Call: H2020-ICT-2020-2

Project reference: 101015956

Project Name:

A flagship for B5G/6G vision and intelligent fabric of technology enablers connecting human, physical, and digital worlds

Hexa-X

Deliverable D7.3
Special-purpose functionalities: final solutions

Date of delivery:	31/05/2023	Version:	1.0
Start date of project:	01/01/2021	Duration:	30 months

Document properties:

Document Number:	D7.3
Document Title:	Special-purpose functionalities: final solutions
Editor(s):	Claudio Casetti (POL), Björn Richerzhagen (SAG), Lucas Scheuvens (TUD), Bin Han (TUK)
Authors:	Claudio Casetti (POL), Björn Richerzhagen (SAG), Lucas Scheuvens (TUD), Bin Han (TUK), Alberto Martínez Alba (SAG), Sokratis Barmpounakis (WIN), Davide Calandra (POL), Carla Fabiana Chiasserini (POL), Panagiotis Demestichas (WIN), Giuseppe Di Giacomo (POL), Mohammad Asif Habibi (TUK), Antonis Karaolanis (WIN), Ingolf Karls (INT), Charalampos Korovesis (WIN), Fabrizio Lamberti (POL), Vasiliki Lamprousi (WIN), Mattia Merluzzi (CEA), Honglei Miao (INT), Martti Moisio (NOF), Federico Mungari (POL), Dinh-Thuy Phan-Huy (ORA), Filippo Gabriele Pratticò (POL), Riccardo Rusca (POL), Hans D. Schotten (TUK), Mohammad Shehab (OUL), Emilio Calvanese Strinati (CEA), Karthik Upadhya (NOF), Amir Varastehhajipour (SAG)
Contractual Date of Delivery:	31/05/2023
Dissemination level:	PU[1]
Status:	Final
Version:	1.0
File Name:	Hexa-X D7.3_v1.0

Revision History

Revision	Date	Issued by	Description
	22.09.2022	WP7	Initial Table of Contents / structure
	27.01.2023	WP7	Draft of technical contributions (per-task)
	16.03.2023	WP7	First full draft of technical content
	26.03.2023	WP7	Version for WP-internal review
	06.04.2023	WP7	Complete version for project-internal review
	01.05.2023	WP7	Complete version for GA review
	31.05.2023	WP7	Final version for submission to EC

[1] CO = Confidential, only members of the consortium (including the Commission Services)

PU = Public

Abstract

This report presents the outcome and final solutions of the Hexa-X Work Package 7 (WP7) regarding special-purpose functionalities, including mapping to other technical enablers, project objectives, KPIs/KVIs and a presentation of the demonstration results.

Keywords

Dependability, sustainable coverage, I4.0, radio resource management, digital twin, key performance indicators

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101015956.

Executive Summary

This report is the final deliverable D7.3 of the Hexa-X Work Package 7 (WP7), “Special purpose functionality”. The main objective of WP7 is to address challenging use cases with special-purpose functionality to contribute to the **network evolution and expansion towards 6G**. Based on the initial gap analysis and work plan outlined in [Hexa-X D7.1] and the intermediate solutions discussed in [Hexa-X D7.2], this deliverable reports on the results achieved in WP7 and on joint work with other technical work packages in Hexa-X. Contributions in this deliverable address the following work package objectives:

- Define ultra-flexible resource allocation procedures in challenging environments such as those populated by mobile devices with special requirements and in need of coverage.
- Develop mechanisms and enablers for high dependability in vertical scenarios, enabling efficient resource support of complex and dynamically changing availability requirements.
- Support the convergence of the biological, digital, and physical worlds with human interaction through novel human-machine interface (HMI) concepts and privacy-preserving high-availability Digital Twins.

The deliverable contains an analysis of the fulfilment of project objectives related to the network evolution and expansion towards 6G, relevant outputs, and measurable results. It maps technical contributions to the Hexa-X Key Performance Indicators (KPIs) and Key Value Indicators (KVIs), updating the initial view presented in [Hexa-X D7.2]. Additionally, architectural enablers based on the end-to-end architecture in [Hexa-X D1.3] and the relation to special-purpose functionality in WP7 are discussed.

Following the structure motivated in the initial work plan in [Hexa-X D7.1], technical contributions in this deliverable are structured into three main topics: **flexible resource allocation in challenging environments**, **dependability in I4.0 environments**, and **Digital Twins and novel HMIs**. Finally, the deliverable contains an in-depth discussion of the **demonstrator**, showcasing extreme performance in handling unexpected situations in industrial contexts, and thereby addressing the collaborative robots use case family defined in [Hexa-X D1.1] and updated in [Hexa-X D1.2] and [Hexa-X D1.3].

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

Term	Description
3GPP	Third Generation Partnership Project
4G	Fourth Generation of Wireless Communications Systems
5G	Fifth Generation of Wireless Communications Systems
5G-PPP	5G Infrastructure Public Private Partnership
6G	Sixth Generation of Wireless Communications Systems
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AP	Access Point
API	Application Programming Interface
AR	Augmented Reality
B5G	Beyond 5G
BS	Base Station
CCDF	Complementary Cumulative Distribution Function
CCI	Cross Channel Interference
CD	Crowd-Detectable
CDF	Cumulative Distribution Function
CF	Cell-Free
CR	Collaborative Robot
CU	Centralized Unit
cMTC	Critical Machine-Type Communications
CNN	Convolutional Neural Network
CPE	Cyber-Physical Environment
CPU	Central Processing Unit
DL	Downlink
DT	Digital Twin
DU	Distributed Unit
E2E	End-to-End
EC	European Commission
ECG	Electrocardiogram
EEG	Electroencephalogram
EI	Emergent Intelligence
EMF	Electromagnetic Field
EMG	Electromyogram
EOG	Electrooculogram
ES	Edge Server

EU	European Union
FA	Functionality Allocation
FE	Functional Entity
FL	Federated Learning
GPS	Global Positioning System
H2020	Horizon 2020
HARQ	Hybrid Automatic Repeat Request
HE	Hosting Entity
HMD	Head-Mounted Display
HMI	Human-Machine Interface
HTC	Human-Type Communications
HW	Hardware
I4.0	Industry 4.0
IBFD	In-band full duplex
ICT	Information and Communication Technologies
In^2-X	Intra-X and Inter-X
IoT	Internet of Things
KPI	Key Performance Indicator
KVI	Key Value Indicator
LSTM	Long Short-Term Memory
LTI	Linear-Time-Invariant
LTV	Linear-Time-Variant
LU	Local User
MAC	Medium Access Control
MEC	Multi-Access Edge Computing
MIMO	Multiple-Input Multiple-Output
MJLS	Markov Jump Linear System
MKS	Mouse-Keyboard-Screen
ML	Machine Learning
mMIMO	Massive MIMO
MR	Mixed Reality
MTC	Machine-Type Communications
MTTF	Mean Time till Failure
NACK	Negative Acknowledgement
NBI	North-Bound-Interface (NBI)
NLOS	Non-Line-of-Sight
NPN	non-public or private networks (NPN)
NR	(5G) New Radio
NTN	Non-terrestrial Network

OSM	Open-Source Management and Orchestration
PD	Partially Distributed
PDU	Packet Data Protocol Unit
PER	Packet Error Rate
PHY	Physical
PLR	Packet Loss Rate
PSO	Particle swarm optimization
QoS	Quality of Service
RAN	Radio Access Network
REM	Radio Environment Map
RRM	Radio Resource Management
RSRP	Reference Signal Received Power
RU	Remote User
SIM	Subscriber Identification Module
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Radio
SW	Software
TCO	Total Cost of Ownership
TCP	Transmission Control Protocol
TD	Totally Distributed
WP	Work Package
UAV	Unmanned Aerial Vehicle
UE	User Equipment
UI	User Interface
UL	Uplink
URLLC	Ultra-Reliable Low-Latency Communications
VR	Virtual Reality
X2I	X-to-Infrastructure
XR	Extended Reality
ZED	Zero-Energy-Device

1 Introduction

Hexa-X is one of the 5G-PPP projects under the European Union (EU) Horizon 2020 framework. It is a flagship project that develops a Beyond 5G (B5G)/6G vision and an intelligent fabric of technology enablers connecting human, physical and digital worlds.

This report is the third and final deliverable of Work Package 7 (WP7) – "Special-Purpose Functionality". It contains the final update on the technical solutions proposed in D7.2 [Hexa-X D7.2] to close the gaps that are identified and analysed in D7.1 [Hexa-X-D7.1]. The solutions are mapped against use cases, the project objectives, and Key Performance Indicators (KPIs) / Key Value Indicators (KVIs) and are put into relation to the enablers discussed for the end-to-end architecture in [Hexa-X D1.3]. The final stage of the demonstrator is also detailed, showcasing core functionality discussed in this deliverable.

1.1 Objective of the document

The objective of this document is to present the final solutions of WP7 addressing the technical challenges identified in D7.1 [Hexa-X D7.1] and the project objective **Network evolution and expansion towards 6G** for I4.0 and IoT scenarios. It provides a self-contained overview on how the research gaps identified in [Hexa-X D7.1] have been addressed, including updates to the intermediate solutions presented in [Hexa-X D7.2] along the four work package objectives:

- Provide an in-depth gap analysis of existing special-purpose functionality that sharpens requirements and formulates first solutions for extreme environments.
- Define ultra-flexible resource allocation procedures in challenging environments such as those populated by mobile devices with special requirements and in need of coverage.
- Develop mechanisms and enablers for high dependability in vertical scenarios, enabling efficient resource support of complex and dynamically changing availability requirements.
- Support the convergence of the biological, digital, and physical worlds with human interaction through novel HMI concepts and privacy-preserving high-availability Digital Twins.

Additionally, the demonstrator and results obtained from its evaluation are detailed in this deliverable.

1.2 Structure of the document

This document is structured as follows. Section 2 provides an overview of all contributions in WP7 and discusses in detail how those contributions address the project objective **Network evolution and expansion towards 6G** and the performance and value indicators set for Hexa-X. It further details the relation of contributions to the end-to-end architecture and other Hexa-X technical enablers. Section 3 reports on the final solutions dealing with the need for ultra-flexible resource allocation. Section 4 discusses final solutions on the modelling, evaluation, and applications of the 6G dependability concept defined in [Hexa-X D1.3], Section 2.2, in future I4.0 environments. Section 5 details the insights and solutions regarding novel Human-Machine-Interfaces and Digital Twins that enable collaboration among machines and humans and increased utilization of intelligence. Section 6 presents the demonstrator showcasing many aspects of the work in WP7 for the use case of handling unexpected situations in an I4.0 scenario with collaborating robots and humans in the loop. Section 7 concludes this document.

2 Overview of solutions and relation to objectives, architecture, and technical enablers

Providing an update on the intermediate results that have been presented in [Hexa-X D7.2], this section gives an overview on the contributions of WP7 and their relation to the objectives, use cases, KPIs and KVIs, and the end-to-end architecture developed in the Hexa-X project. For this deliverable to be self-contained, some aspects of [Hexa-X D7.2] are also again briefly summarized in the following.

Based on the updated set of use case families presented in D1.3 [Hexa-X D1.3], Figure 2-1 provides an updated view on the resulting focus in WP7 on special purpose functionality for *Dependability in Industry 4.0* and *Sustainable Coverage in IoT*. The specific use cases addressed in WP7 have not changed compared to the initial assessment in [Hexa-X D7.1], only the set of use case families and the mapping of individual use cases has been updated to reflect the latest state in [Hexa-X D1.3].

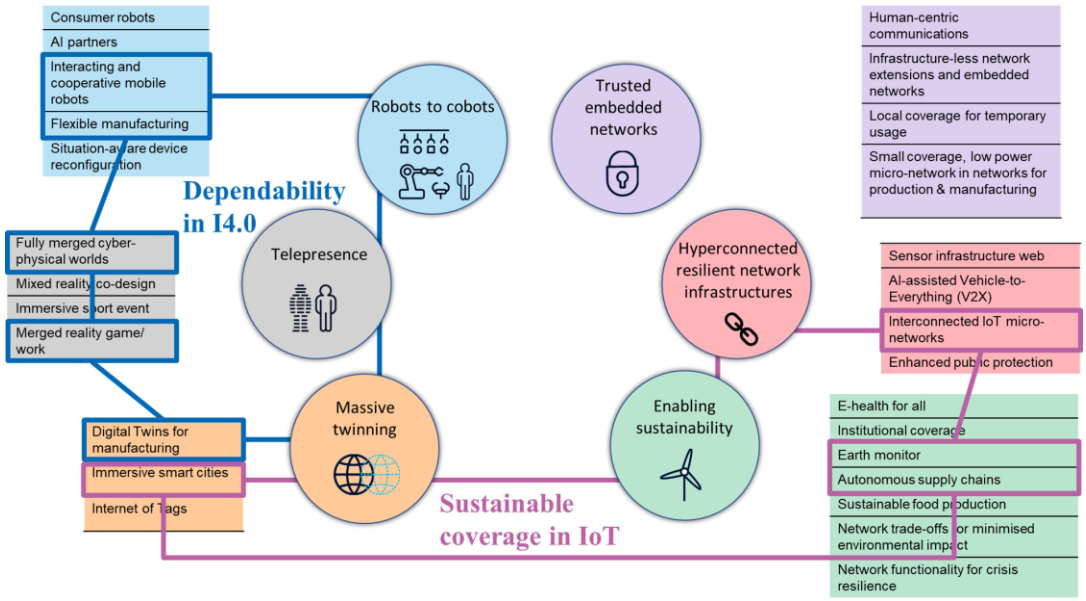


Figure 2-1: Updated set of use cases and use case families from [Hexa-X D1.3] grouped into Dependability in I4.0 and Sustainable Coverage in IoT [Hexa-X D7.1].

The following section provides an overview of the final solutions for increased B5G/6G system performance in extreme environments that are discussed in detail in this deliverable. The solutions are related to the project objectives for the *network evolution and expansion* shared with WP5, with special-purpose functionality-oriented additions to the discussion on fulfilment of these objectives in [Hexa-X D5.3]. Afterwards, in Section 2.2, solutions are mapped to the (updated) set of KPIs and KVIs defined in [Hexa-X D1.3]. The relation of special-purpose functionality solutions with the end-to-end architecture and the architectural principles and technical enablers is discussed in Section 2.3.

2.1 Overview of solutions and relation to objectives

The solutions outlined in this document address the main project-level objective **network evolution and expansion towards 6G**, which is shared by WP5 and WP7. The following WP7 specific objectives have been derived in [Hexa-X D7.2] and on the Hexa-X website[2] to address the project-level objective:

- *Provide an in-depth gap analysis of existing special-purpose functionality that sharpens requirements and formulates first solutions for extreme environments*: An in-depth analysis of use cases, requirements, and State of the Art was published in [Hexa-X D7.1], with updates especially on the topics of HMIs and Digital Twins presented in [Hexa-X D7.2], Section 5. In [Hexa-X D7.1], Section 6, first solutions for addressing extreme environments are outlined.
- *Define ultra-flexible resource allocation procedures in challenging environments such as those populated by mobile devices with special requirements and in need of coverage*: State of the Art, use cases, and requirements were detailed in [Hexa-X D7.1], including a work plan on how to address the resulting gaps. In [Hexa-X D7.2], intermediate solutions were presented, including mechanisms and models for radio-aware trajectory planning (Sec. 3.2), resource allocation in industrial environments (Sec. 3.1, Sec. 3.3), resource provisioning for Federated Learning in resource-constrained IoT environments (Sec. 3.4), utilization of ambient backscatter communication for zero-energy devices (Sec. 3.5), and flexibility in functional splits within the Radio Access Network (Sec. 3.6). The final outcomes of these works are discussed in detail in Section 3 of this deliverable.
- *Develop mechanisms and enablers for high dependability in vertical scenarios, enabling efficient resource support of complex and dynamically changing availability requirements*: The State of the Art, respective use cases and their requirements, including a work plan on how to address the gaps were discussed in [Hexa-X D7.1]. Intermediate solutions presented in [Hexa-X D7.2] included: the framework and model of Communication-Computation-Control-Co-Design (CoCoCoCo) (Sec. 4.1), methods for error identification (Sec. 4.2), increased dependability with Distributed Massive MIMO (Sec. 4.3), the utilization of a Digital Twin for efficient Radio Resource Management (Sec. 4.4), and more details on quantification and monitoring of E2E dependability, also in collaboration with WP1 on KVIs and KPIs (Sec. 4.5). Final insights on these contributions are discussed in Section 4 of this deliverable, structured into technical enablers for dependability and the overall framework of CoCoCoCo.
- *Support the convergence of the biological, digital, and physical worlds with human interaction through novel HMI concepts and privacy-preserving high-availability Digital Twins*: Initial ideas and use cases including KPIs were already outlined in [Hexa-X D7.1]. Details on State of the Art on novel HMIs and privacy-preserving Digital Twins were provided in [Hexa-X D7.2], Section 5, including initial solutions for Digital Twin-empowered collaborative robots, network awareness of Digital Twins, and Digital Twins for Emergent Intelligence. The final outcomes of these works are discussed in detail in Section 5 of this deliverable.

The following **outputs towards the project-level objective** have been delivered in WP7:

- *To enable an intelligent network, with AI and programmability tools, new protocols and components*: WP7 builds on the architectural enablers proposed in WP5 and the enablers for intelligence discussed in WP4 and summarized also in [Hexa-X D1.3] to address selected use cases with special-purpose functionality. To this end, a gap analysis was published in [Hexa-X D7.1]. Initial results and first insights from simulations, deployments in PoCs, and demos of proposed solutions were discussed in [Hexa-X D7.2], including an assessment of the relation of WP7 contributions to the architectural enablers proposed for the end-to-end architecture in [Hexa-X D1.3]. An update on this is provided in Section 2.3 in this deliverable.
- *To develop a fully converged smart connectivity platform for a wide range of network topologies, devices and sub-networks, and a multitude of special-purpose solutions for specific*

[2] https://hexa-x.eu/objectives/

segments of existing and potential new value chains: WP7 focuses on selected use cases as discussed in [Hexa-X D7.1] to study special-purpose solutions for specific segments. Contributions are further discussed in the scope of addressed KPIs and KVIs later in Section 2.2. Contributions address the high demands for dependability in industrial scenarios by studying, among other aspects, a closer coupling and resulting benefits between (control) applications and the network under the umbrella of Communication Computation Control Co-Design (Section 4). The co-existence of different local (sub-) networks and resulting implications of spectrum management in industrial scenarios is modelled in Section 3.1. On the other end of the spectrum of extreme experiences, zero-energy devices, and their integration into a 6G system for the sustainable realization of the use case of earth monitoring and selected aspects of smart cities are studied (Section 3.5). These specialized solutions rely on the overarching architectural principles and enablers for their realization, as discussed in Section 2.3. On all contributions, intermediate results were reported in [Hexa-X D7.2].

- *To support a secure, smart, and flexible network, with a B5G/6G functional architecture, utilising a fully cloud-native Radio Access Network (RAN) and Core Network (CN), reducing the total cost of ownership (TCO) related to network integration and implementation*: Here, WP7 focuses on the flexibility of networks for specific use cases. General (architectural) enablers are discussed in WP5 [Hexa-X D5.1, Hexa-X D5.2, Hexa-X D5.3] and summarized in [Hexa-X D1.3]. The relation of these enablers to the contributions in WP7 discussed in this deliverable is summarized in Section 2.3, providing an update to the earlier assessment in [Hexa-X D7.2], Section 2.

Measurable results for WP7 towards the objective are:

- *Development of a framework for a resource efficient support of complex and dynamically changing availability requirements*: The resource efficient support of complex and dynamically changing requirements is considered along two clusters of use cases, as motivated in [Hexa-X D7.1]: dependability in I4.0 and sustainable coverage in IoT. Intermediate results were presented in [Hexa-X D7.2], and updates to these results are discussed in the following. To address the challenging dependability demands of use cases in the first cluster, the CoCoCoCo-framework is proposed (Section 4.2). Here, application characteristics and behaviour are modelled (Section 4.2.1) to assess the impact of network failures on the application and allow for increased productivity even in the face of transmission errors (Sections 4.2.2 and 4.2.3). As technical enablers, mechanisms for Radio Resource Management that utilize a Digital Twin of the network are proposed (Section 4.1.1). To increase coverage and provide flexible support for changing demands, the utilization of Unmanned Aerial Vehicles (UAVs) is modelled, and their dependability is evaluated for mMTC scenarios (Section 4.1.2). Mechanisms for a deterministic control plane of the underlying programmable network infrastructure are proposed and evaluated, acting as an important building block for determinism even in the face of reconfigurations and flexibility on the control plane (Section 4.1.3). To improve the future factory efficiency and productivity, methods for joint optimization of application QoE and RAN resource management are proposed to achieve the required application-level QoE with the efficient usage of RAN resources and energy (Section 4.1.4). To increase flexibility in the face of dynamically changing requirements, a model and concept for flexible functional splits in the RAN was proposed in [Hexa-X D7.2] and is extended in Section 3.3. To flexibly support local re-use of spectrum (for example in local machine networks), the In2-X-framework from [Hexa-X D7.2] is further extended (Section 3.1). The joint optimization of radio resource assignments and flexible functional splits and the trajectories of (controllable) mobile entities is studied and respective models are proposed and evaluated (Sections 3.2 and 3.3). To address the need of distributed AI applications, an O-RAN compliant way of provisioning resources for Federated Learning is proposed and evaluated (Section 3.4). Finally, to re-use available resources for resource-constraint devices and to achieve a high density of sensors, zero-energy devices and the utilization of backscatter communication is evaluated (Section 3.5). Core aspects of the mechanisms are integrated into the demonstrator detailed in Section 6, showcasing the ability of the overall system to react to unforeseen events by flexible re-allocation of functionality and utilization of human-in-the-loop control.

- *Development of enablers for Human-Machine Interaction (HMI) and fully immersive digital twins*: Recent advances in HMIs and initial solutions have been summarized in [Hexa-X D7.2], with an updated assessment being discussed in Section 5.1 of this deliverable. The general potential of human-machine interaction and its augmentation with a Digital Twin is shown for the use case of collaborative robots in Section 5.3. The impact of human presence on the network (e.g., because of human-robot collaboration on joint tasks) is modelled as part of a network digital twin, shown in Section 5.2. The utilization of DTs for flexible functional splits is discussed in Section 5.4, further utilizing results from [Hexa-X D7.2], Section. 3.6. Finally, with decentralized decision making based on collaboration among different digital twins, the concept of Emergent Intelligence introduced in [Hexa-X D7.2], Section. 5.7 is further developed and analysed also with respect to security concerns in Section 5.5. Key aspects are also integrated and shown in the demonstrator, discussed in detail in Section 6.
- *Architectural solutions enabling connectivity of a wide range of devices and sub-networks, for a wide range of use cases and scenarios*: Adding to the solutions discussed in WP5 [Hexa-X D5.2, Hexa-X D5.3], WP7 considers the utilization and extension of architectural principles and enablers for a wide range of use cases in the two clusters "dependability in I4.0" and "sustainable coverage", as outlined previously. An overview of the corresponding contributions is given in Table 3, Section 2.3.

Quantified targets regarding the network evolution and expansion towards 6G are shared with WP5 and briefly summarized and extended with WP7-specific aspects in constrained or local scenarios in the following. For a more detailed description of the underlying methodology for the E2E view, please refer to Section 7 in [Hexa-X D5.2] and the updated discussion in Section 6 of [Hexa-X D5.3].

- *Access links supporting simultaneous high rate and low E2E latency (>0.1 Tbps @ <1 ms E2E)*: For a detailed discussion of this target and the assumptions in the scope of the overall 6G architecture, please refer to [Hexa-X D5.2], Section 7.1 and [Hexa-X D5.3], Section 6.1. According to the analysis conducted in [Hexa-X D5.3] and additional inputs in [Hexa-X D2.3], it is possible to achieve a user plane latency of below 1 ms for data rates above 0.1 Tbps. For the use cases studied in WP7, such data rates and latency requirements are most relevant within local networks (e.g., in a factory network) as discussed in [Hexa-X D7.1], Section 4, further relaxing the underlying assumptions. Focus of WP7 is on the E2E dependability, taking application behaviour into account, as described in [Hexa-X D7.2], Section 4 and Section 4 in this deliverable.
- *Supporting (>100 bn) connected devices in the network*: For a detailed discussion of this target in the scope of the overall 6G architecture, please refer to [Hexa-X D5.3], Section 6.2. Based on models of two cities with high population density, the findings indicate that the number of connected devices supported in NR exceeds the target connections with 4-5 times. In WP7, specific contributions to the mMTC case are addressing increased coverage on a regional or local scale with the utilization of UAVs (Sec. 4.1.2) and the utilization of backscatter communication for monitoring and asset tracking use cases on a (potentially) global scale (Sec. 3.5). Further benefits on a local scale (e.g., a factory network) are expected from spectrum management according to the In2-X model proposed in [Hexa-X D7.2], Sec. 3.1 and Sec. 3.1 in this deliverable.
- *(>99%) of global population reached with (>1 Mbps) data rates at sustainable cost levels; Full coverage (100%) of world area*: For a detailed discussion of this target in the scope of the overall 6G architecture, please refer to [Hexa-X D5.3], Section 6.3 and Section 6.4. The analysis is based on model assumptions utilizing non-terrestrial networks (NTNs), and additional (terrestrial) infrastructure is assumed to cater for areas with high population density. Works in WP7 contribute to a reduction in load on the network for demanding I4.0 cases (e.g., through CoCoCoCo, Sec. 4.2) or to local/regional increase of coverage with UAVs (Sec. 4.1.2). By supporting local sub-networks ([Hexa-X D7.2], Sec. 3.1 and Sec. 3.1 in this deliverable), coverage can be further increased.

2.2 Assessment of targeted KPIs and KVIs

Based on the initial assessment in [Hexa-X D7.2], an updated discussion of the projects KPIs and KVIs in relation to the solutions for special-purpose functionality outlined in this deliverable is given in the following table. The structure of the following table is unchanged compared to [Hexa-X D7.2], with definitions being provided in [Hexa-X D1.3], Section 2.2.1. However, contributions are described in more detail, including pointers to relevant simulation or measurement results discussed in this deliverable. In addition to the KPIs and KVIs defined in [Hexa-X D1.3], the concept of *resilience* is discussed in relation to dependability and the key values *flexibility* and *trustworthiness*. Additional information on these key values and potential KVIs will also be included in the final deliverable of WP1, D1.4, which is due at the end of the project.

Table 1: Contributions mapped to targeted KPIs in Hexa-X.

		KPI	Contributions
Communication	Dependability	Availability	Understand and model the impact of packet losses etc. on application performance with Communication(-Computation)-Control-Codesign ([Hexa-X D7.2]: Sec. 4.1; Sec. 4.2 in this deliverable), with the goal to achieve high application productivity. Use case is industrial control applications in I4.0, involving Automated Guided Vehicles (AGVs) and robots. Quantification of end-to-end dependability and measurement approaches ([Hexa-X D7.2]: Sec. 4.5). Modelled for the CoCoCoCo approach in Section 4.2.1 to address I4.0 control use cases. Extended to UAV-assisted mMTC use case in Section 4.1.2, allowing for dependability within sustainable coverage use cases (e.g., temporary coverage extension with mobile base stations). Additional technical enablers for dependability as listed in Section 4.1: Radio Resource Management with Digital Twins, data and control-plane guarantees in programmable factory networks, AI based joint application QoE and RAN resource optimization; addressing I4.0 production use cases. *All contributions mentioned under "Quality of Service (QoS) Attributes"*
		Reliability	
		Safety	No dedicated contributions
		Integrity	No dedicated contributions
		Maintainability	Observability with dependability monitoring ([Hexa-X D7.2]: Sec. 4.5) Mechanisms for error identification ([Hexa-X D7.2]: Sec. 4.2)
	QoS Attributes	Service latency	Deterministic latency with control and data plane guarantees ([Hexa-X D7.2]: Sec. 4.5)
		Data rate	Increasing the efficiency of Radio Resource Management (RRM) with Digital Twin ([Hexa-X D7.2]: Sec. 4.4), Section 4.1.1 in this deliverable. Further AI based joint application and RAN resource optimization in Section 4.1.4. Dependability in Massive MIMO ([Hexa-X D7.2]: Sec. 4.3), Section 4.1.2 in this deliverable.
		Resource constraints	
		Scalability	

			Utilizing ambient backscatter communications for resource constrained devices ([Hexa-X D7.2]: Sec. 3.5), Updated in Section 3.5 in this deliverable. Interference management ([Hexa-X D7.2]: Sec. 3.1), further modelled and simulated in Section 3.1 this deliverable.
AI and computation	Dependability Attributes	Agent availability	Inclusion of "Computation" in Communication-Control-Codesign (CoCoCo) ([Hexa-X D7.2]: Sec. 4.1), final solutions on compute aspects are included in Section 4.2.2 in this deliverable. Resource provisioning for Federated Learning in IoT ([Hexa-X D7.2]: Sec. 3.4) showing that in a distributed environment, such as an urban IoT one, FL can be affected by latency and bandwidth. This was further extended towards a realization of a V2X use case in the O-RAN compliant COLOSSEUM emulator in Section 3.4 in this deliverable.
		Agent reliability	
		Safety	No dedicated contributions
		Integrity	No dedicated contributions
		Maintainability	Observability with dependability monitoring ([Hexa-X D7.2]: Sec. 4.5) Mechanisms for error identification ([Hexa-X D7.2]: Sec. 4.2)
	QoS Attributes	AI service RTT	Optimal resource allocation and redistribution ([Hexa-X D7.2]: Sec. 3.3) Resource provisioning for Federated Learning in IoT ([Hexa-X D7.2]: Sec. 3.4), showing the impact of RTT on a testbed FL implementation.
		Inferencing accuracy	No dedicated contributions
		Interpretability level	No dedicated contributions
		Training/model transfer latency	Optimal resource allocation and redistribution ([Hexa-X D7.2]: Sec. 3.3) Resource provisioning for Federated Learning in IoT ([Hexa-X D7.2]: Sec. 3.4), showing that increased latency for few selected FL agents leads to worse overall performance. Emulation in a V2X O-RAN scenario (Section 3.4 in this Deliverable) highlights the magnitude of this problem. Digital Twins for Emergent Intelligence ([Hexa-X D7.2]: Sec. 5.7)
		Resource constraints	
		Scalability	
Localisation and Sensing			Utilization of location information in digital twins (e.g., for trajectory optimization) to allow for more efficient resource utilization ([Hexa-X D7.2]: Sec. 3.2, 4.4), extended in Section 3.2 and 3.3 for the case of radio-aware trajectory planning and for split-aware trajectory planning in the case of flexible functional splits. Modelling the impact of human presence, potentially augmented by sensing capabilities of 6G ([Hexa-X D7.2]: Sec. 5.3), extended in Section 5.2 for the case of industrial environments. Additionally, human pose detection is described in Section 5.3.2.2 for the use case of collaborative robots.

	Selected aspects are studied in a collaboration between WP3 and WP7 on *location and sensing enhanced services* as described in [Hexa-X D3.1], Section 5. For detailed description of the final results and the integration of localization and sensing into the 6G ecosystem, refer to [Hexa-X D3.3], Section 3.

In addition to dependability, the term *resilience* is often used to describe the same target of maintaining a high application productivity, but with more focus on (unexpected) disturbances and degradations of the system. Such degradations could be caused by errors and faults but could also be the result of external factors (e.g., natural disasters, hacking). Resilience characterizes the way the system can handle such impacts and even "come back stronger" from these situations by self-learning and adapting. In [Lap08], a shorthand definition of resilience is given as "the persistence of dependability when facing changes". This is directly linked to the Hexa-X key values of *trustworthiness* and *flexibility*, further elaborated in the final deliverable of WP1, D1.4 [Hexa-X D1.4]. One example of increased resilience because of (self-) learning and utilization of human-in-the-loop to react and maintain high productivity while adapting to unforeseen situations is showcased in the demonstrator, described in Section 6.

Regarding KVIs, Table 2 provides an updated mapping of technical contributions discussed in this deliverable to KVI areas.

Table 2: Mapping of contributions to Key Value Indicators (KVIs).

KVI area	Contributions	Remarks
Sustainability	Ambient backscatter communication [Hexa-X D7.2]: Sec. 3.5, updated in Sec. 3.5 in this deliverable.	Novel zero-energy devices for massive IoT scenarios (e.g., earth monitor). Energy autonomy under optimistic assumptions is shown. The impact of potential integration into the mobile network is discussed.
	Efficient resource allocation [Hexa-X D7.2]: Sec. 3, final solutions in Sec. 3 in this deliverable.	Efficient utilization of infrastructure, adapted to current load and conditions (c.f. flexibility KVI). Also impacted by results on dependability, as e.g., CoCoCoCo enables high application productivity at lower demands on the underlying network by utilizing cross-layer optimization.
Trustworthiness	Dependability-related contributions [Hexa-X D7.2]: Sec. 4 and Sec. 4 in this deliverable.	Increased and observable/quantifiable dependability is expected to contribute to the overall level of trust as an indicator of trustworthiness [Hexa-X D1.3].
	Trustworthy Digital Twin platform [Hexa-X D7.2]: Sec. 5.1, 5.6, 5.7, updates in Sec. 5.5 in this deliverable	Privacy-preserving collaboration among digital twins, benefiting from novel 6G capabilities (e.g., localization, sensing).
Inclusiveness	Novel HMIs (5.2) and interaction with Digital Twins [Hexa-X D7.2]: Sec. 5.4, 5.5, updates in Sec. 5 in this deliverable.	Enable remote interaction, enable inclusion of a more diverse (remote/on-site) workforce. Reduced human exposure to hazardous/dangerous situations. Example shown in the demonstrator (Sec. 6) and for DT-empowered collaborative robots (Sec. 5.3).

Flexibility	Flexible resource allocation [Hexa-X D7.2]: Sec. 3 and final solutions in Sec. 3 in this deliverable.	Mechanisms to adapt to changing requirements, mobility, device constraints, …

As already discussed in [Hexa-X D7.2], special-purpose functionality and the resulting solutions rely on technical enablers from other work packages and are part of the overall end-to-end architecture presented in [Hexa-X D1.3]. When discussing performance targets, achievable results therefore depend on those enablers and architectural principles as well. An updated discussion of the relations to architectural principles and technical enablers from other work packages is provided in the following section.

2.3 Relation to end-to-end architecture, architectural principles, and technical enablers

An initial mapping of special-purpose functionality to the 6G architectural principles and enablers was provided in [Hexa-X D7.2], showing the role of special-purpose functionality in future 6G systems.

Extending the initial assessment in [Hexa-X D7.2], an updated discussion of *architectural principles* for 6G (detailed in [Hexa-X D1.3], Section 3.1.1 and [Hexa-X D5.1]) and their relation to the solutions presented in this deliverable is provided in the following.

Exposure of new and existing capabilities – especially relevant in the digital twin ecosystem discussed in [Hexa-X D7.2] Section 5.1, and when it comes to the utilization of localization and sensing information in the digital twin for, e.g., trajectory optimization of AGVs (Sections 3.2, 3.3); further discussed for the utilization of localization and sensing in [Hexa-X D3.3], Section 3.

Designed for (closed loop) automation – enables the execution of resource allocation algorithms and methods as discussed in Section 3, for efficient placement of network and application elements. Not only on a global scale, but also for local networks or networks-in-networks and compute/AI resources on resource-constrained devices and networks.

Extensibility and flexibility to different topologies – supporting the full range: from models for In-X networks (or network-of-networks) in factory environments that can be utilized for interference management and link scheduling ([Hexa-X D7.2]: Section 3.1), to support for ambient backscatter communications on resource constrained zero-energy devices ([Hexa-X D7.2]: Section 3.5).

Scalability – a consequence of flexible resource assignment and efficient use of available resources, for example because of Communication-Computation-Control-Codesign (Section 4.2) or by benefitting from insights generated by digital twins (Sections 3.2, 3.3).

Resilience and availability – a focus of the work in WP7, with initial results reported in [Hexa-X D7.2], Section 4 and final solutions being discussed in Section 4 in this deliverable. Resilience and availability are increased by understanding the impact of errors on application productivity, and by extending dependability to an end-to-end perspective ([Hexa-X D7.2]: Section 4.2, 4.5). As previously discussed for scalability, an increased resilience is also a consequence of flexible resource assignment and the efficient use of resources ([Hexa-X D7.2]: Sections 4.1, 4.4 and Section 3 in this deliverable).

Exposed interfaces are service-based – this is expected to further ease the integration of network services and capabilities into the digital twin ecosystem, thereby allowing digital twins to benefit from additional network awareness and insights ([Hexa-X D7.2]: Sections 3.2, 4.4, 5.6). Additionally, this principle should allow easier extension of networks with specialized services offering some of the functionality discussed in this deliverable (e.g., on resource allocation, Section 3).

Separation of concerns of network functions – expected to further allow tailored fitting of deployed functions to the respective scenario, especially in local networks (or networks of networks). However, this is not in the focus of the work reported in this deliverable.

Network simplification in comparison to previous generations – considered especially relevant for the management and operation of local networks or networks-in-networks. Complexity also affects the ability of the digital twin to capture all relevant network-related aspects and the ability of humans to interact with the network and its digital twin through novel HMIs (Section 5 in this deliverable and in [Hexa-X D7.2]).

With respect to *technical enablers* for the end-to-end architecture discussed in [Hexa-X D1.3], Section 3, the functionality and solutions proposed in this deliverable benefit from further advances in other work packages. The updated set of enablers relevant to work in this WP is provided in Table 3. The table briefly summarizes the main enablers and details contributions in this WP related to the respective enabler. The final list of enablers for the end-to-end architecture will be included in D1.4 of WP1, due at the end of the project (after publication of this deliverable).

Table 3: Updated technical enablers and related contributions.

Enabler	Details	Contributions in WP7
Distributed large MIMO	[Hexa-X D2.2], [Hexa-X D2.3]	Contributing with dependability in distributed massive MIMO ([Hexa-X D7.2], Sec. 4.3).
Localization and sensing	[Hexa-X D3.1], [Hexa-X D3.2]	Formulate requirements and observe performance bounds and constraints, especially regarding utilization in the digital twin ([Hexa-X D7.2] Sec. 4.4., 5.3, 5.6). Considered for I4.0 / factory automation use case.
UE, network and service programmability	[Hexa-X D5.1], [Hexa-X D5.2], [Hexa-X D5.3]	UEs not explicitly considered - can be an enabler for novel HMI functionality or retrofitting. Network programmability with data and control plane guarantees (Sec. 4.1.3)
Network automation	[Hexa-X D5.1], [Hexa-X D5.2], [Hexa-X D5.3], [Hexa-X D6.1], [Hexa-X D6.2]	Orchestration and handling of unexpected situations (demonstrator), detailed in Sec. 6. Flexible functional split adaptation and joint trajectory optimization (Sec. 3.3). Data analytics assisted AI operation (Sec. 4.1.4)
AI and AI as a Service	[Hexa-X D5.1], [Hexa-X D5.2], [Hexa-X D5.3], [Hexa-X D4.1], [Hexa-X D4.2]	Special-purpose case of federated learning in IoT ([Hexa-X D7.2] Sec. 3.4; extended in Sec. 3.4 in this deliverable), AI and Emergent Intelligence in digital twins ([Hexa-X D7.2] Sec. 5.7; updates in 5.5 in this deliverable). *More detailed X-WP results on AI and AI as a Service are reported after this table.*
Dynamic function placement	[Hexa-X D5.1], [Hexa-X D5.2]	Utilized as enabler for resource allocation in challenging environments ([Hexa-X D7.2] Sec. 3 and updates in Sec. 3 in this deliverable), impact on dependability (e.g., [Hexa-X D7.2] Sec. 4.5).
NTN	[Hexa-X D5.1], [Hexa-X D5.2], [Hexa-X D5.3]	No direct NTN works. Utilization of drones for sustainable coverage extension in mMTC use cases (Sec. 4.1.2)
Mesh / Device-to-Device	[Hexa-X D5.1], [Hexa-X D5.2], [Hexa-X D5.3]	In2-X networks in factories ([Hexa-X D7.2] Sec. 3.1), updates in Sec. 3.1 in this deliverable.
Architectural transformation	[Hexa-X D5.1], [Hexa-X D5.2], [Hexa-X D5.3]	Reduced dependencies between network functions enables more flexible placement and, thereby, adaptation to latency requirements. Relevant for industrial scenarios with increased dependability requirements ([Hexa-X D7.2] Sec. 5.6). Potential utilization in flexible functional

		split adaptation and joint trajectory optimization (Sec. 3.3).
Compute-as-a-Service	[Hexa-X D5.1], [Hexa-X D5.2], [Hexa-X D5.3], [Hexa-X D4.1], [Hexa-X D4.2]	Availability of (trustworthy) compute capabilities for the execution of digital twins ([Hexa-X D7.2] Sec. 5). Can be enriched with allocation strategies, e.g., for federated learning in IoT scenarios ([Hexa-X D7.2] Sec. 3.4). Further studies in relation to CoCoCoCo in Sec. 4.2 of this deliverable.
Automation & Data-driven M&O	[Hexa-X D6.1], [Hexa-X D6.2]	Exposure of local (domain-)knowledge through network-aware collaborating digital twins to aid in overall resource coordination and management ([Hexa-X D7.2] Sec. 4.4, 5.6). Formulation of an ecosystem of digital twins to allow cross-domain optimization and collaboration in a privacy-preserving fashion ([Hexa-X D7.2] Sec. 5). Enabler for resource allocation, e.g., in In-X networks and networks-of-networks ([Hexa-X D7.2] Sec. 3.1).

In addition to the overview given in the table and the architectural enablers, more details on the topic of **AI and AI as a Service** is provided in the following, briefly summarizing X-WP discussions and potential for future work that is not covered by the technical contributions outlined in the later sections of this deliverable.

Flexible Digital Twin of radio propagation environment with minor predictable randomness – With the dense networks enabled by high frequency beam-based transmissions in 6G, a requirement for environmental awareness arises which can be fulfilled by creating a radio-aware digital twin (DT) of the environment where the network operates. Non-RF sensing with sensors such as LiDARs and Cameras are a potential solution to create such real-time DT of the environment which can be used to detect and track the moving entities such as humans or robots. Although such a DT provides more accurate position and dimension data, further improvement is needed to accurately characterise the dynamics of the wireless channels due to such blockages. This improvement is necessitated by the nature of human blockage models proposed in literature which are intended for use in system-level simulations and whose intention is to mimic the statistical impact of human presence on the communication link. On the other hand, calculating the exact parameter (pathloss/SNR etc.) is more precise and can help with proactively resource allocation. Ray tracing-based models appear as a potential solution for this. The main idea is to combine the real-time location and dimension data provided by the DT and ray tracing based human models to generate near real-time deterministic channel data. One can use such data to evaluate/predict link blockages enabling efficient communications. In cases where fully deterministic model is not possible, we can also use statistical models for example to predict the probability of blockage. Even probabilistic models can help in determining optimal usage of radios, for example allocation of beams.

Adapting communication service KPIs to the device state and predicting this state through AI – The problem of computation offloading entails pre-processing, transmitting, and remotely processing data to obtain, e.g., control actions to be sent back to a robotic arm. In this case, signal distortion due to wireless communication or packet losses, may still result in attained desired performance in terms of, e.g., system stability. In WP4, the concept of goal-oriented communications is investigated for the specific use case of an inference service running at the edge for data classification. The paradigm is to adapt communication KPIs (e.g., Packet Error Rate - PER) to the actual outcome of communication (in that case the correct classification of patterns on time). The idea is to not define, a priori, communication KPIs for a specific service, but rather to adapt them to the actual application performance, assumed to be measurable according to a predefined quality metric. In case of edge inference, it is, e.g., the confidence of the classifier in classifying data. The above concept could be applied to the CoCoCoCo paradigm, by considering the stability of a system, as quality metric for the application performance, adapting communication metrics to the actual control operation conditions. This would help further

reducing the communication overhead with respect to the solutions proposed in this deliverable. Different methodologies can be applied, and most of them rely on learning and adaptation algorithms, including (Deep) Reinforcement Learning (DRL), stochastic optimization, contextual multi-armed bandit.

Channel charting for improved resource allocation – With URLLC, reliability is dependent on the tail of the latency distribution. So, when targeting sub-ms latencies or high reliabilities, as is the case with extreme URLLC, it is important to be able to act proactively avoid a packet drop rather than reactively after a packet drop is observed. For instance, the receiver transmitting a NACK after a packet drop and the transmitter retransmitting the whole packet in response to the NACK will incur additional latency that may violate the stringent latency constraints. Positioning information from channel charting, studied in WP4, can help provide crucial contextual information which would enable proactive resource allocation. For instance, suppose that a UE is in the proximity of a location known from previous measurements to have a poor SINR, the network can prepare for a possible packet drop by acting proactively by either lowering the MCS or allocating additional resources in time/frequency/space for redundancy or repetition. As channel charting provides information about the local geographical neighbourhoods of UEs based on their channel measurements, it could be leveraged to detect problematic areas from a radio standpoint. Spatial trajectories of the UEs could be approximated on the learned chart and used to predict future pseudo-locations. Although channel charting is unsupervised in nature, it could be adapted to any supervised task relevant to the considered scenario to provide additional context, e.g., SINR prediction. A short additional training phase would be needed to calibrate the model and annotate the chart with the relevant information before deployment.

3 Flexible resource allocation in challenging environments

In this section, the report of progress made in exploring the use cases of WP7 is presented, specifically addressing the requirement for highly adaptable resource allocation resulting from the scarcity of bandwidth in conventional radio frequencies. Various cases are addressed, ranging from factory environment, digital twins for manufacturing, autonomous or semi-autonomous vehicles, and IoT. An additional case dealing with resource allocation and redistribution of functionalities in industrial environments is treated at length in Section 6 as a Demo.

3.1 In^2-X communication in factory environments

Wireless data traffic in factory environments appear to be a complex mixture of intra-X, inter-X, and X-to-infrastructure traffics where "X" stands for generic machines including vehicles [Hexa-X D7.2], which setup underlay subnetworks below the wide-area network, constructing a Networks-in-Network topology [Hexa-X D1.4]. The coexistence of these three distinct types of communication, especially the first two which we collectively refer to as In^2-X communication, are exhibiting heterogeneous coverages and traffic patterns, and calling for novel solutions for dynamic and intelligent spectrum sharing among underlay subnetworks (and eventually also between the underlay subnetworks and the wide-area network), which minimize the interference while efficiently utilizing the spectrum. The main technical challenges introduced by this demand are including 1) the incompatibility among various high-layer protocols used by different subnetworks regarding timing constraints that declines inter-subnetwork negotiation, 2) the interference between underlay networks and external wide-area network links outside of the factory site, 3) the lack of framework to support aligned spectrum management among underlay subnetworks, and 4) the difficulty in globally aggregating and updating the real-time spectral information from all underlay subnetworks. Are more detailed elaboration is provided in [Hexa-X D1.4], Section 6.

To address these issues, Hexa-X proposes a two-level scheme for dynamic spectrum allocation. On the macro level, a centralized dynamic spectrum licensing mechanism shall be implemented, which periodically senses the radio environment and therewith flexibly allocates different frequency bands to the overlay wide-area network different underlay subnetworks, to reduce the long-term interference especially between overlay and underlay networks. Meanwhile, on the micro level, each underlay subnetwork shall be integrated with an autonomous cognitive radio scheduler, which executes predictive time-domain scheduling to minimize interference with neighbour subnetworks regarding the sensed interference patterns.

The key challenge to the micro-level approach roots in the lack of priori knowledge of each underlay subnetwork about the traffic pattern of its neighbour subnetworks. Usually, every such traffic pattern is a stochastic superpositions of several polycyclic basic patterns, each of which is generated regarding a certain communication protocol. However, the detailed pattern constructions are unknown and non-observable by different subnetworks to each other, making classical adaptive filtering approaches fail to apply due to a huge state space and the therewith associated high computation load. To address this issue, an AI-based mutual scheduling solution was developed. It receives only the overall received interference power in the assigned frequency channel as input, invokes the Long Short-Term Memory (LSTM) method to predict the future interference power, and relies on a neural-network-based scheduler to minimizes the expected collision probability in the assigned channel. Numerical simulations in Figure 3-1 and Table 4 show that the proposed method can effectively predict the pattern of radio interference caused by neighbour underlay subnetworks.

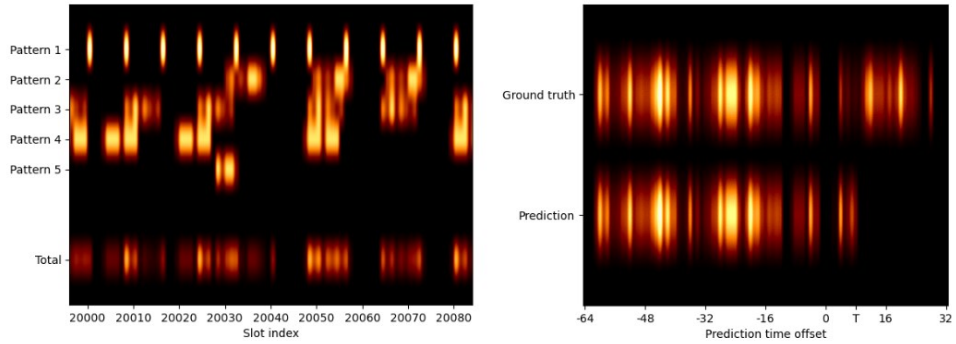


Figure 3-1: Numerical simulation results, where each of the 5 patterns on the left is generated by a particular protocol for inter-X communication. Their sum is then predicted on the right side.

Table 4: Numerical simulation results, including the accuracy in predicting the occupation of shared spectrum, and the normalized mean square error of overall predicted traffic pattern

Prediction step	1	2	3	4	5	6	7	8
Acc [%]	32.9	22.9	21.0	19.5	18.6	18.5	19.1	18.9
NMSE	0.184	0.257	0.318	0.356	0.399	0.410	0.396	0.411

3.2 Radio-aware trajectory planning

The implementation of 6G technology in industrial settings, such as with the Hexa-X use case of Digital Twin for Manufacturing, may pose significant challenges due to the need for low latency and high reliability for mission-critical services, such as industrial ethernet over the air. Additionally, these industrial use cases may also require high-quality video, augmented reality (AR), or virtual reality (VR) which demand high bitrates and bandwidths.

As discussed in [Hexa-X D7.2], Sec. 4.1.2, one can use Radio Aware Digital Twin to gain more knowledge of the radio environment and do intelligent trajectory planning to optimize the target KPIs/KVIs, especially the trade-off between radio performance and sustainability.

At least when operating in a controlled environment (for instance factory floor with private 6G network), one can know exactly where the UEs are and in many cases where they will be in the near future. As an example, AGVs moving on factory floor are typically using fixed trajectories and their routes are planned (at least within some time window in the near future).

[Hexa-X D7.2], Sec. 3.2.2 showed an example how unmanned aerial vehicle (UAV) flight paths can be optimized with radio-aware digital twin, including some quantitative simulation results. The results show that radio-aware trajectory planning allows to optimize the total aggregated throughput within certain constraints, like energy or time-of-travel. Same principles can be applied for other use cases like robots and AGVs in factories and warehouses, self-driving cars in the future etc.

Another useful technique is to share radio quality information of the trajectories between the end devices. For example, if one end device with critical communication needs starts moving to a certain area, the other end device in that area can share quality information to the device which has the intention to transmit/receive in that area. This is especially useful when end devices are moving along fixed trajectories. If the communication needs are extremely critical (in case of for example industrial

applications), the remote UE can assist the first UE by doing dedicated measurements for it – in extreme cases even physically moving for measurement purposes.

Other techniques can be combined with this, like 6G sensing (to sense the environment and obstacles, trace the vehicles and users etc.) and ML/AI techniques.

In this deliverable, the work in [Hexa-X D7.2, Sec. 3.2.2] is extended to the case of UAVs or AGVs being served by an in-band full duplex network. Current 5G BSs and UEs communicate in only one direction, i.e., uplink (UL) or downlink (DL), on a given time-frequency resource (in addition to a side-link where the devices may communicate directly with each other). In-band full-duplex (IBFD), an emerging technology, holds the potential to double the overall throughput by enabling simultaneous transmission and reception over the same time-frequency resource for both BSs and UEs.

A major challenge with implementing IBFD has to do with self-interference that results from the stronger transmit signal interfering with the weaker received signal. Suppressing self-interference at the UEs through passive isolation, beamforming, and digital signal processing-based methods will lead to increased size, cost, and power consumption. An alternative approach involves modifying the network architecture, such that an IBFD-capable BS serves two sets of legacy half-duplex UEs – one set in the uplink and the other in the DL – over the same time-frequency resource.

While self-interference at the BS can be managed with this network architecture, a UE transmitting in the UL will still cause cross-channel interference (CCI) to a UE that is receiving in the DL. The magnitude of this interference will depend on the UL transmit power and the pathloss between the UEs, where the latter pathloss on the relative location of the UEs within a cell. Provided the self-interference is suppressed sufficiently at the IBFD-capable BS, CCI then becomes a major source of interference in this network architecture. This is depicted in Figure 3-2.

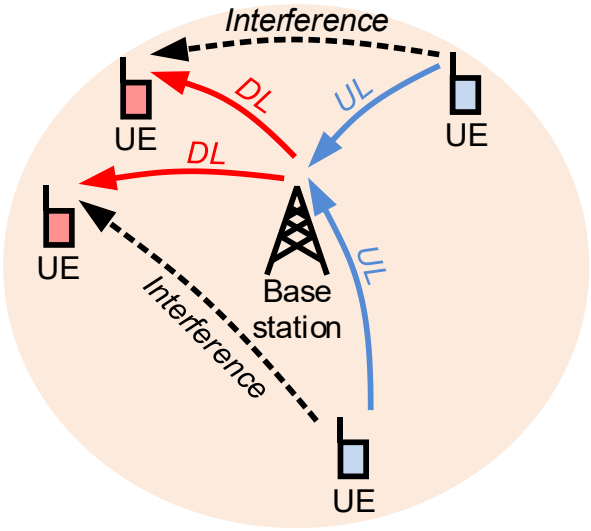


Figure 3-2: CCI in an IBFD network

A solution is proposed for the case where an IBFD-capable BS serves a swarm of UAVs or AGVs equipped with half-duplex radios. Their location is optimized based on the UE traffic pattern, i.e., whether the UE transmits in the UL or DL in the subsequent time-slots, to minimize the impact of CCI or maximize the overall throughput at the BS. For instance, a UE that has a high probability of transmitting in the UL is moved away from a UE that has a high probability of receiving in the DL. This idea is diagrammatically depicted in Figure 3-3 with UAVs.

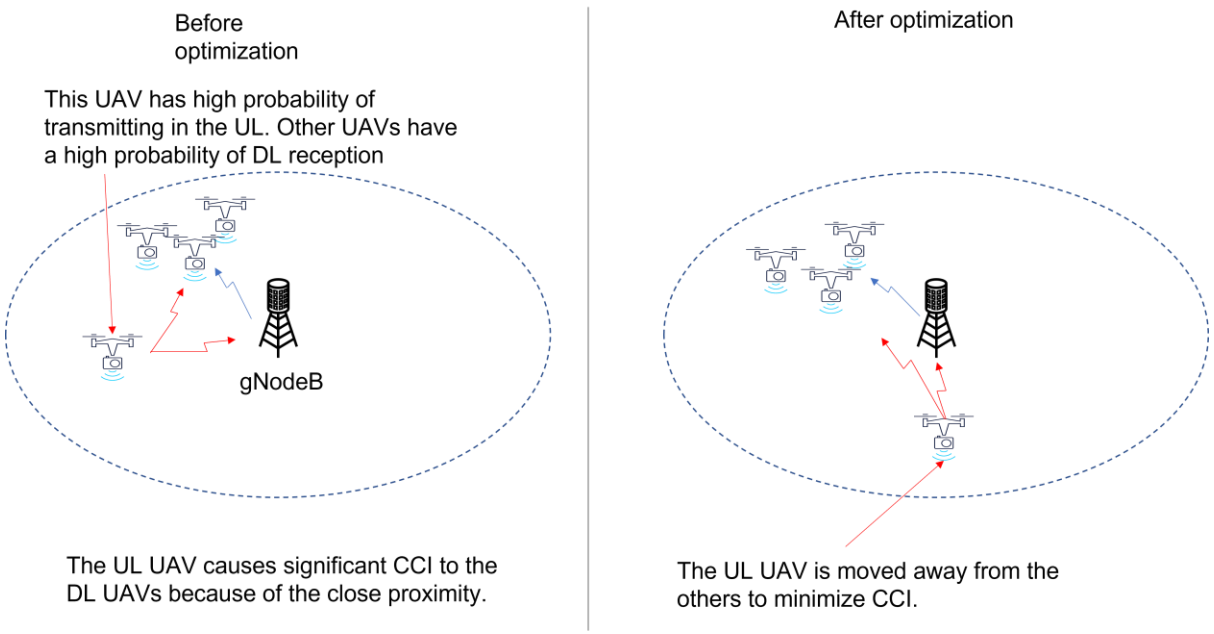


Figure 3-3: Trajectory optimization for minimizing CCI with UAVs

Specifically, consider the case of an IBFD-capable BS communicating with a swarm of K half-duplex single-antenna UEs that are UAVs/AGVs. The BS has $M \times N$ antenna elements (M columns and N rows) and transmits and receives a total of K streams over the same time-frequency resource through spatial multiplexing. A single-cell setup is considered, where the UEs have a single omnidirectional antenna for ease of exposition of the idea, although an extension to the multi-cell multi-antenna case is straightforward.

The existence of a radio-aware digital twin is assumed that can predict the channel condition between the BS and any UE, as well as between any pair of UEs. In addition, the radio-aware DT is equipped with a traffic prediction algorithm that can predict the probability that a UE will transmit or receive in the next T time-slots. Given this, the objective is to optimize the trajectory of the UEs in the swarm to minimize CCI over T future time-slots.

In the following, the notation that is used in the rest of the section is defined.

$\boldsymbol{v}_k[t]$: Position vector of UE k along in time-slot t. The position vector contains three components corresponding to the x, y, and z axes, i.e. $\boldsymbol{v}_k[t] = \left[v_k^x[t], v_k^y[t], v_k^z[t]\right]^T$

$\boldsymbol{h}_k[t]$: Channel vector of UE k at the BS in time-slot t

$g_{km}[t]$: Channel coefficient between UEs k and m in time-slot t

$\alpha_k[t]$: Product of path-loss and array gain between BS and UE k in time-slot t. $\alpha_k[t] \triangleq \mathbb{E}[\|\boldsymbol{h}_k[t]\|^2]$

$\beta_{km}[t]$: Path-loss between UEs k and m in time-slot t. $\beta_{km}[t] \triangleq \mathbb{E}[|g_{km}|^2]$.

$x_k^{ul}[t]$: UL data transmitted by UE k in time-slot t

$x_k^{dl}[t]$: DL payload data transmitted to UE k in time-slot t

η_k^{ul} : UL transmit power

η_k^{dl} : DL transmit power

$\boldsymbol{w}_k[t]$: Precoding vector for UE k in time-slot t

Given that the UEs are half-duplex, a UE can be in one of three states at time instant t, namely, uplink, downlink, and idle. Three variables $\chi_k^{ul}[t], \chi_k^{dl}[t], \chi_k^{idle}[t]$ are defined to indicate the state of the UE in time-slot t. Specifically:

$$\chi_k^{ul}[t] = \begin{cases} 1, & \text{UE '}k\text{' is transmitting in the UL in time-slot } t \\ 0, & \text{otherwise} \end{cases}$$

$$\chi_k^{dl}[t] = \begin{cases} 1, & \text{UE '}k\text{' is receiving in the DL in time-slot } t \\ 0, & \text{otherwise} \end{cases}$$

$$\chi_k^{idle}[t] = \begin{cases} 1, & \text{UE '}k\text{' is idle in time-slot } t \\ 0, & \text{otherwise} \end{cases}$$

The exact transmission state of a UE in a future time-slot is not known with certainty, which means that the variables $\chi_k^{ul}[t], \chi_k^{dl}[t], \chi_k^{idle}[t]$ are random and their statistics are provided by the radio-aware DT. In other words, the radio-aware DT provides us with $p_k^{ul}[t] = Prob(\chi_k^{ul}[t] = 1)$ which is the probability that UE k is transmitting in the UL in time-slot t. Similarly, the radio-aware DT also provides the DL and idle probabilities $p_k^{dl}[t]$ and $p_k^{idle}[t]$, respectively.

Next, it is described in detail how the UE positions are optimized. The received signals in the DL at UE m in time-slot t are defined as

$$r_m[t] = \underbrace{\sqrt{\eta_m^{dl}}\boldsymbol{h}_m^H[t]\boldsymbol{w}_m[t]x_m^{dl}[t]\chi_m^{dl}[t]}_{\text{Desired signal for UE } m} + \underbrace{\boldsymbol{h}_m^H[t]\sum_{k\neq m}\sqrt{\eta_k^{dl}}\boldsymbol{w}_k[t]x_k^{dl}[t]\chi_k^{dl}[t]}_{\text{Interference from other UEs due to spatial multiplexing}}$$
$$+ \underbrace{\sum_k \sqrt{\eta_k^{ul}} g_{km} x_k^{ul}[n]\chi_k^{ul}[n]}_{\text{CCI}} + q[t]$$

Assuming that the BS has exact knowledge of the channel (for simplicity) and is using maximum-ratio precoding, i.e., $\boldsymbol{w}_k[t] = \frac{\boldsymbol{h}_k[t]}{||\boldsymbol{h}_k[t]||}$, the above equation becomes

$$r_m[t] = \sqrt{\eta_m^{dl}}\|\boldsymbol{h}_m[t]\|\chi_m^{dl}[t]x_m^{dl}[t] + \sum_{k\neq m}\sqrt{\eta_k^{dl}}\frac{\boldsymbol{h}_m^H[t]\boldsymbol{h}_k[t]}{\|\boldsymbol{h}_k[t]\|}\chi_k^{dl}[t]x_k^{dl}[t]$$
$$+ \sum_k \sqrt{\eta_k^{ul}} g_{km} x_k^{ul}[t]\chi_k^{ul}[t] + q[t]$$

A lower bound on the average DL throughput for UE m in slot t can be obtained as

$$R_m^{dl}[t] = p_m^{dl}[t]\log_2\left(1 + \frac{\eta_m^{dl}\mathbb{E}[\|\boldsymbol{h}_m[t]\|^2]}{\underbrace{\sum_k \eta_k^{ul}\mathbb{E}[|g_{km}|^2]p_k^{ul}[t]}_{\text{CCI}} + \underbrace{\sum_{k\neq m}\eta_k^{dl}\mathbb{E}\left[\frac{|\boldsymbol{h}_m^H[t]\boldsymbol{h}_k[t]|^2}{\|\boldsymbol{h}_k[t]\|}\right]p_k^{dl}[t] + \sigma^2}_{\text{Non-coherent interference+AWGN}}}\right)$$

The numerator in the above expression is the desired signal power and is proportional to the product of the path-loss and the array gain of the antenna array, i.e., $\alpha_k[t] = \mathbb{E}[\|\boldsymbol{h}_m[t]\|^2]$. The first term in the denominator corresponds to CCI power and the last term is the non-coherent interference that is a combination of AWGN and inter-stream interference from spatial-multiplexing.

Similarly, a lower bound on the average UL throughput for UE m in slot t neglecting self-interference can be obtained as

$$R_m^{ul}[t] = p_m^{ul}[t]\log_2\left(1 + \frac{\eta_m^{ul}\mathbb{E}[\|\boldsymbol{h}_m[t]\|^4]}{\sum_{k\neq m}\eta_k^{ul}\mathbb{E}[|\boldsymbol{h}_m^H[t]\boldsymbol{h}_k[t]|^2]p_k^{ul}[t] + \mathbb{E}[\|\boldsymbol{h}_m[t]\|^2]\sigma^2}\right)$$

With the equations for the UL and DL achievable rate, the UE trajectories can be optimized by solving the following optimization problem:

$$(\boldsymbol{v}_{\text{opt}}[t])_{t=1}^{T} = \arg \max_{(\boldsymbol{v}[t])_{t=1}^{T}} \sum_{t=1}^{T} \sum_{k=1}^{K} \left(R_k^{\text{ul}}[t] + R_k^{\text{dl}}[t]\right)$$

$$\text{subject to}$$

$$\boldsymbol{l}_k[t] \leq \boldsymbol{v}_k[t] \leq \boldsymbol{u}_k[t] \quad \forall\, k = [1, \ldots, K],\ t = [1, \ldots, T]$$

$$\|\boldsymbol{v}_k[t] - \boldsymbol{v}_k[t-1]\| \leq \delta \quad \forall\, k = [1, \ldots, K],\ t = [1, \ldots, T]$$

In the optimization problem, the objective function of the optimization is the sum of the average throughputs of all K UEs over all the T timeslots.

The first set of constraints ensure that the UE is positioned within the coordinates $\boldsymbol{l}_k[t]$ and $\boldsymbol{u}_k[t]$, to ensure that the UEs follow some predetermined trajectory but are free to move in a small area around the planned route to minimize CCI. The second set of constraints ensures that that the difference between the positions of the UE in the current and previous timeslot is less than δ. This is to enforce the practical constraint that the UEs are limited in their movement across each timeslot.

The optimization problem is non-convex and to get a suboptimal solution, a greedy coordinate descent method is utilized where, in each iteration, the position of one of the UEs in one of its timeslots is optimized while keeping the position of rest of the UEs in the remaining time-slots unchanged.

An example scenario where the UEs are UAVs is simulated. 8 UAVs in LOS conditions are considered, placed in a straight line uniformly 500m from the BS. The spacing between the UEs is 25m. The BS is equipped with 64 antenna elements. The UAVs are simulated for a time horizon of $T = 50$ timeslots, with $\delta = 1$s. The UAVs can move a maximum of 12.5m on either side from their starting locations, thereby defines $\boldsymbol{l}[t]$ and $\boldsymbol{u}[t]$.

All UEs transmit/receive in all timeslots and there is no idle UE. In the first timeslot, the probability $p_k^{ul}[1]$ varies linearly from 1 for the left-most UE to 0 for the right-most UE. The UEs in between take values from 0 to 1. As time progresses, the left-most UEs change from UL to DL while the right-most UEs change from DL to UL. At the 50$^{\text{th}}$ time slot, the probabilities are opposite to the values in the first timeslot, i.e., $p_k^{ul}[1]$ is 0 for the left-most UE and 1 for the right-most UE. The remaining UEs have values in between 0 and 1.

Figure 3-4 and Figure 3-5 plot the UL and DL spectral efficiencies (SEs) of the UEs across all timeslots. It is evident that there is a clear improvement in the DL SE performance with the trajectory optimization thanks to a reduction in the CCI. However, this is at the cost of a marginal drop in the UL SE, which is due to the fact that the optimization problem above maximizes the sum of the UL and DL SEs. To control this trade-off, the objective function can be replaced with a weighted sum. In addition, the UL SE in Figure 3-5 exhibits a stepwise behaviour because this is a contrived examples with UAVs located on a straight line and with discrete UL/DL transmission probabilities. Consequently, the UL received power and interference levels are also discrete. However, the same is not true in the downlink, since neither the signal or interference power constrained to a discrete set.

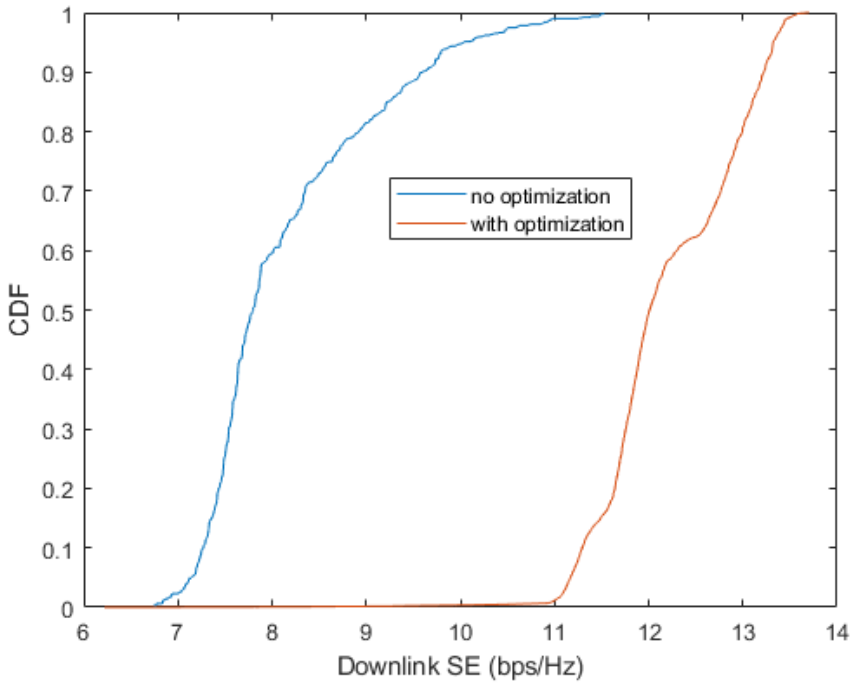


Figure 3-4: CDF of downlink SE over all UEs and timeslots.

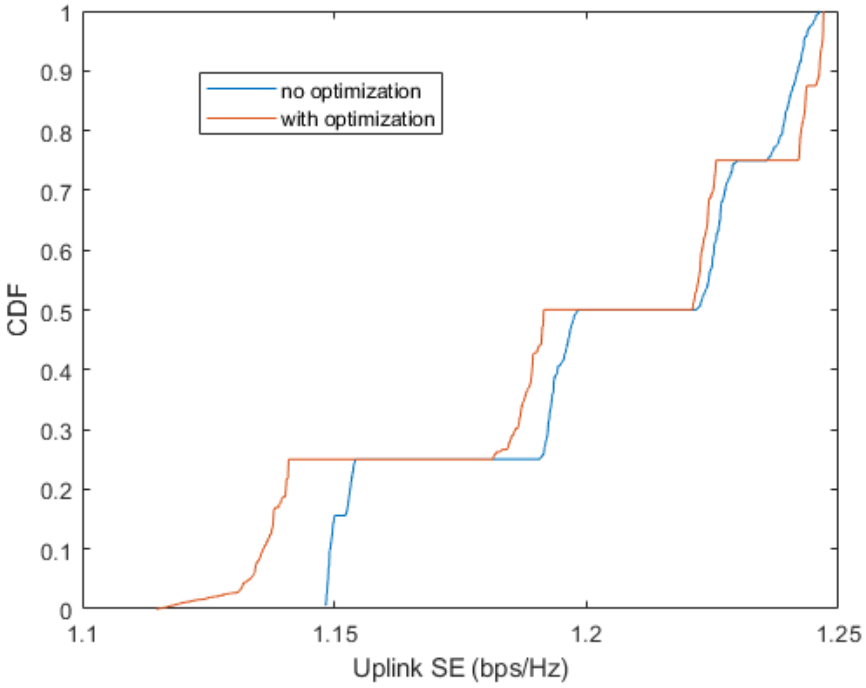


Figure 3-5: CDF of uplink spectral efficiency over all UEs and timeslots.

3.3 Functional-split-aware trajectory planning for industrial vehicles

Autonomous or semi-autonomous vehicles are a very promising option to provide a wide range of services, such as city monitoring, packet delivery, efficient human transportation, communication

relays, etc. Frequently these vehicles require good communication links to their ground stations or between themselves while they drive or fly, which can be in principle accomplished by mobile radio access networks as part of the vehicle-to-everything (V2X) perspective.

Mobile coverage in the inner region of the traversed cells often allows for good-quality connections (high data rates, low latency, low packet-loss probability). Conversely, at the cell edges inter-cell interference may negatively affect connection quality. Cell coordination is an effective way to reduce inter-cell interference at the cell edge, especially in networks with high frequency reuse factors. Nevertheless, the effectiveness of cell coordination is heavily influenced by the centralization level of the protocol stack of the coordinating cells. When the processing of two cells is highly centralized, sophisticated interference mitigation methods can be used, such as coordinated multipoint or joint transmission and reception. If at least one of the cells is not centralized, however, coordination is hindered.

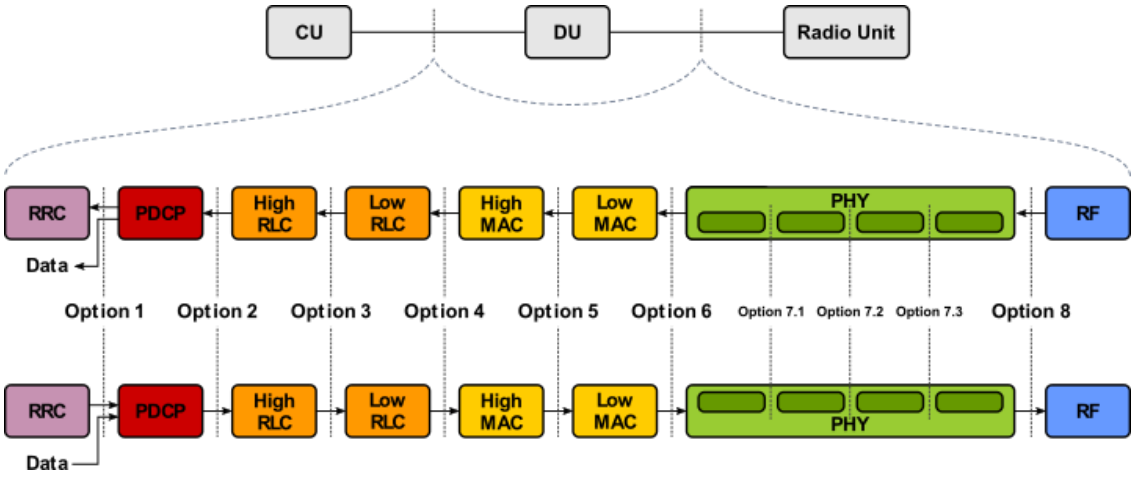


Figure 3-6: Depiction of several functional split options for 5G/6G mobile networks.

The centralization level of a base station is determined by its functional split, which divides the processing functions between those running at the distributed unit (DU) and those running at the centralized unit (CU). The functional split affects the ability of applying interference mitigation techniques between neighbour cells. High centralization levels, such as C-RAN (shown as option 8 in Figure 3-6) or Intra-PHY (options 7.x) enable advanced interference cancellation techniques, such as joint transmission and reception. Less centralized options, such as MAC-PHY (option 6), still allow for less sophisticated techniques, like coordinated scheduling and/or beamforming. Low centralization, as in the PDCP-RLC split (option 2) only allow slow and basic interference mitigation.

Current radio access networks often lack the resources for centralizing the processing of all their base stations. They are limited by the bandwidth of the fronthaul and midhaul networks connecting their centralized, distributed, and remote units, as well as by the computational capacity of each of these units, and in some cases even by the latency introduced by these fronthaul and midhaul networks. As a result, it is only possible to centralize (fully or partially) a subset of all base stations, resulting in pairs of coordinated (fully or partially) and uncoordinated cells.

Given these limitations, current state-of-the-art approaches propose to dynamically adapt the centralization level of the base station to the network conditions: the number, activity, and geographical distribution of users. For each base station, a functional split is defined so that some of its functions are centralized, and the remaining functions are distributed. This functional split is subject to change dynamically if the network operator detects that it is not optimal anymore, because of changes in the network conditions.

Consequently, in a radio access network that dynamically adapts its functional split, the connection quality at the edge between two cells depend on the instantaneous centralization level, so it cannot be known beforehand by the users. However, for a vehicle it would be very beneficial to know which cell edges provide the best connection quality, so that its trajectory planning can take it into account.

A mechanism for the vehicles to request this information to the network, and for the network to answer succinctly yet comprehensively is proposed. This information can be then used by the path finding algorithm at the vehicle to produce path that avoids areas with low-quality connections, which may degrade the service provided or required by the vehicle.

An industrial mobile radio access network (RAN) consisting of multiple base stations is considered, which are connected to an external data network via a mobile core network. Each base station is split into at least two units: a centralized unit (CU) and a distributed unit (DU). The RAN functions deployed at either DU or CU are not fixed, but the operator can change the functional split at will, according to the instantaneous network conditions to modify the overall centralization level of each base station. Further divisions in the protocol stack are possible, for instance to introduce a remote unit (RU). The CU is connected to the core network, which in turns connects the base station to the external data network (such as the internet).

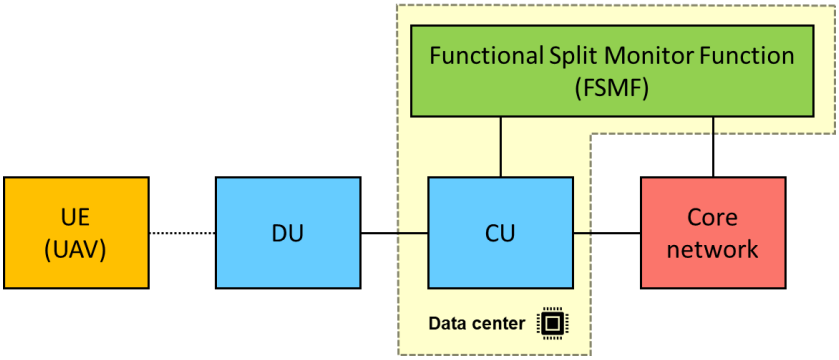


Figure 3-7: Simplified example architecture of the mobile access network. The FSMF is deployed at the same data center as the CU in this example, but other configurations are also possible.

In addition to already existing units, a separate Functional Split Monitor Function (FSMF) is defined, which keeps track of changes in the centralization level (functional split) of all base stations, and performs the required calculations required to inform the UE about the current RAN state. This FSMF can be collocated with the CU and/or the core network, or it can be separately deployed at a location that has access to the updated state of all base stations. The overall RAN architecture is depicted in Figure 3-7.

Each base station serves one or more mobile cells, which are spatially separated but border each other at the cell edges, where inter-cell interference is especially noticeable. Frequency reuse is such that neighbour cells may experience collisions in their used time-frequency resources. The functional split, and hence the centralization level, influences the interference mitigation capabilities for each pair of base stations since it affects the effectiveness of the coordination between base stations. That is, base stations whose lower layers are centralized can implement advanced techniques such as coordinated transmission or reception, which is frequently not possible between base stations with predominantly distributed functions.

Apart from other types of user equipment (UE), the radio access network provides uplink and downlink connectivity to AGVs. These vehicles can move from one point to another as needed by the factory owner, e. g. to move parts between factory locations or perform monitoring. When moving, AGVs may require minimum levels of certain connection performance indicators, such as minimum or average data rate, maximum latency, packet loss probability, etc. These indicators are affected by the signal power and signal-to-interference-and-noise ratio (SINR) associated to each location, which are in turn affected by the centralization levels of neighbouring base stations.

At some point, motivated by manual or automatic input, the AGV needs to move from its current location (origin) to a new destination. Origin and destination are distant enough such that one or more mobile cell edges will be crossed when moving, thus requiring one or more handovers to perform.

Before calculating the optimal path, informs the DU indicating the geographical coordinates of its current position and its intended destination. The DU receives the message and forwards it to the FSMF. Based on the origin and destination coordinates, the central entity computes a considered region containing both points. This considered region should contain all reasonably possible paths between origin and destination. For instance, this region can be the circle whose centre is the midpoint between origin and destination and whose diameter is the distance between origin and destination plus an additional margin.

The FSMF then retrieves the array of centralization levels of the base stations serving the cells fully or partially covered by the considered area (e.g., 50 m). The FSMF maintains a database with the approximate cell boundaries of the radio access network it manages. This cell boundaries can be modelled as a polygonal line, whose corners are stored as 2D (or 3D) coordinates and their connections are represented by and adjacency matrix, adjacency list, incidence matrix, or other alternative graph representation. Consequently, the FSMF retrieves the set of corners of cell boundaries that are contained within the considered region. For each edge in the cell boundaries graph, the FSMF retrieves the pairs of base stations whose cells are separated by that edge. If they are different base stations, the central entity compares its centralization level and assigns a *Risk of Interference* (RoI) value to the edge associated with the lowest centralization level of both cells. This RoI may be an integer value within a convenient range (such as 0 to 7), and can be calculated as a direct reescalation of the centralization level, as a function of the expected degradation in the SINR, maximum data rate, etc.

From the retrieved information, the FSMF answers the UE with two main information elements. First, it includes a matrix or list of coordinates corresponding to relevant corners of cell boundaries. Second, it includes an adjacency list of the corners of each edge together with the RoI value of that edge. This information can be also structured as a matrix. An example of the values of these two information elements for the cell boundaries depicted in Figure 3-8 are shown in the following.

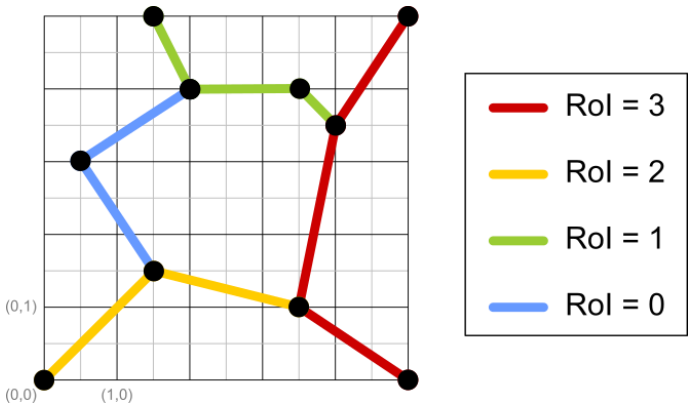


Figure 3-8: Example of cell boundaries with RoI values depicted with different colours.

The first column in $Corners$ represents the X dimension, whereas the second column represents the Y dimension:

$$Corners = \begin{bmatrix} 0 & 0 \\ 1.5 & 1.5 \\ 0.5 & 3 \\ 2 & 4 \\ 1.5 & 5 \\ 3.5 & 4 \\ 3.5 & 1 \\ 4 & 3.5 \\ 5 & 5 \\ 5 & 0 \end{bmatrix}.$$

The first two columns in $Adjacency + RoI$ represent the indices of the connected corners, whereas the third column represents the RoI value of the corresponding edge:

$$Adjacency + RoI = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 6 & 2 \\ 2 & 3 & 0 \\ 3 & 4 & 0 \\ 4 & 5 & 1 \\ 4 & 6 & 1 \\ 6 & 8 & 1 \\ 8 & 9 & 3 \\ 8 & 7 & 3 \\ 7 & 10 & 3 \end{bmatrix}.$$

The AGV now parses the received response and uses the RoI to compute the optimal path that features the best combination of RoI, connection quality, cell load, regulatory constraints, time of flight, energy consumption, obstacle avoidance, etc.

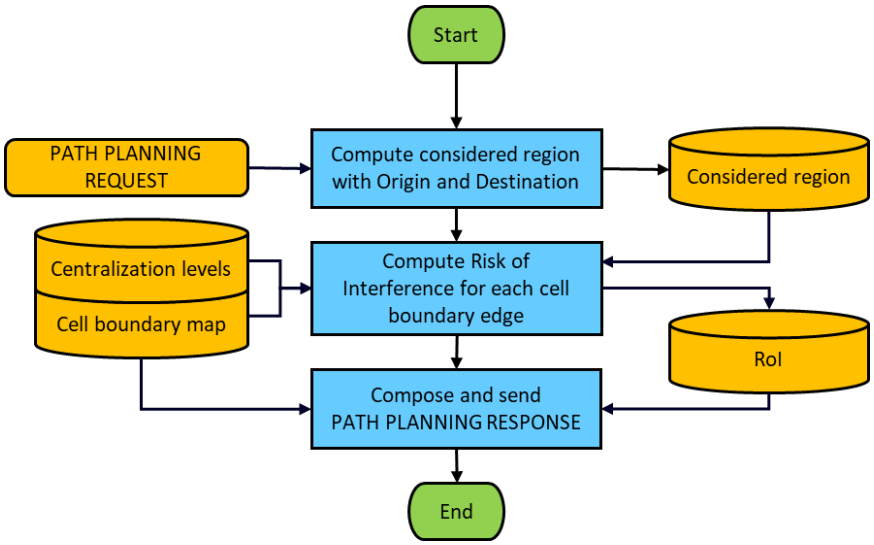


Figure 3-9: Flow diagram of the operation of the FSMF after the reception of a PATH PLANNING REQUEST message from a UE.

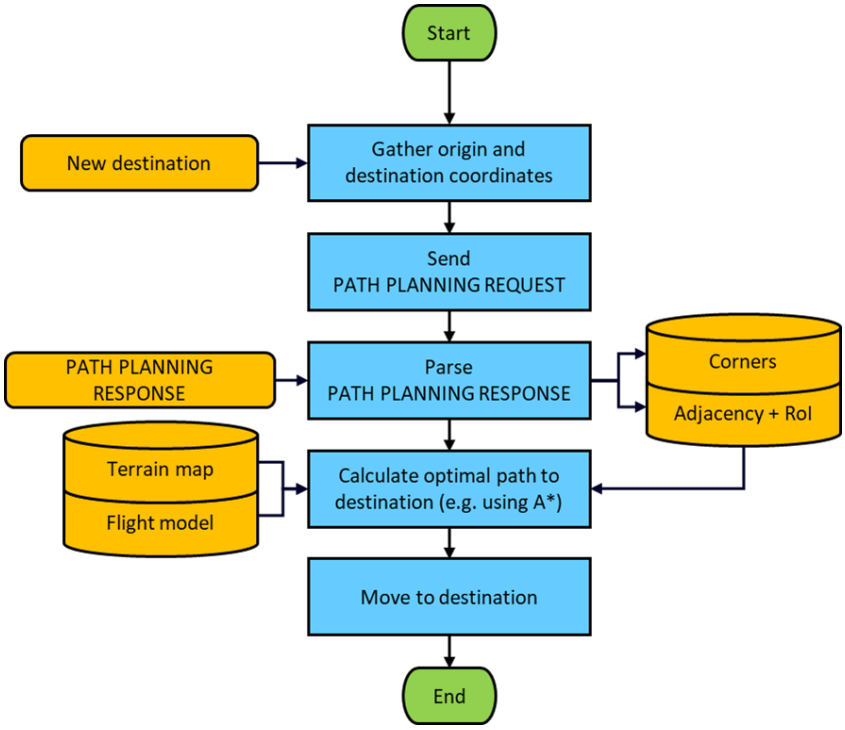


Figure 3-10: Flow diagram of the operation of the UE during the procedure of calculating the optimal path.

A modified state-of-the-art algorithm, such as Dijkstra or A* search, can be used to find the path with the lowest accumulated RoI, or with the lowest combination of accumulated RoI and other metric, such as distance, curvature, etc.

For this, the UE can convert the received Adjacency+RoI matrix into a dual graph, whose nodes are the midpoints of each edge in the Adjacency matrix and the edges connect those nodes that are at the edge of the same cell. In addition, each point as an associated RoI value, which is the RoI of the corresponding edge of the primal graph as indicated in the Adjacency+RoI matrix. In addition, the dual graph contains one node at the initial position of the vehicle and another one at the destination, and the required edges to connect both these nodes to all the dual nodes (midpoints of primal edges) of the cell they are in.

Once this dual graph is constructed, finding the path of lowest accumulated RoI is a matter of applying a shortest path finding algorithm (such as Dijkstra or A*) to the dual graph, where the "distance" associated with each dual edge is the RoI corresponding to the end node, or a combination of RoI and other cost or distance metrics. A summary of the procedure from the point of view of the FSMF and the UE are shown in Figure 3-9 and Figure 3-10, respectively.

A depiction of a RoI-optimal path that the vehicle from Figure 3-11 could obtain is shown in Figure 3-12. Without RoI information, the vehicle would have preferred a shorter path (depicted as an orange dashed line) that crosses cell edges of uncoordinated base stations, which may result in low-quality connections. Notice how the RoI optimal path prefers traversing between neighbouring, centralized base stations, at the expense of possibly longer driving or flying times.

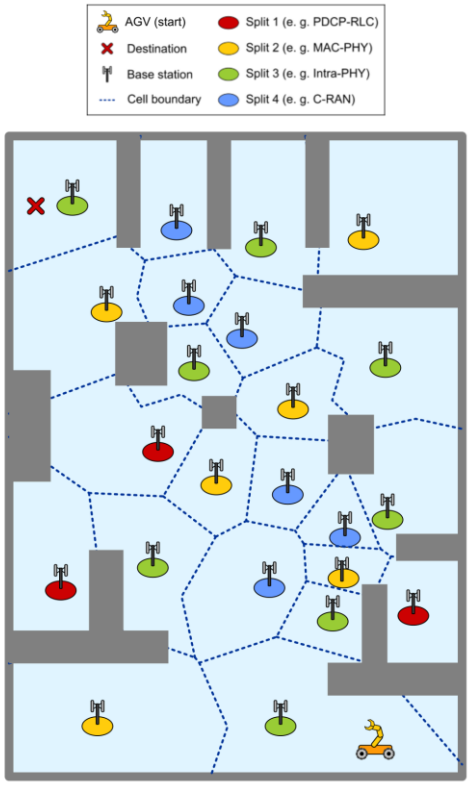


Figure 3-11: Example of a considered region in a radio access network implementing four functional split options and an automated guided vehicle (AGV) that intends to traverse it.

Figure 3-12: Depiction of a RoI-optimal path calculated by the vehicle from the RoI values at the cell edges, which minimizes the crossing over cell edges with high interference risk.

3.4 O-RAN compliant resource provisioning for Federated Learning

Large-scale wireless emulation is gaining momentum nowadays, thanks to its potential in the development and deployment of advanced uses cases for IoT devices in next-generation wireless networks. The development and testing of a wireless application, such as massive MIMO, wireless signal beamforming, and Vehicle-to-Everything (V2X) communications, especially at a large scale and when dealing with mobile nodes, faces several challenges that cannot be solved by simulation frameworks alone. Thus, massive-scale channel emulators are emerging, enabling the emulation of realistic scenarios which leverage real hardware and physical radio signals, exchanged in a dedicated testbed environment thanks to Software Defined Radios (SDRs).

A novel framework is proposed, shown in Figure 3-13, for the design and generation of channel emulation scenarios starting from real mobility traces, either generated by means of dedicated tools, or collected on the field. With the aim of validating the pipeline proposed as part of our framework, the framework is tested in the world's most powerful, one-of-kind wireless network emulator, Colosseum [BJP+21]. In detail a full-fledged vehicular 5G scenario with 13 vehicles moving around an area of 1 km^2, starting from a real mobility dataset called SAMARCANDA [RZM+22] is created in Colosseum.

Thanks to the combination of 128 Standard Radio Nodes and the Massive digital Channel Emulator (MCHEM) backed by an extensive FPGA routing fabric, not only the channel between transmitters and receivers but also the typical effects of the wireless propagation environment can be emulated. Thanks to our framework, it becomes possible to easily create and deploy both cellular-based and Wi-Fi-based scenarios to large-scale emulators. As next step a *Federated Learning framework* in a V2X scenario, O-RAN compliant inside the COLOSSEUM emulator is to be implemented and optimized.

Each client should be able to perform different tasks, also heterogeneous, including training and inference of a Machine Learning model, where the objective is to do image/instance segmentation. The resource allocation problem will be at the core of our framework, keeping into consideration both communication and computation resources (i.e., Physical Resource Block and CPU). The objective function will be minimizing the end-to-end latency and used bandwidth to reach a certain accuracy of the trained model in a federated way and the time for retrieving the most updated model for inference tasks.

To validate the mobility scenario, created with our proposed framework in Colosseum, several metrics were measured, including the Round-Trip-Time (RTT) while the vehicles move around the coverage area, and the measured Downlink Signal-to-Noise-Ratio (SNR) in time. Then, these metrics have been compared to the position of each vehicle and to their distance from the gNB, leveraging the positions available in the SAMARCANDA dataset. As Colosseum is optimized to work over a frequency band around 1 GHz, we selected it as carrier frequency.

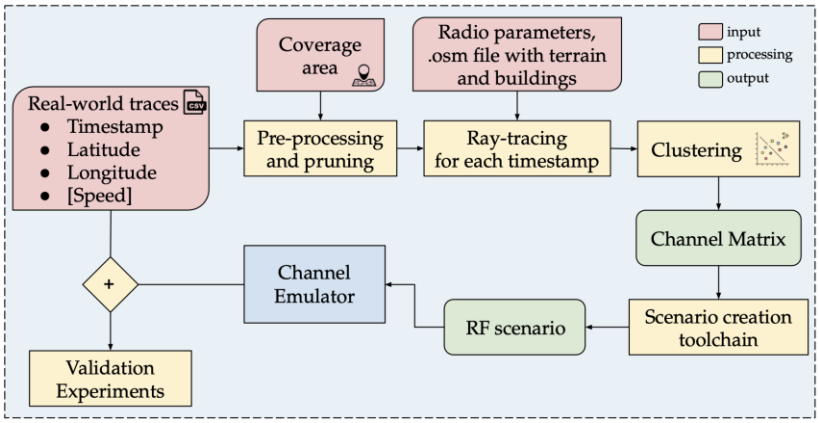


Figure 3-13: Proposed Channel Emulation Framework

Figure 3-14 shows the distance from the gNB as a function of time, comparing it with the RTT between the vehicle and the base station, measured thanks to the ping tool. The dotted lines delimit the time period in which the vehicle is located within the coverage area, while the vertical black lines delimit the moments in time included in the dataset.

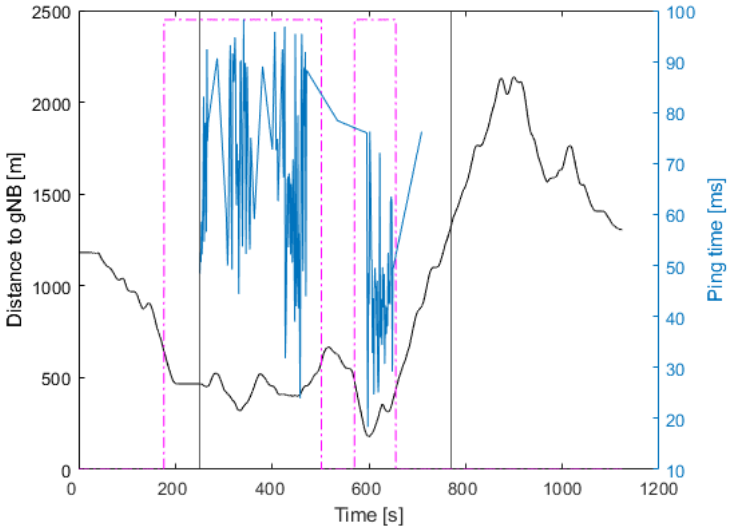


Figure 3-14 Distance from the gNB over time

Figure 3-15 reports instead the SNR as a function of time, compared with the distance from the gNB. Consistently with the previous plot, it shows how a higher SNR is measured as the distance from the gNB decreases, with non-linearities and oscillations due to the effects of terrain and buildings, realistically modelled thanks to our framework and to the Colosseum emulator.

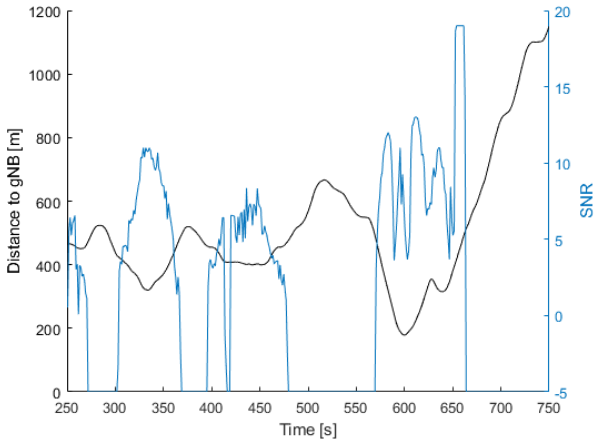


Figure 3-15 SNR and distance to gNB over time

3.5 Ambient backscatter communication: recycling radio waves

In [Hexa-X D7.2], a new method of using radio resources by recycling radio waves, called the *Crowd-Detectable Zero-Energy-Device (CD-ZED)*, a new use-case called "smart tracking out-of-thin air", and experimental results have been reported, and are briefly recalled hereafter. The CD-ZED technique aims at participating to the sustainable development of IoT based services in future 6G networks [URB+21]. As illustrated in Figure 3-16(a), the CD-ZED backscatters DL ambient waves (transporting data and/or control signalling to broadcast its ID number (#77). A smartphone closed to the CD-ZED and connected to the network detects the CD-ZED message simultaneously with the ambient signals [PBR+21] [Ora21]. As illustrated in Figure 3-16(b), in the "smart tracking out-of-thin air" service, the network tracks the location of an asset on which has been stuck a ZED, thanks to the contacts of the ZED with the crowd of geo-localised smartphones connected to the network. Note that the CD-ZED and tracking use-case can be implemented with UL ambient waves instead of DL waves [PRB+21]. Finally, as the smartphone already has a GPS positioning error, associating the position of the smartphone to the position of the tag, adds up an additional positioning error due to the tag-to-smartphone reading distance. As illustrated by Figure 3-16(c), summarizing D7.2 [Hexa-X D7.2] experimental measurements (in outdoor and indoor, sub-urban environment, with high rise buildings making GPS positioning challenging), this additional positioning error is low (in the order or below 10% of the GPS positioning error). Note that [PRB+22] reports additional results for on highways and in ultra-dense urban environments.

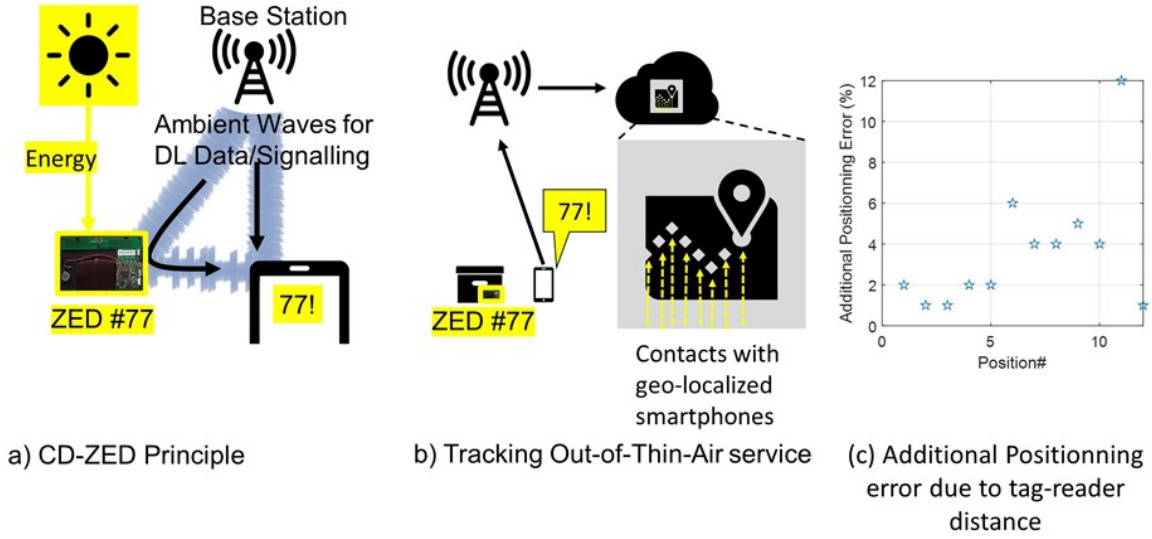


Figure 3-16: CD-ZED principle (a), tracking out-of-thin air service (b), experimental results (c).

In this current deliverable, new results on the following aspects are reported: the validity of the self-powered device concept, the expected impact on mobile network standards, the impact on the smartphone activity.

Validity of self-powered device concept: A measurement-based model of the energy-consumption of the current ZED prototype has been derived [PBR+22]. As illustrated in Figure 3-17(a), the ZED energy-autonomy in days (assuming the battery of 3V initially full) depends on two parameters: the ZED sending mode and the level of energy that can be harvested. In Figure 3-17(a), the ZED prototype can send the 96 bit message permanently (in permanent sending mode) or every 10 seconds (in spaced sending mode), consuming 250 microwatts 24 hours a day, or 54 microwatts 24 hours a day, respectively. Figure 3-17(a) shows that the current ZED prototype could work indefinitely, transmitting messages 24 hours a day (therefore, even at night), by spacing its message transmissions by 10 seconds (and spending 54 microwatts), on condition that it is illuminated 10 hours a day by a bright light

(corresponding to 150 microwatts harvesting), and assuming that the 3V battery is initially charged. In practice, at the end of a trajectory, once arrived at its destination point, the ZED would be sent back to the logistic company, and could be "switched off" (i.e., with the micro-controller switched off) to let the ZED harvest solar energy and charge the battery without discharging it, until it is reused again.

Impact on mobile network standards, service and architecture aspects: An initial impact of such system in mobile network standards has been studied and reported in [PRB+22]. The study took, *as an example*, the 4G standard, as it is an existing and mature network. Regarding service and architecture aspects, to support the out-of-thin-air tracking service, a CD-ZED server would run at the network side, in parallel to a CD-ZED app at the UE side. As illustrated in Figure 3-17 (b), the CD-ZED service would be connected to the Evolved Packet Core (EPC). Upon detection of a ZED by a user equipment (UE), the CD-ZED app of the UE would report the ZED ID to the CD-ZED server, through the BS and the EPC. The CD-ZED app and the CD-ZED server would call the 3GPP localisation service (LCS) and/or a global navigation satellite systems (GNSS) to monitor the UE positioning. Then, the CD-ZED server would associate the ID of the reported ZED with the localisation of the reporting UE, to track position the ZED. Finally, the ID of the UE would be deleted.

Impact on mobile network standards, MAC and PHY layer aspects: Regarding physical and MAC layer aspects, to support an efficient detection of the ZED by the UE, [PRB+22] proposed to exploit existing 4G pilot signals and existing channel estimation blocks in UEs. When connected to the network, i.e. in the standardized mode called Radio Resource Control (RRC) connected mode, the smartphone would monitor the 4G BS following common signals (sent within a 1 ms sub-frame) [36.211]: downlink pilots called Common Reference Signals (CRS) sent in every sub-frame, the Primary Synchronization (PSCH) and the Secondary Synchronization Channels (SSCH) sent every 1 over 5 sub-frames, the Broadcast Channel (BCH) sent every 1 over 10 sub-frames, the Physical Downlink Control Channel (PDCCH) sent every sub-frame. By using channel estimates instead of energy measurements, the UE can detect the influence of the tag on the channel, in terms of magnitude variations and phase shifts (instead of only in terms of only magnitude), and also be robust to bursty data traffic. Indeed, the energy detector fails to detect the ZED when the variations of the received energy are more due to the data traffic variations than the ZED's influence. In a recent study, an example of pilot-based detection (based on CRS) has been successfully tested in [RWL+22]. It exhibited a better performance than the energy-detection receiver used in previous studies. Also, in [RWL+22], the tag's influence on the propagation channel (due to backscattering) is a Doppler frequency shift that is cautiously chosen in the 200 to 1000 Hz range. This range ensures that the ZED does not to degrade the quality of channel estimation and equalization, and that it hence protects the currently on-going ambient communication quality. This range also ensures that the ZED's influence remains separable from usual mobility phenomena (such as pedestrians or cars). Now, one can wonder whether smartphones are connected to the network frequently enough. Figure 3-17 (c), illustrates the connections to the 4G network of a smartphone which is left, without any human interaction for two hours, and with only a google account installed, as an application. The google account generates background traffic and RRC connections. Figure 3-17 (c) plots the durations of the RRC sessions as a function of the delay between the start of current RRC session and the start of the next session. It appears, that even an "untouched" smartphone is connected for 10 to 50 seconds to the network at least every 10 minutes. Therefore, even an "untouched" smartphone has frequent occasions to read a ZED.

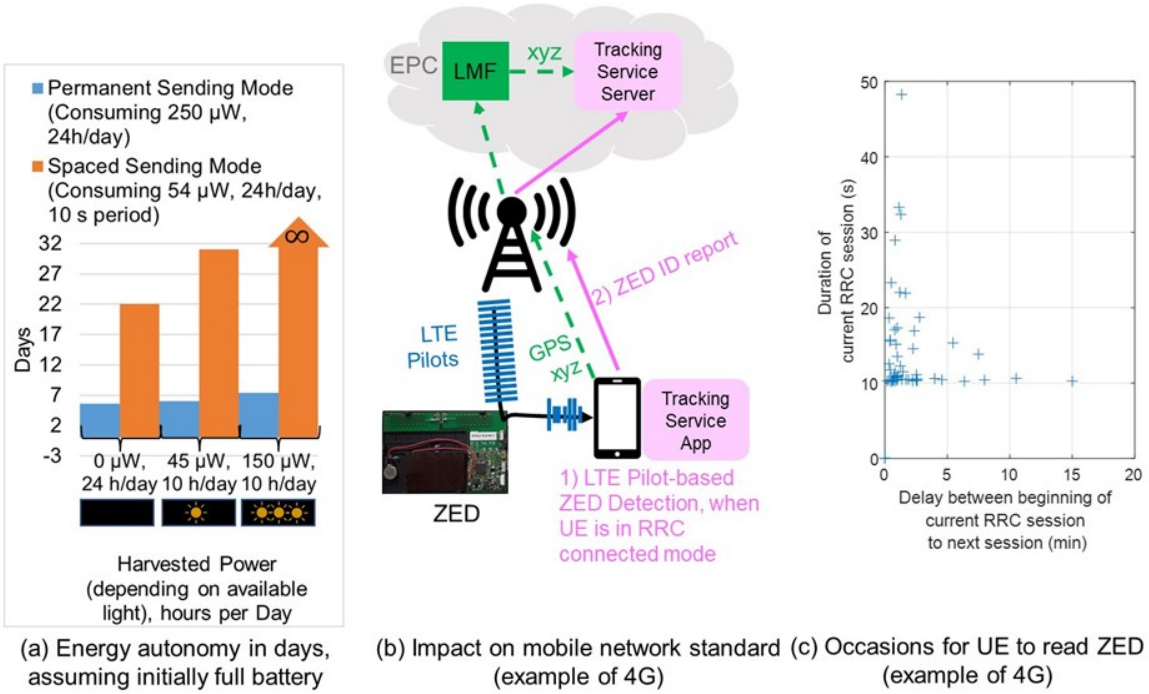


Figure 3-17: ZED energy-autonomy (a), impact on mobile network standard (b), occasions for UE to read ZED (c).

Conclusions: Measurements conducted during Hexa-X project show that the precision of the positioning of the ZED is similar to the precision of the positioning of the smartphone, as the additional positioning error due to the ZED-to-reader distance is negligeable. An energy consumption analysis of the ZED prototype conducted during Hexa-X project shows that the ZED can be autonomous. Some small impacts on mobile network standards have been identified, at service and architecture level, and also at PHY and MAC level, on the example of 4G network as it is an already existing and mature ambient network.

4 Dependability in I4.0 environments

In Hexa-X, dependability is defined as the "ability to perform as and when required" and identified as a critical KPI area that contains numerous related KPIs, including but not limited to availability, reliability, safety, integrity, and recoverability [Hexa-X D1.3]. These underlying KPIs serve as an indicator for the Hexa-X key value *trustworthiness*. In this section, efforts and results are described, which revolve around increasing application dependability through means that extend the thoroughly investigated and well-known ultra-reliable low-latency communications (URLLC) approach. The section is divided into two major directions:

- **Technical enablers** increase the dependability in I4.0 environments through increasing network dependability. In total, three different approaches are presented.
- **Communication-Computation-Control-Codesign** increases the dependability by attempting a joint modelling approach in which interdependencies between the "Co's" are identified and co-optimised, e.g., with respect to higher overall spectral efficiency, or reduced energy consumption.

Additionally, the reader is referred to Section 6 for a description of a practical implementation for achieving dependability in an I4.0 context by identifying errors and acting upon resolving them.

4.1 Technical enablers for dependability beyond URLLC

In essence, well-known URLLC research investigates how a wireless connection can be made as low-latency and reliable as possible, i.e., how a wireless link must be designed to resemble a wired connection as close as possible. In this section, technical enablers beyond the wireless link are investigated, including (1) an improved radio resource management through digital twinning in a network setting, (2) UAV data relays in an IoT network for improving the Age-of-Information (AoI) and energy consumption of sensors, and (3) provision of E2E latency and throughput guarantees for both control and data planes for ensuring that the system will behave in the foreseen way, and (4) network data analytics assisted AI operation for joint application and RAN optimization for future factory network.

4.1.1 Radio Resource Management with Radio-Aware Digital Twin

A Digital twin is an up-to-date digital representation of a physical object. A radio-aware digital twin as a digital representation of the radio propagation environment is proposed, which can be used to predict the link condition between a transmitter and receiver pair. The concept is most suitable for controlled environments like private factories, and warehouses, where the environment is fully under our control; for instance, spectrum usage, UE and BS types (including their position and movement), and traffic sources. The environment can also be enhanced to have better and more favourable radio conditions, for example by choosing the right materials or even by using smart materials like intelligent reflecting surfaces.

In a radio-aware digital twin, in addition to the classical DT features, a set of *relevant* details of the wireless network is modelled, such as:

- The PHY and MAC layers of the access points and mobile devices in the radio network.
- The properties of these transmitting/receiving radio nodes (e.g., their current position and possible pre-defined trajectories).
- Other details such as BS and UE capabilities, beam patterns, RRM algorithms etc. (retrievable from RAN).
- The radio link quality in the whole network with some resolution, for example average Reference Signal Received Power (RSRP) or SINR levels on 5mx5m grid. This data can be stored (and regularly updated) in a database like a radio environment map (REM).
- A mapping between the radio-propagation environment and relevant KPIs/KVIs such as throughput and reliability.

The primary benefit of Radio-Aware DT is that it enables us to predict many things in advance in an anticipatory and pro-active manner. For example, in the case of URLLC, reactively acting upon occurrence of errors might not be practical (when it is often already too late, there might not be a delay budget for retransmissions). In contrast, one can prepare for possible errors in advance. For example, one might be able to predict that a moving AGV will experience very bad quality radio link after 5 meters so one can start preparing for new beams, allocating more or different resources (even on different frequency bands), etc.

The Radio-Aware DT maintains an accurate and dynamic REM. This REM is updated constantly with a combination of enhanced measurements and advanced propagation prediction engine (e.g., ray-tracers). This update procedure makes the radio-aware DT requiring connectivity resources. In general, radio-aware DT has high situational awareness – especially in controlled environment (like private factory) where it knows the current and future positions of the UEs, and even what kind of traffic they are going to send (for example some sensors send fixed sized packets with fixed intervals). By knowing the traffic patterns, the interference patterns can be predicted and the SINR can be estimated in a certain location. For creating the overall situational awareness, many other techniques can be utilized which are studied in Hexa-X, for example ML/AI (c.f. [Hexa-X D4.1]), high-accuracy positioning (c.f. [Hexa-X D3.1]) and simultaneous sensing and communication (especially with sub-THz).

4.1.1.1 Minimizing overhead when training a radio-aware DT

As mentioned earlier, a radio-aware DT requires an accurate model of the propagation environment to be practically useful, and the DT and the REM database also need to be constantly updated to take changes in the environment into account. Conventional approaches to model the radio propagation environment such as ray tracing can be inaccurate (because they do not accurately capture all modes of electromagnetic propagation) and computationally intensive (such as finite element methods) limiting its applicability in real-time operation. Data-driven methods with machine learning offer a good trade-off between modelling accuracy and computational complexity. Such methods will be the focus of our work in this sub-section.

Machine learning models for a radio-aware DT require large amounts of labelled training data that need to be obtained from measurements. Such data is also required to keep the REM database updated. Making these measurements require reference symbols such as UE-specific demodulation reference symbols (DMRS) or channel state information reference symbol (CSI-RS) to be transmitted which increases the overall overhead. This overhead, which is on top of the overhead necessary for communication, may become prohibitively large in time-varying propagation environments with a large number of sensor nodes such as warehouses or ports where measurements need to be made regularly.

These measurements are expected to be performed in the downlink (DL) (with measurement reporting in the UL) since AGVs and UAVs may be equipped with several cheap narrow-band RX-only radio frequency (RF) chains (in addition to a single TX RF chain) for the sole purpose of simultaneously generating measurement data in a wide variety of antenna positions and orientations. Consequently, uplink (UL) measurements (with UL/DL reciprocity at BSs) may not be a viable option to generate labelled training data since this would necessitate expensive wideband transmit chains that need to be sounded separately, thereby increasing the overhead. Note here that measurements made in the DL are returned to the network in the UL.

In this subsection, a method is proposed to obtain labelled training data for a radio-aware DT without requiring reference symbol transmission for this purpose.

There are several approaches in literature and supported by the standards to generate such data. For example, UE-specific CSI-RS and/or DMRS may be transmitted by the network [38.211]. The UE may then use these reference symbols to perform measurements and send them back to the network. However, this approach requires additional overhead that scales with the number of UEs in the network.

Alternatively, minimization of drive tests (MDT) is a standardized feature in 3GPP LTE/NR [37.320] with which it is possible to make the normal UEs to collect certain network KPIs (like Radio Link Failures) which are then reported back to network, including UE positions. This method however

requires additional reference symbols to be transmitted for measurements, in addition to the reference symbols that are used for communication.

Lastly, analog channel state information (CSI) is a well-known concept in prior art where the UE amplifies and forwards the unquantized received observations back to the network [DN06, MYG20]. While this method also has the feature of not requiring additional reference symbol transmission for updating the DT, the analog symbols that are received at the BS are corrupted with additional noise and interference. Consequently, the quality of the labelled training data for the DT may be poor.

In developing a method for generating labelled training data, the following observations were made:

- Training the radio-aware DT is a latency insensitive task: Since the focus is on generating labelled training data for an ML model of the propagation environment, latencies of the order of a few seconds are tolerable.
- The payload symbols in the downlink (PDSCH/PDCCH) are known to the network. If the UE samples the PDSCH and PDCCH transmissions, quantizes them, and feeds these quantized samples back to the network, the network can treat the payload symbols as pilots.

The proposed approach is shown in Figure 4-1: Proposed approach to generate labelled training data and has the following steps:

- The network provides the location of resource elements (REs) in PDSCH or PDCCH over which the UE is expected to make measurements.
- The UE quantizes the received samples and waits for the traffic in the network to drop. Once the network traffic is low, it forwards the samples back to the network along with the location at which the measurements were made.
- Since the network is aware of the payload, the network can estimate the channel from the forwarded samples.
- The network can use the location information and the channel estimate as labelled training data.

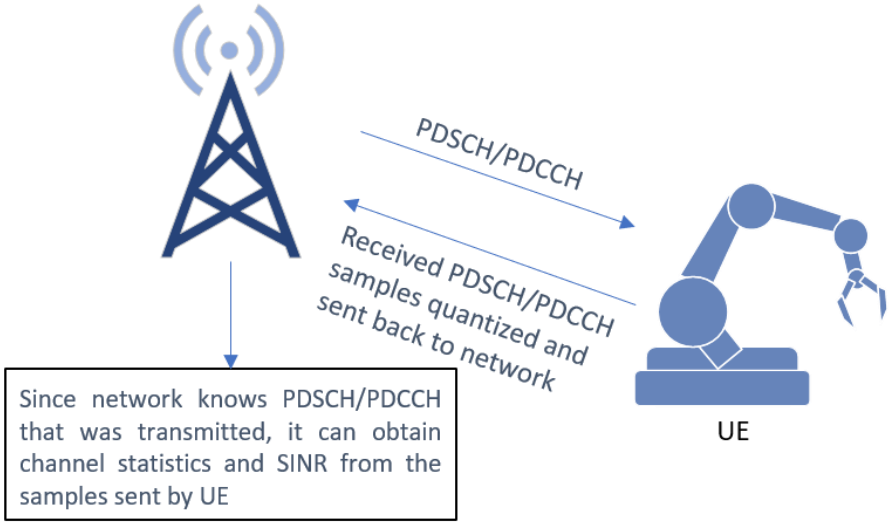


Figure 4-1: Proposed approach to generate labelled training data.

This approach can also be described mathematically. Consider a single-antenna UE receiving payload data in the DL and assume that the network has a total of L cells, which includes a serving cell that the UE is connected to and $L-1$ interfering cells (there could be more than L cells in the vicinity of the UE, but it is assumed that interference from only $L-1$ of them are above the noise floor). In addition, the value of L is small (usually less than 10).

The n^{th} received symbol at the UE can then be written as

$$y_n = \sum_{l=1}^{L} h_l \, x_{ln} + w_n = \boldsymbol{x}_n^T \boldsymbol{h} + w_n$$

where h_l is the channel between the l^{th} BS and the UE, and x_{ln} is the symbol transmitted by BS l and w_n is the additive white Gaussian noise at the UE. We denote $\boldsymbol{x}_n = [x_{1n}, \dots, x_{Ln}]^T$ and $\boldsymbol{h} = [h_1, \dots, h_L]^T$. Assuming a block fading channel and aggregating $P \geq L$ observations in this coherence block $\boldsymbol{y} = [y_1, \dots y_P]^T$, one gets:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{h} + \boldsymbol{w}$$

at the UE, where $\boldsymbol{X} = [\boldsymbol{x}_1^T, .., \boldsymbol{x}_P^T]^T$. The UE quantizes $\boldsymbol{y}$ and feeds it back to the network as $\boldsymbol{y}_Q$. Now, an estimate of $\boldsymbol{h}$ can be obtained as $\widehat{\boldsymbol{h}} = \boldsymbol{y}_Q \boldsymbol{X}^+$, which is accurate if $\boldsymbol{y} \approx \boldsymbol{y}_Q$. Here, $\boldsymbol{X}^+$ is the Moore-Penrose pseudo inverse of $\boldsymbol{X}$. Alternative methods for estimating $\widehat{\boldsymbol{h}}$ exist, such as minimum mean-squared error or a deep neural network when given prior statistical information in the form of a probability distribution or a data-driven model.

The proposed approach has the following benefits:

- Lower DL overhead since no dedicated reference signal transmission is required for the purpose of training a radio-aware DT.
- Note that this method results in an increased UL overhead. However, since there is no stringent latency constraint on transmitting the samples back to the network, they can be transmitted when the UL network traffic is low or when the UE is close to sub-THz access points that have large amount of bandwidth available.
- This method allows for the possibility to generate a large amount of labelled data with minimal DL overhead.
- Since method enables a UE to utilize PDSCH transmissions (meant for it as well as other UEs) for estimating the channel, the time and frequency resolution of measurements are also higher than what can be obtained with CSI-RS or DMRS. This is because frequent transmission of these reference symbols increases the communication overhead.

4.1.1.2 Beam training interval adaptation with radio-aware DT

Any communication system implementing beamforming requires periodic beam refinements to maintain the connection under mobility. Such mobility can be of the terminals themselves, or of other objects in the environment.

Beam refinement can either be done reactively, i.e., it is only done after the link breaks, or proactively, meaning that it is performed before the link breaks. The former method has the obvious disadvantage that the link will be broken at some point, so if, e.g., a URLLC packet is sent at that time, it is likely to be lost. The latter method in contrast is likely to prevent link disruptions but requires more overhead and to potentially run beam refinement even when it is unnecessary.

Arguably, the simplest method to implement proactive beam-refinement is to run the beam-training procedure periodically. This procedure involves transmission of reference symbols while sweeping through all the beams. The receiver then measures the received power on each of these beams and informs the transmitter of the beam index with the highest received power. More sophisticated methods also exist (c.f. AI based beam training in [Hexa-X D4.3] However, in current systems, beam-training is done with a fixed training interval designed to work even in the worst-case scenario of very high mobility. If proactive beam refinement is implemented by running the procedure periodically with a constant update rate, two undesired situations can happen:

- The environment is almost static, or the movements are controlled, so the procedure causes useless overhead.
- There are uncontrolled fast movements in the environment that affect the channel faster than the update period, causing link disruption.

Both cases can happen in the same environment at different times if the update rate is fixed by design. Moreover, in case a radio-aware digital twin is partially managing the network and environment, some of the movements might be planned to not interfere with the communication, therefore increasing the training interval due to those movements will cause unnecessary overhead.

In this subsection, adapting the beam training interval with a radio-aware DT is considered, where there is some amount of randomness still persisting in the environment. In other words, not all objects in the environment may be tracked by the radio-aware DT. Such objects could be humans on the factory floor or vehicles driven or remotely controlled by humans.

In joint communication and sensing, the estimation and analysis of the Doppler spectrum is well known and studied to infer information about the environment (e.g. [PZLJ+16, WPF+09, PMR+21]), however direct usage of the Doppler spectrum to determine the beam-refinement period has not been proposed before, to the best of our knowledge.

Some methods to adapt the training interval have been proposed in the past, for example in [YZ+19] the angular speed of the object is estimated to determine the beam coherence time. In [SMR+18] a more advanced approach is proposed, but also in this case the model requires LoS and only accounts for the mobility of the user and not of other objects in the environment. Our method instead is extremely simple, does not assume LoS and considers the movements of the environment as well.

One approach to predict mobility in an environment is to estimate the Doppler spectrum of the received signal. Such a spectrum will contain high-frequency components if there is significant movement in the area. A radio-aware DT, such as the one described in the previous subsection will be able to provide a prediction of the Doppler spectrum based on the locations and trajectories of the transmitter, receiver, and tracked objects in the environment.

The proposed approach utilizes the mismatch between the Doppler spectrum estimated from the measurements and that is provided by the radio-aware DT to adapt the beam training interval. The approach first involves estimating the channel through reference symbols such as DM-RS/CSI-RS/SRS. From such a channel estimate, the Doppler spectrum can be obtained by performing the Fourier transform of the autocorrelation function or any related spectral estimation method (Welch periodogram, MUSIC, or ESPRIT). Examples of Doppler spectrum with and without movement in the environment can be seen in Figure 4-2.

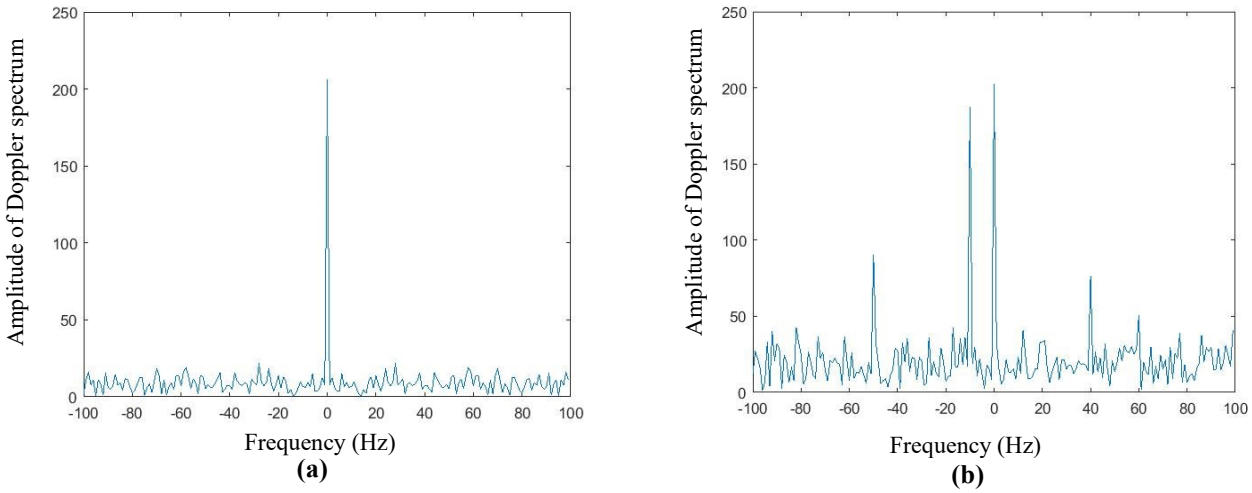


Figure 4-2: Doppler spectrum for (a) static and (b) dynamic environments.

In a static environment, the channel remains almost constant, resulting in a single large component in the Doppler spectrum at 0 Hz. In a dynamic environment, each object moving within the environment relative to the transmitter and receiver will result in a peak in the Doppler spectrum. As shown in Figure 4-3, this frequency is directly proportional to the component of the velocity orthogonal to the ellipse with the transmitter and the receiver at the foci. If $\bar{v}$ is this component of velocity, the Doppler frequency $f_d = 2f_0\bar{v}/c$.

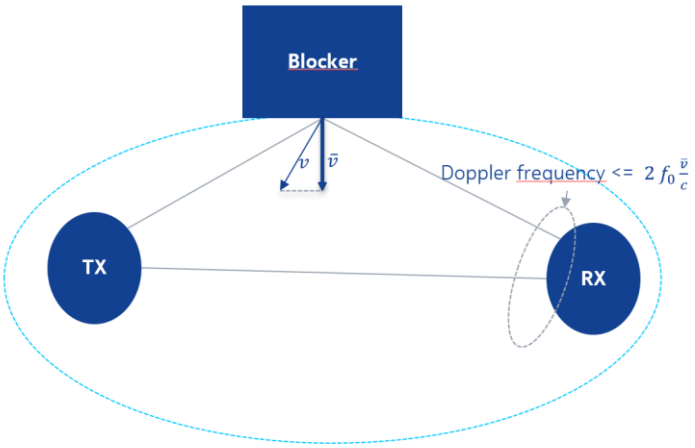


Figure 4-3: Depiction of the interpretation of the Doppler frequency.

Now, a DT that is aware of the objects in the vicinity of the transmitter and receiver can predict the Doppler spectrum that is calculated by these objects. If those objects are under the control of the DT, they do not possess a threat to the communication since the DT can plan the path of the obstacles such that they do not interfere with the communication or notify the network before a disruption. However, objects that are not under the control of the DT may pose a threat to the communication link.

To identify whether there are objects in the vicinity of the transmitter and/or the receiver, the DT provides a list of frequencies in the Doppler spectrum that correspond to the doppler frequencies from known/tracked objects. Then, the receiver compares these frequencies with the estimated doppler spectrum and ignores the frequencies provided by the DT (since they correspond to objects tracked by the DT). This is referred to as the filtered Doppler spectrum. Only frequencies present in the filtered Doppler spectrum are used for determining the beam training interval.

Next, since the biggest threat (i.e., blockage probability) is posed by the fastest moving object, the highest Doppler frequency is considered for beam refinement. For example, if the highest frequency in the filtered Doppler spectrum is f_{max}, the beam training interval is defined as $T = 1/f_{max}$. In general, though, the update interval can be defined as any function of all the detected Doppler frequencies and their amplitude. It should be also noted that only positive frequencies matter, as negative frequencies are associated to objects that are moving away from the receiver, whereas positive frequencies are caused by objects approaching the receiver.

In this case, one can notice that in the static environment where the only present Doppler frequency is 0, one has $T \rightarrow \infty$, i.e., no refinement is required, thus avoiding useless overhead. Conversely, with very fast movements the beam is updated very often in order to guarantee reliability. Note that T is computed from the filtered doppler spectrum, and $T \rightarrow \infty$ if the filtered doppler spectrum only contains the static component. This is because when the objects are under the DT's control, the DT can manoeuvre the objects such that they do not interfere with the link between the transmitter and receiver despite showing up as a positive doppler frequency, thus eliminating the need for adapting the beam training interval.

4.1.2 Dependability in UAV-assisted massive MTC

Many emerging MTC applications rely on information collected by sensor nodes where the freshness of information is an important dependability criterion. Age of Information (AoI) is a dependability metric that quantifies information timeliness, i.e., the freshness of the received information or status update. The importance of AoI minimization lies in improving the dependability of a certain setup by obtaining fresher information about processes. This could be of great relevance in applications such as smart agriculture, V2x, and industrial IoT.

In this context, a setup of deployed sensors in an IoT network is considered, where multiple UAVs serve as mobile relay nodes between the sensors and the base station. An optimization problem is formulated to jointly plan the UAVs' trajectory, while minimizing the AoI of the received messages. This ensures that the received information at the base station is as fresh as possible. The complex optimization problem is efficiently solved using a deep reinforcement learning (DRL) algorithm. In particular, a deep Q-network is proposed, which works as a function approximation to estimate the state-action value function.

AoI is a function of both how often the packets are transmitted and the delay they experience in the system. To begin with, the AoI metric is defined as the time elapsed since the generation of the packet that was most recently delivered to the destination. Minimizing the AoI in IoT applications has attracted lots of attention recently in many applications [KSUB+18]. For instance, reducing the AoI in vehicular communications (e.g, state of traffic lights, vehicles, and road sensor states, etc.) could prevent the occurrence of accidents and thus, enhance dependability of the whole setup. There is also a special interest in scenarios with power-limited sensor nodes (SNs), and their communication with the base station (BS) is difficult or even infeasible most times [HLC21]. The SNs in such scenarios might not transmit the signals with sufficient power and hence will not achieve the signal-to-interference plus noise ratios (SINR) required to decode the data at the BS.

A large area with a set of D low-power single-antenna IoT devices randomly deployed in the 2D plane to monitor different physical processes is considered as in [AEFDS19]. A BS is located at the center of the area. A set of U rotary-wing UAVs is dispatched to collect information from all the deployed devices by flying over different spots within the service area. The main objective is to gather information from the devices in a way which reduces the weighted sum of AoI while reducing the energy consumption for each IoT device. Each UAV then relays the information from the IoT devices to the BS at the center of the map. A set of UAV charging depots is conveniently deployed at fixed positions around this area (for example, at the corners). The system model is illustrated in Figure 4-4.

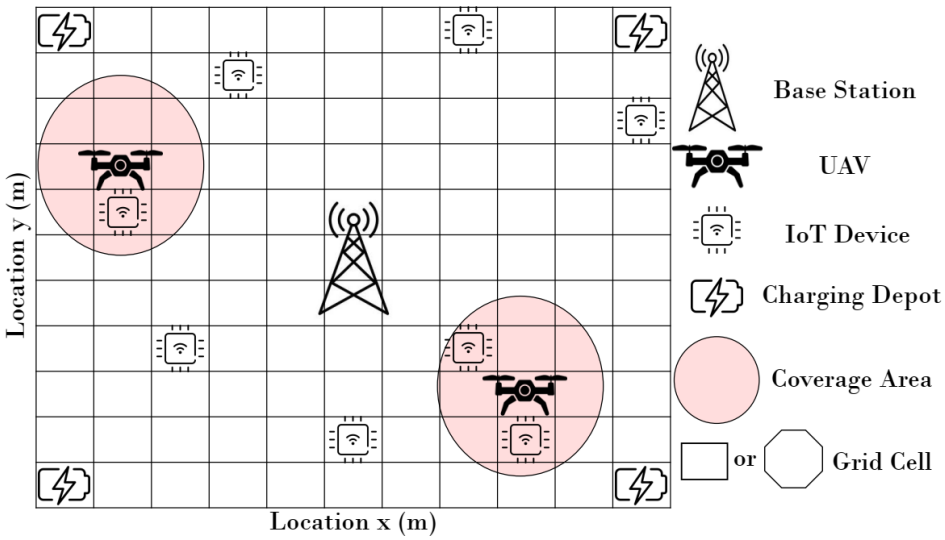


Figure 4-4: The system model comprises a set D of IoT devices served by a set U of rotary-wing UAV. Each UAV relays the information from the IoT devices to the BS in the middle of the map [EPS22].

The target is to optimize the trajectory and scheduling policy of the UAV to jointly minimize the AoI of MTC devices and the energy consumption of both the MTC devices and the UAVs. To do this, deep reinforcement learning (DRL) using Markov decision process (MDP) is utilized. In this scenario, the elements of the MDP are as follows:

- The agent: Multiple UAVs.
- The state space is composed of the location of each UAV, the AoIs of the devices, and the battery levels of the UAVs, respectively given by s(t)=(l(t), A(t), Δ(t))

- The action space consists of the direction of movement and the scheduling policy of the UAV given by: $a(t) = (v_u(t), w(t))$
- The weighted reward function:

$$r_u(t) = \begin{cases} -\sum_{d=1}^{D} \theta_d A_d(t) - z, & d_{u,d} > R_d, \\ -\sum_{d=1}^{D} \boldsymbol{\theta_d A_d(t)}, & otherwise. \end{cases}$$

which is determined according to the distance between the UAV and the scheduled device, whether this distance is withing the coverage radius (red circle) of the UAV R_d or not. The penalty z accounts for the high energy consumed when transmitting at large distance.

Figure 4-5shows the average AoI (i.e., final average age of the simulation time) regarding the number of IoT devices. The age increases with the number of IoT devices and the proposed DRL scheme outperforms the random walk (RW) policy. In addition, the reduction in the age is quite significant while using DRL scheme over the RW policy when the number of IoT devices increases. The 1-UAV DRL almost achieves the same average AoI of the 2-UAV RW policy for 10 devices. Since the UAV serves only 1 device at each time instant, the devices have to wait longer period until being served by the UAV, which increases the age in case of a large number of IoT devices. Therefore, the larger the deployment, the more significant the reduction of the age of the DRL policy compared to the baseline RW policy.

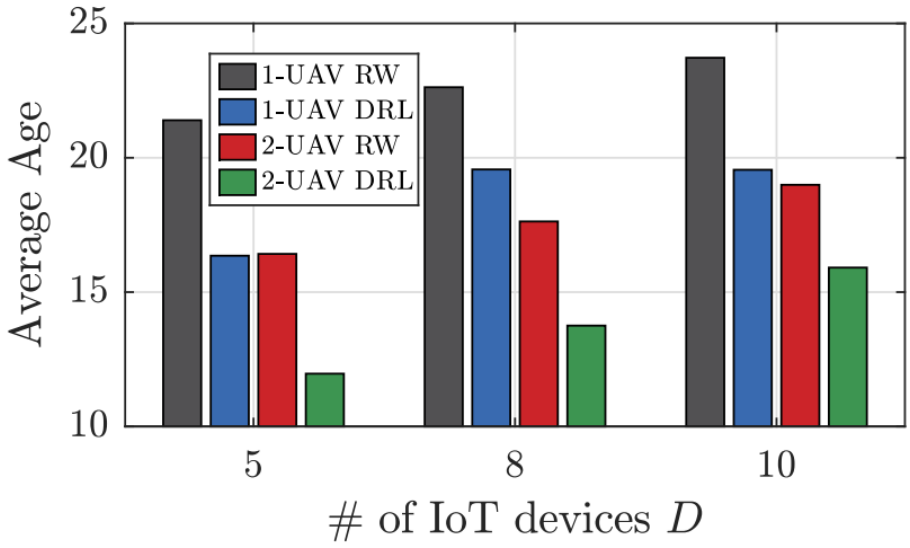


Figure 4-5: Average age of information for the RW and the proposed DRL approach with 1 and 2 UAVs using the 9-directions model as a function of the number of IoT devices D.

Figure 4-6 illustrates the energy consumption for the proposed DRL scheme compared to the random (RW) scheme for 2 UAVs serving 5 IoT devices using both the 5-directions UAV movement model (i.e., east, west, north, south, hovering), and the 9-directions UAV movement model accounting for diagonal movements. The available energy levels presented is the average energy levels of the two UAVs. The DRL scheme finds the optimal policy that can achieve the best combination between the desired low age and saving UAV energy before recharging. The 9-directions model outperforms the 5-directions model in terms of average AoI. In addition, the 9-directions model also overpasses the 5-directions model by saving more energy levels as the 9-directions model has more flexibility in movement and can save time and energy by reaching the optimized location faster than the 5-directions model. It can be observed that the available energy levels of the DRL scheme using the 9-directions

model is almost double the available energy levels of the RW scheme using the 5-directions model after 59 time instants.

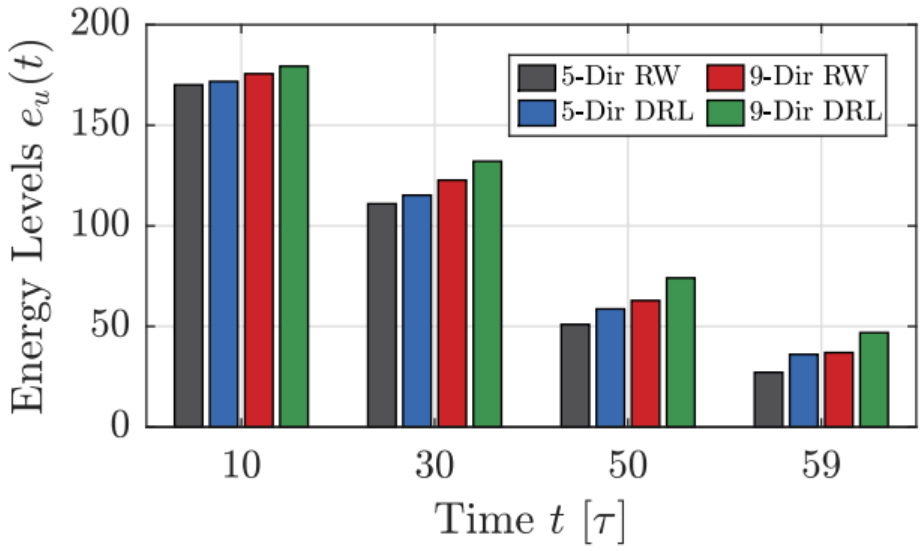


Figure 4-6: Available number of energy levels at the UAV for the random walk and the proposed DRL approach with 2 UAVs serving 5 IoT devices using the 5-directions model and the 9-directions model.

4.1.3 Data and control plane guarantees in programmable industrial networks

Dependable and often deterministic communication is a crucial requirement in Industry 4.0 scenarios, as application resilience is severely limited, especially in brownfield deployments that involve legacy equipment and machinery. Quality of Service (QoS) violations can result in complete standstill or significant decreases in application productivity. To take advantage of the flexibility offered by B5G/6G systems that arise from virtualization and programmability in such conditions, both the data and control planes of the network must demonstrate predictable performance. In addition to predictably forwarding user data, the network must ensure the timely delivery of control plane messages while also considering the impact of virtualization and network update.

Existing state-of-the-art systems either assume full end-host control (e.g., traffic scheduling, policing) as in data centres [AGM+10, LML+19, ZEF+15], or they concentrate only on the data plane, neglecting deterministic operations at the control plane [BDZ+19, GSG+15, JSB+15]. In contrast, here a new system is proposed that combines predictable control plane operations into a data plane, enabling joint data and control plane end-to-end latency and throughput guarantees. Our system uses an in-band control plane, meaning that there is no need for physically isolated dedicated control plane channels, leading to a significant reduction in CAPEX and OPEX. Moreover, our system relies solely on existing mechanisms available in current programmable state-of-the-art switches, such as traffic metering, priority queuing, and label-based forwarding, thus avoiding the creation of new or modification of existing protocol stacks for deployment.

Use case: Reconfigurable Production Lines

In Figure 4-7, an instance of a flexible production line is shown where a single production line handles three different products - circles, squares, and triangles. A robotic arm processes these products depending on the production line mode, with the processed products being represented by dotted ones. In a smart factory setting, a Programmable Logic Controller (PLC) controls the actuator, which, in this case, is the robotic arm. The PLC can be virtualized and run on a server connected to a deterministic network. There are three different PLCs, with each one responsible for a different product – PLC_c, PLC_s, and PLC_t, which control the processing of circles, squares, and triangles, respectively. Initially,

PLC_c controls the robot arm through flow f_c to process circles using a predefined logic, while flows f_s and f_t remain inactive. However, the plant operator may decide to switch from processing circles to squares and later to triangles. It is critical to ensure that the production line never stops since that could lead to revenue loss. Hence, it is essential to transfer control of the robot arm from PLC_c, to PLCs seamlessly. To achieve this, flow f_s must be established on the network in a timely and consistent manner before the deadline T_s. Similarly, to process triangles later, the control of the robot arm needs to be handed over from PLC_s to PLC_t, before the deadline T_t. This is a challenging task because the data plane (DP) must be deterministic and updating the network forwarding logic must occur consistently and timely. Furthermore, adding a flow to the network may require existing flows to be rerouted, making the problem even more complex. Thus, proposing novel deterministic networking systems is essential to industrial scenarios that require strict data plane and control plane (CP) guarantees.

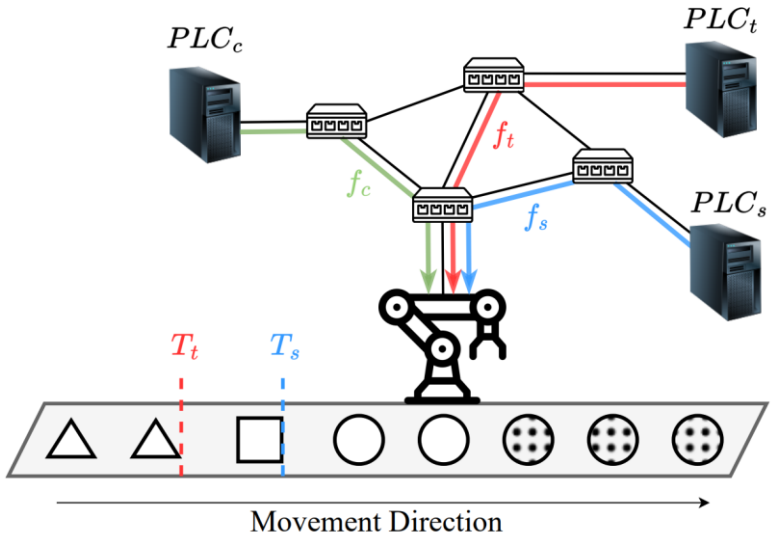


Figure 4-7: The use case for reconfigurable production line

Flow requests are defined as a 5-tuple (src, dst, rate, burst, delay) for end-to-end communication. These requests arrive at the North-Bound-Interface (NBI) of the controller in an online manner. The admission control algorithm is employed by the controller to embed the flows into the network while ensuring the guaranteed performance of these flows, including a bounded end-to-end delay and guaranteed data rate along the path. To achieve these guarantees, the admission control algorithm uses the deterministic network calculus framework [BT01]. Once the flows are embedded, the controller sends network updates to the network devices to enable them to forward the traffic of the new flow.

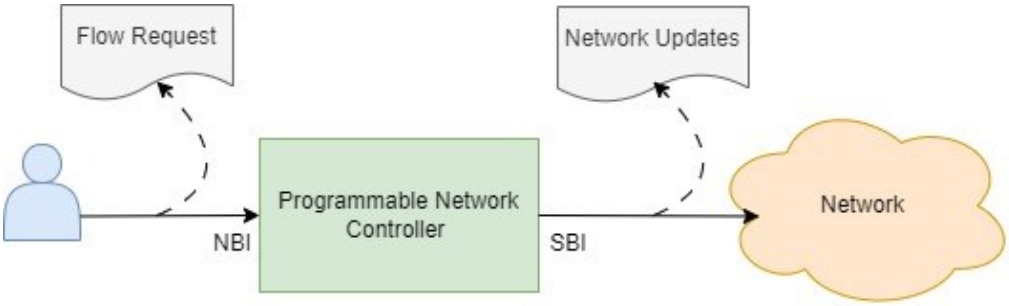


Figure 4-8: System Model

In addition, maximizing the flow acceptance rate is crucial for the network operator to increase revenue. By flow acceptance rate it is meant being able to embed as many flows as possible in the network, i.e.,

increasing network utilization. Therefore, network reconfigurations are implemented as a means of accommodating more flows. In essence, when a new flow request is rejected, some of the existing flows are rerouted to free up the necessary resources for the new flow. Thus, the admission control algorithm outputs a set of network updates required to embed the new flow, which may involve either adding a single forwarding rule or adding a rule along with dependent reconfigurations (rerouting). These network updates need to be transmitted to the forwarding devices through the in-band control plane channel. To achieve this, a fast and efficient scheduler algorithm for predictable and consistent (re)configuration of the network is proposed.

This procedure is designed such that the data plane traffic guarantees are not violated during the network update. To ensure this, network devices need to be predictable, both from data plane and control plane point of view. For instance, one should ensure that they can perform traffic shaping precisely, otherwise, the network calculus framework cannot guarantee the packet latencies in the network. To do that, the measurement testbed is set up as shown in Figure 4-8. The Traffic Generator is the DPDK-based MoonGen [EGR+15] running on a server with Ubuntu 18.04 and equipped with Intel Xeon E5-2650 v4 @2.2 GHz CPU and an Intel X520 NIC. Also, the Device Under Test (DUT) is Wedge 100BF-32X/65X. The tap device copies the traffic and DAG is the network packet capturer card. The DUT device has to shape the incoming traffic to its port 1 and forward the shaped traffic to port 2. Thus, the accuracy of the shaping functionality can be determined. As depicted in Figure 4-9, the shaping function seems very accurate in different flow rates.

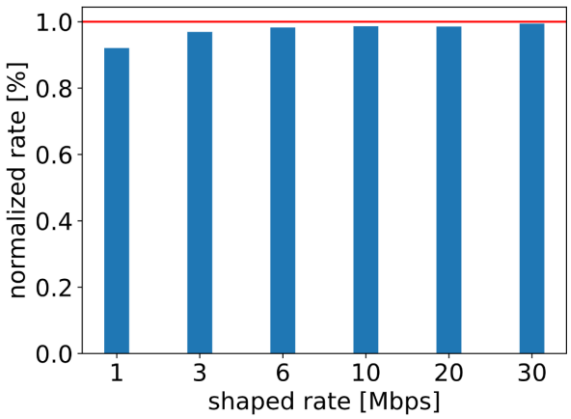


Figure 4-9: Shaping Accuracy

Additionally, a real-world implementation and assessment of our proposed system using a network topology depicted in Figure 4-10 was conducted. The testbed consists of five switches, six hosts, and a controller. Specifically, the PICA-P3297 and P4-enabled network switches were utilized in our testbed.

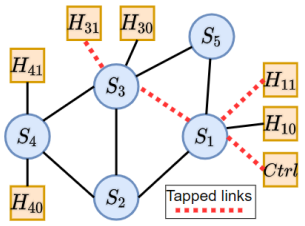


Figure 4-10: Testbed network.

To perform the evaluations, different sets of flow types are used with random source-destination pairs and various characteristics as presented in Table 5.

Table 5: The considered flow types and their characteristics (as uniform distribution).

Flow Type	Rate (Mbps)	Burst (kb)	Delay (ms)
Industrial	U(8, 40)	80	U(5, 10)
Hadoop	U(40, 120)	U(80, 120)	U(10, 100)
Data mining	U(80, 160)	U(12, 160)	U(10, 100)
Strict consistency	U(5,8)	U(80, 120)	U(5, 50)
Adaptive consistency	U(2,4)	U(80, 120)	U(5, 50)

The end-to-end delay is measured in our network by tapping certain links (represented by dashed lines) and using a 10G Endace DAG 10X4-S measurement card to accurately measure latency in nanoseconds. A series of random flows from the table mentioned earlier is then introduced into the network and a subset of them is assessed for evaluation purposes.

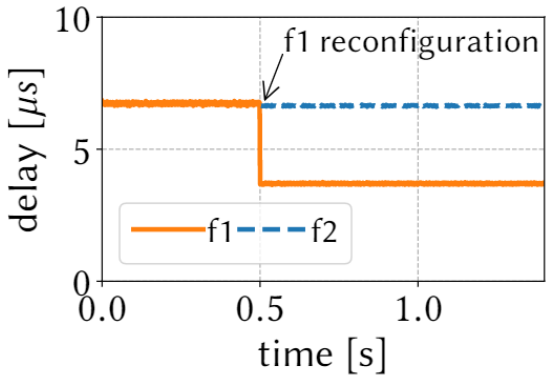


Figure 4-11: Confirming the consistency of the proposed system while reconfiguration.

After successfully embedding more than 100 flows in the network, no packet loss or end-to-end delay violations for any of the flows were observed. To evaluate the system's ability to reconfigure flows without violating their guarantees, an experiment was conducted where flow f1 was added to the network at t=0s, then at t=0.5s, a new flow f2 was added to the same path as f1 while re-routing f1 to a path with lower delay. The end-to-end delay of these flows was measured, and the results are presented in Figure 4-11. Throughout the measurement, the end-to-end delay remained constant, including during the reconfiguration at t=0.5s, and there was no packet loss.

In another experiment, the scalability of the system's control plane was assessed, particularly in terms of the network update rate in a worst-case scenario. To do this, simulations on the Internet2 topology were conducted, with a varying number of nodes and flows arriving at the scheduler over time. Figure 4-12 shows that the proposed scheduling algorithm can maintain a network update rate of up to 280 per

second as the network size increases, whereas state-of-the-art approaches such as [ZHK+21, NCC17] can only maintain around 5 network updates per second.

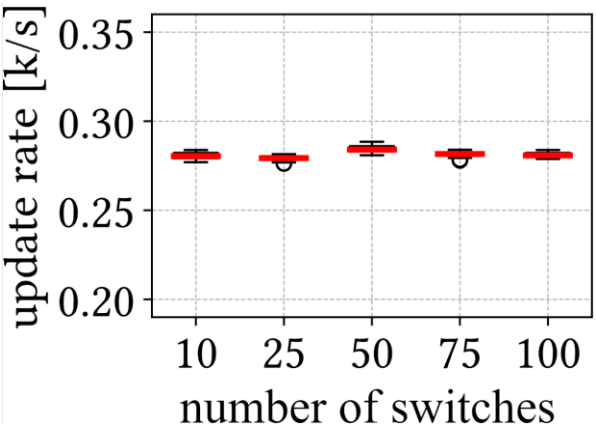


Figure 4-12: Network update rate scalability.

The proposed system acts as one building block for achieving the required QoS attributes for dependable communication while maintaining flexibility in virtualized and programmable network resources.

4.1.4 Network data analytics assisted AI operations for factory network optimization

4.1.4.1 Background and motivation

Industry 4.0 is the next era in industrial production, aiming at significantly improving the flexibility, versatility, usability, and efficiency of future smart factories. Industry 4.0 integrates the Internet of Things (IoT) and related services in industrial manufacturing and delivers seamless vertical and horizontal integration across all layers of the automation pyramid. Connectivity is a key component of Industry 4.0 and will support the ongoing developments by providing powerful and pervasive connectivity between machines, people and objects. Wireless communication is an important means of achieving the required flexibility of production, supporting new advanced mobile applications for workers, and allowing mobile robots and autonomous vehicles to collaborate on the shop floor etc.

Artificial intelligence has been considered as an important technology with huge potential to improve the performance and efficiency of next generation wireless network and various vertical applications powered by the wireless network. AI enabled 5G radio access network performance and efficiency enhancements have been studied in [37.817]. Specifically, according to [37.817], the unified functional framework for AI enabled RAN intelligence operation is illustrated in Figure 4-13.

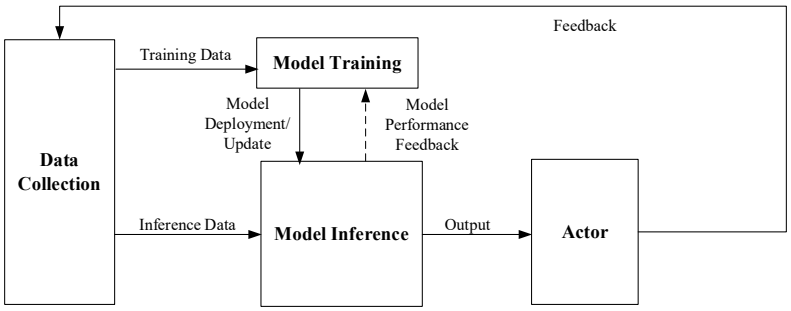


Figure 4-13: Functional framework for RAN intelligence [37.817].

The solutions in [37.817] consider two type of AI function locations in 5G system: 1) AI/ML Model Training is in the operation, administration, and management (OAM) and AI/ML Model Inference is in

the gNB; 2) AI/ML Model Training and AI/ML Model Inference are both located in the gNB. For the future factory of Industry 4.0 with connectivity infrastructure provided by 5G private 5G network, the factory network owner can act simultaneously as the end user and operator of the private 5G factory network so that the factory network owner can have the access to the factory automation applications data as well as the underlay 5G network data analytics. It is envisioned that the factory network operator can employ AI/ML technique by virtue of both application and network data to improve the overall factory efficiency in terms of achieving the application required communication quality of experience (QoE) and optimal network resource and energy efficiency simultaneously.

Different factory applications can have different application characteristics with different required QoEs. For example, for video streaming applications, current video buffer level and playout delay for media start-up are important application QoE parameters while for industrial real-time automation applications, end-to-end (E2E) packet delay and per-packet error/loss event probability are important application QoE parameters. When factory AI/ML operation, i.e., model training and/or interference, is located in 5G gNB, the QoE parameters depending on the factory applications which are involved in AI/ML operation can be exposed to gNB by control plane signalings, i.e., RRC message as defined in [38.331].

It is envisioned that application QoE and RAN resource management can be jointly optimized to achieve the required application-level QoE with the efficient usage of RAN resource and energy. Moreover, AI/ML operation can be used to provide analytics data used for joint application and RAN optimization, and the AI/ML model can be generated and updated by factory network management system (FNMS) comprised of 5G OAM, network data analytics function (NWDAF) and factory application server. As detailed in the following, a new signalling framework to enable the joint application and RAN resource optimization is proposed.

4.1.4.2 AI assisted joint application and RAN optimization

In this section, we propose a method to realize joint application and RAN resource optimization depending on the analytics data generated by AI/ML model which is determined and updated by NWDAF in FNMS. In this method, 6G system is deployed in future factory to provide communication connectivity among those participant entities, i.e., future factory application UE (FFA-UE) and server (FFA-server), involved in industrial automation applications. Moreover, AI/ML operation with the principle illustrated in Figure 4-13 can be used in the factory network to perform joint application and RAN efficiency optimization. For example, the application-level optimization strategy can correspond to the improved application QoE in terms of E2E application message latency and/or per-application message reception loss rate, and the RAN optimization strategy can refer to the network resource scheduling solution to achieve an improved total RAN energy efficiency while meeting the required application QoE requirements. Specifically, the FNMS can be comprised of, among others, NWDAF, FFA-server and OAM of 6G system. Based on the knowledge about the desired application characteristics and RAN resource/energy efficiency obtained from FFA-server and OAM, respectively, the NWDAF can determine the configuration, activation and deactivation of the AI/ML model which is requested or subscribed by next generation RAN (NG-RAN) node to generate a set of Analytics data used for joint application and RAN resource/energy optimization, and further update the respective AI/ML model. To realize the AI/ML operation for joint application and RAN optimization, we propose the signalling procedure illustrated in Figure 4-14, and the respective detailed signalling steps are described below. The proposed signalling method is also based on service-based architecture (SBA) principle adopted in 5G and future 6G system [Section 6, Hexa-X D6.2].

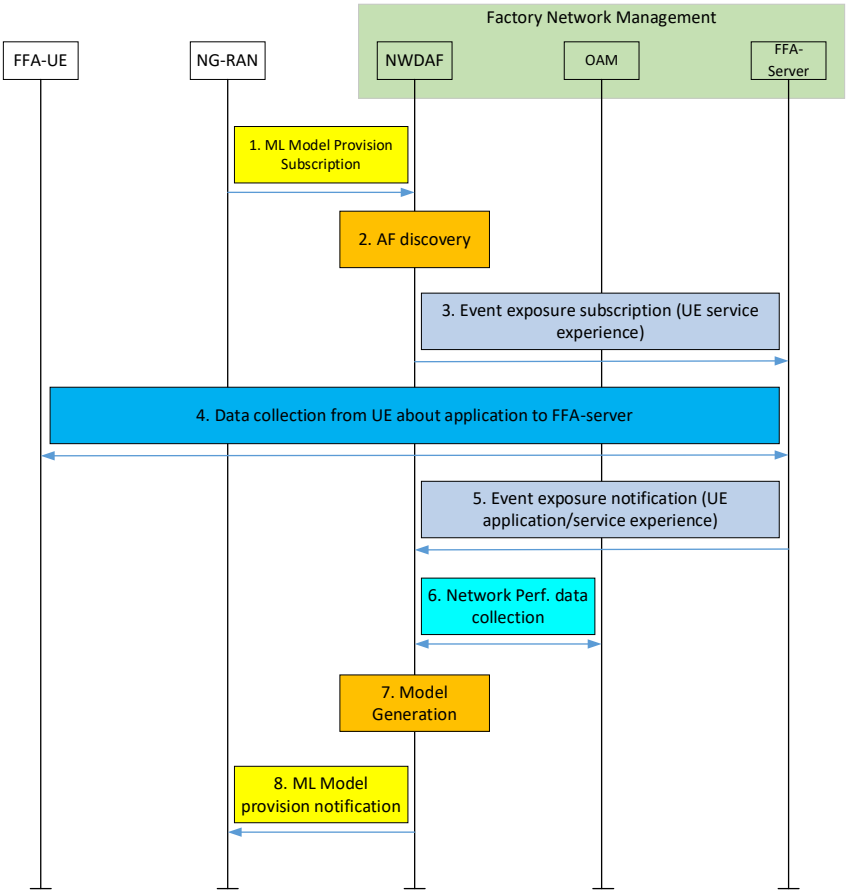


Figure 4-14: NWDAF based AI/ML operation signaling for joint application and RAN optimization.

Step 1: The NG-RAN node as NWDAF service consumer subscribes to, modifies, or cancels subscription for a (set of) trained ML Model(s) associated with a (set of) Analytics used for joint application and RAN optimized radio resource management algorithm by invoking the ML model related subscription and un-subscription messages, e.g., Nnwdaf_MLModelProvision_Subscribe / Nnwdaf_MLModelProvision_Unsubscribe services [23.288]. Upon the reception of the subscription for a trained ML model, the NWDAF containing model training function may initiate several possible processes as described in [23.288].

Step 2: NWDAF discovers the FFA-server that provides data collection (based on the FFA-server profiles registered in network repository function (NRF)) as described in TS 23.502 [23.502].

Step 3: NWDAF subscribes to the FFA-server located in factory network management domain for UE data collection (i.e. input data from UE for analytics), by using Naf_EventExposure_Subscribe as defined in [23.502]. The event is associated with the UE application/service QoE. The NWDAF request contains an Application ID known in the core network and the UE Application provides the Application ID configured in the UE Application. The FFA-server binds the NWDAF request for an Application ID and the UE data collection for an Application ID configured in the UE. Data collection can also be triggered using data collection coordination function (DCCF) described in [23.288].

Step 4: The FFA-server collects the UE data using either direct or indirect data collection procedure in [23.288]. The establishment of the connection can be performed at any time prior to this. The FFA-server links the data collection request from step 3 to the user or control plane connection.

Step 5: The FFA-server in factory network management domain receives the input data from the UE and processes the data according to the service level agreement (SLA) that is configured in the FFA-server and Event ID(s) and Event Filter(s) related to UE application service experience set during step

3. The FFA-server then notifies the NWDAF on the processed data according to the NWDAF subscription in step 3.

If NWDAF requests the same data from multiple UEs, i.e., a determined list of UEs or "any UE" as the Target of Analytics Reporting, the FFA-server can process the data from multiple UEs according to the Event parameters received from NWDAF during step 3 before notifying the NWDAF on the processed data in step 5.

Step 6: The NWDAF subscribes to OAM services to get the status and load information and the resource usage on the Area of Interest according to the procedure [23.288].

Step 7: Model Training at NWDAF containing model training logical function (MTLF). Required measurements and input data from FFA-server and OAM nodes are leveraged to train AI/ML models for a (set of) Analytics used for joint application and RAN network optimization in terms of UE QoE and network resource energy efficiency.

Step 8: If the NG-RAN node subscribes to a (set of) trained ML model(s) associated to a (set of) Analytics, the NWDAF containing MTLF notifies the NG-RAN node with the trained ML Model Information (containing a (set of) file address of the trained ML model), e.g., by invoking Nnwdaf_MLModelProvision_Notify service [23.288] operation. The exemplary content of trained ML Model Information provided by the NWDAF can be referred to [23.288].

The NWDAF containing MTLF can also invoke the notification message, i.e., Nnwdaf_MLModelProvision_Notify, to notify an available re-trained ML model when the NWDAF containing MTLF determines that the previously provided trained ML Model required re-training at step 1.

When the step 1 is for a subscription modification (i.e., identified by Subscription ID), the NWDAF containing MTLF may provide either a new trained ML model different to the previously provided one, or re-trained ML model by invoking the same notification message, i.e., Nnwdaf_MLModelProvision_Notify.

With the described signalling steps above, FNMS comprised of NWDAF, FFA-server and OAM, can consequently generate and update the AI/ML model to provide required analytics data used for realizing joint application and RAN network efficiency optimization to significantly improve the productivity and efficiency of future factories. In principle, the proposed method can be employed to implement the variety of AI/ML assisted 6G network performance enhancements reported in Section 2 of [Hexa-X D4.2]. Moreover, the proposed method can be also utilized to realize the DT assisted RRM approaches in Section 4.1.1 and joint communication-and-control methods in Section 4.2 in the factory network.

4.2 Communication-Computation-Control-Codesign

The term “Communication-Computation-Control-Codesign” (CoCoCoCo) refers to the convergence of building blocks that enable the creation of distributed, scalable, and dependable systems. In this context, "building blocks" refer to the different "Co's," including communication, computation, and control blocks. The term "convergence" means that instead of using static approximations of one domain when designing, analysing, or optimizing the other, the overall system is modelled using a single description.

For example, in the classical separated approach, the communication model may be approximated as a fixed delay when designing a closed-loop control application, or the computing delay may be assumed to be constant when offloading data to an edge cloud server. However, such static approximations fail in two fundamental ways.

Firstly, when considering only a single application instance, the goal for any control application is to satisfy a certain quality of experience (QoE), which might include not exceeding a certain maximum acceleration when performing a task or not deviating by more than a certain maximum offset from a desired value. However, this does not necessarily mean that there is no room for errors in the pursuit of achieving this goal. For example, any packet loss does not necessarily cause the control instance to

exceed the threshold acceleration. A joint model enables understanding the interdependency of the respective domains, making it possible to evaluate the significance of an error in terms of its impact on QoE.

Secondly, in a networked scenario, multiple concurrent agents and applications share available resources, which may be on the air interface (e.g., spectrum) or on a shared computing platform. Taking a simplified static modelling approach yields highly conservative results, as resources must be reserved exclusively for the static assumption to hold. For example, computing resources withheld from other users will be wasted during every time period in which the reserving application instance does not need them. A joint model enables the effective use of all available resources and adherence to a range of additional constraints, such as end-to-end delay bounds for certain agents or maximum EMF exposure in particular areas. As a side effect, identifying the bottleneck of experiencing a higher QoE is possible, and appropriate measures can be taken.

The benefits from having a joint model description are significant and evident as will be shown in later subsections. However, this joint modelling requires substantial effort, especially when attempting to capture co-dependencies of networked systems. This problem is discussed in more detail in the following section that contrasts a novel data-driven framework with the established model-based approaches.

4.2.1 Model-based vs. data-driven approach

System optimization in the context of CoCoCoCo relies on accurate dependability models across the different planes of data transmission, computing & controlling, and application. This model can be very complex, sometimes even intractable when following the classical approach of system reliability modelling, which is based on the stochastic state of multiple components. More specifically, the state of an industrial system is typically considered upon its structure function:

$$X_{sys,t} = \emptyset(X_{1,t}, X_{2,t} \dots X_{I,t})$$

where $X_{sys,t}$ is the system reliability at time t, and each $X_{i,t}$ the reliability of its component i at time t. Thus, the system-level dependability metrics, such as the mean time to failure (MTTF), can be evaluated through a probabilistic integral over the process $X_{sys,t}$, and the overall CoCoCoCo optimization problem can be generically formulated as

$$\begin{aligned} \min_{z} \quad & \mathbf{E}_{P_1}\left[Q_1(z,\omega_1)\right] \\ s.t. \quad & f(z) \le 0, \\ & \mathbf{E}_{P_1 \times P_2}[Q_2(z,\omega_1,\omega_2)] \le 1 - R_{app}(0) \\ & z \in \boldsymbol{R}^n \end{aligned}$$

$$\begin{aligned} Q_1(z,\omega_1) = \min_{z} \quad & \boldsymbol{E}_{P_2}[J(u)] \\ s.t. \quad & g(\dot{x},x,u,\omega_1,\omega_2) = 0, \\ & (x_0,t_0) = v(z,\omega_1), \\ & u \in \boldsymbol{R}^m \end{aligned}$$

$$\begin{aligned} Q_2(z,\omega_1,\omega_2) &= 1 - I(y) \\ y &= h[x,u,v(z,\omega_1),\omega_2] \\ I(y) &= \begin{cases} 1, & y \in D_{app} \\ 0, & o.w. \end{cases} \end{aligned}$$

where the outer stage decision vector z is the design parameters of the communication subsystem that determines the stochastic delays of inner-stage problems; P_1 and P_2 are the probability measures of the stochastic time delay and of the stochastic disturbance in the controlling subsystem, respectively; ω_1 and ω_2 are the samples of time delay and disturbance, respectively. R_{app} is the demanded performance reliability at the application level, J is the optimized expected performance of the control objective, u is the control signal, g is the state-space equations of the controlled object, (x_0, t_0) the set of initial

states that have been observed after the time delay, y the performance of the system at the application level, I an indicator function that determines whether the system's performance is within the demand region.

The most straightforward and classical approach to tackle down this complex multi-level optimization problem is an application-oriented in-loop design, where the controlling, computing, and communication subsystems are optimized in turns iteratively, and each optimization loop is followed by a system-level dependability test on the application layer to drive the optimization to the next round. This approach has a great advantage in its modelling simplicity, requiring no novel model beyond the conventional domain-specific models. However, it requires accumulating expertise from all fields of controlling, computation, and communication throughout an iterative development process, which consumes both time and cost. Furthermore, since the test stage is coupled with a specific application, the final design cannot be flexibly modified in case of system upgrades or scenario adaptation. For a higher generality and flexibility, it appears reasonable to decompose the overall problem into domain-specific subproblems that are coupled with each other, by means of explicitly formulating the key domain-specific metrics, e.g., the MTTF, state error rate, or PER, etc., as functions of each other. This enables a layer-by-layer optimization that can be executed in a bottom-up fashion. Such a decomposition will allow a much more flexible development cycle and therewith reduce the cost, while still allowing some conventional domain-specific design approaches to be easily adopted on each individual optimization stage. Nevertheless, it requires a deep understanding of the dependability coupling across different domains, to select the appropriate domain-wide dependability metrics that have well-studied correlation with each other. As an effort in this direction, different error sources and their impacts have been identified in [Hexa-X D7.2]. Moreover, some emerging metrics such as AoI and age of incorrect information (AoII) have been proposed recently to umbrella multiple domain-specific QoS metrics from different subsystems, which reflects the propagation and transfer of errors through different subsystems. Nevertheless, the state of the art is still insufficient to support a full modelling of such inter-domain coupling, especially when the uncertainty in computing (e.g., the computing latency, and the value errors intrinsically caused by numerical algorithms within a limited latency budget) is also considered. Hexa-X has made efforts to analytically capture such inter-domain correlations, especially regarding the link between computing and communication (Section 4.2.2), as well as that between communication and control (Section 4.2.3), which are built on intermediate results from [Hexa-X D7.1] and [Hexa-X D7.2].

In some cases, when it appears impossible or analytically intractable to model the coupling and correlation across different subsystems, numerical methods shall be developed to obtain satisfactory solutions. Towards this, a generalized black-box model for networked control systems (NCS) and its dedicated CoCoCoCo optimization problems is proposed in [HSM+22], as illustrated in Figure 4-15. The communication, control, and computation domains are jointly modelled as a packaged black-box that is independent from the physical application-domain implementation. This black-box defines a generic interface to an abstracted application module, which is characterized by a QoS vector in a multi-dimensional space of QoS metrics across different subsystems. A certain implementation of the black-box defines a feasibility region within this QoS space, which can be empirically modelled in a data-driven approach. Upon each specific application, a dependability model can be established to describe the application-layer dependability as function of the QoS vector. For example, in an application of remotely controlled inverted pendulum (see [Hexa-X D7.2]), a black-box model can be used to characterize the remote controlling platform that consists of a specific controller (e.g., a PID controller or a model-based controller), a specific communication system (with certain transmission technology and working under certain channel conditions) and a specific computation unit (with certain computing capacity). Its QoS space defines all feasible metric combinations of these subsystems (e.g., any QoS vector that contains an extremely low control latency and an extremely low PER is infeasible). On the other hand, the application module, which contains in this application the inverted pendulum and the cart carrying it, can be characterized by its MTTF as a function of the CoCoCoCo metric combination. Indeed, by replacing the application module with another, we can easily evaluate the capability of the same remote controlling platform for another application. Thus, the CoCoCoCo optimization problem can be transformed into a mathematical programming problem.

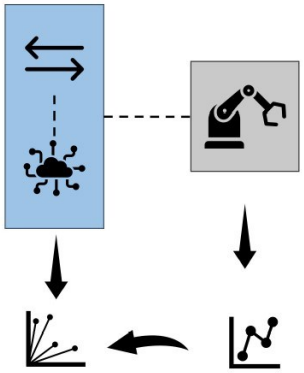


Figure 4-15: A generalized black-box NCS model

Generally, the model-based approaches are more capable of revealing the fundamental mechanisms that different subsystems interact with each other and can exploit such insights about specific applications to better achieve the global optima. In contrast, the data-driven black-box approach relies less on the understanding of the basic principles of the specific system and is more capable of flexibly reusing the same CoCoCoCo infrastructure for different applications. The work on this data-driven modelling approach is still in early stages, hence, no examples can be presented at this stage.

In the following, two model-based approaches are presented. First, a model-based approach in a networked scenario and secondly, a model-based approach for dependability improvements of a single agent.

4.2.2 Identifying the bottleneck of connect-compute services sharing spectrum and computing resources

In [Hexa-X D7.2], [MBC22], the inherent coupling between wireless (transmit energy and delay), computing (processing energy and delay) and the physical world (EMF exposure) is investigated. This is of particular interest for use cases in which data need to be continuously exchanged among heterogeneous intelligent agents to operate complex cooperative tasks, e.g., the cooperative and mobile robot use case.

It has been shown how, with a joint optimisation of radio and computing (joint communication and computation co-design), it is possible to identify the bottleneck of edge computing workloads, which can reside in wireless or computing resources, as well as in EMF exposure constraints, e.g., due to public regulations [ICNIRP20], or any other external restriction or energetic requirement. Indeed, this bottleneck can also depend on the energy availability at different nodes in the network, being them communicating, computing nodes, or both. An adaptive method has been proposed, with the ability of autonomously identifying the bottleneck and proactively dropping packets that cannot be treated within finite end-to-end delay, given the overall capacity of the system. The end-to-end delay entails data transmission and processing, to maximise the data offloading rate (i.e., the number of processed bits per unit time). However, packet losses (due to errors and/or outages) were not considered in the abovementioned study. In previous investigations in [Hexa-X D7.2], it has been shown how control applications are affected by packet losses in different ways, also with different performance depending on the correlation between subsequent losses. A simulation study on an inverted pendulum control use case has also been performed. However, the impact of computing resources and mutual interference is not considered in these studies. In other words, this section extends the solution proposed in [Hexa-X D7.2], [MBC22], considering interference as another source of bottleneck, in a scenario in which heterogeneous services (and systems) co-exist.

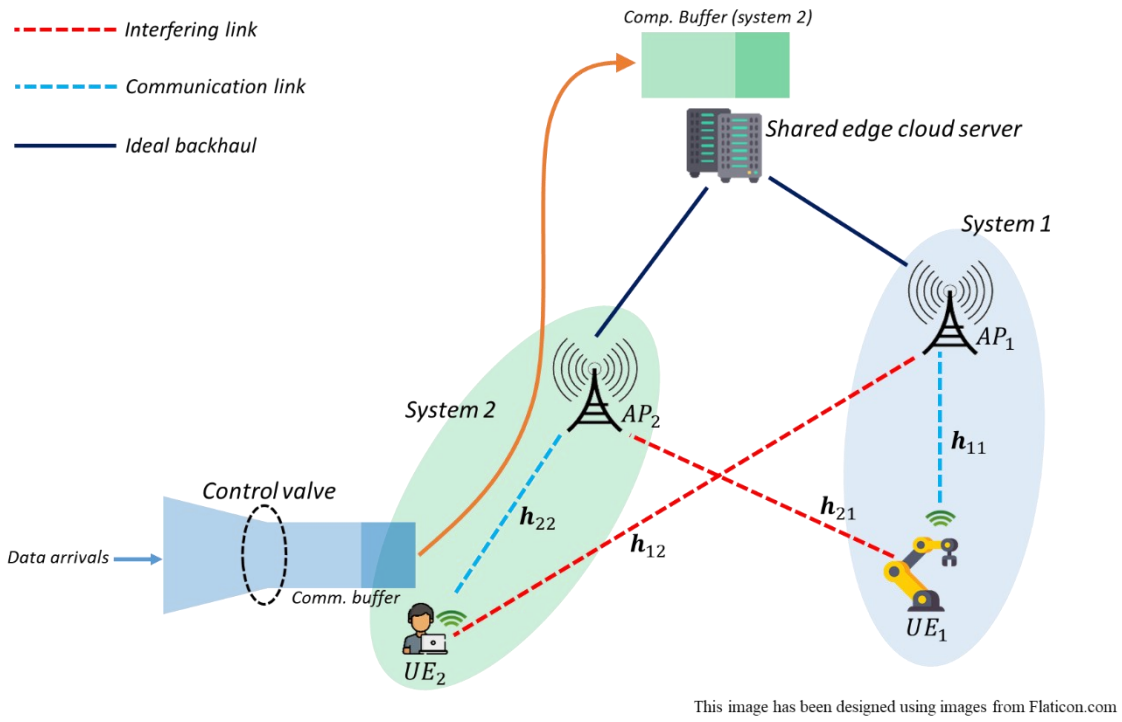


Figure 4-16: Scenario under investigation

A high-level description of the considered system comprising a control system and data offloading is depicted in Figure 4-16: Scenario under investigation. The control system is referred to as *System 1*, and the other offloading system is referred to as *System 2*. More specifically, *System 1* comprises UE_1 and AP_1, while *System 2* comprises UE_2 and AP_2. The two systems share spectrum resource, but also the edge cloud server resources, reached through both APs via high capacity and negligible delay backhaul connections. At the wireless access, it is assumed that *System 1* has the priority, i.e., UE_1 transmits with fixed power (i.e., its maximum power) and target reliability in terms of packet error rate. Conversely, at the computing tier, both systems share the same pool of resources, without a priori priority. At the same time, UE_2 must adapt its transmission to maximise its performance, while guaranteeing *System 1* requirements. This concept is similar to that of cognitive radio settings, in which a secondary user "jumps" into the wireless channel whenever possible, guaranteeing a bounded interference level on a primary user that owns the license of the spectrum resources. In this case, the bottleneck of *System 2* may reside in wireless and/or computing resources, as well as in *System 1* requirements fulfilment.

System model and notation. The wireless channel between UE_i and AP_j, with $i, j = 1,2$ is denoted by h_{ji}. As it will be described in more detail in the following, *System 1* does not make use of buffers, and uploads control data as soon as they are generated, within an end-to-end latency predefined as a service requirement. On the contrary, as in [Hexa-X D7.2] and [MBC22], *System 2* entails a communication buffer of data waiting for uplink transmission, and a computation buffer at the edge server waiting for computation. Also, as in [MBC22], new data arrivals are accepted into the local queue, based on the available offloading capacity, which depends on wireless and computing capacity, whose bottleneck is assumed to be unknown a priori. All other arrivals are proactively dropped to match this capacity, which highly depends on the requirements of both systems, as mutual interference is considered both at wireless and computation tiers. To model system dynamics, time is organised in slots of equal duration τ, at the beginning of which new wireless channels are observed, and a decision is taken on various parameters, as described in the sequel. The idea is to orchestrate wireless and computing resources to explore a trade-off involving: *i) System 1* delay and packet error rate, *ii) System 2* data offloading rate and delay (as in [MBC22]), and *iii)* Edge cloud server power consumption.

In the following, the main KPIs, variables, and models of the two systems are briefly presented.

System 1. In *System 1*, the following KPIs are considered: *i)* Packet Error Rate (PER) during uplink wireless transmission, and *ii)* end-to-end delay, comprising transmission and computation (without buffering). It is assumed that UE_1 transmits, at each slot, short packets, with fixed transmit power, and bandwidth W, thus possibly interfering with UE_2 in those slots in which they both chose to transmit. Then, a time-varying wireless channels, the uplink data rate of *System 1* at time t, in the finite blocklength regime can be written as in [PPV10]:

$$R_1 = W\left[\log_2(1+SINR_1) - \frac{1}{\ln(2)}\sqrt{\frac{V}{n_1}}Q^{-1}(PER_1)\right]$$

where $SINR_1$is the signal-to-noise ratio of *System 1*, i.e., at AP_1, V is the channel dispersion (which generally depends on the channel statistics), n_1 is the blocklength, PER_1 is the packet error rate, and $Q(\cdot)$ represents the Gaussian Q function. At time t, UE_1 experiences a communication delay $D_{comm,1,t}$, which depends on the number of bits to be transmitted (e.g., the information needed to remotely control an actuator). Also, the edge cloud server dynamically assigns computing resources in form of CPU clock frequency. Then, assuming that the computing load ω_1 (in CPU cycles/s) is fixed (e.g., it depends on the number of computations needed to output a control action – AI/ML-based workloads are one possibility), UE_1 experiences a computation delay $D_{comp,1,t} = \frac{\omega_1}{f_{1,t}}$, with $f_{1,t}$ the CPU clock frequency assigned by to it by the edge server. Therefore, neglecting the downlink transmission delay (this can be generalized by taking into account downlink transmission buffering and delay, but it goes beyond the scope of this section), the total offloading delay of *System 1* is $D_{tot,1,t} = D_{comm,1,t} + D_{comp,1,t}$. As a requirement, it is assumed that *System 1* needs to transmit and compute data with a target PER during uplink transmission (which affects the finite blocklength capacity [PPV10], [SSY+19]), and within a finite end-to-end delay, which for simplicity (but without loss of generality) is assumed to equal the slot duration τ, i.e., $D_{tot,1,t} \leq \tau$.

System 2. *In System 2*, the following KPIs are considered: i) data offloading rate, and ii) end-to-end delay. Differently from *System 1*, *System 2* makes use of buffers: a local communication queue, and a remote computation queue. In particular, when data are generated locally at the end user, part of them is proactively dropped, and part of them is accepted into the local queue before transmission. It is assumed that a fixed amount A_{max} new bits arrive at each time slot, and a decision is taken on whether to accept them into the local queue or drop part of them, through the fictitious *control valve* depicted in the left end part of Figure 4-15. Once transmitted, data are buffered remotely at the edge server before computation. This tandem queue model is useful to characterise the end-to-end delay of the data offloaded by UE_2 . By Little's law, the average E2E delay experienced by UE_2 equals the ratio between the total average queue length and the arrival rate. The goal is to maximise the arrivals, while keeping finite buffers (i.e., finite E2E delay) and guaranteeing UE_1 requirements.

Edge server. Edge server resources are shared by the two systems. Then, denoting by $f_{i,t}$ the CPU clock frequency assigned to *System* i at time t, with $i = 1,2$, $f_{1,t} + f_{2,t} \leq f_{\max}$ must hold, with f_{max} the maximum edge server CPU clock frequency. Also, as in [MBC22], the dynamic CPU power consumption can be written as $p_{es,t} = \kappa\left(f_{1,t} + f_{2,t}\right)^3$, where κ denotes a processor dependent parameter, also known as *capacitance.*

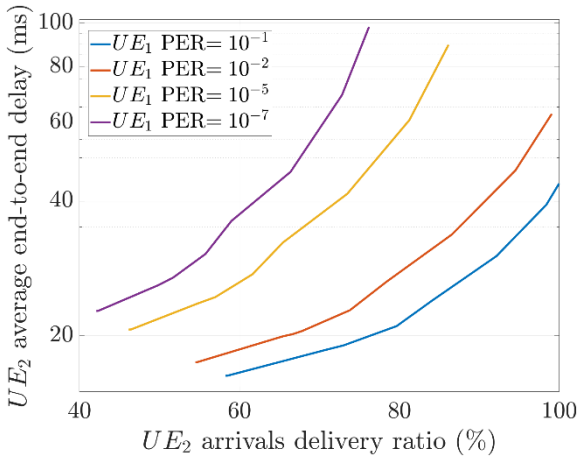
Sketch of problem formulation. Given the described system, the goal is to jointly allocate radio and computing resources to maximise *System 2* data offloading rate (i.e., minimise the number of proactively dropped arrivals) under constraints on: *i)* both queues stability, to ensure finite end-to-end delay for *System 2*, *ii)* target PER of *System 1, iii)* end-to-end delay of *System 1, iv)* long-term power consumption of the edge server, and *v)* feasibility constraints on involved optimisation variables.

Sketch of the proposed solution. The proposed long-term problem is solved via stochastic optimisation tools, in particular the Lyapunov-based approach also presented in [MBC22]. The details are omitted due to the lack of space and to lighten the reader. Lyapunov stochastic optimisation allows the decoupling of the problem into a sequence of per-slot problems, whose formulation includes properly defined state variables that track the behaviour of the system in terms of constraint violations and queue lengths. In this particular case, the per-slot problem is optimally solved through an exhaustive search on a limited set of variables and closed-form expressions. The decoupling is equivalent to solving the original problem, with a single hyper-parameter that trades off *System 2* data offloading rate and end-to-end delay, and edge server power consumption, while guaranteeing *System 1* requirements.

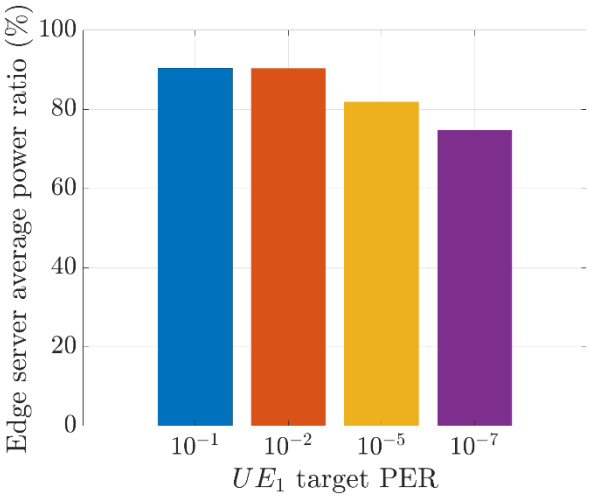
Numerical results and discussion. The system parameters used for the numerical evaluation are reported in Table 6: Simulation parameters.

Table 6: Simulation parameters

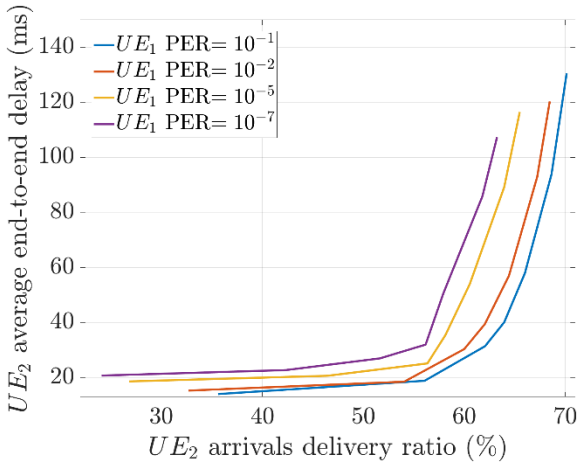
Parameter	Value
UE_1, UE_2, AP_1, AP_2 positions	[5,0], [7,10], [5,20], [8,20]
Number of AP antennas and received filters	8 with maximal ratio combining
Carrier frequency and channels	3.5 GHz, path loss exponent 4, Rayleigh fading
UE_1 transmit power	100 mW
UE_2 transmit power	In [0,100] mW with linear step and 11 values
Bandwidth	180 kHz fully shared
Noise power spectral density	-174 dBm with 3 dB noise figure at receiver
Number of bits transmitted by	1024
Computational load $\omega_1 (UE_1)$	10^7 CPU cycles
Blocklength	32 bits
Maximum CPU clock frequency	3.6 GHz
κ	10^{-27} [MBC22]
Slot duration τ	5 ms
UE_2 computational load	1000 CPU cycles/bit
UE_2 arrivals per slot ($A_{\max}$)	5 kbits



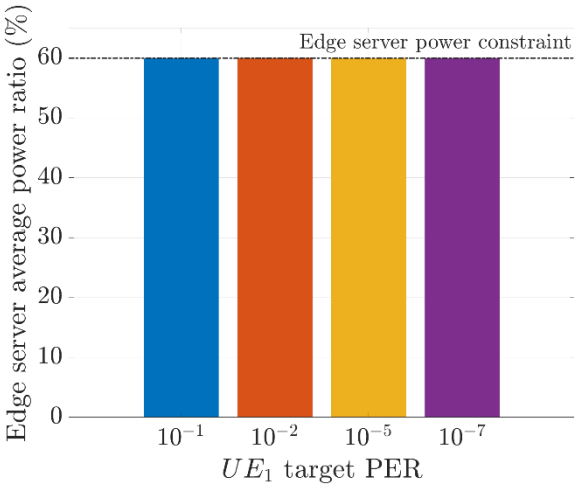
(a) trade-off between UE_2 average end-to-end delay and arrivals delivery ratio, for different UE_1 target PER values



(b) edge server average power consumption ratio vs. target PER



(c) trade-off between UE_2 average end-to-end delay and arrivals delivery ratio, for different UE_1 target PER values and constrained edge server power



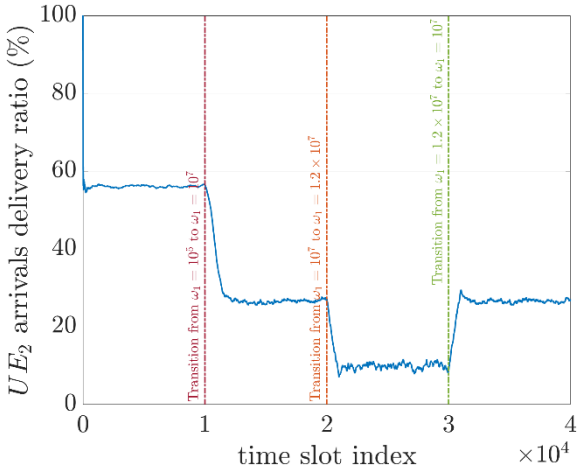
(d) edge server average power consumption ratio vs. target PER, in case of constrained edge server power consumption

Figure 4-17: Trade-off involving UE_2 data offloading rate and E2E delay, UE_1 reliability constraints, and edge server power consumption

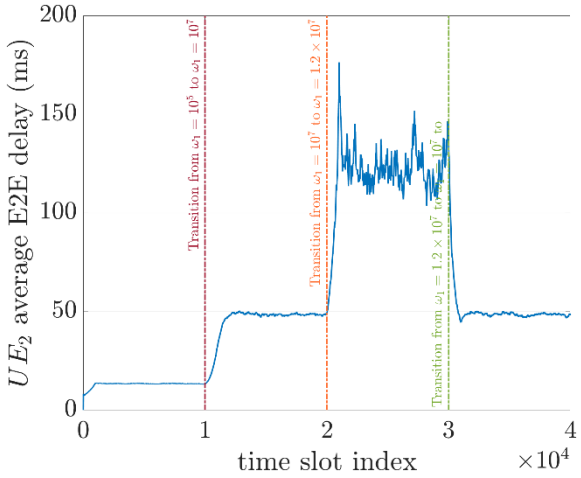
As a first result, Figure 4-17a shows the average end-to-end delay of UE_2 offloading service, as a function of the arrivals delivery ratio, for different values of target PER. In this case, no constraint is imposed on the edge server power consumption. As expected, the average delay increases as the arrivals delivery ratio increases, showing different trade-offs depending on UE_1 target PER. In particular, stricter UE_1 communication constraints lead to worse performance of the computation offloading services provided to *System 2*. This is due to the full sharing of spectrum resources. At the same time, as shown in Figure 4-17b, the edge server power consumption decreases as UE_1 target PER decreases. The edge server power ratio is computed with respect to the maximum power consumption in case of full CPU usage. This is due to the fact that, to ensure lower PER in *System 1, System 2* is required to decrease its transmit power and, as a consequence, the data offloading rate (as also clear from Figure 4-17a). The method is able to learn and adapt resource allocation, to match the system capacity by identifying the bottleneck. While in Figure 4-17a-b, no edge server power constraint is imposed, Figure 4-17c-d show the same trade-off, in the case the edge server is required to consume on average 60% of its full power. In this case it can be noted, in Figure 4-17d, how the adaptive method is able to guarantee

the constraint for all values of UE_1's target PER. At the same time, as visible from Figure 4-17c, the data arrivals delivery ratio is degraded with respect to the previous case, due to the reduced availability of computing resources, of course with different performance depending on *System 1* requirements (i.e., the target PER). In this case, the computing resources represent the bottleneck. However, this bottleneck chokes more arrivals if *System 1* requirements are stricter.

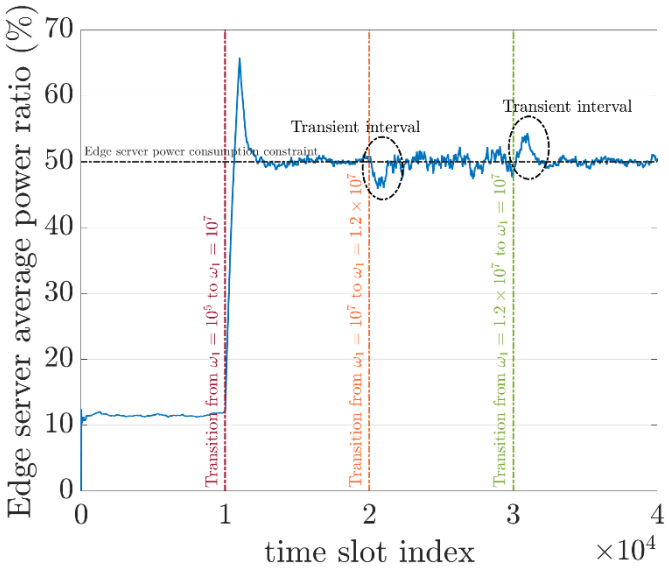
From a complexity point of view, the exploited optimisation approach helps decoupling the problem into per-slot programs that are solved through an exhaustive search over UE_2 transmit power, and with closed form expressions for the computing resource optimisation. It should be noted that, in the considered scenario with 2 UEs, the exhaustive search does not introduce dramatic complexity, which would however increase when scaling the system. In the latter case, heuristics are needed to reduce the complexity of the proposed solutions. This is part of future investigations.



(a) adaptation of arrivals delivery ratio



(b) adaptation of E2E delay



(c) – adaptation of edge server power consumption

Figure 4-18: Adaptation capabilities of the method

Conclusions and future directions. In this section, a joint communication and computation co-design approach has been proposed to identify the bottleneck of different computation offloading services sharing the same spectrum and computing resources, with a primary system owning the spectrum, and the second one adapting communication to maximise its performance while guaranteeing the first

system's requirements. The inherent coupling of radio and computing has been evaluated through a joint optimisation problem that has been solved through stochastic optimisation tools. Further directions include the investigation of more complex scenarios with more users and services, but also the evaluation of control performance based on the reliability constraints considered in this section, to close the loop of control, communication, and computation. A possible way forward is the integration of the proposed method with those proposed in Section 4.2.3.

4.2.3 Packet-loss-aware resource allocation

It is widely known that closed-loop control applications, e.g., the cooperative and mobile robot use case, may not be served over wireless networks that exhibit best-effort packet loss ratios (PLRs) and latencies because they typically fail quickly when they are not provided with QoS guarantees. This gave rise to the development of URLLC (ultra-reliable low-latency communication) wireless networks, which – in essence – has the goal to drive both metrics as close to zero as possible. However, as the wireless channel is typically subject to small-scale fading caused by multi-path propagation and the Doppler effect, the optimal set of transmission settings is commonly unknown immediately before each transmission, which can be shown through straight-forward information-theoretic calculations (refer to channel outage capacity). Only mid- and long-term average channel statistics are known – sufficient for high-data-rate operation but not for high-dependability/low-latency. Diversity, i.e., the parallel transmission over multiple diversity branches (frequency, space, time) is the most widely adapted approach to achieve URLLC, however, it is costly in terms of wireless resources and transmit power and therefore wastes spectrum and/or energy.

Taking a step back, the premise of requiring low PLR for closed-loop control applications was questioned in [Hexa- X D7.2]. It was determined that indeed low PLRs significantly stabilize the control loop, but also that a negative packet loss correlation can achieve the same goal while keeping the overall PLR in the best-effort region (which enables high spectral and energy efficiency). The state-aware resource allocation (SARA) scheme was developed, which effectively reduces burst errors by temporally negatively correlating packet losses at roughly the same cost as best-effort operation. It was shown by means of an AGV position control system how stable it can be operated while still exhibiting a PLR of around 10% (but with highly negative correlation).

In this deliverable, the single-hop analysis presented in [Hexa-X D7.2] is extended by:

- Dual-hop operation and how "packet consecutiveness" may be reformulated more generally as the age of information.
- A generalized optimization framework.

4.2.3.1 Extension to the dual-hop network case

The term "agent" is introduced to refer to a sensor/plant/actuator in any network configuration.

A dual-hop architecture is considered with two transmissions in either UL or DL, or one successive UL and DL connection (see Figure 4-19). This network mode is relevant for wireless sensor networks with relaying nodes and networked closed-loop control applications.

Generally, the UL is assumed to connect an/the agent to a network function (e.g., the networked controller) and the DL connects the network function (back) to the agent (in the dual-hop case through an intermediate node). The term *transmission cycle* is introduced to describe the data processing chain

$$\text{data generation} \rightarrow \text{UL/DL transmission} \rightarrow \text{processing} \rightarrow \text{UL/DL transmission}$$

Each transmission cycle is periodically triggered with the fixed sampling period T_s, which is common for industrial applications [5GACIA20].

The connectivity model bundles the following assumptions:

- small-scale fading is the main cause of failure
- an isochronous communication mode is employed with two transmissions within one transmission cycle

- there are no retransmissions as this leads to unpredictable resource consumption, which might have side effects for other applications served over the same network
- the network manages channel access and a time-frequency grid exists (similar to that in 5G), which allows to assign multiple parallel links to any connection

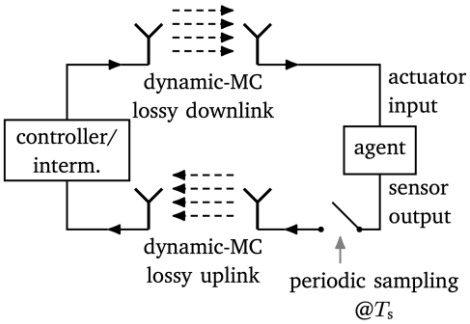


Figure 4-19: Considered network model in the dual-hop network case. A sensor output triggers a transmission cycle consisting of UL transmission, data processing, and DL transmission.

In the single-hop case, the current number of experienced consecutive packet losses could be taken as a direct indicator for the importance of that particular connection to succeed in the next transmission attempt. In the dual-hop case, this is not possible anymore because there are two consecutive transmissions (e.g., one UL and one DL transmission) and it is very relevant for the propagation of information, where the transmissions error occurs. For example, it would not make sense to make the UL more reliable (by spending more resources) when actually the successive DL is the reliability bottleneck. This is in line with the investigations carried out in Section 4.2.2: identifying the bottleneck is a fundamental step towards efficient and effective resource optimisation.

The more general question to ask when assessing the importance of a packet transmission on any given link is: How much does this packet transmission contribute to keeping the information at the data sink fresh/up-to-date? In other terms: How much does a certain packet transmission contribute to keeping the *age-of-information* (AoI) at the data sink low? And successively: How many resources should be spent on that transmission? Figure 4-20 illustrates the relation of latency/packet losses/sampling with the AoI through the well-known saw-tooth diagram. Since the latency-induced AoI can be regarded as low in the presented network models (isochronous operation mode assumption with no retransmissions), and the sampling-induced AoI only depends on the application-specific sampling-period T_s (which is set by the application engineer), variations in the AoI are mainly determined by packet losses. Note here that for the single-hop network case, given a maximum AoI bound, the sampling period T_s and maximum allowable number of consecutive packet losses form a simple multiplicative relation. With a dual-hop network, this relation is not so straightforward anymore as will be discussed in the following.

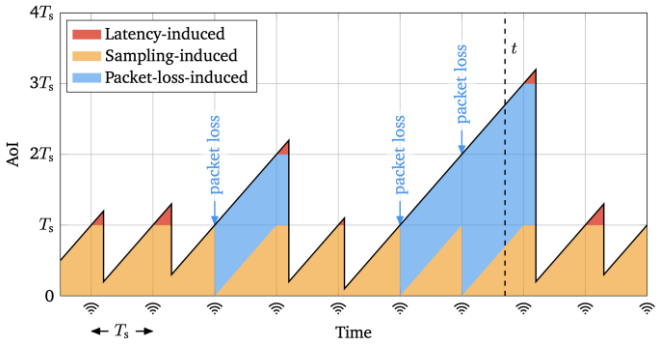


Figure 4-20: The age of information is the "super metric" encompassing the latency, packets losses, as well as the effect of sampling alone.

In [Hexa-X D7.2], a state model was discussed that relates the current number of consecutive packet losses to a certain state, which then triggers assigning a certain number of resources associated with a state transition probability p_k and $\tilde{p}_k$, respectively (Figure 4-21) The dual-hop extension is depicted in Figure 4-22.

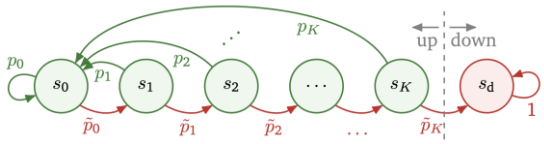


Figure 4-21: Single-hop network case failure model [Hexa-X D7.1].

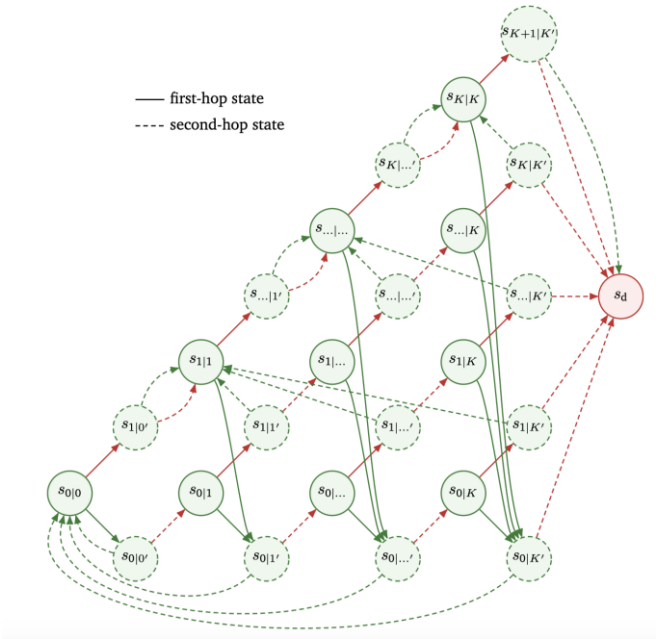


Figure 4-22: Extension of the failure model to the dual-hop network case.

In this model, it is assumed that the intermediate node always tries to forward the most current data in an attempt to decrease the AoI at the destination/data sink. For example, if at time t the first-hop transmission was successful, but the second-hop transmission has failed and at $t+1$, the first-hop transmission has failed, the second-hop transmission will attempt to forward data based on the information provided at t (the most current).

In the model, each state $s_{k_{1h}|k_{2h}}$ refers to having an AoI of k_{1h} at the intermediate node and an AoI of k_{2h} at the destination. For example, in $s_{1|1}$, the currently available information at both, intermediate node and destination, is one sampling period old. As before, green transitions refer to successful transmissions, red transitions refer to unsuccessful transmissions. As stated above, one transmission cycle consists of two transmissions (first hop and second hop) in the dual-hop case. States encircled with a solid line refer to the AoI situation right before a first-hop transmission attempt and are termed *first-hop states* in the following. States encircled with a dashed line describe the AoI situation within a transmission cycle, i.e., after the first hop but before the second hop, and are termed *second-hop states*. All second-hop states are marked additionally through a dash in the second index to highlight that a transmission cycle is in progress and the data is about to be transmitted on the second hop, which will update the AoI at the destination.

The transition probabilities are omitted in Figure 4-22 for readability but are indexed similarly to the single-hop case. Successful transmissions originating in $s_{i|j}$ and $s_{i|j\prime}$, respectively, are denoted by a

transition probability of $p_{i|j}$ and $p_{i|j\prime}$, failed transmissions occur with a probability $\tilde{p}_{i|j} = 1 - p_{i|j}$ and $\tilde{p}_{i|j\prime} = 1 - p_{i|j\prime}$.

The intermediate second-hop states are motivated by the possibility that the information about the success/failure of the first-hop transmission may be used to adjust the success probability (by providing more/fewer channels) for the subsequent second-hop transmission within the same transmission cycle. This is possible in this model because it is assumed that the time between the first and second hop, during which the intermediate node processes the signal, suffices to make a resource assignment decision for the second-hop transmission.

It is apparent that the AoI at the intermediate node cannot exceed the AoI at the destination in between transmission cycles (i.e., for all first-hop states) because the destination is behind the intermediate node in terms of information flow. Hence, the Markov model is composed of a triangle structure. The AoI dependency of the second hop on the first hop also manifests itself through the fact that after a successful second-hop transmission, the destination AoI $k_{2\mathrm{h}}$ can only drop to the current AoI at the intermediate node $k_{1\mathrm{h}}$ instead of zero.

The determining KPI for an agent's failure is the AoI at the destination $k_{2\mathrm{h}}$. The system is assumed to enter the absorbing down state s_{d} if $k_{2\mathrm{h}} > \mathrm{AoI}_{\mathrm{peak}}$ (where $\mathrm{AoI}_{\mathrm{peak}}$ is an application-specific threshold) at which point the control system is assumed to switch off.

It is noteworthy that in certain states, there is no updated information available at the intermediate node, which makes any subsequent second-hop transmission obsolete. For example, in $s_{1|0\prime}$, the most recent first-hop transmission has failed, and the destination already has all the information that the intermediate node has. This is indicated in Figure 4-22 through a double transition (green + red) as the second-hop transmission success/failure has no impact on the AoI at the destination. This is also true for $s_{K+1|K\prime}$, at which point the application is already condemned to fail after the next transmission (the state, however, is included for completeness).

Through this extended modelling, it becomes clear how the AoI extends the single-hop concept of "consecutive packet losses" to dual-hop/multi-hop networks and is therefore the more general concept. While the AoI is able to fully capture the consecutiveness of packet losses when applied to single-hop networks, it generalizes the concept to "information freshness" in dual-hop and multi-hop systems.

4.2.3.2 Generalized optimization framework

Similar to state-aware resource allocation in the single-hop case, the number of channels used for first-hop (solid states) and second-hop transmissions (dashed states) shall be adjusted according to the history of packet losses. However, as the AoI at the destination is now affected by two transmissions, a holistic resource allocation including the AoI at both, the intermediate node and the destination, may be beneficial.

Three objectives are pursued:

(a) increase the mean time to failure (MTTF)
(b) reduce the average number of channels $\bar{l}$ in both transmissions, and
(c) ensure that high AoIs only occur with a low probability.

As these objectives partly counteract another, the derivation of appropriate SARA schemes may be understood as a multi-objective optimization problem, constrained to an integer number of links. Formally:

$$\boldsymbol{l}^{\mathrm{opt}} = \arg\left(\min_{\boldsymbol{l}}\left(\Gamma\left(g_{\mathrm{MTTF}}\left(\mathrm{MTTF}(\boldsymbol{l})\right), g_{\mathrm{AoI}}\left(\mathrm{AoI}(\boldsymbol{l})\right), g_{\bar{l}}(\bar{l}(\boldsymbol{l}))\right)\right)\right) \text{ subject to } l_k \in \{1, \ldots, \hat{l}\} \qquad (4.1)$$

All functions $g_{(.)}$ constitute penalty functions, which must be carefully designed by control system engineers, as MTTF and $\bar{l}$ directly influence operational expenditure, while the PMF of the AoI influences control performance. The function Γ combines all penalty functions to a single value.

For a given maximum acceptable AoI K, the objective is to find the optimal solution among a set of $(Z_{1\mathrm{h}} + Z_{2\mathrm{h}})^{\hat{I}}$ possible resource assignment schemes, where $Z_{1\mathrm{h}}$ and $Z_{2\mathrm{h}}$ denote the number of first-hop and second-hop states, respectively, of the failure model depicted in Figure 4-22.

$$Z_{1\mathrm{h}} = \sum_{i=1}^{K+1} i$$

$$Z_{2\mathrm{h}} = \sum_{i=1}^{K+2} i - 1$$

Since the problem is (a) non-convex, (b) integer-constrained, and (c) grows quickly with K, therefore rendering exhaustive search infeasible for $K > 3$, MATLAB's Surrogate Optimization from the Global Optimization Toolbox is used.

Due to the heterogeneity of potential requirements for control systems, there is not one single optimal SARA solution and the choice of all functions $g_{(.)}$ and Γ is highly subjective. The form (exponential, polynomial, etc.) and weights (prefactors, exponents, etc.) of all functions $g_{(.)}$ and their relation to one another must be carefully balanced to reflect the trade-off between operational expenditure and control performance. The contribution here is not a single optimal SARA solution but should be understood as a framework that yields optimised AoI-based resource allocation for dependable real-time applications.

4.2.3.3 Extensive simulation

In [Hexa-X D7.2], it was shown that a good control performance may be achieved over an error-prone wireless network as long as the temporal packet loss correlation is highly negative. To describe this temporal correlation, an artificial packet loss correlation variable ρ was introduced. Here, extensive simulations of an automated guided vehicle (AGV) traversing a given path were carried out and the results are displayed in the form of a path deviation and acceleration signal (control value) in Figure 4-23 and Figure 4-24. Figure 4-23 builds the reference static single-connectivity case. Note that the MTTF is only 2 minutes (even with tolerating a maximum AoI of $K = 3$), which does not constitute ultra-dependable operation. With a reasonable set of optimization target functions, the extensive simulation results for an optimal SARA scheme according to an example set of exponential penalty functions are plotted in Figure 4-24. Note that from 10^6 simulation runs, the red graph displays the maximum value among all runs and the blue graph the minimum value. It can be seen that the path deviation (left side) is significantly more deterministic than that of single-connectivity. Also the control value (right side) is more deterministic for the SARA scheme and less deterministic for single-connectivity. At the same time, the MTTF is improved by a factor 2×10^7 (sic!) over single-connectivity (2 minutes compared to 74 years) while spending only 18% more resources (in terms of average number of channels used).

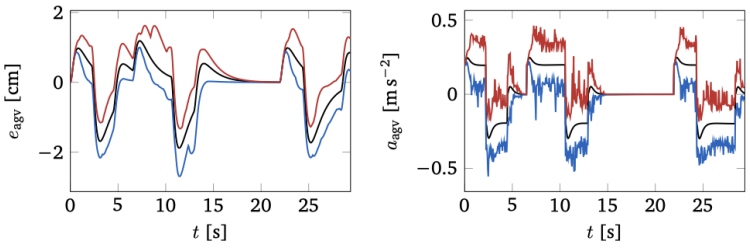


Figure 4-23: Single-Connectivity, $p_{\mathrm{loss}} = 10\%$, $K = 3$, MTTF = 2 minutes, $\bar{l} = 1$. The red curve denotes the maximum value out of 10^6 simulation runs, the blue curve the minimum, and the black curve the case without any packet losses. On the left, the deviation from the planned trajectory is depicted. The right shows the acceleration value.

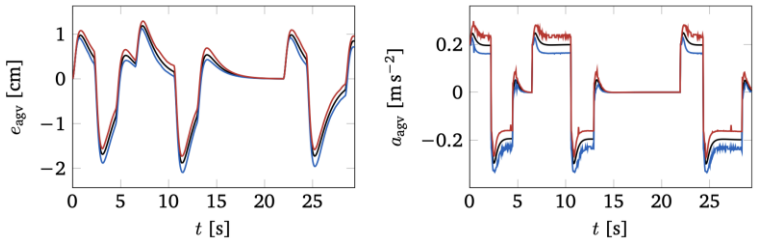


Figure 4-24: SARA, $p_{loss} = 10\%$, $K = 3$, MTTF $= 74$ years, $\bar{l} = 1.18$. The red curve denotes the maximum value out of 10^6 simulation runs, the blue curve the minimum, and the black curve the case without any packet losses. On the left, the deviation from the planned trajectory is depicted. The right shows the acceleration value.

4.2.3.4 Non-Integer-constrained optimization

In the previous section, it was shown how the AoI-based SARA framework may be utilized to derive a per-channel packet loss probability requirement p_{loss} that ensures desired KPIs relevant to real-time operation in industrial scenarios, namely MTTF, $\bar{l}$, and AoI. Thereby, the number of channels, which can be assigned to an agent in parallel, was assumed to be integer-constrained, i.e., $l \in \{1, \dots, \hat{l}\}$, which causes discontinuities in all individual penalty values of Figure 4-25.

This integer constraint was motivated by multi-connectivity as a tool to increase a transmission's success probability, enabled by selectively combining an integer number of uncorrelated system channels that individually feature an identical packet loss probability p_{loss}. This way, the combined packet loss probability for a transmission could be controlled by the number of parallel channels assigned for one transmission. In this section, this assumption is replaced by allowing to divide the available spectrum arbitrarily. The derivation of a PHY and MAC that allows such arbitrary distribution will be left for future work.

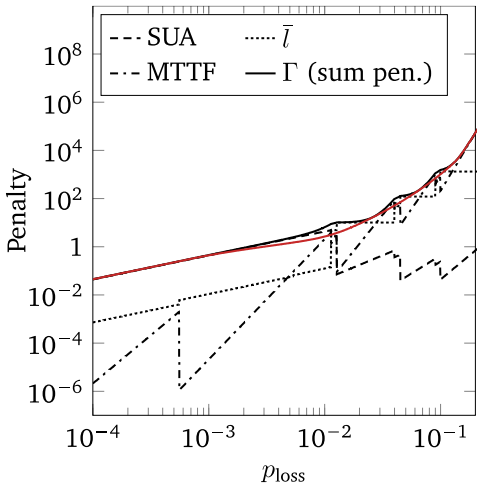


Figure 4-25: The integer constraint on the number of channels causes discontinuities in the individual penalty contributions. Removing the constraint yields a smoother and more optimal sum penalty (red).

The optimization of Eq. (4.1) is repeated without the integer constraint ($l_k \in [1, \hat{l}]$) and the results are plotted in Figure 4-25 (red) for a direct comparison with the constrained case (blue). As expected, the discontinuities in the individual penalties are resolved and intuitively, the optimal non-constrained sum penalty Γ lies below the constrained sum penalty (because – per definition – it is not constrained).

This demonstrates that the presented framework is not restricted to networks with an integer-constrained number of parallel channels, but that it can also be applied to networks that are able to dynamically (on a per-transmission basis) alter the transmission success probability by more elaborated means than multi-connectivity with selection combining. It remains an open research question, how channels may

be mapped to physical resources (in the time-frequency-coding space) to yield optimal spectral efficiency for a given target transmission success probability. With such a mapping, the penalty function for the average channel consumption $\bar{l}$ in Eq. (4.1) could be replaced by an advanced penalty function that penalises actual PHY resource usage.

5 Digital Twins and novel HMIs

As discussed in [Hexa-X D7.1] and [Hexa-X D7.2], novel technologies of DT and HMI are playing a key role in enabling new 6G use cases and establishing the 6G human-centric industry ecosystem. Built on the intermediate analyses and results in [Hexa-X D7.2], this section presents the novel contributions of Hexa-X regarding the novel HMI and DT technologies and their applications in future 6G systems.

5.1 Novel HMI for mobile human-machine and human-CPE interaction

Several types of feedback in multi-sensory HMI technologies have been named in [Hexa-X D7.2], including holographic vision, tactile, smell, and taste. Especially, there is an intense recent interest of employing holographic vision interfaces in future 6G-driven industrial use cases, such as collaborative robots and massive twinning with human in the loop.

5.1.1 Holographic vision for collaborative robots

The terminology holographic vision, also known as Augmented Reality (AR), is used to refer to a family of technologies that allow a user (human) to access some piece of digital information superimposed on top of the real world. There are three main ways of providing content to the user:

- Spatial AR (or projection-based AR) where digital content is projected directly on top of the physical environment or integrated by means of large displays. The drawback of this approach is that is rather difficult to stereoscopically display 3D objects to the end-user.
- Hand-held AR, which can be delivered easily through smartphones or tablet devices; in this case, the digital content is usually combined with a video camera feed and shown to the user using the display of the device. This category is often less preferred in the industrial domain, since it prevents the user to use his or her hands freely.
- Head-worn AR, the most common form of augmentation that takes advantage of optical see-through lenses built-in HMD allowing the user to directly perceive the real world, and augmented content is superimposed from his or her point of view. Recently, also video see-through HMDs are reaching a good degree of diffusion, but their use in industrial applications is still scarcely explored.

The application of holographic vision in collaborative robotics is a promising approach to enhance the interaction between collaborative robots (cobots) and humans in a shared workspace. The combination of these technologies enables experiences that provide a new level of awareness and understanding of the cobot work environment. For instance, the operator's ability to understand the cobots' movements and anticipate potential issues can be enhanced by exploiting holographic vision. In fact, holographic vision can be exploited to provide human operators with visual (and auditory) cues about the cobots' perception, planned trajectories of the cobot, the status of the machinery available in the workspace, digital twins (DTs) or replicas of tools or workpieces, and so forth. These functionalities can be applied to a variety of scenarios and use cases ranging from cobot programming [Hexa-X D7.2], to supporting an efficient production cycle, remote assistance, or operator training, as discussed later in Section 5.3.

5.1.2 Spinal cord and brain stimulation

Besides the classical approach of exploiting the natural human perception organs, it is also possible to generate artificial sensory feedback by directly stimulating the human neural system, i.e., spinal cord and brain. With a long history in the field of neuroscience, this concept has been so far mainly studied in the context of medical applications, e.g., prosthetic implants that can simulate tactile sense for amputees [TSK+15]. Over the past years, significant research efforts have been made to merge such technologies with the future 6G wireless solution towards a so-called brain-type communications (BTC) scenario [MNB+21], which is supposed to exhibit outstanding effectiveness of triggering sense in

flexible environments. However, the concept of BTC is remaining challenged by the high implementation cost and potential risks in human safety.

5.1.3 Synesthesia

Additionally, the so-called synesthetic phenomena, in which one sense can be triggered by another, raises research interest for its potential in certain use scenarios. For example, there are both behavioural [PBT+18] and neurological [LGR+18] evidence, that certain type of videos/audios can trigger in some audience a special gentle touch-alike tactile sense accompanied by a deep relaxation and pleasure, which is known as the autonomous sensory meridian response (ASMR). Such observations are suggesting a new class of solutions to resolve/soften the negative mental status (such as depression, stress, anxiety, and anger, as identified in [Hexa-X D7.2]) of human participants in industrial processes, without introducing significant disturbance to them.

To explore the potential of ASMR audios in relaxation and demonstrate the feasibility of exploiting them, we conducted a cyber-psychological study that is reported in [FHC+22]. First, a set of 10-second clips were taken from four audios recorded from real sounds of different natures, including breathing, mixing soft cream, puffing a spray, and typing on a keyboard. Spectral and cyclostationary features were extracted from each audio clip to capture its acoustic characteristics. More specifically, the mean and variance of Discrete Fourier Transform (DFT), the mean, variance, and Gini coefficient of the average spectral correlation function (SPD), as well as the mean, variance, and Gini coefficient of the peak SPD were obtained. Afterwards, an online psychological survey was conducted to evaluate the emotional impacts of each audio clip on voluntary human participants when played to them. The collected behavioural data was then analysed with the extracted acoustic features to evaluate the correlation between them. More specifically, a linear mixed-effect regression model (LMM) was applied. The regression results revealed a significant relation between the extracted audio features and the triggered emotional effects, but showing very low linearity. Furthermore, deep convolutional generative adversarial networks (DCGANs) were used to synthesize random artificial ASMR audio clips with similar acoustic features, which were observed to generate similar (but slightly weaker) psychological effects in the behavioural experiment, as shown in Table 7.

Table 7: Perception of feelings triggered by ASMR audio clips

Perceived Feeling	Average of recorded clips		Average of synthesized clips	
	Mean score	Deviation	Mean score	Deviation
Negative[1]	1.6244	0.7302	1.5827	0.6353
Positive[1]	1.5527	0.9109	1.2521	0.3605
Relaxed[1]	1.6489	0.9953	1.3924	0.5093
Attentive[1]	2.4112	1.6582	2.0072	1.1980
ASMR experience[2]	0.6552	0.7727	0.5093	0.6232

1: graded into one of five levels, from 1 (not at all) to 5 (extremely)

2: graded into one of three levels:0 (no), 1 (almost), or 2 (yes)

This demonstration revealed to us the feasibility of exploiting synaesthesia as a new type of HMI solutions. Towards practical implementation and commercial deployment, deeper study is still required to setup a reliable model of the human sensory system, where DT of human can play an important role.

5.2 Modelling the impact of human presence on industrial deployment of Digital Twins

Here we consider the radio-aware digital twin discussed in Section 4.1.1, which is an up-to-date digital twin of the radio propagation environment. We employ the radio-aware digital twin to proactively manage radio-resources in a quasi-controlled environment. In such an environment, the position information of fixed objects in the propagation environment such as gNBs, walls, pillars, etc. are expected to be known with less than $\lambda/2$ precision. In addition, positioning of UEs and other mobile objects are expected to be known to a reasonable degree of accuracy (of the order of a few cm). Lastly, depending on whether the factory is a greenfield or brownfield setup, the materials and other properties are expected to be known to a reasonable degree of accuracy, with higher accuracy in the former.

In such an environment, the location and trajectories of AGVs and UAVs can be planned and controlled, and this can be done based on the radio-aware DT. For an environment where all objects are under the DT's control, the DT can leverage accurate knowledge of the propagation environment and location of objects within it to set the communication parameters aggressively and proactively such as training overhead, training interval, MCS, etc.

However, when humans are present in the propagation environment, their positions cannot be controlled. At mmWave frequencies, the effect humans have on the propagation environment depend on their physical properties such as height, size, clothing, body and limb location, and orientation. Not accounting for these factors could lead to errors in the digital twin. Human presence brings uncertainty and may hamper functioning, especially if the DT is setting parameters aggressively assuming a fully controlled environment.

Human models at mmWave are typically meant to capture the effect of a human body blocking the line-of-sight between a transmitter and receiver. Such models are typically designed to be *statistically* accurate for use in system-level simulations and predict the "average" effect of humans on a communication link. Deterministic models for body blockage also exist, but such models do not always extend easily to the case of multiple humans.

Lastly, the aforementioned models are for body blockage. However, models for human *presence* are limited. Like body blockage, human presence can be modelled either stochastically or deterministically, where the latter is done through ray tracing or other simplified models such as multiple knife edge or flat sheet.

In this work, we validate the need to account for human presence in a radio-aware DT. To accomplish this, we utilize geometrical optics (ray tracing) and physical optics solvers for modelling the impact of humans on propagation environment. Since we are working at mmWave frequencies, we replace human tissue with a perfect electric conductor and the human body as a hollow shell. These simplifications are valid at mmWave frequencies.

We consider a simulation at 30 GHz where we have a human placed at the end of an L-shaped corridor. The corridor has length 6 m, width 2 m, and height 3 m. The transmitter has a $\lambda/2$ patch antenna, whereas the receiver has $\lambda/2$ patch antennas arranged as an antenna array in a 4×2 configuration. A human is placed in the corner of the corridor such that the distance to each of the corner walls is 1 m. The position of the human is offset from this position on a grid of size 10 cm (i.e., one wavelength at 30 GHz). The orientation of the human is chosen from {0, 15, …, 90} degrees, where 0 degrees implies that the human is facing the transmitter and 90 degrees means that the human is facing the receiver. This setup is shown in Figure 5-1.

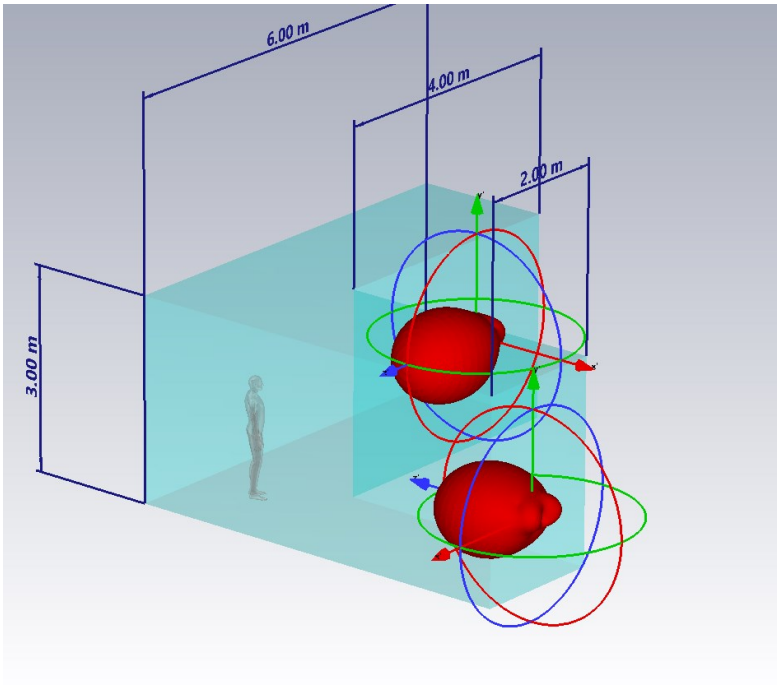


Figure 5-1: Simulation setup with the human in the corner of a narrow corridor. The dimensions of the corridor are 3m (height), 6m (depth), and 2m (width).

In this experiment, we assume that the physical optics solver is the ground truth. This assumption is justified by our earlier work presented in [Hexa-X D7.2] where it was shown that the physical optics can accurately predict pathloss.

In Figure 5-2 we plot the CDF of the pathloss for different locations of the human on the grid across all frequencies. It is evident here that there could be a substantial difference of the order of a few dBs in pathloss between the case where the human is present and the human is absent. Furthermore, since the human here reflects energy from the transmitter to the receiver, the pathloss also varies along with the orientation of the human.

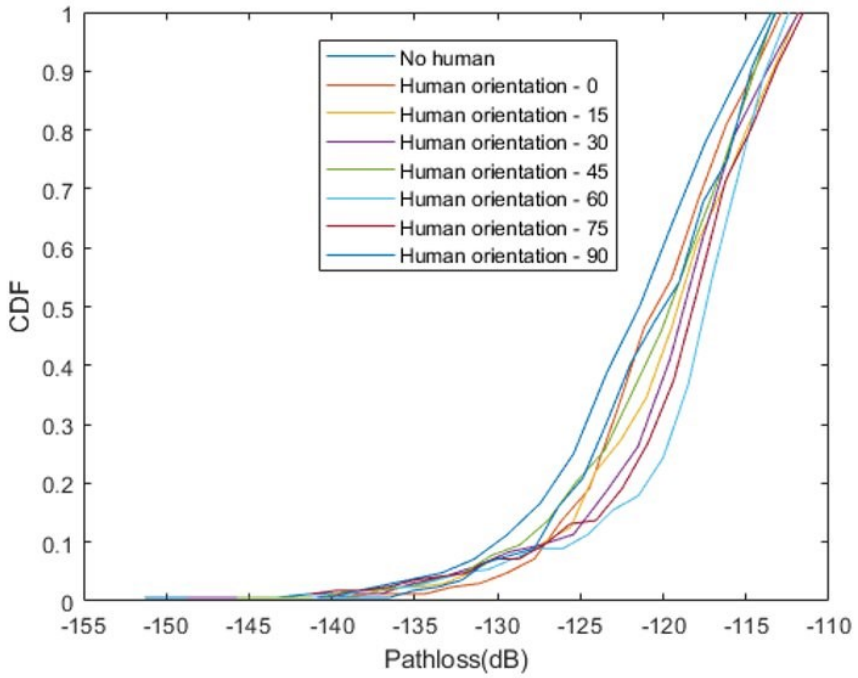


Figure 5-2: CDF of pathloss between transmitter and receiver

In Figure 5-3 we plot the pathloss vs. frequency for the same setup. Here, the effect of the human on the multipath characteristics is visible, where human presence reduces pathloss and changes the delay spread. Lastly, simple ray tracing does not predict all the paths/clusters hence shows a reduced frequency selectivity compared to the other methods that are considered. Hence, more advanced methods like physical optics are required to capture the different propagation effects.

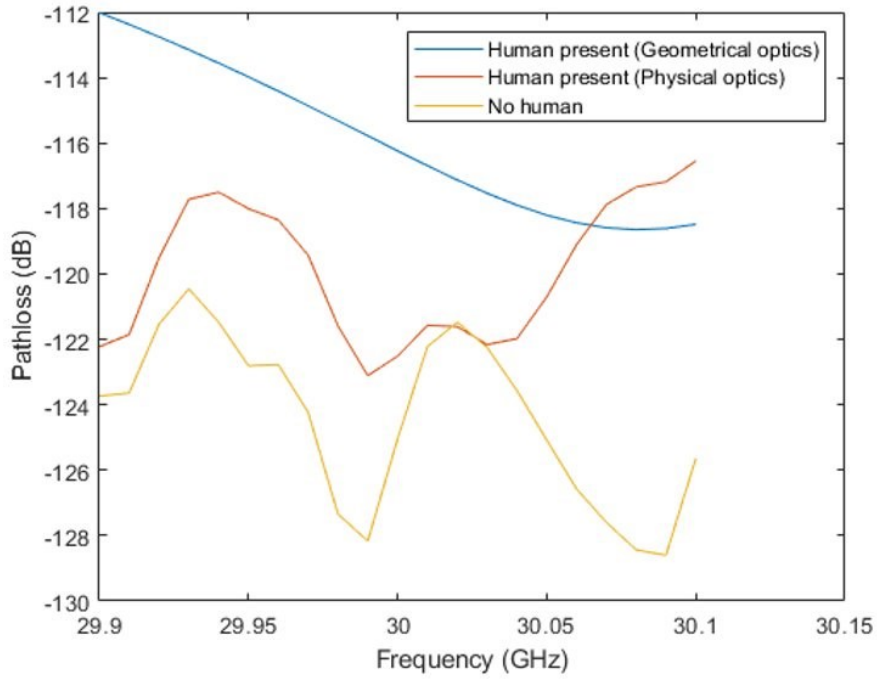


Figure 5-3: Pathloss between transmitter and receiver when human is placed with zero offset and is oriented at an angle of 45°

From the figures, we conclude that human presence needs to be modelled NLoS scenarios, in addition to the LoS scenarios, since it affects pathloss. In the absence of such modelling, aggressive radio-resource management by the radio-aware DT may cause link errors. Alternatively, in the presence of a human in the environment, the radio-aware DT may need to adopt a more conservative approach to resource management in the absence of modelling.

5.3 Digital-Twin-empowered collaborative robots

Activities illustrated in this section have been performed as a continuation of the work reported in [Hexa-X D7.2], which focused on the analysis, from a 6G perspective, of an approach designed to support the remote programming of a Collaborative Robot (cobot) featuring Digital Twin (DT)- and eXtended Reality (XR)-based tele-presence capabilities.

5.3.1 Introduction

The new activities reported herewith focused on extending two capabilities that were not yet particularly developed in the first design. These are the DT of the workpiece, in terms of localization (position and rotation) and visualization (3D mesh), and the DT of the LU, in terms of a full-body pose reconstruction (i.e., not limited to the head pose as in the initial design). In these contexts, Computer Vision- (CV) and more in general, Artificial Intelligence- (AI) based techniques find ample space, enabling new and more sophisticated HMI scenarios. At the same time, the employment of these technologies to implement the new capabilities inevitably leads to additional connected hardware (e.g., sensors and processing nodes), and the previously considered network requirements need to be re-evaluated.

To investigate the potentialities of the new iteration over the said design, the original use case (remote collaborative programming) was considered as not fully representative, as the two users were assumed to be skilled in the considered procedure. Hence, it was decided to test the capabilities of the new iteration on a different, more traditional scenario, quite typical in the training context. In this case, on the one hand, the LU is typically an operator who has no experience about how to perform a given procedure with the cobot, and is not well-trained in terms of human-cobot interaction.

The extended XR platform was evaluated against this scenario, by building on the laboratory setup already employed in the previous experiments. Network measurements were executed on each network node, during the repeated execution of the given task by pairs of individuals playing the parts of the inexperienced LU and the experienced RU. The collected results were used to update the previous requirements in terms of network dependability, as well as to evaluate the possibility to deploy the scenario on current and next-generation mobile networks.

5.3.2 Background

In this section, the reasons behind the modifications with respect to the original setup [Hexa-X D7.2] are reported and discussed.

5.3.2.1 Extended Digital Twin reconstruction

As said, the first iteration of the XR-based telepresence platform was designed to take advantage of any appliance that can provide its DT reconstruction (e.g., the cobot and the VR system of the RU). In addition to this, an RGB-D camera was used to provide a reconstruction, in the form of a point cloud, of the other workspace elements, similar to what proposed in [PGB+21]. Although the presence of the RGB-D camera allows a visual representation of the LU and the workpiece, the result is not a true DT, as information regarding the state of the reconstructed elements is not provided to the XR platform. Focusing on the working area (in particular, on the workpiece), a viable solution is to perform a kind of six degrees of freedom (6-DOF) tracking of the elements which cannot provide a DT by themselves. By leveraging one of these techniques, it is possible to provide a DT of passive but known elements present in the working area. The DT can then be used to perform automatic checks on the actual state of the real element (e.g., the system can detect the presence of a new workpiece on a roller conveyor).

At the same time, a possible remote (VR, in the considered scenario) user can be made aware of the evolution over time of the real workspace.

5.3.2.2 Human pose detection

The representation of the user, namely an avatar, is a critical aspect to cope within a shared collaborative Virtual Environment (VE) [BYMS06]. In the first iteration of the XR platform, the system provided an avatar reconstruction only for the RU. This simplification allowed to avoid the use of additional hardware and had a limited impact on the use case, as the LU was not required to know the accurate pose of the other user. However, in the case of different use cases, the knowledge of the actual pose of the LU's body may play a more important, if not fundamental, role. For example, the RU, playing the part of a remote expert, may be in charge of identifying possible hazardous situations or misbehaviours related to the actions of the LU in the vicinity of the cobot. In the previous programming use case, the two users were considered as comparable in terms of expertise, and the presence of the point cloud provided the RU with a rough but sufficient representation of what was happening in the real robot cell. However, if the actual pose of the LU becomes more important, both the RU and the system may require this kind of numeric information, which cannot be easily inferred from the point cloud.

Human-pose estimation is one of the most widely studied use cases involving the use of CV and ML. In general, it consists of the detection of specific points of a human body (joints or keypoints) from images or video feeds [KDF+23]. OpenPose [ZHT+21], BlazePose/MediaPipe Pose [BGR+20], and YOLOv7 [WBM22] are just few relevant examples of these techniques. The possible uses of the information related to the LU's body pose are manifold. For example, if the pose of the LU becomes the object of an evaluation (e.g., in a training use case), not only the RU would require a more precise visual representation of the LU's body, but also the system may automatically detect body stances associated with overfatiguing, errors or hazards, and signal them to both the LU and RU. Data coming from a pose estimation algorithm may be also used to precisely drive a visual full-body avatar representation like the one already used for the RU, but with a higher level of fidelity with respect to the use of IK.

5.3.3 Use case – multi sensor-based remote training

As mentioned above, the use case explored in the previous analysis was updated in order to cover a number of aspects for which the original setting was not suitable.

It was decided to move from a remote collaboration scenario in which both users shared the same level of expertise, to a remote assistance one, in which the LU has to be trained on a given task by the RU. In this context, is assumed that the LU has never seen or worked with a cobot, but is widely experienced for what it concerns the actual task that has to be performed in collaboration with it (it can be assumed that, before introducing a cobot, the given task was executed by a human operator alone or in cooperation with another operator). The main reason behind this choice was to move the attention of the RU from the cobot programming to the actions performed by the LU, hence giving more importance to the LU's visual representation (and related actions).

Another important difference with the previous use case is related to the cobot presence itself. Being the LU completely inexperienced in human-robot collaboration, it is reasonable that the first training steps should not involve the use of a real cobot (e.g., for safety reasons). In the considered use case, the LU is an operator of a production line/cell, that is planned to be completed by the introduction of a cobot, and he or she is requested to be trained to work with the robot in advance as part of a pre-commissioning step. This allows the operator to be ready to engage with the robot at the time of its integration. A similar use of a shared XR experience (VR for the remote expert and AR for the assisted operator) to provide remote assistance in a manual task was explored by Wang et al. in [WWB+23].

The selected task that the LU will have to perform in collaboration with the cobot is the riveting of a metal plate. This task was selected since common in the context of human-robot collaboration [LMS+19], and explored in training scenarios similar to the one here described in which the human operator is requested to practice with a simulated cobot, physical tools and cell environment

[MVGM17]. In the devised use case, the simulated cobot is equipped with a riveting flange, and the role of the operator is to position and keep in place the plate during the riveting operation performed by the cobot. Between one riveting action and the next one, the operator must reposition himself or herself to clear the path for the following motion. In this context, the workflow of the remote training is the following:

1. The LU tells the RU how the task shall be performed, and the role that the cobot should have in the production cycle.
2. The RU programs the movement of the simulated cobot in VR based on the information provided by the LU, in order to represent how the real cobot will likely behave when deployed in the cell.
3. The LU receives a preview of the planned motion in form of AR content (positioned where the real cobot will be located) and can evaluate its motion within the real environment.
4. The LU attempts the execution of the task in collaboration with the simulated cobot, while the RU supervisions it. During the execution, the RU can see the virtual robot, the DT of the workpiece, and a representation of the LU's body pose. On the basis of this information, he or she can provide advice or report errors and unsafe behaviours in terms of human-robot interaction.
5. The LU reiterates the actions until he or she feels comfortable, or until the remote expert considers it necessary.

In case the cobot program is already available at the training time, the initial two steps can be avoided as the RU would not be required to program the robot on the fly. However, for the sake of the current evaluation from a networking standpoint, it was decided to include the first programming step, assuming that the robot program was not available at the time of the training. In that case, being the LU not experienced in cobot programming, the RU is the one mostly in charge of preparing and deploying the program.

5.3.4 Materials and methods

In this section, the implementation of the new functionalities of the XR platform required for the experiment, the modified experimental setup, and the methods adopted for the analysis are reported.

5.3.4.1 Protocol

The alternate programming concept [CPC+22] was the focal point of the original investigation, and the basis on which the previous iteration of the XR platform was designed and implemented. Despite the change of focus in terms of use case, the possibility for the RU to create a "slice" of the cobot program (referred to as "clip") and send it to the LU (and then, to the real robot) resulted as particularly useful for the new use case as well. In particular, since the real cobot is not available in the LU workspace, the RU can take advantage of the remote programming feature to draft a rough version of the motion that the cobot will likely perform during the foreseeable production (Figure 5-4). This is done based on the information provided by the LU, which has to describe his or her needs (e.g., which parts of the task are performed by the LU, which ones will be delegated to the cobot, or when they will have to interact).

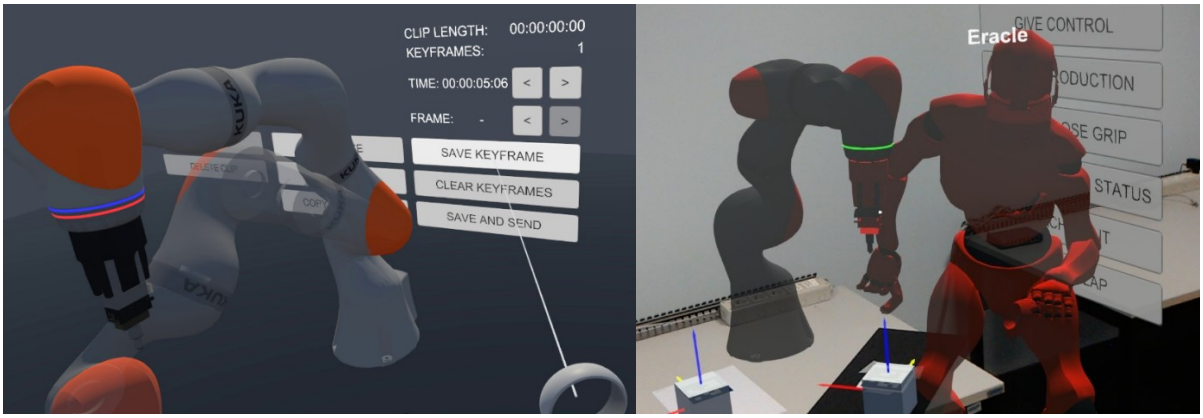


Figure 5-4: RU (expert) in the act of creating a program clip to be used to train the LU (left), and RU's avatar as seen by the LU during the same activity (right).

Afterwards, the LU will execute the program on the local (simulated) cobot to verify if the motion covers all the aspects required to complete the task. If this is the case, the planned motion is accepted, and the program is executed. During the execution, the LU performs his or her part of the task, possibly collaborating with the simulated cobot when required. At the same time, the RU observes the LU's actions and can intervene at any time. The resulting production cycle can be exploited to repeat the training till desired.

5.3.4.2 Extension of the XR platform

Although the XR platform was capable of supporting a remote assistance session on a real cobot, some modifications were necessary to support the functionalities required for the considered use case. Modifications included the positional tracking of the workpiece, the full-body pose estimation of the LU, and the possibility to replace the real robot with a simulated one.

Architecture

The foundations of the system were not modified, hence most of the architecture was kept unchanged. The new experimental architecture is depicted in Figure 5-5.

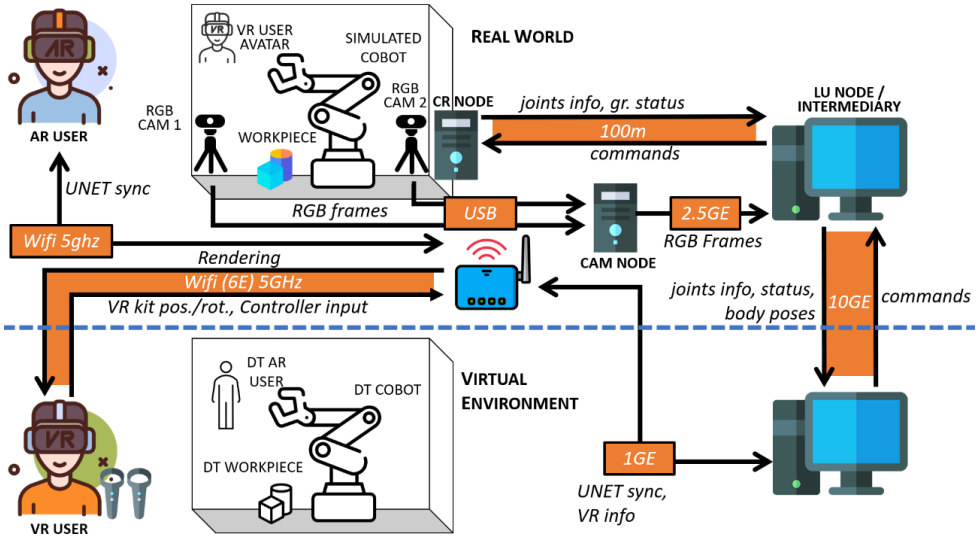


Figure 5-5: Architectural overview of the modified concept scenario that has been implemented in the laboratory for the new analysis, based on [Hexa-X D7.2].

The *LU* and *RU nodes* were again hosted on the same machines of the reference work. The *LU node* was connected to a node emulating the real robot controller (*cobot node*).

The *Intermediary* broker application, based on the *ZeroMQ* (ZMQ) library, was again deployed on the *LU node*, whereas the *RU node* acted as ZMQ node (*VR client*) and as *UNET host* (contemporarily client and server) for the XR platform. The connection between the AR HMD (running a *UNET client*), the VR HMD, and the *RU node* was again provided with a 5Ghz Wi-Fi 6E router (ASUS RT-AX55).

The configuration of the XR Platform was similar to that of the original setup, the AR and VR HMDs did not differ from the previous iteration (a Microsoft HoloLens 1st Gen, and a Meta Quest 2), and the experience was not significantly modified in terms of user interface and experience.

The first relevant difference with respect to the original configuration is the removal of the Depth camera, since in the selected use case it is assumed that all the relevant elements of the working area are going to provide a DT. In its place, two generic RGB cameras, configured to run at 30fps with a resolution of 1280x720 pixels, were used to capture two points of view of the working area for the human pose estimation algorithm. The cameras were connected to the USB ports of a dedicated node (referred to as the *CAM node*). This choice allowed the possibility to take two synchronized frames (one per camera) without the need for additional synchronization logic. The *CAM node* was then in turn connected through a USB3 to 2.5GE Ethernet dongle to a 10GE Ethernet port of the *Intermediary* (*LU node*), on which the *Human Pose Estimation client*, described in the following, receives the synchronized frames used to estimate the LU's pose.

Workpiece tracking

In the previous iteration, the workpiece reconstruction was provided by the point cloud mentioned above, which was obtained through an RGB-D sensor placed in the robotic cell. However, the result did not provide a clear and precise representation of the workpiece, due to the mentioned noisy nature of the point cloud.

Given these serious limitations, it was decided to rely on the ArUco marker detection, which was already running on the device without issues, to track the workpiece. The Mixed Reality ToolKit's (MRTK) Spectator View ArUco detection plug-in was then modified in order to support at the same time the stationary and moving detection strategies [BCS+20] so that the former could be still used to detect the first marker used to align the AR content with the world, and the latter exploited to detect a moving item (the workpiece). Hence, a further ArUco marker was printed and attached to the metal plate representing the workpiece, whereas a CAD model of the same element was displayed to the RU, effectively acting as a real-time DT of the workpiece (Figure 5-6).

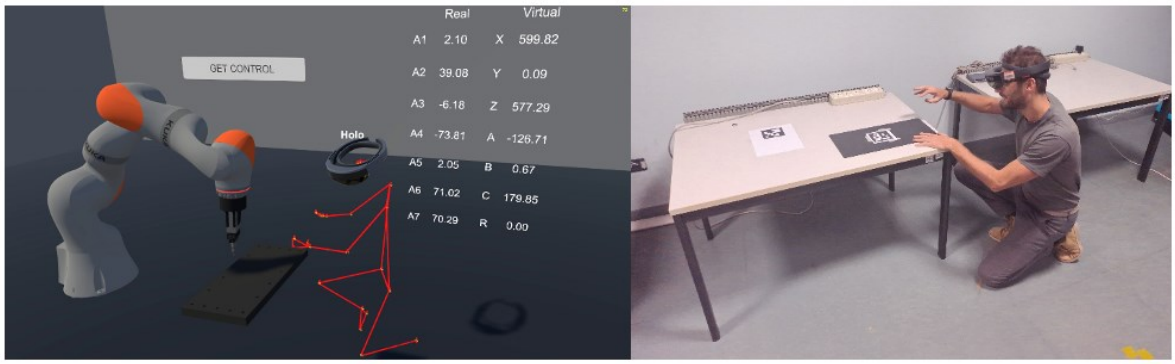


Figure 5-6: LU's DT represented by means of MediaPipe Pose landmarks during the training with the simulated cobot, as seen by the RU (left), and picture of the LU performing the task with the workpiece (tracked through an ArUco marker) in the real workspace (right).

RGB camera client

The *RBG Camera client* was implemented as OpenCV-based python script, which opens one or more Video Captures (one per camera connected to the machine) and sends the raw frames to the *Human Pose Estimation client*. The client, running on the *CAM node*, uses the ZMQ bindings for python (PyZMQ v25.0.0) to establish a connection with the *Intermediary*. The client maintains a uniform resolution among the cameras (1280x720x32) and a maximum of 30 fps. Hence, the maximum size of one second of the 2-cameras RGB stream was 1.23 Gb.

Human pose estimation client

For what concerns the LU's body pose estimation, an approach similar to the one described above for the workpiece was pursued, as different libraries and solutions were evaluated for the purpose.

MediaPipe runs extremely fast on CPU, provides both 2D and 3D poses (the 3D poses are provided in local coordinates with respect to the center of the left and right body hips). YOLOv7, in turn, is faster when ran on a GPU, but provides 2D poses only, thus requiring the implementation of a calibration for the two cameras, along with triangulation. Finally, OpenPose is characterized very poor real-time performance on machines comparable to those used for the experiments[3]. Considering the significantly lower complexity and the sufficiently high performance for the considered use case, MediaPipe was selected for the purpose. A python client was designed to receive a stream of raw images from two or more *Camera clients*, and to generate a single 3D pose as a result of the processing, which is then sent to the XR Platform through the *Intermediary*. Similarly to what was done in the previous iteration with the *Depth Camera client*, the *Human Pose Estimation client* was executed on the same machine already acting as the *Intermediary/LU* node.

Despite the non-necessity of additional points of view to obtain a 3D pose, the multi-camera configuration was adopted to mitigate the occlusion-related downsides. In particular, each of the 32 MediaPipe landmarks[4] is provided with a visibility value, expressed in a range between 0 (not visible) and 1 (perfectly visible). To merge the 3D poses coming from different cameras, a weighted average (considering the visibility value) is performed for each of the three (x, y, z) components. Theoretically, the 3D pose generated by MediaPipe does not provide information on the actual position of the human body in the 3D world, being the landmarks in local coordinates with respect to the body hip. However, in this context, the AR HMD already provides information related to the position and orientation of the LU, thanks to the inside-out 6-DOF tracking of the device itself. With the help of this information, it was possible to move and rotate the 3D pose to align it with the head of the LU, thus effectively overcoming the additional complexity related to a possible calibration. The so-calculated 3D pose is then sent through the *Intermediary* to the XR server running on the *RU node*, which uses the data to draw a representation of the human pose in terms of landmarks and connections between them (Figure 5-6.

Collaborative Robot simulation

As mentioned before, for the sake of the considered use case the real collaborate robot (Kuka Iiwa 14 R280[5]) was replaced with a simulated cobot. The model of the simulated cobot was not changed with respect to the real one, whereas the anthropomorphic hand previously used as robot tool was replaced with a riveting flange left). In order to properly simulate the behaviour of the real cobot, as well as to provide the same functionalities of the Sunrise program used in the previous iteration, a porting of the KUKA Sunrise application (the *cobot client*) was performed (from Java to C#), avoiding the implementation of everything that was not strictly useful for the selected use case.

[3] OpenPose: https://docs.google.com/spreadsheets/d/1-DynFGvoScvfWDA1P4jDInCkbD4lg0IKOYbXgEq0sK0/edit#gid=0

[4] Google MediaPipe Pose: https://google.github.io/mediapipe/solutions/pose

[5] KUKA LBR Iiwa: https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/lbr-iiwa

5.3.5 Results and discussion

In this section, the results collected from the execution of the newly defined use case on a remote training task selected for the evaluation are reported and discussed. As done in the previously reported activity, the users were requested to repeat the task 10 times to stress the platform under representative operating conditions (the duration of one execution was ~10 minutes).

The selected task was performed by two users, one playing the part of an expert (the RU, in VR), and the other one (the LU, in AR) playing the part of an operator that is skilled in the production task to be carried out, but is totally inexperienced in terms of human-robot collaboration. The production task that is considered as training subject is a fictional collaborative assembly task (the riveting of a metal plate). As explained before, for the purpose of the training, the real cobot is replaced with a simulated cobot. The LU is located inside the real space in which the real cobot will be deployed, and handles the real workpiece (a plastic prop representing the metal plate); the workpiece is provided with an ArUco marker, which allows for the creation and representation of its DT.

Since it is assumed that the program of the cobot has still to be defined, the remote training sessions begin with a first phase in which the RU gets the programming token from the LU and drafts a clip that represents what the cobot will have to do with the workpiece. In particular, from the starting position, the cobot should place the tip of the riveting flange on the first socket. Then, it should perform the riveting action on a series of sockets. Finally, it should move away from the riveting plate to let the LU reposition in order to free the path to the next series of sockets. This set of actions need be repeated for each series on the plate, which requires the operator to change his or her pose so as to allow the cobot to access it minimizing the risk of collisions.

After the definition of the program clip, the RU sends it to the LU, which starts the execution. Then, during the execution of the task, the LU tries to perform the task in collaboration with the simulated cobot in the same way as he or she would do it with a physical cobot. The RU takes advantage of the DT of the workpiece (tracked with the marker) and of the LU (in the form of a stick-based 3D pose) to evaluate the actions of the latter and possibly provide pieces of advice.

For each execution, statistics regarding bandwidth and latency between the various network nodes were gathered using a custom probe logic included in the developed software.

For what concerns the VR rendering, since the conditions were the same of the previous experiment and the software is not under the control of the developers (the AirLink functionality of the Meta Quest 2 is proprietary and cannot be customized by end-users), it was decided not to perform additional measures, assuming the previous values for bandwidth (~72 Mbps) and *motion-to-photon* latency ($L_{VR\ HMD\text{-}RU}$ =~60 ms) could still be valid.

Regarding the UNET layer, the latency between the rendering of the virtual content on the AR HMD and the same representation in VR running on the *RU node* was $L_{AR\ HMD\text{-}RU}$ = 16.94 ms (min 3.50 ms, max 79.50 ms); it showed a lower average value but a higher maximum value, probably related to a temporary congestion of the network. It should be noted that the $L_{AR\ HMD\text{-}RU}$ includes the latency of the rendering loop on the VR node (about 11.1 ms), which shall be subtracted from the overall value (corrected $L_{AR\ HMD\text{-}RU}$ = 5.84 ms). For what it concerns the bandwidth occupation $B_{AR\ HMD\text{-}RU}$ = 165.87 kbps (min 15.81 kbps, max 363.47 kbps), it was higher on average than in the previous case. This bandwidth differences are probably due to the fact that the new use case is more demanding than the previous one due to the UNET messages used to synchronize the representation of the AR user's pose (landmarks transforms) and of the workpiece, as well as due to the higher reliance on the VOIP chat for the assistance. As for the latency between the AR HMD and the simulated cobot, even trying to simulate the characteristics of the real cobot, the measured values ($L_{CR\text{-}AR\ HMD}$ = 20.41 ms, min 3.90 ms, max 107.70ms) were sensibly better than those observed in the previous evaluation, which seems to support the speculations outpointed in [Hexa-X D7.2] about some networking issues of the real cobot controller (a KUKA Sunrise Cabinet) probably due to the interlocking system among the real-time OS, the API (Sunrise OS), and the high-level OS (Windows 7) on that machine.

For what it concerns the *Camera client* (abbreviated as CAM) and the *Human Pose Estimation client* (abbreviated as HPE), the measured latencies between former and the latter ($L_{CAM-HPE}$ = 15.79 ms, min 10.00 ms, max 60.00 ms) and between the latter and the *VR client* (L_{HPE-VR} = 23.24 ms, min 2.90 ms, max 72.40 ms) summed up to a total latency between the LU's motion in the real world and his or her representation in form of MediaPipe Pose landmarks on the *VR client* of L_{CAM-VR} = 38.90 ms (min 22.90 ms, max 92.40 ms). For what it concerns the bandwidth, the obtained results between *CAM* and *HPE* were in line with the quantities hypothesized in [Hexa-X D7.2] ($B_{CAM-HPE}$ = 1.27 Gbps, min 1.13 Gbps, max 1.36 Gbps) whereas, after the processing, the stream of body poses (30 per second) was relatively small (B_{HPE-VR} = 0.18 kbps, min 0.05 kbps, max 0.20 kbps).

Finally, considering all the ZMQ-related traffic not included in the two RGB streams, the measured bandwidth passing through the *Intermediary (LU)* node was $B_{NO\ RGB}$ = 12.68 kbps (min 0.12 kbps, max 38.30 kbps).

Summarizing the results, it appears that, similarly to the previous evaluation, the actual 4G and 5G mobile networks cannot completely support the newly explored use case. The main limitation is related to the required bandwidth, which exceeds the peak bandwidth that today's real-world 5G networks can reach (1 Gbps)[6], way below the minimum for the considered setup (1.23 Gbps for the sole RGB). On the other hand, for what it concerns the latencies, the performance of 5G results again sufficient. In both cases, the employment of a 6G network would dramatically widen the potentialities due to the significantly better theoretical performance (from 10 ms and 10 Gbps[7] of 5G to 1 ms and 1 Tbps[8] of 6G).

The proposed arrangement and scenario places strict constraints on the amount of data that must be transmitted and the delays for data receipt, which may clash with the capabilities of current mobile networks. This research activity hence investigated the benefits that are anticipated to be introduced by the adoption of 6G technology to address such constraints. The results regarding latency and bandwidth, obtained from a laboratory implementation of the system, show that it is theoretically possible to deploy the investigated paradigm with 5G technology, but that it is not fully practical to do so with present 5G networks (particularly given the remarkable quantity of data to be transferred).

5.4 DT-based functional split adaptation for industrial networks

Building upon the adaptiveness proposed in the previous Section 3.3, a Digital-Twin-based controller deciding on the optimal functional split for a large industrial 5G/6G radio access network (RANs) with (possibly) heterogeneous cells is envisioned. The RAN provides connectivity to fixed stations and moving AGVs, which are deployed with the purpose of moving parts within the industrial area, monitor the activity, etc. An exemplary depiction of such a network is shown in Figure 5-7.

[6] UK Mobile Performance in Review 2H 2020: UK-wide, nation, and metro area results: https://rootmetrics.com/en-GB/content/uk-mobile-performance-in-review-2H-2020

[7] 5G vs. 4G: How does the newest network improve on the last?: https://www.digitaltrends.com/mobile/5g-vs-4g/

[8] 6G: Going Beyond 100 Gbps to 1 Tbps: https://www.keysight.com/it/en/assets/7121-1152/whitepapers/6G-Going-Beyond-100-Gbps-to-1-Tbps

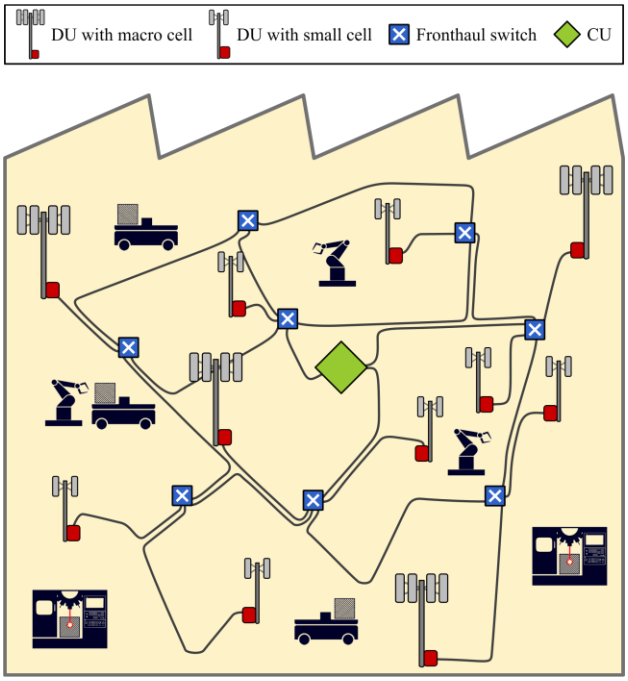


Figure 5-7: Depiction of an industrial network where fixed stations and AGVs are served by a 5G/6G heterogeneous RAN.

Owing to the necessity of acquiring dedicated licenses or to limitations of the radio equipment, it is assumed that radio spectrum resources may be scarce, which motivates a high frequency reuse. In such a scenario, inter-cell interference may be problematic at the cell edges, leading to quality of service (QoS) degradation or even connectivity losses.

Inter-cell interference can be countered with interference coordination and cancellation techniques. Nevertheless, interference mitigation requires sub-millisecond coordination between base stations, which is only achievable when coordinating functions are centralized into the same computational equipment. However, with state-of-the-art technology, full centralization is usually not possible. This is due to the capacity limitations of the midhaul network connecting centralized units (CUs) and distributed units (DU), and/or to the computational capabilities of the centralized unit.

In order to overcome these limitations, the industrial operator can opt for a dynamic functional split adaptation, in which the functions that are centralized change during runtime, according to the instantaneous requirements of the network. This is, in principle, a beneficial approach, but implementing it efficiently may be rather challenging.

To provide an efficient implementation of the dynamic functional split adaptation, development of a digital twin (DT) of the RAN is proposed, focusing on the activity of fixed stations, AGVs, and the current state of the functional split for each base station. The real system provides up-to-date information about the instantaneous state of the network, whereas the DT simulates the evolution of the activity and position of the AGVs and fixed station and takes the appropriate decisions regarding required changes in the functional split. In addition, the DT can be exploited for accurate monitoring of the state of the RAN. In Figure 5-8 a summary of the interactions between the DT and the real system is shown.

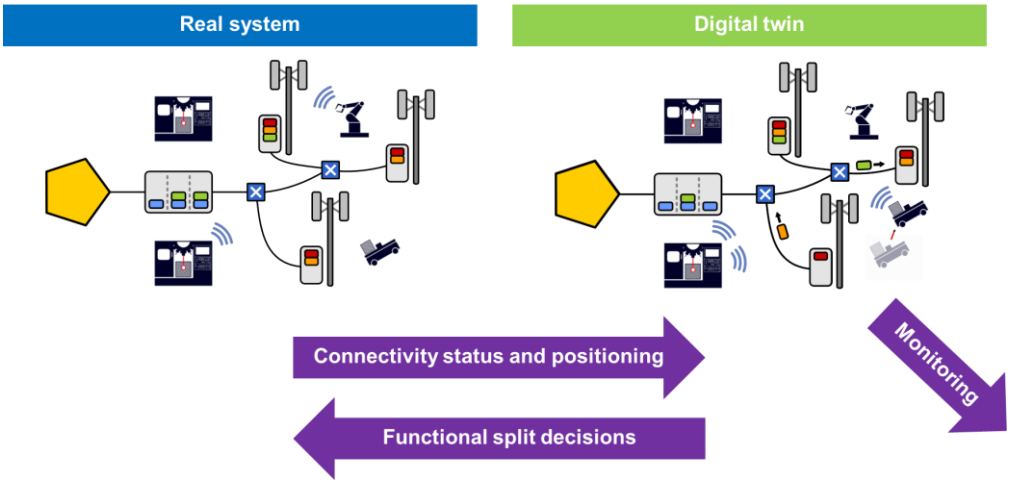


Figure 5-8: Interactions between the real system and the DT.

Consequently, the DT requires information about the activity and position of the UEs, which can be accomplished by a variety of methods, such as sheer 5G localization, ultra-wideband (UWB) positioning, RSSI fingerprinting, etc. Besides, dedicated traffic models are used to track the connectivity status of AGVs and fixed stations. Using either knowledge-based or machine-learning-based approaches, the DT transforms this information into a prediction of the future position and activity of the AGVs and fixed stations.

In addition, the DT offers a highly advanced RAN monitoring, which can provide up to date (and even future) information regarding the instantaneous functional split status, the QoS and connectivity situation of all AGVs and fixed stations, an assessment of the interference situation, and the existence of present or future connectivity requirements and other challenges that the network is facing.

Using state-of-the-art optimization approaches for functional split adaptation, the DT also decides when the functional split should be adapted based on updated simulation. In general, there may be two reasons for adaptation. On the one hand, it is mandatory to change the functional split whenever the known constraints of midhaul network capacity or computational capabilities are in risk of being violated. Otherwise, the RAN may fail to provide the required QoS to the AGVs or fixed stations, which may lead to costly service disruptions. On the other hand, even when constraints are being met, the DT may notice the existence of a better state that also meet the constraints. As a result, changing the functional split to reach this state lower the operating cost without compromising connectivity or QoS.

To assist in the decision-making part of the DT, multiple cost and risk factors must be included. Firstly, the cost of having a network that can perform this adaptation needs to be modelled. Such a network requires advanced hardware and software platforms that enable live migration of RAN functions, whose cost is understandably higher than simpler networks. Secondly, the cost or risk of operating the network and those of a possibly better state needs to be modelled, so that one can monitor the current operating cost and predict the existence of better functional splits. Thirdly, an estimation of the cost and risk of performing a change in the functional split is required, since this may potentially affect the QoS of connected devices or imply a large resource consumption. Finally, it is also necessary to model the cost or risk of erring the positioning/activity prediction, adapting too late, failing to adapt, or otherwise producing an undesired effect resulting from an adaptation, since for unclear scenarios it might be better to opt for conservative decisions. Within the project lifetime, this concept for the utilization of digital twins and initial models for the aspects discussed previously have been developed based on intermediate results reported in [Hexa-X D7.2]. A full realization of this interaction between a DT and the network for the adaptation of functional splits is beyond the scope of this project; however, the approach motivates the utilization of different Hexa-X enablers for the realization of flexible and dependable networks.

5.5 Digital Twins for Emergent Intelligence

In [Hexa-X D7.2], Emergent Intelligence (EI) has been identified as a competitive solution to deliver decentralized intelligence. It exhibits advantages in aspects of computational complexity, latency, robustness, security, privacy, and scalability, which makes it a good match for 6G use cases. However, it has also been pointed out in the same report, that EI is exposed to a special security risk of data injection attacks. In this section, we demonstrate how the DT technology can be used to 1) enhance the performance of EI in mobile communication scenario and 2) counter the threat of data-injection attacks.

5.5.1 Digital Twins enhance the performance Emergent Intelligence

For demonstration, a use case of multi-UAV-based chemical spill handling is considered in [YHK+22], where multiple UAVs are deployed at different locations over a chemical manufacturing site. Supposed to localize the spill spot and arrive there to take all necessary repairing measures, each UAV (also referred to as agent hereafter) is equipped with not only a positioning module to obtain its own position, but also sensors to measure the chemical densities in the air, from which it can estimate its distance to the spill spot according to the diffusion model. However, due to the isotropic diffusion of chemicals in the air, no single agent can estimate the relative direction it is distanced from the spill spot. Thus, a joint localization and routing must be carried out, which exploits the measured data of multiple agents. As a classical instance of EI, the particle swarm optimization (PSO) algorithm is invoked, where each agent collects in every iteration the latest position and distance to target of every other agent, and therewith update its speed and direction of motion. After enough steps, all agents eventually converge at the target position.

This EI approach, as discussed in [Hexa-X D7.2], decentralizes the decision making to all agents, which makes it robust against local failures and immune to model tampering attacks. However, it shall be noted that a full information exchange among N agents, when realized in a conventional cellular approach, takes at least $2N(N-1)$ successful data transmissions. Although 6G is supposed to provide sufficient radio resource for massive devices to access the network simultaneously, this quadric increase of communication load regarding system dimension can raise significant challenges to radio resource allocation and interference management of the network when N is large.

This issue, as shown in [YHK+22], can be addressed by deploying DT for all involved agents at the same MEC server. By migrating the information exchange and decision making from the physical UAVs to their DTs, the communication load can be reduced to a linear level w.r.t. the agent number N, so that the convergence performance is improved even under stringent radio resource constraints, as shown in Figure 5-9. Meanwhile, the privacy advantage of EI can be still guaranteed by virtually isolating the DTs.

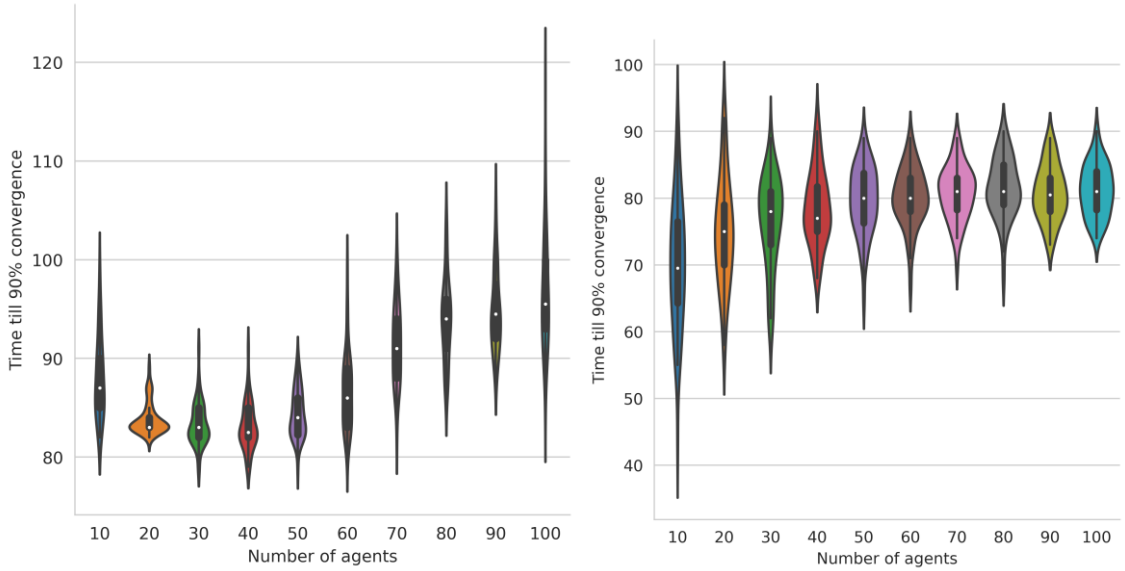


Figure 5-9: Convergence of the PSO algorithm in [YHK+22] with (left) and without (right) DT.

Beyond the enhancement to protocol efficiency of information exchange among agents, DT can potentially bring further benefits to EI systems in the 6G context. For example, with a DT of the radio environment at the deployment scenario, significant gains in radio efficiency can be expected by invoking DT-based radio resource management and agent trajectory planning, as discussed in Sections 3.2 and 4.1.1.

5.5.2 Security concerns in EI

Instead of decision making based on any shared model, EI relies on the information sharing among massive independent agents in an ad-hoc fashion, which makes it immune to some classical attacks such like model tampering. However, this mechanism design is also magnifying the dependency of different agents on each other, which exposes EI systems to a special type of security risk, i.e., data injection attacks, as pointed out in Sec. 2.4 of [Hexa-X D7.2] and Sec. 7.4.2 of [Hexa-X D1.4].

To assess the threat of data injection attacks to EI, numerical simulations are carried out in [HKZ+22] where a small group of agents (penetration rate between 3% to 10%) in the use case aforementioned in Sec. 5.6.1 are manipulated to maliciously and randomly (by chance of 10%) send incorrect information to other agents. The result, as depicted in Figure 5-10 shows that even at a low attacking rate, it takes only a few malicious agents with appropriate attacking strategy to significantly compromise the overall performance of an EI system.

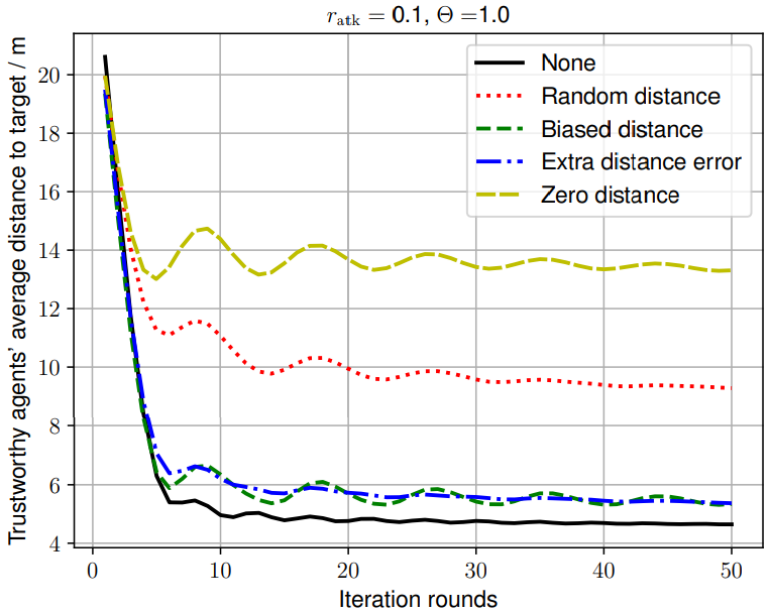


Figure 5-10: The impact of data-injection attacks with different strategy on the PSO performance

5.5.3 Trust-aware particle swarm optimization

Intrinsically, data-injection attacks are exploiting and abusing the trust among agents to each other. To counter such attacks, agents shall be aware of the trustworthiness of each other, and therewith selectively exploit the information they receive from others, i.e., discriminate some agents against others regarding their trustworthiness.

As a demonstration, a trust-aware PSO algorithm is developed in [HKZ+22], which relies on a generic anomaly detector to distinguish possible maliciously injected fake data from normal data, and therewith adjusts the trust score of the data sources (i.e., agents). Furthermore, instead of simply rejecting data shared by suspicious agents, it is capable of exploiting such data with a certain trust-score-related penalty. Simulation results show that the proposed solution can efficiently improve the robustness of the EI system against data injection attacks, despite of the non-ideality of the anomaly detector, as illustrated in Figure 5-11.

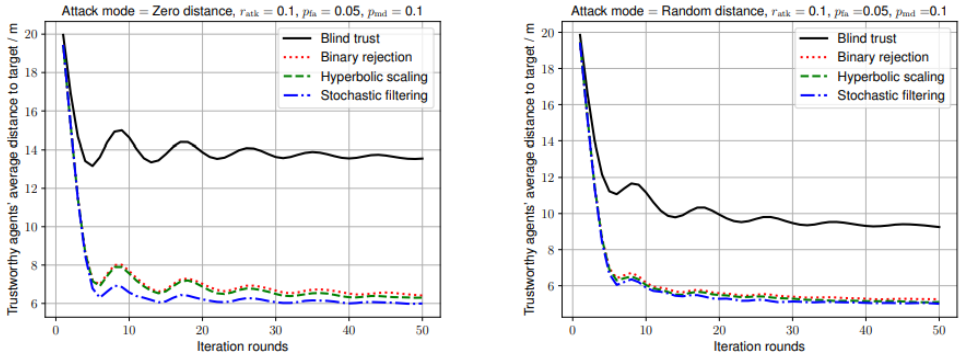


Figure 5-11: Performance of PSO with different trust-awareness solutions under data-injection attacks.

5.5.4 DT-based trust awareness

It shall be noted that to realize an efficient trust awareness in 6G-based EI systems, as many as possible aspects of the agent behaviour shall be observed and analysed. Such observation shall not be limited within the domain of individual application or use case, but all applications/use cases that the agent is involved in. In other words, trustworthiness as a special type of context information of an agent (which can be a UE, a network node, or a service provider, etc.), shall be measured by diverse virtual sensors, and the various measurement shall be merged through a data fusion process. This demand is calling for a framework that creates for every agent a tightly coupled data entity, which shall be accessible by all other authorized agents, and capable of being online updated in real time. Fortunately, the DT technology is offering all these essential features, which encourages to implement the envisioned trust awareness system based on the DT framework.

6 Demonstrator: extreme performance in handling unexpected situations in industrial contexts

During the study and implementation of the Hexa-X project, Demo#4 "Extreme performance in handling unexpected situations in industrial contexts" was created, among other demonstrations. This demo is closely related to the "interacting and cooperative mobile robots" and the "digital twins for manufacturing" Hexa-X use cases, which are part of the "from robots to cobots" and the "massive twinning" use case families respectively [Hexa-X D7.2]. WP7 (all tasks) and WP6 contribute to this demo; hence "Network evolution and expansion towards 6G" and "Connecting intelligence towards 6G" Hexa-X objectives are explored and targeted. The demo's aim is to address advanced orchestration, monitor, diagnostics, management, redistribution of functionality in an automated way as part of the 6G continuum, while the ultimate goal is the connection of the Human, Physical and Digital worlds. By recognizing faults and taking appropriate action to address them, Demo#4 offers a practical implementation for ensuring dependability in an I4.0 scenario.

Within this demo, Human Machine Interaction (HMI), and user's interaction with digital twins (DTs) and Virtual Reality (VR) technology with immersive realistic 3D graphics is investigated, driven by the enormous interest on infrastructure automation, cobots and VR on a massive scale in the industrial sector. Additionally, the emulation of impairments through advanced predictive orchestration and diagnostics as part of the 6G continuum, and the management of failures through redistributing functionality and roles is studied since the failure of a robotic service or device is crucial and rather costly for the production and transportation of products.

For this demo, a set of robots is utilized, working in an industrial environment, moving products, and performing quality checks. These robots are cobots, in the sense that they interact and collaborate in order to accomplish a task by sensing, perceiving, planning and controlling independently, towards a common goal and without explicit instructions from a human. The robots are continuously providing feedback of their status (exact location in space, RAM, CPU, battery level, their possible malfunctioning components etc.), and the services assigned and running on them.

Some of the challenges are: (i) the massive twinning application, (ii) the cooperation of real robots on industrial tasks, (iii) the handling of unexpected situations through hybrid cloud infrastructure, intelligent orchestration, diagnostic and monitoring mechanisms, and (iv) the control, supervision and monitoring using digital twin and/or VR goggles enabling the "human in the loop" for interactions, repairs, or even manual teleoperation.

6.1 Scenarios

In this demo, a series of cooperative robots (three LoCoBots in particular) performing a full cycle of an integrated operation in an industrial environment is presented. They currently utilise Wi-Fi/4G/5G networks to communicate with each other and with the infrastructure. One of the robots performs product quality check and passes the product with the help of the robotic arm to one of the other two robots, which transports the product to repair or shipping locations depending on the outcome.

In comparison to the ones in the previous deliverable [Hexa-X D7.2], the scenarios have been enhanced and extended. Therefore, three scenarios are provided to showcase all the anticipated functionalities and features, named: 1) Unified orchestration across the Cloud – Extreme edge continuum, 2) Functionality Allocation, and 3) Predictive orchestration and maintenance. Each scenario leverages the work of the previous ones, while each one introduces a different enabler. A description of each scenario follows. However, refer to [Hexa-X D6.3] for more details related to management and orchestration.

Scenario 1: Unified orchestration across the Cloud – Extreme edge continuum

The first scenario exploits the benefits of B5G/6G network technology on performance and efficiency of production lines in an industrial environment with the utilisation of a unified orchestration process

across the Cloud - Extreme edge continuum. The unified orchestration process developed for this scenario can provide automation, reconfigurability and reduces human interventions. Specifically, this enabler is showcased with the development of the example digital twin, VR application, HMI, and teleoperation allowing human-in-the-loop activities.

The digital twin developed for this demo represents in real time and in detail the three robots moving in the room with 3D graphics. Each part of the robot (e.g., robotic arm, camera, lidar) is pictured in the digital twin in detail. Each robot has a popup window on top, containing status information (CPU, RAM, battery level, etc.) and the services that are assigned and run on the robot with a health check of each service. Also, the feature of a real time streaming is enabled, of onboard cameras on each robot as well as static cameras placed inside the room. The teleoperation can be performed at any time by a remote user for remote maintenance and configuration by utilising informative user interfaces (UIs). The VR application of the digital twin developed can provide an advanced experience to the user when for instance there is a need of remote repair.

Scenario 2: Functionality Allocation

The second scenario aims to study and demonstrate the fast and close to optimum management and redistribution of functionality in an automated way. This functionality allocation component is of great importance in the case of an unexpected situation, a problem in the flow of industrial production (e.g., a part of a robot goes out of order). Of the same importance are the AI/ML based anomaly detection and performance degradation analysis which are utilised in closed-loop control mechanisms (monitoring and performance diagnosis) to increase network efficiency by alerting/triggering corrective mechanisms if needed. Hence, various metrics and KPIs are monitored by closed loop control mechanism and in the case of the detected/predicted accomplishment of target KPI, corrective actions occur immediately. The AI/ML actuators were analysed and studied for the greater efficiency of the functions and services, and also to obtain the necessary KPIs/KVIs. More details of the software components mentioned can be found in the next section.

Scenario 3: Predictive orchestration and maintenance

The third scenario addresses the challenge of anticipating situations and predicting the behaviour of services and the various components of the production cycle, with the use of AI/ML enablers. With the use of monitoring data from selected services or components, predictive models are trained to be able to identify accurately upcoming critical events. This functionality increases network efficiency, reduces operational and maintenance costs as well as minimises the impact of unexpected situations. For a more detailed view on this scenario, refer to [Hexa-X D6.3]

6.2 Demonstration specific architecture

For implementing and materialising the described scenarios, various software components have been developed and efficiently linked together. Figure 61 presents the architecture of the Demo#4 implementation. The upper layer of this schema consists of the Intelligent orchestration components, meaning Service Registry, Predictive orchestration, Functionality allocation, and Diagnostics along with the Orchestrator (OSM). These components are closely connected with the monitoring framework named Monitoring as a Service (MaaS) and decide if there is a need of action and accordingly, they trigger OSM for executing the potential decision. The middle layer consists of the Kubernetes cloud, master and edge as well as the MaaS. All these components are closely connected with the three robots for exchanging services and collecting data, respectively. Digital twin and VR applications are mostly connected with the three robots for exchanging the status and location information mentioned earlier. This figure also shows the network connections utilised for each component. A more detailed description of these components follows.

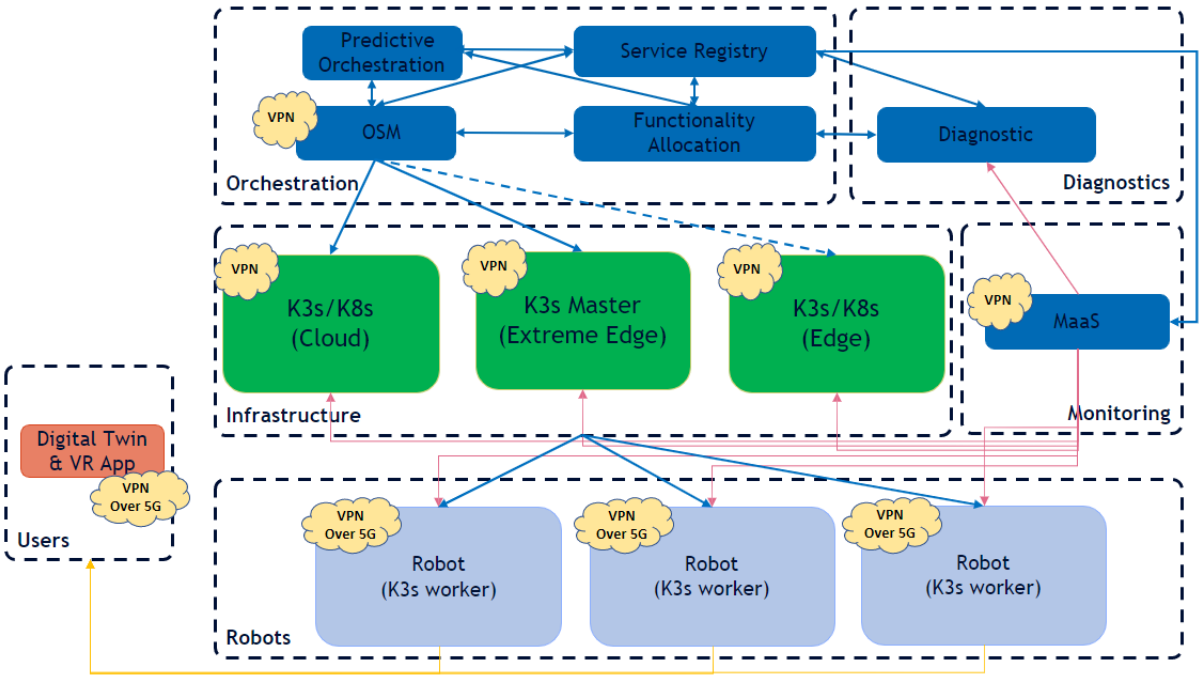


Figure 61: Demo#4 software architecture. The upper layer consists of the intelligent orchestration components, the middle layer consists of the Kubernetes infrastructure together with the monitoring component, and the lower layer consists of the three robots. The digital twin and the VR application are placed on the left side, being closer to the robots with whom they interact the most.

6.2.1 System software components

6.2.1.1 Monitoring (6G network metrics supporting robotics applications)

The monitoring platform utilised is the Monitoring as a Service (MaaS) platform. It consists of three main parts, the MaaS server, MaaS client(s) for implementing front-end and Bridge which translates the MaaS server probes request to the cloud platform. The probes are placed in a probe catalogue and provide metadata for allowing MaaS to choose the deployment decision. The probes push data in a time series format, to a database and stores them there. The Bridge implements the creation, update, delete, status information of the available probes in the cloud. For further details refer to [Hexa-X D6.3].

6.2.1.2 Performance diagnosis (Error identification)

Performance diagnosis tool, as was described in [Hexa-X D7.2], collects information during the execution of a certain application or service. The data collected is analysed by this tool and insights of the performance of the various elements that comprise the experiment are produced. Performance diagnosis tool operates as part of a set of management operations, retrieved from agents deployed on the individual parts of the infrastructure such as robots, and other network nodes. It is responsible for detecting anomalies in the behaviour of the various elements and can identify the root cause of possible problems during the experiment in most cases.

In the case of a performance degradation in a deployed service, the Diagnosis component's operation is synchronised with communicated interfaces for providing real-time insights to Functionality allocation component. This is succeeded without unnecessary traffic load or functional overlaps, because the processes of data collection, pre-processing and analysis, take place on the spot.

6.2.1.3 Functionality allocation

[Hexa-X D7.2] introduced a mechanism for optimal resource allocation and redistribution of functionalities in industrial environments named Functionality Allocation. The problem statement and

the problem formulation were presented in that document, with the proposed solution. Subsequently, this subsection describes the updated formulation utilised, the solution approach, the performance tests and results obtained and the integration of this mechanism to the overall architecture of Demo#4.

Updated problem formulation

The reallocation of functionality and redistribution of tasks of a robotic industrial system in an orchestrated manner, can help to overcome a possible issue like the increased latency or the unavailability of used robotic units in an industrial environment. A Functionality Allocation algorithm was developed for this reason, which is an optimisation algorithm that redistributes the functionalities to the various nodes/robots with minimum cost as well as with ensured efficient operation of the robotic system. The input to this algorithm is the set of Functional Entities (FEs), such as services, tasks, as well as the set of Hosting Entities (HEs), such as robotic units, edge/core nodes of the system. Each FE has some computational requirements (e.g., CPU, RAM) and some functional requirements related mostly to robot specific services/tasks (e.g., need of camera, arm, wheels, specific location in the space to be executed). Each HE has some capabilities (e.g., available CPU, RAM, battery level, camera, arm, wheels if applicable) and the location in space applicable mostly to robotic units. The functional graph (directed acyclic graph) and the system layout graph are additional input to the algorithm. In the first graph each node represents a FE, and each edge connects two interacting FEs, showing the direction of data transferred and the second graph shows the communication channels among HEs with the bandwidth capacity of each link.

The optimal allocation/placement of the FEs to the available HEs is succeeded when the following objective function (mixed integer programming formulation) is minimised respecting some constraints:

$$\min_{y,x,z}\Bigg(a_1 \sum_{j=1}^{m}\left(y_j b_j\right) + a_2 \sum_{j=1}^{m}\left(\left(W_{max}^{j} - W_{idle}^{j}\right)U_{CPU}^{j}\left(x_{i,j}\right) + W_{idle}^{j}\right) + a_3 \sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_{i,j} D(l_i, l_j)\right) + a_4 \sum_{i=1}^{n}\sum_{i'=1}^{n}\sum_{j=1}^{m}\sum_{j'=1}^{m}\left(z_{i,i',j,j'}\frac{k_{i,i'}}{cap_{j,j'}}\right)\Bigg)$$

The notations utilised can be found in Table 8. The objective function consists of four weighted terms. The first term is associated with the battery level of the utilised HEs (b_j is zero when it is fully charged, or it is not battery-powered, and close to 1 when the battery level is very low). The second term is associated with the power consumption of the system. In this term, the CPU utilisation rate $U_{CPU}^{j}\left(x_{i,j}\right) = \frac{cpuHE_{total}^{j} - cpuHE_{free}^{j} + \sum_{i=1}^{n}\left(cpuFE_i x_{i,j}\right)}{cpuHE_{total}^{j}}$ is assumed that counts for most to power consumption, compared to memory, bandwidth and disk storage, as it is proposed in [AAR22]. The third term is related to the travel distance, meaning the distance between the HE's (robot) location and the FE's (task) location where it should be executed if it is applicable. Finally, the fourth term is associated with the transmission delay of the data transferred between interacting FEs. Each term is weighted (a_1, a_2, a_3, a_4) depending on the use case requirements.

The constraints utilised are:

- $\sum_{j=1}^{m} x_{i,j} = 1,\ \forall\, i = \{1, \dots, n\}$, all FEs are allocated,
- $\sum_{i=1}^{n} x_{i,j} cpuFE_i \le cpuHE_{free}^{j}\ \forall\, j \in \{1,\ \dots,\ m\}$ and $\sum_{i=1}^{n} x_{i,j} memFE_i \le memHE_j \quad \forall\, j = \{1, \dots, m\}$, the maximum available resources of each HE is respected.
- $\sum_{i=1}^{n} x_{i,j}/n \le y_j,\ \forall\ j = \{1, \dots, m\}$, all HEs that are utilized ($y_j = 1$) have at least one FE assigned on them.
- $x_{i,j} \le w_{i,j},\ \forall\, i = \{1, \dots, n\}, j = \{1, \dots, m\}$, the feasibility of assigning a FE to a HE in terms of functionality types that can be supported by the HE is respected.

- $z_{i,i',j,j'} \geq x_{i,j} + x_{i',j'} - 1$ and $z_{i,i',j,j'} = z_{i,i',j',j}$, $\forall\, i, i' = \{1, \dots, n\}, j, j' = \{1, \dots, m\}$, where $i \neq i'$, $j \neq j'$ and $k_{i,i'} \neq 0$, the communicating HEs should have communicating FEs assigned on them.

Table 8: Functionality allocation notations.

Notation	Definition	Notation	Definition
n, m	Total number of FEs and HEs respectively	W^j_{max}, W^j_{idle}	Power consumption of HE H_j when fully loaded and when idle
$i,\ i'$	Indexes of FEs	l_j	Location of HE (robot) H_j
$j,\ j'$	Indexes of HEs	$D(l_i, l_j)$	Euclidean distance of locations l_i, l_j
$F = \{F_1, \dots, F_i, \dots, F_n\}$	Set of FEs	$U^j_{CPU}(x_{i,j})$	CPU utilisation rate of HE H_j
$H = \{H_1, \dots, H_j, \dots, H_m\}$	Set of HEs.	$w_{i,j}$	Constant {0,1} showing if FE F_i can be assigned to HE H_j in terms of functionality types that can be supported by the HE
$cpuFE_i, memFE_i$	CPU and memory requirements of FE F_i	a_1, a_2, a_3, a_4	Weights of the objective function's terms
$cpuHE^j_{total}$, $cpuHE^j_{free}$, $memHE_j$	Total CPU, available CPU and available memory of HE H_j	y_j	Decision variable that takes 1(0) depending on whether HE H_j is (is not) utilized
b_j	Cost related to the battery level of HE H_j	$x_{i,j}$	Decision variable that takes 1(0) depending on whether FE F_i is (is not) assigned to HE H_j
$k_{i,i'}$	Data transferred between FEs F_i and $F_{i'}$	$z_{i,i',j,j'}$	Decision variable that takes 1(0) depending on whether HE H_j and $H_{j'}$ are (are not) communicating due to communicating FE F_i, $F_{i'}$ assigned on them.
$cap_{j,j'}$	Capacity of the links among HEs H_j and $H_{j'}$		

Solution approach

The Functionality allocation problem was initially solved with the use of a Mixed Integer Programming (MIP) python solver named GNU Linear Programming Kit (GLPK) provided by the open-source PuLP [MSD11]. Then it was also solved with the development of a meta-heuristic algorithm based on the Genetic algorithm paradigm [MS96].

In Functionality Allocation meta-heuristic algorithm, some appropriate optimization steps and additional features were utilised. Firstly, each "chromosome" in this algorithm is a sequence of HEs, where each HE represents the "proposed" placement for each FE and the length of each "chromosome" equals to the number of FE. A penalty function was used to handle the constraints arising from the computational requirements of FE and the functionality types that each HE can support. This penalty is added to the fitness function each time a constraint is violated; hence it is defined carefully to ensure that the final solution is feasible. Last but not least, an appropriate early stopping criterion was utilised for termination and convergence of the algorithm. A schematical example of Functionality Allocation deployment is presented in Figure 6-2. In this case robot HE3 becomes unavailable and the algorithm

reallocates the FEs of this robot to the remaining HEs based on their computing (CPU, RAM, etc.) and functional requirements (camera, arm, etc.).

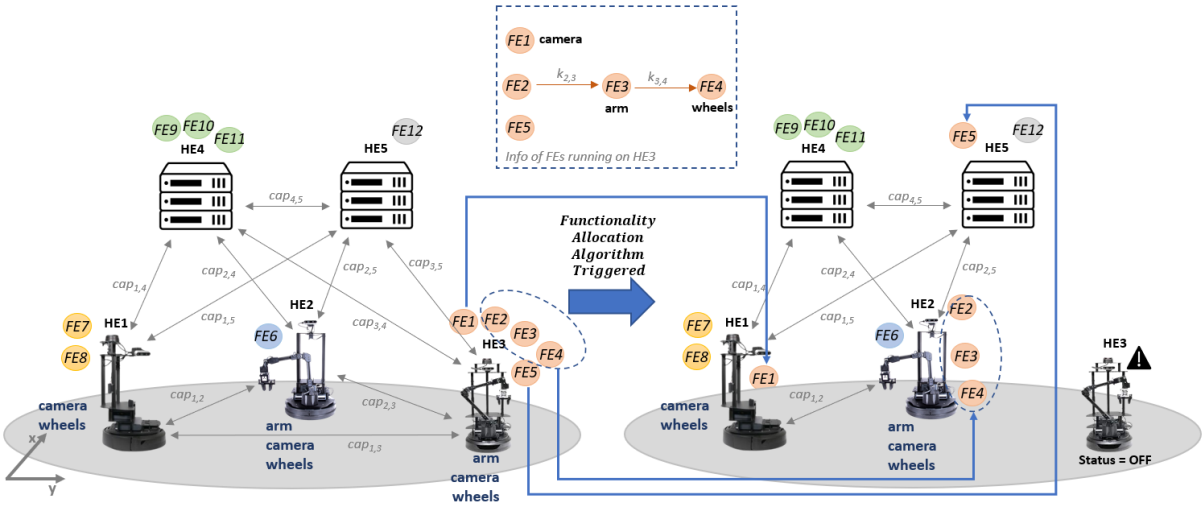


Figure 6-2: Schematical Functionality Allocation algorithm utilisation.

Performance tests and theoretical results

The output of the MIP python solver PuLP was used for testing the performance of the meta-heuristic algorithm developed, based on the genetic algorithm. Figure 6-3 shows the performance testing of the proposed algorithm based on genetic algorithm compared to PuLP GLPK MIP solver. In this case a fixed number of FEs is assumed ($n = 9$) with {500,750,1500} MIPS levels of required CPU, {256,512,2048} MB levels of available memory, 0-10 Mb range of data transferred, and various functional requirements, e.g., arm, camera, wheels. The number of HEs increases and have {2000,2600,3000} MIPS levels of available and total CPU, {2048,4096,8192} MB levels of available memory, {260,360,460} W levels of power consumption when fully loaded, {70,100,170} W levels of power consumption when idle, various additional capabilities {arm, wheels, camera, none} and 0-20 Mbps range of capacity links. Additionally, the "population" (possible solutions) of each generation of the genetic algorithm was 100, with 0.8 crossover rate and 0.15 mutation rate.

Although MIP solvers are known to provide in most cases the optimal solution, they are computationally demanding in large scale, hence an execution time limit of 100 sec was imposed to the PuLP GLPK solver. When solver exceeds the time limit, the best solution obtained till then is provided (may not be the optimum). The graphs in Figure 6-3 show that the developed algorithm based on genetic algorithm obtains scores close to the ones obtained by PuLP solver in significantly less time. In particular, when the number of HEs exceeds 30, the developed metaheuristic algorithm is preferable since it obtains approximately the same results (scores) and sometimes better than the ones obtained by the solver till the time limit of 100sec in much better (lower) execution time.

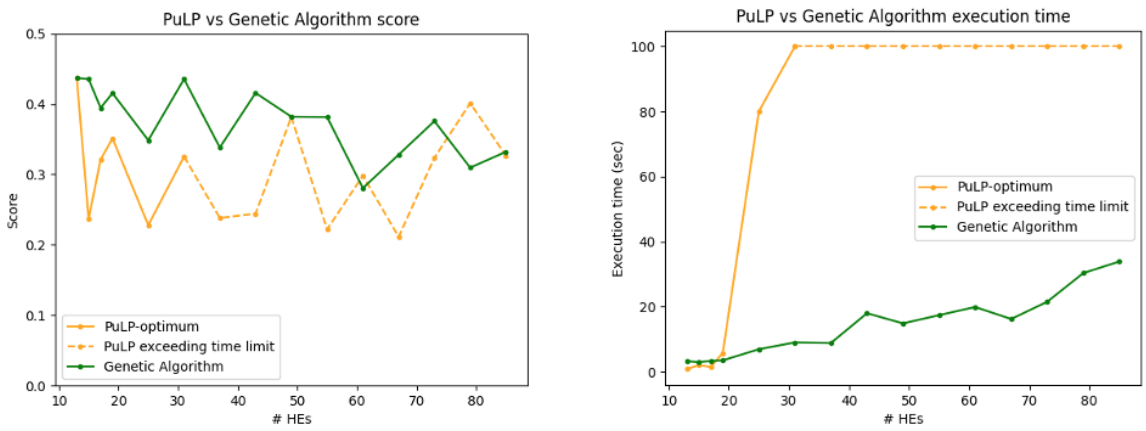


Figure 6-3: Comparison of the scores (left) and the execution time (right) achieved with increasing number of HEs for the proposed model based on Genetic Algorithm and PuLP GLPK MIP solver.

Integration to the Demo#4 architecture

Functionality allocation component has three main features. It can: (i) efficiently place/allocate of some new FEs/services to the network's available nodes, (ii) change the placement of a service that performs badly (e.g., a service which exceeds the execution time limit), and (iii) reallocate the services running on a HE/node which becomes unavailable for a reason (Figure 6-2).

Functionality allocation algorithm is containerized using Docker[9] and it is placed in the central infrastructure along with the rest of Intelligent orchestration components. The FastAPI framework[10] was utilised to enable the communication of Functionality Allocation component with the rest of the demo's components (Figure 6-4).

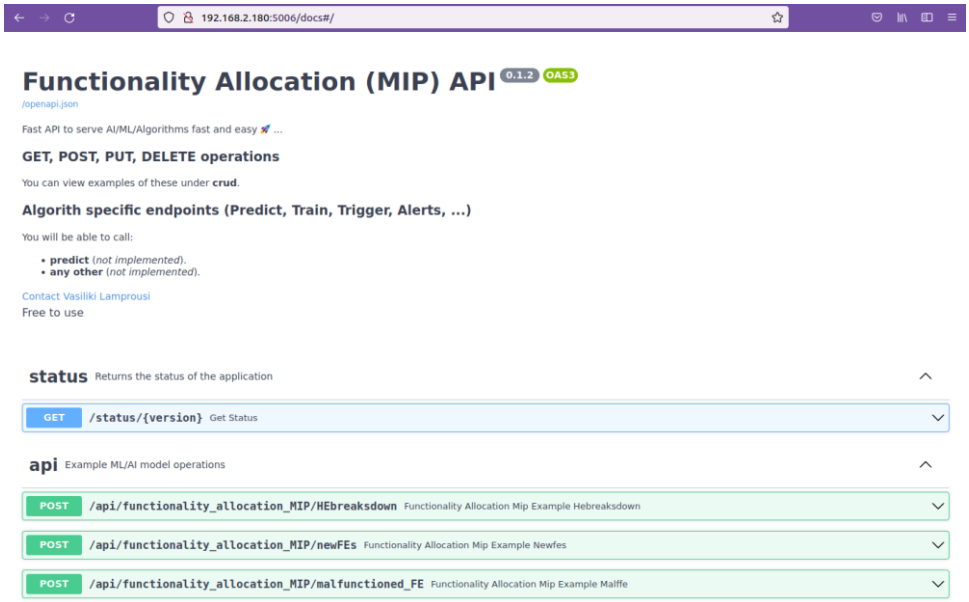


Figure 6-4: Functionality Allocation FastAPI

[9] Docker. What is a Container? Available from: https://www.docker.com/resources/what-container.

[10] https://fastapi.tiangolo.com/

6.2.2 Orchestration - recovery in robotic scenarios

As it was mentioned above, when the Intelligent orchestration components decide (by analysing input from MaaS) that there is a need of action, they trigger the orchestrator OSM for executing the potential decision taken (e.g., service reallocation). This decision and service placement is immediately perceived by the user. Similar to the renewal of the position of the robots in space, the end user is continuously informed of the services running on the robots. In the Unity digital twin application there is a "services" tab menu with all the functional services running on each robot, divided into three subcategories (machine learning, software, hardware). The live feeding of the information of the services as well as their transfer from one robot to the other gives the VR user direct supervision of the process status.

A specialized software has been developed, to run the robotic scenario in an industrial context. It handles the operation by utilizing the services provided by the robots, i.e., controls the flow of movement, object detection, environment manipulation and interaction based on collected data from sensors. Therefore, is named controller. In literature such software is also known as a task scheduler. Keeping the conformity of the greater architecture, it constitutes a deploy-able service, isolated from undesired interactions and events.

The controller periodically scans the network for changes, to identify if there are available robots to fulfil the scenario, or in the event of an error. It can detect if all the services needed are present or not, regardless of the number of robots involved. The intelligence encapsulated in it, makes it able to run using a single robot or more, depending each time, on how the services have been allocated. Its dynamic nature allows it to alter the number of robots commanded at run time, minimizing the idle time that is introduced when a fault occurs and maximizing productivity by taking advantage of newly available robotic systems.

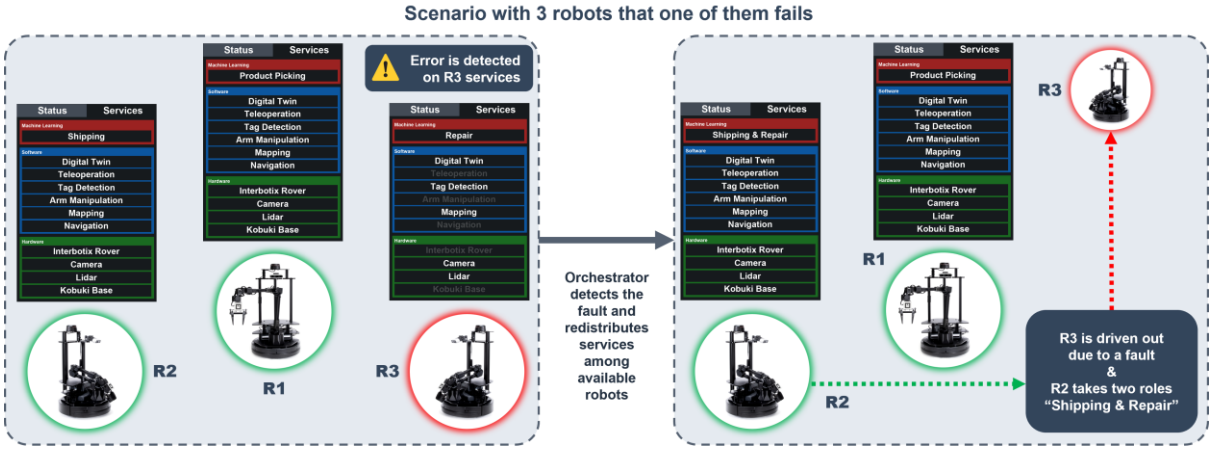


Figure 6-5: Schematical representation of the scenario where one out of three robots fails.

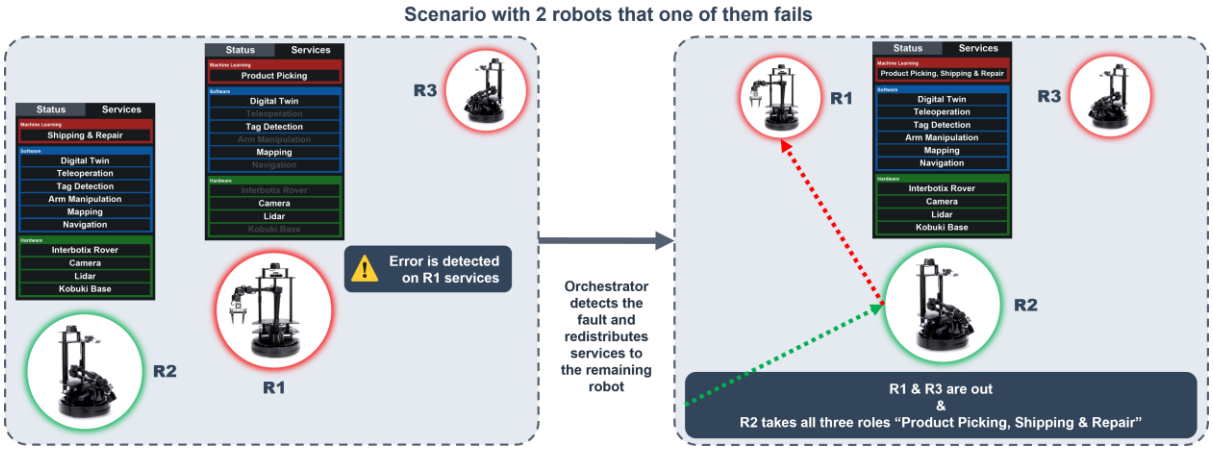


Figure 6-6: Schematical representation of the scenario where two out of three robots fail.

Figure 6-5 and Figure 6-6 schematically present the case/scenario where robots gradually become unavailable, and the rest take upon their services. The controller can mandate the remaining fleet to keep the scenario running. This process can also work in reverse, the controller can detect that services have been reallocated to newly introduced robots and provide commands.

As a result, the possible scaling of such a solution becomes transparent. Its modularity and event handling, realize a robust paradigm that can promote further automation and reliability.

6.2.3 Digital twin, VR application and teleoperation

6.2.3.1 Digital twin

A digital twin is a virtually constructed model designed in such a way that can accurately reflect the physical properties and functions of the real object. In digital twin applications, important roles have the kinematic models, the transmission of information between each other and the end user, but also the environment that contains these models, such as the real space that may not be static or even the obstacles that may suddenly appear.

According to the scenario we are studying, the environment is a small industrial sector. The space consists of a room on specific dimensions (3x2 m) and at the same time with certain fixed prescribed properties (landmarks, apriltags). Within the area there are three points of interest (production, shipping, repair point) where the following operations are carried out, the receipt of items from the production location, the identification of defective or non-defective products (red or blue respectively), the exchange of products between collaborative robots and the transportation of the items to the respective locations (shipping if it is a good product or repair if it is defective).

The objects we are studying in this project are robots which are equipped with servo motors, robotic arms, cameras, batteries, lidars and sensors which are vital elements for their operation. Robots produce and receive data about physical processes they are conducting, such as moving in space to specific points of interest, performing object recognition, avoiding obstacles, grab and carry products (see Figure 6-7); they transmit information to each other as well as inform the user. Their goal is their cooperation for the completion of automatic processes within the framework of the factory installation (Co-bots). Data is transmitted from one robot to another, processed by a controller algorithm, and all the necessary information is displayed to the end user. Finally, all the above information is applied back to the corresponding digital twins of the robots.

Once the application is updated with the data, the virtual digital twin models can use it to perform complex processes, finding better solutions such as their location and routing path in space, avoiding obstacles, distribute tasks between them, exchanging their roles, measuring their performance, identifying a possible problem in a robot or process level, and other functions which can then be applied to the real objects. Figure 6-7 shows the digital twin interface where the two robots cooperate to exchange a product. On the right side we can see the digital robots and on the left side we can see the real robots as they are captured by the live streaming cameras (onboard cameras on each robot and cameras in space).

Figure 6-7: Digital twin interface showing the moment a product is exchanged between two robots.

Digital twins use a real-time two-way flow of information, in contrast to simulations, which do not benefit from real-time data transmission. This means that our robots feed the system with data that are displayed to the user (real time camera feed of the robots, sensors values and arm rotation, status, battery level, etc.), at the same time these values used by the system control to find solutions to various problems (exchange of roles, service allocation, etc.) and feeding back the system with information from the user (HMI), also from the controller (controller decision making) and from other co-bots that are participating in the processes. Always, the goal is to improve the processes within the actual factory space and inform the end user.

Based on the project scenario, we tried to simulate an entire factory production unit. All necessary systems are synchronized in such a way that they operate at maximum efficiency with minimum delays. Digital twins can help us determine which operations need to be performed for optimal system timing and efficiency.

6.2.3.2 VR application – Unity 3D application

Apart from the digital twin, it was also developed a VR application in Unity 3D game engine. Unity is software particularly popular in the game development industry but is also extending to other areas such as robotics and industrial application development. Thus, it was used in the composition of all 3D graphics, environment and objects (3d graphics - environment) and programming and operation of the robotic twins (3d kinematic models) and in the development of the UI. With Unity as the main tool, we can choose a development platform such as Windows or Android OS and turn it into a Virtual Reality immersive environment application.

As it was described in the previous subsection, the real space of the industrial space used for this demo, consists of fixed and prescribed boundaries where the robots can move within. The VR application was built using data from the robots, for example what they see through their cameras (depth camera) combined with the 3D point map (simplified point cloud) coming from the navigation system of the robots (lidar). Then the rest of the fixed elements were added, such as the locations of the three points of interest (production, shipping, repair point) and the images that help the robots to recognize the space (apriltags, QR codes).

The digital twins of the robots (Interbotics Locobots PX200 & PX250S) were initially constructed using 3D graphics (3D modelling) with the Autodesk 3ds Max software in such a way as not to burden the application with unnecessary information but instead to have a photorealistic result and serve their purpose as game objects properly (low-poly models & 3d texturing). Within the Unity 3D environment, the kinematic models of the robots were created, i.e., the way they can move within the space (wheels,

move base) but also all the necessary joints (pivot points, rotation axis) that make up the kinematic models of the robotic arms. Afterwards, these objects were programmed (C# scripts) to become digital twins and to virtually transfer all the necessary information, just as the real robots to the end user.

VR Users can:

- Navigate inside the digital space of the robots (virtual space camera) using the VR controls (joystick) as well as the settings offered by the controls (Oculus Quest 2 options).
- Access to real-space robot cameras. Live streaming feed in separate windows for each. The user can also check the identification of the items as defective or ready for delivery.
- Choose to connect, start, or stop the automatic process.
- Select each robot and access to its menu.
- Select remote control of each robot (transform & rotation) in future version manual remote control of each robotic arm will be added.
- Access the status menu of the robot (Ip Address, Role, Mode, Battery level, Power bank level, Total CPU, Usage CPU, Total RAM, Free RAM)
- Access to the Services menu of the robots.
- Access to the Notifications & Debugging menu where the user can intervene in any eventuality to avoid errors.
- Browser access.
- Language selection.

In the event of an error in the system, a network problem, a robot failure due to a status or a servo problem, the user is informed via notifications and messages, while the following solutions are given:

- The user can stop the automatic process as soon as the error occurs.
- The user has also the option to choose remote manual control of any robot, lead it to a point where it does not disturb the automatic process, while at the same time the system activates the corresponding mechanism for functionality allocation, redistributes the roles of the robots and continues the scenario without the problematic robot.
- The user can restart the automatic process when the problem is fixed.

Figure 6-8: The VR application interface with the user wearing the VR headset utilised.

Figure 6-8 shows the VR application developed for Demo#4. It pictures the VR interface and the user wearing the VR headsets used (Oculus Quest 2).

6.2.3.3 Teleoperation

Teleoperation service is included in the desired features of the required HMI - digital twin interface. The main reasons governing the existence of such functionality can be seen below:

- Remote control of robot, in case of necessary removal from the automated production line (especially when a vital sensor for localization fails, hardware failure)
- Localization algorithm fails to detect the actual robot's position (software failure)
- Need for creating a new map due to changes in the industrial environment.

Teleoperation service is a Robot Operating System (ROS) node hosted in a docker environment, which is placed in each robot's PC. This node is an executable file within a ROS package. ROS is an open-source robotics middleware suite. ROS nodes use a ROS client library to communicate with other nodes. They can publish or subscribe to a Topic. Nodes can also provide or use a Service. This specific node, teleoperation service, is responsible for converting the movement triggered from the HMI, in a motion command to the robot's motor base. Figure 6-9 shows the schematical representation of the teleoperation functionality developed for Demo#4.

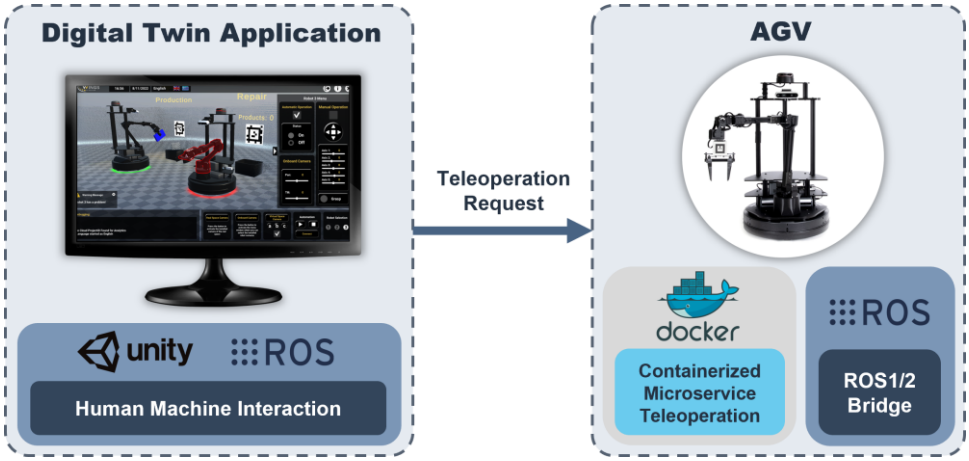


Figure 6-9: Schematical representation of teleoperation service.

6.3 Demonstration results

The orchestration and diagnostic related results from Demo#4 are reported in [Hexa-X D6.3]. The rest of the results obtained are reported in this subsection.

The automated reallocation of functionality and reorchestration of cobots' roles reflects thus to a more resilient of the system, also when considering the energy consumption and the cycle execution times of the system. Detailed measurements were performed related to the energy consumption and operation cycle execution times for the various scenarios. Figure 6-10 shows the battery level with time of a robot having three roles (Product Picking, Shipping, Repair) on it, when having two roles and when having one role. Intuitively, the more roles a robot is allocated with, the higher its energy consumption is in time, thus the faster this robot is exhausted in terms of available energy. Figure 6-11 shows the total system energy consumption when using one robot conducting all three roles, when using two robots and when using three. As it was expected, the more robots are used to complete a process, the higher the energy requirements for the overall system are.

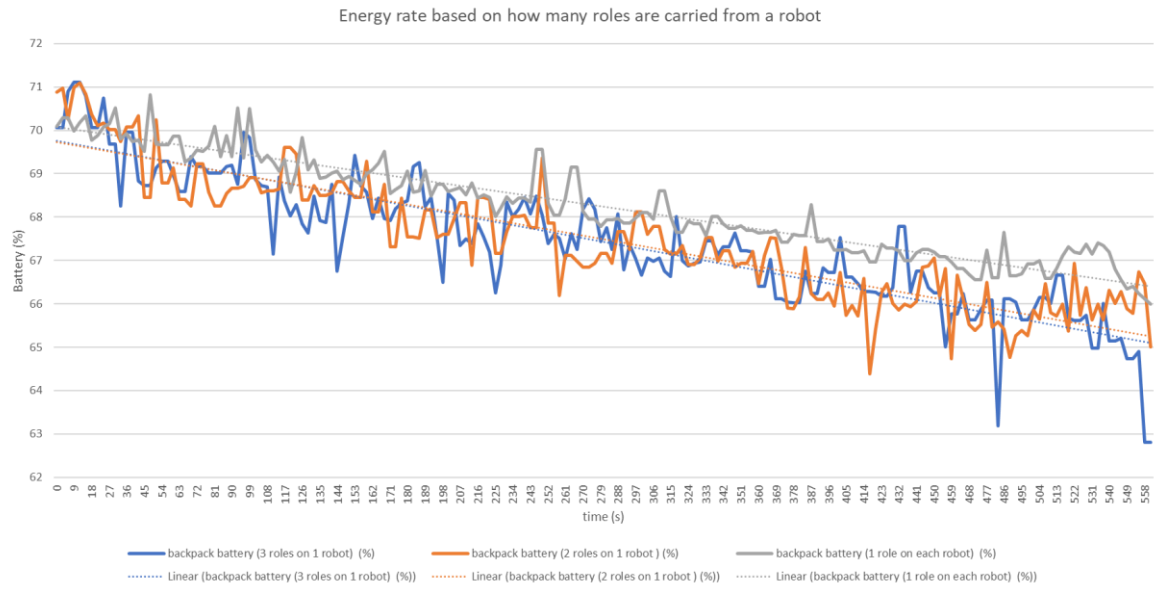


Figure 6-10: Battery level of a robot having 3 roles, e.g., Product Picking, Shipping, Repair (blue), 2 roles (orange) and 1 role (grey) on it with time.

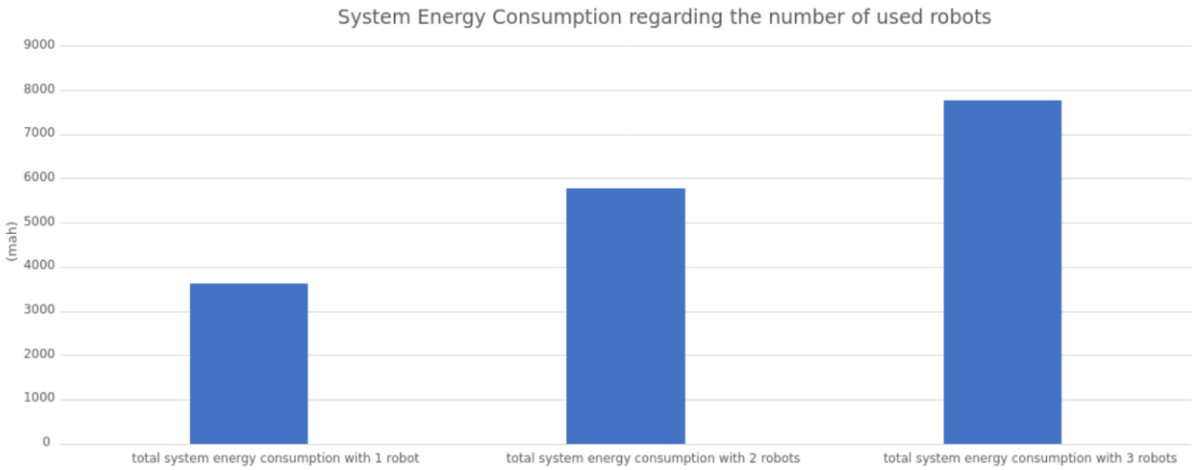


Figure 6-11: Total system energy consumption when using one, two and three robots.

However, saving energy is not always the priority. Following the previous rationale, in the case of one or two robots undertaking more roles in the system –in the case of a fault- will lead to a decrease of the lifetime of the system. Figure 6-12 shows the average lifespan when one role, two or three roles are on each robot. The more roles are assigned to a robot, the less lifespan is observed. Figure 6-13 shows the number of completed rounds succeeded in a lifespan when one role, two or three roles are on each robot. It is observed that the more roles are assigned on a robot, the smaller number of completed rounds are succeeded. Particularly, a system with 3 cobots has a lifespan of T=275 min, during which it conducts roughly ~150 operation cycles, meaning that at the end of the 275 min window, their available energy is depleted. In a second case, it is assumed that a fault occurs, at ~160 min. A reallocation is triggered automatically, which takes place in the system in average in the order of ~275 milliseconds, by the orchestration components, supported by functionality allocation algorithm. This is an end-to-end latency involving triggering the alert, running the algorithm, and forwarding the functionality allocation-related decision making to the various involved interfaces for reorchestrating the whole process. The reallocation-related latency is thus almost negligible compared to the cycle operation time. In the case of the 3 cobot-operation, 150 operation cycles are overall conducted until the cobots'

batteries are depleted. In case of the fault occurring and 2 cobots undertaking all roles, 100 operations are only conducted, since the two robots have been doing more work, thus the remaining robot energy is depleted considerably sooner. Specifically, the energy consumption per operation cycle, in the 3 cobot-case is 330 mAh; on the other hand, in the 2 cobot- and 1-cobot cases, this increases to ~500 and ~600mAh respectively (resulted from Figure 6-12 and knowing that a battery has 50000mAh).

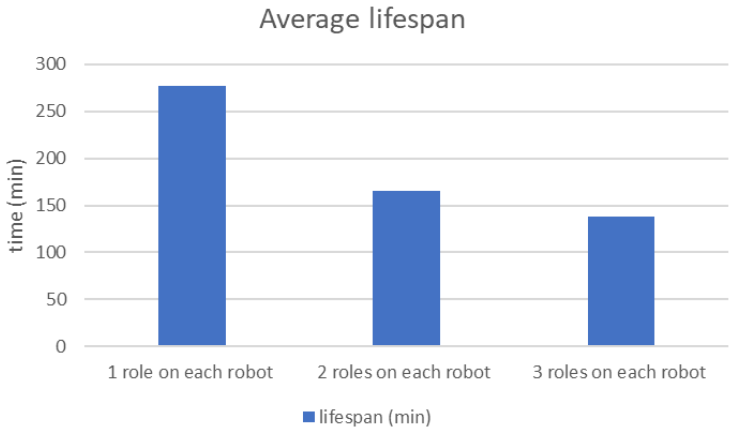


Figure 6-12: Average lifespan when 1, 2, or 3 roles are on each robot.

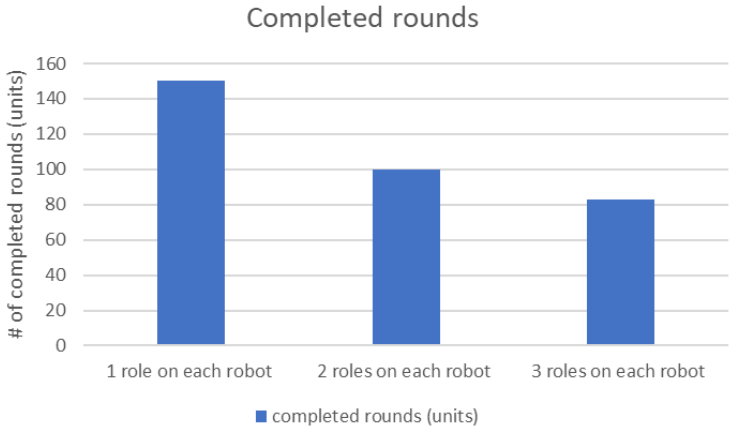


Figure 6-13: Number of completed rounds succeeded in a lifespan (right) when 1,2, or 3 roles are on each robot.

The results of the measurements around the energy consumption for the execution of a robotic process come more or less to confirm what was expected and aimed at, from the beginning. The system can continue to operate, with a higher fatigue / energy consumption for the devices that remain in operation. The system capability to seamlessly reallocate the functionality among the cobots, along with the ability to measure the real energy consumption needs for the various robot roles and the system as a whole, enables the effective design of the system, considering the various energy consumption – operation cycles trade-offs.

Some additional results follow. Figure 6-14 and Figure 6-15 presents the latency derived using the ping method when WiFi, 4G or commercial 5G connectivity is utilised. An Internet Control Message Protocol (ICMP) packet is sent with ping method to the host of interest waiting for an ICMP echo reply. Figure 6-14 refers to network layer latency meaning the latency observed when running a ping to a random server from a robot (LoCoBot) and Figure 6-15 refers to service layer latency meaning the latency observed when an ICMP packet is sent to a service that is enclosed in a Docker environment with ros1_bridge, using ROS2 middleware. ROS2 was selected because it is known to enable the

creation of a fully distributed system, keeping all nodes independent as well as offering a rich variety of QoS policies that allow configuring communication between nodes. The network bridge ros1_bridge enables the exchange of messages between ROS1 and ROS2. The throughput of the WiFi, 4G and 5G network used, and the maximum, minimum and mean values of latency, can be found in Table 9. In both Figure 6-14 and Figure 6-15, WiFi gives the lowest latency performance. 4G and commercial 5G have almost identical curves in both graphs, but if we compare the mean values shown in Table 9, commercial 5G appears to have a bit lower latency performance than 4G in both network and service layer. The range though is approximately 40 ms in network layer latency and 900 ms in service layer latency. This is probably due to interference problems since measurements were collected in indoor settings.

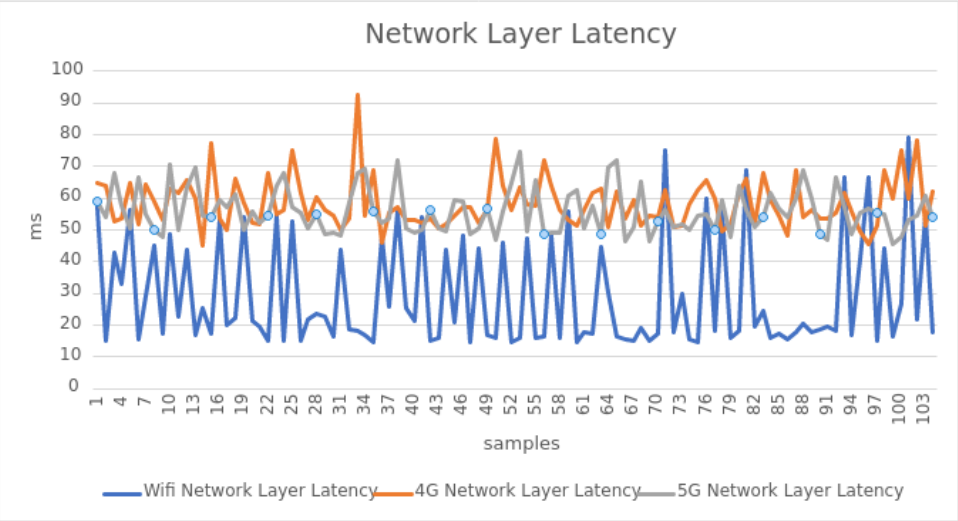


Figure 6-14: Network layer latency measurements with WiFi, 4G and commercial 5G connectivity.

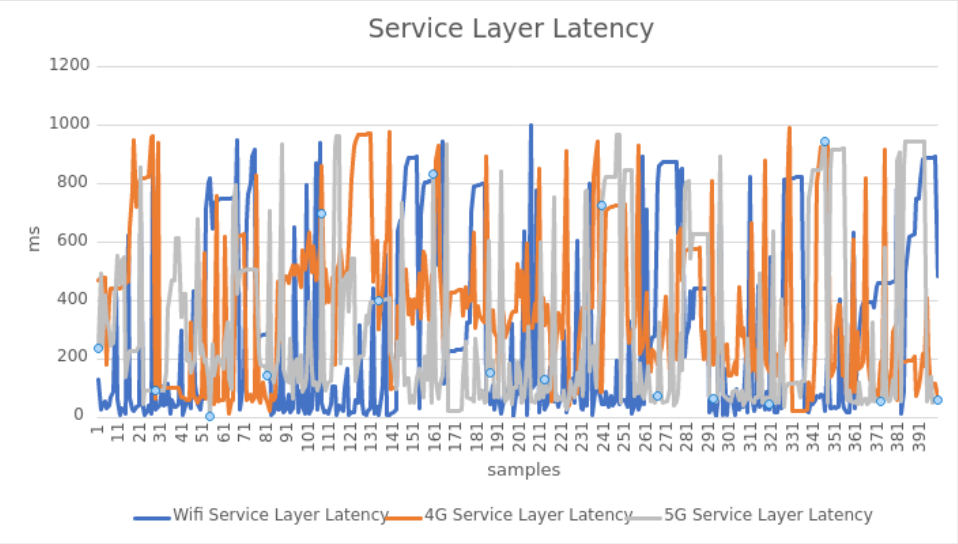


Figure 6-15: Service layer latency measurements with WiFi, 4G and commercial 5G connectivity.

Table 9: Wifi, 4G and 5G throughput and latency information.

	Throughput (Mbps)		Network layer latency (ms)			Service layer latency (ms)		
	Download	Upload	mean	max	min	mean	max	min
Wi-Fi	92.77	87.91	**29.38**	79	14.7	**299.84**	1000.41	4.90
4G	158.66	20.96	**58.38**	92.8	45.3	**371.02**	994.35	22.34
Commercial 5G	280.68	68.42	**56.01**	74.5	45.7	**298.66**	969.39	5.23

The average execution time of the Functionality Allocation algorithm in the scenario of Demo#4 where 5 HEs are assumed (3 robots and 2 servers) and approximately 8 FEs (services) need to be reallocated since they were hosted on a robot which appears with fault, is 0.2759 sec.

Figure 6-16 shows the voltage and current measurements of robot's battery when all services and only the bare-minimum services are hosted on the robot.

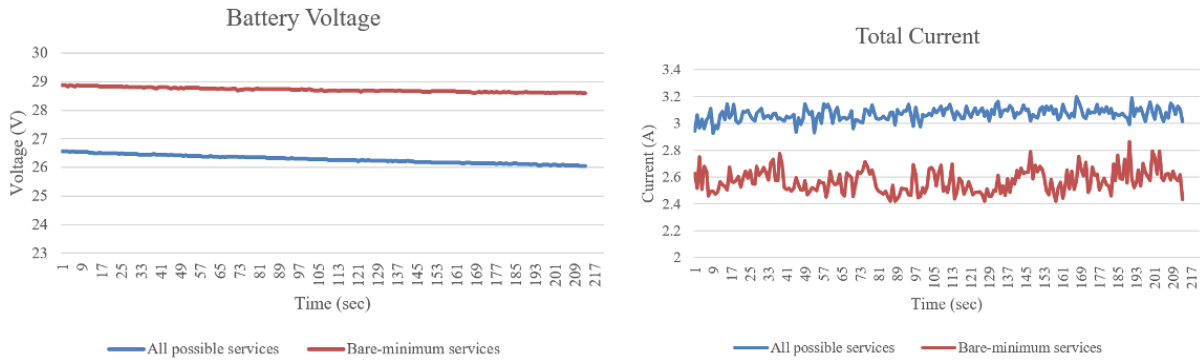


Figure 6-16: Voltage (left) and current (right) measurements of robot's battery when it is loaded with all possible services and when only bare-minimum services are loaded.

Moreover, battery level percentage was measured with time, in the case where all services are assigned on the robot and the case where only bare-minimum services are assigned (Figure 6-17). In both cases, it is obvious the decrease of battery level with time. As expected, the battery level percentage is higher when only bare-minimum services are loaded on robot.

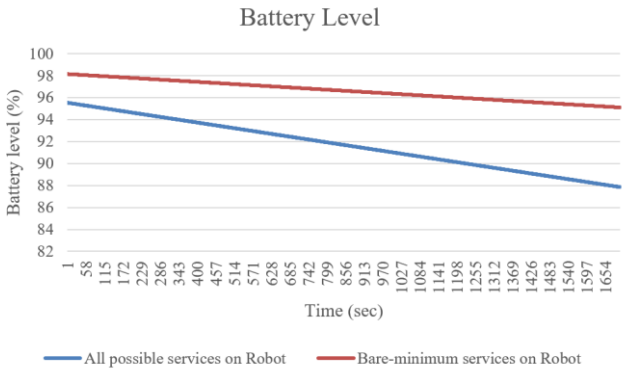


Figure 6-17: Battery level measurements of robot (LoCoBot) when it is loaded with all possible services and when only bare-minimum services are loaded.

Figure 6-18 and Figure 6-19 present RAM and CPU usage when bare-minimum services and when all possible services are loaded to the robot, respectively. Previous voltage, current, power consumption, temperature measurements and these RAM and CPU usage measurements are mainly presented to show how demo's architecture and components (e.g., Functionality allocation, intelligent orchestration) can contribute to energy efficiency.

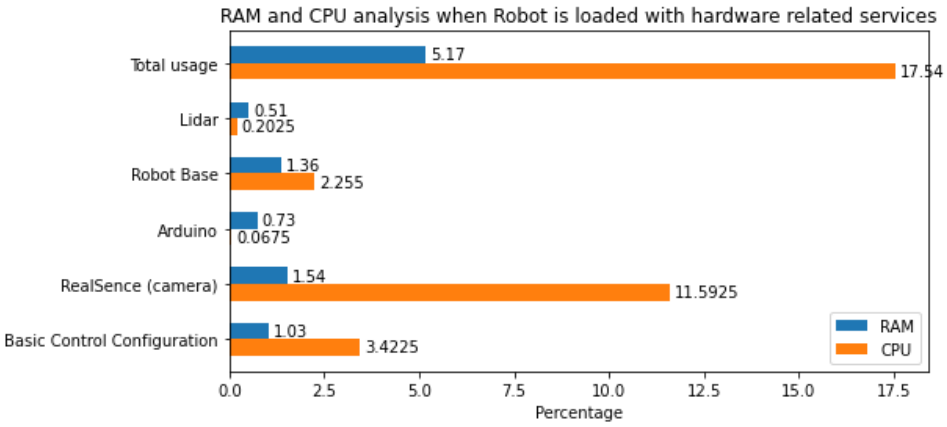


Figure 6-18: RAM and CPU analysis when only hardware related services are loaded on robot.

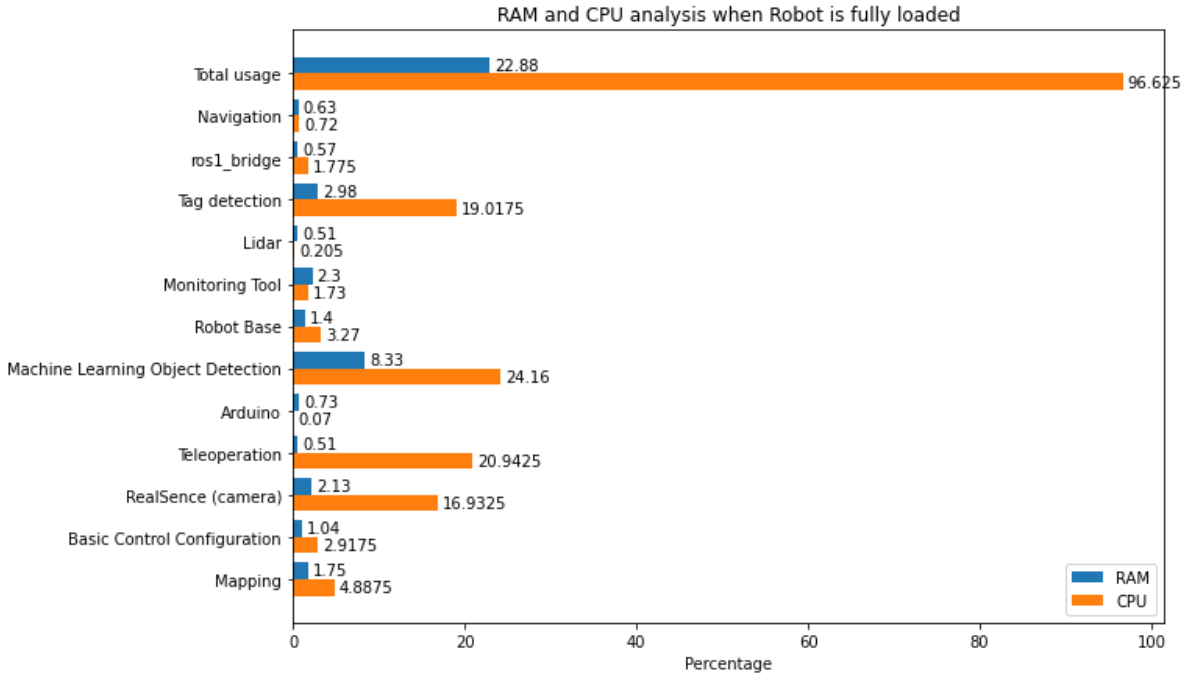


Figure 6-19: RAM and CPU analysis all possible services are loaded on robot.

Additional results are presented in [Hexa-X D6.3], specifically the workflow time, i.e., the time that it takes from the unexpected event appearance until the service is available again and the down time (time that service was unavailable) for each of the three scenarios described previously in Section 6.1, handling four types of events that can occur during the automated operations in the industrial context (robot malfunction, increasing load/low battery etc.). Workflow time consist of: (i) the time it takes for the monitoring system to check the status of the HE, (ii) the time it takes for the monitoring system to detect the issue, (iii) the time it takes for the Functionality allocation to be triggered, (iv) the time it

takes for Functionality allocation to be completed (trigger and receive FEs requirements and available HEs, status and capabilities, calculation of best solution) and (v) the time it takes for the service to be restored due to service initialisation or management orchestrations. Results showed that the third scenario, having predictive orchestration and Functionality allocation, had the best workflow time and down time. Workflow time was 8.7-11.9 s and the down time was approximately 2 s. If we compare the down time of the first scenario, having typical management and orchestration, which was 21-515 s with the third scenario (2 s), we can say that the maintainability KPI is high.

Concluding we could say that there were important research findings from this PoC and future work could be the assessment on node's trustworthiness when reallocating functionalities and services as well as the integration of networking enablers, e.g., flexible topologies described in WP5 [Hexa-X D5.2], [Hexa-X D5.3].

7 Conclusions

This deliverable presented the final solutions of **WP7: special-purpose functionality** with respect to the project objective **network evolution and expansion towards 6G**, addressing the need for special-purpose functionality in extreme environments. Based on the intermediate results presented in [Hexa-X D7.2], this deliverable reported the final technical contributions for *flexible resource allocation in challenging environments* (Task T7.2, reported in Section 3), *dependability in I4.0 environments* (Task T7.3, reported in Section 4) and *Digital Twins and novel HMIs* (Task T7.4, reported in Section 5). Additionally, the results of the demonstrator *extreme performance in handling unexpected situations in industrial contexts* were presented in detail (Section 6). The relation of all contributions to the technical and architectural enablers in Hexa-X, the KPIs and KVIs, as well as the objectives, outputs, and measurable results was detailed (Section 2). In the following, we briefly summarize the main contributions towards the work package objectives included in this deliverable and in earlier WP7 deliverables [Hexa-X D7.1] and [Hexa-X D7.2].

In-depth gap analysis of existing special-purpose functionality, sharpening of requirements and formulation of first solutions for extreme environments: A detailed analysis of use cases with extreme environments and requirements motivating special-purpose solutions was published in [Hexa-X D7.1], grouped into "dependability in Industry 4.0" and "sustainable coverage in IoT". A detailed work plan for the technical tasks on dependability and flexible resource allocation in WP7 based on discussion of relevant State of the Art was proposed in [Hexa-X D7.1] and further updated in [Hexa-X D7.2], Section 5 for HMIs and Digital Twins. First solutions were discussed in [Hexa-X D7.2].

Ultra-flexible resource allocation procedures in challenging environments such as those populated by mobile devices with special requirements and in need of coverage: State of the Art was discussed in [Hexa-X D7.1] and intermediate solutions were outlined in [Hexa-X D7.2]. This final deliverable included the results on mechanisms and models for radio-aware trajectory planning (Section 3.2), resource allocation in industrial environments (Sections 3.1 and 3.3), resource provisioning for Federated Learning in resource-constrained IoT environments (Section 3.4), and utilization of ambient backscatter communication for zero-energy devices (Section 3.5).

Mechanisms and enablers for high dependability in vertical scenarios, enabling efficient resource support of complex and dynamically changing availability requirements: State of the Art, respective use cases and their requirements, including a work plan on how to address the gaps were discussed in [Hexa-X D7.1]. Initial solutions were presented in [Hexa-X D7.2]. The final results that were discussed in this deliverable addressed technical enablers for dependability in Section 4.1 (radio resource management with digital twins, UAV-assisted mMTC, data and control plane guarantees and network data analytics assisted AI operations) and the framework of Communication-Computation-Control-Co-Design in Section 4.2. Works on dependability in [Hexa-X D7.2] and resilience and the related KVI trustworthiness have been extended in collaboration with WP1 and are published in [Hexa-X D1.4].

Human interaction through novel HMI concepts and privacy-preserving high-availability Digital Twins to support the convergence of the biological, digital, and physical worlds: Initial ideas and use cases including KPIs were outlined in [Hexa-X D7.1]. Details on State of the Art on novel HMIs and privacy-preserving Digital Twins were provided in [Hexa-X D7.2], Section 5, including initial solutions for Digital Twin-empowered collaborative robots, network awareness of Digital Twins, and Digital Twins for Emergent Intelligence. In this deliverable, we provided an updated assessment of novel HMIs in Section 5.1. A model for the impact of human presence on DTs was discussed in Section 5.2. The use case of collaborative robots and its realization with DTs was detailed in Section 5.3. The flexible functional split discussed already in [Hexa-X D7.2] was augmented with DTs in Section 5.4. The topic of emergent intelligence and related trust and security concerns were discussed in Section 5.5, further augmenting the discussion of trustworthiness in [Hexa-X D1.4].

Even during the last months of the Hexa-X project, additional potential for collaboration and follow-ups have been identified towards the realization of 6G systems. The authors hope to address some of

these identified aspects in Hexa-X-II[11] or in other common activities in the future and would like to thank all colleagues and partners who contributed to Hexa-X and especially the work in WP7 and all joint activities during the project lifetime.

[11] https://hexa-x-ii.eu/

References

[23.288] 3GPP TS 23.288 V18.1.0 (2023-3) Architecture enhancements for 5G System (5GS) to support network data analytics services (Release 17)

[23.502] 3GPP TS 23.502 V18.0.0 (2022-12) Procedures for the 5G System (5GS), Stage 2

[36.211] 3GPP TS 36.211 V8.9.0 (2009-12) Technical Specification 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 8).

[37.320] 3GPP TS 37.320 V16.2.0 (2020-11) Radio measurement collection for Minimization of Drive Tests

[37.817] 3GPP TS 37.817 V17.0.0 (2022-4) E-UTRA and NR; Study on enhancement for data collection for NR and EN-DC (Release 17)

[38.211] 3GPP TS 38.211 V17.4.1 (2022-07) Physical channels and modulation

[38.331] 3GPP TS 38.331 V17.4.0 (2023-03) NR; Radio Resource Control (RRC) Protocol Specification (Release 17)

[38.801] 3GPP TR 38.801 V14.0.0 (2017-03) Study on new radio access technology: Radio access architecture and interfaces

[38.901] 3GPP TR 38.901, "Study on channel model for frequencies from 0.5 to 100 GHz," Release 16 (v16.1.0), 2019.

[5GACIA20] 5G Alliance for Connected Industries and Automation, "Key 5G Use Cases and Requirements", White Paper, 2020. [Online].

[AAR22] Alharbe, N.; Aljohani, A.; Rakrouki, M.A. A Fuzzy Grouping Genetic Algorithm for Solving a Real-World Virtual Machine Placement Problem in a Healthcare-Cloud. Algorithms 2022, 15, 128. https://doi.org/10.3390/a15040128

[AEFDS19] Mohamed A. Abd-Elmagid, Aidin Ferdowsi, Harpreet S Dhillon, and Walid Saad. Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks. In 2019 IEEE Global Communications Conference (GLOBECOM), pages 1–6. IEEE, 2019.

[AGM+10] Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B., Sridharan, M., "Data center tcp (dctcp)," In Proceedings of the ACM SIGCOMM 2010 Conference (pp. 63-74), August 2010.

[AZA+21] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and m. Grundmann, "Objectron: A Large Scale Dataset of Object-Centric Videos in the Wild with Pose Annotations." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1-11. doi:10.48550/arXiv.2012.09988, 2021.

[BCS+20] E. Battegazzorre, D. Calandra, F. Strada, A. Bottino, and F. Lamberti, "Evaluating the Suitability of Several AR Devices and Tools for Industrial Applications." Vol. 12243, in Augmented Reality, Virtual Reality, and Computer Graphics, by Lucio Tommaso De Paolis and Patrick Bourdot, 248-267. Cham: Springer. doi:10.1007/978-3-030-58468-9_19, 2020.

[BDZ+19] Van Bemten, A., Đerić, N., Zerwas, J., Blenk, A., Schmid, S., and Kellerer, W., "Loko: Predictable latency in small networks," In Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies (pp. 355-369), 2019.

[BGR+20] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. L. Zhu, F. Zhang, and M. Grundmann, "BlazePose: On-device Real-time Body Pose tracking." ArXiv abs/2006.10204: 1-4. doi:10.48550/arXiv.2006.10204, 2020.

[BJP+21] L. Bonati, P. Johari, M. Polese, S. D'Oro, S. Mohanti, M. Tehrani-Moayyed, V. Davide, et al. "Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation." 2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN). IEEE, 2021.

[BT01] Le Boudec, Jean-Yves, and Patrick Thiran, eds. “Network calculus: a theory of deterministic queuing systems for the internet.” Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[BYMS06] J. N. Bailenson, N. Yee, D. Merget, and R. Schroeder, “The Effect of Behavioral Realism and Form Realism of Real-Time Avatar Faces on Verbal Disclosure, Nonverbal Disclosure, Emotion Recognition, and Copresence in Dyadic Interaction.” Presence 15 (4): 359-372. doi:10.1162/pres.15.4.359, 2006.

[CPC+22] D. Calandra, F. G. Pratticò, A. Cannavò, C. Casetti, and F. Lamberti, “Digital twin- and extended reality-based telepresence for collaborative robot programming in the 6G perspective.” Digital Communications and Networks 1-16. doi:10.1016/j.dcan.2022.10.007, 2022.

[DN06] D. Samardzija and N. Mandayam, "Unquantized and uncoded channel state information feedback in multiple-antenna multiuser systems," in IEEE Transactions on Communications, vol. 54, no. 7, pp. 1335-1345, July 2006.

[EGR+15] Emmerich, P., Gallenmüller, S., Raumer, D., Wohlfart, F., and Carle, G. “Moongen: A scriptable high-speed packet generator.” In Proceedings of the 2015 Internet Measurement Conference (pp. 275-287), 2015.

[EPS22] E. Eldeeb et al., "A Learning-Based Trajectory Planning of Multiple UAVs for AoI Minimization in IoT Networks," 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Grenoble, France, 2022.

[FHC+22] Zexin Fang, Bin Han, C. Clark Cao, and Hans D. Schotten, “Artificial ASMR: A cyber-psychological study,” preprint, *arXiv:2210.14321*, October 2022.

[GMMM14] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion.” Pattern Recognition 47 (6): 2280-2292. doi:10.1016/j.patcog.2014.01.005, 2014.

[GSG+15] Grosvenor, M. P., Schwarzkopf, M., Gog, I., Watson, R. N., Moore, A. W., Hand, S., & Crowcroft, J., “Queues don’t matter when you can JUMP them!”. In 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15) (pp. 1-14), 2015.

[Hexa-X D1.1] Hexa-X, “Deliverable D1.1: 6G Vision, use cases and key social values”, March 2021.

[Hexa-X D1.2] Hexa-X, “Deliverable D1.2: Expanded 6G vision, use cases and societal values – including aspects of sustainability, security and spectrum”, April 2021.

[Hexa-X D1.3] Hexa-X, “Deliverable D1.3: Targets and requirements for 6G – initial E2E architecture”, March 2021.

[Hexa-X D1.4] Hexa-X, “Deliverable D1.4: Hexa-X architecture for B5G/6G networks – final release”, to be submitted, June 2023.

[Hexa-X D2.1] Hexa-X, “Deliverable D2.1: Towards Tbps Communications in 6G: Use cases and Gap Analysis”, June 2021.

[Hexa-X D2.2] Hexa-X, “Deliverable D2.2: Initial radio models and analysis towards ultra-high data rate links in 6G”, January 2022.

[Hexa-X D2.3] Hexa-X, “Deliverable D2.3: Radio models and enabling techniques towards ultra-high data rate links and capacity in 6G”, April 2023.

[Hexa-X D3.1] Hexa-X, “Deliverable D3.1: Localisation and sensing use cases and gap analysis”, Jan. 2022.

[Hexa-X D3.2] Hexa-X, “Deliverable D3.2: Initial models and measurements for localisation and sensing”, October 2022.

[Hexa-X D3.3] Hexa-X, “Deliverable D3.3: Final models and measurements for localisation and sensing”, May 2023.

[Hexa-X D4.1] Hexa-X, “Deliverable D4.1: AI-driven communication & computation co-design: Gap analysis and blueprint”, August 2021.

[Hexa-X D4.2] Hexa-X, “Deliverable D4.2: AI-driven communication & computation co-design: initial solutions”, July 2022.

[Hexa-X D4.3] Hexa-X, "Deliverable D4.2: AI-driven communication & computation co-design: final solutions", April 2023.
[Hexa-X D5.1] Hexa-X, "Deliverable D5.1: Initial 6G Architectural Components and Enablers", December 2021.
[Hexa-X D5.2] Hexa-X, "Deliverable D5.2: Analysis of 6G architectural enablers applicability and initial technological solutions", November 2022.
[Hexa-X D5.3] Hexa-X, "Deliverable D5.3: Final 6G architectural enablers and technological solutions", May 2023.
[Hexa-X D6.1] Hexa-X, "Deliverable D6.1: Gaps, features and enablers for B5G/6G service management and orchestration", July 2021.
[Hexa-X D6.2] Hexa-X, "Deliverable D6.2: Design of service management and orchestration functionalities", May 2022.
[Hexa-X D7.1] Hexa-X, "Deliverable D7.1: Gap analysis and technical work plan for special-purpose functionality", July 2021.
[Hexa-X D7.2] Hexa-X, "Deliverable D7.2: Special-purpose functionalities: intermediate solutions", May 2022.
[HKZ+22] Bin Han, Dennis Krummmacher, Qiuheng Zhou, and Hans D. Schotten, "Trust-awareness to secure swarm intelligence from data injection attack," to appear in the *2023 IEEE International Conference on Communications (ICC)*, Rome, Italy, June 2023.
[HLC21] Mohammad Hatami, Markus Leinonen, and Marian Codreanu. AoI minimization in status update control with energy harvesting sensors. IEEE Transactions on Communications, 69(12):8335–8351, 2021.
[HS22] Bin Han and Hans D. Schotten, "Multi-sensory HMI to enable digital twins with human-in-loop: A 6G vision of future industry," in the *4th International IEEE Workshop on Social (Media) Sensing (SMS 2022)*, Rhodes Island, Greece, June 2022.
[HSM+22] Bin Han, Muxia Sun, Lai-Kan Muk, Yan-Fu Li, and Hans D. Schotten, "Flexible and dependable manufacturing beyond xURLLC: A novel framework for communication-control co-design," in the *2022 IEEE International Workshop on Predictive Maintenance (PM 2022)*, Guangzhou, China, December 2022.
[ICNIRP20] International Commission on Non-Ionizing Radiation Protection (ICNIRP), "ICNIRP guidelines for limiting exposure to electromagnetic fields (100 kHz to 300 GHz)", 1998.
[JSB+15] Jang, K., Sherry, J., Ballani, H., and Moncaster, T., "Silo: Predictable message latency in the cloud," In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (pp. 435-448), 2015.
[KDF+23] S. Kulkarni, S. Deshmukh, F. Fernandes, A. Patil, and V. Jabade, "PoseAnalyser: A Survey on Human Pose Estimation." SN Computer Science 4 (136): 2661-8907. doi:10.1007/s42979-022-01567-2, 2023.
[KKS+19] S. Knopp, P. Klimant, R. Schaffrath, E. Voigt, R. Fritzsche, and C. Allmacher, "Hololens AR - Using Vuforia-Based Marker Tracking Together with Text Recognition in an Assembly Scenario." 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). Beijing, China. 63-64. doi:10.1109/ISMAR-Adjunct.2019.00030, 2019.
[KSUB+18] Igor Kadota, Abhishek Sinha, Elif Uysal-Biyikoglu, Rahul Singh, and Eytan Modiano. Scheduling policies for minimizing age of information in broadcast wireless networks. IEEE/ACM Transactions on Networking, 26(6):2637–2650, 2018.
[Lap08] J. Laprie, "From Dependability to Resilience." Dependable Systems and Networks," 38th IEEE/IFIP Int. Conf. On dependable systems and networks, 2008.
[LGR+18] B. C. Lochte, S. A. Guillory, C. A. Richard, and W. M. Kelley, "An fMRI investigation of the neural correlates underlying the autonomous sensory meridian response (ASMR)," BioImpacts : BI, vol. 8, p. 295, 2018.

[LML+19] Li, Y., Miao, R., Liu, H. H., Zhuang, Y., Feng, F., Tang, L. and Yu, M., "HPCC: High precision congestion control," In Proceedings of the ACM Special Interest Group on Data Communication (pp. 44-58), 2019.

[LMS+19] A. Luxenburger, J. Mohr, T. Spieldenner, D. Merkel, F. Espinosa, T. Schwartz, F. Reinicke, J. Ahlers, and M. Stoyke, "Augmented Reality for Human-Robot Cooperation in Aircraft Assembly." 2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR). San Diego, CA: IEEE. 263-2633. doi:10.1109/AIVR46125.2019.00061, 2019.

[MBC22] M. Merluzzi, S. Bories and E. C. Strinati, "Energy-Efficient Dynamic Edge Computing with Electromagnetic Field Exposure Constraints," 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Grenoble, France, 2022.

[MNB+21] R. C. Moioli, P. H. J. Nardelli, M. T. Barros, et al., "Neurosciences and wireless networks: The potential of brain-type communications and their applications," in IEEE Communications Surveys & Tutorials, 23.3, 2021.

[MS96] Z. Michalewicz and M. Schoenauer, "Schoenauer, m.: Evolutionary algorithms for constrained parameter optimization problems. Evolutionary computation 4(1), 1-32," Evolutionary Computation, vol. 4, pp. 1–32, 03, 1996.

[MSD11] Mitchell S, O'Sullivan M, Dunning (2011) Pulp: A linear programming toolkit for python. Accessed May 1, 2013, https://code.google.com/p/pulp-or/

[MVGM17] R. Mueller, M. Vette, A. Geenen, and T. Masiak, "Improving Working Conditions in Aircraft Productions Using Human-Robot-Collaboration in a Collaborative Riveting Process." SAE Technical Papers. doi:10.4271/2017-01-2096, 2017.

[MYG+20] M. B. Mashhadi, Q. Yang and D. Gündüz, "CNN-Based Analog CSI Feedback in FDD MIMO-OFDM Systems," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

[Ora21] Orange Press Release at MWC'21, 28th June 2021, available at: https://www.orange.com/sites/orangecom/files/documents/2021-06/MWC-2021-press-folder-en.pdf, 2021.

[PBR+21] D. -T. Phan-Huy, D. Barthel, P. Ratajczak, R. Fara, M. d. Renzo and J. d. Rosny, "Ambient Backscatter Communications in Mobile Networks: Crowd-Detectable Zero-Energy-Devices," 2021 IEEE International Conference on RFID Technology and Applications (RFID-TA), 2021.

[PBR+22] D. -T. Phan-Huy, D. Barthel, P. Ratajczak, R. Fara, M. d. Renzo and J. de Rosny, "Ambient Backscatter Communications in Mobile Networks: Crowd-Detectable Zero-Energy-Devices," in IEEE Journal of Radio Frequency Identification, vol. 6, pp. 660-670, 2022.

[PBT+18] G. L. Poerio, E. Blakey, T. J. Hostler, et al., "More than a feeling: Autonomous sensory meridian response (ASMR) is characterized by reliable changes in affect and physiology," in PloS One, 13.6, 2018.

[PGB+20] F. De Pace, G. Gorjup, H. Bai, A. Sanna, M. Liarokapis, and M. Billinghurst, "Assessing the Suitability and Effectiveness of Mixed Reality Interfaces for Accurate Robot Teleoperation." Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology. Association for Computing Machinery. 1-3. doi:10.1145/3385956.3422092, 2020.

[PGB+21] F. De Pace, G. Gorjup, H. Bai, A. Sanna, M. Liarokapis, and M. Billinghurst, "Leveraging Enhanced Virtual Reality Methods and Environments for Efficient, Intuitive, and Immersive Teleoperation of Robots." 2021 IEEE International Conference on Robotics and Automation (ICRA). Xi'an, China. 12967-12973. doi:10.1109/ICRA48506.2021.9560757, 2021.

[PL21] F. G. Pratticò and F. Lamberti, "Towards the adoption of virtual reality training systems for the self-tuition of industrial robot operators: A case study at KUKA." Computers in Industry 129: 103446. doi:10.1016/j.compind.2021.103446, 2021.

[PMR+21] J. Pegoraro, F. Meneghello and M. Rossi, "Multiperson Continuous Tracking and Identification From mm-Wave Micro-Doppler Signatures," in IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 4, pp. 2994-3009, April 2021.

[PPV10] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," IEEE Transactions on Information Theory, vol. 56, no. 5, pp. 2307–2359, 2010.

[PZLJ+16] J. Pan, C. Zhou, B. Liu and K. Jiang , "Joint DOA and Doppler frequency estimation for coprime arrays and samplers based on continuous compressed sensing," 2016 CIE International Conference on Radar (RADAR), 2016.

[PZLJ16] J. Pan, C. Zhou, B. Liu, and K. Jiang, "Joint DOA and Doppler frequency estimation for coprime arrays and samplers based on continuous compressed sensing." International Conference on Radar (RADAR), 2016.

[RWL+22] K. Ruttik, X. Wang, J. Liao, R. Jäntti and P. -H. Dinh-Thuy, "Ambient backscatter communications using LTE cell specific reference signals," 2022 IEEE 12th International Conference on RFID Technology and Applications (RFID-TA), Cagliari, Italy, 2022.

[RZM+22] F. Raviglione, S. Zocca, A. Minetto, M. Malinverno, C. Casetti, C. Chiasserini, and F. Dovis, "From collaborative awareness to collaborative information enhancement in vehicular networks," Vehicular Communications (Elsevier), vol. 36, 2022.

[SMR+18] M. Scalabrin, N. Michelusi and M. Rossi, "Beam Training and Data Transmission Optimization in Millimeter-Wave Vehicular Networks," IEEE Global Communications Conference (GLOBECOM), 2018.

[SSY+19] C. Sun, C. She, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications," IEEE Transactions on Wireless Communications, vol. 18, no. 1, pp. 402–415, 2019.

[SSY+19] C. Sun, C. She, C. Yang, T. Q. S. Quek, Y. Li, and B. Vucetic, "Optimizing resource allocation in the short blocklength regime for ultra-reliable and low-latency communications," IEEE Transactions on Wireless Communications, vol. 18, no. 1, pp. 402–415, 2019.

[TSK+15] D. W. Tan, M. A. Schiefer, M. W. Keith, et al., "Stability and selectivity of a chronic, multi-contact cuff electrode for sensory stimulation in human amputees," in Journal of Neural Engineering, 12.2, 2015.

[URB+21] M. A. Uusitalo, P. Rugeland, M. R. Boldi, E. C. Strinati, P. Demestichas, M. Ericson, G. P. Fettweis, M. C. Filippou, A. Gati, M.-H. Hamon, M. Hoffmann, M. Latva-Aho, A. Pärssinen, B. Richerzhagen, H. Schotten, T. Svensson, G. Wikström, H. Wymeersch, V. Ziegler and Y. Zou, "6G Vision, Value, Use Cases and Technologies From European 6G Flagship Project Hexa-X," in IEEE Access, vol. 9, pp. 160004-160020, 2021.

[WBM22] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." ArXiv abs/2207.02696: 1-15. doi:10.48550/arXiv.2207.02696, 2022.

[WPF+09] M. Wildemeersch, J. Petit and J. Fortuny-Guasch, "Doppler radar and postprocessing techniques for small area surveillance," IEEE 10th Workshop on Signal Processing Advances in Wireless Communications, 2009.

[WWB+23] P. Wang, Y. Wang, M. Billinghurst, H. Yang, P. Xu, and Y. Li., "BeHere: a VR/SAR remote collaboration system based on virtual replicas sharing gesture and avatar in a procedural task." Virtual Reality (Springer) 1434-9957. doi:10.1007/s10055-023-00748-5, 2023.

[YHK+22] Siyu Yuan, Bin Han, Dennis Krummacker, and Hans D. Schotten, "Massive twinning to enhance emergent intelligence," in the *2nd International Workshop on Distributed and Intelligent Systems (DistInSys 2022)*, Rhodes Island, Greece, June 2022.

[YZ+19] L. Yang and W. Zhang, "Beam Tracking and Optimization for UAV Communications," in IEEE Transactions on Wireless Communications, vol. 18, no. 11, pp. 5367-5379, Nov. 2019.

[ZEF+15] Zhu, Y., Eran, H., Firestone, D., Guo, C., Lipshteyn, M., Liron, Y., and Zhang, M., "Congestion control for large-scale RDMA deployments," ACM SIGCOMM Computer Communication Review, 45(4), 523-536, 2015.

[ZHT+21] C. Zhe, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields." IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (1): 172-186. doi:10.1109/TPAMI.2019.2929257, 2021.