Timing Performance Analysis of the Deterministic Ethernet Enhancements Time-Sensitive Networking (TSN) for Use in the Industrial Communication

Zeitliche Leistungsanalyse der deterministischen Ethernet Erweiterungen Time-Sensitive Networking (TSN) für die Verwendung in der industriellen Kommunikation

vom
Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

Genehmigte Dissertation
von

Dipl.-Ing. Seifeddine Nsaibi
geboren in Tunis, Tunesien

D 386

Tag der mündlichen Prüfung	20.05.2020
Dekan des Fachbereichs:	Prof. Dr.-Ing. Ralph Urbansky
Promotionskommission	
Vorsitzender:	Prof. Dr.-Ing. Norbert Wehn
Berichterstattende:	Prof. Dr.-Ing. Hans D. Schotten
	Prof. Dr.-Ing. Ludwig Leurs

Erklärung

Ich erkläre,

dass ich die vorliegende Dissertation selbst angefertigt habe und alle von mir benutzten Hilfsmittel angegeben habe,

dass ich die Dissertation oder Teile hiervon noch nicht als Prüfungsarbeit für eine staatliche oder wissenschaftliche Prüfung eingereicht habe,

dass ich nicht die gleiche oder eine andere Abhandlung bei einem anderen Fachbereich oder einer anderen Universität als Dissertation eingereicht habe.

_____ _____

(Ort, Datum) (Seifeddine Nsaibi)

Publications

I have authored or co-authored the following publications:

Journal Papers

- Seifeddine Nsaibi, Ludwig Leurs; „*Time Sensitive Networking Leistungsversteigerung von Industrial Ethernet*“; atp edition, pp. 54-61, October 2016
- Seifeddine Nsaibi, Ludwig Leurs; „*Abbildung von TDMA-basierten Industrial Ethernet Protokollen auf TSN am Beispiel von Sercos III*“; Springer (122-135) Kommunikation und Bildverarbeitung in der Automation; December 2017
- Seifeddine Nsaibi, Ajitesh Mishra; “*Experimental Evaluation of TSN performance”*; Industrial Ethernet Book, pp. 33-36; Issue 106 ISSN1470-5745, Mai 2018

Conference Papers

- Seifeddine Nsaibi, Ludwig Leurs; „*Chancen und Grenzen der Leistungssteigerung von Industrial-Ethernet Systemen bei der Verwendung von Ethernet Time Sensitive Networking*“; AUTOMATION Secure & Reliable in the digital world; Baden-Baden Germany, June 2016
- Seifeddine Nsaibi, Ludwig Leurs; „*Ableitung von zukünftigen Steuerungsarchitekturen aus der Kombination von Gigabit- und TSN-Ethernet“;* VDE Congress 2016 Internet of Things; Mannheim Germany, November 2016
- Seifeddine Nsaibi, Ludwig Leurs; „*Abbildung von TDMA-basierten Industrial Ethernet Protokollen auf TSN am Beispiel von Sercos III*“; KommA 2016 – Jahreskolloquium Kommunikation in der Automation; Lemgo Germany, November 2016
- Seifeddine Nsaibi; „*Gegenüberstellung der deterministischen Kommunikationsmechanismen von TSN*: Netzwerkplanung und Telegrammunterbrechung“; AUTOMATION 2017 Technology Networks Processes; Baden-Baden Germany, June 2017
- Seifeddine Nsaibi, Ludwig Leurs, Hans D. Schotten; “*Formal- and Simulation-based Timing Analysis of Industrial-Ethernet Sercos III over TSN”*; IEEE/ACM DS-RT; The 21st IEEE International Symposium on Distributed Simulation and Real-Time Applications; Rom Italy, October 2017
- Seifeddine Nsaibi, Ludwig Leurs; Hans D. Schotten; “Worst-Case Timing Analysis of Integrating TSN in the Field-Level”; KommA 2017 – Jahreskolloquium Kommunikation in der Automation; Magdeburg Germany, November 2017

Technical Reports

- Seifeddine Nsaibi; “*Sercos III over TSN”*; Bosch Networks and Bus Systems; Renningen Germany; April 2017
- Seifeddine Nsaibi; “*It’s about Time – An experimental Evaluation of TSN for integration in the industrial- and automotive fields”*; Bosch PhD Conference; Bosch Corporate Research – Renningen Germany; October 2017
- Seifeddine Nsaibi; “*TSN Missing Functionalities in the Field-level of the Automation industry”*; 2nd German IEEE Meet-Up; Stuttgart Germany; February 2017
- Seifeddine Nsaibi; “*Worst-Case Timing Analysis of Integrating TSN in the Field-level*”; Industrial-Ethernet Workshop; Erbach Germany, December 2017;

Contents

Acknowledgement

This thesis is written during my time as a PhD student at Bosch Rexroth AG in Lohr am Main. I thank God for guiding me through my life and I offer this PhD to my parents Amor and Saliha my dear Khawla, Alaeddine and Khalil.

My deep gratitude goes to Prof. Dr.-Ing. Hans D. Schotten for supervising my thesis and many fruitful discussions. His early advice and his pragmatic mentoring style helped me during my research and contributed on my career path.
A big thank to Prof. Dr.-Ing. Ludwig Leurs, who took on my supervision at Bosch Rexroth AG. I will not forget his willingness to contribute and extend my technical know-how nor our technical discussions.
My appreciation also extents to my Bosch department (DC-AE/EAS) for the financial support to extend the research results.
Last but not least I thank my former students: Adarsh Nayak, Ajitesh Mishra, Kamran Khadir Mohammed and Karima Saif Khandaker for the unforgettably successful brainstorming and the close cooperation that made it possible to publish our research results in multiple journals, workshops and conferences. Many thanks!

Lohr am Main, Dezember 2018 Seifeddine Nsaibi

Abstract

Ethernet has become an established communication technology in industrial automation. This was possible thanks to the tremendous technological advances and enhancements of Ethernet such as increasing the link-speed, integrating the full-duplex transmission and the use of switches. However these enhancements were still not enough for certain high deterministic industrial applications such as motion control, which requires cycle time below one millisecond and jitter or delay deviation below one microsecond. To meet these high timing requirements, machine and plant manufacturers had to extend the standard Ethernet with real-time capability. As a result, vendor-specific and non-IEEE standard-compliant "Industrial Ethernet" (IE) solutions have emerged.

The IEEE Time-Sensitive Networking (TSN) Task Group specifies new IEEE-conformant functionalities and mechanisms to enable the determinism missing from Ethernet. Standard-compliant systems are very attractive to the industry because they guarantee investment security and sustainable solutions. TSN is considered therefore to be an opportunity to increase the performance of established Industrial-Ethernet systems and to move forward to Industry 4.0, which require standard mechanisms.

The challenge remains, however, for the Industrial Ethernet organizations to combine their protocols with the TSN standards without running the risk of creating incompatible technologies. TSN specifies 9 standards and enhancements that handle multiple communication aspects. In this thesis, the evaluation of the use of TSN in industrial real-time communication is restricted to four deterministic standards: IEEE802.1AS-Rev, IEEE802.1Qbu IEEE802.3br and IEEE802.1Qbv. The specification of these TSN sub-standards was finished at an early research stage of the thesis and hardware prototypes were available.

Integrating TSN into the Industrial-Ethernet protocols is considered a substantial strategical challenge for the industry. The benefits, limits and risks are too complex to estimate without a thorough investigation. The large number of Standard enhancements makes it hard to select the required/appropriate functionalities.
In order to cover all real-time classes in the automation [9], four established Industrial-Ethernet protocols have been selected for evaluation and combination with TSN as well as other performance relevant communication features.

The **objectives** of this thesis are to

(1) Provide theoretical, simulation and experimental evaluation-methodologies for the timing performance analysis of the deterministic TSN-standards mentioned above. Multiple test-plans are specified to evaluate the performance and compatibility of early version TSN-prototypes from different providers.

(2) Investigate multiple approaches and deduce migration strategies to integrate these features into the established Industrial-Ethernet protocols: Sercos III, Profinet IRT, Profinet RT and Ethernet/IP. A scenario of coexistence of time-critical traffic with other traffic in a TSN-network proves that the timing performance for highly deterministic applications, e.g. motion-control, can only be guaranteed by the TSN scheduling algorithm IEEE802.1Qbv.

Based on a **requirements** survey of highly deterministic industrial applications, multiple network scenarios and experiments are presented. The results are summarized into two case studies. The **first case study** shows that TSN alone is not enough to meet these requirements. The **second case study** investigates the benefits of additional mechanisms (Gigabit link-speed, minimum cycle time modeling, frame forwarding mechanisms, frame structure, topology migration, etc.) in combination with the TSN features.

An **implementation prototype** of the proposed system and a **simulation case study** are used for the evaluation of the approach. The prototype is used for the evaluation and validation of the simulation model. Due to given scalability constraints of the prototype (no cut-through functionalities, limited number of TSN-prototypes, etc…), a realistic simulation model, using the network simulation tool OMNEST / OMNeT++, is conducted.

The obtained **evaluation results** show that a minimum cycle time ≤1 ms and a maximum jitter ≤1 µs can be achieved with the presented approaches.

Zusammenfassung

Ethernet hat sich erfolgreich als Kommunikationstechnologie in der industriellen Automatisierung etabliert. Dies war dank der enormen technologischen Fortschritte und Verbesserungen des Ethernets möglich, z. B. Mehrfache Erhöhung der Datenrate, Einführung der Vollduplex-Übertragung sowie die Verwendung von Switches für die komplette Vermeidung von Kollisionen. Dennoch waren diese Features nicht ausreichend für bestimmte hoch deterministische industrielle Anwendungen, z. B. für die Bewegungssteuerung, die Zykluszeiten unter einer Millisekunde und Jitter bzw. Verzögerungsabweichungen unter einer Mikrosekunde erfordert. Um diesen hohen zeitlichen Anforderungen gerecht zu werden, mussten die Maschinen- und Anlagenhersteller das Standard-Ethernet um die harte Echtzeitfähigkeit erweitern. Infolgedessen sind herstellerspezifische und nicht IEEE-Standard-kompatible "Industrial Ethernet" -Lösungen (IE) entstanden. Dies wiederspricht sich mit dem Industrie 4.0 Konzept, welches ein einheitliches Kommunikationsstandard bevorzugt.

Die TSN-Taskgruppe IEEE Time-Sensitive Networking spezifiziert neue IEEE-konforme Funktionen und Mechanismen, um den fehlenden Determinismus von Ethernet zu ermöglichen. Standardkonforme Systeme sind für die Industrie sehr attraktiv, da sie Investitionssicherheit und nachhaltige Lösungen garantieren. Daher wird TSN als eine Gelegenheit betrachtet, die Leistung der etablierten Industrial-Ethernet-Systeme zu steigern und zu Industrie 4.0 überzugehen, die Standardmechanismen erfordert.
Die Herausforderung für die Industrial Ethernet-Organisationen besteht jedoch weiterhin darin, ihre Protokolle mit den TSN-Standards zu kombinieren, ohne dass dabei ein Technologiebruch entsteht. TSN spezifiziert 9 Sub-Standards, die mehrere Kommunikationsaspekte handhaben.

Um die Verwendung von TSN in der industriellen Echtzeitkommunikation zu bewerten, ist der Fokus dieser Arbeit auf die vier deterministischen TSN Sub-Standards: IEEE802.1AS-Rev, IEEE802.1Qbu IEEE802.3br und IEEE802.1Qbv.
Diese Sub-Standards wurden im Laufe dieser Dissertation teilweise fertig spezifiziert und als Hardware-Prototypen zur Verfügung gestellt.

Eine Integrationsstrategie von TSN in die Industrial-Ethernet-Protokolle gilt als eine große Herausforderung für die Industrial-Ethernet-Organisationen. Die Vorteile, Grenzen und Risiken sind zu komplex, um sie ohne eingehende Untersuchung abzuschätzen. Die große Anzahl von Standards-Erweiterungen erschwert die Auswahl der erforderlichen / geeigneten Funktionen. Um alle Echtzeitklassen [9] in der Automatisierung abzudecken, wurden vier etablierte Industrial-Ethernet-Protokolle mit unterschiedlichen Leistungsmerkmalen für die Bewertung und Kombination mit TSN sowie andere Kommunikationsfunktionen ausgewählt, die die Leistungsdaten beeinflussen.

Die Ziele dieser Arbeit sind:

- **(1)** Bereitstellung einer experimentellen Evaluierungsmethode für die Timing-Performance-Analyse der oben genannten deterministischen TSN-Standards. Es werden mehrere Testpläne angegeben, um die Leistung und Kompatibilität von TSN-Prototypen früherer Versionen verschiedener Anbieter zu bewerten.
- **(2)** Untersuchung verschiedener Ansätze sowie die Ableitung von Migrationsstrategien, um diese Funktionen in die etablierten Industrial-Ethernet-Protokolle in TSN zu integrieren: Sercos III, Profinet IRT, Profinet RT und Ethernet / IP. Ein Szenario der Koexistenz zeitkritischer Verkehr mit anderen Verkehrsträgern in einer TSN-Cloud beweist, dass die Zeitsteuerungsleistungen für Steuerungsdaten nur durch den TSN-Scheduling-Algorithmus IEEE802.1Qbv garantiert werden können.

Basierend auf einer Anforderungsübersicht über deterministische industrielle Anwendungen werden mehrere Netzwerkszenarien und Experimente vorgestellt. Die Ergebnisse werden in

zwei Fallstudien zusammengefasst. Die erste Fallstudie zeigte, dass TSN alleine nicht ausreicht, um diese Anforderungen zu erfüllen. Die zweite Fallstudie untersucht die Vorteile zusätzlicher Mechanismen (z. B. Gigabit-Verbindungsgeschwindigkeit, Modellierung der minimalen Zykluszeit, Frame-Forwarding-Mechanismen, Frame-Struktur, Topologiemigration usw.) in Kombination mit den TSN-Funktionen.

Ein Implementierungsprototyp des vorgeschlagenen Systems und eine Simulationsfallstudie werden zur Bewertung des Lösungsansatzes verwendet. Der Prototyp dient zur Bewertung und Validierung des Simulationsmodells. Aufgrund gegebener Skalierbarkeitsbeschränkungen des Prototyps (keine Durchschneidefunktionalitäten, begrenzte Anzahl von TSN-Prototypen usw.) wird ein realistisches Simulationsmodell, basierend auf dem Netzwersimulationstool ONEST / OMNeT++, durchgeführt.

Die erhaltenen Auswertungsergebnisse zeigen, dass mit den vorgestellten Lösungsansätzen eine minimale Zykluszeit ≤ 1 ms und ein maximaler Jitter ≤ 1 µs erreicht werden kann.

List of Abbreviations

µs	microsecond
ABB	Asea Brown Boveri
AT	Acknowledge Telegram
AVB	Audio-Video Bridging
BE	Best-Effort
C++	programming language
CBS	Credit-Based Shaper
CIP	Common Industrial Protocol
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
CT	Cut-Through
DFP	Dynamic Frame Packing
DLR	Device Level Ring
DUT	Device Under Test
e.g.	exempli gratia (= for example)
FCnt	Fragment Counter
FCS	Frame Check Sequence
FPGA	Field Programmable Gate Array
Gbps	Gigabit per second
GM	Grand Master Clock
gPTP	generalized Precision Time Protocol
GUI	Graphical User Interface
I210	Intel Network Interface Card
ID	Identifier
IE	Industrial Ethernet
IEEE	Institute of Electrical and Electronics Engineers
IF	Individual Frame
IFG	Inter-Frame Gap
IIC	Industrial Internet Consortium
IP	Internet Protocol
I-PC	Industrial-PC
IRT	Isochronous Real-Time
IT	Information Technology
KPI	Key Performance Indicator
LAN	Local Area Network
LLC	Link Layer Control OSI Layer2b
LNI4.0	Lab Network Industry 4.0
MAC	Media Access Control
MB	Megabyte
Mbps	Megabit per second
MC	Model Checking
MCRC	MAC Merge Cyclic Redundancy Check
MDT	Master Data Telegram
ms	millisecond
MST	Master-Sync-Telegram
MTU	Maximum Transmission Unit
NIC	Network Interface Card
NP-SPA	Non-Preemptive Strict-Priority Algorithm
NP-TAS	Non-Preemptive Time-Aware Shaper
NRT	Non-Real-Time
ns	nanosecond
NTC	Non-Time-Critical
NTP	Network Time Protocol
OSI	Open Systems Interconnection
OT	Operational Technology

PC	Personal Computer
PCI	Peripheral Component Interconnect
PCP	Priority Code Point VLAN Priority field
PER	Packet Error Rate
PHY	Physical Layer
PI	Profinet International
PL	Payload
PLC	Programmable Logic Controller
PPS	Pulse Per Second
P-SPA	Preemptive Strict-Priority Algorithm
P-TAS	Preemptive Time-Aware Shaper
PTP	Precision Time Protocol
QoS	Quality of Service
RAM	Random Access Memory
RC	Rate-Constrained
RJ45	Registered Jack 45 (8/4 wire connector used in networking)
RT	Real-Time
RTC	Real-Time Channel
RTOS	Real-Time Operating System
s	second
S&F	Store&Forward
SDP	Software Defined Pin
Sercos	Serial Realtime Communication System
SF	Summation Frame
SFD	Start of Frame Delimiter
SMD-Cx	Start MAC merge frame Delimiter – continuation fragment
SMD-E	Start MAC merge frame Delimiter - Express
SMD-Sx	Start MAC merge frame Delimiter – Start fragment
SR	Stream Reservation
SRT	Soft-Real-Time
TAP	Terminal Access Points
TAS	Time-Aware Shaper
TC	Time-Critical
TCP	Transmission Control Protocol
TLV	Type Length Value
TSA	Transmission Selection Algorithm
TSN	Time-Sensitive Networking
TT	Time-Triggered
UC	Unified Communication
UCC	Unified Communication Channel
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network

1 Introduction

Real-time applications exist in multiple domains such as digital control, signal processing, multimedia applications and industrial automation [1].
The performances of deterministic communication networks can be measured by the following features that need to be guaranteed for critical data streams [2]:

- High time synchronization accuracy in the nanosecond range
- Guaranteed end-to-end latency for flow reservations through minimal packet loss ratios (10^{-6} to below 10^{-10}) asserted by software and hardware components
- Network configuration and management software functionality as well as protocols to reserve resources (buffers and schedulers) for critical data streams
- A single network able to sustain converged data streams, critical and best-effort, and other QoS features

Typical communication technologies are fieldbus and Ethernet. Due to its limited bandwidth and the increasing demand and amount of exchanged data, the fieldbus technologies are getting replaced by Ethernet. The wide use of Ethernet and its continuous improvement made it suitable for use in the automation domain.
Currently Ethernet is enhanced with new real-time functionalities for use in highly deterministic domains without any vendor-specific modifications.
The focus of this work is to investigate the communication performance of the new IEEE Ethernet enhancements, Time-Sensitive Networking (TSN), to meet the timing requirements of the deterministic industrial applications.

1.1 Deterministic Networking

1.1.1 Deterministic Industrial Applications

1.1.1.1 Classification

The deterministic industrial applications can be classified into the following dominant classes [3]:

- **Condition Monitoring** – Of the three classes, these applications have the least stringent real-time communication requirements. They primarily constitute of the monitoring of the condition of electromechanical, pneumatic or hydraulic components using, for example, an elaborate network of sensors on the factory floor to collect measurements of currents, vibrations or temperatures. The main requirement is to maintain a common time base for the communication signals which requires node synchronization.
- **Process Automation** – With comparatively stricter requirements, this class concentrates on maintaining a high quality in applications of mainly mass or batch production. Such networks record and propagate large amounts of data from extensive networks of devices, shifting the focus to providing higher link-speeds and coexistence.
- **Factory Automation** – This group of applications require closed loop feedback systems consisting of sensors and actuators to support discrete manufacturing processes (product assembly, testing, packing etc.). Such networks demand very high real-time communication requirements to provide the needed precision and speed.

The focus of this thesis is analyzing different approaches to meet the communication requirements of the highly deterministic applications in factory automation. Typical examples of this class of application are: machine tools, packaging machines, and printing machines.
Machine tools use computerized numerical control machines to manufacture geometrically complex products with high precision. An example of such an application is a high speed milling machine, constituting of a large number of communication components with complex control movements. The communication cycle commonly involves transmitting 50-byte data packets from master to each slave and vice versa per cycle, while real-time data needs to be transmitted at 0.5 ms intervals. Additionally, each slave needs to be synchronized to accuracy of 1 µs to reach the needed precision at high speeds. Cooperating robot arms, also machine tools, generally having greater dimensions tend to be slower and thus have less demanding cycle time requirements (1 ms). Existing systems operate with a Packet Error Rate (PER) of 10^{-9} [3].

Packaging Machines are made up of multiple subsystems for the different functions, each equipped with a controller unit. Additionally, the system needs to be provided with supplementary materials, which may need to be processed first, and the finished products have to be transported, in accordance with the main functionalities. An example of a subsystem of packing machines is a pick-and-place machine, which sorts products continuously moving on a conveyor belt into boxes to be transported. Commonly the communication in such a system involves 65 individual components. On average 40 Bytes of data need to be transmitted from master to slaves and vice versa every 2 ms (with jitter 5 µs). These systems require a PER of less than 10^{-8} [3].

Printing machines in industry consist of two main functional systems: flexography and injector printing systems. The cylinders of a flexography system need to be synchronized precisely to produce clear images, as a 5 µm jitter can result in visible defects. The printing material in sophisticated flexography systems move at a speed of 20 m/s, limiting the tolerable cycle time jitter to 0.25 µs while cycle times in the range to 10 ms is acceptable. Injector printing systems use moving print heads, that need to be quick to change their direction at frequent intervals, setting the required cycle time at 2 ms. Flexography systems can have up to 100 communicating nodes, receiving and transmiting 20 Bytes of control information and data per cycle. The PER of these systems need to be less than 10^{-7} [3] .

1.1.1.2 Key Performance Indicators (KPIs)

Definition
KPIs are the computable results of a performance analysis, based on a set of measured values [4].
For example, the network delay KPI in the Industrial Protocol Performance is a value computed by using primitive information like the message data size, the link-speed, the message timestamps, the forwarding mechanism and the number of forwarding hops. These primitives provide a way to compute the end-to-end delay of a message throughout the followed path.
According to [5] metrics can be classified as economic-, reliability- or performance-related. While economic metrics generally pertain to costs, performance metrics and reliability metrics are more relevant to this research. Table 1 summarizes the different metrics that are commonly used [5] and those specific to industrial networks [6].

Economic Metrics	Reliability Metrics	Performance Metrics
Acquisition Costs Development Costs Installation Costs (Upkeep Costs)	Reliability Maintainability Availability Packet loss rate	Latency Jitter Bandwidth Cycle Time Length Number of Frames Energy Consumption Computation complexity Start-Up Speed

		EMC (Electromagnetic Compatibility) Dimension Delivery time (measured at application layer) Synchronization accuracy Possible number of end nodes Number of switches between end nodes Throughput of real-time data Non-real-time bandwidth Basic network topology Redundancy recovery time

Table 1: Metrics Classification of communication services

[7] **Network Performance** KPIs measure the performance of the underlying network. In the Process Control system, the network is a critical communication link for all sub-systems. All the nodes are connected and communicate through the network connection. As a critical communication channel, any network delay or failure will impact the performance of the entire continuous production system. For example, the Controller relies on real-time sensor information to compute the desired actuator values to maintain stable control. Also the HMI relies on real-time data to display the up-to-date process information for operators. Each computer has a network packet analyzer tool installed to capture all the inbound and outbound network traffic of that computer. Post-processing is performed to compute the desired KPIs.

Key Performance Indicator (KPI)	Description
Packet Path Delay	Measures the time delay along the path from transmitter to receiver.
Inter-packet Delay	Measures the difference between the packet path delay of two packets.
Round Trip Time	Measures the amount of time for the source node to receive the acknowledgement of receipt (ACK) from the destination node.
Bit Rate	Measures the rate of bits transmitted or received over a specific amount of time.
Packet Rate	Measures the rate of packets transmitted and received over a specific amount of time
Network Utilization	Measures the percentage of network capacity utilized over a specific amount of time.
Packet Size	Measures the number of bytes contained in a packet.
Packet Loss Rate	Measures the percentage of packets that failed to reach the destination node over a specific amount of time.

Table 2: Network Key Performance Indicators

Computing Resources Performance KPI measures the performance of the computing hardware and software. This is especially important for sub-systems that are software intensive, such as the HMI, Data Historian, and the OPC Server. Major functionalities in these sub-systems are implemented in software. For example, the HMI is a software application powered by the Rockwell *FactoryTalk* software suite. Software applications consume computing resources to execute. Therefore, the availability of computing resources, such as processor time, memory, disk usage, and network access, directly affect the performance of these software applications. A lack of computing resources will delay the execution of the HMI software, which in turn delays the display of the manufacturing process information.
To measure the performance of the computing resources, the Microsoft Resource Monitor and the Microsoft Performance Monitor are used. These tools are included in the Windows 7 installation and have access to many computing resources and Windows operating system statistics. Another Microsoft tool, TCPView, is used to capture network usage and TCP connections per software application.
Since each computer has a multi-core processor, the software application being measured is pinned to run in a specific core that has the lightest load. To perform the measurement, a data

collection set is first defined in the tools describing which KPI and the specific processor core to record.

Key Performance Indicator (KPI)	Description
CPU Utilization	Measures the percentage of central processing unit (CPU) utilized over a specific amount of time.
Memory Utilization	Measures the percentage of memory utilized over a specific amount of time.
Network Throughput	Measures the mean rate and standard deviation of packets transmitted and received per software application.

Table 3: Computing Resource Key Performance Indicators

[7] **Industrial Protocol Performance** KPI measures the performance of the industrial communication network. The industrial protocol being used in the Process Control System is DeviceNet, which handles the communication between the Plant Simulator and the PLC. It is based on the Controller Area Network (CAN) protocol, a serial message-based communication protocol, with an additional application and physical layer specification.
In understanding the performance of the DeviceNet, it helps to better understand the low-level details of the industrial control network, and this information can help to identify the performance impact on the system.
To measure the performance of the industrial protocol, logging capacity is added to the software DeviceNet interface. The interface timestamps and captures all inbound and outbound DeviceNet traffic. A post processing is performed to compute the desired KPIs.

Key Performance Indicator (KPI)	Description
Network Utilization	Measures the percentage of network capacity utilized over a specific amount of time.
Packet Path Delay	Measures the time delay along the path from transmitter to receiver.
Packet Rate	Measures the rate of packets transmitted and received over a specific amount of time
Data Size	Measures the number of application payload data in bytes in a packet.

Table 4: Industrial Protocol KPIs

To reach the hard-real-time requirements of the highly deterministic industrial applications the networks are migrating from fieldbus to Ethernet technology.

1.1.1.3 Communication Requirements and Timing Constraints

The deterministic industrial applications cited in section 1.1.2 have different communication requirements. A comparison is shown in Table 5. As a subset of the requirements cited in section 1.1.3, the focus in the comparison is on the:

- Synchronization Accuracy in microsecond [µs]
- Spatial dimension (of the network) in meter [m]
- Number of communicating nodes
- Payload per Node in Byte
- End-To-End delay in microsecond [µs]
- End-To-End delay variation (jitter) in nanosecond [ns]
- Cycle Time in millisecond [ms]

[7] Typical values of the requirements for the deterministic industrial applications are listed below.

Industrial Applications	Cycle [ms]	Sync. Accuracy [µs]	No. Nodes	Payload/Node [Byte]	Distance [m]	Topology
Condition Monitoring	100	1	1000	300	1000	Star – Tree
Process Automation	10 - 100	1000	300	1500	100	Star – Tree
Machine Tool	0,5	0,25	50	30	7	Line - Ring - Baum
Packaging Machines	1	1	100	50	5	Line - Ring - Baum
Printing Machines	4	0,25	200	50	25	Line - Ring - Baum

Table 5: Overview - Communication Requirements of deterministic industrial Applications

1.1.2 OSI Model

The Open Systems Interconnection (OSI) model, created by the International Organization for Standardization, refers to a conceptual framework used to categorize the functions of a telecommunication or computing system. It supports the development of diverse but interoperable communication protocols. Any communication system can be divided into seven logical layers, wherein each layer serves a specific purpose [8]. The table below lists the different layers and elaborates on their functions as well the unit of information transmitted at the level.

Layers	Protocol data unit (PDU)	Function
7. Application	Data	High level APIs, including resource sharing, remote file access
6. Presentation		Translation of data between a networking service and an application including character encoding, data compression and encryption/decryption
5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmission between two nodes
4. Transport	Segment (TCP)/ Datagram (UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
3. Network	Packet	Structuring and managing a multi-node network including addressing, routing and traffic control.
2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium

Table 6: OSI-Model - Layers Description

1.1.3 Automation Pyramid

The automation pyramid (Figure 1) is also a conceptual framework that illustrates the different industrial communication levels based on certain performances and applications.

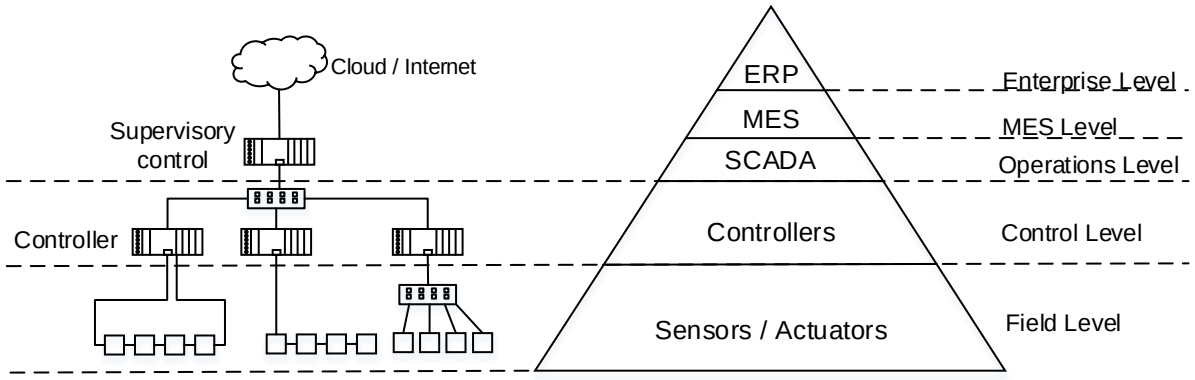


Figure 1: Automation Pyramid and hierarchical structure of industrial manufacturing

The deterministic industrial applications can be mapped to the automation pyramid as shown in Figure 1. For simplification purposes, the three upper levels have been illustrated as unified (to a single level): Plant-Level.
Extensive research has been conducted in the applications at the lower levels (control and field levels) through the development of fieldbus and Industrial-Ethernet protocols. These solutions offer different real-time performances. As Ethernet is the focus technology of this work, its functionalities and communication performances are analyzed in the next section(s). Industrial communication networks are inevitably heading in the direction of Industrial Ethernet, to benefit from the continued progress of the IEEE Ethernet technology [9]. High performance requirements, the need for a consolidation between the operational technology (factory systems) and IT-systems (OT-IT convergence) as well as the appeal of Industrial Internet of Things (IIoT) are added factors supporting the transition [10].
The figure below maps the timing requirements of the deterministic industrial applications to the appropriate levels of the automation pyramid.

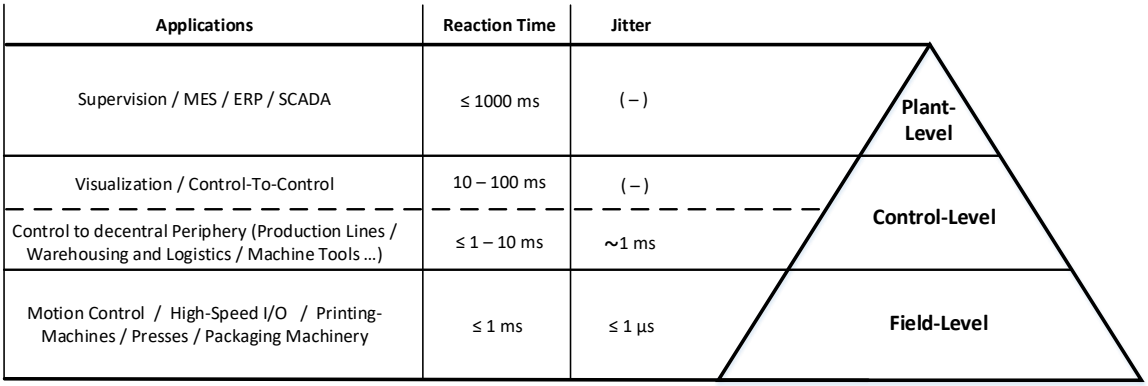


Figure 2 - Timing requirements of the deterministic industrial-applications mapped to the levels of the automation pyramid ([1] and [2]) [11]

1.1.3.1 Closed loop control systems

A closed loop control system is essentially composed of sensor(s) and actuator(s) that are connected over a communication network to a centralized controller device. Time- and safety-critical control data is exchanged between the devices. Depending on certain communication mechanisms different timing performances can be reached. Authors differentiates basically between

- event-driven configuration: A node starts an activity only when an event occurs. E.g. when an information from another node is received and
- clock-driven configuration: A node starts an activity at predefined time. E.g. a node runs periodically [12].

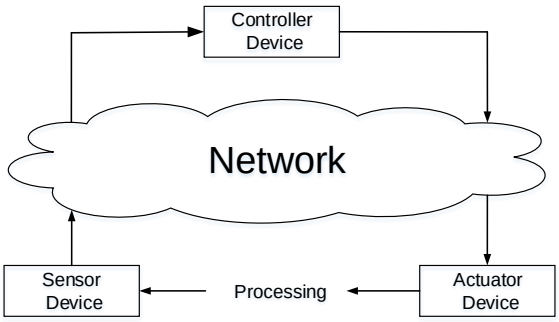


Figure 3: Closed loop control system

1.1.3.2 Established Industrial Communication Technologies

The statistics show the market shares for fieldbus and Ethernet interfaces in the field of industrial manufacturing automation worldwide in 2017. Experts from the company HMS classified industrial networks into the categories of wireless (6%) and wired (94%). The market share of wired Industrial networks are almost equally divided into various fieldbus technologies (blue bars, 48%) and industrial Ethernet systems (grey bars, 46%).

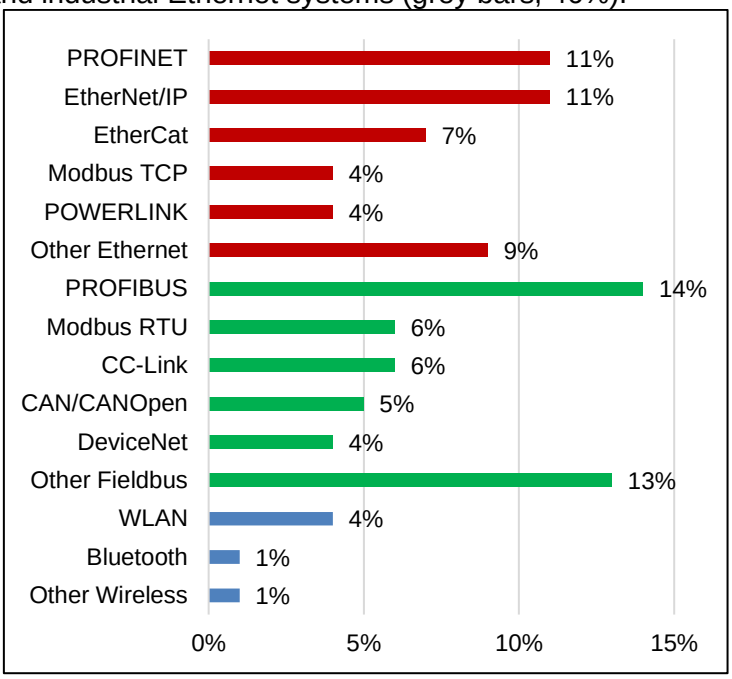


Figure 4: Market shares of the industrial networks worldwide for 2017 [13]

According to the statistics, industrial Ethernet is growing faster than previous years, with an annual growth rate of 22%, compared to 4% for fieldbus technologies and 32% for wireless technology. [12]
Owing to the indisputable importance of industrial Ethernet in deterministic industrial communication, it has been selected as the focus of the thesis.

1.2 Ethernet

Ethernet is a commonly used local area network (LAN) technology, specified in the IEEE 802.3 standards family. Although it was originally developed for LAN applications, it has over time evolved into a networking technology with much higher capabilities, supporting the communication of time-critical data in various sectors, including industrial and automotive [14]. Its evolution over time and the add-ons are described in this section.

1.2.1 IEEE Ethernet Evolution

Since its initial standardization in 1985, standard Ethernet has undergone a multitude of improvements (Figure 5) to enhance its specifications in multiple domains e.g. office-, audio/video- , avionic, automotive and industrial communication.

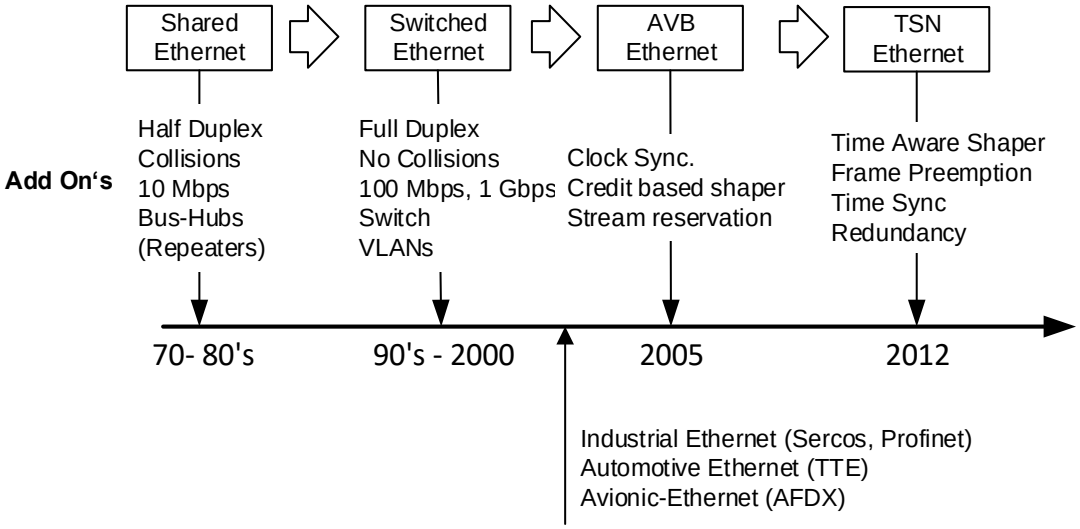


Figure 5: Ethernet Evolution

1.2.1.1 Shared-Ethernet

Collisions on an Ethernet medium can lead to a significant drop in throughput, owing to data corruption and requiring retransmission of the data. It occurs when two stations attempt to transmit simultaneously over a channel. At its initial phase, Ethernet addressed the collision problem using the carrier sense multiple access with collision detection (CSMA/CD) scheme. CSMA/CD is acceptable for smaller networks but not very effective for larger networks [14].

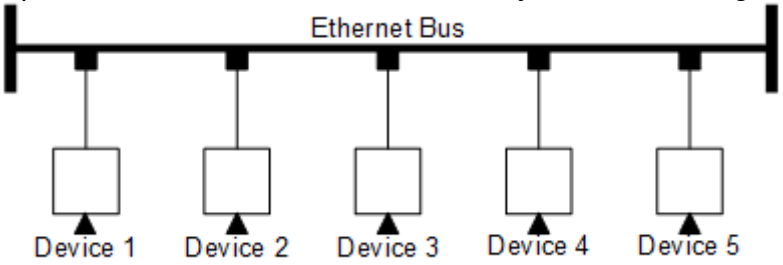


Figure 6: Original Ethernet implementation: Shared-Medium collision-prone

Introduction of Repeaters and Hubs

For the purpose of extending an Ethernet network, coaxial segments and repeaters proved insufficient. Long length of coaxial cable segments inherently degrades the signal and fails to meet the timing requirements. Although using repeaters can double the size of the network (two-port), support star topology (multi-port repeaters) networks and resolves some of the concerns with Ethernet segments, the network size limitation is still not acknowledged. Since a repeater broadcasts received traffic to all ports, it makes the network more prone to collisions [14]. It also does not allow upgrading segments to higher link-speeds. The introduction of hubs in 1985 allowed the regeneration of signals and supported collision detection at each port [15].

1.2.1.2 Switched-Ethernet

Ethernet switches were able to resolve the collision problem entirely by adhering to the MAC protocol (i.e. queueing, if a frame is already in transmission at the port, until the port is available) and moving away from the broadcast method used by hubs [Ethernet-Based Real-Time and Industrial Communications – D. New Evolutions]. Switched-Ethernet, sometimes

called bridged-Ethernet, enabled the switches (or bridges) to build address tables, using the source addresses of the received frames, to identify the port it had to be forwarded to. Unlike repeaters, bridges can mix speeds and are not restricted by a limit on the number of segments between two points [14]. The frames are completely received by the switches, checked and then forwarded.

1.2.1.3 AVB-Ethernet

The increasing demand for high synchronization and preference of Ethernet technologies in the audio-video sector, led to the formation of the Audio-Video Bridging (AVB) group in 2005. AVB extended the functionality of Ethernet to include the Credit-Based Shaper (CBS), on top of the existing Strict-priority Algorithm (SPA) of the IEEE 802.1Q standard, however, were still unable to reduce the interference delay to their satisfaction. In the meanwhile, Ethernet started becoming more appealing to the industrial, avionic and automotive sectors, which led to the eventual transition of the AVB group to the Time-Sensitive Networking (TSN) task group. With the introduction of the nine TSN standards, Ethernet was finally capable of solving the collision problem with the time-synchronization and traffic scheduling standards [15].
The Audio Video Bridging task group were mainly focused on audio and video streaming applications using IEEE 802 networks. They specified four IEEE 802.1 standards, two of which were amendments to IEEE 802.1Q (“Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks”) [14] and to IEEE802.1AS. The prime outcomes of the AVB-standards for Ethernet-based systems is the improvement in the clock synchronization of the communicating nodes and reduction in the network delay.
[16] The key challenges in the development of AVB were:

- Centralized Control is better than Decentralized Control for industrial applications
- Need for a new traffic shaping method to eliminate the congestion loss
- Expanding the market from audio-video sector to control system applications of industrial and automotive sectors brings new challenges and requirements

1.2.1.4 TSN-Ethernet

Time-Sensitive Networking (TSN) is an IEEE 802.1 task group formed with the charter of providing deterministic services through IEEE 802 networks, i.e., guaranteed packet transport with bounded low latency, low packet delay variation, and low packet loss. The TSN Task Group has been evolved from the former 802.1 Audio Video Bridging Task Group and extended its application fields from the transmission of time-sensitive audio-video data to the transmission of time-sensitive and safety-critical control-data in the industrial and automotive networks.
A brief overview of the most fundamental TSN-standards and enhancements are illustrated below.

IEEE Standards	Title	Description
802.1 AS-Rev	Timing and Synchronization for Time-Sensitive Applications	Defines time synchronization mechanisms for faster fail-over of clock grandmasters
802.1 Qbv	Enhancements for Scheduled Traffic	Planning transmission time points and reserving time-slots to transmit scheduled data traffics and to eliminate the interference with the non-scheduled traffic and to reduce the end-to-end delay and guarantee an ultra-low jitter.
802.1 Qbu	Frame-preemption	Preempting the transmission of low-priority traffics to reduce the interference delay of the time-critical traffic
802.3 br	Interspersing Express Traffic	
802.1 Qcc	Stream Reservation Protocol (SRP)	Specifies new enhancements and improvements for stream reservation

	Enhancements and Performance Improvements	
802.1 Qca	Path Control and Reservation	Identification and control of redundant communication paths and reservation
802.1 CB	Frame Replication and Elimination for Reliability	Handling the redundant transmission of data frames (frame replication and elimination)
802.1 Qci	Per-Stream Filtering and Policing	Configurable limitation of the bandwidth utilization
802.1 Qch	Cyclic Queuing and Forwarding	Peristaltic traffic shaping to forward isochronous traffics in the same cycle

Table 7: Brief description of the TSN-Standards and -Enhancements

[16] TSN enables control systems to build a converged network, in which time-critical, mission-/safety-critical and best-effort data traffics can share the same network resources while maintaining the timing performances of highly deterministic applications. The key benefits of TSN are as listed below:

- **Determinism** – Guaranteed upper-bound latency, ultra-low delay variation (jitter) and zero congestion loss for all critical control loops.
- Enabling higher **link-speeds** (1Gbps, multi Gigabit)
- **Convergence** of different traffic classes on a single network
- Familiar, widely used, open, interoperable standard with continuous development, which ensures long-term availability, innovation and lower costs

The thesis focuses on the first three IEEE TSN standards, as listed in table 7. Unlike the other TSN amendments, their specifications completed during an early stage of the thesis and were available in hardware prototypes.

1.2.2 Ethernet Frame Format

The Ethernet frame format, as defined by IEEE802.3 [17], is basically composed of an Ethernet header (26 Bytes) and a payload field (46-1500 Bytes) followed by 12 Bytes transmission pause named Inter-Frame Gap (IFG).
The Ethernet data fields illustrated in figure 7 are described below:

- ***Preamble*** – Used to inform the recipient station of an incoming Ethernet frame; Seven Bytes of alternating zeroes and ones [18]
- ***Start Frame Delimiter*** – One Byte as per the Ethernet 802.3 standard ending with two consecutive ones for the purpose of synchronization
- ***Destination and Source MAC addresses*** – The 6-Byte destination MAC address may be a unicast, multicast or broadcast address, while the 6-Byte source MAC address is inherently always a unicast address.
- ***VLAN Tag*** – is an optional data-field that contains provisions for quality of service prioritization and define virtual network ID
- ***EtherType*** – Specification of the upper-layer protocol used after the Ethernet overhead processing has completed
- ***Length*** – Specifies the numbers of bytes of data ensuing
- ***Payload*** – The actual data used by an upper-layer protocol (as specified in the EtherType field); Required to be at least 46 Bytes
- ***Frame Check Sequence (FCS)*** – 4-Byte cyclic redundancy check (CRC) value used to detect corrupt frames; Generated at the transmitting station and recalculated and verified at receiving station

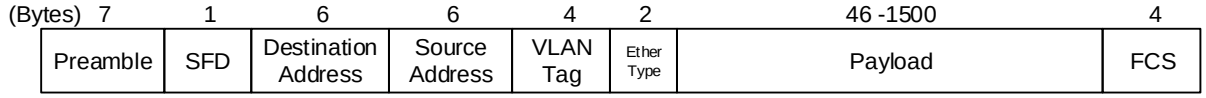


Figure 7: IEEE802.3 VLAN-tagged Ethernet frame formats

1.2.3 Real-Time Ethernet

The evolution of Ethernet in terms of collision avoidance, frames prioritization, latency reduction, and higher bandwidth made it an attractive technology for use in deterministic sectors such as industry, automotive and avionic to transmit time-critical control traffics.
Depending on the timing requirements of each application field, certain Ethernet features and capabilities are activated.

1.2.3.1 Industrial-Ethernet

[19] For companies in the automation industry, the development of real-time Ethernet to connect devices is of high economic interest to replace conventional fieldbus systems. Therefore, many approaches for adapting Ethernet to real-time requirements come from industrial applications.
[20] Industrial-Ethernet is the use of Ethernet in an industrial environment combined with communication protocols that provide determinism and real-time control to meet the timing requirements of the deterministic industrial applications.
Problems with the switched Ethernet
A bounded transmission time of a frame cannot be guaranteed using the switched Ethernet MAC. A fair treatment of the different data traffics by the MAC is difficult, referred to as the capture effect [21].
Certain Industrial Ethernet protocols used the IEEE802 Ethernet standards without modification to meet the timing requirements of the applications. Others modified the Ethernet MAC layer, which require hardware support.
More than twenty Industrial Ethernet protocols have been created since the integration of Ethernet in deterministic industrial applications. A survey of their performances, advantages and disadvantages is featured in [19].

1.2.3.2 Avionic-Ethernet

"The Avionics Full DupleX Switched Ethernet (AFDX) was designed with the avionics industry in mind. The network is based on full-duplex switched-Ethernet for real-time applications. A data stream is referred to as a Virtual Link and the network is typically of multicast streams sporadic in nature." [22] Avionics Full DupleX Switched Ethernet (AFDX) is a real-time switched Ethernet network that has been defined specifically for avionics sector. It avoids the signal collision by supporting the full-duplex mechanism. The exchanged traffic flows on an AFDX network are multicast sporadic flows called Virtual Link.

1.2.3.3 Automotive-Ethernet

[23] Time-Triggered Ethernet, often called TTEthernet or TTE, is a computer network technology that expands standard-Ethernet with features and functionalities to meet requirements of time-critical and safety-related applications primarily in the automotive and aerospace sectors.
TTE defines three traffic classes:

- **Time-Triggered (TT) Traffic** − are scheduled Ethernet frames that are sent at predefined transmission time-points and take precedence over all the remaining traffic types. TT messages allow designing strictly deterministic distributed systems but require a system wide synchronized time base.
- **Rate-constrained (RC) Traffic** − are used for applications with less stringent determinism and real-time requirements. These messages guarantee that bandwidth is predefined for each application and temporal deviations have defined upper bounds limits. RC messages are typically used for multimedia systems
- **Best-Effort (BE) Traffic** − have no timing guarantees whether and when they can be transmitted. They use the remaining network bandwidth and have the lowest priority. Typical applications are web services and applications without QoS requirements

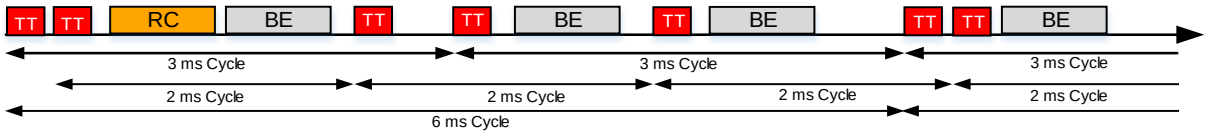


Figure 8: Traffic classes of TTEthernet: TT, RC and BE [23]

1.3 Problem Statement and Thesis Contribution

1.3.1 Scientific Gaps

TSN has not yet been throughly evaluated specifically for the integration in deterministic industrial communication, which is considered an important sector to increase the range of TSN applications.
In order to be integrated into the automation industry, certain communication mechanisms, such as frame structure (individual and summation), switch forwarding mechanisms (store&forward, cut-through), and network topologies (line, ring, star and tree), need to be taken into consideration for the timing analysis of the deterministic TSN features.

The large number of IEEE TSN-standard-enhancements and the complex mechanisms complicate the selection of the required features for integration in the appropriate Industrial Ethernet Protocol. Currently the industrial sector is divided into two main groups that aim to utilize TSN for real-time improvements in two different and incompatible ways.

- The first group aims to support IEEE802.1Qbv to schedule time-critical control data traffic throughout the whole network. The goal of this group is to reach the highest timing performance, ultra-low jitter, delays and cycle time, by eliminating the interference with the background traffic. However this approach may require complex configuration and planning of the network. Further, some of the available bandwidth could be wasted either by not completely using the guard-band window for transmission or by scheduling too large time-slots.
- The second group aims to support IEEE802.1Qbu and IEEE802.3br in combination with 1Gbps link-speed (Section 3.3), which reduces the interference delay with non-time-critical traffic. Nevertheless, interference cannot be completely avoided. Unlike the previous approach, no complex configurations are required.

Neither approach is ideal but each have their benefits for the appropriate fields of application. Therefore a deep timing analysis of integrating IEEE802.1Qbu and IEEE802.1Qbv in established Industrial-Ethernet protocols is required. Based on concrete integration scenarios through extensive simulation and TSN prototypes, the benefits and limitations of TSN are analyzed.

Multiple TSN prototypes and test setups (e.g. LNI 4.0, IIC TSN test setups) [Reference for Test setups] have been presented in different fairs and plug-fests the last few years. For integration in the industrial-sectors, the TSN-features and the hardware-prototypes need to be evaluated. This article presents a test-plan to evaluate the following TSN-Sub-Standards: IEEE802.AS, IEEE802.1Q, IEEE802.1Qbu, IEEE802.3br and IEEE802.1Qbv.

The scientific questions that this work is investigating are listed below:

- *Are the deterministic TSN-features defined in IEEE802.1Qbv and IEEE802.1Qbu sufficient to reach the hard-real-time requirements of highly-deterministic industrial-applications such as "High-Speed I/O" and "Motion-Control"? Which are the missing features? Which TSN-mechanism(s) can be integrated in which Industrial-Ethernet protocol(s)?*

- *How to evaluate the TSN features and prototypes for use in the industrial-communication?*
- *In which level(s) of the automation pyramid and for which deterministic industrial-applications can TSN be integrated? What are the challenges and missing features?*

1.3.2 Related Works

[9] discussed in 2007 the use of Gigabit Ethernet as the last possible approach to decrease the cycle time of the Industrial-Ethernet Protocols from the highest real-time class. However the analysis covered only the two industrial-Ethernet protocols EtherCAT and Profinet IRT that support cut-through. The authors presented in [24] a new approach to reduce the propagation delay and the frame transmission duration in a PROFINET network. The concept is not suitable for other Industrial-Ethernet Systems and is not IEEE conform since it requires a modification of the Preamble and the MAC address.
The performances of more Industrial-Ethernet Protocols were compared in [25] without the context of TSN.

The author of this thesis presented the Sercos over TSN concept in [26] and simulation results [27] of improving the timing performances of Sercos III by increasing the link-speed, integrating the TSN standard-enhancements IEEE802.1AS-Rev and IEEE802.1Qbv as well as migrating from line/ring to tree topology. The concept requires a time-aware TSN-switch between the Master and the slaves to enable higher topology flexibility and to reduce the cycle time.

To analyze the impact of the cross-traffic (Best-Effort) on the robustness of a deterministic system, the two real-time Ethernet approaches: AVB[1] Credit-Based Shaper (CBS) and TTE[2] Time-Triggered Ethernet were compared in [28]. It was proven that the traffic scheduling approach can guarantee better timing performances than reserving transmission credit. The authors recognized an automotive network with few devices.
[29] presented a formal analysis of IEEE802.1Qbv and IEEE802.1Qbu for use in automotive. The nonuse of tight time synchronization as well as the soft-real-time configuration of 10 ms cycle times for few number of nodes made the results not appropriate to the highly deterministic control systems. However it was shown that the results reached by frame-preemption and frame-scheduling are very close, which favorites the use of IEEE802.1Qbu for the automotive sector, since no set-up complexity is required.

[30] presented a concept to integrate TSN in the high-deterministic telecommunication application “Common Public Radio Interface” (CPRI). The simulation results compared the performances of IEEE802.1Qbu and IEEE802.1Qbv. It was shown that the ultra-low CPRI jitter requirement of 8.138 ns can only be met with the frame scheduling approach. Frame-preemption was not suitable even with high link-speeds of up to 40Gbps. The CPRI jitter requirements exceeds by far the jitter requirements of all deterministic industrial applications, which are mostly met with a few hundred nanoseconds or a few microseconds. The use of complex tree topology with high link-speeds of up to 40Gbps make the experiment configurations not comparable to the industrial control systems (mostly 100Mbps).

None of the known works evaluated TSN in the context of automation communication or analyzed the integration of TSN in Industrial-Ethernet and its combination with certain features such as Gigabit, topology migration, cut-through forwarding, summation frames, etc. Furthermore the coexistence of TSN-capable devices and legacy devices is an important factor that still not been covered. TSN adds a set of enhancements to standard Ethernet and thus enables Ethernet capabilities such as multi-gigabit and higher topology flexibility. The thesis aims to cover these gaps and gives an answer to the question whether TSN is absolutely required on the field-level to improve the timing behavior of multiple Industrial-Ethernet protocols.

1.3.3 Contributions

The thesis presents surveys on the

- Timing requirements of multiple deterministic industrial applications
- Cycle Time Modeling of Industrial-Ethernet Protocols
- Impact of frame structure on the efficiency of bandwidth-utilization
- Impact of the frame forwarding mechanisms on reducing the delay
- Mapping the deterministic TSN-features to the industrial applications
- TSN-integration scenarios in the field- and machine levels of the automation communication

The TSN-evaluation can be illustrated as

- Designing a formal-, simulation- and experimental evaluations of the following TSN-standards and enhancements:
  - IEEE802.1AS
  - IEEE802.1Q
  - IEEE802.1Qbu & IEEE802.3br
  - IEEE802.1Qbv
- Designing Test-Plans for an experimental Evaluation to verify the compatibility and measure the performance of the TSN prototypes in real-operation
- Formal and simulative performance comparison of the following Industrial-Ethernet protocols: Sercos III, Profinet/IRT, Profinet RT and Ethernet/IP. Multiple network scenarios (multiple topologies, link-speeds, cycle time model, frame structures, frame forwarding mechanism, etc…) have been used.
- TSN-Integration scenarios in the Industrial-Ethernet communication
  - Development of new concepts to improve the timing behavior of Industrial-Ethernet Protocols with TSN and / or other communication features such as increasing the link-speed and supporting cut-through instead of store & forward.
  - Designing a concept for hardware-adaptors to tunnel frames of “specific” Industrial-Ethernet Protocols (e.g. Sercos III) through a TSN-sub-network
- Presenting an evaluation matrix to map the TSN-features to the appropriate applications and levels of the automation pyramid

1.4 Analysis Approaches

[31] classified the existing approaches for worst-case delay analysis of a real-time switched Ethernet network into three main categories

- End-to-end delay distribution - calculation of the end-to-end delay based on various scenarios, that results in a distribution of the delays with the maximum pertaining to the worst-case delay. Simulation is generally the opted approach for this analysis.
- Upper bound end-to-end delay – computation of a definite upper bound of the end-to-end delay. This is obtained using the formal approach (using for example, Network Calculus) and gauge the upper bound of all the parts of the delay. It may be possibly scaled.
- Exact value of worst-case end-to-end delay – A Model Checking (MC) approach is used to design different scenarios in order to determine an accurate value of the worst-case end-to-end delay. It is a formal approach and cannot be scaled.

This thesis used the following three approaches:

- **Formal approach**: has been derived from the specifications of multiple Industrial-Ethernet specifications that are considered in this work. The formal approach presents

formulas to compute the timing KPIs of the TSN-features and the Industrial-Ethernet protocols: e.g. bounds of end-to-end delay, forwarding delay, cycle time and minimum cycle time, etc.

- **Simulation approach**: computes the jitter (delay variation) and the channel throughput for complex network scenarios, of which only the bounds can be mathematically computed.
- **Experimental approach**: is a TSN-test setup composed of TSN-prototypes from multiple vendors to **verify** the formal and simulative results of the same communication scenarios and to **evaluate** the performances, interoperability and standard-conformance of the prototypes.

The results of the simulation and experimental approaches need to be within the range given by the bounds of the formal approach. Values outside of the range were analyzed and used to improve the test setup.

1.4.1 Formal Analysis Approach

The objective of the formal approach [9] - [24] - [25] is to compare the timing performances of the established Industrial-Ethernet protocols and the new TSN-features IEEE802.1Qbu and IEEE802.1Qbv based on the computation methodologies of the performance indicators.

1.4.2 Simulation Analysis Approach

Simulation experiments are configured based on models of a certain system. The main purpose of simulation is obtaining information regarding the behavior of the system under a predefined setup. Since numerous configurations of industrial networks are possible, the simulation is focused on a specific setups that have a relevant impact on the selected metrics [31].

1.4.2.1 OMNEST / OMNeT++

OMNEST or OMNeT++, referring to the commercial or academic free license versions respectively, is a framework for discrete event simulation. Being modular and object-oriented, it is a convenient tool for performance evaluations of different communication aspects and systems. It allows for a range of functionalities and applications. As well as modelling communication networks (both wired and wireless), it can also be used to model queueing, protocols, and distributed hardware systems [OMNeT++ Simulation Manual Documentation – Section 1 Introduction].INET Framework

- The INET Framework further supports the development of communication networks, providing models for protocols (Ethernet, IEEE 802.11, etc.), applications, components etc. It is a model library, which is open-source and offers the following [32]: link-speeds of 10Mbps, Fast-Ethernet, Gigabit and Multi-Gigabit-Ethernet,
- Full-duplex mechanism
- Multiple device-models: as well as hub model (EtherHub) and switch model (EtherSwitch), node model (EtherHost).
- And component-models: e.g. MAC model (EtherMAC), LLC model (EtherLLC) and frame model (EtherFrame).

The existing implementation of the physical and MAC layers of Ethernet makes INET and OMNeT++/OMNEST a very attractive solution for researchers to focus on building up new features and mechanisms on top of the existing modules.
The thesis used OMNEST and INET frameworks and built up multiple IEEE TSN-mechanisms and Industrial-Ethernet features (e.g. cut-through and fast-forwarding approaches) upon it.

1.4.3 Experimental Analysis Approach (Test setup)

In the course of the thesis, a test setup is built parallel to the IIC TSN test setup. The test setup is composed of measurement equipment (network TAPs and oscilloscope), TSN hardware prototypes (I210 Ethernet controller, switches and industrial-PC's as end-devices) from multiple vendors and a configuration host. The goal of the test setup is to evaluate the prototypes in terms of timing performance, standard compliancy and interoperability. In addition, the formal and simulative results of certain models and network scenarios are checked against the real-hardware.
Implementation prototypes of the investigated TSN-features and the communication- and network-scenarios are used for evaluation of the solution approach, including the formal-computation and the simulation-models. Due to given scalability constraints of the prototypes and missing features (e.g. cut-through), a realistic simulation model of the industrial networks is mandatory.

The formal-approach computes the upper-bounds of the end-to-end delay and jitter of the frames and gives an important range for the simulative and experimental results. Values above and / or below the bounds help to evaluate the performance of the prototypes and to adjust the simulation and the prototypes implementation if necessary.

2 Background and State of the Art

This chapter gives an overview about the state-of-the-art, which is fundamental to this thesis. Section (2.1) handles the classification, performance and functionality of Industrial-Ethernet. Section (0) describes the specifications of the deterministic TSN-Standards for use in the automation industry. The last two sections (2.3) and (2.4) give a mathematical model for the timing analysis of the topics in (2.1) and (0) and a survey on the communication features and their impact on the key performance indicators for an industrial communication.

2.1 Industrial-Ethernet

2.1.1 Real-Time Classes

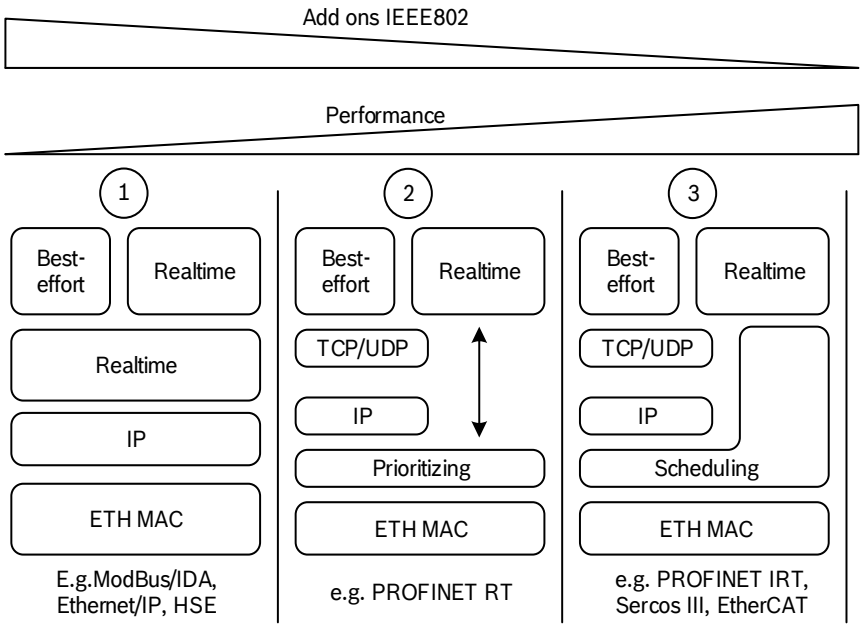


Figure 9: Classification of industrial-Ethernet Protocols

Multiple Industrial-Ethernet Protocols with different communication approaches and performances have been developed and discussed in the literature. [9] classified the protocols into three main categories, as shown in Figure 1. The timing performance is increasing from category 1 to category 3, requiring additional functionality at the MAC layer.
The **first category** uses standard, unmodified-Ethernet hardware as well as standard TCP/IP software stacks for process communication and adds an industrial-automation specific application layer on top of it. Typical representatives are Mod-Bus/IDA and Ethernet/IP. Because of using the whole TCP/IP protocol stack, high timing performances cannot be reached. Typical cycle times of ~10 to 100 milliseconds can be achieved. However, some implementations improved the TCP/IP stacks to provide better performances.
Industrial communication systems of the **second category** are using standard, unmodified Ethernet hardware with the priority scheme at the MAC layer according to IEEE802.1D/Q [33] and bypass the transport and network layers (no TCP/IP or UDP/IP stacks are used) for the transmission of time-critical data to improve the performance. Cycle times in the range of 1 to 10 milliseconds can be achieved.
Further improvements can be achieved by modifying the Ethernet MAC layer with the traffic scheduling, cut-through and/or modify on the fly procedures. This requires specific hardware or software support. Protocols scheduling the time-critical traffics belong to the **third category**. Cycle times below one millisecond can be achieved. Typical representatives are Sercos III, Profinet IRT and EtherCAT.

2.1.2 Communication Features

This section describes the most important communication features affecting the metrics and key performance indicators of the deterministic industrial applications.

2.1.2.1 Minimum Cycle Time

In industrial control systems, the controller, e.g. PLC acts as a master by initiating all communications with the connected slave devices (sensors and actuators). The communication is characterized by its controller cycle time period [25], which is divided into three phases, as shown in Figure 10:

- *Phase 1: Input Reading* – Is the time needed to collect new sensor data and to be refreshed in the controller memory. Further called upstream communication
- *Phase 2: Processing Time* – Is the time needed by the controller to process the sensor data and generate new command data
- *Phase 3: Output Writing* – Is the time needed to transmit and update the actuator(s). Further called downstream communication

Phase 1 and 3 represent communication time, which should be less than the difference between the controller cycle time and the processing time. Unlike Phase 2, the Input reading and output writing phases depend on the used communication features. Multiple approaches, such as TSN, Gigabit, forwarding mechanism etc… are used to reduce these two phases. However, the time performance of each Ethernet product was compared according to a constraint named minimum cycle time, which was defined in [25] as: *"The minimum cycle time was the communication time required by the controller to both collect and update the data memories of all sensors and actuators."*

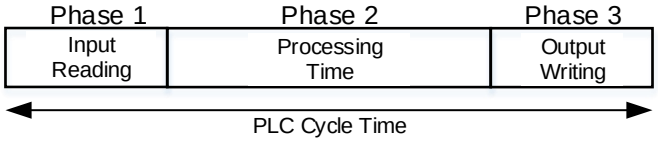


Figure 10: PLC cycle time

[9] The line topology is very important at the field level of industrial automation systems. Unlike the office communication, the wiring structures in the industrial automation domain must follow mechanical conditions and cabling channels of machines and plants. This leads to a structure of a line topology (...) When using Ethernet, an industrial control system may have tens or even up to hundreds of active nodes or switches cascaded.

Figure 11 illustrates a PLC acting as a Master connected to 5 slaves in a line topology. Shown are the frames of the up- and downstream communications using store&forward (left) and cut-through (right). The PLC cycle time is computed as follows:

PLC Cycle Time computation – Store&Forward

$d_{UL} = d_{Rx} + d_{fwd} * (n_{dev} - 1) + \sum_0^{n_{Interf}} d_{Interf}$; $d_{DL} = d_{Tx} + d_{fwd} * (n_{dev} - 1) + \sum_0^{n_{Interf}} d_{Interf}$

$T_{PLC} = d_{UL} + d_{PLC\_Px} + d_{DL}$; $T_{PLC(min)} = \max(d_{UL}; d_{DL})$

If the frames of the up- and downstream communications have the same sizes and do not face interferences with the background traffics, then $d_{UL} = d_{DL} = T_{PLC(min)}$otherwise the minimum PLC cycle time is equal to the maximum of both.

PLC Cycle Time computation – Cut-Through

Case 1 – if no interferences occur and the cut-through forwarding duration is less than a single frame transmission duration ($d_{fwd(CT)} < d_{Tx}$) than

$d_{UL} = (d_{Rx} + d_{IFG}) * n_{devices} - d_{IFG}$; $d_{DL} = (d_{Tx} + d_{IFG}) * n_{devices} - d_{IFG}$

Case 2 – if an interference occurs on each egress port or ($d_{fwd(CT)} > d_{Tx}$) than

$d_{UL} = d_{Rx} + d_{fwd(CT)} * (n_{devices} - 1)$; $d_{DL} = d_{Tx} + d_{fwd}(CT) * (n_{devices} - 1)$

In the second case the flow diagram supporting cut-through is similar to the store&forward (Figure 3 - left).

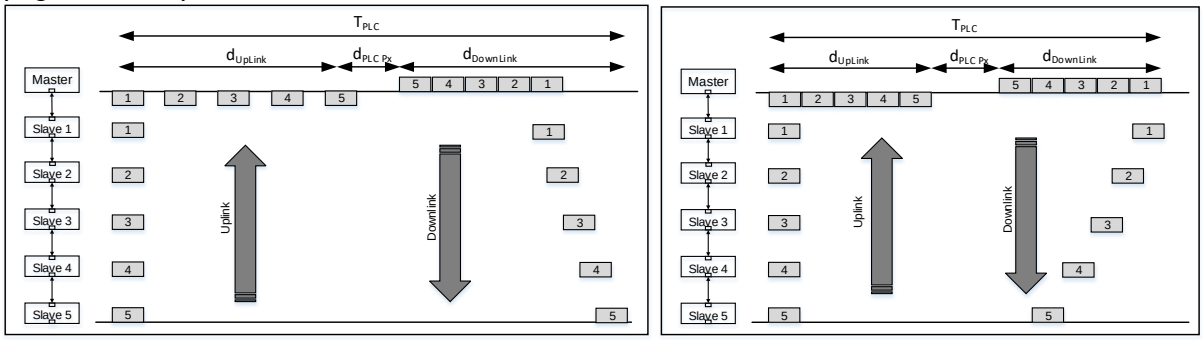


Figure 11: Synchronous PLC Cycle Time Model in a line topology using Store&Forward (left) vs. using cut-through (right)

2.1.2.2 Cycle Time Modeling

[34] introduced three CIP-Motion cycle timing models for Ethernet/IP based systems. These models includes the actuators processing time, which is depending on the software stack and is not part of the communication. Therefor it will not be further considered in this thesis.
[25] extracted the controller and actuators processing times and restricted the cycle time to the sensors and actuators update times (minimum cycle time) and introduced two main models:

- Model 1: also called 3 cycle mode in [34] presents simultaneous upstream and downstream communications: values from sensors and to actuators are updated simultaneously. The master and slave devices periodically generate and transmit their messages at the same times. Typical protocols supporting this model are: Ethernet/IP, Profinet RT, Profinet IRT, etc...
- Model 2: Downstream then upstream communication: First all actuators are updated then sensors values are collected and updated in the controller memory. Typical protocols supporting this model are: Sercos III and EtherCAT, where the controller (master) is the one who generates all frames (for reading and writing) and he slave devices read and write on the fly.

These two models are used for the evaluation of TSN and the industrial-Ethernet protocols.

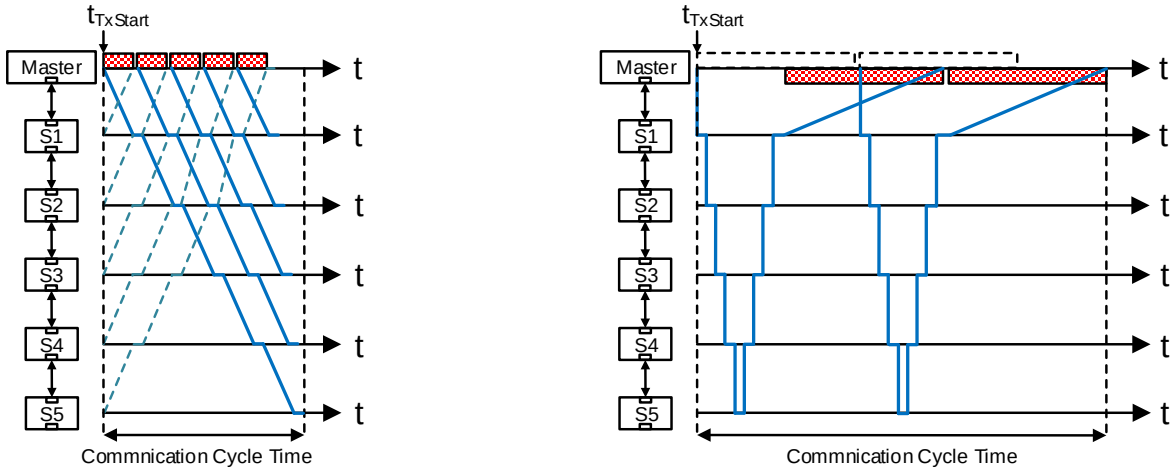


Figure 12: Minimum Cycle Time Models: (left) Model 1 - (right) Model 2

2.1.2.3 Link-speed

[9] analyzed the effect of increasing the link-speed from 100Mbps to 1Gbps on improving the timing performance (mainly reducing the cycle time) of the Industrial-Ethernet protocols Profinet IRT and EtherCAT. For a payload of 16 Byte per device and 100 devices in line, it has been shown that Gigabit-Ethernet reduced the cycle time of Profinet IRT from ~700µs to ~100µs and EtherCAT from ~400µs to ~ 160µs.
Because of its high impact on the timing performances of industrial-Ethernet, increasing the link-speed is applied in combination with TSN in this thesis.

2.1.2.4 Forwarding Mechanism

[35] „Both, Cut-Through and Store-and-Forward Layer 2 switches base forward decisions on destination MAC address of the packet. However, a Store-and-Forward switch makes the decision after the whole packet has been received. A Cut-Through switch makes a forwarding decision after analyzing the destination MAC address and depending on the specific protocol other field set in the first part of the frame.
In a Store-and-Forward switch, the latency time includes the time needed to receive the whole frame. Thus, the latency times are drastically worsened compared to Cut-Through switches. In best and worst latency cases (the node is injecting a frame into the ring or not) Cut-Through switches offer better latency times than Store-and-Forward because the frame forwarding can start immediately“.
Typical Industrial-Ethernet protocols supporting cut-through forwarding mechanism are: Sercos III, PROFINET IRT and EtherCAT with cut-through forwarding delays between 0,8 and 3 microsecond.
[36] identified the following issues and performance limitations of cut-through. Cut through is only useful when going between two ports running at a equal link-speed. Going from a slower port to a faster port, will run out of bits to transmit in the middle of a packet.

Forwarding Delay Computation
The minimum forwarding delay of a bridge [24] consists of three main components:

- physical layer (transmission and reception PHYs), typically below 300 nanosecond each
- minimum amount of data to be received in order to determine the destination port: It corresponds to the whole frame size for store&forward and few read bytes for cut-through
- logic delay for a correct forwarding of the incoming frame to the corresponding outgoing port. Typically few microseconds.

$$d_{SwFwd(S\&F)} = d_{RxPHY} + d_{Rx} + d_{SwPx} + d_{Interference} + d_{TxPHY}$$

$$d_{SwFwd(CT)} = d_{RxPHY} + d_{MinReadByte} + d_{TxPHY}$$

	100Mbps		1Gbps	
Frame Size	**Min. (72Byte)**	**Max. (1530Byte)**	**Min. (72 Byte)**	**Max. (1530 Byte)**
Store&Forward	19,76 µs	136,4 µs	14,57 µs	26,24 µs
Cut-Through	2,52 µs	2,52 µs	0,79 µs	0,79 µs

Table 8: Forwarding Delay Computation - Store&Forward vs. Cut-Through

2.1.2.5 Transmission Selection Algorithm and Traffic-Shaping

“Traffic shaping is a bandwidth management technique used in networks to delay certain traffics to bring them into compliance with a desired *traffic profile*.” [37]
IEEE802.1Q standard and its enhancements IEEE802.1Qav, specified by the AVB task group, IEEE802.1Qbu and IEEE802.1Qbv, specified by the TSN task group, define multiple transmission selection algorithms to identify time-critical traffics, favor their transmission over the remaining traffics and to reduce their end-to-end delay.

[38] “For a given Port and supported value of traffic class, frames are selected from the corresponding queue for transmission if and only if

- The operation of the transmission selection algorithm supported by that queue determines that there is a frame available for transmission; and
- For each queue corresponding to a numerically higher value of traffic class supported by the Port, the operation of the transmission selection algorithm supported by that queue determines that there is no frame available for transmission.”

IEEE802.1Q defined the **Strict-Priority Algorithm**, which classifies the traffics into time-critical with high priority and non-time-critical with low priorities. Frames are identified based on their priority field (PCP 3bits) within the VLAN-Tag. Up to 8 priorities from 0 (lowest) to 7 (highest) are possible. Frames with highest priority are transmitted first.

IEEE802.1Qav defined the **Credit-Based Shaper Algorithm** regenerated the priority and introduced two Stream Reservation (SR) classes, SR class A and SR class B. [39].
“For a given queue that supports credit-based shaper transmission selection, the algorithm determines that a frame is available for transmission if the following conditions are all true:

- The queue contains one or more frames.
- The transmission credit is ≥ 0.”

[40] The transmission credit, in bits, that is currently available to the queue. If, at any time, there are no frames in the queue, and the *transmit* parameter is FALSE, and credit is positive, then credit is set to zero. During the transmission of a frame the credit is decremented. For more fairness in the transmission of the low-priority traffics, the maximum allowed credit is 75% of the available bandwidth.
Compared to the strict-priority algorithm, the credit-based shaper does not improve the timing behavior (delay and jitter), which is mainly due to the interference with the remaining traffics.

To reduce the interference delay, the transmission preemption of non-time-critical traffics was introduced in IEEE802.1Qbu and IEEE802.3br. The two enhancements did not specified any new transmission selection algorithms but they introduced two new traffic classes: express and preemptable that are further identified through the priority field (PCP).
For certain high deterministic communication applications, meeting the hard-real-time requirements: ultra-low jitter below 1 microsecond and short end-to-end delay times are only feasible if the interference is completely avoided. That led to the introduction of the frame scheduling through the **Time-Aware Shaper Algorithm** in the IEEE802.1Qbv enhancement. The traffics are further classified into scheduled and non-scheduled. Specific time-slots and transmission time-points are predefined and protected through a guard-band window with the size of the Ethernet MTU (1542Byte). If the transmission of non-scheduled frames does not end before the start of the time-slot, they will be blocked and transmitted after the time-slot.

2.1.2.6 Frame Structure

In order to understand the motivation behind the integration of summation frames in the automation communication, the Ethernet frame structure as well as its minimum and maximum size need to be described.
Each Ethernet frame is composed of an Ethernet overhead (30Byte), optional protocol(s) header(s) (depending on the used OSI-layer for the data encapsulation), payload (42-1500Byte) and Inter Frame Gap (IFG ≥12Byte). The Ethernet overhead is composed of the following control data: Preamble (7 Byte), Start of Frame Delimiter (1Byte), destination MAC address (6Byte), source MAC address (6Byte), optional VLAN-Tag (4Byte), EtherType (2Byte) and Frame Check Sequence (4Byte).
The minimum possible Ethernet frame size is 84Byte = Ethernet Overhead (30Byte) + Protocol Overhead (Optional) + Payload (≤ 42Byte) + IFG (12Byte). In order to transmit 2Byte of payload, 40Byte of padding Bytes need to be added, which results into a disproportional high

overhead of 80Byte to transmit small user data of 2Byte. The maximum possible Ethernet frame size is 1542Byte.
Unlike office technology, the exchanged payload per device in automation technology are generally very small (e.g. status data of 4Byte). Using individual frames to each device would increase the overall overhead and leads to a poor bandwidth utilization. For example, if status data of 4Bytes per device for 50 devices were sent individually, that would take up 4200 bytes = 50*84 bytes altogether (smallest packet size with Ethernet: 64 bytes). However, only 200 bytes would be used productively for the application; ~ 5% of the bandwidth.

In order to overcome this problem, summation frames have been introduced. Only one single frame will be used to transmit all payloads from and to all devices. The devices, which are involved in the cycle real-time communication, receive their data at a predefined position of the total frame [9]. This basic concept is well-known from fieldbus systems like INTERBUS [41] and is further used in multiple Industrial-Ethernet protocols such as Sercos III, EtherCAT and PROFINET IRT. “In Sercos frames for example, up to 1,494 bytes of all device user data is packed together with an additional 56 - 44 bytes of overhead (depending the Sercos frame type). With packets that are a maximum size of 1,542 bytes, the bandwidth available for productive data increases up to 97%.” [42]
[24] introduced the *dynamic frame packaging* approach to increase the performance of the Industrial-Ethernet system PROFINET. The basic idea of this approach is to transmit a summation frame containing payloads to all devices. Each device extracts its own data field and forward the shortened frame. However the approach has not been supported by other Industrial-Ethernet protocols.
To summarize three frame structures are used in Industrial-Ethernet protocols: Individual (most popular), summation and dynamic. The last two are mainly useful for the daisy-chain line or ring topologies and small payloads.

2.1.3 Industrial-Ethernet Protocols

In order to cover the integration of TSN in the industrial communication, four Industrial-Ethernet protocols with different communication mechanisms and timing performances are selected for analysis. The following “typical” representative(s) of the three real-time class are listed below:

- Ethernet/IP from the first real-time class
- Profinet RT from the second real-time class
- Profinet IRT and Sercos III from the third real-time class

The protocols are compared based on the *real-time class* they belong to, the achievable *cycle time performance*, supported *minimum cycle time model*, *transmission selection algorithm*, network *topology*, *frame structure* for time-critical data transfer, *forwarding mechanism* and *synchronization protocol*.

2.1.3.1 Ethernet/IP

Ethernet/IP is an industrial-Ethernet protocol from the **first real-time class** that was initially released in 2000 by Rockwell Automation and ODVA (Open DeviceNet Vendors Association). Ethernet/IP runs on standard Ethernet hardware and uses TCP/IP to transfer “explicit” query/reply telegrams for configuration and data acquisition. Real-time “implicit” I/O messages are sent to each device **individually** using UDP/IP with compacter format and smaller overhead (Figure 5). The 82Byte overall frame overhead for real-time data transfer is composed of 30Byte Ethernet overhead, 28Byte UDP/IP, 24Byte encapsulation header [43]. The V-LAN Tag is an optional field, therefore it is marked in the frame structure below.

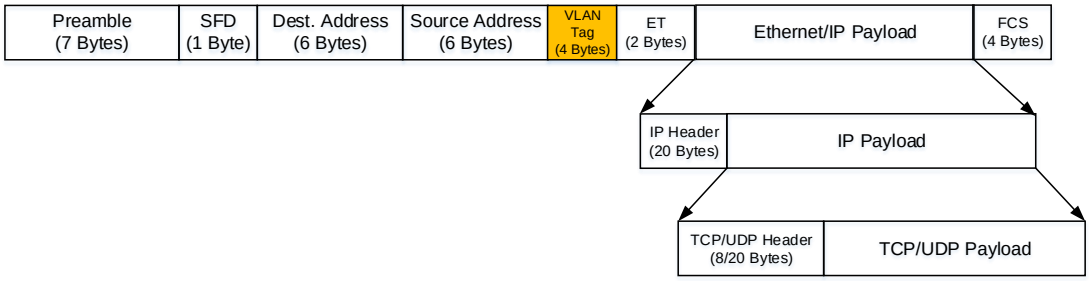


Figure 13: Ethernet/IP frame structure.

The computation of an Ethernet/IP frame size is illustrated below

$$B_{EthernetIP} = B_{TotalOverhead} + B_{Payload}$$

$$B_{TotalOverhead} = B_{EthernetOverhead} + B_{UDP/IP} + B_{Encapsulation}$$

To transmit a payload of 8Byte to an actuator, a 90Byte Ethernet/IP frame is required

$$B_{EthenetIP} = 82 + 8 = 90\ Byte$$

Since Ethernet/IP is built on on Commercial Off-the-Shelf (COTS), Store&Forward switch-based architectures are possible [25]". Typically a **star topology** is used preventing data collisions using full-duplex approach and **strict-priority** transmission selection algorithm.

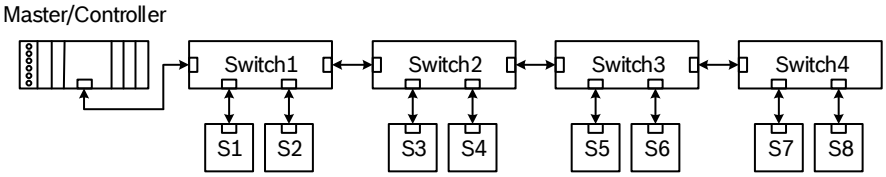


Figure 14: Network Topology

Minimum cycle time model 1 as defined in [44] is typically used to achieve soft real-time performance with cycle times around **10 milliseconds**. Other cycle time models have been introduced in [34].

Using the full-duplex feature the uplink- and downlink-stream transmissions start simultaneously. The minimum cycle time is calculated as the maximum of downlink- and uplink-stream communication durations as shown in Figure 4 (left).

The Ethernet/IP CIP Sync mechanism synchronizes the network through distributed clocks as specified in the **IEEE1588** standard. Combined with CIP Motion, motion applications such as servo motor control are feasible.

2.1.3.2 Profinet RT

Profinet RT is an industrial-Ethernet protocol from the **second real-time class** that was released by Profibus & Profinet International (PI), originally named Profinet Soft Realtime (SRT), a term that was later on dropped for marketing reasons [45]. Meanwhile mainly addressed as PROFINET I/O.

Profinet RT runs on standard Ethernet hardware and bypasses TCP- and UDP/IP stacks to transfer real-time process data that are sent to each device **individually** (Figure). "Whereas diagnostics and configuration data are sent using UDP/IP" [46].

The 36Byte overall frame overhead for real-time data transfer is composed of 30Byte Ethernet overhead and 6Byte protocol header. The VLAN-Tag is optional, therefore it is marked in the frame structure below.

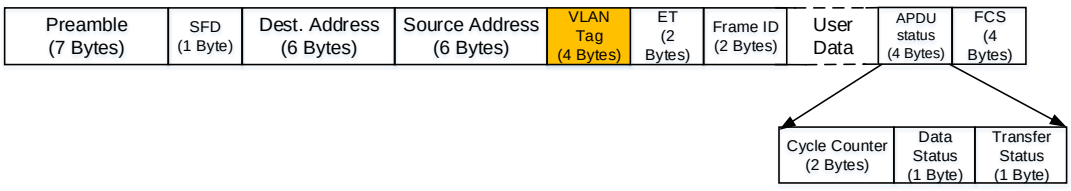


Figure 15: Profinet RT frame structure.

The computation of Profinet RT frame size is illustrated below

$$B_{P/RT} = B_{TotalOverhead} + B_{Payload}$$
$$B_{TotalOverhead} = B_{EthernetOverhead} + B_{ProtcolHeader}$$

To transmit a payload of 8Byte to an actuator, a 45Byte Profinet RT frame is required

$$B_{P/RT} = 36 + 8 = 45\ Byte$$

Profinet RT builds **Store&Forward** switch-based architectures. **Star-** and **tree-topologies** can be built using **strict-priority**-based managed switches.

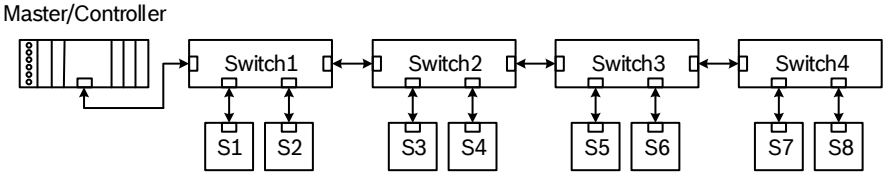


Figure 16: Typical Network Structure of Profinet RT

Minimum cycle time model 1 as defined in [44] is typically used to achieve soft real-time performance with cycle times of ~**5 to 10 milliseconds**.
Using the full-duplex feature the uplink- and downlink-stream transmissions can start simultaneously. The minimum cycle time is calculated as the maximum of downlink- and uplink-stream communication durations as shown in Figure 4 (left).

2.1.3.3 Profinet IRT

For higher timing performances Profibus & Profinet International (PI) released the Profinet IRT industrial-Ethernet protocol that belongs to the **third real-time class** [45]. Like Profinet RT, Profinet IRT is also addressed as PROFINET I/O. Both are an extension of the Profinet CBA. Profinet IRT requires special hardware support to achieve sufficient performance and high synchronization accuracy for industrial applications with hard-real-time requirements.
Time-critical data is sent in layer 2 frames by **planning and scheduling predefined time-slots**. [46].
The 32Byte overall frame overhead for real-time data transfer is composed of 26Byte Ethernet overhead and 6Byte protocol header [47]. Profinet IRT frames are not VLAN tagged, hence the Ethernet overhead is reduced by 4 Bytes.

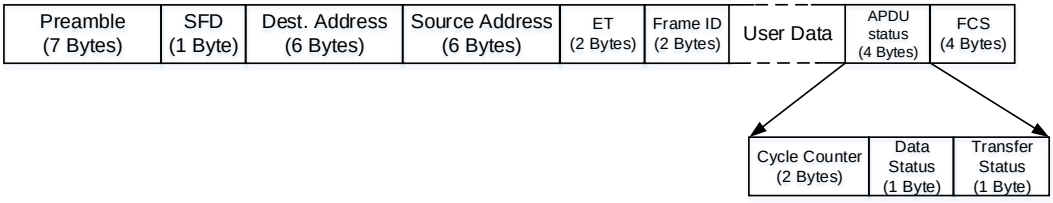


Figure 17: Profinet IRT frame structure.

The computation of Profinet IRT frame size is illustrated below

$$B_{P/IRT} = B_{TotalOverhead} + B_{Payload}$$
$$B_{TotalOverhead} = B_{EthernetOverhead} + B_{ProtcolHeader}$$
$$B_{ProtcolHeader} = B_{FrameID} + B_{APDUstatus}$$

To transmit 8Byte Payload to a slave device a Profinet IRT frame of the size 40Byte is required.

$$B_{P/IRT} = 32Byte + 8Byte = 40\ Byte$$

Unlike Ethernet/IP and Profinet RT, Profinet IRT does not rely on COTS hardware and presents special **cut-through** switch-based architectures. **Line, Ring, Star-** and **tree-topologies** can be built using **time-aware** configurable switches.

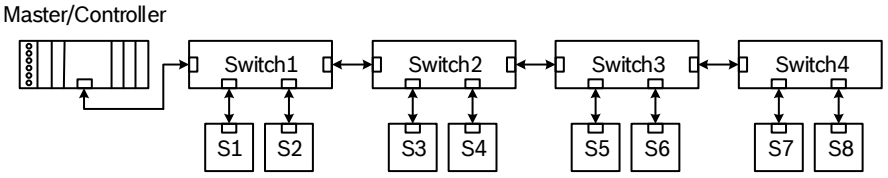


Figure 18: Typical Ntework Structure of EtherNet/IP

Minimum cycle time model 1 as defined in [25] is used (Figure 3).
[46] PROFINET IRT offers clock-synchronized cycle times below **one millisecond**, which satisfies the hard-real-time requirements of Motion Control applications. PROFINET IRT implements therefore a time-division multiple multiplex (TDMA) mode based on specially managed, hardware-synchronized switches and supports the "**Dynamic Frame Packing** (DFP) approach, defined in [24], to optimize cycle times making use of the **summation frame** principle for a certain set of devices in the network." However individual frames are also supported to transmit time-critical data.

2.1.3.4 Sercos III

Sercos III is a typical representative of the **third real-time class**. It has been released by the Sercos International e.V. (SI) in 2003 and is predominantly used in Motion-based automation systems, where high timing performances are required.
Similar to Profinet IRT, Sercos III uses also solution that require specific hardware and software support.
Time-critical data is sent in layer 2 frames that are protected from interfering with the remaining traffics by **planning and scheduling predefined time-slots**. Sercos III is master-slave based and builds a converged network, in which scheduled-time-critical traffics (e.g. control data and alarm signals) or non-scheduled remaining traffics (reason of an alarm signal, diagnostics, etc…) share the same network resources.
The Sercos III communication cycle time follows the model 2 described in (2.1.2.2) (figure 4 (right)) and is divided into two channels: Real-Time Channel (RTC), in which only scheduled Sercos frames are transmitted, and Unified Communication Channel (UCC), in which all remaining traffics can be transmitted.
The communication cycle times are defined as 31,25µs, 62,5µs, 125µs up to 65ms [48].

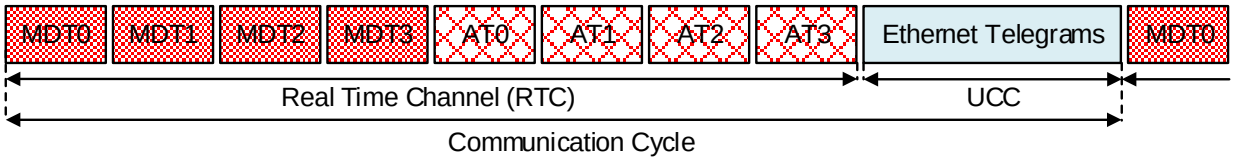


Figure 19: Sercos III communication cycle

The master device generates and initiates the real-time data exchange through the Sercos **summation frames**: Master Data Telegrams (MDT[1]) and Acknowledge Telegrams (AT[2]), which are cyclically transmitted to the slave devices (drive- and I/O-modules). The slaves reads the MDT frames and adjust the ATs with their current data on the fly. All payloads of real-time data are encapsulated in one frame. Once the frame reaches the maximum possible Ethernet frame size, next frame is used. Up to 4 MDTs and 4 ATs are possible. The frames have similar structure but different overheads: 56Byte for MDT0, 52Byte for AT0 and 48Byte for MDT1,2,3 and AT1,2,3.

[1] MDTs transmit new commands to the actuators, e.g. position. MDTs are read-telegrams
[2] AT transmits last status from the sensors, e.g. temperature. ATs are write-telegrams

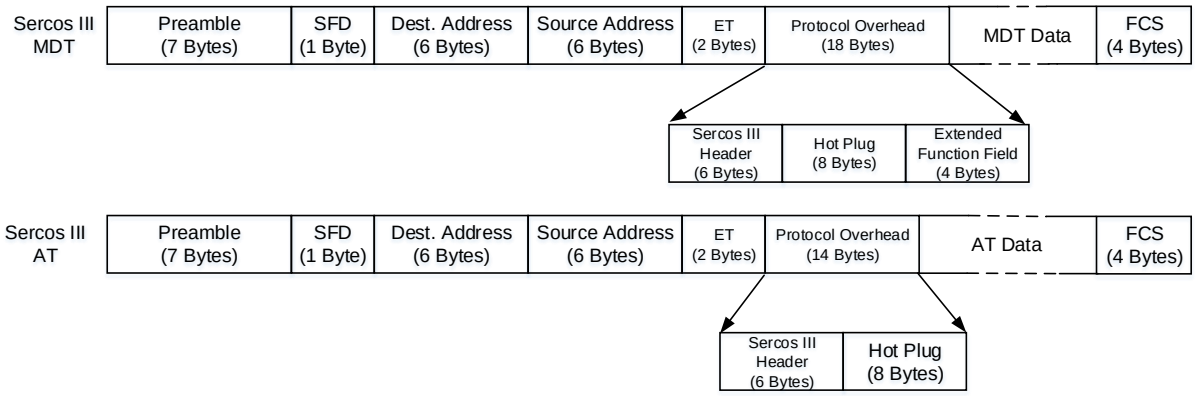


Figure 20: MDT and AT Sercos III frame structures

The computation of Sercos III frame size is illustrated below

$$B_{Sercos} = B_{TotalOverhead} + (B_{Payload} * n_{slave-devices})$$
$$B_{TotalOverhead} = B_{EthernetOverhead} + B_{ProtcolHeader}$$

To transmit 8Byte Payload to 20 slave devices MDT0 frame of the size 216Byte is required.

$$B_{MDT0} = 56Byte + 8Byte * 20devices = 216Byte$$

Similar to Profinet IRT, Sercos supports **cut-through** forwarding mechanism through a two ports switched slave end-devices but is restricted to **line, Ring and broken Ring (two lines)**. No external switches are used (Figure 21). Cut-through forwarding delay below one microsecond (~800 nanoseconds) is possible.

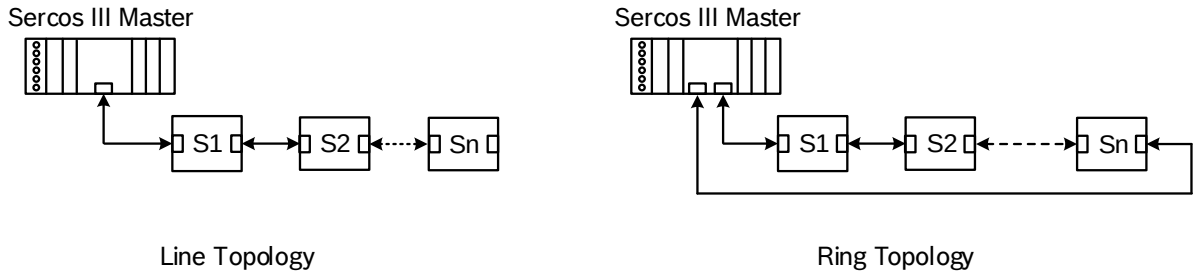


Figure 21: Line and Ring Topologies of Sercos III

2.1.4 Summary

The table below summarizes the features of multiple selected Industrial-Ethernet Protocols from real-time class 1, 2 and 3.

Real-time Class	IE Protocol	Features										
		Cycle Time Guarantees	Frame Structure	Topology				Nodes	Forward Mechanism	Forwarding Delay	
				Line	Ring	Star	Tree			100Mbps	1Gbps
1	Profinet CBA	10 -100 ms	Individual	✓	✓	✓	✓	C-Sw-D[4]	S&F[3]	-	-
	Ethernet/IP[6]			✓	✓[5]	✓	✓	C-Sw-D	S&F	-	-
	Modbus/TCP			✓	✓	✓	✓	C-Sw-D	-	1	1
2	Profinet RT	1 -10 ms		✓	✓	✓	✓	C-Sw-D	S&F	-	-
	Powerlink			✓	✓	✓	✓	C-Hub-D	-	0.5	-
3	Profinet IRT	< 1 ms	Summation	✓	✓	✓	✓	C-Sw-D	Cut-Through	3	0.6
	SercosIII			✓	✓	×	×	C-D	Cut Through	0.8	×
	EtherCAT			✓	✓	✓	✓	C-D	Cut Through	1.35	0.85

[3] S&F – Store@Forward
[4] C-Sw-D – Controller (Master) to Switch to Drive (Slave)
[5] DLR-topology: Device Level Ring
[6] Cycle times below 10ms are reachable with Ethernet/IP by modifying the stack and the driver

Table 9 Features of certain Industrial-Ethernet Protocols from real-time classes

2.2 Deterministic TSN-Standards

TSN specifies multiple IEEE standards and enhancements for reliability, security, configuration, synchronization, timing performances, etc... The focus of this thesis is improving the timing performances of the industrial communication by integrating TSN in the field- and control-levels. Therefore only the deterministic TSN-features IEEE802.1AS-Rev, IEEE802.1Qbu, IEEE802.3br and IEEE802.1Qbv are further recognized. These functionalities are described in this section.

2.2.1 IEEE802.1AS-Rev

[49] "This standard specifies the protocol and procedures used to ensure that the synchronization requirements are met for time-sensitive applications, such as audio, video, and time-sensitive control, across network. This includes the maintenance of synchronized time during normal operation and following addition, removal, or failure of network components and network reconfiguration. It specifies the use of the IEEE Standard 1588 specifications where applicable in the context of IEEE Standard 802.1Q. (…)
This standard enables stations attached to bridged LANs to meet the respective jitter, wander, and time synchronization requirements for time-sensitive applications. This includes applications that involve multiple streams delivered to multiple endpoints. To facilitate the widespread use of bridged LANs for these applications, synchronization information is one of the components needed at each network element where time-sensitive application data are mapped or demapped or a time-sensitive function is performed. This standard leverages the work of the IEEE 1588 Working Group by developing the additional specifications needed to address these requirements."
[16] "The Precision Time Protocol (PTP), as described in IEEE1588v2 uses physical layer timestamps to compute network delays and define synchronization events.
For TSN systems a 1588 profile was developed (IEEE 802.1AS) that defines less options, but extends some physical layer options. This profile was developed for Audio-Video Bridging (AVB) to provide the following PTP features:

- Provides performance specifications for switches as "Time-aware Bridges";
- Uses accumulated "Neighbor Rate Ratio" calculations to improve accuracy and speed up convergence
- Includes Plug and Play operation and startup with a specified Best Master Clock Algorithm (BMCA) used by switches.
- Requires Two-Step Delay message processing (Sync & Follow-Up Messages); "

PTP Systems
[50] "A PTP system is a distributed, networked system consisting of a combination of PTP and non-PTP devices. PTP devices include ordinary clocks, boundary clocks, end-to-end transparent clocks, peer-to-peer transparent clocks, and management nodes. Non-PTP devices include bridges, routers, and other infrastructure devices, and possibly devices such as computers, printers, and other application devices."
PTP device types
Generalized PTP (gPTP) IEEE802.1AS defined the **clock** as a „physical device that is capable of providing a measurement of the passage of time since a defined epoch" [51].
[52] There are five basic types of PTP devices, each implements one or more aspects of the protocol.

Clock Type	Description
Ordinary clock	"Supports a single copy of the protocol and has a single PTP state. The ordinary clock can be the grandmaster clock in a system, or it can be a slave clock in the master–slave hierarchy." [53]
Boundary clock	"typically has several physical ports, each communicating with the network via two logical interfaces. (…) typically used only as a network element and is not normally associated with application devices such as sensors or actuators" [53]
End-to-end transparent clock	"forwards all messages just as a normal bridge, router, or repeater." [53] Non PTP switches can be used
Peer-to-peer transparent clock	Only difference to the end-to-end transparent clock is correcting and handling the PTP timing messages. [53]
Management node	"is a PTP device that: a) Has one or more physical connections to the network b) Serves as an human or programmatic interface to PTP management messages c) May be combined with any of the clock types" [53]

Table 10: Description of the PTP clock types as defined in [53]

PTP port roles

[55] defines 4 port roles that are illustrated in the table below.

Port role	Description
Grand Master	Can be given only to one port in the whole PTP network with the highest clock accuracy
Master Port	Any port, P, of the time-aware system that is closer to the root than any other port of the gPTP communication path connected to P [55]
Slave Port	The one port of the time-aware system that is closest to the root time-aware system. If the root is grandmaster-capable, the SlavePort is also closest to the grandmaster. The timeaware system does not transmit Sync or Announce messages on the SlavePort. [55]
Passive Port	Any port of the time-aware system whose port role is not MasterPort, SlavePort, or DisabledPort. [55]
Disabled Port	Any port of the time-aware system for which portEnabled, pttPortEnabled, and asCapable are not all TRUE. [55]

Table 11: Port role definitions

gPTP message classes

[56] IEEE802.1AS defines two message classes, "the event message class and the general message class. **Event messages** are time-stamped on egress from a time-aware system and ingress to a time-aware system. **General messages** are not time-stamped."

[57] "The set of event messages consists of:

- Sync (see 13.6)
- Delay_Req (see 13.6)
- Pdelay_Req (see 13.9)
- Pdelay_Resp (see 13.10)

The set of general messages consists of:

- Announce (see 13.5)
- Follow_Up (see 13.7)
- Delay_Resp (see 13.8)
- Pdelay_Resp_Follow_Up (see 13.11)
- Management (see Clause 15)

- Signaling (see 13.12)“

[57] “The Sync, Delay_Req, Follow_Up, and Delay_Resp messages are used to generate and communicate the timing information needed to synchronize ordinary and boundary clocks using the delay request-response mechanism.
The Pdelay_Req, Pdelay_Resp, and Pdelay_Resp_Follow_Up messages are used to measure the link delay between two clock ports implementing the peer delay mechanism. The link delay is used to correct timing information in Sync and Follow_Up messages in systems composed of peer-to-peer transparent clocks.
Ordinary and boundary clocks that implement the peer delay mechanism can synchronize using the measured link delays and the information in the Sync and Follow_Up messages. The Announce message is used to establish the synchronization hierarchy.
The management messages are used to query and update the PTP data sets maintained by clocks. These messages are also used to customize a PTP system and for initialization and fault management. Management messages are used between management nodes and clocks.
The signaling messages are used for communication between clocks for all other purposes. For example, signaling messages can be used for negotiation of the rate of unicast messages between a master and its slaves.
All messages can be extended by means of a standard type, length, value (TLV) extension mechanism. For example, the PATH_TRACE message extensions can be used to detect rogue frames; see 16.2.1 for more detail on rogue frames.”

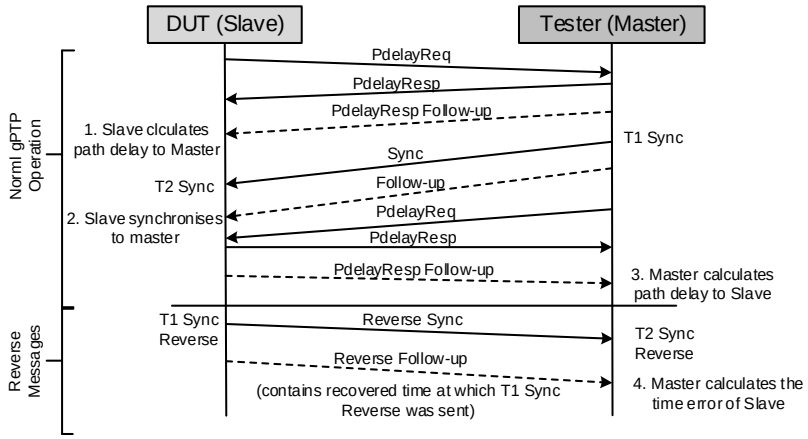


Figure 22: Sync Message Exchange

gPTP messages transmission intervals

[58] “For each of the message types Announce, Sync, Delay_Req, and Pdelay_Req, the mean time interval between successive messages shall be represented as the logarithm to the base 2 of this time interval measured in seconds on the local clock of the device transmiting the message. The values of these logarithmic attributes shall be selected from integers in the range −128 to 127 subject to further limits established in an applicable PTP profile. These intervals are communicated via the logMessageInterval field of PTP messages.”

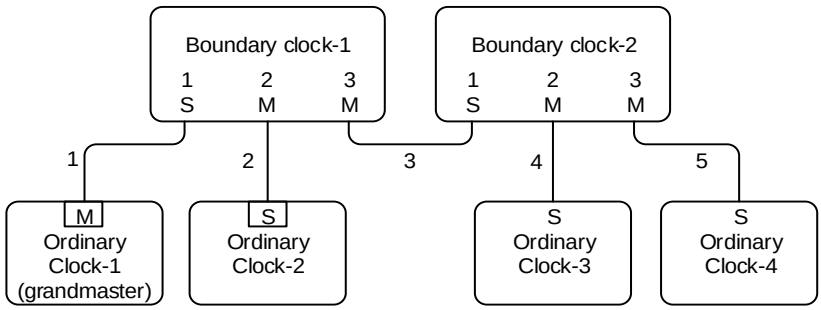


Figure 23: Simple master−slave hierarchy [59]

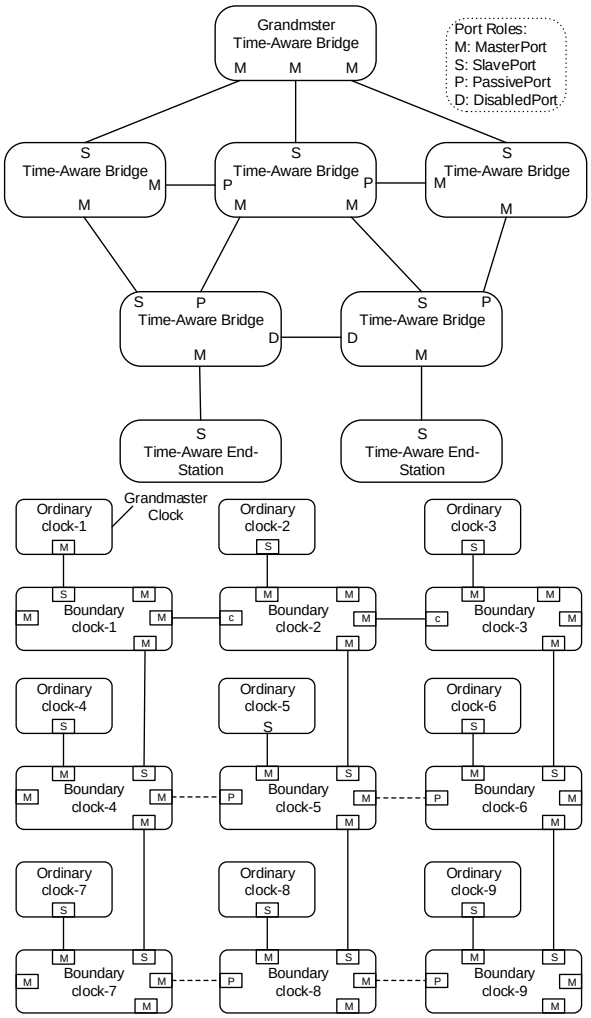


Figure 24: Master/Slave hierarchy example of time-aware systems [60]

Best Master Clock Algorithm

[61] "The best master clock algorithm compares data describing two clocks to determine which data describes the better clock. This algorithm is used to determine which of the clocks described in several Announce messages received by a local clock port is the best clock. It is also used to determine whether a newly discovered clock—a foreign master—is better than the local clock itself. The data describing a foreign master is contained in the grandmaster fields of an Announce message. The data describing the local clock is contained in the defaultDS data set of the clock."

2.2.2 IEEE802.1Qbv

IEEE802.1Qbv is an enhancement to the IEEE802.1Q standard and is titled as "*Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Enhancements for Scheduled Traffic*" [62].
IEEE802.1Qbv specifies a time-aware queuing procedure that enables bridges and end stations to schedule the transmission of scheduled time-critical traffics. Therefore a high time synchronization accuracy is required. This can be derived from the IEEE802.1AS-Rev standard. The amendment introduces a new traffic shaper (Time-Aware Shaper) that provides higher timing performances than the strict-priority algorithm defined in IEEE802.1Q and the credit-based shaper defined in IEEE802.1Qav. An upper bound worst case latency and ultra-low jitter can be guaranteed by avoiding the interference with the background non-scheduled traffics.

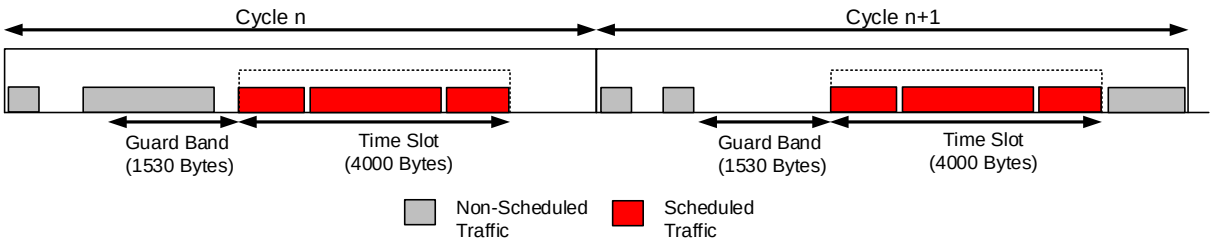


Table 12: A reserved Time Slot protected by a guard band window as defined in IEE802.1Qbv

Functionality

Using the scheduling approach defined in the time-aware shaper, time-critical traffics can be scheduled. That means transmission time points and time-slots can be planned, therefor the egress ports need to be configured. IEEE802.1Qbv associates a *transmission gate* [63] to each queue, which based on the state of the transmission gate, can be selected for transmission or not. The states are:

- Open: queued frames can be selected for transmission.
- Closed: queued frames are not selected for transmission.

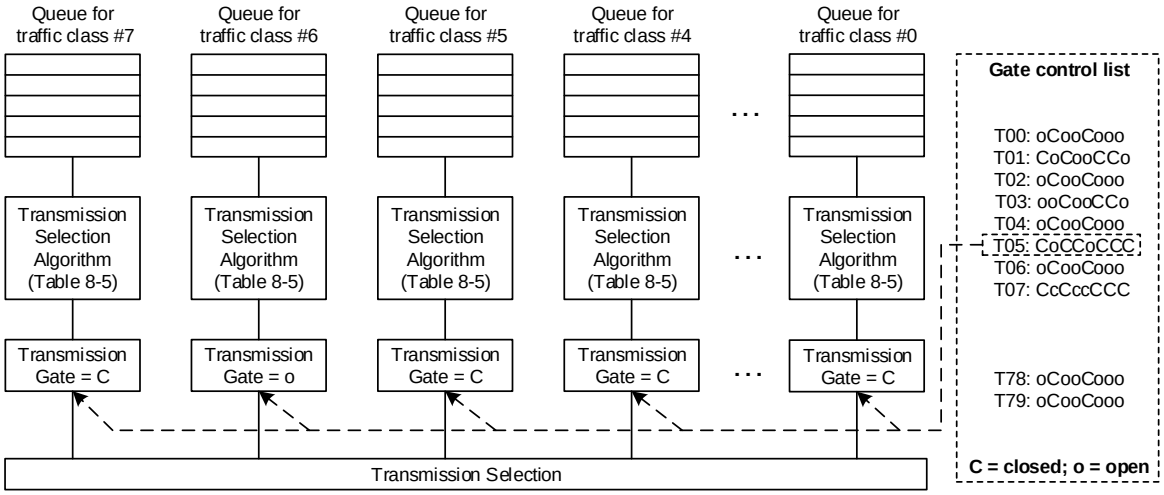


Figure 25: Time-Aware Shaper with transmission gates

A transmission queue of a time-aware egress port can be configured to support multiple transmission windows with different cycle times (Figure 25). The configuration is summarized in a *gate control list* that contains an ordered list of gate operations.

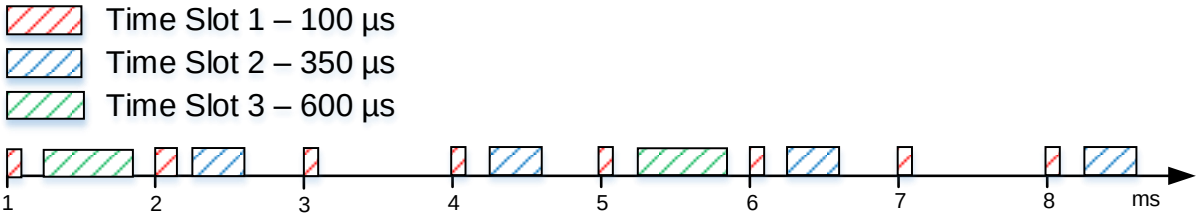


Figure 26: Multiple time-slots for the same transmission queue of an egress port

Unlike the strict-priority algorithm and the credit-based shaper, time-critical scheduled frames must not have the highest priority PCP. E.g. the transmission queue 3 with the frames of priority PCP3 can be configured to transmit scheduled traffics. In this case, frames of PCP7 that are ready for transmission during the time-slot of PCP3 are blocked and need to be transmitted outside of the time-slot.

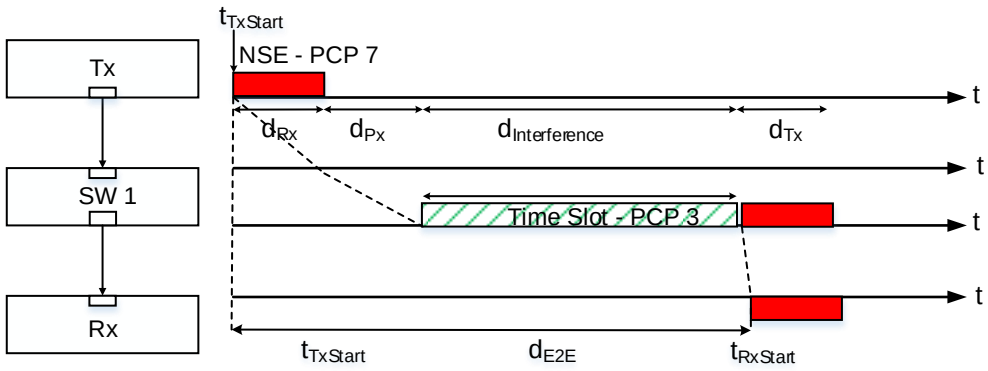


Figure 27: PCP7 frame ready within the TSlot of PCP3 frame and is therefore delayed and forwarded outside of the time-slot

2.2.3 IEEE802.1Qbu - IEEE802.3br

Background

Since IEEE Ethernet used to be non-preemptive transmission before the specification of IEEE802.1Qbu and IEEE802.3br, an interfering non-time-critical frame is guaranteed to finish without interruption. This increases the delay of a time-critical frame with higher priority, once it is ready for transmission during the transmission of the lower priority frame. The interference delay for a single forwarding hop goes up to ~123,36 at 100Mbps or ~12,34µs at 1Gbps and is equal to the transmission duration of the interfering frame (64 to 1530Byte) and the IFG duration (≥12Byte). The interference can occur at each forwarding process, which leads to big delay variation (jitter) and high end-to-end delay. Time-critical applications, especially with line or ring topologies, cannot tolerate such not deterministic timing behavior.
As a solution for this problem, the frame-preemption functionality has been specified in IEEE802.1Qbu and IEEE802.1br. With the preemption of the transmission of an interfering non-time-critical frame the interference delay can be significantly reduced to 11,44µs at 100Mbps or 1,14µs at 1Gbps.

IEEE802.1Qbu vs. IEEE802.3br

[29] "Frame-preemption in Ethernet is specified in the IEEE 802.3br (interspersing express traffic) standard (…), which defines two MAC interfaces":

- Express MAC interface to transmit frames of the express traffic class. These frames cannot be preempted.
- Preemptable MAC interface to transmit frame the preemptable frames. These can be preempted only by express frames. Preemptable frames cannot be preempted by other preemptable frames even if they have higher priority.

IEEE 802.1Qbu (frame-preemption) is defined in the context of TSN and adds management and configuration mechanisms for frame-preemption.

Functionality

Figure 28 illustrates the difference between non-preemptive- and preemptive-frame transmissions. Two frames high priority *hp* and low priority *lp* are concurring on the egress port of an Ethernet node (switch or end-device). During the transmission of *lp*, *hp* becomes ready for transmission. Without preemption, *hp* needs to wait until *lp* finishes its transmission. Under preemption the transmission of *lp* can be preempted and *hp* can be sent earlier. However certain preemption overhead [29] due to the inter frame gap duration is introduced. *hp* can be transmitted only after the transmission end of the first fragment of *lp* and the IFG. After the transmission of *hp* and IFG duration *lp* frame can be resumed by transmitting its second fragment with a preamble and "other information, which is required to reassemble the preempted frame at the receiving end" [29].

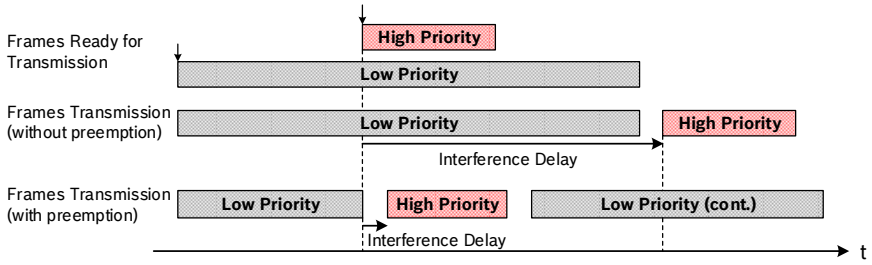


Figure 28: non-preemptive vs. preemptive frame transmission

Frame Format

In order to make frame-preemption transparent to the physical layer (PHY) of Ethernet, the original Ethernet frame format as defined in IEEE802.3 need to be preserved for the resulting new fragments. Therefor IEEE802.3br defines new formats to the express frame and to the preemptable frame and its fragments.

The format of express and preemptable frames is very similar to the original format. Only the start of frame delimiter (SFD) is replaced by the SMD-E (**S**tart **M**AC merge frame **D**elimiter - **E**xpress) for an express frame and by SMD-Sx (**S**tart **M**AC merge frame **D**elimiter – **S**tart fragment) for preemptable frame [29]. Further the Frame Check Sequence FCS is replaced for preemptable frames by an MCRC (Mac Merge Cyclic Redundancy Check).

All remaining fragments are signaled by an SMD-Cx (SMD – continuation fragment) and a fragment counter (FCnt). For the last fragment the MCRC is again replaced by the FCS.

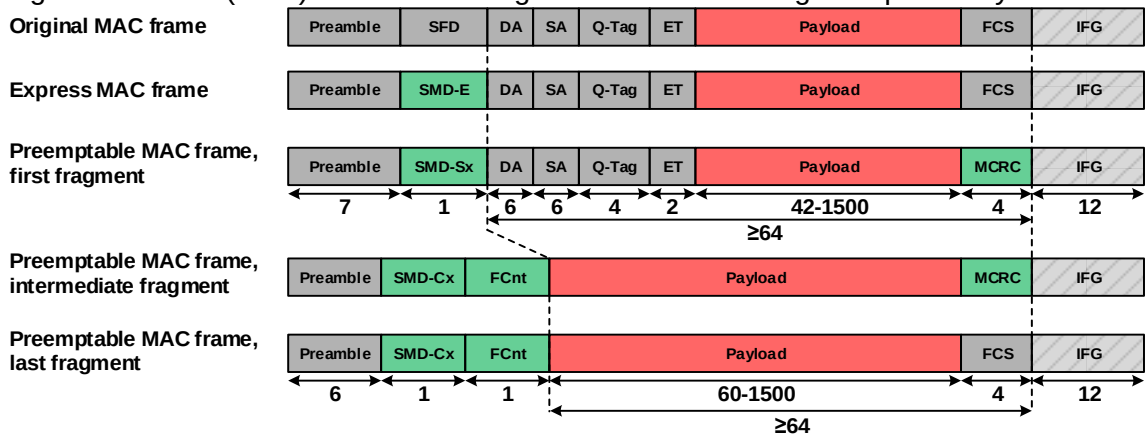


Figure 29: MAC frame formats: original IEEE 802.3 MAC frame format on the top followed by four IEEE 802.3br frame formats (all sizes in bytes) [29]

Worst-Case Computation

In order to analyze the timing behavior of frame-preemption [29] computed the longest preemptable frame that cannot be preempted and would block an express frame. This is set to 143 bytes. The resulting fragments after preempting a preemptable frame must fulfill the minimum Ethernet frame size requirement of 84 Bytes (including the preamble, SFD and IFG).

The totoal payload PL is splitted into PL1 of fragment 1 and PL2 of fragment 2.

PL1 = Minimum Frame Size – Fragment1Overhead = 84Byte – 42Byte = 42Byte

PL2 = Minimum Frame Size – Fragment2Overhead = 84Byte – 42Byte = 60Byte

PL = PL1 + PL2 = 102Byte

Minimum Preemptable Frame Size = Ethernet Overhead + Payload (PL) = 144Byte

Thus the size of the minimum preemptable frame that cannot be preempted is 143Byte.

Each single preemption presents an overhead of 24Bytes due to the data fields Preamble, SMD-Cx, FCnt, MCRC, and IFG that need to be sent with the new fragment.

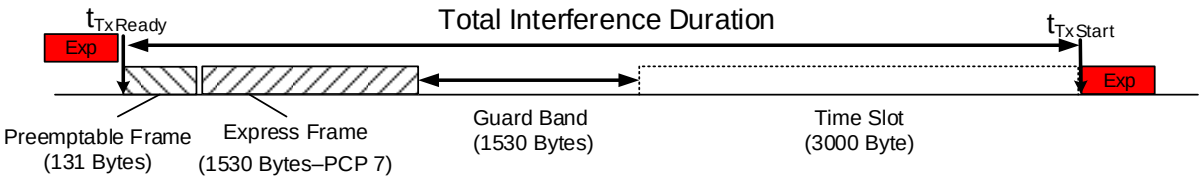


Figure 30: Figure – Upper Worst-Case Interference delay of an express frame over a single hop – Interference with preemptable (<131Byte), express (1530Byte), guard band (1542Byte), time-slot (3000Byte)

Summary

The motivation behind these standards is improving the end-to-end delay of time-critical traffics by reducing the delay caused by the interference with the background low-priority traffics. The two standards address this problem by introducing new enhancements to the IEEE Standard 802.1Q-2014 and defining a new forwarding process that support the preemption of the transmission of time-critical traffics. The implementation of the frame-preemption is mainly in the MAC layer and requires hardware support.

2.3 Formal Timing Analysis

2.3.1 Traffic classification

Basically two traffics types are considered in the thesis: time-critical and non-time-critical. Following the activated transmission selection algorithm, the data traffics are further classified into multiple traffic classes (Table 13).

Transmission Selection Alg.	Time-Critical Traffic	Non-Time-Critical Traffic
Non-Preemptive Strict-Priority Algorithm (IEEE802.1Q)	High Priority	Low Priority
Preemptive Strict-Priority Algorithm (IEEE802.1Qbu)	Express	Preemptable
Non-Preemptive Time-Aware Shaper (IEEE802.1Qbv)	Scheduled[7]	Non-Scheduled Low Priority
	Non-Scheduled High Priority	
Preemptive Time-Aware Shaper (IEEE802.1Qbv - IEEE802.1Qbu – IEEE802.3br)	Scheduled Express	Non-Scheduled Preemptable
	Non-Scheduled Express	

Table 13: Classification of the data traffics [11]

Time-critical traffics in the industrial automation are typically divided into

- *Isochronous control-data* – is synchronously exchanged at a predefined periodic rate
- *cyclic- control-data* – is exchanged at defined rate
- *Alarms and events* – are sporadically transmitted. An upper-bound end-to-end delay need to be guaranteed and traffic loss need to be avoided

2.3.2 Traffic Modeling

Time-critical traffics are generally described by the following parameters:

$$M_{i,j} = \{X_i, Y_i, Z_i\}$$

where i s the traffic type and j the traffic class, X_i, Y_i, Z_i are specific parameters.

For *Non-Preemptive Strict-Priority Algorithm* (IEEE802.1Q) the time-critical (*tc)* high-priority (*hp)* traffics are given as follow:

$$M_{tc,hp} = \{T_i, P_i, C_i\}$$

where T_i is the transfer period, P_i its priority and C_i the message size.

[7] Scheduled Traffic (without preemption) is independent of the priority

For *Preemptive Strict-Priority Algorithm* (IEEE802.1Qbu) the time-critical *tc* express *exp* traffics are given as follow:

$$M_{tc,e} = \{T_i, P_i, C_i\}$$

where T_i is the transfer period, P_i its priority and C_i the message size.

For *Preemptive and Non-Preemptive Time-Aware Shaper* (IEEE802.1Qbv with / without IEEE802.1Qbu) the time-critical *tc* scheduled *s* traffics are given as follow:

$$M_{tc,s} = \{T_i, P_i, S_i, C_i\}$$

where T_i is the transfer period, P_i its priority, S_i its reserved time-slot and C_i the message size.

2.3.3 Evaluation Metrics

[11] "There are multiple evaluation metrics of a deterministic communication system. [5] and [64] defined methods to deliver the metrics and classified them into empiric and model-based. The model-based methodology is divided into formal (analytical) and simulative. Metrics indicates the performance of a network and are required to evaluate certain communication features [65]. To evaluate the different transmission selection algorithms of Ethernet, the thesis focusses on three metrics:

- *End-To-End delay*
- End-To-End delay variation: *Jitter*
- and *throughput* or *bandwidth utilization* "

2.3.4 Flow Timing Model

[11] "The end-to-end (E2E) delay of a time-critical frame is given as Last Out – Last In (LOLI) measurement methodology. Figure 31 shows the E2E delay of a frame sent from the transmitter end-device over a switch to the receiver end-device."

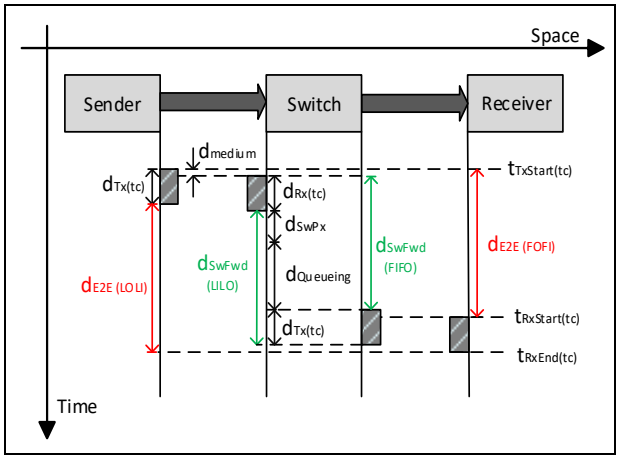


Figure 31 - Flow timing model for a Store&Forward Switch showing the queuing delay in case of interference between two frames from different data traffics

2.3.5 Formal Computation

End-To-End delay

[11] "In order to guarantee a deterministic communication, the upper-bound worst-case End-To-End delay d_{E2E} of a time-critical frame *tc* allover its path between its transmitter and its receiver needs to be computed. The worst case scenario depends on the queuing delay of *tc*, which is determined by the transmission selection algorithm.

$$d_{E2E(LOLI)} = d_{Tx(tc)} + d_{medium} + (d_{SwFwd} + d_{medium}) * n_{Sw} \quad (1)$$

$d_{Tx(tc)}$ is the transmission duration of *tc* and depends on the *tc* size and the link-speed. The switch forwarding delay d_{SwFwd} depends on the forwarding mechanism and the

transmission selection algorithm. d_{SwFwd} counts for a single switch and is multiplied by the number of traversed switches n_{Sw}. For an accurate computation the medium delay d_{medium} of each link is considered. d_{medium} is a constant parameter and is set to 5 nanoseconds per meter twisted pair copper-based segment)."

Upper-Bound Worst-Case Switch Forwarding Delay

[11] "The definition of the LILO (Last In – Last Out) measurement methodology of the switch forwarding delay is: once received at the switch, *tc* faces the switch processing delay d_{SwPx}, defined in [2] as the delay caused by the switch internal processing unit and the search algorithm (address-lookup) in order to "enqueue" *tc* in the appropriate transmission queue. d_{SwPx} is independent of the link-speed, the frame size and the transmission selection algorithm. Once enqueued and ready for transmission, *tc* could face a queuing delay $d_{Queuing}$ before getting selected for transmission. Unlike Cut-Through (CT), *tc* needs to be retransmitted by each forwarding hop for the Store&Forward (S&F) mechanism. "

$$d_{SwFwd(S\&F-LILO)} = d_{SwPx} + d_{Queuing} + d_{Tx(tc)} \quad (2)$$

Upper-Bound Worst-Case Queuing Delay

[11] "The Queueing Delay is the duration of time, that *tc* spends in its transmission queue from the time point of being ready for transmission until its transmission starts. This is caused either by an interference with a concurring frame *cf* on the same egress port (3) or by a poor scheduling design causing *tc* to wait until its scheduled time-slot starts (4), which can be avoided by starting the time slot early. In this work we follow only the interference queuing delay (3)."

$$d_{Queuing} = d_{Interference} = d_{Tx(cf)} + d_{IFG} \quad (3) \qquad d_{Queuing} = d_{Waiting\_Offset} \quad (4)$$

Worst-Case Interference Scenario: *the forwarding port starts transmitting a concurring frame cf (1530 byte MTU) from another transmission queue just before tc is ready for transmission. cf can be a time-critical or a non-time-critical frame. tc needs to wait until the transmission of cf (1530Byte) and the the IFG duration (≥12Byte) have been elapsed.*

The upper-bound worst-case interference delay $d_{Interference}$ is given by the transmission duration of the concurring frame and depends on the link-speed and is the difference $\Delta(t_{TxReady(tc)}, t_{TxStart(tc)})$ between the time points: when *tc* is ready for transmission $t_{TxReady(tc)}$ and when the transmission actually starts $t_{TxStart(tc)}$.

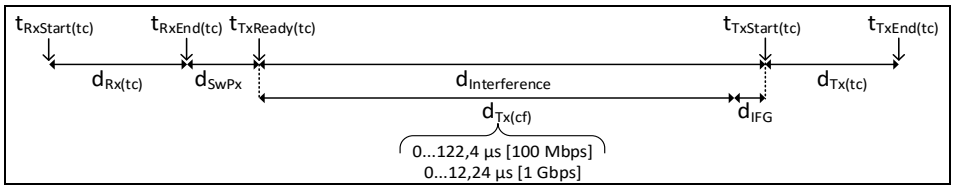


Figure 32 - Delay Parameters [11]

End-To-End Delay Variation (Jitter)

We define the Jitter of a time-critical frame *tc* as the average deviation of its End-To-End delay over the sample time.

$$Jitter = Average\ Deviation\ (d_{E2E}) = \frac{\sum_{i=1}^{n} |d_{E2E(i)} - \overline{d_{E2E}}|}{n} \quad (5)$$

where *n* is the sample set.

Channel Throughput

The throughput is the rate of successful message delivery over a communication channel in bits per second [17] and is limited by the channel capacity given by the link-speed or link-speed.

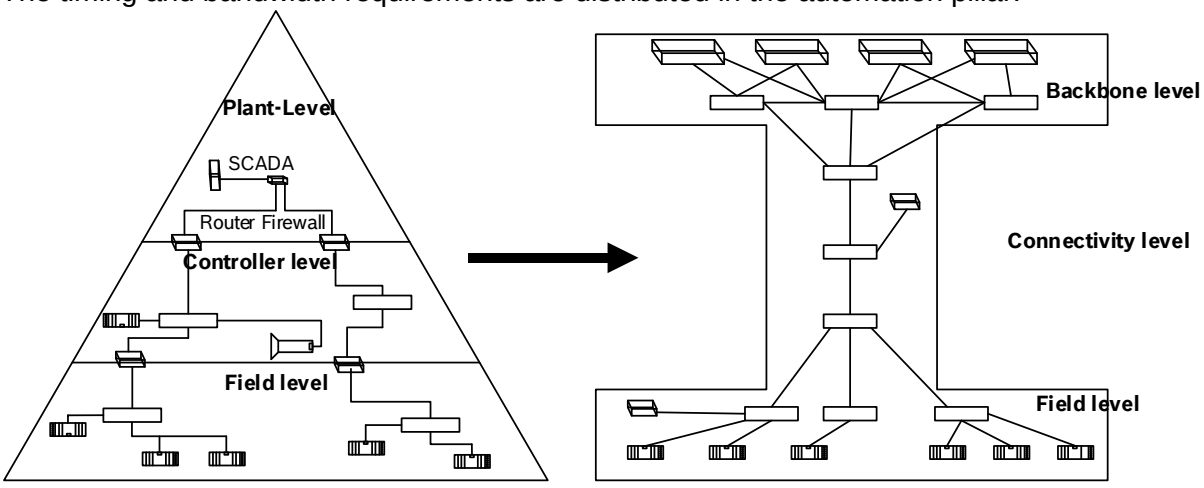
$$Throughput_{Channel} = \frac{Frame_{size_{bit}} * n_{frames/Sec}}{d_{oneSecond}} \quad (6)^8 \qquad n_{frames} = \left\lfloor \frac{d_{oneSecond}}{((d_{TxFrame}+d_{IFG})*d_{BitRate})} \right\rfloor \quad (7)^9$$

2.4 Surveys

2.4.1 From Automation Pyramid to Automation Pillar

[66] A new vision that is getting spread with the integration of TSN in the industry is the architecture transition from the "classic" *automation pyramid* with its static architecture and localized functionalities to the *automation pillar* with a dynamic architecture and virtualized functionalities.

The automation pillar deletes the controller level and moves its functionalities either down to the field level or up to the backbone level. The controller level is transiting to a "connectivity level" with the purpose of connecting the applications of the backbone- and field-levels. This enables a "high-speed" data exchange between the two levels.
The timing requirements increase while going from upper- to the lower-levels of the automation pyramid. While the bandwidth requirements increases in the opposite direction, since higher amount of data exchange are required.
The timing and bandwidth requirements are distributed in the automation pillar.



2.4.2 Communication Features vs. Delay Parameters

In order to build high deterministic control systems, ***communication features*** influencing the timing behavior of the exchanged traffics need to be analyzed. The most common are listed below:

- **Link-speed**: 100Mbps, 1Gbps or Multi-Gigabit
- **Transmission Selection Algorithm**: Preemptive-/Non-preemptive Strict-priority Algorithm and Preemptive-/Non-preemptive Time-Aware Shaper
- **Forwarding Mechanism**: Cut-Through vs. Store&Forward
- **Frame Structure**: Individual Frame vs. Summation Frame
- **Network Topology**: Line vs. Ring vs. Star vs. Tree
- **Number of forwarding hops**: Up to 100
- **Frame Size**: 64Byte – 1530Byte

[8] The frame size is given in bit and does not include the preamble, the start of frame delimiter (SFD) and the inter-frame gap (IFG)
[9] The frame transmission duration given in (7) includes the preamble and the SFD

Each communication feature is influencing one or more ***evaluation metric(s)***:

- end-to-end delay: is further divided into the following *delay parameters*: frame transmission and reception duration, interference delay and IFG duration
- Jitter
- Channel throughput
- Bandwidth loss

The communication features as well as the evaluation metrics (KPIs) are illustrated in table 14. Green cells shows whether a parameter can influence a particular KPI. If not than the cells are colored in red.

	Delay				Jitter	Channel Throughput	Bandwidth Loss
	d_Tx , d_Rx	d_Interference	d_fwd	d_E2E			
Data Rate	✓	✓	✓	✓	✓	✓	✓
Transmission Selection Algorithm	×	✓	✓	✓	✓	✓	✓
Forwarding Mechanism	×	×	✓	✓	×	×	×
Frame Structure	✓	×	✓	✓	×	✓	✓
Topology	×	✓	×	✓	✓	×	×
Number Of forwarding Hops	✓	✓	×	✓	✓	✓	✓
Frame Size	✓	×	✓	✓	×	✓	×

Table 14: Communication Features vs. Evaluation Metrics

- The **link-speed** is the only communication feature influencing all delay parameters and evaluation metrics. E.g. increasing the link-speed from 100Mbps to 1Gbps results into a ten time reduction of each of the delay parameters and jitter and a 10 time higher bandwidth utilization.
- The **transmission selection algorithm** on the other side has no effect on the frame reception or transmission duration but can reduce or even completely avoid the interference delay which as directly affect the jitter behavior.
- Supporting **cut-through** instead **of store&forward** reduces the single forwarding delays and thus the overall end-to-end delay. To get the highest benefit of the cut-through forwarding approach scheduling the time-critical frame is required.
- Unlike in **individual frame**, with the **summation frame** the Ethernet- and protocol-overheads required for each payload can be used for multiple payloads at once. This results into a higher bandwidth utilization and less bandwidth loss. However summation frames are mainly suitable for daisy-chain topologies (line, Ring).
- In a **star topology** the number of connected devices to the same forwarding hop, e.g. switch, is higher than in a **line / ring topology**. This might increase the interference probability on the egress ports. However the downlink stream communication is reduced in a star topology compared to a line / ring topology since the frames are forwarded over a single hop.

Modeling the communication features and evaluation metrics

For the configuration of a certain communication scenario (Figure 33), a set of communication features is selected as input parameters, which are influencing one or multiple evaluation metrics that are shown here as output parameters.

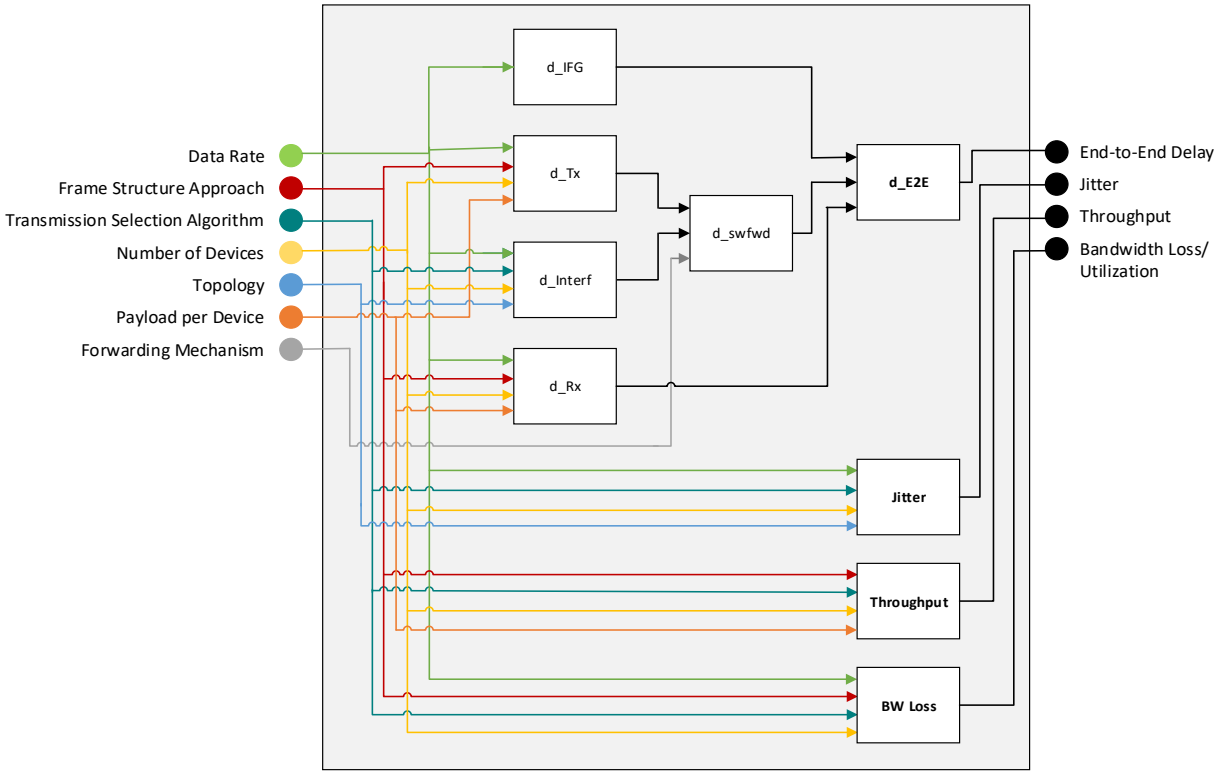


Figure 33: Modelling the communication features and evaluation metrics

Impact of Influencing Parameters on end-to-end delay

To forward a time-critical frame (1530Byte) over a single switch, different timing performances can be reached. The figure below shows the results for a store&forward switch with ~100Mbps network load. In order to cover the upper-bound worst-case interference, it is assumed that the background traffic is 1530Byte for IEEE802.1Q and 131Byte for IEEE802.1Qbu.

The calculated KPIs: end-to-end delay and jitter are very close for frame-preemption and frame-scheduling procedures for a single forwarding hop. For multiple forwarding hops and in the coexistence of concurring time-critical traffics, the timing performances of IEEE802.1Qbv is much better IEEE802.1Qbu.

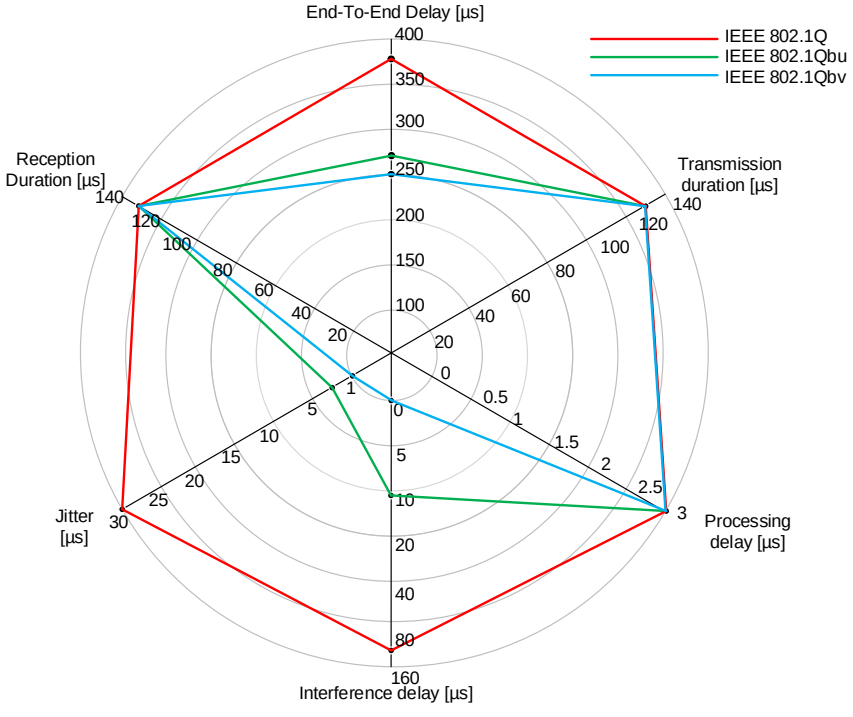


Figure 34: Partition of single-forwarding delay

2.4.3 Delay parameters

A single frame forwarding delay d_{fwd} is composed of the four delay parameters [67]:

- d_{RxTC}: frame reception duration and is dependent of the frame size and the activated link-speed on the reception / ingress port
- d_{PHY}: PHY reception and PHY transmission delays, ~300 nanoseconds each. It is not invesitigated in this study.
- d_{SwPx}: switch processing time and is dependent of the lookup search table (e.g. 4 microseconds for store&forward and 1 microsecond for cut-through)
- $d_{Interference}$: Interference delay and is equal to the transmission duration of the interfering (concurring) frame and the IFG duration. It is depending of the activated link-speed of the forwarding / egress port, the forwarding mechanism and the transmission selection algorithm

The forwarding mechanism, the link-speed of the ingress- and egress-ports as well as the transmission selection algorithm, defined in IEEE802.1Q and its enhancements .1Qbu and .1Qbv affect directly the frame forwarding delay. The impact of these features need to be computed and compared in order to reduce the forwarding delay and thus the overall end-to-end delay and cycle time. Therefor an example of 100Byte time-critical frame and 1530Byte interfering (non-time-critical) frame are taken as an example.

- **Green**: Interference delay $d_{Interference}$
- **Blue**: Reception Duration of time-critical frame d_{Rx}
- **Red**: Switch Processing delay d_{Px}

	.1Q - 100Mbps - S&F		.1Q - 1Gbps - S&F		.1Q - 100Mbps - CT		.1Q - 1Gbps - CT	
d_{RxTC}	8 [µs]	5,95 [%]	0,8 [µs]	4,7 [%]	8 [µs]	6,09 [%]	0,8 [µs]	5,70 [%]
d_{SwPx}	4 [µs]	2,98 [%]	4 [µs]	23,5 [%]	1 [µs]	0,76 [%]	1 [µs]	7,12 [%]
d_{Interf}	122,4 [µs]	91,07 [%]	12,24[µs]	71,8 [%]	122,4 [µs]	93,15 [%]	12,24[µs]	87,18 [%]
d_{Fwd}	134,4 [µs]	100 [%]	17,04[µs]	100 [%]	131,4 [µs]	100 [%]	14,04[µs]	100 [%]

Table 15: Forwarding Delay Results of certain Communication scenarios with IEEE802.1Q

	.1Qbu - 100Mbps - S&F		.1Qbu - 1Gbps - S&F		.1Qbu - 100Mbps – CT		.1Qbu - 1Gbps – CT	
d_{RxTC}	8 [µs]	34,13 [%]	0,8 [µs]	13,46 [%]	8 [µs]	39,14 [%]	0,8 [µs]	27,17 [%]
d_{SwPx}	4 [µs]	17,06 [%]	4 [µs]	67,29 [%]	1 [µs]	4,89 [%]	1 [µs]	33,97 [%]
d_{Interf}	11,44 [µs]	48,81 [%]	1,14 [µs]	19,25 [%]	11,44 [µs]	55,97 [%]	1,14 [µs]	38,86 [%]
d_{Fwd}	23,44 [µs]	100 [%]	5,94 [µs]	100 [%]	20,44 [µs]	100 [%]	2,94 [µs]	100 [%]

Table 16: Forwarding Delay Results of certain Communication scenarios with IEEE802.1Qbu

	.1Qbv - 100Mbps - S&F		.1Qbv - 1Gbps - S&F		.1Qbv - 100Mbps – CT		.1Qbv - 1Gbps – CT	
d_{RxTC}	8	66,67%	0,8	16,67%	8	88,89%	0,8	44,44%
d_{SwPx}	4	33,33%	4	83,33%	1	11,11%	1	55,56%
d_{Interf}	0	0,00%	0	0,00%	0	0,00%	0	0,00%
d_{Fwd}	12	100%	4,8	100%	9	100%	1,8	100%

Table 17: Forwarding Delay Results of certain Communication scenarios with IEEE802.1Qbv

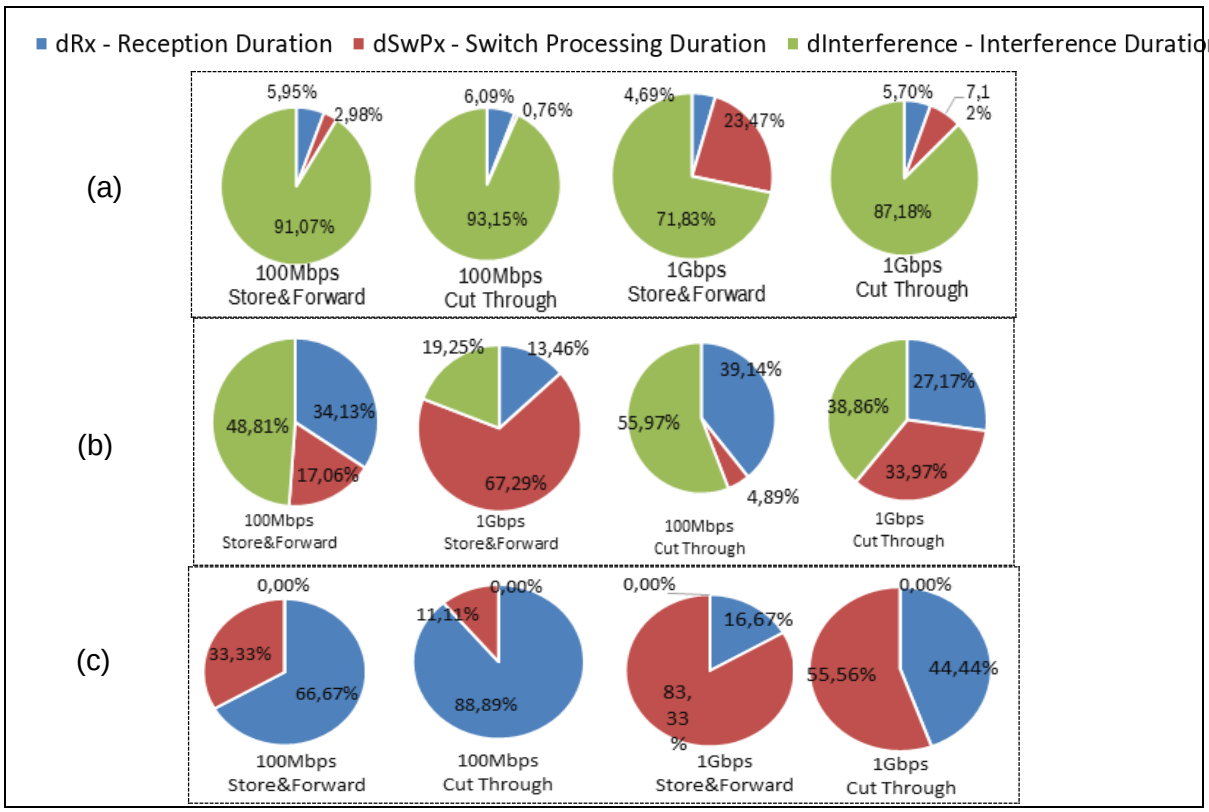


Figure 35: Delay parameters Distribution for (a) IEEE802.1Q (b) IEEE802.1Qbu (c) IEEE802.1Qbv

For all communication scenarios using the Non-Preemptive Strict-Priority Algorithm as described in IEEE802.1Q, the interference delay is always the biggest part of a single forwarding delay with values between 71% and 93% (figure12 (a)). Activating the frame-preemption strongly reduces the interference delay, which makes only 19% (figure12 (b)) once combined with Gigabit-Ethernet. Scheduling transmission time-slots and planning the transmission time points of the time-critical traffics can avoid the interference (figure12 (c)). The reception duration is higher than the switch processing delay for the time-aware shaper (IEEE802.1Qbv) with 100Mbps and is less with 1Gbps.

2.4.4 Impact of Frame Structure on timing behavior

This section analyzes the impact of the frame structure on the network performances (cycle time and bandwidth utilization). Because of its importance for the automation communication, the study focusses on the line topology. Considered are summation and individual frames.
The frame structure is basically affected by three parameters: payload per device (given in Byte), link-speed (100Mbps and 1Gbps) and number of connected devices in a line topology (maximum 100 devices).

Figure 14 shows the maximum possible number of devices that can be connected in a line topology for an increasing payload per device from 1 to 1450 Byte, such that the overall cycle time does not exceed 1 millisecond. In order to consider the impact of the frame structure without the effect of the transmission selection algorithms, it is assumed that the transmitted frames do not interfere with background traffics.

- For individual as well as summation frames, increasing the payload per device decreases the maximum number of devices that can be connected in a line topology, such that the overall cycle time does not exceed 1 millisecond.
- For the ***individual*** frame structure maximum 65 devices with 36 Byte payload each can be supported with 100Mbps and 100 devices with 512 Byte payload each for 1Gbps.

- For the ***summation*** frame structure maximum 100 devices with 44 Byte payload each can be supported with 100Mbps and 100 devices with 467 Byte payload each for 1Gbps.
- The benefit of summation frame over individual frame, in terms of maximum possible number of devices, is mainly for small payload per device (below 100Byte) in 100Mbps.

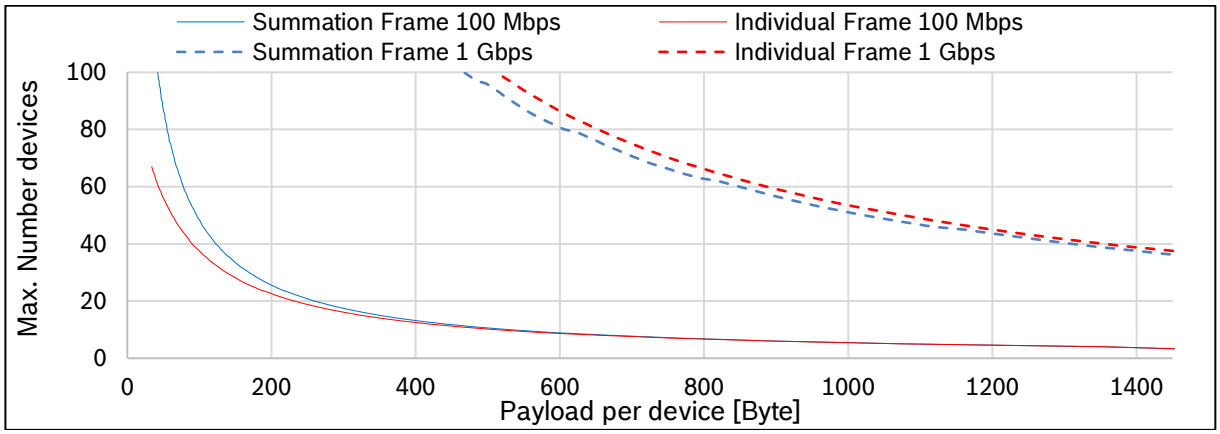


Figure 36: Maximum number of Devices for a cycle time <= 1 ms with an increasing Payload Per Device (No Interference is considered)

The overall cycle time increases linearly with the increasing number of devices in line topology.
For small payload per device (12Byte) and 100Mbps link-speed summation frame can strongly reduce the cycle time compared to individual frames. A further reduction of the cycle time is possible by increasing the link-speed to 1Gbps. However the effect of frame structure is no more noticeable (Figure 37 (Left)).
Figure 37 right shows a diagram, which compares the cycle times using summation-frame and individual-frame structures as a function of the payload per device. The study considers 100 devices connected in a line topology. For 100Mbps summation frame structure offers less cycle time than individual frame and is always worst for 1Gbps.
The maximum performance of individual frame over summation frame structure is reached at a payload of 82Byte per device. “This value represents the point, where the bottleneck moves from the propagation time $d_{fwd} + d_{medium}$ to the frame transmission duration d_{Tx}.” [9]

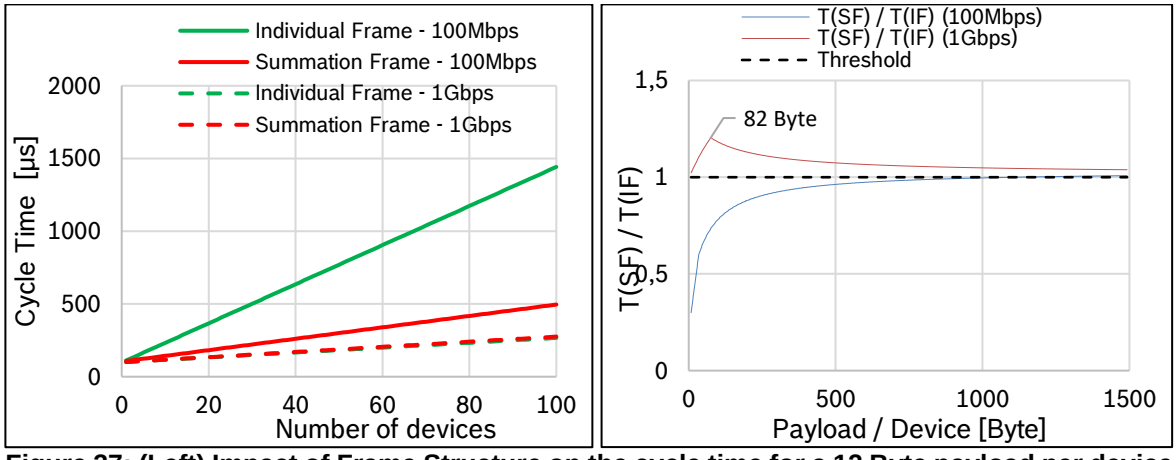


Figure 37: (Left) Impact of Frame Structure on the cycle time for a 12 Byte payload per device and 100Mbps - (Right) Individual Frame (IF) vs. Summation Frame (SF) as a function of the payload per device and 100 Devices in Line topology, T = Cycle Time (UpLink + Processing + DownLink durations)

The impact of the frame structure on the bandwidth utilization (effective used bandwidth to transmit payload data) is illustrated in Figure 38. Considered is a time-critical frame with an overhead of 50Byte (independent of the frame structure) and a constant payload of 50Byte per device (Figure 38 (left)).

- While the individual frame structure gives a constant bandwidth loss of 50%, the summation frame strongly reduce it below 5% for an increasing number of devices from 1 to 200. (Figure 38 (left))
- For a constant number of devices and an increasing payload per device the bandwidth loss remains below 11% very low for summation frame structure and strongly decreases for higher payload per device with individual frames. From 752Byte Payload per device there is no difference between the two frame structures (Figure 38 (right)).

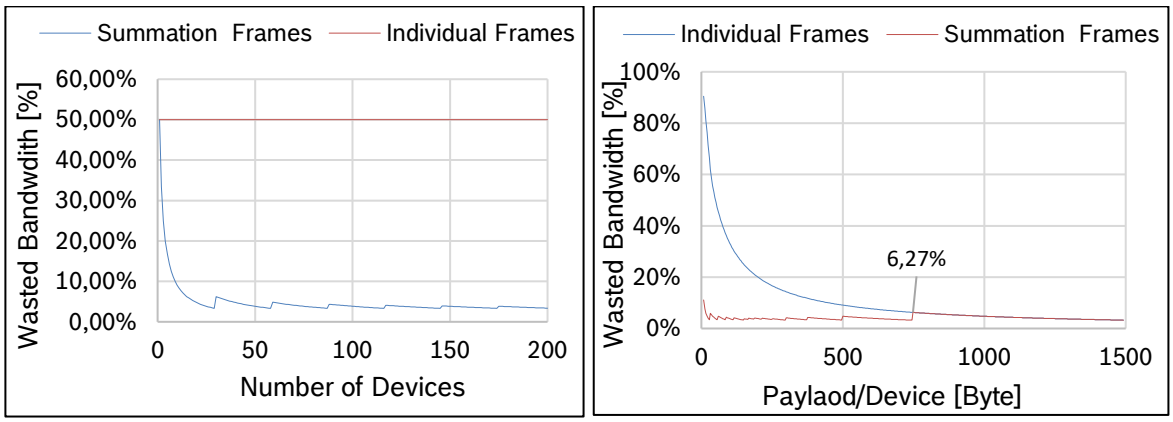


Figure 38: Impact of Frame structure on Bandwidth Utilization (100Mbps link-speed) (Left) constant payload/device = 50Byte - (Right) constant number of devices = 50 Devices

3 Formal and Simulative Timing-Analysis of IEEE 802.1 TSN

3.1 Introduction

In order to meet the hard timing requirements

- cycle times in the microseconds range e.g. packaging machines or machine tools (section 1.1.1.3),
- ultra-low end-to-end latency
- and latency variations (only for synchronization jitter) in the nanosecond range, e.g. printing machines

of highly deterministic industrial applications, such as Motion-Control, TSN specifies two mechanisms: Either *preempting the transmission* of the non-time-critical concurring frames or *scheduling* the time-critical frames to avoid any interference with the remaining traffic. This chapter analyses the timing behavior of the deterministic TSN-sub-standards, listed below (Table 18), for integration in the automation communication. Formal timing computation and extensive simulation scenarios have been conducted.

Each of the covered standards defines an appropriate traffic shaper (Table 18).

IEEE Standard	Transmission Selection Algorithm
IEEE802.1Q	Non-Preemptive Strict-Priority Algorithm (NP-SPA)
IEEE802.1Qbu	Preemptive Strict-Priority Algorithm (P-SPA)
IEEE802.1Qbv	Non-Preemptive Time-Aware Shaper (NP-TAS)
IEEE802.1Qbu and IEEE802.1Qbv	Preemptive Time-Aware Shaper (P-TAS)

Table 18: IEEE Standards and the appropriate Transmission Selection Algorithm

The communication parameters, affecting the timing behavior of the traffic, are listed below.

Communication Features	Variants
Transmission Selection Algorithm	• Strict-Priority Algorithm • Preemptive-Strict-priority Algorithm • Non-Preemptive Time-Aware Shaper
Link-speed	• Fast-Ethernet 100Mbps • Gigabit-Ethernet 1Gbps
Frame Forwarding Mechanism	• Store&Forward • Cut-Through
Frame Structure / Size	• Individual Frames – typically below 200Byte • Summation frames – typically above 500Byte
Topology	• Line, Ring, Star, Tree
Number of forwarding hops	• 5 – 50
Network Load	• 0% - 20% - 40% - 60% - 80% - 100%

Table 19: Communication Features affecting the KPIs of the deterministic applications

Multiple communication scenarios are designed with respect to the typical conditions of the automation communication. Each scenario is a combination of the features shown in Table 19. The scenarios are compared to each other based on the results given by the KPIs. The goal is to check which scenarios can meet the timing requirements of the industrial applications. The next sections show that the TSN features alone cannot meet the timing requirements of the deterministic industrial applications. Therefore each of the transmission selection algorithms listed in Table 18 is combined with other communication features such as increasing the link-speed and supporting cut-through instead of store&forward.

3.1.1 Contributions

The major contributions of this study are

- Comparing the timing behavior of the Ethernet transmission selection algorithms shown in Table 18
- The TSN-features IEEE802.1Qbu and IEEE802.1Qbv are not sufficient to meet the timing requirements of certain highly deterministic industrial applications, e.g. motion control.
- Deducing the parameters influencing the communication in the industrial automation and analyzing their timing behaviors
- Designing a multiple communication scenarios, based on a set of combinations between the *transmission selection algorithms* and the *additional communication features*, to meet the timing requirements of the deterministic industrial applications

3.1.2 Key Performance Indicators

The timing performance of the deterministic industrial applications is given by the key performance indicators, listed below, and are strongly affected by the communication features and parameters as shown in Table 19.

- Upper-Bound Worst Case end-to-end delay from transmitter to receiver
- Forwarding delay of a single hop
- Interference delay
- Delay Variation (Jitter)
- Bandwidth Utilization

3.1.3 Network Design

Because of its importance for the automation communication, a line topology [9] has been selected. In order to evaluate the timing behavior of Ethernet, its transmission selection algorithms (table) are combined with multiple communication features. Each set of combination builds a proper scenario.
The basic network design shown below, is adjusted in each sceanrio with a certain combination of the communication features to get the best performances.

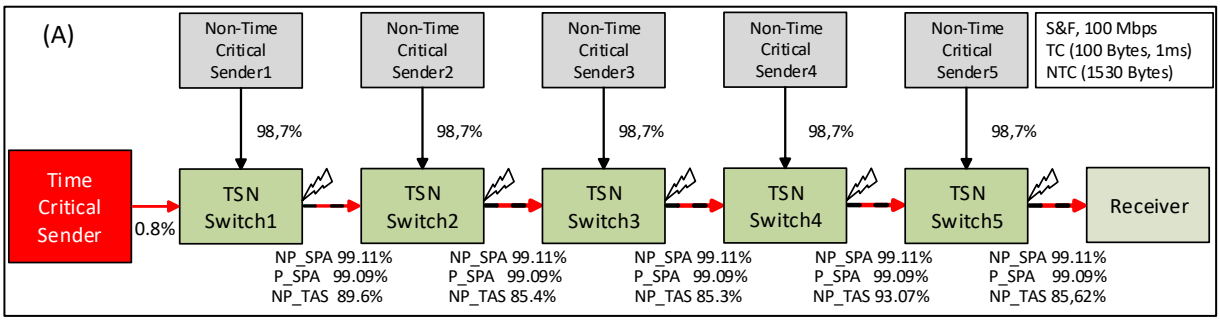


Figure 39: Network Design

The network consists of four types of devices:

- time-critical transmitter, generating and transmitting a time-critical traffic
- non-time-critical transmitters, each generating and transmitting a non-time-critical traffic
- traffic receiver, receiving all data-traffics
- forwarding hops (TSN-switches), each receiving and forwarding the traffics.

Each forwarding hop is connected on three ports to devices. Two are ingress ports, on which time-critical and non-time-critical traffics are received and one is an egress port, on

which all received traffics are forwarded. All devices support 100Mbps and 1Gbps link-speed but only one link-speed is selected at a time to avoid a bottleneck.
In order to compare the timing performances of the transmission selection algorithms, the following concept is considered: A time-critical frame is concurring with multiple background traffics on the network resources (egress port of each forwarding hop). The communication features of the network infrastructure (forwarding hops) affect the KPIs (Section 3.1.2) for the time-critical frame. Multiple scenarios are considered in the next sections.

3.1.4 Traffics Configuration

There are basically two types of traffics: time-critical and non-time-critical. A time-critical traffic transmitter generates and transmits a 100 Byte frame each 1 millisecond, which results into a throughput of 0.74 Mbps at 100Mbps or ~0.074 Mbps at 1Gbps link-speed. The time-critical frame is forwarded over all forwarding hops to the receiver device following this path:
Time-Critical Transmitter – TSN-Switch1 – TSN-Switch2 – … – TSN-Switch(n) – Receiver
The remaining traffics are synchronously transmitted from the non-time-critical transmitters and are considered as “concurring” frames. Based on the activated transmission selection algorithm an interference between the time-critical and non-time-critical frames might occur, reduced or avoided, which results into different timing behavior: end-to-end delay and jitter. The egress port of each forwarding hop supports the following transmission selection algorithms: preemptive and non-preemptive strict-priority algorithm as well as the non-preemptive time-aware shaper.

Traffic Type	Size [Byte]	Transmission Interval [ms]	Throughput [Mbps]
Time-Critical	100	1	0,736
Non-Time-Critical1	1530	Stream	0 to 98,7
Non-Time-Critical2	1530	Stream	0 to 98,7
Non-Time-Critical3	1530	Stream	0 to 98,7
Non-Time-Critical4	1530	Stream	0 to 98,7
Non-Time-Critical5	1530	Stream	0 to 98,7

Figure 40: Common Traffics configuration

The upcoming sections analyses the timing behavior of the Ethernet Transmission Selection Algorithms. Each section presents multiple scenarios, in which the formal timing behavior of the transmission selection algorithm is described. Since certain KPIs, such as the delay variation (jitter) or whether the formally computed maximum and minimum possible end-to-end delay are reachable, can only be answered through simulation with predefined parameters (e.g. variable amount of background traffics from 0% to ~100% network loads). Each scenario is simulated and present the results in two graphs:

- distribution of the end-to-end delay under a network load of 20%
- minimum, maximum, average and average deviation of the end-to-end delay under increasing network loads from 0% to ~100%

Note: The non-time critical traffics are used to interfere with the time-critical frames and are synchronously transmitted from several nodes.

3.2 IEEE802.1Q – Non-Preemptive Strict-Priority Algorithm

The non-preemptive strict-priority algorithm, defined in IEEE802.1Q, is a traffic shaper that extends the Ethernet frame format with the 4 Byte optional data-field: “VLAN-Tag”. The VLAN-tag is used to identify the time- or mission-critical traffics in order to accelerate their forwarding through the network.
The main drawback of this traffic shaper is the limited timing performances, due to the interference with the background traffics. The jitter increases with the amount of exchanged traffics within a network.

3.2.1 Formal Flow Analysis

For an egress port supporting only the non-preemptive strict-priority algorithm, as defined in IEEE802.1Q, a time-critical frame with the highest priority PCP7 might be ready for transmission but cannot be directly selected due to an interference with another frame in transmission. For the upper-bound worst-case scenario, the worst-case interference delay is given by a concurring frame of the size 1530Byte. Together with the IFG the resulting interference delay is 1542Byte means 123.36µs at 100Mbps and ~12.34µs at 1Gbps.
The upper-bound worst-case forwarding delay over a single hop and over multiple hops are respectively illustrated in figure 41 and figure 42.

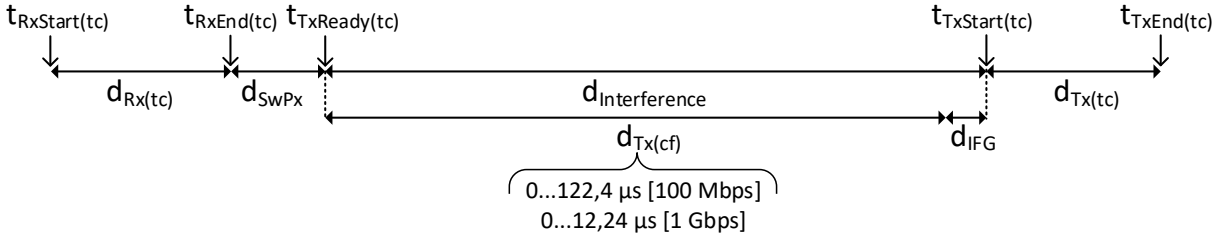


Figure 41: Formal - Upper Bound Worst Case Single Forwarding Delay

Figure 42 illustrates the flow diagram of a time-critical frame (red) forwarded over 5 hops supporting the non-preemptive strict-priority algorithm. In a worst case scenario, the time-critical frame would interfere with a background frame (white) in each forwarding process. Two cases of the time-critical end-to-end delay are shown: (left) with store&forward and (right) cut-through.

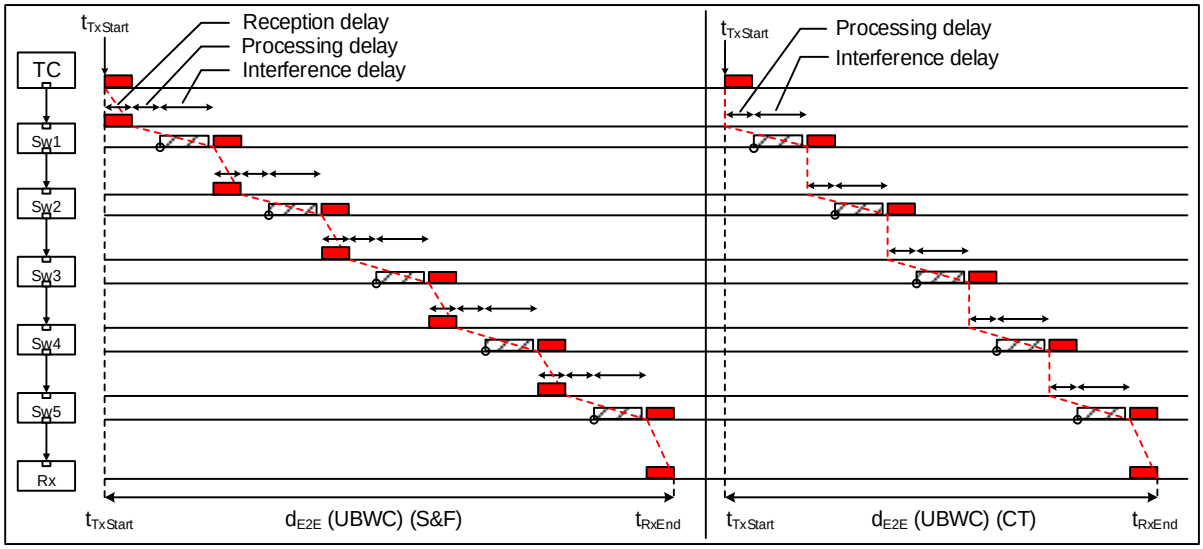


Figure 42: Flow Diagram of a time-critical frame using non-preemptive Strict-Priority Algorithm (IEEE802.1Q) in combination with Store&Forward (left) vs. Cut-Through (right)

3.2.2 Scenario 1

Configuration

Table 20 lists the activated communication features for the first scenario.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps
Frame Forwarding Mechanism	Store&Forward

Table 20: Features Configuration

Simulation Results

The distribution of the end-to-end delay for a sample duration of 1 second (=1000 time-critical cycles) and a network load of ~20% is shown on the left side, while the max, min,

average and average deviation (jitter) are shown on the right side for an increasing network load from 0 to ~100%.

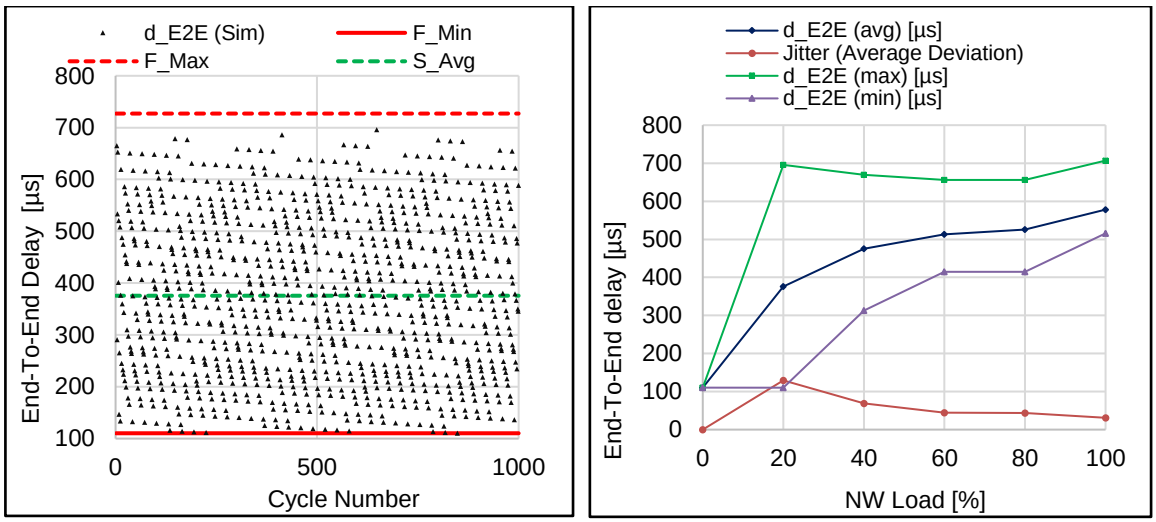


Figure 43:

- **(left) End-To-End delay distribution over 5 (Store&Forward, 100Mbps) switches using IEEE802.1Q under 20% Network load**
- **(right) Timing behavior over 5 (Store&Forward, 100Mbps) switches using IEEE802.1Q under different network loads**

Results Observation

Due to the interference with background traffics, the simulation results of the end-to-end delay (left) over 5 hops is distributed between 110µs and 695µs, resulting in a big distribution range of ~600µs and an end-to-end average delay of ~375µs. The simulation values are within the computed formal range [110,3µs – 727,1µs]. These formal values are used to check the correctness of the simulation model. The minimum formal value is given by zero interference, while the maximum is due to an upper-bound worst-case interference delay on each forwarding hop. The results prove that the IEEE802.1Q does not deliver reliable timing behavior, which clarifies why it is not used in highly deterministic applications. This is the reason why the TSN enhancements of IEEE802.1Q have been specified. Neither the resulting jitter nor the relative big delay are suitable for most deterministic applications.

The figure on the right shows that increasing the network load of background traffics increases the end-to-end delay (min, average, max) but decreases the jitter from ~129µs at 20% to ~31µs at 100%. This is due to the increased probability of interference, which decreases the difference between the minimum and maximum resulting end-to-end delay. However most networks have a background traffic below 80%. This network setup is only suitable for relatively small networks (number of devices in a line toplogy < 5), where a jitter of few hundreds microsecond can be tolerated. Such soft-real-time applications are not limited to the line topology, which might reduce the delays between the master- and the single slave-devices.

Correlation

The correlation coefficient r is a measurement that shows the strength and direction of a relationship between two variables using a value between -1 and 1. The closer r to positive or negative 1, the stronger the relationship.

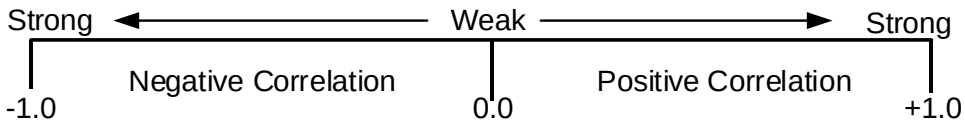


Figure 44: Correlation coefficient r shows the strength and direction of correlation

The correlation coefficient r is

- **strong positive** (0,887) between the average end-to-end delay and network load
- **weak negative** (-0,125) between the jitter and the network load

The results of the combination of store&forward, 100Mbps and non-preemptive strict-priority cannot meet the timing requirements of motion applications.

3.2.3 Scenario 2

Configuration

The activated communication features are listed below.

Communication Features	Variants
Link-speed	Gigabit-Ethernet 1Gbps
Frame Forwarding Mechanism	Store&Forward

Table 21: Features Configuration

Simulation Results

The distribution of the end-to-end delay for a sample duration of 500 millisecond (=500 time-critical cycles) and a network load of ~20% is shown on the left side The max, min, average and average deviation (jitter) are shown on the right side for an increasing network load.

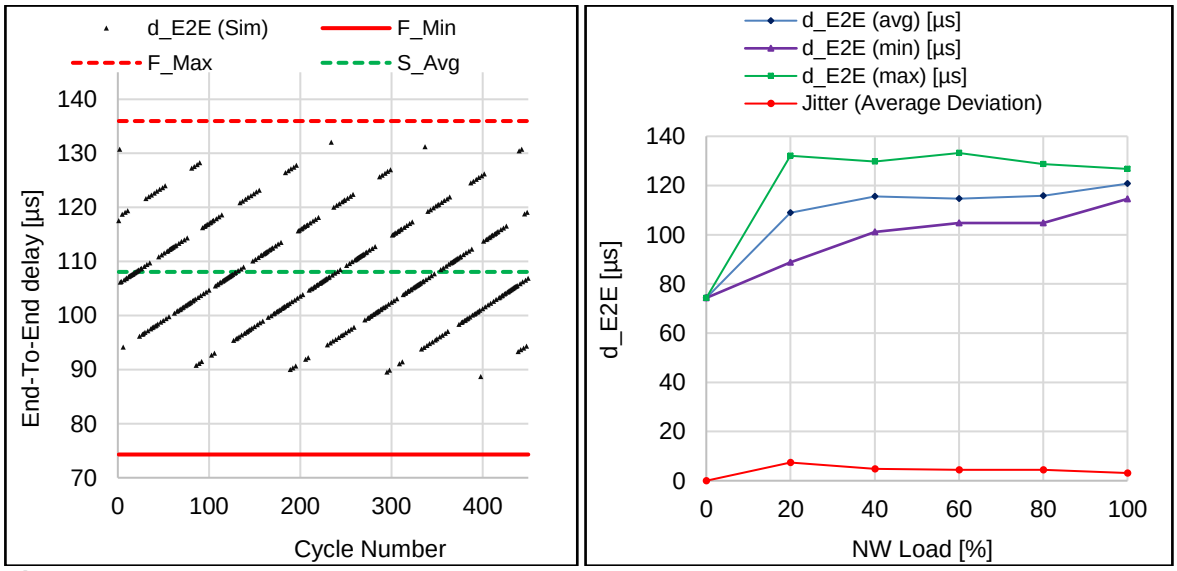


Figure 45:

- **(left) End-To-End delay distribution over 5 (Store&Forward, 1Gbps) switches using IEEE802.1Q under 20% Network load**
- **(right) Timing behavior over 5 (Store&Forward, 1Gbps) switches using IEEE802.1Q under different network loads**

Results Observation

The figure on the left shows the distribution of the end-to-end delay (points) for the time-critical traffic over 450 samples at a constant network load of 20%. The continuous red line refers to the formal calculated end-to-end delay (74,3µs) without any interference throughout the path from the transmitter to the receiver. The dashed line (red) shows the formal calculated upper-bound worst-case delay (~136µs) if the time-critical traffic interferes on each forwarding switch with the background traffic (max. size).

Increasing the link-speed from 100Mbps (figure 43) to 1Gbps (figure 45) reduces the the end-to-end delay and the jitter. E.g. the maximum end-to-end delay at 100% network load is strongly reduced from ~706µs at 100Mbps to only ~126µs at 1Gbps and the jitter from ~31µs to ~3µs. For machine-to-machine communication the Gigabit-Ethernet with store&forward approach is enough for most industrial applications.

The distribution range at 20% network load is also reduced from ~375µs = [110 ; 695] to ~108µs = [88 ; 132].
Increasing the network load using 1Gbps link-speed does not strongly increase the end-to-end delay compared to the previous scenario. The average value is increased from ~74µs without background traffic to ~120µs with ~100% network load. The jitter on the other side is halved from ~7µs at 20% to ~3µs at ~100%.

The simulation results of the combination: store&forward, 1Gbps and non-preemptive strict-priority come very close to the timing requirements of motion applications for small number of devices but cannot meet it (end-to-end delay variation above 1µs). However this is suitable for more soft-real-time applications especially for the machine-to-machine communication.

3.2.4 Scenario 3

Configuration
The activated communication features are listed in below.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps
Frame Forwarding Mechanism	Cut-Through

Table 22: Features Configuration

Simulation Results
Activating cut-through instead of store&forward as in scenario1 does not strongly reduce the distribution range of the end-to-end delay, which remains by ~600µs unsuitable for use in highly deterministic applications. This is because the most benefit of cut-through can only be reached if the time-critical traffic frames do not interfere with other data-traffics.

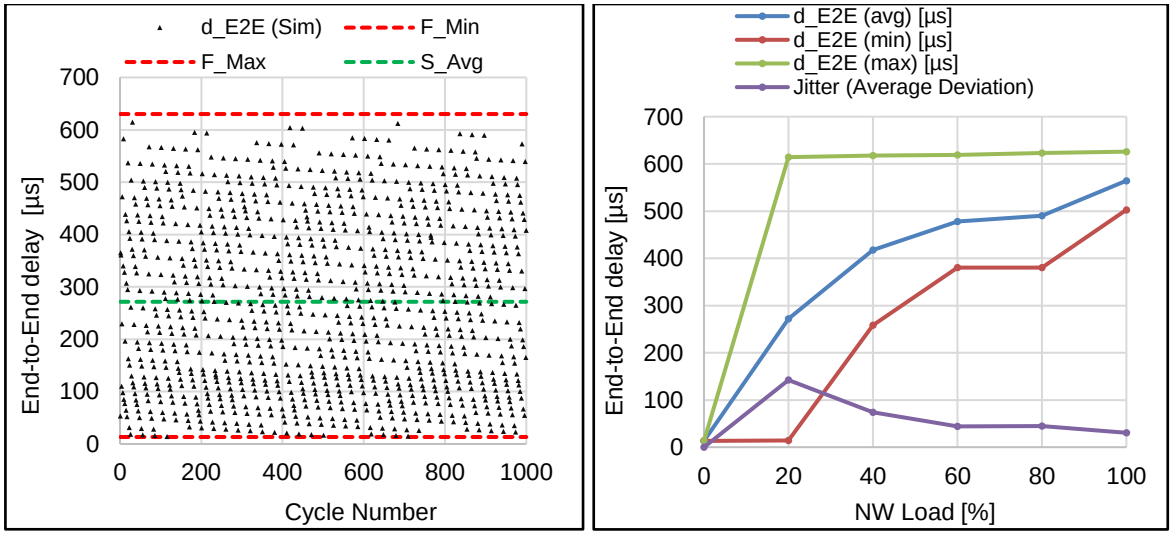


Figure 46

- **(left) End-To-End delay distribution over 5 (Cut-Through, 100Mbps) switches using IEEE802.1Q under 20% Network load**
- **(right) Timing behavior over 5 (Cut-Through, 100Mbps) switches using IEEE802.1Q under different network loads**

Results Observation
The maximum end-to-end delay at ~100% network load and 100Mbps link-speed is weakly reduced from ~706µs with store&forward to ~625µs with cut-through.

Combined with 100Mbps and non-preemptive strict-priority, cut-through forwarding has no big impact on the timing behavior. This is due to the interference with the background traffic, which does not evolve the big benefit of the cut-through approach. In order to get the maximum benefit from this approach, avoiding or strongly reducing the interference is mandatory. This combination is therefor still not suitable for motion applications. But is

sufficient for most applications in the industrial communication with soft-real-time requirements.

3.2.5 Scenario 4

Configuration

Scenario 4 combines the cut-through forwarding mechanism and 1Gbps link-speed with non-preemptive strict-priority algorithm.

Communication Features	Variants
Link-speed	Gigabit-Ethernet 1Gbps
Frame Forwarding Mechanism	Cut-Through

Table 23: Features Configuration

Simulation Results

Increasing the link-speed to 1Gbps and supporting cut-through instead of store&forward reach the best timing behavior for the non-preemptive strict-priority algorithm IEEE802.1Q. Although the distribution range of the end-to-end delay for 20% network load has been strongly reduced from 375µs (scenario1) to ~61µs = [~6 ; ~67], the timing behavior is still far away from the requirements of the highly deterministic industrial applications. With focus on those using industrial-Ethernet protocols, in which the synchronization information is triggered by the reception of the control-data (e.g. sercos).

The average end-to-end delay under 20% network load is reduced from ~375µs in scenario1 (store&forward, 100Mbps) to ~108µs in scenario2 (store&forward, 1Gbps), ~271µs in scenario3 (cut-through, 100Mbps) and ~30µs in scenario4 (cut-through, 1Gbps). However this timing behavior is sufficient for most applications in the industrial automation.

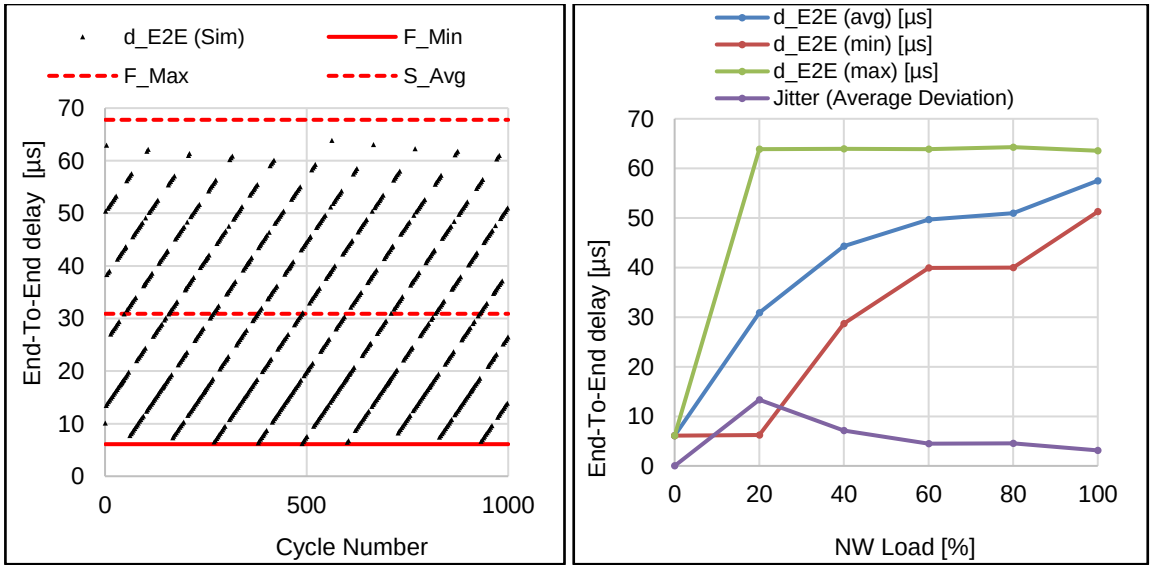


Figure 47:

- **(left) End-To-End delay distribution over 5 (Cut-Through, 1Gbps) switches using IEEE802.1Q under 20% Network load**
- **(right) Timing behavior over 5 (Cut-Through, 1Gbps) switches using IEEE802.1Q under different network loads**

3.2.6 Summary

The simulation results of the end-to-end delay and jitter of a time-critical frame transmitted each one millisecond from a transmitter to a receiver over 5 switches supporting the non-preemptive strict-priority algorithm and under ~100% network load are shown in the figure below.

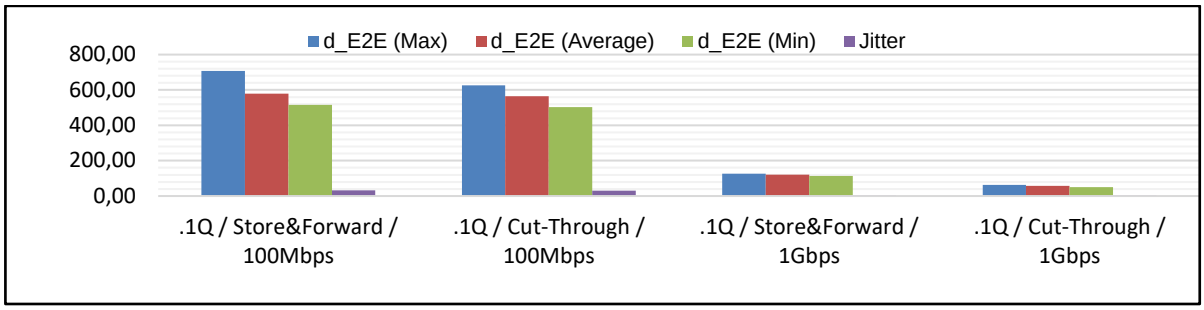


Figure 48: Timing improvement for IEEE802.1Q (Non-Preemptive Strict-priority Algorithm) at 100% network load

Supporting cut-through instead of store&forward under a high network load has no significant impact on reducing the end-to-end delay compared to increasing the link-speed to 1Gbps. Cut-through reduces the forwarding delay of a particular frame if no interference occurs.
The best timing behavior is reached with 1Gbps and cut-through. While the end-to-end delay KPI of this combination can be accepted for the highly deterministic industrial applications, the jitter is still too high and cannot be tolerated.

3.3 IEEE802.1Qbu – Preemptive Strict-priority Algorithm

3.3.1 Formal Flow Analysis

The interference of an express time-critical frame with the remaining traffics cannot be avoided using the preemptive strict-priority algorithm. The interference delay depends on the class and size of the concurring frame(s).

- The upper-bound worst-case scenario is given by an interference: either with a time-slot of a scheduled express frame(s) or with a single non-scheduled express frame. In both cases the blocked express frame needs to wait until the transmission of other express frame(s) is finished.
- The express frame could also interfere with a preemptable frame that, due to its size (<131Byte), cannot be preempted or with a preemptable frame at its tail.
- The best case is an interference with a preemptable frame. The resulting interference delay is given by the durations of the preemption-operation and the IFG.

The figures below illustrate the types of interferences described above within an end-to-end delay.

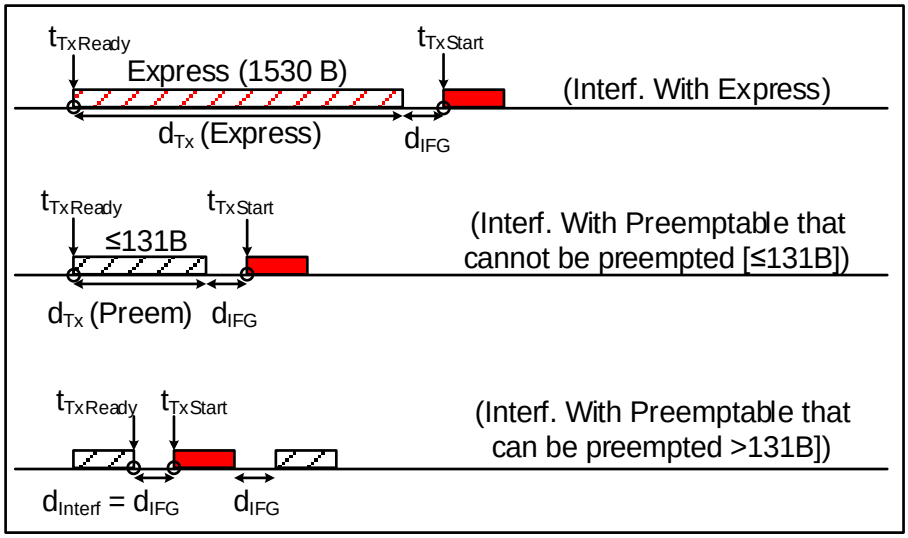


Figure 49: Formal - Upper Bound Worst Case Single Interference Delay of a time-critical frame using preemptive Strict-Priority Algorithm (IEEE802.1Qbu)

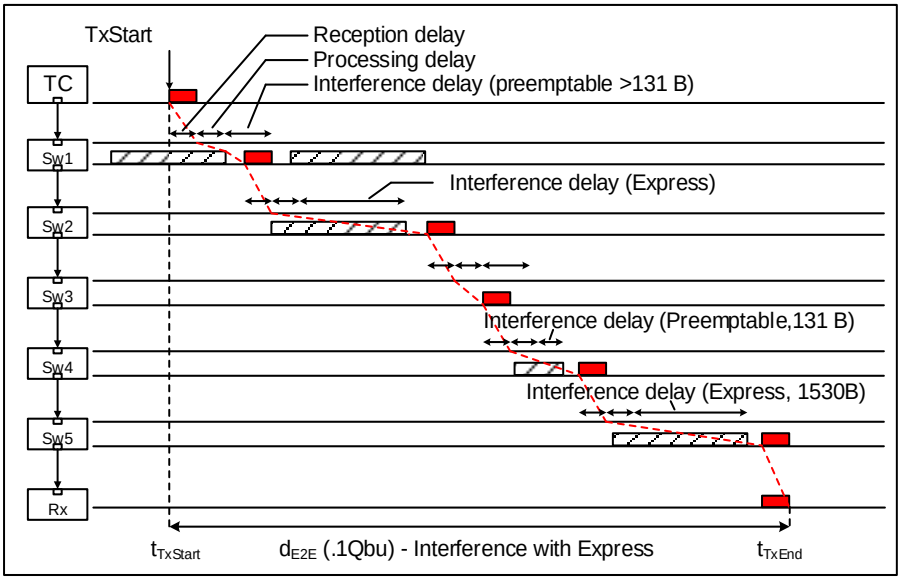


Figure 50: Flow Diagram of a time-critical frame using preemptive Strict-Priority Algorithm (IEEE802.1Qbu) – Interference with express- (red) and preemptable-frames (green)

Since it is hard to estimate the timing behavior in such a context, the described cases are analyzed in the simulation. The scenarios and the results are discussed in the next sections.

3.3.2 Interference with Preemptable Traffic (1530Byte)

3.3.2.1 Scenario 1

Configuration

In this scenario the time-critical traffic could interfere on each forwarding switch with a 1530Byte preemptable frame from the background traffics. The preemptable frames are synchronously transmitted from transmitters distributed over the forwarding switches. Fragments of preempted frames are not re-transmitted after the preemption process.
The activated communication features are listed below.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps
Frame Forwarding Mechanism	Store&Forward

Table 24: Features Configuration

Results Observation

The distribution of the end-to-end delay for a sample duration of 1 second (=1000 cycles of the time-critical traffic) for network load of ~20% is illustrated below.

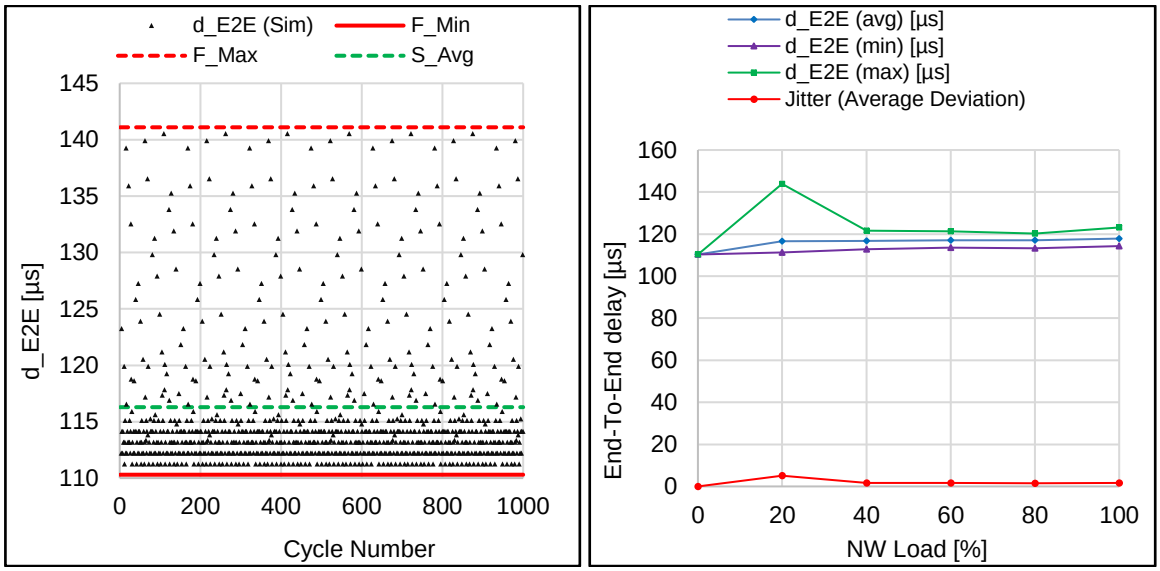


Figure 51:

- **(left) End-To-End delay distribution over 5 (Store&Forward, 100Mbps) switches using IEEE802.1Qbu under 20% Network load**
- **(right) Timing behavior over 5 (Store&Forward, 100Mbps) switches using IEEE802.1Qbu under different network loads**

Few simulative values are very close to the formal computed maximum end-to-end delay. But the distribution density is closer to the minimum formal computed value, which gives an average delay of 115µs. This is due to the strongly shortened interference delay from ~125µs to ~11µs by preempting the transmission of the background traffic. However in certain cycle the preemption is not possible, if the remaining bytes to be transmitted are less than the minimum Ethernet frame length 64Byte.

As shown in the right figure, increasing the network load from 20% to ~100% has no significant impact on the average end-to-end delay and the resulting jitter. This is due to the strongly reduced interference delay as explained above.

The timing behavior of the preemptive approach is better than all scenarios with the non-preemptive strict-priority algorithm.

3.3.2.2 Scenario 2

Configuration
Scenario 2 increases the link-speed to 1Gbps.

Communication Features	Variants
Link-speed	Gigabit-Ethernet 1Gbps
Frame Forwarding Mechanism	Store&Forward

Table 25: Features Configuration

Simulation Results

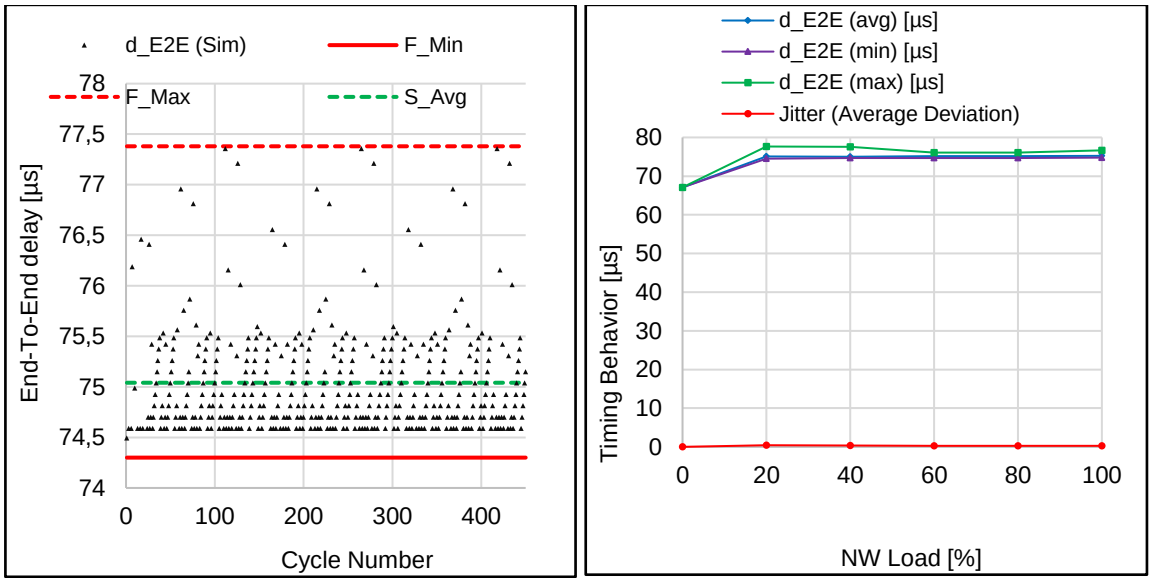


Figure 52:

- **(left) End-To-End delay distribution over 5 (Store&Forward, 1Gbps) switches using IEEE802.1Qbu under 20% Network load**
- **(right) Timing behavior over 5 (Store&Forward, 1Gbps) switches using IEEE802.1Q under different network loads**

Results Observation
Increasing the link-speed to 1Gbps reduces the overall end-to-end delay as expected with the formal computation. Unlike the non-preemptive strict-priority algorithm, the average delay parameter could not be strongly reduced by simply increasing the link-speed. Few simulation values are close to the formal computed maximum delay. Increasing the link-speed from 100Mbps to 1Gbps reduces only the transmission duration by a factor of 10 but not the processing delay of each forwarding hop.
In general increasing the network-load has no significant impact on the timing behavior. The minimum and maximum values remains in the range between 70µs and 80µs. However the jitter remains below 1µs.
The combination: non-preemptive strict-priority algorithm, 1Gbps and store&forward meets the timing requirements of the highly determinsitic industrial applications for small number of devices in line.

3.3.2.3 Scenario 3

Configuration
The activated communication features are listed below.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps
Frame Forwarding Mechanism	Cut-Through

Table 26: Features Configuration

Simulation Results

The simulation results of the end-to-end delay and the jitter for the time-critical traffic are illustrated below.

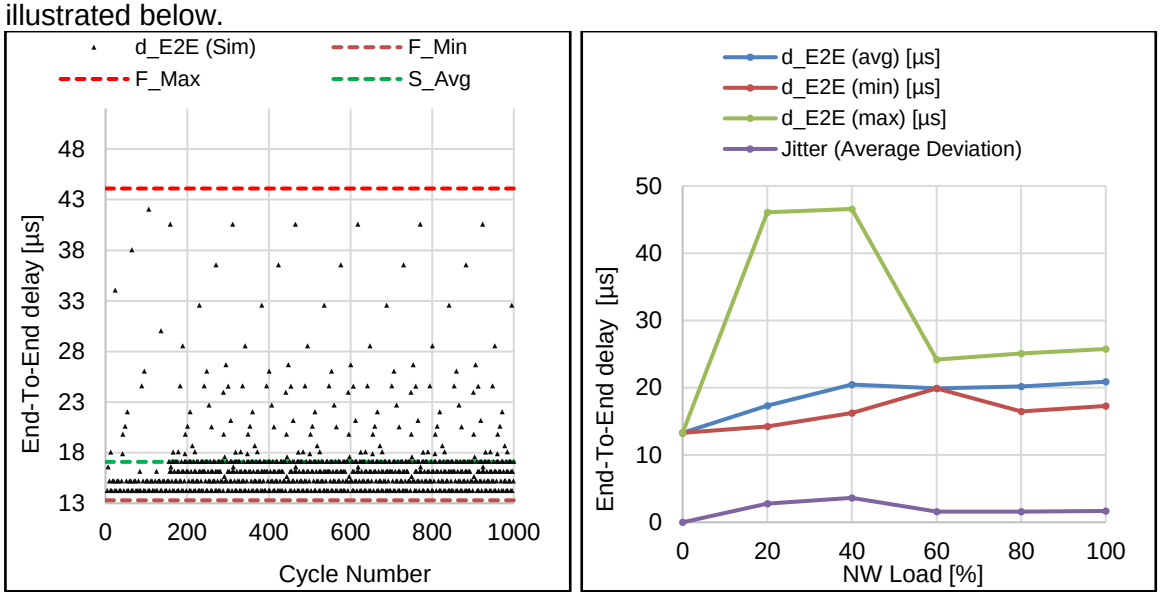


Figure 53:

- **(left) End-To-End delay distribution over 5 (Cut-Through, 100Mbps) switches using IEEE802.1Q under 20% Network load**
- **(right) Timing behavior over 5 (Cut-Through, 100Mbps) switches using IEEE802.1Q under different network loads**

Results Observation

Supporting cut-through instead of store&forward significantly reduced the formal and simulative end-to-end delay by almost 70%. Increasing the network load has further no strong impact on the timing behavior. Combining cut-through with the preemptive approach does not deliver better timing behavior than the Gigabit and Store&Forward approach. However this is due to the huge impact of increasing the link-speed on reducing the resulting jitter.

3.3.2.4 Scenario 4

Configuration

The activated communication features are listed below.

Communication Features	Variants
Link-speed	Gigabit-Ethernet 1Gbps
Frame Forwarding Mechanism	Cut-Through

Table 27: Features Configuration

Simulation Results

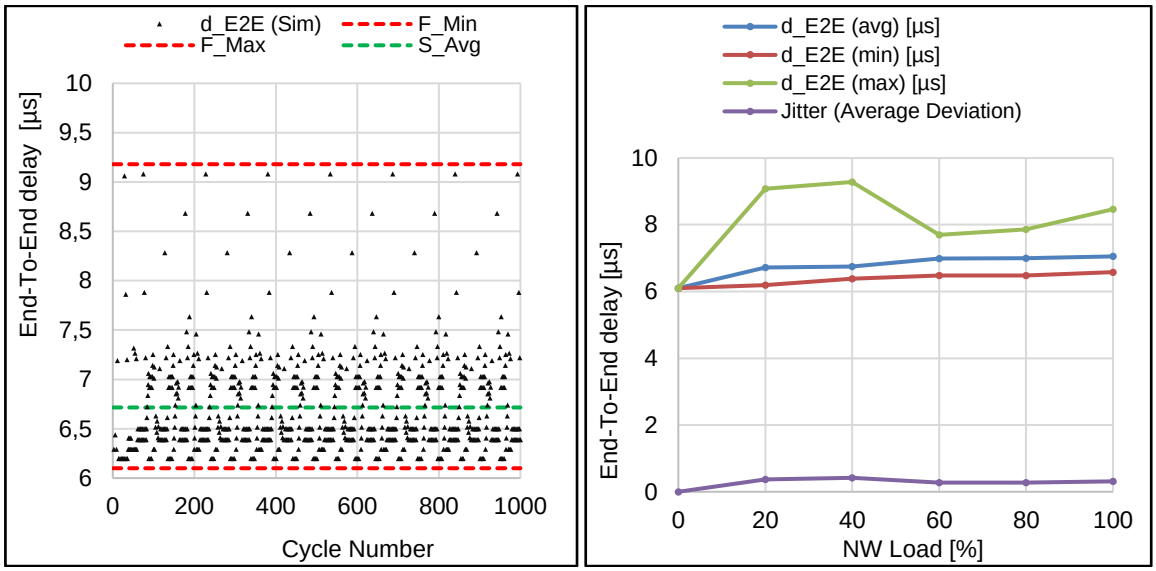


Figure 54:

- **(left) End-To-End delay distribution over 5 (Cut-Through, 1Gbps) switches using IEEE802.1Qbu under 20% Network load**
- **(right) Timing behavior over 5 (Cut-Through, 1Gbps) switches using IEEE802.1Qbu under different network loads**

Results Observation

Combining cut-through and 1Gbps with the preemptive strict-priority algorithm has the best timing behavior compared to the previous analyzed scenarios. Independent of the amount of the background traffic, an end-to-end delay below 10µs and a jitter below 1µs over 5 switches can be reached.

Increasing the link-speed, preempting the transmission of the interfering background traffic as well as supporting the cut-through approaches reduce the end-to-end delay and thus the resulting jitter more than all scenarios of the non-preemptive strict-priority approach.

This combination is very attractive for deterministic industrial applications especially for the machine-to-machine communication.

3.3.2.5 Summary

The timing behavior of a non-scheduled express frame transmitted over 5 switches with configurable forwarding mechanism (store&forward, cut-through) and link-speed (100Mbps, 1Gbps) is illustrated below.

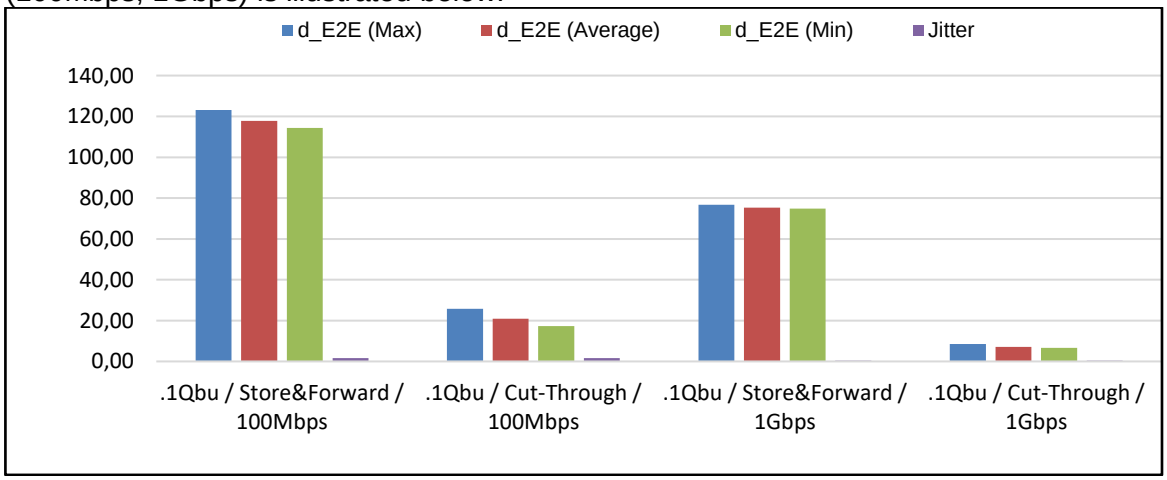


Figure 55: Timing improvement for IEEE802.1Qbu (Preemptive Strict-priority Algorithm)

All scenarios meet the requirement of an end-to-end delay below one millisecond and a jitter below 1 microsecond.

3.3.3 Interference with Preemptable Traffic (131 Byte)

Scenario Description

A non-scheduled express frame interferes with preemptable frames of the size 131Byte. Because of their size, the preemptable frames cannot be preempted and would delay the non-scheduled express frame in case of interference.

The scenario recognizes five store&forward switches and covers different link-speeds: 100Mbps (left – scenario1) and 1Gbps (right – scenario2) as well as a network load of 20%.

Traffics Configuration

Traffic Types	Size [Byte]	Priority (PCP)	Transmission Interval
Non-Scheduled Express	100	7	1 ms
Non-Scheduled Preemptable	131	1	Stream

Table 28: Traffics Configuration

Simulation Results

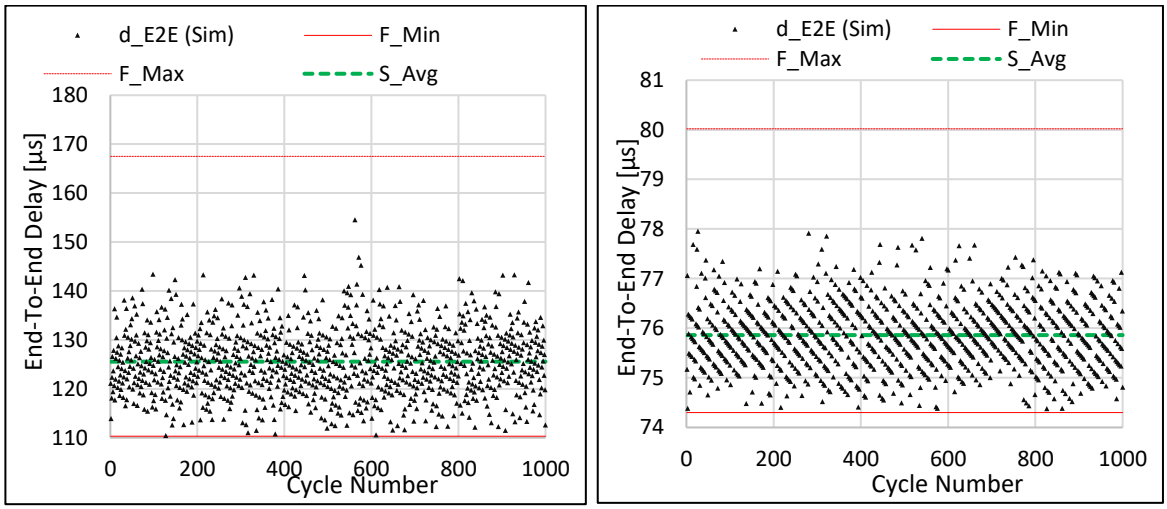


Figure 56: End-To-End delay distribution over 5 Store&Forward switches using IEEE802.1Qbu under 20% Network load – Interference with 131Byte preemptable frames. [Left - 100Mbps; Right - 1Gbps]

Results Observation

An interference with a preemptable frame that cannot be preempted results into an additional delay of almost ~12µs per forwarding hop. The formal upper-bound worst-case delay is given by multiplying the single interference delay by the number of forwarding hops. This means that the non-scheduled express frame would be ready for transmission few nanoseconds after the preemptable frame has been selected for transmission.

The simulation results shown above proves that this case is very hard to reach. That explains why the formal maximum value could not be reached also after a simulation duration of 6 hours. Compared to scenario1 and scenario2 of the previous section, the interference with preemptable traffics that cannot be preempted does not have a significant impact on the timing behavior. The values have been slightly increased.

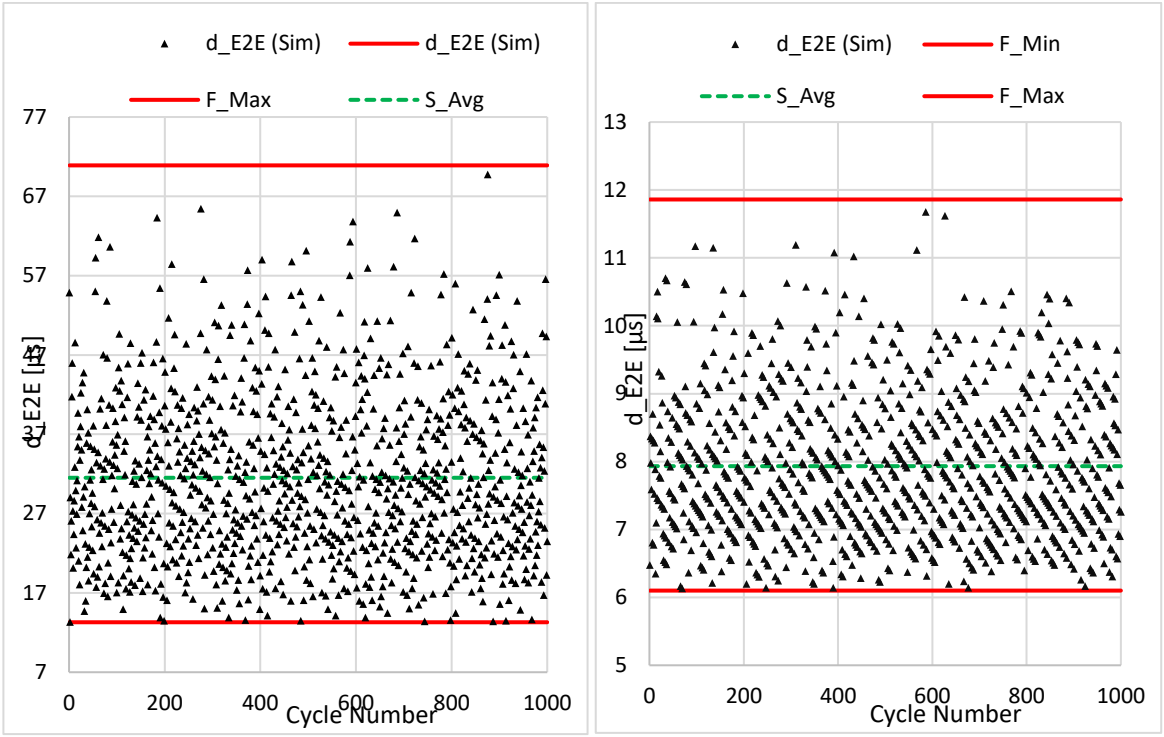


Figure 57: End-To-End delay distribution over 5 Cut-Through Ethernet switches using IEEE802.1Qbu under 20% Network load – Interference with 131Byte preemptable frames. [Left - 100Mbps; Right - 1Gbps]

3.3.4 Interference with Express- and Preemptable-Traffics

In this section the non-scheduled express traffic 1 interferes with the background traffics: *non-scheduled express traffic 2* on the egress port of the second switch and *non-scheduled preemptable* traffics on the remaining switches. The background traffics make together a network load of ~100%. The traffics configurations are illustrated below. The non-scheduled express traffics 1 and 2 are transmitted from different nodes. Unlike the earlier cases, the goal of this section is to investigate the timing behavior of time-critical traffic in the coexistence of other time-critical traffic, of which the transmission cannot be preempted.

Traffics Configuration

Traffic Types	Size [Byte]	Priority (PCP)	Transmission Interval
Non-Scheduled Express 1	100	7	1 ms
Non-Scheduled Express 2	1530	7	1 ms
Non-Scheduled Preemptable 1	1530	1	Stream

Table 29: Traffics Configuration

The two express traffics have the same transmission interval and are simultaneously transmitted from their transmitters.

3.3.4.1 Scenario 1

Configuration

The activated communication features are listed belowTable 20.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps
Frame Forwarding Mechanism	Store&Forward

Table 30: Features Configuration

Results observation

The distribution of the end-to-end delay for a sample duration of 1 second (=1000 time-critical cycles) for network load of ~20% is illustrated below.

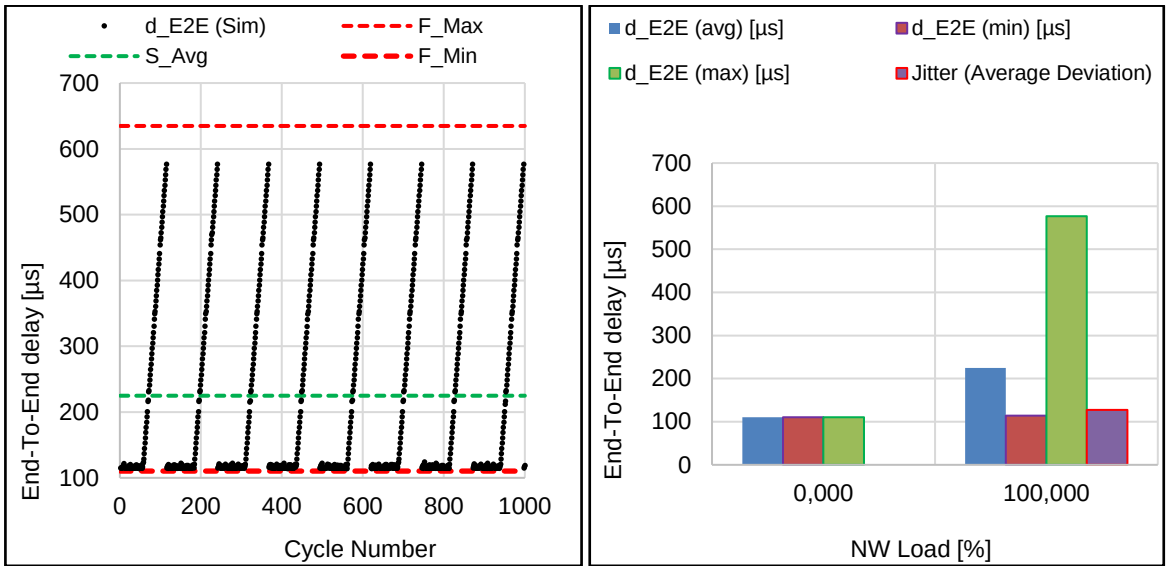


Figure 58:

- **(right) End-To-End delay distribution over 5 (Store&Forward, 100Mbps) switches using IEEE802.1Qbu under 20% Network load**
- **(left) Timing behavior over 5 (Store&Forward, 100Mbps) switches using IEEE802.1Qbu under different network loads**

Results Observation

Because of its interference with the “express traffic 2”, which has the same cycle time of one millisecond, the express traffic 1 revealed a different distribution of the end-to-end delay. Because of the small jitter, the cyclical interference faces a slight movement over time. The cycle time collisions / overlapping over time results into a periodic increasing end-to-end delay that remains to ~0 after a certain number of cycle collisions. Compared to scenario 1 of the interference with a preemptable traffic of the size 1530Byte (above 131Byte), the delay has been strongly increased from a maximum of ~139µs to ~634µs. This proves that the frame-preemption cannot be used as an efficient solution to reduce the delay variation. The range is of almost 524µs [110µs – 634µs] cannot be accepted for deterministic applications such as motion control.

3.3.4.2 Scenario 2

Configuration

The activated communication features are listed below.

Communication Features	Variants
Link-speed	Gigabit-Ethernet 1Gbps
Frame Forwarding Mechanism	Store&Forward

Table 31: Features Configuration

Results

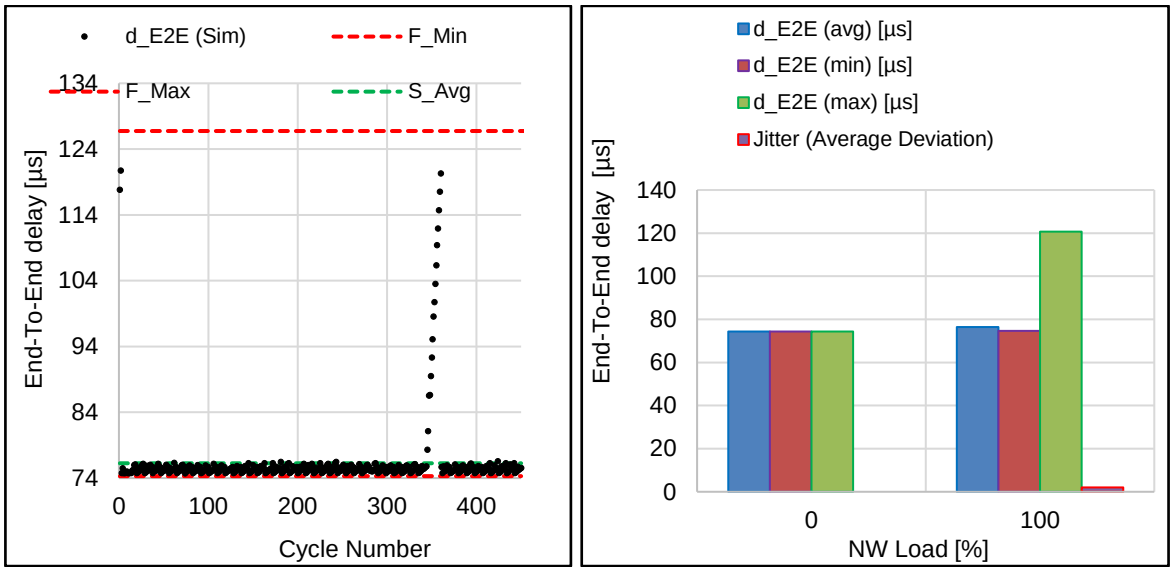


Figure 59:

- **(right) End-To-End delay distribution over 5 (Store&Forward, 1Gbps) switches using IEEE802.1Qbu under 20% Network load**
- **(left) Timing behavior over 5 (Store&Forward, 1Gbps) switches using IEEE802.1Q under different network loads**

Results Observation

Increasing the link-speed strongly reduces the interference delay and thus the overall end-to-end delay. The formal upper-bound worst-case delay is reduced by 80% from ~634µs to ~126µs.

3.3.4.3 Scenario 3

Configuration

The activated communication features are listed below.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps
Frame Forwarding Mechanism	Cut-Through

Table 32: Features Configuration

Results Observation

Supporting cut-through instead of store&forward slightly improves the timing behavior but has not the same impact as increasing the link-speed to 1Gbps.

Unlike the previous two scenarios, the simulation results of this scenario did not reach the pessimistic formal upper-bound worst-case delay. This is due the fact that an interference with the express-frame (non-preemptable) is not cyclically. Further the calculated

In case of an interference, the benefit of reducing the forwarding delay using cut-through is not feasbale.

While the end-to-end delay does not jitter for 0% network load, the coexistence of other traffics (express and preemptable) in the same path of the time-critical traffic increased the maximum measured end-to-end delay to 140µs. Nevertheless the average deviation remained weak by ~26µs.

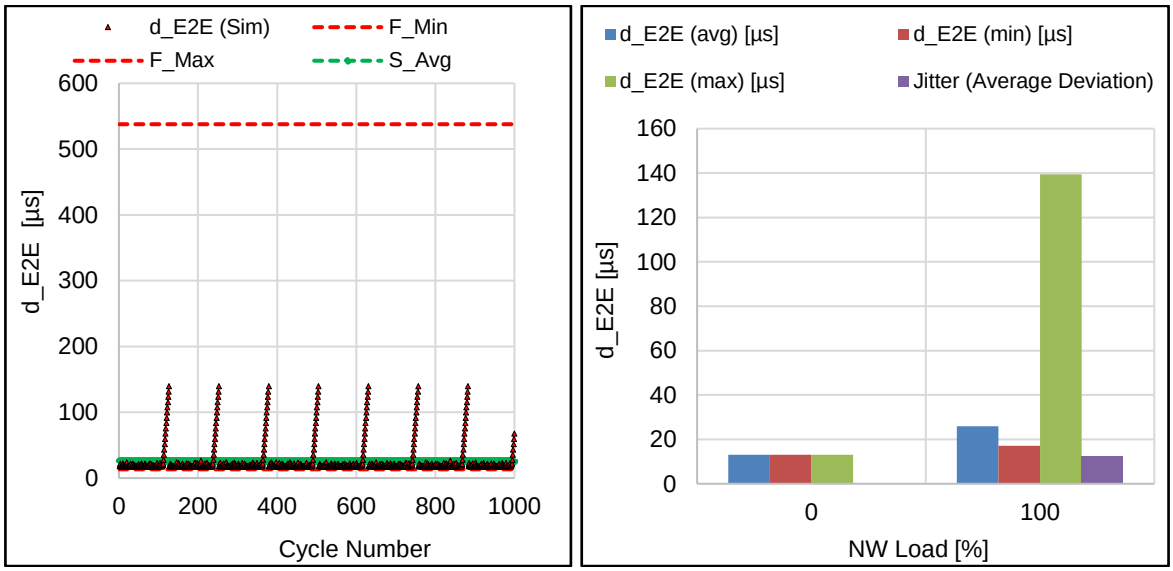


Figure 60:

- **(left) End-To-End delay distribution over 5 (Cut-Through, 100Mbps) switches using IEEE802.1Q under 20% Network load**
- **(right) Timing behavior over 5 (Cut-Through, 100Mbps) switches using IEEE802.1Q under different network loads**

3.3.4.4 Scenario 4

Configuration

Scenario 4 combines cut-through with Gigabit-Ethernet for the non-preemptive strict-priority algorithm.

Communication Features	Variants
Link-speed	Gigabit-Ethernet 1Gbps
Frame Forwarding Mechanism	Cut-Through

Table 33: Features Configuration

Simulation Results

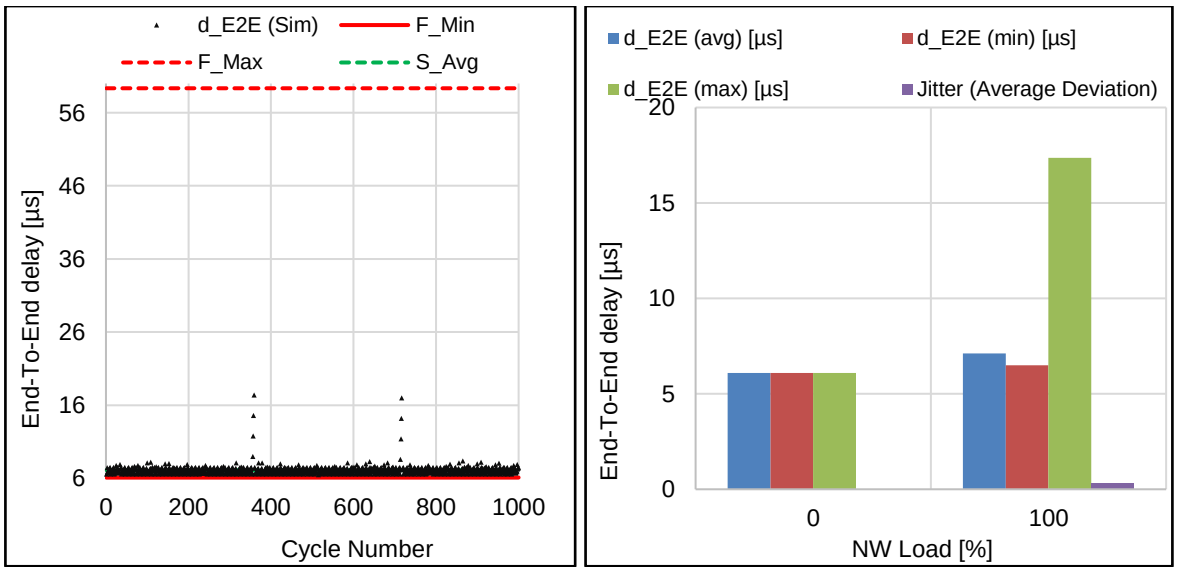


Figure 61:

- **(left) End-To-End delay distribution over 5 (Cut-Through, 1Gbps) switches using IEEE802.1Qbu under 20% Network load**
- **(right) Timing behavior over 5 (Cut-Through, 1Gbps) switches using IEEE802.1Qbu under different network loads**

Results Observation

Scenario 4 with cut-through and 1Gbps link-speed offers the best improvement of the timing behavior. The resulting jitter, min, max and average delay meets the requirements of highly deterministic industrial-applications for short line (less of devices in daisy-chain). The overall end-to-end delay is given by the single forwarding delays and the transmission duration.
The forwarding delay can be reduced by

- increasing the link-speed, which reduces the transmission duration of the background as well as the time-critical traffic
- supporting cut-through instead of store&forward to avoid the internal switch processing delay
- supporting frame-preemption to reduce the interference duration

3.3.4.5 Summary

The figure below illustrates the timing behavior of the preemptive-strict-priority algorithm, defined in IEEE802.1Qbu and IEEE802.3br, under four different scenarios. In each scenario covers a different combination of link-speed (100Mbps or 1Gbps) and forwarding-mechanism (store&forward or cut-through). The traffic shaper remains constant for all scenarios.

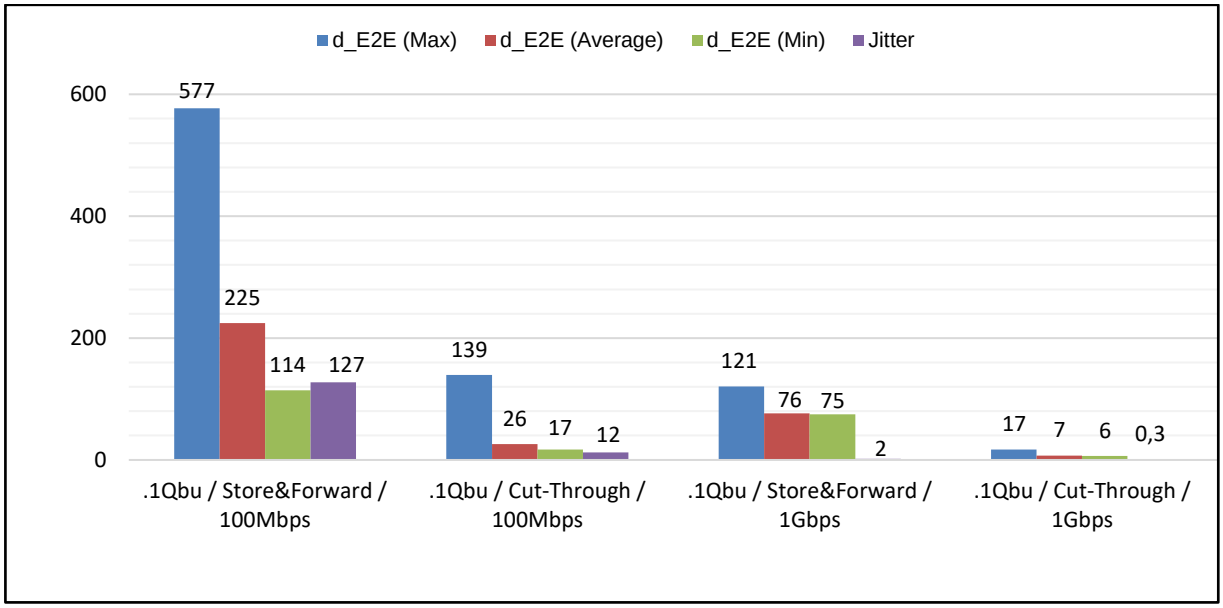


Figure 62: Timing improvement for IEEE802.1Qbu (Preemptive Strict-priority Algorithm)

The study proves that supporting cut-through instead of store&forward and increasing the link-speed from 100Mbps to 1Gbps offers the best strategy to reduce the end-to-end delay and to guarantee an ultra-low jitter below one microsecond.

3.4 IEEE802.1Qbv – Time-Aware Shaper

3.4.1 Formal Flow Analysis

The worst-case delay analysis of the previous scenarios proved that the interference of time-critical traffics with the background traffics all-over the path between the transmitter and the receiver has a high impact on the timing behavior with the focus on the jitter KPI. This can be improved through preemption, cut-through and Gigabit-Ethernet but is restricted to a network structure with few number of devices and limited amount of high priority background traffic. Increasing the number of devices might increase the amount of time-critical (express) traffics that cannot be preempted and would lead a bad timing behavior.
Therefor the interference must be completely avoided and not only reduced. This can be reached by scheduling the time-critical traffics, planning and protecting the transmission

time points all over the path from the transmitter to the receiver through certain mechanisms such as time-slots and guard-band window. This section shows how the delay variation range can be strongly reduced using the time-aware shaper defined in IEEE802.1Qbv. In order to reduce the overall delay, the transmission selection algorithm is combined with other features such as cut-through forwarding and Gigabit-Ethernet. The benefit of supporting cut-through in combination with traffic scheduling is shown below.

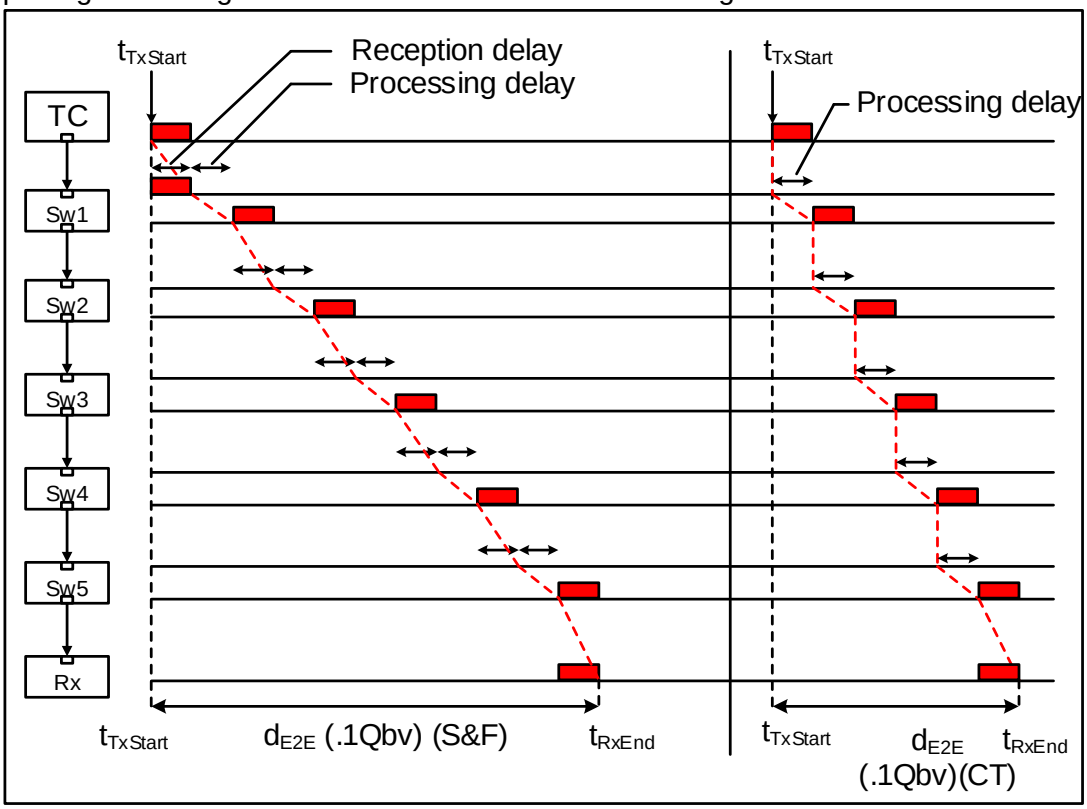


Figure 63: Flow Diagram of a time-critical frame using Time-Aware Shaper (IEEE802.1Qbv) with Store&Forward (left) vs. with Cut-Through (right)

Since the scheduled traffic is protected against any interference, increasing the network load has no effect on the timing behavior. This was proven by extensive simulation experiments. Therefore the results in the following scenarios show the timing behavior under 20% network load.

3.4.2 Scenario 1

Configuration

The activated communication features are listed below.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps (left) Gigabit-Ethernet 1Gbps (right)
Frame Forwarding Mechanism	Store&Forward

Table 34: Features Configuration

Results Observation

The distribution of the end-to-end delay for a sample duration of 1 second (=1000 time-critical cycles) for network load of ~20% is illustrated below.

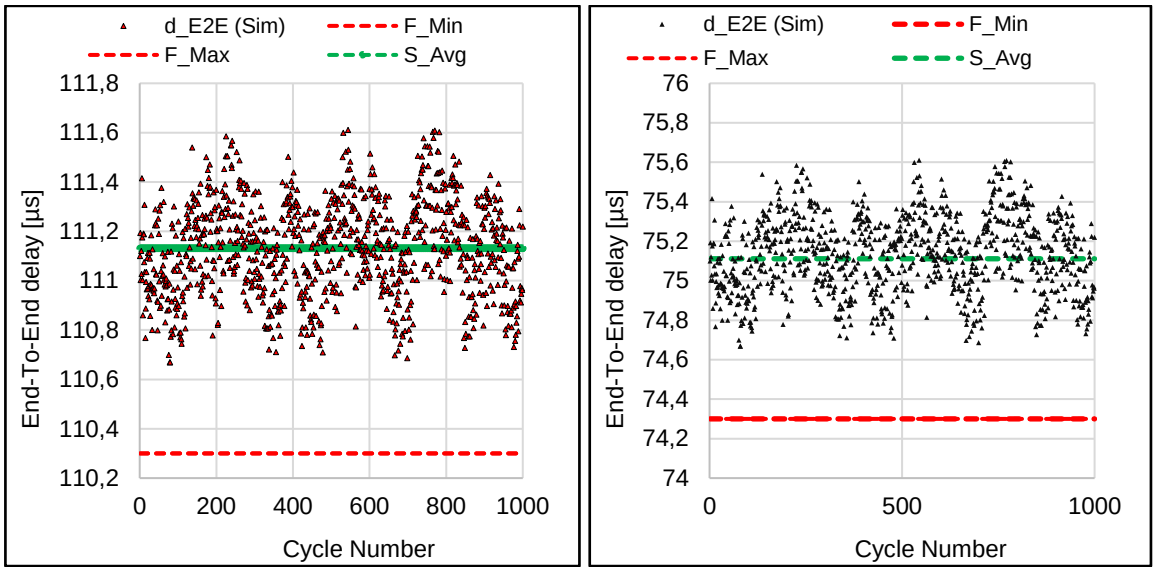


Figure 64:

- **(left) End-To-End delay distribution over 5 (Store&Forward, 100Mbps) switches using IEEE802.1Qbv under 20% Network load**
- **(right) End-To-End delay distribution over 5 (Store&Forward, 1Gbps) switches using IEEE802.1Qbv under 20% Network load**

Results Observation

For both link-speeds an ultra-low delay variation (jitter) below 1µs can be reached. This is due to the avoided interference. The transmission time-points of the scheduled time-critical frames are protected by the guard-band window. The resulting end-to-end delay variation is due to the slightly jittering clocks of the forwarding devices.

The resulting average end-to-end delay is ~111µs for 100Mbps and ~75µs for 1Gbps. Both results meet the hard timing requirements of the highly deterministic industrial applications.

3.4.3 Scenario 2

Configuration

The activated communication features are listed below.

Communication Features	Variants
Link-speed	Fast-Ethernet 100Mbps (left) Gigabit-Ethernet 1Gbps (right)
Frame Forwarding Mechanism	Cut-Through

Table 35: Features Configuration

Simulation Results

The distribution of the end-to-end delay for a sample duration of 1 second (=1000 time-critical cycles) for network load of ~20% is illustrated below.

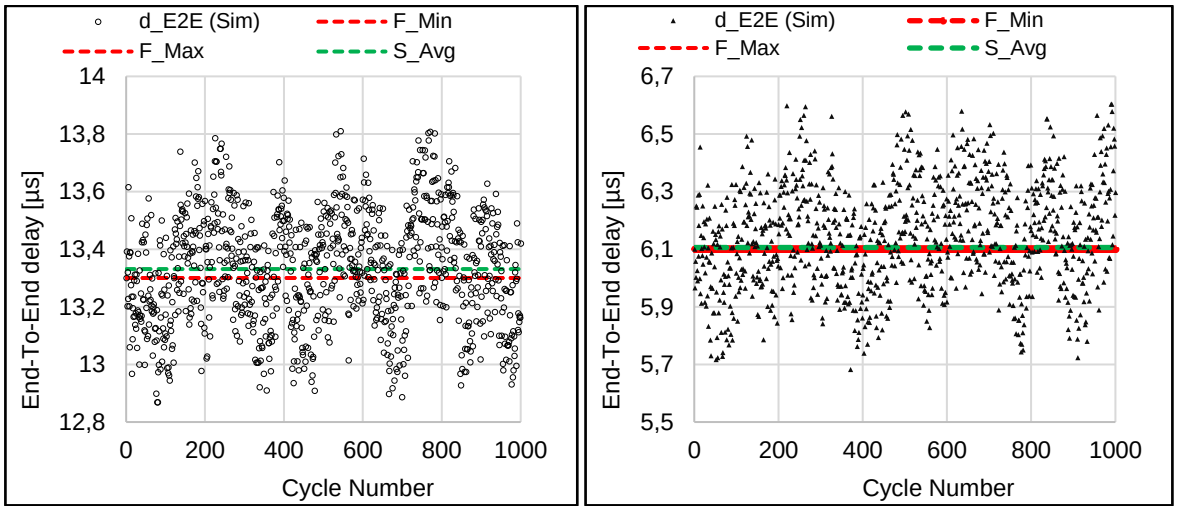


Figure 65:

- **(right) End-To-End delay distribution over 5 (Cut-Through, 100Mbps) switches using IEEE802.1Qbv under 20% Network load**
- **(left) End-To-End delay distribution over 5 (Cut-Through, 1Gbps) switches using IEEE802.1Qbv under 20% Network load**

Results Observation

Supporting cut-through has much higher impact on improving the timing behavior of the scheduled express traffic than increasing the link-speed to 1Gbps. Even with 100Mbps end-to-end delays of ~13µs can be reached.

3.4.4 Summary

The best timing behavior is given by the time-aware shaper in combination with different features, such as increasing the link-speed and supporting cut-through instead of store&forward. Unlike the frame-preemption, the scheduling approach guarantees an ultra-low jitter also in the presence of other time-critical traffics. The guard-band can protect the distributed transmission time-points all over the network between the transmitter and the receiver. However the planning complexity is the price for the high deterministic behavior. In order to well use the available network bandwidth, the time-slot need to be computed and shifted for each forwarding hop. This requires knowing or estimating the internal switch processing delay, the supported link-speed on each link and the remaining bandwidth on each forwarding port. The computation-effort becomes quickly complex especially if the bandwidth has been already allocated by other scheduled streams.

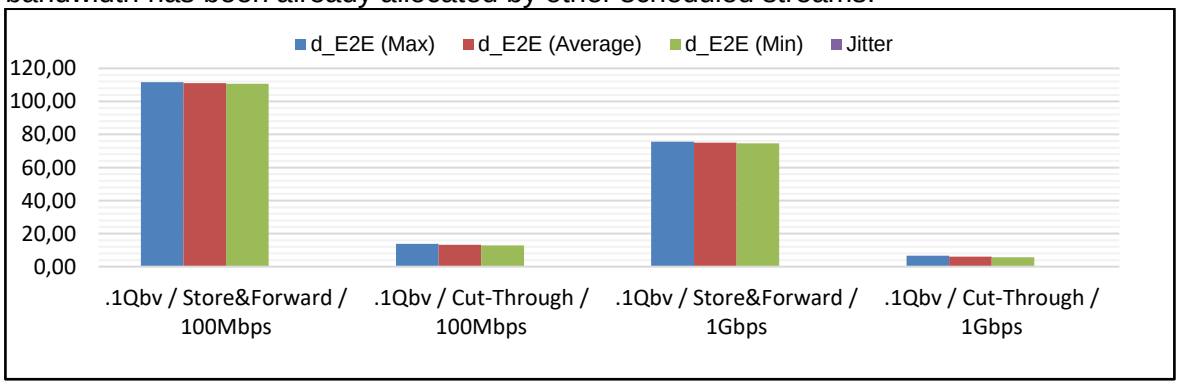


Figure 66: Timing improvement for IEEE802.1Qbv (Non-Preemptive Strict-priority Algorithm)

3.5 Evaluation of Transmission Selection Algorithms

The figure below illustrates the flow diagram of a time-critical frame transmitted from a transmitter to a receiver through five switches. Depending on the activated transmission selection algorithm on the egress ports of the switches, the frame has different timing

behavior that is shown below as three possible flow diagrams. Each flow is given by a different traffic shaper: non-preemptive strict-priority algorithm, preemptive strict-priority algorithm and time-aware shaper. It is assume that the link-speed and the forwarding mechanism remains constant.
The shortest end-to-end delay is given by the time-aware shaper.

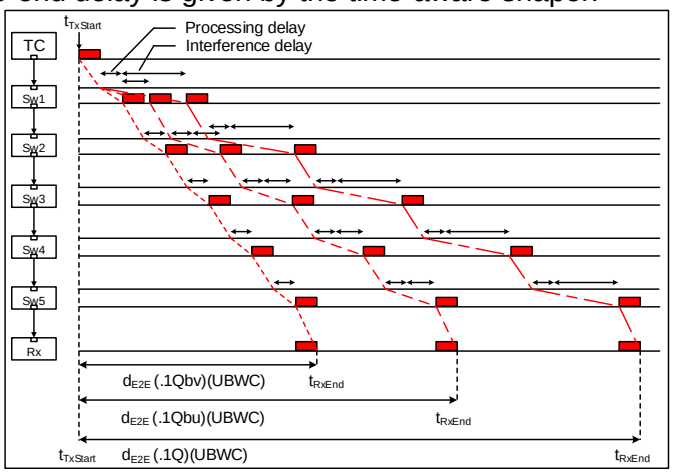
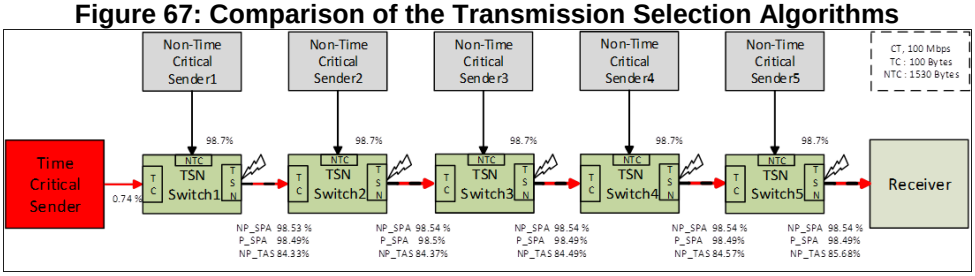


Figure 67: Comparison of the Transmission Selection Algorithms



Link-speed	Forwarding Mechanism	Reserved Time Slot	Channel Throughput [Mbps]		
			IEEE802.1Q	IEEE802.1Qbu	IEEE802.1Qbv
100Mbps	S&F	30	98,5	[98,48 to 98,5]	[85,37 to 93,07]
	Cut-Through		[98,53 to 98,54]	[98,48 to 98,5]	[84,33 to 85,68]
1Gbps	S&F	20	98,68	98,68	[95,54 to 95,55]
	Cut-Through		98,68	98,68	[95,54 to 95,55]

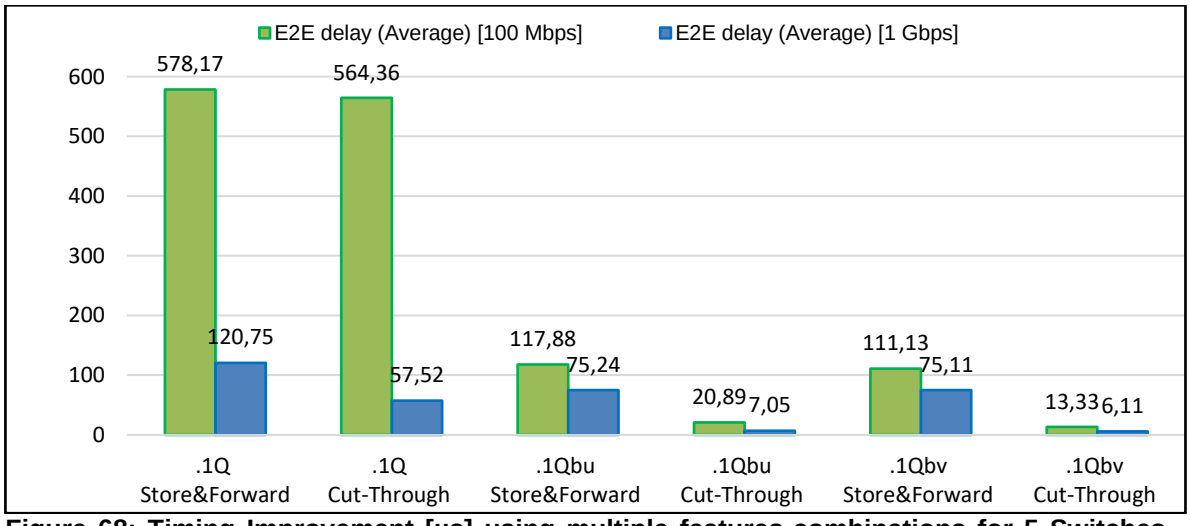


Figure 68: Timing Improvement [µs] using multiple features-combinations for 5 Switches, ~100% network load, 100Mbps or 1Gbps

Increasing the number of forwarding devices between the transmitter and the receiver is another important factor that has a huge impact on the end-to-end delay. Depending on the selected features, the frame might interfere with more traffics and need to be forwarded by

each device on the path. The results for 50 forwarding switches instead of 5 switches are shown below.

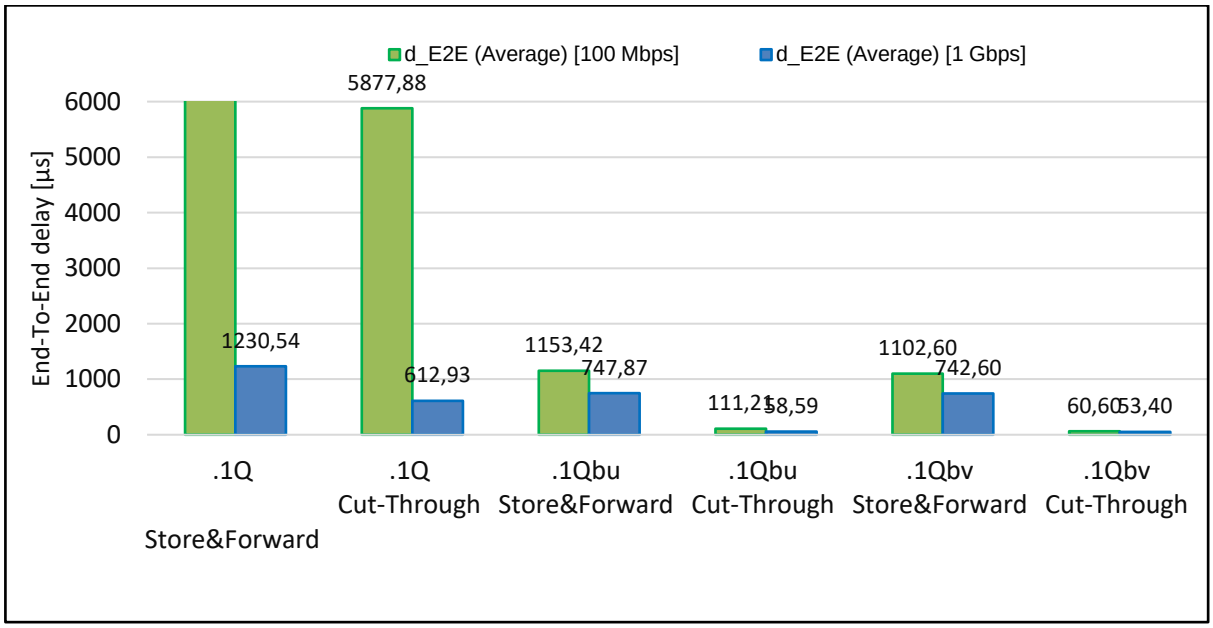


Figure 69: Timing Improvement [µs] using multiple features-combinations for 50 Switches, ~100% network load, 100Mbps or 1Gbps

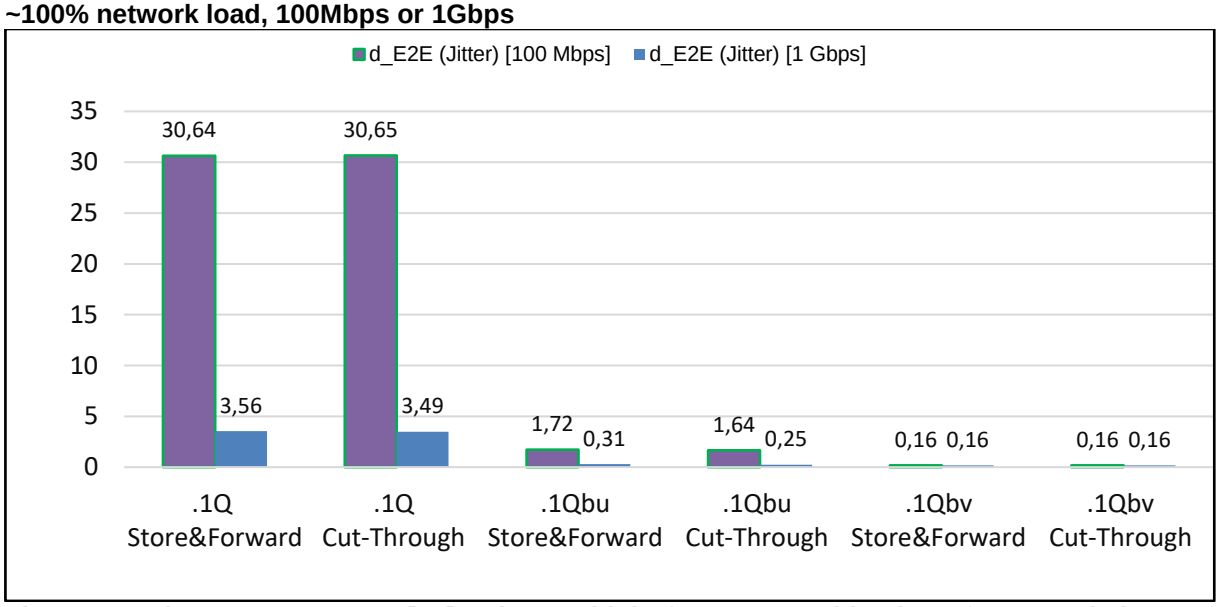


Figure 70: Jitter Improvement [µs] using multiple features-combinations for 50 switches, ~100% network load, 100Mbps or 1Gbps

3.6 Mapping TSN to the deterministic Industrial Applications

Table 36 summarizes all possible combinations of the communication features. Based on the typical communication requirements of the deterministic industrial applications (Table 37) the end-to-end delay, a key performance metrics (KPIs), has been computed.
The cells of the table are colored in

- "Red" if the delay requirements are not met (>2x more than the double)
- "Yellow" if the delay requirements are not met but are close (<2x)
- "Green" if the requirements are met

E.g. Machine tools applications require a cycle time below 0.5 millisecond (section 1.1.1.3) for 100 nodes connected in a line topology with 30Byte Payload per node. This requirement can be reached with the features combination (IEEE802.1Qbu – 100Mbps – Cut-Through).

Industrial Applications	Requirements & typical values of Key Performance Indicators					
Examples	Cycle Time [ms]	Sync Jitter [µs]	No Nodes	Payload/Node [Byte]	Distance [m]	Topology (-ies)
Monitoring	100	1	1000	300	1000	
Process Automation	10-100	1000	300	1500	100	
Machine Tool	0,5	0,25	50	30	7	Line - Ring - Baum
Packaging Machines	1	1	100	50	5	Line - Ring - Baum
Printing Machines	4	0,25	200	50	25	Line - Ring - Baum

Table 37: Requirements and Configuration of the industrial applications

	IEEE802.1Q (Strict Priority Algorithm)				IEEE802.1Qbu (Frame-Preemption)				IEEE802.1Qbv (Time-Aware Shaper)			
Industrial Applicatior	Store&Forward		Cut-Through		Store&Forward		Cut-Through		Store&Forward		Cut-Through	
Examples	100 Mbps	1 Gbps	100 Mbps	1 Gbps	100 Mbps	1 Gbps	100 Mbps	1 Gbps	100 Mbps	1 Gbps	100 Mbps	1 Gbps
Monitoring												
Process Automation												
Factory Automation												
Machine Tool												
Packaging Machines												
Printing Machines												

Table 36: Mapping TSN to the deterministic industrial applications

3.7 Summary

The formal and simulation results of the covered communication scenarios proved that the new TSN transmission selection algorithms: *Non-preemptive strict-priority algorithm* and *time-aware shaper*

- Are not needed for the industrial applications *condition monitoring* and *process automation*, requiring cycle times ~10-100ms (section 1.1.1.3). The soft-real-time requirements of these applications can be met with the established standard features: IEEE802.1Q, 100Mbps link-speed and the Store&Forward mechanism.
- are not sufficient to meet the hard-real-time requirements (mainly short cycle times) of the highly deterministic applications of the factory automation: machine tools, packaging- and printing-machines once combined with Store&Forward and 100Mbps or 1Gbps.

Further it was proven that

- Supporting cut-through forwarding mechanism is mandatory for industrial applications with line topologies and hard-real-time requirements. Cut-through has a higher impact on reducing the end-to-end delay than increasing the link-speed.
- Certain hard-real-time applications using Industrial-Protocols with frame-based synchronization, e.g. Sercos, it is necessary to avoid the interference between the time-critical control data and the remaining traffics. Therefore cut-through and the scheduling mechanisms are mandatory.
- IEEE802.1Qbu in combination with Gigabit-Ethernet can reach close timing performances as IEEE802.1Qbv as long as the size and amount of the coexisting epress (non-preemptable) traffics remains very low. The coexistence of multiple express traffics increases the cycle time, end-to-end delay and the jitter. This big restriction is feasible only in closed networks.
- The best technical combination for hard-real-time communication is Gigabit, IEEE802.1Qbv and Cut-Through. However this is not necessary for most industrial applications.

4 Industrial-Ethernet Protocols over TSN

4.1 Introduction

TSN specifies a set of standards and mechanisms to improve the timing behavior of Ethernet. Integrating TSN in the factory automation requires the definition of communication scenarios that combine the new TSN mechanisms with the established Industrial-Ethernet protocols.
To analyze the timing behavior, the timing behavior of these communication solutions need to be investigated first. Four Industrial-Ethernet protocols are selected to cover all possible performance- and application-areas they are used for.

- Ethernet/IP from real-time class 1
- Profinet RT from real-time class 2
- Profinet IRT and Sercos III from real-time class 3

For a fair comparison of the timing performances of the protocols, a common network- and traffic-configuration is used for each scenario.

- link-speed – 100Mbps or 1Gbps
- payload per device – 36Byte
- number of devices – 50
- topology – Line
- Network load – 30%
- Forwarding-mechanism and –delay – protocol dependent.
- Size and number of required frames – protocol dependent

The main characteristics of the selected Industrial-Ethernet protocols are listed below.

Protocol	Frame-Structure	Frame Overhead [Byte]	Forwarding Mechanism	Transmission Selection Algorithm	Min. Cycle Time[10] Model
Ethernet/IP	Individual	82	Store&Forward	IEEE802.1Q	1[11]
Profinet RT	Individual	36	Store&Forward	IEEE802.1Q	1
Profinet IRT	Individual	32	Cut-Through	Scheduled	1
Sercos III	Summation	44 - 56	Cut-Through	Scheduled	2

Table 38: Overview and comparison of the selected Industrial-Ethernet protocols

To analyze the improvement of their timing performances, the link-speed is increased from 100Mbps to 1Gbps and the protocols are extended with the deterministic TSN features: frame-preemption (IEEE802.1Qbu, IEEE802.3br) and / or frame-scheduling (IEEE802.1Qbv).

4.1.1 Network Description

For a fair comparison of the timing behavior between the different Industrial-Ethernet protocols a “common” network structure of a line topology has been selected. The network consists of a master device (PLC) connected in a line topology to 50 slave devices (Figure 71). The master device generates and transmits cyclically time-critical control data to the slave devices (downstream communication) that simultaneously respond back (upstream communication). In case of Sercos III, the master is the only device that generates time-critical control-data traffic. The Sercos traffic is transmitted throughout the slave-devices and is received back at the master port.

[10] See [25] and section 2.1.2.1Minimum Cycle Time
[11] For EtherNet/IP the following charachteristics are considered: single-port devices and line topology Store&Forward mechanism (not DLR with Cut-Through).

The slave devices are two ports switching end-points. In real factories, the devices contain integrated three-port switches with one port connected internally to the device. In case of protocols using the individual frame approach, the last device does not forward the frame. In case of summation frame protocol, the last device is a loop-back node that transmits all received control-data frames at the same port to the previous device.
36 Bytes of payload for the downlink and the uplink communications are exchanged cyclically. Each protocol is encapsulating the payload-data in individual- or summation frames. This results into different frame sizes.

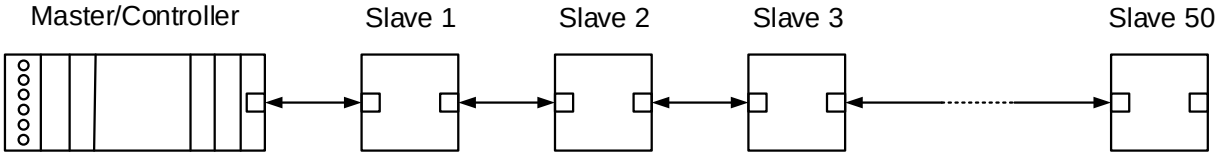


Figure 71: Common Network Structure – Line Topology

This chapter is composed of multiple sections. Each analyzes the timing performance of a single industrial-Ethernet protocol. The following scenarios are recognized:

- Increasing the link-speed from 100Mbps to 1Gbps
- Supporting frame-preemption(IEEE802.1Qbu – IEEE802.3br)
  - with 100Mbps
  - with 1Gbps
- Supporting frame-scheduling (IEEE802.1Qbv)
  - with 100Mbps
  - with 1Gbps

The line topology, number of devices (50), network load of background traffics (30%) and the payload per device (36Byte) are common for all scenarios. The background traffic is transmitted from each slave in both directions: to the master and to the slave. This is required to cover the timing behavior for the uplink and downlink communication.

Computing the cut-through forwarding delay
Based on the activated transmission selection algorithm, more or less bytes need to be read to identify the time-critical frames for a cut-through forwarding. For a non-preemptive strict-priority algorithm as well as the time-aware shaper, a time-critical frame is identified through its priority field (PCP) within the VLAN-tag. This results into reading 24Bytes before taking the decision of activating the cut-through forwarding. For the preemptive strict-priority algorithm a time-critical (express) frame can be identified through the modified preamble and start of frame delimiter. In order to forward it through the designated egress port, the destination address must be read. This results into reading 14Bytes.
The cut-through forwarding delay is composed of the delays caused by the reception and transmission PHYs, set to 300 nanosecond typical value each, as well as the duration of time of receiving the reading bytes.
$d_{ct} = d_{RxPHY} + d_{ReadingBytes} + d_{TxPHY}$
Another factor that affects the cut-through forwarding delay is the link-speed. The delays are illustrated below.

Transmission selection algorithm	Reading Bytes	100Mbps	1Gbps
Non-Preemptive Strict-Priority Algorithm	24	2,52µs	~ 0,8µs
Preemptive Strict-Priority Algorithm	14	1,72µs	0,71µs
Non-/ Preemptive Time-Aware Shaper	24	2,52µs	~ 0,8µs

Table 39: Computing the cut-through forwarding delay

A comparison of the cut-through forwarding delays of Sercos III and Profinet IRT are illustrated below.

Industrial-Ethernet Protocol	100Mbps	1Gbps
Sercos III (*)[12]	1µs	1µs
Profinet IRT [9]	3µs [9]	0,6µs [9]

Table 40: cut-through forwarding delays for Sercos III and Profinet IRT (sources: Jasperneite und Schumacher IE Improvement with Gigabit + Sercos specifications);

4.1.2 Contributions

- Analyzing and comparing the timing performances of the selected Industrial-Ethernet protocols.
- Analyzing the timing performances of the industrial-Ethernet protocols adding
  - Higher link-speed – 1Gbps
  - Frame-preemption – IEEE802.1Qbu and IEEE802.3br
  - Frame scheduling – IEEE802.1Qbv

4.2 Ethernet/IP

4.2.1 Formal Timing Analysis

Ethernet/IP supports store&forward and the non-preemptive strict-priority algorithm to forward the cyclically generated time-critical control-data frames [74]. Each slave receives and generates its own “individual” frame for the uplink stream. In case of Device Level Ring (DLR) Ethernet/IP supports cut-through forwarding. This is not covered in this investigation. The cycle time is equal to the maximum duration from transmitting the first bit of the first frame until receiving the first bit at the furthest slave device (downlink stream) or from transmitting the first bit of the last frame from the furthest slave device until receiving the first bit of that frame at the master device (upstream). In order to reach a minimum cycle time, the frame to the last device is sent first.
Because of the non-preemptive strict-priority algorithm, it is not possible to avoid the interferences with the background traffics. The best-case scenario is given by zero interferences and the worst-case scenario by n interferences, where n is the number of slave-devices minus one. The space-time diagram below illustrates the best- and worst-case scenarios for the down- and upstream communications of the Ethernet/IP traffic flow.

[12] For Sercos there are no published measurements or studies for 1Gbps. Since mainly the reception- and transmission-latencies, caused by the physical layer, are reduced, the thesis considered the forwarding delay the same for 100Mbps and 1Gbps.

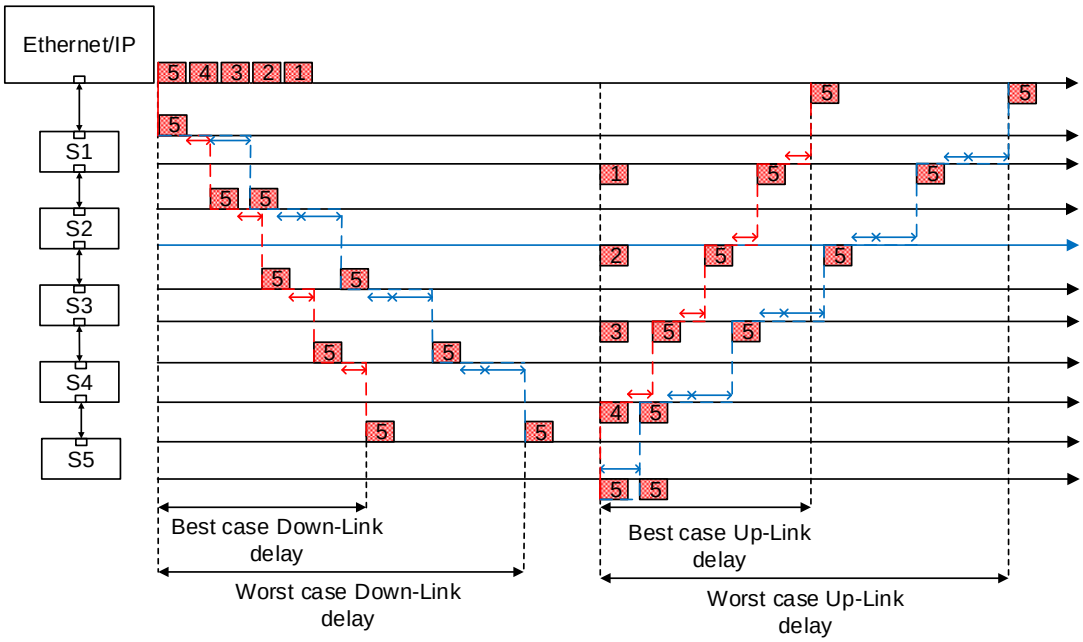


Figure 72: Worst case interference scenario in Ethernet/IP during downlink communication Minimum cycle time computation (In red box: time-critical frame; red arrow: switch internal processing delay; blue arrow: interference delay)

Note: A higher timing performance can be reached using star topology for EtherNet/IP. But because of its importance for the industrial automation this thesis and this [9] publication focuse on line topology.

If the transmission duration of a single individual frame d_{Tx} is bigger or equal to the medium delay d_{medium} and the forwarding duration d_{fwd} than the minimum cycle time is equal to the maximum of the duration of the uplink or downlink communication streams.
$d_{cycleEthernet/IP} = \max(d_{dl-Ethernet/IP}\ , d_{ul-Ethernet/IP})$
This calculation is given by the case of three cycle model (Ethernet/IP specifications), in which the uplink stream is parallelly transmitted to the downlink stream.
The downlink and uplink are respectively given by the duration of time to transmit the frame from the controller to the last device or from the last device to the controller.
$d_{dl-Ethernet/IP} = d_{Tx} + d_{fwd} * (n_{devices} - 1) + (d_{medium} * n_{devices})$
$d_{ul-Ethernet/IP} = d_{Tx} + d_{fwd} * (n_{devices} - 1) + (d_{medium} * n_{devices})$
A single forwarding delay is given by the processing, interference and transmission delays.
$d_{fwd} = d_{Px} + d_{Interference} + d_{Tx}$

Common traffic configurations

The following traffic configuration is common for all scenarios. It is assumed that all Ethernet/IP frames, independent of the stream direction, have the same size 118Byte.

Protocol Overhead [Byte]	Payload / Dev [Byte]	Frame Size [Byte]	No of frames	Cycle Time [ms]	Throughput [Mbps]
82	36	118	50	10	4,72

Table 41: Traffic Configuration for Ethernet/IP

4.2.2 Timing Analysis with Gigabit-Ethernet

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[13] [%]
100Mbps	619,25	6787,25	2371,58	3438,42	3876,08	216,36	4,72
1Gbps	194,45	811,25	269,74	399,84	518,13	27,97	0,472

Table 42: formal and simulation results for Ethernet/IP using Non-Preemptive Strict-Priority Algorithm for 100Mbps and 1Gbps

The simulation results for a sample duration of 9 seconds for 100Mbps and 1Gbps link-speeds are shown in Figure 73.

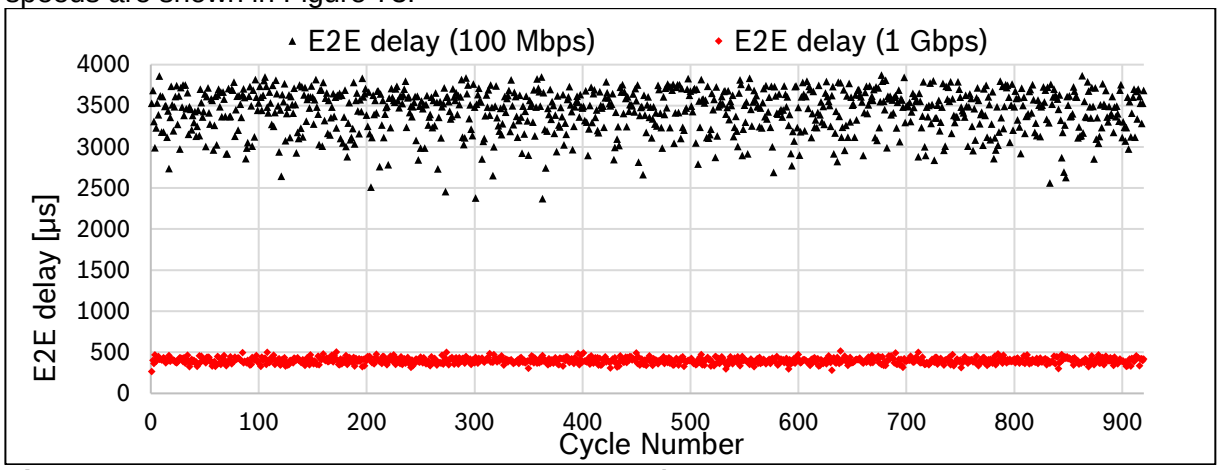


Figure 73: Minimum Cycle Time [µs] of Ethernet/IP using Non-Preemptive Strict-priority Algorithm – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load

Note: *Practical meaning: The asynchronous data exchange between the slaves with each other and with the master is the reason for considering the interference between the time-critical control data and the background traffics.*

Results Observation

The End-To-End delay distribution of the Ethernet/IP frames for 100Mbps (black) using non-preemptive strict-priority algorithm shows a huge jitter (>>1µs). The range is almost 1.5 millisecond [2371,58 µs, 3876,0 µs]. The delay is highly non-deterministic and is not suited for applications belonging to the third real-time class. Increasing the link-speed from 100Mbps to 1Gbps results into a strong reduction of the End-To-End Delay and the jitter. Since Ethernet/IP uses store&forward, the end-to-end delay for a line topology goes easily above one millisecond even for less number of cascaded slave devices (<10).

4.2.3 Timing Analysis with Frame-Preemption (IEEE802.1Qbu)

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[14] [%]
100Mbps	619,25	1189,25	673,09	705,30	730,50	6,61	4,72
1Gbps	194,45	251,45	194,74	195,80	198,88	0,48	0,472

Table 43: formal and simulation results for Ethernet/IP using Preemptive Strict-Priority Algorithm for 100Mbps and 1Gbps

[13] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)
[14] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)

The simulation results for a sample duration of 9 seconds for 100Mbps and 1Gbps link-speeds are shown in Figure 74.

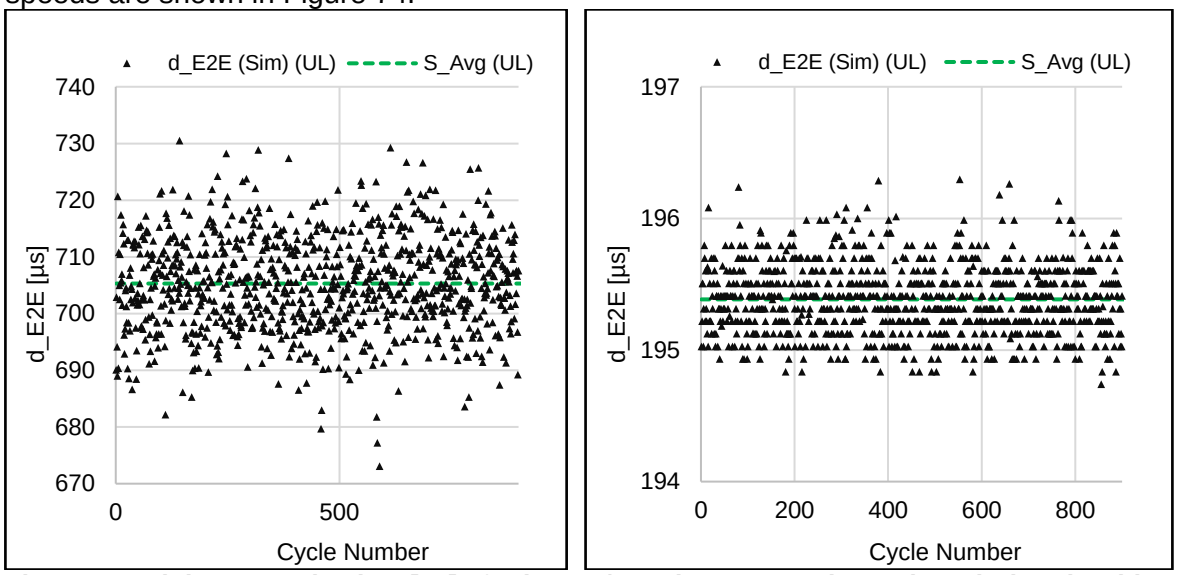


Figure 74: Minimum Cycle Time [µs] of Ethernet/IP using Preemptive Strict-priority Algorithm – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation

This simulated scenario activates the frame-preemption for the strict-priority transmission selection algorithm. A significant timing performance improvement has been observed with the simulation for both link-speeds 100Mbps and 1Gbps. The end-to-end delay as well as the jitter are strongly reduced compared to the non-preemptive scenario. The distribution range of the end-to-end delay is reduced from 1.5 millisecond (no-preemption) to ~57 microsecond (preemption) [673.09 µs, 730.50 µs] for 100Mbps. And the jitter (average delay deviation) from 216.36 µs (no-preemption) to 6.61µs (preemption). A further reduction of the distribution range of the end-to-end delay is possible by increasing the link-speed from 100Mbps to 1Gbps. The range size is reduced to almost four microseconds [194.75 µs, 198.8 µs], which results into an ultra-low jitter (average deviation) below one microsecond. The results prove that combining the frame-preemption with 1Gbps link-speed (with no express background traffic) is sufficient to meet the requirements of the highly deterministic industrial applications that are reached today with the scheduling approach of protocols from the third real-time class.

4.2.4 Timing Analysis with Frame-Scheduling (IEEE802.1Qbv)

The formal and simulation results for the cycle time, jitter and resulting bandwidth for the Ethernet/IP traffic using IEEE802.1Qbv are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[15] [%]
100Mbps	619,25		619,32	619,65	621,57	0,38	35,00
1Gbps	194,45		194,29	194,85	196,77	0,38	35,00

Table 44: formal and simulation results for Ethernet/IP using Non-Preemptive Time-Aware Shaper for 100Mbps and 1Gbps

The figure below illustrates the minimum cycle time of Ethernet/IP using the non-preemptive time-aware shaper over 50 slave devices and under 30% network load. The scheduled time-slots have the same size but not the starting time-point and thus are shifted from a forwarding hop to the other to better use the available network bandwidth.

[15] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)

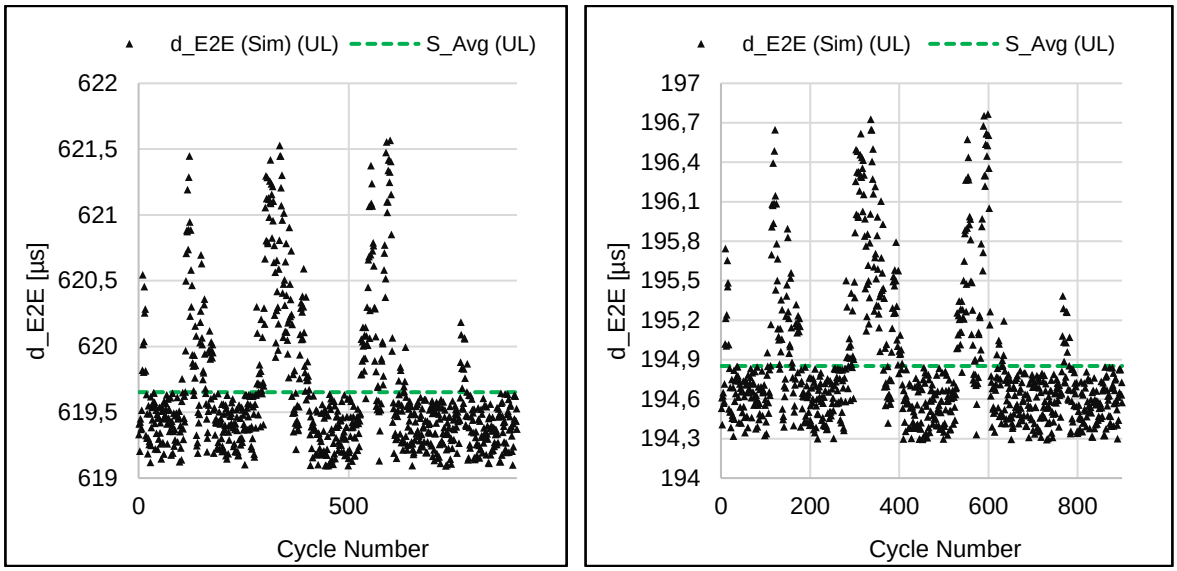


Figure 75: Minimum Cycle Time [µs] of Ethernet/IP using Non-Preemptive Time-Aware Shaper – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation

The TSN time-aware shaper offers the best performance results for the Ethernet/IP time-critical traffic. For 100Mbps and 1Gbps link-speed, the average end-to-end delay variation (jitter) is << 1 µs (~380 nanoseconds). Scheduling the traffics makes the Ethernet/IP protocol efficient enough to be classified in the third real-time class. The high time-synchronization accuracy can be reached with the IEEE802.1AS or IEEE1588 as it is used today.

4.2.5 Summary

The figure below summarizes the network timing performances of Ethernet/IP today and compares it with the other scenarios discussed above. The results shows a strong performance improvement. All scenarios guarantee a cycle time below one millisecond, which is a reduction by a factor of four.

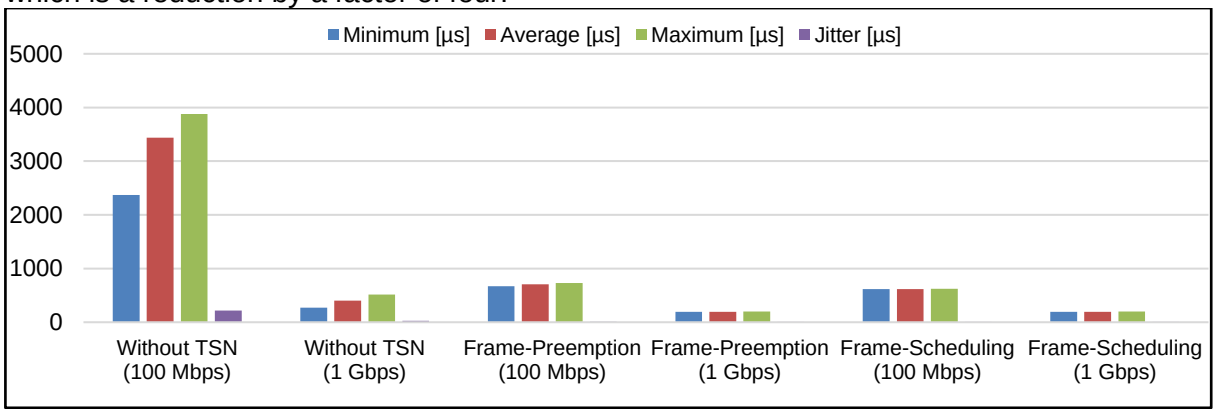


Figure 76: Timing performance comparison of Ethernet/IP using different communication features

4.3 Profinet RT

4.3.1 Formal Timing Analysis

Similar to Ethernet/IP, Profinet RT uses store&forward and non-preemptive strict-priority algorithm to forward the cyclically generated time-critical control-data frames. The frames have less overhead, since the payload is not encapsulated in a UDP/IP (28Byte) and Ethernet/IP encapsulation headers (24Byte) (section 2.1.3.1). The space-time diagram of

the traffic flow as well as the computation of the minimum cycle time is similar to Ethernet/IP (using line topology-based model and Store&Forward).

Minimum cycle time computation

$d_{cycleProfinetRT} = \max(d_{dl-ProfinetRT}, d_{ul-ProfinetRT})$

The downlink and uplink are respectively given by the frame transmission duration from the controller to the last device or from the last device to the controller.

$d_{dl-ProfinetRT} = d_{Tx} + d_{fwd} * (n_{devices} - 1) + (d_{medium} * n_{devices})$

$d_{ul-ProfinetRT} = d_{Tx} + d_{fwd} * (n_{devices} - 1) + (d_{medium} * n_{devices})$

A single forwarding delay is given by the processing, interference and transmission delays.

$d_{fwd} = d_{Px} + d_{Interference} + d_{Tx}$

Common traffic configurations

The following traffic configuration is common for all scenarios. It is assumed that all Profinet RT frames, independent of the stream direction, have the same size 72Byte. Compared the Ethernet/IP, Profinet RT saves 46Bytes overhead per frame by bypassing the UDP/IP stack.

Protocol Overhead [Byte]	Payload / Dev [Byte]	Frame Size [Byte]	No of frames	Cycle Time [ms]	Throughput [Mbps]
36	36	72	50	5	5,76

Table 45: Traffic Configuration for Profinet RT

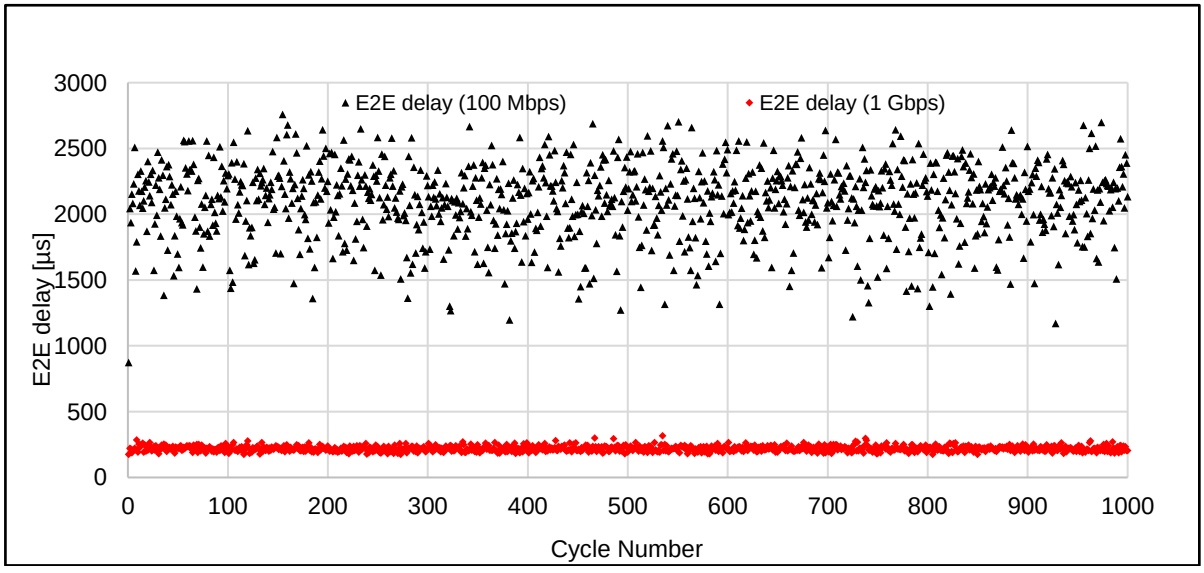
4.3.2 Timing Analysis with Gigabit-Ethernet

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[16] [%]
100Mbps	435,25	6603,25	874,82	2104,08	2761,10	208,84	5,76
1Gbps	176,05	792,85	176,05	223,43	316,57	16,82	0,576

Table 46: Summary of the formal and simulation results for Profinet RT using non-preemptive strict-priority algorithm for 100Mbps and 1Gbps

The simulation results for a sample duration of 9 seconds for 100Mbps and 1Gbps link-speeds are shown below.



[16] The total bandwidth utilization includes the throughput of all traffics (Profinet RT and background)

Figure 77: Minimum Cycle Time of Profinet RT [µs] using Non-Preemptive Strict-priority Algorithm – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load

Results Observation
The figure above shows highly non-deterministic latency of the Profinet RT frames caused by the interference with the background traffics. An improvement of the timing performance can be observed by increasing the link-speed. The jitter is reduced from 208µs for 100Mbps to 16.82µs for 1Gbps showing a reduction factor of more than factor 10.
However Porifnet RT does not require highly deterministic latency performances. For these particular applications, e.g. motion-control, the Profinet community uses Profinet IRT (Isochronous Real-Time) Protocol.

4.3.3 Timing Analysis with Frame-Preemption (IEEE802.1Qbu)

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[17] [%]
100Mbps	435,25	1005,25	442,88	464,79	496,51	6,45	5,76
1Gbps	176,05	233,05	181,44	184,06	188,639	0,86	0,576

Table 47: Summary of the formal and simulation results for Profinet RT using preemptive strict-priority algorithm for 100Mbps and 1Gbps

The simulation results for a sample duration of 9 seconds for 100Mbps and 1Gbps link-speeds are shown in **Figure 78**.

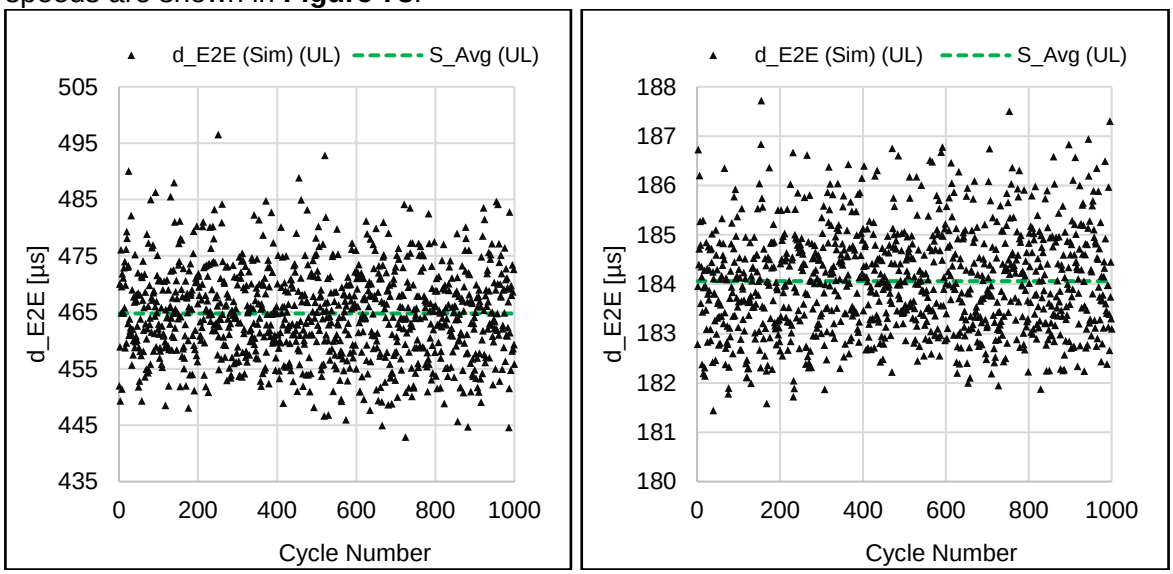


Figure 78: Minimum Cycle Time [µs] of Profinet RT using Preemptive Strict-Priority Algorithm – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation
Using the preemptive strict-priority algorithm (IEEE 802.1Qbu) as a transmission selection algorithm reduces the delay variation for 100Mbps and 1Gbps. An average jitter below one microsecond can only be reached with 1Gbps link-speed. The graph shows a difference of about five microseconds between the minimum and maximum end-to-end delay.

[17] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)

4.3.4 Timing Analysis with Frame-Scheduling (IEEE802.1Qbv)

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[18] [%]
100Mbps	435,25		436,07	436,36	436,63	0,32	35,01
1Gbps	176,05		176,87	177,16	177,43	0,13	3,5

Table 48: Summary of the formal and simulation results for Profinet RT time-aware shaper for 100Mbps and 1Gbps

Similar to Ethernet/IP, supporting the frame-scheduling approach for the Profinet RT traffic reduces the cycle time to the microsecond range and the average jitter to the nanosecond range.
The figures below compare the impact of increasing the link-speed on the timing performances on Profinet RT.

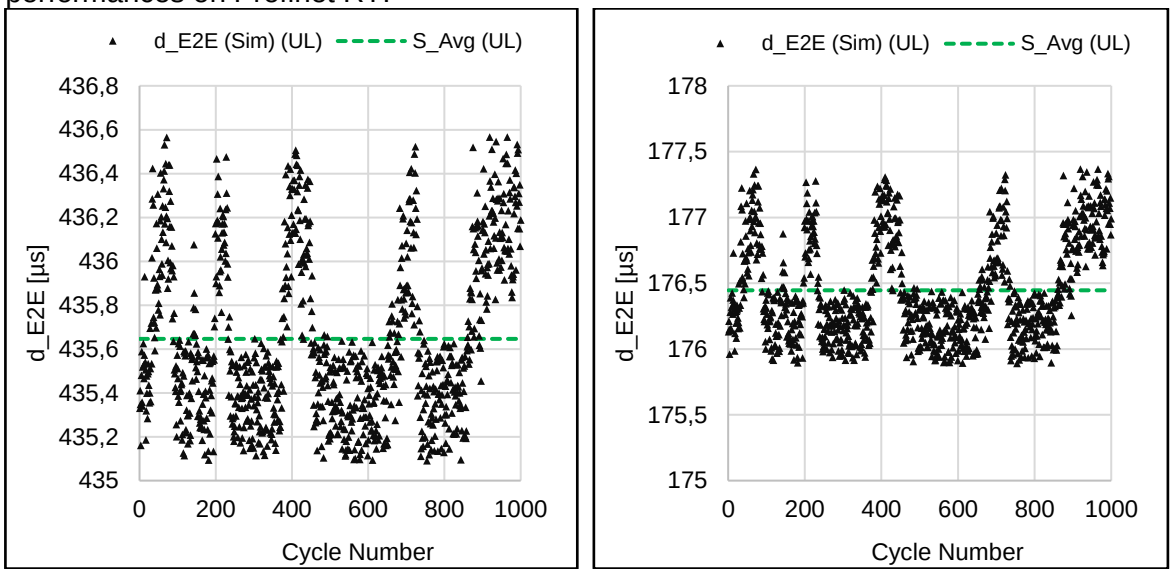


Figure 79: Minimum Cycle Time [µs] of Profinet RT using Non-Preemptive Time-Aware Shaper – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

4.3.5 Summary

All scenarios described in this section are summarized in the figure below. It is proven that combining the frame-preemption approach with gigabit promises similar timing performances to the frame-scheduling approach. However the results strongly affected by the moderate traffic configuration. In real networks cyclic and sporadic express traffics that cannot be preempted might coexist in the network and share the resources with the control-data traffics.
This would results into higher delays and jitter.

[18] The total bandwidth utilization includes the throughput of all traffics (Profinet RT and background)

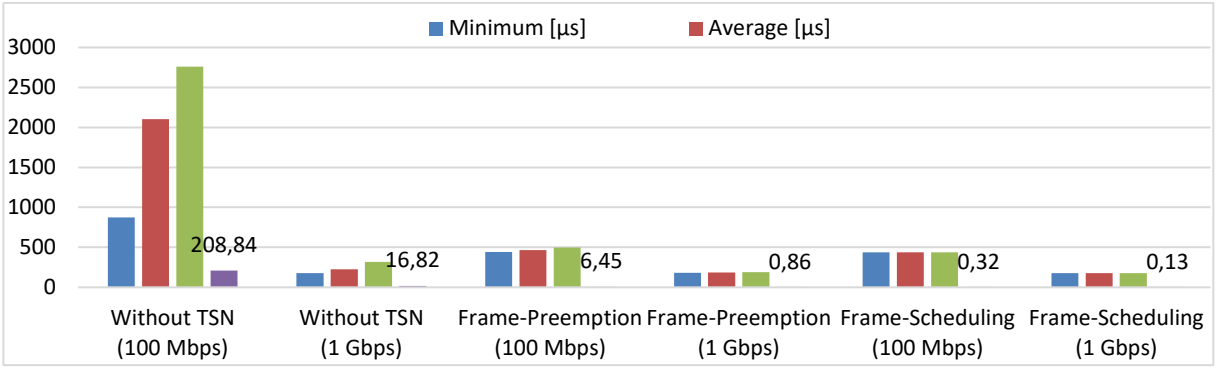


Figure 80: Timing performance comparison of Profinet RT using different communication features

4.4 Profinet IRT

4.4.1 Formal Timing Analysis

A Profinet IRT system uses cut-through forwarding mechanism to forward the cyclically generated isochronous control-data frames. Each slave receives and generates its own "individual" frame.
Similarly to the previous two protocols Ethernet/IP and Profinet RT, the cycle time of Profinet IRT is calculated as the maximum duration of downlink and uplink streams. Since the time-critical frames are scheduled, the interference with the background traffics can be avoided for both stream directions. This results into an ultra-low jitter and a highly deterministic timing behavior.

Minimum cycle time computation
It is assumed that the same payload size is used for both stream directions. Therefore the down- and up-stream durations are equal.
$d_{cycleProfinetIRT} = d_{dl-ProfinetIRT} = d_{ul-ProfinetIRT}$
The downlink and uplink are respectively given by the duration of time to transmit the frame from the controller to the last device or from the last device to the controller.
$d_{dl-ProfinetIRT} = d_{ul-ProfinetIRT} = d_{Tx} + d_{ct} * (n_{devices} - 1) + (d_{medium} * n_{devices})$
Profinet IRT capable devices support cut-through to forward the time-critical isochronous traffic.

Common traffic configurations
The following traffic configuration is common for the all scenarios. It is assumed that all frames, independent of the stream direction (upstream and downstream), have the same size. For 36Byte payload per device and 36Byte total overhead, the minimum Ethernet frame size of 72Byte (including 8 byte preamble) cannot be reached, therefore four padding Bytes are added.

Protocol Overhead [Byte]	Payload / Dev [Byte]	Frame Size [Byte]	No of frames	Cycle Time [ms]	Throughput [Mbps]
32	36	72	50	1	28,8

Table 49: Traffic Configuration for Profinet IRT

4.4.2 Timing Analysis with Gigabit-Ethernet

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[19] [%]
100Mbps	335,04		334,89	335,24	335,56	0,11	2,88
1Gbps	33,504		33,3	33,59	33,79	0,08	0,29

Table 50: Summary of the formal and simulation results for Profinet IRT using non-preemptive strict-priority algorithm for 100Mbps and 1Gbps

The simulation results for a sample duration of nine seconds for 100Mbps and 1Gbps link-speeds are shown in Figure 81.

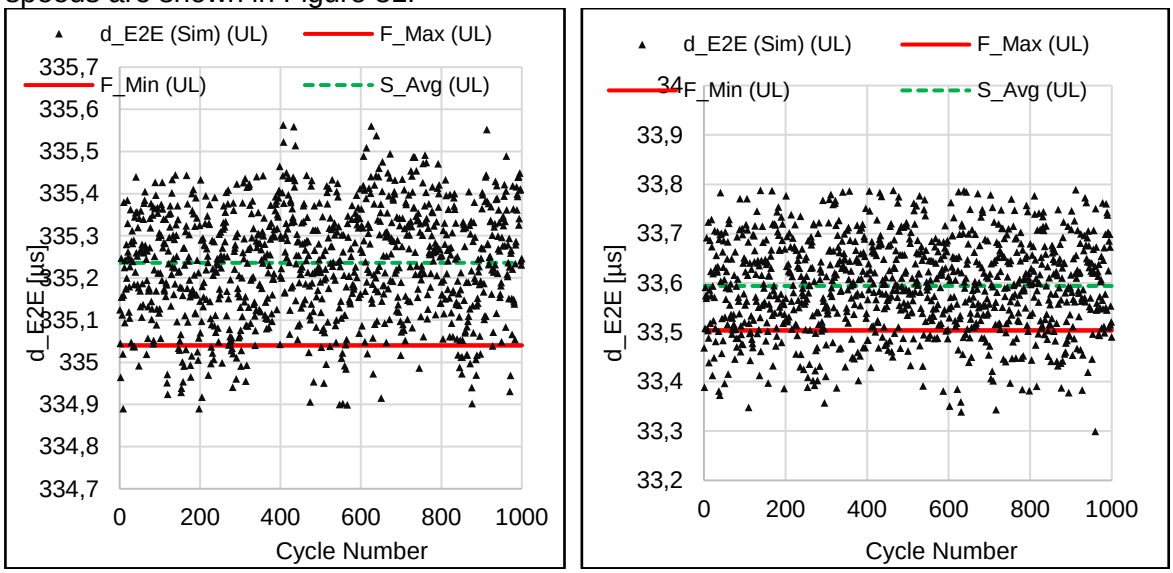


Figure 81: Minimum Cycle Time [µs] of Profinet IRT using Non-Preemptive Time-Aware Shaper – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation

Compared to the previous two protocols Ethernet/IP and Profinet RT, a strong reduction of the end-to-end delay is observed with Profinet-IRT. This is mainly due to two factors: supporting the cut-through forwarding mechanism and scheduling the control-data traffic. The latter ensures that the frames are not interfering with the background traffics throughout the network. An end-to-end delay below one millisecond can be easily achieved. This can be reduced almost by a factor of 10 with gigabit-Ethernet. Avoiding the interference results into an ultra-low delay variation below << 1µs for 100Mbps and 1Gbps.

4.4.3 Timing Analysis with Frame-Preemption (IEEE802.1Qbu)

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[20] [%]
100Mbps	335,04	905,04	335,05	340,87	341,13	0,29	2,88
1Gbps	40,03	92,71	42,72	46,49	49,99	0,98	0,29

Table 51: Summary of the formal and simulation results for Profinet IRT using preemptive strict-priority algorithm for 100Mbps and 1Gbps

[19] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)
[20] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)

The formal computation of the minimum and maximum possible cycle times gives the upper bound ranges for the simulation model. Further it is used to guarantee the boundaries of the time-critical control data especially for highly deterministic applications, e.g. motion-control. These need to be recognized in the planning and the network configuration.
For industrial applications with soft-real-time requirements, the formal computation could be used to improve the bandwidth utilization.

The simulation results for a sample duration of one seconds for 100Mbps and 1Gbps link-speeds are shown in **Figure 82**.

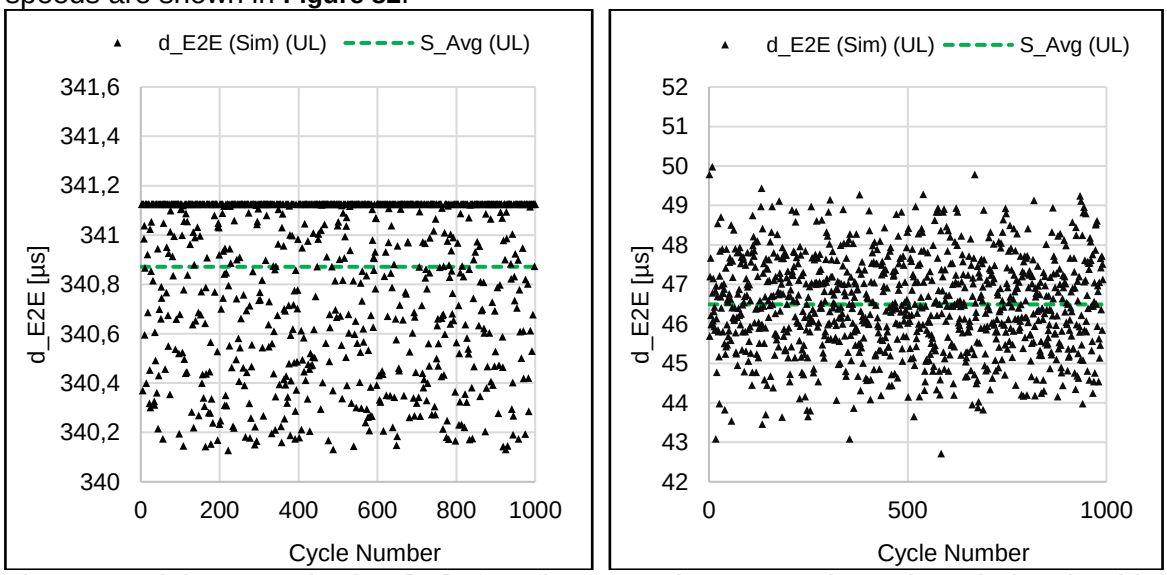


Figure 82: Minimum Cycle Time [µs] of Profinet IRT using Preemptive Strict-Priority Algorithm – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation

Replacing the scheduling approach by frame-preemption does not improve the timing performance. On the contrary the end-to-end delay range is increased from 334µs to 341µs. The jitter slightly increased from 110 nanoseconds to 290 nanosecond but still below one millisecond. A tradeoff can be made between the increased end-to-end delay and the network design complexity. The preemptive strict-priority algorithm does not require a separate time synchronization protocol and also complex traffic scheduling and planning.

4.4.4 Timing Analysis with Frame-Scheduling (IEEE802.1Qbv)

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[21] [%]
100Mbps	335,04		334,89	335,236	335,563	0,11	30,586
1Gbps	40,026		39,85	40,15	40,38	0,13	3,06

Table 52: Summary of the formal and simulation results for Profinet IRT using non-preemptive time-aware shaper for 100Mbps and 1Gbps

The figures below compare the timing performances for 100Mbps and 1Gbps link-speeds.

[21] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)

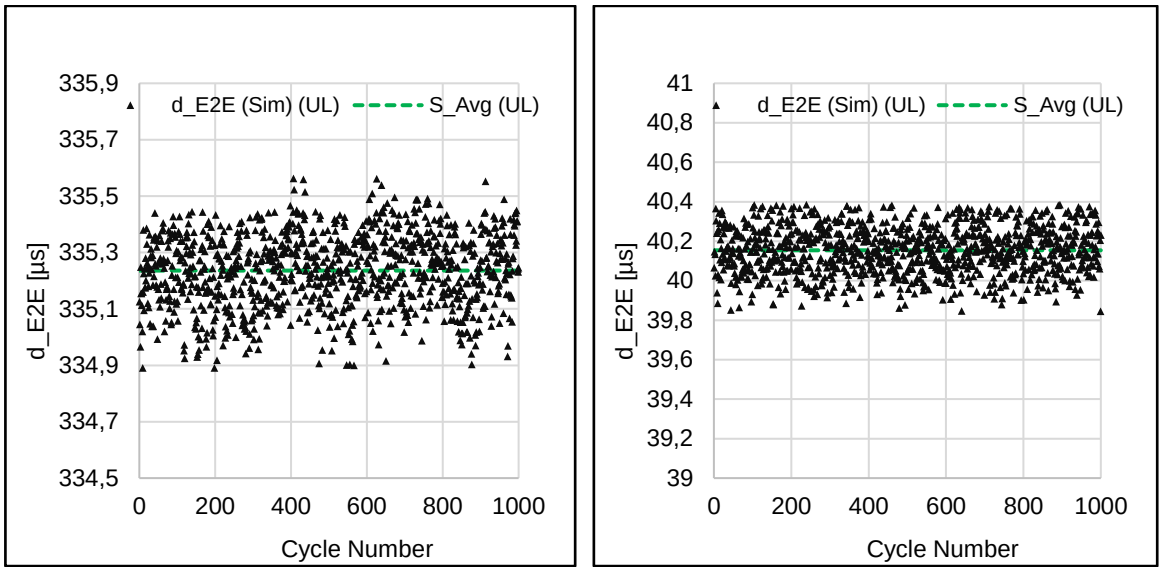


Figure 83: Minimum Cycle Time [µs] of Profinet IRT using Non-Preemptive Time-Aware Shaper – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

4.4.5 Summary

All scenarios reached an ultra-low jitter in the submicroseconds. The difference between the maximum and minimum delays is kept small. Cycle times in the microsecond range can be guaranteed for 50 devices connected in line. The highest performance improvement is reached by increasing the link-speed. Combining Gigabit-Ethernet with frame-preemption has even better performance than the frame-scheduling approach with Fast-Ethernet as long as no express traffics coexist on the network. In real networks, safety- and time-critical traffics that are not-preemptable (express) would coexist and thus worsen the timing behavior of the covered control-data traffic.

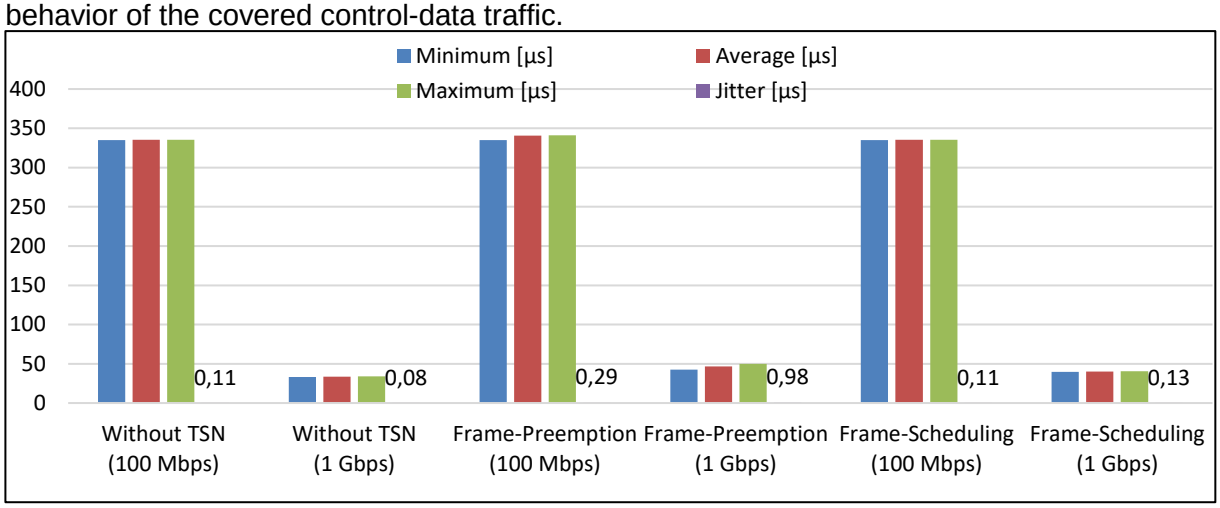


Figure 84: Timing performance comparison of Profinet IRT using different communication features

4.5 Sercos III

4.5.1 Formal Timing Analysis

Similar to Profinet IRT, Sercos III supports cut-through to forward the scheduled isochronous Sercos frames. Because of using the scheduling approach the frames are not interfering with the background traffics. Unlike the previous protocols the Sercos frames are summation-frames (Chapter 2). For the selected 36Byte payload per device four Sercos frames (MDT0 – MDT1 – AT0 and AT1) are required for the 50 slave devices. The frames

are generated and transmitted from the master to all slave devices and are received back at the master.

[27] The minimum cycle time is defined as the elapsed time from the time point of transmitting the first bit of the first Sercos frame (MDT0) until the time point of receiving the first bit of the last Sercos frame (in this case AT_1). It is assumed in this thesis, that the Sercos minimum cycle time T_{sercos} includes the transmission durations of the Sercos MDTs and ATs (d_{MDT} ; d_{AT}) and the network delay $d_{Network}$ (without Best-Effort Frames).

$T_{sercos} = \sum_{i=0}^{3} d_{MDT_i} + \sum_{i=0}^{3} d_{AT_i} + d_{Network}$

$d_{Network} = d_{ct} * ((2 * n_{devices}) - 1)$

The transmission duration depends on the supported link-speed and the frame size. For an MDT Frame it can be calculated as: $d_{MDT_i} = \frac{MDT_{iSize}}{bit\ rate}$

Each Sercos frame consists of 42 byte Ethernet Overhead d_{EthOv}, 6 byte Sercos Telegram Overhead $d_{SercosOv}$, 42 to 1500 byte Payload $d_{SercosPayload}$ and at least 12 byte Inter Frame Gap d_{IFG}.

Sercos Frames does not support VLAN-Tagging in current line and ring topologies and need to be extended with 4 byte VLAN-Tag for tree topology.

$MDT_{iSize} = d_{EthOv} + d_{SercosOv} + d_{SercosPayload} + d_{IFG}$ (3)

The network delay $d_{Network}$ of a Sercos frame corresponds to the forwarding delays caused by the individual slaves and depends on the topology.

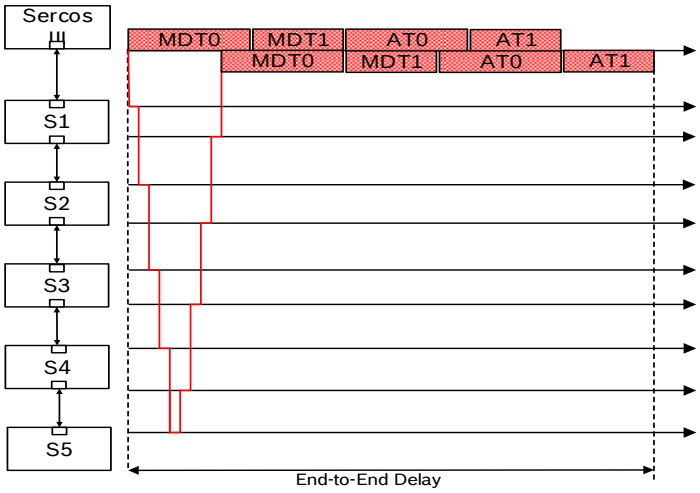


Figure 85: Space-Time diagram of the Sercos traffic for a line topology

Common traffic configurations

The Sercos III traffic configuration is illustrated below.

Sercos III Frame	Overhead [Byte]	Payload / Dev [Byte]	Frame Size [Byte]	Cycle Time [ms]	Throughput [Mbps]
MDT0	44	36	1520	1	29,984
MDT1	32	36	356		
AT0	40	36	1516		
AT1	32	36	356		

Table 53: Traffic Configuration for Sercos III

4.5.2 Timing Analysis with Gigabit-Ethernet

Table 54 shows the formal and simulation results for the Sercos traffic using the TDMA mechanism. All Sercos III control-data traffics have been scheduled all over the network to avoid any interference with the background traffics.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[22] [%]
100Mbps	403,18		403,18	403,47	403,74	0,13	29,98
1Gbps	129,868		129,55	129,84	130,11	0,13	3,00

Table 54: Summary of the formal and simulation results for Sercos III using non-preemptive strict-priority algorithm for 100Mbps and 1Gbps

The simulation results for a sample duration of 1 second (1000 cycles) for 100Mbps and 1Gbps link-speeds are shown in Figure 86.

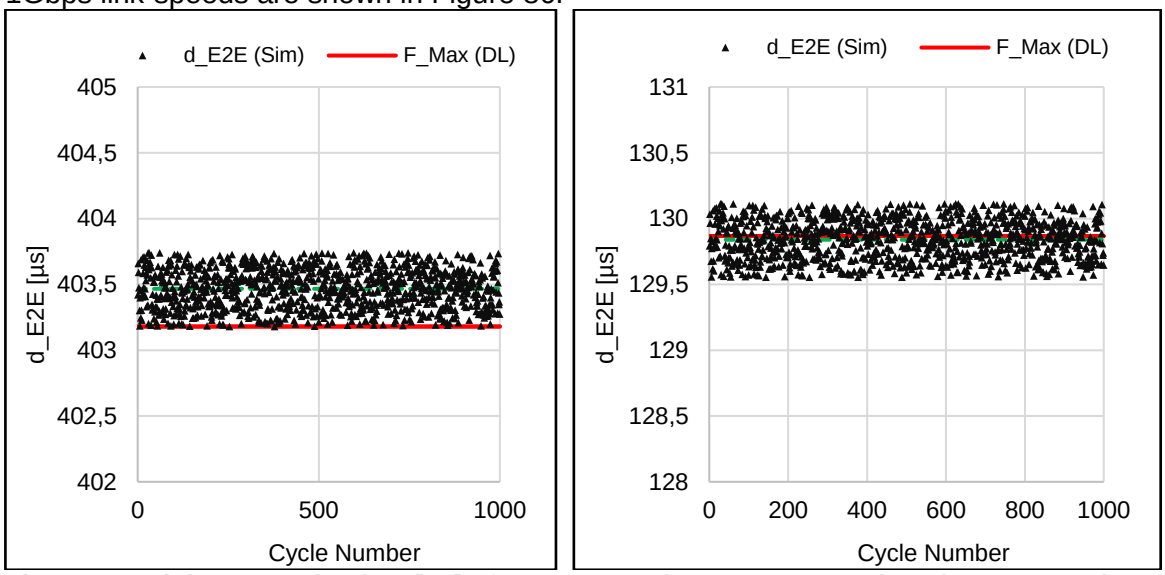


Figure 86: Minimum Cycle Time [µs] of Sercos III using Non-Preemptive Time-Aware Shaper over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation

The cycle time model of Sercos III is based on round trip communication. The end-to-end delay is not twice as compared to Profinet IRT because Sercos uses summation frames. Hence, for the same payload and same number of devices assumed in this work, the collective frame sizes are less than Profinet IRT. Similar to Profinet IRT cut-through and the scheduling approach are also supported by Sercos. Therefore, the end-to-end delay variation is in sub-microsecond range (13 nanoseconds). A reduction of the end-to-end delay is achieved once the link-speed is increased from 100Mbps to 1Gbps. Because of the cut-through delay is a dominant factor in end-to-end delay, the delay is not reduced by a factor of 10 but roughly by a factor of 3.

4.5.3 Timing Analysis with Frame-Preemption (IEEE802.1Qbu)

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[23] [%]
100Mbps	556,86	1610,22	562,74	576,71	591,42	3,48	30,112
1Gbps	110,39	214,93	110,43	112,12	115,69	0,71	3,0112

[22] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)
[23] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)

Table 55: Summary of the formal and simulation results for Sercos III using preemptive strict-priority algorithm for 100Mbps and 1Gbps

The simulation results for a sample duration of 1 second (1000 cylces) for 100Mbps and 1Gbps link-speeds are shown in **Figure 87**.

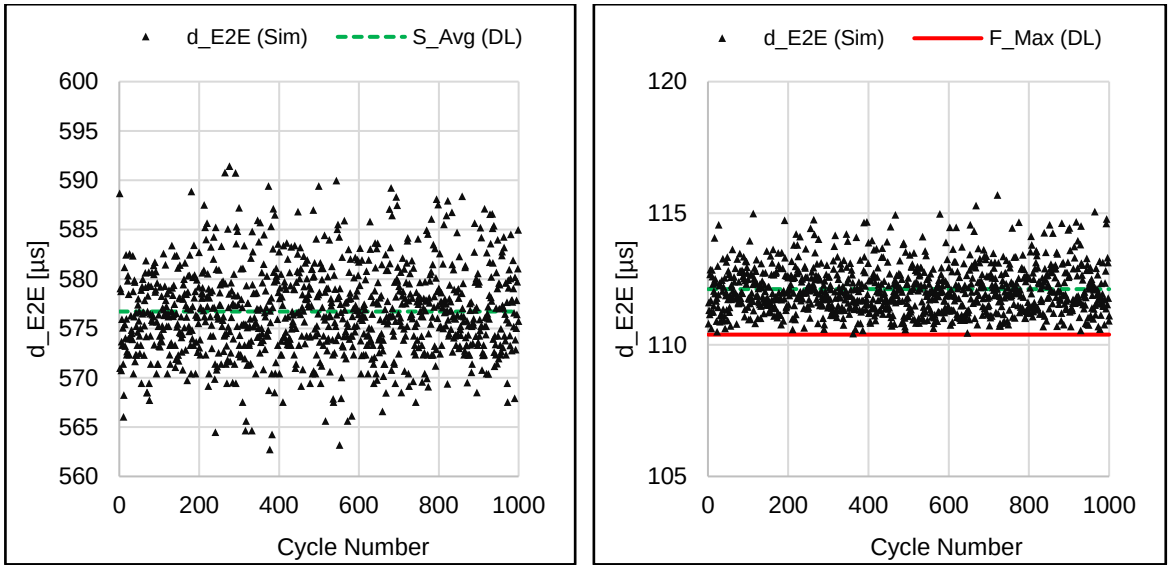


Figure 87: Minimum Cycle Time [µs] of Sercos III using Preemptive Strict-Priority Algorithm – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation

The frame-preemption approach, which is not conform to the Sercos specifications, increases the latency of Sercos III. For 100Mbps the average jitter is increased from the nanosecond range to 3.48 µs. The jitter behavior is again below 1µs with Gigabit-Ethernet. The difference between the minimum and maximum end-to-end delay is ~30µs for 100Mbps and ~5µs for 1Gbps link-speeds. The high synchronization accuracy of Sercos is achieved by the MST field of the MDT0 frame. Since the frame-preemption cannot avoid an interference with the background traffic like the scheduling approach, the synchronization accuracy is decreased with the increasing background traffics.

4.5.4 Timing Analysis with Frame-Scheduling (IEEE802.1Qbv)

The formal and simulation results are illustrated below.

Link-speed	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Total Bandwidth Utilization[24] [%]
100Mbps	556,86		554,12	554,41	554,68	0,13	29,90
1Gbps	110,388		109,95	110,24	110,51	0,13	~3

Table 56: Summary of the formal and simulation results for Sercos III using non-preemptive time-aware shaper for 100Mbps and 1Gbps

The figures below compare the timing performances of the Sercos traffic using different link-speeds.

[24] The total bandwidth utilization includes the throughput of all traffics (Ethernet/IP and background)

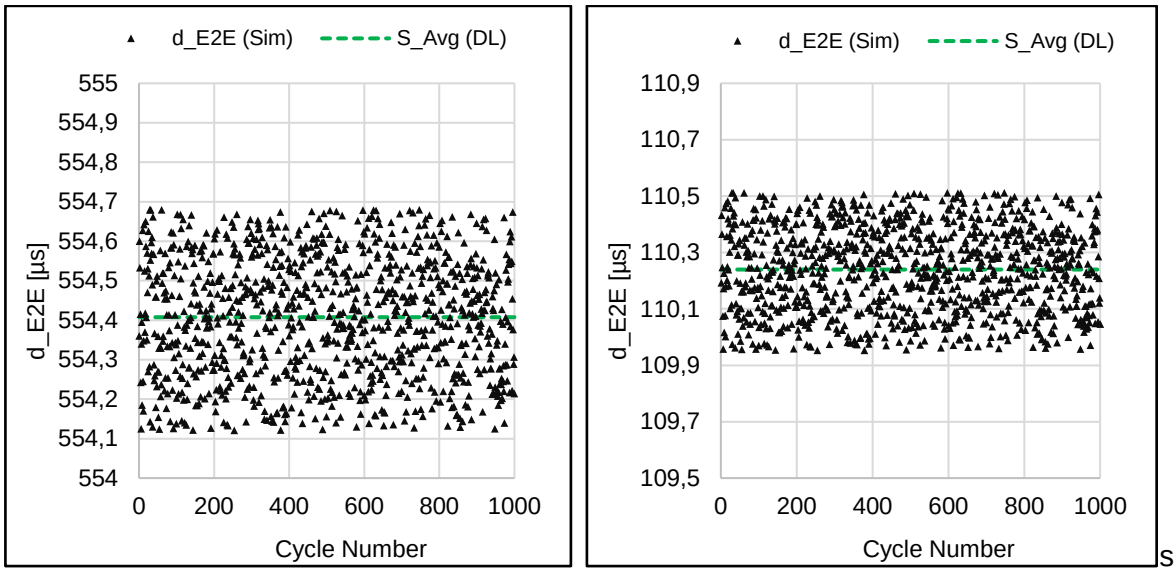


Figure 88: Minimum Cycle Time [µs] of Sercos III using Non-Preemptive Time-Aware Shaper – over 50 Slave devices - (S&F, Processing Duration of 3µs) - 30% Network Load (left: 100Mbps – Right: 1Gbps)

Results Observation

The frame-scheduling approach as supported by Sercos today is the most suitable solution to reach the highest timing performances. For 100Mbps and 1Gbps link-speeds the jitter is in the nanosecond range. The difference is visible for the cycle time parameter. Supporting gigabit promises much higher reduction of the cycle time, which enables connecting more slave devices in the line topology without performance degradation.

4.5.5 Summary

All scenarios promise cycle times below one microsecond. However since the Sercos time synchronization approach depends on the variation of the reception time point of the MDT0 frame, the frame-preemption approach is not suitable without supporting a separate synchronization protocol such as gPTP as defined in IEEE802.1AS-rev.
The performances are shown in the figure below.

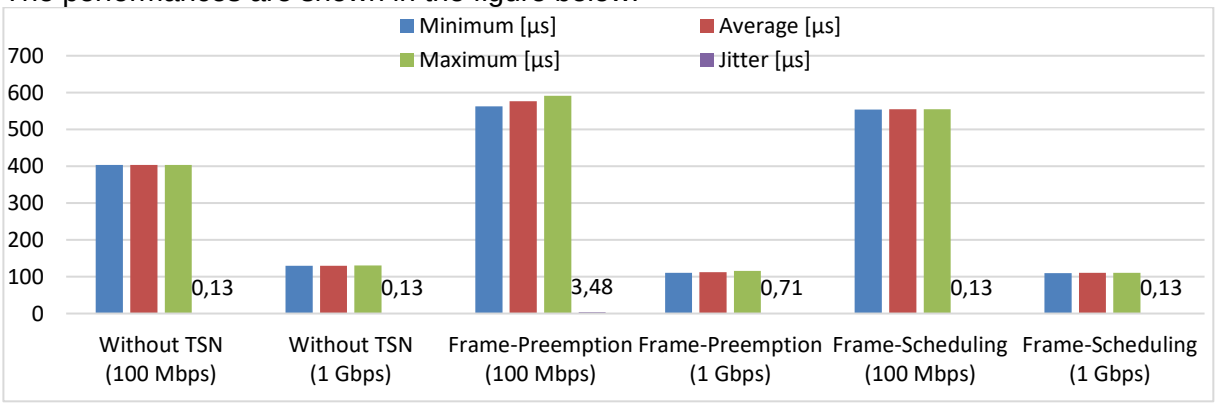


Figure 89: Timing performance comparison of Sercos III using different communication features

4.5.6 Challenges and Benefits of Integrating TSN in Sercos

This section investigates the challenges and benefits of tunneling the Sercos traffic over a TSN subnetwork (one or multiple TSN hops). Different scenarios are illustrated. The focus of this section is:

- Enabling a topology flexibility by integrating a TSN-switch between the master and slaves.

- Highlighting the required functionalities that need to be supported by the Sercos devices and TSN-hops to tunnel Sercos traffic through a TSN subnetwork.
- Presenting an adapter device that adjusts the Sercos frames on the edges of the TSN-subnetwork to avoid the hard- and software modifications of the Sercos master- and slave-devices.
- Analyzing the influence of the topology migration from line/ring to tree on improving the timing behavior, the availability and the safety.
- Topology Migration

Today three network topologies are possible with Sercos: single line, ring or double line (broken ring). This topology limitation has a direct impact on the cycle time. More topology flexibility would not only influence the timing behavior but also enable and simplify certain use-cases, such as cloud to sensor applications.
The Sercos traffic is generated only by the master device and is forwarded through all devices for the downlink and uplink stream communications. Therefore the number of the connected devices has a direct impact on the resulting network delay and thus the overall cycle time. The ring topology, for example, halves the network delay of the line topology, since the Sercos traffic is only forwarded for one stream direction, and thus reaches the master earlier. However for the worst case scenario, in which the communication to the closest slave device to the master is lost (e.g. medium damaged), a line topology is rebuild. Therefore the ring topology cannot be used to reduce the minimum cycle time although the network delay is strongly reduced.
Integrating one or multiple TSN-switches between the Sercos master and multiple slave lines results into a tree topology as shown in Figure 90. The tree topology of *Sercos over TSN* is generated by dividing a long Sercos line into multiple short-lines as shown below.

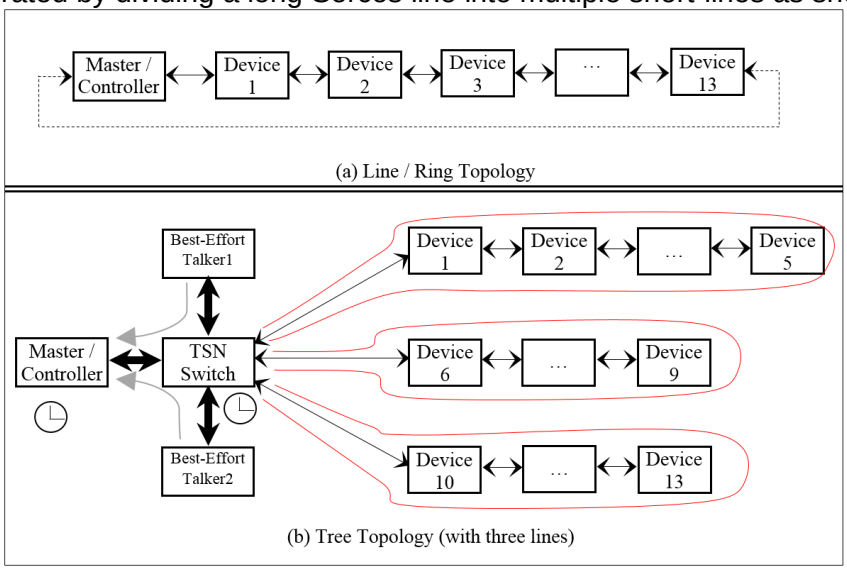


Figure 90: Topology Migration from Ring/Line to a Tree [11]

Now the Sercos traffic follows a new path and will be forwarded by the store&forward TSN-Switch. Cut-through forwarding will not be possible since the link to the Master supports higher link-speed (1Gbps) than the links to the short-lines. Instead of transmiting up to eight Sercos frames to one Sercos *line* or *ring*, the master could transmit now up to eight Sercos frames to each short-line. Periodically a *block* of MDT frames and a *block* of AT frames will be exchanged in the gigabit link between the Master and the TSN-Switch. Based on their VLAN-ID the TSN-Switch forward each frame in 100Mbps link-speed to the corresponding short-line. [11]
The Sercos over TSN cycle time will be calculated as follow:

$$T_{SoTSN} = d_{MDT\_Block} + d_{AT\_Block} + d_{SwFwdToLines} + d_{ShortLine} + \sum_{i=0}^{3} d_{MDTi} + \sum_{i=0}^{3} d\_AT_i + d_{SwFwdToMaster} + d_{MDT\_Block} + d_{AT\_Block}$$

Challenges

- Time Synchronization with two different mechanisms: Sercos encapsulates the synchronization pattern within the MDT0 frame. TSN on the other side, presents a protocol-based synchronization (gPTP) with specific synchronization frames to exchange the synchronization values between the devices. Unlike Sercos, the gPTP frames are independent of the control-data traffic. Since this TSN feature requires hardware support, the gPTP between the Sercos Master and the TSN-switch.
- Sercos frames are broadcast-based and need to be extended with a VLAN-Tag to be identifed as time-critical by the TSN-switch and to be forwarded to the correct short-line. Since the current Sercos slaves does not support VLAN's, the switch need to remove the Tag in the short-line direction and add it in the Master direction. This feature is also limited between the Sercos Master and the TSN-switch.
- To maintain the short-lines synchronous, the TSN-switch must forward the Sercos MDT0 frames, which include the synchronization pattern, simultaneously to the short-lines. This requires highly synchronized clocks of the TSN-Switch and the Master to guarantee a transparent integration of the TSN-Switch in a Sercos Network.

Assumptions

- The cut-through forwarding delay d_{DevFwd} of a single Sercos device is ~ 800ns [4] and is assumed to be 1µs. It includes the delays of reception and transmission PHYs as well as the delay of the forwarding logic. The impact of gigabit on the forwarding delay is igonred because only the PHY delays are reduced by increasing the link-speed
- The cable delay between two connected devices is set to 50ns. This corresponds to 10meter copper-based segment and has to be counted twice (forward and reverse channels) for each line of cascaded field-devices.
- The number of Sercos frames depends on the number of devices and the payload per device. We assume the same size of payload per device for each slave in the downlink (MDT's) as well as for the uplink (AT's[25]).
- The division of the total number of Sercos slaves yields in to multiple lines with an equal number of slaves or with exactly 1 slave more.

4.5.6.1 Feasibility Analysis (Challenges)

4.5.6.1.1 Challenge 1

The Sercos frames, that are cyclically generated and transmitted from the master device, are broadcast-based (destination MAC). Tunneling these frames through a standard Ethernet network could quickly flood the non-Sercos domains, since the TSN-switches would forward the Sercos frames to all ports. In case of connecting a Sercos ring to the TSN-edge switch, the broadcast Sercos frames would create a bridge loop (frames would circulate endlessly within between the slave devices), which leads to a bad bandwidth utilization and quickly to overflow the Sercos network.

Solution approach

The problem can be solved by modifying the Sercos frames.

- Either supporting multicast instead of broadcast destination addresses or
- Extending the Sercos frames by a VLAN-tag with the highest priority 7 in order to be identified and handled as scheduled traffics at the TSN-subnetwork.

4.5.6.1.2 Challenge 2

Today Standard-Ethernet frames generated by the Sercos slave device within the UC channel can be transmitted outside of the Sercos network. This is possible either through

[25] AT: Acknowledgement Telegram – includes the current values (e.g. position) of the Sercos slaves

a special network infrastructure device, Sercos master device or by connecting a Standard-Ethernet device at the end of a Sercos line.
Tunneling Sercos frames through TSN subnetwork requires tagging all received Sercos control-data (MDTs & ATs) at the TSN-switch port connected to the Sercos slave device(s) in order to avoid flooding non-Sercos domain(s) by the broadcast based MDT and AT frames.
Since these Sercos frames received back from the slaves are not tagged, it is hard for the switch at the edge of the TSN-subnetwork to identify them in order to force their forwarding to the appropriate port. However this would lead to force also non-Sercos frames received at the UC channel and results into a restriction of the network convergence of Sercos.

Solution approach
This restriction can be avoided by identifying the untagged Sercos frames, which is a special functionality that can be either supported by the TSN-switch or an additional adapter device. The Sercos frames identification can be either

- data-field-based – reading the EtherType or or VLAN-tag (Inter vlan routing – source: [68])
- time-based – reception time
- Inter-VLAN Routing [75]

Adapter functionalities
The figures below show the modification of the Sercos frames bypassing the adapter device. Only the overhead is adjusted for the downstream and the upstream directions.

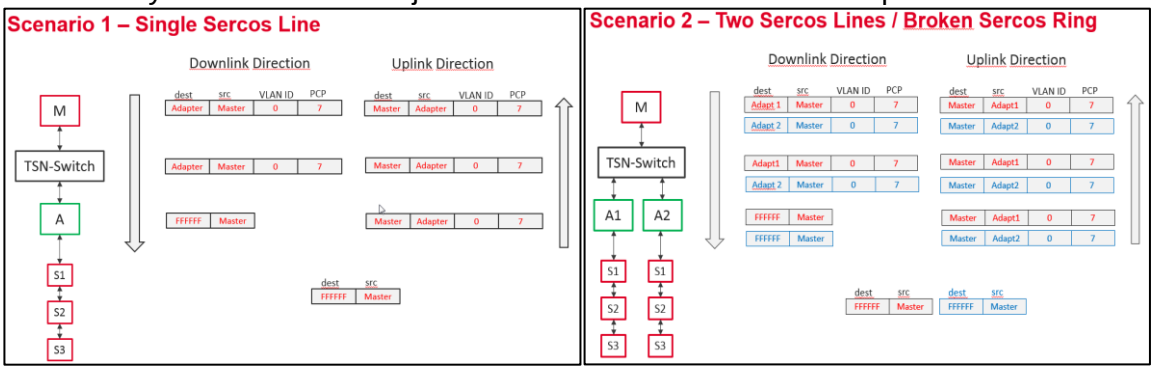


Figure 91: Two Scenarios for the adapter functionalities of Sercos over TSN Concept (left: Single Sercos Line, Right: Two Sercos Line or a physically broken Sercos Ring)

The main purpose of introducing an adapter is making the Sercos traffic conformant to standard-Ethernet in the TSN network edge(s) and the TSN-netwrok. For simplification reasons the TSN network shown in the figure above is composed of a single TSN-switch. Tunneling the Sercos traffic is explained below:

- **Master** generates modified unicast Sercos frames with the destination address of the adapter instead of broadcast so that the broadcast-based Sercos traffic does not flood the TSN-network. The frames are VLAN-tagged with the highest priority PCP 7, which could be used in order to be identified as scheduled traffic in the TSN-switches and forwarding hops. Further assigning the highest VLAN priority is mandatory to transmit an isochronous data-traffic in a time-critical Ethernet network. The scheduling is done on the forwarding ports of the switches.
- **TSN-Switch** forwards the Sercos frames within the reserved time-slot(s) for the downlink and uplink directions.
- **Adapter:** for the downlink stream direction the Sercos traffic is modified: The destination MAC address is kept broadcast and the VLAN-tag is removed. For the uplink stream direction the destination address is converted to unicast (destination MAC address of the master device). The source MAC address gets the address of the adaptor and a VLAN-Tag (highest priority, PCP7) is added.

The required features to tunnel the Sercos traffic over a non-Sercos (TSN-) subnetwork is summarized in the table below. An acceptance condition for the Sercos over TSN concept is keeping the Sercos slave devices unmodified.

Device	Necessary features
Sercos Master	• IEEE802.1AS(-Rev) for time synchronization • Time-Triggered Transmit or IEEE802.1Qbv for traffic scheduling • Gigabit link-speed (in case of multiple Sercos lines/rings)
TSN-Switch	• IEEE802.1AS(-Rev) for time synchronization • IEEE802.1Qbv – for traffic scheduling • Gigabit link-speed (on the port to the Sercos master in case of multiple Sercos lines/rings) • VLAN tagging of the Sercos MDT and AT frames to the Master • VLAN untagging of the Sercos MDT and AT frames to the slaves •
Adapter	• Dependent of the scenario, must support at least 2 ports (100Mbps) • Requires an own MAC address • VLAN-Tagging/Untagging of the Sercos frames (could also be done by the TSN-Switch) • Modifying / adjusting the MAC addresses of the Sercos frames • Cut-Through and modify on the fly • Sercos frame identification either by reading the EtherType or time-based. For the time-based approach the adapter need to be synchronized by the MST field like the slave devices.
Sercos Slaves	• Unmodified, fully backward compatible

Table 57: Functionalities of the devices

4.5.6.1.3 Challenge 3

In case of redundancy, i.e. if Sercos slaves are connected back to the TSN-switch, the UC broadcast traffic would lead to the circulating frames on the Sercos line. The scenario is even more complicated if a Sercos ring breaks again into two lines.

Solution approach

This could be solved using common functionality of switches – "channel bonding" and Spanning-tree protocol. These functions can recognize the scenario with ring or line and automatically switch forwarding of broadcast UC traffic in TSN-switch on the ports connected to the Sercos slaves

4.5.6.2 Timing Analysis (Benefits)

[27] "The flow of the Sercos frames for the tree topology with three short-lines, as given in topology figure above, has been illustrated in the figure below. The Master transmits six Sercos frames to the Switch. At the Switch, the MDT0's are forwarded simultaneously to the three short-lines in order to maintain the slaves of different lines synchronous. The AT's are forwarded as soon as the MAC of the forwarding port is idle again. As shown in the figure below, the frames become 10 time bigger on the *Switch←→Line* links than on the *Master←→Switch* Links. This is due to the different link-speed. Once received by all the slaves of the corresponding short-lines, the frames are forwarded back to the Switch. Because of the different link-speed, forwarding the frames back to the master is much faster than receiving them from the slaves. This results into a gap between the MDT's and AT's which requires a bigger time slot on the backward direction Switch-to-Master.

$$TimeSlot_{SwitchToMaster} = TimeSlot_{MasterToSwitch} + Gap$$

For more network bandwidth, the timeslot can be reduced. The start time point of the reserved Time Slot get shifted by the duration of the gap. That means the MDT's get delayed although they are ready to be forwarded to the master."
Figure 92 shows the simulated end-to-end delay variation (jitter) of the MDT0 frame that is transmitted through TSN-switch to a line of 5 slave devices. The Frame is *"timestamped"* as it is received back at the Master. The followed path is Master → TSN-switch → short Line1 downstream direction → short Line1 upstream direction→ TSN-switch → Master.
Over 600 values are plotted for a simulation duration of 1.3 sec for a cycle time of 2 millisecond. The Jitter is ±0,9 µs.

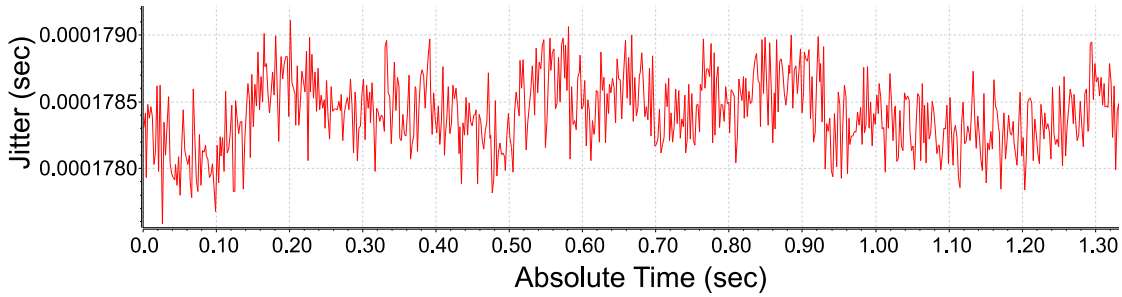


Figure 92: Delay Variation (Jitter) [sec] of the Sercos MDT0 frame over 130 seconds

Figure 94 shows the impact of the topology and the link-speed on reducing the Sercos cycle time. The cycle time for a network composed of 100 Sercos slave devices and a network of 13 Sercos devices is strongly reduced by increasing the number of the connected Sercos short-lines. Compared are the Sercos line and ring topologies with the Sercos over TSN concept with multiple shorter lines. Two cases are covered. For the first case a gigabit link-speed is limited to the master – switch link and for the second case the slaves support gigabit as well.
Increasing the number of short-lines without modifying the link-speed at the slave level continuously reduces the cycle time. Increasing the link-speed to 1Gbps for the whole network combined with the tree topology migration promise an ultra-low end-to-end latency of the time-critical frames and thus enable cycle times below 200µs even for 100 slave devices. [27].
The impact of the topology migration becomes almost linear for more than two short-lines.
An exemplar payload of 137Byte per device is selected for the network of 13 slave devices and 20Bytes per device for 100 slave devices.

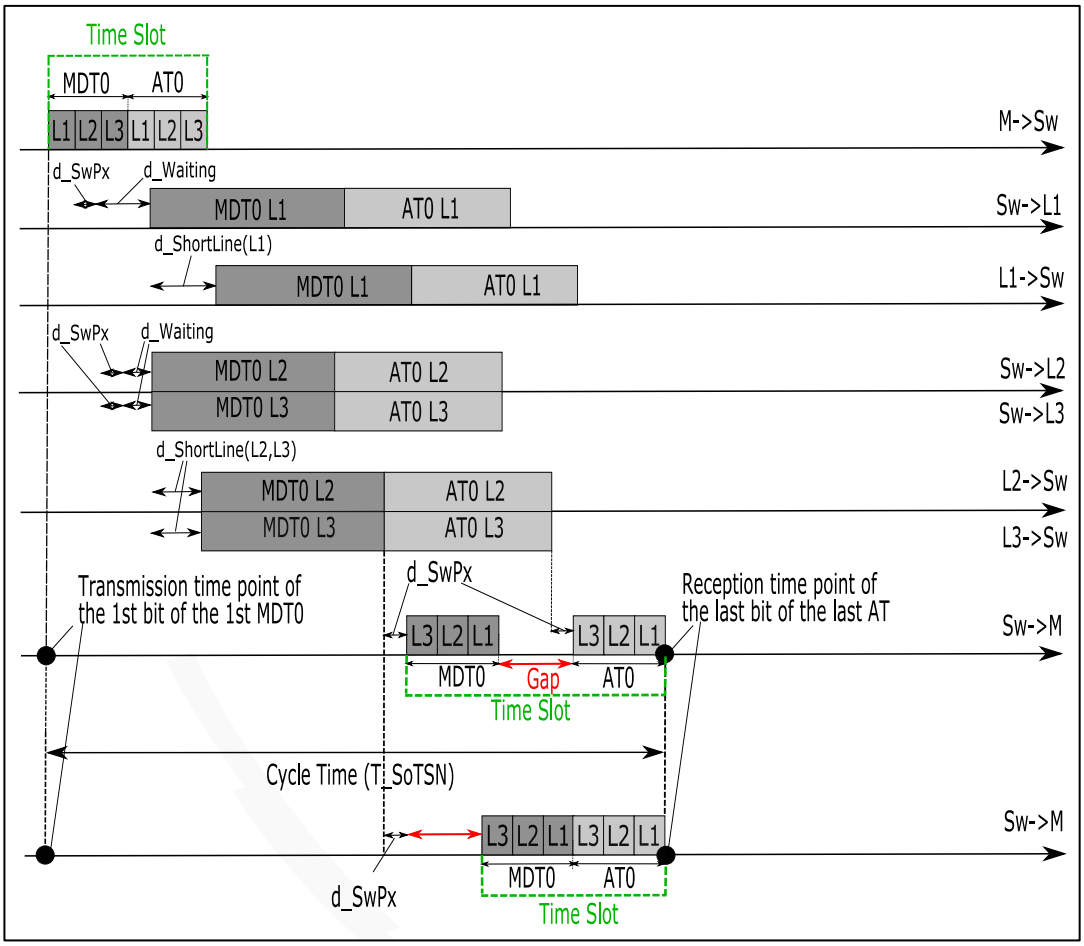


Figure 93: Space-Time diagram of the Sercos III traffic

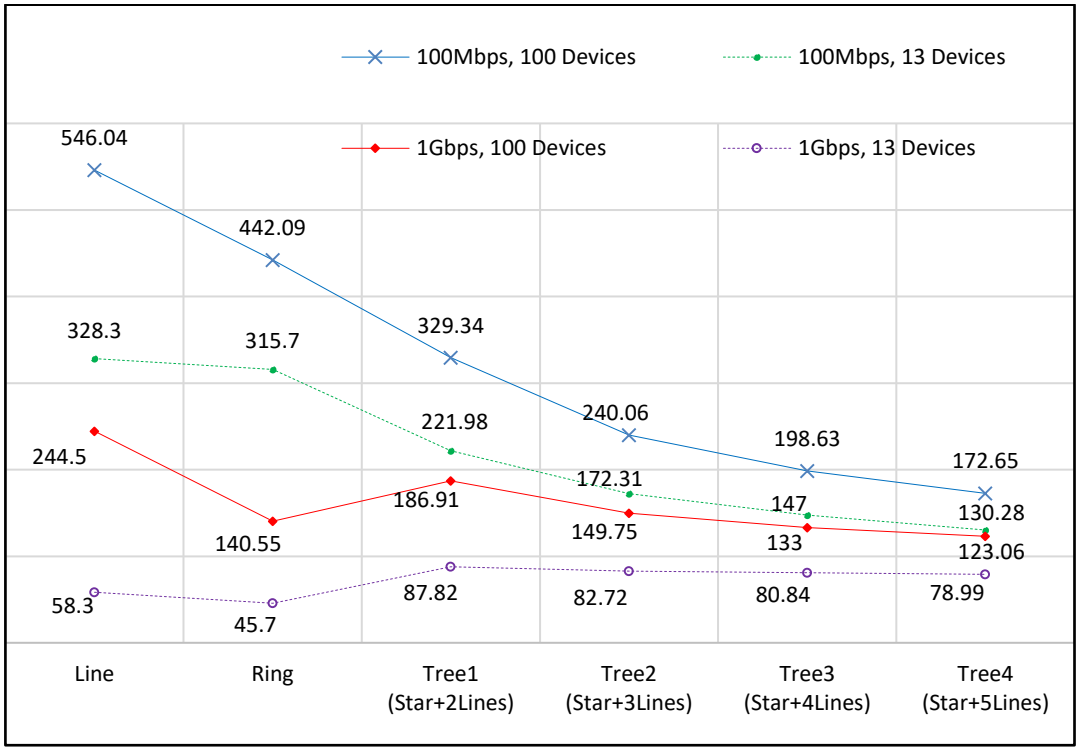


Figure 94: Comparison of the timing performance of Sercos and Sercos over TSN using different link-speeds and topologies

The diagram below compares the cycle times for Sercos line and tree topologies as a function of the number of slave devices. The figure on the right side shows the reduction of the Sercos network delay as a function of the number of slave devices.
For both figures increasing the number of short-lines strongly reduces the delay and thus the resulting cycle time.

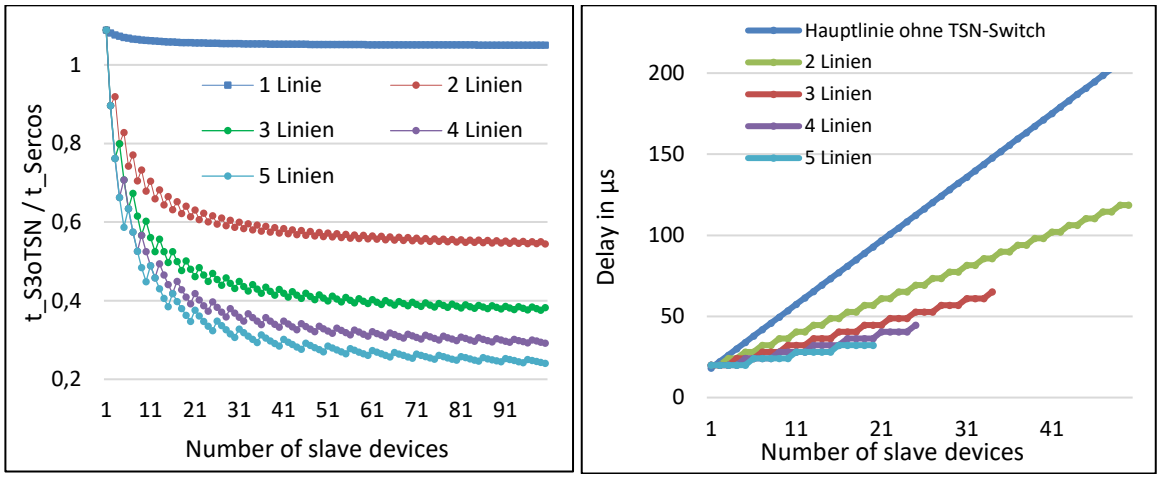


Figure 95: Sercos III cycle time reduction using different topologies

4.6 Converged Network

Typically information technology (IT) and operational technology (OT) networks are strictly separated. IT systems are used for data-centric computing and have typically high bandwidth requirements. OT systems are used for process / events monitoring and adjusting industrial operations with real-time requirements and quick response time.
The convergence of IT and OT networks would improve manufacturing operations, productivity and security. Further it is considered as an enabler for the Industrial Internet of Things (IIoT).
Because of its importance, a converged IT/OT network is considered in this section. The timing behavior of the time- and safety-critical traffics of the industrial devices (PLCs and drives) are sharing the network infrastructures (Ethernet switches) with video streams and diagnostic traffics from IT devices (camera, PCs).
In order to further guarantee the timing requirements of the OT systems, TSN-features are used in the Ethernet switches. Two TSN approaches are covered.
The first approach aims to guarantee the highest timing performances (ultra-low jitter, delay and cycle time) by avoiding the interference. This is possible through scheduling and planning the control data traffics throughout the whole network. Therefore IEEE802.1Qbv is required. The configuration complexity of the network is a big challenge for this approach.
The second approach aims to reach performances that are accepted by most deterministic applications without adding any configuration complexity. This is possible by increasing the link-speed to gigabit and supporting the frame-preemption approach, as defined in IEEE802.1Qbu and IEEE802.3br.

4.6.1 Network Description

In order to evaluate the two approaches, described above, a network model (Figure 96) has been simulated. The network model consists of a TSN-subnetwork, composed of four switches, and multiple end-devices. The end-devices can be divided into talkers (transmitter of the traffics) and listeners (receivers).

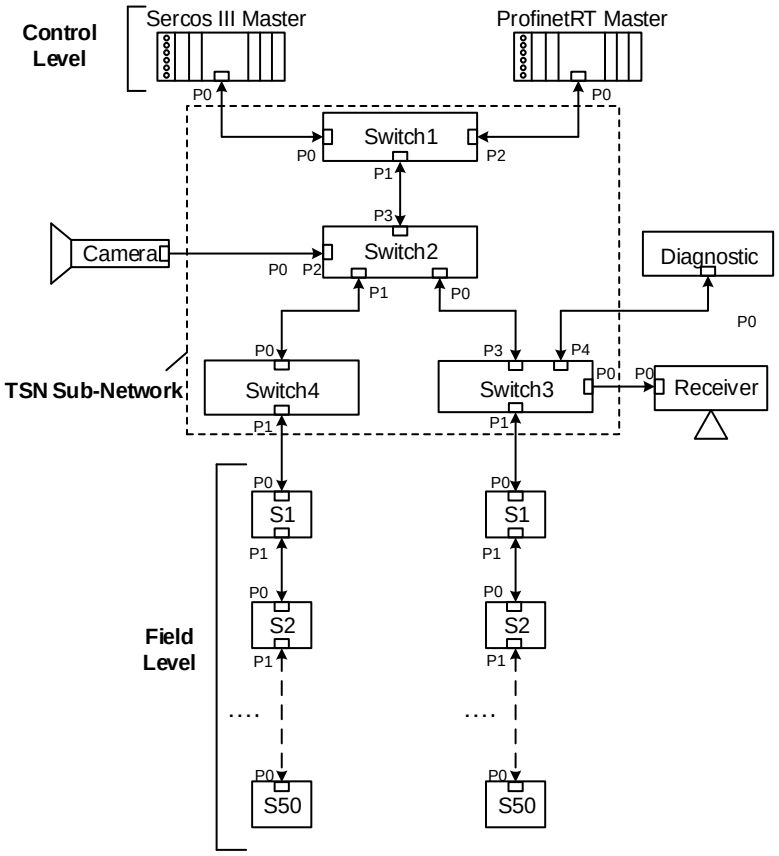


Figure 96: Converged network model showing the integration of a TSN-subnetwork within the industrial-Ethernet network

Traffics from different classes are exchanged through the TSN-subnetwork and are sharing common network resources.

This chapter analyzes the timing behavior of tunneling two control-data traffics through the TSN subnetwork:

- *non-scheduled express* industrial-Ethernet protocol Profinet RT
  Path: Profinet RT Master – Switch 1 – Switch 2 – Switch 3 – Slave devices and back
- *scheduled express* industrial-Ethernet protocol Sercos III
  Path: Sercos III Master – Switch 1 – Switch 2 – Switch 4 – Slave devices and back

The control-traffics are sharing the network resources with the following background traffics:

- *non-scheduled express* video traffic transmitted as a “block” of frames
  Path: Camera – Switch 2 – Switch 3 –Receiver device
- *non-scheduled preemptable* diagnostic traffic 1 transmitted each 3ms
  Path: Diagnostic PC – Switch 3 – Profinet RT Slave 1 ... Slave 25
- *non-scheduled preemptable* diagnostic traffic 2 transmitted each 3ms
  Path: Profinet RT Salve 25 ... Slave 1 – Switch 3 – Diagnostic PC

The traffic configurations is illustrated below.

Traffic Name	Traffic Type	Traffic Class	Size (Bytes)	Number Frames	Cycle Time (ms)	Throughput [Mbps]
Sercos III	TC	Scheduled Express	1524 - 360 1520 - 360	4	2	1,5056
Profinet RT	TC	Non-Scheduled Express	72	50 – Downlink 50 – Uplink	3	1,44
Video	TC	Non-Scheduled Express	1530 - 307	68	33	2,49
Diagnostic 1 Diagnostic 2	NTC	Non-Scheduled Preemptable	131	4	3	0,034

Figure 97: Traffics configuration

A payload of 36Byte per device has been selected for both control-data traffics. This results into four summation frames MDT0, MDT1, AT0 and AT1 for Sercos III and 100 individual frames for Profinet RT: 50 for the downstream and 50 for the upstream communications.

Network Configuration

A link-speed of 1Gbps is supported within the TSN subnetwork and 100Mbps outside of the TSN network. The TSN-switches support frame scheduling, frame-preemption as well as the store&forward approach, since cut-through is not IEEE compliant. Frame scheduling as well as frame-preemption is supported throughout the TSN-subnetwork.

The Sercos and the Profinet RT lines consist each of 5o slave devices and support 100Mbps link-speed. Cut-through approach with a forwarding delay of 1µs per slave device is supported on the Sercos line. And the store&forward approach with a processing delay of 3µs per slave device is supported on the Profinet RT line.

4.6.2 Simulation Results

4.6.2.1 Profinet RT

The Profinet RT traffic flow, illustrated in the figure below, faces multiple interferences. For the downstream communication, the Profinet RT frames are delayed at the egress ports

- P1 of switch 1 by 20µs, each second Profinet RT cycle, due to the reserved time-slot for the scheduled Sercos traffic.
- P0 of switch 2, each 11th Profinet RT cycle, due to the interference with the non-scheduled express video traffic. It is assumed that video data is transmitted as a stream (as a block of multiple frames).
- P1 of switch 3, cyclically, due to an interference with the non-scheduled preemptable diagnostic traffic. The diagnostic traffics considered in the thesis are IP-traffics and not time-critical control data.

For the upstream communication, the Profinet RT frames are delayed at the egress port of the Profinet RT slave device 25, which is transmitting a diagnostic traffic to the diagnostic transmitter device.

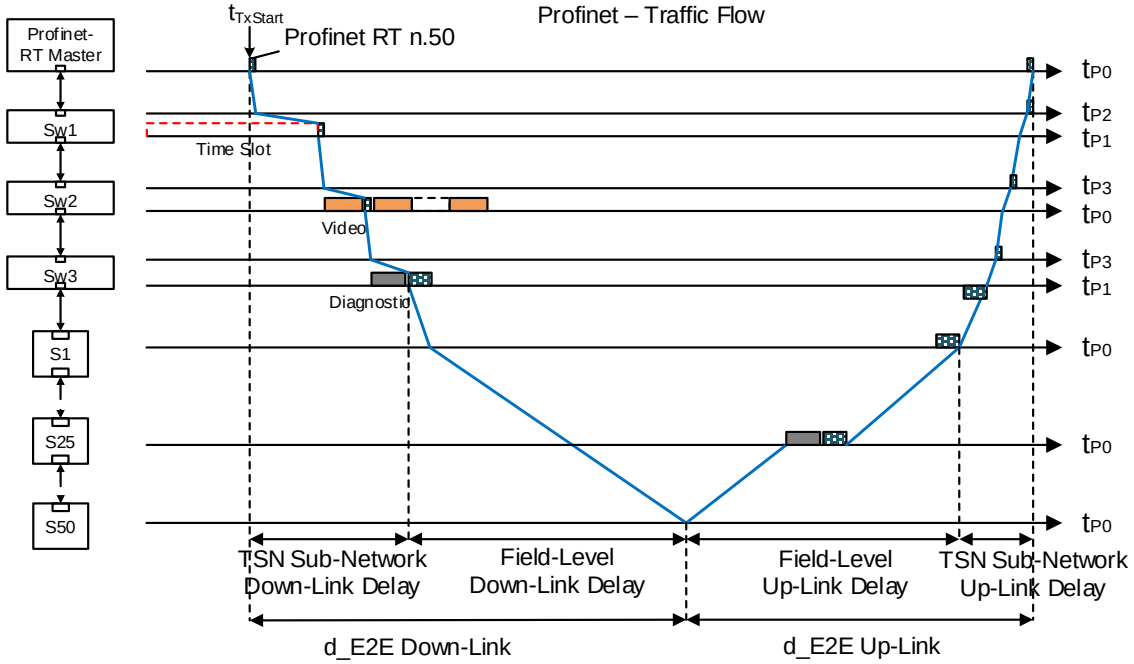


Figure 98: Space-Time diagram of the Profinet RT traffic: Down- and Uplink communications (Worst-case delay)

The simulation results are illustrated in the table below.

Traffic	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Bandwidth Utilization [%]
Profinet RT	710,47	748,97	710,473	724,876	746,568	8,507	0,96

Table 58: Resulting end-to-end delay (minimum cycle time) of Profinet RT traffic

Figure 99 shows a distribution of the delay for the non-scheduled express Profinet RT traffic. It is proven that supporting the frame-preemption approach even combined with gigabit in an open network structure cannot guarantee an ultra-low jitter below one microsecond but it can guarantee an upper-bound worst-case delay. The main reason is the interference with background traffics that cannot be preempted. Even non-scheduled preemptable traffics might be not preempted, if the preemption condition is not met. This additional unpredictable sporadic delay can be a problem especially for link-speed below 100Mbps.

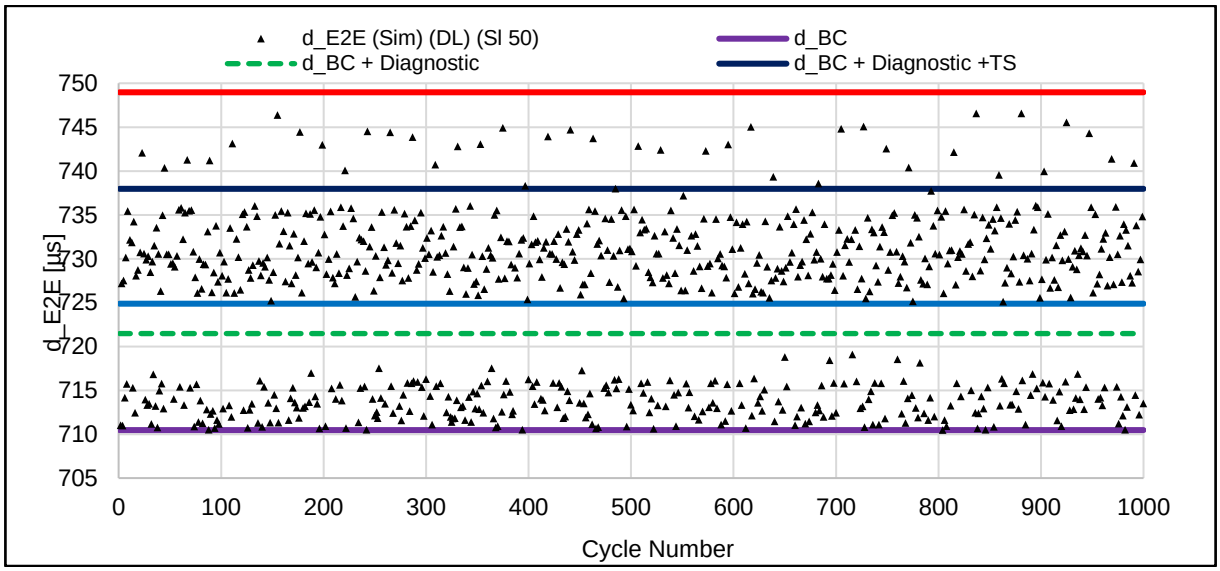


Figure 99: Minimum Cycle Time (using three cycle-time model) [µs] of Profinet RT with Preemptive Strict-Priority Algorithm on the TSN-subnetwork (3x TSN-Switches) and Non-Preemptive Strict-Priority Algorithm on the Industrial-Ethernet Level (50 slave devices)

The two figures below compare the down- and upstream durations for the Profinet RT through the TSN-subnetwork. The downstream traffic is higher delayed than the upstream, due to the higher amount of the background traffics.

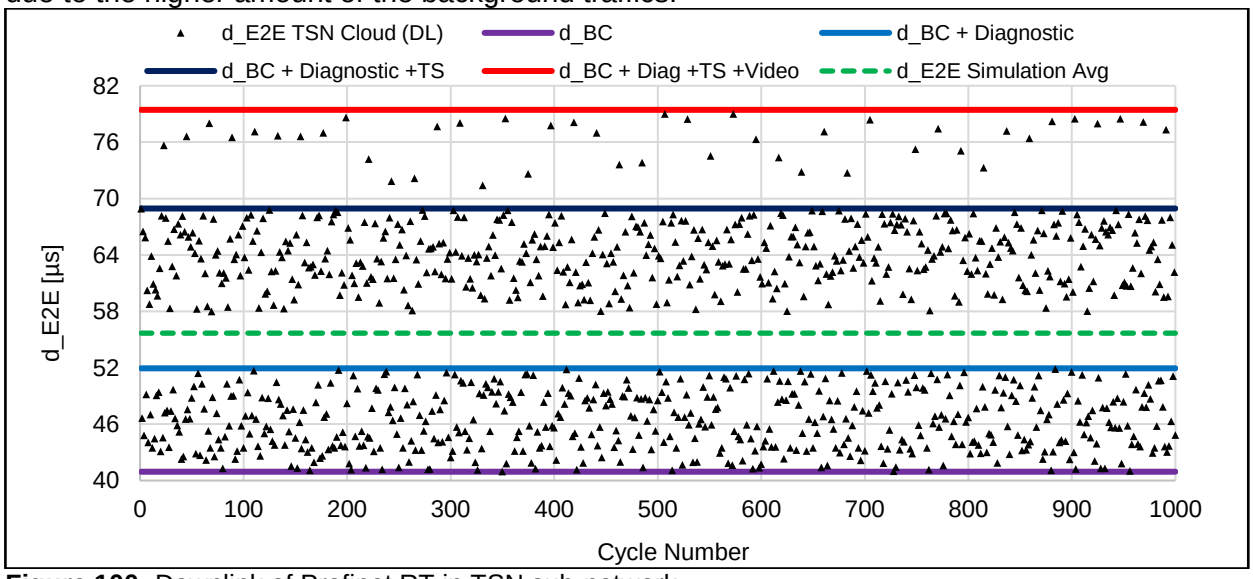


Figure 100: Downlink of Profinet RT in TSN sub-network

4.6.2.2 Sercos III

Since the Sercos III traffic is scheduled throughout the TSN-subnetwork and the Sercos line, interferences are avoided for down- and upstream communications. The traffic flow is illustrated below.
On the ingress port of TSN-switch 4 connected to the Sercos line, a gap is observed between the received Sercos frames. This is due to the difference of the link-speed. For a better use of the bandwidth on the reserved TSN resources for the Sercos upstream path, the first received Sercos frames MDT0 and MDT1 could be delayed and transmitted together with the AT0 and AT1 frames in one single shorter time slot instead of reserving multiple time slots or a big single time slot. Another approach is to reserve two time-slots to improve the bandwidth-utilization to transmit non-scheduled background traffics.

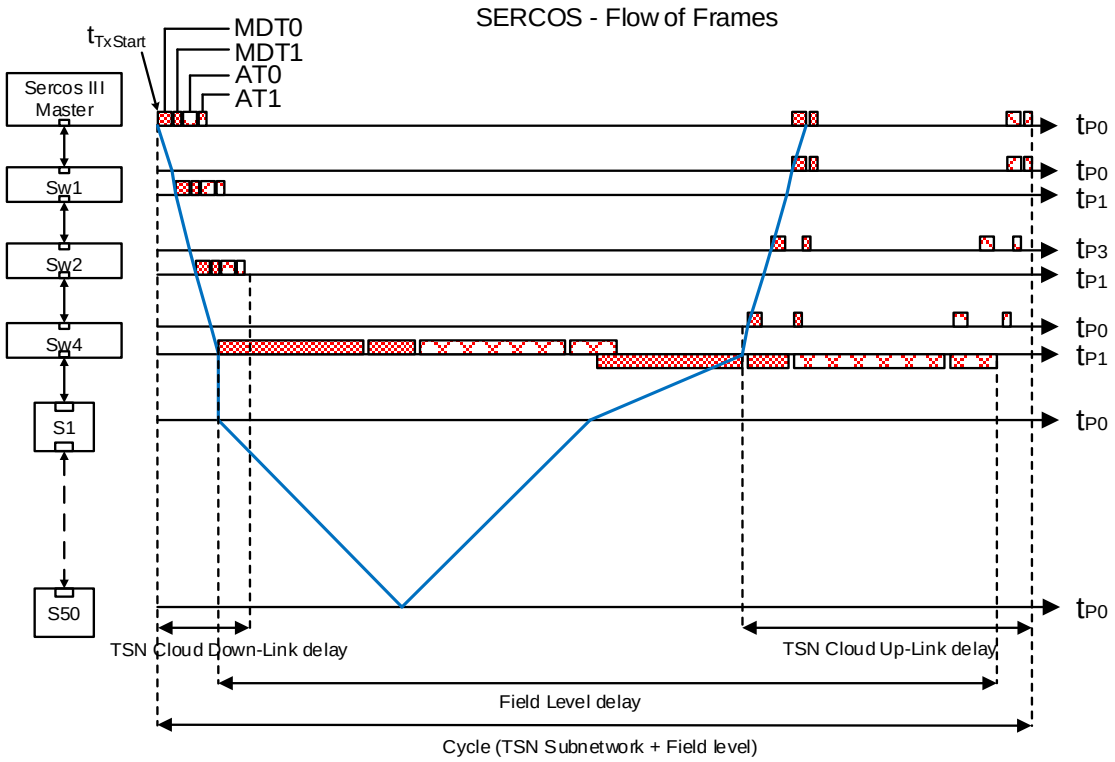


Figure 101: Space-Time diagram of the Sercos III traffic: Down- and Uplink communications

The formal and simulation results are illustrated in the table below.

Traffic	Formal		Simulation				
	T(Min) [µs]	T(Max) [µs]	T(Min) [µs]	T(Average) [µs]	T(Max) [µs]	Jitter [µs]	Bandwidth Utilization [%]
Sercos III	467,272		467,686	467,98	468,82	0,13	1,5056

Table 59: Summary of the formal and simulation results for the Sercos III traffic

Figure 102 shows a distribution of the end-to-end delay for the scheduled Sercos traffic.

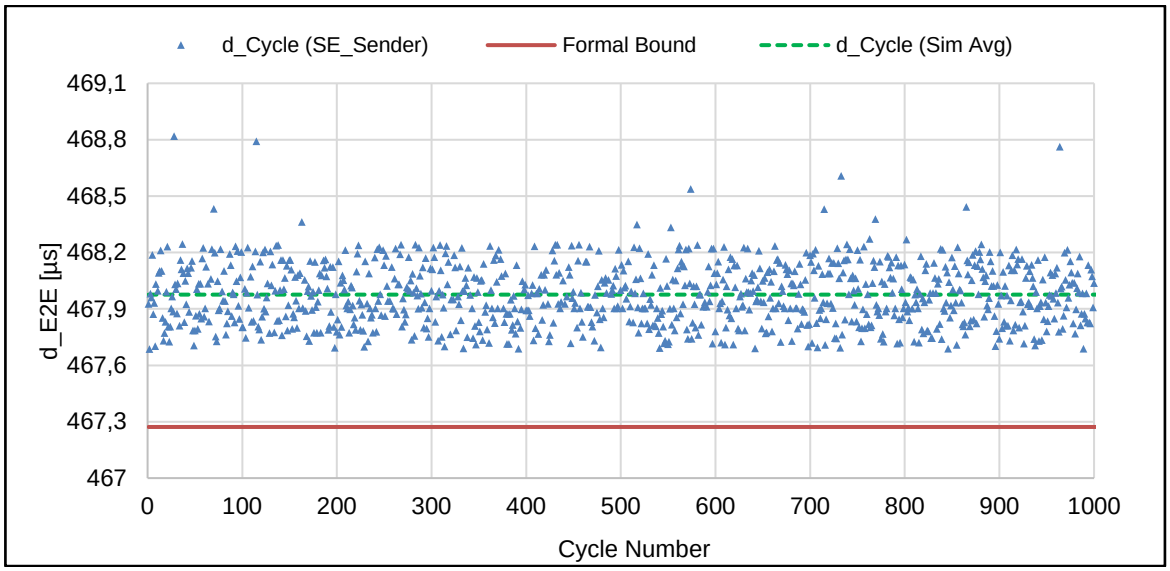


Figure 102: Minimum Cycle Time [µs] of Sercos III using Non-Preemptive Time-Aware Shaper on the TSN-subnetwork (3x TSN-Switches) and Industrial-Ethernet Level (50 slave devices)

4.6.3 Conclusion

This study proved that in order to guarantee an ultra-low jitter in the nanoseconds range and short cycle times below one millisecond, scheduling the traffic throughout the whole network is the only suitable approach. However the complexity of the network configuration need to be investigated.

Combining gigabit link-speed with frame-preemption can strongly reduce the jitter and cycle times without a high complexity of the network configuration. However an upper-bound worst-case delay cannot predicted in the coexistence of other express traffics that cannot be preempted. This approach might be more suitable for closed systems, in which the controller device is the only initiator of the exchanged frames. Otherwise the bandwidth should be limited for other traffics with higher priority. However this condition would strongly limit the network convergence.

5 Experimental Evaluation of TSN-prototypes

Unlike the formal- and simulation-based investigations described in the previous sections, the goals of this chapter is to test the TSN timing behavior using hardware prototypes and to give an early feedback of the readiness and maturity of TSN-prototypes.
Certain challenges, such as standard-conformance tests and interoperability of prototypes from different vendors, can only be verified using physical hardware.
This chapter describes a test setup built during the thesis to evaluate the implementation of four out of nine TSN-specifications described in the (sub-) standards: IEEE802.1AS[26], IEEE802.1Q, IEEE802.1Qbu, IEEE802.3br and IEEE802.1Qbv. Multiple TSN-prototypes from different vendors are tested for standard-conformance and interoperability.

5.1 Test setup Description

Available TSN test setups worldwide
Two TSN test setups have been developed during the research phase of the thesis.

- IIC TSN Test setup at two locations: Erbach (Germany) and Austin (USA).
- LNI 4.0 in Augsburg (Germany).

The main goals of the test setups is exchanging know-how and guaranteeing the interoperability of the TSN-prototypes at an early development stage, especially that the participating companies are from different fields:

- **Industrial automation**: Bosch Rexroth AG, Schneider Electric, National Instruments, ABB, B&R, etc…
- **Semiconductor**: Renesas and Intel
- **IT and Network infrastructure**: Cisco, MOXA, Hirschmann and TTTech

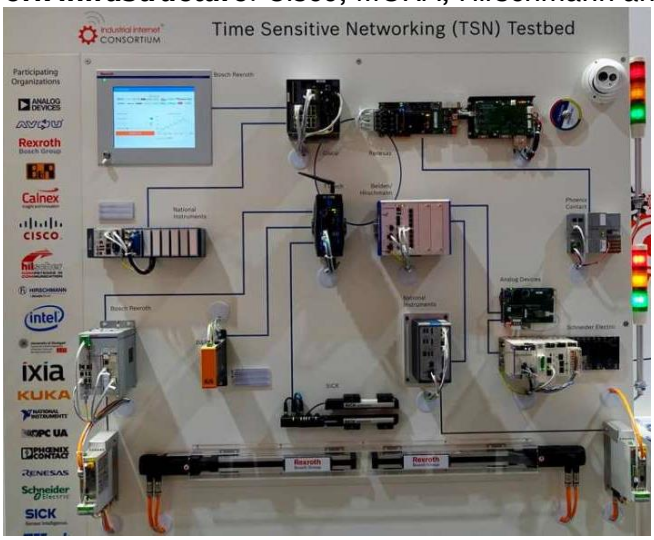


Figure 103: Industrial-Internet Consortium (IIC) TSN Test setup [69]

Market available TSN prototypes (2016/2017)
Multiple TSN-devices that are available on the market during the research phase of this thesis were tested and used to build and investigate a TSN network. Certain prototypes are used in this research. The technical features are described in Table 60: Summary of selected TSN prototypes.

Features	Intel	Renesas	Cisco	TTTech	
	I210 Chip	CEP TSN-Switch	IE4000	Hermes	DE IP Solution Edge

[26] The specifications of the TSN-standard IEEE802.1AS-Rev was not finished during the research-phase of the thesis. Therefore, the vendors implemented the AVB-standard IEEE802.1AS in the TSN-prototypes.

Sync. Protocol	1588	-	✓	✓	✓	✓
	.1AS	✓	✓	✓	✓	✓
	PPS signal	-	✓	✓	-	✓
Transm. Selection Algorithm	.1Qbv	-	-	✓	✓	✓
	Time-Triggered Transmit	✓	✓	-	-	-
	.1Qbu	-	-	✓	-	-
Ports	Number	-	1x	4x	20x	4x
	Gigabit	-	✓	✓	✓	✓

Table 60: Summary of selected TSN prototypes (2017)

Although the I210 chip from Intel did not support any the TSN-standards, at least during the research phase, it was the most suitable chip to be used for a time-triggered transmission of the frames from the end-devices.
The TSN-switches from Renesas, Cisco and TTTech supported the TSN-sub-standards needed to evaluate their timing behavior and interoperability.
Testing the interoperability of the IEEE802.1Qbu and IEEE802.3br implementations was not possible, since it was only implemented by the Renesas TSN-Switch CEP.

TSN Testing solutions

Because of the big market trend to deploy TSN in the automotive and industrial markets, the need to test the available TSN prototypes and to guarantee an interoperable integration becomes more and more urgent.
The technical features of the TSN testing solutions, available on the market during the research phase of the thesis, are listed below.

Vendor	Solution	Features
IXIA	AVB/TSN Conformance Test Solution	AVB Standards: IEEE802.1BA, Qav, Qat, AS and 1722/1733 TSN Standards: IEEE802.1Qbv, gPTP-Rev Supports Link-speed from 100Mbps to 40Gbps
Spirent	Automotive AVB/TSN Conformance Test Solution	AVB Standards: IEEE802.1BA, Qav, Qat, AS and 1722 TSN Standards: None Supported Link-speed 100Mbps and 1Gbps
Calnex	Calnex Paragon x	2x Ethernet Ports Sync. Protocols: 1588 (PTP) – SyncE – NTP – CES – E-OAM –IEEE802.1AS Supports Link-speed from 100Mbps to 10Gbps

Table 61: Market available TSN testing solutions during the research phase of the thesis

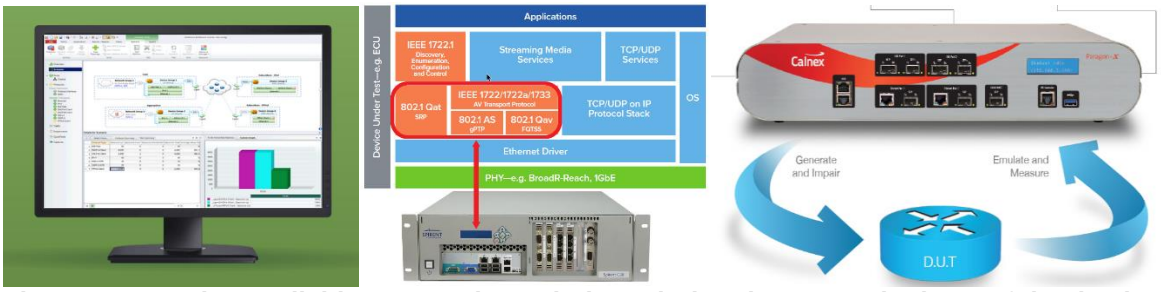


Figure 104: Market available TSN testing solutions during the research phase of the thesis

The TSN testing solutions are built to test the TSN-prototypes. These prototypes are the device under test (DUT). However the testing solutions have benefits and drawbacks:

- Benefits
  - Deep TSN Knowledge is not required
  - Ease of Use (GUI)

        - Automated / Deep Tests are possible
        - Producing log files with all results
        - Plotting the results during the test
    - Drawbacks
        - High prices
        - Cannot be extended to cover special use-cases; e.g. modifying the size, number and class of the frames, adjusting the cycle times and/or the time-slots
        - Requires Support

Especially the high prices of the TSN testing solutions makes it hard to further investigate the TSN-development and to accelerate its deployment for this thesis. Research institutes and small companies for example are facing a huge challenge to build similar solution for their field of research.

The financial and technical challenges are the main motivation to build a test setup that is able to test the timing behavior of the TSN-features in real hardware operation and to check the interoperability of the available TSN-prototypes with respect to limited financial resources.

5.1.1 TSN Test setup

This section describes the TSN-test setup built during the research phase of the thesis.
The figure below illustrates the network design of the TSN-test setup to evaluate the TSN standard-enhancements using four TSN-switches. The TSN-switches are the Devices Under Test (DUTs) and are physically connected in a line topology. Four industrial PCs, that generate multiple data traffics are distributed all-over the network. The Time-Critical (TC) and the Non-Time-Critical (NTC) Transmitter 1 are connected to TSN-switch 1. NTC Transmitter 2 and 3 are respectively connected to TSN-switch 2 and 3. A Time-Critical (TC) Receiver is connected to TSN-switch 4. All four TSN-switches and four traffic transmitters are configured through the Development Host. A Standard-Ethernet Switch is used to distribute the configuration files (TFTP).
A network TAP (for clarity purpose and less cabling the TAP is not shown in the figure below) with four channels is used to measure the End-To-End delay, single switch forwarding delays, interference delays and the jitter of the time-critical traffic. The captured frames are time-stamped by the TAP and uploaded to the development host.

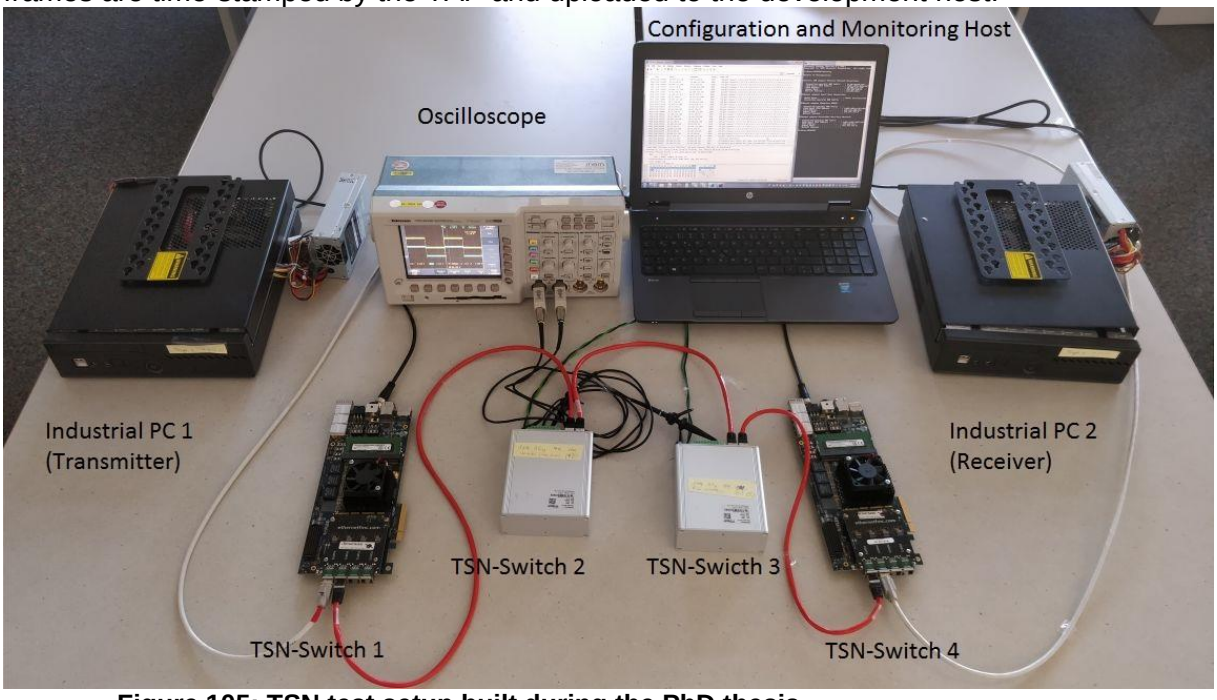


Figure 105: TSN test setup built during the PhD thesis

The physical connections of the devices in the TSN test setup that cannot be clearly visible on the previous photo are drawn below.

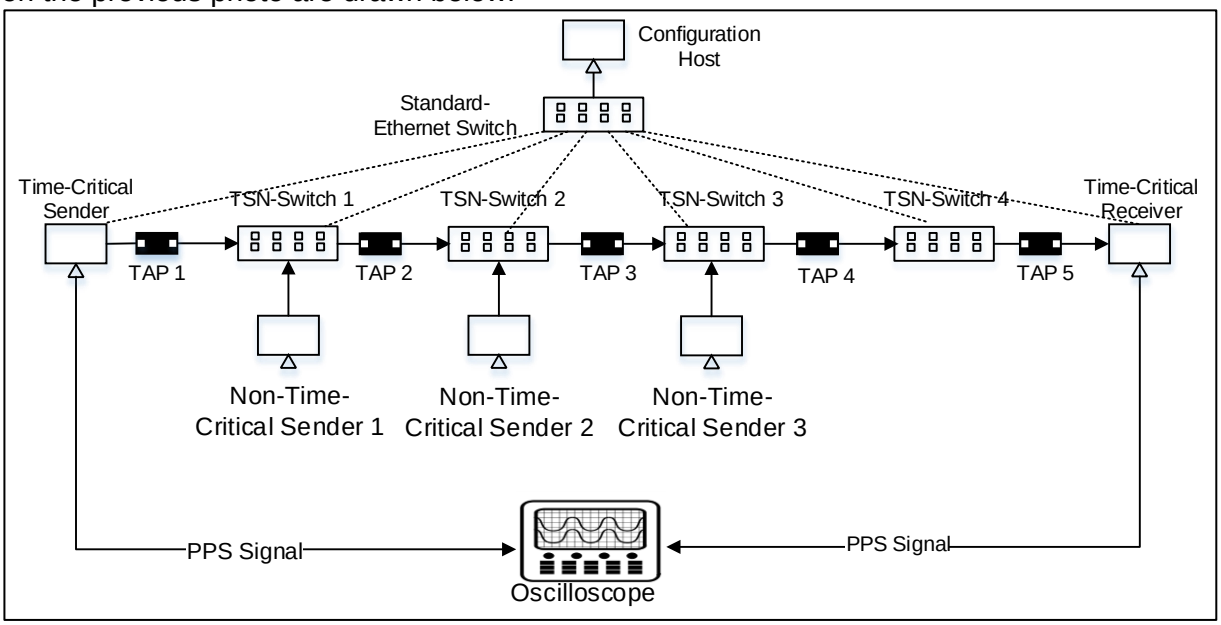


Figure 106: Technical Architecture of the TSN test setup

The Table below illustrates all devices and their functionalities.

Class	Devices	Functionality
Measurement Equipments	Oscilloscope	to visualize and measure the clock synchronization accuracy
	Network TAPs	to capture with timestamp and upload time-critical- and best-effort data traffics into the traffic analyzer
Traffic Transmitters	1x industrial PC	generates *time-critical traffic* at a cycle time of 1 millisecond. For a highly accurate cyclic transmission, the time-critical transmitter should support time-triggered transmit functionality and IEEE802.1AS time synchronization protocol. The time-critical application requires an RTOS to enable cycle times below 1 millisecond
	3x PCs	generate non-time-critical data-traffics to create an interference with the time-critical traffic on the network. (RTOS not required)
Traffic Receiver	1x Industrial PC	receives and consumes the time-critical traffic.
Traffic Analyzer	Standard PC	connected to the network TAPs. It requires Wireshark to monitor the network traffics. The time-stamps can be processed in excel to compute the End-To-End delay and delay variation (jitter).
Configuration Host	Standard PC	to configure the TSN-switches and the traffic transmitters / receiver. It could be also the traffic analyzer.
Network Infrastructure	Standard-Ethernet Switch	to distribute the configurations
Devices Under Tests (DUTs)	4x Store&Forward TSN-Switches	each with the following features - 4x RJ-45 Ethernet ports - IEEE802.1Qbv - IEEE802.1Qbu & IEEE802.3br - IEEE802.1AS

Table 62: Description of the devices within the TSN-Test setup

5.1.1.1 Devices under Test (DUTs)

Industrial-PCs
Two industrial PCs (I-PCs) are used as end devices. One is the transmitter of the time-critical frames. The second is the receiver. Each I-PC is equipped with an Intel® i7 processor, 2048 MB RAM, one Intel® I210 Ethernet controller, one Arensys Ethernet controller. The end devices support the real-time operating system VxWorks. The Intel® i210 controller is tested for integration in the deterministic industrial communication.
The deterministic communication between the end-devices is built using a software and a hardware parts.

- Software: RTOS-based time-critical transmitter and receiver applications
- hardware: i210 controller

For this reason, it is important to test the suitability of the i210 chip for use in highly deterministic communication. After all the communicating applications are the base for an end-to-end delay and jitter measurement and a verification of the TSN-standards on the DUTs (TSN-switches).

Intel® Ethernet Controller I210
The Intel® I210 Ethernet Controller is a single port hardware component that supports Gigabit Ethernet. I210 offers a fully integrated Media Access Control (MAC) and physical (PHY) port as well as PCI express bus [70]. The supported functionalities and communication features are listed below:

- Audio Video Bridging
  - IEEE1588 and IEEE802.1AS for precision time synchronization
  - IEEE802.1Qav for credit-based shaping
- Time-Triggered Transmit to transmit the scheduled traffic(s) at predefined transmission time points to ensure that there will be no interference with the remaining traffics. The functionality makes I210 suitable behave as the time-aware shaper defined in IEEE802.1Qbv for an end-device
- VLAN tagging
- Four transmit and four receive queues
- Four Software Defined Pins (SDPs) can be used e.g. to visualize the synchronization signal on the oscilloscope

TSN-Switches (Device under Test – DUT)
For experimental verification, TSN-switches from three different vendors (A, B and C) have been configured and used to transmit the data traffics. Each switch has four RJ45 Ethernet ports operating at 100Mbps/1Gbps link-speed. The switches support IEEE802.1AS, IEEE 802.1Q (VLAN tagging), time-triggered transmit, IEEE802.1Qbv and/or IEEE802.1Qbu.

5.1.1.2 Measurement Equipment

Several measurement devices are used to analyze the timing behavior of the DUTs.

- Terminal Access Points (TAPs) to probe the Ethernet connections. A TAP line has two ports. The frames are received on the first port, where they are mirrored and time stamped. The time-stamped frames are sent to the connected host via the uplink port. The original frame is sent to the network through the second TAP port. The TAP used in this work is capable of synchronous probing of 4 Ethernet connections at 100Mbps.
- Oscilloscope: The pulse per second (PPS) signals of the IEEE 802.1AS-capable devices are visualized using an oscilloscope. This is required to visualize the time error (clock drift) between the clocks of the connected devices in real-time.

- I210 Network controller can time stamp the received and transmitted frames in an accuracy of ±20nanoseconds [71]. This functionality is required to measure the end-to-end delay of the exchanged time-critical frames. Only time-triggered frames can be time-stamped.

Measurement Errors:
According to its data sheet [72] the Network TAP gives a time-stamp accuracy of 40 ns with a resolution time-stamp of 1 ns. A forwarding delay of 500-600 ns is added by the TAP ports. This means each Ethernet frame forwarded by the TAP gets is delayed by 500-600 ns. For an endl-to-end delay a frame could be forwarded three or four times over the TAP between the transmitter and the receiver end-devices. Which leads to additional delay [1 to 2 µs] and worsen the jitter. For this reason, the TAP is only used to measure the forwarding delay of single switches. E.g. to measure the internal processing delay of a switch for different frame sizes and link-speeds.
The end-to-end delay measurement is realized by time-stamping the frames with i210 controller.
The measurement results of the switch internal delay proved that the later depends on the activated link-speed. Different values have been observed depending on the activated link-speed (Table 63). The frame size has no effect.

Link-speed	Switch Processing delay [µs]	
	Vendor A	Vendor B
100Mbps	14	7
1Gbps	11	2

Table 63: TSN-switch Processing delay (vendor A)

5.1.1.3 Configuration and Monitoring Subnetwork

The Test Applications are developed in C++ programming language to generate, transmit and receive time-critical and non-time-critical frames. To perform a regression testing of the timing performance, network load is introduced through a non-time-critical transmitter application. Wireshark is used to monitor the exchanged network traffic. The applications are configured according to the covered communication scenario.

Test Applications:
Three test applications are created:

Time-critical Transmitter: This application generates the time-critical frames following the configuration parameters set by the user

- Cycle duration
- Frame Size
- VLAN priority (PCP)
- Number of frames per cycle

The frames are sent using the time-triggered functionality of the Intel®I210 controller. The application calculates the launch-time of the frames in each cycle. This launch-time is used by the Intel®i210 controller to transmit the frames in time-triggered mode. The Ethernet frames used in this application are Layer-2 Ethernet frames. The application bypasses the UDP/IP and TCP/IP stacks and transmits the frame using the routines provided by the driver of the controller. This helps reducing the delay between the application and the controller. The activity diagram of the time-critical transmitter application is shown in Figure 107.

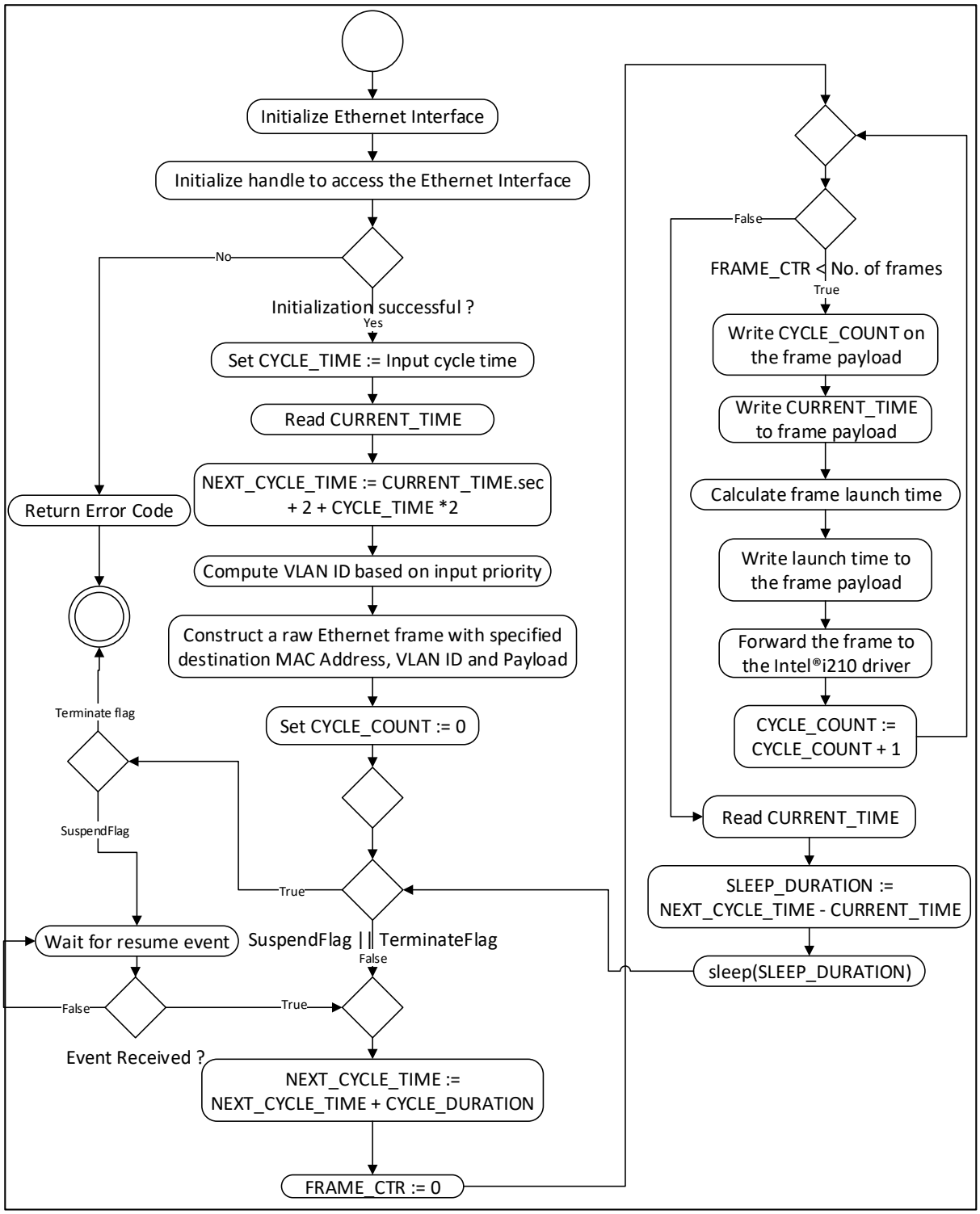


Figure 107: Activity diagram of Time-Critical transmitter application

Non Time-critical Transmitter: This application is used to manipulate the bandwidth usage of the network by generating and transmitting non time-critical frames. The configuration parameters set by the user are:

- Frame size
- Network load (0% - 25 %- 50% -75% -100%) and is the used bandwidth
- Cycle duration
- VLAN priority

The application calculates the number of frames to be sent in a given cycle time. The number of frames depends on the link-speed and the frame size. For the same size, eleven

times more frames are sent in 1Gbps as compared to 100Mbps to achieve the same network load.

Receiver: This application receives and time-stamps the Ethernet frames and generates a log file on the local storage. The log file contains the timing information of the received frames (transmit time-stamp, reception time-stamp). The time-stamps are then used to calculate the resulting end-to-end delay for each frame.

end-to-end delay = transmit time-stamp – reception time-stamp

Wireshark: is used to monitor the uplink port of the TAP, from which all network traffics are forwarded to the traffic analyzer PC. Wireshark dissects the timing information of the frames and is used to calculate and visualize the end-to-end delay of the time-critical frames.

5.2 Time Synchronization

This section describes the topologies and hardware setup as well as the configuration used to measure and evaluate the time-synchronization protocol specified in IEEE802.1AS.
The KPIs are:

- **Synchronization accuracy**: time error between two or more clocks from different devices in the same time domain. For highly deterministic industrial applications, the synchronization accuracy should be in the nanosecond range
- **Start-up phase**: is the time duration until the clocks are synchronized below 1 microsecond
- **Time duration** until the clocks are constantly synchronized below 100 nanosecond

The KPIs depend on the physical connection of the gPTP devices, the topology and the configuration of the gPTP profile (synchronization interval or synchronization frequency). The synchronization accuracy is measured with an oscilloscope connected to the hardware pins of the gPTP devices. For the sake of clarity, the oscilloscope is not shown in the figures.

5.2.1 Peer-To-Peer gPTP Connection

In this configuration, two devices are connected point-to-point over 1Gbps link-speed. The devices can be end-device(s) and/or TSN-switch(es).
The Network figure is shown in Figure 108.

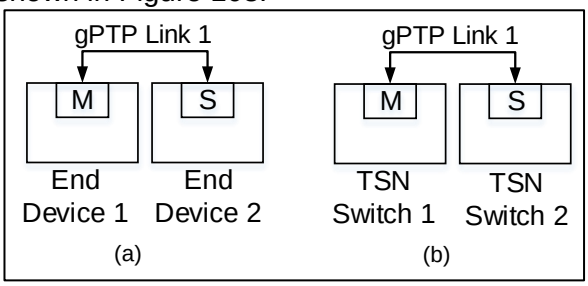


Figure 108: End device to end device time synchronization

Once connected, gPTP daemon, on each device, starts generating and analyzing the exchanged gPTP frames. The Best Master Clock Algorithm (BMCA) assigns the port state of each gPTP port (Table 64).
It is possible to manually assign a certain gPTP state to a port. Therefor the BMCA can be deactivated for certain scenarios.

Hosts	gPTP Port State	Configuration type
End Device 1	Master	BMCA activated
End Device 2	Slave	BMCA activated

Table 64: gPTP Port Configuration for end device to end device time synchronization

The gPTP daemon is executed automatically when the switches are powered-on. And the BMCA algorithm assigns the gPTP port states.

Hosts	gPTP Port State	Configuration type
TSN-Switch1 Port1	Master	BMCA activated
TSN-Switch2 Port1	Slave	BMCA activated

Table 65: gPTP Port Configuration for end to end synchronization of two TSN-switches

The results of the time synchronization is summarized in Table 66.

Scenario	Start-up phase [sec] for a sync. accuracy < 1µs	Start-up phase [sec] for a sync. accuracy < 100ns
End Device – End Device	131	199
TSN-switch – TSN-switch	< 5	< 10

Table 66: Timing error of peer to peer time synchronization

Around two minutes were required to synchronize two end-devices below 1µs and three minutes to bring it below 100nanosecond. The synchronization behavior of the TSN-switches were much better. Only five seconds for a synchronization accuracy below 1µs and five more seconds to get it constantly below 100nanoseconds.
This is due to the different gPTP daemon software and the supported clock quartz used in each device.
In order to reduce the start-up phase, the synchronization interval, which is set to 8Hz per standard, has been increased to 128Hz. The figures below shows the impact of this approach.

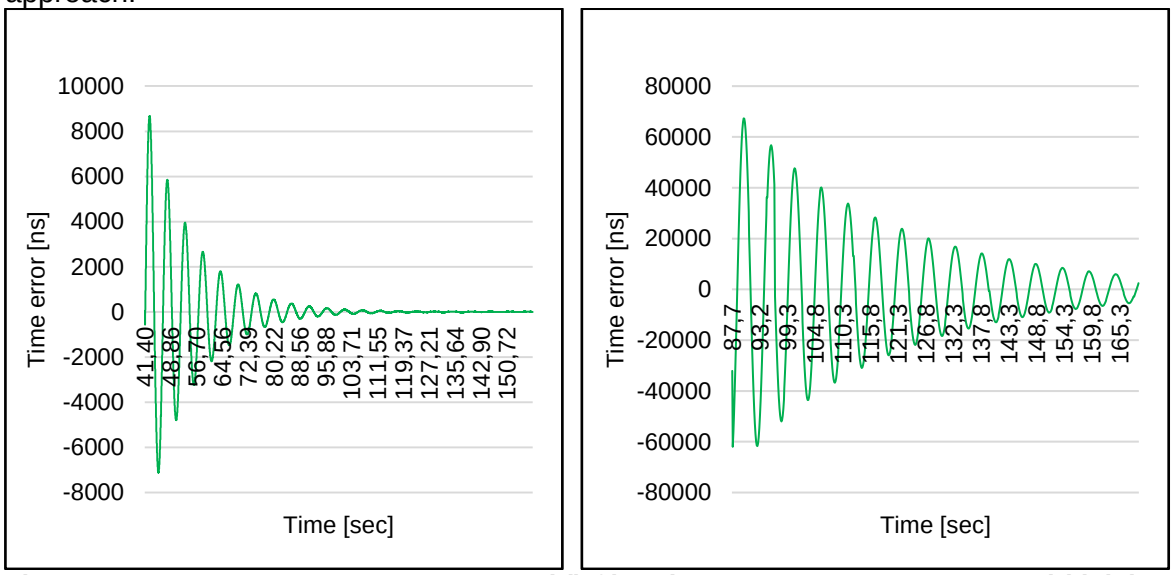


Figure 109: 128 Sync messages per second (left) and 8 Sync Messages per second (right)

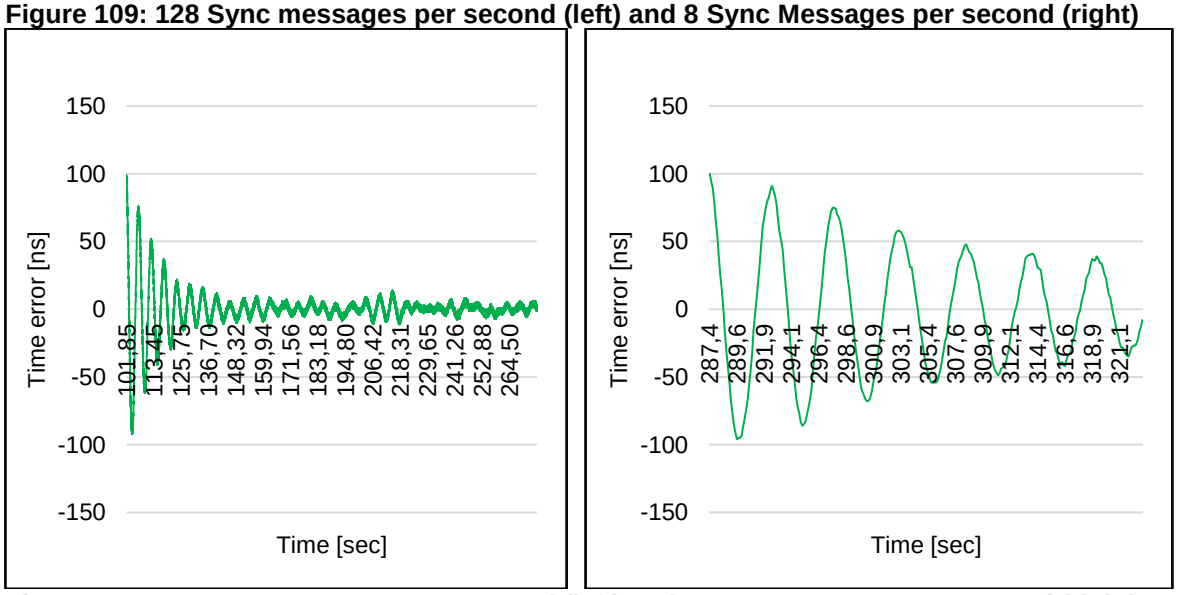


Figure 110: 128 Sync messages per second (left) and 8 Sync Messages per second (right)

5.2.1.1 Reducing the synchronization start-up phase

In automotive and industrial sectors require different synchronization accuracy. Typically higher synchronization accuracy (below 5µs) are needed in the industrial automation.
Independent of the synchronization accuracy, a control system must be synchronized before the deterministic communication starts. Therefore the start-up time duration until the clocks of a control system are synchronized is as important as the synchronization accuracy. This section shows the impact of the synchronization interval on the duration until a synchronization below 1 microsecond and 100 nanosecond is reached.
A peer-to-peer physical connection consisting of two end-devices connected over 1Gbps link-speed is used (Figure 108 (a)). In order to reduce the start-up phase, the synchronization frequency, starting from 8 Hz [56] is increased. This results into five experiments with a decreasing synchronization interval of 2^{-n} seconds, where $n \in [3, 4, 5, 6, 7]$.
Increasing the frequency means increasing the number of exchanged gPTP frames per second and thus the duration of the synchronization interval.
The results of the improvement are shown in Table 67.

n	Sync. Interval [ms]	Sync. Frequency [Hz]	Start-up phase [sec] for a sync. accuracy < 1µs	Start-up phase [sec] for a sync. accuracy < 100ns
3	125	8	131	199
4	62,5	16	67	105
5	31,25	32	34	68
6	15,62	64	31	63
7	7,81	128	28	60

Table 67: Improvement in time elapsed for time error under 1 µs

The impact of increasing the synchronization frequency on reducing the synchronization start-up phase is shown in Figure 111. The graph illustrates the behavior for a synchronization accuracy below one microsecond and below 100 nanoseconds.

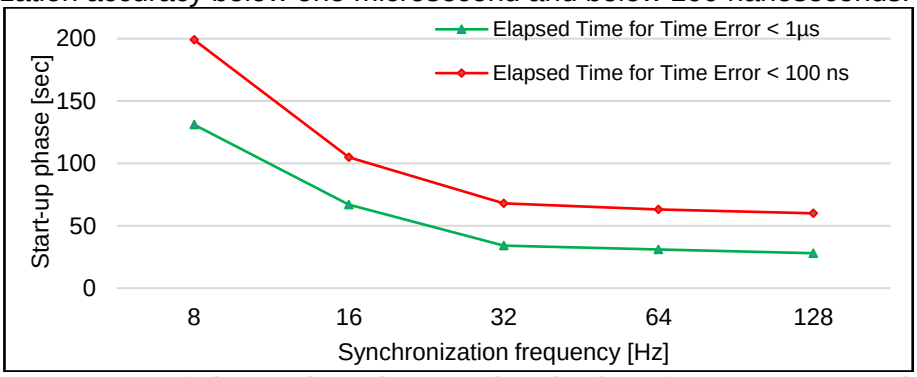


Figure 111: Impact of increasing the synchronization frequency on reducing the synchronization start-up phase for an accuracy below one microsecond and below 100 nanosecond

5.2.2 End-To-End gPTP Connection

5.2.2.1 Scenario 1 – Homogenous switches

In this section, the gPTP time synchronization is realized over a varying number of TSN-switches. The different network setups are shown in Figure 112. The network is to be called homogenous if all TSN-switches between the end-devices are from the same vendor and supporting the same gPTP implementation. For the scenarios (a) and (b) switches from vendor B are selected. For scenario (c) switches from vendor A are selected.

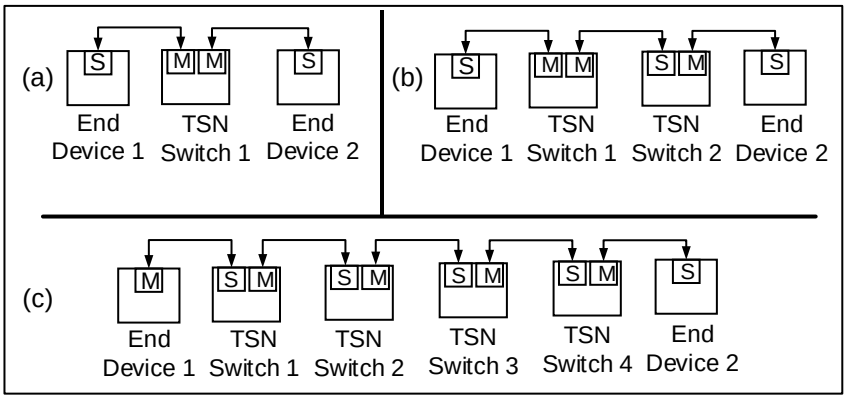


Figure 112: Time Synchronization of two end devices with homogenous TSN-switches in line topology

TSN-Switch1 is the grand-master (GM) device in Figure 112 (a) and (b). The grand-master in (c) is the end-device. The measurement points are set on the end-devices 1 and 2 for 1Gbps link-speed.The results of the time synchronization are shown in the table below.

Scenario	Start-up phase [sec] for a sync. accuracy < 1µs	Start-up phase [sec] for a sync. accuracy < 100 ns
(a) – Single TSN-switch	124.9	220.1
(b) – Two TSN-switches	123	214
(c) – Four TSN-switches	1721	1739

Table 68: Timing error of two end devices and homogenous TSN-switch in line topology

5.2.2.2 Scenario 2 –Heterogeneous switches

In this scenario, two end-devices are connected to eight heterogeneous TSN-switches (3 types) in line topology. The network setup is shown in Figure 113.

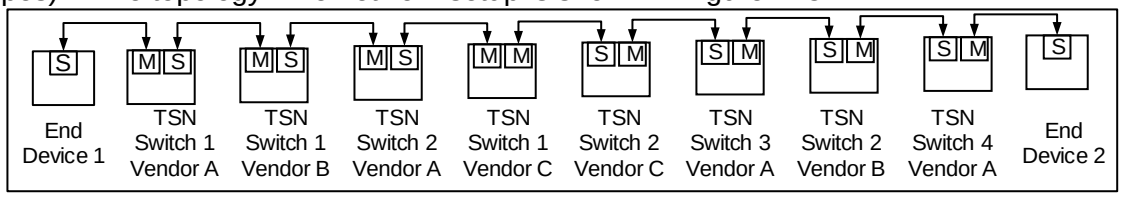


Figure 113: Time Synchronization of two end devices and four TSN-switches in line topology

TSN-Switch1 from vendor C has been selected as the grand-master (GM) in figure Figure 113. The measurement points are set on the end-devices 1 and 2 for 1Gbps link-speed. Results of the time synchronization are shown below.

Sync. Accuracy	Start-up phase [sec]
< 1 µs	741
< 300 ns	815

Table 69: Timing error of two end devices and eight TSN-switches in line topology

The clocks of end devices connected in line topology eight Switches are highly synchronized. The timing error is under ±250-300 ns between end devices.

5.2.2.3 Scenario 3 – Star Topology

In this scenario, two end-devices are connected to three TSN-switches in star topology. The network setup is shown in Figure 114.

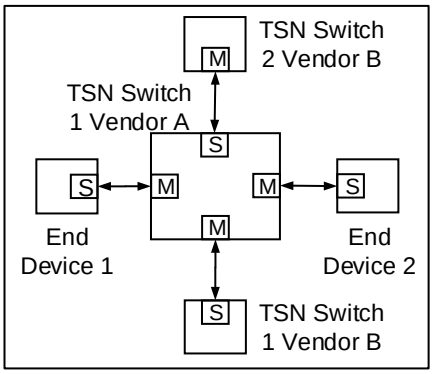


Figure 114: Time Synchronization of two end devices and three TSN-switches in star topology

TSN-Switch1 from vendor B has been selected as the grand-master (GM) in Figure 114. The measurement points are set on the end-devices 1 and 2 for 1Gbps link-speed.

The results of the time synchronization are shown in Table 70.

Sync. Accuracy	Start-up phase [sec]
< 1 µs	304
< 100 ns	399

Table 70: Timing error of two end devices and three TSN-switches in star topology

5.3 Intel® I210 Evaluation

The purpose of this sector is to evaluate the integration of i210 in the end-devices of industrial control systems to meet the deterministic requirements of the automation applications. Therefore the transmission accuracy of the i210 is tested and measured under different scenarios.
The i210 supports multiple transmission selection algorithms:

- Strict-priority algorithm
- Credit-based shaper
- Time-triggered transmit

These are evaluated and compared to each other.

5.3.1 Strict-priority Algorithm vs Credit Based Shaper

In this scenario the Strict-priority Algorithm and Credit-based transmit functionality of Intel®I210 controller are examined. This scenario is designed to investigate the use of credit-based shaper as a transmission selection algorithm in highly critical industrial applications.

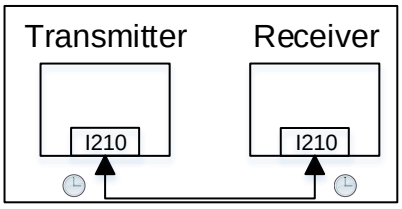


Figure 115: Network Setup for evaluation of Intel®I210 controller

The network setup for the scenario is shown in **Figure 115**. In this experiment two successive frames are sent. In the first step the credit based shaper is activated in the Intel®i210 controller. This is done by modifying the driver of the controller in the kernel image. In the next step only the time-triggered transmit is activated. The traffic configuration is shown in Table 71.

Type	Size [Bytes]	Priority	Period [ms]	No of frames	Tx start time [µs]
Scheduled	1400	7	1	2	50

Table 71: Traffic configuration for evaluation of Intel®I210 controller

The transmission duration of the frame at 100Mbps link-speed is calculated as:

$$d_{Tx} = \frac{1400 * 8}{100 \times 10^6} = 112\ \mu s$$

The transmission duration of the frame at 1Gbps link-speed is calculated as:

$$d_{Tx} = \frac{1400 * 8}{1000 \times 10^6} = 11.2\ \mu s$$

The credit is set to 0 (no bandwidth reservation with the credit-based shaper CBS). The credit becomes negative after a frame is sent. As a result, the second frame has to wait before the credit is positive again. The time taken to accumulate the credit is given by $sendSlope$ and $idleSlope$. In this scenario, they are set equal. The earliest time at which the second frame could be sent is calculated as:

$$d_{waiting} = d_{Tx} + d\_timeToPositveCredit$$

Since the rate of reduction is equal to the rate of accumulation, the time taken to reach zero credit is equal the transmission duration of frame. Hence, the waiting time is

$$d_{waiting} = \begin{cases} 112 + 112 = 224\ \mu s\ for\ 100\ Mbps \\ 12.2 + 12.2 = 24.4\ \mu s\ for\ 1\ Gbps \end{cases}$$

In case of time-triggered transmit without credit-based shaper, the waiting time of second frame is reduced only to the transmission duration of first frame.

$$d_{waiting} = d_{Tx} + d\_timeToPositiveCredit$$

$$d_{waiting} = \begin{cases} 112\ \mu s\ for\ 100\ Mbps \\ 12.2\ \mu s\ for\ 1\ Gbps \end{cases}$$

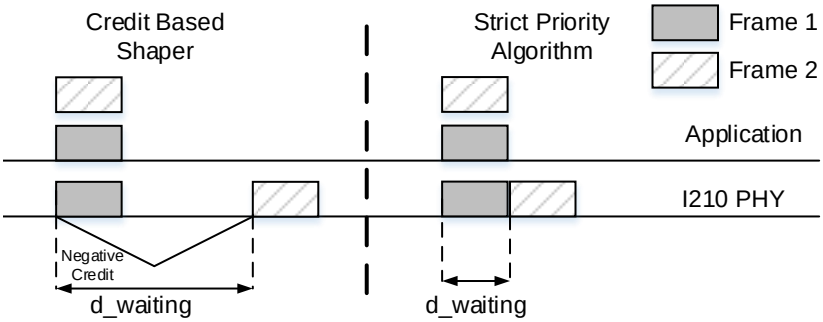


Figure 116: Frame flow diagram using Credit-based shaper

The result of the experiment with credit-based shaper is shown in Figure 117 (left) for 100Mbps link-speed and in Figure 117(right) for 1Gbps link-speed.
Credit-based shaper is used in IEEE802.1 AVB to reduce the boost, caused by the audio-video streams to fill the reception buffer. Therefore this shaper it is not suitable for highly deterministic applications with closed-loop control network.

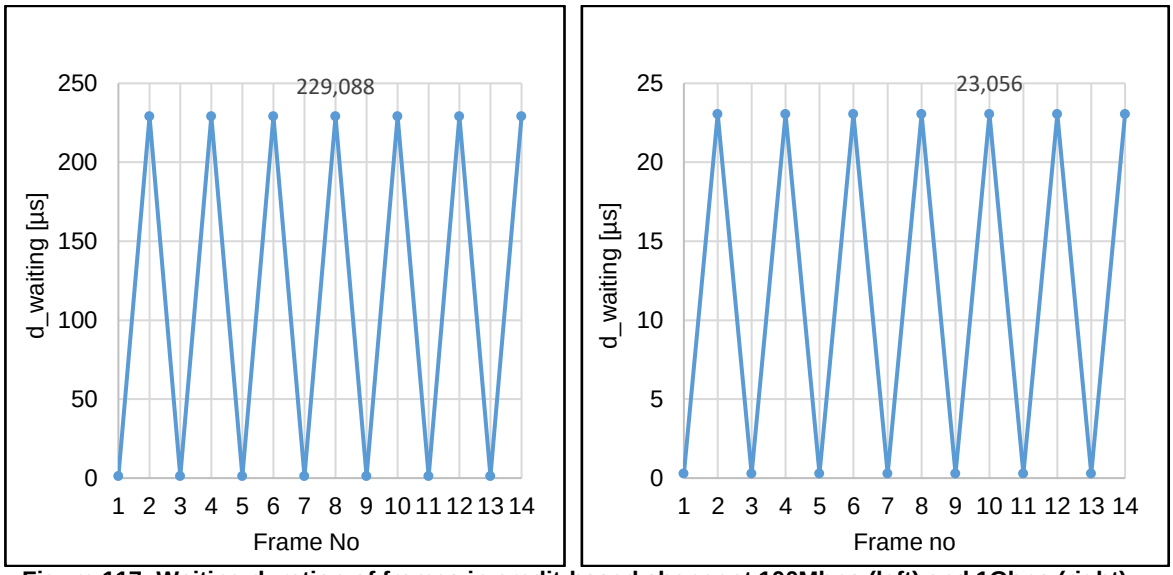


Figure 117: Waiting duration of frames in credit-based shaper at 100Mbps (left) and 1Gbps (right)

From the graphs it is clear that the second frame has to wait for double the transmission duration of the first to be eligible for transmission. This delay cannot be accepted by high-deterministic applications with very low period. Therefore, a credit-based shaper could not be used to cater to needs of time-sensitive traffics.

The time-triggered transmit behavior of Intel®I210 is evaluated as shown in Figure 118 for 100Mbps and 1Gbps respectively.

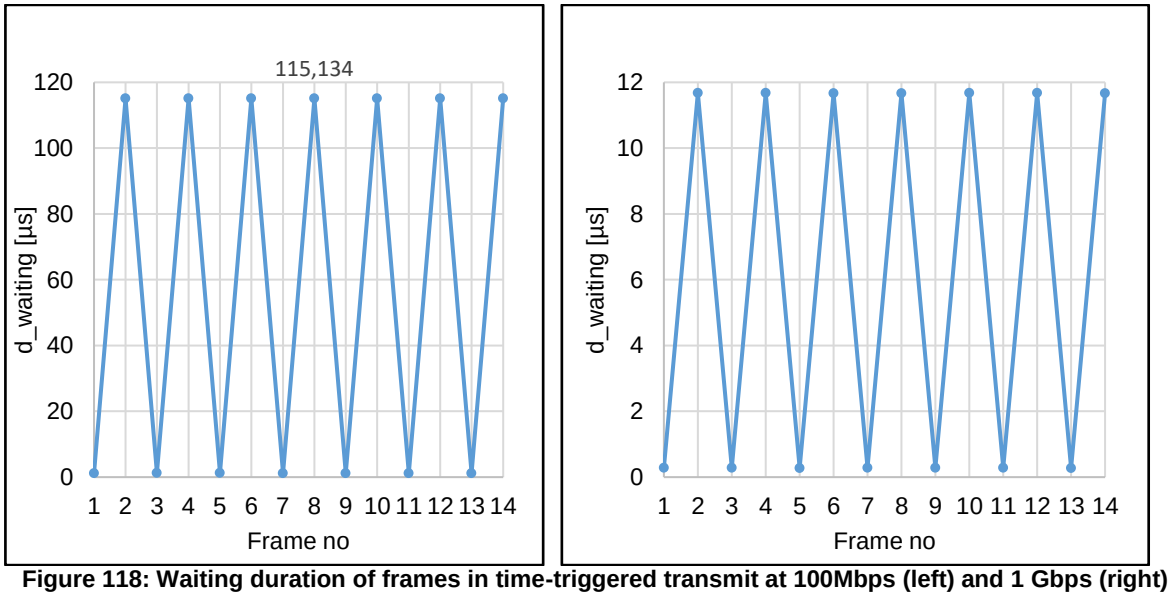


Figure 118: Waiting duration of frames in time-triggered transmit at 100Mbps (left) and 1 Gbps (right)

The graphs validates the mathematical calculations. The second frame is sent right after the first frame with an Inter-frame gap in between. Hence the frames do not incur an extra delay in comparison to the credit-based shaper.

5.3.2 Strict-priority Algorithm vs Time-Triggered Transmit

In this scenario, the strict-priority algorithm of Intel®i210 controller is evaluated against its time-triggered transmit functionality. The network setup is same as shown in Figure 115. The algorithm is activated on the controller by modifying the driver of the controller.

Traffic Configuration:

In both the modes, same traffic configuration is used. The configuration is mentioned in Table 72.

Type	Size [Bytes]	Priority	Period [ms]	Tx start time [µs]
Time-Critical	1400	7	1	500
Best-Effort	1500	2	1	500

Table 72: Traffic configuration for evaluation of strict-priority algorithm and time triggered transmit of Intel®I210 controller

Only one time-critical frame is sent per cycle. The best-effort frames are sent to introduce a network load. The count of frames and the network load is different for 100Mbps and 1Gbps.
It is listed in Table 73.

Datarate	No of Frames	Network Load
100Mbps	7	72 %
1Gbps	60	84 %

Table 73: Number of Best-Effort frames and network load for evaluation of strict-priority algorithm and time-triggered transmit of I210 controller

Evaluation Metrics:
In this experiment, the delay incurred by the frame after it is sent by the application to the controller and before it is finally time-stamped by the MAC of the controller is measured. The jitter observed in this delay is used to evaluate the readiness of i210 controller in the deterministic control applications.

Experimental Analysis:

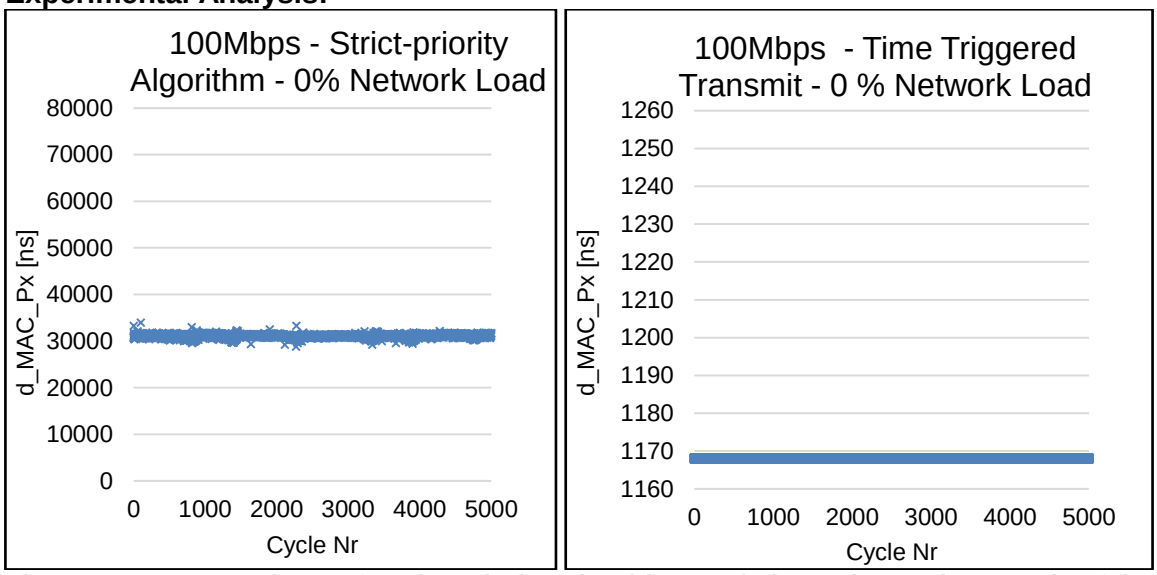


Figure 119: Comparison of Strict-priority algorithm and time triggered transmit of i210 controller at 100Mbps without network load

In strict-priority algorithm, the frames are sent as soon as they are received by the controller from the application. In Figure 119 (left), although the processing time is high, it is deterministic with very low jitter. In Figure 119 (right), the frames are sent in time-triggered mode and they are given a predefined launch-time. The delay between the launch-time given by the application and the time-point at which the frame has been stamped at the frame's egress from I210's MAC is shown in the graph. The delay as well as the jitter is highly deterministic, making Intel®I210 controller suitable for integration in deterministic networks.

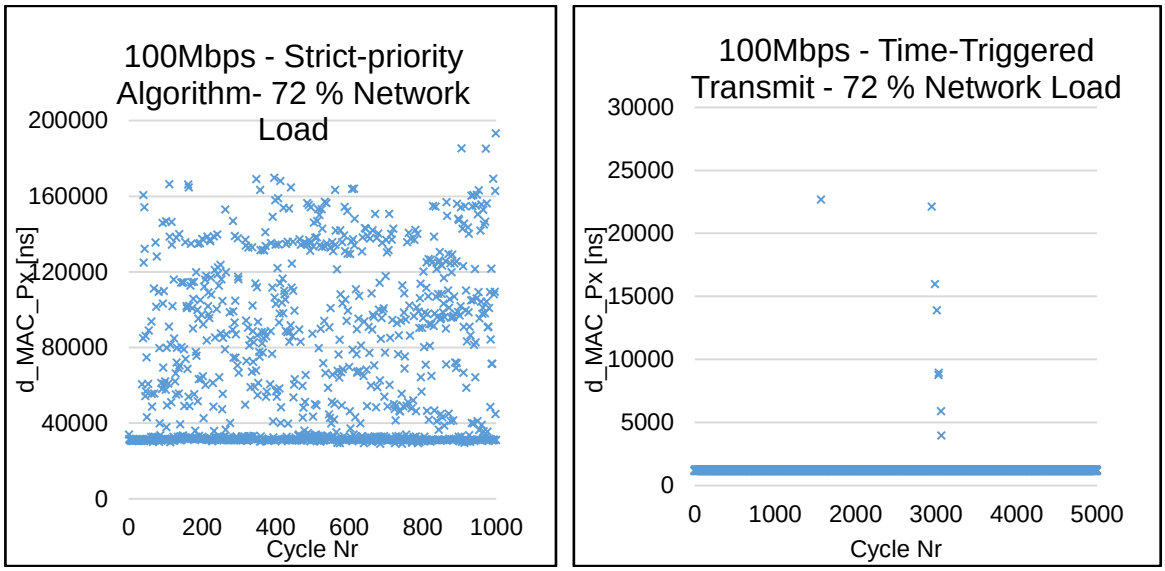


Figure 120: Comparison of Strict-priority algorithm and time triggered transmit of i210 controller at 100Mbps with 72 % network load

Figure 120 shows the behavior of the Intel®i210 controller under 72% network loads. In the strict-priority mode, the controller is used to transmit the time-critical and non-time-critical frames. Since no launch-time is given, the transmission time point of the time-critical frames is not protected, which causes them to be delayed. In Figure 120(right), the frames are sent with a launch-time. In this case, the transmission start time of the time-critical frames is protected by a guard band and the frames have a deterministic latency and jitter within the scheduled transmission queue. Because of the launch-time, the time-critical frames are called time-triggered.
The transmission of a non-time-triggered frame is to be delayed if it does not finish before the scheduled launch-time of a time-triggered frame. The non-time-triggered frame is than selected for transmission following its priority.

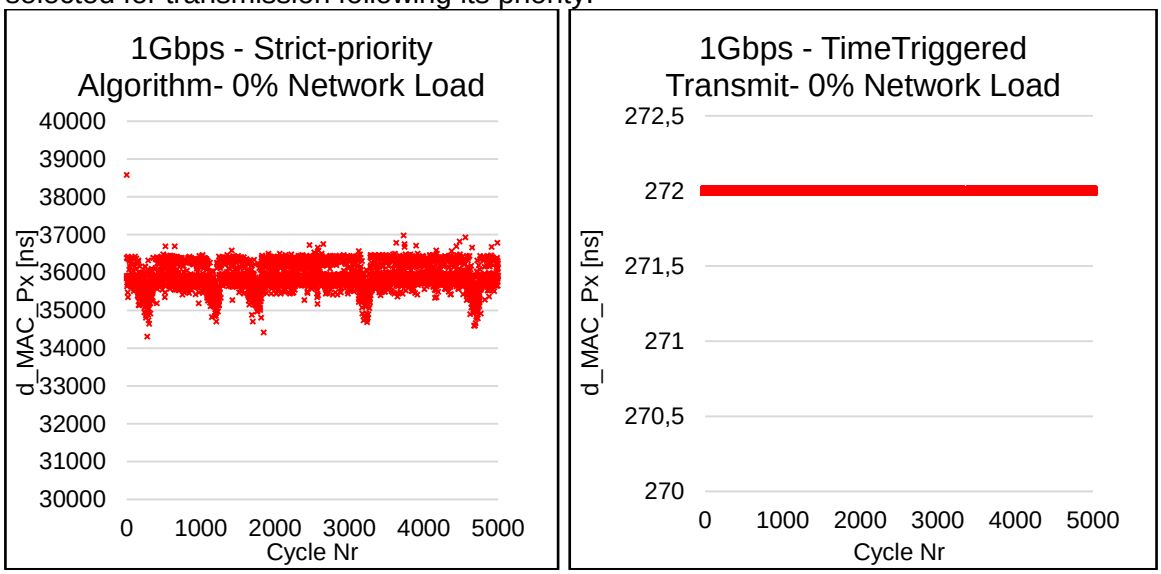


Figure 121: Comparison of Strict-priority algorithm and time triggered transmit of i210 controller at 1Gbps with 0 % network load

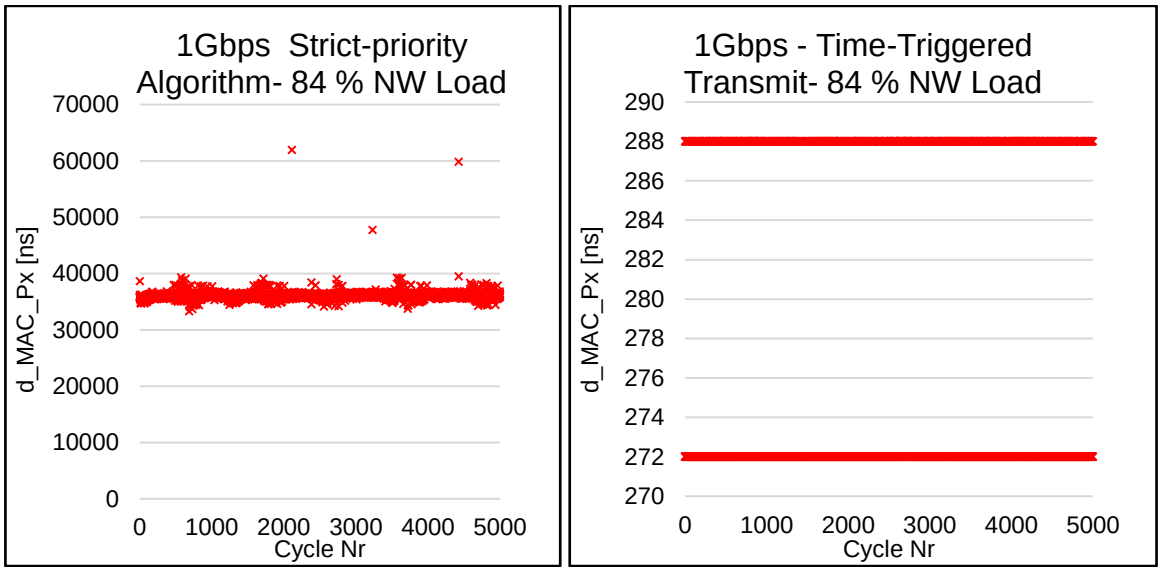


Figure 122: Comparison of Strict-priority algorithm and time triggered transmit of i210 controller at 1Gbps with 84 % network load

Figure 121 and Figure 122 show the two transmission modes of i210 controller: strict-priority algorithm and time-triggered transmit, for 1Gbps link-speed. The processing time of the frame is considerably reduced for 1Gbps link-speed but the jitter is not altogether reduced in the strict-priority algorithm. Even at 1Gbps, a deterministic processing latency is achieved in the time-triggered transmit mode.

5.4 Traffic Shaping

This section investigates the timing behavior of the traffic shapers specified in IEEE802.1Q, IEEE802.1Qbu / IEEE802.3br and IEEE802.1Qbv. Therefor different scenarios have been covered. Each scenario has a set of configuration parameters:

- Topology: line or tree
- link-speed: 100Mbps or 1Gbps
- frame forwarding mechanism: Store&Forward or cut-through
- Number of forwarding hops
- network load: used bandwidth by the non-time-critical traffic(s)

The configuration parameters influence the traffic shaper's KPIs:

- end-to-end delay; given in microsecond [µs]
- end-to-end delay variation over time (referred as jitter); given in microsecond [µs] and
- use of bandwidth; given in [%]

The KPIs are measured by time-stamping the Ethernet frames using a network TAP and the i210 network controllers on the transmitting and receiving end-devices. The results of each scenario are shown in a tabular and graphical form.

Network configuration:

Link-speed	100Mbps / 1Gbps
Frame Forwarding Mechanism	Store&Forward
Topology	Line
Number of forwarding hops	2 or 4
Network Load	0% -25%- 50%-75%-100%

Table 74: Network configuration

5.4.1 Scenario1 - Two forwarding Hops

Network description

The network is composed of five communicating devices: three end-devices and two forwarding hops. The end-devices are: time-critical sender, non-time-critical sender and a receiver. The forwarding hops are the TSN-switches. A TAP has been used to measure the single forwarding and processing delay of the TSN-switches. It is not used for the end-to-end delay measurement. The i210 network controller on the sender- and receiver-devices are used to calculate the end-to-end delay.

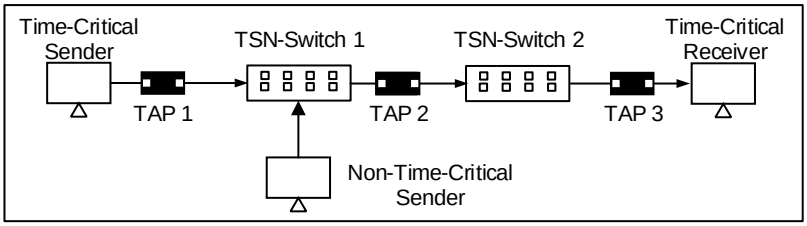


Figure 123: Network Setup for TSN Evaluation over 2 hops

Traffic configuration

The time-critical sender generates and transmits a time-critical 1400Byte frame with the highest priority PCP7 each millisecond. This results into 1000 frames per second. The path of the time-critical traffic is: Time-critical sender – TSN-Switch1 – TSN-Switch2 – Time-Critical Receiver.

The non-time-critical sender generates and transmits a 1514Byte time-critical frame with the priority PCP2 as a stream. The path of the time-critical traffic is: Non-time-critical sender – TSN-Switch1 – TSN-Switch2 – Non-Time-Critical Receiver.

For purpose of clarity, the Non-Time-Critical receiver device is not shown in the figure.

Traffic Type	Size [Byte]	Transmission Interval [ms]	Priority
Time-Critical (TC)	1400	1 [27]	7
Non-Time-Critical (NTC)	1514	Stream[28]	2

Table 75: Traffic configuration for TSN Evaluation over 2 hops

Results

This section shows the measured KPIs: end-to-end delay, jitter and use of bandwidth under different combinations of the configuration parameters shown in the table above.

End-To-End delay for Fast-Ethernet link-speed (100Mbps)

The measurement results of the end-to-end delay for the time-critical traffic TC for a constant link-speed 100Mbps and network load 96% (~100%) strongly depends on the activated traffic shaper. The high variation over time is due to the interferences with the background traffic. The maximum interference is given by the transmission duration of a single background frame and the following inter-frame gap IFG. For 100Mbps link-speed this would be around 122µs.

The results varies from ~252µs to ~379µs using strict-priority algorithm. This results into a huge jitter. The upper-bound end-to-end delay is strongly reduced to ~260µs using frame-preemption and to 252µs using time-aware shaper. The delay variation is still observed with the frame-preemption approach. This is due to the transmission duration of an NTC-fragment or the rest of the transmission an NTC-frame that cannot be preempted anymore. E.g. because the resulting fragment has a size below 64Byte (minimum Ethernet frame size).

The minimum value (252µs) of the end-to-end delay is given if TC does not interfere with the background traffic NTC. It is equal to the transmission duration (112µs) of a single TC-frame at TSN-switch1 and TSN-switch2 plus their internal processing delay (14µs each).

[27] Cycle time = 1ms , Reserved time slot = 200µs for 100Mbps and 1Gbps , Guard-Band = 1542Byte

[28] The non-time-critical traffic is not cyclic and is transmitted as a stream.

The maximum value (380µs) is given by the full interference.

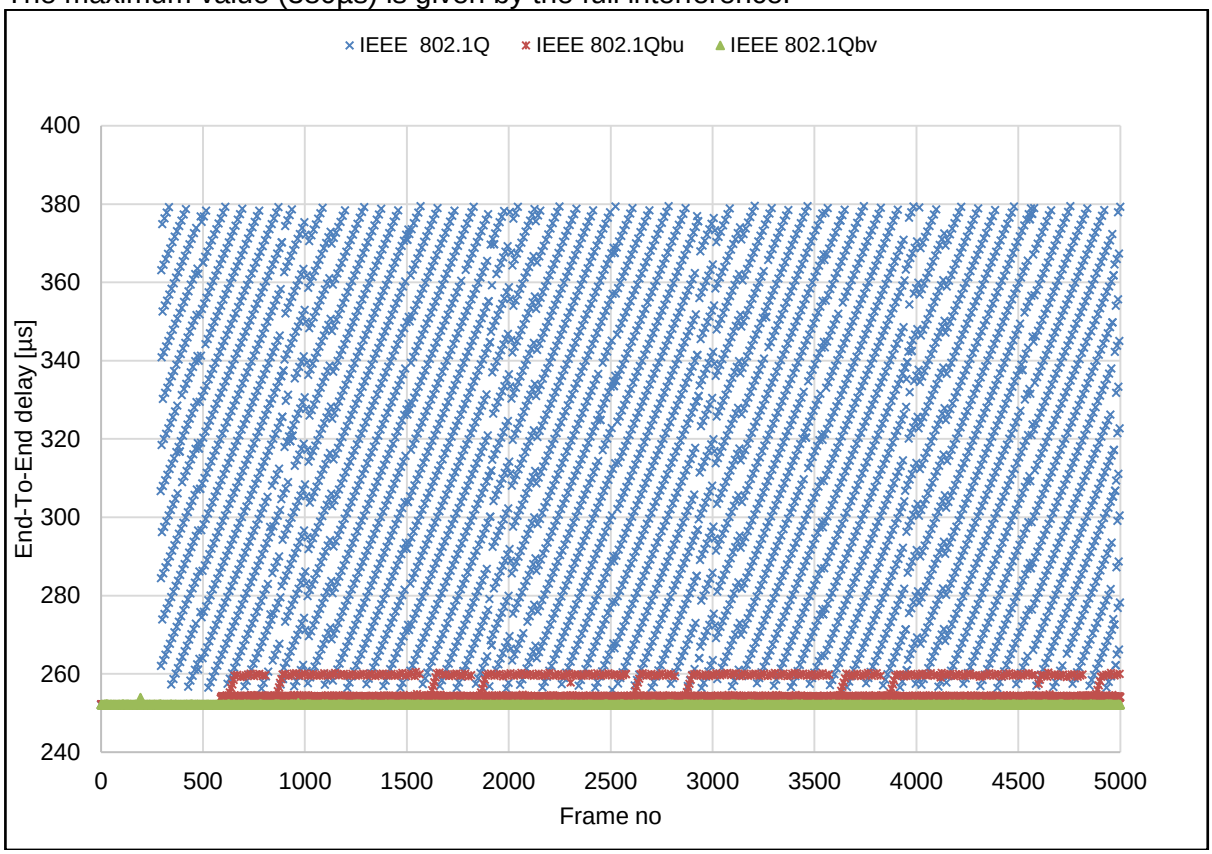


Figure 124: Behavior of the End-to-End delay over two TSN-switches for a data-traffic at 100Mbps and ~100% Network load using different traffic-shapers: (blue) strict-priority IEEE8020.1Q, (red) frame-preemption IEEE802.1Qbu and (green) time-aware shaper IEEE 802.1Qbv

In order to cover the end-to-end delay of TC traffic under an increasing network load for all transmission selection algorithms (TSA) 15 experiments were required. That means five experiments per TSA. Each experiment has a sample duration of 50 seconds. The results are plotted in figures below.

Before running the experiments, it was expected to get no difference between the minimum (min), average (avg) and maximum (max) values for 0% network load. The measurements shows that the TSN-switches does not perform well enough. Depending on the activated TSA, the difference between min and max is between 4µs and 10µs at 0% network load.

Further increasing the network load has almost no impact on the max end-to-end delay values for the strict-priority algorithm.

The time-aware shaper shows some bad behavior. Since it was expected to have an almost constant end-to-end delay in the sub-microseconds range. However, the overlapping lines of the average and minimum end-to-end delay proves that for most samples (50.000 samples in total) the time-aware shaper was deterministic. Anyway this behavior cannot be tolerated for certain time- and safety-critical applications. For most deterministic applications the timing behavior of the time-aware shaper used in these experiments is acceptable.

In general the experiments proved that increasing the network load increases the end-to-end delay. This behavior can clearly be observed with the average curve in all three graphics and is caused by early prototype implementation.

The timing behavior of the frame-preemption approach fulfils the timing requirements of most deterministic applications.

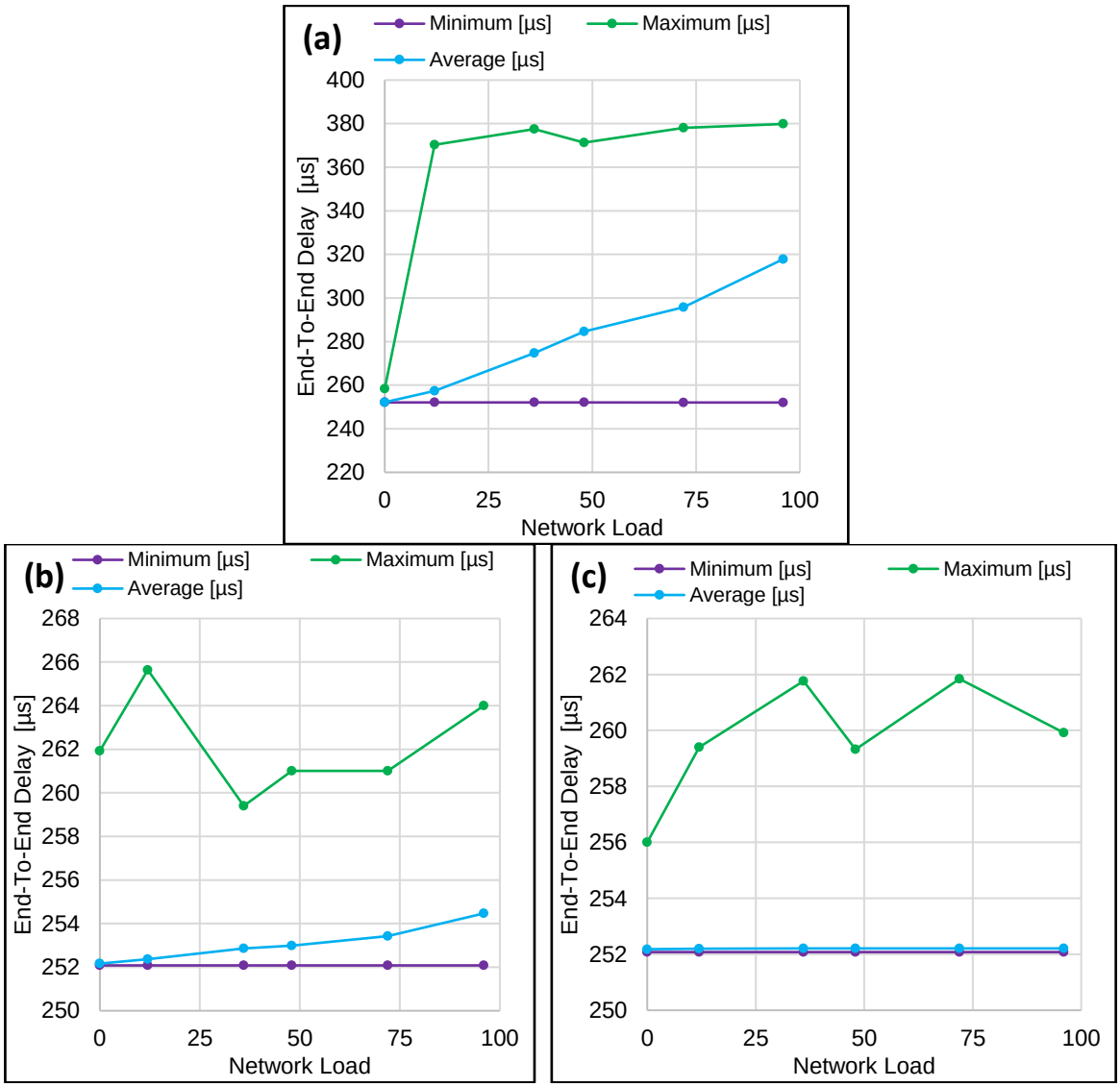


Figure 125: Behavior of the End-to-End delay for a data-traffic at a constant link-speed 100Mbps and increasing network-load from 0% to ~100% using:
(a) strict-priority algorithm IEEE802.1Q (b) frame-preemption IEEE8020.1Qbu and (c) time-aware shaper

End-To-End delay for Gigabit-Ethernet link-speed (1Gbps)

Unlike the end-to-end delay for 100Mbps, the difference between the performances of the three traffic shapers is strongly decreased with Gigabit-Ethernet. The higher link-speed reduces the transmission- and interference-durations by a factor of 10. This results into much shorter end-to-end delay and much smaller differences between the min and max values.

As expected the best behavior is given by the time-aware shaper. Because of its simplicity compared to the scheduling approach of time-aware shaper, the frame-preemption mechanism can be favorited in combination with Gigabit-Ethernet. The difference between all traffic-shapers is expected to be more negligible with higher link-speed e.g. 10Gbps.

The values are given for an almost constant network-load of ~100%. Three experiments were required. Each experiment is performed for a single traffic-shaper with 5000 samples.

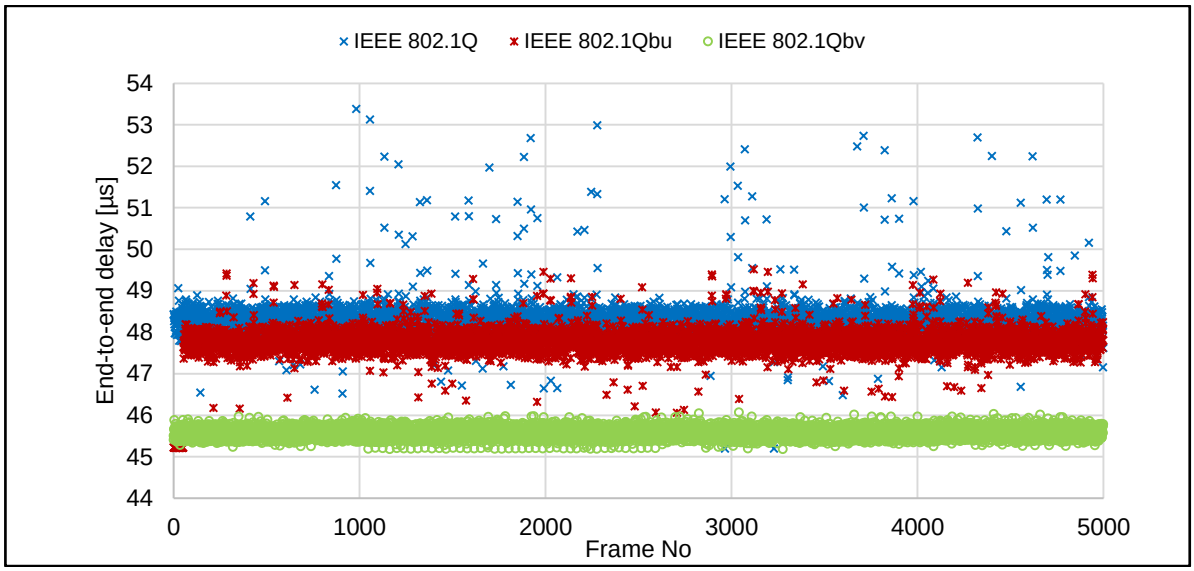


Figure 126: Behavior of the End-to-End delay for a data-traffic at 1Gbps and ~100% Network load using different traffic-shapers: (blue) strict-priority IEEE8020.1Q, (red) frame-preemption IEEE802.1Qbu and (green) time-aware shaper IEEE 802.1Qbv

The impact of increasing the link-speed by a factor of 10, on improving the timing behavior, can be clearly observed for all traffic shapers (Figure 127 below). The most profiting shaper is the strict-priority algorithm. The difference between the min and max values is decreased by ~110µs from ~130µs to ~20µs. Unlike with 100Mbps, the average curve is slowly increasing with higher network-load.

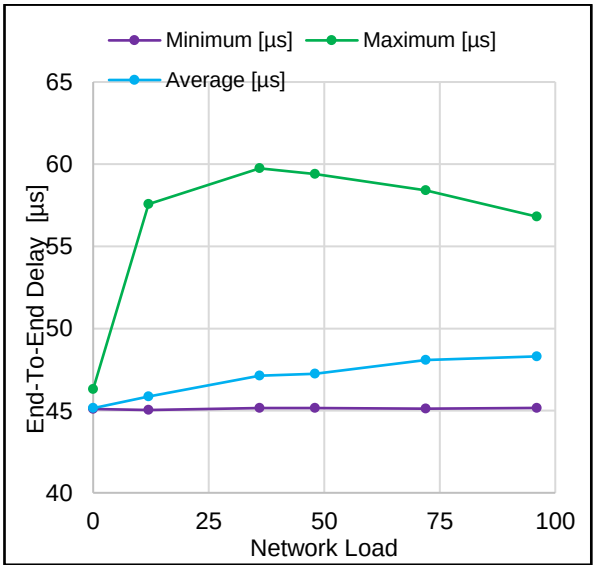


Figure 127: Behavior of the End-to-End delay for a data-traffic at a constant link-speed 1Gbps and increasing network-load from 0% to ~100% using strict-priority algorithm IEEE8020.1Q

The timing performance of the time-aware shaper supported by the TSN-switches, taken under test, is worsen by increasing the link-speed. This is already observed with 12% network-load. The min and max curves of the end-to delay are no more overlapping as with 100Mbps.
The frame-preemption approach shows almost the same behavior as with 100Mbps.
In general it is shown that the strict-priority algorithm is the most profiting traffic shaper from the increased link-speed. Its timing performance is very similar to the one of frame-preemption.

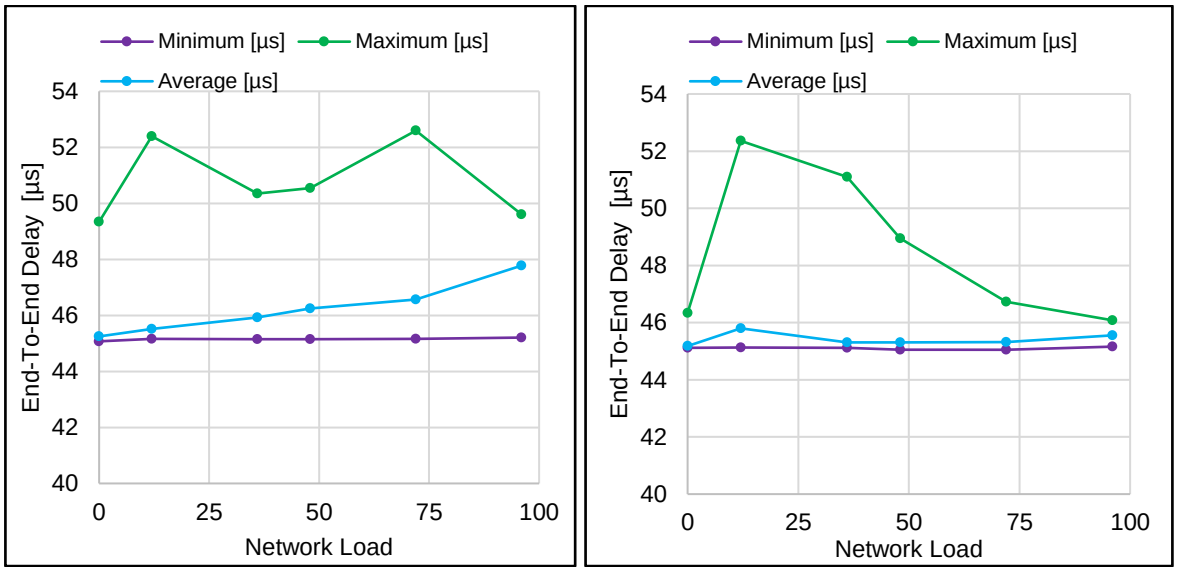


Figure 128: Behavior of the End-to-End delay for a data-traffic at a constant link-speed 1Gbps and increasing network-load from 0% to ~100% using frame-preemption IEEE8020.1Qbu (left) and time-aware shaper (right)

Summary results of end-To-end delay

The measurements results of the end-to-end delay for six experiments are shown in Figure 129 below. The experiments are given by different combinations of the link-speed and the traffic shaper. E.g. strict-priority algorithm IEEE802.1Q with 100Mbps.

The figure demonstrates the impact of the activated transmission selection algorithm (TSA) and the link-speed on the end-to-end delay.

The best results are reached with Gigabit-Ethernet independently of the activated transmission selection algorithms. This proves that the timing benefits of the scheduling approach and the frame-preemption mechanism are restricted to frame transmission over multiple forwarding hops and/or for link-speed below 10Gbps.

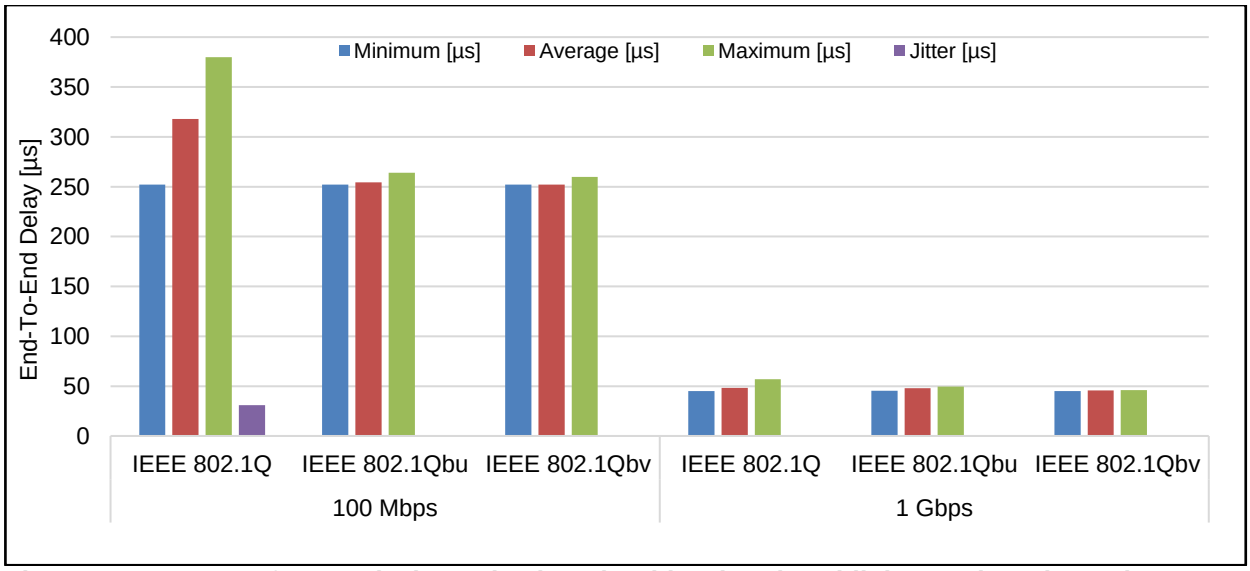


Figure 129: Impact of transmission selection algorithm (TSA) and link-speed on the end-to-end delay over two TSN-switches and 96% network load

Jitter

The measurement results for 100Mbps and 1Gbps link-speed and increasing network load are shown below (figure left). For the purpose of clarity the figure is zoomed the axis is reduced from 35µs to 3µs (figure right).

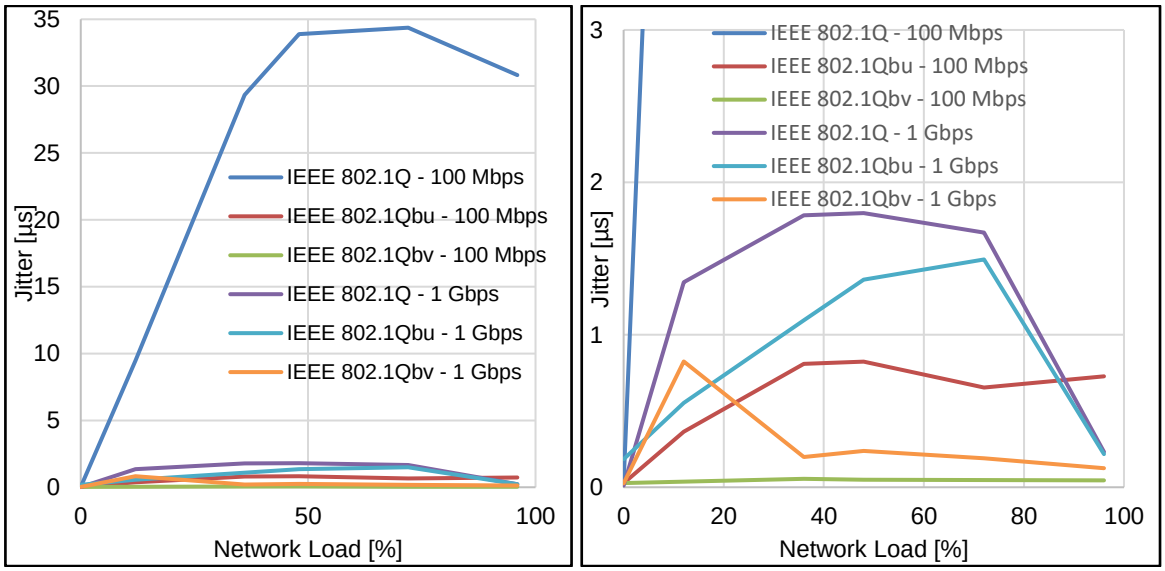


Figure 130: Jitter under increasing Network Load – 2 hops

Use of Bandwidth

Guaranteeing an upper bound for the end-to-end delay and an ultra-low jitter are specific requirements for deterministic applications e.g. industrial motion-control. But the price should not be the loss of the bandwidth. E.g. (1) reserving much bigger time-slot than the transmission duration of the scheduled frame(s). (2) if the transmission of a non-scheduled frame cannot be finished within the guard-band window, used to protect the scheduled traffic from any arbitrary additional delay, is also a reason for more loss of bandwidth.

An optimal use of the bandwidth is an important KPI for a successful IT/OT network convergence. Therefore, it is measured and is considered as a relevant acceptance criterion for the different traffic-shapers and under different configuration parameters e.g. 100Mbps and 1Gbps link-speed.

In order to measure the use of bandwidth, the non-time-critical traffic is transmitted as a burst from the transmitter to the receiver and shares the physical link between the TSN-switches with the time-critical traffic. Depending on the activated communication features: link-speed and traffic-shaper, different results are observed.

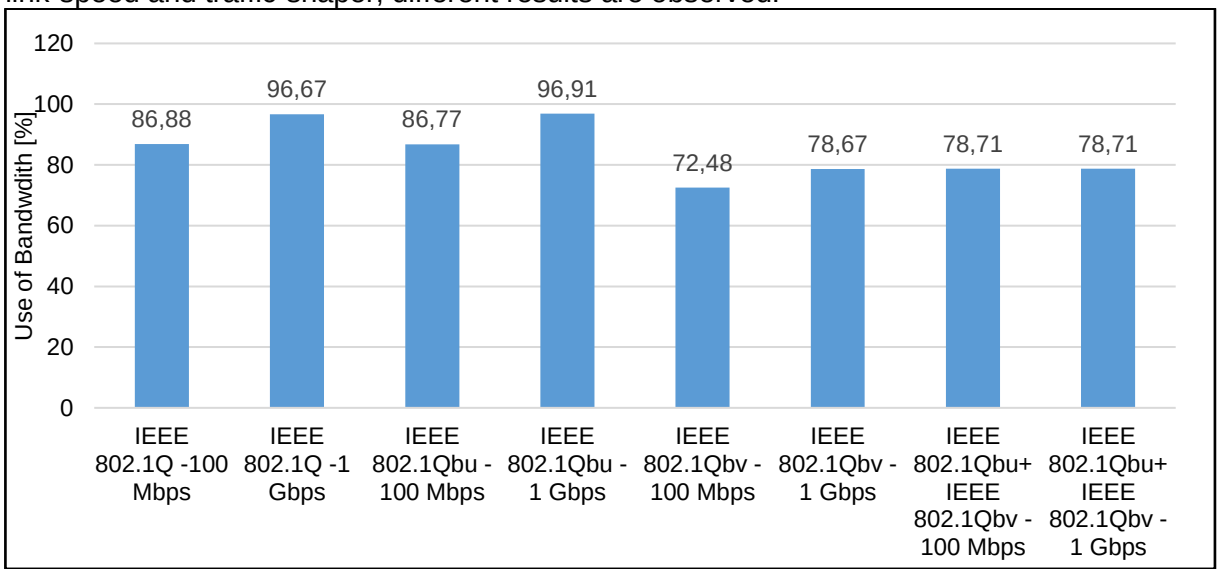


Figure 131: Throughput of all TSAs defined in IEEE802.1Q, IEEE802.1Qbu and IEEE802.1Qbv

Eight experiments are performed. No packet loss is observed for the time-critical frames. The measurement shows different throughput for the each experiment.

The best bandwidth use is reached with Gigabit link-speed in combination with strict-priority or frame-preemption. Almost 96% of the network bandwidth could be used. The value decreases to fast 87% with Fast-Ethernet link-speed 100Mbps.

The worst use of bandwidth is given by the time-aware shaper. A time-slot of 200µs each 1millisecond cycle time is reserved to transmit 1400Byte scheduled frame independently of the link-speed. The high frequency of the time-slot 1000x/sec and the guard-band are the main reason for the loss of bandwidth. The results shows that the configuration for 1Gbps link-speed is not effective, since no more than 20µs should be reserved to transmit the scheduled frame.
In order to benefit from the high timing performance (delay and jitter) of time-aware shaper and to improve its use of bandwidth, the frame-preemption is activated in the last two experiments (right). The combination led to 6% improvement for use of bandwidth.

Summary
The impact of the time-aware shaper and the frame-preemption depends on several parameters, such as the link-speed (100Mbps or 1Gbps), the forwarding mechanism (store&forward, cut-through) and the number of forwarding hops between the transmitter and receiver end-devices. The last configuration parameter was set to two (TSN-switches). This restricted the performance comparison between the traffic shapers.
For a more fair and real comparison, the scenario is repeated for the double number of TSN-switches. The next section shows the performance of the traffic shapers using four forwarding hops between the transmitter and receiver end-devices.

5.4.2 Scenario2 – Four forwarding Hops

Unlike the previous scenario, this scenario increases the number of forwarding hops from two to four TSN-switches. Its impact on the resulting timing behavior is analyzed and compared to the previous scenario.

Network description

Sceanrio2 is an extension of scenario1. In order to investigate the impact of increased number of forwarding hops, two TSN-switches have been added. The Time-Critical (TC) traffic generated and transmitted cyclically from the TC-sender is now forwarded over four TSN-switches to the receiver device. Depending on the activated traffic shaper, the TC-frames could interfere with the background traffics transmitted from NTC-Senders 1, 2 and 3.

A TAP has been used to measure the single forwarding- and processing-delays of the TSN-switches. Because of the jittering TAP forwarding delay [500ns to 600ns] the TAP is removed for the end-to-end delay measurement. The i210 network controller on the sender- and receiver-devices are used to calculate the end-to-end delay, which is given by the difference between the time-stamp for the same frame at its transmission- and reception-time-points.

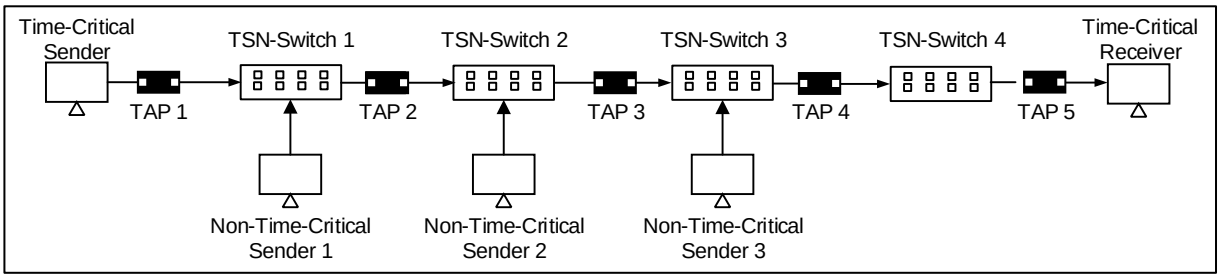


Figure 132: Network Setup for TSN Evaluation over 4 hops

Traffic configuration

The configurations of the TC-traffic and NTC-traffics are similar to the one described in section 5.4.1. The only difference is the additional two NTC-traffics.

Traffic Type	Size [Byte]	Transmission Interval [ms]	Priority
Time-Critical (TC)	1400	1	7
Non- Time-Critical1 (NTC1)	1514	Stream[29]	2
Non- Time-Critical2 (NTC2)	1514	Stream	2
Non- Time-Critical3 (NTC3)	1514	Stream	2

Table 76: Traffic configuration for TSN Evaluation over 4 hops

Results

This section shows the measured KPIs: *end-to-end delay*, *jitter* and *use of bandwidth* under different combinations of the configuration parameters shown in the table above.

End-To-End delay for Fast-Ethernet link-speed (100Mbps)

The measurement results of the end-to-end delay for the time-critical traffic TC, a constant link-speed 100Mbps and network load 96% (~100%) strongly depends on the activated

[29] The non-time-critical traffic is not cyclic and is transmitted as a stream.

traffic shaper. Three traffic shapers are activated: non-preemptive strict-priority algorithm (IEEE802.1Q), preemptive strict-priority algorithm (IEEE802.1Qbu, IEEE802.3br) and time-aware shaper (IEEE802.1Qbv).
The highest possible end-to-end delay (~996µs) using IEEE802.1Q is given by the transmission duration of the time-critical frame over four TSN-switches (114µs*4), the sum of the switches internal processing delays (14µs*4) and the interference delay on each forwarding process (121µs*4). The maximum measured delay over 5000 samples is ~822µs. The theoretical maximum value of ~996µs is hard to reach because it assumes four consecutive and full interference durations on each forwarding hop.
Similar to scenario1 with two forwarding hops, activating the frame-preemption or the scheduling mechanism instead has a huge impact on reducing the resulting end-to-end delay, which is kept below 518µs. There is no considerable difference between the timing performances of IEEE802.1Qbu and IEEE802.1Qbv for less number of forwarding hops between the transmitting and the receiving end-devices.

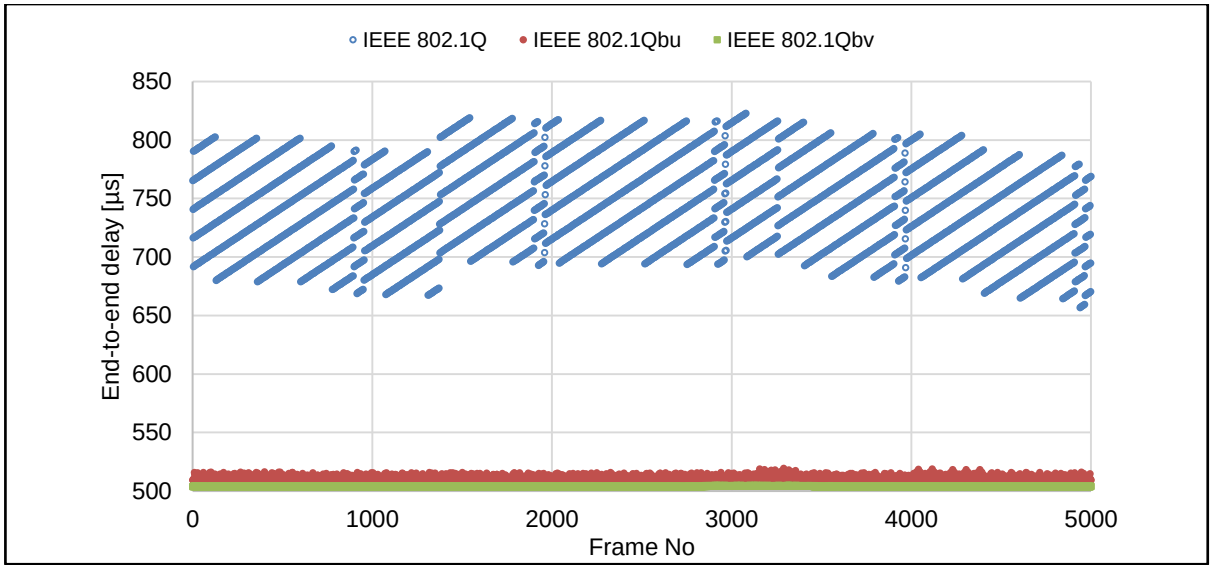


Figure 133: Behavior of the End-to-End delay over four TSN-switches for 100Mbps and ~100% Network load using different traffic-shapers: (blue) strict-priority IEEE8020.1Q, (red) frame-preemption IEEE802.1Qbu and (green) time-aware shaper IEEE 802.1Qbv

In order to cover the end-to-end delay of TC traffic under an increasing network load for all transmission selection algorithms (TSA), 15 experiments were required. That means five experiments per TSA. Each experiment has a sample duration of 5000 cycles ~50 seconds. The results are plotted below.
Unlike the previous scenario1 with two forwarding hops, there is no considerable difference anymore between the minimum (min), average (avg) and maximum (max) end-to-end delay for 0% network load. This should be kept constant also for different traffic-shapers since there is no interference with the non-existing background traffic.
Increasing the network load from 0% to ~100% increases the maximum and average values of the end-to-end delay for the non-preemptive as well as the preemptive strict-priority algorithms. While the average delay increases from 520µs to ~671µs at ~100% network with IEEE802.1Q, it is kept almost the same ~510µs with IEEE802.1Qbu and IEEE802.3br.
Similar to the previous scenario with two forwarding hops, the time-aware shaper keeps showing a bad timing behavior. Instead of keeping the end-to-end delay almost constant in the sub-microseconds range, the maximum end-to-end delay showed certain “jumps” e.g. from 504µs to ~509µs at 25%; 50% and 75% network loads. This behavior cannot be tolerated for certain time- and safety-critical applications. For most deterministic applications the timing behavior of the time-aware shaper used in these experiments is acceptable.

In general the experiments proved that increasing the network load increases the end-to-end delay. This behavior can clearly be observed with the average curve in all three graphics.
The timing behavior is caused by the used switches in the experiment, since these are from different vendors.

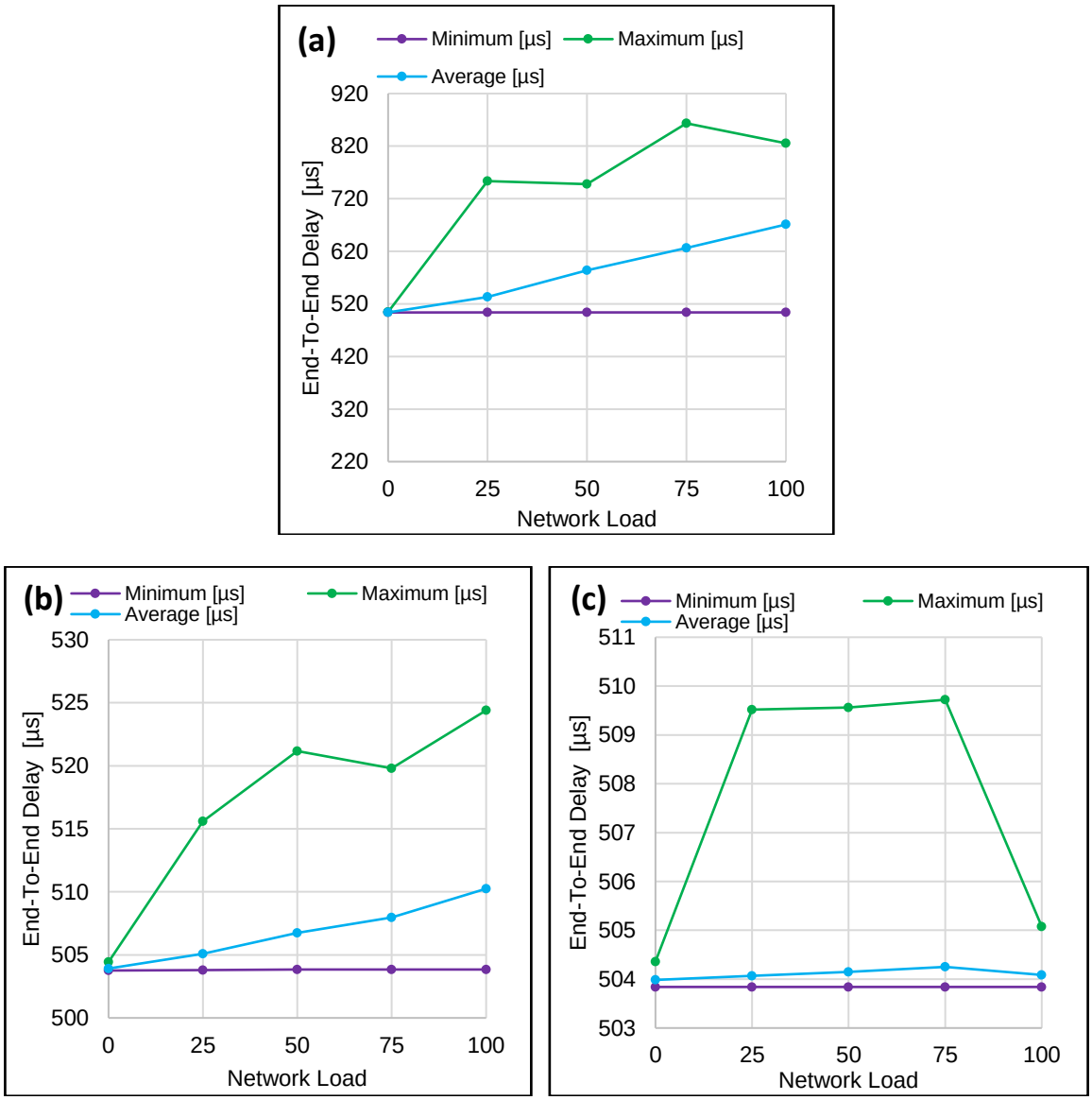


Figure 134: Behavior of the End-to-End delay over four TSN-switches for a data-traffic at a constant link-speed 100Mbps and increasing network-load from 0% to ~100% using: (a) strict-priority algorithm IEEE802.1Q (b) frame-preemption IEEE8020.1Qbu and (c) time-aware shaper

End-To-End delay for Gigabit-Ethernet link-speed (1Gbps)

Unlike the end-to-end delay for 100Mbps, the difference between the performances of the three traffic shapers is strongly decreased with Gigabit-Ethernet. The higher link-speed reduces the transmission- and interference-durations by a factor of eight. This results into much shorter end-to-end delay and much smaller differences between the minimum and maximum values of the end-to-end delays.
Also for four forwarding hops, the frame-preemption in combination with Gigabit Ethernet approach can be clearly favorited over the scheduling approach with time-aware shaper. Not only because of its simple configuration but also for its deterministic timing behavior. However the simulation showed that increasing the number of forwarding hops has a huge impact on the difference between the two approaches. The time-aware shaper shows much bette timing behavior for 50 forwarding hops or more.

The values are given for an almost constant network-load of ~100%. Three experiments were required. Each experiment is performed for a single traffic-shaper with 5000 samples.

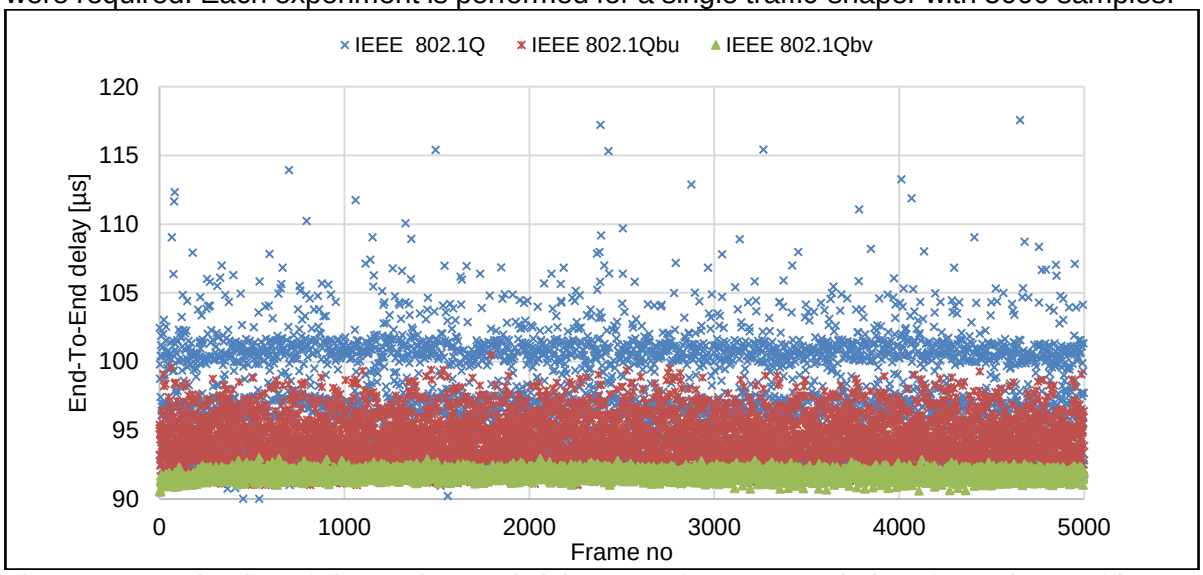


Figure 135: Behavior of the End-to-End delay over four TSN-switches for a data-traffic at 1Gbps and ~100% Network load using different traffic-shapers: (blue) strict-priority IEEE8020.1Q, (red) frame-preemption IEEE802.1Qbu and (green) time-aware shaper IEEE 802.1Qbv

The impact of increasing the link-speed (from 100Mbps to 1Gbps) on improving the timing behavior can be clearly observed for all traffic shapers (Figure 136 below). The difference between the minimum and maximum end-to-end delays for IEE802.1Q at ~100% network-load is decreased by ~700µs from ~820µs to ~117µs.

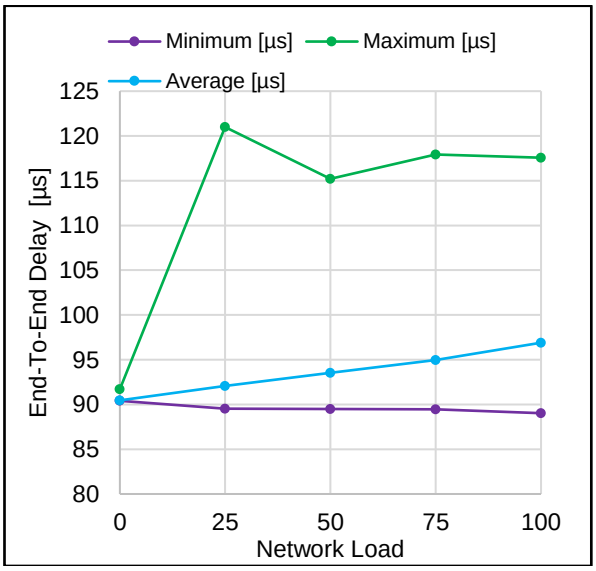


Figure 136: Behavior of the End-to-End delay over four TSN-switches for a data-traffic at a constant link-speed 1Gbps and increasing network-load from 0% to ~100% using non-preemptive strict-priority algorithm IEEE8020.1Q

The timing behavior using the frame-preemption approach shows almost parallel curves of the minimum, average and maximum end-to-end delays.
The minimum and average curves using the time-aware shaper supported by the TSN-switches are almost parallel. This proves that for most of the cycles there was no considerable difference. However it is observed that the maximum end-to-end delay is decreasing for a network load higher than 50%. The amount of background traffics should have no impact on the timing behavior of the scheduling approach. Therefore it is assumed

that the implementation of IEEE802.1Qbv in the switches under test has a bug. E.g. the starting time-point of the scheduled time-slot is not protected by the guard-band.

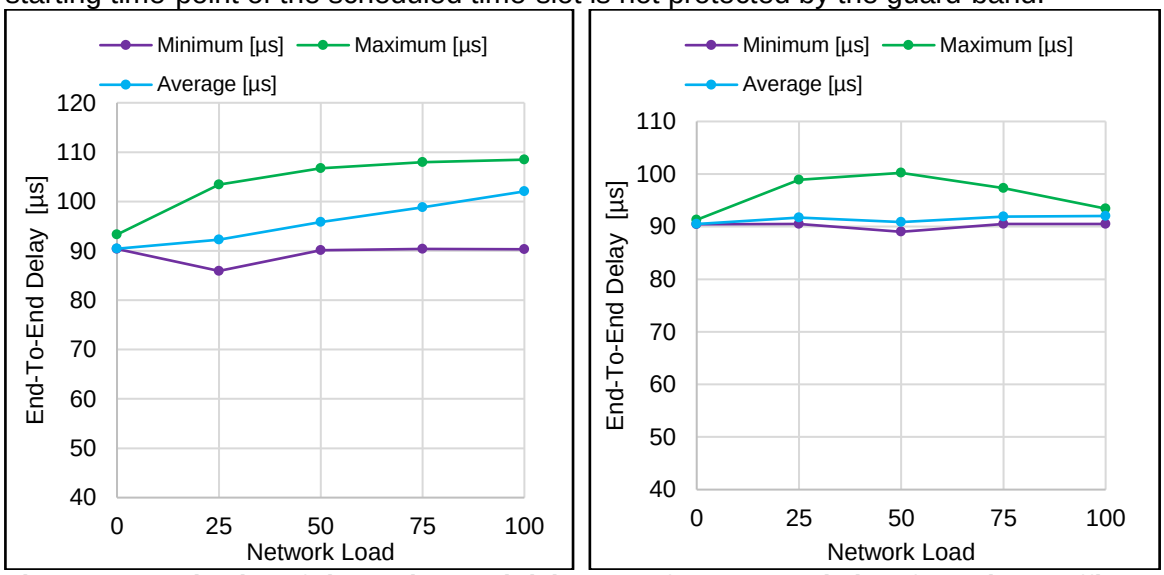


Figure 137: Behavior of the End-to-End delay over four TSN-switches for a data-traffic at a constant link-speed 1Gbps and increasing network-load from 0% to ~100% using:
(a) strict-priority algorithm IEEE802.1Q (b) frame-preemption IEEE8020.1Qbu and (c) time-aware shaper

Summary results of end-To-end delay

The measurements results of the end-to-end delay for six experiments are shown in the Figure 138 below. The experiments are given by different combinations of the link-speed and the traffic shaper. E.g. strict-priority algorithm IEEE802.1Q with 100Mbps.
The figure demonstrates the impact of the activated transmission selection algorithm (TSA) and the link-speed on the end-to-end delay.
Similar to the scenario with two forwarding hops, the best results are reached with Gigabit-Ethernet independently of the activated transmission selection algorithms. An upper-bound end-to-end delay below 100µs has been measured for all traffic-shapers.

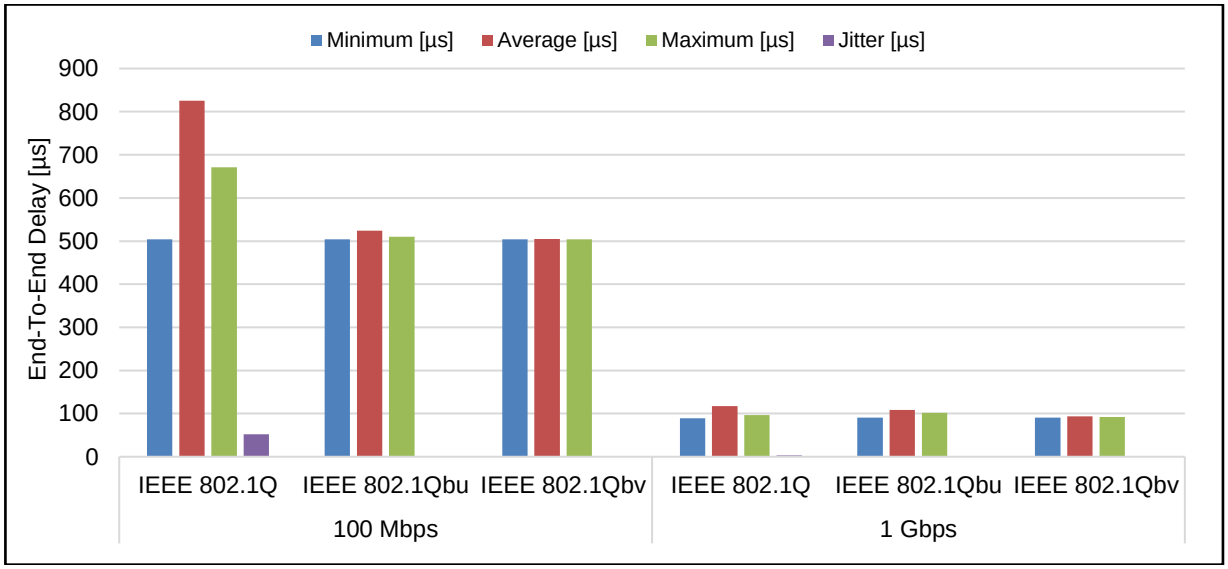


Figure 138: Impact of transmission selection algorithm (TSA) and link-speed on the end-to-end delay over four TSN-switches and 96% network load

Jitter

The measurement results for 100Mbps and 1Gbps link-speed and increasing network load are shown below (figure left). For the purpose of clarity the figure is zoomed, the axis is reduced from 60µs to 4µs (figure right).

An end-to-end delay variation below 1µs is measured using the combinations

- IEEE802.1Qbv and 100Mbps
- IEEE802.1Qbv and 1Gbps
- IEEE802.1Qbu for 100Mbps

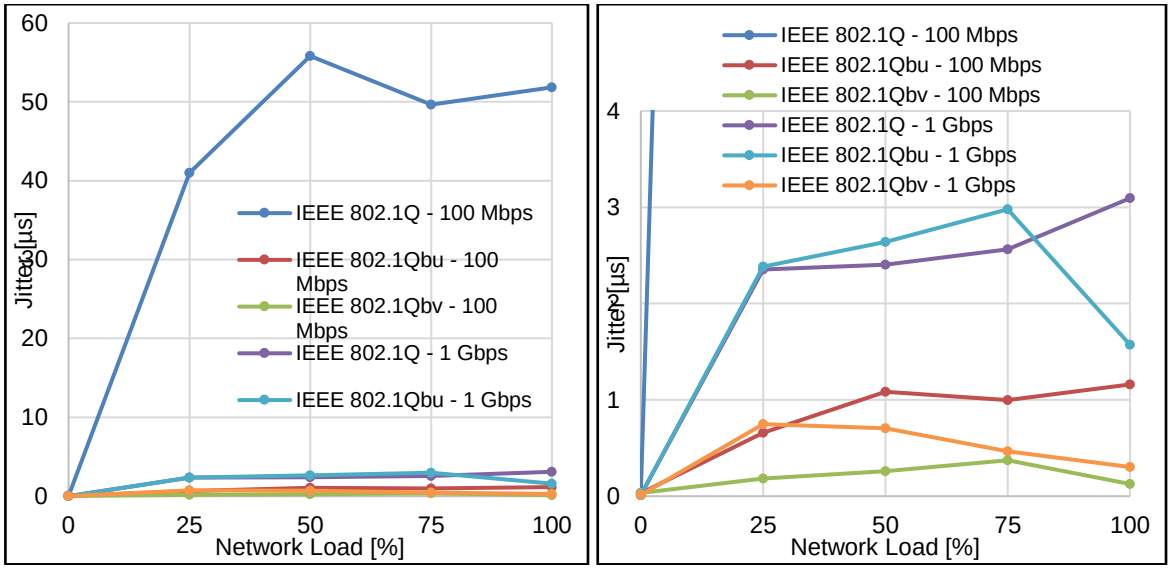


Figure 139: Comparison of Jitter in all TSAs at 100Mbps and 1Gbps over four TSN-switches

5.4.3 Coexistence of express and scheduled traffics

The previous scenarios did not covered the coexistence of two express traffics. Therefor the results for the frame-preemption approach is considered as “optimistic”, since the transmission of the background traffics can always be preempted as long as the size of the preemptable frame is below 131Byte. For a fair comparison between the frame-preemption and the scheduling approaches, this section considers the coexistence of two express traffics: the first is scheduled and thus protected against interferences and the second is non-scheduled. In case of interference between two express frames from different traffics, the non-scheduled frame must wait until the transmission duration of the scheduled frame and / or the time-slots finishes.

5.4.3.1 Scenario1 – Two forwarding Hops

Network configuration

The same network configuration as in section 5.4.1 is considered.

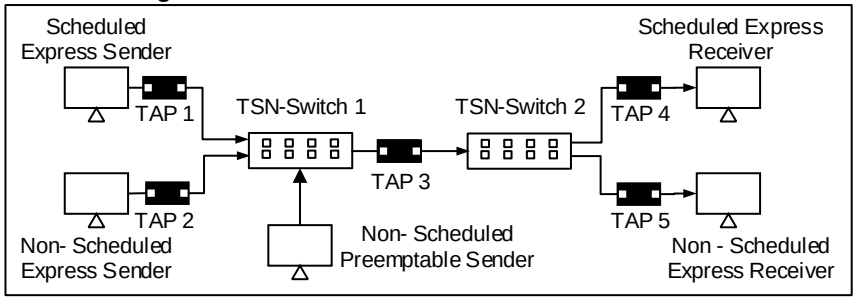


Figure 140: Network Setup for coexistence of express and scheduled traffic over 2 Hops

Network Configuration	Link-speed	100Mbps
	Frame Forwarding Mechanism	Store&Forward
	Topology	Line
	Number of forwarding hops	2
	Network Load	0% -25%- 50%-75%-100%

Table 77: network configuration

Traffic configuration

Three traffics with the same frame size 1514Byte are considered: scheduled express, non-scheduled express and non-scheduled preemptable.

Traffic Class	Size [Byte]	Transmission Interval [ms]	Priority	Frames / second
Scheduled Express[30]	1514	1	7	1000
Non-Scheduled Express	1514	1	6	1000
Non-Scheduled Preemptable	1514	Stream[31]	2	Depends on Network Load

Table 78: Traffic configuration for coexistence of express and scheduled traffic

Results

The figures below show the impact of extending the time-aware shaper with the frame-preemption approach on improving the use of bandwidth and reducing the end-to-end delays of the non-scheduled express traffic.

The increasing network load refers to the amount of non-scheduled preemptable frames transmitted from the non-scheduled preemptable device and forwarded over the TSN-Switch2 to the non-scheduled (Express) receiver.

The end-to-end delay of the non-scheduled express traffic is ~270µs in both scenarios (with and without the frame-preemption approach) and is given by its two transmission durations (2*121µs) and the internal processing time of both TSN-Switches (2 * 14µs).

By activating the background traffics (scheduled express and non-scheduled preemptable), the end-to-end delay of the non-scheduled express traffic increases exponentially from ~270µs to ~600µs if no frame-preemption is supported. Once the frame-preemption is activated, the average end-to-end delay remains almost the same ~277µs at 25% and 50% network-load and jumps by ~28µs at 75% and 100%.

This experiment proves that supporting the frame-preemption with the scheduling approach has a huge benefit on time-critical sporadic traffics that are not cyclic and thus should not be scheduled in order to better use the bandwidth. E.g. Sporadic safety-critical signals can be set as non-scheduled express traffics in parallel to the cyclic control-data traffics, which should be set as scheduled traffic.

[30] Reserved time-slot= 200 µs

[31] The non-time-critical traffic is not cyclically and is transmitted as a stream.

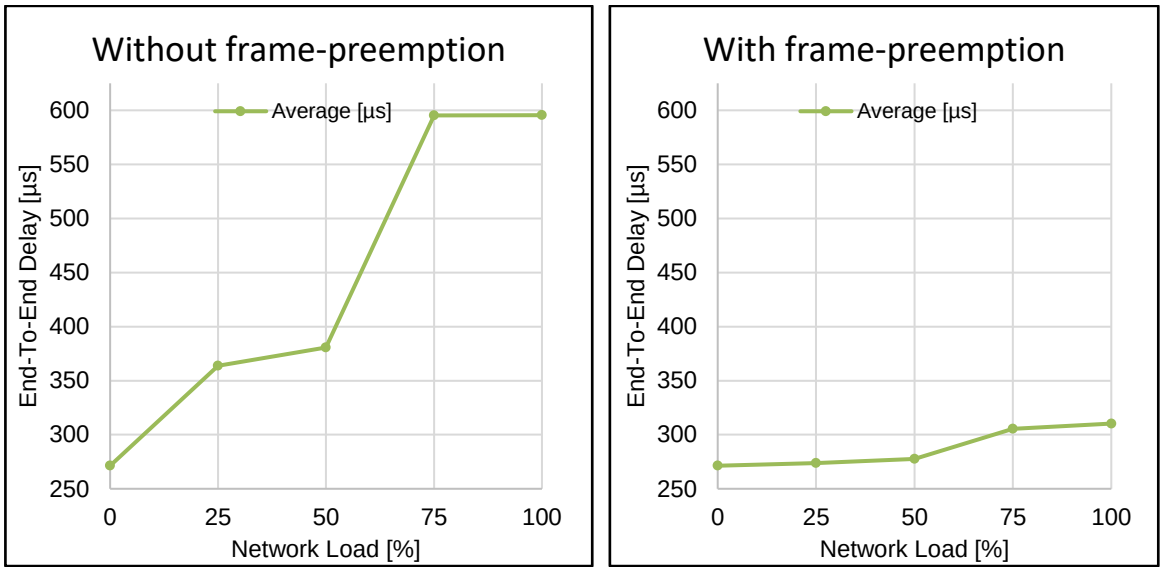


Figure 141: Comparison of the average end-to-end delay for non-scheduled Express traffic (100Mbps) using non-preemptive time-aware shaper (left) with preemptive-time-aware shaper (right) over 2 hops

5.4.3.2 Scenario2 – Four forwarding Hops

Network Setup

The same scenario as in 5.4.3.1 is repeated for four TSN-Switches.

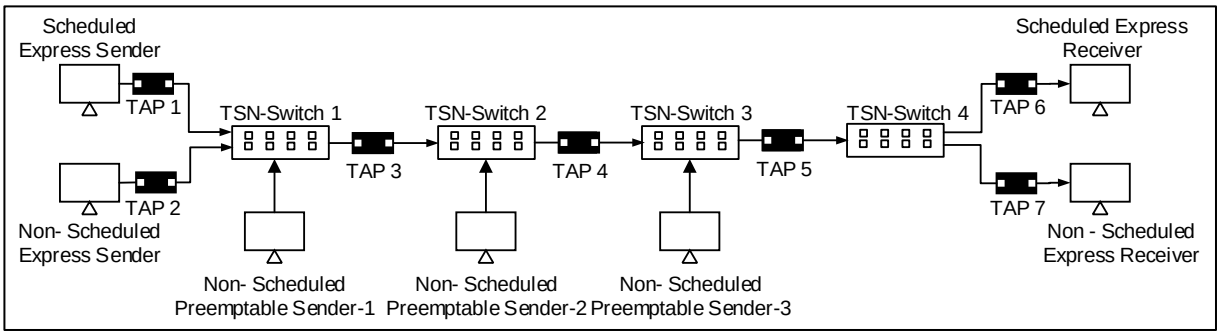


Figure 142: Network Setup for coexistence of express and scheduled traffic over 4 Hops

Traffic configuration

The express traffics remain the same as in the previous scenario 5.4.3.1. The non-scheduled preemptable traffics are set to three with a 1514Byte frame size each. These frames are transmitted in order to increase the probability of interference with the non-scheduled express traffic on each forwarding TSN-Switch (1 to 4).

Traffic Class	Size [Byte]	Transmission Interval [ms]	Priority	Frames / second
Scheduled Express[32]	1514	1	7	1000
Non-Scheduled Express	1514	1	6	1000
Non-Scheduled Preemptable 1	1514	Stream[33]	2	Depends on Network Load
Non-Scheduled Preemptable 2	1514	Stream[34]	2	Depends on Network Load
Non-Scheduled Preemptable 3	1514	Stream[35]	2	Depends on Network Load

Table 79: Traffic configuration for coexistence of express and scheduled traffic

Results

Unlike the previous scenario (5.4.3.1) with two TSN-Switches, the average end-to-end delay of the non-scheduled express traffic using four TSN-Switches shows a much better impact of activating the frame-preemption in addition to the scheduling approach in order to maintain an almost constant end-to-end delay for the express traffic. The average end-to-end delay increases from ~504µs to ~870µs without frame-preemption and to only 507µs with frame-preemption.

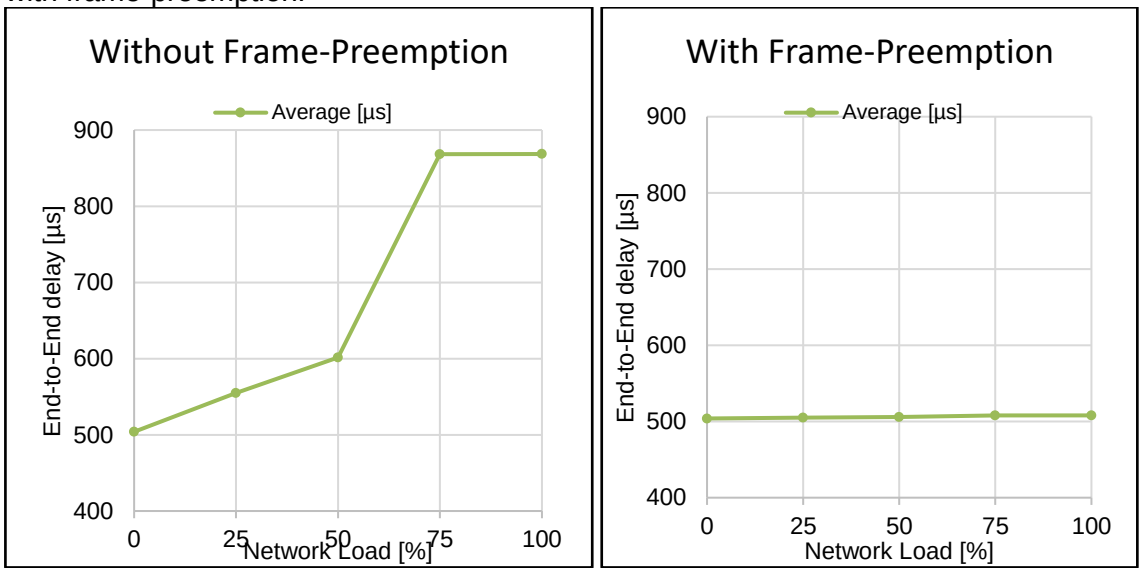


Figure 143: Comparison of the average end-to-end delay for non-scheduled Express traffic (at 100Mbps) using non-preemptive time-aware shaper (left) with preemptive-time-aware shaper (right) over 4 hops

The figure below illustrates the jitter (end-to-end delay variation) behavior of the non-scheduled express traffic transmitted over four TSN-Switches with and without activating the frame-preemption approach. In red is the scenario using the Preemptive Time-Aware Shaper (P-TAS) and in blue is the scenario using the Non-Preemptive Time-Aware Shaper (NP-TAS).

The jitter increases from 0,04µs at 0% network-load to ~245µs at 75% or 100% network-load if no preemption is activated. This is due to the high amount of interferences with the background traffics.

The jitter remains below 2µs by activating the frame-preemption approach even at 100% network-load.

[32] Reserved time-slot= 200 µs

[33] The non-time-critical traffic is not cyclically and is transmitted as a stream.

[34] The non-time-critical traffic is not cyclically and is transmitted as a stream.

[35] The non-time-critical traffic is not cyclically and is transmitted as a stream.

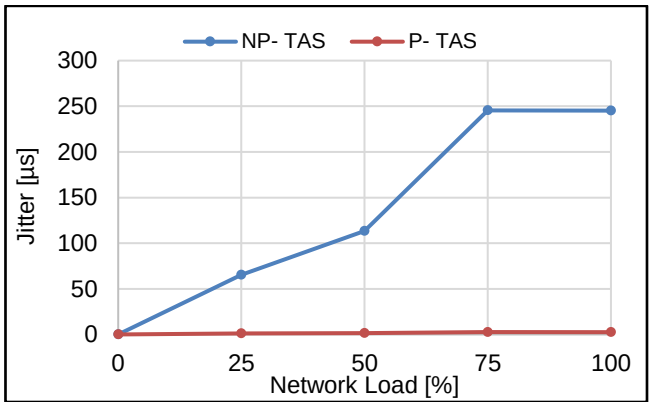


Figure 144: Jitter of non-scheduled express frame using non-preemptive and preemptive time-aware shaper

5.4.4 Impact of the background frames sizes on jitter

The previous scenarios (5.4.3.1 and 5.4.3.2) showed the benefit of supporting the frame-preemption approach on improving the use of bandwidth and especially on improving the end-to-end delay and jitter of the non-scheduled express traffics. But only the network-load (number of background traffics) was taken into account as a factor that affects the timing behavior of the time-critical traffics.
This section investigates the impact of the size of the background traffics on increasing the end-to-end delay variation (jitter) of the non-scheduled express traffic.

Network Configuration

Link-speed	100Mbps
Frame Forwarding Mechanism	Store&Forward
Topology	Line
Number of forwarding hops	2/4
Network Load	80% -100%

Figure 145: Network Configuration

Traffic configuration
Four data traffics are recognized: a single 1400Byte time-critical traffic and three non-time-critical traffics of different sizes. Since the frame-preemption approach has its best impact if the preemptable frames can be preempted: e.g. preemptable frame is bigger than 131Byte and / or the remaining Bytes to be transmitted is above 64Byte.

Traffic Type	Size [Byte]	Transmission Interval [ms]	Priority	Frames / second
Time-Critical	1400	1	7	1000
Non-Time-Critical 1	64-131	Stream[36]	2	Depends on Network Load
Non-Time-Critical 2	131	Stream[37]	2	Depends on Network Load
Non- Time-Critical 3	131-1514	Stream[38]	2	Depends on Network Load

Figure 146: Traffic Configuration

5.4.4.1 Scenario1 – Two forwarding Hops

Network Setup

[36] The non-time-critical traffic is not cyclically and is transmitted as a stream.
[37] The non-time-critical traffic is not cyclically and is transmitted as a stream.
[38] The non-time-critical traffic is not cyclically and is transmitted as a stream.

The network is composed of two talkers: Time-Ctritical Sender and Non-Time-Critical Sender transmitting two traffics to the receiver device over TSN-Switch1 and TSN-Switch2. With the Network TAP device, three positions on the network are used to measure the forwarding delays of the single switches and the overall end-to-end delay.

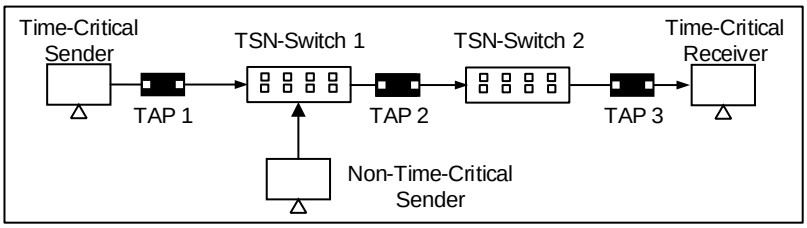


Figure 147: Network Setup for P-SPA Evaluation over 2 hops

Results

Around 2000 samples have been measured. During the first half of the measurement, the time-critical traffic is interfering with NTC-traffic of a variable size between 64 to 131Byte. During the second half with a NTC-traffic of a fixed size 1514Byte.

Since preemptable frames of a size below 131Byte cannot be preempted into two Ethernet fragments (minimum Ethernet frame size rule), the resulting interference delay is even higher than the interference delay of the preemptable frames. This can be seen in the figure below.

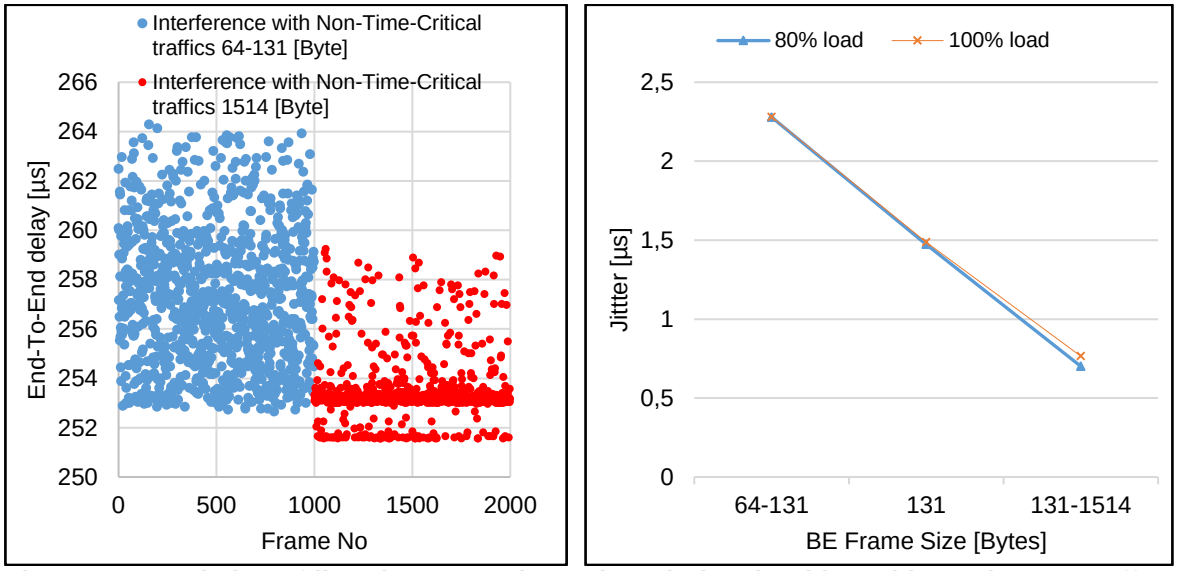


Figure 148: Variation of jitter in preemptive strict-priority algorithm with varying Best Effort Frame size over 2 Hops

5.4.4.2 Scenario2 – Four forwarding Hops

Network Setup

The network in the previous scenario is extended with two more TSN-Switches and two more Non-Time-Critical Senders. This increases the end-to-end delay as well as the interference delay of the time-critical traffic.

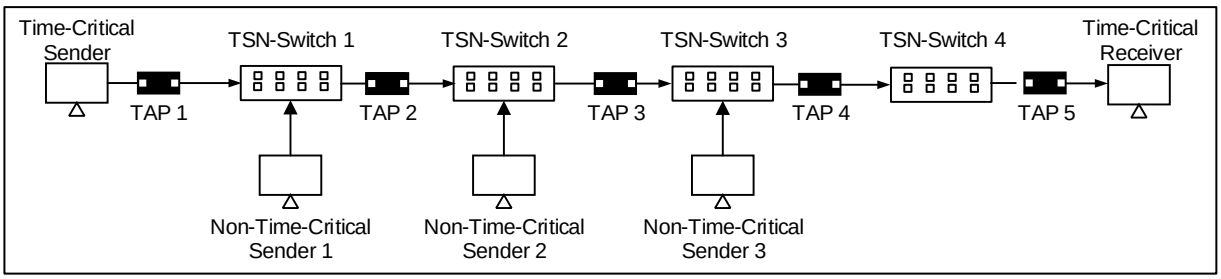


Figure 149: Network Setup for P-SPA Evaluation over 2 hops

Results

The figures below highlights the impact of the preemptable frames that cannot be preempted, due to the size (below 131Byte) on the interference delay and the resulting overall end-to-end delay for the time-critical traffic.

Increasing the amount of background traffics from 80% to ~100% has no significant effect on the jitter. This is due to the high interference between the time-critical and the background traffics.

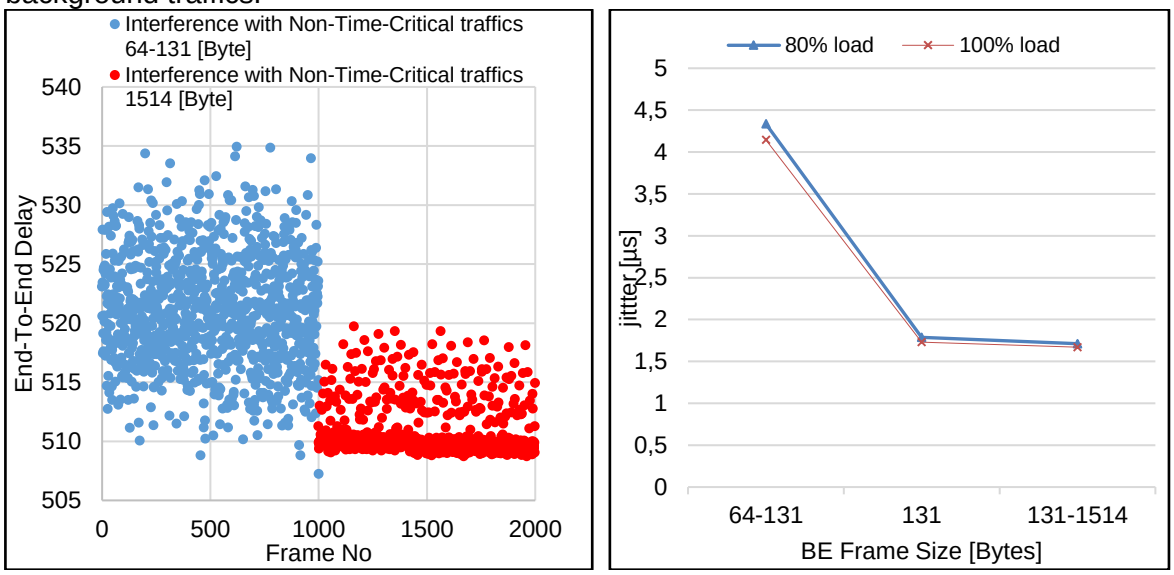


Figure 150: Variation of jitter in preemptive strict-priority algorithm with varying Best Effort Frame size over 4 Hops

6 Conclusion and Outlook

This chapter summarizes the topics covered in this thesis and provides an outlook for further researches.

6.1 Overview and Contributions

The thesis investigates the timing performances of TSN-Ethernet for use in highly deterministic industrial applications such as Motion-Control.

TSN specifies communication features for safety-critical applications (through redundancy and path reservation), deterministic communication and security. The thesis covers the sub-standards required for a deterministic communication. Four out of the TSN nine sub-standards are evaluated for use in the industry:

- IEEE802.1AS for time-synchronization
- IEEE802.1Qbu and IEEE802.3br for frame-preemption and interspersing express-traffic
- IEEE802.1Qbv for frame scheduling

The thesis investigated the real-time performances of the TSN sub-standards through a set of KPIs. Multiple communication scenarios have been then defined with respect to the required conditions in the factory. Three investigation methods are used:

- ***Formal***: through mathematical computations based on specifications of certain industrial-Ethernet protocols and queuing theory.
- ***Simulative***: through extensive simulation scenarios, in which the number and communication-features of the network-devices can be easily extended to simulate real industrial plants.
- ***Experimental***: through the development of a TSN-test-setup composed of configuration devices, TSN-devices under-test and some network-analysis and testing equipments. The goal of the TSN-test-setup is to evaluate the TSN-features in real-operation and to test the interoperability of different TSN-implementations in their early releases.

After defining and comparing a set of the established Industrial-Ethernet protocols (Sercos III, Profinet RT, Profinet IRT and EtherNet/IP), the integration of the established Industrial-Ethernet protocols in TSN have been introduced.

It has been proven that TSN lacks a very important feature in order to reach cycle times in the microsecond range and thus to be used in the field-level of the industrial automation pyramid: Cut-Through forwarding.

The thesis introduces an alternative to the complex TDMA concept with a similar timing behavior. The new concept is a combination of frame-preemption with Gigabit-Ethernet and cut-through, which offers similar real-time performance to the scheduling approach but with much easier communication configuration and higher bandwidth usage. However the concept is suitable especially for networks in which the coexistence of non-preemptive express traffics is limited.

Certain established Industrial-Ethernet protocols have been chosen to be combined with the TSN-sub-standards mentioned above. Protocols that already support the scheduling approach (e.g. EtherCAT, Sercos, Profinet IRT) have no big benefit from extending their system with the TSN-features, at least in terms of timing behavior. Only certain KPI, such as bandwidth usage, can be further improved. Therefore it remains critical for the organizations specifying these protocols to extend their system architecture with TSN.

Therefore certain organizations, e.g. EtherCAT Technology Group, are trying to keep their field-devices unmodified in the field-level and to benefit from TSN and its high bandwidth by introducing some TSN-modules and extending their PLC-master. This approach has been already covered in the thesis with a very similar protocol “Sercos”.
Organizations with small communities have a new chance to specify a successor for their protocols and are investing in the upcoming OPC UA Pub/Sub over TSN.
The “fieldbus war” is starting again!

6.2 Outlook

This section provides certain research gaps that need to be investigated to extend the use of TSN in the industrial communication.

Network Configuration
The TSN network configuration can be considered as a critical factor for a successful integration of TSN in the industry. Today most Industrial-Ethernet networks support the central configuration model. This means all communication-hops are configured by a central device, typically the PLC.
Certain Industrial-Ethernet Protocols, such as EtherCAT, provides variable frame size, which makes it hard to schedule the time-critical traffics without certain lost of bandwidth, since the reserved time-slot is not completely used. Therefore providing an approach for a dynamic time-slot reservation could be an attractive feature. The approach should present a solution in which the size of the time-slots can be adjusted online.
Further, the network configuration device should be able to identify and configure all TSN capable devices. The available solutions are not compatible, which limits its employment.

Safety-critical Communication
The TSN standard IEEE802.1CB provides new mechanisms for a seamless redundancy. This standard should be evaluated for use in the industrial-communication. Industrial-Ethernet Protocols such as Sercos III provides a medium-redundancy. The frames are duplicated and received by the PLC. TSN enables the elimination of redundant frames by other devices than the destination device. The bandwidth utilization and the safety levels should be investigated with IEEE802.1CB.

Bibliography

[1] J. Liu; "Real-Time Systems"; Chapter 1 pp. 15; edition 2002

[2] N. Finn; "Current state of IEEE 802.1 Time-Sensitive Networking Task Group"; Internet Engineering Task-Force (IETF 91 Meeting); Honolulu Hawai, November 2014

[3] S. Dietrich, G. May, O. Wetter, H. Heeren, G. Fohler; "Performance Indicators and Use Case Analysis for Wireless Networks in Factory Automation"; 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA);Limassol Cyprus, September 2017

[4] T. Zimmerman; "Metrics and KPI for Robotic Cybersecurity Performance Analysis"; NISTIR 8177 National Institute of Standards and Technology (NIST), Gaithersburg USA, 2016]

[5] T. Steinbach; "Echtzeit-Ethernet für Anwendungen im Automobil: Metriken und deren simulationsbasierte Evaluierung am Beispiel von TTEthernet"; Section 3.3 pp. 40; February 2011

[6] IEC 61784 Industrial communication networks - Profiles - Series of standard ISO/IEC, International Electrotechnical Commission (IEC) Standard 2014

[7] C. Tang; "Key Performance Indicators for Process Control System Cyber Security Performance Analysis";NISTIR 8188 National Institute of Standards and Technology (NIST); August 2017

[8] L. Leurs; "Industrielle Steuerungstechnik"; Chapter 7; June 2013

[9] M. Schumacher, J. Jasperneite, K. Weber; "Limits of increasing the performance of Industrial Ethernet protocols"; IEEE Emerging Technologies and Factory Automation ETFA; Patras Greece, September 2007

[10] T. Carlsson; "Industrial Ethernet and Wireless are growing fast — Industrial network market shares 2017"; HMS and Statista.com ; February 2017

[11] S. Nsaibi, L. Leurs; H. D. Schotten; “Worst-Case Timing Analysis of Integrating TSN in the Field-Level”; KommA 2017 – Jahreskolloquium Kommunikation in der Automation; Magdeburg Germany, November 2017

[12] J. Nilsson; "Real-Time Control Systems with Delays"; Department of Automatic Control, Lund Institute of Technology (LTH)1998

[13] Statista.com; [https://de.statista.com/statistik/daten/studie/457627/umfrage/marktanteile-industrieller-netzwerke-weltweit/]; 2018

[14] Ethernet Book search for a name

[15] S. Nsaibi; „Gegenüberstellung der deterministischen Kommunikationsmechanismen von TSN: Netzwerkplanung und Telegrammunterbrechung“; AUTOMATION 2017 Technology Networks Processes; Baden-Baden Germany, June 2017

[16] P. Didier, G. Ditzel; "Time Sensitive Network (TSN) Protocols and use in Ethernet/IP Systems"; ODVA 2015 Industry Conference And 17th Annual Meeting; Texas USA , October 2015

[17] IEEE Standard for Ethernet, 802.3-2012 – section 1; 2012-12-28. pp. 53

[18] Uni-Koblenz; "Rechnerorganisation" – Chapter 7 Ethernet/IEEE802.3

[19] P. Danielis et al.; "Survey on Real-Time Communication Via Ethernet in Industrial Automation Environments"; Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA); Barcelona Spain, September 2014

[20] L. Zihong, S. Pearson; "An inside look at industrial Ethernet communication protocols"; Whitepaper 2018

[21] J.-D. Decotignie; "Ethernet-Based Real-Time and Industrial Communications"; Proceedings of the IEEE; https://ieeexplore.ieee.org/document/1435741 , May 2005

[22] X. Li; "Worst-case delay analysis of real-time switched Ethernet networks with flow local synchronization"; PhD Chapter 4; September 2013

[23] Scalable Real-Time Ethernet Platform; TTTech "TTEthernet - A Powerful Network Solution for All Purposes";

https://paginas.fe.up.pt/~ee04007/tese/docs/TTEthernet_Article.pdf; 2009

[24] M. Schumacher, J. Jasperneite, K. Weber; "A new Approach for Increasing the Performance of the Industrial Ethernet System PROFINET", IEEE International Workshop on Factory Communication Systems (WFCS); Dresden Germany, May 2008

[25] J. Robert, J.-P. Georges, E. Rondeau, T. Divoux, "Minimum Cycle Time Analysis of Ethernet-Based Real-Time Protocols";INT J COMPUT COMMUN, Vol.7 pp. 744-758; November 2012

[26] S. Nsaibi, L. Leurs; „Abbildung von TDMA-basierten Industrial Ethernet Protokollen auf TSN am Beispiel von Sercos III“; KommA 2016 – Jahreskolloquium Kommunikation in der Automation; Lemgo Germany, November 2016

[27] S. Nsaibi, L. Leurs, H. D. Schotten; “Formal- and Simulation-based Timing Analysis of Industrial-Ethernet Sercos III over TSN”; IEEE/ACM DS-RT; The 21st IEEE International Symposium on Distributed Simulation and Real-Time Applications; Rom Italy, October 2017

[28] T. Steinbach et al.; "Beware of the Hidden! How Cross-traffic Affects Quality Assurances of Competing Real-time Ethernet Standards for In- Car Communication", IEEE LCN, Clearwater Beach USA, October 2015

[29] R. Ernst; "Formal Worst-Case Performance Analysis of Time-Sensitive Ethernet with Frame-preemption"; IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA); Berlin Germany, September 2016

[30] T. Wan et al.; "A Performance Study of CPRI over Ethernet with IEEE 802.1Qbu and 802.1Qbv Enhancements"; IEEE Global Communications Conference (GLOBECOM); San Diego USA, December 2015

[31] X. Li; "Worst-case delay analysis of real-time switched Ethernet networks with flow local synchronization"; PhD Chapter 1.5 pp. 19; September 2013

[32] INET Framework; "INET Framework Manual"; Section 1; Version 2016

[33] ANSI/IEEE Std 802.1Q. "Media Access Control (MAC) Bridges"; ANSI/IEEE Std 802.1D-1998.; IEEE. Virtual Bridged Local Area Networks; New York, 1998

[34] ODVA; CIP Motion System Performance Considerations

[35] A. Astarola et al.; "FPGA Implemented Cut-Through vs Store-and-Forward Switches for Reliable Ethernet Networks"; IEEE Design of Circuits and Integrated Systems; Madrid Spain, November 2014

[36] P. Thaler; "Issues regarding cut-through"; January 2017

[37] "Traffic control and congestion control in B-ISDN"; "https://www.itu.int/rec/T-REC-I.371-200403-I/en"; ITU-T Telecommunication Standardization Sector of ITU - I.371 (March 2004)

[38] IEEE Std 802.1QTM-2014 – Revision of IEEE Std 802.1Q-2011 - Section 8.6.8 Transmission selection pp. 127

[39] IEEE Std 802.1QTM-2014 – Revision of IEEE Std 802.1Q-2011 - Section 6.9.4 Regenerating priority Page 77

[40] IEEE Std 802.1QTM-2014 – Revision of IEEE Std 802.1Q-2011 - Section 8.6.8.2 Credit-based shaper algorithm Page 128

[41] IEC, Geneva; "Digital data communication for measurement and control – Fieldbus for use in industrial control systems"; IEC 61158 Ed.3 CDV, Type 8 (INTERBUS); 2001

[42] Sercos.rog - https://www.Sercos.org/technology/advantages-of-Sercos/performance/

[43] ODVA Literature Library; "Introduction to EtherNet/IP – CIP Encapsulation Message"; pp. 18

[44] J. Robert, J.-P. Georges, E. Rondeau, T. Divoux, "Minimum Cycle Time Analysis of Ethernet-Based Real-Time Protocols";INT J COMPUT COMMUN, Vol.7 Section 2.5; November 2012

[45] "Industrial-Ethernet Overivew" – Section: PROFINET RT (V2) - pp. 10

[46] L. Lachello, S. Schönegger et al.; "Industrial-Ethernet Facts - System Comparison" – 3rd Edition

[47] "TI Designs Certified PROFINET IRT v2.3 Device With 1-GHz ARM Application Processor"; http://www.ti.com/lit/ug/tiduak0/tiduak0.pdf, Texas Instruments; October 2015
[48] "Sercos Specifications"; Section S-0-1002 Communication Cycle time (tScyc)
[49] P802.1AS-Rev – Timing and Synchronization for Time-Sensitive Applications "https://1.ieee802.org/tsn/802-1as-rev/" Draft D7.4; November 2018
[50] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems" – pp. 18 – Section 6.3; July 2008
[51] "IEEE802.1AS-2011" – Section 3.3 – pp. 5
[52] "IEEE1588 2008" – Section 6.5.1 – pp. 18
[53] "IEEE1588 2008" – Section 6.5 – pp. 20-29
[54] "IEEE1588 2008" – Section 6.5 – pp. 29
[55] "IEEE802.1AS-2011" – Table 10-1
[56] "IEEE802.1AS-2011" – Section 8.4.2.2
[57] "IEEE1588 2008" – Section 6.4 – Page 17
[58] "IEEE1588 2008" – Section 7.7.2 – Page 61
[59] "IEEE1588 2008" – Section 6.6.2.4 – Page 32
[60] "IEEE802.1AS-2011" – Figure 10-10
[61] "IEEE1588 2008" – Section 6.6.2.3 – Page 31
[62] "802.1Qbv, Draft 3.1 comment dispositions"; Draft 3.1; http://www.ieee802.org/1/pages/802.1bv.html
[63] "IEEE802.1Qbv – Amendment25: Enhancements for Scheduled Traffic"
[64] M. Haas, W. Zorn; "Handbuch der Informatik" Bd. 2: Methodische Leistungsanalyse von Rechensystemen. München : Oldenbourg, 1995. – ISBN 3-486-20779-2
[65] R. Baumann, S. Heimlicher, M. Strasser and A. Weibel; "A Survey on Routing Metrics". Computer Engineering and Networks Laboratory, ETH-Zentrum, Switzerland, TIK Report 262, 2007, pp. 1-53. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.304.7863&rep=rep1&type=pdf
[66] Hummen; O. Kleineberg; "TSN and Cyber security Webinar"; 2018 https://event.on24.com/eventRegistration/console/EventConsoleApollo.jsp?simulive=y&eventid=1582016&sessionid=1&username=&partnerref=&format=fhvideo1&mobile=false&flashsupportedmobiledevice=false&helpcenter=false&key=7FA568925986CF8419134979BC22475E&text_language_id=en&playerwidth=1000&playerheight=650&overwritelobby=y&eventuserid=197217109&contenttype=A&mediametricsessionid=160563714&mediametricid=2267993&usercd=197217109&mode=launch
[67] J. Jasperneite, M. Schumacher; "Forschungsprojekt ESANA: Echtzeit-Ethernet für die Sensor/Aktorvernetzung"; Lemgo Germany, February 2009; https://www.hs-owl.de/init/uploads/tx_initdb/esana_f.pdf
[68] G. Workman ; "Single hop inter-VLAN routing – a capability that EtherNet/IP I/O can take advantage of – a feature that suppliers of advanced, managed, industrial Ethernet switches should consider implementing"; ODVA Industry Conference & 16th Annual Meeting; Phoenix Arizona USA, March 2014; https://www.odva.org/Portals/0/Library/Annual_Meeting_2014/2014_ODVA_Conference_Workman_Single_hop_inter-VLAN_router_FINAL.pdf
[69] T. Brandl; "So gelingt die Interoperabilität von OPC-UA und TSN" ; July 2017; https://www.industr.com/de/gemeinsame-sprache-fuer-die-zukunft-2295448
[70] "Intel Ethernet Controller I210 Datasheet"; 2017
[71] "Intel Ethernet Controller I210 Datasheet"; table 7-62; 2017
[72] Beckhoff; "ET2000 Industrial-Ethernet-Multichannel-Probe"; Product Datasheet; Version 2.2; January 2016 datasheet pp-8
[73] S. Nsaibi, A. Mishra; "*Experimental Evaluation of TSN performance"*; Industrial Ethernet Book, pp. 33-36; Issue 106 ISSN1470-5745, Mai 2018
[74] Common Industrial Protocol – CIP – Volume 2
[75] ODVA Annual Meeting 2014 – ODVA Conference Workman Single hop inter-VLAN router FINAL

https://www.odva.org/Portals/0/Library/Annual_Meeting_2014/2014_ODVA_Conference_Workman_Single_hop_inter-VLAN_router_FINAL.pdf

List of figures

List of tables

Lebenslauf

Dipl.-Ing. Seifeddine Nsaibi

LinkedIn Profil **XING** Profil

Persönliche Angaben

Name, Vorname	Nsaibi, Seifeddine

Ausbildung

07.2015 – 06.2018	**Promotion (Dr.-Ing.) –**TU Kaiserslautern (EIT) – Bosch Rexroth AG – Titel: *„Timing Performance Analysis of the deterministic Ethernet Enhancements Time-Sensitive Networking (TSN) for use in the industrial communication "*
04.2010 – 05.2015	**Studium (Dipl.-Ing.) –** Informationstechnik **– Vertiefung:** Embedded-Systems Technische Universität Kaiserslautern (TU-KL) – Deutschland
12.2014 - 05.2015	**Diplomarbeit –** extern bei **Bosch Rexroth AG** – Erbach – Functional Safety Titel: *„Untersuchung des Optimierungspotentials vom CIP Safety Protokollstack"* (**Note 1,3**) Sehr gut
07.2014 - 11.2014	**Studienarbeit:** extern bei **Bosch Rexroth AG** – Lohr am Main – Functional Safety Titel: *„Messung der Laufzeit von CIP Safety on Sercos in der Rexroth Safety-Steuerung SafeLogic"* (**Note 1,3**) Sehr gut
	Abitur: Schwertpunkt *„Sciences Experimentelles"* – Tunesien

Praktische Erfahrungen

05.2018 - heute

Product Manager – Hoch Automatisiertes Fahren – ETAS GmbH (Bosch Gruppe)

- Messtechnik Zulieferer an TIER1 und OEM
- Koordination zwischen Entwicklung, Strategie, Einkauf, Logistik und Sales
- Agile & Lean Management
- Requirements Engineering – Dokumentation von Pflichten- und Lastenheft
- Priorisierung von Features & User Stories – Features Backlog
- Entwicklung Business Strategien
- Erstellung und Betreuung der ETAS Marketing – Messen & Test-setupen
- E/E Architektur
- Testing ADAS und HAD (Highly Autonomous Driving) Lösungen

07.2015 - 07.2018

Industrie Promotion bei Bosch Rexroth AG

- Zahlreiche wissenschaftliche Veröffentlichungen
- Wissenschaftlicher Beitrag zur Bewertung von IEEE Time-Sensitive Networking (TSN)
- Teilnahme am öffentlich geförderten Projekt FIND
- Betreuung von Studenten und Abschlussarbeiten
- Strategische Nutzwertanalyse der Integration von TSN in die Automatisierung
- Bewertung und Evaluierung von Kommunikationsprototypen & -Szenarien für Industrie 4.0
- Teamführung, strukturierte Arbeitsweise, hohes Engagement
- Mit-Entwicklung des europäischen IIC TSN Test setups

Projekte

- **Bosch Daimler Athena Projekt** – Hochautomatisiertes Fahren
- **ÖgP FIND** – Öffentlich gefördertes Projekt – Future Industrial Network Architectures
- **IIC TSN Test setup** – Industrial Internet Consortium
- **Sercos International Task Force** – Fachlicher Netzwerk Experte
- **Continental AG & VPE Institut** – Virtuelle Produktentwicklung (Robotic)
- **Fraport AG & MEC Institut** – Entwicklung von Avionic Hydraulics

Teamleitung

- Erstellung und Führung eines Rexroth internen Teams von 6 Masteranden
- Fachliche Führung eines Bosch internen Entwicklerteams (Synergien der Bosch Automotive- & Industriellen-Sektoren)
- Scrum Agile Entwicklung
- Aufbau eines Test-setups zur Evaluierung neuer Ethernet-basierten Kommunikationsszenarien
- Netzwerk Simulation (OMNEST) von zukünftiger Industrie 4.0 Netzwerkarchitekturen zur Integration von IEEE Ethernet TSN in die Automatisierung

07.2014 - 05.2015

2x Abschussarbeiten + 1x Praktikum bei Bosch Rexroth AG

- Functional Safety (Industry) – Vom TÜV zertifizierte Arbeitsergebnisse
- Entwicklung Safety Firmware Stack

- Erstellung eines großen Test setup von 3 Steuerungen und 70 Antriebs- und I/O-Module
- Hardware Messung zur Evaluierung der Stack Laufzeiten
- Systematische Firmware Optimierung zur Reduzierung der Safety Zykluszeiten
- Mitarbeit an zwei Bosch Rexroth Standorten und Schnittstelle zwischen zwei Teams

03.2013 - 06.2014 **Wissenschaftliche Hilfskraft**, Produktentwicklung – Automatisiertes Fahren
Institut für Virtuelle Produktentwicklung (VPE) & **Continental AG**

- Product Lifecycle Management – PLM
- V-Modell (Anforderungsanalyse: Funktionale-/ Nicht funktionale Anforderungen)
- Mindstorms Roboter zum Automatisierten Fahren
- Matlab Simulink – Produkt Design
- Studentenbetreuung – Platz 1 bei einem Wettbewerb – Continental AG
- Erstellung und Korrektur von Prüfungen

08.2013 - 10.2013 **Wissenschaftliche Hilfskraft**, Avionic Hydraulics
Lehrstuhl für Mechatronik (MEC) & **Fraport AG**

01.2014 - 05.2014 **Wissenschaftliche Hilfskraft**, Wireless Networked Control Systems (inverted pendulum)
Lehrstuhl für Automatisierungstechnik (AT)

- C++ Programmierung
- Matlab/Simulink zur Simulation eines interierten Pendels
- Regelungstechnik Messungen an einem realen Test setup – Inverted Pendulum

11.2009 – 02.2010 **VISA Dienst Bonn GmbH & Co.** Standort – Frankfurt am Main

- Visa Dienste mit Konsulaten und Botschaften in Frankfurt am Main
- Termine Planung, Management- und Administrative Aufgaben

Sprachen

Deutsch - Englisch	Verhandlungssicher
Arabisch - Französisch	Muttersprache (Muttersprache)
Italienisch	Grundkenntnisse

Kenntnisse

Agile Methoden	Scrum – Kanban – Scrum of Scrum – SAFe (Scaled Agile Framework for enterprise) XML / SysML Mind Mapping – MindJet TTM Project Management
Veröffentlichungen	6x Konferenzveröffentlichungen 3x Fachartikeln in Journals 4x Scientific Reports
Programmieren	Matlab / Simulink C/C++ Software Architecture Safety Defensive Programmierung – Safety Integrity Levels (SILs)
Netzwerke - Kommunikation	Wireshark – OMNeT++ / OMNEST – Netzwerk Simulation Network Management Deterministic (Real-Time) Communication IEEE802 Ethernet Standard – IEEE802 AVB – IEEE802 TSN Industrial-Ethernet Protocols – CAN – FlexRay – Time-Triggered Ethernet (TTE)
Verification / Evaluation	Entwicklung / Aufbau von Test-setupen – Messen / Workshops Hardware Messung (Oscilloscope – Network TAPs – Wireshark) Evaluation & Verification von Standard-Compliancy
IT	Microsoft Office Outlook Enterprise Architect