Comparative Analysis of Open Source 5G Core Implementations and Design and Evaluation of a 5G Testbed for Industrial Communication Master's Thesis shortened Draft

vorgelegt von
Mohamed Mesto, MCs TU Berlin[1]

Technical University Berlin, Berlin, Germany,
Department of Telecommunication Systems Next Generation Networks Architekturen der Vermittlungsknoten (AV) Betreuer:
Prof. Dr. Th. M. ,
Gutachter: Prof. Dr.-Ing. A. W.
Dr.-Ing. R. K.

`M.mesto@campus.tu-berlin.de`
`Matrikelnummer:3900**`

Abstract. The integration of the Fifth-generation (5G) networks with Time-Sensitive Networking (TSN) technology is believably one of the most considerable argument topics in industrial automation nowadays. Companies that operate an intelligent manufacturing system aim to develop an extensive, ubiquitous, and permanent connectivity of the technologies and features of networking to achieve real-time requirements. That would create advantages such as building up smart factories completely connected, enhanced, and compliant with all the necessary industrial technology requirements. Moreover, the fifth-generation (5G) networks attempt to play a significant role in connecting massive wireless devices, which are very useful in factories. Furthermore, it has a tremendous positive impact on improving smart factories' capabilities, performance, scalability, and compliance. All essential elements of our research are discussed in the 3GPP release 15 specifications when the 5G service-based architecture (SBA) was reviewed. The fundamental technological components in the 5G core network are the separation of control plane and user plane, service-based interface (SBI), modularization, and network function virtualization. SBA Architecture provides a single API calling interface. All the network functions (NF) are interconnected via an interface for calling the other NF. After Authorization, the network functions (NFs) able to access the other NF using SBI. Network function virtualization (NFV) empowers the network function to be virtualized and deployed on any cloud environment. The support of virtualization in the 5G, the limits between traditional Evolved Packet Core (EPC) network components (MME, SGW, and PGW) will come to exit [30].
The paper targets on forming and handling a hybrid network comprising Industrial Ethernet/Time Sensitive Networking (TSN) and 5G, as the

key communication system representatives of operational technology (OT) and information and communication technology (ICT) industry. It is organized as follows: the subsections below introduce 5G mobile networks, Industrial Ethernet and TSN, section II describes scenarios to combine both communication technologies, section III describes approach to model and configure hybrid networks and their transitions, and section IV concludes the paper.
The thesis scope is to investigate the integration of 5GC with TSN And implement a Simulation of a 5G Core Network Deployment and Testing with an open-source 5G SA gNB emulator (Rel. 16) gNBSIM. Besides, It studies the opportunity of transmitting data packets from end-to-end (E2E). Wireshark and Iperf apps will be employed to capture Data Packets. They help achieve a comparative analysis and get the measurement of time-sync, Reliability, Latency, and Determinism.

Keywords: 3GPP, Architecture, Open 5G core System, NextGen, Time-Sensitive Networking (TSN), Free5GC, NSA, SA, 5GC, SBA, Smart Factories

Table of Contents

List of Figures

List of Tables

List of Listings

1 Introduction

Simultaneously to the increasing demands of full connectivity between people and machines, Industrial automation is changing towards matching these enormous growth requirements regarding reliability and latency. It has been progressed through several stages from particular Fieldbus systems over Industrial Ethernet to industrial wireless communication [26]. Likewise, the need for continuous connectivity and improved network technologies to meet real-time systems has become critical [12]. The continuous digitalization of the production systems pointed to Industry 4.0. Nevertheless, Industry 4.0 is inspiring the manufacturing area with the provision of decentralized management, control, monitoring and support for real-time systems, giving industrial automation flexibility and making it typical [14][4]. In Consequence, The internet of things (IoT) and machine-to-machine communication (M2M) have been employed to strengthen smart factories in terms of communications and self-monitoring [18]. In contrast to the predecessors, 5G is exceptionally beneficial for intelligent manufacturing. It points to support a wide range of vertical industries by appending ultra-reliable low-latency communication(uRLLC), flexibility, and decrease the dependence on cables [12]. On the other side, to answer real-time requirements, extra functionalities such as summation frame communication, Time Division Multiple Access (TDMA), or polling-based communication have been included. These additions present rigorous real-time capabilities, but require changes of the original IEEE Ethernet MAC [33]. QoS, Timing, synchronization, and resource reservation are some of the promised features by TSN [7]. Nevertheless, the integration of 5G with Time-Sensitive Networking (TSN) has significant advantages: supporting the entire connection in industrial automation, and provide all needed requirements on intelligent factories. To grant satisfactory QoS for end-to-end communication (E2E) in integrated networks [34]. Furthermore, the Integration of 5GC with TSN will be described in three scenarios regarding the distribution of Industrial parts (sensors, actuators, and controllers). Besides, a couple of convenient use cases of integrating 5G mobile networks in the industries area of the future will be covered [4]. Although, ITU classified 5G applications into three types: Enhanced mobile broadband (eMBB), massive machine-type communication (mMTC), and uRLLC shown in figure 1. At the moment, according to the 3GPP R15 standard, 5G network focuses on eMBB. In order to deploy the 5G architecture, 3GPP defines Non-Standalone (NSA), and Standalone (SA) architectures. You can see the NSA and SA architecture in figure 2. According to 3GPP release 15

[30], 3GPP will release the NSA architecture first. NSA will use the existing 4G infrastructure to deploy the 5G network to support eMBB services. SA is the aim architecture from 3GPP, which will not depend on the 4G infrastructure, will use the 5G radio network, and will afford the eMBB, mMTC, and uRLLC services [2].

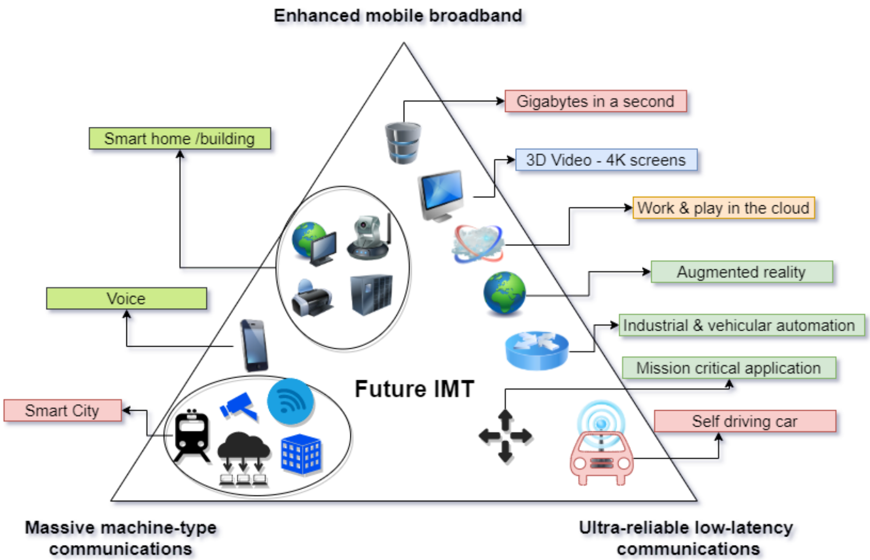

Fig. 1: IMT 2020 Use cases [23]

In order to reach the topic's goal, we need to dive into the details of 5G Network Architecture; That is, where we have Service-based architecture (SBA) outlines the network function (e.g., AMF), and Reference point architecture proves how the Network function services are interacting with the Network function (NF)[30]. This paper summarily describes the favorable deployment scenario for an integrated 5G and TSN system in Industry 4.0 field. Additionally, the Integration concepts will clarify the transparent and non-transparent approaches (represented in Tunneling, gateways, and proxies) [15] [21].
Last but not least, the following chapters will explain the details, including establishing a Protocol Data Unit (PDU) session between ingress TSN Translator (TT) and egress TT over Ethernet, and the required functions that need to be supported by the Open Source 5GC.

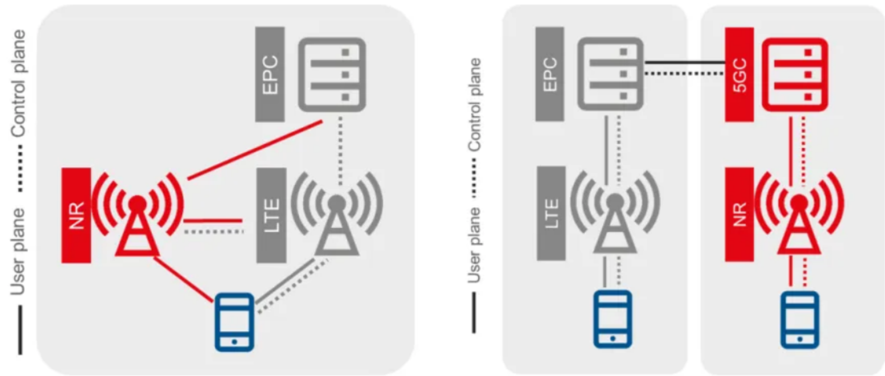

Fig. 2: SA and NSA Architecture[2].

2 Problem Statement

2.1 Reliability and Low Latency

Enhanced mobile broadband (eMBB) term is already in handling in 4G Networks. However, For 5G, two fundamental classifications have been determined: massive machine-type communication (mMTC) to support an unlimited amount of associated nodes in industrial and consumer fields, and ultra-reliable low-latency communication (uRLLC) for critical communication and correlated control systems [16]. Figure 4 depicts the uRLLC features[12].

The abilities uRLLC empowers 5G to achieve the essential requests of time-sensitive communication, reliability, latency and wireless deterministic. According to the ITU-R[27], Studies and research in 5G networks domain showed that End-to-End reliability is near to 99,999% and latency of 1 ms of Data packets. However, the data rates higher to many Gb/s, processing entrance for up to a million devices per square kilometer. 5G Networks supports ultra-reliability for control and data channels by providing different techniques, such as encouraging multiple carriers and packet duplication over independent radio links, multi-antenna transmission.

Time-sensitive networking is a collection of Ethernet standards currently developed by the IEEE 802.1 working group[32]. TSN is an enabler of Industry 4.0 by providing flexible data access and full connectivity for a smart factory. The following reasons support this perspective, but we address them by mentioning and not limiting them. TSN empowers Industrial Automation by performing deterministic communication over Ethernet for real-time applications. By accomplishing scheduled traffic and synchronization, i.e., TSN grants guaranteed latency limitations [16]. On the other side, in a hybrid network, TSN supports many applications with various QoS requirements, e.g., closed-loop control, to grant the best-effort traffic over an individual standard Ethernet infrastructure[12]. Figure 3 draws Time-Sensitive Networking (TSN) features (Selection and Use of TSN tools).

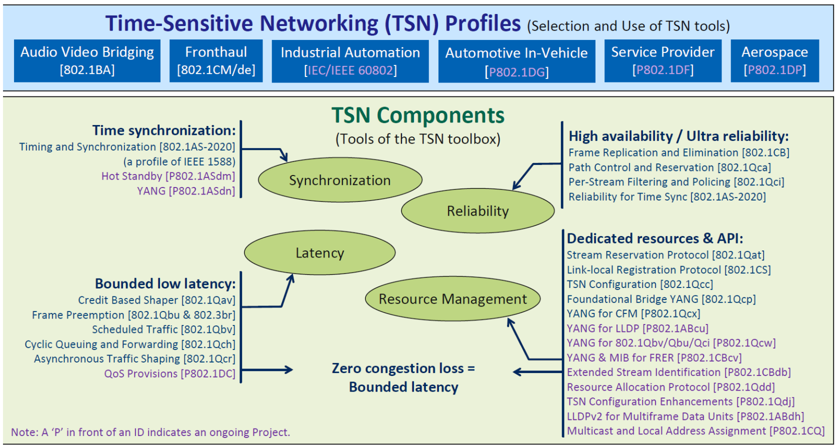

Fig. 3: Time-Sensitive Networking (TSN) Profiles [32].

2.2 Real-Time (Wireless) Communications

Wired networks have always played a primary role in industrial automation. However, the advantages and high efficiency of wireless communication demonstrated by performance indicators in smart manufacturing made it very attractive. Not only in the case of reducing energy consumption, expanding the range and increasing the bandwidth, but also in limiting packet collision. All this gives wireless networks flexibility and mobility. Nevertheless, the most significant factor for real-time systems is preventing collisions and thus ending data packets lost [24]. On the other hand, some applications like closed-loop control have a demand of periodic uRLLC, or uRLLC with a high refresh rate. This confirms that Synchronous and Hybrid Architecture for Real-Time (RT) Performance (SHARP) is the ideal solution. Which includes both required features uRLLC and Best-Effort traffic [28]. Therefore, the trend towards hybrid communication network systems was observed in the past and will continue into the future. To address Real-Time (Wireless) Communications problems in industrial networks, previous mobile network generations and 5G cellular radio systems include Time synchronization as a fundamental element of their operation. Besides, the wireless network parts themselves are also time-synchronized, e.g., during the precision time protocol telecom profile [13]. All those features make it a strong candidate for time-critical systems. Figure4 illustrates uRLLC features. There is a close relation between reliability and time synchronization, i.e., the draw reviews that 5G uses time synchronization for its operations, the multiple antennas, and radio channels that afford reliability[12]. 5G also offers promising ideas and methods for managing low latency and resources, which can be combined to provide superior reliability and low latency. What arouses curiosity; the 5G system (5GS) also contributes solutions in the core network (CN) for Ethernet networking and URLLC. The 5G CN offers native Ethernet protocol data unit (PDU) sessions. 5G assists in establishing unnecessary user plane paths through the 5GS, including RAN, the CN, and the carrier network. 5GS also allows for a redundant user plane separately between CN and RAN nodes and between the UE and the RAN nodes [20].

3 Existing Work

To place the paper's contribution in context and identify the gap the work is intended to fill, a short literature survey will provide. We will touch on a brief overview of industrial automation's rival approaches, from Fieldbus to TSN, TSN over 5G in 3GPP Rel.15 and beyond. There were complete solutions and methods to connect field controllers, sensors, and actuators, which Fieldbus technology has granted[31]. Those make it flexible regarding the architecture and give it the ability to support particular Qos for different applications. Nevertheless, Fieldbus Foundation, a not-for-profit organization, designed standards that increase operability, safety, therefore lowering cost when employing Fieldbus technology. Hence Fieldbus Foundation could guarantee the tested devices to ensure that they meet Fieldbus Foundation specifications and identified with √FF for interoperability assurance. Consequently, Fieldbus technology was the ideal solution, and it took over at the A level in industrial automation from 1970 until November 2012, when the TSN Task Force was formed[29]. Time-sensitive networking (TSN) is a collection of standards under development by the IEEE 802.1 [32].The race was frantic among engineers and researchers working in industrial automation to make wired networks meet the need for time-critical systems. The reason for this was the inefficiency of the original IEEE 802.3 Ethernet standard. Besides, protocols based on Ethernet, like TCP, UDP, and IP, typically do not recognize real-time requirements. The race was frantic among engineers and researchers working in industrial automation to make wired networks meet the need for time-critical systems. The reason for this was the inefficiency of the original IEEE 802.3 Ethernet standard. Besides, protocols based on Ethernet, like TCP, UDP, and IP, typically do not recognize real-time requirements. Therefore, the Open Systems Interconnection model (OSI model) modification rises to the highest level to match the conditions of the real-time application, which gives a strong impetus to improve the Industrial Ethernet schemes abilities. i.e., To overcome obstacles to standard Ethernet and protocol compatibility TCP/IP or UDP/IP with real-time systems requirements[9]. Nevertheless, Ethernet's progress constant over the decades has seen its way into protocols such as Profinet, PowerLink, EtherCAT, and EtherNet/IP [31].

As a review from the above, TSN standards are essentially for IEEE Std *intro* 802.3 Ethernet, which indicates they employ all the advantages of standard Ethernet, such as universality, flexibility, and economic operation price. Various worthy tools are included in those standards like reliability, time synchronization, traffic shaping, resource management, as shown in figure 5 TSN characteristics are developed upon the base IEEE 802.1 bridging standards, making them vital to be carried in industrial automation.

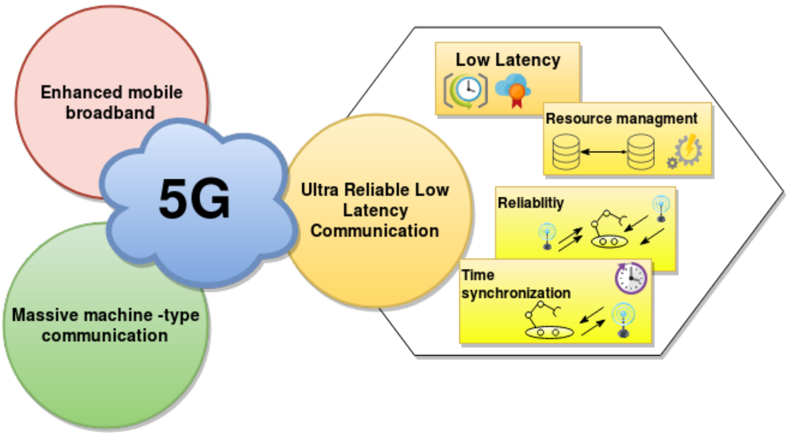

Fig. 4: 5G URLLC overview of TSN components [12]

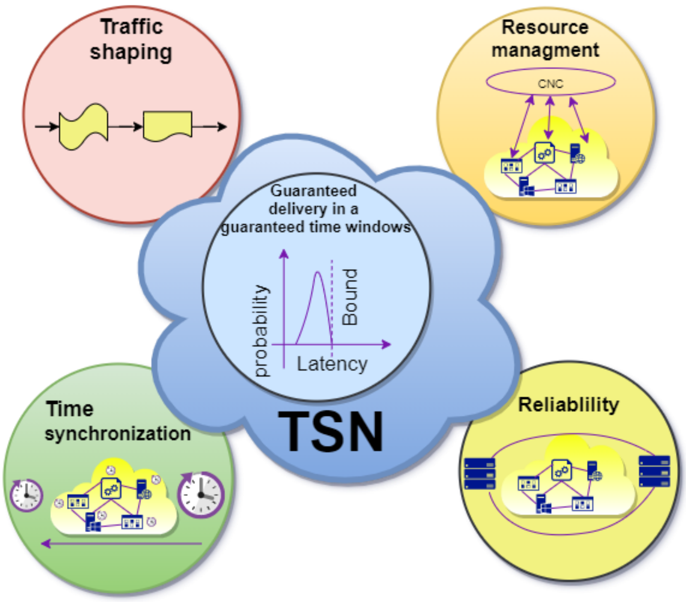

Fig. 5: Valuable tools within the TSN toolbox that enable deployments in industrial automation [35].

3GPP also provided the conditions for high data rates and traffic densities in 3GPP TS 22.261[1]; in this 50Mbps is the essential requirement of eMBB service for down link (DL). Up Link (UL) and DL requirement for high data rates and traffic densities can be seen in the **table 1**.

Scenario	**Experience data rate (DL)**	**Experience data rate (UL)**	**Area Traffic Capacity (DL)**	**Area Traffic Capacity (UL)**	**Overall user density**
Urban	50 Mbps	25 Mbps	100 Gbps/km2	50 $Gbps/km^2$	$10000/km^2$
Rural	50 Mbps	25 Mbps	1 $Gbps/km^2$	500 $Gbps/km^2$	$100/km^2$
Indoor hotspot	1 Gbps	500 Mbps	15 $Tbps/km^2$	2 $Tbps/km^2$	$250000/km^2$
Dense urban	300 Mbps	50 Mbps	750 $Gbps/km^2$	125 $Gbps/km^2$	$25000/km^2$
High- speed vehicle	50 Mbps	25 Mbps	100 $Gbps/km^2$	50 $Gbps/km^2$	$4000/km^2$

Table 1: Performance requirements for high data rates and traffic densities[1].

3.1 5G Network Architecture

3GPP determined the specification of the 5G architecture. 3GPP defined the 5G architecture in two methods, first as a service-based and the second reference point architecture, which shows the intercommunication between network functions [30].

- Service-based architecture (SBA) describes the network function (e.g., AMF) of the control plane and also defines how it can authorize other network functions to access the services of other network functions. You can see the SBA in figure 7
- **Reference point architecture explains how the NF services are interacting with the Network function. All of these interactions represented by a point-to-point reference (e.g., N5) between two network functions (e.g., UDM and NRF). As it shown in figure 8**

All essential elements of our research are discussed in the 3GPP release 15 specifications when the 5G service-based architecture (SBA) was reviewed. The fundamental technological components in the 5G core network are the separation of control plane and user plane, service-based interface (SBI), modularization, and network function virtualization. SBA Architecture provides a single API calling interface. All the network functions (NF) are interconnected via an interface for calling the other NF. After Authorization, the network functions (NFs) able to

Use case category	User Experienced Data Rate	E2E Latency	Mobility
Broadband access in dense areas	DL: 300 Mbps UL: 50 Mbps	10 ms	On demand, 0-100 km/h
Indoor ultra-high broadband access	DL: 1 Gbps, UL: 500 Mbps	10 ms	Pedestrian
Broadband access in a crowd	DL: 25 Mbps UL: 50 Mbps	10 ms	Pedestrian
50+ Mbps everywhere	DL: 50 Mbps UL: 25 Mbps	10 ms	0-120 km/h
Ultra-low cost broadband access for low ARPU areas	DL: 10 Mbps UL: 10 Mbps	50 ms	on demand: 0-50 km/h
Mobile broadband in vehicles (cars, trains)	DL: 50 Mbps UL: 25 Mbps	10 ms	On demand, up to 500 km/h
Airplanes connectivity	DL: 15 Mbps per user UL: 7.5 Mbps per user	10 ms	Up to 1000 km/h
Massive low-cost/long-range/low-power MTC	Low (typically 1-100 kbps)	Seconds to hours	on demand: 0-500 km/h
Broadband MTC	See the requirements for the Broadband access in dense areas and 50+Mbps everywhere categories		
Ultra-low latency	DL: 50 Mbps UL: 25 Mbps	<1 ms	Pedestrian
Resilience and traffic surge	DL: 0.1-1 Mbps UL: 0.1-1 Mbps	Regular communication: not critical	0-120 km/h
Ultra-high reliability & Ultra-low latency	DL: From 50 kbps to 10 Mbps; UL: From a few bps to 10 Mbps	1 ms	on demand: 0-500 km/h
Ultra-high availability & reliability	DL: 10 Mbps UL: 10 Mbps	10 ms	On demand, 0-500 km/h
Broadcast like services	DL: Up to 200 Mbps UL: Modest (e.g. 500 kbps)	<100 ms	on demand: 0-500 km/h

Fig. 6: User Experience Requirements[5].

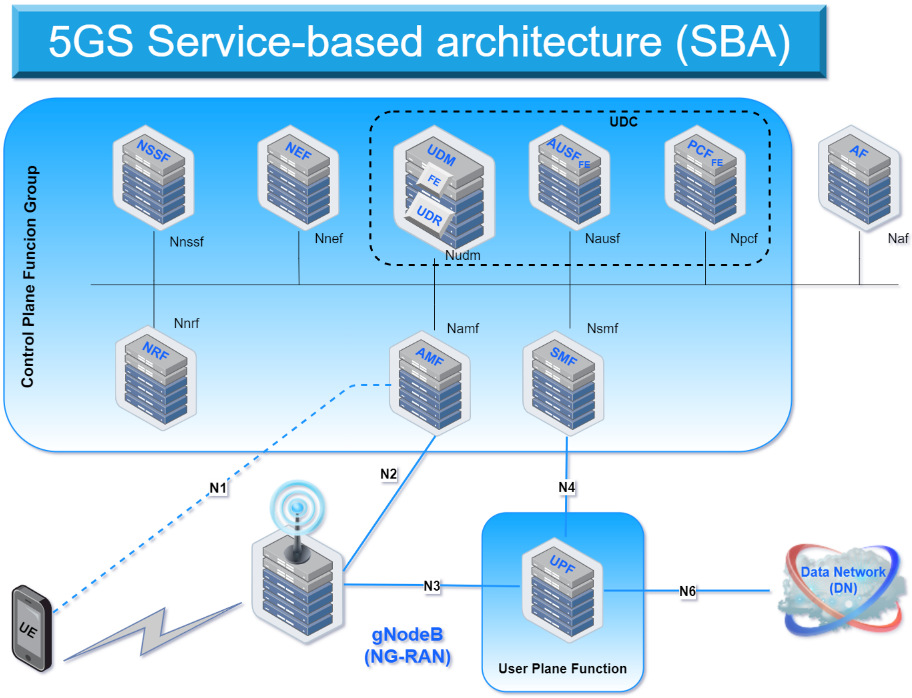

Fig. 7: 5GS Service-based architecture (SBA)[11].

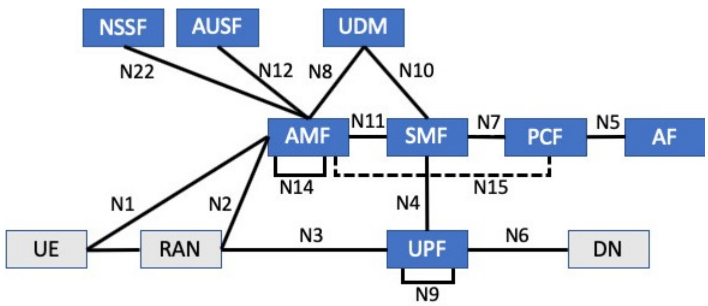

Fig. 8: 5G Reference Point Architecture

access the other NF using SBI. Network function virtualization (NFV) empowers the network function to be virtualized and deployed on any cloud environment. The support of virtualization in the 5G, the limits between traditional Evolved Packet Core (EPC) network components (MME, SGW, and PGW) will come to exit [30].

In 5G Service-Based architecture, Many advantages noticed; independent logical network, sharing the infrastructure either wholly or partly, and being deployed on different infrastructure. I.e., 5GC Supports network slicing. Furthermore, SBA interfaces are beneficial; 5GC promotes working with 3rd parties and can customize network slice capabilities via AF[30]. Another important feature; 5G architecture empowers us to take full advantage of the most advanced virtualization and software technologies. The SBA architecture provides a foreseen performance and flexibility for the network. The following are the 5GC network functions[11]:

- AMF: Access and Mobility Management Function - PCF: Policy Control Function.
- SMF: Session Management Function - UPF: User Plane Function.
- NRF: Network Repository Function - AF: Application Function
- NEF: Network Exposure Function - UDM: User Data Management.
- AUSF: Authentication Server Function - NSSF: Network Slice Selection Function.
- SBI: Service Based Interfaces (Namf, Nsmf, Nudm, Nnrf, Nnssf, Nausf, Nnef, Nsmsf, Nudr, Npcf)

Table 2 indicates 5G Interfaces and Functional Description of them.
Another significant technical concept is Network Slicing, clearly “5G slice”, which strengthens the communication service of a particular connection model with a particular method of handling the User-Plane and Control-Plane for this service. To this purpose, a 5G slice is composed of 5G network functions (NF) and precise RAT settings combined together for the exact use case or business model. figure **9** represents an instance of multiple 5G slices concurrently served on the same infrastructure[5].

3.2 5GC-TSN Integration Scenarios

The Integration of the 5G core network and TSN has many possibilities. This Thesis will handle three scenarios of them. Every scenario has its unique advantages to award them to particular user applications[3].

- Scenario 1: Within the first scenario, we have the production sectors for the production area. The connection between those isles is using Industrial Ethernet technology [3]. Consequently, all communication endpoints give network interfaces according to a similar model. In this scenario, 5GS supports the data interchange amongst the isles, which becomes a demand in Industry 4.0 and digitization [28]. In this instance, figure **10** shows that 5GC networks are not affected by this scenario, while the focus is restricted to the Radio Access Network (RAN) and the backhaul of the 5GS figure **13(A)**.

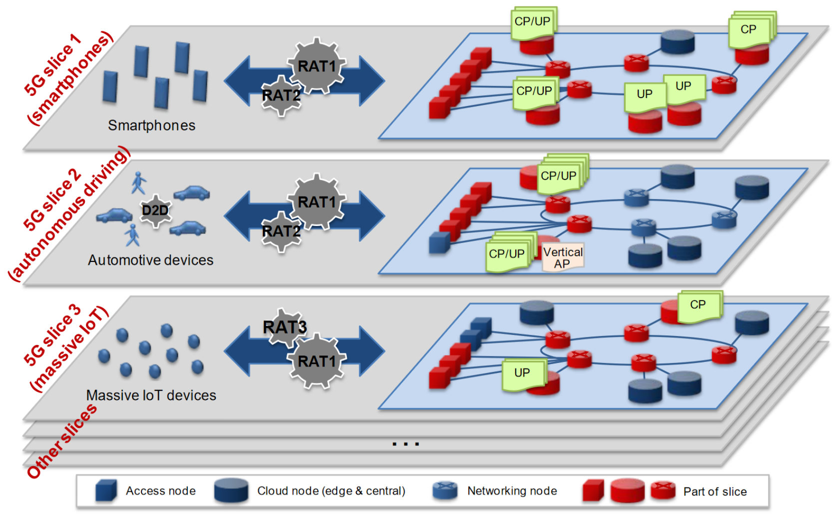

Fig. 9: Network Slicing for different use-cases [5].

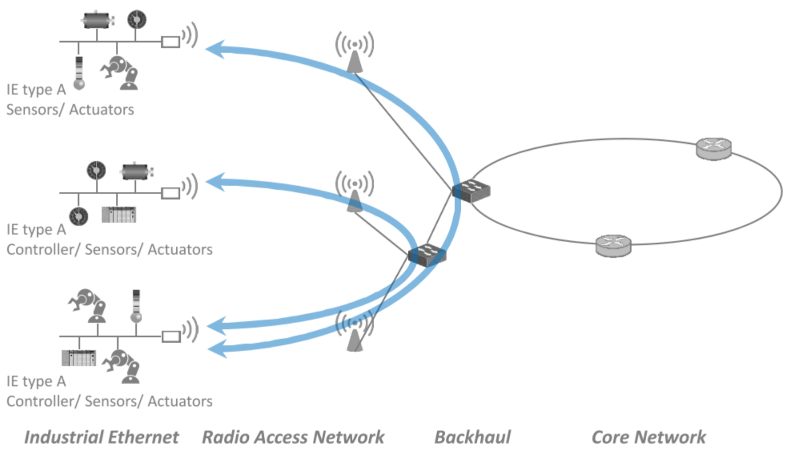

Fig. 10: Scenario 1: Connected homogeneous isles [21].

5G interfaces	Functional Description
N1	Between UE and AMF (Access and Mobility Management Function)
N2	Between RAN (Radio Access Network) or gNB (i.e. 5G base station) and AMF
N3	Between RAN or gNB (i.e. 5G base station) and UPF (User Plane Function)
N4	Between SMF (Session Management Function) and UPF
N5	Between PCF (Policy Control Function) and AF (Application Function).
N6	Between UPF and DN (Data Network)
N7	NG7 is reference point between SMF and PCF
N8	Between Unified Data Management (UDM) and AMF
N9	Between two core UPFs
N10	Reference point between UDM and SMF
N11	Between SMF and SMF
N12	Between AMF and AUSF (Authentication Server Function)
N13	Between UDM and AUSF
N14	Between two AMFs
N15	Between PCF and AMF (in Nonroaming scenario)

Table 2: Shows the 5G Interfaces and Functional Description.

- Scenario 2:
  This scenario is an ideal decision for utilizing cloud-based control [17]. The motivation of that is working mechanism and structure. I.e., as a main noticed difference between the first and second scenario, the controller's physical object is moved from an allocated machine at the production line to a virtualized entity in the 5G network[3]. The complete 5GS is covered in this scenario; since the virtualized controller is connected to the core network. In figure **11**, we figure out that, Qos is required at a high level because the entire 5G layout is considered in the control loop. The virtualized controller performs a standard Ethernet IP-based connection[3]. Conversely, two access methods: Industrial Ethernet or native 5G radio, probably are donated by the actuators or sensors at the production line figure **13(B)**.
- Scenario 3: A surprise that snatched the spotlight between the three scenarios is scenario 3 shown in figure **13(C)**. It is similar to the second scenario in terms of a virtualized controller but is distinct from its design, including a remote production entity connected to 5GC Network [3]. Hence, its significant advantage demonstrates by supporting various types of communications technologies: 5G radio, standard Ethernet IP-based protocols, and different Industrial Ethernet protocols. Scenario 3 diagram illustrated in figure **12**.

The mentioned scenarios have the feasibility to apply through many use cases of 5GC-TSN integration, which takes its ways in Industrial automation [4]. Domains that could benefit from the first scenario include; mobile robots, closed-loop

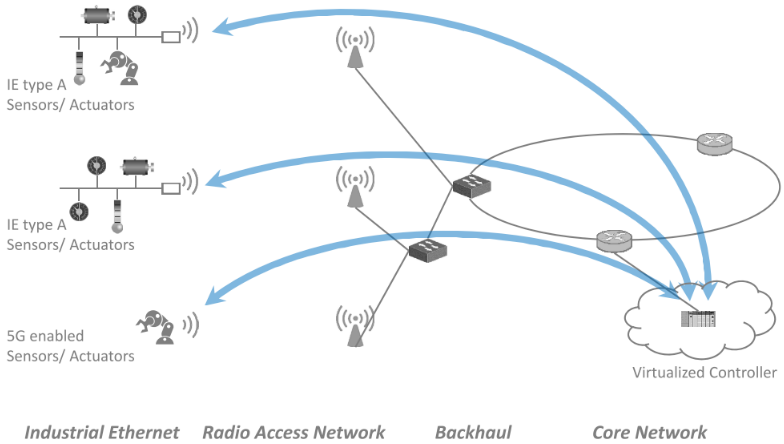

Fig. 11: Scenario 2: Virtualized controller [21].

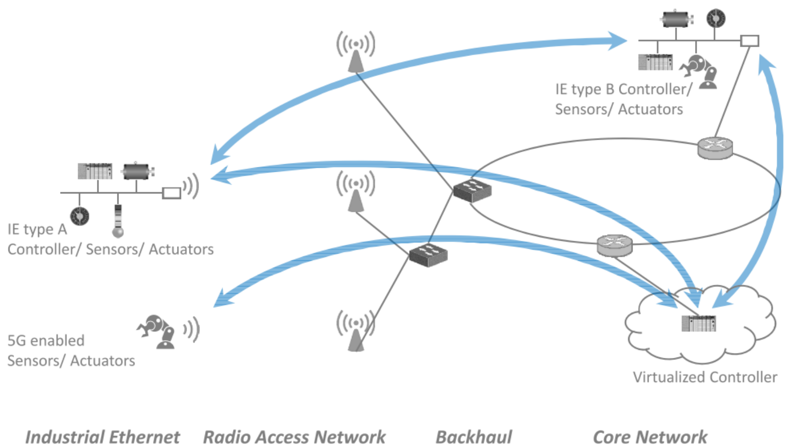

Fig. 12: Scenario 3: Versatility with virtualization and remote site [21].

control in process automation, connectivity for the factory floor, mobile control panels, or modular assembly areas. On the other hand, Scenario 2 concerns control-to-control communication, process monitoring, extensive sensor networks, plant asset management. Last but not least, Scenario 3 is fit for inbound logistics or remote access and maintenance [21].

In addition, Mobility indicates the system's ability to support continuous service activity for movable users. Furthermore, the identified 5G use cases show that 5G networks will support an increasingly large spectrum of static users/devices to mobile users. Regardless of the massive support for 5G networks of a large number of mobile and fixed nodes, 5GS should support mobility-on-demand only. From low mobility or stationary devices like smart meters to very high mobility, like high-speed trains/airplanes [5]. In all cases, the conditional speed between the network edge and the user refers to the mobility needs. In other words, The harmony of the user's activity must be guaranteed. Use case specific mobility requirements are shown in figure **6**.

4 Own approach

This part will identify TSN requirements over 5G testbed and the automatic setup of selected open-source 5G testbed.

4.1 The integration of 5G with TSN configuration approach

The Thesis will deal with the use cases of integration 5G-TSN in the Manufacturing Domains, where 5G technology does not has any action on E2E protocols. I.e., This refers to the transparent methods of 5G-TSN integration[21]. On the contrary, the study will not discuss the non-transparent methods of 5G-TSN integration presented by Tunneling, gateways, and proxies. By reason of not being able to meet requirements of the combining of 5G and TSN.

Other related concepts are P802.1Qcc, and Stream Reservation Protocol (SRP) Enhancements and Performance Improvements [32]. 802.1Qcc enables the coexistence between centralized configuration management and decentralized configuration, configurable SR (stream reservation) classes and streams, benefits heterogeneous configurations for legacy Audio Video Bridging (AVB) equipment [19], support for Layer 3 streaming, fully centralized configuration, deterministic stream reservation convergence, and fully distributed configuration of the Stream Reservation Protocol (SRP)[32]. TSN Configuration fully distributed model showed in figure **14**.

- Fully centralized configuration of IEEE 802.1Qcc: Regarding IEEE 802.1 sense, the system is classified into two characters of equipment: bridges and end stations (Talker or Listener) [32]. The talker end station is the source or a machine that generates data, e.g., (sensors). The listener end station is the destination or a machine that receives and uses data, e.g., controller or monitoring device [15]. Stream: a unidirectional movement of data from

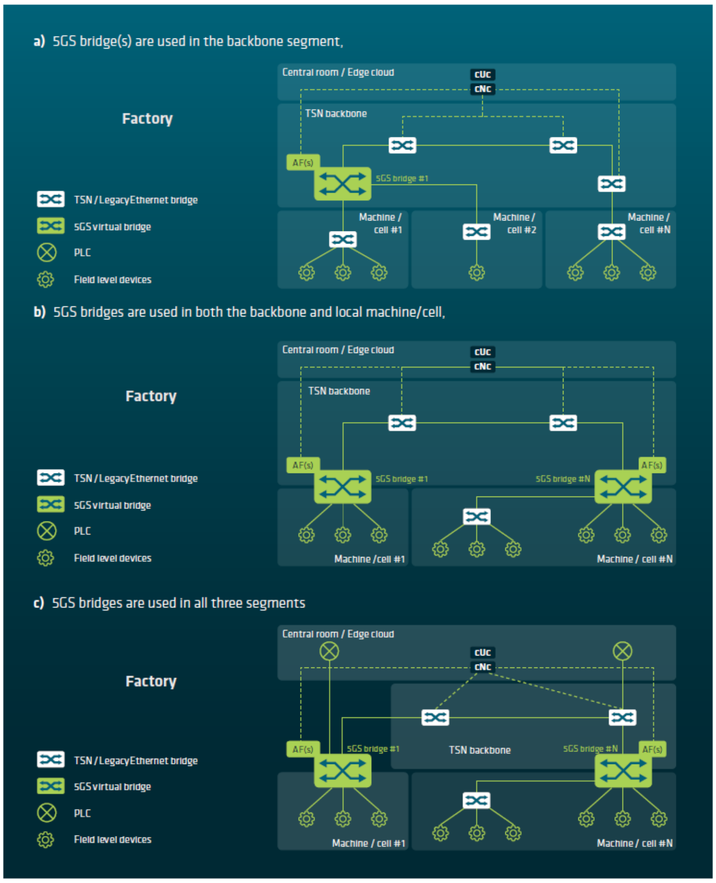

Fig. 13: The role of 5GS bridges in industrial automation[3].

a Talker to one or more further Listeners. As depicted in figure **15**, the technology consists of the following elements: Centralized User Configuration (CUC), Centralized Network Controller (CNC), and Bridges. The Bridge is a design , see figure **17**, that includes Media Access Control (MAC) Bridge or Virtual Local Area Network (VLAN) Bridge element functionality [20]. Time-synchronization is at a high level in this topology. I.e., The topology has a collection of time-aware bridges. On the other hand, end stations and these bridges are synchronized to a master clock in the system. Besides, the complete network is sensible of the global time—however, The interaction between the previously mentioned elements detailed in figure **16**. In addition, (CNC) deals with network devices (bridges), while (CUC) deals with user devices (end stations). The CNC and CUC present the control plane (CP) rather than distributed protocols; i.e., the fully centralized configuration model attends a software-defined networking (SDN) approach. In opposition, distributed control protocols are utilized in the fully distributed model, where there is no CNC or CUC [12].

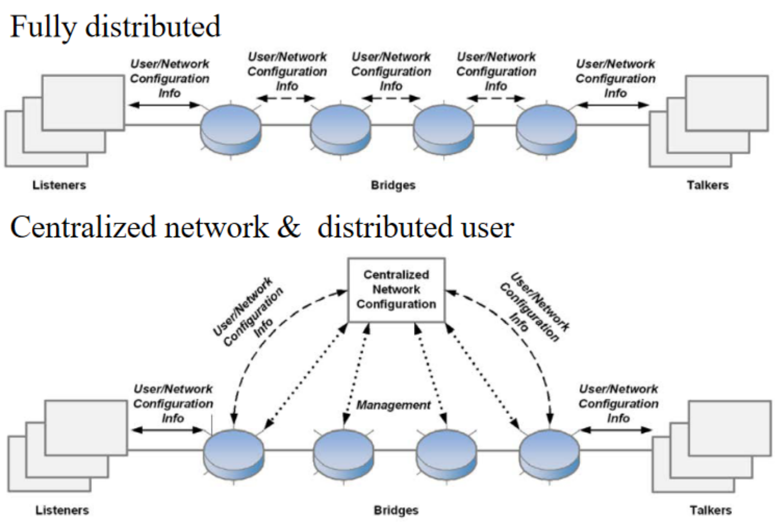

Fig. 14: TSN Configuration fully distributed [802.1Qcc] [32].

– 3GPP 5GS mobile network within TSN Transparent integration: In order to focus on the thesis goal, we perform a zoom in on a TSN bridge of the 802.1Qcc Fully Centralized Configuration Model, that illustrates in figure **18** considering TSN features that shown in figure **5**. Regarding 3GPP Release 16,

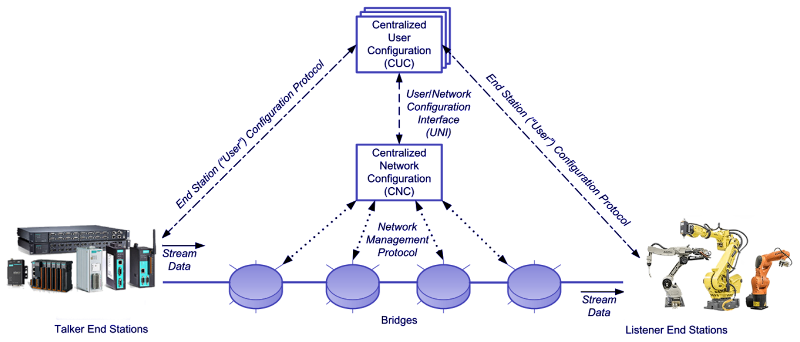

Fig. 15: 802.1Qcc Fully Centralized Configuration Model [16]

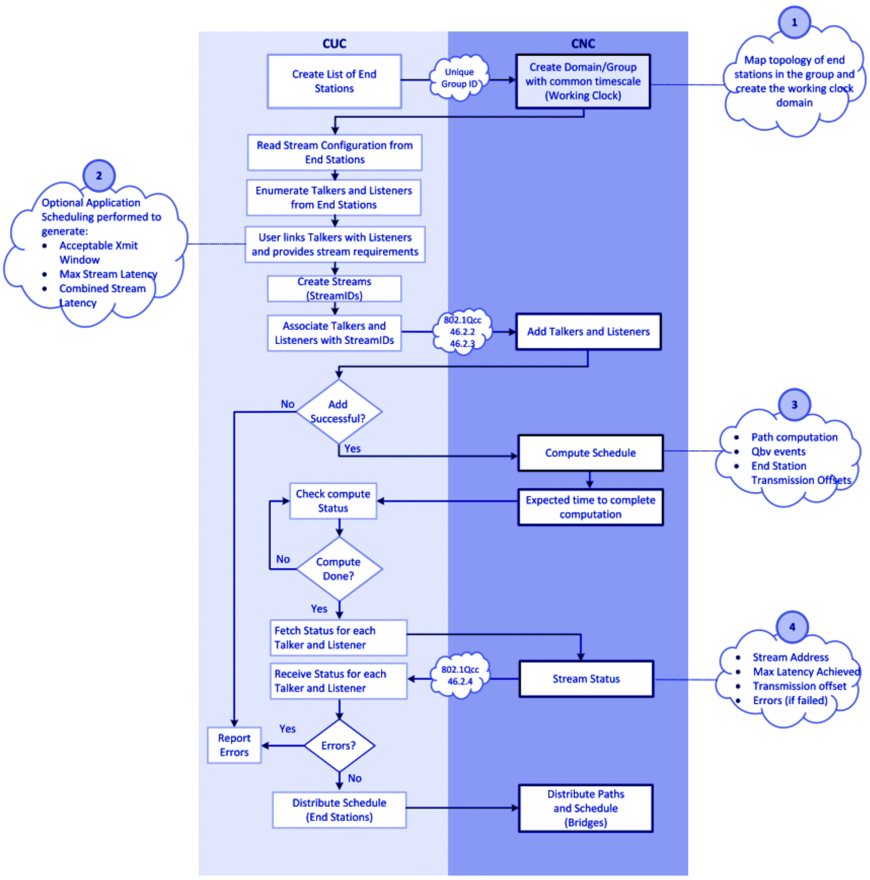

Fig. 16: CUC-CNC Interactions for Industrial TSN Domain Configuration [15].

two operative items are provided; TSN Translator (TT) and an adaptation interface (AIF). Both items represent the concept of the integration of 5G into TSN in industrial automation. I.e., TT and AIF encapsulate the 5G network as a virtual bridge in the TSN network. These two objects modify the TSN performances into identical movements in the 5G and vice versa. Besides, within the TT functionality, the 3GPP 5GS empowers TSN bridge ingress and egress port additions. To characterize these two advantages, the TTs provide hold, and forward functionality for de-jittering [12]. Link Layer Discovery Protocol (LLDP) and Precision Time Protocol (PTP) also play an essential operational purpose in promoting the integration [21]. We also note in the diagram showing the case of two TSN streams associated with two PDU sessions and two EU devices. Furthermore, other angles of the coexisting concept between 5G Network and TSN clarify by supporting 5GS the coherence process of bridges and joining an end station to a bridged network[12]. The deployment contains only an actual UE among two PDU sessions utilizing duplicated connectivity in Radio Access Network (RAN). Consequently, the entire 5G network will look like a TSN bridge.

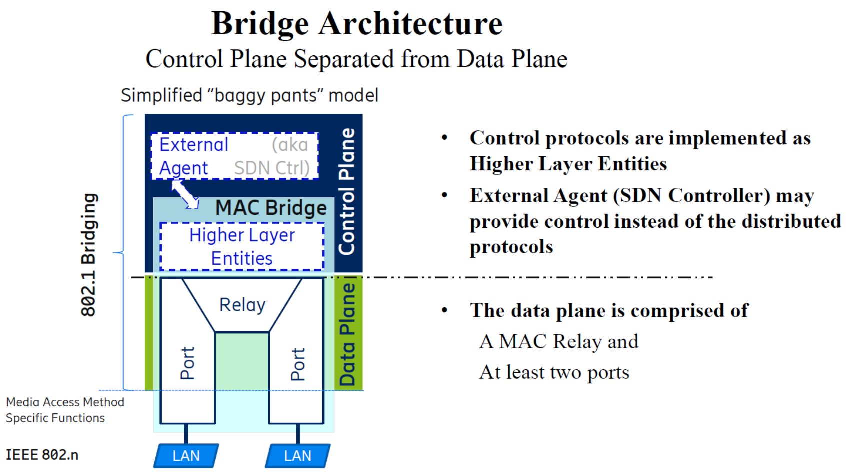

Fig. 17: Bridge Architecture

We conclude from the above that The 5GC supports a Protocol Data Unit (PDU) Connectivity Service, i.e., a service that provides the exchange of PDUs between a UE and a data network identified by a Data Network Name (DNN) [25]. The PDU Connectivity Service is supported via PDU Sessions that are established upon request from the User Equipment (UE), between ingress TSN Translator (TT) and egress TT over Ethernet(not necessarily RAN between 5G base station GNodeB (gNB) and User Equipment (UE)) as illustrated by figure 18. Typically, These PDUs can be IP, Ethernet, and Unstructured, besides DNN (Data Network

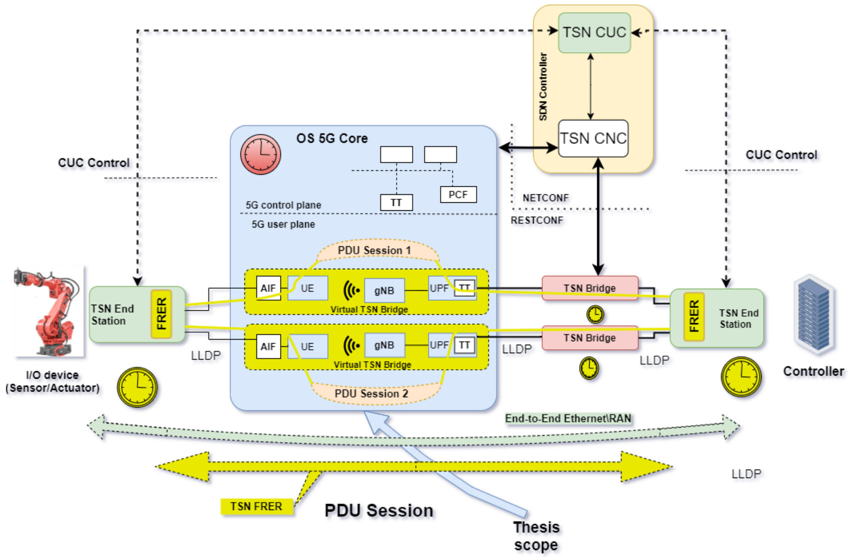

Fig. 18: Open source 5G Core Testbed[12]

Name) is employed to distinguish various destination networks outside these 5G networks. Figure 19 represents a simplified PDU Session Establishment call flow, highlighting the key Network Functions included as well as the steps taken in the process [25].

However, all universities and research institutes are closed due to the corona pandemic crisis these days. Nevertheless, Evolved Node B (eNB) or Next Generation NodeB (gNB) provision is impossible because of technical issues and others, related to security and license. Hence, Own approach will be limited to Implementations and Design and Evaluation of a 5G Testbed via Simulation method. Likewise, 5G Service Based Architecture (SBA) core network will be deployed using docker and docker-compose. Nevertheless, dsTest as a gNB emulator. DsTest offers server emulation and client simulation capabilities for comprehensive testing of 3GPP core network interface functionality and performance[10].

Two 3GPP 5G research platforms, OpenAirInterface (OAI) [6], and Free5GC [8], are involved in utilizing to deploy 5G Core Network components in virtual and physical machines to achieve Thesis goals.

4.2 SA 5GC Testbed Setup

Through the simulation approach, different physical and virtual machines will be used. I.e.:

1. On the first stage: Virtualbox is utilized to run several virtual machines, Ubuntu 18.04 operating system and Ubuntu 18.04 Server, to apply the

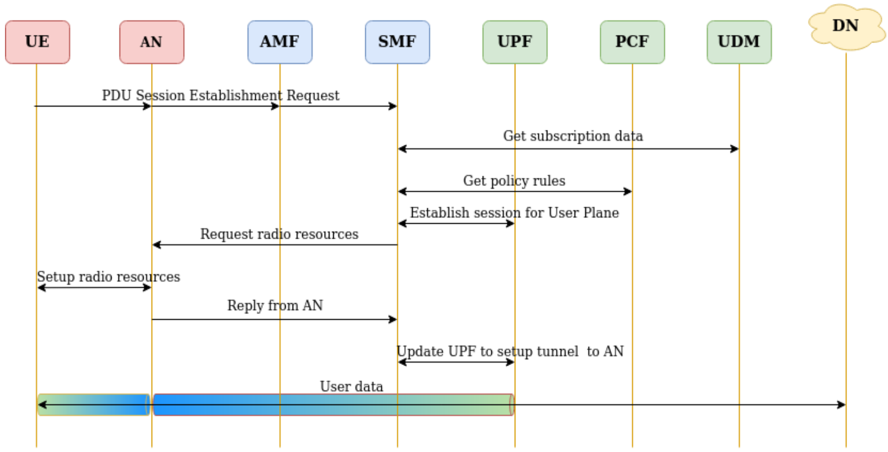

Fig. 19: Simplified PDU Session Establishment procedure [25]

simulation of 5GC elements (AMF, SMF, NRF, and UPF) provided by OpenAirInterface (OAI) community.

2. On the second stage: Physical Ubuntu 18.04 OS is utilized to apply the simulation of 5GC elements, as shown in figure 20, provided by Free5GC community If the expected results are not satisfactory in part 1.

Furthermore, several open source technologies are involved in the study to achieve the aim of this chapter as follows: Docker container, Ansible, Wireshark, Iperf, and dsTest. These terms will be explained in this section and the following sections.

Ansible will make DevOps tasks in this part of the Thesis more effective and less time-consuming. It is an open-source IT automation efficacious tool, which is widely adopted and trusted: because of using simple YAML language and supports different types of infrastructure, e.g., Clouds, Virtual machines. These advantages make it easy for everyone to accept. Furthermore, employ it in all kinds of IT tasks such as configuration management and application deployment. Nevertheless , Ansible is agentless or remotely employed. Besides Ansible components are:

1. Ansible Modules: small programs that get executed on target machines.
2. Ansible Playbook: instructions of the executed programs.
3. Ansible Inventor: list of hosts where those programs get executed.

Openairinterface (OAI): The OAI Public License V1.1. Openairinterface permits the researchers, students, and developers to practice and use the OAI 3GPP 5G Core network projects for studying purposes. Figure 21 depicts the 5GC elements in orange. In other words, it shows fulfilling OAI 5G Core Network components, AMF, SMF, NRF, and UPF (SPGW-U-tiny). Nevertheless, the scenario includes deploying a 5G Service Based Architecture (SBA) core network

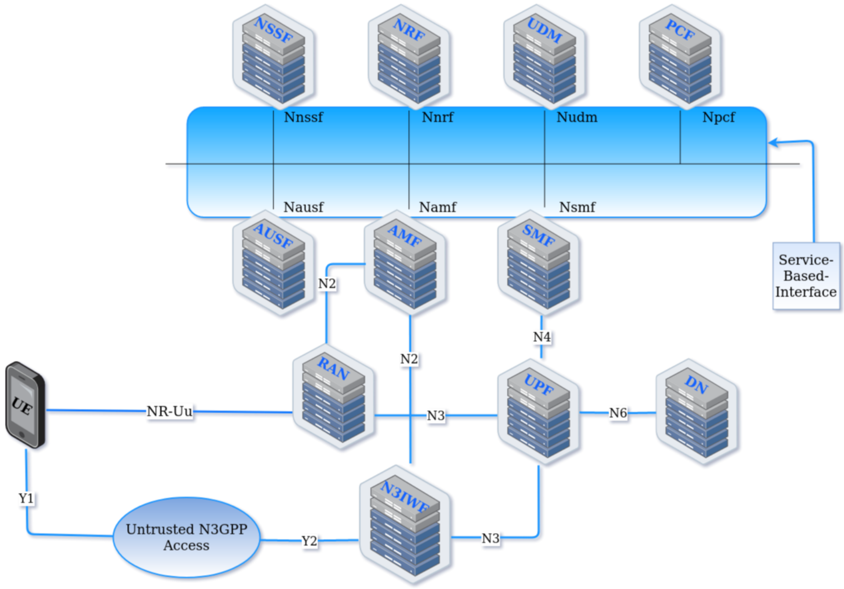

Fig. 20: Stage 2 architecture of free5GC[8]

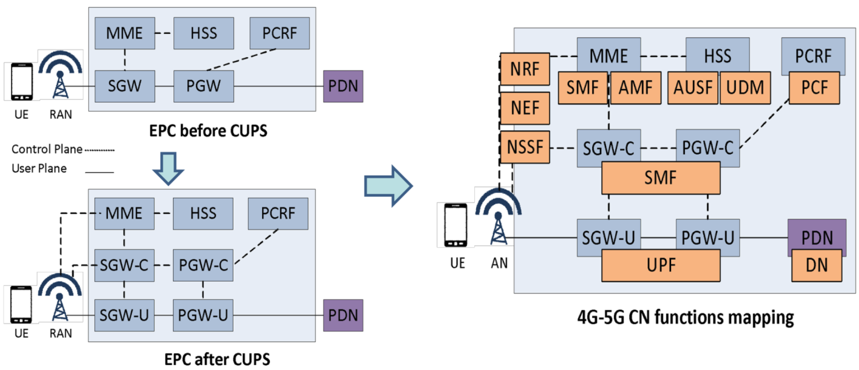

Fig. 21: 5GC Elements corresponding to the 4G [6]

using docker-compose, dsTest as gNB emulator, attaching and detaching UE, and single public data network (PDN) session establishment.

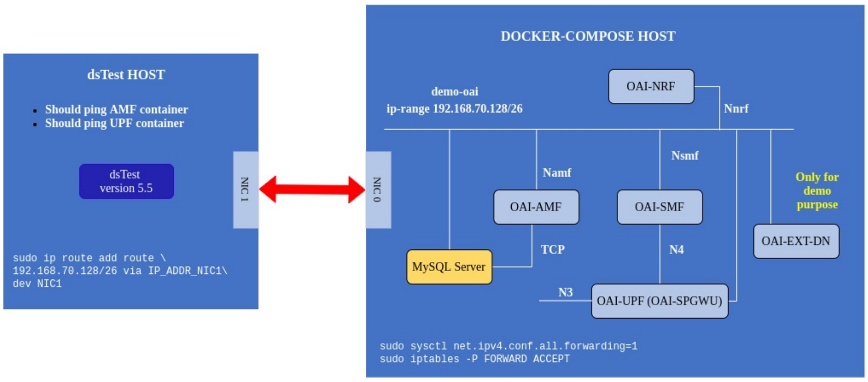

Fig. 22: 5G Core Network Deployment using Docker-Compose and Testing with dsTest [6]

The testbed obtains two host machines; As can be noticed in figure **22**. DsTest host and Docker-compose host, which will be the center of consideration. All 5G core components are running in the Docker-compose host, and they are attached to the same demo-oai. The dstest is deployed in the other machine. The figure shows that the docker-compose host includes an extra container oai-ext-dn, which is only required for the demo goal. Notwithstanding, this container is used in the demo to simulate the downlink traffic. However, because of reasons related to financing, we cannot use a commercial paid gNB emulator (dsTest). Figure **23** shows how can we appropriate gNBsim instead of dstest. Gnbsim is an open-source 5G SA gNB emulator (Rel. 16) for testing 5GS which is written in golang. It simulates NG Application Protocol (NGAP), Non-Access-Stratum (NAS) protocol, and GPRS Tunnelling Protocol User Plane (GTP-U). However, the available gNBsim supports a simulation for one UE and one gNB.

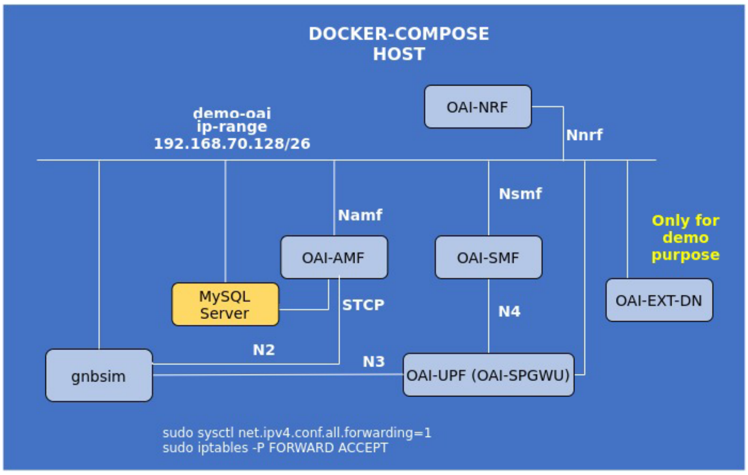

Fig. 23: 5GCN gNBsim [6].

Pre-requisites of deploying the SA 5GC testbed are described in figure 24. Moreover, the following are the required steps during the setup, which will be explained in detail in the Thesis paper:

1. Installation of host operating system Ubuntu 18.04.4 LTS and container operating system Ubuntu 18.04.
2. Docker engine, docker-compose, python, and building container images.
3. Configuring host machines, and configuring OAI 5G Core network functions.
4. Deploying OAI 5GC network, getting a gnbsim docker image, and executing gnbsim Scenario.
5. Wireshark, and Iperf.

Software	Version
docker engine	19.03.6, build 369ce74a3c
docker-compose	1.27.4, build 40524192
Host operating system	Ubuntu 18.04.4 LTS
Container operating system	Ubuntu 18.04
dsTest (Licensed)	5.5
tshark	3.4.4 (Git commit c33f6306cbb2)
wireshark	3.4.4 (Git commit c33f6306cbb2)

Fig. 24: Pre-requisites OAI 5GCN [6].

According to the above, utilizing deployment Docker via Ansible will be more beneficial and operative. Furthermore, the recommendation of Dr.-Ing. Kreuch to present the initially draft instructions of the Ansible playbook of 5GC Network Deployment and Testing with gnbsim will be provided in this section. They are not entirely available yet; because the study and the investigation are ongoing. Nevertheless, this instructions will provide at the abbreviation section in the final thesis paper.

Ansible playbook instructions will directly imported and viewed from the files: docker-compose.yaml and docker-compose-gnbsim.yaml successively [6].

docker-compose.yaml [6]:

```yaml
version: '3.8'
services:
    oai-nrf:
        container_name: "oai-nrf"
        image: oai-nrf:latest
        environment:
            - NRF_INTERFACE_NAME_FOR_SBI=eth0
            - NRF_INTERFACE_PORT_FOR_SBI=80
            - NRF_INTERFACE_HTTP2_PORT_FOR_SBI=9090
            - NRF_API_VERSION=v1
            - INSTANCE=0
            - PID_DIRECTORY=/var/run
        networks:
            public_net:
                ipv4_address: 192.168.70.130
        volumes:
            - ./nrf-healthcheck.sh:/openair-nrf/bin/nrf-healthcheck.sh
        healthcheck:
            test: /bin/bash -c "/openair-nrf/bin/nrf-healthcheck.sh"
            interval: 10s
            timeout: 5s
            retries: 5
    mysql:
        container_name: "mysql"
        image: mysql:5.7
        volumes:
            - ./oai_db.sql:/docker-entrypoint-initdb.d/oai_db.sql
            - ./mysql-healthcheck.sh:/tmp/mysql-healthcheck.sh
        environment:
            - TZ=Europe/Paris
            - MYSQL_DATABASE=oai_db
            - MYSQL_USER=test
            - MYSQL_PASSWORD=test
            - MYSQL_ROOT_PASSWORD=linux
        healthcheck:
            test: /bin/bash -c "/tmp/mysql-healthcheck.sh"
            interval: 10s
            timeout: 5s
            retries: 5
        networks:
            public_net:
                ipv4_address: 192.168.70.131
    oai-amf:
        container_name: "oai-amf"
        image: oai-amf:latest
        environment:
```

```
            - TZ=Europe/paris
            - INSTANCE=0
            - PID_DIRECTORY=/var/run
            - MCC=208
            - MNC=95
            - REGION_ID=128
            - AMF_SET_ID=1
            - SERVED_GUAMI_MCC_0=208
            - SERVED_GUAMI_MNC_0=95
            - SERVED_GUAMI_REGION_ID_0=128
            - SERVED_GUAMI_AMF_SET_ID_0=1
            - SERVED_GUAMI_MCC_1=460
            - SERVED_GUAMI_MNC_1=11
            - SERVED_GUAMI_REGION_ID_1=10
            - SERVED_GUAMI_AMF_SET_ID_1=1
            - PLMN_SUPPORT_MCC=208
            - PLMN_SUPPORT_MNC=95
            - PLMN_SUPPORT_TAC=0xa000
            - SST_0=222
            - SD_0=123
            - SST_1=1
            - SD_1=12
            - AMF_INTERFACE_NAME_FOR_NGAP=eth0
            - AMF_INTERFACE_NAME_FOR_N11=eth0
            - SMF_INSTANCE_ID_0=1
            - SMF_FQDN_0=oai-smf
            - SMF_IPV4_ADDR_0=0.0.0.0
            - SMF_HTTP_VERSION_0=v1
            - SELECTED_0=true
            - SMF_INSTANCE_ID_1=2
            - SMF_FQDN_1=oai-smf
            - SMF_IPV4_ADDR_1=0.0.0.0
            - SMF_HTTP_VERSION_1=v1
            - SELECTED_1=false
            - MYSQL_SERVER=192.168.70.131
            - MYSQL_USER=root
            - MYSQL_PASS=linux
            - MYSQL_DB=oai_db
            - OPERATOR_KEY=63bfa50ee6523365ff14c1f45f88737d
            - NRF_IPV4_ADDRESS=192.168.70.130
            - NRF_PORT=80
            - NF_REGISTRATION=yes
            - SMF_SELECTION=yes
            - USE_FQDN_DNS=yes
            - NRF_API_VERSION=v1
            - NRF_FQDN=oai-nrf
            - AUSF_IPV4_ADDRESS=127.0.0.1
            - AUSF_PORT=80
            - AUSF_API_VERSION=v1
        depends_on:
```

```
            - mysql
        volumes:
            - ./amf-healthcheck.sh:/openair-amf/bin/amf-healthcheck.sh
        healthcheck:
            test: /bin/bash -c "/openair-amf/bin/amf-healthcheck.sh"
            interval: 10s
            timeout: 15s
            retries: 5
        networks:
            public_net:
                ipv4_address: 192.168.70.132
    oai-smf:
        container_name: "oai-smf"
        image: oai-smf:latest
        environment:
            - TZ=Europe/Paris
            - INSTANCE=0
            - PID_DIRECTORY=/var/run
            - SMF_INTERFACE_NAME_FOR_N4=eth0
            - SMF_INTERFACE_NAME_FOR_SBI=eth0
            - SMF_INTERFACE_PORT_FOR_SBI=80
            - SMF_INTERFACE_HTTP2_PORT_FOR_SBI=9090
            - SMF_API_VERSION=v1
            - DEFAULT_DNS_IPV4_ADDRESS=192.168.18.129
            - DEFAULT_DNS_SEC_IPV4_ADDRESS=192.168.18.129
            - AMF_IPV4_ADDRESS=0.0.0.0
            - AMF_PORT=80
            - AMF_API_VERSION=v1
            - AMF_FQDN=oai-amf
            - UDM_IPV4_ADDRESS=127.0.0.1
            - UDM_PORT=80
            - UDM_API_VERSION=v1
            - UDM_FQDN=localhost
            - UPF_IPV4_ADDRESS=192.168.70.134
            - UPF_FQDN_0=oai-spgwu
            - NRF_IPV4_ADDRESS=192.168.70.130
            - NRF_PORT=80
            - NRF_API_VERSION=v1
            - NRF_FQDN=oai-nrf
            - REGISTER_NRF=yes
            - DISCOVER_UPF=yes
            - USE_FQDN_DNS=yes
        depends_on:
            - oai-nrf
        volumes:
            - ./smf-healthcheck.sh:/openair-smf/bin/smf-healthcheck.sh
        healthcheck:
```

```
            test: /bin/bash -c "/openair-smf/bin/smf-healthcheck.sh
    "
            interval: 10s
            timeout: 5s
            retries: 5
        networks:
            public_net:
                ipv4_address: 192.168.70.133
    oai-spgwu:
        container_name: "oai-spgwu"
        image: oai-spgwu-tiny:latest
        environment:
            - TZ=Europe/Paris
            - PID_DIRECTORY=/var/run
            - SGW_INTERFACE_NAME_FOR_S1U_S12_S4_UP=eth0
            - SGW_INTERFACE_NAME_FOR_SX=eth0
            - PGW_INTERFACE_NAME_FOR_SGI=eth0
            - NETWORK_UE_NAT_OPTION=yes
            - NETWORK_UE_IP=12.1.1.0/24
            - SPGWC0_IP_ADDRESS=192.168.70.133
            - BYPASS_UL_PFCP_RULES=no
            - MCC=208
            - MNC=95
            - MNC03=095
            - TAC=40960
            - GW_ID=1
            - REALM=openairinterface.org
            - ENABLE_5G_FEATURES=yes
            - REGISTER_NRF=yes
            - USE_FQDN_NRF=yes
            - UPF_FQDN_5G=oai-spgwu
            - NRF_IPV4_ADDRESS=192.168.70.130
            - NRF_PORT=80
            - NRF_API_VERSION=v1
            - NRF_FQDN=oai-nrf
            - NSSAI_SST_0=222
            - NSSAI_SD_0=123
            - DNN_0=default
        depends_on:
            - oai-nrf
        cap_add:
            - NET_ADMIN
            - SYS_ADMIN
        cap_drop:
            - ALL
        privileged: true
        volumes:
            - ./spgwu-healthcheck.sh:/openair-spgwu-tiny/bin/spgwu-
    healthcheck.sh
        healthcheck:
```

```
            test: /bin/bash -c "/openair-spgwu-tiny/bin/spgwu-healthcheck.sh"
            interval: 10s
            timeout: 5s
            retries: 5
        networks:
            public_net:
                ipv4_address: 192.168.70.134
    oai-ext-dn:
        image: ubuntu:bionic
        privileged: true
        container_name: oai-ext-dn
        entrypoint: /bin/bash -c \
              "apt update; apt install -y iptables iproute2 iperf3 iputils-ping;"\
              "iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE;"\
              "ip route add 12.1.1.0/24 via 192.168.70.134 dev eth0; sleep infinity"
        depends_on:
            - oai-spgwu
        networks:
            public_net:
                ipv4_address: 192.168.70.135
networks:
    # public_net:
    #     external:
    #         name: demo-oai-public-net
    public_net:
        driver: bridge
        name: demo-oai-public-net
        ipam:
            config:
                - subnet: 192.168.70.128/26
        driver_opts:
            com.docker.network.bridge.name: "demo-oai"
```

Listing 1.1: docker-compose.yaml[6].

Following will be the contents of docker-compose-gnbsim.yaml [6]:

```yaml
version: '3.8'
services:
    gnbsim:
        container_name: gnbsim
        image: gnbsim:latest
        privileged: true
        environment:
            - MCC=208
            - MNC=95
            - GNBID=1
            - TAC=0x00a000
            - SST=222
            - SD=00007b
            - PagingDRX=v32
            - RANUENGAPID=0
            - IMEISV=35609204079514
            - MSIN=0000000031
            - RoutingIndicator=1234
            - ProtectionScheme=null
            - KEY=0C0A34601D4F07677303652C0462535B
            - OPc=63bfa50ee6523365ff14c1f45f88737d
            - DNN=default
            - URL=http://www.asnt.org:8080/
            - NRCellID=1
            - USE_FQDN=no
            - NGAPPeerAddr=192.168.70.132
            - GTPuLocalAddr=192.168.70.136
            - GTPuIFname=eth0
        networks:
            public_net:
                ipv4_address: 192.168.70.136
        healthcheck:
            test: /bin/bash -c "ifconfig gtp-gnb"
            interval: 10s
            timeout: 5s
            retries: 5
    gnbsim2:
        container_name: gnbsim2
        image: gnbsim:latest
        privileged: true
        environment:
            - MCC=208
            - MNC=95
            - GNBID=2
            - TAC=0x00a000
            - SST=222
            - SD=00007b
            - PagingDRX=v32
```

```
            - RANUENGAPID=0
            - IMEISV=35609204079514
            - MSIN=0000000032
            - RoutingIndicator=1234
            - ProtectionScheme=null
            - KEY=0C0A34601D4F07677303652C0462535B
            - OPc=63bfa50ee6523365ff14c1f45f88737d
            - DNN=default
            - URL=http://www.asnt.org:8080/
            - NRCellID=1
            - USE_FQDN=no
            - NGAPPeerAddr=192.168.70.132
            - GTPuLocalAddr=192.168.70.137
            - GTPuIFname=eth0
        networks:
            public_net:
                ipv4_address: 192.168.70.137
        healthcheck:
            test: /bin/bash -c "ifconfig gtp-gnb"
            interval: 10s
            timeout: 5s
            retries: 5
    gnbsim3:
        container_name: gnbsim3
        image: gnbsim:latest
        privileged: true
        environment:
            - MCC=208
            - MNC=95
            - GNBID=3
            - TAC=0x00a000
            - SST=222
            - SD=00007b
            - PagingDRX=v32
            - RANUENGAPID=0
            - IMEISV=35609204079514
            - MSIN=0000000033
            - RoutingIndicator=1234
            - ProtectionScheme=null
            - KEY=0C0A34601D4F07677303652C0462535B
            - OPc=63bfa50ee6523365ff14c1f45f88737d
            - DNN=default
            - URL=http://www.asnt.org:8080/
            - NRCellID=1
            - USE_FQDN=no
            - NGAPPeerAddr=192.168.70.132
            - GTPuLocalAddr=192.168.70.138
            - GTPuIFname=eth0
        networks:
            public_net:
```

```
                ipv4_address: 192.168.70.138
        healthcheck:
            test: /bin/bash -c "ifconfig gtp-gnb"
            interval: 10s
            timeout: 5s
            retries: 5
    gnbsim4:
        container_name: gnbsim4
        image: gnbsim:latest
        privileged: true
        environment:
            - MCC=208
            - MNC=95
            - GNBID=4
            - TAC=0x00a000
            - SST=222
            - SD=00007b
            - PagingDRX=v32
            - RANUENGAPID=0
            - IMEISV=35609204079514
            - MSIN=0000000034
            - RoutingIndicator=1234
            - ProtectionScheme=null
            - KEY=0C0A34601D4F07677303652C0462535B
            - OPc=63bfa50ee6523365ff14c1f45f88737d
            - DNN=default
            - URL=http://www.asnt.org:8080/
            - NRCellID=1
            - USE_FQDN=no
            - NGAPPeerAddr=192.168.70.132
            - GTPuLocalAddr=192.168.70.139
            - GTPuIFname=eth0
        networks:
            public_net:
                ipv4_address: 192.168.70.139
        healthcheck:
            test: /bin/bash -c "ifconfig gtp-gnb"
            interval: 10s
            timeout: 5s
            retries: 5
    gnbsim5:
        container_name: gnbsim5
        image: gnbsim:latest
        privileged: true
        environment:
            - MCC=208
            - MNC=95
            - GNBID=5
            - TAC=0x00a000
            - SST=222
```

```
            - SD=00007b
            - PagingDRX=v32
            - RANUENGAPID=0
            - IMEISV=35609204079514
            - MSIN=0000000035
            - RoutingIndicator=1234
            - ProtectionScheme=null
            - KEY=0C0A34601D4F07677303652C0462535B
            - OPc=63bfa50ee6523365ff14c1f45f88737d
            - DNN=default
            - URL=http://www.asnt.org:8080/
            - NRCellID=1
            - USE_FQDN=no
            - NGAPPeerAddr=192.168.70.132
            - GTPuLocalAddr=192.168.70.140
            - GTPuIFname=eth0
        networks:
            public_net:
                ipv4_address: 192.168.70.140
        healthcheck:
            test: /bin/bash -c "ifconfig gtp-gnb"
            interval: 10s
            timeout: 5s
            retries: 5
    gnbsim-fqdn:
        container_name: gnbsim-fqdn
        image: gnbsim:latest
        privileged: true
        environment:
            - MCC=208
            - MNC=95
            - GNBID=5
            - TAC=0x00a000
            - SST=222
            - SD=00007b
            - PagingDRX=v32
            - RANUENGAPID=0
            - IMEISV=35609204079514
            - MSIN=0000000035
            - RoutingIndicator=1234
            - ProtectionScheme=null
            - KEY=0C0A34601D4F07677303652C0462535B
            - OPc=63bfa50ee6523365ff14c1f45f88737d
            - DNN=default
            - URL=http://www.asnt.org:8080/
            - NRCellID=1
            - USE_FQDN=yes
            - AMF_FQDN=amf.oai-5gc.eur
            - GTPuIFname=eth0
        networks:
```

```
            public_net:
        healthcheck:
            test: /bin/bash -c "ifconfig gtp-gnb"
            interval: 10s
            timeout: 5s
            retries: 5
    gnbsim-vpp:
        container_name: gnbsim-vpp
        image: gnbsim:latest
        privileged: true
        environment:
            - MCC=208
            - MNC=95
            - GNBID=5
            - TAC=0x00a000
            - SST=222
            - SD=00007b
            - PagingDRX=v32
            - RANUENGAPID=0
            - IMEISV=35609204079514
            - MSIN=0000000035
            - RoutingIndicator=1234
            - ProtectionScheme=null
            - KEY=0C0A34601D4F07677303652C0462535B
            - OPc=63bfa50ee6523365ff14c1f45f88737d
            - DNN=default
            - URL=http://www.asnt.org:8080/
            - NRCellID=1
            - USE_FQDN=no
#           - USE_FQDN=yes
#           - AMF_FQDN=amf.oai-5gc.eur
            - NGAPPeerAddr=192.168.71.132
            - GTPuLocalAddr=192.168.72.141
            - GTPuIFname=eth1
        networks:
            public_net_core:
                ipv4_address: 192.168.71.141
            public_net_access:
                ipv4_address: 192.168.72.141
        healthcheck:
            test: /bin/bash -c "ifconfig gtp-gnb"
            interval: 10s
            timeout: 5s
            retries: 5
networks:
    public_net:
        external:
            name: demo-oai-public-net
    public_net_core:
        name: oai-public-core
```

```
        ipam:
            config:
                - subnet: 192.168.71.0/24
    public_net_access:
        name: oai-public-access
        ipam:
            config:
                - subnet: 192.168.72.0/24
# Incase the user wants docker-compose to create a bridge rather
    than creating the bridge manually then uncomment the below
    lines
#     public_net:
#         driver: bridge
#         name: demo-oai-public-net
#         ipam:
#             config:
#                 - subnet: 192.168.70.128/26
#         driver_opts:
#             com.docker.network.bridge.name: "demo-oai"
```

Listing 1.2: docker-compose-gnbsim.yaml[6].

As introduced previously in the draft copy of docker-compose.yaml and docker-compose-gnbsim.yaml both ansible files are preconfigured for executing the gnbsim scenario and will be modified for a test. Then we run the following command to create and launch gnbsim.

```
oai-cn5g-fed/docker-compose$ docker-compose -f docker-compose-
    gnbsim.yaml up -d gnbsim
```

Listing 1.3: Run gnbsim[6].

Before going to the next step, we need to check that all services' statuses are working fine, as displayed in figure 25.

```
oai-cn5g-fed/docker-compose$ docker ps -a
CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS                     PORTS                           NAMES
2ad428f94fb0   gnbsim:latest             "/gnbsim/bin/entrypo…"   33 seconds ago   Up 32 seconds (healthy)                                    gnbsim
c25db05aa023   ubuntu:bionic             "/bin/bash -c ' apt …"   4 minutes ago    Up 4 minutes                                               oai-ext-dn
31b6391a3a41   oai-amf:latest            "/bin/bash /openair-…"   4 minutes ago    Up 4 minutes (healthy)     80/tcp, 9090/tcp, 38412/sctp    oai-amf
753ae61f715f   oai-spgwu-tiny:latest     "/openair-spgwu-tiny…"   4 minutes ago    Up 4 minutes (healthy)     2152/udp, 8805/udp              oai-spgwu
84c164ab8136   oai-smf:latest            "/bin/bash /openair-…"   4 minutes ago    Up 4 minutes (healthy)     80/tcp, 9090/tcp, 8805/udp      oai-smf
6f0ce91e4efb   oai-nrf:latest            "/bin/bash /openair-…"   4 minutes ago    Up 4 minutes (healthy)     80/tcp, 9090/tcp                oai-nrf
565617169b42   mysql:5.7                 "docker-entrypoint.s…"   4 minutes ago    Up 4 minutes (healthy)     3306/tcp, 33060/tcp             mysql
```

Fig. 25: Gnbsim is healthy[6].

Momentarily we can implement some traffic tests:

- Ping test: Figure 26 shows us how we ping the UE From the external DN container.
- Iperf test (server/client): Between gnbsim UE and the external DN container, we can apply iperf traffic tests because we are able to create any node as an

iperf server/client. Figure 27, figure 28 show the iperf traffic tests and results in the server and client state.

```
$ docker exec -it oai-ext-dn ping -c 3 12.1.1.2
PING 12.1.1.2 (12.1.1.2) 56(84) bytes of data.
64 bytes from 12.1.1.2: icmp_seq=1 ttl=64 time=0.235 ms
64 bytes from 12.1.1.2: icmp_seq=2 ttl=64 time=0.145 ms
64 bytes from 12.1.1.2: icmp_seq=3 ttl=64 time=0.448 ms

--- 12.1.1.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.145/0.276/0.448/0.127 ms
rohan@rohan:~/gitrepo/oai-cn5g-fed/docker-compose$
```

Fig. 26: ping UE from external DN container [6].

5 Evaluation

Wireshark and Iperf, as open-source tools, are employed to take measurements and read the values to get the best tolerable results. Wireshark is a free open-source network packet analyzer. It is used to capture data packets and display the most accurate details achievable about them. Similarly, Iperf is an open-source tool for active measurements of the peak possible bandwidth on IP networks. Besides, it supports harmonizing many factors related to timing and protocols (Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Stream Control Transmission Protocol (SCTP) with IPv4 and IPv6)[22]. Other significant advantages are that Iperf can be utilized in two forms: server and client modes. In addition, it supports End-to-end (E2E) testing and affords a throughput measurement report between the two ends in one or both directions. It handles both data streams: Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). The primary target of (E2E) testing is to examine the end user's activity by simulating the real user scenario and validating the system under test and its elements for combination and data integrity. It is beneficial for analyzing network traffic in real-time. Figure **29** presents the Wireshark user interface. We start the 5GC Network components: NRF, Mysql, AMF, SMF, and SPGWU. Then we capture packets on the docker-compose host as shown in the bottom section. The initial data exchanges between the core network components are evident here. Furthermore, capturing these packets is fundamental, allowing us to see the expected exchange between SMF, RNF, and UPF. SMF can discover UPF using the service discovery feature of NRF. The core network should be correctly configured and healthy, as displayed in figure **30**. The completely evaluation process, including taking measurements and monitoring time synchronization, reliability, latency, and determinism, will be explained in detail in the master's paper[6].

```
$ docker exec -it oai-ext-dn iperf3 -s
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 12.1.1.2, port 43339
[  5] local 192.168.70.135 port 5201 connected to 12.1.1.2 port 55553
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  73.8 MBytes   619 Mbits/sec
[  5]   1.00-2.00   sec  76.3 MBytes   640 Mbits/sec
[  5]   2.00-3.00   sec  77.8 MBytes   653 Mbits/sec
[  5]   3.00-4.00   sec  66.7 MBytes   560 Mbits/sec
[  5]   4.00-5.00   sec  71.9 MBytes   603 Mbits/sec
[  5]   5.00-6.00   sec  80.2 MBytes   673 Mbits/sec
[  5]   6.00-7.00   sec  76.5 MBytes   642 Mbits/sec
[  5]   7.00-8.00   sec  78.6 MBytes   659 Mbits/sec
[  5]   8.00-9.00   sec  74.5 MBytes   625 Mbits/sec
[  5]   9.00-10.00  sec  75.5 MBytes   634 Mbits/sec
[  5]  10.00-10.01  sec   740 KBytes   719 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.01  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.01  sec   753 MBytes   631 Mbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

Fig. 27: Iperf client traffic test between gnbsim UE and external DN node [6].

```
$ docker exec -it gnbsim iperf3 -c 192.168.70.135 -B 12.1.1.2
Connecting to host 192.168.70.135, port 5201
[  5] local 12.1.1.2 port 55553 connected to 192.168.70.135 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  77.6 MBytes   651 Mbits/sec   29    600 KBytes
[  5]   1.00-2.00   sec  76.2 MBytes   640 Mbits/sec    0    690 KBytes
[  5]   2.00-3.00   sec  77.5 MBytes   650 Mbits/sec    4    585 KBytes
[  5]   3.00-4.00   sec  66.2 MBytes   556 Mbits/sec  390    354 KBytes
[  5]   4.00-5.00   sec  72.5 MBytes   608 Mbits/sec    0    481 KBytes
[  5]   5.00-6.00   sec  80.0 MBytes   671 Mbits/sec    0    598 KBytes
[  5]   6.00-7.00   sec  76.2 MBytes   640 Mbits/sec    7    684 KBytes
[  5]   7.00-8.00   sec  78.8 MBytes   661 Mbits/sec    3    578 KBytes
[  5]   8.00-9.00   sec  75.0 MBytes   629 Mbits/sec    1    670 KBytes
[  5]   9.00-10.00  sec  75.0 MBytes   629 Mbits/sec    5    554 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec   755 MBytes   633 Mbits/sec  439             sender
[  5]   0.00-10.00  sec   753 MBytes   631 Mbits/sec                  receiver

iperf Done.
```

Fig. 28: Iperf client traffic test between gnbsim UE and external DN node [6].

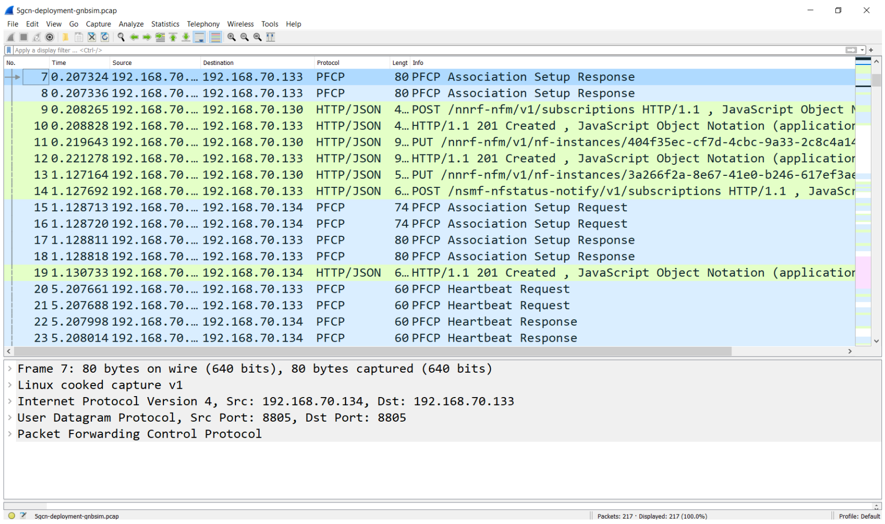

Fig. 29: 5GC Network Deployment gNBsim [6]

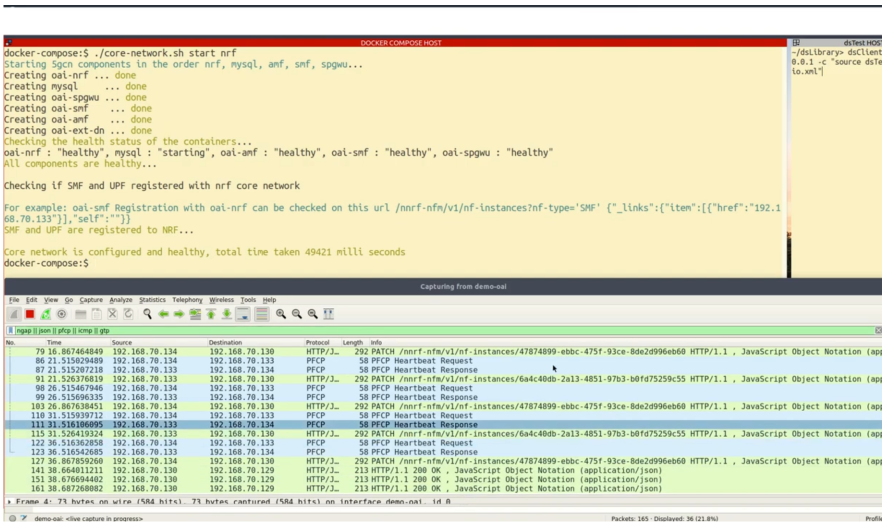

Fig. 30: Core Network are healthy and configured [6].

6 Expected Result

The expected result usually will be: releasing a signal with 5g logo and uplink or downlink data after turning on the WiFi in all connected mobile phones, the ability to connect a massive number of mobile phones, all phones can watch HD video at the same time. However, the expected result will be limited to the captured packets in Wireshark and Iperf tools because of international health status and costs. As we previously explained, The initial data exchanges between the core network components are evident At the bottom window of the Wireshark. Additionally, capturing these packets is significant, seeing the expected exchange between SMF, RNF, and UPF. Furthermore, SMF can discover UPF using the service discovery feature of NRF. We can observe intermediary request responses between UPF and SMF. Wireshark shows the signaling process and when UE is accessed, such as registration, authentication, PDU session establishment, an uplink, or downlink GTP packet. Within viewing the AMF log, which showed in figure 31, we can see that there is already a user Observed.

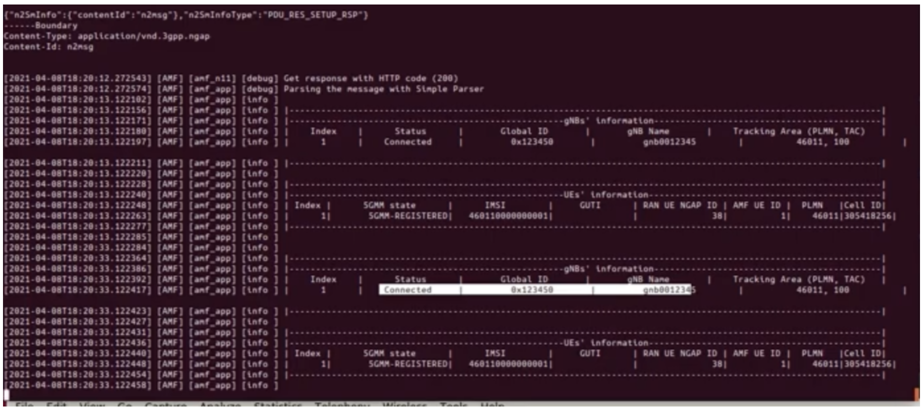

Fig. 31: A user observed in AMF log [6].

In the SMF log, we can understand that the IP sent by SMF is the same as the phone. Nevertheless, it displays data and other session messages. After stopping the 5GC components services, we can see that the phone loses the signal in case of the possibility to use a phone. All the logs file, including 5gcn-deployment-gnbsim.pcap, amf.log, initialmessage.log, smf.log, nrf.log, and spgwu.log will be viewed in the final paper.

7 Work Plan

The 5G-TSN integration is a crucial topic of importance for all communication companies.The 5G and TSN mixture is ideal for intelligent factories, given ultra-reliability and low latency characteristics. Although, a particular level of integration of the couple technologies is required to present End-to-End Ethernet connectivity to match the industrial requirements. That allows us to imagine; How much the integrated time synchronization through wired TSN and wireless 5G domains affords a regular reference time for industrial endpoints. 5G is also integrated with the given TSN tool used in an unusual deployment to accommodate limited low latency. End-to-End Ultra-reliability and high availability are granted by the arrangement of the disjoint forwarding paths of the 5G and TSN segments [12]. This study aims to understand the open-source 5G core Network benefits and Time-sensitive Network (TSN) features. The integration between both technologies is the key To achieve eMBB, mMTC, and uRLLC. In this Thesis, we tried to design and implement the best effort testbed environment that can be provided by integrating open source 5GC and TSN. Chapter 2 (State of the Art) in the thesis paper can give us all the background that we need to understand this thesis. It provides more knowledge about architecture. In addition to that, we will also deploy a Standalone 5GC Network. Chapters 4, 5, and 6 (Design and Specifications) explain the implementation's needed knowledge in detail. Last but not least, Wireshark and Iperf are utilized to achieve a Comparative Analysis of OS 5G Core Implementations and Design and Evaluation of a 5G Testbed.

7.1 Thesis Structure

Thesis Structures illustrated in figures **32**,**33**,**34**,**35**,

TABLE OF CONTENTS

Fig. 32: Thesis Structure TOC1

viii *TABLE OF CONTENTS*

Fig. 33: Thesis Structure TOC2

TABLE OF CONTENTS ix

Fig. 34: Thesis Structure TOC3

x *TABLE OF CONTENTS*

6.8 Conclusion 47

7 Summary and Further Work **49**
7.1 Overview 49
7.2 Conclusions and Impact 50
7.3 Outlook 50
7.3.1 Enhancement of Hardware 50
7.3.2 Integration of hardware 50
7.3.3 Supported scheduled TSN traffic 50

A Specifications **I**
A.1 Specification 1 I
A.2 Specification 2 I

B Test Results **III**
B.1 Conformance Results III
B.2 Performance Results IV
B.2.1 Histograms IV
B.2.2 Lineplots IV

Acronyms **V**

Glossary **VII**

Bibliography **IX**

Fig. 35: Thesis Structure TOC4

7.2 Gantt

As mandate by Regulations Governing General Study and Examination Procedures (AllgStuPO), the period to work for a master thesis should be in six months. The proposed timeline to commence this work is depicted in figure 36 below.

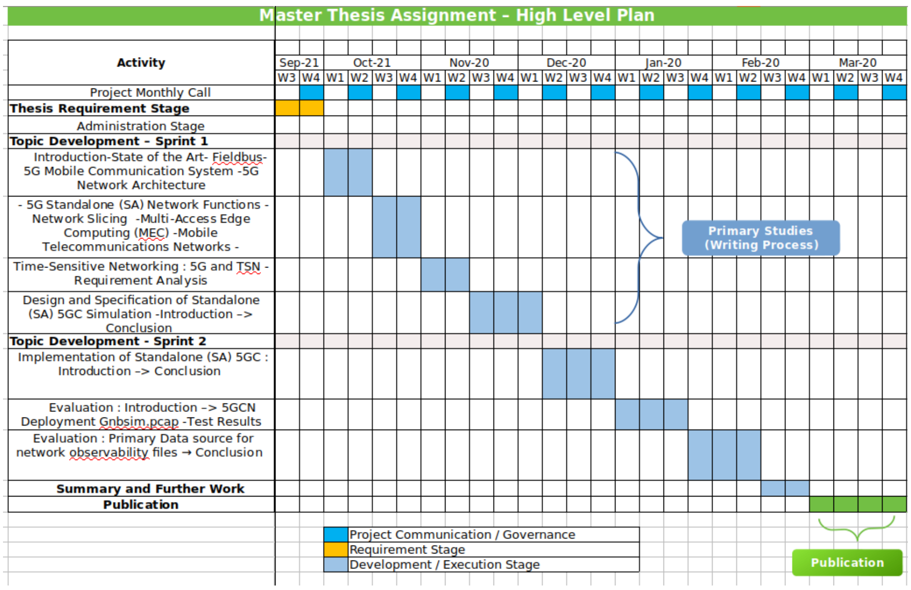

Fig. 36: Master Thesis High level Timeline.

References

[1] G. T. 22.261. "5G Service requirements for next generation new services and markets," in: (2018). (version 15.5.0 Release 15 2018).

[2] *5G implementation guidelines,* GSMA , 2019. URL: https://www.gsma.com/futurenetworks/wiki/5g-implementation-guidelines/#8cb93af06277b80cf7b9805cf9960882.

[3] *5G-ACIA White Paper Integration of 5G with Time-Sensitive Networking for Industrial Communications.* 5G Alliance for Connected Industries and Automation (5G-ACIA), 2021. URL: https://www.5g-acia.org,. (accessed on Feb.2021).

[4] J. M. Meredith et al. *3GPP Technical Report 22.804 Version 1.0.0: Study on Communication for Automation in Vertical domains (CAV).* 3GPP. 2017. URL: https://www.3gpp.org/DynaReport/22804.htm. (Online; accessed May 15, 2018).

[5] NGMN Alliance. “5G white paper”. In: *Next generation mobile networks, white paper* 1 (2015).

[6] OpenAirInterface Software Alliance. *OpenAirInterface | 5G software alliance for democratising wireless innovation 2014*. URL: https://openairinterface.org/. (accessed: January 2021).

[7] Lucia Lo Bello. “Novel trends in automotive networks: A perspective on Ethernet and the IEEE Audio Video Bridging”. In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE. 2014, pp. 1–8.

[8] Hsiaotsui Chen. *Open-source project for 5th generation (5G) mobile core networks 2019*. URL: https://free5gc.org/. (accessed: January 2019).

[9] Peter Danielis et al. “Survey on real-time communication via ethernet in industrial automation environments”. In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. IEEE. 2014, pp. 1–8.

[10] *Developing Solutions, Inc. Unique 3G/LTE/5GC Core Network Testing Solutions*. Developing Solutions, Inc., 2009-2021. URL: https://www.developingsolutions.com/,.

[11] ETSI. “5G: System architecture for the 5G system”. In: (Tech. Rep.,2018). (3GPP ts 23.501 version 15.2.0 release 15),

[12] J. Farkas, G. Miklós B. Varga, and J. Sachs. “5G-TSN integration meets networking requirements for industrial automation”. In: (2019), pp. 1–11.

[13] ITU-T G.8275.1. *Precision time protocol telecom profile for phase/time synchronization with full timing support from the network,* ITU-T, URL: https://www.itu.int/rec/T-REC-G.8275.1/en.

[14] Eric Gardiner. “The Avnu alliance theory of operation for TSN-enabled industrial systems”. In: *IEEE Communications Standards Magazine* 2.1 (2018), pp. 5–5.

[15] Eric Gardiner. *Theory of Operation for TSN-enabled Systems Applied to Industrial Markets*. Avnu Alliance, 2017. URL: https://avnu.org/knowledgebase/theory-of-operation/,.

[16] David Ginthor et al. “Analysis of multi-user scheduling in a TSN-enabled 5G system for industrial applications”. In: *Proceedings - IEEE International Conference on Industrial Internet Cloud, ICII 2019* Icii (2019), pp. 190–199.

[17] Omid Givehchi et al. “Control-as-a-service from the cloud: A case study for using virtualized PLCs”. In: *2014 10th IEEE Workshop on Factory Communication Systems (WFCS 2014)*. IEEE. 2014, pp. 1–4.

[18] Henning Kagermann, Wolfgang Wahlster, Johannes Helbig, et al. “Recommendations for implementing the strategic initiative Industrie 4.0: Final report of the Industrie 4.0 Working Group”. In: *Forschungsunion: Berlin, Germany* (2013).

[19] Rick Kreifeldt et al. “AVB for professional A/V use”. In: *Whitepaper, Knorrstrasse* 147 (2009), p. 80788.

[20] Christian Mannweiler et al. “Reliable and deterministic mobile communications for industry 4.0: Key challenges and solutions for the integration

of the 3GPP 5G system with IEEE time-sensitive networking". In: *24. ITG-Symposium on Mobile Communication - Technologies and Applications* (2020), pp. 64–69.

[21] Arne Neumann et al. "Towards integration of Industrial Ethernet with 5G mobile networks". In: *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS* 2018-June (2018), pp. 1–4.

[22] Jon Dugan - Seth Elliott - Bruce A. Mah - Jeff Poskanzer - Kaustubh Prabhu. *iPerf - The ultimate speed test tool for TCP, UDP and SCTP.* Distributed Applications Support Team (DAST), URL: https://iperf.fr/,.

[23] François Rancy. "IMT for 2020 and beyond". In: *5G Outlook-Innovations and Applications* (2016), p. 69.

[24] Felix Ring et al. *Wireless real-time communication for industrial automation systems in simulation and reality.* na, 2012.

[25] Stefan Rommer et al. "Chapter 6 - Session management". In: *5G Core Networks.* Ed. by Stefan Rommer et al. Academic Press, 2020, pp. 111–136.

[26] T. Sauter. "The Three Generations of Field-Level Networks—Evolution and Compatibility Issues". In: *IEEE Transactions on Industrial Electronics* 57.11 (2010), pp. 3585–3595.

[27] M Series. "Minimum requirements related to technical performance for IMT-2020 radio interface (s)". In: *Report* (2017), pp. 2410–.

[28] "SHARP: Towards the Integration of Time-Sensitive Communications in Legacy LAN/WLAN". In: *2018 IEEE Globecom Workshops, GC Wkshps 2018 - Proceedings* (2019), pp. 2–8.

[29] Gil Shoshani, Stephen Mitschke, and Stan Stephan. "Industrial Fieldbus technology and Fieldbus cable overview—Cable standards and electrical qualifications". In: *2010 Record of Conference Papers Industry Applications Society 57th Annual Petroleum and Chemical Industry Conference (PCIC).* IEEE. 2010, pp. 1–10.

[30] G. Std. *3GPP: Technical Specification Group Services and System Aspects: System Architecture for the 5G System,* 3GPP TR 23.501, vol. V15.2.0, 2018.

[31] J-P Thomesse. "Fieldbus technology in industrial automation". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1073–1101.

[32] *Time-Sensitive Networking (TSN) Task Group,* IEEE , 2019. URL: https://1.ieee802.org/tsn/,. (accessed on February 3, 2021).

[33] Lukasz Wisniewski et al. "Fast and simple scheduling algorithm for PROFINET IRT networks". In: *2012 9th IEEE International Workshop on Factory Communication Systems.* IEEE. 2012, pp. 141–144.

[34] M. Wollschlaeger, T. Sauter, and J. Jasperneite. "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0". In: *IEEE Industrial Electronics Magazine* 11.1 (2017), pp. 17–27.

[35] Jordon Woods. *Time Sensitive Networking: Five Ways the IEEE Standards Will Advance Industry 4.0.* URL: https://www.embeddedcomputing.com/application/networking-5g/ethernet-serial/time-sensitive-

networking-five-ways-the-ieee-standards-will-advance-industry-4-0. (accessed:Jul 09, 2020).