DETERMINISTIC6G

Edge computing and situational awareness via digital twinning

D3.3

The DETERMINISTIC6G project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement no 1010965604.

Edge computing and situational awareness via digital twinning

Grant agreement number: Project title: Project acronym: Project website: Programme:	101096504 Deterministic E2E communication with 6G DETERMINISTIC6G Deterministic6g.eu EU JU SNS Phase 1

Deliverable type:	Report
Deliverable reference number:	D3.3
Contributing workpackages:	WP3
Dissemination level:	PUBLIC
Due date:	M18
Actual submission date:	28-06-2024
Responsible organization:	SAL
Editor(s):	Armin Hadziaganovic
Version number:	V1.0
Status:	Final
Short abstract:	This deliverable describes different architectural deployment scenarios for edge computing platforms, i.e. MEC. It also includes initial performance results and gives insights into the benefits of local MEC versus cloud-based deployment for low delay variation. Additionally, the integration aspects of Time-Sensitive Networking with Edge computing to support cloudified applications are explored. The second part of the deliverable presents a concept of gaining situational awareness through the interaction of operation technology digital twin and 6G system digital twin. It shows in an example what parameters are required to generate situational awareness, what critical scenarios can be predicted and which action measures can be applied to the communication system and factory controlling system to avoid failure.
Keywords:	Edge computing, MEC, TSN, 6G, Adaptive manufacturing, Digital twin, Situational awareness, failsafe

Contributor(s):	János Harmatos (ETH) Dávid Jocha (ETH) Mahin Ahmed (SAL)

	Armin Hadziaganovic (SAL) Damir Hamidovic (SAL) Raheeb Muzaffar (SAL) Jose Costa Requena (CMC) Gourav Prateek Sharma (KTH) Joachim Sachs (EDD), Drissa Houatra (OR)

Reviewers:	Gourav Prateek Sharma (KTH) Giulia Bigoni (IUVO) Inés Álvarez Vadillo (ABB)

Revision History

04/06/2024	Draft version for first internal review
21/06/2024	PMT review draft
28/06/2024	Final submission version

Disclaimer

Executive summary

The advancements in digital transformation and high-demanding applications requiring reliability and low latency necessitate robust network infrastructure for End-to-End (E2E) dependable time-critical communication services, integrating wired, wireless, and computing domains with digital representations of physical objects like robots and sensors. To achieve this, control, monitoring, and maintenance must shift towards an edge computing paradigm, enhancing flexibility and providing computational and storage resources. This report explores the Edge Computing platform (MEC), proposed to support low-latency, E2E applications by deploying time-aware applications closer to each other. It identifies various deployment scenarios where MEC can be integrated into or connected to Mobile Network Operator (MNO) infrastructure to minimize latency and delay variation compared to cloud-based deployments. The report also examines the integration of Time-Sensitive Networking (TSN) applications into MEC, presenting initial solutions. A framework proposal for the seamless support of the IEEE 802.1Qbv scheduled traffic standard in an E2E scope for cloudified time-critical applications is also described.

Additionally, the report highlights the value of Digital Twins (DTs) in optimizing and maintaining Cyber-Physical Systems (CPS) across domains like manufacturing, and communication. It discusses the need for interaction between DTs to enhance Situational Awareness (SA) for critical services. A use case involving manufacturing, 6G communication, and mobile User Equipment (UE) is analyzed to identify critical operating points and how SA can predict and mitigate the risk of failure in these scenarios. The report details the situational information generated by each DT, the relevant parameters for deriving SA, and the interaction procedures for joint task planning and execution, addressing the challenges of running and maintaining DTs.

Revision History 1
Disclaimer 2
Executive summary 3
List of Figures 6
1 Introduction 7
1.1 DETERMINISTIC6G Approach 7
1.2 Relation to other work packages 8
1.3 Objective of the document 9
1.4 Structure and scope of the document 10
2 Edge computing 10
2.1 Background 10
2.1.1 ETSI MEC architecture 10
2.1.2 3GPP MEC architecture integration 13
2.2 3GPP Edge computing deployment 16
2.2.1 3GPP Edge computing application deployment flow 16
2.2.2 Deterministic services in MEC 18
2.3 Architecture integration of 3GPP edge computing and TSN support 19
2.3.1 Aspects of control plane design and integration with edge computing 23
2.4 Integration aspects of compute domain with TSN communication to seamless support of IEEE 802.1Qbv scheduled traffic 25
2.4.1 Enablers and capabilities for supporting cloudified, time-critical applications 25
2.4.2 Issues with seamless support of TSN 802.1Qbv-based communication for cloudified applications 28
2.4.3 IEEE 802.1Qbv-aware traffic handling in Edge computing domain 32
3 Situational awareness via Digital twinning 38
3.1 Background 38
3.2 Cyber-Physical System (CPS) 39
3.2.1 Operation Technology DT 39
3.2.2 6G DT 40
3.3 Situational awareness via DT interaction 40
3.4 Example use case 41
3.4.1 Critical operating points 42
3.5 Resource and environmental awareness 43

3.5.1 Parameters relevant to situational awareness 46
3.6 Impact on data-driven latency predictions 49
3.7 Action measures 50
3.7.1 6G DT action measures 50
3.7.2 OT DT action measures 51
3.8 Costs and challenges 51
4 Conclusion 52
References 53
List of abbreviations 55

List of Figures

Figure 1.1 - Relationship of D3.3 to other work packages and deliverables 9
Figure 2.1 - ETSI MEC architecture 11
Figure 2.2 - High-level view of the MEC architecture 11
Figure 2.3 - 3GPP TS 23.558 defined edge computing architecture 14
Figure 2.4 - Setup for validating MEC support of TSN applications 17
Figure 2.5 - Rout Trip (RTT) measurements between UE and edge server (on the right) or cloud server (on the left). 18
Figure 2.6 - Edge computing platform integrated into 3GPP system 20
Figure 2.7 - Standalone edge computing deployment 22
Figure 2.8 - Example time sensitive SDN proposal (from [MPA19]) 24
Figure 2.9 - Dedicated vs. cloudified application deployment 29
Figure 2.10 - Packet ordering variances in case of deadline scheduling 30
Figure 2.11 - Effect of packet ordering uncertainties 31
Figure 2.12 - Coordinated time-gating mechanism on container level 33
Figure 2.13 - Operation of the coordinated time-gating mechanism 33
Figure 2.14 - Centralized 802.1Qbv-aware traffic handling mechanism 35
Figure 2.15 - Operation details of the centralized 802.1Qbv-aware traffic handling mechanism 35
Figure 2.16 - Packet arrival for per-container shaping (left) and centralized shaping (right). 37
Figure 3.1 - Situational awareness via DTs interaction 40
Figure 3.2 - Adaptive manufacturing use case 41
Figure 3.3 - Resource awareness, cell resource utilization 44
Figure 3.4 - Radio awareness, signal coverage 45
Figure 3.5 - Data traffic requirements awareness 46
Figure 3.6 - Interaction procedure between DTs 48

1 Introduction

The advancements in the digital transformation of industries and society and the emergence of high-demand applications present unique requirements on the underlying network infrastructure to support E2E dependable time-critical communication services. An E2E communication service includes the integration of wired, wireless, and computing domains together with its interaction with the digital representation of physical objects such as robots, machines, sensors, and devices. Therefore, the control, monitoring, and maintenance functionality need to be moved towards an edge computing paradigm in a distributed fashion to enable flexibility while providing computation and storage facilities. While the existing edge computing systems can support services to host applications requiring high computation, enhancements are needed to ensure dependable, time-critical, and deterministic communication performance. The challenge to enable E2E deterministic communication service can be addressed by integrating features of TSN and Deterministic Networking (DetNet) into the edge computing domain. In addition, predicting the behavior of the physical systems and processes while generating different alternative scenarios can further enrich the E2E time-aware communication system. This report presents an architectural framework and mechanisms for enabling dependable and deterministic communication at the edge computing domain and provides initial concepts on obtaining SA from 6G and operational digital twins that can be used to manage and configure 6G deterministic communication more effectively.

1.1 DETERMINISTIC6G Approach

Time-critical communication has in the past been mainly prevalent in industrial automation scenarios with special compute hardware like Programmable Logic Controller (PLC), and is based on a wired communication system, such as EtherCat and Powerlink, which is limited to local and isolated network domains which are configured to the specific purpose of the local applications. With the standardization of TSN and DetNet, similar capabilities are being introduced into the Ethernet and IP networking technologies, which thereby provide a converged multi-service network allowing time-critical applications in a managed network infrastructure allowing for consistent performance with zero packet loss and guaranteed low and bounded latency. The underlying principles are that the network elements (i.e. bridges or routers) and the PLCs can provide a consistent and known performance with negligible stochastic variation, which allows to manage the network configuration based on the needs of time-critical applications with known traffic characteristics and requirements.

It turns out that several elements in the digitalization journey introduce characteristics that deviate from the assumptions that are considered as baseline in the planning of deterministic networks. There is often an assumption for compute and communication elements, and also applications, that any stochastic behavior can be minimized such that the time characteristics of the element can be clearly associated with tight minimum/maximum bounds. Cloud computing provides efficient scalable compute, but introduces uncertainty in execution times; wireless communications provide flexibility and simplicity, but with inherently stochastic components that lead to packet delay variations exceeding significantly those found in wired counterparts; and applications embrace novel technologies (e.g. Machine Learning (ML)-based or machine-vision-based control) where the traffic characteristics deviate from the strictly deterministic behavior of old-school control. In addition, there will be an increase in dynamic behavior where characteristics of applications, and network or compute elements may change over time in contrast to a static behavior that does not change during runtime. It turns out that these deviations of stochastic characteristics make traditional approaches

to planning and configuration of end-to-end time-critical communication networks such as TSN or DetNet, fall short in their performance regarding service performance, scalability and efficiency. Instead, a revolutionary approach to the design, planning and operation of time-critical networks is needed that fully embraces the variability but also dynamic changes that come at the side of introducing wireless connectivity, cloud compute and application innovation. DETERMINISTIC6G has as objective to address these challenges, including the planning of resource allocation for diverse time-critical services E2E over multiple domains, providing efficient resource usage and a scalable solution [SPS+23].

DETERMINISTIC6G takes a novel approach towards converged future infrastructures for scalable cyber-physical systems deployment. With respect to networked infrastructures, DETERMINISTIC6G advocates for: (I) the acceptance and integration of stochastic elements (like wireless links and computational elements) with respect to their stochastic behavior captured through either short-term or longer-term envelopes. Monitoring and prediction of Key Performance Indicators (KPIs), for instance latency or reliability, can be leveraged to make individual elements plannable despite a remaining stochastic variance. Nevertheless, system enhancements to mitigate stochastic variances in communication and compute elements are also developed. (II) Next, DETERMINISTIC6G attempts the management of the entire E2E interaction loop (e.g. the control loop) with the underlying stochastic characteristics, especially embracing the integration of compute elements. (III) Finally, due to unavoidable stochastic degradations of individual elements, DETERMINISTIC6G advocates allowing for adaptation between the applications running on top such converged and managed network infrastructures. The idea is to introduce flexibility in the application operation such that its requirements can be adjusted at runtime based on prevailing system conditions. This encompasses a larger set of application requirements that (a) can also accept stochastic E2E KPIs, and (b) can possibly adapt E2E KPI requirements at run-time in harmonization with the networked infrastructure. DETERMINISTIC6G builds on a notion of time-awareness by ensuring accurate and reliable time synchronicity while also ensuring security-by-design for such dependable time-critical communications. Generally, we extend a notion of deterministic communication (where all behavior of network and compute nodes and applications is pre-determined) towards dependable time-critical communication, where the focus is on ensuring that the communication (and compute) characteristics are managed in order to provide the KPIs and reliability levels that are required by the application. DETERMINISTIC6G facilitates architectures and algorithms for scalable and converged future network infrastructures that enable dependable time-critical communication E2E, across domains and including 6G.

1.2 Relation to other work packages

This deliverable is part of work package 3 and has linkages with other technical work packages, as presented in Figure 1.1. The deliverable takes inputs in terms of edge computing and SA using digital twining from the use case requirements and the DETERMINISTIC6G architecture investigated in WP1, as formulated in deliverables [DET23-D11] and [DET24-D12]. The deliverable [DET23-D21] on 6G centric enablers from WP2 discusses architectural aspects of latency prediction which are used as basis on identifying gaps to produce accurate delay predictions for SA. The E2E scheduling aspects discussed in deliverable [DET23-D31] provides inputs to the traffic scheduling mechanisms in integrating the edge compute domain. The outcomes of this deliverable, specifically the integration aspects of edge computing with TSN for scheduling will serve as an input to WP4 in developing the validation framework.

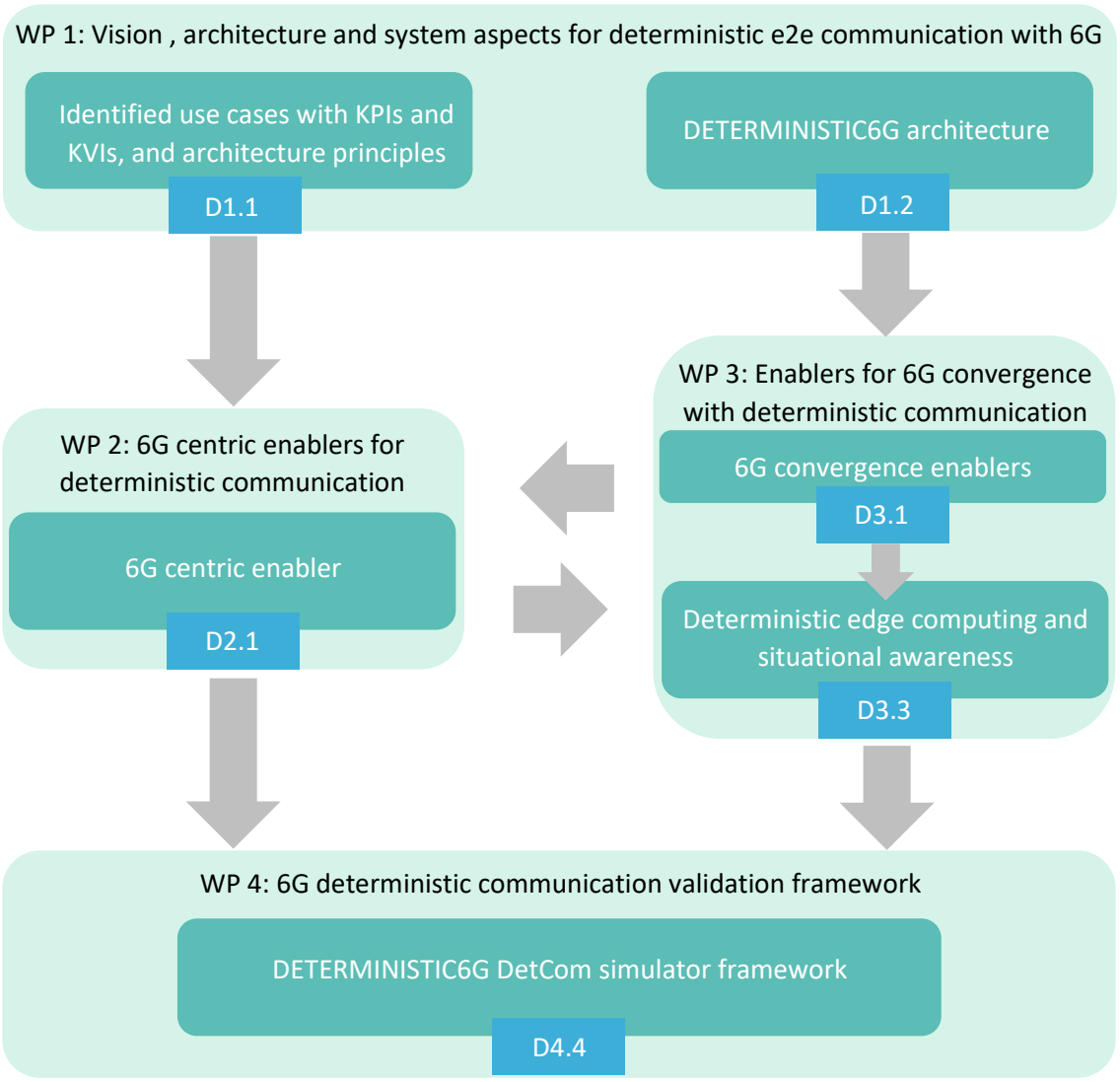

Figure 1.1 - Relationship of D3.3 to other work packages and deliverables

1.3 Objective of the document

This deliverable presents a first report on the architectural and functional integration of edge computing capabilities with time-sensitive communication to ensure E2E time-aware communication for applications that have computation and low-latency communication requirements. To further enrich E2E time-aware communication in 6G networks, initial concepts on deriving SA through 6G and operational digital twins are explored. The highlight is on understanding the values and goals of developing the two digital twins for predicting the future state of the CPS. Moreover, this deliverable also discusses parameters needed from the two digital twins for their interaction to derive SA and real-time decision-making. These interactions are then explained with an example of industrial use case. The impact of the developed concepts on data-driven latency prediction is explored with the potential of using digital twins to enhance SA and facilitate real-time decision-making in 6G environments.

1.4 Structure and scope of the document

The document is structured as follows. After the introduction in chapter 1, chapter 2 focuses on edge computing aspects. Section 2.1 provides a background on edge computing from the 3rd Generation Partnership Project (3GPP) and European Telecommunications Standards Institute (ETSI) perspective with the analysis of its available features supporting deterministic communication services. Section 2.2 explores edge computing application deployment flow and identifies categories of deterministic services that can be deployed at a 5G/6G MEC system. Section 2.3 discusses edge computing architectures as standalone, with or without TSN capabilities, and for controlling wired and wireless devices. Lastly, section 2.4 discusses integration aspects of the compute domain with TSN communication to support scheduled traffic. Chapter 3 of the document relates to SA concepts. Section 3.1 provides a background on the elements and benefits of the use of digital twins. Section 3.2 explains digital twinning from a cyber-physical systems perspective, including the operational digital twin and 6G digital twin. Section 3.3 defines concepts on how SA can be perceived using digital twins including the interactions between the factory and 6G digital twins. Section 3.4 further explores the concepts with an industrial use case example. Section 3.5 proposes parameters that can be helpful in creating resources and environmental awareness. Section 3.6 studies the impact of SA on data-driven latency predictions. Section 3.7 suggests measures that can be applied to 6G or operational systems with the help of SA knowledge and lastly section 3.8 analyses cost and challenges of having SA systems. Finally, chapter 4 concludes the deliverable with discussions of future work.

2 Edge computing

2.1 Background

Edge computing or Multi-access Edge Computing (MEC) is defined initially by the ETSI as a mechanism to reduce latency. ETSI proposed a MEC architecture [ETSI MEC Arch] to facilitate the integration of IT and cloud computing capabilities with mobile networks. Following the ETSI proposed MEC architecture, 3GPP has defined the network functions and interfaces required to integrate edge computing features into mobile networks.

2.1.1 ETSI MEC architecture

The ETSI architecture is structured in three layers as shown in Figure 2.1. The lowest layer consists of the network infrastructure that provides the basic connectivity i.e., local network and 3GPP mobile network, between the devices and the MEC platform. The middle layer provides the platform that is hosting the edge computing infrastructure, including the virtualization components required to run the edge applications and the management system that handles the available resources on the host where the MEC platform is deployed. The highest layer consists of a system-level management that provides overall visibility of the devices and edge computing platform.

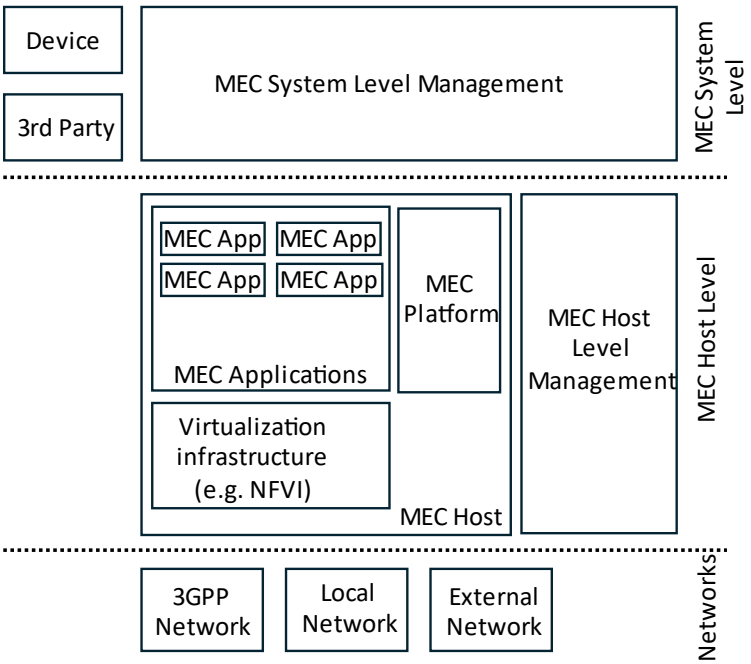

Figure 2.1 - ETSI MEC architecture

The ETSI MEC architecture is a major edge computing specification for the telecommunication services domain, providing a standard edge computing service framework for fixed and mobile communications and IT services at the edge of operator networks. It is an edge computing standard with specificities related to the access network (cellular 5G and beyond, fixed wireless, private networks, etc.), focusing on edge computing support for a variety of access networks and hosted IT services. MEC can now be seen as a key 5G technology enabler, and its role is expected to be more important for communication technologies beyond 5G, and especially for 6G technologies. Figure 2.2 shows a high-level view of the MEC architecture, and its operational environment.

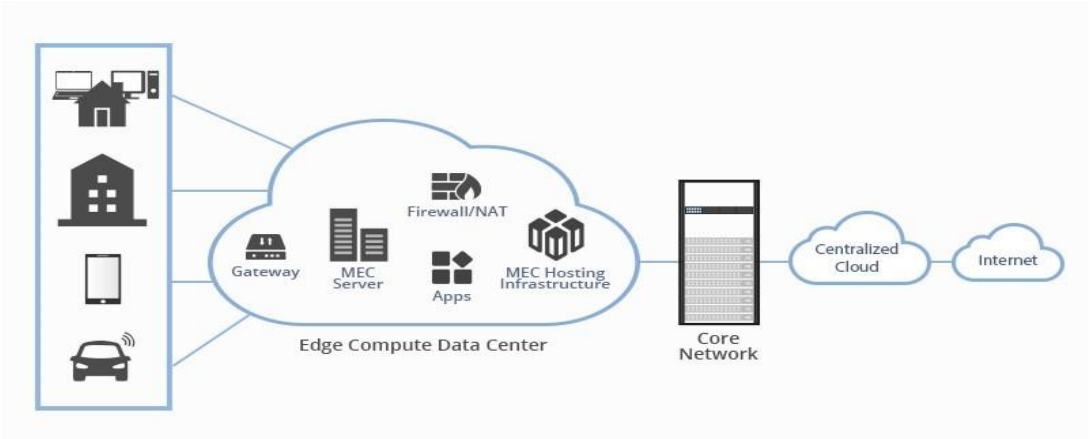

Figure 2.2 - High-level view of the MEC architecture

On the left of this figure, we have a representation of the various access technologies, including (but not limited to) fixed (wired and wireless) residential and professional (office) access, wireless mobile

personal and professional access, or wireless mobile vehicular access. In the center of the figure, we have the MEC service environment itself, which is exploited as a cloud computing service provided on an edge datacenter, consisting of an aggregation of MEC servers, a MEC hosting infrastructure and related applications, gateway, and firewall/Network Address Translation (NAT) devices, etc. On the right of the figure, we have the core network and a centralized cloud environment, and the Internet. This figure also highlights the view of the MEC infrastructure as ideally positioned between the access and the core network infrastructure.

MEC evolved from an earlier ETSI initiative called “Mobile Edge Computing”, that started with the production of an initial white paper in September 2014 and established its foundations. Now, most of the times MEC is used to refer to Multi-access Edge Computing, and sometimes to Multi-service Edge Computing. As an edge computing technology, MEC means proximity of the compute infrastructure for end-users vis-à-vis of physical distance, availability of compute resources at the periphery of public networks (typically placed between core and access networks), decentralized and centrifuged (i.e., resources available outside the center, at the periphery, as an alternative solution to centralization) general purpose resources in contrast to cloud datacenter and datalakes, as a solution to the “data gravity” problem as pointed out by Dave McCrory in 2010 [DGC2010], as an analogy to the physical way that objects with more mass naturally attract objects with less mass. In the context of cloud datacenters, a dataset is analogous to an object with a mass, and the size of that dataset is analogous to the mass of that object. The more a dataset is heavy (large), the more it tends to attract applications and other datasets towards the same datacenter, a driving force for data centralization. The main advantages of MEC can be classified into three categories, described as follows:

- Resource management: this includes bandwidth, processing, and storage resource management, with the possibility to save bandwidth (between the MEC and upwards) with local processing, and the possibility to scale computing and storage resources at datacenters and datalakes to include computing and storage resources available at the MEC infrastructure.
- User experience: latencies experienced by users, and the real-time constraints of services are sometimes better met when those services are placed on the MEC infrastructure, instead of datacenters, datalakes or other IT servers at the other end of an operator network. The user experience of interactive services, especially low-latency human-machine or machine-to-machine interactions are expected to be cut both in network and computing latencies with the use of a MEC infrastructure to host services.
- Dependable computing: reliable service delivery, the availability of services, security, and conformance with regards to data locality are among the dependable computing advantages that are expected from the use of a MEC service infrastructure. Reliability and availability are related to the possibility of network failures between the MEC infrastructure and upwards (i.e., between the MEC, the transport network and a centralized datacenter cloud), or the possibility of computing and storage service failure between the MEC infrastructure and upwards. Security and data locality are potentially reinforced by the use of MEC resources.

One of the main achievements of the ETSI concerning MEC is the specification of a reference system architecture for running MEC services. The reference architecture provides a description of a computing service framework, identifying a set of entities, service components and interfaces involved in a MEC system. This reference architecture introduces an application development and hosting paradigm, offering cloud computing capabilities and an IT service environment at the edge of the network to application developers and content providers. Figure 2.1 represents this service

environment, with the entities, services components, and interfaces. The bottom of Figure 2.1 shows the MEC host environment, consisting of managed MEC hosts and MEC platforms, the MEC platform providing a service environment running on a virtualization infrastructure, and being used by MEC applications. Interactions between the different entities and component are done via reference point interfaces. The top of Figure 2.1 shows the MEC system environment, with a particular emphasis on a multi-access edge orchestrator, an Operating System Support (OSS) and their reference points and links with external entities (devices, etc.). The application development framework provides Application Programming Interfaces (API) for a variety of MEC services, application support (for developing and hosting applications), service management, radio network information collection, location and UE entity management, device tracking, bandwidth management, fixed access and Wireless Local Area Network (WLAN) information, and V2X information services support. Typical MEC applications targeted by this application development framework are found in the following domains: automotive, e-health, gaming, industrial automation, smart cities, smart grids, video-streaming, VR/AR/XR, among others. When related to 5G and 6G networks, the ETSI MEC has investigated opportunities that can be expected from the integration of 5G and MEC, studying the organization and use of MEC as a 5G Application Function (AF), in addition to control plane interactions with 5G Core, functional split between MEC and 5GC and related APIs, or interactions of MEC and Radio Access Network (RAN). The integration of MEC and cellular systems beyond 5G is expected to continue for 6G systems, and there is room to shape the evolution of future MEC standards with 6G and the use of MEC in future 6G standards. Concerning DETERMINISTIC6G, the design and study of deterministic services for MEC could be an interesting topic to investigate.

2.1.2 3GPP MEC architecture integration

Following the architecture defined by ETSI for MEC, 3GPP has defined the network functions and interfaces to realize the proposed edge computing. The objective is to bring the data processing of applications closer to their physical location, either end user or data source. In addition to reducing delay, edge computing is also reducing traffic load between the mobile operator infrastructure. The edge computing allows to outsource the computing process to edge nodes closer to the user.

3GPP SA2 group introduced features to support edge computing defined in TS 23.548 [3GPP23-23548]. This specification outlines three connectivity models supported by the 5G core to enable edge computing. It defines various functionalities for traffic steering and User Plane Function (UPF) selection for realizing these different connectivity models.

The TS23.558 [3GPP23-23558], specified by the 3GPP SA6 group described Application Layer Architecture for enabling edge applications, is illustrated in Figure 2.3 and the related interfaces are briefly discussed in Table 2.1.

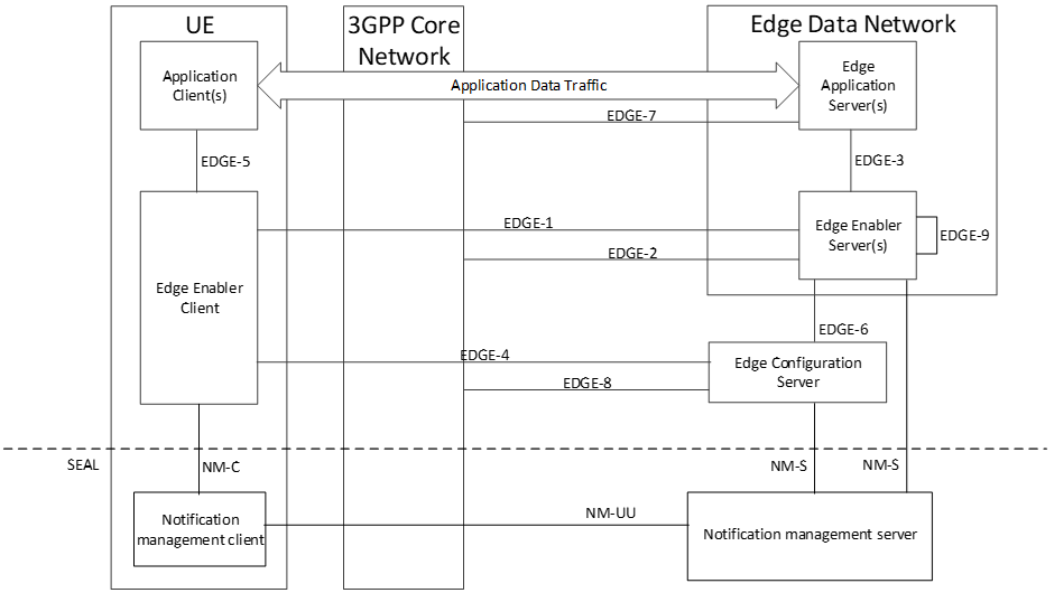

Figure 2.3 - 3GPP TS 23.558 defined edge computing architecture

Table 2.1 – 3GPP-defined interfaces in TS23.558 for enabling edge applications

Interface Name [3GPP TS23.558]	Entities involved	Functions
EDGE-1	EES-EEC	Retrieval and provisioning of Edge Application Server (EAS) configuration information. Discovery of EASs available in the Edge Data Network (EDN).
EDGE-2	EES-5GC	Retrieval of network capability information via Network Exposure Function (NEF).
EDGE-3	EES-EAS	Registration of EASs with availability information. Deregistration from Edge Enabler Server (EES). Providing network capability information (UE location information).
EDGE-4	EEC-ECS	Provisioning of Edge configuration Information to Edge Enabler Client (EEC).
EDGE-5	AC-EEC	Edge Domain Name Service (DNS) Client Provisioning/Interactions. Note: The DNS client provisioning will be provided by modifying the Session Management Function (SMF) configuration file. It will be static.
EDGE-6	EES-ECS	Registration to/Deregistration from Edge Configuration Server (ECS).
EDGE-7	EAS-5GC	Interactions with 5GC via Network Exposure Function (NEF).
EDGE-8	ECS-5GC	Interactions with 5GC via NEF.
EDGE-9	EESs in different EDN	Discovery of EAS relocation/AC relocation. EEC Context relocation.

This architecture makes the UE edge-aware, which means that all the devices include an Edge Enabler Client (EEC). The EEC communicates with the ECS, which provides the required configuration and supporting functions to set a data session up from the application client (hosted by the UE) to the EAS.

TS 23.548 and TS 23.558 3GPP define a set of Network Functions (NFs), listed in Table 2.2, to provide all the default functionality to register, authenticate and provide data sessions to the devices to support edge computing and interact with the rest of the NFs in the 5G System (5GS).

Table 2.2 - 3GPP-defined edge computing support Network Functions

Acronym	Definition	Functionality
AC	Application Context [Client]	It is not a function itself but a set of data about the Application Client that resides in the Edge Application Server.
ACT	Application Context Transfer	Refers to the transfer of the Application Context between the source Edge Application Server and the target Edge Application Server.
ACR	Application Context Relocation	Refers to the end-to-end service continuity procedure of relocating the Edge Application Server due to some reason (e.g., user plane change, AF request, etc.).
AF	Application Function	Control plane function that is interacting with NEF within 5G core network, to provide application services to the subscriber.
EAS	Edge Application Server	Application software resident in the edge performing the server function.
EASDF	Edge Application Server Discovery Function	A DNS resolver/server locally deployed by the 5GC operator within the local data network, responsible for resolving UE DNS queries into suitable EAS IP address(es).
ECS	Edge Configuration Server	Provides configurations to the EEC to connect with an EAS.
EDN	Edge Data Network	A local data network that supports the architecture for enabling edge applications.
EEC	Edge Enabler Client	Provides support functions, such as EAS discovery to the ACs in the UE (DNS client).
EES	Edge Enabler Server	Formerly responsible for enabling discovery of the EAS.

Moreover, the 3GPP specifications define a list of parameters to model the available MEC resources that allow to identify whether the application requirements will be met when deployed in the MEC

platform. Following are the parameters that 3GPP specifications have selected for exposing the capabilities that EAS supports, to check whether the EAS can meet the requirements of the application that will be installed in the EAS. This capability exposure functionality is utilized by the functional entities (i.e. EES, EAS and ECS) depicted in the architecture for enabling the edge application. A message flow describing the message exchange between EAS and ECS to identify common capabilities is shown in section 2.2. The performance results of applications running on the public cloud or MEC platform are presented in that section.

Table 2.3 summarizes the EAS profile KPI parameters specified in the 3GPP specifications (Table 8.2.5-1 TS 23 558 Rel 17)

Table 2.3 - MEC KPI parameters

KPI Parameter	KPI Description
Maximum Request Rate	Maximum requests per second that server should handle
Maximum Response Time	Maximum processing delay on incoming requests
Availability	Indicates whether edge computing resources are available
Available compute	Computing resources available
Available Memory	Memory available for the application on MEC
Available Storage	Storage memory available for the application on MEC
Connection Bandwidth	Bandwidth available for the application on MEC

These KPI values supported by the EAS are used to deploy the discovery procedures that enable entities in an edge deployment to obtain information about EAS and their available services. On the client side, the EEC can obtain information about available EASs of interest. The discovery of the EASs is done by the EEC and is based on matching EAS discovery filters provided in the request.

2.2 3GPP Edge computing deployment

In this section an off-the-shelf 5G network is used to validate the usage of 3GPP MEC platform to deliver required performance for deterministic applications. Initial prototype of MEC Rel 17 platform is implemented and next section describes the flow to discover the optimal location for deploying deterministic application using the network functions defined in the 3GPP MEC architecture.

2.2.1 3GPP Edge computing application deployment flow

A practical development of a 3GPP edge computing architecture and its integration with a commercial base station has been completed to validate the support of MEC to the deployment of TSN applications. The TSN application will discover the best matching MEC platform.

The setup used for the validation is shown in Figure 2.4 below and consists of a 3GPP edge computing platform that runs an off-the-shelf iperf3 server and an Android mobile device running an iperf3 client. In the setup we use iperf3 to measure throughput and ping to measure the delay between UE and the iperf3 server either in the edge computing server or in public cloud.

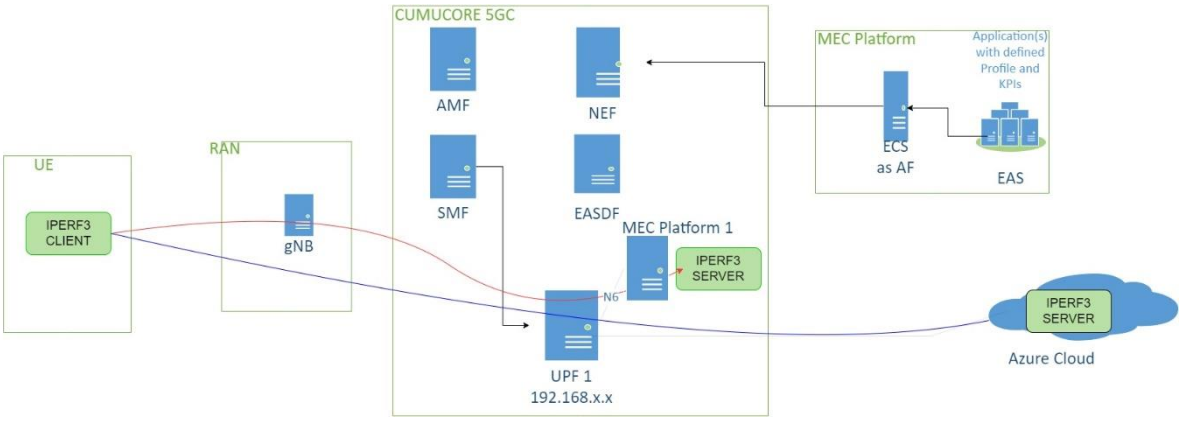

Figure 2.4 - Setup for validating MEC support of TSN applications

The edge application location is selected based on Uplink (UP) parameters combined with profile and KPIs of EAS and Application (AC) e.g. TSN talker or listener.

1. The EAS registers to ECS (considered as an AF) with its profile information. EAS's profile also includes KPIs supported by the application.
2. ECS deploys EAS's registration data in an internal mongodb database to be accessed by other NFs.
3. ECS then requests the list of active UEs from the CN, i.e., subscribers showing a CONNECTED status in the Graphical User Interface (GUI). Since multiple slices can be active, the request is done per Group ID to identify UEs connected to different slices.
    a. Using the list of active UEs from the previous step, the ECS requests the aggregated delay information of each UE from the Network Data Analytic Function (NWDAF). The NWDAF continuously collects information about all NFs, their load and performance.
    b. The ECS computes the average delay per slice.
    c. ECS also requests the slice capacity from NWDAF.
    d. In this context, the ECS is considered as an internal AF. Therefore, it sends an NF_discovery request to Network Repository Function (NRF) to learn about active UPFs. The response includes IP addresses of UPFs and their associated slices.
4. The ECS then selects the slice with the lowest delay and high enough capacity to support MEC traffic and updates the EAS information with the corresponding UP path (IP address).
5. To change a subscriber's traffic routing, ECS requests the Policy Control Function (PCF) for authorization providing traffic routing information.
6. The PCF instructs the Session Management Function (SMF) for a Packet Data Unit (PDU) session modification with the IP address of selected UPF.
7. PDU session modification messages are sent between the UE, the Access and Mobility Management Function (AMF), and SMF. The SMF may also request EAS's deployment information.
8. UE's PDU session is updated with new traffic routing information to establish data traffic towards the edge application.

After the discovery of the EAS to deploy the application a set of performance testing was completed. Figure 2.5 shows the differences in delay i.e. Round Trip Time (RTT) between the UE and the cloud server on the left and between the UE and the edge server on the right. Comparing the results we see a long tail distribution of the delay when accessing the iperf3 server running on the public cloud as shown in left of Figure 2.5 compared to the delay distribution on right of Figure 2.5. The results show that the usage of MEC platform can reduce the delay variation between 8-25ms compared to more distributed delay variation between 4-200ms without MEC platform.

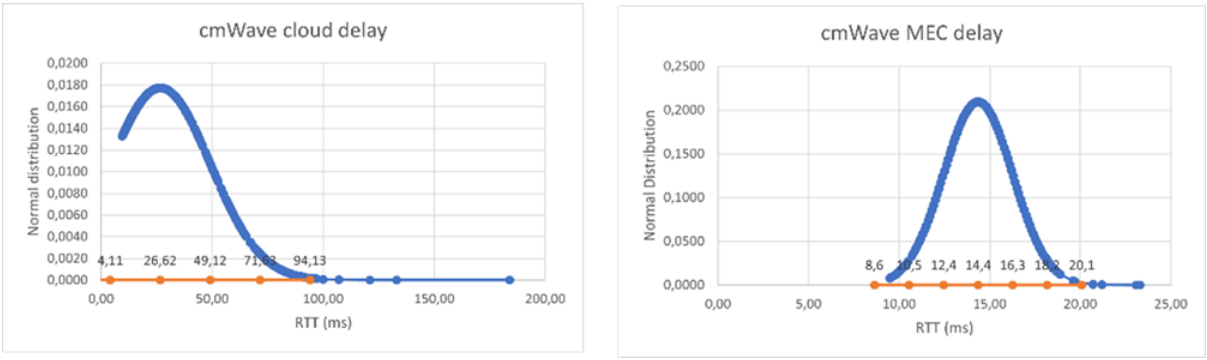

Figure 2.5 - Rout Trip (RTT) measurements between UE and edge server (on the right) or cloud server (on the left).

2.2.2 Deterministic services in MEC

Considering the geographical distribution of end devices and users expected in many future MEC applications, a classical scenario for creating a MEC-capable network in the future would be to realize an interconnected network of MEC nodes, a MEC node being an edge computing node providing MEC services. Each of the interconnected MEC nodes looks like the Edge Compute Datacenter cloud present in the high-level view of the MEC architecture in Figure 2.2, and it includes a set of interconnected MEC hosts (i.e., servers hosting MEC services), used to run MEC servers, MEC hosting infrastructure and applications, and other software. MEC hosts and other devices in a MEC node are linked together with a (local or metro) network internal to each MEC node to form a distributed computing node at a small (local or metro) scale, and MEC nodes in a MEC-enabled 5G or 6G network are linked together to form a (metro or large scale) distributed MEC infrastructure. A typical MEC installation in such a scenario can be seen as a hierarchical distributed system, with at-least two levels of hierarchy, i.e., a (large or metro scale) distributed system in which each node is also a (local or metro scale) distributed system. This MEC system is also an open and programmable environment, with APIs and the possibility to dynamically support the execution of newly created applications, or to stop/delete running applications. We are therefore faced with open, hierarchical distributed MEC systems that require dependable, time-critical 5G/6G communications, dependable, real-time computing tasks for many of the hosted IT services, and potentially resource demanding (compute/storage-intensive) tasks for some of the applications. The use of an open environment, with fluctuating capacity demands of resources provided as a service, has the potential to make it even more challenging to provide deterministic guarantees, compared to embedded/close environments.

Deterministic service requirements for the MEC system are inherent to some of its infrastructure services and hosted applications, and many deterministic service characteristics can be derived from the specifications of the target applications and the main objective of edge computing and the MEC. Time awareness, real-time computing, and low-latency for example, are among the main motivations for using a MEC system to host real-time or latency-constrained applications closer to the end user. Reliable services, and service availability (both networks and computing) are also among the major MEC service (application and infrastructure) requirements, motivating the use of a MEC system. Others are system management, application interfacing with APIs, etc. Each of these requirements and service characteristics are present in a MEC (application and/or infrastructure) service or other, and they are included in the capabilities targeted by a MEC system. However, how these deterministic services capabilities can be guaranteed with a MEC system remains for the most part an open question.

Three categories of deterministic services can be identified for the open and hierarchical distributed MEC system that can be deployed for a 5G/6G MEC system:

- Dependable, time-critical communications: the local and metro network connections internal to a MEC node (between MEC hosts and other devices within a MEC node), and the metro and long-distance network connections between MEC nodes, need to provide dependable time-critical communications in general between the software entities and devices linked by these networks. Dependable time-critical communications are required in many of the hosted MEC applications, in the MEC infrastructure services, and in cyber-physical communications between the MEC system and external devices.
- Dependable computing: dependable computing is required, mostly real-time computing capabilities, relying primarily on the use of important capacity resulting from aggregated computing resources, storage, and specialized hardware. But the use of large capability is usually not enough. Additional mechanisms need to be implemented in the MEC infrastructure to ensure real-time and dependable computing. Section 2.4.1 discusses major enablers and capabilities of a cloud deployment for supporting time-critical applications.
- Dependable distributed computing: some of the MEC applications and infrastructure services may be designed to run on several MEC hosts within a MEC node, and/or in several MEC nodes. The correct execution of some of these services sometimes requires dependable, real-time communications and computing services.
  In some cases, dependable computing and communications may need to be used together/jointly, and in a coordinated way. As an illustrative example of the integration and coordinated operation of the compute and communication domains, section 2.4.3 showcases a traffic handling framework in the compute domain for the seamless support of the IEEE 802.1Qbv scheduled traffic concept.

2.3 Architecture integration of 3GPP edge computing and TSN support

The edge computing facilitates the deployment of use cases that have time-critical requirements. Typical use cases include Extended Reality (XR), Occupational Exoskeletons (OE) with function offloading to the edge, and various industry automation use cases, where the controller application is deployed on an edge computing host. If TSN functionalities are intended to be utilized in these use cases, then the integration of TSN and edge computing/MEC is necessary, which requires some architecture design that enables the handling of time synchronization and packet scheduling.

The integration of TSN and edge computing/MEC requires some conceptual analysis. In the case that the TSN Talker/Listener (e.g., cloudified application instance) is moved to the edge computing domain connected through a mobile network, then we require mapping the 3GPP defined service endpoints into TSN control interfaces.

Two basic deployment scenarios have been identified in Figure 2.6 and Figure 2.7. In the first scenario shown in Figure 2.6, the edge computing service is enabled by the 5G system and connected to the 5G/6G virtual TSN bridge. In the second scenario, the edge computing service is deployed by using a standalone, dedicated infrastructure to host the TSN Talkers/Listeners as shown in Figure 2.7.

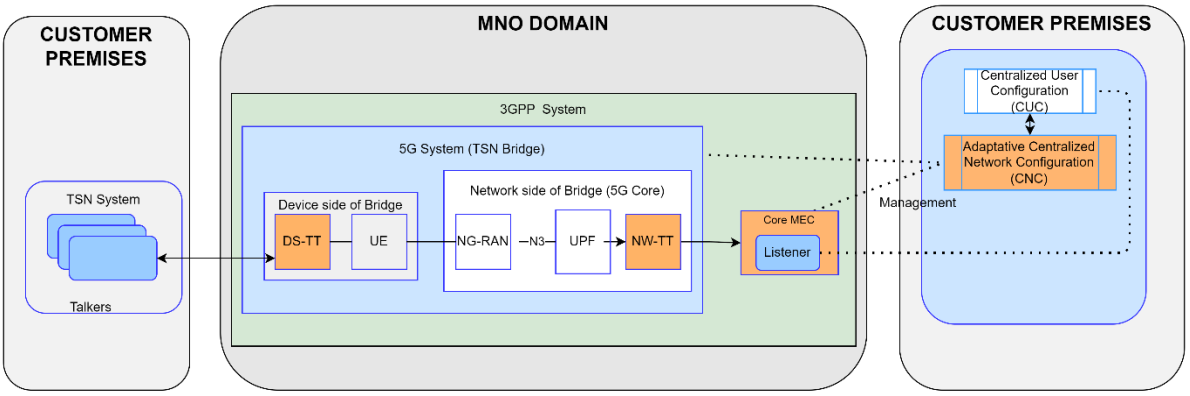

Figure 2.6 - Edge computing platform integrated into 3GPP system

In this first scenario (Figure 2.6), the edge computing is tightly integrated with the 3GPP system and leverages the distributed cloud infrastructure resource principle. This deployment uses the 3GPP SA6 specified edge computing support features (details are outlined in [3GPP23-23558]).

The 3GPP architecture provides the flexibility to support multiple deployments, applicable to our scenario. In the case of an Enterprise deployment, for instance, where the 3GPP domain may act as a Standalone Non-Public Network (SNPN), a shared computing infrastructure could host the 3GPP (cloudified NFs) and edge services.

Another deployment is such that the edge computing is enabled by the MNO, leveraging that the geographical density of RAN sites enables the deployment of edge resources within approximately 10km range from the end device. In this case the MEC platform will be using MNO premises acting as edge computing service provider to their industrial customers or 3rd party edge service providers.

Applicable mainly to the second deployment option, where the edge computing platform is integrated with the MNO network, there are two main infrastructure integration options to consider:

- The first option involves having separate dedicated infrastructure for both the MNO network and the edge cloud offered as a service. This setup ensures physical isolation between the network components running in the datacenter and the user components. However, this approach requires double cloud management overhead, as each infrastructure needs to be managed independently.

- Alternatively, the second integration option involves a single cloud operated by the MNO, hosting both its own network components and user components. While this reduces management overhead and enables load optimization, it necessitates the implementation of strict isolation solutions, described in Section 2.4.1.

In both deployments, the operator edge cloud potentially offers a low-level service, such as Infrastructure as a Service (IaaS) or bare metal hosting. However, this is unlikely due to the tight integration required with the operator infrastructure and network. Instead, a higher-level edge cloud service, such as Platform as a Service (PaaS) hosting containers, is envisioned. This higher-level service can also be prepared to support time-sensitive workloads.

The cloudification of various application components (i.e., virtualize the application and running an instance in a container) and offloading them to the edge compute domain may result in a deployment scenario where some TSN Talkers are connected to the edge applications via the 3GPP system, while others are connected via a legacy, wired TSN domain. This requires an integrated connectivity solution, which enables both wired and wireless devices to be served from a single edge computing premise.

One possible realization could be having the migration point of the traffic coming from the wired and wireless devices provided in the edge computing domain as a platform or infrastructure functionality. For the wireless devices, the existing 3GPP TSN support function can be leveraged, while for handling the wired devices, a TSN Gateway (TSN-GW) functionality is ensured by the edge computing platform. The wired devices can then use a legacy wired TSN infrastructure, and a certain bridge of this infrastructure can be connected to the TSN-GW of the edge computing platform. In an Enterprise deployment, the clear advantage of this solution is the easy implementation since both the edge platform and the edge infrastructure can be easily configured by the owner of the edge deployment to realize the required TSN-GW functionality. In the operator-enabled edge service case, however, it is a bit more challenging, since the PaaS or IaaS offered by the operator must contain (or must be capable of hosting) the required TSN-GW features for handling the wired devices.

Another option could be to leverage the support of non-3GPP capable devices (e.g., legacy industrial devices without UE functionality) as specified by 3GPP TS 23.501 [3GPP18-23501] and TS 23.316 [3GPP18-23316]. In this case, the convergence point for the communication paths of the wired and wireless devices is the 3GPP system, which handles the wired devices as non-3GPP capable devices. To this end, according to 3GPP TS 23.316, a Wireline Access Gateway Function (W-AGF) must be deployed in the 3GPP system, which ensures the connectivity between the wired devices and the UPF. However, it is not obvious how this solution can be integrated with the 3GPP TSN support architecture, so further study is required in this area.

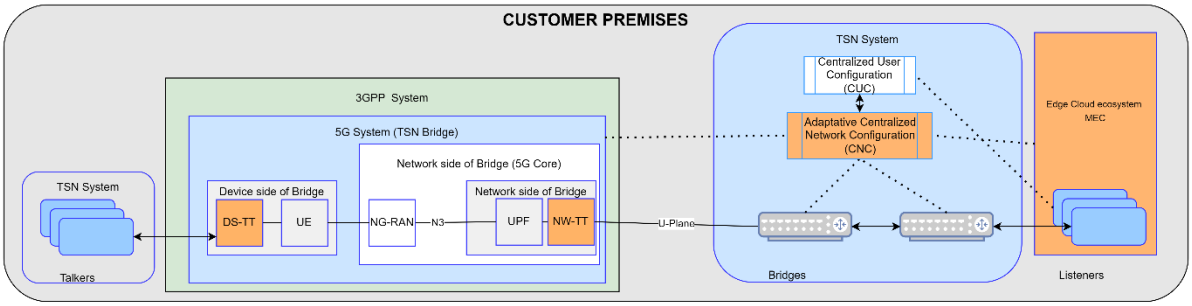

Figure 2.7 - Standalone edge computing deployment

In the second scenario (see Figure 2.7), the edge computing platform is isolated from the mobile infrastructure, and a standalone datacenter infrastructure is deployed to host the cloudified applications (aka TSN Talkers/Listeners). This type of edge deployment scenario enables lots of flexibility, such as distributed edge infrastructure for ensuring high reliability, customized traffic handling, and networking solutions in the virtualized domain tailored to tight interworking with TSN domain.

The simplest deployment option for this scenario is to have a single, standalone datacenter, deployed at the enterprise premise. To ensure higher redundancy at the infrastructure level, multiple datacenters can be deployed using separate infrastructure resources (e.g., in a factory deployment, one datacenter can be deployed in different buildings). To leverage the robustness of this deployment, a federated management of the datacenters is required, enabling the cloud manager (e.g., Kubernetes) to handle the multiple datacenters in an integrated way. From the customer's perspective, the multiple Kubernetes clusters are seen as a single one. This enables the orchestration of multiple application instances while considering redundant infrastructure resources seamlessly.

This deployment enables flexibility in the usage of the cloud execution environment, since even in the case of single, standalone datacenter deployment, multiple execution environments, such as virtual machines and containers, can be deployed on the same hardware infrastructure. Furthermore, an important characteristic of this deployment is that the owner of the edge has full control over the entire cloud stack, providing significant freedom to implement all the necessary features to support applications with real-time requirements and to ensure seamless integration with the TSN domain in a fully customized way. Illustratively, the edge owner can select the type of virtualization platform, operating system of the host (e.g., real-time Linux), configure the usage of Graphic Processing Units (GPUs), various hardware accelerations and hardware offloads, set up low-level Central Processing Unit (CPU) scheduling and resource allocation, as well as configure host networking features for TSN-aware traffic handling. Furthermore, if needed, customized Container Network Interface (CNI) plugins can be installed, which contain capabilities tailored to the application deployments and TSN communication. On the other hand, to properly utilize the above-mentioned capabilities, a deep, cloud domain-specific knowledge is required. Additionally, the edge owner is responsible for maintaining the deployment, including the infrastructure, software, virtualization layers, as well as the control and management planes and the required security features. Hence, in some cases, a 3rd party integrator steps in to take care of the deployment-specific maintenance, while the customer handles only the application deployment and life-cycle management aspects.

The seamless support of wired and wireless devices can be easily provided without placing any requirement on the edge computing domain. As can be seen in Figure 2.7, legacy TSN bridge(s) can be deployed between the virtual TSN bridge of the 3GPP system and the edge computing domain. This TSN segment corresponds to the TSN backbone in section 5.2.1.2 of DETERMINISTIC6G D1.2 [DET24-D12] and acts as the integration point for the traffic coming from the wireless and wired devices.

2.3.1 Aspects of control plane design and integration with edge computing

The purpose of this section is to provide a high-level introduction to a view of a 6G network controller as an edge computing service, based on Software Define Network (SDN). The controller is intended to control the packet forwarding engine (the data plane) of a 6G network under dependable, time-critical communications constraints. It is intended to be designed and implemented as an edge computing service, providing an edge control service for SDN applications, and providing a form of integration of the 6G network control plane with edge computing systems. The integration of control plane with edge computing requires first to analyze the requirements of dependable, time-critical networks and check current state of the art for different options to implement those requirements into mobile networks.

Control requirements

The TSN control plane for example focuses on time-sensitiveness. The intentions here with 6G network controller designed and implemented as an edge computing service based on SDN are to go beyond the current TSN time-sensitiveness, with two possible directions:

- Real-time and dependability: in some cases, the controller itself needs to be wireless-aware and to react in real-time, and to guarantee a set of dependability constraints for considering the 6G wireless domain.
- Network programmability: in some cases, other programmable network features may be useful or even required, especially those related to the computing model, the system and software architecture, mechanism, algorithms and protocols, or APIs. The use of SDN controllers running on edge computing resources in particular, provides an adequate framework for programmable 6G network controllers.

The controller should support the programmability aspects of dependable 6G communications. It should be possible to program a control plane of the 6G packet forwarding engine (or data plane) to change the behaviour of a 6G network controller (a controller that controls the 6G packet forwarding engine – or data plane), and to adapt dynamically to dependable, time-critical communication requirements. As a particular consequence, it should be possible to use both centralized and decentralized controllers, or mixed controllers. It should support the programmability aspects of dependable 6G communications. Dependable and time critical communications are studied, and in our context, enabling control plane programmability means we need not only to bring time-sensitive or real-time capabilities to SDN controllers (in some cases the SDN controllers themselves need to be real-time), but also to bring dependable communication capabilities. The SDN controllers for dependable and time-critical communications should be able to integrate time-sensitive constructs and provide real-time control services (timeliness), in addition to dependability and network programmability. The SDN controller should also be able to integrate dependability concepts ideally related to availability, safety, reliability, recoverability, integrity and maintainability, in the long term.

Moreover, the SDN controller should also be able to identify the different threats to the network and communication services (intentional or unintentional faults, errors, failures, …) to provide services with mechanisms to implement one or many of the standard means to ensure dependable services, namely fault-prevention, fault-tolerance, fault-removal, or fault-forecasting. Deterministic SDN controllers in the context of DETERMINISTIC6G means SDN controllers that are designed and implemented as edge computing services, that are capable to accommodate real-time constraints at the deferent levels of the SDN environment, and that are capable to guarantee the various dependability concepts and attributes in the ideal case.

State of the art of control plane design

In the fully centralized TSN configuration model, the SDN Controller is called Centralized Network Control (CNC), which centrally controls and configures network elements. As an SDN Controller, a CNC is in charge with absolute authority for the control of a configuration domain. Therefore, the CNC controls the network to support all traffic flows, i.e., traffic flows with requirements of dependable, time-critical communication services as well as best effort flows. An existing SDN Controller can be extended to make it a CNC, i.e., to add capabilities for the control of TSN devices. Figure 2.8 shows an example of such work, dealing with the extension of SDN controllers, e.g., extensions to OpenFlow to be able to control real-time Ethernet data plane, i.e., TSN devices.

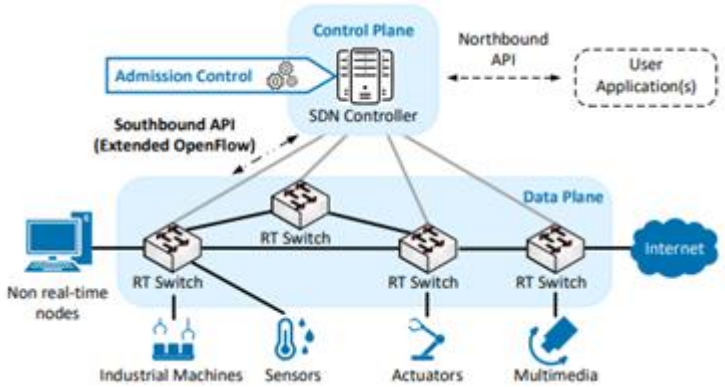

Figure 2.8 - Example time sensitive SDN proposal (from [MPA19])

These proposals for extensions to SDN controllers are applicable to several domains and types of networks, including vehicular and industrial networks, and they need to be developed for (or adapted to) public 6G networks. They are mentioned here as example (state-of-art) work towards the direction of dependable time critical controllers building on TSN and SDN concepts. These are first steps, we need to go beyond these example proposals with real-time controllers, dependability and network programmability concepts.

Future directions on control plane design

The IETF DetNet Controller Plane Framework (CPF) is a framework to support and manage the control aspect of Deterministic Networking. The framework is expected to become an RFC soon, but the further detailed design could benefit from deterministic SDN controller considerations in drawing its requirements. It will also need to be “populated” with actual design proposals and feedback from

experiments for future refinements of its specifications. IETF DetNet CPF has identified a set of requirements linked to the use of edge computing capabilities. We believe it is possible to find similarities with the requirements of Centralized User Configuration (CUC)/CNC controllers in the TSN architectures, and/or the evolution of SDN controllers to consider time-sensitive constraints. This could lead to a potentially interesting basis for implementing deterministic controllers with edge computing resources, and interesting feedback for future DetNet CPF, TSN, or SDN standards.

The main focus of future work in this area, in the context of DETERMINISTIC6G, could be to design and study a control plane based on deterministic SDN controllers. Furthermore, we foresee the work on the deterministic SDN controllers within DETERMINISTIC6G as a potential future design proposal that could be considered as an input for detailed design of the IETF CPF.

In summary of this section, the main intention is to investigate the design and study of 6G SDN controllers for dependable time-critical communications: not only time considerations, but also dependability and programmability concepts inherent to SDN. The main motivations for that are:

- Evolution of CNC; particularly to make the CNC wireless-aware and wireless-friendly.
- In some cases, current TSN is not enough. Other programmable features are required, especially those related to the computing model, the system and software architecture, algorithms and protocols, APIs.

2.4 Integration aspects of compute domain with TSN communication to seamless support of IEEE 802.1Qbv scheduled traffic

Cloudification of the applications enables to utilize the generic advantages of cloud systems, such as access to huge amounts of resources (compute, memory, storage), dynamic and elastic resource handling and scaling, flexible, adaptive application deployment management, robustness, etc.

Additionally, the edge computing paradigm provides reduced latency between the client and the server application, enabling the cloudification of applications with low-latency requirements (referring to Figure 4 in [DET23-D21]). However, although edge computing can significantly reduce the E2E latency compared to cloud computing, it cannot circumvent the problem of latency variation caused by the uncertainties in the compute domain[1]. Consequently, guarantees for time-critical services cannot only be provided by the usage of edge computing paradigm, in itself. If the support features of the compute deployment for time-awareness are not used for applications on an edge premise, seamless integration with a TSN-based time aware communication system (e.g., using IEEE 802.1Qbv) cannot be implemented properly.

2.4.1 Enablers and capabilities for supporting cloudified, time-critical applications

Two main **challenges** can be identified for cloud execution when trying to address deterministic communication. The first one is about guaranteeing time-bounded execution of the time critical applications, summarized in this section. The second one relates to the timely and coordinated delivery of messages of the time critical applications, more in the focus of our projects, discussed in section 2.4.2.

[1] For an illustrative example, please see Figure 2.5, which shows that an edge deployment can significantly decrease the latency, but the latency variation may still remain significant.

To address the first challenge, the goal is to mitigate the harmful uncertainties of the compute components, by leveraging the resource allocation, task scheduling, isolation, etc., toolset of cloud computing in a coordinated way, i.e., using a real-time cloud (optionally with optimized hardware like GPUs).

In our case the goal of the cloud environment is to provide **similar behavior to native** execution for the real-time application. This requires spatial as well as temporal separation of the applications.

Traditionally the virtualization environments aim to provide **spatial isolation** or protection, meaning that private resources like compute, memory, and storage of an application cannot be accessed by other applications. The typical cloud virtualization environments we consider are virtual machines and containers. Serverless approaches, also known as Functions as a Service (FaaS), are out of scope for us, as they require a different design principle compared to traditional time-sensitive applications.

Kubernetes (often abbreviated as K8s) is an open-source platform for automating the deployment, scaling, and management of containerized applications. It provides features for orchestrating containers, scaling applications, and ensuring high availability and reliability. Kubernetes is widely used for managing containerized workloads in diverse environments, including cloud data centers, on-premise data centers, and hybrid setups. Container isolation is achieved through the use of container runtimes, such as Docker or Containers, which provide a layer of isolation between individual containers running on the same host. Kubernetes leverages the capabilities of these container runtimes to ensure that each container operates independently from others, with its own filesystem, networking, and process space. This isolation helps to prevent interference and ensures that applications running within containers remain isolated and secure from one another. In Docker, **CPU scheduling** refers to the allocation and management of CPU resources for containerized applications running on a Docker host. Docker provides CPU scheduling capabilities through the use of the CFS (Completely Fair Scheduler) in the Linux kernel, which manages the allocation of CPU time among competing tasks, including Docker containers. With Docker, one can specify CPU shares and CPU quotas for individual containers, which allows to control how CPU resources are distributed among containers on the host. This helps in prioritizing and allocating CPU resources based on the needs of specific containers, ensuring fair distribution of CPU time and preventing any single container from monopolizing the CPU resources.

Regarding virtual machines, **OpenStack** is an open-source cloud computing platform that provides a range of services for building and managing public and private clouds. It includes components for computing, storage, networking, and dashboard management. It supports various virtualization technologies, including KVM (Kernel-based Virtual Machine), Xen, VMware, and Hyper-V. KVM is one of the most commonly used virtualization options in OpenStack due to its open-source nature and integration with the Linux kernel. Virtual machines are created and managed using the Compute service, also known as Nova. Nova allows users to launch and manage instances of virtual machines on demand, providing the necessary computing resources in a flexible and scalable manner. Users can specify the characteristics of the virtual machines they need, such as CPU, memory, and storage. In OpenStack, scheduling refers to the process of determining where and how to place workloads, such as virtual machines or containers, across the available compute resources within the cloud environment. We rather refer to this process as **orchestration**, to differentiate it from the lower level scheduling of processes to CPUs. The orchestration decision takes into account factors such as resource availability, workload requirements, and policy constraints too.

Temporal isolation means that timing related constraints of an application do not depend on or interact with other applications. This is a key issue for real-time application in a cloud environment, as a cloud environment usually makes use of resource (e.g. CPU) sharing. To provide temporal isolation, either significant overprovisioning is needed which goes against statistical multiplexing benefit of the cloud, or clever techniques are required. By default, both K8S and OpenStack provide limited temporal isolation, meaning that a given fraction of CPU (e.g. 25%) can be allocated to an application, but the application is not guaranteed to get execution time in a constrained time interval (e.g. the 25% processor time is valid on average in a minute timescale, but the app is not guaranteed to run in e.g. every 2 ms).

The **native environment** of a time critical application can be (i) a dedicated special hardware, in this case the application is typically low-level code close to hardware, or (ii) a more abstract application running on a Real-Time Operating System (RTOS).

A RTOS is a specialized type of operating system designed to manage and execute tasks with precise timing constraints. It is used in applications where timely and predictable responses are critical, such as in embedded systems, industrial automation, robotics, and control systems. RTOSs are characterized by their ability to guarantee a maximum response time to events, ensuring that critical tasks are executed within specific time intervals. They are classified into two main categories:

- Hard Real-Time Operating Systems: in hard real-time systems, tasks must be completed within strict and deterministic time constraints. Missing a deadline in a hard real-time system is considered a system failure and can have serious consequences.
- Soft Real-Time Operating Systems: soft real-time systems have less strict timing requirements, allowing some flexibility in meeting deadlines. While meeting deadlines is important in soft real-time systems, occasional misses may be tolerable without causing system failure.

RTOSs (E.g. real-time Linux, like Real-time Ubuntu) are optimized for low-latency, rapid context switching, and deterministic task scheduling, making them suitable for applications that require precise timing and control. CPU scheduling plays a critical role in ensuring that time-sensitive tasks are executed within their specified deadlines. RTOSs typically employ deterministic and priority-based **scheduling algorithms** to manage task execution. The primary CPU scheduling algorithms [MOH09] include:

- Rate-Monotonic Scheduling (RMS): RMS is a priority-driven scheduling algorithm that assigns priorities to tasks based on their periods. Shorter-period tasks are assigned higher priorities, allowing for efficient management of periodic real-time tasks.
- Earliest Deadline First (EDF): EDF is another priority-driven scheduling algorithm that assigns priorities based on the time remaining until each task's deadline. This approach ensures that the task with the earliest deadline is given the highest priority, facilitating timely execution of time-critical tasks.
- Fixed-Priority Preemptive Scheduling: this approach assigns fixed priorities to tasks, and the scheduler ensures that higher-priority tasks preempt lower-priority tasks. It provides determinism and predictability in meeting timing constraints for real-time tasks.

A Linux CPU scheduler based on the EDF and Constant Bandwidth Server (CBS) algorithms, called SCHED_DEADLINE is considered suitable for real-time applications.

Applying **real-time capabilities to virtualization** introduces challenges [GCL14] due to the inherent complexities of virtualized environments. Achieving deterministic behavior and low-latency performance for time-critical tasks within virtualized systems requires addressing several key hurdles like:

- Hypervisor Overhead: virtualization introduces overhead, as the hypervisor must manage and schedule resources for multiple Virtual Machines (VMs) running on a shared physical host. Mitigating this overhead is essential to ensure that time-critical tasks within VMs can meet their timing constraints.
- Resource Contention: in a virtualized environment, multiple VMs compete for access to physical resources, such as CPU, memory, and I/O. Contention for resources can impact the predictability and determinism of time-critical operations.
- Latency Variability: virtualization can introduce variability in task execution times due to factors such as virtual machine migration, shared resource contention, and scheduling overhead. This variability can hinder the predictable behavior required for real-time applications.
- Real-Time Scheduling: ensuring that real-time scheduling and prioritization mechanisms are effectively implemented within the virtualization stack is crucial. This involves providing mechanisms to prioritize time-critical VMs and enforce deterministic scheduling of tasks.
- I/O and Networking: achieving low-latency I/O and networking performance within virtualized environments presents challenges due to the shared nature of hardware resources and the potential for interference from other VMs.

Addressing these challenges requires careful design and configuration of the virtualization infrastructure, including the selection of real-time capable hypervisors, the allocation of dedicated resources to critical VMs, and the use of real-time extensions and scheduling policies within the virtualization stack.

2.4.2 Issues with seamless support of TSN 802.1Qbv-based communication for cloudified applications

The IEEE 802.1Qbv, Enhancements to Traffic Scheduling: Time-Aware Shaper (TAS), is designed to enhance Ethernet networks for time-critical applications with cyclic communication. Its essence is to split the communication into fixed-length, repeating time cycles. Within these cycles, time slices are configured and assigned to certain Ethernet priorities. The careful design of these slices across all devices (end stations and bridges) in the network allows the time-sensitive traffic to pass through a network node-by-node at specific times, according to the scheduling plan. This results in the minimization of the jitter and ensures predictable latency for the applications.

Considering the use cases investigated in the project (e.g., Occupational Exoskeletons and Adaptive Manufacturing, described in detail in Deliverable D1.1 [DET23-D11]), the usage of 802.1Qbv allows the guaranteed transmission of control messages between the control application and the device, ensuring the timely execution of mission-critical tasks.

Illustratively, in Smart Manufacturing, in the edge cloud-enabled control of Automated Guided Vehicles (AGVs) the usage of IEEE 802.1Qbv ensures the synchronized (control) data forwarding towards the AGV devices on the shopfloor from the AGV Swarm coordination application deployed in the edge domain, which provides the operation of the AGVs as a coordinated swarm.

In the Occupational Exoskeletons use case, particularly in the long-term offload scenario, the low-level control functions are offloaded to the edge domain, meaning that there is a 1ms budget for the control loop, considering the compute and network domains. In this case, on one hand, the usage of IEEE 802.1Qbv ensures predictable timing in the communication network domain, on the other hand, this predictability in the network domain provides more time budget for the compute domain, which is crucial considering the very tight 1ms time budget requirement.

Based on section 2.4.1, it can be concluded that it is not enough to a) leverage the real-time enabler features of the compute domain and b) properly configure the legacy TSN and/or 6G DetCom domains, but the still existing stochastic uncertainties caused by the characteristics of the cloud environment and deployment must be considered by the CNC in the preparation of the 802.1Qbv scheduling plan.

The main source of these uncertainties comes from the different design paradigms and capabilities of the specific hardware-based and a cloudified application. As illustrated in Figure 2.9, in the case of a hardware-based controller (left hand side of the figure), the controller software and hardware of an end-host are designed to guarantee deterministic performance for the application, and there is a tight relationship between the application function and the Network Interface Controller (NIC) of the end-host. This guarantees, for instance, in the case of 802.1Qbv scheduled traffic, that the application timing and NIC traffic handling configurations, calculated by the CUC and CNC, can be fully and precisely enforced.

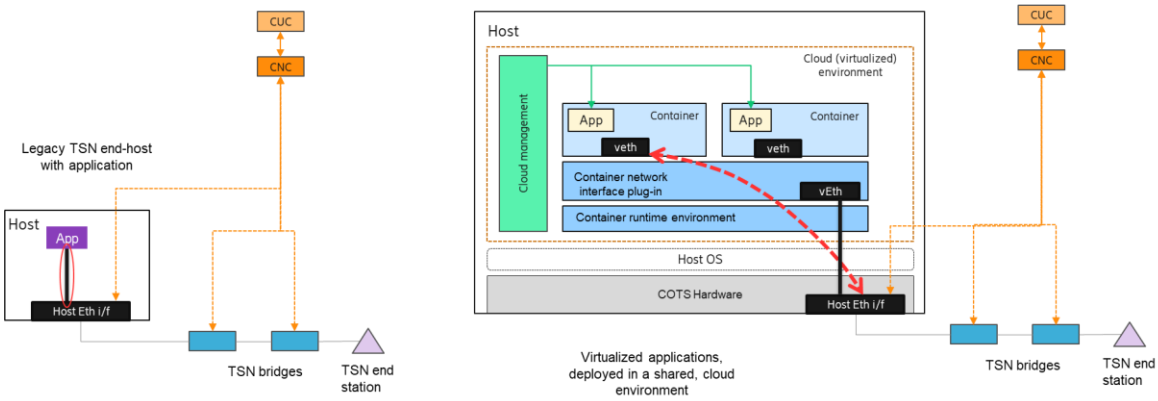

Figure 2.9 - Dedicated vs. cloudified application deployment

In contrast, in case of a cloudified application, the effect of virtualization, the shared resource paradigm, and handling of the timing of multiple application instances by the cloud management have to be considered. Furthermore, in the case of a containerized application, the application instance is linked to the virtual Ethernet port of the containers, instead of the host Ethernet interface and there is a networking in the virtualized environment between the application instance and the NIC. This results in a complex ecosystem, with multiple, interrelated factors that make timing uncertain for the applications.

Albeit the effect of virtualization can be measured or predicted, there is no way to expose this towards the TSN control plane in a standardized way, so this factor cannot be considered in an automatic manner in the design of TSN traffic forwarding. Furthermore, the timing of the virtualized networking is not deterministic, either due to the shared resource paradigm and due to changes that may occur in the networking configuration over time. The virtualized networking capabilities and details are also hidden from the CNC, so the virtual Ethernet interfaces cannot be configured directly by the TSN control plane.

Another crucial factor is the precision of the application timing and scheduling[2], which primarily determines the achievable timing accuracy. As it was mentioned in section 2.4.1 for the RTOS, proper resource allocation for application workloads and the usage of deadline scheduling in a combined way can ensure bounded execution times for time-critical applications. A related work is presented in [FAC22], where the authors propose an architecture framework with the ability to deploy real-time software components within containers in cloud infrastructures. The essence of the idea is the deployment of containers with guaranteed CPU scheduling by using a deadline scheduling policy. The framework can ensure time execution guarantees for the application components that have time-bounded characteristics. On the other hand, as illustrated in Figure 2.10., the bounded execution time does not automatically mean the proper ordering of the applications, which is essential to generate the packets, in accordance with the planned 802.1Qbv schedule in the TSN communication domain. This experiment is produced by using the DETERMINISTIC6G simulation framework [DET23-D41] features for edge computing simulations. We simulate five applications that periodically send packets (one packet in every 10ms) to a common receiver where the packet arrival time is measured. Each application is modelled by a separate host in the simulation to test the effects of SCHED_DEADLINE without cross-application interference.

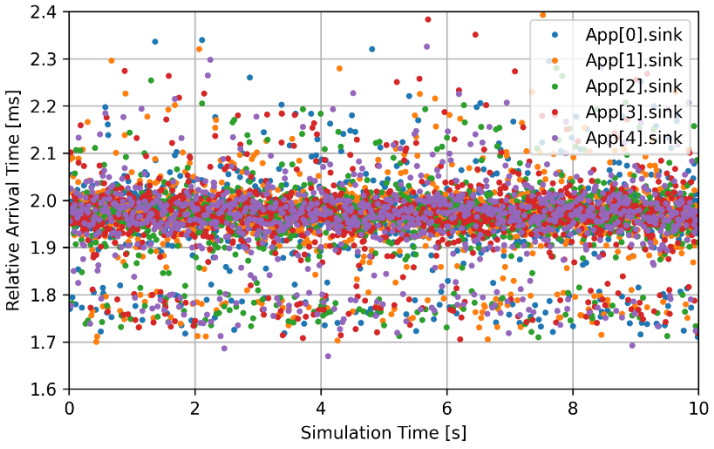

Figure 2.10 - Packet ordering variances in case of deadline scheduling

As depicted on Figure 2.10, the deadline scheduling is an appropriate solution to ensure bounded latency for application execution (latency bound for application execution is around 2.4ms). However,

[2] The term scheduling is used here for the low-level scheduling of resources, i.e., CPU timeslot allocation to a certain application process.

it does not itself result in the constant order of application execution (the order of execution of applications can vary in different cycles), so the order of the packets can vary.

The main issue with the inadequate ordering of packets resulting by the stochastic variances in application scheduling and execution times, is that it may result in unpredictable packet (re-)ordering in other bridges within the network as illustrated in Figure 2.11. These detrimental consequences are very difficult to detect and may propagate to other devices, and ultimately lead to the collapse of the 802.1Qbv scheduling plan in the network.

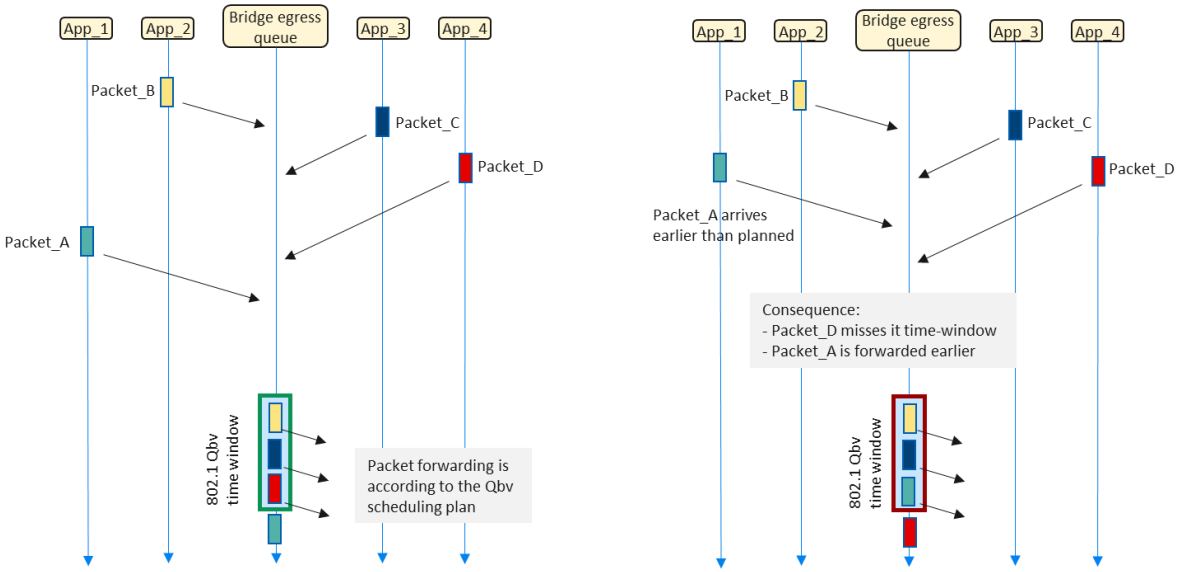

Figure 2.11 - Effect of packet ordering uncertainties

The left side of the figure shows the case when each packet arrives to an intermediate network bridge in the correct order, which ensures that the packets will be scheduled on the egress interface according to the 802.1Qbv scheduling plan. The right side of the figure shows a case where Packet_A arrives earlier than expected and is hence queued before Packet_D. It results that both Packet_A and Packet_D are forwarded in the wrong time-window (in other words, Packet_D is delayed due to the incorrect scheduling of Packet_A). The incorrect scheduling causes a latency increase for Packet_D, which could lead to Packet_D missing its deadline. On the other hand, it also introduces uncertainty throughout the rest of the network because the impact of incorrect scheduling can propagate to other parts of the network, adversely affecting the 802.1Qbv scheduling plan there as well[3].

Based on section 2.4.1 , it can be concluded that the proper usage of real-time compute features can significantly reduce latency variations and can ensure timing guarantees on the application execution time. This is a crucial step for providing time-aware, dependable (edge-)computing execution environment for the time-critical services. However, even if the above-mentioned toolset is applied,

[3] IEEE 802.1Qci (Per-Stream Filtering and Policing) provides mechanisms for filtering and policing individual data streams to ensure robust performance by preventing congestion and ensuring compliance with predefined traffic policies. The usage of IEEE 802.1Qci can help to mitigate the harmful effect in parts of the network, but it cannot solve the origin of the problem.

TSN-grade timeliness cannot still be achieved due to the stochastic characteristics of a virtualized environment, especially if the 802.1Qbv traffic scheduling concept is applied in the communication domain, and hence it should be supported by the computing domain.

2.4.3 IEEE 802.1Qbv-aware traffic handling in Edge computing domain

The E2E seamless support of IEEE 802.1Qbv, considering cloudified applications, requires supporting the 802.1Qbv scheduling paradigm in the virtualized deployment. As mentioned above, the cornerstone of the 802.1Qbv scheduling paradigm is to ensure that the data sent from the virtualized applications arrives at the NIC within the proper time window and in the correct order, as configured to the NIC by the CNC.

To fulfill this, in the rest of this section a framework is proposed by combining the time-bounded application execution design with two alternatives of 802.1Qbv-aware traffic handling scheme in the virtualized network domain. This approach guarantees that the packets will arrive to the NIC of a certain host according to the configured Qbv scheduling plan.

Design of application execution and scheduling

As highlighted in the above section, for deploying a seamless 802.1Qbv support for cloudified applications, several important considerations must be taken into account. First and foremost, the application must be orchestrated to an appropriate infrastructure component. This involves not only ensuring that the typical resource requirements such as CPU and memory are met, but also addressing special needs such as GPU resources. Additionally, selecting fast and reliable hardware, potentially with some over-dimensioning, is crucial for optimal performance. Failure handling (resulting in re-orchestration, or requiring redundancy) is not in our current scope, but should also be considered in the full picture. Furthermore, the infrastructure node hosting the application should support real-time behavior, using a real-time operating system with real-time scheduling capabilities. In the scenario where the application is packaged as a virtual machine rather than a container, it is essential for the guest VM to also support real-time scheduling. It is important to note that using a virtual machine introduces the complexity of multiple schedulers affecting application behavior, making bare metal a preferred option over VM due to its ability to minimize virtualization overhead. Of course, the application itself should be designed as a real-time application. This requirement by itself is not (edge) cloud specific, however specific additional considerations may be needed, as described later. Additionally, it is essential that all components involved, including both the host and guest components, are time synchronized, ideally utilizing Precision Time Protocol (PTP). The entire networking solution, encompassing both the host and guest components, should support 802.1Qbv. Further details on proposed options for achieving this will be discussed in upcoming sections. Complying with 802.1Qbv requirements necessitates the implementation of deadline scheduling to ensure the application's real-time performance.

Shared, coordinated gating mechanism to support 802.1Qbv scheduling

The essence of this solution is to apply the TAPRIO[4] (Time-Aware Priority Shaper) queuing discipline (qdisc) on the virtual Ethernet interface of the containers of the time-critical applications (aka TSN

[4] https://manpages.ubuntu.com/manpages/focal/en/man8/tc-taprio.8.html

Talkers) to realize a coordinated, time-aware gating scheme for the traffic forwarding as illustrated in Figure 2.12.

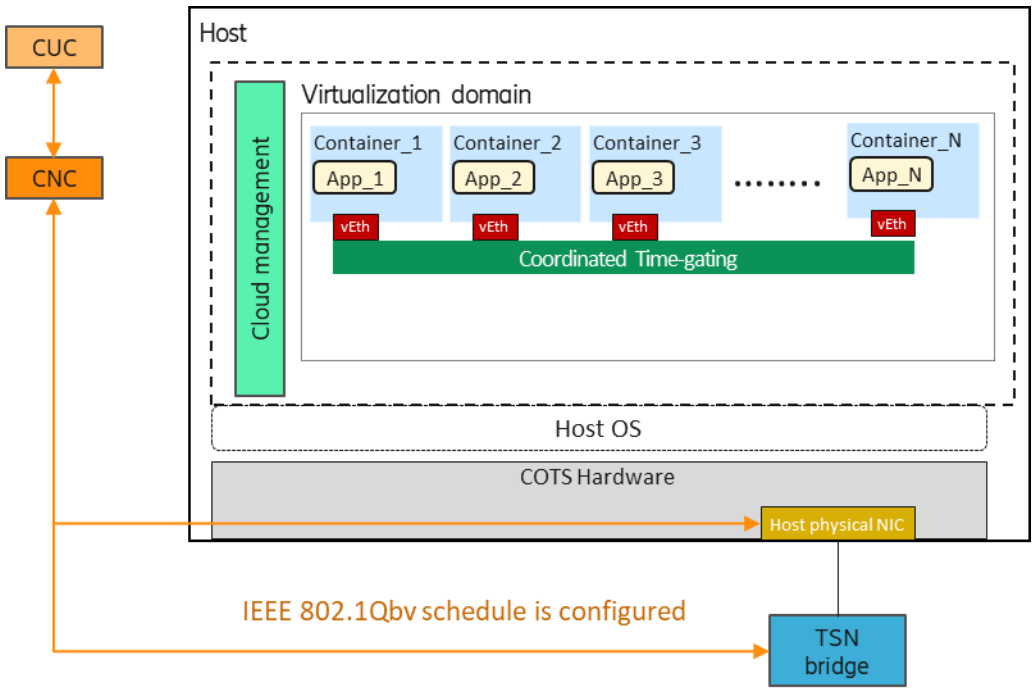

Figure 2.12 - Coordinated time-gating mechanism on container level

The operation of the method is showcased in Figure 2.13.

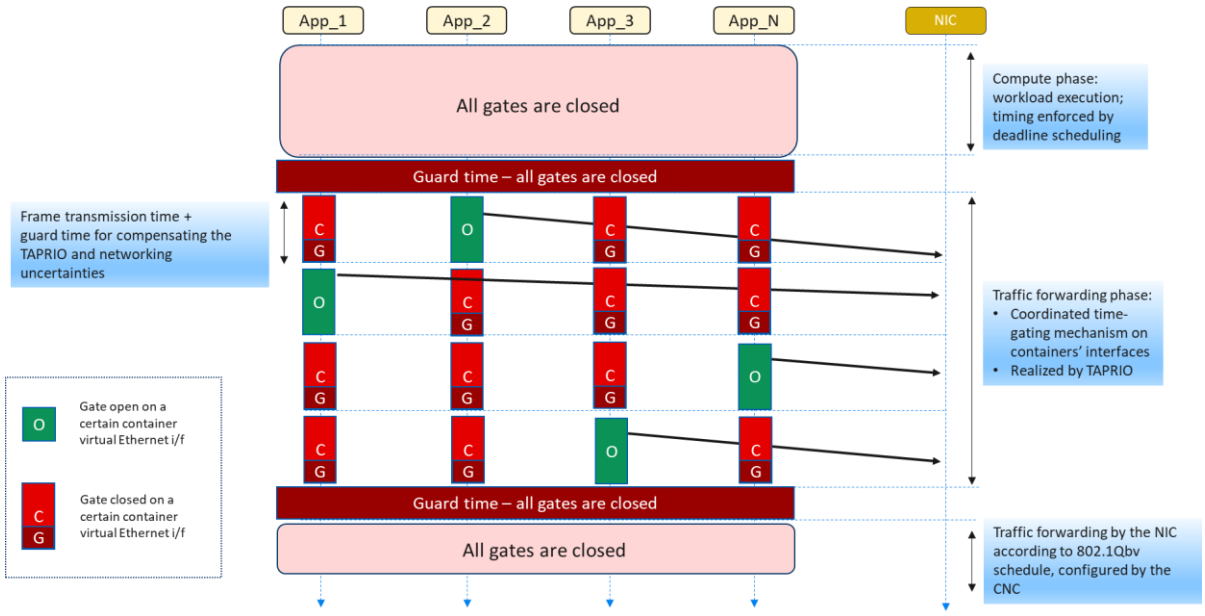

Figure 2.13 - Operation of the coordinated time-gating mechanism

Leveraging the design of application execution and scheduling described above, the operation is divided into three phases. During the compute phase, the gates of all containers are closed, so traffic forwarding to the NIC is not allowed by any application instance. In this phase, the applications are scheduled for computing according to the deadline scheduling algorithms of the host's (real-time) operating system. As mentioned, the requirement is to complete all the application computation tasks within a defined deadline, thus guaranteeing that the computation phase is time-bounded for all time-critical applications.

Due to shared resource paradigm of the virtualized environment, uncertainties may arise even with deadline scheduling. To handle this effect, a guard time is planned after the compute phase, while all the gates of containers' interfaces remain closed. Then, the next phase, the traffic forwarding in the virtualized networking is started, when a TAPRIO-based, coordinated gate scheduling is applied. As depicted in Figure 2.13, the TAPRIO qdisc is configured on the containers' interfaces in a manner that allows only one containerized application to forward traffic to the host's physical NIC at any given time. Ideally, the duration of the gate opening for a specific container should only be calculated based on the transmission time of the frame(s) to be forwarded. However, due to the uncertainties inherent in TAPRIO and the underlying virtualized networking, an additional guard time must be also considered[5] between the opening times for the traffic of consecutive containers. This guard time ensures that even in the case of inaccuracy, unwanted packet re-ordering can be avoided[6].

The configuration of the gating (the order of the packets to be forwarded to the NIC) is determined based on the 802.1Qbv schedule configured by the CNC on the NIC. However, this schedule contains only Quality of Service (QoS)-class level information, therefore in addition, application-level information should also be considered by invoking the CUC.

When all the packets arrive to the NIC, they can be forwarded to the next TSN bridge according to the configured 802.1Qbv schedule. Alternatively, packet forwarding could be started by the NIC already during the traffic forwarding phase in the virtualized domain, after the first packet has arrived to the NIC. However, in this case, the guard times should be considered in the 802.1Qbv schedule planning by the CNC, resulting in much worse utilization (no traffic forwarding is allowed during the guard times).

To sum up, the main advantage of this solution is its relatively simple implementation: only a per-container TAPRIO qdisc needs to be configured in a coordinated manner. The disadvantage of the solution arises from its shared operation; in order to secure the proper packet ordering, guard times are required to compensate the stochastic inaccuracies of the cloud deployment. This introduces some delay, as the length of the guard periods must also be taken into account, in addition to time required for packet forwarding to the NIC.

[5] The calculation of the guard time could be complex, as it depends on the specific edge cloud deployment, including various hardware and virtualization aspects. As a rule of thumb, the inaccuracy of around ~20us - 100us could be expected.
[6] Due to the stochastic operation of a cloud system, 100% guarantees cannot be ensured, but proper setting of guard times ensures that packet re-ordering may happen only in extraordinary cases.

Centralized traffic scheduling mechanism to 802.1Qbv scheduling

This solution intends to implement a centralized traffic handling scheme built on an Open Virtual Switch (OVS) – based container network interface (CNI) plugin. As depicted in Figure 2.14, in contrast to the previous approach, in this case the time-gating mechanism is realized on the egress interface of the OVS, which is directly connected to the NIC of the host.

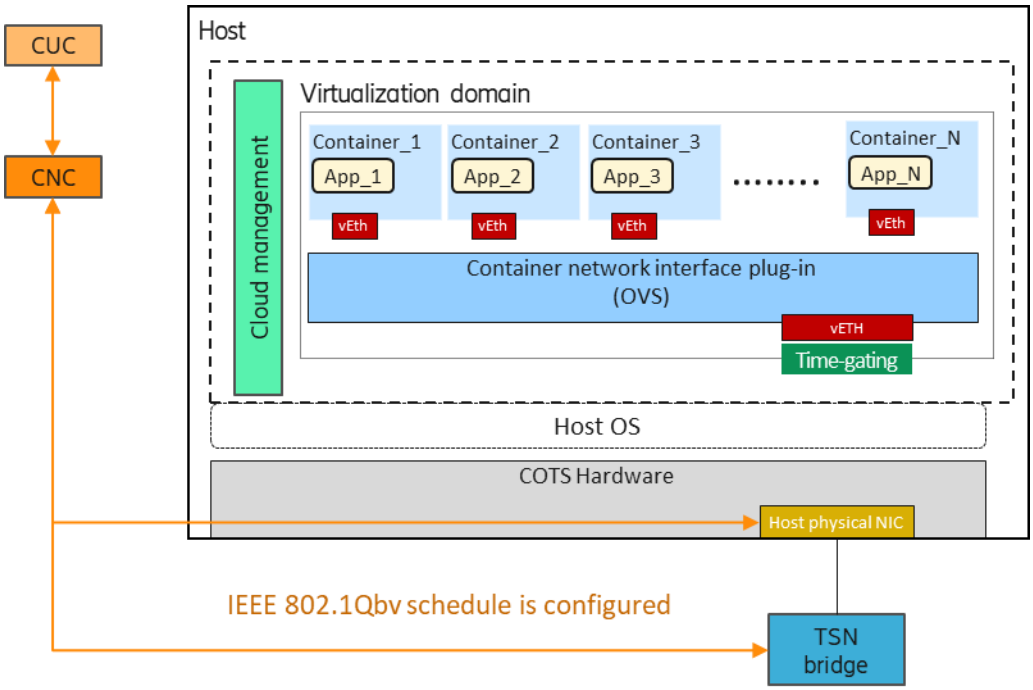

Figure 2.14 - Centralized 802.1Qbv-aware traffic handling mechanism

The essence of the operation of the method can be seen in Figure 2.15.

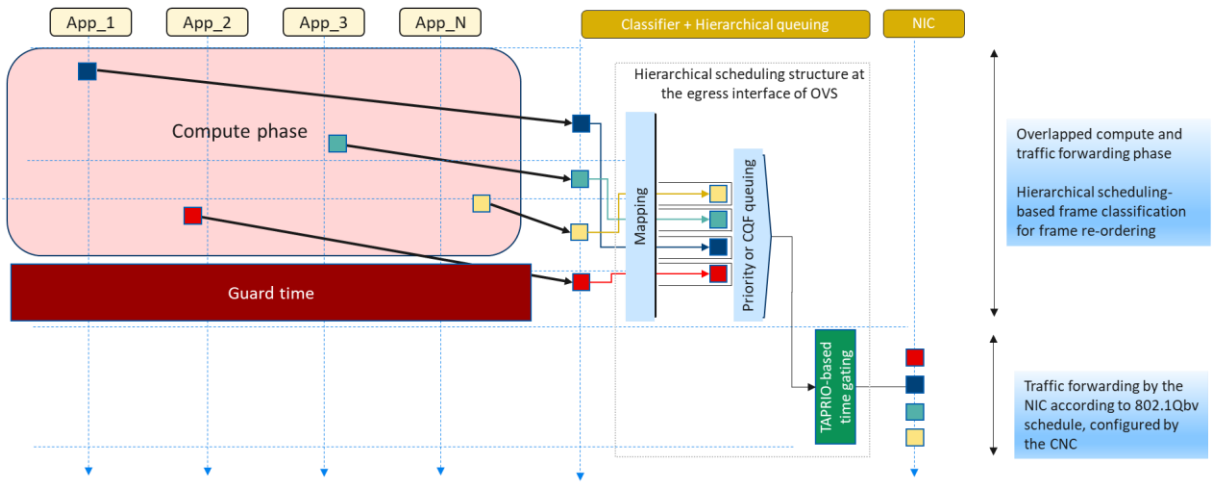

Figure 2.15 - Operation details of the centralized 802.1Qbv-aware traffic handling mechanism

In the case of this solution, there is no time-gating on the virtual interfaces of the containers. Therefore, once an application has generated its packet, it is immediately forwarded towards the physical NIC of the host via the CNI plug-in. This means that the compute phase and the traffic forwarding (in the virtualized networking domain) phase overlap, eliminating the need for guard times ensuring minimal latency. A minor guard time is only necessary at the end of the compute phase in order to ensure that all the packets arrive at the OVS before the forwarding to the NIC begins.

However, depending on the order of compute resource scheduling for the application instances, inappropriate ordering of the packets may occur. Albeit the TAPRIO qdisc can differentiate multiple QoS classes, if multiple applications share the same QoS class, TAPRIO-based differentiation and re-ordering is not possible. Similarly, once the packets arrive to the NIC's buffer, re-ordering is also not possible. Therefore, to solve the packet re-ordering, a hierarchical scheduling concept is applied at the egress interface of the OVS, which combines the priority queuing or Cyclic-Queuing and Forwarding (CQF) concepts and the TAPRIO qdisc.

Initially, a priority queuing or CQF mechanism is used to ensure the proper ordering of the frames, regardless of the incoming order. The number of queues is equal to the number of the different application instances. The incoming packets are processed by a component, called classifier, which maps a corresponding application packet to a specific queue. The packet that needs to be forwarded first by the NIC according to the configured 802.1Qbv schedule is mapped to the queue with the highest priority, while subsequent packets are assigned to lower-priority queues accordingly. The mapping can be configured on various filter criteria, for example, the source address of the container hosting the application. By using this mechanism, when a packet arrives, it is immediately mapped to the corresponding queue, according to its order based on the 802.1Qbv schedule.

The next scheduling level is the TAPRIO qdisc-based traffic forwarding, this begins when all packets have arrived and are queued. When the TAPRIO-based forwarding begins, the highest priority queue is served first, followed by the queues with lower priorities. This ensures that packets arrive to the NIC's buffer in accordance with the configured 802.1Qbv schedule, regardless the scheduling of the application instances.

One important characteristic of this method is that all packets belonging to a queue will be served (forwarded to the NIC) before serving the other queues with lower priority. Since there is not any gating (traffic policing) mechanism on the containers' virtual Ethernet interface level, it may happen that a malfunctioning application instance may send more packets than allowed, and all these packets will be queued according to the corresponding priority classification. This can cause a malfunctioning application belonging to a higher priority queue to starve the applications belonging to lower priority queues, ultimately causing significant negative impact on the 802.1Qbv scheduling plan. To mitigate this effect, a possible solution is to limit the queue lengths, according to the planned traffic volume of the applications, forcing the harmful extra packets to be dropped.

The described method is a very efficient solution for ensuring 802.1Qbv-aware traffic handling in a containerized compute environment when the number of application instances is big enough so that multiple applications are belonging to the same traffic class. Then within each traffic class, the priority or CQF scheduling is used to ensure the proper order of frames. Albeit the configuration of this traffic handling method is more complicated (hierarchical scheduling must be configured accordingly) than

the distributed method, it results in much better utilization, since guard times between traffic forwarding of individual application instances are not required.

To sum up, there is a trade-off between the shared, container-based time-gating and the centralized traffic handling methods. The shared approach definitely requires simpler configuration, but results in higher average delay and worse utilization (due to the guard times). The centralized approach ensures minimized latency increase and maximized utilization. However, this requires a more complicated scheduling configuration setup, which is more sensitive to any malfunctioning of the applications.

Comparing the per-container and the centralized scheduling mechanism by simulations

We employ our simulation framework [DET23-D41] to evaluate the impact of the per-container shaping mechanism and the centralized shaping mechanism. To this end, we simulate five applications that periodically initiate the transmission of 1000 byte packets every 10ms. As before, we model each application on a separate host to employ SCHED_DEADLINE without cross-application interference. To evaluate the per-container shaping mechanism, we connect each application's host to a separate TSN bridge, where TAPRIO qdisc-based traffic shaping takes place. The uncertain forwarding delays, induced by TAPRIO and virtualized networking, are modelled by a uniform delay between $100\mu s$ and $200\mu s$. In contrast, to evaluate the centralized shaping mechanism, each application's host is connected to a common TSN bridge without any additional stochastic delays. We report the relative arrival time for each packet, i.e., the sum of the application's processing delay by using deadline scheduling (as specified in [DET23-D41]) and the packet's end-to-end latency. With a simulation time of 10s, we simulate 1000 packet transmissions per application.

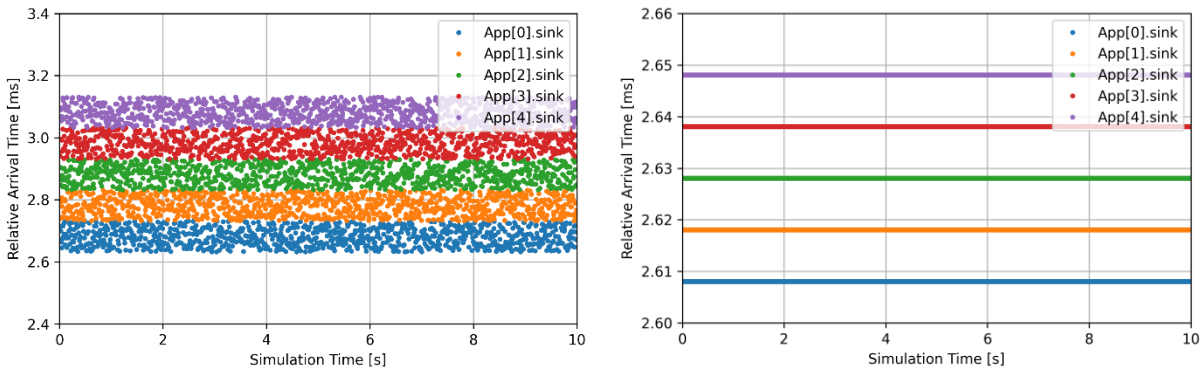

Figure 2.16 - Packet arrival for per-container shaping (left) and centralized shaping (right).

Figure 2.16 shows that both shaping mechanisms ensure a deterministic packet arrival order among the five applications. However, we find that employing the per-container shaping mechanism introduces increase in both the packets' delay variation and the packets' experienced end-to-end latency. The increase in the delay variation leads back to the additional uncertainty that is induced by the TAPRIO and virtualized networking. As it was mentioned above, to handle this, a guard timing

structure is introduced, which however, causes the increase in the delay[7]. In the centralized mechanism, much stricter requirements can be met, since in this case the jitter is compensated due to the nature of centralized operation.

3 Situational awareness via Digital twinning

3.1 Background

Digital twins (DTs) are entities, whether machines or computer-based models, that emulate, simulate, or mirror the behavior and lifecycle of a corresponding physical entity, which can be an object, process, human, or human-related feature [RHO+15]. Each DT is uniquely linked to its physical counterpart through a distinct identifier, establishing a bijective relationship between the two.

Regardless of the implementation, DTs must have certain essential characteristics. First, an appropriate data structure is required to represent the real-world state and knowledge at a detailed level suitable for the scenario, from low-level hardware to aggregated characteristics of services and networks. Second, there must be continuous synchronization between the real world and the twin, allowing them to evolve in parallel. Depending on the scenario, the synchronization timescale can range from milliseconds to days or longer. Typically, the main data flow consists of measurements and events from the real-world asset to the twin, necessary for characterizing performance and behavior. Configurations and control actions can also be sent back to the real world using suitable control mechanisms [ÖJO+22]. DTs are dynamic, intelligent, and evolving representations, continuously synchronizing with their physical twins to monitor, control, and optimize processes and functions [BR16]. They possess predictive capabilities, allowing for the anticipation of future statuses and the testing of novel configurations to preventively apply maintenance operations. DTs interact, communicate, and synchronize with their physical twins and the surrounding environment, exchanging and updating descriptive data in real-time. Through data fusion algorithms, big data analytics, and Artificial Intelligence (AI) techniques, digital twins evolve alongside their physical twins, uncovering hidden patterns, correlations, and system descriptions. They enable the application of predictive and prescriptive techniques for forecasting failures, testing solutions, and activating self-healing mechanisms in real-time [BRC+19]. Furthermore, DTs facilitate remote access and monitoring of physical twin status for all users and stakeholders, fostering improved cooperation and faster decision-making. Strictly speaking, a DT embodies the digital representation of an individual physical entity. It constitutes a system designed to meticulously replicate a physical object in a virtual realm, requiring intelligence and continual evolution. Conversely, a DT network extends beyond singular entities, catering to modeling groups of interconnected physical objects. In essence, a DT network expands upon the DT paradigm by facilitating communication among multiple digital twins [WZZ21].

Several application domains have been investigated for the benefits of DTs including manufacturing, aviation, and healthcare. In this report, we look at the application of DTs in manufacturing, which is a sector advanced in exploring DTs as a technology. In this context, we investigate particular use cases that benefit from (6G wireless) communication, in order to explore the relationship of DTs with the communication system. DTs serve as integral components in manufacturing, offering many benefits and embodying key characteristics that propel efficiency and innovation within the industry. Acting as

[7] If the impact of latency increases and uncertainties due to TAPRIO and virtualized networking is smaller, then the guard times can also be reduced accordingly.

digital counterparts to physical entities, they enable real-time monitoring, optimization, and predictive maintenance throughout the product lifecycle. DTs hold immense importance in manufacturing for several reasons. Firstly, they optimize all facets of the manufacturing process by providing highly detailed virtual models of physical entities, from design to production stages. Secondly, they facilitate a modular approach to smart manufacturing [DEP+12], empowering autonomous modules to execute tasks independently, thus ensuring flexibility and efficiency. Thirdly, DTs rely on continuous communication between the virtual model and its physical counterpart, ensuring accurate representation and enabling timely decision-making. Finally, they facilitate predictive maintenance by continuously monitoring physical systems and predicting future statuses, thus enabling proactive maintenance and averting costly disruptions.

3.2 Cyber-Physical System (CPS)

The concept of 'cyber-physical system' was introduced in 2006 by Helen Gill to describe complex systems that became hard to define using traditional IT terminology. The concept of CPS was derived from the application of embedded systems which represents the direct interaction between the physical and digital worlds through sensors and actuators. However, CPSs mainly differ from embedded systems by not referring to a specific application or implementation. CPS is more a scientific, instead of an engineering category. The short definition of CPS is that it represents the interaction between computational and physical processes, while the concept of DT represents a digital 'copy' of a physical system to perform real-time optimizations [FQL+19]. It is important to note that the cyber (computational) system may affect more than one physical system, meaning a CPS represents a one-to-many relation.

CPS can be seen as a platform that defines the interactions between the physical and digital worlds, which may include interactions between DTs and their physical representations as well as interactions between DTs within a DT network.

Industry 4.0/5.0 assumes use-cases of complex systems and systems-of-systems with multi-layer system-hierarchy. Focusing on the scope of this project, we focus on the two main systems, the physical/operation technology (OT) system and the 6G system. Therefore, the CPS consists of these two systems and their DTs: OT DT and 6G DT. There are other commonly used terms, such as 'factory DT', 'enterprise DT', or 'industrial automation DT', which have a similar meaning. Different terms exist because of different physical environments/industries where a DT represents a physical system. The term CPS DT represents a digital 'copy' of the entire CPS including physical systems and all the interactions with the cyber/computational systems, including all DTs.

The CPS in this case assumes the interaction between the OT system and its DT, 6G system and its DT, and interaction between these two DTs. In the following subsections, the OT system, the 6G System (6GS) and their DTs will be defined, and interactions, benefits and limitations will be explained in detail.

3.2.1 Operation Technology DT

Referring to Figure 3.1, the OT system consists of the control systems, system of actuators, sensors, processing, and monitoring of these systems, as defined in [DET23-D11]. Therefore, OT DT represents a precise digital 'copy' of the OT system, including its subsystems. Each subsystem can have its DT resulting in the OT DT being a network of DTs of the OT subsystems.

The flow of information between the OT system and its DT is bidirectional. The state parameters of each component (sensor value, actuator state, position, speed, and other physical parameters) are

input to the OT DT where the running process is analysed in real-time and predicted in a certain time horizon (from milliseconds to hours in advance). The outputs of the DT are the simulated state values in real-time that are compared with the physical system to make corrections to the DT models. The predicted system state values that are expected in the future in the physical system are used to influence and correct the behavior of the physical system in advance to optimize its efficiency and resource utilization, and to increase its reliability and safety, e.g., by avoiding predicted failures.

3.2.2 6G DT

From Figure 3.1, the 6GS consists of the 6G core network, RAN part and UEs. Therefore, 6G DT is a virtual replica of the 6GS that contains information about users/flows configuration, network load, network connectivity characteristics, and other relevant system parameters. The 6G DT continuously monitors the network behavior and predicts the achievable performance such as latency, packet loss and resource occupation in advance. Predictions can be used to act and modify configurations within the 6GS by anticipating changes before they happen in the real network. The main aim of predictions of 6G DT is a reliable network operation that serves the communication needs of the applications in a dependable way, while utilizing the network resources in an efficient way, e.g. avoiding excessive overprovisioning.

While this and the previous subsection focus on the scope and interaction between a physical system and its own DT (OT system with OT DT and 6GS with 6G DT), the next section focuses on the interaction between these two DTs to fully benefit from SA.

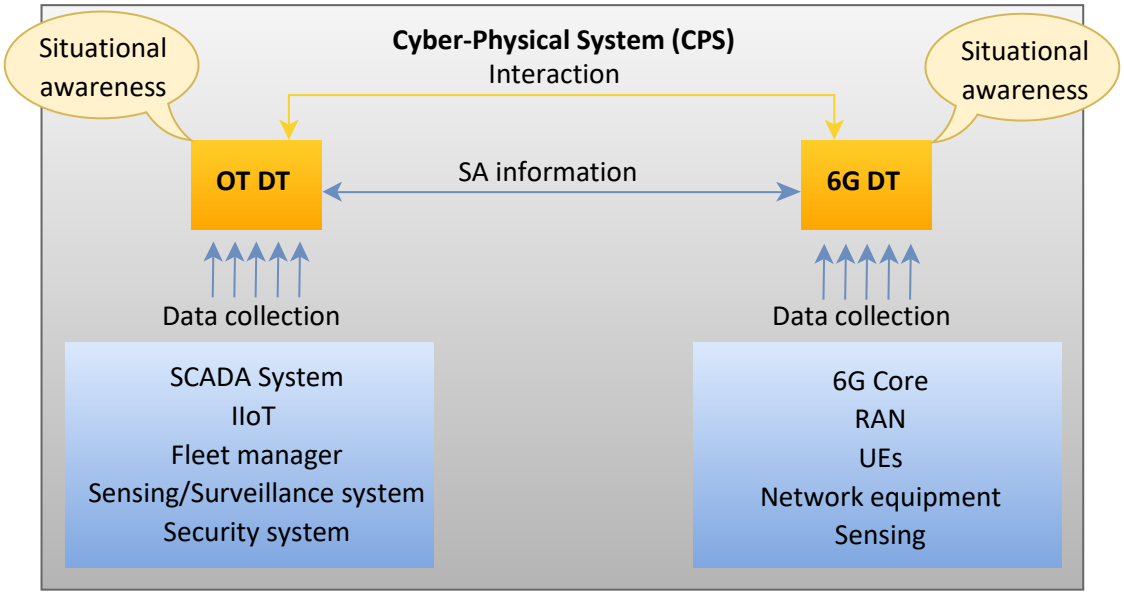

Figure 3.1 - Situational awareness via DTs interaction

3.3 Situational awareness via DT interaction

Situational Awareness (SA) has at first been investigated in the aspect of human dynamic decision-making in various domains. It has different definitions but the most commonly used one is by Endsley:

"The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future" [End95]

If the same SA model is applied to the 6G network or to the entire CPS, the three main steps towards obtaining the SA are perceiving or sensing the environment and the system behavior, analyzing the collected data to understand the current state, and finally, projecting the future state of CPS. The projection is then fed to the decision-making logic and the performance of consequent actions is fed back to the next SA analysis. Considering the computation capabilities that exist nowadays and continuous advancements in AI technologies, it is the right time to bring SA into the 6G network and the CPS. Nevertheless, the collection of context and situational information should not be excessive, but shall rather be motivated by a purpose and value that this information can provide to the DT. Various sensors are integrated into the machinery and also the factory floor environment. CPS subsystems often rely on precise positioning, Light Detection and Ranging (LiDAR) sensors and visual sensors such as cameras that can be fixed installed for surveillance or mobile deployed as a part of AGVs, temperature sensors, light curtains, etc. This sensing data is collected by the OT DT and translated to SA which is then used to enhance e.g. AGV navigation and production optimization. Although the 6G network is a separate subsystem of CPS, sensed data and the situational information available at the OT DT can provide valuable information for the 6G DT. The 6G DT itself generates SA for communication services from a different set of parameters that it receives from the 6GS such as radio resource usage, network coverage, provided QoS, etc. However, additional information received from the OT DT can greatly improve the timeliness and trustworthiness of 6G SA. On the other hand, the OT DT can also benefit from data provided by the 6GS such as service availability or localisation or sensing data obtained via the Joint Communication And Sensing (JCAS) solutions. The availability of produced data in the digital domain in the form of DT results in the capability to gain SA in the digital domain as well. Multiple loops of the SA model can be executed to increase precision and only the most optimized decision will be mapped to the physical domain. In the following, we will present some examples of how SA could be used to support industrial safety-critical communication.

3.4 Example use case

To motivate the benefits of SA and interaction between operational DT and network DT, we analyze the cooperative AGV use case from [DET23-D11].

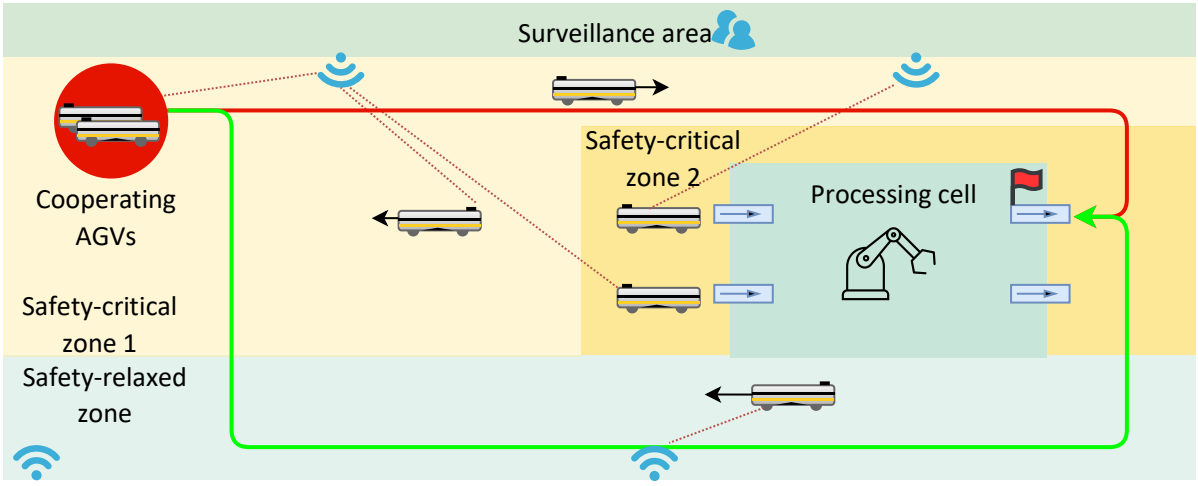

Figure 3.2 - Adaptive manufacturing use case

The use case describes a three-stage task of two collaborative virtually coupled AGVs. In the first stage, as illustrated in Figure 3.2, the cooperative AGVs circled in red, carry an object together to the processing cell where they will cooperate with processing cell robots to modify the object. On their path, they move through a safety-critical zone 1 where human presence is possible and therefore there is a need for safety-critical communication that complies with the functional safety standards. In the second stage, the AGVs carry the object through the processing cell where different stationary robots operate on the object itself. By the use case specification, this is an enclosed area without human presence and thus lowered safety-critical communication requirements. However, precisely coordinated movements between cooperative AGVs and between the AGVs and the processing robots demand even lower tolerance to communication latency with similar reliability requirements. This is because stationary robots continuously monitor and instruct the AGVs on how to orientate and position the object relative to themselves during the object modification task. Finally, after the modifications on the object are completed, the AGVs carry the object to the storage area through a safety-relaxed zone where human access is restricted. This use case implies for the wireless communication network a high degree of mobility of communicating devices and dynamically changing traffic requirements. To achieve low latency requirements for periodic traffic specified in the description of the use case, the 6GS would have to apply features and configuration that provide reliable low latency communication [LSW+19] [SAA+19], such as establishing wireless resource allocation via configured grant access, which is configured to provide periodic uplink transmission opportunities to UEs according to their traffic requirements. However, in this use case, the traffic requirements of UEs change depending on their tasks and position on the factory floor. By becoming aware of the simultaneous or planned tasks and movements of other UEs in the same operating area, this provides additional information to the 6GS and can help in planning future resource allocations in order to fulfill the promised QoS for all UEs.

3.4.1 Critical operating points

Under the assumption that proper radio planning has been performed and the 6G network is designed to support the factory requirements without significant resources overhead, in that case, it is possible to focus only on critical operating points that can occur and stress the network capabilities. The clustering of UEs is identified as a critical operating point in the covered use case.

In an industrial scenario where we have hundreds of mobile devices, it can easily occur that the capacity of one radio cell or Remote Radio Unit (RRU) is saturated if devices get clustered in a small area with poor radio signals. This can result in the unavailability of the RRU to serve the next incoming UE or even in losing the communication service for existing UEs due to a lack of resources. The reason is that although the 6G network can be aware of UE position and estimated trajectory through its localization services, it cannot predict sudden changes or new tasks given to the UEs nor can it influence the movement paths or traffic requirements.

The second critical point is the navigation of UEs through the area with poor coverage. The signal coverage on the factory floor can vary significantly which strongly depends for a certain radio network deployment on radio propagation and signal blockage. To maintain the promised QoS, more resources are needed when UEs enter areas with weaker signal coverage. This may potentially lead to resource deficiency with similar consequences as in the first critical point.

The third critical point specific to this use case is a change of operation mode from relaxed to demanding criticality by the UEs. As explained in the use case description, the traffic requirements that a UE demands, depend on its location. When the UE moves from the safety-relaxed zone to the safety-critical zone, the application requires a lower bound for the maximum latency. When the UE then enters the processing cell, the latency upper bound reduces further and the throughput rises. This means that 6GS has to overprovision the resources to cope with the varying traffic patterns or use some advanced mechanisms to manage the resources promptly on demand.

3.5 Resource and environmental awareness

To improve the resilience of 6GS and factory operation system the usage of DTs on both systems is proposed, not only for individual optimizations but rather for interactive coordination to gain and utilize SA on both sides. Both systems can benefit from their DTs and gain different optimizations and enhanced resource utilization. However, in an industrial scenario, these two systems are not independent from an operation point of view. Change of operation in a factory will affect the network and vice versa. For this reason, an active interaction between the systems and joint planning and execution of new tasks is desirable. This interaction can be lifted to the DT level where every task could be evaluated in many different scenarios and configuration options without affecting the ongoing processes. Interaction between DTs is used for exchanging collected data from their physical twins and for sending and receiving action requirements for steering their physical twins. The action can be initiated from both DTs, however, this report focuses on the 6G DT and one motivation is to provide dependable communication from the 6GS for the use case. As per the definition of DT, the DT has to be continuously fed with real-time data measurements to represent a trustworthy image of its physical twin. In our proposed concept, the 6G DT consumes relevant data from the 6G system components, including control and user plane functions and the Operation Administration and Management (OAM) system. This data is used for the operation of the 6G DT. A second part of information that extends the conventional operation of the 6G DT is received from the OT DT and it represents the current state of the factory controlling system and ongoing as well as planned activities. The 6G DT makes a broader situational assessment based on the information received from both sources.

From the analysis of critical operating points mentioned in the use case section, it can be assessed if communication failures are likely to occur, e.g. due to resource deficiency at a certain point in time or a certain location. We propose one desirable way to detect or even predict these situations on time and try to avoid resource starvation through various optimization activities. SA is key information for detecting, predicting, and handling such critical scenarios.

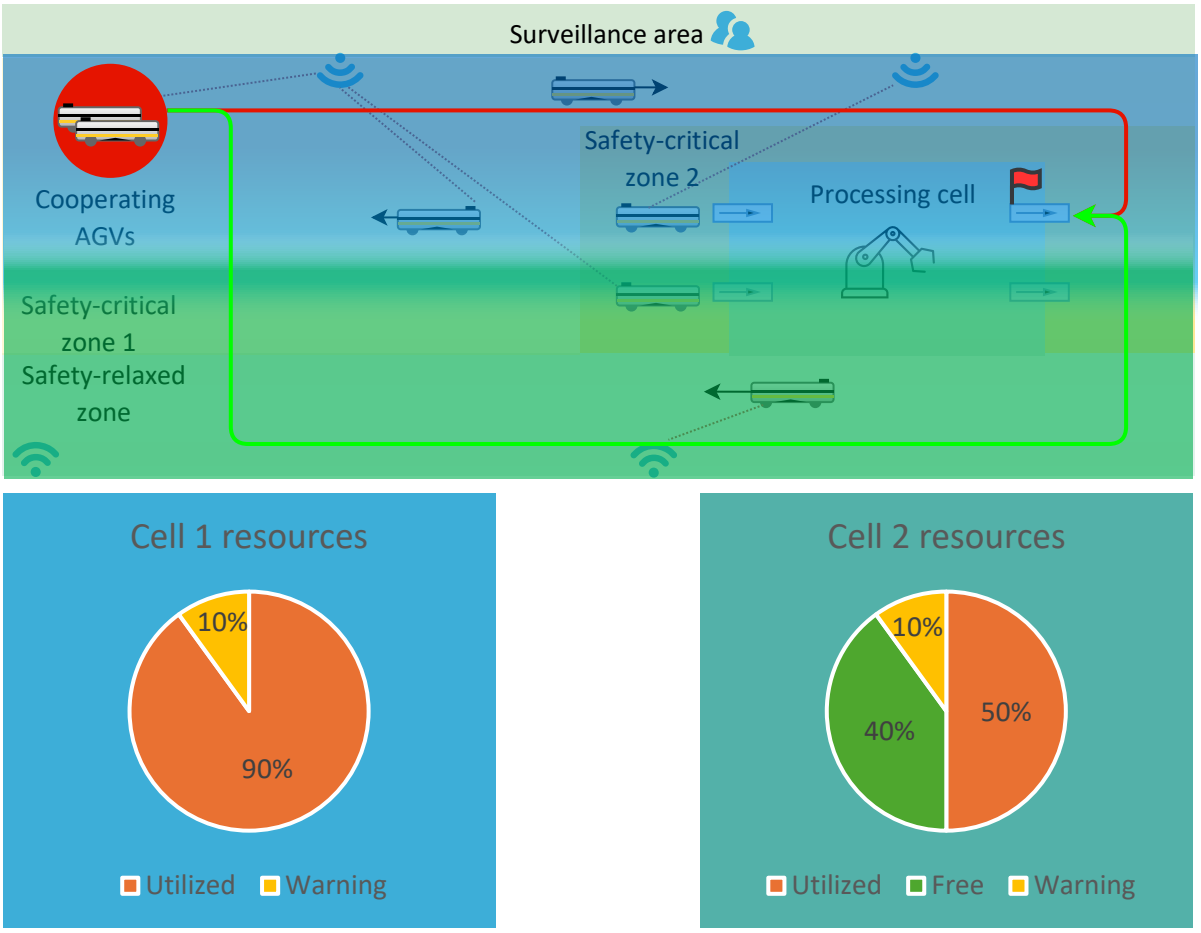

Figure 3.3 - Resource awareness, cell resource utilization

Figure 3.3 illustrates the scenario where the factory operation could lead to the clustering of UEs and the first critical operating point. The factory area is covered with multiple RRUs of which some are dedicated to cell 1 and some to cell 2, depicted with blue and green colors respectively. The blue cell serves the UEs in the more demanding processing area and has 90% of available resources occupied, while the green cell has 50% of resources occupied. The factory fleet manager plans to send two cooperative AGVs on a new task towards the processing area. By the default behavior, the fleet manager would pick the shortest possible way for the navigation, highlighted with a red arrow, which leads through the safety-critical zone 1 thus demanding the service of the blue cell. Being at 90% resource utilization, it would be easy to analyze if the blue cell can support two more UEs in its current state. However, it is important to emphasize that it is not sufficient to perform the analysis only once but rather to simulate the state of the network for each future point of UE trajectory (dots on a trajectory in Figure 3.6) from its current location to its end station in the processing area, taking into account temporal and spatial information. The path planning of AGVs is performed at the OT DT as it is in the factory responsibility domain. The fleet manager of the CPS maintains the navigation maps and has information about AGV battery status, docking station location, charging schedule, maintenance schedules, and capabilities of each AGV. It ensures maximum utilization of its mobile fleet. Therefore, for joined planning of new assignments for AGVs the OT DT informs 6G DT of possible trajectories as shown in Figure 3.6 by giving the necessary parameters for intermediate points of each trajectory. The first point of each trajectory is the current location of the UE and with known relations

between parameters given by OT DT for trajectory point and current parameters collected from 6GS, the 6G DT can precisely calculate the current resource needs and availability. If DTs joint analysis is done for each AGV then 6G DT knows their mobility too. Therefore, the same analysis can be done for future trajectory points by taking into account where each UE will be at those time points. However, predicting the radio conditions in future time points is a challenge.

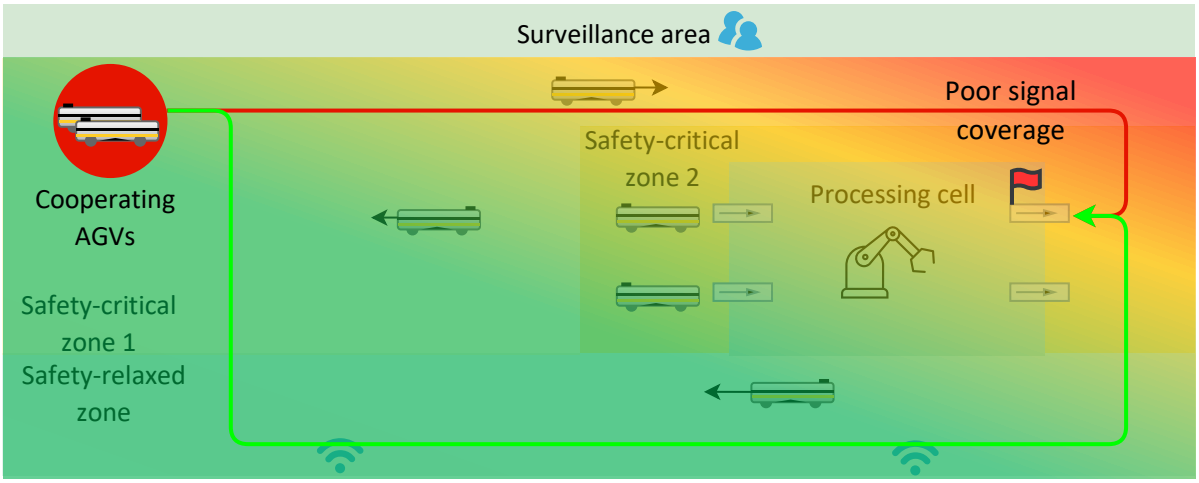

Figure 3.4 - Radio awareness, signal coverage

Knowledge about radio conditions, i.e. radio awareness, enables detection and acting on our second described critical point of the use case. The radio-aware 6G DT could capture information of the Physical (PHY) and Media Access Control (MAC) layer operations of UE and RRU [KHP+22]. Based on the capabilities of the sender and receiver, including physical coordinates, mobility attributes and radio network performance and capacity insights, the radio network performance and capacity can be continuously calibrated based on measurement of many UEs that are moving on the factory floor. Dynamic effects may be caused by the changing radio environment, mobile equipment such as AGVs carrying large metal objects, and external interferences from other networks. Predictability of such changes in radio environment might be improved through environmental sensing. Various sensors are constantly used in the factory as described in the section about SA and produced data is available to OT DT but not directly to 6G DT. Through the proposed interaction between the DTs, the OT DT can forward the sensing data to the 6G DT and based on such data around the environment and radio performance measurements of UEs collected at the 6G DT, the 6G DT can detect the causes of potential signal degradations. This can allow to indicate areas of the factory where disturbances have been experienced or may be expected. 6G DT can create an overlay map of the radio environment with link quality information and update it periodically. Figure 3.4 illustrates the scenario where unfavorable channel conditions exist in certain areas of the factory floor. It shows our use case with a channel-quality overlay layer where the green color depicts a favorable signal condition and the red color shows an area of unfavorable signal conditions. As in the previous case, the trajectory planning for the AGVs could consider this information provided by the 6G DT, which may suggest a longer path to the processing area to avoid the section of the factory with poor signal coverage as shown in Figure 3.4. Generally, the planning and operation of the CPS can improve if current and predicted network performance and capacity information can be provided by the 6G DT to the OT DT. To this end, the 6G DT should continuously assess the 6G network characteristics throughout the network, based on

measurements obtained from UEs, based on information collected from the network and based on a toolset for network performance analysis.

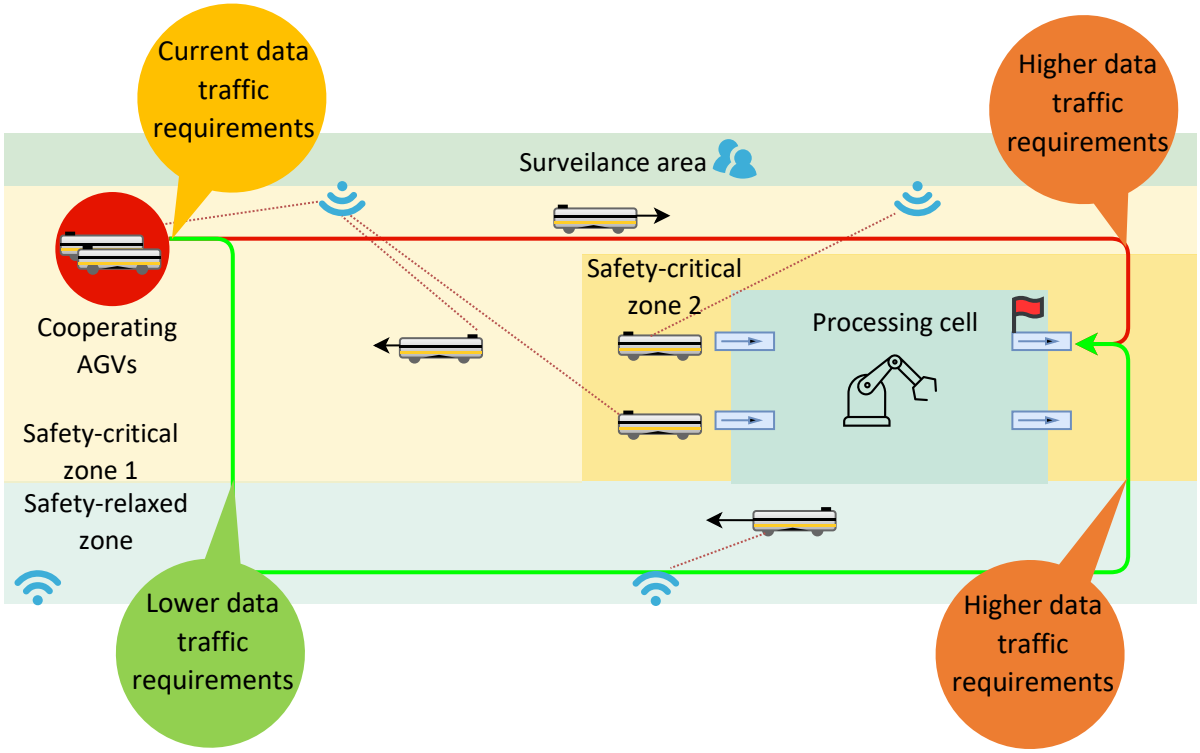

Figure 3.5 - Data traffic requirements awareness

Traffic requirements are the third critical point explained in the use case section. Due to different safety-critical areas and phases of the task, different AGV trajectories can demand different data traffic requirements in the form of throughput, latency, message cycle rate, and reliability. Moreover, the data traffic requirements can change within the same trajectory. This is shown in Figure 3.5. Without SA, the UE would have to specify the maximum data requirements it would need or the exact ones at the moment they are needed. However, it is difficult for the resource allocation in the 6GS to ensure resource availability at these critical points without reservation of resources in advance. Therefore, the 6GS has to assess future traffic demands and resource requirements (including worst-case scenarios) to plan for future resource needs (and potentially reserve resources for anticipated critical applications).

3.5.1 Parameters relevant to situational awareness

The crucial point of this approach is the accuracy of predictions. There is an enormous number of parameters that can be used to derive a SA. However, it is possible to extract the most relevant ones. Table 3.1 lists relevant parameters and their sources.

Table 3.1 - List of relevant parameters for situational awareness

Parameter	Source	Type	Remark
Communication endpoints	OT DT	dynamic	IP addresses

Analysis points	OT DT	dynamic	list of coordinates with time
Movement speed	OT DT	dynamic	per analysis point
Throughput requirement	OT DT	dynamic	per analysis point
Delay upper bound	OT DT	dynamic	per analysis point
Packet sending rate	OT DT	dynamic	per analysis point
Packet loss tolerance	OT DT	dynamic	per analysis point
Number of cells	RAN	static	per cell
Bandwidth	RAN	static	per cell
Time Division Duplex (TDD) pattern	RAN	static	per cell
Position of RRU	RAN	static	per cell
Numerology	RAN	static	per cell
Multiple-Input Multiple-Output (MIMO) support	RAN	static	per cell
Resource utilization	RAN	dynamic	per cell, measured
No. of serving UEs	RAN	dynamic	per cell, measured
5G QoS Identifier (5QI)	UDM/PCF	static	per UE
Subscriber data	UDM	static	per UE
Position	LMF	dynamic	per UE, measured
Channel Quality Indicator (CQI)	RAN	dynamic	per UE, measured
Data rate	RAN	dynamic	per UE, measured
Uplink (UL)/Downlink (DL) latency	RAN	dynamic	per UE, measured
Packet Delay Variation (PDV)	RAN	dynamic	per UE, measured

There are static parameters that are usually configured once and are not changing during the operation of CPS and dynamic ones whose value can change in different time intervals. Furthermore, the dynamic parameters can be given by certain services or NFs or they can be acquired through the measurements. We assume that parameters related to factory operation and controlling are provided by OT DT as it represents a trustworthy image of factory operations. OT DT can provide communication endpoints for each new task as this indicates if communication will be established between UE and an application on the local edge platform or cloud, or between two UEs where RAN has to provide resources for both UEs. Following the third critical point described in the use case, the communication requirements of AGVs depend on their location. Therefore, it is not sufficient for OT DT to specify only the start and end points of the trajectory, but rather intermediate points. Accordingly, parameters that depend on AGV position are given for each intermediate point as indicated in Table 3.1. in the fourth column. Static parameters related to the 6GS cell or to the UE are specified in RAN or in core NFs and can be given to the 6G DT in the network deployment phase or after the network reconfiguration. Another set of relevant parameters are dynamic parameters that are measured by the RAN and related to end application experience. The most relevant one is resource utilization as it shows the current cell capacity and CQI, data rate, and Packet Delay Variation (PDV) that shows the QoS the UE is experiencing.

By having access to all mentioned parameters from both parts of CPS, the planned trajectories of AGVs can be analyzed to estimate if the required network performance and capacity can be provided. As illustrated in Figure 3.6, the next step is sending the SA feedback in the form of e.g. estimated latency distribution or estimated maximum Throughput (TP) to the OT DT. The OT DT includes this information in its path planning and gives feedback to the physical twin to execute the task with AGVs following

the more reliable route (green route). However, the dynamic environment such as factory floor, requires continuous monitoring and adaptation. Therefore, during the movement of AGVs, the interaction between DTs continues to verify the existing projection and also to steer the AGVs towards a potentially new more suitable trajectory. The OT DT periodically sends new SA analysis requests from the current position of AGVs with all defined parameters together with the sensing data. The 6G DT uses this data together with data collected from the 6GS to calculate the deviation of the initially reported SA from the actual measurements. Additionally, it uses the new insights into the network state to generate a new projection for the new list of points. By comparing the initial projection with the new one, the 6G DT also detects changes in the network that occurred in the meantime. The unforeseen changes might be the result of varying radio channel conditions or mobile UEs that are not following their trajectories due to obstacle evasion or some third sporadic cause. Regardless of the reason, if the deviation from the initial projection is significant, new action measures are to be applied to the 6GS or delegated to the OT DT.

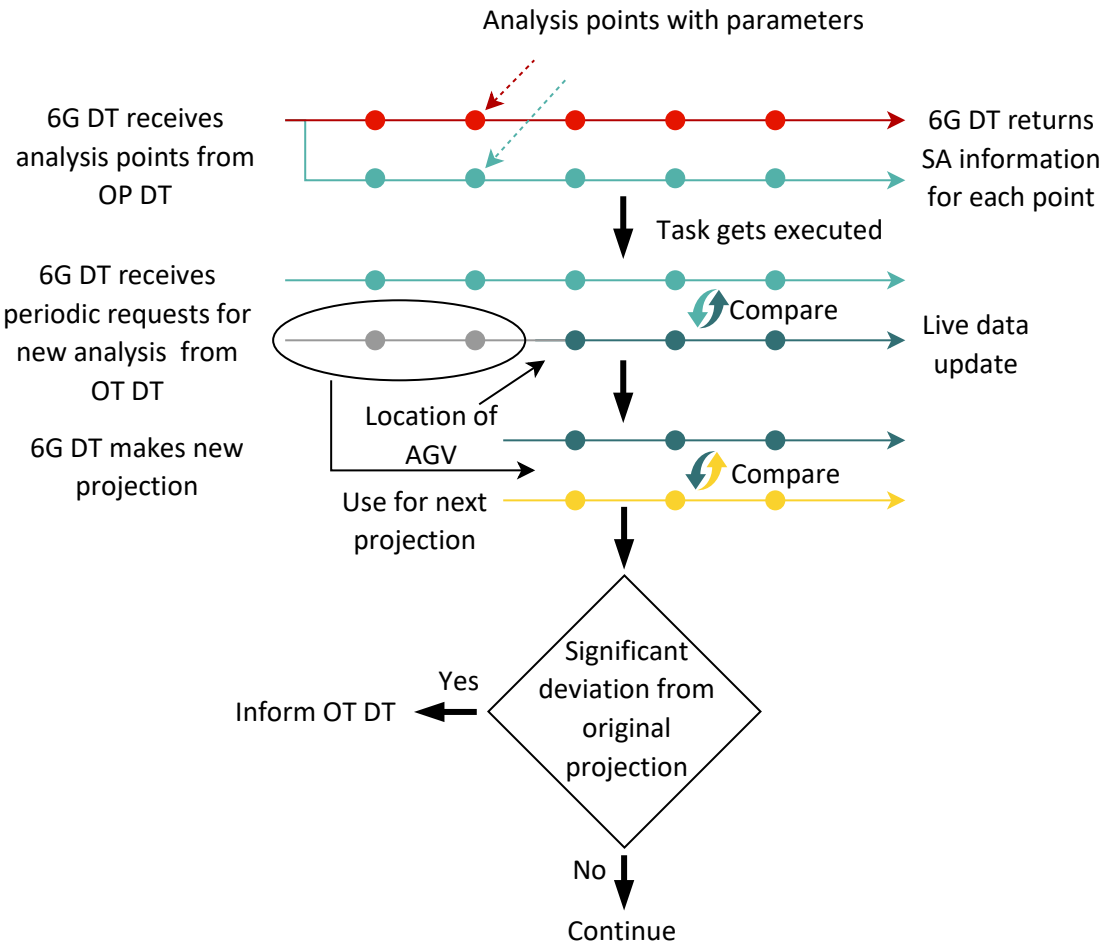

Figure 3.6 - Interaction procedure between DTs

3.6 Impact on data-driven latency predictions

AGV operations on the factory floor are highly influenced by current and anticipated communication performance. By using data-driven delay prediction models operating within the 6G DT, it can be verified if a considered AGV trajectory can fulfill the delay requirements at all times.

On one hand, data-driven delay prediction models can benefit from the SA information gathered by the OT DT. On the other hand, various planning tasks with the OT DT can leverage these delay predictions. Next, we motivate how SA can be beneficial for data-driven delay predictions and vice versa.

The traditional approach for delay predictions in 5G-Advanced/6G networks primarily relies on using network and traffic as well as historical delay samples to infer packet delay [MSG23]. Key conditions considered include the position of UEs, channel CQI, available Physical Resource Blocks (PRB) and other 5G system parameters. These conditions are fed to sophisticated data-driven AI/ML models to predict packet delay, which in turn can be used to potentially manage network resources dynamically. However, the traditional approach method of latency prediction has some limitations. This approach lacks SA of the operating environments, leading to potentially less accurate predictions and therefore suboptimal resource allocation. Due to the lack of insights into the physical state and motions within the CPS, it can be challenging to produce accurate delay predictions with sufficient prediction horizons as there can be sudden changes in traffic patterns or mobility of UEs.

By incorporating SA information from the operational environment, such as the positions and trajectories of UEs, more accurate delay predictions could be generated. For instance, consider a group of AGVs carrying a payload and arriving at a certain location on the factory floor, leading to the saturation of Radio Unit (RU) and resulting in higher packet delays. Predicting this increase in packet delay for UEs (corresponding to a group of AGVs) well in advance is challenging due to the dynamic nature of AGV movements, the finite sampling rate of network conditions, and the time required to run predictive analyses. This challenge is further increased by the fact that UEs have different traffic requirements along their paths. However, by using situational information from OT DT alongside traffic and network conditions in 6G DT, more accurate delay predictions can be achieved for longer prediction horizons. For example, the planned trajectory of AGVs, including intermediate points and the corresponding traffic requirements at those points, can be used to predict load on individual RU. Therefore, this approach enables delay prediction models to provide longer-term predictions as compared to the approach agnostic of SA information.

For the adaptive manufacturing use case, prior to the actual execution of the task when the AGVs start moving on the factory floor, the paths of all the participating AGVs need to be planned. To this end, the path planner determines route(s) or path(s) from a starting point to a goal on the factory floor. As mentioned before, different AGVs might have different delay requirements at different locations depending on the phase of the task and also different network conditions can exist in different intermediate points of the path. Therefore, the path planning algorithm should take into account the network requirements corresponding to the different locations on the factory floor while computing potential paths for AGVs. The static parameters can be used for delay predictions that are used for path planning. These include position of RUs, trajectories of UEs, channel models (e.g., obtained using ray tracing, simulation or analytical models), traffic profiles at different locations, etc. AGVs must navigate the complex environments after their paths are planned and would require tight coordination among neighboring AGVs. Therefore, it is crucial to also predict communication delays so that operations can be adjusted dynamically, e.g., altering paths or adjusting speed. To this end, continuous monitoring of parameters such as position of UEs, their channel quality, available network

resources and historical delay samples are required. The generated predictions can be used to not only control and optimize the 6G network but also adjust operation in the OT DT.

3.7 Action measures

The goal of DTs is not only to detect potential failures but also to generate precise and effective action measures to mitigate the risk of failure. The 6G DT can apply the changes to the 6G physical twin or require production changes via the OT DT.

3.7.1 6G DT action measures

An important aspect of 6G communication that could be optimized with the presented concept is handovers. Handovers in 5G are critical for maintaining seamless connectivity as UE moves between different cells or coverage areas. The process involves several steps, including measurement and reporting of signal quality, handover decision-making, preparation, execution, and completion. 5G utilizes both traditional handover methods and advanced techniques like dual connectivity, allowing devices to connect to multiple cells simultaneously. This ensures a smoother transition and reduced latency which is one of the most important KPIs of industrial communication. The classical handover procedures in 5GS are based on channel quality measurements such as Reference Signal Receive Power (RSRP), Received Signal Strength Indicator (RSSI) and Reference Signal Received Quality (RSRQ) from serving and neighboring cells. Therefore, handovers typically occur at the edge of the cell where signal quality from both serving RRUs is low. The 6G DT can analyze the trajectories of UE and determine the number of handovers on each trajectory, their effect on communication latency, and also determine the best possible position and time to perform the handover. Furthermore, from the aspect of cell capacity, resource awareness is an additional parameter that can trigger the handover. If the UE is in a range of two cells with different resource utilization, the output of DT simulations can indicate the need to perform the handover at a certain moment of UE mobility for load balancing between the cells.

To overcome the covered critical scenarios, the 6G DT can influence scheduling or simulate different link adaptation options on the trajectory. The link adaptation procedure is a highly dynamic measure that ensures the specified Block Error Rate (BLER) is not violated. It relies on channel state estimation to modify transmission parameters such as modulation scheme, channel coding, and transmission power. As mentioned, the link adaptation could be used to simulate resource needs in different points of UE trajectory and gain SA, but also, the SA can be used to enhance the link adaptation. The link adaptation is a reactive measure of the current 5GS, meaning, the system allows bit errors up to a certain BLER, after which the link adaptation reacts and changes the Modulation and Coding Scheme (MCS) to keep the BLER under the defined bound. With radio-aware 6GS and SA, the link adaptation could also be proactive where MCS changes could be applied in advance, before an area of the factory where the rise of BLER is anticipated. Potentially, higher spectral efficiency can be maintained in the area of the factory where the sudden increase of BLER is unlikely to occur. However, this highly depends on the trustworthiness of the SA. The SA can also be used to overcome the challenges of link adaptation with Configured Grant (CG) scheduling. CGs are static for a given period, which may not align well with dynamic link adaptation. CG is scheduled based on current channel state estimation and calculated MCS might not be optimal for the entire trajectory of the UE. Of course, the network can periodically perform the link adaptation and adjust the transmission parameters and scheduled grants, however, the aim is to reduce the control signaling and to ensure the resource availability

when needed. Finally, to boost reliability, the SA can enable timely scheduling of packet repetitions and Hybrid Automatic Repeat Request (HARQ) retransmissions. The link adaptation is a critical radio management function and as such it would not depend on DT input but rather it would be enhanced with additional SA information that DT provides. Further investigation is needed to better understand how this action measure would influence the reliability of 6GS.

3.7.2 OT DT action measures

Depending on the manufacturing process and the flexibility of production, OT DT can push various changes to the physical twin to reduce the risk of potential communication failures. Changing the trajectory of AGVs to ensure reliable connectivity is already explained in previous sections. Another possible action is changing the speed of AGVs, which can provide robustness against larger latencies in the control loops. One reason for this modification is to avoid cell resource starvation or to avoid temporary radio disturbances that were detected on the trajectory of AGV. The second reason is the inability of 6GS to maintain latency under the promised bound. This action relates to safety-critical communication which relies on low and reliable latency. However, the latency upper bound depends on AGV speed as per functional safety standards, the AGV has to reach the safe state within a strictly defined period. The safe state reaction time can be translated to travel distance from the moment the safety hazard is detected until the safe state is reached, e.g. the AGV has stopped. Therefore, if latency predictions generated by the 6G DT indicate an increased probability of higher latencies, then the AGV could reduce speed and tolerate increased latency. In certain use cases control applications support multiple operation modes that could be switched depending on communication quality [DET23-D11]. Changing the operation mode can influence the movement speed of the device, trigger turning off or on certain non-critical services, change stream quality in the case of video streaming, etc. SA information received from 6G DT can enhance this control application flexibility.

3.8 Costs and challenges

Running and maintaining the DT of the 6G network requires a significant amount of real-time data provisioning from base stations, UEs, sensors, and network management systems. In the presented concept, the data is also collected from factory devices, sensors, actuators, controllers and OT DT. Additionally, the data has to be filtered, denoised and properly stored. Low-quality data might reduce the capabilities of DTs. Handling the data is one of the challenges of running the DT as it requires significant processing, storage and data throughput capabilities. If both DTs are deployed on the local edge computing platforms or even in the cloud, they can be efficiently connected via wired transport. DTs also need to be connected to their physical counterparts and rely on up-to-date data collection to provide timely SA. Therefore, careful network planning and traffic management are required to avoid the negative impact of data provisioning for DTs on physical networks.

Another important aspect and challenge are security and privacy. Provisioning data for a 6G DT and interaction between the DTs require high network exposure capabilities. This introduces the risk of unauthorized access and potential difficulties in maintaining data integrity. Interaction between 6GS and 6G DT has to be trusted, however, if 6G DT receives data from OT DT which is not part of 6GS, then this data source also has to be trusted. Data flows between DT and its physical twin as well as between two DTs have to be thoroughly analyzed. Moreover, user privacy has to be ensured at all times.

4 Conclusion

The Edge computing platform, i.e. MEC, has been proposed to support applications that require low latency end-to-end. There are different specifications to define the MEC architecture, i.e. ETSI, 3GPP, but the objective is to define a platform where time-sensitive applications can be deployed closer to each other. In this report we have identified different deployment scenarios where the MEC platform can be either integrated and provided as part of the MNO infrastructure or it can be connected to the MNO infrastructure to provide low latency features. In order to showcase the importance of the integration of the compute and communication domains a novel framework is proposed, which combines the application scheduling and traffic handling in the compute domain to ensure the seamless support of IEEE 802.1Qbv for cloudified applications. Using the DETERMINISTIC6G simulation framework, it was also demonstrated how two alternative realization options of the proposed framework can ensure E2E 802.1Qbv-aware traffic handling in the compute domain. In addition, the report includes some initial performance results of the MEC platform and identifies the benefits of MEC versus cloud-based deployment to reduce delay variation.

Digital Twins are a valuable asset in novel CPSs. They are used for the optimization and predictive maintenance of CPSs in different application domains including manufacturing, aviation, healthcare, communication and others. It is not uncommon that there is a dependency between these domains and therefore a need for interaction between their corresponding DTs. The second part of the report describes the concept of establishing the interaction between the DTs for gaining and fully utilizing situational awareness in CPS to e.g. support critical services. A realistic use case that includes manufacturing, 6G communication and mobile UEs is analyzed and several critical operating points that could lead to potential production downtime or degradations are identified. The document describes how to use situational awareness to predict and act in these critical scenarios and what kind of situational information each DT can generate. Moreover, relevant parameters that can be collected from both systems and are required to derive the SA are discussed. Finally, the report illustrates a possible interaction procedure between the DTs for joint task planning and execution and describes the challenges of running and maintaining the DTs.

References

[3GPP18-23316]	*3GPP TS 23.316, "Wireless and wireline convergence access support for the 5G System (5GS)," v18.5.0, March 2024, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3576*
[3GPP18-23501]	*3GPP TS 23.501, "System Architecture for the 5G system," v18.4.0, Dec. 2023, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144*
[3GPP23-23548]	*3GPP TS 23.548, "5G System Enhancements for Edge Computing; Stage 2", technical specification, April 2023, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3856*
[3GPP23-23558]	*3GPP TS 23.558, "Architecture for enabling Edge Applications", technical specification, March 2023, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3723*
[BÖG+22]	*G. Bottari, P. Öhlén, F. Gunnarsson, F. Chernogorov, M. Puleri, V. Ayadurai, J. Kronander, F. de Laval, "Digital twins: bridging the physical and virtual worlds," in Ericsson Perspectives, June 2022, https://www.ericsson.com/en/about-us/new-world-of-possibilities/imagine-possible-perspectives/digital-twins/*
[BR16]	*S. Boschert and R. Rosen, "Digital twin—The simulation aspect," in Mechatronic Futures. Basel, Switzerland: Springer, 2016, pp. 59–74*
[BRC+19]	*Barricelli, Barbara Rita; Casiraghi, Elena; Fogli, Daniela (2019): In IEEE Access 7, pp. 167653–167671. DOI: 10.1109/ACCESS.2019.2953499.*
[DEP+12]	*J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, "Smart manufacturing, manufacturing intelligence and demand-dynamic performance," Comput. Chem. Eng., vol. 47, pp. 145–156, Dec. 2012*
[DET23-D11]	*DETERMINISTIC6G, Deliverable 1.1, "DETERMINISTIC6G use cases and architecture principles," Jun. 2023, https://deterministic6g.eu/index.php/library-m/deliverables*
[DET23-D21]	*DETERMINISTIC6G, Deliverable 2.1, "DETERMINISTIC6G first report on 6G centric enablers," Dec. 2023, https://deterministic6g.eu/index.php/library-m/deliverables*

[DET23-D31]	*DETERMINISTIC6G, Deliverable 3.1, "Report on 6G convergence enablers towards deterministic communication standards" Dec. 2023, https://deterministic6g.eu/index.php/library-m/deliverables*
[DET23-D41]	*DETERMINISTIC6G, Deliverable 4.1, "DETERMINISTIC6G DetCom simulator framework release 1'', Dec. 2023, https://deterministic6g.eu/index.php/library-m/deliverables*
[DET24-D12]	*DETERMINISTIC6G, Deliverable 1.2, "DETERMINISTIC6G first report on DETERMINISTIC6G architecture," Apr. 2024, https://deterministic6g.eu/index.php/library-m/deliverables*
[DGC2010]	*Dave McCrory, "Data Gravity – in the Clouds", dec. 2010: https://datagravitas.com/2010/12/07/data-gravity-in-the-clouds/*
[End95]	*Endsley, Mica R. (1995): Toward a Theory of Situation Awareness in Dynamic Systems. In Hum Factors 37 (1), pp. 32–64. DOI: 10.1518/001872095779049543.*
[ETSI MEC Arch]	*ETSI GS MEC 003 V3.1.1 " Multi-access Edge Computing (MEC); Framework and Reference Architecture" https://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/03.01.01_60/gs_mec003v030101p.pdf*
[FAC22]	*Stefano Flori, Luca Abeni, Tommaso Cucinotta, "RT-kubernetes: containerized real-time cloud computing", SAC '22: The 37th ACM/SIGAPP Symposium on Applied Computing, April 2022, pp. 36-39, DOI: 10.1145/3477314.3507216*
[GCL14]	*M. García-Valls, T. Cucinotta, C. Lu, "Challenges in real-time virtualization and predictable cloud computing", Journal of Systems Architecture, Volume 60, Issue 9, 2014. https://doi.org/10.1016/j.sysarc.2014.07.004.*
[FQL+19]	*F.Tao, Q. Qi, L. Wang, A. Y C Nee, Digital Twins and Cyber-Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison, Engineering, pp. 653-661, 2019*
[KHP+22]	*Kuruvatti, Nandish P.; Habibi, Mohammad Asif; Partani, Sanket; Han, Bin; Fellan, Amina; Schotten, Hans D. (2022): Empowering 6G Communication Systems With Digital Twin Technology: A Comprehensive Survey. In IEEE Access 10, pp. 112158–112186. DOI: 10.1109/ACCESS.2022.3215493.*
[LSW+19]	*O. Liberg, M. Sundberg, Y.-P. E. Wang, J. Bergman, J. Sachs, G. Wikström, Cellular Internet of Things - From Massive Deployments to Critical 5G Applications, Academic Press, second edition, ISBN: 9780081029022, October 2019*
[MOH09]	*A. Mohammadi, "Scheduling Algorithms for Real-Time Systems" PhD Thesis Queen's University Kingston, Ontario, Canada, 2009. https://api.semanticscholar.org/CorpusID:1358285*
[MPA19]	*L. Moutinho, P. Pedreiras and L. Almeida, "A Real-Time Software Defined Networking Framework for Next-Generation Industrial Networks," in IEEE Access, vol. 7, pp. 164468-164479, 2019, doi: 10.1109/ACCESS.2019.2952242.*
[MSG23]	*S. Mostafavi, G. P. Sharma and J. Gross, "Data-Driven Latency Probability Prediction for Wireless Networks: Focusing on Tail Probabilities," GLOBECOM 2023 - 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 2023, pp. 4338-4344, doi: 10.1109/GLOBECOM54140.2023.10437281.*
[ÖJO+22]	*P. Öhlén, C. Johnston, H. Olofsson, S. Terrill, F. Chernogorov, "Network digital twins – outlook and opportunities," in* Ericsson Technology Review*, vol. 2022,*

	no. 12, pp. 2-11, December 2022, doi: 10.23919/ETR.2022.9999174. https://ieeexplore.ieee.org/document/9999174
[RHO+15]	*J. Ríos, J. Hernández, M. Oliva, and F. Mas, ‘‘Product avatar as digital counterpart of a physical individual product: Literature review and implications in an aircraft,’’ in Transdisciplinary Lifecycle Analysis of Systems (Advances in Transdisciplinary Engineering), vol. 2. 2015, pp. 657–666*
[SAA+19]	*J. Sachs, L.A.A.Andersson, J. Araújo, C. Curescu, J. Lundsjö, G. Rune, E. Steinbach, G. Wikström, "Adaptive 5G Low-Latency Communication for Tactile InternEt Services," in Proceedings of the IEEE, vol. 107, no. 2, pp. 325-349, Feb. 2019, doi: 10.1109/JPROC.2018.2864587. https://ieeexplore.ieee.org/document/8454733*
[SPS+23]	*G.P.Sharma, D. Patel, J. Sachs, et. al., “Toward Deterministic Communications in 6G Networks: State of the Art, Open Challenges and the Way Forward”*
[WZZ21]	*Y. Wu, K. Zhang, and Y. Zhang, ‘‘Digital twin networks: A survey,’’ IEEE Internet Things J., vol. 8, no. 18, pp. 13789–13804, Sep. 2021.*

List of abbreviations

3GPP	3rd Generation Partnership Project
5GS	5G System
5QI	5G QoS Identifier
6GS	6G System
AC	Application Context [Client]
ACR	Application Context Relocation
ACT	Application Context Transfer
AF	Application Function
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
API	Application Programming Interface
AR	Augmented Reality
BLER	Block Error Rate
CBS	Constant Bandwidth Server
CG	Configured Grant
CFS	Completely Fair Scheduler
CNC	Centralized Network Controller
CNI	Container Network Interface
CPF	Controller Plane Framework
CPS	Cyber-Physical System
CPU	Central Processing Unit
CQF	Cyclic-Queuing and Forwarding
CQI	Channel Quality Indicator
CUC	Centralized User Controller

DetNet	Deterministic Networking
DL	Downlink
DT	Digital Twin
E2E	End-to-End
EAS	Edge Application Server
EASDF	Edge Application Server Discovery Function
ECS	Edge Configuration Server
EDF	Earliest Deadline First
EDN	Edge Data Network
EEC	Edge Enabler Client
EES	Edge Enabler Server
ETSI	European Telecommunications Standards Institute
FaaS	Functions as a Service
GPU	Graphic Processing Unit
GUI	Graphical User Interface
HARQ	Hybrid Automatic Repeat Request
IaaS	Infrastructure as a Service
JCAS	Joint Communication And Sensing
KPI	Key Performance Indicators
KVM	Kernel-based Virtual Machine
LiDAR	Light Detection and Ranging
MAC	Media Access Control
MCS	Modulation and Coding Scheme
MEC	Multi-access Edge Computing
MIMO	Multiple-Input Multiple-Output
MNO	Mobile Network Operator
NAT	Network Address Translation
NF	Network Function
NIC	Network Interface Controller
NRF	Network Repository Function
NWDAF	Network Data Analytic Function
OAM	Operation Administration and Management
OS	Operating System
OSS	Operating System Support
OT	Operation Technology
OT DT	Operation Technology Digital Twin
OVS	Open Virtual Switch
PaaS	Platform as a Service
PCF	Policy Control Function
PDU	Packet Data Unit
PDV	Packet Delay Variation

PHY	Physical
PLC	Programmable Logic Controller
PRB	Physical Resource Block
PTP	Precision Time Protocol
QoS	Quality of Service
RAN	Radio Access Network
RMS	Rate-Monotonic Scheduling
RRU	Remote Radio Unit
RSRP	Reference Signal Receive Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indicator
RTOS	Real-Time Operating System
RU	Radio Unit
SA	Situational awareness
SDN	Software Defined Network
SMF	Session Management Function
SNPN	Standalone Non-Public Network
TAPRIO	Time-Aware Priority Shaper
TAS	Time-Aware Shaper
TDD	Time Division Duplex
TP	Throughput
TSN	Time-Sensitive Networking
TSN-GW	TSN Gateway
UE	User Equipment
UL	Uplink
UPF	User Plane Function
VM	Virtual Machine
VR	Virtual Reality
W-AGF	Wireline Access Gateway Function
WLAN	Wireless Local Area Network
XR	Extended Reality