Developing a Translation Technique for Converged TSN-5G Communication

Zenepe Satka, David Pantzar, Alexander Magnusson, Mohammad Ashjaei, Hossein Fotouhi,
Mikael Sjödin, Masoud Daneshtalab, Saad Mubeen
Mälardalen University, Sweden
{zenepe.satka, mohammad.ashjaei, hossein.fotouhi, mikael.sjödin, masoud.daneshtalab, saad.mubeen}@mdh.se

***Abstract*—Time Sensitive Networking (TSN) is a set of IEEE standards based on switched Ethernet that aim at meeting high-bandwidth and low-latency requirements in wired communication. TSN implementations typically do not support integration of wireless networks, which limits their applicability to many industrial applications that need both wired and wireless communication. The development of 5G and its promised Ultra-Reliable and Low-Latency Communication (URLLC) integrated with TSN would offer a promising solution to meet the bandwidth, latency and reliability requirements in these industrial applications. In order to support such an integration, we propose a technique to translate the traffic between TSN and 5G communication technologies. As a proof of concept, we implement the translation technique in a well-known TSN simulator, namely NeSTiNg, that is based on the OMNeT++ tool. Furthermore, we evaluate the proposed technique using an automotive industrial use case.**

***Index Terms*—TSN, 5G, URLLC, 3GPP, TSN Translator.**

I. Introduction

Many industrial applications require convergence of wired and wireless networks with deterministic end-to-end latency [1], [2]. Such a converged network can lead to a more transparent network communication, allowing parts of the Operational Technology (OT) and Information Technology (IT) sectors in a smart factory to have a homogeneous layout. In today's industrial networks, the OT domain is made up of 90% vendor-locked wired technologies with limited throughput [3]. With parts of the smart factory network being wireless, one near-term benefit would be the significant reduction in utilization of cables, which in turn would reduce production costs [1].

Consider the automotive domain where an autonomous mine or a quarry consists of several autonomous vehicles and their control center. These vehicles can be equipped with numerous high data-rate sensors that can generate hundreds of megabytes of data per second (e.g., radars, Lidars and video cameras). Furthermore, the large amount of data acquired from these sensors needs to be communicated with predictable and low latencies between the computing units within the vehicles as well as among the vehicles and their control centre. Similar applications can be found in the other domains. In these applications, the IEEE TSN standards[1], based on the switched Ethernet, stand out as a promising solution to provide high-bandwidth and low-latency onboard communication [4]. Similarly, 5G offers a promising solution to support low-latency communication among these vehicles as well as between each vehicle and its remote control center. A converged TSN and 5G network can meet the high-bandwidth and low-latency end-to-end communication requirements, lower the number of vendor-specific requirements, and introduce a greater level of flexibility in the network communication.

In order to support such a converged end-to-end network communication, we propose a technique to translate the traffic between TSN and 5G communication technologies. This translation acts as a gateway between the two technologies by taking the necessary properties from TSN and mapping them to the 5G Quality of Service (QoS) according to the 3GPP specifications [5] and vice versa. We present a proof-of-concept implementation of the proposed technique in a commonly used TSN network simulator NeSTiNG [6] that is based on OMNeT++. Furthermore, we evaluate the translation technique using an automotive industrial use case. We show that the technique can assist network designers to evaluate various holistic TSN-5G network configurations.

II. Background and Related Work

A. Time Sensitive Networking (TSN)

TSN is a set of standards based on switched Ethernet. It supports high-bandwidth and low-latency communication, gaining attention in time-critical industrial applications such as in the industrial automation [7] and automotive domains [4], [8]. To improve the QoS of Ethernet, the TSN task group proposed several features; e.g, time-aware traffic shaper (IEEE 802.1 Qbv), clock synchronization (IEEE 802.1AS), frame preemption (IEEE 802.1Qbu), and path control and reservation (IEEE 802.1Qca), among others.

The TSN bridges (switches) are time-synchronized using IEEE 802.1AS. There are eight different classes of priority for TSN frames. This priority is defined using the Priority Code Point (PCP) field added in 802.1Q-2018 VLAN. We use two scheduling techniques available in TSN: the AVB Credit-Based Shaper (CBS) standardized in 802.1Qav, and the Time-Aware Shaper (TAS) based on Time Division Multiple Access (TDMA), where critical and non-critical traffic are assigned different time slots. For each priority queue, there is a gate that controls the egress data flow. A Gate Control List (GCL) contains the gate states of each queue at each time slot. A model of the functions of TSN bridge is presented in Fig. 1. It contains traffic class queues, a transmission scheduling algorithm, and a gate control list.

[1]https://1.ieee802.org/tsn/

The queues are numbered from 0 to 7 and Best Effort (BE), with 7 being the highest priority queue.

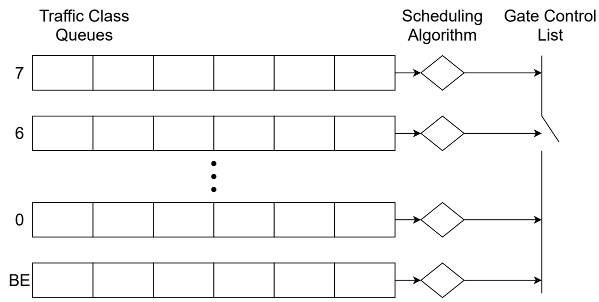


Fig. 1: A TSN bridge model with the traffic class queues, transmission scheduling algorithm and gate control list.

To be able to guarantee that the schedule is followed and no frame/message[2] exceeds its time slot, a guard band of the size of the next message, or maximum message size is included at the end of each scheduling cycle. Such a schedule can be obtained by using existing methods, e.g., [9]. On the other hand, CBS is based on a token bucket shaper. When messages are pending in a queue the credit increases, while it decreases when the messages are transmitted. This helps in preventing starvation of low-priority traffic and allows predictable traffic transmission.

B. 5G and URLLC

The fifth generation of wireless communications (5G) [10], [11] provides significant improvements to the long term evolution (LTE) technology. It is designed to achieve low latency and reliability, providing the built-in flexibility required by Industry 4.0. 5G includes three generic services: enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and URLLC, [12], [13], [14]. The eMBB supports high data rates, higher user mobility, high density, and fixed-mobile convergence. The mMTC provides efficient connectivity for a massive number of heterogeneous IoT devices with a variety of characteristics and demands. URLLC is a set of features for 5G to support critical applications with low-latency and reliability requirements. The standardization for URLLC started with 3GPP Release 15 and evolved until Release 17. With URLLC features, the new 5G Radio Access Network (RAN) [15] can achieve ultra-low latency down to 1 ms and reliability up to 99.9999%. Within the core network, latency is typically below 1 ms [16]. The desired QoS requirements for URLLC depend on the applications as shown in Table I.

C. 5G as a Logical TSN Bridge

Integration of 5G into TSN is based on either having 5G with the capabilities of TSN or integrating 5G as a logical TSN bridge. In the first approach, 5G is seen as a cable link between the devices [5], while in the second approach, 5G is seen as a black-box TSN bridge. 5G as a logical TSN bridge approach is the most focused in the existing works [17], [18]. 3GPP provides two design approaches for using 5G as a logical TSN bridge [19]. The first approach has the translator located within the User Plane Function (UPF). In this architecture, the UPF and TSN Translator (TT) are seen as one component, where the translation of parameters between TSN and 5G takes place within the UPF. In the second design approach, the TSN translators are established on the device-side (DS-TT) and the network-side (NW-TT) of the logical TSN bridge. From the user plane's (UP) perspective, the logical TSN Bridge is a virtual tunnel between the UE and the TSN Network. The DS-TT translates the necessary parameters of TSN to 5G QoS to establish the message's priority on the device-side. This is transmitted from the RAN to the network-side, which holds the TSN network. The NW-TT handles the translation from the 5G QoS to TSN QoS so that the frame maintains correct priority within the integrated network [5]. The introduction of the TSN translators at the device-side and network-side makes it possible to reuse many of the existing interfaces defined for the 5G systems. Integrating the translator at the UPF would require the 5G system functionalities to communicate via the Session Management Function (SMF) [19].

[2]We consider the messages that fit only one frame.

TABLE I: Expected QoS requirements for URLLC [14], [20].

Industry	Error Rate/Reliability	Latency (ms)
Augmented/Virtual Reality	10^{-5} - 10^{-3}	5 - 10
Autonomous/guided vehicles	$\geq 10^{-3}$	5 - 10
Automated Industry	10^{-9} - 10^{-5}	1
Internet of things/Tactile Internet	10^{-5}	1

D. Related Works

There are very few works that focus on the integration of TSN and 5G communication. Most of the works consider challenges related to clock synchronization, while bridging between the two domains has received little attention. For example, Schüngel *et al.* [21] consider the integration of 5G as a TSN virtual bridge. They provide a single message mechanism for signalling timing information through the virtual TSN bridge leveraging the underlying synchronization of the 5G system. For evaluation, they use a discrete event simulator OMNEST [22] (a commercial version of OMNeT++).

There are very few works that address simulation of integrated TSN and 5G networks. Ginthör et al. [23] present a system-level simulator considering the impact and requirements of TSN end-to-end systems. They use OMNeT++ with NeSTiNg model to simulate a TSN-5G network by following the specifications of 3GPP Release 15. By converting a 4G architecture, they added characteristics such as Ethernet PDU sessions and packet filter sets supporting MAC addressing, mini-slots, high-reliability modulation, and 5G quality-of-service indicators, which are required for 5G communication. Their simulation setup consisted of multiple user equipment that are connected to one base station with a strict prioritization scheme. In comparison, we do not convert 4G to 5G. Furthermore, we consider 3GPP Release 16, which provides more details for time-sensitive communications.

Martenvormfelde et al. [24] present a simulation model for integrating 5G into TSN as a transparent bridge. They utilize OMNET++, NeSTiNg and a 5G user plane model. Their bridge model is limited to the user plane, derived from the 3GPP 5G architectural model, and is capable of uplink and downlink traffic. Certain characteristics of the New Radio frame structure and sub-carrier spacing of their model affected the end-to-end delay, even in smaller networks. This paper claims that to provide QoS guarantees in large networks, the model should handle 5G quality-of-service indicators, enabling priorities and queues similar to the TSN IEEE 802.1Q. To do so, we map the TSN QoS to the 5G QoS.

To the best of our knowledge, the research on TSN-5G integration is still in its infancy, mainly focusing on timing information and not on the traffic mapping with QoS management. We present a technique to translate the traffic between TSN and 5G domains considering the properties in the 3GPP-R16 specifications. Furthermore, we provide a proof-of-concept implementation of the technique in a well-known simulator for TSN.

III. TSN-5G Translator Design

In this section, we present the design of the TSN-5G translator. First we describe the focused QoS parameters. Then we present the translator's design and its proof-of-concept implementation in OMNET++ simulator.

A. 5G QoS Indicators (5QIs)

The 5G QoS indicators is a list of parameters representing commonly used values for certain types of traffic [5]. Some of the 5QIs focused in this work are as follows.

- *Resource Type:* In 5G, the resource type parameter indicates how the Packet Delay Budget, Packet Error Rate, and Maximum Data Burst Volume should be handled. The resource can be of type Guaranteed Bit Rate (GBR), Non-GBR, or Delay-Critical GBR.
- *Default Priority Level:* The priority level indicates the scheduling priority of a QoS message. The standardized 5QIs assign their own default priority values, indicating the highest priority message with the lowest value of the default priority level parameter.
- *Packet Delay Budget (PDB):* It defines an upper bound on how long a packet can be delayed between the User Equipment (UE) and the User Plane Function (UPF). The UPF represents the communication scheme between the base station (gNB) and the NW-TT.
- *Packet Error Rate (PER):* It defines the level of reliability by providing an upper bound on the number of messages that can be processed and sent by the 5G node, but never arrive at their intended destination.
- *Default Maximum Data Burst Volume (MDBV):* It indicates the amount of data that can be sent within a PDB.
- *Default Averaging Window:* Indicates the calculation time of Guaranteed Flow Bit Rate (GFBR) and Maximum Flow Bit Rate (MFBR). 5G is expected to provide a guaranteed bit rate that is represented by GFBR. The MFBR defines the maximum value of an actual bit rate.

A representation of 5QIs for Delay-Critical GBR resources that are recommended for integration with TSN is shown in Table. II. The listed parameters are part of a more extensive list of statically assigned parameters specified in the 3GPP specifications [5]. These parameters can also be set dynamically for highly specified scenarios.

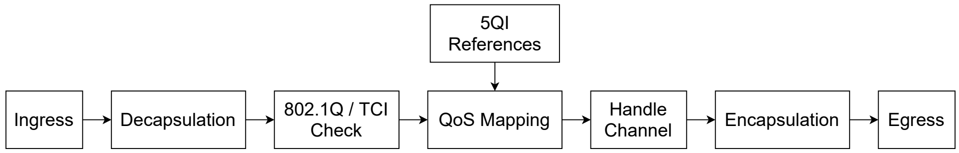
TABLE II: 5QIs for standardized Delay-Critical GBR.

5QI Value	Resource Type	Default Priority Level	PDB	PER	MDBV (bytes)	Default Averaging Window
82	Delay Critical GBR	19	10 ms	10^{-4}	255	2000 ms
83	Delay Critical GBR	22	10 ms	10^{-4}	1354	2000 ms
84	Delay Critical GBR	24	30 ms	10^{-6}	1354	2000 ms
85	Delay Critical GBR	21	5 ms	10^{-5}	255	2000 ms

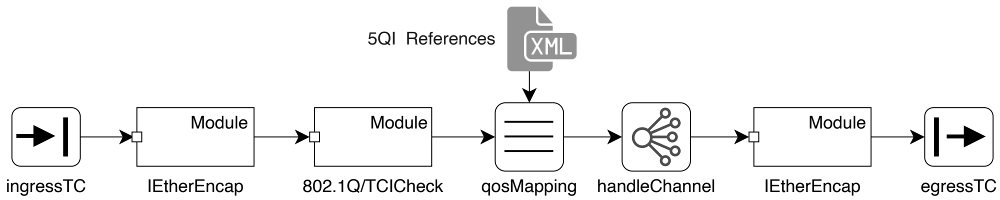
We focus on the user plane and translation of incoming frames so that they can be transmitted between TSN and 5G networks. Mapping of QoS messages to maintain the priority of the message in both networks is done by checking the Priority Code Point (PCP) value in the TSN frame. This value indicates how the 5G system should change its 5QI, which are parameters pre-determined as discussed in Table II. Once a frame arrives at the translator, it is first determined what type of translation has to be done by checking the data's interface. There are two possible types of translation in this scenario: *i)* TSN to 5G translation, and *ii)* 5G to TSN translation. As a proof of concept, each component of the translator is presented as a model or sub-model in OMNeT++ simulator.

1) TSN to 5G - Translation Flow: To guarantee that the QoS in both 5G and TSN message is upheld, there are specific attributes of TSN that need to be mapped to the 5QIs and vice versa. The TSN to 5G translation design is presented in Fig. 2(a), whereas its proof-of-concept implementation in OMNET++ is depicted in Fig. 2(b). The Ingress and Egress modules in the translator handle the reception and transmission of the message respectively. These two modules are realized in OMNeT++ with the egressTC and ingressTC sub-modules as shown in Fig. 2(b) respectively. The contents of the message received at the Ingress module are decapsulated by the Decapsulation module. Similarly, the message contents are encapsulated by the Encapsulation module before the message is transmitted by the Egress module. These two modules are realized by the IEtherEncap module in OMNET++. We introduce a new module, namely Tag Control Information (TCI) of the 802.1Q Header or 802.1Q/TCI Check module, after the Decapsulation module as shown in Fig. 2(a). The 802.1Q/TCI Check module checks the priority level of the message received from the TSN network. Based on the priority level, the QoS requirement is mapped to the 5QI reference established earlier.

The 5QI reference is a pre-configured XML document representing the parameters that should be configured in the logical TSN bridge. The 802.1Q/TCI Check module is realized in OMNeT++, as shown in Fig. 2(b). A QoS mapping



(a) TSN to 5G Translation Flow.



(b) A proof-of-concept implementation of TSN to 5G Translation Flow in OMNeT++.

Fig. 2: The translation flow from TSN to 5G networks.

algorithm is applied between the priority level of the TSN message and the default priority level parameter stored in the pre-configured XML document. As this implementation does not have access to a 5G medium, the translator instead configures the channel within the logical TSN bridge to act as a 5G medium. This is done in the Handle Channel sub-module as shown in Fig. 2 (a). The handleChannel module is realized in OMNeT++, before encapsulating the message contents by the IEtherEncap module as in Fig. 2 (b). The channel gets the 5QI parameters as listed in the 3GPP standardized delay-critical GBR Table II. Once the channel is configured correctly, the translator then encapsulates the message and sends it over the channel.

Mapping a TSN QoS message to a 5QI requires a systematic technique to check its priority level and handle it via specific QoS references. In this regard, the translation from TSN to 5G and corresponding QoS mapping is discussed in Algorithm 1. The structure of a TSN message is shown in Fig. 3, where the first 24 bytes consists of a preamble, destination MAC (DST MAC) and source MAC (SRC MAC) [25]. The 802.1Q header contains the TCI which identifies the data fields that state how the message should be prioritized. Lastly, the Ethernet Type (ETH Type), the Payload (Data), and the CRC fields.

8 bytes	6 bytes	6 bytes	4 bytes	2 bytes	46-1500 bytes	4 bytes
Preamble	DST MAC	SRC MAC	802.1Q HDR	ETH type	Payload	CRC

Fig. 3: TSN message structure.

To map the message to a 5QI, the message is de-encapsulated down to the 20th byte, i.e., the Preamble, DST MAC, and SRC MAC are identified. The next 16 bits correspond to the TPID field, which has the value of 0x8100 for a TSN message [25]. The next 3 bits represent the PCP field containing the TSN message's priority value (0-7). This field is part of a larger field called Tag Control Information (TCI), which also contains the Drop Eligible Indicator (DEI) and the VLAN Identifier (VID). However, in our prototype implementation, it is assumed that all packets are not dropable and belong to the same VID. This is done to reduce the number of parameters in the initial stages of the implementation. The structure of the 802.1Q Header Frame is shown in Fig. 4.

Algorithm 1 TSN to 5G translation flow

begin
qosMapping_List $\leftarrow 5QIReferences$
for all *Messages_at_the_ingress_port* **do**
  decapsulate down to the TPID
  decapsulate next 3 bits
  TCI_check PCP value in the TSN message
  assign 5QI to message.PCP
  handle_channel with the 5QI parameters
  encapsulate message
  send message_to_the_egress_port
end for
end

16 bits	3 bits	1 bit	12 bits
TPID	TCI		
	PCP	DEI	VID

Fig. 4: 802.1Q header frame structure.

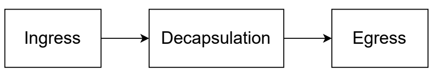
Once the PCP value has been derived from the frame, it must be mapped to a 5QI. The TSN message is listed as a Delay-critical GBR [26] to pre-allocate dedicated network resources to TSN.

2) 5G to TSN - Translation Flow: 5G utilizes GPRS Tunnelling Protocol (GTP) to encapsulate the frames sent over a tunnel. In the user-plane, the GTP-U version is used [5]. The GPRS Tunnelling Protocol User Plane (GTP-U) frame structure is presented in Fig. 5. It starts with an outer header that specifies the source (SRC) and destination (DST) addresses. The QoS Flow Identifier (QFI) and the Tunnel Endpoint Identifier (TEID) are included in the GTP-U header. The QFI represents the priority level of the message, while the TEID indicates the tunnel ID for the PDU session anchor. Part of the 5G frame structure is also the IP message with its DST and SRC IP. In our case, the payload of the 5G GTP-U frame structure is indicated by the TSN frame that we aim to transmit over 5G.

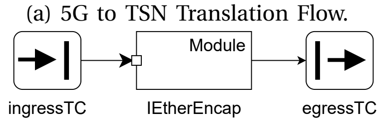
Outer Header		GTP-U Header		IP-Package		Payload
GTP-DST	GTP-SRC	QFI	TEID	IP-DST	IP-SRC	TSN-Frame

Fig. 5: 5G - GTP-U Frame Structure.

The 5G to TSN translation is done by decapsulating the frame, removing the GTP-U and IP Header, to make the frame function as a TSN frame again. The frame is then sent to the TSN bridge which can maintain the QoS by prioritising the frame appropriately depending on the PCP value. The 5G to TSN translation is graphically depicted in Fig. 6 and algorithmically presented in Algorithm 2.



(a) 5G to TSN Translation Flow.



(b) A proof-of-concept implementation of 5G to TSN Translation Flow in OMNeT++.

Fig. 6: The translation flow between 5G and TSN networks.

Algorithm 2 5G to TSN translation flow

begin
for all *Messages_at_the_ingress_port* **do**
decapsulate down to the payload
send message_to_the_egress_port
end for
end

IV. Simulation Development

In this section, we present our simulation development, starting with describing the simulation environment, and then explaining the simulation setup of a 5G node inside a TSN network environment. All the flow information is brought together to the finalized design implemented in OMNeT++/NeSTiNg. The source code for the developed TSN-5G simulator is openly provided in gitlab[3].

A. Simulation Environment - NeSTiNg

The OMNeT++ simulation environment offers ease of extension to incorporate various network protocols thanks to its modular architecture. The translation of traffic between 5G and TSN is implemented in OMNeT++ by leveraging NeSTiNg, which is a TSN simulation framework built over OMNeT++ [6]. NeSTiNg is built as an enhancement of the Ethernet protocol provided by the INET framework. The main features of TSN supported by NeSTiNg are scheduling, gate control, queuing, and frame preemption.

To make full use of the simulation model, we perform a study on the capabilities and limitations of NeSTiNg. For scheduling of various traffic in TSN, NeSTiNg supports both Credit Based Shaper (CBS) and Time Aware Shaper (TAS) [6]. One of the main challenges in NeSTiNg is the lack of global simulation clock, which is needed as all TSN bridges are synchronized. NeSTiNg does not support the IEEE 802.1AS standard [27] for time synchronization in TSN networks. Furthermore, it neglects two major TSN implementations, which are IEEE 802.1CB Frame Replication and Elimination for Reliability (FRER) and IEEE 802.1Qci Per-Stream Filtering and Policing (PSFP).

B. Simulation Setup

The implementation is done in OMNeT++ by integrating modules and sub-modules into a network. Only some of the modules had to be re-written to fulfil the requirements of the translator: IEtherEncap and handleChannel. The 5G Node contains an Ethernet gate (ethg), the TSN-Translator (TT), and the message dispatcher between the two. It also has a few sub-modules including interfaceTable, filteringDatabase, scheduleSwap, oscillator, legacyClock, and the clock. The interface table and the filteringDatabase indicate where the traffic should go once it has been handled. The indication is done by establishing port and destination mapping rules in an XML document. The oscillator and clock modules deal with the time ticks in the modules and synchronize to the simulation time for time-stamping of logged data. A visual representation of the 5G node is shown in Fig. 7, where various entities indicate submodules, the horizontal line is the message dispatcher, and the arrows are channels connecting the modules. Note that the submodules shown on the left side of the horizontal line in Fig. 7 do not require any connectors and should be seen as a way to access information elsewhere in the system.

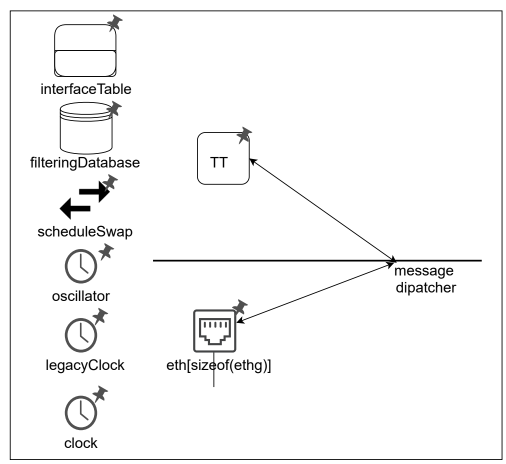


Fig. 7: 5G Node in OMNeT++.

The line underneath the eth submodule leading to the module's border indicates the possibility of other modules, such as VlanEtherSwitchPreemptable (TSN Bridges), to connect to the 5G Node. Note that the arrows in the 5G node are bidirectional, i.e., both the TT and the eth modules can send and receive data.

Each of the gates connects to a channel; this channel can be seen as a submodule with established parameters. They are initialized to default values, which can be changed during runtime. These parameters include delay of transfer,

[3]https://gitlab.com/DavidPantzar/5GTSNTranslator

packet error rate, or data rate. One of the TSN-Translator (TT) functions is to configure these parameters of the channel to correct values to simulate the max-delay of the PDB to establish a simulated 5G transfer. Similarly, as to how the QFI would change the parameters in an actual 5G node, the TT looks at the incoming transmissions PCP value and changes the channel parameters to pre-determined values that are set in the XML files.

5QI - XML Integration: When the TT function is initialized, it collects the data structures of pre-determined QoS parameters of the channel in the XML file. The XML file also allows the users to establish these parameters beforehand. This XML file is akin to the 5QI values in an actual 5G node. As this proof-of-concept implementation of the translator focuses on the user-plane, an XML file loaded during initialization is an elegant solution. The loaded file contains the delay, data rate, and PER of the channel. The translator chooses the parameters that correspond to the PCP value being sent through the device. There are eight different levels of PCP in TSN. Hence, eight 5QIs can be used. The delayPar parameter sets the delay of the channel to the indicated value, the errorRatePar sets the packet error rate, and the dataratePar sets the datarate. A detailed description of the implementation in simulation environment is presented in [28].

V. Evaluation: Automotive Industrial Use case

This section presents an industrial use case that is used to evaluate the proposed technique and its proof-of-concept implementation.

A. Use Case Setup

The use case is a part of an autonomous recycling site that contains several autonomous vehicles (recycling cranes and haulers). Each vehicle uses TSN for onboard backbone communication. The vehicles communicate with each other and with their remote control center using 5G. We consider a part of one of these vehicles and the remote control center as shown in Fig. 8.

There are four nodes that are connected to a TSN bridge within the vehicle. Similarly, there are two nodes that are connected to one TSN bridge in the remote control center. Within the vehicle, the actuator node (A) is controlled by a camera input that is acquired by node (S). The camera node is connected to an aggregation node (Agg) that performs aggregation of data and computation of control signals. The 5G gateway node (G1) is responsible for communicating with the 5G gateway node (G2) in the remote control center. The Remote Computer node (RC) in the remote control center computes the actuators' states in each vehicle. The visual representation of the use case in the simulation environment is depicted in Fig. 9.

The traffic flow in the use case is as follows:

(S)–>(AGG)–>(G1)–>(G2)–>(RC)–>(G2)–>(G1)–>(A)

The node (S) sends its data to the (Agg) node that, in turn, sends the computed data to the (RC) node through the 5G network. The (RC) node then sends a message back to the vehicle through the 5G network to node (A). We focus on investigating the channel propagation delay in the message that originates from node (S) and terminates at the actuator node (A).

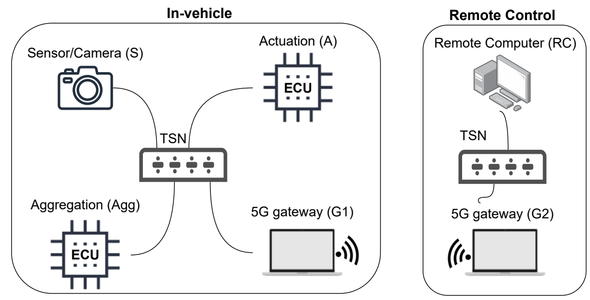


Fig. 8: Automotive industrial use case utilizing TSN & 5G.

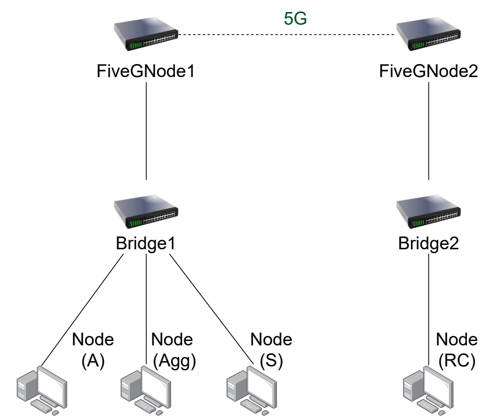


Fig. 9: Model of the use case in the simulation mode.

B. Simulation Parameters

The simulation parameters are set either on the channels or in the omnetpp.ini file. Each wired channel has a 100Mbit/s bandwidth. The transmission delay of a TSN message with maximum payload (1546 Bytes) on each link is equal to $123\mu s$. The processing delay in each of the node (S), (Agg), (A) and (RC) is equal to $20\mu s$, while the processing delay in each of the nodes G1 and G2 is set to 0.5ms to simulate maximum latency of L2/L3 flow as per 3GPP to fulfil the URLLC requirements in the UP [29]. The 5G channel delay is set to the parameters indicated by the XML-file. The TSN bridges are set with a pre-configured offline schedule. The gates on the TSN bridges are set to StrictPriority, which indicates that they both check if a gate is open and the PCP value of the incoming traffic to determine which message to prioritize. The end-to-end deadline of the message of interest is 50ms according to the requirement specification of the use case.

C. Scheduling Parameters

The schedule used in this use case is created to showcase that the tool functions even with gates not being open at all times. The (S) and (Agg) assume the same priority and set their PCP values to 1, while the (RC) node has its traffic assigned to a PCP value of 2. Each of the devices is set with a period of 10ms and varying offsets depending on the arrival time of the previous message. For example, the message sending task in (RC) has an offset of 2500 μs. This offset corresponds to the time it takes for the message to arrive from the (Agg) node. Table III shows the period and offset of each of the task in the corresponding nodes. The scheduling sub-module was already part of the NeSTiNg packet in OMNeT++ and was derived from the simulation environment [6]. The schedule is designed so that the gates are open as the message arrives at Bridge1 but the gates open with a slight delay when a message arrives at Bridge2. This was done to simulate a potential configuration where other messages with different priorities go through the port. The scheduling works by having gates opened or closed for a certain length over a set cycle. The gates are represented by bit-vectors where the value '0' indicates the closed state of the gate, and the value '1' indicates that the gate has an open state.

TABLE III: Scheduling parameters.

MessageID	Name	Start time (μs)	Period (μs)
1	Camera	10	10000
2	Aggregator	300	10000
3	Remote Control	2500	10000

D. Evaluation Results

We focus on two main aspects in the evaluation: end-to-end delay and channel manipulation by the translator design during the run-time.

Fig. 10 shows the delay at each hop in the converged TSN-5G network depicted in Fig. 9. The horizontal axis shows the events that are explained in Table IV. For example, event 1 shows the processing delay in Node (S). Similarly, event 17 shows the delay of the message received in Node (A). The vertical axis in Fig. 10 shows the cumulative delay at various events with respect to the start of the flow (Node (S) sensing the values). The blue line indicates the measured cumulative delay for each event in the simulator, while the highlighted red values represents the cumulative delay for the flow: Node (S) -> Node (Agg) -> Node (RC) -> Node (A). For example, the cumulative delay from sensor Node (S) to aggregator Node (Agg) is 0.27ms. The end-to-end delay from the sensor (Node (S)) to the actuator (Node (A)) in the traffic flow in the use case in Fig. 8 is 5.57ms.

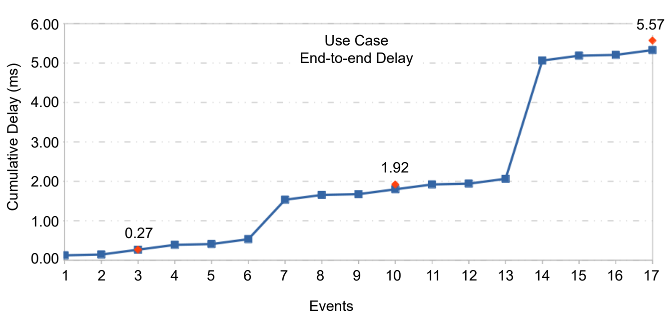


Fig. 10: End-to-end delay in the automotive use case.

It is interesting to not that there are considerable jumps in the delay between the events 6 and 7 and between the events 13 to 14. These jumps show the transmission time over the 5G channel with the slight difference between each due to the different priority level used to manipulate the 5G channel as described below.

TABLE IV: Representation of each event included in the transmission flow from Sensor to Actuator.

Event	Event Interpretation
1	Processing delay in Node (S)
2	Transmission delay between Node (S) and Bridge1
3	Transmission delay between Bridge1 and Node (Agg)
4	Node (Agg) sends the message back to Bridge1
5	Translation delay from TSN to 5G
6	The message is sent to the FiveGNode1
7	Transmission delay from FiveGNode1 to FiveGnode2 (PCP 1)
8	Translation delay from 5G to TSN
9	The message is sent to Bridge2
10	Bridge2 sends the message to Node (RC)
11	Node (RC) sends the message back to Bridge2
12	Translation delay from TSN to 5G
13	The message is sent to the FiveGNode2
14	Transmission delay from FiveGNode2 to FiveGnode1 (PCP 2)
15	Translation delay from 5G to TSN
16	The RC message is sent to Bridge1
17	Bridge1 sends the RC message to Node (A)

The manipulation of the channel is indicated by first writing the PCP value and then the corresponding XML value, shown by an output of XMLInfo as depicted in Fig. 11. To make sure that the values are properly configured, the channel parameters are also output to the console. This flow of output can be seen in Fig. 11 which shows how different PCP values provide different outputs and corresponding channel delay.

```
INFO: PCPValue: 1
INFO: XMLInfo: Delay: 0.001
INFO: XMLInfo: PER: 0.0001
INFO: XMLInfo: Datarate: 100000000
INFO: Channel Delay: 0.001
INFO: Channel PER: 0.0001
INFO: Channel Datarate: 1e+08
```

(a)

```
INFO: PCPValue: 2
INFO: XMLInfo: Delay: 0.003
INFO: XMLInfo: PER: 0.0001
INFO: XMLInfo: Datarate: 100000000
INFO: Channel Delay: 0.003
INFO: Channel PER: 0.0001
INFO: Channel Datarate: 1e+08
```

(b)

Fig. 11: 5G channel manipulation based on different priority levels: a) PCP 1 output for message sent from Node (Agg) to Node (RC), and b) PCP 2 output for message sent from Node (RC) to Node (A).

PCP 1 is used from Node (Agg) to Node (RC) and is indicated by events 6 to 7. Similarly, PCP 2 is used from Node (RC) to Node (A) and is denoted by events 13 to 14. In Fig. 11, the PCP value is first read and then matched to the PCPValue of the XML document as listed by the XMLInfo. This is applied to the channel, indicated by the Channel Delay, PER, and Datarate. For PCP 1 and 2, the delays of 0.001s and 0.003s correspond to the delay-time increase when looking at events 6 to 7, and 13 to 14.

The parameters' values for each PCP correspond to the predefined values in the XML file. This shows that the translator performed well on channel manipulation during run-time. The use case shows a change in channel parameters depending on the read PCP value and is, therefore, the first step towards an improved converged TSN-5G network within the NeSTiNg simulation tool.

VI. Conclusions

In this paper, we proposed a technique to integrate TSN and 5G communication mainly focusing on the translation of the flows between them. Furthermore, we presented a proof-of-concept implementation of the proposed technique in a commonly used free simulation tool, namely NeSTiNg. We utilized an automotive industrial use case to evaluate the performance of the proposed technique and the simulator. We showed that the proposed technique can be useful for the network designers to evaluate TSN-5G heterogeneous network configurations with regards to end-to-end delays.

The current implementation in the simulator assumes that the clock synchronization is perfect between the devices in use. This is not the case in a real-world scenario where TSN and 5G have their clock synchronization schemes. The sharing of time information between the TSN and 5G system is a required step for a fully synchronized system. Therefore, an important step forward is to design and implement such a synchronization. Moreover, other features of 5G according to URLLC can be integrated, which entails another future work.

Acknowledgment

The work in this paper is supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA) via the PROVIDENT project and by the Swedish Knowledge Foundation via the projects DPAC & HERO. We thank all our industrial partners, especially Arcticus Systems & HIAB.

References

[1] J. Sachs, K. Wallstedt, F. Alriksson, and G. Eneroth, “Boosting smart manufacturing with 5G wireless connectivity,” Ericsson Technology Review, February 2019, [Online]. Available: https://www.ericsson.com/49232f/assets/local/reports-papers/ericsson-technology-review/docs/2019/5g-and-smart-manufacturing.pdf.

[2] S. Mubeen, P. Nikolaidis, A. Didic, H. Pei-Breivold, K. Sandström, and M. Behnam, “Delay Mitigation in Offloaded Cloud Controllers in Industrial IoT,” *IEEE Access*, vol. 5, pp. 4418–4430, 2017.

[3] HMS, “Industrial ethernet is now bigger than fieldbuses,” February 2018, [Online]. Available: https://www.hms-networks.com/news-and-insights/2018/02/27/industrial-ethernet-is-now-bigger-than-fieldbuses.

[4] M. Ashjaei, L. L. Bello, M. Daneshtalab, G. Patti, S. Saponara, and S. Mubeen, “Time-sensitive networking in automotive embedded systems: State of the art and research opportunities,” *Journal of Systems Architecture*, vol. 117, p. 102137, 2021.

[5] “Technical Specification Group Services and System Aspects; System architecture for the 5G system (5GS)(Release 16),” 3GPP TR 23.501, Technical report, 10 2020.

[6] J. Falk, D. Hellmanns, B. Carabelli, N. Nayak, F. Dürr, S. Kehrer, and K. Rothermel, “Nesting: Simulating IEEE Time-Sensitive Networking (TSN) in OMNeT++,” in *2019 International Conference on Networked Systems*, 2019, pp. 1–8.

[7] L. Lo Bello and W. Steiner, “A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems,” *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1094–1120, 2019.

[8] L. L. Bello, R. Mariani, S. Mubeen, and S. Saponara, “Recent advances and trends in on-board embedded and networked automotive systems,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1038–1051, 2019.

[9] B. Houtan, M. Ashjaei, M. Daneshtalab, M. Sjödin, and S. Mubeen, “Synthesising schedules to improve qos of best-effort traffic in tsn networks,” in *29th International Conference on Real-Time Networks and Systems*, April 2021.

[10] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A Survey on Low Latency Towards 5G: RAN, Core Network and Caching Solutions,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.

[11] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, “What will 5G be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[12] S. Z. Shancang Li, Li Da Xu, “5G internet of things: A survey,” *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[13] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, “A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges,” *IEEE Access*, vol. 6, 2018.

[14] M. A. Siddiqi, H. Yu, J. Joung,“5G ultra-reliable low-latency communication implementation challenges and operational issues with IoT devices,” Electronics, vol. 8, no. 9, p. 981, Sep. 2019.

[15] M. Säily, C. B. Estevan, J. J. Gimenez, F. Tesema, W. Guo, D. Gomez-Barquero, and D. Mi, “5G Radio Access Network Architecture for Terrestrial Broadcast Services,” *IEEE Transactions on Broadcasting*, vol. 66, no. 2, pp. 404–415, 2020.

[16] A. Zaidi, A. Bränneby, A. Nazari, M. Hogan, and C. Kuhlins, Cellular IoT in the 5G era: Realizing cellular IoT for industrial transformation, June, 6. 2019. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/white-papers/cellular- iot-in-the-5g-era.

[17] I. Godor, M. Luvisotto, S. Ruffini, K. Wang, D. Patel, J. Sachs, O. Dobrijevic, D. P. Venmani, O. L. Moult, J. Costa-Requena, A. Poutanen, C. Marshall, and J. Farkas, “A Look Inside 5G Standards to Support Time Synchronization for Smart Manufacturing,” *IEEE Communications Standards Magazine*, vol. 4, no. 3, pp. 14–21, 2020.

[18] M. Schüngel, S. Dietrich, D. Ginthör, S.-P. Chen, and M. Kuhn, “Analysis of Time Synchronization for Converged Wired and Wireless Networks,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2020, pp. 198–205.

[19] “Technical Specification Group Services and System Aspects; Study on enhancement of 5G System (5GS) for vertical and Local Area Network (LAN) services (Release 16),” 3GPP TR 23.734, Technical report, 06 2019. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3487

[20] B. Holfeld, D. Wieruch, T. Wirth, L. Thiele, S. A. Ashraf, J. Huschke, I. Aktas, and J. Ansari, “Wireless Communication for Factory Automation: an opportunity for LTE and 5G systems,” *IEEE Communications Magazine*, vol. 54, no. 6, pp. 36–43, 2016.

[21] M. Schüngel, S. Dietrich, D. Ginthör, S.-P. Chen, and M. Kuhn, “Single Message Distribution of Timing Information for Time Synchronization in Converged Wired and Wireless Networks,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2020, pp. 286–293.

[22] OMNEST. [Online]. Available: https://omnest.com/.

[23] D. Ginthör, J. von Hoyningen-Huene, R. Guillaume, and H. Schotten, “Analysis of Multi-user Scheduling in a TSN-enabled 5G System for Industrial Applications,” in *2019 IEEE International Conference on Industrial Internet*, 2019, pp. 190–199.

[24] L. Martenvormfelde, A. Neumann, L. Wisniewski, and J. Jasperneite, “A Simulation Model for Integrating 5G into Time Sensitive Networking as a Transparent Bridge,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2020, pp. 1103–1106.

[25] Cisco, "Inter-Switch Link and IEEE 802.1Q Frame Format", [Online]. Available: https://www.cisco.com/c/en/us/support/docs/lan-switching/8021q/17056-741-4.html".

[26] A. Larrañaga, M. C. Lucas-Estañ, I. Martinez, I. Val, and J. Gozalvez, “Analysis of 5G-TSN integration to support Industry 4.0,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation*, vol. 1, 2020, pp. 1111–1114.

[27] M. D. Johas Teener and G. M. Garner, “Overview and timing performance of IEEE 802.1AS,” in *2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, 2008, pp. 49–53.

[28] Alexander Magnusson and David Pantzar, "Integrating 5G Components into a TSN Discrete Event Simulation Framework", Master Thesis, School of Innovation, Design and Engineering, Mälardalen University, Sweden, 2021.

[29] “Technical Specification Group Services and System Aspects; Study on Scenarios and Requirements for Next Generation Access Technologies (Release 14),” 3GPP Rep. 38.913, Technical report, 2017.