AUTOWARE

Wireless Autonomous, Reliable and Resilient Production Operation Architecture for Cognitive Manufacturing

D2.2b AUTOWARE Deterministic Ethernet Communications

Document Owner	**TTTech Computertechnik AG**		
Contributors	SmartFactoryKL		
Reviewers	SmartFactoryKL		
Dissemination level	Public	**Dissemination nature**	Report
Date	26/06/2019	**Version**	1.0

European Commission

Version History

Nr.	Date	Author (Organization)	Description
0.1	20/02/2019	TTT	ToC defined
0.2	15/03/2019	TTT	Description TSN standards
0.3	12/04/2019	TTT	Scheduling description
0.4	30/04/2019	SFKL	Input SFKL regarding usage of TSN at SFKL
0.5	30/05/2019	TTT	Dynamic (re-)configuration decsription
0.6	07/06/2019	TTT	Introduction, Conclusion
0.7	24/06/2019	SFKL	Added review comments
1.0	26/06/2019	TTT	Updates based on review comments

Project partners

Software Quality Systems	SQS
Asociación de Empresas Tecnológicas Innovalia	INNO
Technologie Initiative SmartFactoryKL e.V.	SmartFactory
Josef Stefan Institute	JSI
TTTech Computertechnik AG	TTT
Consiglio Nazionale Delle Ricerche	CNR
imec	imec
Robovision	Robovision
Universidad Miguel Hernández	UMH
Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.	FhG
Blue Ocean Robotics	BOR
Fundación Tekniker	Tekniker
SMC Pneumatik GmbH	SMC

Table of Contents

1 Introduction 7

- 1.1 Purpose and Scope 7
- 1.2 Contributions to other WPs and deliverables and document structure 7
- 1.3 Target Audience 8

2 Continuous Development of TSN 9

- 2.1 TSN Profile for Industrial Automation 10

3 Scheduling 12

4 Dynamic (Re-)Configuration 16

- 4.1 IEEE 802.1Qcc – Stream Reservation Protocol 16

5 Deterministic Communication Network Configuration Tool 21

6 Applicability in AUTOWARE 24

- 6.1 TSN in SmartFactoryKL 24

7 Conclusions and Future Work 26

List of Figures

Figure 1: TSN Standardization Status (February 2019) 9
Figure 2: TSN Profile for Industrial Automation Standardization Status (February 2019) 11
Figure 3: Overview of a TSN switch, with IEEE 802.1Qbv 12
Figure 4: Example network 14
Figure 5: Communication Schedule for TSN network 14
Figure 6: (Re-)configuration System – Identifying changes in the network 18
Figure 7: Self-configuration Framework 19
Figure 8: Physical topology view 21
Figure 9: Path/logical topology view 22
Figure 10: Scheduled path Visualization 22
Figure 11: Scheduled gates visualization 23
Figure 12: Network Topology 24

List of Tables

Table 1: Configured gate control list for IEEE 802.1Qbv 25

Executive Summary

This deliverable D2.2b provides an overview of the work that is being performed within the AUTOWARE project in the field of Deterministic Ethernet Communication and dynamic reconfiguration under consideration of deterministic requirements and changing environments with real-time constraints. The work presented here is the follow-up from the work described in the predecessor deliverable D2.2a and the continuation of the work performed in Task 2.2 – Deterministic Ethernet Communication and CPS control/networking led by partner TTTech.

This document will mainly focus on two parts IEEE 802.1 Time Sensitive Networking technology, namely scheduling (targeting the IEEE 802.1Qbv standard) and dynamic (re-)configuration (targeting the IEEE 802.1Qcc standard). As already mentioned in the predecessor deliverable, the IEEE 802.1 standards are still under development, meaning that the applicability of these standards is also still under development and will continue to be over the scope of the AUTOWARE project. Additionally, the work presented belongs to the core technologies of TTTech and is therefore still heavily under development and will remain also so for the coming years.

Scheduling and (re-)configuration are two main parts for constantly changing Deterministic Communication networks and therefore this document will mainly focus on the work performed in these areas. It will provide an overview of the work performed and a description of the available technologies. Additionally, it will give a brief overview of the available network configuration tool available for modelling, configuring and scheduling TSN communication networks.

Keywords

Deterministic communication, Deterministic Ethernet, IEEE 802.1, Time-Sensitive Networking (TSN), dynamic (re-)configuration

Acronyms

CBS	Credit Base Shaper
CNC	Centralized Network Configuration
CPSoS	Cyber-Physical Systems of Systems
CUC	Central User Configuration
GCL	Gate Control List
IIoT	Industrial Internet of Things
ILP	Integer Linear Programming
IT	Information Technology
JSON	JavaScript Object Notation
OT	Operational Technology
SMT	Satisfiability Modulo Theories
SRP	Stream Reservation Protocol
TAS	Time Aware Shaper
TSN	Time-Sensitive Networking
UNI	User Network Interface
VLAN	Virtual Local Area Network
FIFO	Fist-in-first-out
RT-IoT	Real-Time Internet of Things

1 Introduction

1.1 Purpose and Scope

One of the objectives of the AUTOWARE project is to target distributed safety-critical applications and for that it considers Deterministic Ethernet as the communication backbone to guarantee that communication between the different components of the network is reliable and robust at all times.

Nowadays, there is a large collection of Industrial Ethernet protocols available on the market. In most cases, the Industrial Ethernet protocol selected for use in industrial applications or devices differs dependent on the vendor it belongs to, which results in limited compatibility with other devices coming from different vendors. This is the so-called manufacturer lock-in. It forces customers to buy just from a single vendor or to put in a lot of effort getting equipment from different vendors compatible with each other, which many, especially SMEs, don't have the time and money to accomplish.

The introduction of the Industrial Internet of Things (IIoT) and Industry 4.0 demands for larger automation and greater insights into the manufacturing processes and to become more interoperable, flexible and seamless in nature. Additionally, as processes are becoming more and more critical, real-time connectivity has become essential for executing these processes. IEEE 802.1 Time-Sensitive Networking (TSN) has been introduced as a new deterministic communication technology and offers real-time connectivity capabilities that match or even exceed the current Industrial Ethernet protocols, but additionally provides the added flexibility of IEEE standards. TSN as an open standard provides enough functionality and flexibility to fulfill the requests from the industry. TSN can be applied as the common communication protocol that connects industrial equipment from various vendors, simultaneously fulfilling the challenging requirements of current and future industrial applications.

1.2 Contributions to other WPs and deliverables and document structure

The work performed in this task and deliverable has relations with the other work performed in WP2. The partners active in WP2 (i.e. CNR, UMH and TTT) have been trying to incorporate the different technologies developed together into a single demonstrator, which has been the aim of the overall WP. Additionally, the work here has been reflected in WP1, where the AUTOWARE Reference Architecture has been defined and has taken the concept of communication into account. Finally, relations to WP5 has been identified, where the pilots have been considering using the available TSN technologies to be integrated in their demonstrators.

The deliverable is built up as follows:

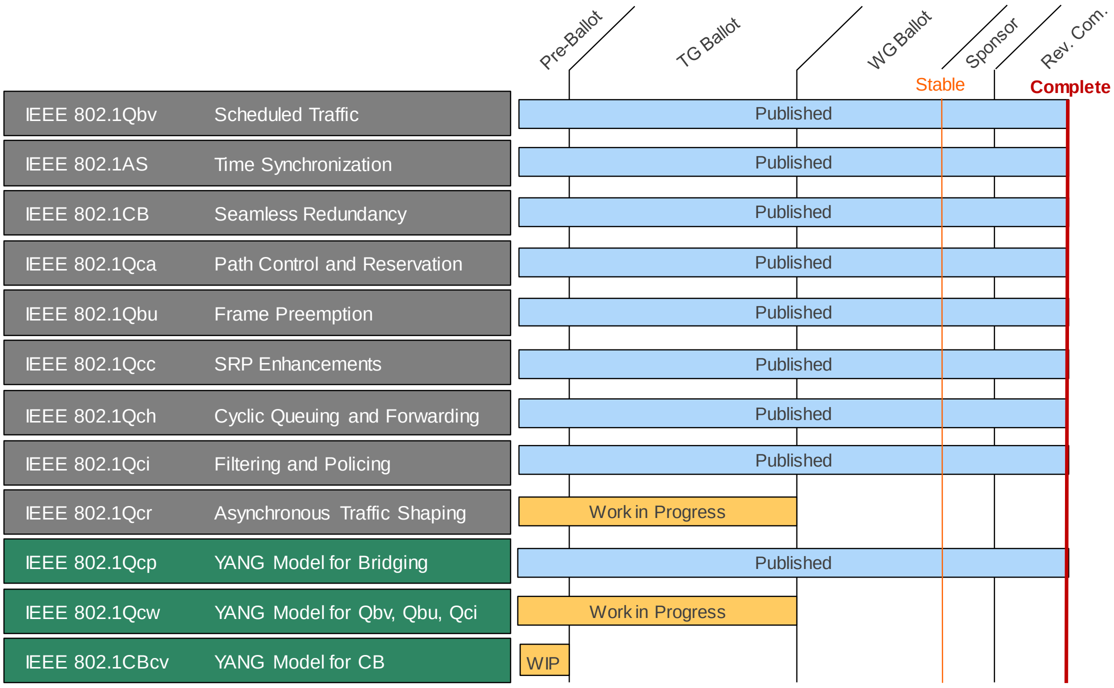
- Chapter 2 gives an overview of the status of TSN and how the different standards have been further developed.
- Chapter 3 goes into more detail on the work performed for the scheduling of Deterministic Ethernet network, which is a crucial technology needed for establishing determinism in communication.
- Chapter 4 presents the updated work on the dynamic (re-)configuration concept developed for deterministic communication networks.
- Chapter 5 provides a brief overview of the network configuration tool available for planning, defining and scheduling networks that provide deterministic communication.
- Chapter 6 gives a description of how TSN has been used in the neutral facility of SmartFactory^{KL}, which is the use case where TSN has been applied inside the AUTOWARE project.
- Finally, Chapter 7 concludes the document and gives a look into future work on Deterministic Ethernet Communication.

1.3 Target Audience

The deliverable is intended to provide an overview of the possibilities of IEEE 802.1 TSN for Deterministic Ethernet Communication, targeting mainly industrial applications, like manufacturing. It is in the first place targeting the AUTOWARE partners providing an overview of TSN and its functionalities. Additionally, the deliverable targets a larger audience, mainly potential users that aim to use this technology in their future plants. These users can vary from system integrators, software developers, hardware technologist but also policy makers that want to push TSN to be the field bus of the future.

2 Continuous Development of TSN

As already mentioned in the predecessor deliverable D2.2a, TSN is a set of IEEE 802.1 sub-standards. This set of standards is under development and new standards are being included in the set. Figure 1 provides a current overview of the current standards available in the IEEE 802.1 and its development status. As can be seen, quite some standards are published, whereas some others are still Work in Progress or in the very early process of definition.



IEEE 802.1Qbv	Scheduled Traffic	Published
IEEE 802.1AS	Time Synchronization	Published
IEEE 802.1CB	Seamless Redundancy	Published
IEEE 802.1Qca	Path Control and Reservation	Published
IEEE 802.1Qbu	Frame Preemption	Published
IEEE 802.1Qcc	SRP Enhancements	Published
IEEE 802.1Qch	Cyclic Queuing and Forwarding	Published
IEEE 802.1Qci	Filtering and Policing	Published
IEEE 802.1Qcr	Asynchronous Traffic Shaping	Work in Progress
IEEE 802.1Qcp	YANG Model for Bridging	Published
IEEE 802.1Qcw	YANG Model for Qbv, Qbu, Qci	Work in Progress
IEEE 802.1CBcv	YANG Model for CB	WIP

Figure 1: TSN Standardization Status (February 2019)

The TSN set of standards can be interpreted as a flexible toolbox, from which a network designer can select only those standards that are needed for the targeted applications, when it is needed. One way to categorize the TSN standards is to sort them in collections based on the support they provide to the main pillars of TSN, i.e. timing and synchronization, bounded low latency, reliability, and resource management. However, these collections will not be disjunct, as some of the standards contribute to more than one concept of TSN. As an example, the IEEE 802.1AS standard addresses time synchronization, but is also relevant to the reliability concept.

Therefore, within this document, the focus will be on a limited selection of standards that are of importance to the work performed in the AUTOWARE project.

- **IEEE 802.1Qbv – Enhancements for Scheduled Traffic:** Scheduling of traffic is a core concept in TSN. Based on the shared global time provided by IEEE 802.1AS, a schedule is created and distributed between participating network devices. IEEE 802.1Qbv defines the mechanisms for controlling the flow of queued traffic through gates at the egress of a TSN switch. The transmission of messages from these queues is executed during scheduled time windows. Other queues will typically be blocked from transmission during these time windows, therefore removing the chance of scheduled traffic being impeded by non-scheduled traffic. This means that the delay through each switch is deterministic and that message latency through a network of TSN-enabled components can be guaranteed. More information is provided in Section 3.

- **IEEE 802.1Qcc – Stream Reservation Protocol (SRP) Enhancements:** The enhancements to SRP include support for more streams, configurable stream reservation classes and streams, better description of stream characteristics, support for Layer 3 streaming, deterministic stream reservation convergence, and User Network Interface (UNI) for routing and reservations. 802.1Qcc supports offline and/or online configuration of TSN network scheduling. More information is provided in Section 4.

2.1 TSN Profile for Industrial Automation

While Operational Technology (OT) is designed to fulfil properties, such as real-time and safety-critical behaviour, reliability, availability, Information Technology (IT) is usually not designed with those properties in mind. Consequently, the challenge is to make OT and IT co-exist. One notable example is communication. In fact, the significant increase in the demand for networking and the availability of high-speed Ethernet equipment, that nowadays is cheaper than the one of special-purpose digital technologies, result in critical traffic flows (e.g. time-sensitive ones) and non-critical traffic flows sharing the same network. TSN is the foundation to provide connectivity to time and mission-critical applications over converged Ethernet networks. With TSN, a network can consist of multiple vendor devices that can interlock and can be configured via a single standard interface. However, in order to deploy converged networks able to simultaneously support OT traffic and IT traffic, developers, vendors and users of interoperable bridged time-sensitive networks for industrial automation need guidelines for selecting features, configurations, protocols, and procedures of bridges, end stations and LANs. The answer to this need is the ongoing standard named IEC/IEEE 60802 – Time-Sensitive Networking Profile for Industrial Automation (TSN-IA), that is a joint project of IEC SC65C/MT9 and IEEE 802.

The IEC/IEEE 60802 standard defines profiles for network bridges and end stations for Time-Sensitive Networking in industrial automation that are based on standards published by IEEE 802.3 and IEEE 802.1. The choice of using the IEEE 802.3 and IEEE 802.1 standards as the building blocks of the lower communication stack layers and the management is to avoid the proliferation of divergent implementations of deterministic real-time Ethernet networks, while exploiting the advantages provided by Ethernet networks in terms of deterministic transmission, data rate availability (from 10 Mbps to 1 Tbps), time synchronization, etc.

At the moment, the IEC/IEEE 60802 standard is still in progress[1] (see Figure 2).

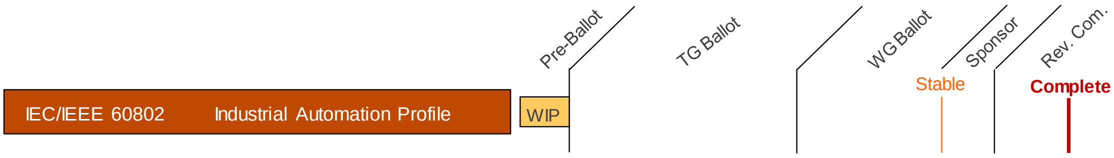


Figure 2: TSN Profile for Industrial Automation Standardization Status (February 2019)

---

[1] https://1.ieee802.org/tsn/iec-ieee-60802-tsn-profile-for-industial-automation/

3 Scheduling

Traffic planning, also called scheduling, is the most important part of the TSN network to guarantee deterministic communication over the network. Here we are concerned with the beforementioned IEEE 802.1Qbv standard, targeting the minimization of transmission latency. Within the research performed for scheduling, it is assumed that the components of the network (i.e. end stations and switches) are capable to execute the time-triggered paradigm with a sufficient level of quality. This may require hardware mechanisms to be in place to ensure the timely accuracy of the execution of transmission and forwarding events in end systems and switches.

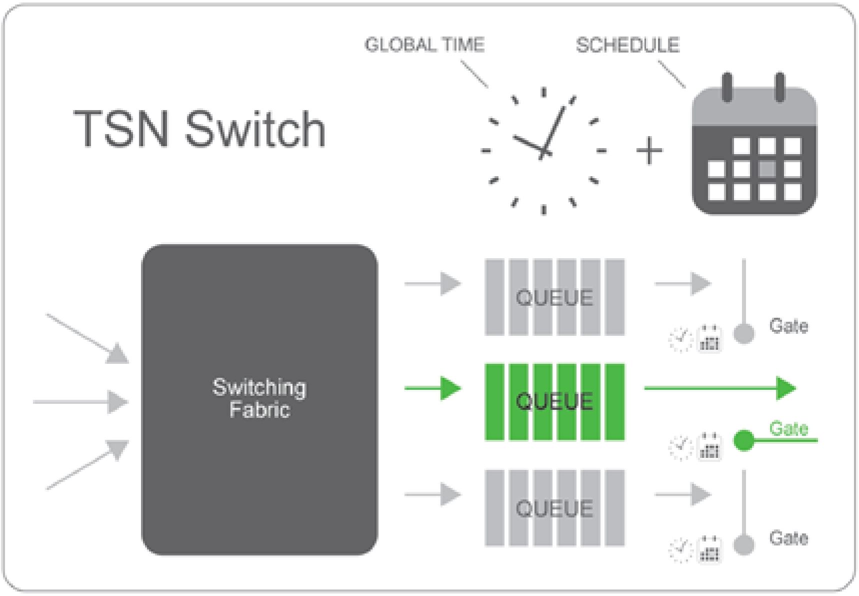


Figure 3: Overview of a TSN switch, with IEEE 802.1Qbv

One approach to solve the scheduling problem is by using general-purpose tools, like e.g., Satisfiability Modulo Theories (SMT)-solvers.

Nodes communicate with each other by the concepts of streams and frames. A stream is a periodic multicast data transmission from one talker (the sender node) to one or more listeners (the receiver nodes). Typically, the sender and receiver nodes are end systems in the network, whereas switches will function as forwarding nodes. While the stream defines the overall end-to-end communication between senders and receivers, the concept of frame identifies a particular message communicated between any two nodes. A frame is characterized by a frame length and a frame period. The period of a frame is equal to the period of the stream, while the length of the frame is calculated based on the data size of the stream and the link speed.

This queue-based model of communication handles frames according to a first-in-first-out (FIFO) paradigm. (see Figure 3). The TSN switch depicted here contains four ports, namely three ingress ports on the left side and one egress port on the right side. The frames on the ingress ports are identified typically based on the information in the Ethernet frame header and assigns them accordingly to the available queues. Additionally, each queue is assigned to a gate, which can be any time be in one of the two states *open* or *close*. When the gate of a specific queue is in the open state, the frames can then be selected for transmission. In case the gate is closed, frames within the respective queues cannot be selected for transmission. The scheduling problem of a switch in TSN time-triggered communication can be summarized as to find points in time for the open and close events of the queues inside the switch.

The Time-Aware Shaper (TAS) defined in IEEE 802.1Qbv [1] is essentially a gate mechanism dynamically enabling or disabling the selection of frames from output queues based on a predefined cyclic Schedule referred to as the Gate Control List (GCL). Within the GCL, the entries in the list indicate for each port and traffic class the points in time when to set the gate state into the open or close state.

Gate states are statically scheduled with respect to a synchronized time and defined at design time of the network. This means, that as synchronized time proceeds, a node continually checks whether a state change for one of its gates is scheduled. If this is the case, the state change of the gate is executed.

Figure 4 provides a schematic overview of an example network, where six end stations are involved (V_1, V_2, V_4, V_5, V_7 and V_8) and two switches (V_3 and V_6)

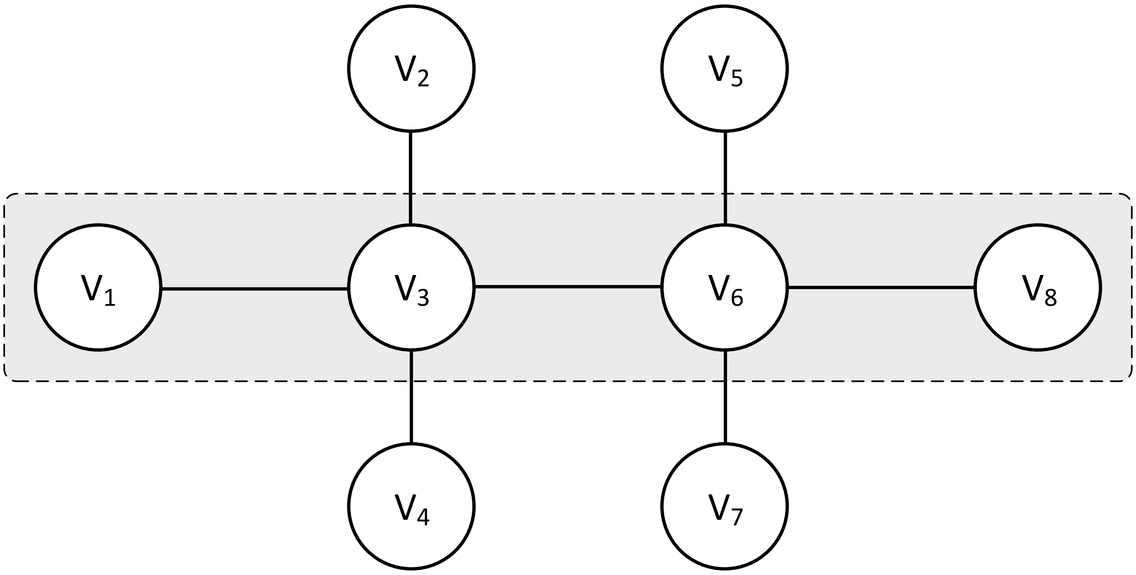


Figure 4: Example network

Figure 5 represents a schedule for the transmission of frame 1 and frame 2 from end stations V_1 and V_2 to V_8, while passing through the switches V_3 and V_6. It shows the time times when the gates are opened and closed, so called windows, when the frames are transmitted. Each window is thus defined by an open and close event, which defines the core of the scheduling part, namely to generate concrete values for these events.

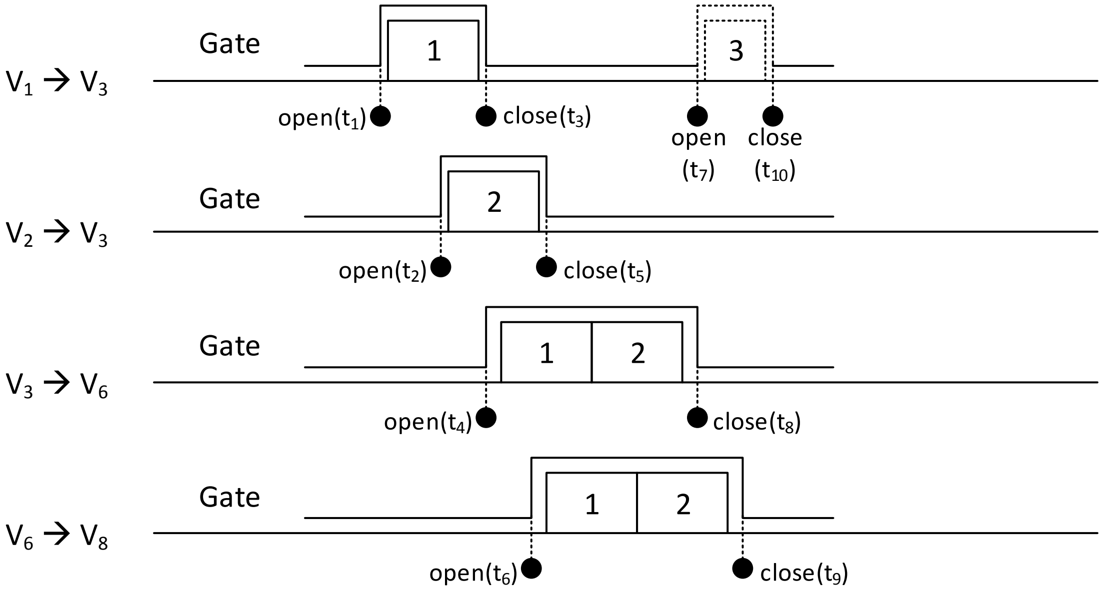


Figure 5: Communication Schedule for TSN network

These open and close events need to be in a certain relation with each other, which can mostly be expressed in a mathematical form. These relations are so called *constraints*. Simple networks (with very few nodes) have only a limited amount of constraints, but as soon as the network starts to grow the constraints will also increase. The number of constraints grows with the number of nodes in the topology and even quadratic with the number of messages. Industrial networks, which contain many switches and end devices (e.g. robots, feeders, conveyer belts, etc.) can easily involve over a million constraints.

Unfortunately, the generation of a schedule for a TSN network in real-life scenarios turn out to be often NP-complete [2], which mean there is no known algorithm that generates the communication schedule in polynomial time. The generation of communication schedules is an active field of research and many results are available. There is a differentiation between two kinds of research going on in this field: 1) the aim to construct specialized search algorithms, e.g., by deploying heuristics, meta-heuristics or genetic algorithms and 2) by applying general purpose tools, like integer linear programming (ILP) or SMT solvers. IEEE 802.1Qbv is still being heavily researched and there are currently only a few, targeting the conversion of scheduling from TTEthernet to TSN [3], formal sets of scheduling constraints [4][5] and schedule synthesis performance numbers based on SMT [6].

4 Dynamic (Re-)Configuration

Nowadays communication networks in Cyber-Physical Systems of Systems (CPSoS) are becoming larger and larger and are also subject to change (e.g. adding or removing new devices). The application areas of these networks can be, for example, smart factories or smart cities. Still, these systems need to uphold to the requirements of Real-Time. Therefore, these systems are also sometimes dubbed Real-Time Internet of Things (RT-IoT).

This brings new features to the communication networks in the beforementioned areas. First, the network should be a real-time network, meaning that the network should have the capability of transmitting data in a deterministic manner. Secondly, the network should be designed in such a way that it can grow in size as much as possible. And finally, the system must be dynamically adaptable, meaning that the topology of the network should be changeable, either by removing or adding new devices to the network, modifying functionality of the network, change communication between different nodes or perform maintenance tasks.

The challenge in these situations is to achieve and to uphold determinism. Typically in such networks to achieve determinism, is to create a static configuration, but these will not be sufficient anymore if the network is undergoing modifications. To establish the required features and current industrial requirements, dynamic configuration and management of real-time networks for CPSoS are being investigated. Here, first findings and thoughts will be discussed.

4.1 IEEE 802.1Qcc – Stream Reservation Protocol

The standard that deals with the configuration of TSN networks is IEEE 802.1Qcc. This standard is an enhancement of the Stream Reservation Protocol (SRP) (IEEE 802.1Qat) designed for the resource management in networks using the Credit Base Shaper (CBS) (IEEE 802.1Qav).

One of the main elements for the configuration of TSN networks is the User Network Interface (UNI). On the user side of this interface are the talkers and the listeners, whereas on the network side of the interface are the bridges. The concept of the UNI is that the user specifies the requirements for the streams that they want to transmit without having all the details about the network. The network then analyzes this information with network capabilities and configures the bridges to meet the user requirements. IEEE 802.1Qcc defines three configuration models (already introduced in the predecessor deliverable D2.2a [7]), which provides the realization of the configuration paradigm:

- Fully Distributed Model: The UNI is located between the Talker/Listener (user side) and the bridge to which it is connected to (network side). The user transmits its requirements and the network propagates them through the relevant paths. The management of bridges is performed locally in the bridges just with the information that is currently available to the respective bridge. This model is used to configure the CBS and for that the SRP can be used as UNI. One limitation of this model is the lack of a centralized view with complete knowledge of the network that makes it not suitable for the configuration of the TAS.
- Centralized Network / Distributed User Model: In this model, the centralized network configuration (CNC), is installed. The CNC can be realized by another end station, which could be connected to any bridge. The UNI remains between the Talker/Listener and the bridge towards which the end systems connect. The configuration requests are not addressed locally anymore by the bridges in the network, but are forwarded to the CNC (which is located in an end system). The CNC generates configurations for all bridges affected by the talker/listener request and provides the configuration to the individual bridges. The CNC thus has a globalized view of the available resources in the network.
- Fully Centralized Network: The Fully Centralized Network introduces another new element, namely the Central User Configuration (CUC). This model places the UNI between the CNC and the CUC, rather than directly at the talkers/listeners. The end stations communicate their communication needs with the CUC that requests configuration updates from the CNC. The CNC produces a new configuration, distributes the configuration over the available bridges in the network and returns some status information to the CUC, regarding the possible configuration. The CUC informs the end systems about the relevant configuration of the network. The CUC and CNC can both be implemented side by side in and end station.

In the previous deliverable D2.2a, we introduced the proposed reconfiguration system for identifying changes in the network (see Figure 6), containing the following functional elements:

- Monitor: Observes the network traffic and gathers measurements to identify traffic patterns. The goal is to recreate from the identified patterns the original real-time constraints defined by the currently running parameters.
- Extractor: Derives traffic parameters based on the traffic patterns observed by the Monitor and previous knowledge of the network and applications. This part is the learning phase of the reconfiguration.

- Scheduler: Uses the traffic parameters obtained by the Extractor to generate a schedule for the network, that maintains and improves the deterministic guarantees.
- Reconfigurator: In charge of updating the network configuration to follow the new communication schedule generated by the Scheduler.

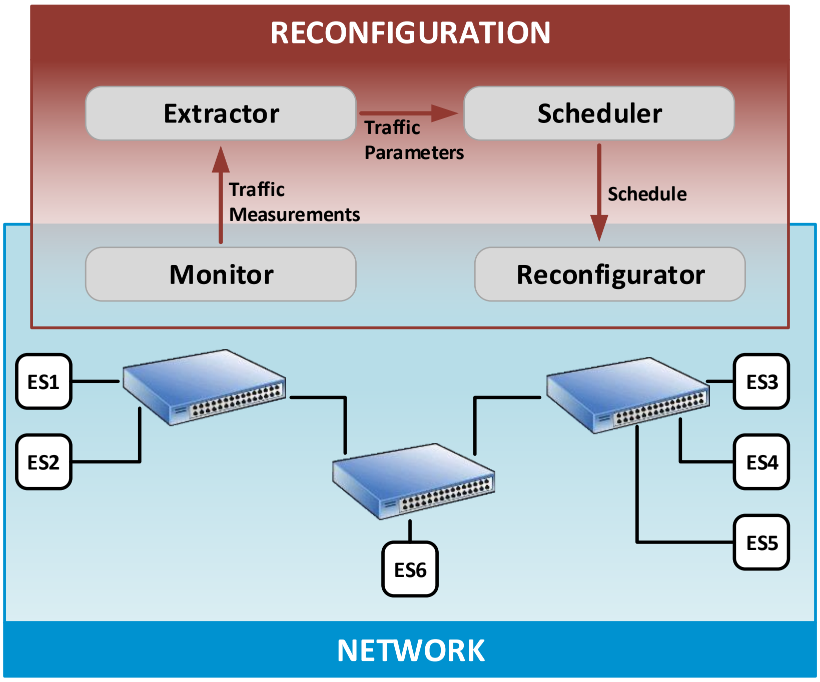


Figure 6: (Re-)configuration System – Identifying changes in the network

The centralized network model, described before and proposed by IEEE 802.1Qcc fit very well with the global view that the Configuration System needs to calculate the new configuration for the network. The Extractor and the Scheduler should be implemented as part of the CNC, as it has the overall view of the whole network. In fact, the CNC will always need to have some scheduling capacity if it is meant to be used to configure the TAS.

In addition to a generic definition of the UNI, TSN also provides a concrete realization of this interface for the centralized models. This concrete formalization is defined by using the data modelling language YANG [8]. These YANG models may be communicated between the Talkers/Listeners or the CUC and the CNC by protocols like e.g. NETCONF or RestConf.

The overall framework for reconfiguration or self-configuration is depicted in Figure 7.

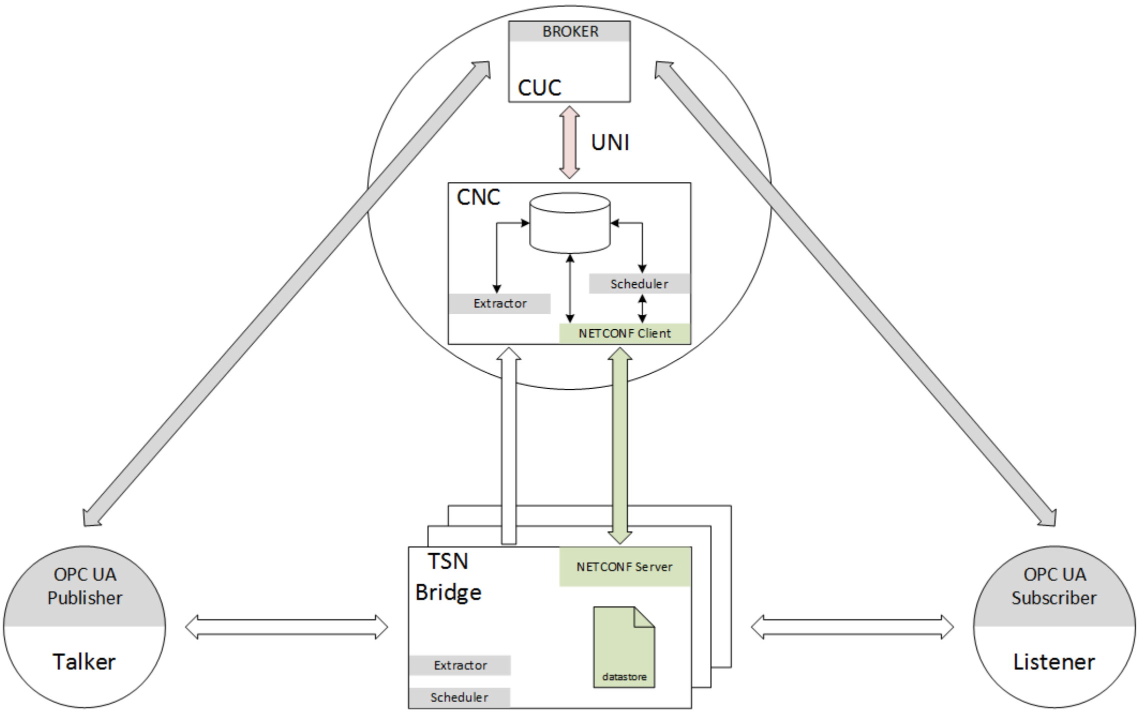


Figure 7: Self-configuration Framework

The self-configuration framework provides us with all the required elements and the ability to have a clear central overview of the topology of the network. A potential workflow of the self-configuration of a network could be the following, starting with the initial configuration of the actual network. We assume that the CNC doesn't have any knowledge about the network. Additionally, the end-systems (i.e. talkers and listeners) don't actively notify the CNC about their communication needs. The CNC (located in an end-system) has the capability to acquire the required knowledge from the network it is attached via a bridge. The initial configuration is as follows:

1) The CNC established a connection with the available bridges using the NETCONF protocol. Using this connection the CNC is aware of the capabilities of the different bridges.
2) The CNC uses the Monitor and Extractor part to extract information from the bridges that are capable of learning information from the network. Using the NETCONF protocol, the CNC changes these bridges behaviour, so they can identify the changes in the network.
3) The specific bridges send the detected and updated information to the CNC, using the available bandwidth in the network.
4) The CNC collects all the information, and as soon as enough information has been collected, it creates an initial configuration for the total network, including a schedule for the (deterministic) communication.

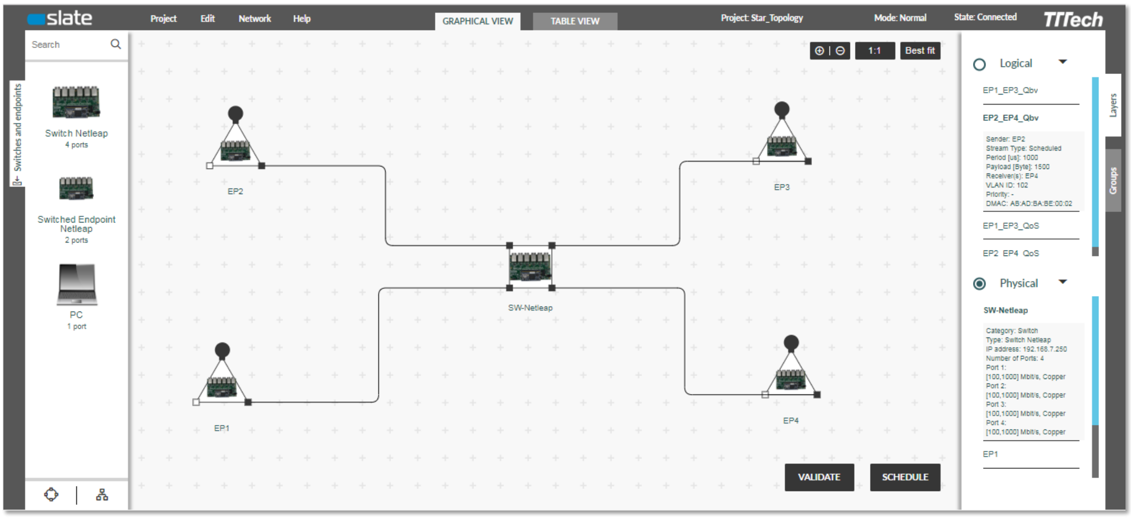
5) The CNC distributes the new configuration of the bridges and the CUC. It uses the NETCONF protocol to communicate the new configuration to the different bridges and the UNI interface to communicate it to the CUC.

After the initial configuration has been distributed over the total network, the bridges are now operating in a mode that enables them to identify or learn about changes in the network. These changes could either be in the topology or in the messages to be sent. If a change occurs inside the network, e.g. changes in the traffic caused by the insertion or removal of devices, the CNC has now two possibilities of detecting these changes. One option is that the added end system notifies the CNC of the change in the network. This will be established by communicating with the CUC using OPC UA. The other option is through the "learning" capabilities of the bridges that are in communication with the CNC using the NETCONF protocol. This could also be done periodically in fixed periods. If a change has been detected, steps 4) and 5) of the initial configuration process will be repeated.

5 Deterministic Communication Network Configuration Tool

The deterministic communication network configuration tool has been introduced in the predecessor deliverable D2.2a. The goal of the tool is to model network topologies, create schedules and deploy configurations for TSN networks.

The tool enables the user to define the physical topology of the network (see Figure 8), so it becomes clear which nodes are part of the network, and how the different components (end-systems, bridges) are connected to each other. It integrates the IEEE 802.1Qbv scheduling concept and supports derived YANG models for IEEE 802.1Qbv and IEEE 802.1Qcp. A collection of devices is being supported, and users can define their own devices using the JSON (JavaScript Object Notation) data-interchange format.



Sender: EP2
Stream Type: Scheduled
Period [us]: 1000
Payload [Byte]: 1500
Receiver(s): EP4
VLAN ID: 102
Priority: -
DMAC: AB:AD:BA:BE:00:02

Category: Switch
Type: Switch Netleap
IP address: 192.168.7.250
Number of Ports: 4
Port 1:
[100,1000] Mbit/s, Copper
Port 2:
[100,1000] Mbit/s, Copper
Port 3:
[100,1000] Mbit/s, Copper
Port 4:
[100,1000] Mbit/s, Copper

Figure 8: Physical topology view

Additionally, the tool enables the user to define data streams and physical connections between the different end-systems and switches. Within the logical topology view (see Figure 9), the user can identify and visualize which end systems are logically connected to each other, so identifying which end systems (i.e. talker) sends messages to which other end-system (i.e. listener).

Furthermore, the system designer can adjust various parameters of the network that influence the communication, such as cable length and type, VLAN ID, period and packet size. Further user constraints can also be defined such as sending time, receiving time, end-to-end latency and transmission gaps. These are all parameters and constraints that have influence on the schedule to be generated.

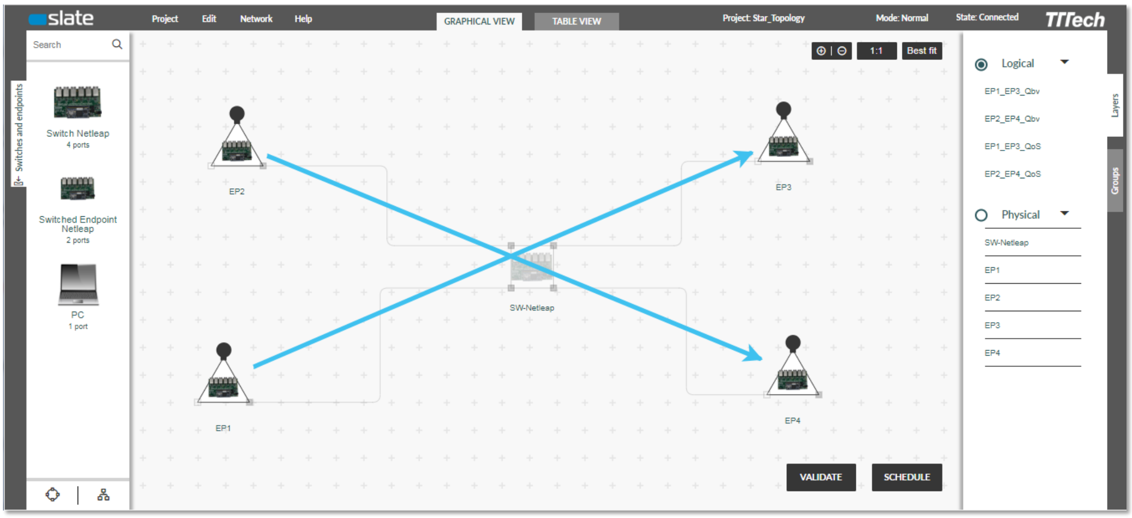


Figure 9: Path/logical topology view

The scheduling engine integrated into the tool have algorithms that use the beforementioned parameter input to create TSN schedules. Incremental scheduling because of new components and/or data streams is feasible. The scheduling engine handles TSN parameters such as gate control and time windows duration. Additionally, it aims to optimize bandwidth for non-scheduled traffic (Ethernet messages), and visualizes the schedule in a graphical user interface (see Figure 10 and Figure 11).

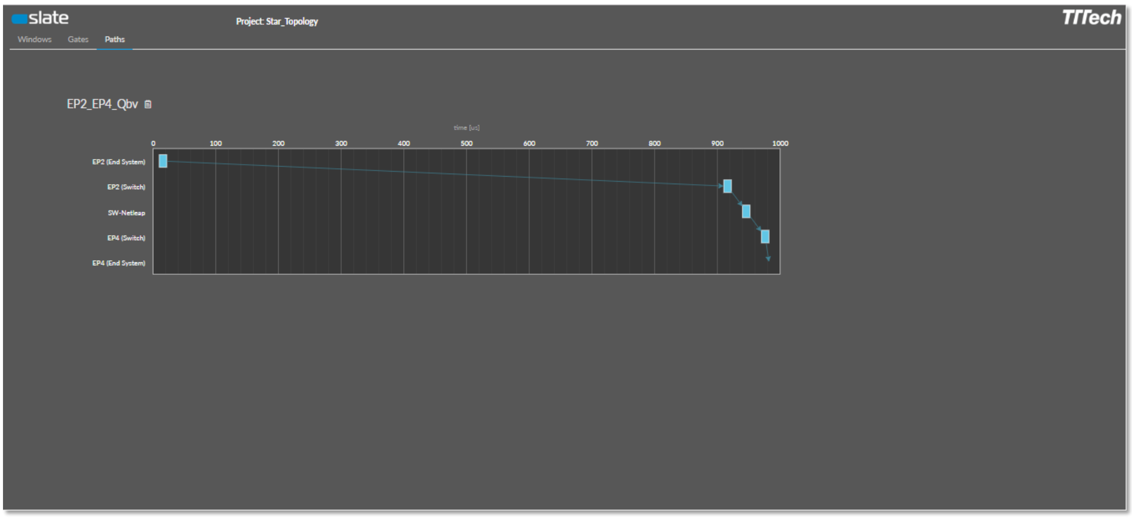


Figure 10: Scheduled path Visualization

Figure 11: Scheduled gates visualization

Finally, information can be shared seamlessly between standard compliant elements. The generated configurations comply to the YANG models for TSN. Additionally, one can automatically deploy configurations to devices (i.e. switches, end systems) that are compliant with NETCONF.

6 Applicability in AUTOWARE

6.1 TSN in SmartFactoryKL

In AUTOWARE, TSN-capable hardware was implemented in the production modules and infrastructure nodes of the SmartFactory testbed. To test the implementation, several high-traffic IP cameras are installed in the modules and connected to the network. The TSN is scheduled to prioritize production and safety relevant functions, but also set single frames of the IP camera on a higher priority for other use-cases within the AUTOWARE project.

At the time of implementation, the IEC/IEEE 60802 TSN Profile for Industrial Automation is not yet finished, thus a suitable subset of standards considering both the use-case and supported TSN features by the hardware had to be chosen.

Switches must support IEEE 802.1Qbv and IEEE 1588v2 allowing to slice the available network bandwidth into timeslots to guarantee determinism to important traffic. End points must only support PTP to synchronize egress traffic with their respective timeslots.

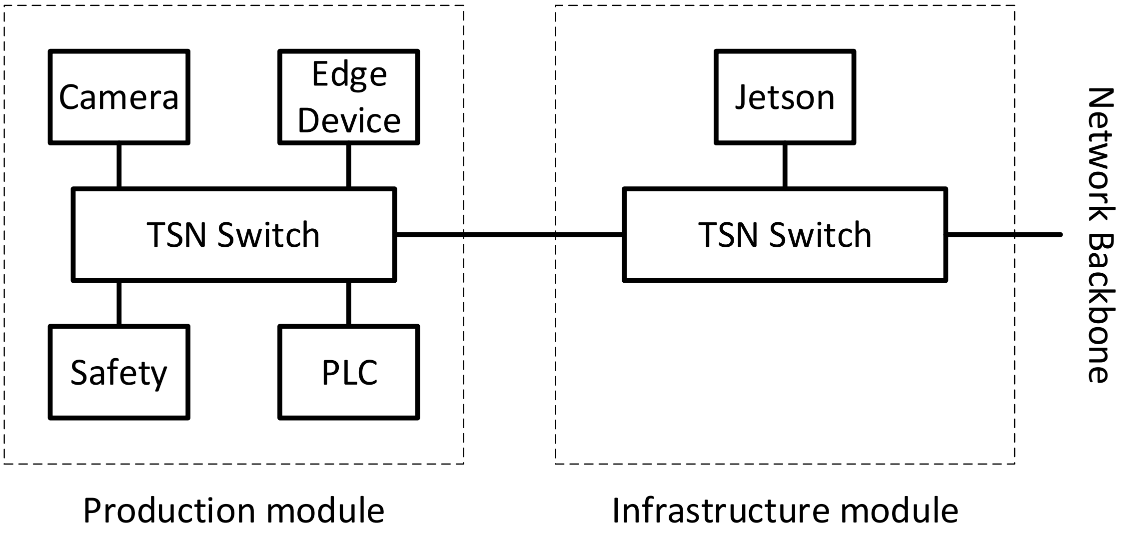


Figure 12: Network Topology

Shown in Figure 12 is the relevant network topology. An edge device connected via a RSPE 35 TSN switch to the Cisco 6050 IP camera streaming to an NVIDIA Jetson TX2 for further analytics at 30 FPS 1080p using a priority of 4. These single frames are sent with a higher priority of 5 of the single shared ethernet cable during the assigned timeslot described in Table 1.

Slot description	Priorities	Interval
Safety, Network Administration	6, 7	100 µs
Camera Streams	3, 4, 5	400 µs
Low Priority (Best Effort, etc.)	0, 1, 2	500 µs

Table 1: Configured gate control list for IEEE 802.1Qbv

Additionally, both hard real-time traffic (safety) and best effort traffic are utilizing the shared network link without any disturbance due to the time-aware schedulers of the switches.

Correct operation and scalability by overloading the camera timeslot with an UDP flood generated was verified by two laptops running the iperf3 tools over the shared link. The higher priority single frames within the same timeslot are not affected, because of strict priority-based scheduling. The same is true for the other tow time slots due to time-aware scheduling. Thus, trying to add too many IP cameras to the network will only impair the video streams, but importantly not the prioritized single frames of each camera of any other network services in other time slots.

7 Conclusions and Future Work

Time-Sensitive Networking (TSN) has been identified as the communication backbone to reshape the industrial communication landscape and create the foundation for the convergence of Information Technology (IT) and Operations Technology (OT). TSN is getting more and more support from the large industries. A large consortium of major suppliers in the industrial automation (incl. Siemens, Rockwell, Beckhoff, etc.) has publicly announced to push TSN together with OPC UA as their future mainstream communication solution[2]. This marks an important milestone that even caused key competitors in industrial automation to proclaim: “The (fieldbus) war is over!”[3].

This deliverable provides an insight into the capabilities and feasibilities of Time-Sensitive Networking and some of its core technologies. As this deliverable has been a follow-up of its predecessor deliverable D2.2a, the focus has been here on two main standards and technologies providing Deterministic Ethernet Communication:

- IEEE 802.1Qbv – Enhancements for Scheduled Traffic
- IEEE 802.1Qcc – Stream Reservation Protocol

These two standards are described in more detail and how they are being used and applied to establish scheduling and (re-)configuration of TSN communication networks. For the (re-)configuration concepts, a first approach has been studied, where the overall and final implementation goes beyond the scope of the project, but first tests will be performed to test the functionality.

TSN has been applied in the SmartFactory^{KL} neutral facility, where it is used to communicate time-critical and non-critical information over the same network.

Future work on TSN involves the updating of the standards (where TTTech is heavily involved), where results from the AUTOWARE project can be of interest. Furthermore, the overall dynamic (re-)configuration concept is still under development and is a topic for future research within TTTech.

---

[2] https://opcfoundation.org/news/press-releases/major-automation-industry-players-join-opc-ua-including-tsn-initiative/

[3] https://www.linkedin.com/feed/update/urn:li:activity:6473453300460056576/

References

[1] IEEE 802.1Qbv – Enhancements for Scheduled Traffic. [Online] http://www.ieee802.org/1/pages/802.1bv.html, April 2019

[2] H. Kopetz and G. Bauer, „The time-triggered architecture“, *Proceedings of the IEEE*, 91(1):112-126, Jan 2003

[3] S.S. Craciunas, R. Serna Oliver, M. Chmelik and W. Steiner, "Scheduling real-time communication in IEEE 802.1Qbv time sensitive networks", in *Proceedings of the 24th International Conference on Real-Time Networks and Systems*, ACM, 2016, pp. 183-192

[4] S.S. Craciunas, R. Serna Oliver and W. Steiner, "Formal Scheduling Constraints for Time-Sensitive Networks", *CoRR*, vol. abs/1712.02246, 2017. [Online]. Available: http://arxiv.org/abs/1712.02246

[5] R. Serna Oliver, S.S. Craciunas and W. Steiner, "IEEE 802.1Qbv gate control list synthesis using array theory encoding", in *2018 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, 2018, pp. 13-24

[6] W. Steiner, S.S. Craciunas and R. Serna Oliver, "Traffic Planning for Time-Sensitive Communication", *IEEE Communication Standards Magazine*, vol. 2, no. 2, 2018

[7] AUTOWARE Deliverable D2.2a, "AUTOWARE Deterministic Ethernet Communications", July 2018

[8] M. Bjorkland, "YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF)", Internet Engineering Task Force, October 2010