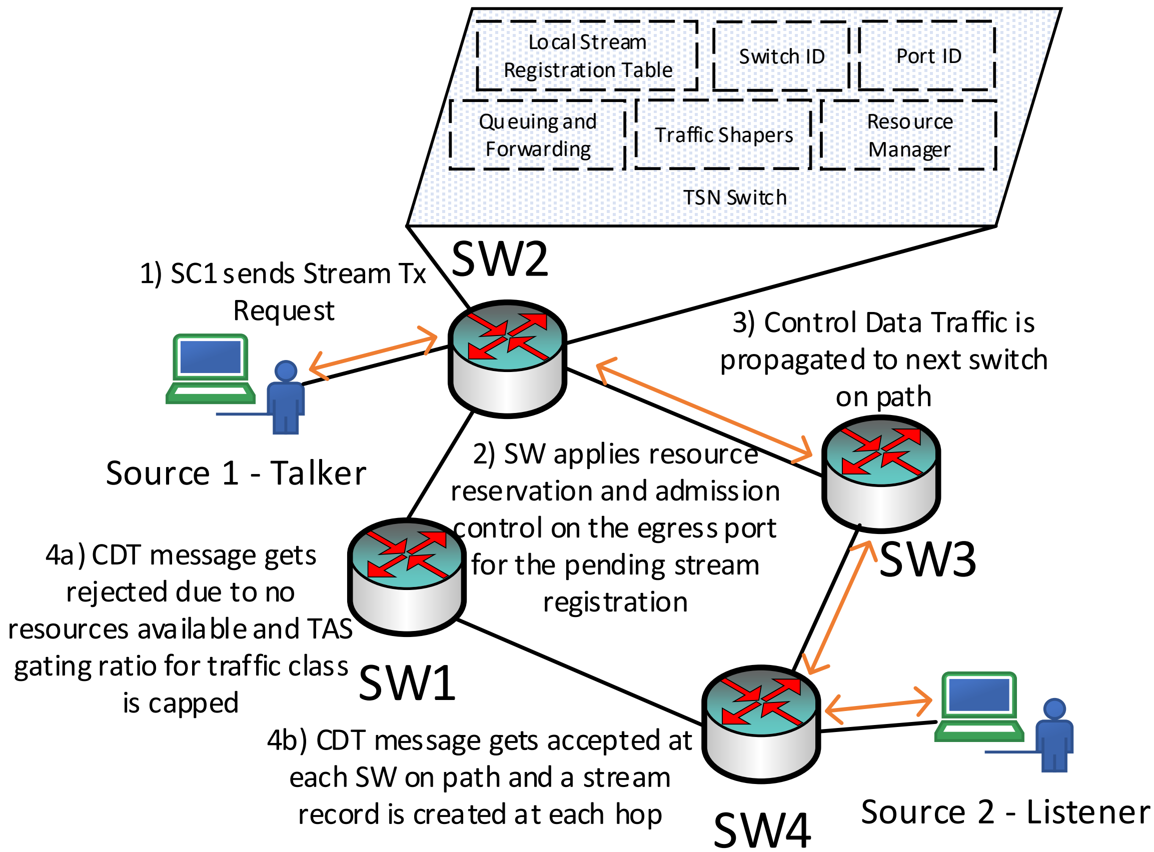


Figure 4.3: A TSN fully distributed configuration model example illustrating the general strategy and logic of each TSN switch with TAS support. In the absence of a CNC to centrally manage network parameters, each switch performs admission control and resource reservation (according to the TAS time slot load) and propagates the information to the next hop on the stream path. A single rejection on one hop terminates the forwarding of the CDT, and sends another CDT on the reverse path indicating the stream rejection outcome. If all switches on the path accept the stream, then the source is notified of the stream acceptance outcome and can begin forwarding in the next TAS cycle. In our model, CDT traffic has higher priority than non-CDT traffic (including ST). The formal definition of the CDT traffic is left for future work.

All bridges are synchronized to the same gating schedule GCL Cycle Time (CT) given by the CNC's flow schedule element (CT indicates the time period for the GCL to repeat).

4.4 Decentralized Model Design and Framework Considerations

This section presents our design methodology and framework for the TAS reconfiguration in the decentralized (fully distributed) model. Our current proposed architecture generally follows the steps enumerated below and illustrated in Fig. 4.3. Our description focuses on the additions to the design of RAP over LRP, e.g., TAS slot computation/reservations.

1. At each egress port (Port Identifier, PID), the TSN switch maintains a local stream registration table that includes information, such as flow ID, gateway (i.e., the first bridge that a talker is connected to), destination address(es), the traffic injection rate per GCL cycle time, and the calculated port bandwidth requirement. The traffic injection rate is not computed, rather the traffic injection rate is reported by the source (talker) to the network devices. It indicates the bandwidth requirements of a stream. Bandwidth for a bridge egress port needed for a stream is computed using the ST injection rate (or ST rate), the average packet size, and the bridge TAS timing configuration (e.g., the CT and current traffic class slot time). This information is carried and communicated between bridges using the CDT packet type identifier (or message type).

2. A source (talker) can send a stream transmission request, i.e., a CDT message of type Stream Transmission Request, to register its stream and to use the TSN service for ST.

3. Each switch maintains a resource manager module for each port. If the newly incoming stream is accepted (due to available resources and TAS slot space). The TAS slot size for a specific traffic class is governed by the CT and traffic class gating ratio (in time). The TAS ST slot can be configured/reconfigured

according to stream requests and terminations. The stream registration message is then propagated towards the next switch, and a map is maintained for the stream (and any other streams) pending approval.

4. If accepted by the last switch, then the stream registration record is added to the local stream registration table, and bandwidth resources are allocated for the stream and TAS slot space is modified (if necessary) on the reverse path. The main reason for allocating the resources in the reverse path is as follows. If we allocate the resources in the forward direction but a switch in the next hop rejects the stream (due to lack of resources), then we have to release the resources reserved earlier for the stream. Therefore, we avoid the allocation until all hops provide assurance that the stream will be accommodated.

5. Each switch receiving the pending registration message adds the stream record to its local table, allocates the necessary resources and TAS slot reservation, and propagates the registration message towards the source gateway.

6. The source gateway receives the pending stream registration message and similarly allocates the resources and finally sends an approval granted message towards the source, which prompts the source to start sending data in the next available TAS cycle.

4.4.1 Core Components

This section outlines the main components required to successfully implement stream admission control and resource reservation within switches that support the TAS traffic shaper in a distributed fashion. Fig. 4.4 illustrates the typical registration/reservation procedure for all streams within the TSN domain.

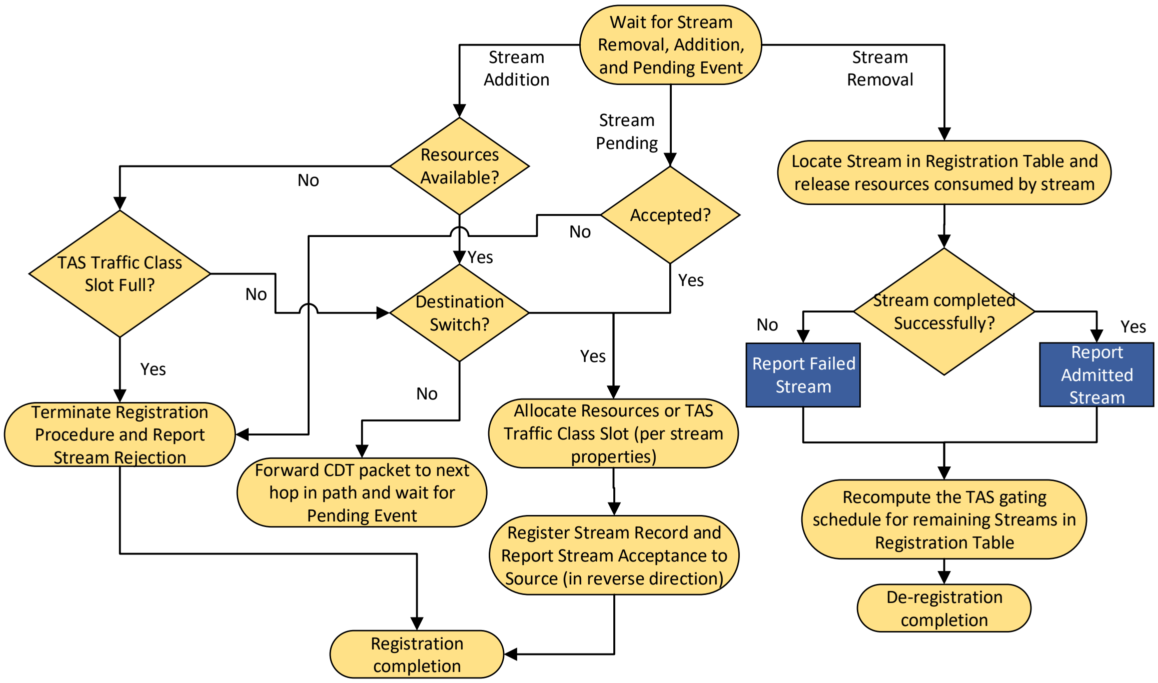


Figure 4.4: The main logical steps performed by each switch along the stream's path are shown to apply stream registration and reservation. Each switch generally waits for an event (addition, removal, or pending) for each stream. For instance, a stream removal is usually based on the resource utilization time (stream lifetime) that was specified at stream establishment. The bridges that allocated resources for the stream can remove the stream after the stream lifetime has expired. For the cases of stream addition or pending, the event is the CDT message received (whether in the forward or reverse direction). Towards completing (finalizing, confirming) a stream reservation (registration), the pending event is the event when a CDT message is received in the reverse direction where each switch (not the last switch) waits for the approval (confirmation of reservation) of the next-hop switch.

Admission Control

The admission control element extracts the necessary information from the CDT packet and forwards the information according to the CDT type. The switch forwarding mechanism applies several steps to accurately decide whether to accept or reject the stream transmission request. Note that the stream transmission request corresponds to a CDT request. The switch consults the resource manager module to check if enough resources (bandwidth) are available for the new stream. In particular, a given stream's bandwidth requirement is calculated by multiplying the ST injection rate with the average packet size and dividing by the current ST slot size. Note that the traffic class TAS slot time is the time during which the TAS gate is open to transmit frames belonging to the considered class. Also note that all GCEs are executed during each CT. If the CT is smaller than the aggregate of the GCEs, then we need to either increase the CT or reject streams that cause the CT to be exceeded.

Flow Scheduling

This element currently takes the Time-Aware Shaper into consideration. Due to the TAS's requirements on time synchronization between network components (switches, hosts, etc.), all switches/hosts follow the TAS GCL timescale cycle time (e.g., 50 μs). Depending on the number of supported traffic classes, the TAS cycle time can be divided into appropriate slots for each traffic class load. The TAS CT is divided among all the traffic classes (in our evaluation model, we consider two traffic classes, BE and ST). Currently, in our evaluations, the CT is initially predefined to 50 microseconds. Note that the CT could be changed/configured dynamically. The dynamic adaptation of the CT with respect to new stream additions, application specifications, or other events is a topic for future research.

Stream Registration Table

In our evaluations, stream creation follows a Poisson process with a prescribed stream generation rate π. Different scenarios with varying mean stream lifetimes (durations) enable analysis of how reconfiguration works in multiple settings. The stream registration table contains the characteristics of the source streams that are established for the corresponding bridge egress port. Each record gets populated (if accepted) on the reverse path taken by the stream's registration message (after reaching the destination switch).

Traffic Shaper — Time-Aware Shaper (TAS)

The TAS is the main shaping and scheduling mechanism that controls the gating schedules for all the traffic classes within the TSN domain. All bridges are synchronized to the same gating schedule GCL CT that is initially predefined by the network administrator. Ideally, we want the CT to be large enough for all streams from all traffic classes to be accommodated and short enough that all streams meet their delay requirements. In our current evaluations, the CT is predefined at 50 microseconds.

Reconfiguration Calculus

The reconfiguration (dynamic configuration) of the TAS schedules (switch GCL/GCE) for each egress port is dynamically invoked according to two principle events, *i*) adding a new stream, and *ii*) removing an existing stream. The switch's gating ratio (for a particular stream belonging to a defined traffic class) reports certain parameters (e.g., packet injection rate, maximum packet size, latency requirement, deadline, and application response time) which are then used to check if enough slot time is available (which corresponds to attempting bandwidth reservation). In the event that no slots are available, the GCE slot size is recomputed (according to the registered stream properties within the registration table), generally by allocating more resources from BE Traffic. The stream lifetime is reported by the source to the network as User/Network Information (UNI). Each UNI is propagated by each switch along the path which allows the switch to register the stream and store the stream's resource utilization time (stream lifetime) among other critical information. Any information pertaining to the UNI of a stream is recorded in the stream registration table. In terms of GCEs for TAS with the support of ST and BE traffic classes, only two GCEs within a GCL (1/0 (ST/BE) for the first entry and 0/1 (BE/ST) for the second entry) are necessary with a total of three outbound queues for each egress channel port in a TSN switch; two queues for each traffic class (ST and BE), and another queue for CDT traffic (signaling traffic).

Upon initialization, each switch allocates 20% of the CT to ST traffic, and BE traffic is initially allocated the remaining 80% of the CT. These initial settings are chosen arbitrarily to start up the network system. As streams get registered, the ST slot time is recomputed (according to the stream packet size, ST injection rate, and current slot time). If the stream is the first stream to the switch, then the ST slot size is set at a minimum to 11% (a minimum of 1% step size for the added ST flow plus the minimum ST partition of 10%) of the CT. Thus, as ST streams are admitted and exit the system, the ST vs. BE allocation is dynamically adapted in the reconfiguration scenarios. The minimum step size of 1% of the CT is considered so as to limit the adaption granularity to a reasonable level. Note that the ST to BE slot size (or gating ratio) is limited to 10% and 90% for the lower and upper limits, respectively. The main reasoning behind this design choice is to avoid any starvation of lower priority traffic.

Path Computation

While a path computation module is fundamentally necessary in any switch (in a decentralized/distributed network), we define static shortest path routing tables for destination addresses and associated ports on each switch. Essentially, we assume a procedure to compute paths, i.e., we assume that there is a path computation module, e.g., Path Computation Engine (PCE), that is used in both centralized and distributed configuration models (the path computation can be accelerated with hardware modules [240, 325, 386], if needed). We make this assumption to simplify operations and place emphasis on the TAS reconfiguration technique.

Network Resource Table

To remove certain overheads of the configuration procedure, the network resource table operates in tandem with the stream registration table to accurately determine the required network resources (mainly bandwidth for our traffic model) per switch egress port. The network resource table classifies streams based on periodic and sporadic stream properties, though currently our focus is on periodic ST streams. Any stream that has been approved by a switch has an associated record in the network resource table, located within each switch, which can be called to compute and store current and remaining link/port loads for each switch. Each egress port has a network resource table.

4.5 Performance Evaluation

4.5.1 System Overview and Simulation Setup

This section explains the simulation setup and model. Furthermore, the topology and simulation scenarios will be presented. Throughout, we employ the OMNet++ [420] simulation environment. For each evaluation for a given set of parameters, we conduct 5 independent simulation replications; each replication simulates the network for 20 seconds. The widths of the resulting 95% confidence intervals are smaller than 5% of the corresponding sample means and are therefore omitted from the plots to avoid clutter.

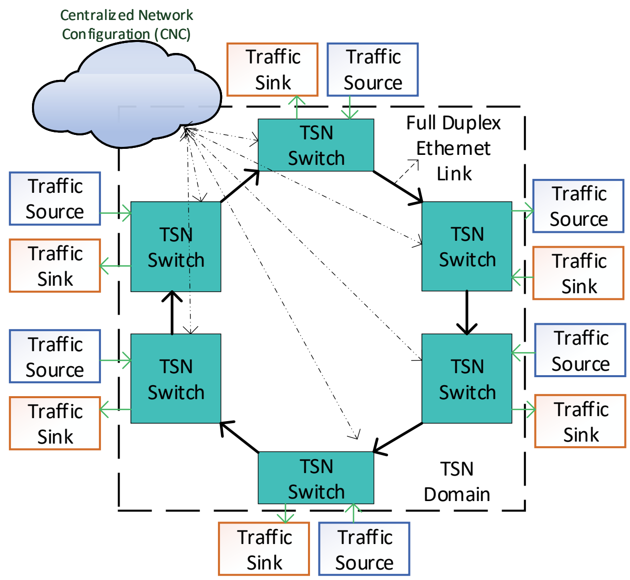


Figure 4.5: Industrial control loop topology [190]: Each source generates stream data with varying hop counts and packet rates unidirectionally or bidirectionally across the six switches ultimately destined to a sink

Table 4.1: Simulation Parameters

Key	Symbol	Value
Simulation Duration	Sim_{limit}	100 seconds
Initialized Cycle Time	GCL_{CT}	50 μs
Initialized Gating Ratio	ST^{R}_{init}	20% (i.e., 10 μs)
Average Streams per Second	π	$1-20$
Average stream duration	τ	$2-5$ seconds
BE Traffic Intensity	ρ_L	$0.1, 1.0, 2.0$ Gbps (580 byte packets)
ST sources	S	6
Queue Size	Q_{size}	512 KB
Link Capacity	R	1 Gbps

Network Model

The network topology is modeled around an industrial control loop topology that consists of six core switches in a ring topology. In the case of the centralized model, a CNC is used with out-of-band connections to each of the core switches; while in the distributed approach, the signaling is in-band and can interfere with data traffic within the TSN domain, as shown in Fig. 4.5. Each switch-to-switch link operates as a full-duplex Ethernet link with a capacity (transmission bitrate) $R = 1$ Gbps. Each switch can act as a gateway for a number of traffic sources and one sink. The distance between two successive switches along the ring is fixed to 100 m and the switch-to-switch propagation delay is set accordingly to 0.5 μs. The out-of-band connections have exactly the same configurations as the normal full-duplex Ethernet links in the data plane, i.e., the same bitrate and propagation delay. All switches are configured to use 802.1Qbv TAS as the traffic shaper for each switch-to-switch egress port whose flow schedule (ST gating ratio and cycle time) is configured by the CNC in the centralized (hybrid) model and independently in the decentralized (fully distributed) model. For all simulation runs, the ST slot size is initialized to 20% of the CT. For the operation without reconfiguration, the ST slot size is kept at 20% of the CT; whereas, for the operation with reconfiguration, the ST slot size is dynamically recomputed when the first stream transmission request arrives.

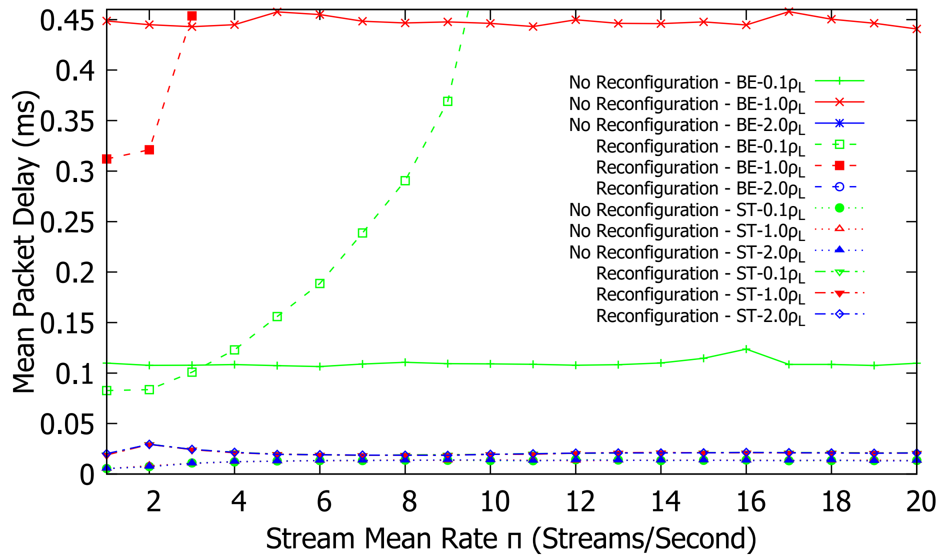
Traffic Model

We consider periodic (pre-planned) traffic for the ST traffic and sporadic (random) Poisson traffic for the BE traffic. To emulate dynamic conditions in the network, we employ several distributed ST sources that generate π ST streams according to the network and traffic parameters shown in Table 4.1. The stream generation follows a Poisson process with a prescribed mean rate of π generated streams per second. We refer to the stream generation rate π also as the stream mean rate π. Each ST stream injects one packet of size 64 bytes per cycle. A destination address is assigned by the number of switch-to-switch hops. A given stream that has been generated at the traffic source attached to a given TSN switch is destined to the traffic sinks at the other five TSN switches with a uniform probability of 1/5. Furthermore, at stream creation, each stream is given a start time (usually the current runtime), and a finish time according to a stream lifetime (duration) that follows the exponential distribution with mean τ. The exponential stream lifetime is considered as call level dynamics in communication networks often follow Poisson process dynamics, i.e., exponential call lifetimes. As TSN networks become more commonly deployed, it will be important to verify the stream lifetime dynamics through real system measurements.

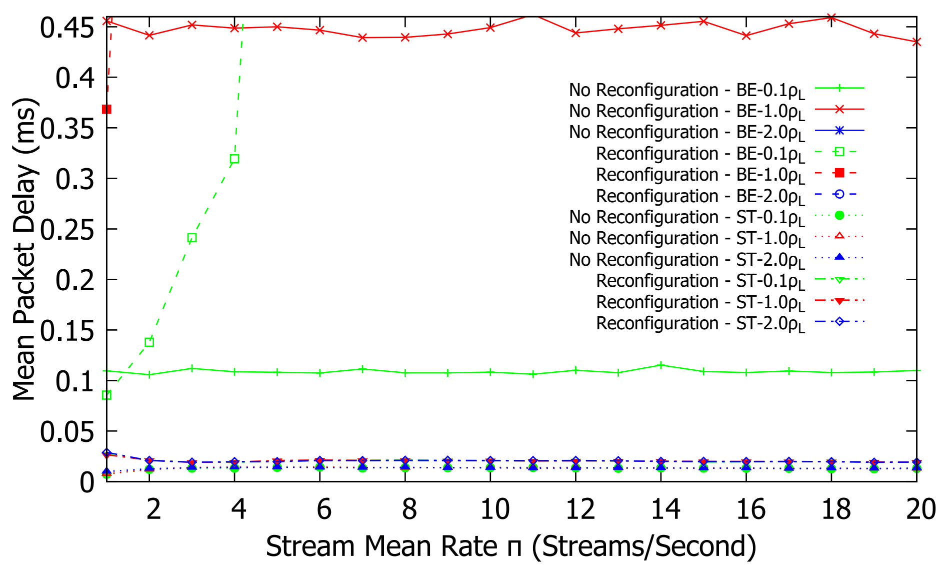
We consider admission as the completion of the reservation of the network resources for the flow from the source node to the destination node. Each source is attached to a core TSN switch gateway (first hop switch). While the TSN switches operate with time synchronization, the ST sources (outside the TSN domain) do not need to be synchronized. However, note that the ST traffic follows an isochronous traffic class, as specified by IEC/IEEE 60802, whereby the sources are synchronized with the network after stream registration is completed. In particular, the ST sources inject the ST traffic in just-in-time fashion, i.e., the transmission of the ST packets out of a source starts at the instant of the start of the ST transmission slot at the switch that is directly attached to the source. Packets are time stamped for the packet delay measurement at the time instant when the packet transmission out of the source commences.

4.5.2 Centralized (Hybrid) Model Evaluation

In evaluating the proposed solution described in Section 4.3, we consider both periodic ST traffic and sporadic BE traffic, as described in Section 4.5.1. We evaluated the CNC with TAS shaper on the industrial control loop for the unidirectional and bidirectional topologies and results are collected for the simulation parameters shown in Table 4.1.



(a) $\tau = 2$



(b) $\tau = 5$

Figure 4.6: Centralized (Hybrid) Unidirectional Topology: Mean end-to-end delay for ST and BE traffic for mean stream durations $\tau = 2$ and $\tau = 5$ seconds under different BE loads ρ_L, and ST stream rates π.

Unidirectional Ring Topology

Fig. 4.6 shows the average mean delay for ST traffic and for BE traffic for the centralized unidirectional ring topologies. The average delays are generally short and

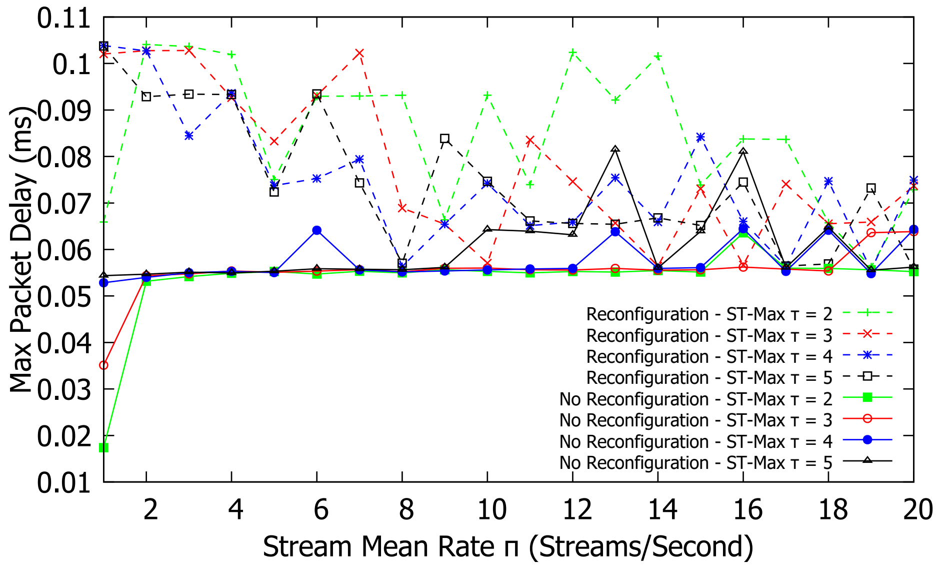


Figure 4.7: Centralized Unidirectional Topology: Maximum packet delay for TAS with centralized configuration (CNC) management.

stable for both BE and ST traffic. Since the CNC manages the ST traffic streams and therefore guarantees the bandwidth rates needed across the stream's path, the ST delays are less than 100 μs for all average stream durations τ. The ST streams with reconfiguration at the CNC experience higher delays than for the no reconfiguration scenarios since we essentially push more ST traffic into the network which increases the queuing delay. BE traffic experiences much higher delays than ST. With the no reconfiguration approach, the BE traffic delay is nearly constant since the gating ratio is left unchanged. The BE mean delay increases dramatically (up to 21 ms) with reconfiguration since the accepted ST streams tend to consume the full permitted 90% of the CT, leaving only very limited transmission resources for BE traffic.

TSN needs to limit the maximum delay in order to deterministically forward traffic across a TSN domain. Fig. 4.7 shows the maximum delay for the ST traffic. We observe from Fig. 4.7 that for the unidirectional ring topology with a maximum of five hop streams, the maximum delay for the reconfiguration approach is below

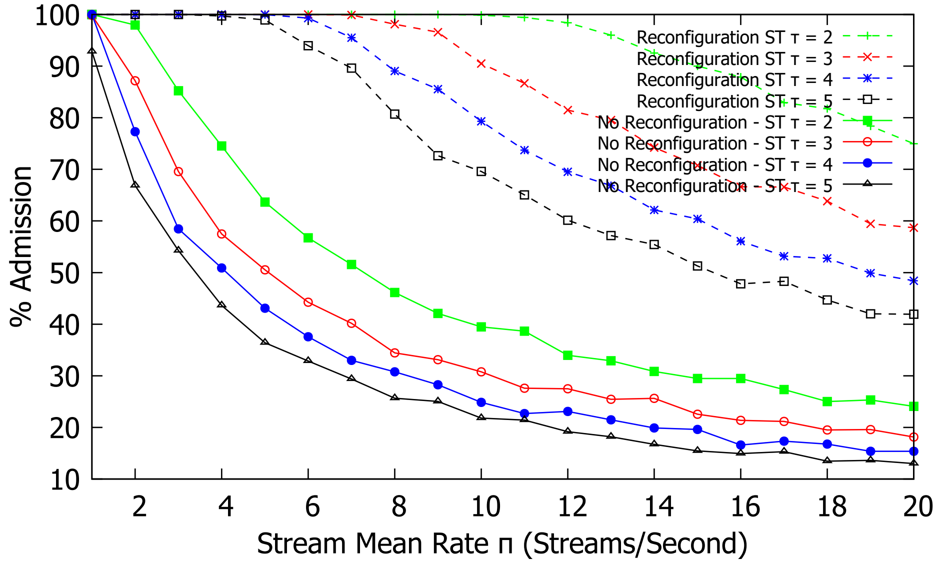


Figure 4.8: Centralized Unidirectional Topology: Stream Admission percentage for TAS with centralized configuration (CNC) management.

0.11 ms. On the other hand, for the "no reconfiguration" approach, the maximum delay is below 60μs due to lower frame residence time on each switch; however, reconfiguration increases the admission of ST streams as examined next in Fig 4.8. TAS in conjunction with the CNC registration and reservation procedure provide a prescribed bandwidth share of the egress port using time division multiplexing. With our empirically chosen parameters, the maximum delays is capped to approximately 100μs which is suitable for the considered topology and time-critical ST traffic that requires less than 1 ms of delay.

While QoS metrics are important, another factor that determines the performance gains is the admission ratio for the system. Fig. 4.8 shows the stream admission ratio for both reconfiguration and no reconfiguration. In general, each generated stream needs a data rate of about 11.5 Mbps for a 50μs CT (which corresponds to approximately 45μs of maximum ST slot size since we permit ST traffic to take up at most 90% of the CT) for each egress port on the stream's path with 1 packet injected

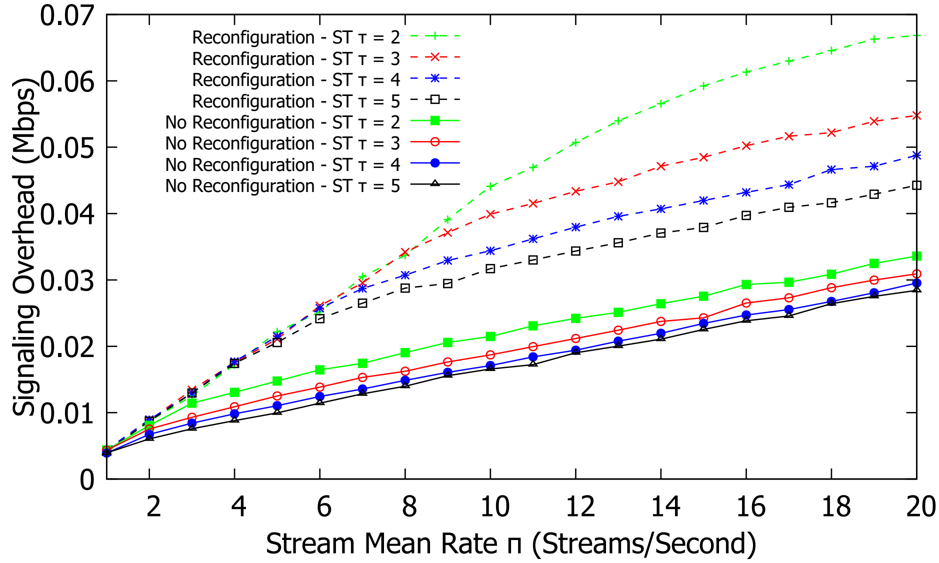


Figure 4.9: Centralized Unidirectional Topology: Stream average signaling Overhead for TAS with centralized configuration (CNC) management.

by an ST per CT and a fixed packet size of 64 B. With an egress port channel capacity of $R = 1$ Gbps, approximately 86 streams can be accommodated. Compared to the "no reconfiguration" approach, the reconfiguration approach significantly improves the admission rates at the expense of higher BE traffic delays, since the ST slot borrows BE time slots to accommodate the ST streams. We also note that increasing the maximum ST allocation above 90% would increase the ST stream admission ratio, at the expense of starving the BE traffic.

CDT traffic that requests transmission guarantees from the CNC experiences some delay before being either admitted or rejected. Since the control plane is out-of band from the data plane within the TSN domain, the delay is constant (around 4 μs) throughout the simulation run.

Stream registration and reservation introduce some control plane overhead. Fig. 4.9 shows the signaling overhead. More specifically, the overhead is measured as the signaling traffic rate in Mbit/s at the CNC for both incoming and outgoing control

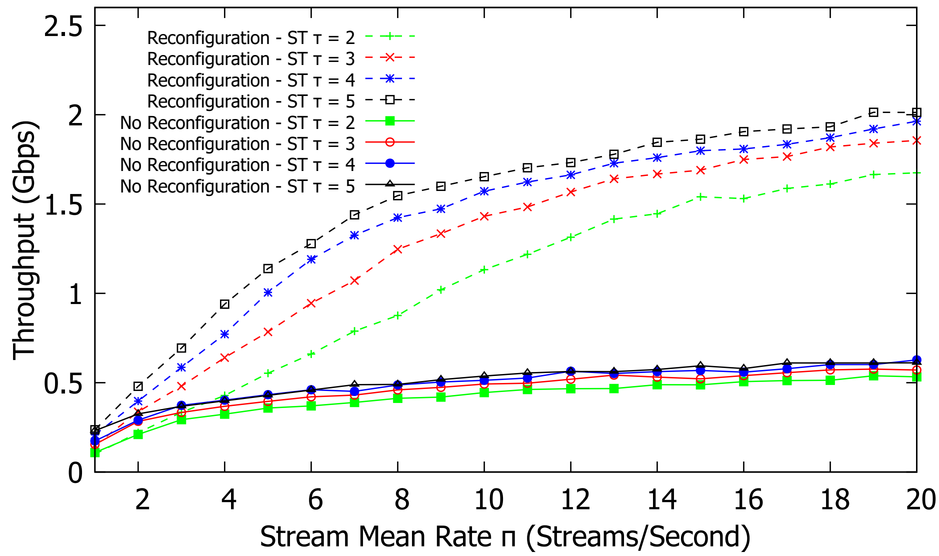
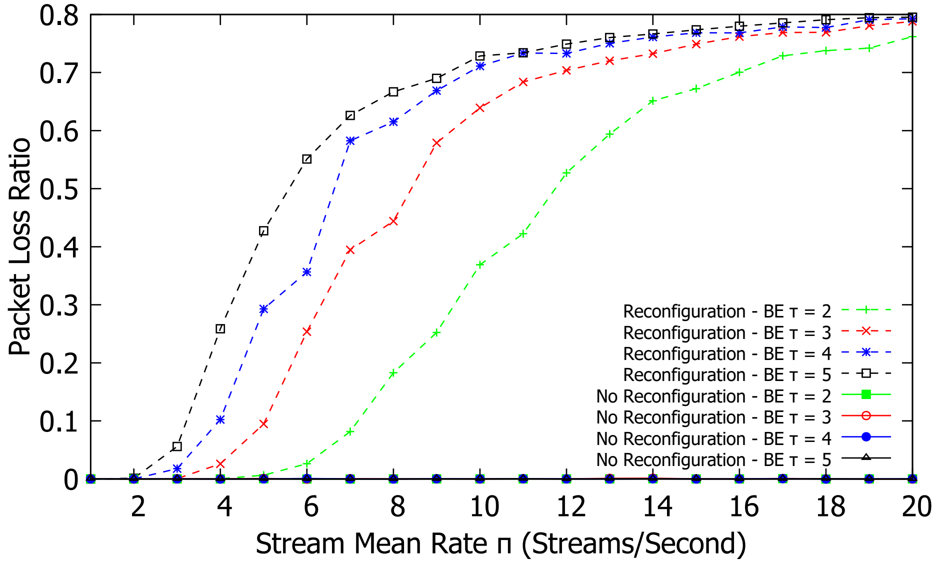


Figure 4.10: Centralized Unidirectional Topology: ST total average throughput measured at the sink for TAS with centralized configuration (CNC) management.

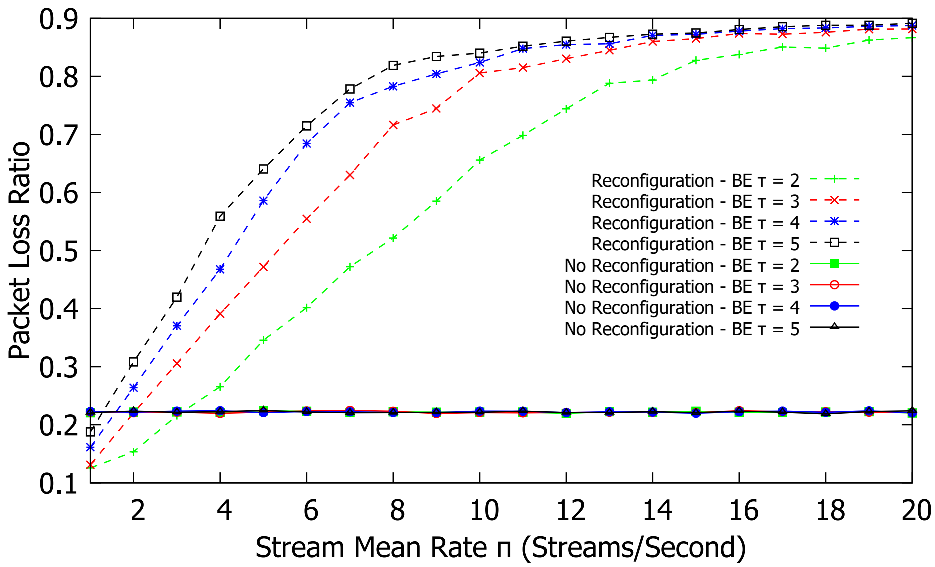
(CDT) traffic. Generally, the reconfiguration introduces more signaling overhead; however, Ethernet generally has large bandwidths, thus the CDT traffic rates are minuscule compared to the link capacities. Furthermore, when $\tau = 2$, we observe higher signaling overhead due to accepting larger numbers of streams (rejections are inexpensive compared to acceptance) both with and without reconfiguration.

Fig. 4.10 shows the average throughput measured at the sink for ST traffic. We observe from Fig. 4.10 that the reconfiguration substantially increases the throughput compared to the no reconfiguration scenario. Typically, the throughput is more than doubled by the reconfiguration.

To examine the reliability performance, Fig. 4.11 shows the BE packet loss ratio for mid and high BE traffic loads ρ_L; we omitted the low BE traffic load which has negligible losses. Since the CNC manages only ST streams, the TSN guarantees (which include zero packet loss since retransmissions are in general too expansive for ST traffic) are only valid for ST streams. As the ST traffic load increases in the



(a) Mid BE Traffic Load $\rho_L = 1$ Gbps



(b) High BE Traffic Load $\rho_L = 2$ Gbps

Figure 4.11: Centralized Unidirectional Topology: BE frame loss ratio for TAS with centralized configuration (CNC) management.

reconfiguration scenario, the BE packet loss increases. For the "no reconfiguration" approach, the BE packet loss is typically constant even for high loads of BE traffic.

For a benchmark comparison of the TSN effectiveness, and specifically TAS, we conducted additional evaluations for the scenario in Fig. 4.6 without the TSN slot reservation, admission control, and TAS scheduling. Specifically, we considered an ST stream mean generation rate of 1–20 streams per second with a mean lifetime $\tau = 5$ seconds with the mid and high BE traffic loads of $\rho_L = 1.0$ Gbps and 2.0 Gbps. We employed strict priority scheduling at each switch without any TSN slot reservation, i.e., each switch output port schedules and transmits all ST packets before any BE packets.

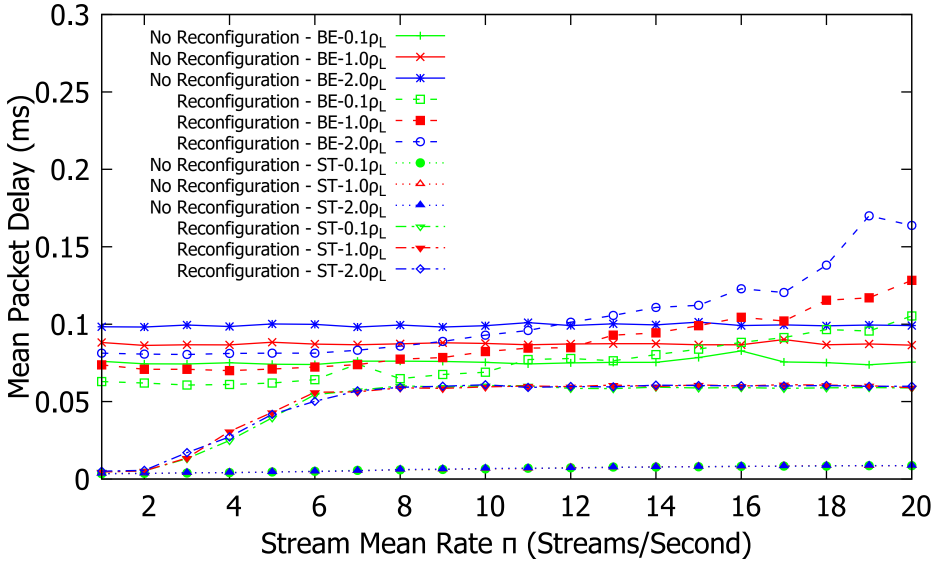
We outline three main observations for the unidirectional ring topology. First, while the mean delays were generally very low for ST traffic (34–55 μs for the low traffic load range $\pi = 1$ to 5 ST streams per second), the priority scheduling of the ST packets can severely starve the low-priority BE traffic (for the high $\rho_L = 2.0$ Gbps BE load, the mean BE packet delays increase from a minimum of 15 ms to a maximum of around 0.1 s as the ST load increases from 1 to 20 streams per second; whereas, with TSN, the mean BE packet delays increase from around 10 ms to 21 ms, which is outside the plotted range of Fig. 4.6(b)). Additionally, compared to TSN, the maximum delays and jitter increase more strongly as the BE and ST loads increase (the ST maximum packet delays range from 34 μs to 20 ms; while, with the TSN operation, the ST maximum packet delays hover around 55–101 μs; see Fig. 4.7). This stronger increase of the maximum ST packet delays is a result of the BE packet traffic interfering with the ST packet traffic due to the lack of TAS operation. In particular, ST packets are blocked from transmission during an ongoing transmission of a 580 byte BE packet (as we considered non-preemptive priority scheduling). Second, since no admission control based on TSN slot reservation is used, congestion arises for ST traffic loads of $\pi = 6$ to 20 ST streams per second, causing high mean and maximum delays for both ST and BE traffic. Third, due to the congestion, packet drops occur at high ST loads for both ST and BE packet traffic. We also note that since no signaling traffic is used, the priority scheduling benchmark without TSN operation provides a performance reference for both the centralized and the decentralized TSN model.

Overall, we conclude that the proposed centralized (hybrid) reconfiguration approach provides a means to ensure that dynamically varying numbers of ST streams are accommodated as permitted by the available link capacity in the unidirectional ring network. However, the unidirectional ring network does not involve any distinct routing choices towards the destination. In order to examine the performance of the proposed centralized reconfiguration in a network with different routing paths, we next consider the operation of the ring network topology as a bidirectional ring network.

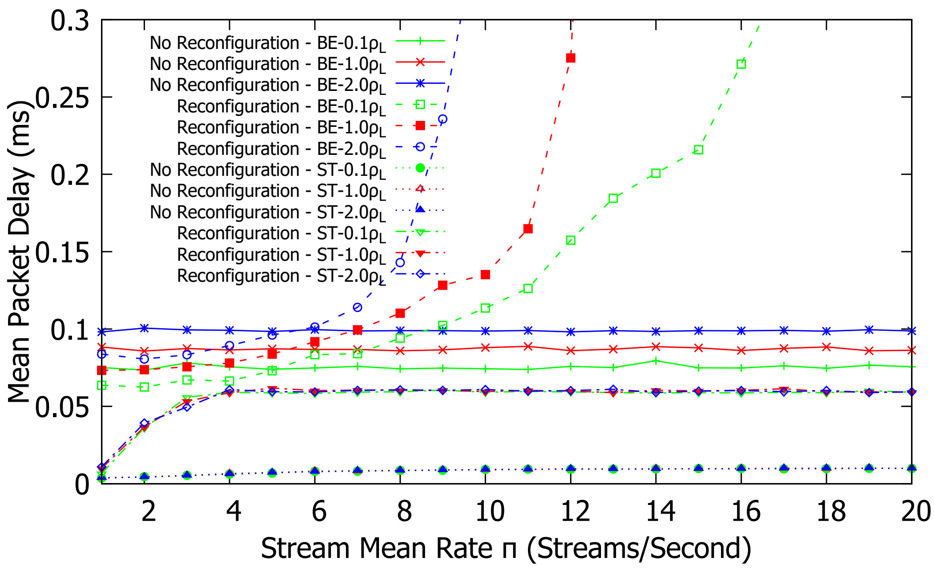
Bidirectional Ring Topology

The unidirectional ring topology certainly simplifies the calculation of the ST slot window in the reconfiguration. In order to examine whether the proposed centralized (hybrid) reconfiguration approach can efficiently utilize the higher capacity of a more complex network with multiple routing options, we examine the bidirectional ring network. In the bidirectional ring network, each two-port switch has now two paths to the destination. We employ shortest path routing according to the hop count. We set the edge link (source to first ring switch and last ring switch to sink) capacities to 2 Gbps to avoid congestion on the edge links (which the CNC does not control).

Fig. 4.12 shows the average mean ST and BE packet delay for different stream lifetimes τ. Compared to the unidirectional topology (see Fig. 4.6), the bidirectional significantly reduces the packet delay since an extra port with full-duplex link support now provides extra capacity to service streams giving more slot reservations to BE even at high ST stream loads.



(a) $\tau = 2$



(b) $\tau = 5$

Figure 4.12: Centralized Bidirectional Topology: Mean end-to-end delay for ST and BE traffic for varied mean stream lifetime τ for different BE loads ρ_L, and ST stream rates π.

Fig. 4.13 shows the maximum ST packet delays for the bidirectional ring topology with CNC. We observe from Fig. 4.13 in comparison with the corresponding maximum

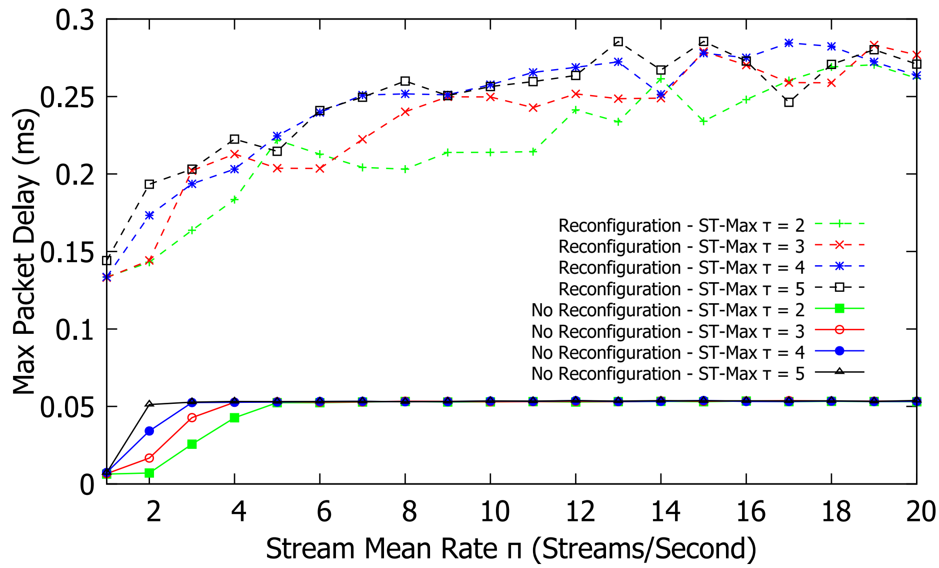


Figure 4.13: Centralized Bidirectional Topology: Maximum ST packet delay for TAS with centralized configuration (CNC) management.

packet delay plot for the unidirectional ring in Fig. 4.7, that the bidirectional topology with configuration gives higher maximum packet delays, which is mainly due to the substantially increasing ST stream acceptance, as examined next in Fig 4.14. The "no reconfiguration" keeps the ST slot size at the initialized value (20% of CT, i.e., 10 μs), resulting in a constant maximum delay of around 50 μs, albeit at the expense of rather low admission rates, see Fig 4.14.

Fig. 4.14 shows the stream admission ratio (percentage). With the high stream generation rate $\pi = 20$ streams/s and long average stream lifetime $\tau = 5$ s, the admission rate is still slightly above 90% for the bidirectional topology with CNC reconfiguration. The bidirectional ring thus achieves a substantially increased (close to 50% higher) admission rate compared to the unidirectional ring examined in Fig. 4.8. In contrast, the increases of the admission ratio of the no reconfiguration approach with the bidirectional ring compared to the unidirectional ring are more modest (roughly 20%). This is mainly because the initialized gating ratio is too restrictive and severely

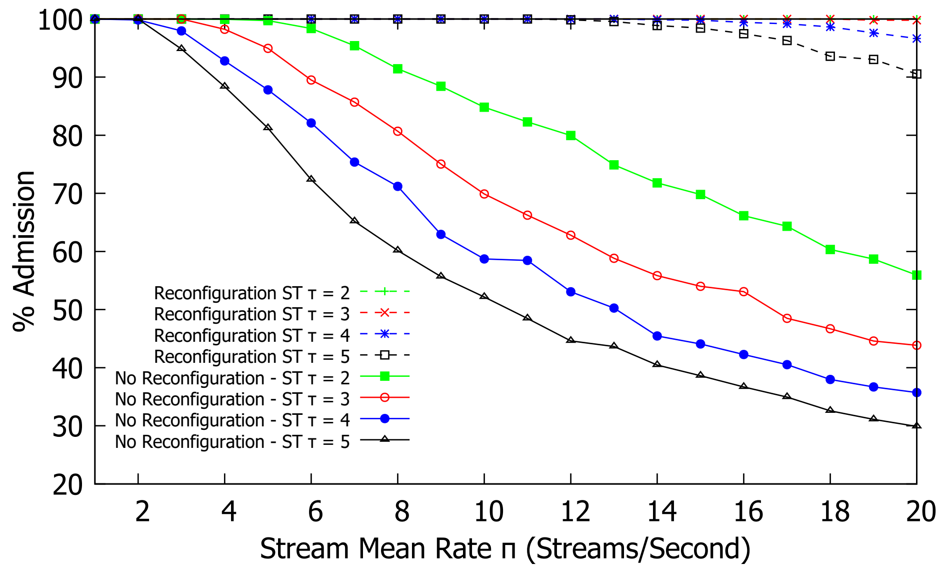


Figure 4.14: Centralized Bidirectional Topology: Stream admission percentage for TAS with centralized configuration (CNC) management.

underutilizes the links. We found in additional evaluations that are not included to reduce clutter that different BE loads ρ_L do not impact the ST stream performance due to the TAS operation, i.e., TAS effectively partitions the traffic at the egress switch/port (BE traffic does not block ST traffic).

Similar to the unidirectional ring, the bidirectional ring topology provides constant signaling delay (around 3.5 μs) due to the CNC out-of band signaling channels. The average signaling delay is slightly lower than in the unidirectional ring (which had a signaling delay around 4 μs), mainly since the signaling hop distances in the bidirectional ring are shorter than in the unidirectional ring.

Fig. 4.15 shows the signaling overhead. Since the bidirectional ring topology is effectively the same as the unidirectional ring topology (albeit having another port to the switch), the signaling overhead in the bidirectional ring network is in general very similar to the signaling overhead in the unidirectional topology. Note that while the hop traversal is reduced (since the stream can take one of two paths to the

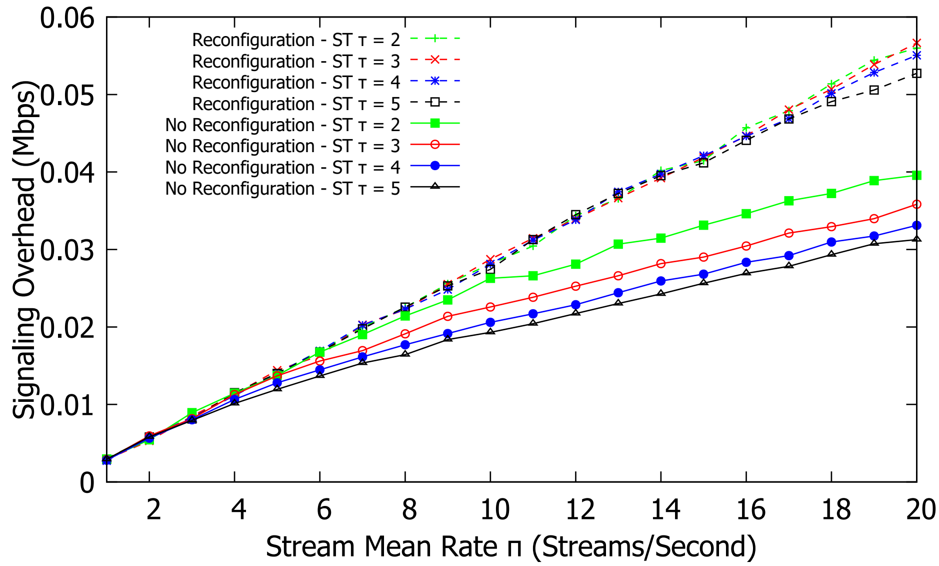


Figure 4.15: Centralized Bidirectional Topology: Average stream signaling overhead for TAS with centralized configuration (CNC) management.

destination governed by shortest path, i.e., the smallest hop count), the number of sent and received CDT frames are generally the same. Similar to the unidirectional topology, the reconfiguration approach generates more CDT traffic. Note that admissions are in general more costly in terms of sent and received CDT frames in the network. Therefore, the higher the admission rate, the more overhead is observed in the control plane, though based on Fig. 4.15, the overall overhead is well below 1 Mbps and therefore is minuscule compared to the channel capacity. We also observe from Fig. 4.15 that the results for different stream lifetimes τ differ only very slightly since for any τ value, almost all the streams are accepted, generating the same total overhead.

Fig. 4.16 shows the average overall throughput measured at the ST sinks for the bidirectional ring topology. Compared to the unidirectional ring (see Fig. 4.10), the throughput for the bidirectional ring is much higher, typically increased by a factor of two.

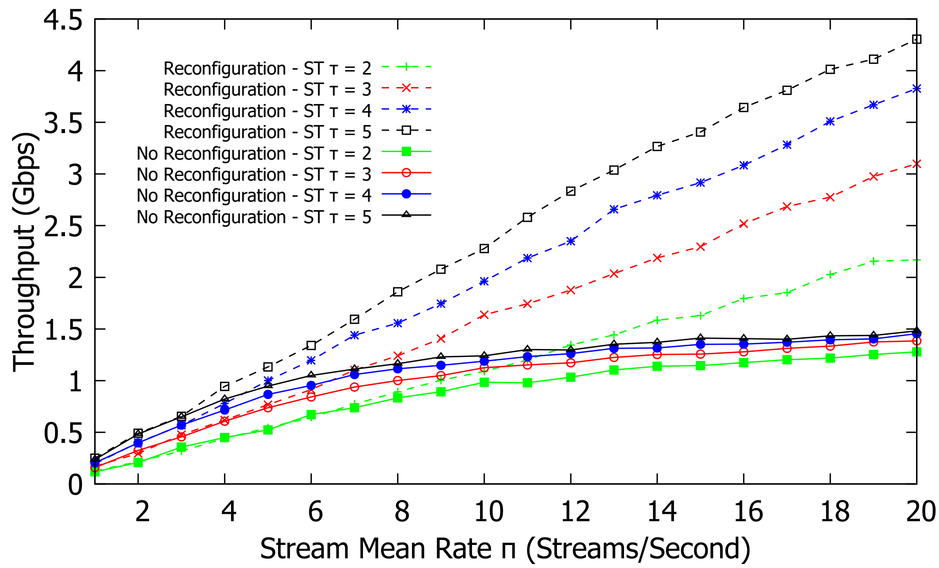
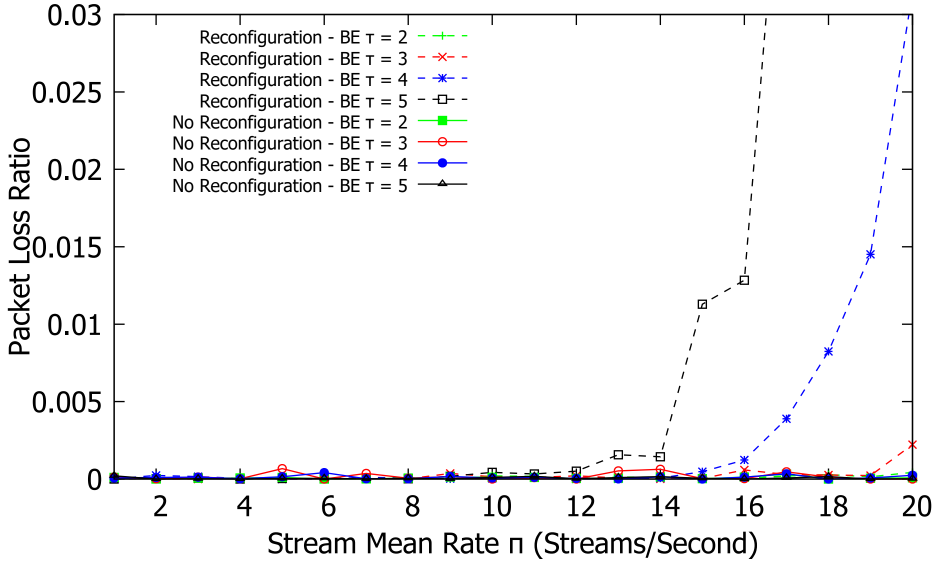


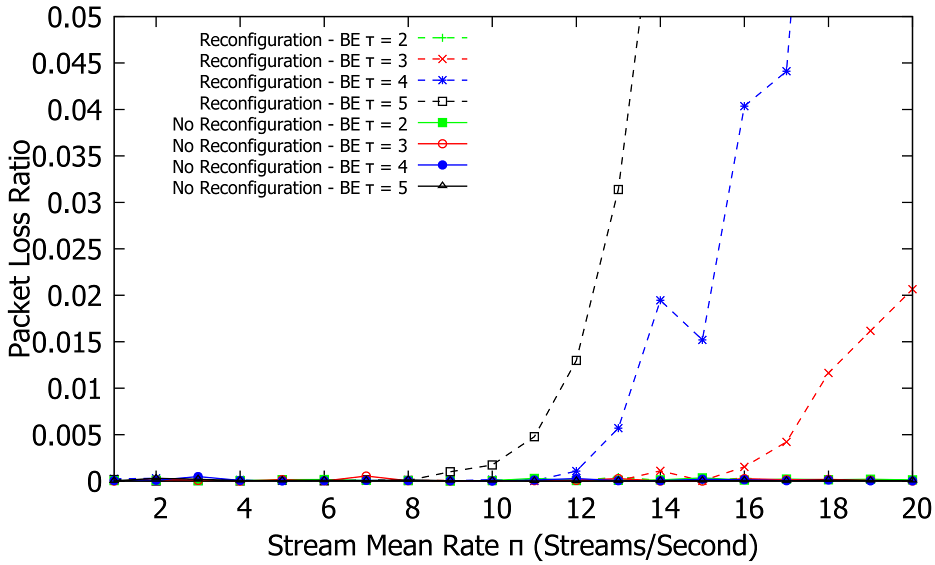
Figure 4.16: Centralized Bidirectional Topology: ST Total average throughput measured at the sink as a results of TAS with centralized configuration (CNC) management.

Similar to the unidirectional ring topology, the bidirectional topology achieves zero loss for ST streams while significantly reducing the BE packet loss rate. Fig. 4.17 shows the BE packet loss ratio for the bidirectional ring network. The maximum BE loss for the high BE traffic intensity $\rho_L = 2.0$ is around 30% which is a significant reduction from the unidirectional topology (of around 90%, see Fig. 4.11).

In contrast to the unidirectional topology, the bidirectional topology with central (hybrid) CNC reconfiguration achieves improved QoS metrics and admission rates. Overall, the ST traffic throughput is typically doubled in the bidirectional ring network compared to the unidirectional ring network. We can thus conclude that our proposed centralized (hybrid) CNC reconfiguration can effectively utilize the higher capacity provided by the bidirectional ring network for dynamic ST traffic, with random ST flow generations and random ST flow lifetimes.



(a) Mid BE Traffic Load $\rho_L = 1$ Gbps



(b) High BE Traffic Load $\rho_L = 2$ Gbps

Figure 4.17: Centralized Bidirectional Topology: BE frame loss ratio for TAS with centralized configuration (CNC) management.

4.5.3 Decentralized Model Evaluation

Analogous to the centralized (hybrid) reconfiguration evaluation, we evaluate our proposed decentralized reconfiguration from Section 4.4 with both periodic ST traffic and sporadic (random) BE traffic, as specified in Section 4.5.1. As before, we evaluate the network with TAS shaper on the industrial control loop unidirectional and bidirectional topology and collect results for the simulation parameters shown in Table.4.1.

Unidirectional Ring Topology

The decentralized model essentially transfers some of the CNC functions (e.g., TAS reconfiguration and resource reservation modules) from the centralized model down to the TAS enabled egress ports of the TSN switches in the data plane. The main difference between the centralized and decentralized models is the signaling performance which is now in-band and can affect data traffic. In additional evaluations that are not included to reduce clutter, we have found that with the in-band CDT traffic in the decentralized model, the average ST and BE packet delays are about the same as the centralized model in Fig. 4.6. Typically, the ST stream's average delay is minimal to near constant for both the reconfiguration and "no reconfiguration" approaches. For BE, the "no reconfiguration" approach produces constant average delay for each BE ρ_L traffic intensity.

Fig. 4.18 shows the maximum ST packet delay for the unidirectional ring network using the decentralized model. In contrast to the average ST packet delay, the maximum delay is affected by the in-band CDT traffic. In the decentralized model, the CDT traffic is given the highest priority above both ST and BE traffic. Therefore, the maximum delays can reach about 150 μs, which is somewhat higher than for the

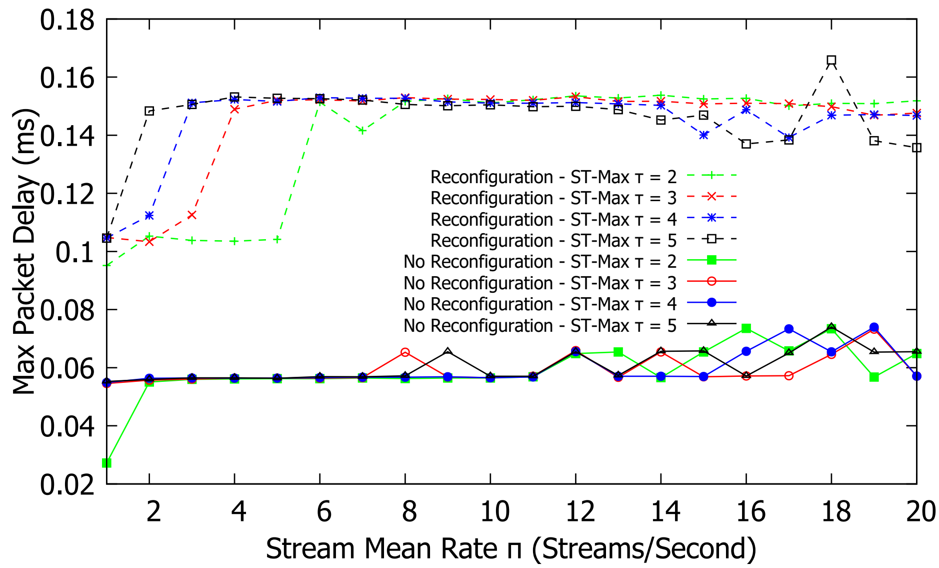


Figure 4.18: Decentralized Unidirectional Topology: Max delay for TAS.

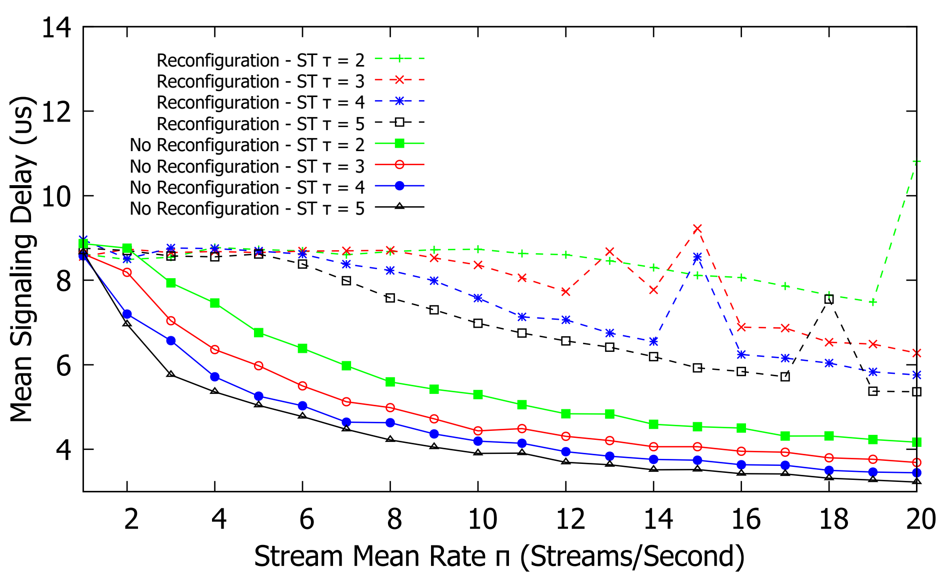


Figure 4.19: Decentralized Unidirectional Topology: Average stream signaling delay for TAS.

centralized reconfiguration in Fig. 4.7, but still well below 1 ms.

The stream admission rate for the decentralized model is very similar to the centralized model (see Fig. 4.8) and is not displayed in detail. Fig. 4.19 shows the signaling delay for ST stream registration in the decentralized model. In contrast to

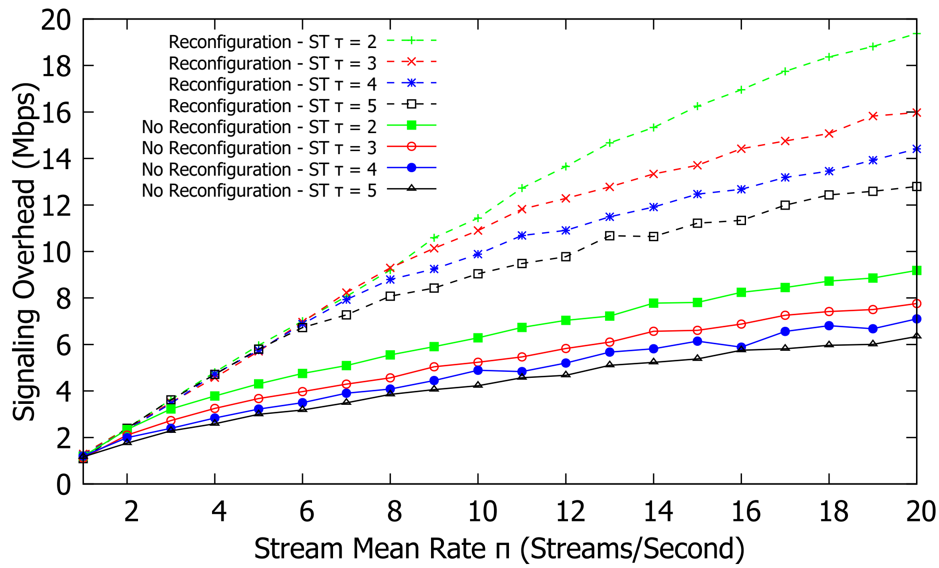


Figure 4.20: Decentralized Unidirectional Topology: Stream Signaling Overhead for TAS.

the centralized model, the decentralized model's in-band CDT traffic implies varied stream signaling delays. As the streams generation rate π increases, the overall average signaling delay decreases which is due to the increased rejections as more streams attempt to request network resources. In the decentralized model, a rejection by an intermediate bottlenecked switch implies a termination of the reservation attempt and a notification to any previous pending stream records to cancel the potential reservation and eventually notify the source of the rejection. If this rejection happens closer to the source, then the average signaling delay will be shorter compared to a stream acceptance. In general, the average stream signaling delay is on the order of microseconds which is reasonable for most industrial control systems applications.

Generally, the decentralized model produced greater signaling overhead than the centralized model (cf. Fig. 4.9) since CDT traffic is measured at each data switch traffic port for incoming and outgoing as shown in Fig. 4.20. Analogous to the signaling delay, the more ST streams are accepted, the more overhead is observed. Therefore, as the stream lifetime τ increases and consequently, the more rejections occur, the lower the overhead. Overall, the comparison of the signaling overhead for the decentralized model in Fig. 4.20 with the centralized model in Fig. 4.9 indicates that the decentralization increases the signaling overhead by over two orders of magnitude. However, the aggregate signaling overhead bitrate in the decentralized model is still below 20 Mbps and thus below 2% of the 1 Gbps link capacity.

Throughput results are generally the same when compared to the unidirectional centralized model (cf. Fig. 4.10) and are therefore omitted. Similarly, the packet loss rate is nearly similar to the unidirectional centralized model (cf. Fig. 4.11). However, the unidirectional topology with either the centralized or decentralized approach generally gets bottlenecked at lower traffic loads compared to the bidirectional ring network. Therefore, BE traffic suffers as more ST streams request TAS slot reservations. We next examine the bidirectional ring network for decentralized operation to determine how the BE traffic performance can be improved while maintaining the ST traffic performance.

Bidirectional Ring Topology

For the bidirectional topology using the decentralized model we found that the inband CDT traffic affects the data traffic similar to the decentralized unidirectional model, i.e., maximum ST packet delay is somewhat increased while the mean ST packet delay is essentially unchanged. As the ST stream lifetime τ is increased, i.e., the number of ST streams at any time increases, the BE slots are reallocated to ST streams which increases the mean BE packet delay which is similar to the centralized model (cf. Fig. 4.12) and is therefore omitted.

Fig. 4.21 shows the maximum ST packet delay. While the reconfiguration approach looks very similar to the centralized model (cf. Fig. 4.13), the no reconfigura-

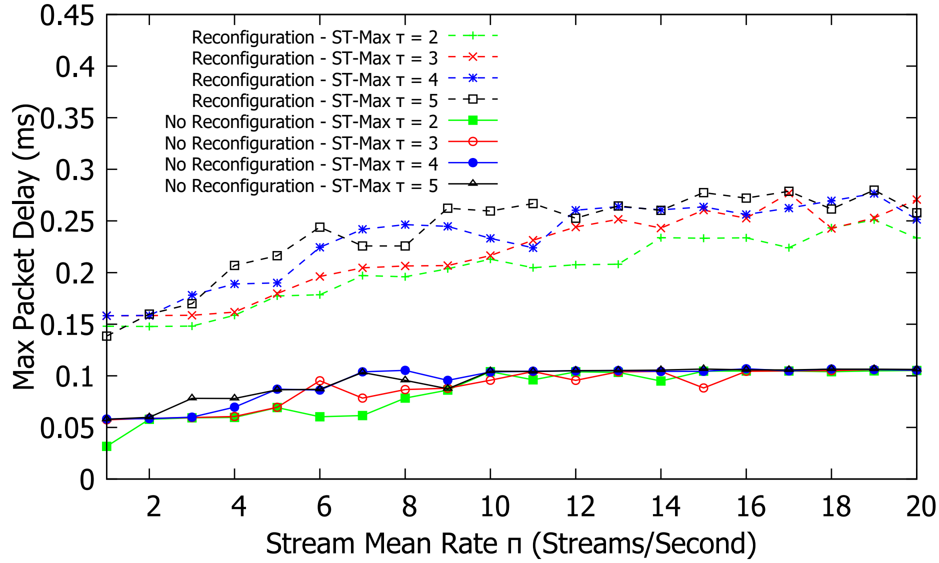


Figure 4.21: Decentralized Bidirectional Topology: Maximum ST packet delay for TAS.

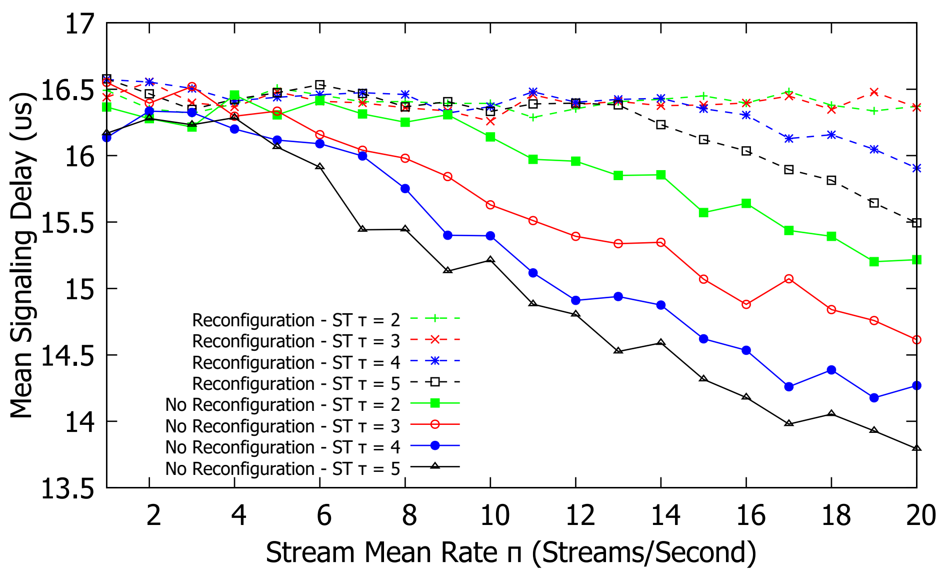


Figure 4.22: Decentralized Bidirectional Topology: Average stream signaling delay for TAS.

tion approach is affected by the in-band CDT traffic which raises the maximum ST packet delay in some no reconfiguration scenarios to around 100 μs.

The admission rate is exactly the same as for the centralized model (cf. Fig. 4.14).

Fig. 4.22 shows the average signaling delay for ST stream registration. Similar to the unidirectional topology, the mean signaling delay starts to decrease as the load increases due to higher rejections.

We found that the stream signaling overhead with the decentralized bidirectional model is similar to the decentralized unidirectional model (cf. Fig. 4.20), albeit slightly lower due to the shorter signaling hop counts in the bidirectional ring network.

Additional evaluations have found that the throughput of the bidirectional decentralized model is nearly identical to the centralized model. We observed only very slightly reduced throughput with the decentralized model compared to the centralized model since the decentralized model carries the control traffic in-band, which very slightly reduces the link utilization for data traffic.

Similar to all the preceding models and topologies, ST streams have zero traffic drops. The BE packet loss rates for the decentralized bidirectional model are nearly identical to the centralized bidirectional model. Similarly, the overall performance is largely improved under the bidirectional topology compared to the unidirectional model due to the additional port and path.

The decentralized model was found to operate nearly identically to the centralized model in terms of QoS metrics and overall admission rate. Thus, the segregation of traffic based on the class of service can be accomplished with the proposed decentralized model without the overhead complexities of a CNC node. A main disadvantage of the decentralized model is the in-band CDT traffic which can delay ST streams, particularly affecting the maximum ST packet delays. A potential workaround to explore in future research is to service all the ST streams first, and then service CDT frames before servicing the BE traffic, though this might lead to additional signaling delays depending on the ST load.

4.6 Conclusions and Future Work

The IEEE 802.1Qcc framework and the 802.1Qbv traffic shaper enable the implementation of a deterministic forwarding plane that provides strict bandwidth guarantees to ST flows without any flow or congestion control mechanism at the source. Using an automated network configuration is an imperative tool set to provide a unified communication platform based on commercial of the shelf (COTS) full-duplex Ethernet with high bandwidth and low complexity compared to Controller Area Networks (CANs), Local Interconnect Networks (LINs), and specialized field-buses in industrial control system applications (e.g., industrial control, automotive, and avionics).

Network designs based on the IEEE 802.1Qcc framework and the 802.1Qbv traffic shaper can form a contract with the source to forward mission critical traffic and to automate the network configuration process using 802.1Qcc for the full lifetime of the stream. Additionally, depending on the forwarding plane port traffic shaper (e.g., TAS), the required schedules can be passed to the switch servers using general user/network information protocols (e.g., TLV, NETCONF/Yang, and SNMP).

In this chapter, we have investigated the impact of TAS reconfigurations in response to dynamic network conditions, i.e., the addition and removal of transient ST streams (flows) with different lifetimes. We have demonstrated the effectiveness of TAS with and without the CNC, i.e., for centralized (hybrid) vs. decentralized (fully distributed) models. We have examined network QoS traffic characteristics when admitting ST flows based on an iterative heuristic approach that computes TAS schedules for current and newly requested ST streams.

Based on the insights from the present study we outline the following future research directions. First, it would be interesting to judiciously change the GCL time for switches during reconfiguration whilst satisfying QoS requirements. The studied reconfiguration techniques should also be examined in alternate approaches for providing deterministic QoS, e.g., [310, 382] as well as in the context of related QoS oriented routing approaches, e.g. [112, 191].

Another interesting future research direction is to adapt the reconfiguration mechanisms that have been developed in this study to the interactions between TSN and fifth generation (5G) wireless communication systems that operate with Ultra-Reliable Low-Latency Communication (URLLC). A few recent studies have begun to explore the use of TSN in the 5G URLLC context, see e.g., [160, 228, 248, 330], indicating significant potential for improving 5G URLLC services by exploiting TSN. The TSN reconfiguration mechanisms developed in this study can potentially help in flexibly providing high-quality 5G URLLC services for varying traffic dynamics. Similarly, TSN reconfiguration may aid low-latency real-time services in future WiFi networks, which may incorporate TSN, see e.g., [32, 94].

In the wider context of QoS networking and related applications, deterministic networking should be examined in the context of emerging multiple-access edge computing (MEC) [140, 178, 286, 385], in particular MEC settings for low-latency applications [149, 439, 454]. As an alternative approach to coordinating the reconfigurations, emerging softwarized control and virtualization paradigms can be explored [55, 136, 138, 226, 374]. Regarding the reliability aspects, a potential future research direction is to explore low-latency network coding mechanisms, e.g., [31, 115, 150, 177, 270, 275], to enhance networking protocols targeting reliable low-latency communication.

Chapter 5

LARGE SCALE DETERMINISTIC NETWORKING: A SIMULATION EVALUATION

5.1 Introduction

The open access and ubiquitous use of Ethernet switched networking technology is propelling the use of full-duplex Ethernet standards in LANs and WANs for a variety of real-time and traditional background applications on converged Ethernet switches and links. The use of Ethernet in industrial environments provides increased bandwidth and better interoperability among other benefits. While the idea of using Ethernet devices in Operational Technology (OT), i.e., automotive, avionics, and industrial control systems, is not new (see Fig. 5.1), the IEEE 802.1Q, Time-Sensitive Networking (TSN), task force recently released a set of standards that augment standard Ethernet switches providing determinism and low latency communication ideal for OT applications.

A key question that needs defining is what constitutes a deterministic system or determinacy in the context of networking and communication? We can establish that it does not mean increased throughput or reduced latency. We conclude that a deterministic system is a system in which *no* randomness is involved and therefore can be modeled or characterized to produce the same output from the same starting conditions (i.e., initial state).

In this chapter, we implement and utilize Cyclic Queuing and Forwarding (CQF), and the Paternoster scheduling mechanism on a standard industrial control closed-loop unidirectional ring topology. Furthermore, we study and analyze the scheduling

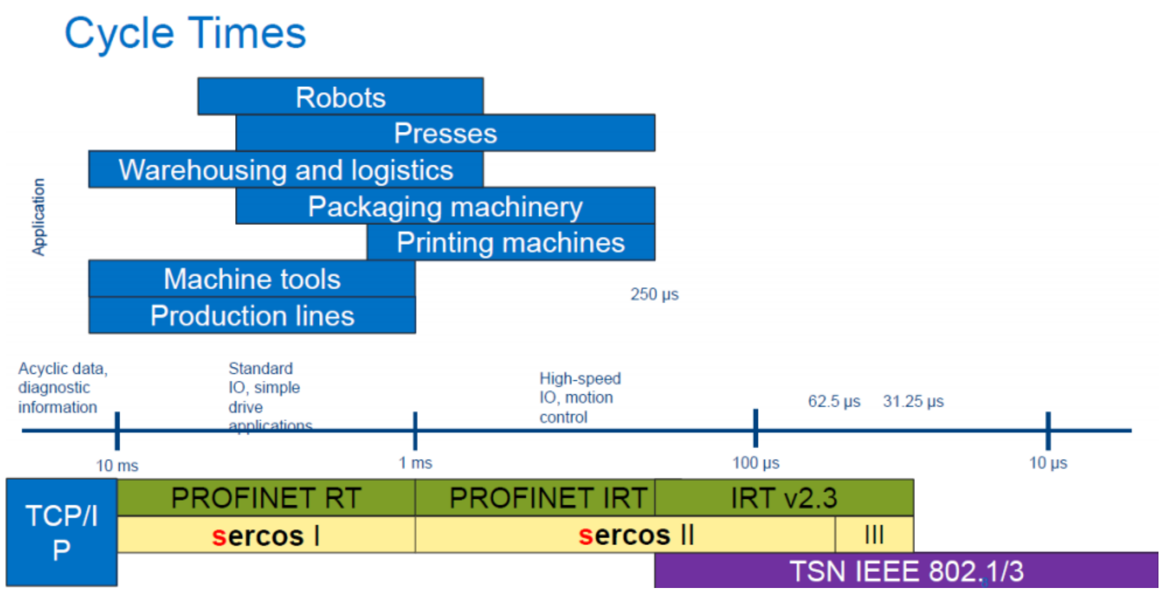


Figure 5.1: Industrial QoS between different protocols related to OT applications.

mechanism's efficacy for different propagation delays and traffic intensity emulating large-scale networks with both sporadic and periodic traffic. The main goal is to ensure the deterministic attributes governed by the scheduling mechanism used in ensuring proper TSN QoS.

5.1.1 Related Work

Groundwork on CQF, which was also previously known as Peristaltic shaper, was conducted by Thangamuthu et al. [407]. Moreover, Thiele et al. [412] have conducted a theoretical analysis of the blocking factors for CQF and TAS. Zhou et al. [460, 461] have conducted a simulation study on Paternoster, but only for one-hop transmission (they did not consider a full multi-hop network). In [64] authors model a routing problem in TSN as an ILP. In [339] authors propose a joint optimization problem of routing and scheduling in one step. In [198] authors propose a bandwidth optimization based queuing technique.

5.1.2 Contributions

We make the following contributions:

i) We implement both standard CQF and Paternoster scheduling models.

ii) We comprehensively evaluate and analyze the two models for both sporadic and periodic sources with cross-interference of BE traffic and varying propagation delays emulating large-scale networks.

iii) We elucidate recommendations and limitations of each model according to the results.

5.1.3 Organization

This chapter is organized as follows. Section 5.2 provides the necessary background information to understand the mechanisms of CQF and Paternoster. Section 5.3 describes and illustrates the simulation environment, network/traffic model, and shows the results collected and metrics involved in analyzing the scheduling mechanisms. Finally, Section 5.5 concludes the chapter.

5.2 Background: IEEE 802.1 Time Sensitive Networking

This section provides a brief background overview on TSN standardization, specifically CQF and Paternoster. TSN is a suite of standards aimed at applying deterministic behavior to the traditional best-effort Ethernet standards.

5.2.1 CQF

The published IEEE 802.1Qch (CQF) [26] standard proposes to coordinate enqueue/dequeue operations within a switch in a cyclic fashion. Fig 5.2 shows a simplified illustration (or snapshot) of the CQF mechanism. The CQF cyclic operation

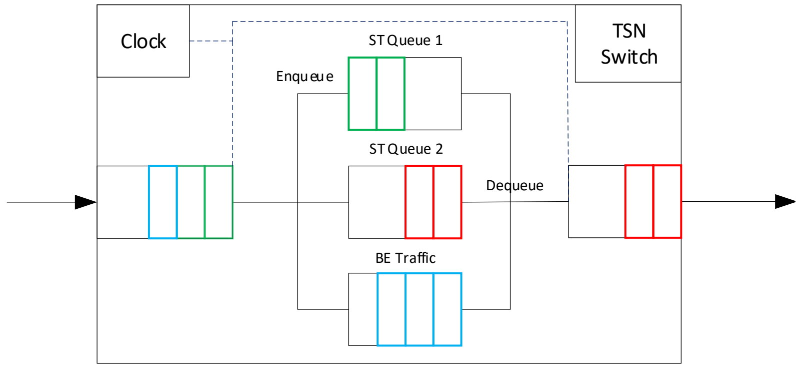


Figure 5.2: Simplified CQF Mechanism in TSN switch with two ST queues

results in an easily calculable latency bound governed by the chosen Cycle Time and the number of end-to-end hops between communicating parties. In CQF, time is divided into slots or intervals. For a given traffic class, two queues are used to enable the cyclic property. Frames arriving in interval x will be transmitted in interval $x+1$. Similarly, frames arriving in interval $x+1$ are transmitted in interval $x+2$, and so on. The maximum and minimum frame delay bounds in CQF with H and CT representing the number of hops and cycle time duration, respectively, are

$$D_{Max} = (H+1) \times CT \quad (5.1)$$

$$D_{Min} = (H-1) \times CT. \quad (5.2)$$

Two queues are used to handle enqueue and dequeue operations in separate time intervals. For example, frames arriving in even intervals will be enqueued in one queue, while the frames that were enqueued during the previous interval will be transmitted from the other queue. In CQF, a frame sent by an upstream switch in cycle x must be received by the downstream at cycle x, i.e., the propagation delay must be less than the selected cycle time. Therefore, the cycle time is constrained by the link distance (network scale in general). Essentially, the smaller the network size, the easier it is to guarantee the TSN QoS by CQF. Additionally, CQF has a

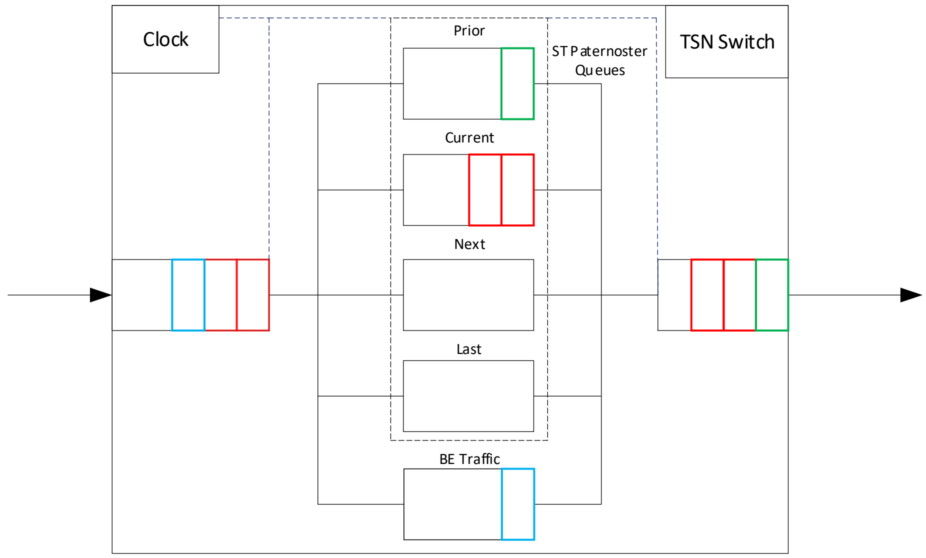


Figure 5.3: Simplified Paternoster Mechanism in TSN switch

few challenges that limit its viability, such as *i*) accurately determining the appropriate cycle time, and *ii*) cycle duration misalignment where due to processing and transmission delays, a frame can be received in the wrong cycle (i.e., be placed in the wrong outbound queue).

5.2.2 Paternoster

The Paternoster algorithm is a proposed enhancement by Seamen et al. [382] to standard CQF. Fig 5.3 shows a simplified illustration (or snapshot) of the Paternoster mechanism. Paternoster provides bounded latencies and lossless service for flows that are successfully registered across the network without a time synchronization requirement. For each egress port, the Paternoster protocol defines a counter for stream reservation and four output queues (*prior*, *current*, *next*, *last*), whereby all switches under Paternoster operate under an *epoch* timescale where the start/end of the *epochs* are not synchronized with other switches. In each *epoch* window, frames in the *prior* queue are transmitted first until all frames are transmitted. Once the *prior* queue is depleted, the *current* queue is selected for transmission until the end of the current epoch. While frames are being transmitted from the *prior* and *current* queues, received frames are enqueued in the *current* queue until the bandwidth capacity is reached for the current *epoch*. Any additional frames are enqueued in the *next* and *last* queues in a similar manner, i.e., until the reservation capacity for the current epoch is reached while additional frames are dropped if the *last* queue is completely reserved for the current *epoch*. Note that all ST traffic streams are given guaranteed bandwidth, while BE traffic is given the leftover bandwidth. When a new epoch starts, the previous *current* queue operates as the *prior* queue while the *next* and *last* queues become the *current* and *next* queues, respectively. The previous *prior* queue (which should be empty, and if not, we purge all the contents and register the packets as lost) becomes the new *last* queue. The Paternoster operation repeats at each *epoch*, while the four queues alternate during each *epoch*. While four queues are expected to be sufficient for many LDN scenarios, very long propagation delays may necessitate that another queue into the past and another queue into the future are added, for a total of six queues [382].

In summary, the Paternoster approach uses four queues that alternate every epoch (also known as cycle) using only frequency synchronization, i.e., the epoch duration is the same across the nodes. In contrast to CQF, the Paternoster approach gives up some delay predictability in exchange for not requiring clock synchronization and for reducing the average delay.

The evaluations reported for Paternoster in this chapter considered random time shifts of equal-duration cycles in the switches. In particular, each switch had an independent uniformly distributed time shift between zero and the cycle time with respect to a common time base.

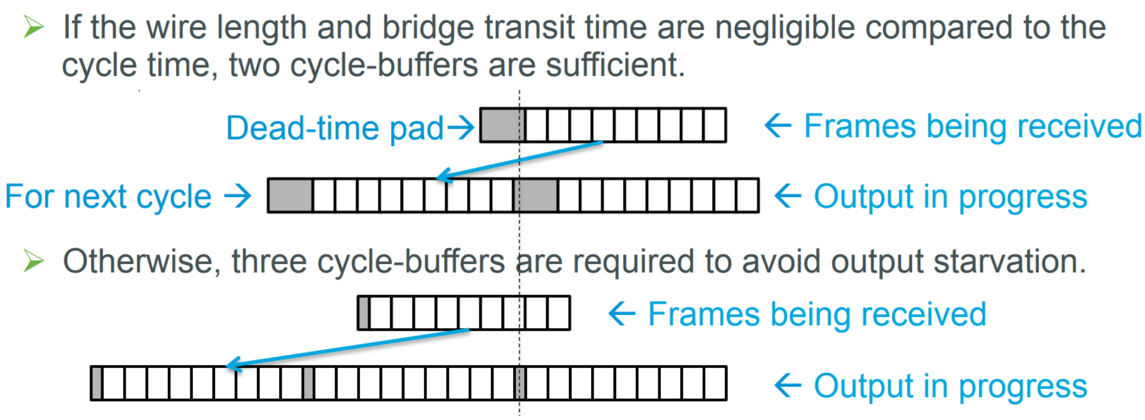


Figure 5.4: Standard vs. 3-Queue CQF example illustration.

5.2.3 *3-Queue CQF*

A critical requirement for the standard CQF is that a frame sent during a cycle has to be received during the same cycle such that the worst case delay is constrained by the cycle time and hop count. The 3-Queue CQF has been proposed to handle networks that have propagation delays that approach and exceed the cycle time [171].

When traffic arrives in the wrong cycle, a third queue is needed to handle such traffic so as to prevent disruption to traffic that conforms to the requirement for CQF. This is illustrated in Fig. 5.4. Though the general or principle idea behind the 3-Queue CQF is interesting, some questions remain that need solutions so that a full-fledged implementation and evaluation is possible.

How would the third queue (or waiting queue) be used in such a environment without affecting other traffic? Every cycle is needed to send traffic from an egress port, especially for periodic traffic. Therefore, when should traffic that gets enqueued into the waiting queue be dequeued? How would the dead-time be calculated or computed? If the propagation delay exceeds the cycle time for all periodic traffic, wouldn't this delay be consistent for all traffic and therefore act as a constant in the

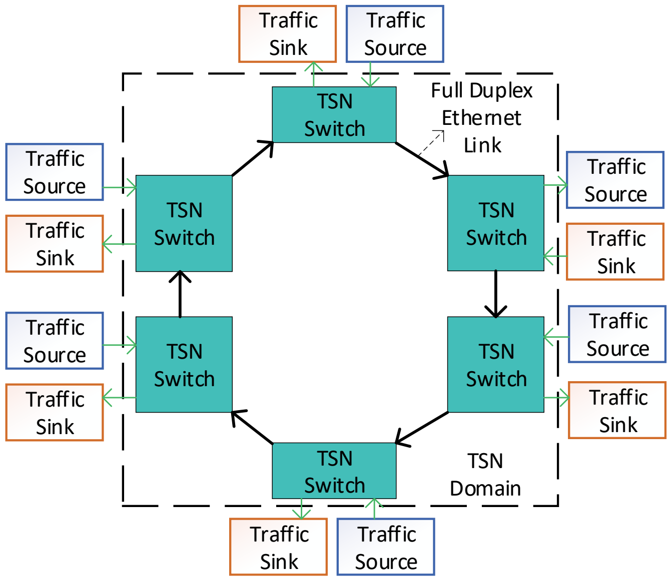


Figure 5.5: Unidirectional Ring Topology

overall worst case delay?

Our tests in Section 5.3 indicate that a propagation delay of 50μs for a 50μs cycle time (where ST is given 25μs) gives twice (from 200μs to 400μs) the maximum or worst case delay than a propagation delay of 25μs. We can hypothesize that for sporadic sources, we can use the strict priority scheduler between the dequeuing queue and the waiting queue, so that any traffic in the waiting queue can be transmitted if no traffic is waiting in the dequeuing queue.

5.3 Performance Evaluation

5.3.1 System Overview and Simulation Setup

This section describes the simulation setup and model for both standard CQF and the Paternoster scheduling protocols. Furthermore, the topology and simulation scenarios will be presented. Throughout, we employ the OMNet++ [420] simulation environment.

Table 5.1: Simulation Parameters

Key	Symbol	Value
Simulation duration	Sim_{limit}	100 seconds
Initialized cycle time	GCL_{CT}	50 μs
Initialized gating ratio	ST^{R}_{init}	50% (i.e., 25 μs)
Total streams	γ	6
Stream duration	τ	100 seconds
Link propagation delay	α	500 ns, 25μs, 50μs
Number of frames/packets per cycle for periodic traffic	π	$1 - 40$
Sporadic traffic intensity	ρ_I	$0.1 - 2.0$ Gbps
ST sources	S	6
ST stream hop count	TTL	3
Hurst parameter	H	0.5
Queue size	Q_{size}	512 Kb
Link Capacity	R	1 Gbps

Network Model

The topology used to test the CQF and Paternoster scheduling mechanisms is modeled as shown in Fig. 5.5. Table. 5.1 shows the simulation parameters used in testing CQF and Paternoster in the unidirectional ring. Each switch-to-switch link operates as a full-duplex Ethernet link with a capacity (transmission bitrate) $R = 1$ Gbps. Each switch can act as a gateway for a number of traffic sources and one sink. The propagation distance is varied between 500 ns and 50μs. Each switch operates either CQF or Paternoster scheduling between switch to switch egress ports.

Traffic Model

We consider periodic (pre-planned) traffic and sporadic self-similar Poisson ($H = 0.5$) traffic for ST traffic, while solely sporadic traffic for BE. Six sources are used to generate traffic each attached to a TSN switch gateway. A single stream is initiated at the start of the simulation for the entire duration of the simulation. Each frame/packet's destination address is specified by the switch to switch hops around the ring, which

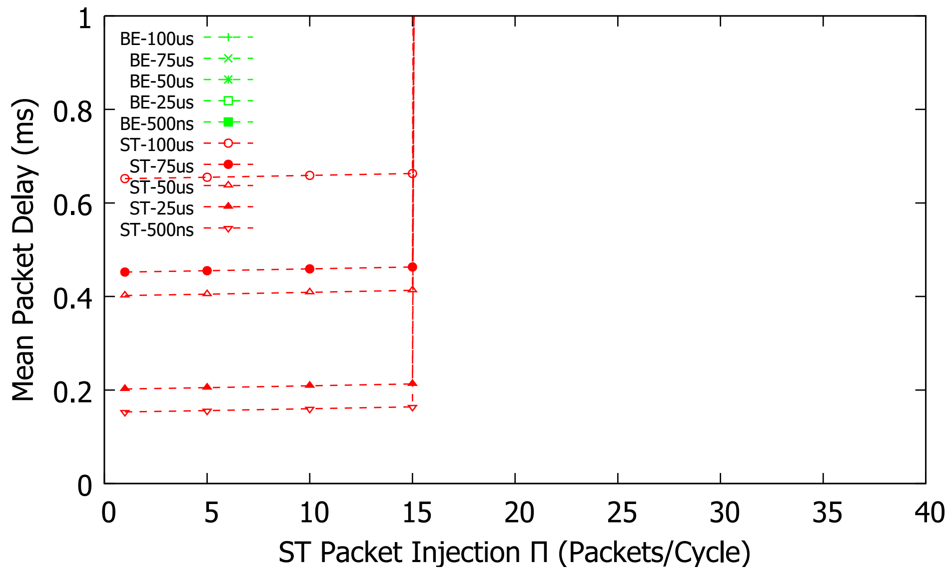


Figure 5.6: CQF mean packet delay for periodic ST traffic sources

is predefined to 3 hops as shown the Table. 5.1. The size of a frame is 64 bytes for ST and 580 bytes for BE. The traffic intensity is varied in each simulation run where the ST injection rate $(1 - 40)$ is used for periodic ST traffic and the ρ_I traffic intensity. Note that the BE traffic intensity in periodic ST source tests is set to 1.0 Gbps.

5.3.2 *Cyclic Queuing and Forwarding (CQF) Evaluation*

Periodic

Fig. 5.6 and Fig. 5.7 show the mean and maximum/minimum delays, respectively, for periodic ST traffic using CQF based scheduling under different propagation delays. The BE traffic intensity is set to a constant value of 1.0 Gbps and exhibits the same mean delay of 28 ms for all ST injection rates (due to TAS isolation). As the ST periodic traffic intensity increases, both the mean and maximum delays are constant up to an ST packet injection rate of $\pi = 16$ packets/cycle, which causes an immediate spike in both mean and maximum delays due to over-utilizing the link resources, no preventive measures of admission control policies, and none-adaptive

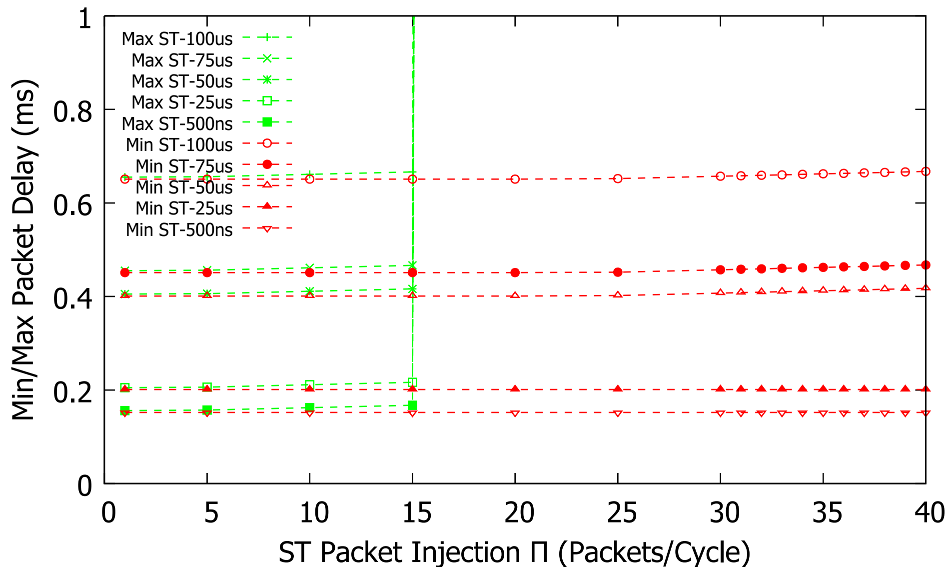


Figure 5.7: CQF maximum/minimum packet delay for periodic ST traffic sources

TAS slot ratios that change according to the bandwidth consumption. Since CQF gives a simple method of calculating the worst-case end-to-end delay of a stream (shown in Section 5.2), the maximum delay shown in Fig. 5.7 illustrates that for the CQF mechanism, the delay is a function of and bounded by the number of hops and GCL time. More precisely, since the cycle time (GCL) is set to 50μs, the total worst case delay for a three hop stream is $50 \cdot (3+1) = 200$ μs which is shown in both figures (except for networks initialized with 50μs propagation delays).

Furthermore, as the propagation delay is increased and approaches the cycle time, the end-to-end delay approaches the CQF worst-case delays, i.e., the maximum/minimum and mean delays are bounded and characterized by the number of hops and cycle time. Note that the ST gating ratio (due to TAS operating in the egress port) is half the cycle time (25μs), while the BE traffic is allocated the rest of the transmission time opportunity. When the propagation approaches the cycle time, it is considered twice the ST gating ratio which translates to twice the worst case delay of 400μs.

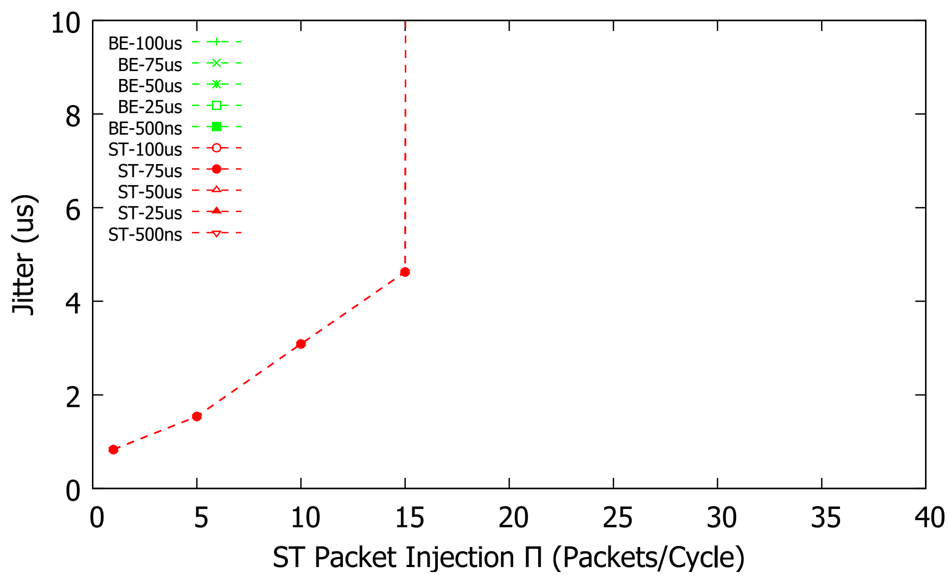


Figure 5.8: CQF jitter for periodic ST traffic sources

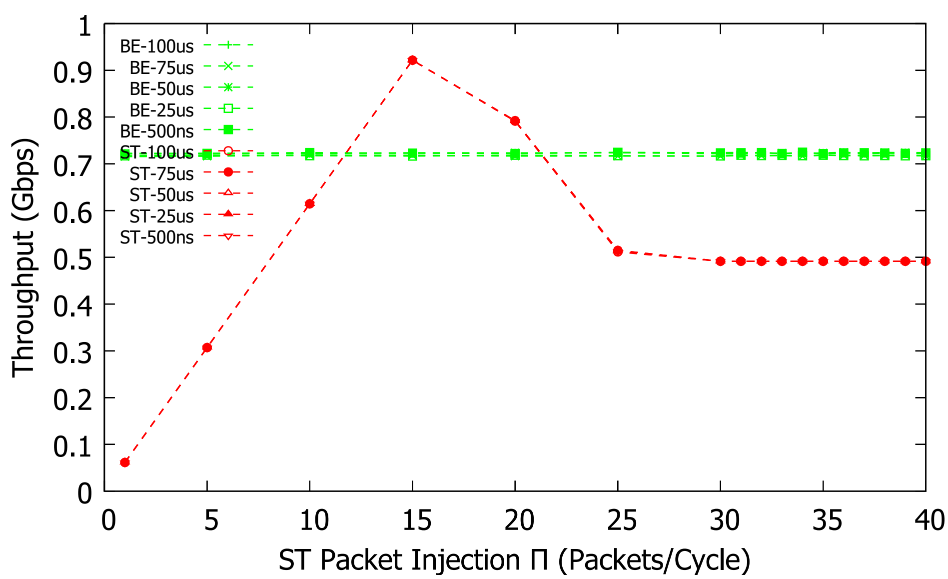


Figure 5.9: CQF average throughput for periodic ST traffic sources

Fig. 5.8 shows the network jitter between a source and sink. The jitter is calculated as the standard deviation of the mean delay. As shown in the figure, jitter is around 4μs but then spikes very quickly due to over-utilization of resources and consequently causing congestion.

Fig. 5.9 and Fig. 5.10 show the throughput and loss respectively. As the ST

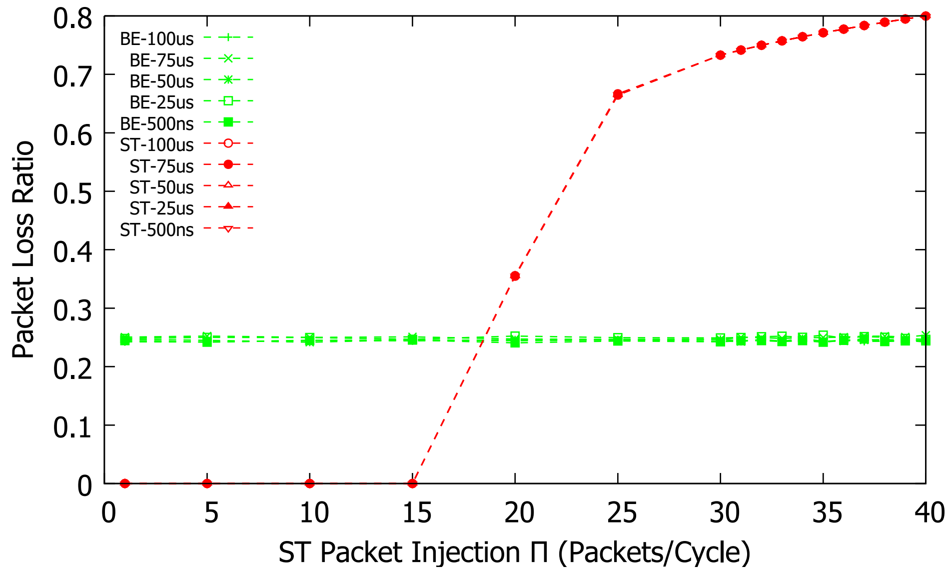


Figure 5.10: CQF loss packet ratio for periodic ST traffic sources

injection rate is increased, the throughput increases linearly. However, the throughput sharply declines due to the congestion caused by injecting more bits than the ST slot can handle within each cycle (16 packets or 1892 bits can be sent by one source per cycle into the network). In terms of packet/frame loss, BE traffic experiences more or less the same loss due to having the same traffic intensity (1.0 Gbps) in all runs (around 0.25% loss). ST, on the other hand, stays constantly at 0 loss until $\pi = 16$ rate. Due to congestion, the loss increases sharply with higher ST traffic intensity.

Sporadic

Fig. 5.11 and Fig. 5.12 show the mean and maximum/minimum delays respectively for sporadic ST sources. In contrast to periodic ST traffic sources, the use of sporadic traffic with uncontrollable bursts can severely degrade the operation of CQF as shown in both figures. The mean delay for both sporadic traffic classes quickly increases as the traffic intensity increases. The TSN QoS (bounded maximum/minimum delays, zero loss, and low jitter) are violated mainly due to the uncontrollable bursts in the

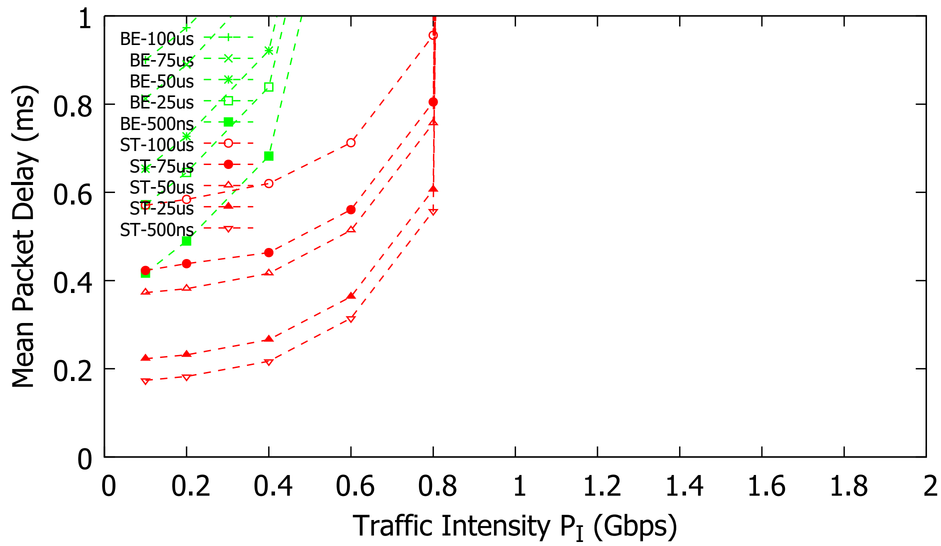


Figure 5.11: CQF mean packet delay for sporadic ST traffic sources

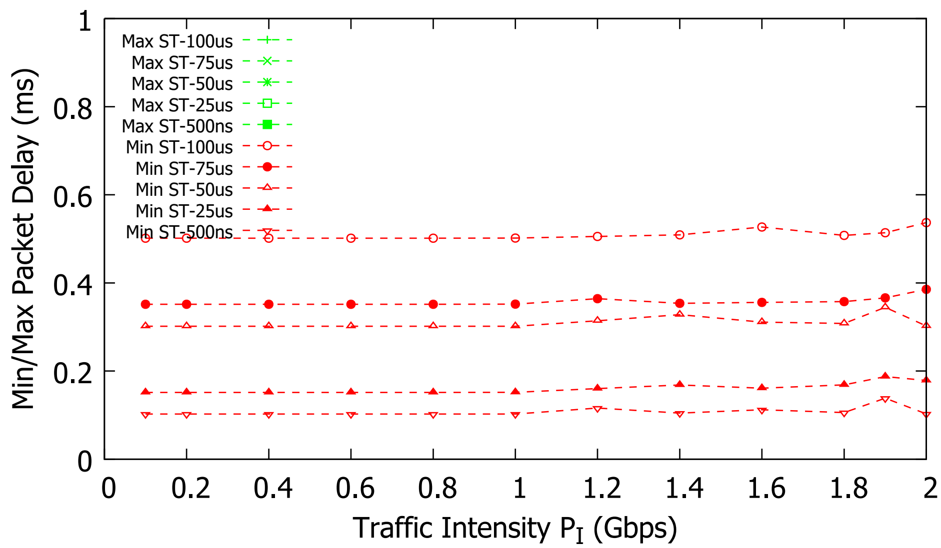


Figure 5.12: CQF maximum/minimum packet delay for sporadic ST traffic sources

sporadic ST sources.

Fig. 5.13 shows the network jitter between source and sink. Similar to the mean and maximum/minimum delays figures, the jitter is much higher compared to the periodic jitter results.

Fig. 5.14 and Fig. 5.15 show the throughput and packet loss for sporadic ST

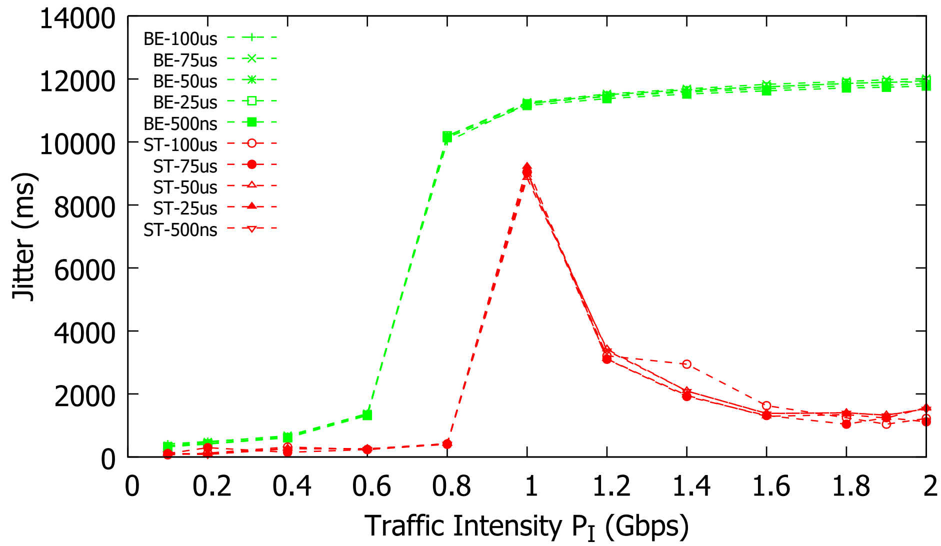


Figure 5.13: CQF jitter for sporadic ST traffic sources

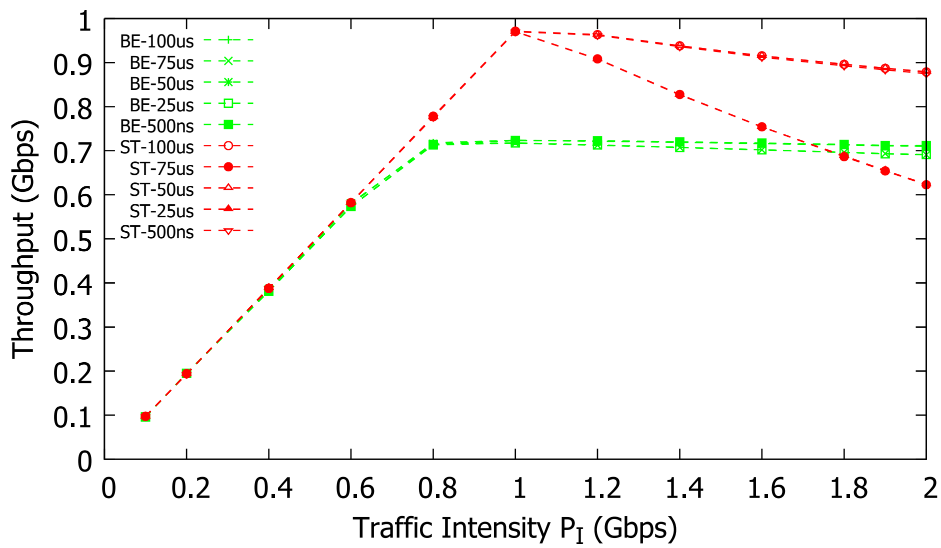


Figure 5.14: CQF average throughput for sporadic ST traffic sources

sources. Throughput increases as the traffic intensity increases up to traffic intensity, $\rho_I = 1.0$, which causes a large drop in throughput due to congestion in the network. Similarly, the packet loss shows large increase after $\rho_I = 1.0$ for ST, while BE starts to lose more packets earlier.

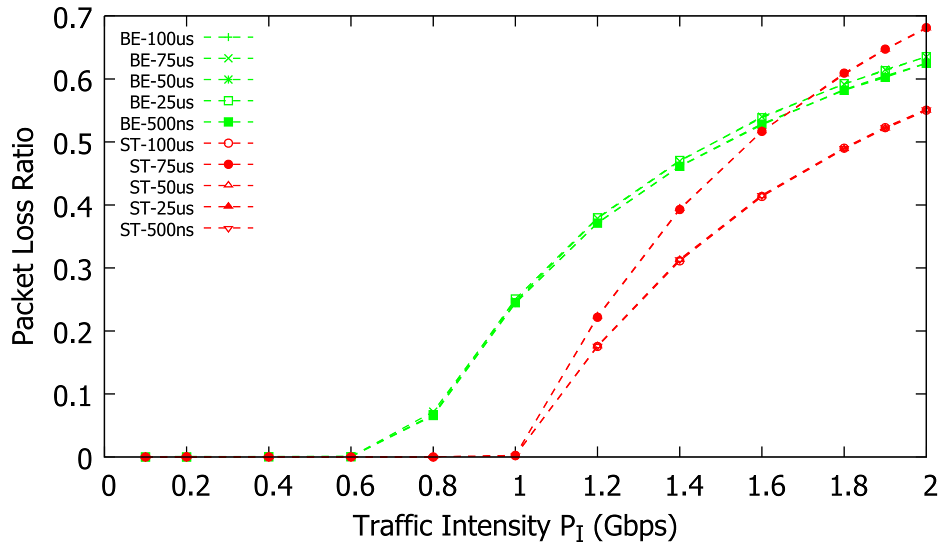


Figure 5.15: CQF loss packet ratio for sporadic ST traffic sources

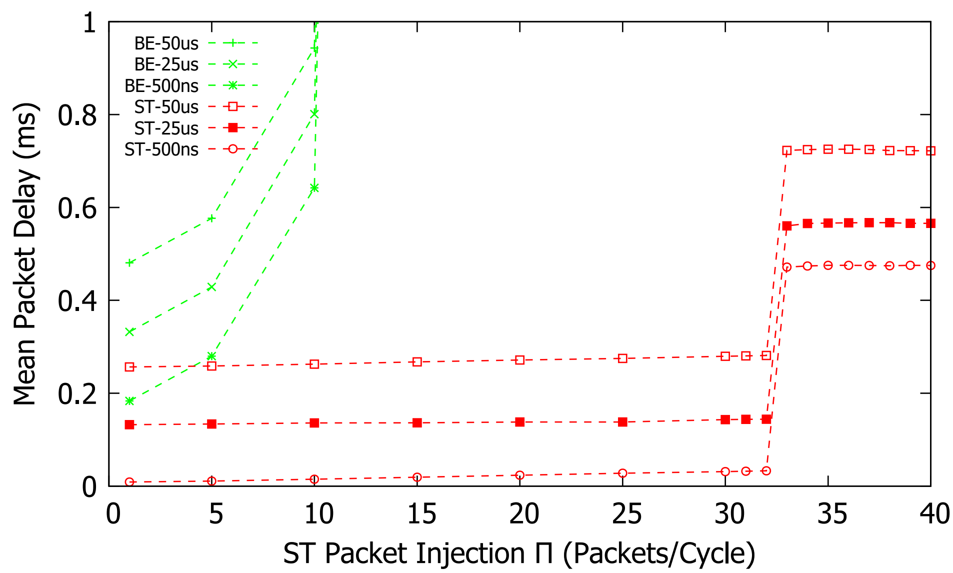


Figure 5.16: Paternoster mean packet delay for periodic ST traffic sources

5.3.3 Paternoster

Periodic

Fig. 5.16 and Fig. 5.17 show the mean and maximum/minimum delays for periodic ST sources and sporadic BE sources using switches that operate Paternoster. Initially,

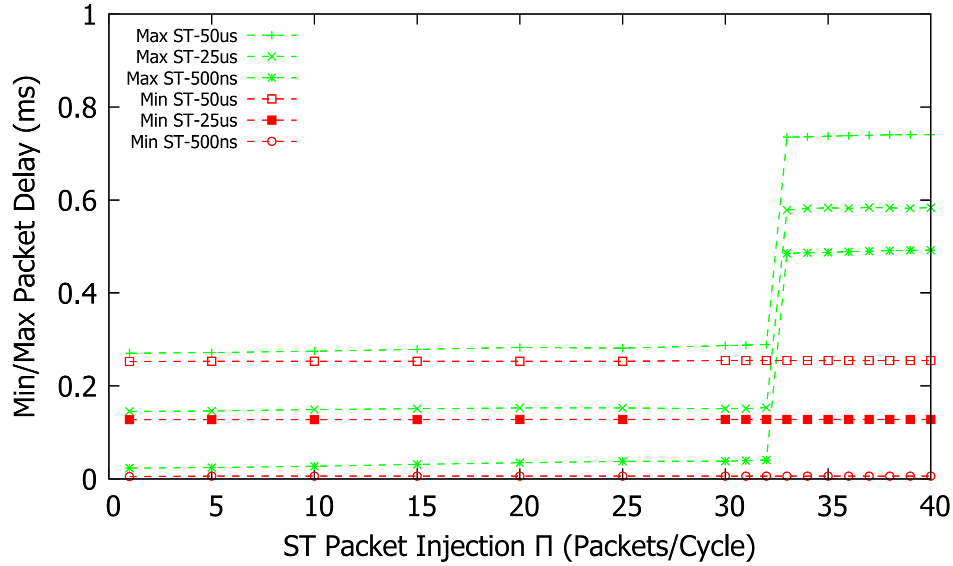


Figure 5.17: Paternoster maximum/minimum packet delay for periodic ST traffic sources

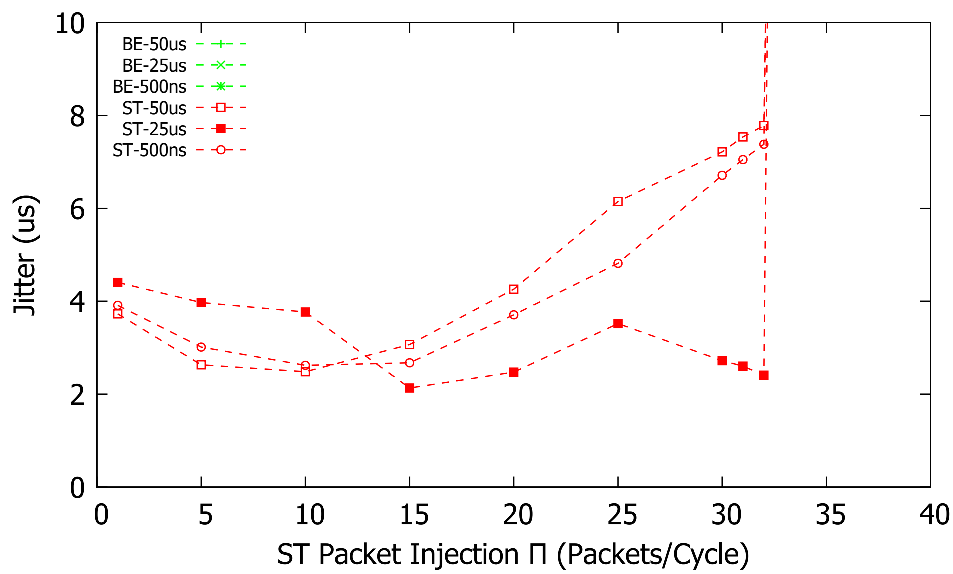


Figure 5.18: Paternoster jitter for periodic ST traffic sources

we observe from Fig. 5.16 that the mean delays for ST are lower when compared against the CQF performance. However, BE get starved by ST when $\pi = 33$ since all transmission opportunities during an epoch/cycle are consumed by ST. ST's delay stabilizes after the spike due to purging the prior queue in Paternoster.

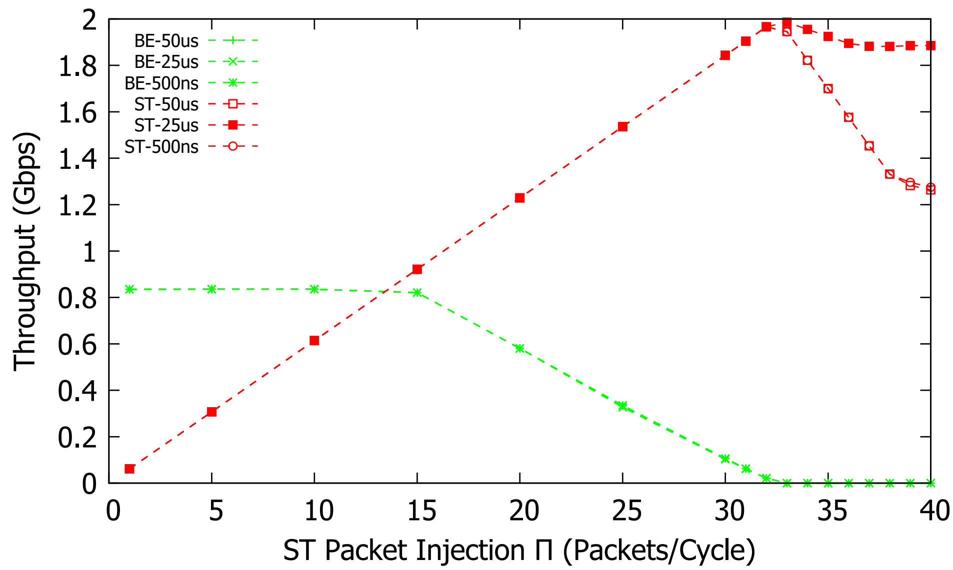


Figure 5.19: Paternoster average throughput for periodic ST traffic sources

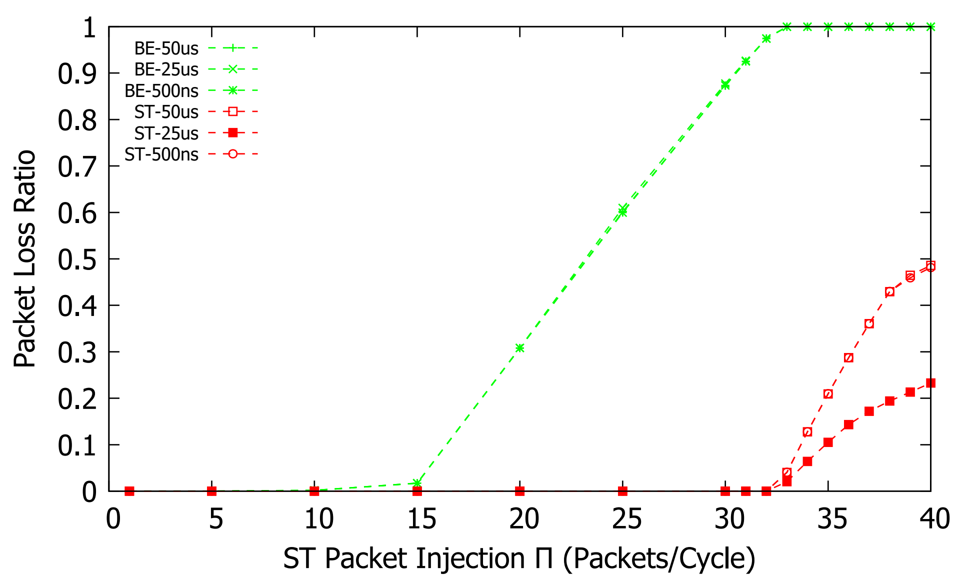


Figure 5.20: Paternoster loss packet ratio for periodic ST traffic sources

Fig. 5.18 shows network jitter. While the jitter is comparable to the CQF protocol, the varying changes as the traffic intensity increases show the unpredictability in Paternoster compared to CQF.

Fig. 5.19 and Fig. 5.20 show the throughput and packet loss experienced at the sink. At $\pi = 16$, we see the BE traffic (with $\rho_I = 1.0$ Gbps) starts to drop proportional

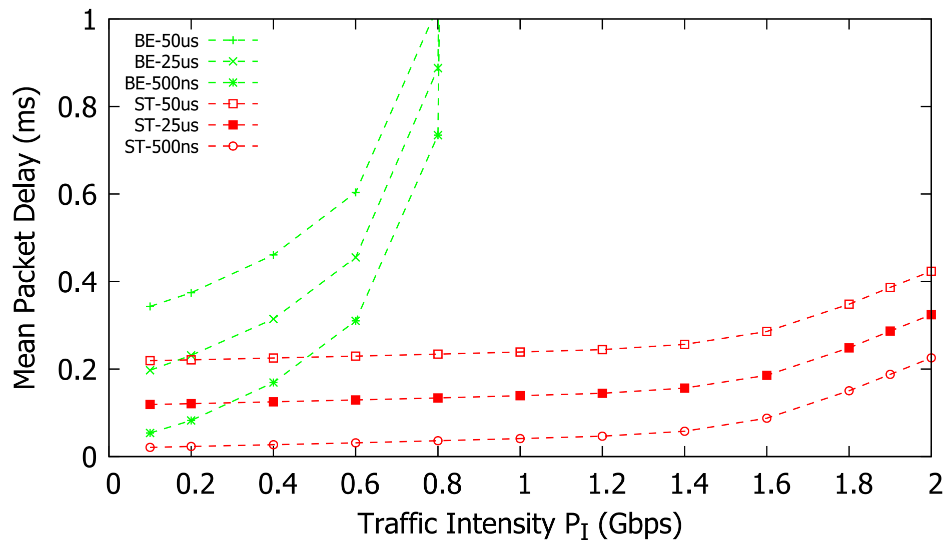


Figure 5.21: Paternoster mean packet delay for sporadic ST traffic sources

to the increase in ST before being starved at $\pi = 33$. Any additional increase causes packet loss and congestion which drops the throughput to below optimum levels. Similarly, the loss shows a complement of the throughput and at $\pi = 16$, the BE traffic starts to accumulate loss linearly as the traffic intensity keeps increasing.

Sporadic

Fig. 5.21 and Fig. 5.22 show the mean and maximum/minimum delays for sporadic ST sources using Paternoster. Compared to the periodic results, the Paternoster performs better for sporadic traffic sources. However, accurately predicting the worst case delays still remains difficult compared to CQF. Since strict priority scheduling is employed at the egress port, the maximum or worst case delays are highly unpredictable compared to CQF and also Paternoster under the periodic ST sources.

Fig. 5.23 shows the network jitter. Since strict priority scheduling is used to arbitrate between competing traffic classes, BE (lower priority) can block ST (higher priority) if the port is currently transmitting BE traffic when ST waits for the trans-

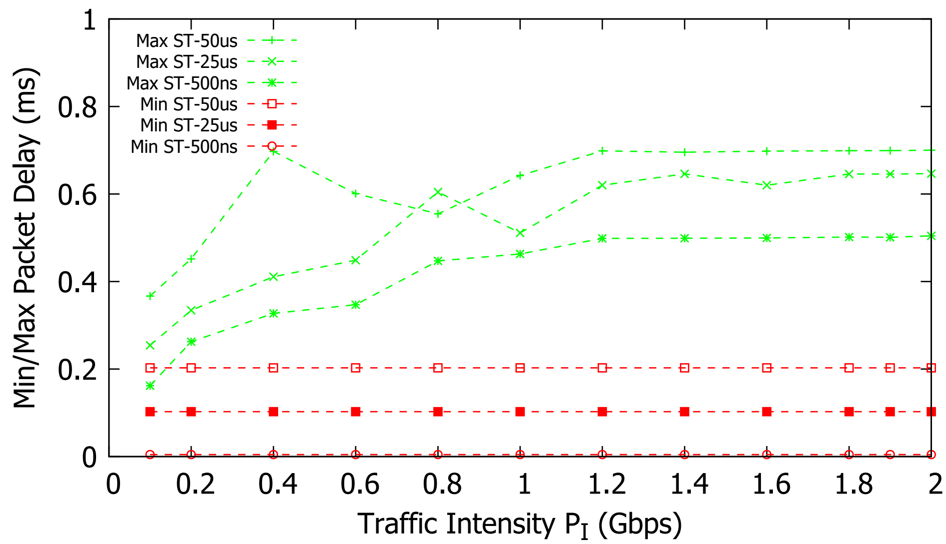


Figure 5.22: Paternoster maximum/minimum packet delay for sporadic ST traffic sources

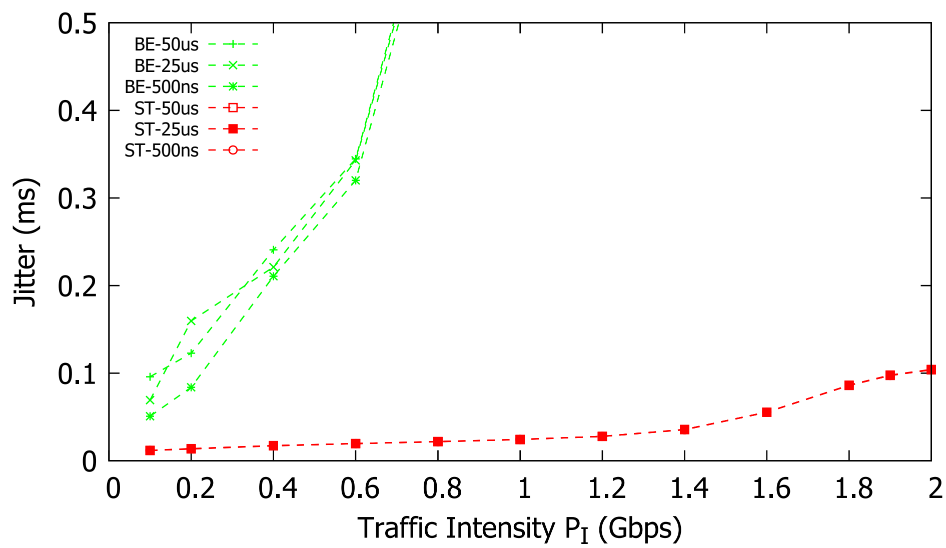


Figure 5.23: Paternoster jitter for sporadic ST traffic sources

mission to finish. This causes higher unpredictable jitter, which is seen Fig. 5.23.

Fig. 5.24 and Fig. 5.25 show the throughput and loss at the sink. Similar to the periodic case, BE traffic throughput decreases after 1.0 Gbps while its loss increases. In contrast, the ST throughput continues to increase, but ST does experience some

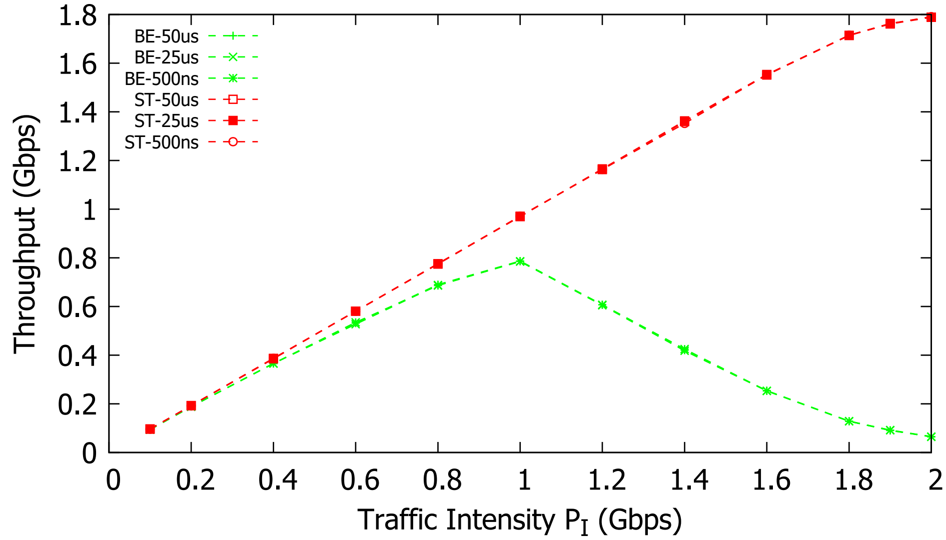


Figure 5.24: Paternoster average throughput for sporadic ST traffic sources

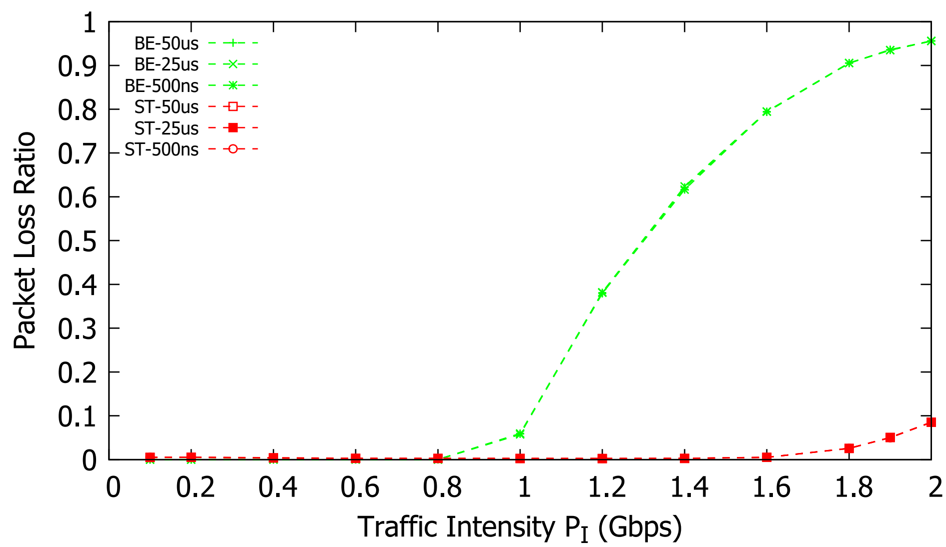


Figure 5.25: Paternoster loss packet ratio for sporadic ST traffic sources

loss after about an injection rate π corresponding to 1.6 *Gbps*.

5.4 Comparison Analysis

Both the CQF and Paternoster scheduling algorithms attempt to enhance the data-link layer with some form of deterministic behavior. CQF coordinates the ingress and egress operations to provide bounded delays and zero traffic loss that conforms to its reservation, while Paternoster uses four queues per port to spread traffic burstiness/bunching and guarantees bounded delays for traffic that conforms to its resource reservation. In essence, Paternoster applies the concept of Credit-Based scheduling (CBS) [19] with the cyclic attribute of CQF.

Paternoster is considered an enhancement to CQF since it guarantees TSN QoS while removing the time synchronization requirement, i.e., an asynchronous scheduling protocol (though frequency synchronization is still needed to keep the cycle/epoch duration the same at all switches). While the results are favorable towards Paternoster in terms of minimizing mean and maximum delay, CQF is generally more predictable and therefore more deterministic than Paternoster, particularly for OT applications with critical QoS and hard deadline requirements. In particular, for periodic traffic and using the CQF protocol (with 50% gating ratio for ST, or 25μs transmission opportunity), all the streams with $\pi \leq 16$ have mean/maximum/minimum delays between 150μs and 200μs regardless of the path and cross traffic. This bounded delay guarantee is not easily predictable for Paternoster due to the fundamental loss of time synchronization between switches. Additionally, since Paternoster uses the strict priority scheduling at the egress, it contains elements of best-effort service which causes loss of determinism.

CQF provides complete isolation between ST and BE (due to TAS being used at the egress port) and as a result performs fairer in resource allocation between ST And BE, especially at high traffic intensity. Paternoster does not isolate traffic classes (though it does provide resource allocation) which can degrade the predictability and deterministic behavior for ST. This effect is evident in the periodic jitter results where the mean jitter at varying traffic intensities for CQF is monotonically increasing (up to a bounded jitter value), while the jitter measurement for Paternoster is highly erratic.

In terms of packet loss, both CQF and Paternoster guarantee zero loss for streams that conform to their reservations. However, Paternoster does perform slightly worse when packets remain in the *prior* queue and a cycle change occurs causing the *prior* queue to be purged of its content. This rarely happens with periodic traffic sources, but can occur more frequently with sporadic traffic sources where a switch can abruptly receive a large number of traffic before a cycle change.

Since Paternoster uses strict priority scheduling at the egress, Paternoster achieves significantly better delays due to having more transmission opportunities than the CQF protocol, i.e., CQF's use of TAS at the egress divides and isolates the transmission opportunities and does not adapt these opportunities to varying changes in traffic intensity. Moreover, the main issue with CQF in guaranteeing QoS for sporadic ST streams is that the burst usually is much greater than the allowable bandwidth per cycle (the transmission opportunities given). Applying ingress policing and admission control (either centralized or distributed) can mitigate this issue by using control signals and negotiating network resources and QoS to streams that request it (this has been investigated in [309] switches utilizing TAS only).

5.5 Conclusions and Future Work

A performance evaluation has been conducted in this chapter to compare standard CQF and Paternoster. Since Paternoster uses more queues, i.e., more complexity, and provides less deterministic behavior compared to CQF, CQF performs better for OT applications with hard real-time requirements. While Paternoster performs worse than CQF in ensuring deterministic properties, it provides a relaxed traffic predictability in networks that do not have time synchronization. While this performance evaluation used statically defined traffic slots in the cycle (for CQF/TAS), an adaptive method (Adaptive TAS [312]) can be used to accurately determine the needed slot duration to optimally service registered traffic classes (in this case, BE and ST). Using TAS in Paternoster at the egress port between ST and BE instead of strict priority scheduling is another recommendation that can reduce the jitter for streams. While the tests in this evaluation involved uniform link transmission and propagation delays, a more complex problem that involves different link transmission and propagation delays is an interesting direction for future research since it can cause cycle misalignment between adjacent ports according to the standard.

In the wider context of QoS networking and related applications, QoS oriented routing approaches, e.g. [112, 191] should be investigated. Furthermore, deterministic networking should be studied in the context of emerging multiple-access edge computing (MEC) [140, 178, 286, 385, 428], in particular MEC settings for low-latency applications [149, 439, 454]. As an alternative approach to coordinating the reconfigurations, emerging softwarized control paradigms, such as software defined networking can be explored [55, 136, 138, 226, 374]. Regarding the reliability aspects, a potential future research direction is to explore low-latency network coding mechanisms, e.g., [31, 115, 150, 177, 270, 275, 438], to enhance networking protocols targeting reliable low-latency communication.

Chapter 6

DESIGN AND EVALUATION OF FAULT TOLERANT ASPECTS IN TIME SENSITIVE NETWORKING USING MACHINE LEARNING

6.1 Introduction

As operational and informational networks get increasingly more interconnected and complex requiring higher bandwidth and Ultra-Low Latency (ULL), traditional control networks that deploy semi-proprietary communication for safety critical systems cannot satisfy the required specification needed for appropriate use, particularly the fault tolerance and interoperability requirements. Fault tolerance is a critical part of industrial networks involving safety critical systems. A general principle to enable fault tolerance is to introduce redundancy, i.e., send multiple copies of frames (or packets) across a network (e.g., High Availability Redundancy Protocol (HSR) and Parallel Redundancy Protocol (PRP)). In the event of a failure of an intermediate node (bridge or switch) or link, the delivery is still possible from redundant packets sent upstream. Additionally, Electro Magnetic Interference (EMI) and Radio Frequency Interference (RFI) from operational equipment in factory and production floors can cause transmission errors. Using optical networks can reduce these types of failures but comes with a large cost expenditure. Therefore, using modern standardized Ethernet and communication links that dominate in the informational network (none-real-time communication context) at the operational level is imperative.

IEEE 802.1CB [27] coupled with IEEE 802.1Qcc [24] allows for bridges to replicate streams across different ports at one end and eliminate redundant copies at another end. The path computed and taken is managed by IEEE 802.1Qca [23]. While the specifications provided presents a general architectural view of managing and using such a framework, testing such a framework is yet to be investigated. For instance, high priority flows could have reservations of dedicated resources, while low priority flows could share a common reserved resource. The dedicated resources would enable the instantaneous recovery of the high priority TSN flows; albeit, at the expense of a slight reduction of the overall network efficiency due to stream redundancy. In the event of failure for a low priority traffic flow, the connection could be re-established with a new flow path considering that the flows can tolerate delays on the order of the connection re-establishment time. Centralized Software Defined Networking (SDN) management can also provide the flexibility of dynamic path computation and resource reallocation in the event of failures. This chapter investigates the wide range of trade-offs and optimizations that arise with reliability through frame replication and elimination. Additionally, an empirical preliminary evaluation is conducted and computer simulations showcases several approaches to fault tolerance in the TSN context. Moreover, a Machine Learning (ML) model is trained and used as a viable alternative to perform redundancy in TSN with random fault scenarios.

6.1.1 Motivation

IEEE 802.1 Time Sensitive Networking (TSN) provides a standardized framework of tools for providing deterministic reliable ULL [170, 313]. Fault tolerance in TSN is an imperative aspect that needs a thorough evaluation on trade-offs involving network utilization, packet/frame loss, useful bandwidth and throughput in the context of FRER. The general idea behind FRER is to use spatial redundancy that tags and copies frames belonging to specific streams and transmits them across disjoint paths. However, the main problem of where to start replicating the streams and duration of the replication process needs to be investigated. This chapter proposes a Machine

Learning intelligent approach that determines where to replicate from and the approximate duration given collected network conditions at runtime using a pre-trained classification model.

6.1.2 Related Works

Generally, fault tolerance in traditional communication networks involve variants of Automatic Repeat Requests (ARQ) that uses a sliding window to track packet order and respond with (negative) acknowledgment. However, for real time systems, this approach is not feasible since timely delivery is needed and retransmission is too costly in such scenarios, e.g., Audio and Video delivery. In the context of TSN, there are numerous literature that investigates reliability mainly through the lens of Path Computation and routing, and packet replication and redundancy. Generally, reliability is classified by two main approaches, *i*) seamless (proactive) and *ii*) none-seamless (reactive). The seamless approach employs protocols that apply redundancy without path switchover and reconfiguration. In contrast, the none-seamless approach employs redundancy after an interruption is detected which results in Ethernet frames being dropped during path switchover (i.e., the recovery time, t_{rec}).

One of the early studies in fault tolerance in TSN (or Audio/Video Bridging (AVB) when the paper was published) is by Kleinberg et al. [233, 234]. Kleinberg et al. notes that the redundancy protocols (e.g., Spanning Tree Protocol, Rapid Spanning Tree Protocol, Media Redundant Protocol, etc.) largely depends on the application requirements and financial constraints. Therefore, fault tolerance in AVB (and TSN) needs to be flexible according to the target application. The fault tolerance protocol proposed in AVB networks [234] decouples stream reservation and redundancy, i.e., create an abstraction layer to allow for more flexibility. In the decoupling approach, the stream reservation protocol registers and reserves the streams independently of the redundancy requirements. This decoupled approach allows for arbitrary redundancy protocols to be utilized. In contrast, the harmonized approach integrates establishment of the reservation and the redundancy requirements, e.g., the usage of 802.1CB with 802.1Qca and 802.1Qcc. Note however that in AVB networks, stringent timing requirements for traffic is more relaxed than in TSN since the main scheduling algorithm generally uses the Credit-Based Shaper which provided soft-real time guarantees.

Building on Kleinberg et al. investigation, Kehrer et al. [225] compares the two main concepts to fault tolerance in TSN, *i*) redundancy protocols integrated into the main stream reservation protocol (harmonized or TSN approach), and *ii*) decoupling the redundancy protocol from the stream reservation protocol (decoupled approach). The qualitative comparative analysis presented in [225] shows that the harmonized approach, due to the tight coupling and enforced usage of 802.1CB for applications that have soft-real time deadlines, could lead to vendors opting out of using the TSN approach. However, the TSN approach has a distinct advantage when protocol overhead and bandwidth consumption are considered.

Focusing on the TSN approach, Alvarez et al. [51] highlights the limitation of 802.1CB in terms of considering transients faults in the transmission channels. 802.1CB mainly deals with spatial (or permanent faults, e.g., link breakage) and does not consider temporal (or transient) faults, e.g., frames that are received erroneously due to EMI. The solution, Proactive Transmission of Replicated Frames (PTRF), presented transmits redundant frames over the channel. The number of copies to be sent is dependent on a number of factors, 1) the loss probability, 2) the target reliability, and 3) the selected redundancy approach used. Two approaches are compared, *i*) a source based replication approach, and *ii*) bridge based replication approach. In the source based approach, the transmitter (or talker) creates the replicas and bridges only forwards them while the bridge based approach creates replicas at each bridge along the path. Additionally, the proposed solution is extended and analyzed further in [49] for mixed spatial and temporal redundancy. Another redundancy approach is presented that is similar to the bridge based solution but changes the number of replicated frames according to the reliability of the forwarding link using so called "error counters" that relates the bit error rate to the number of replicas needed. Highlighting the fact that the TSN approach is limited by node failures in addition to link failures, Alvares et al. [46] presents a fault tolerant architecture, called a replicated star topology, that identifies and creates replicas of nodes and links to reduce any potential error, e.g., single point of failure, etc. Finally, the PTRF solution is extended and analyzed further coupled with a simulation evaluation given in [47]. The environment is tested by injecting faults across a linear path and recording the tolerated scenarios. However, the real impact on reliability was not shown.

Danielis et al. [129] presents a FRER implementation model using the NetSTiNG TSN implementation [156]. A reliability model is first presented and analyzed which is then used to start the FRER process at each bridge on the stream's path. A model that selects frames for redundancy is given depending on the frame's priority (i.e., criticality) and the reliability of the bridges and links along the stream's path. While the results presented promotes the use of FRER, it does not provide any runtime reconfiguration or stress testing with high throughput and loss.

Since FRER uses disjoint paths to transmit multiple copies, the cases where non-disjoint paths are used can result in unintended frame elimination. This problem is investigated by Ergenc et al. [152]. A different path selection strategy is used called reassurance that selects paths between talker and listener such that their path overlap is minimal and the potential junction node (where the original and duplicated streams converge) is close to the receiver.

Utilizing Machine learning to detect faults in TSN has started to gain traction. Desai et al. [137] proposes a ML based configuration synthesis mechanism that optimizes bandwidth utilization based on metrics involving link probability to failure. The paper promotes the idea but no evaluation is shown. Similarly, Daniel et al. [128] proposes the use of deep learning as a strategy for fault detection in TSN specifically during the startup phase of the network. Both of these works involve ML during startup and synchronization phase while this chapter deals with the FRER process during runtime.

6.1.3 Contributions

We design and evaluate the performance of different approaches to fault tolerance in TSN. More specifically, a framework is designed and implemented in OMNet++ that given a topology can produce the needed training data for a supervised model to then use and develop a learning model that inferences where and when a fault is likely to happen given network data at runtime.

i) A TSN fault tolerance framework is implemented in OmNet++.

ii) An ML model is provided that interacts with the TSN framework that is queried on the FRER process

6.1.4 Organization

This chapter is organized as follows. Section 6.2 presents the implemented framework, while Section 6.3 presents the evaluation of the training and testing phase related to different fault tolerance approaches. Finally conclusions and future work are outlined in Section 6.4.

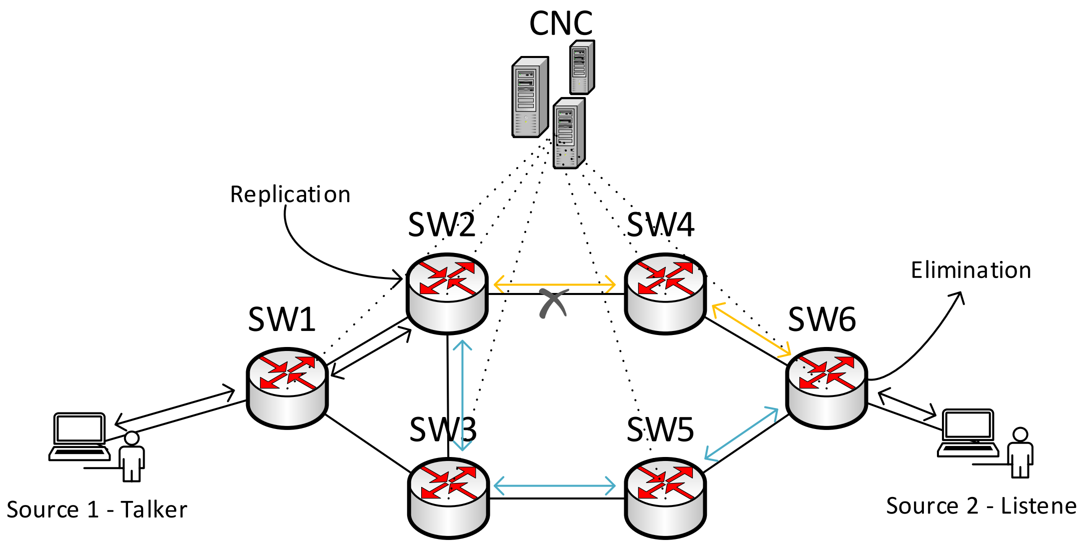


Figure 6.1: Typical Example using Replication Framework for TSN Switches: Centralized User Configuration (CNC) is used to manage and configure the TSN switches and install routes as needed for replicated streams. Replication and elimination processes are transparent and seamless to the Talker/Listeners.

6.2 Seamless Reliability Framework

Our reliability component is built upon the Hybrid TAS reconfiguration framework given in [309]. Two approaches to replication are considered, *i*) ideal, and *ii*) intelligent. The ideal approach identifies immediately the degraded or broken link and starts replication exactly at the affected port. The intelligent approach uses a pre-trained supervised machine learning model which can target specific ports for replication, i.e., a classification model that predicts where and when to start the replication process that enables complete network connectivity in the presence of intermittent or permanent losses. Machine learning techniques are adopted to detect and monitor seamless reliability as dictated by 802.1CB. Typically, the training phase (needing high computational capability and a huge amount of data) is typically done off-line, while the inference phase (e.g. applying the stream redundancy technique to the incoming traffic) can be done in real time involving light computation in a centralized environment.

A simplistic example is shown in Fig. 6.1 where link E_{2-4} fails and the stream is replicated across link E_{2-3}. The elimination procedure is done at SW_6 based on sequence numbering tagged at the replication point, i.e., SW_2. Typically, 802.1CB is not responsible to find a disjoint path. In our approach, the path for redundant traffic streams is selected based on the shortest path on the logically created virtual network that removes the affected link, i.e., the complete explicit path handled by the CNC. If no path is found, then the replication process fails to start.

6.2.1 Model Design and Implementation

The FRER design in this framework follows Fig. 6.2 and the process flow in Fig. 6.3. The switch is initialized with empty source and destination FRER tables. The source FRER table corresponds to any streams where replication originates at the switch and the FRER process start there, while the destination FRER table corresponds to any streams where the replication ends and the FRER process ends there. Before any FRER process or any streams having their replication flag on, the CNC performs admission control and resource reservation to the stream that needs FRER. The admission control and resource reservation follows the same architecture given in Chapter 4 Fig 4.1. Once a stream is accepted for FRER, the table is populated with the stream's information and any traffic belonging to the stream get replicated and have their sequence numbering split and adjusted at the data plane. Finally, a frame is considered lost 1) if the frame is not tagged as replicated and has a bit error when received at the downstream hop or 2) when the sending port gets disconnected due to a permanent break in the channel and therefore gets dropped at the egress point of the port.

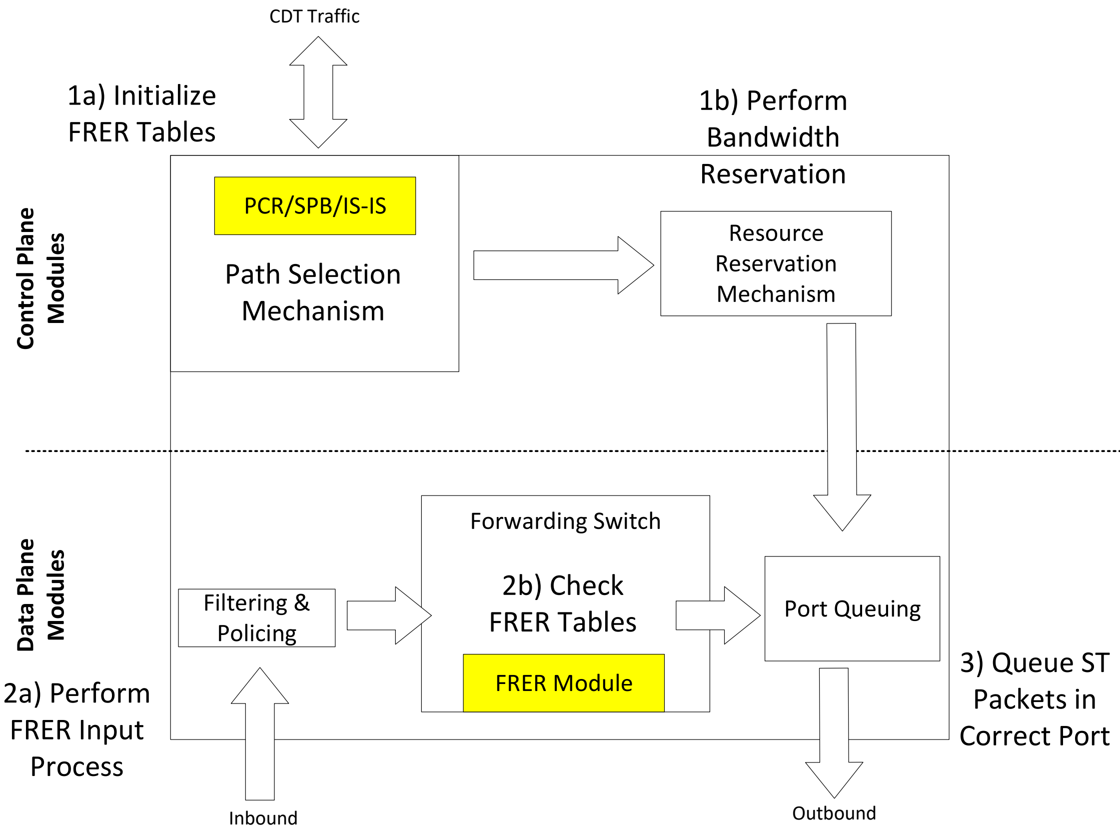


Figure 6.2: High level overview of the switch model. At startup, the switch is initialized with a empty FRER source and destination table that is populated with zero streams. As streams get added by the CNC in the control plane, admission control and bandwidth reservation is used to control the streams. If FRER is required by any port (due to link degradation), the FRER source and destination tables are used to record the streams that are being replicated across the network.

TSN Model and Notation

The traffic and network model follows the same characteristics shown in [309]. A stream (or flow) τ_i^j is denoted by an identifier (or flow ID i) and attached gateway (j). Note that all streams belong to the same traffic class (Scheduled Traffic class) and therefore have the same priority and are hence queued in the same FIFO queue. Each stream is characterized by $\tau_i = (C_i, T_i, U_i, R)$ where C_i is the packet size in bytes, T_i is the period (or inter-arrival time of consecutive packets), U_i is the stream utilization

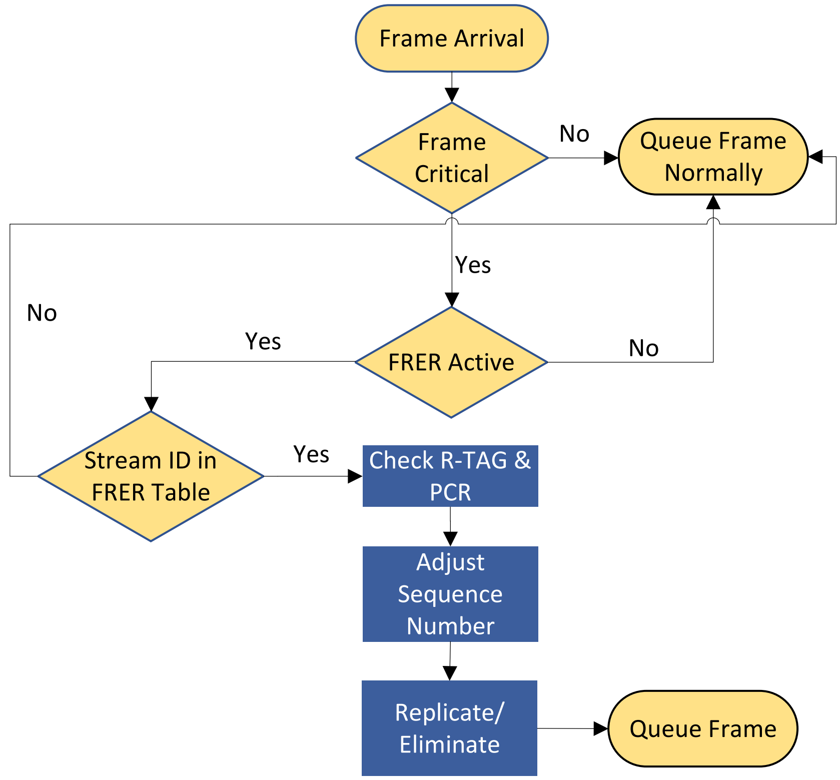


Figure 6.3: Switch Flow incorporating FRER process. As frames arrive, the frame's criticality is checked, and if the FRER process is active, the process starts to replicate/eliminate depending on the source or destination FRER tables.

time (not to be confused with packet deadline), and R is the replication flag. A stream is considered critical when its losses can cause severe loss of operational continuity. This is a notion that is referred to as frame/stream criticality. In our model, all ST streams are critical traffic streams and are therefore eligible for FRER.

ML Parameters

The main parameters are:

i) Number of ports per switch.

ii) Number of ST packets sent per port/switch.

iii) Number of lost ST packets due to link failure per port/switch.

iv) Number of disconnections for each port/switch.

v) Packet error rate for each port/switch.

Using these as input parameters, we can use a Keras sequential dense model (with backend TensorFlow) Feed-forward Neural Network (FNN) to determine the start of the replication process (location) and for how long (duration). The trained multi-class classification model will be given current runtime input from the simulation which decides the (if any) start replication process. The main trade-off of such model is the useful bandwidth consumed over the total bandwidth used for replication. Any packet that arrives at an elimination point and ends up being discarded due to a previously correct packet arriving is considered wasted bandwidth. Generally, we need optimize our model to mitigate such wasted bandwidth where the details of such optimization is left for future work.

System and Error Model

Each component (mainly the CNC and SW) in Fig.6.1 plays an active role in providing seamless reliability in TSN. Each switch inbound port is governed by the Packet Error Rate (PER) that injects bit errors in the frame when it is received. As ST packets are sent across a faulty channel, the probability governed by the PER can cause the measured PER at the destination to shift. If it approaches a pre-defined threshold, the switch contacts the CNC and reports the current switch channel characteristics. An abstract switch model and its corresponding flow process is given in Fig. 6.2 and Fig. 6.3. Note that every switch has 2 FRER tables (one table that records source streams, and another for destination streams).

Table 6.1: Simulation Parameters

Key	**Symbol**	**Value**
Simulation duration	Sim_{limit}	20 seconds
Number of Samples for Training	$Samples$	10000
Initialized cycle time	GCL_{CT}	50 μs
Initialized gating ratio	ST^{R}_{init}	20% (i.e., 10 μs)
Stream duration	τ	approximately 3 seconds
Stream ST Injection Per Cycle	λ	1 Packet Per Cycle
Streams Per Second	π	10
Link propagation delay	α	500 ns
ST sources	S	6
ST stream hop count	TTL	$1-5$
Hurst parameter	H	0.5
Queue size	Q_{size}	512 Kb
Link Capacity	R	1 Gbps

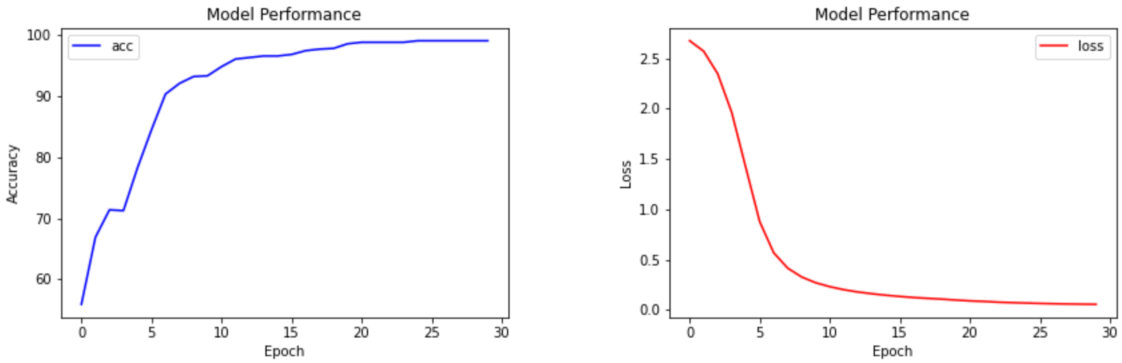
6.3 Performance Evaluation

The topology used to test our proposed FRER model is based on Fig. 6.1 and constructed in OMNet++ simulation environment where the ML agent server is written in Spyder IDE with Keras and TensorFlow libraries. Table. 6.1 shows the simulation parameters used.

6.3.1 Machine Learning Training Evaluation

The training phase is started by having the simulation run various fault random scenarios and collecting results. The results are then passed to the FNN ML agent to generate the ML model. The ML model is used for inferencing at runtime.

In Fig. 6.4, the accuracy and loss of the training phase for 10000 data samples are shown and are compiled using the Spyder-IDE under the Anaconda software suit. Additionally, Keras and TensorFlow is used to create a shallow Neural Network (dense sequential model) with a single hidden layer and a ReLu activation function is used. The multi-class categorical classification model's input is given in Section 6.2.1 while

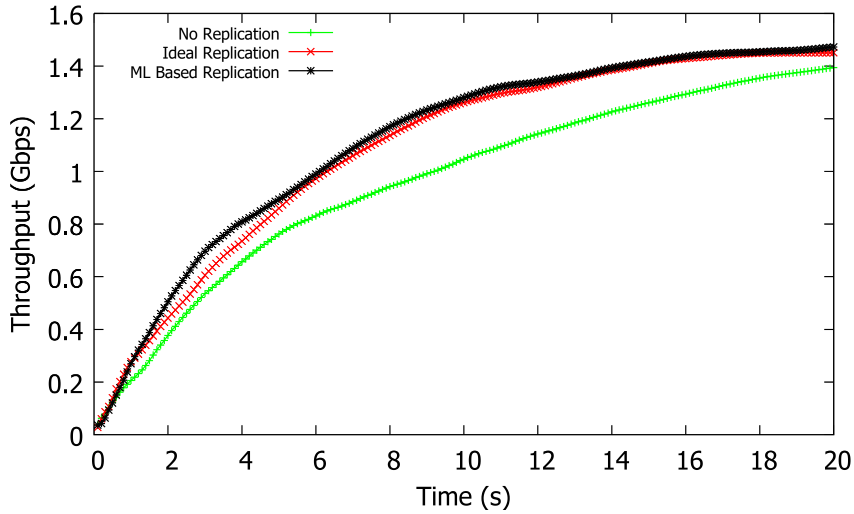


(a) Accuracy

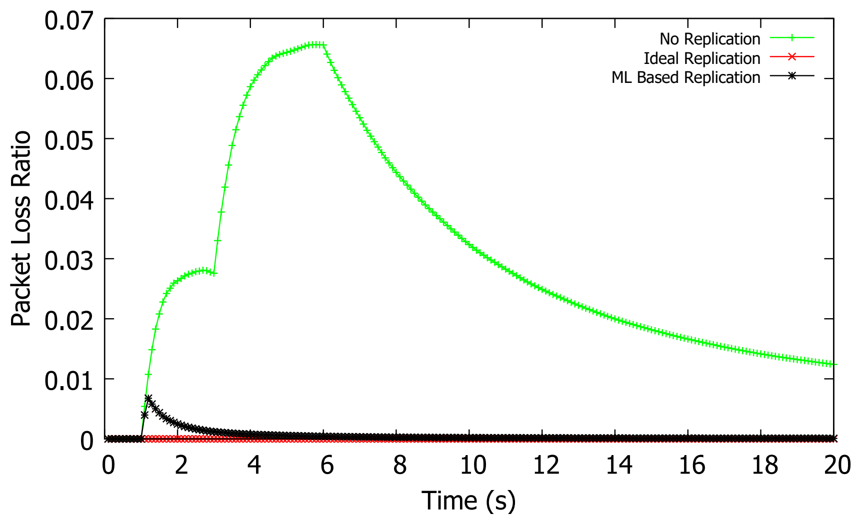
(b) Lose

Figure 6.4: The reported accuracy and sparse categorical cross-entropy lose of the training model in Tensorflow.

the output corresponds to the total number of ports the network is comprised. For example, the network in Fig. 6.1 has a total of 14 ports. Therefore, the output nodes in the FNN has 14 nodes and one extra node that corresponds to no replication, i.e., 15 total nodes for the output of the FNN. For each sample input, we produce probability values for each output node using the Softmax algorithm. The maximum value indicates the best possible point to start replication (or no replication for the last label). The duration for the FRER process is given by a regression model that follows the classification model in a similar fashion. Generally, the duration is considered optimal if we do not underestimate or severely overestimate the faulty scenario. However, that is a non-trivial task which requires large amount of data to build a model on. Therefore, a semi-supervised or none-supervised (e.g., reinforcement) learning model can be used since labeled data is not readily available. Note that the semi-supervised and other approaches to ML are out of scope and left for future work.



(a) Throughput with FRER



(b) Reported Loss Across the Network

Figure 6.5: The recorded throughput and loss at the sink.

6.3.2 *Multi-Class Classification Model Evaluation*

To test the generated ML model, we record the throughput and loss for five independent trial runs and average the results. Fig. 6.5 show the throughput and loss for each FRER approach. The ideal approach is used a reference point to compare the other approaches. While FRER should be seamless (i.e., proactive), we loss some frames at the beginning of the link degradation since the pre-defined threshold had to be higher to not start FRER process while not enough data is collected in the presence of faults. Furthermore, if the PER threshold selected is too low (close to zero), the switch would contact the CNC with incomplete information that the ML model then struggles to issue the correct response producing a false negative or positive for replication. A method to work around such reactive loss is to send multiple copies across a faulty link (i.e., temporal redundancy) though this is out of scope for this dissertation.

The link faults is controlled manually at each simulation run and start at 1 second into the run. At 3 seconds, a permanent link failure starts which causes all frames to be dropped. However, frame losses are only counted if the frame is lost and no redundant frame is sent to the destination using FRER. The permanent link failure is restored at 6 seconds. The throughput shows that most of the frames sent are received at the destination nodes compared to "no replication" where the throughput drops by about 200 Mbps after the link degradation event occurs. Given the ML trained model, our reported validation accuracy is close to 97% indicating that a ML model can predict where and for how long a link/port is faulty and can be used for FRER to eliminate dropped frames. However, a small change in the network topology can cause the model to not work correctly. Therefore, for any changes in the network topology, a new ML model needs to be trained off-line and used online. Also note that stream admission with FRER active can lead to lower streams accepted due to link bandwidth usage by replicated streams.

6.4 Conclusions and Future Work

FRER is standardized to provide seamless redundancy for streams that are considered critical for operational continuity. In this chapter, we worked with our previous implemented model in 4 where admission control and resource reservation has been designed and implemented. We added the FRER process to the CNC and switch model to handle sequence generation, identification, splitting, and recombining features to handle seamless spatial redundancy based on the IEEE 820.1CB standard. Our preliminary investigation indicates the viability of using ML for fault tolerance with high accuracy given a generated synthetic database to train a supervised model on.

In terms of future work, the simulation will be augmented to handle more than one failure simultaneously. Additionally, to reduce the reactive loss of frames before the FRER process is started, the temporal redundancy mechanism will be used to create copies of critical frames on the same channel. Finally, different ML models will be used to validate the best supervised model for inferencing and training based on the collected results during the training phase.

Chapter 7

CONCLUSIONS

Time Sensitive Networking provides the means of converging operational technology and information technology hence improving interoperability and connectivity while using a converged Ethernet technology that augments the network with the benefits of Ethernet's high capacity and reliability. However, Ethernet traditionally does not operate well for real time applications that require ultra low bounded latencies (sub-microsecond) with minimal jitter, i.e., data delivery QoS guarantees. TSN standardization emerged as a suite of standards that enhance Ethernet's protocol at the data link layer to achieve bounded low latencies, low jitter, and zero packet loss. While the standards are an ongoing process with several amendments, several research and studies are needed to investigate the standardization specifications and highlight the limitations and lack of mechanisms to operate correctly or as intended.

To that end, Chapter 2 surveys the standards and research literature related to TSN and a a significant portion of 5G ULL with regards to TSN as an application domain of interest. Several pitfalls and classifications are noted with potential future work directions. Using the findings in Chapter 2, Chapter 3 investigates the flow control element of TSN. More specifically, the Time Aware Shaper and Asynchronous Traffic Shaper are analyzed. Several enhancements are proposed and a comparative analysis is done coupled with a comprehensive empirical evaluation. An adaptive TAS that incorporates the enhancements is compared to the standard TAS and results show that adaptive TAS can reduce mean latencies below the 100μ s bounded delay. However, since the adaptive operation is reactive, the maximum bounded worst case delays are unpredictable. To mitigate such reactive unbounded delays, Chapter 4 shows the reconfiguration framework designed to integrate the centralized management entity with TAS such that the gating timing operation is managed through signaling events across the core TSN domain. Both a centralized (hybrid) model and a decentralized model are designed, implemented, and evaluated. Results show that the worst case delays can be bounded while maximizing the total number of high priority streams accepted and keeping low priority traffic from starvation. However, predicting the delay (queuing delay) precisely is still a challenge. Therefore, cyclic queuing and forwarding is implemented in Chapter 5 and compared with the asynchronous traffic scheduler, Paternoster. While CQF works well in small scale networks, increasing the network size can lead to unpredictable responses. Therefore, a large scale deterministic network is constructed and used to test the operation of CQF and Paternoster. Finally, in Chapter 6, fault tolerance in TSN is investigated and a machine learning supervised model is proposed as a viable alternative to predict the FRER function process. Preliminary results are showcased and indicate that the machine learning model can predict with certain accuracy the location and duration of a fault.

REFERENCES

[1] "3GPP Specification series for 5G NR", URL http://www.3gpp.org/DynaReport/38-series.htm (????).

[2] "Control and user plane separation of EPC nodes (CUPS)", URL http://www.3gpp.org/news-events/3gpp-news/1882-cups (????).

[3] "CPRI specification", URL http://www.cpri.info/ (????).

[4] "IEEE P802.1CM/D2.0 time-sensitive networking for fronthaul", URL http://www.ieee802.org/1/files/private/cm-drafts/d2/802-1CM-d2-0.pdf (????).

[5] "Low latency in 4.9G/5G: Solutions for millisecond latency", URL https://onestore.nokia.com/asset/201407/Nokia_Low-latency_in_4dot9_and_5G_Networks_White_Paper_EN.pdf (????).

[6] "Next generation fronthaul interface (1914) working group", URL http://sites.ieee.org/sagroups-1914 (????).

[7] "Requirements for the eCPRI transport network v1.0", URL http://www.cpri.info/downloads/Requirements_for_the_eCPRI_Transport_Network_V1_0_2017_10_24.pdf (????).

[8] "IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security", IEEE Std 802.1AE-2006 pp. 1–150 (2006).

[9] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002) pp. 1–300 (2008).

[10] "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", IEEE Std 802.1AR-2009 pp. 1–77 (2009).

[11] "IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 12 Forwarding and Queuing Enhancements for Time-Sensitive Streams", IEEE Std 802.1Qav-2009 (Amendment to IEEE Std 802.1Q-2005) pp. C1–72 (2009).

[12] "IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks Amendment 14: Stream Reservation Protocol (SRP)", IEEE Std 802.1Qat-2010 (Revision of IEEE Std 802.1Q-2005) pp. 1–119 (2010).

[13] "IEEE Standard for Local and metropolitan area networks–Port-Based Network Access Control", IEEE Std 802.1X-2010 (Revision of IEEE Std 802.1X-2004) pp. 1–205 (2010).

[14] "IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", IEEE Std 802.1AS-2011 pp. 1–292 (2011).

[15] "IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Security Amendment 1: Galois Counter Mode–Advanced Encryption Standard– 256 (GCM-AES-256) Cipher Suite", IEEE Std 802.1AEbn-2011 (Amendment to IEEE Std 802.1AE-2006) pp. 1–52 (2011).

[16] "IEEE Standard for Device Discovery, Connection Management, and Control Protocol for IEEE 1722(TM) Based Devices", IEEE Std 1722.1-2013 pp. 1–366 (2013).

[17] "IEEE Standard for Local and metropolitan area networksMedia Access Control (MAC) Security Amendment 2: Extended Packet Numbering", IEEE Std 802.1AEbw-2013 (Amendment to IEEE Std 802.1AE-2006) pp. 1–67 (2013).

[18] "IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control Amendment 1: MAC Security Key Agreement Protocol (MKA) Extensions", IEEE Std 802.1Xbx-2014 (Amendment to IEEE Std 802.1X-2010) pp. 1–107 (2014).

[19] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks", IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011) pp. 1–1832 (2014).

[20] "IEEE Standard for Ethernet Amendment 5: Specification and Management Parameters for Interspersing Express Traffic", IEEE Std 802.3br-2016 (Amendment to IEEE Std 802.3-2015 as amended by IEEE St802.3bw-2015, IEEE Std 802.3by-2016, IEEE Std 802.3bq-2016, and IEEE Std 802.3bp-2016) pp. 1–58 (2016).

[21] "IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption", IEEE Std 802.1Qbu-2016 (Amendment to IEEE Std 802.1Q-2014) pp. 1–52 (2016).

[22] "IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic", IEEE Std 802.1Qbv-2015 (Amendment to IEEE Std 802.1Q— as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, and IEEE Std 802.1Q—/Cor 1-2015) pp. 1–57 (2016).

[23] "IEEE Standard for Local and metropolitan area networks– Bridges and Bridged Networks - Amendment 24: Path Control and Reservation", IEEE Std 802.1Qca-2015 (Amendment to IEEE Std 802.1Q— as amended by IEEE Std 802.1Qcd-2015 and IEEE Std 802.1Q—/Cor 1-2015) pp. 1–120 (2016).

[24] "IEEE Draft Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements", IEEE P802.1Qcc/D2.0, October 2017 pp. 1–207 (2017).

[25] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 28: Per-Stream Filtering and Policing", IEEE Std 802.1Qci-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, and IEEE Std 802.1Qbz-2016) pp. 1–65 (2017).

[26] "IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks–Amendment 29: Cyclic Queuing and Forwarding", IEEE 802.1Qch-2017 (Amendment to IEEE Std 802.1Q-2014 as amended by IEEE Std 802.1Qca-2015, IEEE Std 802.1Qcd(TM)-2015, IEEE Std 802.1Q-2014/Cor 1-2015, IEEE Std 802.1Qbv-2015, IEEE Std 802.1Qbu-2016, IEEE Std 802.1Qbz-2016, and IEEE Std 802.1Qci-2017) pp. 1–30 (2017).

[27] "IEEE Standard for Local and metropolitan area networks–Frame Replication and Elimination for Reliability", IEEE Std 802.1CB-2017 pp. 1–102 (2017).

[28] "IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Security - Amendment 3:Ethernet Data Encryption devices", IEEE Std 802.1AEcg-2017 (Amendment to IEEE Std 802.1AE-2006 as amended by IEEE Std 802.1AEbn-2011 and IEEE Std 802.1AEbw-2013) pp. 1–143 (2017).

[29] "IEC/IEEE TSN Profile for Industrial Automation, Use Cases IEC/IEEE 60802 V0.61", (2018).

[30] Abbas, N., Y. Zhang, A. Taherkordi and T. Skeie, "Mobile edge computing: A survey", IEEE Internet of Things Journal **5**, 1, 450–465 (2018).

[31] Acevedo, J., R. Scheffel, S. Wunderlich, M. Hasler, S. Pandi, J. Cabrera, F. H. P. Fitzek, G. Fettweis and M. Reisslein, "Hardware acceleration for RLNC: A case study based on the Xtensa processor with Tensilica instruction-set extension", Electronics, in print (2018).

[32] Adame, T., M. Carrascosa-Zamacois and B. Bellalta, "Time-sensitive networking in ieee 802.11 be: On the way to low-latency wifi 7", Sensors **21**, 15, 4954 (2021).

[33] Addanki, V. and L. Iannone, "Moving a step forward in the quest for deterministic networks (detnet)", in "2020 IFIP Networking Conference (Networking)", pp. 458–466 (IEEE, 2020).

[34] Agiwal, M., A. Roy and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey", IEEE Communications Surveys & Tutorials **18**, 3, 1617–1655 (2016).

[35] Agyapong, P. K., M. Iwamura, D. Staehle, W. Kiess and A. Benjebbour, "Design considerations for a 5G network architecture", IEEE Communications Magazine **52**, 11, 65–75 (2014).

[36] Ahmad, I., W. Chen and K. Chang, "LTE-railway user priority-based cooperative resource allocation schemes for coexisting public safety and railway networks", IEEE Access **5**, 7985–8000 (2017).

[37] Ahmed, A. and A. Shami, "RPR–EPON–WiMAX hybrid network: A solution for access and metro networks", IEEE/OSA Journal of Optical Communications and Networking **4**, 3, 173–188 (2012).

[38] Aijaz, A., M. Dohler, A. H. Aghvami, V. Friderikos and M. Frodigh, "Realizing the tactile internet: Haptic communications over next generation 5G cellular networks", IEEE Wireless Communications **24**, 2, 82–89 (2017).

[39] Aijaz, A., M. Simsek, M. Dohler and G. Fettweis, "Shaping 5G for the tactile internet", in "5G Mobile Communications", pp. 677–691 (Springer, Cham, Switzerland, 2017).

[40] Al-Anbagi, I., M. Erol-Kantarci and H. T. Mouftah, "A survey on cross-layer quality-of-service approaches in WSNs for delay and reliability-aware applications", IEEE Communications Surveys & Tutorials **18**, 1, 525–552 (2016).

[41] Al-Hares, M., P. Assimakopoulos, D. Muench and N. J. Gomes, "Scheduling in an Ethernet fronthaul network", in "Proc. IEEE European Conf. on Networks and Communications (EuCNC)", pp. 1–5 (2017).

[42] Al-Hares, M. K., P. Assimakopoulos, D. Muench and N. J. Gomes, "Modeling time aware shaping in an Ethernet fronthaul", in "Proc. IEEE GLOBECOM", pp. 1–6 (2017).

[43] Al-Hares, M. K., P. Assimakopoulos, D. Muench and N. J. Gomes, "Traditional queuing regimes and time-aware shaping performance comparison in an Ethernet fronthaul network", in "Proc. Int. Conf. on Transparent Optical Netw. (ICTON)", pp. 1–4 (2017).

[44] Al-Maqri, M. A., M. A. Alrshah and M. Othman, "Review on QoS provisioning approaches for supporting video traffic in IEEE802.11e: Challenges and issues", IEEE Access **6**, 55202–55219 (2018).

[45] Alimi, I. A., A. L. Teixeira and P. P. Monteiro, "Towards an efficient C-RAN optical fronthaul for the future networks: A tutorial on technologies, requirements, challenges, and solutions", IEEE Communications Surveys & Tutorials **20**, 1, 708–769 (2018).

[46] Álvarez, I., M. Barranco and J. Proenza, "Towards a fault-tolerant architecture based on time sensitive networking", in "2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)", vol. 1, pp. 1113–1116 (IEEE, 2018).

[47] Álvarez, I., D. Čavka, J. Proenza and M. Barranco, "Simulation of the proactive transmission of replicated frames mechanism over tsn", in "2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", pp. 1375–1378 (IEEE, 2019).

[48] Alvarez, I., J. Proenza and M. Barranco, "Mixing time and spatial redundancy over time sensitive networking", in "Proc. IEEE/IFIP Int. Conf. on Dependable Systems and Networks Workshops (DSN-W)", pp. 63–64 (2018).

[49] Alvarez, I., J. Proenza and M. Barranco, "Mixing time and spatial redundancy over time sensitive networking.", in "DSN Workshops", pp. 63–64 (2018).

[50] Alvarez, I., J. Proenza, M. Barranco and M. Knezic, "Towards a time redundancy mechanism for critical frames in time-sensitive networking", in "Proc. IEEE Int. Conf. on Emerging Techn. and Factory Autom. (ETFA)", pp. 1–4 (2017).

[51] Alvarez, I., J. Proenza, M. Barranco and M. Knezic, "Towards a time redundancy mechanism for critical frames in time-sensitive networking", in "2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", pp. 1–4 (IEEE, 2017).

[52] Alvizu, R., G. Maier, N. Kukreja, A. Pattavina, R. Morro, A. Capello and C. Cavazzoni, "Comprehensive survey on T-SDN: Software-defined networking for transport networks", IEEE Communications Surveys & Tutorials **19**, 4, 2232–2283 (2017).

[53] Ameigeiras, P., J. J. Ramos-Munoz, L. Schumacher, J. Prados-Garzon, J. Navarro-Ortiz and J. M. Lopez-Soler, "Link-level access cloud architecture design based on SDN for 5G networks", IEEE Network **29**, 2, 24–31 (2015).

[54] Amin, R., M. Reisslein and N. Shah, "Hybrid SDN networks: A survey of existing approaches", IEEE Communications Surveys & Tutorials, in print (2018).

[55] Amin, R., M. Reisslein and N. Shah, "Hybrid SDN networks: A survey of existing approaches", IEEE Communications Surveys & Tutorials **20**, 4, 3259–3306 (2018).

[56] Amjad, M., F. Akhtar, M. H. Rehmani, M. Reisslein and T. Umer, "Full-duplex communication in cognitive radio networks: A survey", IEEE Communications Surveys & Tutorials **19**, 4, 2158–2191 (2017).

[57] Amjad, M., M. H. Rehmani and S. Mao, "Wireless multimedia cognitive radio networks: A comprehensive survey", IEEE Communications Surveys & Tutorials **20**, 2, 1056–1103 (2018).

[58] Antonakoglou, K., X. Xu, E. Steinbach, T. Mahmoodi and M. Dohler, "Towards haptic communications over the 5G tactile internet", IEEE Communications Surveys & Tutorials, in print (2018).

[59] Arestova, A., K.-S. J. Hielscher and R. German, "Design of a hybrid genetic algorithm for time-sensitive networking", in "International Conference on Measurement, Modelling and Evaluation of Computing Systems", pp. 99–117 (Springer, 2020).

[60] Arif, F. A. R. and T. S. Atia, “Load balancing routing in time-sensitive networks”, in “Proc. IEEE Int. Scientific-Practical Conf. Problems of Infocommunications Science and Technology”, pp. 207–208 (2016).

[61] Ashraf, S. A., I. Aktas, E. Eriksson, K. W. Helmersson and J. Ansari, “Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G”, in “Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation”, pp. 1–8 (2016).

[62] Assimakopoulos, P., M. K. Al-Hares and N. J. Gomes, “Switched Ethernet fronthaul architecture for cloud-radio access networks”, IEEE/OSA Journal of Optical Communications and Networking **8**, 12, B135–B146 (2016).

[63] Assimakopulos, P., G. S. Birring, M. K. Al-Hares and N. J. Gomes, “Ethernet-based fronthauling for cloud-radio access networks”, in “Proc. IEEE Int. Conf. on Transparent Opt. Netw. (ICTON)”, pp. 1–4 (2017).

[64] Atallah, A. A., G. Bany Hamad and O. Ait Mohamed, “Multipath routing of mixed-critical traffic in time sensitive networks”, in “Advances and Trends in Artificial Intelligence. From Theory to Practice”, edited by F. Wotawa, G. Friedrich, I. Pill, R. Koitz-Hristov and M. Ali, pp. 504–515 (Springer International Publishing, Cham, 2019).

[65] Aurzada, F., M. Lévesque, M. Maier and M. Reisslein, “FiWi access networks based on next-generation PON and gigabit-class WLAN technologies: A capacity and delay analysis”, IEEE/ACM Trans. on Netw. **22**, 4, 1176–1189 (2014).

[66] Aurzada, F., M. Scheutzow, M. Reisslein, N. Ghazisaidi and M. Maier, “Capacity and delay analysis of next-generation passive optical networks (NG-PONs)”, IEEE Trans. on Commun. **59**, 5, 1378–1388 (2011).

[67] Bachhuber, C., E. Steinbach, M. Freundl and M. Reisslein, “On the minimization of glass-to-glass and glass-to-algorithm delay in video communication”, IEEE Transactions on Multimedia **20**, 1, 238–252 (2018).

[68] Badr, A., A. Khisti, W. T. Tan and J. Apostolopoulos, “Perfecting protection for interactive multimedia: A survey of forward error correction for low-delay interactive applications”, IEEE Signal Processing Magazine **34**, 2, 95–113 (2017).

[69] Balasubramanian, V., M. Aloqaily and M. Reisslein, “An sdn architecture for time sensitive industrial iot”, Computer Networks **186**, 107739 (2021).

[70] Bannour, F., S. Souihi and A. Mellouk, “Distributed SDN control: Survey, taxonomy, and challenges”, IEEE Communications Surveys & Tutorials **20**, 1, 333–354 (2018).

[71] Bartelt, J., P. Rost, D. Wubben, J. Lessmann, B. Melis and G. Fettweis, “Fronthaul and backhaul requirements of flexibly centralized radio access networks”, IEEE Wireless Communications **22**, 5, 105–111 (2015).

[72] Battiti, R. and G. Tecchiolli, "The reactive tabu search", ORSA Journal on Computing **6**, 2, 126–140 (1994).

[73] Belliardi, R., J. Dorr, T. Enzinger, F. Essler, J. Farkas, M. Hantel, M. Riegel, M.-P. Stanica, G. Steindl, R. Wamßer, K. Weber and S. A. Zuponcic, "Use Cases IEC/IEEE 60802, V1.3", Available from http://www.ieee802.org/1/files/public/docs2018/60802-industrial-use-cases-0918-v13.pdf; Last accessed Feb. 19, 2019 (2018).

[74] Bello, L. L., "Novel trends in automotive networks: A perspective on ethernet and the IEEE audio video bridging", in "Proc. IEEE Emerging Technology and Factory Automation", pp. 1–8 (2014).

[75] Bello, L. L. and W. Steiner, "A perspective on IEEE time-sensitive networking for industrial communication and automation systems", Proc. IEEE **107**, 6, 1094–1120 (2019).

[76] Benacer, I., F. Boyer and Y. Savaria, "A high-speed, scalable and programmable, traffic manager architecture for flow-based networking", IEEE Access **7**, 2231–2243 (2019).

[77] Bennis, M., M. Debbah and H. V. Poor, "Ultra-reliable and low-latency wireless communication: Tail, risk and scale", arXiv preprint arXiv:1801.01270 (2018).

[78] Beyranvand, H., M. Lévesque, M. Maier, J. A. Salehi, C. Verikoukis and D. Tipper, "Toward 5G: FiWi enhanced LTE-A HetNets with reliable low-latency fiber backhaul sharing and WiFi offloading", IEEE/ACM Transactions on Networking **25**, 2, 690–707 (2017).

[79] Bhagavatula, R. and P. Bhagra, "Centrally managed time sensitive fog networks", US Patent App. 15/687,396 (2018).

[80] Bhushan, N., J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5G", IEEE Communications Magazine **52**, 2, 82–89 (2014).

[81] Bidkar, S., J. Galaro and T. Pfeiffer, "First demonstration of an ultra-low-latency fronthaul transport over a commercial TDM-PON platform", in "Proc. IEEE/OSA Optical Fiber Commun. Conf. and Exposition (OFC)", pp. 1–3 (2018).

[82] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, URL `https://rfc-editor.org/rfc/rfc6087.txt` (2011).

[83] Bierman, A., M. Bjorklund and K. Watsen, "RESTCONF Protocol", RFC 8040, URL `https://rfc-editor.org/rfc/rfc8040.txt` (2017).

[84] Bjomstad, S., D. Chen and R. Veisllari, "Handling delay in 5G ethernet mobile fronthaul networks", in "Proc. IEEE European Conf. on Networks and Commun. (EuCNC)", pp. 1–9 (2018).

[85] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, URL `https://rfc-editor.org/rfc/rfc6020.txt` (2010).

[86] Boudec, J.-Y. L., "A theory of traffic regulators for deterministic networks with application to interleaved regulators", arXiv preprint arXiv:1801.08477 (2018).

[87] Briscoe, B., A. Brunstrom, A. Petlund, D. Hayes, D. Ros, J. Tsang, S. Gjessing, G. Fairhurst, C. Griwodz and M. Welzl, "Reducing internet latency: A survey of techniques and their merits", IEEE Communications Surveys & Tutorials **18**, 3, 2149–2196 (2016).

[88] Busari, S. A., K. M. S. Huq, S. Mumtaz, L. Dai and J. Rodriguez, "Millimeter-Wave massive MIMO communication for future wireless systems: A survey", IEEE Communications Surveys & Tutorials **20**, 2, 836–869 (2018).

[89] Cai, Y., Z. Qin, F. Cui, G. Y. Li and J. A. McCann, "Modulation and multiple access for 5G networks", IEEE Communications Surveys & Tutorials **20**, 1, 629–646 (2018).

[90] Cao, J., M. Ashjaei, P. J. L. Cuijpers, R. J. Bril and J. J. Lukkien, "An independent yet efficient analysis of bandwidth reservation for credit-based shaping", in "Proc. IEEE Int. Workshop on Factory Commun. Systems (WFCS)", pp. 1–10 (2018).

[91] Caporuscio, M., P.-G. Raverdy, H. Moungla and V. Issarny, "ubiSOAP: A service oriented middleware for seamless networking", in "Proc. Int. Conf. on Service-Oriented Computing, LNCS 5364", pp. 195–209 (Springer, Berlin, Heidelberg, 2008).

[92] Carapellese, N., A. Pizzinat, M. Tornatore, P. Chanclou and S. Gosselin, "An energy consumption comparison of different mobile backhaul and fronthaul optical access architectures", in "Proc. European Conf. on Opt. Commun. (ECOC)", pp. 1–3 (2014).

[93] Carvajal, G., L. Araneda, A. Wolf, M. Figueroa and S. Fischmeister, "Integrating dynamic-TDMA communication channels into COTS ethernet networks", IEEE Transactions on Industrial Informatics **12**, 5, 1806–1816 (2016).

[94] Cavalcanti, D., S. Bush, M. Illouz, G. Kronauer, A. Regev and G. Venkatesan, "Wireless tsn-definitions use cases & standards roadmap", Avnu Alliance pp. 1–16 (2020).

[95] Cerovic, D., V. D. Piccolo, A. Amamou, K. Haddadou and G. Pujolle, "Fast packet processing: A survey", IEEE Communications Surveys & Tutorials, in print (2018).

[96] Chanclou, P., L. A. Neto, K. Grzybowski, Z. Tayq, F. Saliou and N. Genay, "Mobile fronthaul architecture and technologies: A RAN equipment assessment", IEEE/OSA Journal of Opt. Commun. and Netw. **10**, 1, A1–A7 (2018).

[97] Chanclou, P., A. Pizzinat, F. Le Clech, T.-L. Reedeker, Y. Lagadec, F. Saliou, B. Le Guyader, L. Guillo, Q. Deniel, S. Gosselin *et al.*, "Optical fiber solution for mobile fronthaul to achieve cloud radio access network", in "Proc. IEEE Future Network and Mobile Summit", pp. 1–11 (2013).

[98] Chang, C.-Y., N. Nikaein, R. Knopp, T. Spyropoulos and S. S. Kumar, "FlexCRAN: A flexible functional split framework over ethernet fronthaul in cloud-RAN", in "Proc. IEEE ICC", pp. 1–7 (2017).

[99] Chang, C.-Y., N. Nikaein and T. Spyropoulos, "Impact of packetization and scheduling on C-RAN fronthaul performance", in "Proc. IEEE GLOBECOM", pp. 1–7 (2016).

[100] Chatzopoulos, D., C. Bermejo, Z. Huang and P. Hui, "Mobile augmented reality survey: From where we are to where we go", IEEE Access **5**, 6917–6950 (2017).

[101] Checko, A., H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger and L. Dittmann, "Cloud RAN for mobile networks—a technology overview", IEEE Communications Surveys & Tutorials **17**, 1, 405–426 (2015).

[102] Chen, F., "Resource Allocation Protocol (RAP) based on LRP for Distributed Configuration of Time-Sensitive Streams", Http://ieee802.org/1/files/public/docs2017/tsn-chen-RAP-whitepaper-0917-v01.pdf (2017).

[103] Chen, F., F.-J. Goetz, M. Kiessling and J. Schmitt, "Aggregation of micro-streams into one common stream", Available at http://ieee802.org/1/files/public/docs2018/dd-chen-flow-aggregation-1218-v02.pdf, Last accessed Dec. 20, 2018 (2018).

[104] Chen, J. and J. Li, "Efficient mobile backhaul architecture offering ultra-short latency for handovers", in "Proc. IEEE Int. Conf. Transparent Optical Networks", pp. 1–1 (2016).

[105] Chen, P.-Y. and M. Reisslein, "FiWi network throughput-delay modeling with traffic intensity control and local bandwidth allocation", Optical Switching and Networking **28**, 8–22 (2018).

[106] Chen, S., R. Ma, H.-H. Chen, H. Zhang, W. Meng and J. Liu, "Machine-to-machine communications in ultra-dense networks—a survey", IEEE Communications Surveys & Tutorials **19**, 3, 1478–1503 (2017).

[107] Chen, W., S. A. Patel, P. Gaal, H. Xu and T. Luo, "Techniques for handling channel state information (CSI) in ultra low latency ULL-LTE", US Patent App. 14/977,163 (2015).

[108] Chen, Y., H. Zhang, N. Fisher, L. Y. Wang and G. Yin, "Probabilistic per-packet real-time guarantees for wireless networked sensing and control", IEEE Trans. on Industrial Informatics **14**, 5, 2133–2145 (2018).

[109] Chitimalla, D., K. Kondepu, L. Valcarenghi, M. Tornatore and B. Mukherjee, "5G fronthaul-latency and jitter studies of CPRI over Ethernet", IEEE/OSA Journal of Optical Communications and Networking **9**, 2, 172–182 (2017).

[110] Choudhury, D. and T. Inoue, "Guest editorial special issue on 5G wireless communication systems and technologies", IEEE Transactions on Microwave Theory and Techniques **64**, 7, 2205–2206 (2016).

[111] Chuang, C.-C., T.-H. Yu, C.-W. Lin, A.-C. Pang and T.-J. Hsieh, "Online stream-aware routing for tsn-based industrial control systems", in "2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", vol. 1, pp. 254–261 (IEEE, 2020).

[112] Chunduri, U., A. Clemm and R. Li, "Preferred Path Routing - a next-generation routing framework beyond Segment Routing", in "Proc. IEEE Global Commun. Conf. (GLOBECOM)", pp. 1–7 (2018).

[113] Chundury, R., "Mobile broadband backhaul: Addressing the challenge", Planning Backhaul Networks, Ericsson Review , 3, 4–9 (2008).

[114] Cizmeci, B., X. Xu, R. Chaudhari, C. Bachhuber, N. Alt and E. Steinbach, "A multiplexing scheme for multimodal teleoperation", ACM Trans. on Multimedia Computing, Communications, and Applications (TOMM) **13**, 2, 21:1–21:28 (2017).

[115] Cohen, A., D. Malak, V. B. Bracha and M. Medard, "Adaptive causal network coding with feedback for delay and throughput guarantees", arXiv preprint arXiv:1905.02870 (2019).

[116] Condoluci, M., T. Mahmoodi, E. Steinbach and M. Dohler, "Soft resource reservation for low-delayed teleoperation over mobile networks", IEEE Access **5**, 10445–10455 (2017).

[117] Cox, C., *An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications* (John Wiley & Sons, Hoboken, NJ, 2012).

[118] Cox, J. H., J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley and H. L. Owen, "Advancing software-defined networks: A survey", IEEE Access **5**, 25487–25526 (2017).

[119] Craciunas, S. S. and R. S. Oliver, "Combined task-and network-level scheduling for distributed time-triggered systems", Real-Time Systems **52**, 2, 161–200 (2016).

[120] Craciunas, S. S. and R. S. Oliver, "An overview of scheduling mechanisms for time-sensitive networks", Tech. rep., Real-time Summer School, L'Ecole d'Ete Temps Reel (ETR) (2017).

[121] Craciunas, S. S., R. S. Oliver, M. Chmelík and W. Steiner, "Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks", in "Proc. ACM Int. Conf. on Real-Time Networks and Systems", pp. 183–192 (2016).

[122] Craciunas, S. S., R. S. Oliver and W. Steiner, "Formal scheduling constraints for time-sensitive networks", arXiv preprint arXiv:1712.02246 (2017).

[123] Craciunas, S. S., R. S. Oliver and W. Steiner, "Demo abstract: Slate XNS– an online management tool for deterministic TSN networks", in "Proc. IEEE Real-Time and Embedded Techn. and Applications Symp. (RTAS)", pp. 103–104 (2018).

[124] Cussans, D., "Trigger Logic Unit (TLU) design ready", Tech. rep., Advanced European Infrastructures for Detectors at Accelerators (AIDA-2020), AIDA-2020-MS43 (2017).

[125] Cvijetic, N., A. Tanaka, K. Kanonakis and T. Wang, "SDN-controlled topology-reconfigurable optical mobile fronthaul architecture for bidirectional CoMP and low latency inter-cell D2D in the 5G mobile era", Optics Express **22**, 17, 20809–20815 (2014).

[126] da Silva, R. B. and E. S. Mota, "A survey on approaches to reduce BGP interdomain routing convergence delay on the internet", IEEE Communications Surveys & Tutorials **19**, 4, 2949–2984 (2017).

[127] Dai, L., B. Wang, Z. Ding, Z. Wang, S. Chen and L. Hanzo, "A survey of non-orthogonal multiple access for 5G", IEEE Communications Surveys & Tutorials **20**, 3, 2294–2323 (2018).

[128] Daniel, O., G. E. Juan, C. Lua and O. Roman, "Failure detection in tsn startup using deep learning", in "2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC)", pp. 140–141 (IEEE, 2020).

[129] Danielis, P., H. Parzyjegla, G. Mühl, E. Schweissguth and D. Timmermann, "Frame replication and elimination for reliability in time-sensitive networks", arXiv preprint arXiv:2109.13677 (2021).

[130] De la Oliva, A., J. A. Hernández, D. Larrabeiti and A. Azcorra, "An overview of the CPRI specification and its application to C-RAN-based LTE scenarios", IEEE Communications Magazine **54**, 2, 152–159 (2016).

[131] De La Oliva, A., X. C. Pérez, A. Azcorra, A. Di Giglio, F. Cavaliere, D. Tiegelbekkers, J. Lessmann, T. Haustein, A. Mourad and P. Iovanna, "Xhaul: toward an integrated fronthaul/backhaul architecture in 5G networks", IEEE Wireless Communications **22**, 5, 32–40 (2015).

[132] Dehos, C., J. L. González, A. De Domenico, D. Ktenas and L. Dussopt, "Millimeter-wave access and backhauling: the solution to the exponential data traffic increase in 5G mobile communications systems?", IEEE Communications Magazine **52**, 9, 88–95 (2014).

[133] Delaney, D., T. Ward and S. McLoone, "On consistency and network latency in distributed interactive applications: A survey—-Part I", Presence: Teleoperators & Virtual Environments **15**, 2, 218–234 (2006).

[134] Delaney, D., T. Ward and S. McLoone, "On consistency and network latency in distributed interactive applications: A survey—-Part II", Presence: Teleoperators and Virtual Environments **15**, 4, 465–482 (2006).

[135] Delgado, R. A., K. Lau, R. H. Middleton and T. Wigren, "Networked delay control for 5G wireless machine-type communications using multiconnectivity", IEEE Transactions on Control Systems Technology, in print (2018).

[136] Deric, N., A. Varasteh, A. Basta, A. Blenk, R. Pries, M. Jarschel and W. Kellerer, "Coupling VNF orchestration and SDN virtual network reconfiguration", in "Proc. Int. Conf. on Networked Systems (NetSys)", (2019).

[137] Desai, N. and S. Punnekkat, "Enhancing fault detection in time sensitive networks using machine learning", in "2020 International Conference on Communication Systems & NETworkS (COMSNETS)", pp. 714–719 (IEEE, 2020).

[138] Destounis, A., S. Paris, L. Maggi, G. S. Paschos and J. Leguay, "Minimum cost SDN routing with reconfiguration frequency constraints", IEEE/ACM Transactions on Networking **26**, 4, 1577–1590 (2018).

[139] Ding, Z., X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends", IEEE Journal on Selected Areas in Communications **35**, 10, 2181–2195 (2017).

[140] Doan-Van, T., A. Kropp, G. T. Nguyen, H. Salah and F. H. Fitzek, "Programmable first: Automated orchestration between MEC and NFV platforms", in "Proc. IEEE Consumer Commun. & Netw. Conf. (CCNC)", pp. 1–2 (2019).

[141] Docquier, T., Y.-Q. Song, V. Chevrier, L. Pontnau and A. Ahmed-Nacer, "Iec 61850 over tsn: Traffic mapping and delay analysis of goose traffic", in "2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", vol. 1, pp. 246–253 (IEEE, 2020).

[142] Doudou, M., D. Djenouri and N. Badache, "Survey on latency issues of asynchronous MAC protocols in delay-sensitive wireless sensor networks", IEEE Communications Surveys & Tutorials **15**, 2, 528–550 (2013).

[143] Douik, A., S. Sorour, T. Y. Al-Naffouri and M. S. Alouini, "Instantly decodable network coding: From centralized to device-to-device communications", IEEE Communications Surveys & Tutorials **19**, 2, 1201–1224 (2017).

[144] Durisi, G., T. Koch, J. Östman, Y. Polyanskiy and W. Yang, "Short-packet communications over multiple-antenna rayleigh-fading channels", IEEE Transactions on Communications **64**, 2, 618–629 (2016).

[145] Durisi, G., T. Koch and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets", Proceedings of the IEEE **104**, 9, 1711–1726 (2016).

[146] Dürr, F. and N. G. Nayak, "No-wait packet scheduling for IEEE time-sensitive networks (TSN)", in "Proc. ACM Int. Conf. on Real-Time Networks and Systems", pp. 203–212 (2016).

[147] Dutta, S., M. Mezzavilla, R. Ford, M. Zhang, S. Rangan and M. Zorzi, "Frame structure design and analysis for millimeter wave cellular systems", IEEE Transactions on Wireless Communications **16**, 3, 1508–1522 (2017).

[148] Einspieler, S., B. Steinwender and W. Elmenreich, "Integrating time-triggered and event-triggered traffic in a hard real-time system", in "Proc. IEEE Industrial Cyber-Physical Systems (ICPS)", pp. 122–128 (2018).

[149] Elbamby, M. S., C. Perfecto, M. Bennis and K. Doppler, "Toward low-latency and ultra-reliable virtual reality", IEEE Network **32**, 2, 78–84 (2018).

[150] Engelmann, A., W. Bziuk, A. Jukan and M. Médard, "Exploiting parallelism with random linear network coding in high-speed Ethernet systems", IEEE/ACM Transactions on Networking (TON) **26**, 6, 2829–2842 (2018).

[151] Enns, R., M. Bjorklund, A. Bierman and J. Schönwälder, "Network Configuration Protocol (NETCONF)", RFC 6241, URL `https://rfc-editor.org/rfc/rfc6241.txt` (2011).

[152] Ergenç, D. and M. Fischer, "On the reliability of ieee 802.1 cb frer", in "IEEE INFOCOM 2021-IEEE Conference on Computer Communications", pp. 1–10 (IEEE, 2021).

[153] Erol-Kantarci, M. and H. T. Mouftah, "Energy-efficient information and communication infrastructures in the smart grid: A survey on interactions and open issues", IEEE Communications Surveys & Tutorials **17**, 1, 179–197 (2015).

[154] Eveleens, J., "Ethernet AVB overview and status", in "Proc. IEEE SMPTE Technical Conf. Exhibition", pp. 1–11 (2014).

[155] Falk, J., F. Dürr and K. Rothermel, "Time-triggered traffic planning for data networks with conflict graphs.", in "RTAS", pp. 124–136 (2020).

[156] Falk, J., D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer and K. Rothermel, "NeSTiNg: Simulating IEEE time-sensitive networking (TSN) in OMNeT++", in "Proc. IEEE Int. Conf. on Networked Systems (NetSys)", pp. 1–8 (2019).

[157] Fan, L., Z. Dong and P. Yuan, "The capacity of device-to-device communication underlaying cellular networks with relay links", IEEE Access **5**, 16840–16846 (2017).

[158] Farkas, J., L. L. Bello and C. Gunther, "Time-sensitive networking standards", IEEE Communications Standards Magazine **2**, 2, 20–21 (2018).

[159] Farkas, J., N. Bragg, P. Unbehagen, G. Parsons, P. J. Ashwood-Smith and C. Bowers, "IS-IS Path Control and Reservation", RFC 7813, URL `https://rfc-editor.org/rfc/rfc7813.txt` (2016).

[160] Farkas, J., B. Varga, G. Miklós and J. Sachs, "5g-tsn integration meets networking requirements for industrial automation", Ericsson: Stockholm, Sweden pp. 0014–0171 (2019).

[161] Farzaneh, M. H. and A. Knoll, "An ontology-based plug-and-play approach for in-vehicle time-sensitive networking (TSN)", in "Proc. IEEE Information Technology, Electronics and Mobile Communication Conf.", pp. 1–8 (2016).

[162] Farzaneh, M. H. and A. Knoll, "Time-sensitive networking (TSN): An experimental setup", in "Proc. IEEE Vehicular Netw. Conf. (VNC)", pp. 23–26 (2017).

[163] Farzaneh, M. H., S. Kugele and A. Knoll, "A graphical modeling tool supporting automated schedule synthesis for time-sensitive networking", in "Proc. IEEE Int. Conf. on Emerging Techn. and Factory Autom. (ETFA)", pp. 1–8 (2017).

[164] Farzaneh, M. H., S. Shafaei and A. Knoll, "Formally verifiable modeling of in-vehicle time-sensitive networks (TSN) based on logic programming", in "Proc. IEEE Vehicular Netw. Conf.", pp. 1–4 (2016).

[165] Ferrari, L., N. Karakoc, A. Scaglione, M. Reisslein and A. Thyagaturu, "Layered cooperative resource sharing at a wireless SDN backhaul", in "Proc. IEEE Int. Conf. on Commun. Workshops (ICC Workshops), Int. Workshop on 5G Architecture (5GARCH)", pp. 1–6 (2018).

[166] Fettweis, G. P., "The tactile internet: Applications and challenges", IEEE Vehicular Technology Magazine **9**, 1, 64–70 (2014).

[167] Fidler, M. and A. Rizk, "A guide to the stochastic network calculus", IEEE Communications Surveys & Tutorials **17**, 1, 92–105 (2015).

[168] Finn, N., "IEEE Draft Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Link-local Registration Protocol", IEEE P802.1CS/D1.2 December 2017 (2017).

[169] Finn, N., "Introduction to Time-Sensitive Networking", IEEE Communications Standards Magazine **2**, 2, 22–28 (2018).

[170] Finn, N., "Introduction to time-sensitive networking", IEEE Communications Standards Magazine **2**, 2, 22–28 (2018).

[171] Finn, N., J.-Y. L. Boudec, E. Mohammadpour, J. Zhang, B. Varga and J. Farkas, "DetNet Bounded Latency", Internet-Draft draft-finn-detnet-bounded-latency-03, Internet Engineering Task Force, URL `https://datatracker.ietf.org/doc/html/draft-finn-detnet-bounded-latency-03`, work in Progress (2019).

[172] Finzi, A., A. Mifdaoui, F. Frances and E. Lochin, "Incorporating TSN/BLS in AFDX for mixed-criticality avionics applications: Specification and analysis", arXiv preprint arXiv:1707.05538 (2017).

[173] Finzi, A., A. Mifdaoui, F. Frances and E. Lochin, "Incorporating TSN/BLS in AFDX for mixed-criticality applications: Model and timing analysis", in "Proc. IEEE Int. Workshop on Factory Commun. Systems", pp. 1–10 (2018).

[174] Finzi, A., A. Mifdaoui, F. Frances and E. Lochin, "Network calculus-based timing analysis of AFDX networks with strict priority and TSN/BLS shapers", in "Proc. Int. Symp. on Industrial Embedded Sys.", pp. 1–10 (2018).

[175] Fitzek, F. H., S.-C. Li, S. Speidel, T. Strufe, M. Simsek and M. Reisslein, *Tactile internet: With human-in-the-Loop* (Academic Press, 2021).

[176] Ford, R., M. Zhang, M. Mezzavilla, S. Dutta, S. Rangan and M. Zorzi, "Achieving ultra-low latency in 5G millimeter wave cellular networks", IEEE Communications Magazine **55**, 3, 196–203 (2017).

[177] Gabriel, F., S. Wunderlich, S. Pandi, F. H. Fitzek and M. Reisslein, "Caterpillar RLNC with feedback (CRLNC-FB): Reducing delay in selective repeat ARQ through coding", IEEE Access **6**, 44787–44802 (2018).

[178] Gao, Y., W. Tang, M. Wu, P. Yang and L. Dan, "Dynamic social-aware computation offloading for low-latency communications in iot", IEEE Internet of Things Journal, in print (2019).

[179] Gao, Z., L. Dai, D. Mi, Z. Wang, M. A. Imran and M. Z. Shakir, "Mmwave massive-MIMO-based wireless backhaul for the 5G ultra-dense network", IEEE Wireless Communications **22**, 5, 13–21 (2015).

[180] Gardiner, E., "The Avnu alliance theory of operation for TSN-enabled industrial systems", IEEE Commun. Standards Mag. **2**, 1, 5–5 (2018).

[181] Gavrilut, V. and P. Pop, "Scheduling in time sensitive networks (TSN) for mixed-criticality industrial applications", in "Proc. IEEE Int. Workshop on Factory Commun. Systems (WFCS)", pp. 1–4 (2018).

[182] Gavriluţ, V. and P. Pop, "Traffic-type assignment for tsn-based mixed-criticality cyber-physical systems", Acm Transactions on Cyber-physical Systems **4**, 2, 1–27 (2020).

[183] Gavriluţ, V., L. Zhao, M. L. Raagaard and P. Pop, "AVB-aware routing and scheduling of time-triggered traffic for TSN", IEEE Access **6**, 75229–75243 (2018).

[184] Georgiadis, L., R. Guérin, V. Peris and K. N. Sivarajan, "Efficient network QoS provisioning based on per node traffic shaping", IEEE/ACM Transactions on Networking **4**, 4, 482–501 (1996).

[185] Ghazisaidi, N., F. Paolucci and M. Maier, "SuperMAN: Optical-wireless integration of RPR and WiMAX", OSA Journal of Optical Networking **8**, 3, 249–271 (2009).

[186] Glover, F., "Tabu search—–Part I", ORSA Journal on Computing **1**, 3, 190–206 (1989).

[187] Gomes, N. J., P. Chanclou, P. Turnbull, A. Magee and V. Jungnickel, "Fronthaul evolution: from CPRI to Ethernet", Optical Fiber Technology **26**, Part A, 50–58 (2015).

[188] Gong, Y., X. Wang, M. Malboubi, S. Wang, S. Xu and C.-N. Chuah, "Towards accurate online traffic matrix estimation in software-defined networks", in "Proc. ACM SIGCOMM Symp. on Software Defined Netw. Res.", pp. 1–7 (2015).

[189] Groß, F., T. Steinbach, F. Korf, T. C. Schmidt and B. Schwarz, "A hardware/software co-design approach for ethernet controllers to support time-triggered traffic in the upcoming IEEE TSN standards", in "Proc. IEEE Int. Conf. on Consumer Electr.–Berlin", pp. 9–13 (2014).

[190] Guck, J. W., M. Reisslein and W. Kellerer, "Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks", IEEE Transactions on Industrial Informatics **12**, 6, 2050–2061 (2016).

[191] Guck, J. W., A. Van Bemten, M. Reisslein and W. Kellerer, "Unicast QoS routing algorithms for SDN: A comprehensive survey and performance evaluation", IEEE Communications Surveys & Tutorials **20**, 1, 388–415 (2018).

[192] Gutiérrez, C. S. V., L. U. S. Juan, I. Z. Ugarte and V. M. Vilches, "Time-sensitive networking for robotics", arXiv preprint arXiv:1804.07643 (2018).

[193] Gutierrez, M., A. Ademaj, W. Steiner, R. Dobrin and S. Punnekkat, "Self-configuration of IEEE 802.1 TSN networks", in "Proc. IEEE Int. Conf. on Emerging Techn. and Factory Autom. (ETFA)", pp. 1–8 (2017).

[194] Gutiérrez, M., W. Steiner, R. Dobrin and S. Punnekkat, "Synchronization quality of IEEE 802.1 AS in large-scale industrial automation networks", in "Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)", pp. 273–282 (2017).

[195] Habiba, U. and E. Hossain, "Auction mechanisms for virtualization in 5G cellular networks: Basics, trends, and open challenges", IEEE Communications Surveys & Tutorials **PP**, 99, 1–1 (2018).

[196] Häckel, T., P. Meyer, F. Korf and T. C. Schmidt, "Software-defined networks supporting time-sensitive in-vehicular communication", arXiv preprint arXiv:1903.08039 (2019).

[197] He, F., L. Zhao and E. Li, "Impact analysis of flow shaping in Ethernet-AVB/TSN and AFDX from network calculus and simulation perspective", Sensors **17**, 5, 1–33 (2017).

[198] Heilmann, F. and G. Fohler, "Size-based queuing: an approach to improve bandwidth utilization in TSN networks", ACM SIGBED Review **16**, 1, 9–14 (2019).

[199] Heise, P., F. Geyer and R. Obermaisser, "TSimNet: An industrial time sensitive networking simulation framework based on OMNeT++", in "Proc. IEEE Int. Conf. on New Technologies, Mobility and Security", pp. 1–5 (2016).

[200] Heise, P., F. Geyer and R. Obermaisser, "Self-configuring deterministic network with in-band configuration channel", in "Proc. IEEE Int. Conf. on Software Defined Systems (SDS)", pp. 162–167 (2017).

[201] Hellmanns, D., J. Falk, A. Glavackij, R. Hummen, S. Kehrer and F. Dürr, "On the performance of stream-based, class-based time-aware shaping and frame preemption in tsn", in "2020 IEEE International Conference on Industrial Technology (ICIT)", pp. 298–303 (IEEE, 2020).

[202] Hellmanns, D., A. Glavackij, J. Falk, R. Hummen, S. Kehrer and F. Dürr, "Scaling tsn scheduling for factory automation networks", in "2020 16th IEEE International Conference on Factory Communication Systems (WFCS)", pp. 1–8 (IEEE, 2020).

[203] Herlich, M., J. L. Du, F. Schörghofer and P. Dorfinger, "Proof-of-concept for a software-defined real-time Ethernet", in "Proc. IEEE Int. Conf on Emerg. Techn. and Factory Autom. (ETFA)", pp. 1–4 (2016).

[204] Hisano, D., T. Kobayashi, H. Ou, T. Shimada, H. Uzawa, J. Terada and A. Otaka, "TDM-PON for accommodating TDD-based fronthaul and secondary services", IEEE/OSA J. Lightwave Techn. **35**, 14, 2788–2796 (2017).

[205] Hisano, D., Y. Nakayama, T. Kubo, T. Shimizu, H. Nakamura, J. Terada and A. Otaka, "Gate-shrunk time aware shaper: Low-latency converged network for 5G fronthaul and M2M services", in "Proc. IEEE Global Commun. Conf. (GLOBECOM)", pp. 1–6 (2017).

[206] Hoeschele, T., C. Dietzel, D. Kopp, F. H. Fitzek and M. Reisslein, "Importance of internet exchange point (ixp) infrastructure for 5g: Estimating the impact of 5g use cases", Telecommunications Policy **45**, 3, 102091 (2021).

[207] Holness, M., "IEEE Draft Standard for Local and metropolitan area networks–Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: YANG Data Model", IEEE P802.1Qcp/D0.7 December 2016 (2016).

[208] Hong, S. H., J. Sun, M. Yu, X. Zhang and A. Xu, "Collaborative coexistence management scheme for industrial wireless sensor networks", IEEE Access **7**, 1617–1626 (2019).

[209] Hossain, E. and M. Hasan, "5G cellular: key enabling technologies and research challenges", IEEE Instrumentation & Measurement Magazine **18**, 3, 11–21 (2015).

[210] Huang, J., M. Zhao, Y. Zhou and C. C. Xing, "In-vehicle networking: Protocols, challenges, and solutions", IEEE Network, in print pp. 1–7 (2018).

[211] Huang, T., F. R. Yu, C. Zhang, J. Liu, J. Zhang and Y. Liu, "A survey on large-scale software defined networking (sdn) testbeds: Approaches and challenges", IEEE Communications Surveys & Tutorials **19**, 2, 891–917 (2017).

[212] IETF, "Flexible ethernet (FlexE) deep dive - IETF datatracker", URL `https://datatracker.ietf.org/meeting/98/materials/slides-98-ccamp-102-flexe-technology-deep-dive/` (????).

[213] Jagadeesan, N. A. and B. Krishnamachari, "Software-defined networking paradigms in wireless networks: A survey", ACM Computing Surveys (CSUR) **47**, 2, 27:1–27:11 (2015).

[214] Jia, W.-K., G.-H. Liu and Y.-C. Chen, "Performance evaluation of IEEE 802.1 Qbu: Experimental and simulation results", in "Proc. IEEE Conf. on Local Computer Networks (LCN)", pp. 659–662 (2013).

[215] Jiang, X., H. Shokri-Ghadikolaei, G. Fodor, E. Modiano, Z. Pang, M. Zorzi and C. Fischione, "Low-latency networking: Where latency lurks and how to tame it", Proc. IEEE **107**, 2, 280–306 (2019).

[216] Jiang, Y., "A basic result on the superposition of arrival processes in deterministic networks", arXiv preprint arXiv:1804.10973 (2018).

[217] Jin, X., C. Xia, N. Guan, C. Xu, D. Li, Y. Yin and P. Zeng, "Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries", IEEE Access **8**, 6751–6767 (2020).

[218] Jungnickel, V., K. Habel, M. Parker, S. Walker, C. Bock, J. F. Riera, V. Marques and D. Levi, "Software-defined open architecture for front-and backhaul in 5G mobile networks", in "Proc. IEEE Int. Conf. Transparent Optical Networks", pp. 1–4 (IEEE, 2014).

[219] Jungnickel, V., D. Schulz, J. Hilt, C. Alexakis, M. Schlosser, L. Grobe, A. Paraskevopoulos, R. Freund, B. Siessegger and G. Kleinpeter, "Optical wireless communication for backhaul and access", in "Proc. IEEE European Conf. on Opt. Commun.", pp. 1–3 (2015).

[220] Kakalou, I., K. E. Psannis, P. Krawiec and R. Badea, "Cognitive radio network and network service chaining toward 5G: Challenges and requirements", IEEE Communications Magazine **55**, 11, 145–151 (2017).

[221] Kakar, J., S. Gherekhloo, Z. H. Awan and A. Sezgin, "Fundamental limits on latency in cloud-and cache-aided hetnets", in "Proc. IEEE Int. Conf. on Communications", pp. 1–6 (2017).

[222] Kani, J.-I., J. Terada, K.-I. Suzuki and A. Otaka, "Solutions for future mobile fronthaul and access-network convergence", IEEE/OSA Journal of Lightwave Techn. **35**, 3, 527–534 (2017).

[223] Karimi, A., K. I. Pedersen, N. H. Mahmood, J. Steiner and P. Mogensen, "5G centralized multi-cell scheduling for URLLC: Algorithms and system-level performance", IEEE Access **6**, 72253–72262 (2018).

[224] Kawabata, A., B. C. Chatterjee, S. Ba and E. Oki, "A real-time delay-sensitive communication approach based on distributed processing", IEEE Access **5**, 20235–20248 (2017).

[225] Kehrer, S., O. Kleineberg and D. Heffernan, "A comparison of fault-tolerance concepts for IEEE 802.1 time sensitive networks (TSN)", in "Proc. IEEE Emerging Techn. and Factory Autom.", pp. 1–8 (2014).

[226] Kellerer, W., P. Kalmbach, A. Blenk, A. Basta, M. Reisslein and S. Schmid, "Adaptable and data-driven softwarized networks: Review, opportunities, and challenges", Proceedings of the IEEE **107**, 4, 711–731 (2019).

[227] Kentis, A. M., M. S. Berger and J. Soler, "Effects of port congestion in the gate control list scheduling of time sensitive networks", in "Proc. Int. Conf. on the Netw. of the Future (NOF)", pp. 138–140 (2017).

[228] Khoshnevisan, M., V. Joseph, P. Gupta, F. Meshkati, R. Prakash and P. Tinnakornsrisuphap, "5g industrial networks with comp for urllc and time sensitive network architecture", IEEE Journal on Selected Areas in Communications **37**, 4, 947–959 (2019).

[229] Kim, J., D. Kim and S. Choi, "3GPP SA2 architecture and functions for 5G mobile communication system", ICT Express **3**, 1, 1–8 (2017).

[230] Kim, Y. and M. Nakamura, "Automotive Ethernet Network Requirements", URL `http://www.ieee802.org/1/files/public/docs2011/new-avb-KimNakamura-automotive-network-requirements-0311.pdf`, last accessed 07/21/2018 (2011).

[231] Kirrmann, H., M. Hansson and P. Muri, "IEC 62439 PRP: Bumpless recovery for highly available, hard real-time industrial networks", in "Proc. IEEE Conf. on Emerging Technologies and Factory Automation (ETFA)", pp. 1396–1399 (2007).

[232] Kirrmann, H., K. Weber, O. Kleineberg and H. Weibel, "HSR: Zero recovery time and low-cost redundancy for industrial ethernet (high availability seamless redundancy, IEC 62439-3)", in "Proc. IEEE Conf. on Emerging Techn. & Factory Automation (ETFA)", pp. 1–4 (2009).

[233] Kleineberg, O., P. Fröhlich and D. Heffernan, "Fault-tolerant ethernet networks with audio and video bridging", in "ETFA2011", pp. 1–8 (IEEE, 2011).

[234] Kleineberg, O., P. Fröhlich and D. Heffernan, "Fault-tolerant audio and video bridging (AVB) Ethernet: A novel method for redundant stream registration configuration", in "Proc. IEEE Conf. on Emerging Technologies & Factory Automation (ETFA)", pp. 1–8 (2012).

[235] Knightly, E. W. and N. B. Shroff, "Admission control for statistical QoS: Theory and practice", IEEE Network **13**, 2, 20–29 (1999).

[236] Ko, J., J.-h. Lee, C. Park and S.-k. Park, "Research on optimal bandwidth allocation for the scheduled traffic in IEEE 802.1 AVB", in "Proc. IEEE Int. Conf. on Vehicular Electronics and Safety (ICVES)", pp. 31–35 (2015).

[237] Kobayashi, T., D. Hisano, T. Shimada, J. Terada and A. Otaka, "Bandwidth allocation scheme based on simple statistical traffic analysis for TDM-PON based mobile fronthaul", in "Proc. OSA Optical Fiber Commun. Conf.", pp. W3C–7–1–W3C–7–3 (2016).

[238] Kobzan, T., S. Schriegel, S. Althoff, A. Boschmann, J. Otto and J. Jasperneite, "Secure and time-sensitive communication for remote process control and monitoring", in "Proc. IEEE Int. Conf. on Emerging Techn. and Factory Autom. (ETFA)", vol. 1, pp. 1105–1108 (2018).

[239] Kopetz, H., A. Ademaj, P. Grillinger and K. Steinhammer, "The time-triggered ethernet (TTE) design", in "Proc. IEEE Int. Symp. on Object-Oriented Real-Time Distr. Computing (ISORC)", pp. 22–33 (2005).

[240] Kovacevic, M., V. Skobic, M. Knezic and Z. Ivanovic, "Towards implementation of frame preemption mechanism on fpga platform", in "2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)", pp. 1–7 (IEEE, 2020).

[241] Kovácsházy, T., "Towards a quantization based accuracy and precision characterization of packet-based time synchronization", in "Proc. IEEE Int. Conf. Precision Clock Sync. for Meas., Ctrl., and Commun.", pp. 1–6 (2016).

[242] Kovacshazy, T. and A. E. Hollos, "Low cost field test measurement method and prototype measurement device implementation for timing accuracy evaluation of IEEE 1588 solutions", in "Proc. IEEE Workshop on Metrology for Industry 4.0 and IoT", pp. 72–77 (2018).

[243] Krolikowski, J., S. Martin, P. Medagliani, J. Leguay, S. Chen, X. Chang and X. Geng, "Joint routing and scheduling for large-scale deterministic ip networks", Computer Communications **165**, 33–42 (2021).

[244] Kumar, G. N., K. Katsalis and P. Papadimitriou, "Coupling source routing with time-sensitive networking", in "2020 IFIP Networking Conference (Networking)", pp. 797–802 (IEEE, 2020).

[245] Kumar, M., M. Boehm, J. Ohms, O. Shulha and O. Gebauer, "Evaluation of the time-aware priority queueing discipline with regard to time-sensitive networking in particular ieee 802.1 qbv", in "Proceedings of International Conference on Applied Innovation in IT", vol. 7, pp. 1–6 (Anhalt University of Applied Sciences, 2019).

[246] Kuo, P.-H. and A. Mourad, "Millimeter wave for 5G mobile fronthaul and backhaul", in "Proc. IEEE European Conf. on Networks and Communications", pp. 1–5 (2017).

[247] Lakkakorpi, J., H. Flinck, J. Heinonen, P. Korja, T. Partti and K. Soranko, "Minimizing delays in mobile networks: With dynamic gateway placement and active queue management", in "Proc. IEEE Wireless Days", pp. 1–3 (2016).

[248] Larrañaga, A., M. C. Lucas-Estañ, I. Martinez, I. Val and J. Gozalvez, "Analysis of 5g-tsn integration to support industry 4.0", in "2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", vol. 1, pp. 1111–1114 (IEEE, 2020).

[249] Laursen, S. M., P. Pop and W. Steiner, "Routing optimization of AVB streams in TSN networks", ACM SIGBED Review **13**, 4, 43–48 (2016).

[250] Lee, B., S. Park, D. J. Love, H. Ji and B. Shim, "Packet structure and receiver design for low latency wireless communications with ultra-short packets", IEEE Transactions on Communications **66**, 2, 796–807 (2018).

[251] Lee, H., J. Lee, C. Park and S. Park, "Time-aware preemption to enhance the performance of audio/video bridging (AVB) in IEEE 802.1 TSN", in "Proc. IEEE Int. Conf. on Computer Commun. and the Internet", pp. 80–84 (2016).

[252] Lee, J. and S. Park, "New interconnection methodology of TSNs using V2X communication", in "Proc. IEEE Computing and Commun. Workshop and Conf.", pp. 1–6 (2017).

[253] Lee, K., J. Kim, Y. Park, H. Wang and D. Hong, "Latency of cellular-based V2X: Perspectives on TTI-proportional latency and TTI-independent latency", IEEE Access **5**, 15800–15809 (2017).

[254] Lema, M. A., A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl and M. Dohler, "Business case and technology analysis for 5G low latency applications", IEEE Access **5**, 5917–5935 (2017).

[255] Lema, M. A., A. Laya, T. Mahmoodi, M. Cuevas, J. Sachs, J. Markendahl and M. Dohler, "Business case and technology analysis for 5G low latency applications", IEEE Access **5**, 5917–5935 (2017).

[256] Lévesque, M. and D. Tipper, "A survey of clock synchronization over packet-switched networks", IEEE Communications Surveys & Tutorials **18**, 4, 2926–2947 (2016).

[257] Li, C.-P., J. Jiang, W. Chen, T. Ji and J. Smee, "5G ultra-reliable and low-latency systems design", in "Proc. IEEE European Conf. on Networks and Commun.", pp. 1–5 (2017).

[258] Li, R., R. Wang, N. Halachmi, Q. Zhong, W. Cheng, L. Wang and J. Wang, "X-ethemet: Enabling integrated fronthaul/backhaul architecture in 5G networks", in "Proc. IEEE Int. Conf. Standards for Communications and Networking", pp. 121–125 (IEEE, 2017).

[259] Li, Z., H. Wan, Y. Deng, K. Xiong and X. Song, "A resource-efficient priority scheduler for time-sensitive networking switches", CCF Transactions on Networking **3**, 1, 21–34 (2020).

[260] Liebeherr, J., D. E. Wrege and D. Ferrari, "Exact admission control for networks with a bounded delay service", IEEE/ACM Transactions on Networking **4**, 6, 885–901 (1996).

[261] Liß, C., M. Ulbricht, U. F. Zia and H. Müller, "Architecture of a synchronized low-latency network node targeted to research and education", in "Proc. IEEE Int. Conf. on High Performance Switching and Routing", pp. 1–7 (2017).

[262] Liu, J., B. Bai, J. Zhang and K. B. Letaief, "Cache placement in Fog-RANs: From centralized to distributed algorithms", IEEE Trans. on Wireless Commun. **16**, 11, 7039–7051 (2017).

[263] Liu, J., H. Guo, H. Nishiyama, H. Ujikawa, K. Suzuki and N. Kato, "New perspectives on future smart FiWi networks: Scalability, reliability, and energy efficiency", IEEE Communications Surveys & Tutorials **18**, 2, 1045–1072 (2016).

[264] Liu, J., P. Zhang, H. Wang and C. Hu, "CounterMap: Towards generic traffic statistics collection and query in Software Defined Network", in "Proc. IEEE/ACM Int. Symp. on Quality of Service (IWQoS)", pp. 1–5 (2017).

[265] Liu, J. and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications", IEEE Access **6**, 12825–12837 (2018).

[266] Liu, X. and F. Effenberger, "Emerging optical access network technologies for 5G wireless", IEEE/OSA Journal of Optical Commun. and Netw. **8**, 12, B70–B79 (2016).

[267] Liu, X., F. Effenberger, N. Chand, L. Zhou and H. Lin, "Demonstration of bandwidth-efficient mobile fronthaul enabling seamless aggregation of 36 E-UTRA-like wireless signals in a single 1.1-GHz wavelength channel", in "Proc. OSA Optical Fiber Commun. Conf. and Exhibition (OFC)", pp. 1–3 (2015).

[268] Liu, X., H. Zeng, N. Chand and F. Effenberger, "Experimental demonstration of high-throughput low-latency mobile fronthaul supporting 48 20-MHz LTE signals with 59-Gb/s CPRI-equivalent rate and 2-μs processing latency", in "Proc. IEEE European Conf. on Optical Communication", pp. 1–3 (2015).

[269] Lu, F., L. Cheng, M. Xu, J. Wang, S. Shen and G. K. Chang, "Orthogonal and sparse chirp division multiplexing for MMW fiber-wireless integrated systems", IEEE Photonics Technology Letters **29**, 16, 1316–1319 (2017).

[270] Lucani, D. E., M. V. Pedersen, D. Ruano, C. W. Sørensen, F. H. Fitzek, J. Heide, O. Geil, V. Nguyen and M. Reisslein, "Fulcrum: Flexible network coding for heterogeneous devices", IEEE Access **6**, 77890–77910 (2018).

[271] Luo, J., J. Jin and F. Shan, "Standardization of low-latency TCP with explicit congestion notification: A survey", IEEE Internet Computing **21**, 1, 48–55 (2017).

[272] Luo, Y. and N. Ansari, "Limited sharing with traffic prediction for dynamic bandwidth allocation and QoS provisioning over Ethernet passive optical networks", OSA Journal of Optical Networking **4**, 9, 561–572 (2005).

[273] Luo, Y., F. J. Effenberger and N. Ansari, "Time synchronization over Ethernet Passive Optical Networks", IEEE Communications Magazine **50**, 10, 136–142 (2012).

[274] Luvisotto, M., Z. Pang and D. Dzung, "Ultra high performance wireless control for critical applications: Challenges and directions", IEEE Transactions on Industrial Informatics **13**, 3, 1448–1459 (2017).

[275] Ma, Z., M. Xiao, Y. Xiao, Z. Pang, H. V. Poor and B. Vucetic, "High-reliability and low-latency wireless communication for internet of things: Challenges, fundamentals and enabling technologies", IEEE Internet of Things Journal, in print (2019).

[276] Maallawi, R., N. Agoulmine, B. Radier and T. B. Meriem, "A comprehensive survey on offload techniques and management in wireless access and core networks", IEEE Communications Surveys & Tutorials **17**, 3, 1582–1604 (2015).

[277] Macchiaroli, R., S. Mole and S. Riemma, "Modelling and optimization of industrial manufacturing processes subject to no-wait constraints", International Journal of Production Research **37**, 11, 2585–2607 (1999).

[278] Mach, P. and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading", IEEE Communications Surveys &Tutorials **19**, 3, 1628–1656 (2017).

[279] MacHardy, Z., A. Khan, K. Obana and S. Iwashina, "V2X access technologies: Regulation, research, and remaining challenges", IEEE Communications Surveys & Tutorials **20**, 3, 1858–1877 (2018).

[280] Mahfouzi, R., A. Aminifar, S. Samii, A. Rezine, P. Eles and Z. Peng, "Stability-aware integrated routing and scheduling for control applications in ethernet networks", in "2018 Design, Automation Test in Europe Conference Exhibition (DATE)", pp. 682–687 (2018).

[281] Maier, M., M. Chowdhury, B. P. Rimal and D. P. Van, “The tactile Internet: Vision, recent progress, and open challenges”, IEEE Communications Magazine **54**, 5, 138–145 (2016).

[282] Maier, M., M. Reisslein and A. Wolisz, “A hybrid MAC protocol for a metro WDM network using multiple free spectral ranges of an arrayed-waveguide grating”, Computer Networks **41**, 4, 407–433 (2003).

[283] Makowitz, R. and C. Temple, “FlexRay–a communication network for automotive control systems”, in “Proc. IEEE Int. Workshop on Factory Commun. Systems”, pp. 207–212 (2006).

[284] Mao, M., R. Giddings, B. Cao, Y. Xu, M. Wang and J. Tang, “DSP-enabled reconfigurable and transparent spectral converters for converging optical and mobile fronthaul/backhaul networks”, Optics Express **25**, 12, 13836–13856 (2017).

[285] Martenvormfelde, L., A. Neumann, L. Wisniewski and J. Jasperneite, “A simulation model for integrating 5g into time sensitive networking as a transparent bridge”, in “2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)”, vol. 1, pp. 1103–1106 (IEEE, 2020).

[286] Martín-Pérez, J., L. Cominardi, C. J. Bernardos, A. De la Oliva and A. Azcorra, “Modeling mobile edge computing deployments for low latency multimedia services”, IEEE Transactions on Broadcasting, in print (2019).

[287] Mathis, M., J. Semke, J. Mahdavi and T. Ott, “The macroscopic behavior of the TCP congestion avoidance algorithm”, ACM SIGCOMM Computer Commun. Rev. **27**, 3, 67–82 (1997).

[288] Maxim, D. and Y.-Q. Song, “Delay analysis of AVB traffic in time-sensitive networks (TSN)”, in “Proc. ACM Int. Conf. on Real-Time Networks and Systems (RTNS)”, pp. 18–27 (2017).

[289] McGarry, M. P., M. Reisslein, F. Aurzada and M. Scheutzow, “Shortest propagation delay (SPD) first scheduling for EPONs with heterogeneous propagation delays”, IEEE Journal on Selected Areas in Communications **28**, 6, 849–862 (2010).

[290] Mehaseb, M. A., Y. Gadallah, A. Elhamy and H. Elhennawy, “Classification of LTE uplink scheduling techniques: An M2M perspective”, IEEE Communications Surveys & Tutorials **18**, 2, 1310–1335 (2016).

[291] Meng, X., J. Li, D. Zhou and D. Yang, “5G technology requirements and related test environments for evaluation”, China Communications **13**, Supplement 2, 42–51 (2016).

[292] Mercian, A., M. P. McGarry and M. Reisslein, “Offline and online multi-thread polling in long-reach PONs: A critical evaluation”, IEEE/OSA Journal of Lightwave Technology **31**, 12, 2018–2028 (2013).

[293] Messenger, J. L., "Time-sensitive networking: An introduction", IEEE Communications Standards Magazine **2**, 2, 29–33 (2018).

[294] Metaal, M. A., R. Guillaume, R. Steinmetz and A. Rizk, "Integrated industrial ethernet networks: Time-sensitive networking over sdn infrastructure for mixed applications", in "2020 IFIP Networking Conference (Networking)", pp. 803–808 (IEEE, 2020).

[295] Metcalfe, R. M., "Computer/network interface design: Lessons from Arpanet and Ethernet", IEEE Journal on Selected Areas in Communications **11**, 2, 173–180 (1993).

[296] Metcalfe, R. M. and D. R. Boggs, "Ethernet: Distributed packet switching for local computer networks", Communications of the ACM **19**, 7, 395–404 (1976).

[297] Meyer, P., T. Steinbach, F. Korf and T. C. Schmidt, "Extending IEEE 802.1 AVB with time-triggered scheduling: A simulation study of the coexistence of synchronous and asynchronous traffic", in "Proc. IEEE Vehicular Networking Conf. (VNC)", pp. 47–54 (2013).

[298] Mezzavilla, M., M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan and M. Zorzi, "End-to-end simulation of 5G mmWave networks", IEEE Communications Surveys & Tutorials **20**, 3, 2237–2263 (2018).

[299] Migge, J., J. Villanueva, N. Navet and M. Boyer, "Insights on the performance and configuration of AVB and TSN in automotive Ethernet networks", in "Proc. Embedded Real-Time Software and Systems", pp. 1–10 (2018).

[300] Mikaeil, A. M., W. Hu, T. Ye and S. B. Hussain, "Performance evaluation of XG-PON based mobile front-haul transport in cloud-RAN architecture", IEEE/OSA Journal of Optical Communications and Networking **9**, 11, 984–994 (2017).

[301] Miyanabe, K., K. Suto, Z. M. Fadlullah, H. Nishiyama, N. Kato, H. Ujikawa and K. i. Suzuki, "A cloud radio access network with power over fiber toward 5G networks: QoE-guaranteed design and operation", IEEE Wireless Commun. **22**, 4, 58–64 (2015).

[302] Moallemi, C. C. and M. Sağlam, "The cost of latency in high-frequency trading", Operations Research **61**, 5, 1070–1086 (2013).

[303] Mogensen, P., K. Pajukoski, E. Tiirola, J. Vihriala, E. Lahetkangas, G. Berardinelli, F. M. Tavares, N. H. Mahmood, M. Lauridsen, D. Catania and A. Cattoni, "Centimeter-wave concept for 5G ultra-dense small cells", in "Proc. IEEE VTC", pp. 1–6 (2014).

[304] Mohammadpour, E., E. Stai, M. Mohiuddin and J.-Y. L. Boudec, "End-to-end latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping", arXiv preprint arXiv:1804.10608 (2018).

[305] Mohammadpour, E., E. Stai, M. Mohiuddin and J.-Y. L. Boudec, "End-to-end latency and backlog bounds in time-sensitive networking with credit based shapers and asynchronous traffic shaping", arXiv preprint arXiv:1804.10608 (2018).

[306] Monti, P., Y. Li, J. Mårtensson, M. Fiorani, B. Skubic, Z. Ghebretensaé and L. Wosinska, "A flexible 5G RAN architecture with dynamic baseband split distribution and configurable optical transport", in "Proc. IEEE Int. Conf. Transparent Optical Netw.", pp. 1–1 (2017).

[307] Mouradian, C., D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges", IEEE Communications Surveys & Tutorials **20**, 1, 416–464 (2018).

[308] Nagata, S., L. H. Wang and K. Takeda, "Industry perspectives", IEEE Wireless Communications **24**, 3, 2–4 (2017).

[309] Nasrallah, A., V. Balasubramanian, A. Thyagaturu, M. Reisslein and H. ElBakoury, "Reconfiguration algorithms for high precision communications in time sensitive networks", in "2019 IEEE Globecom Workshops (GC Wkshps)", pp. 1–6 (IEEE, 2019).

[310] Nasrallah, A., V. Balasubramanian, A. Thyagaturu, M. Reisslein and H. ElBakoury, "TSN algorithms for large scale networks: A survey and conceptual comparison", arXiv preprint arXiv:1905.08478 (2019).

[311] Nasrallah, A., A. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research", IEEE Commun. Surv. & Tut. **21**, 1, 88–145 (2019).

[312] Nasrallah, A., A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein and H. ElBakoury, "Performance comparison of IEEE 802.1 TSN Time Aware Shaper (TAS) and Asynchronous Traffic Shaper (ATS)", IEEE Access **7**, 44165–44181 (2019).

[313] Nasrallah, A., A. S. Thyagaturu, Z. Alharbi, C. Wang, X. Shao, M. Reisslein and H. ElBakoury, "Ultra-low latency (ULL) networks: The IEEE TSN and IETF DetNet standards and related 5G ULL research", IEEE Comm. S. & T. **21**, 1, 88–145 (2019).

[314] Navarro-Ortiz, J., 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 **22**, 2, 905–929 (2020).

[315] Navet, N., J. Migge, J. Villanueva and M. Boyer, "Pre-shaping bursty transmissions under IEEE802.1Q as a simple and efficient QoS mechanism", Tech. rep., SAE Technical Paper 2018-01-0756 (2018).

[316] Nayak, N. G., F. Duerr and K. Rothermel, "Routing Algorithms for IEEE802. 1Qbv Networks", in "RTN workshop, ECRTS", (2017).

[317] Nayak, N. G., F. Duerr and K. Rothermel, "Routing algorithms for ieee802. 1qbv networks", ACM SIGBED Review **15**, 3, 13–18 (2018).

[318] Nayak, N. G., F. Dürr and K. Rothermel, "Software-defined environment for reconfigurable manufacturing systems", in "Proc. IEEE Int. Conf. on the Internet of Things", pp. 122–129 (2015).

[319] Nayak, N. G., F. Dürr and K. Rothermel, "Time-sensitive software-defined network (TSSDN) for real-time applications", in "Proc. ACM Int. Conf. on Real-Time Networks and Systems", pp. 193–202 (2016).

[320] Nayak, N. G., F. Dürr and K. Rothermel, "Incremental Flow Scheduling & Routing in Time-sensitive Software-defined Networks", IEEE Trans. on Industrial Inform. **14**, 5, 2066–2075 (2017).

[321] Nayak, N. G., F. Dürr and K. Rothermel, "Incremental flow scheduling and routing in time-sensitive software-defined networks", IEEE Trans. on Industrial Informatics **14**, 5, 2066–2075 (2018).

[322] Nguyen, K.-G., Q.-D. Vu, M. Juntti and L.-N. Tran, "Energy efficient precoding C-RAN downlink with compression at fronthaul", arXiv preprint arXiv:1703.05996 (2017).

[323] Ni, J., K. Zhang, X. Lin and X. S. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions", IEEE Communications Surveys & Tutorials **20**, 1, 601–628 (2018).

[324] Nichols, K., V. Jacobson, A. McGregor and J. Iyengar, "Controlled delay active queue management", RFC 8289, URL `https://rfc-editor.org/rfc/rfc8289.txt` (2018).

[325] Niemiec, G. S., L. M. Batista, A. E. Schaeffer-Filho and G. L. Nazar, "A survey on fpga support for the feasible execution of virtualized network functions", IEEE Communications Surveys & Tutorials **22**, 1, 504–525 (2019).

[326] Norman, F., "Time-sensitive and Deterministic Networking Whitepaper", (2017).

[327] Noseworthy, B., "Network-based application-independent time-error and direct port latency measurement", in "Proc. IEEE Int. Symp. on Precision Clock Synchronization for Measurement, Control, and Communication", pp. 1–6 (2016).

[328] Nsaibi, S., L. Leurs and H. D. Schotten, "Formal and simulation-based timing analysis of industrial-ethernet Sercos III over TSN", in "Proc. IEEE/ACM Int. Symp. on Distr. Simulation and Real Time Appl. (DS-RT)", pp. 1–8 (2017).

[329] Nunes, B. A. A., M. Mendonca, X.-N. Nguyen, K. Obraczka and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks", IEEE Communications Surveys & Tutorials **16**, 3, 1617–1634 (2014).

[330] Ohms, J., M. Böhm and D. Wermser, "Concept of a tsn to real-time wireless gateway in the context of 5g urllc", in "2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)", pp. 1–6 (IEEE, 2020).

[331] Ojewale, M. A. and P. M. Yomsi, "Routing heuristics for load-balanced transmission in tsn-based networks", ACM Sigbed Review **16**, 4, 20–25 (2020).

[332] Ojewale, M. A., P. M. Yomsi and B. Nikolić, "Multi-level preemption in tsn: feasibility and requirements analysis", in "2020 IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC)", pp. 47–55 (IEEE, 2020).

[333] Oliver, R. S., S. S. Craciunas and W. Steiner, "IEEE 802.1Qbv gate control list synthesis using array theory encoding", in "Proc. IEEE Real-Time and Embedded Techn. and Applications Symp. (RTAS)", pp. 13–24 (2018).

[334] Olsson, M., C. Cavdar, P. Frenger, S. Tombaz, D. Sabella and R. Jantti, "5GrEEn: Towards green 5G mobile networks", in "Proc. IEEE Int. Conf. on Wireless and Mobile Comp., Netw. and Commun. (WiMob)", pp. 212–216 (2013).

[335] Oran, D., "OSI IS-IS Intra-domain Routing Protocol", RFC 1142, URL `https://rfc-editor.org/rfc/rfc1142.txt` (1990).

[336] Osseiran, A., F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus and M. Fallgren, "Scenarios for 5G mobile and wireless communications: The vision of the METIS project", IEEE Communications Magazine **52**, 5, 26–35 (2014).

[337] Pagé, J. and J.-M. Dricot, "Software-defined networking for low-latency 5G core network", in "Proc. IEEE Int. Conf. Military Commun. and Inform. Sys.", pp. 1–7 (2016).

[338] Pahlevan, M. and R. Obermaisser, "Evaluation of time-triggered traffic in time-sensitive networks using the OPNET simulation framework", in "Proc. Euromicro Int. Conf. on Parallel, Distr. and Network-based Processing (PDP)", pp. 283–287 (2018).

[339] Pahlevan, M., N. Tabassam and R. Obermaisser, "Heuristic list scheduler for time triggered traffic in time sensitive networks", ACM Sigbed Review **16**, 1, 15–20 (2019).

[340] Park, C., J. Lee, T. Tan and S. Park, "Simulation of scheduled traffic for the IEEE 802.1 time sensitive networking", in "Information Science and Applications, LNEE Vol. 376", pp. 75–83 (Springer, Singapore, 2016).

[341] Park, D., J. Lee, C. Park and S. Park, "New automatic de-registration method utilizing a timer in the IEEE 802.1 TSN", in "Proc. IEEE Int. Conf. on Computer Commun. and the Internet", pp. 47–51 (2016).

[342] Park, P., S. C. Ergen, C. Fischione, C. Lu and K. H. Johansson, "Wireless network design for control systems: A survey", IEEE Communications Surveys & Tutorials **20**, 2, 978–1013 (2018).

[343] Parkvall, S., E. Dahlman, A. Furuskar and M. Frenne, "NR: The new 5G radio access technology", IEEE Commun. Standards Mag. **1**, 4, 24–30 (2017).

[344] Parvez, I., A. Rahmati, I. Guvenc, A. I. Sarwat and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions", IEEE Communications Surveys & Tutorials, in print (2018).

[345] Patel, S. A., W. Chen, A. Damnjanovic, P. Gaal, M. S. Vajapeyam and H. Xu, "Traffic data allocations in low latency LTE downlink communications", US Patent App. 14/925,501 (2015).

[346] Pedreiras, P., L. Almeida and P. Gai, "The FTT-Ethernet protocol: Merging flexibility, timeliness and efficiency", in "Proc. IEEE Euromicro Conf. on Real-Time Systems", pp. 134–142 (2002).

[347] Pedreiras, P., P. Gai, L. Almeida and G. C. Buttazzo, "FTT-Ethernet: A flexible real-time communication protocol that supports dynamic QoS management on Ethernet-based systems", IEEE Transactions on Industrial Informatics **1**, 3, 162–172 (2005).

[348] Pflug, F. and T. Fingscheidt, "Robust ultra-low latency soft-decision decoding of linear PCM audio", IEEE Transactions on Audio, Speech, and Language Processing **21**, 11, 2324–2336 (2013).

[349] Pieska, M. and A. Kassler, "TCP performance over 5G mmWave links—-tradeoff between capacity and latency", in "Proc. IEEE Int. Conf. on Wireless and Mob. Comp., Netw. and Commun.", pp. 385–394 (2017).

[350] Pilz, J., M. Mehlhose, T. Wirth, D. Wieruch, B. Holfeld and T. Haustein, "A tactile internet demonstration: 1 ms ultra low delay for wireless communications towards 5G", in "Proc. IEEE INFOCOM Wkshps", pp. 862–863 (2016).

[351] Pizzinat, A., P. Chanclou, F. Saliou and T. Diallo, "Things you should know about fronthaul", IEEE/OSA Journal of Lightwave Technology **33**, 5, 1077–1083 (2015).

[352] Ploder, O., N. Palaoro, B. Etzlinger and A. Springer, "A cross-layer approach for ultra-low-latency machine type communication", in "Proc. IEEE ICC", pp. 1–6 (2017).

[353] Pocovi, G., K. I. Pedersen and P. Mogensen, "Joint link adaptation and scheduling for 5G ultra-reliable low-latency communications", IEEE Access **6**, 28912–28922 (2018).

[354] Pop, P., M. L. Raagaard, S. S. Craciunas and W. Steiner, "Design optimisation of cyber-physical distributed systems using IEEE time-sensitive networks", IET Cyber-Physical Systems: Theory Applications **1**, 1, 86–94 (2016).

[355] Pop, P., M. L. Raagaard, M. Gutierrez and W. Steiner, "Enabling fog computing for industrial automation through time-sensitive networking (TSN)", IEEE Communications Standards Magazine **2**, 2, 55–61 (2018).

[356] Pop, P., M. L. Raagaard, M. Gutierrez and W. Steiner, "Enabling fog computing for industrial automation through time-sensitive networking (TSN)", IEEE Commun. Standards Mag. **2**, 2, 55–61 (2018).

[357] Popovski, P., K. F. Trillingsgaard, O. Simeone and G. Durisi, "5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view", IEEE Access **6**, 55765–55779 (2018).

[358] Prinz, F., M. Schoeffler, A. Lechler and A. Verl, "Dynamic real-time orchestration of I4.0 components based on time-sensitive networking", Procedia CIRP **72**, 910–915 (2018).

[359] Qian, K., F. Ren, D. Shan, W. Cheng and B. Wang, "Xpresseth: Concise and efficient converged real-time ethernet", in "Quality of Service (IWQoS), 2017 IEEE/ACM 25th International Symposium on", pp. 1–6 (IEEE, 2017).

[360] Quan, W., W. Fu, J. Yan and Z. Sun, "Opentsn: an open-source project for time-sensitive networking system development", CCF Transactions on Networking **3**, 1, 51–65 (2020).

[361] Raagaard, M. L., P. Pop, M. Gutiérrez and W. Steiner, "Runtime reconfiguration of time-sensitive networking (TSN) schedules for fog computing", in "Proc. IEEE Fog World Congress (FWC)", pp. 1–6 (2017).

[362] Radwan, A., M. F. Domingues and J. Rodriguez, "Mobile caching-enabled small-cells for delay-tolerant e-health apps", in "Proc. IEEE Int. Conf. on Communications Workshops", pp. 103–108 (2017).

[363] Rahman, G. M. S., M. Peng, K. Zhang and S. Chen, "Radio resource allocation for achieving ultra-low latency in fog radio access networks", IEEE Access **6**, 17442–17454 (2018).

[364] Raza, E., H. Le-Minh, N. Aslam, S. Hussain, M. Imran, R. Tafazolli and H. X. Nguyen, "Dynamic priority based reliable real-time communications for infrastructure-less networks", IEEE Access **6**, 67338–67359 (2018).

[365] Reisslein, M., K. W. Ross and S. Rajagopal, "Guaranteeing statistical QoS to regulated traffic: The single node case", in "Proc. INFOCOM", vol. 3, pp. 1061–1072 (1999).

[366] Reisslein, M., K. W. Ross and S. Rajagopal, "A framework for guaranteeing statistical QoS", IEEE/ACM Transactions on Networking **10**, 1, 27–42 (2002).

[367] Rimal, B. P., D. P. Van and M. Maier, “Mobile edge computing empowered fiber-wireless access networks in the 5G era”, IEEE Communications Magazine **55**, 2, 192–200 (2017).

[368] Rostami, A., P. Ohlen, K. Wang, Z. Ghebretensae, B. Skubic, M. Santos and A. Vidal, “Orchestration of RAN and transport networks for 5G: An SDN approach”, IEEE Communications Magazine **55**, 4, 64–70 (2017).

[369] Rumpf, S., T. Steinbach, F. Korf and T. C. Schmidt, “Software stacks for mixed-critical applications: Consolidating IEEE 802.1 AVB and time-triggered ethernet in next-generation automotive electronics”, in “IEEE Int. Conf. on Consumer Electronics–Berlin (ICCE-Berlin)”, pp. 14–18 (2014).

[370] Sabella, D., A. De Domenico, E. Katranaras, M. A. Imran, M. Di Girolamo, U. Salim, M. Lalam, K. Samdanis and A. Maeder, “Energy efficiency benefits of RAN-as-a-service concept for a cloud-based 5G mobile network infrastructure”, IEEE Access **2**, 1586–1597 (2014).

[371] Salem, M., A. Adinoyi, H. Yanikomeroglu and D. Falconer, “Fair resource allocation toward ubiquitous coverage in OFDMA-based cellular relay networks with asymmetric traffic”, IEEE Transactions on Vehicular Technology **60**, 5, 2280–2292 (2011).

[372] Saliou, F., G. Simon, P. Chanclou, M. Brunero, L. Marazzi, P. Parolari, M. Martinelli, R. Brenot, A. Maho, S. Barbet, G. Gavioli, G. Parladori, S. Gebrewold and J. Leuthold, “Self-seeded RSOAs WDM PON field trial for business and mobile fronthaul applications”, in “Proc. OSA Optical Fiber Commun. Conf.”, pp. M2A–2–1–M2A–2–3 (2015).

[373] Samii, S. and H. Zinner, “Level 5 by Layer 2: Time-sensitive networking for autonomous vehicles”, IEEE Communications Standards Magazine **2**, 2, 62–68 (2018).

[374] Sándor, H., B. Genge and Z. Szántó, “Infrastructure and framework for response and reconfiguration in Industry 4.0”, in “Proc. IEEE Int. Symp. on Digital Forensic and Security (ISDFS)”, pp. 1–6 (2018).

[375] Scheutzow, M., M. Maier, M. Reisslein and A. Wolisz, “Wavelength reuse for efficient packet-switched transport in an AWG-based metro WDM network”, IEEE/OSA Journal of Lightwave Technology **21**, 6, 1435 (2003).

[376] Schmidt, M., R. Obermaisser and C. Wurmbach, “Dynamic resource allocation of switched ethernet networks in embedded real-time systems”, in “Proc. Int. Conf. on Inform. Techn. in Biomedicine”, pp. 353–364 (Springer, Cham, Switzerland, 2018).

[377] Schneele, S. and F. Geyer, “Comparison of ieee avb and afdx”, in “Proc. IEEE/AIAA Digital Avionics Systems Conf. (DASC)”, pp. 7A1–1–7A1–9 (2012).

[378] Schriegel, S., T. Kobzan and J. Jasperneite, "Investigation on a distributed SDN control plane architecture for heterogeneous time sensitive networks", in "Proc. IEEE Int. Workshop on Factory Commun. Systems (WFCS)", pp. 1–10 (2018).

[379] Schulz, P., M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste and M. Windisch, "Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture", IEEE Communications Magazine **55**, 2, 70–78 (2017).

[380] Schüngel, M., S. Dietrich, D. Ginthör, S.-P. Chen and M. Kuhn, "Analysis of time synchronization for converged wired and wireless networks", in "2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", vol. 1, pp. 198–205 (IEEE, 2020).

[381] Schweissguth, E., D. Timmermann, H. Parzyjegla, P. Danielis and G. Mühl, "Ilp-based routing and scheduling of multicast realtime traffic in time-sensitive networks", in "2020 IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)", pp. 1–11 (IEEE, 2020).

[382] Seaman, M., "Paternoster policing and scheduling, Revision 2.1", Available from http://www.ieee802.org/1/files/public/docs2019/cr-seaman-paternoster-policing-scheduling-0519-v04.pdf, Last accessed May 25, 2019 (2019).

[383] Sengupta, A., R. Tandon and O. Simeone, "Fog-aided wireless networks for content delivery: Fundamental latency tradeoffs", IEEE Trans. on Info. Th. **63**, 10, 6650–6678 (2017).

[384] Seo, J.-B. and V. C. Leung, "Performance modeling and stability of semi-persistent scheduling with initial random access in LTE", IEEE Trans. on Wireless Commun. **11**, 12, 4446–4456 (2012).

[385] Shantharama, P., A. S. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein and A. Scaglione, "LayBack: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing", IEEE Access **6**, 57545–57561 (2018).

[386] Shantharama, P., A. S. Thyagaturu and M. Reisslein, "Hardware-accelerated platforms and infrastructures for network functions: A survey of enabling technologies and research studies", IEEE Access **8**, 132021–132085 (2020).

[387] She, C., C. Yang and T. Q. S. Quek, "Radio resource management for ultra-reliable and low-latency communications", IEEE Communications Magazine **55**, 6, 72–78 (2017).

[388] Shrestha, D., Z. Pang and D. Dzung, "Precise clock synchronization in high performance wireless communication for time sensitive networking", IEEE Access **6**, 8944–8953 (2018).

[389] Simon, C., M. Maliosz and M. Mate, "Design aspects of low-latency services with time-sensitive networking", IEEE Communications Standards Magazine **2**, 2, 48–54 (2018).

[390] Simsek, M., A. Aijaz, M. Dohler, J. Sachs and G. Fettweis, "5G-enabled tactile internet", IEEE Journal on Selected Areas in Communications **34**, 3, 460–473 (2016).

[391] Smirnov, F., M. Glaß, F. Reimann and J. Teich, "Formal timing analysis of non-scheduled traffic in automotive scheduled TSN networks", in "Proc. IEEE Design, Automation Test in Europe Conf. Exhibition", pp. 1643–1646 (2017).

[392] Smirnov, F., M. Glaß, F. Reimann and J. Teich, "Optimizing message routing and scheduling in automotive mixed-criticality time-triggered networks", in "Proc. ACM/EDAC/IEEE Design Autom. Conf. (DAC)", pp. 1–6 (2017).

[393] Sommer, S., A. Camek, K. Becker, C. Buckl, A. Zirkler, L. Fiege, M. Armbruster, G. Spiegelberg and A. Knoll, "RACE: A centralized platform computer based architecture for automotive applications", in "Proc. IEEE Int. Elec. Vehi. Conf.", pp. 1–6 (2013).

[394] Specht, J., "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/D0.2, November 2017 (2017).

[395] Specht, J. and S. Samii, "Urgency-based scheduler for time-sensitive switched ethernet networks", in "Proc. IEEE Euromicro Conf. on Real-Time Systems", pp. 75–85 (2016).

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

[397] Stanton, K. B., "Distributing deterministic, accurate time for tightly coordinated network and software applications: IEEE 802.1AS, the TSN profile of PTP", IEEE Communications Standards Magazine **2**, 2, 34–40 (2018).

[398] Steiner, W., G. Bauer, B. Hall, M. Paulitsch and S. Varadarajan, "TTEthernet dataflow concept", in "Proc. IEEE Int. Symp. on Network Computing and Applications (NCA)", pp. 319–322 (2009).

[399] Steiner, W., S. S. Craciunas and R. S. Oliver, "Traffic planning for time-sensitive communication", IEEE Communications Standards Magazine **2**, 2, 42–47 (2018).

[400] Sutton, G., J. Zeng, R. Liu, W. Ni, D. Nguyen, B. Jayawickrama, X. Huang, M. Abolhasan and Z. Zhang, "Enabling ultra-reliable and low latency communications through unlicensed spectrum", IEEE Network **32**, 2, 70–77 (2018).

[401] Takeuchi, J., "Requirements for Automotive AVB System Profiles, White Paper Contributed to AVnu Alliance", URL http://avnu.org/wp-content/uploads/2014/05/Contributed-Automotive-Whitepaper_April-2011.pdf, last accessed 07/21/2018 (2011).

[402] Taleb, T., K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration", IEEE Communications Surveys & Tutorials **19**, 3, 1657–1681 (2017).

[403] Tan, Z., C. Yang and Z. Wang, "Energy consume analysis for ring-topology TWDM-PON front-haul enabled cloud RAN", IEEE/OSA Journal of Lightwave Techn. **35**, 20, 4526–4534 (2017).

[404] Taori, R. and A. Sridharan, "Point-to-multipoint in-band mmwave backhaul for 5G networks", IEEE Communications Magazine **53**, 1, 195–201 (2015).

[405] Tashiro, T., S. Kuwano, J. Terada, T. Kawamura, N. Tanaka, S. Shigematsu and N. Yoshimoto, "A novel DBA scheme for TDM-PON based mobile fronthaul", in "Proc. OSA/IEEE Optical Fiber Commun. Conf. and Exhibition (OFC)", pp. 1–3 (2014).

[406] Teener, M. D. J., A. N. Fredette, C. Boiger, P. Klein, C. Gunther, D. Olsen and K. Stanton, "Heterogeneous networks for audio and video: Using IEEE 802.1 audio video bridging", Proceedings of the IEEE **101**, 11, 2339–2354 (2013).

[407] Thangamuthu, S., N. Concer, P. J. L. Cuijpers and J. J. Lukkien, "Analysis of Ethernet-switch traffic shapers for in-vehicle networking applications", in "Proc. IEEE Design, Automation Test in Europe Conf. Exhibition", pp. 55–60 (2015).

[408] Thi, M.-T., S. B. H. Said and M. Boc, "Sdn-based management solution for time synchronization in tsn networks", in "2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", vol. 1, pp. 361–368 (IEEE, 2020).

[409] Thiele, D. and R. Ernst, "Formal analysis based evaluation of software defined networking for time-sensitive ethernet", in "Proc. Design, Automation Test in Europe Conf. Exhibition", pp. 31–36 (2016).

[410] Thiele, D. and R. Ernst, "Formal worst-case performance analysis of time-sensitive ethernet with frame preemption", in "Proc. IEEE 21st Int. Conf. on Emerging Techn. and Factory Automation (ETFA)", pp. 1–9 (2016).

[411] Thiele, D. and R. Ernst, "Formal worst-case timing analysis of ethernet TSN's burst-limiting shaper", in "Proc. IEEE Design, Automation Test in Europe Conf. Exhibition", pp. 187–192 (2016).

[412] Thiele, D., R. Ernst and J. Diemer, "Formal worst-case timing analysis of ethernet TSN's time-aware and peristaltic shapers", in "Proc. IEEE Vehicular Networking Conference (VNC)", pp. 251–258 (2015).

[413] Thyagaturu, A., Z. Alharbi and M. Reisslein, "R-FFT: Function split at IFFT/FFT in unified LTE CRAN and cable access network", IEEE Transactions on Broadcasting **64**, 3, 648–665 (2018).

[414] Thyagaturu, A. S., Y. Dashti and M. Reisslein, "SDN-based smart gateways (Sm-GWs) for multi-operator small cell network management", IEEE Transactions on Network and Service Management **13**, 4, 740–753 (2016).

[415] Trestian, R., I. S. Comsa and M. F. Tuysuz, "Seamless multimedia delivery within a heterogeneous wireless networks environment: Are we there yet?", IEEE Communications Surveys & Tutorials **20**, 2, 945–977 (2018).

[416] Trois, C., M. D. Del Fabro, L. C. de Bona and M. Martinello, "A survey on SDN programming languages: Toward a taxonomy", IEEE Communications Surveys & Tutorials **18**, 4, 2687–2712 (2016).

[417] Tyagi, R. R., F. Aurzada, K. Lee and M. Reisslein, "Connection establishment in LTE-A networks: Justification of Poisson process modeling", IEEE Systems Journal **11**, 4, 2383–2394 (2017).

[418] Unbehagen, P., N. Bragg, D. Allan, D. Fedyk and P. J. Ashwood-Smith, "IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging", RFC 6329, URL `https://rfc-editor.org/rfc/rfc6329.txt` (2012).

[419] Vardakas, J. S., I. T. Monroy, L. Wosinska, G. Agapiou, R. Brenot, N. Pleros and C. Verikoukis, "Towards high capacity and low latency backhauling in 5G: The 5G STEP-FWD vision", in "Proc. IEEE Int. Conf. on Transparent Optical Networks", pp. 1–4 (IEEE, 2017).

[420] Varga, A. and R. Hornig, "An overview of the OMNeT++ simulation environment", in "Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops", p. 60 (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008).

[421] Vasseur, J., A. Farrel and G. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, URL `https://rfc-editor.org/rfc/rfc4655.txt` (2006).

[422] Vlk, M., Z. Hanzálek, K. Brejchová, S. Tang, S. Bhattacharjee and S. Fu, "Enhancing schedulability and throughput of time-triggered traffic in ieee 802.1 qbv time-sensitive networks", IEEE Transactions on Communications **68**, 11, 7023–7038 (2020).

[423] von Arnim, C., M. Drăgan, F. Frick, A. Lechler, O. Riedel and A. Verl, "Tsn-based converged industrial networks: Evolutionary steps and migration paths", in "2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)", vol. 1, pp. 294–301 (IEEE, 2020).

[424] von Arnim, C., A. Lechler, O. Riedel and A. Verl, “Fragmentation in reconfigured real-time production networks”, in “Annals of Scientific Society for Assembly, Handling and Industrial Robotics”, pp. 105–115 (Springer Vieweg, Berlin, Heidelberg, 2020).

[425] Vu, T. K., C. F. Liu, M. Bennis, M. Debbah, M. Latva-aho and C. S. Hong, “Ultra-reliable and low latency communication in mmWave-enabled massive MIMO networks”, IEEE Communications Letters **21**, 9, 2041–2044 (2017).

[426] Wan, T. and P. Ashwood-Smith, “A performance study of CPRI over Ethernet with IEEE 802.1Qbu and 802.1Qbv enhancements”, in “Proc. IEEE Globecom”, pp. 1–6 (2015).

[427] Wan, T., B. McCormick, Y. Wang and P. Ashwood-Smith, “ZeroJitter: An SDN based scheduling for CPRI over Ethernet”, in “Proc. IEEE GLOBECOM”, pp. 1–7 (2016).

[428] Wang, M., N. Karakoc, L. Ferrari, P. Shantharama, A. S. Thyagaturu, M. Reisslein and A. Scaglione, “A multi-layer multi-timescale network utility maximization framework for the sdn-based layback architecture enabling wireless backhaul resource sharing”, Electronics **8**, 9, 937 (2019).

[429] Wang, X., L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon and G. Chen, “Millimeter wave communication: A comprehensive survey”, IEEE Communications Surveys & Tutorials **20**, 3, 1616–1653 (2018).

[430] Waqar, M., A. Kim and P. K. Cho, “A transport scheme for reducing delays and jitter in ethernet-based 5G fronthaul networks”, IEEE Access **6**, 46110–46121 (2018).

[431] Wei, J. Y. and R. I. McFarland, “Just-in-time signaling for WDM optical burst switching networks”, IEEE/OSA Journal of Lightwave Technology **18**, 12, 2019–2037 (2000).

[432] Wollschlaeger, M., T. Sauter and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the Internet of Things and Industry 4.0”, IEEE Industrial Electronics Magazine **11**, 1, 17–27 (2017).

[433] Wong, E., E. Grigoreva, L. Wosinska and C. M. Machuca, “Enhancing the survivability and power savings of 5G transport networks based on DWDM rings”, IEEE/OSA Journal of Optical Commun. and Netw. **9**, 9, D74–D85 (2017).

[434] Wright, D., “The history of the IEEE 802 standard”, IEEE Communications Standards Magazine **2**, 2, 4–4 (2018).

[435] Wu, J., S. Guo, H. Huang, W. Liu and Y. Xiang, “Information and communications technologies for sustainable development goals: State-of-the-art, needs and perspectives”, IEEE Communications Surveys & Tutorials **20**, 3, 2389–2406 (2018).

[436] Wu, J., C. Yuen, N. M. Cheung, J. Chen and C. W. Chen, "Enabling adaptive high-frame-rate video streaming in mobile cloud gaming applications", IEEE Transactions on Circuits and Systems for Video Technology **25**, 12, 1988–2001 (2015).

[437] Wubben, D., P. Rost, J. S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy and G. Fettweis, "Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN", IEEE Signal Processing Magazine **31**, 6, 35–44 (2014).

[438] Wunderlich, S., F. Gabriel, S. Pandi, F. H. Fitzek and M. Reisslein, "Caterpillar RLNC (CRLNC): A practical finite sliding window RLNC approach", IEEE Access **5**, 20183–20197 (2017).

[439] Xiang, Z., F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein and F. H. Fitzek, "Reducing latency in virtual machines: Enabling tactile internet for human-machine co-working", IEEE Journal on Selected Areas in Communications **37**, 5, 1098–1116 (2019).

[440] Xu, C., J. Zhao and G. M. Muntean, "Congestion control design for multipath transport protocols: A survey", IEEE Communications Surveys & Tutorials **18**, 4, 2948–2969 (2016).

[441] Xu, H., W. Yu, D. Griffith and N. Golmie, "A survey on industrial internet of things: A cyber-physical systems perspective", IEEE Access, in print pp. 1–1 (2018).

[442] Xu, M., X. Liu, N. Chand, F. Effenberger and G.-K. Chang, "Flex-frame timing-critical passive optical networks for delay sensitive mobile and fixed access services", in "Proc. OSA Optical Fiber Commun. Conf.", pp. Th4B–6–1–Th4B–6–3 (2017).

[443] Xu, S., M. Perez, K. Yang, C. Perrenot, J. Felblinger and J. Hubert, "Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-Trainer simulator", Surgical Endoscopy **28**, 9, 2569–2576 (2014).

[444] Yamada, T. and R. Nakano, "Job shop scheduling", in "Genetic Algorithms in Engineering Systems, IEE Control Engineering Series 55", edited by A. Zalzala and P. Fleming, pp. 134–160 (The Institution of Electrical Engineers, 1997).

[445] Yang, H.-S., M. Herzog, M. Maier and M. Reisslein, "Metro WDM networks: Performance comparison of slotted ring and AWG star networks", IEEE Journal on Selected Areas in Communications **22**, 8, 1460–1473 (2004).

[446] Yang, S., P. Wieder, M. Aziz, R. Yahyapour, X. Fu and X. Chen, "Latency-sensitive data allocation and workload consolidation for cloud storage", IEEE Access **6**, 76098–76110 (2018).

[447] Yassine, A., H. Rahimi and S. Shirmohammadi, "Software defined network traffic measurement: Current trends and challenges", IEEE Instrumentation & Measurement Magazine **18**, 2, 42–50 (2015).

[448] Yazıcı, V., U. C. Kozat and M. O. Sunay, "A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management", IEEE Communications Magazine **52**, 11, 76–85 (2014).

[449] Yu, Q. and M. Gu, "Adaptive group routing and scheduling in multicast time-sensitive networks", IEEE Access **8**, 37855–37865 (2020).

[450] Zhang, C., Y. Wang, R. Yao, B. Zhou, L. Cheng, Y. Xu, X. Li, J. Cheng and B. Liu, "Packet-size aware scheduling algorithms in guard band for time sensitive networking", CCF Transactions on Networking **3**, 1, 4–20 (2020).

[451] Zhang, H. and D. Ferrari, "Rate-controlled static-priority queueing", in "Proc. IEEE Infocom", pp. 227–236 (1993).

[452] Zhang, H. and D. Ferrari, "Rate-controlled service disciplines", Journal of High Speed Networks **3**, 4, 389–412 (1994).

[453] Zhang, J., Y. Ji, S. Jia, H. Li, X. Yu and X. Wang, "Reconfigurable optical mobile fronthaul networks for coordinated multipoint transmission and reception in 5G", IEEE/OSA Journal of Optical Communications and Networking **9**, 6, 489–497 (2017).

[454] Zhang, K., S. Leng, Y. He, S. Maharjan and Y. Zhang, "Mobile edge computing and networking for green and low-latency Internet of Things", IEEE Communications Magazine **56**, 5, 39–45 (2018).

[455] Zhang, Q., J. Liu and G. Zhao, "Towards 5G enabled tactile robotic telesurgery", arXiv preprint arXiv:1803.03586 (2018).

[456] Zhang, Q., X. Wang, M. Huang, K. Li and S. K. Das, "Software defined networking meets information centric networking: A survey", IEEE Access **6**, 39547–39563 (2018).

[457] Zhang, Z.-L., Z. Liu, J. Kurose and D. Towsley, "Call admission control schemes under generalized processor sharing scheduling", Telecommunication Systems **7**, 1-3, 125–152 (1997).

[458] Zhao, L., P. Pop and S. S. Craciunas, "Worst-case latency analysis for IEEE 802.1Qbv time sensitive networks using network calculus", IEEE Access **6**, 41803–41815 (2018).

[459] Zhao, L., P. Pop, Z. Zheng and Q. Li, "Timing analysis of AVB traffic in TSN networks using network calculus", in "Proc. IEEE Real-Time and Embedded Techn. and Applications Symp. (RTAS)", pp. 25–36 (2018).

[460] Zhou, Z., M. S. Berger, S. R. Ruepp and Y. Yan, "Insight into the IEEE 802.1Qcr asynchronous traffic shaping in time sensitive network", Advances in Science, Technology and Engineering Systems Journal **4**, 1, 292–301 (2019).

[461] Zhou, Z., Y. Yan, M. Berger and S. Ruepp, "Analysis and modeling of asynchronous traffic shaping in time sensitive networks", in "Proc. IEEE Int. Workshop on Factory Commun. Syst. (WFCS)", pp. 1–4 (2018).

[462] Zhou, Z., Y. Yan, S. Ruepp and M. Berger, "Analysis and implementation of packet preemption for time sensitive networks", in "Proc. IEEE Int. Conf. on High Performance Switching and Routing (HPSR)", pp. 1–6 (2017).

[463] Zhu, Y. and M. Ma, "IPACT with grant estimation (IPACT-GE) scheme for Ethernet passive optical networks", IEEE/OSA Journal of Lightwave Techn. **26**, 14, 2055–2063 (2008).