GRANT AGREEMENT NO.: 732174

Call: H2020-ICT-2016-2017
Topic: ICT-13-2016
Type of action: RIA

Orchestration and Reconfiguration Control Architecture

D2.5: Final Evaluation of the Showcases

Revision: v.1.0

Work package	WP 2
Task	Task 2.3
Due date	31/1/2020
Submission date	31/1/2020
Deliverable lead	TUD
Version	1.0
Authors	Roberto Bomfin (TUD), Joao Santos (TCD), Wei Liu (IMEC), Muhammed Aslam (IMEC), Xianjun Jiao (IMEC), Clemens Felber (NI), Funing Xia (NI), Prashanth K H Sheshagiri (NI), Seyed Ali Hassani (KUL)
Reviewers	Ingrid Moerman (IMEC), Walter Nitzold (NI)

Abstract	This document intends to describe the ORCA showcases based on the development of year 3. In particular, this set of showcases focus on an industrial wireless communication, where various ORCA functionalities are applied, including mmWave, low-latency remote control of robots, multi-rat and end-to-end network slicing.
Keywords	mmWave, low-latency, multi-rat, remote control, slicing

Disclaimer

The information, documentation and figures available in this deliverable, is written by the ORCA (Orchestration and Reconfiguration Control Architecture) – project consortium under EC grant agreement 732174 and does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Confidential *- The information contained in this document and any attachments are confidential. It is governed according to the terms of the project consortium agreement*

Copyright notice

Acknowledgment

This report has received funding from the EC under the grant agreement 731274.

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	R*	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CI	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to ORCA project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

EXECUTIVE SUMMARY

One target of ORCA is to develop and offer end-to-end (E2E) communication systems as a research facility to the research community, which utilize novel Software-defined Radio (SDR) and Software-defined Networking (SDN) solutions. This deliverable focuses on the E2E demonstrators of ORCA based on the achievements of year 3, which are designed to address the challenges encountered in an industrial environment. The demonstrators include the following 4 showcases. Showcase 1 covers transmission over the 26 GHz mmWave band with real-time beam tracking in a mobile scenario. Showcase 2 demonstrates a dynamic control functionality with very low latency and ultra-high reliability for controlling multiple robotic arms, and a Doppler radar is implemented to sense the velocity of nearby objects to increase reliability and security. Showcase 3 demonstrates the deployment of customised and isolated end-to-end network slices via hybrid SDR/SDN orchestration system, to support communication services with diverging service requirements. Finally, showcase 4 brings the flexible multi-RAT demonstrator that implements WiFi, LTE and a 5G link, which can be parametrized in real-time depending on different network and radio metrics. The individual aspects of each showcase are given in detail through this document. This deliverable reports the aforementioned ORCA showcases, which demonstrate the capability of the ORCA facilities embedding advanced ORCA SDR functionality, and how the functionalities can be used in the future to design novel experiments for advanced wireless application scenarios, covering diverging requirements for 5G and beyond.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AGV	Automated Guided Vehicle
AP	Access Point
API	Application Programming Interface
APP	Application
BS	Base Station
CN	Core Network
CPU	Control Processing Unit
DC	Dual Connectivity
FPGA	Field Programmable Gate Array
GFDM	Generalized Frequency Division Multiplexing
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
IoT	Internet of Thigs
IP	Internet Protocol
ISR	Interrupt Service Routine
LTE	Long Term Evolution
LWA	LTE-WLAN Aggregation
MAC	Multiple Access Control
mmWave	Milimeter Wave
NP	Network Provider
NSaaS	Network Slicing as a Service
PAN	Personal Area Network
PHY	Physical Layer
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RT	Real Time
SDR	Software Defined Radio
SP	Service Provider
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TN	Transport Network
TS	Terminal Station
UART	Universal Asynchronous Receiver/Transmitter
UD	User Device

UDP	User Datagram Protocol
USB	Universal Serial Bus

1 INTRODUCTION

This deliverable describes the showcases developed in the 3rd year of ORCA project. Each of the showcases has different performance attributes and focus area. The showcases are designed in the context of a future factory, it manufactures Organic Recyclable Consumable Additives (ORCA). It should be noted that the context described here motivates how the ORCA showcases can be used in real-life scenario, and should not be taken as the demonstration setup.

The factory has a large area used as warehouse, it is used to store parts and also final products. The parts, many of them recycled from old products, are brought to the assemble line by Automatically Guided Vehicles (AGV). And the final products are brought back to the warehouse also by AGVs. The AGVs are wirelessly controlled with multiple Radio Access Technologies (multi-RATs), including LTE, WiFi, and a 5G link. The presence of multiple RATs makes it possible to take advantage of available facilities on site, and increases the reliability. This is the contribution of showcase 4.

After the parts are brought to the assemble line, they are assembled into the Consumable Additives by robotic arms, which are remotely controlled via wireless connection. The control loop of the robotic arm consists of input from sensor devices (angle sensors, accelerator sensors, speed sensors etc), and output to the motor engines (engines to raise or turn the arm). The loop requires ultralow latency between the input and output and very high reliability. The realization of the control loop is the key contribution of showcase 2.

The final product is visually inspected with a camera installed on another moving robotic. The video is sent via mmWave link with real time beam tracking feature. Beamtracking is necessary to cope with the movement of the robot. In addition, a second mmWave is used as a backhaul to transfer the video stream to remote operator. The remote operator can also steer the robot for the purpose video inspection. The robot's remote control is achieved with a low-latency GFDM link at sub 6GHz frequency band. This is the functionality developed in showcase 1.

The AGVs, robotic arm control, and video streaming are applications that trigger service requests at local gateways when needed. The gateways then send service requests to a hyperstrator, which initiates network slices corresponding to the service requests across multiple network segments. This is achieved by delegating resource allocation and other necessary tasks to underlying orchestrators and lower level network controllers hierarchically. Part of the network slice may involve wired networks and therefore an SDN controller may be used. The construction of network slices through a hierarchical orchestration approach is tackled in showcase 3.

The rest of this deliverable is organized as follows, first we illustrate how the 4 showcases are setup as demos physically forming a future factory at IMEC's IoT Lab in Section 1.1. And then a summary of performance attributes of each of the showcases is given in Section 1.2. After that more details of the showcases are introduced in subsequent sections.

1.1 Demonstration locations

The demos will be setup in the Internet of Things (IoT) lab of IMEC, a high level floorplan is given in Figure 1. We indicate the showcase locations also in this map. Showcase 4 will be visualized with the left part of the map where components and products are stored. Showcase 1 and showcase 2 will be closely located, both showcases use robots and are intended as features used on an assembly line. Showcase 3 is at a global level, it uses most compute resources, since the hyperstrator and other orchestrator entities reside in virtualized containers running on servers, this showcase uses SDRs as infrastructure gateway in order to demonstrate features of radio controllers that would not be possible with commercial hardware.

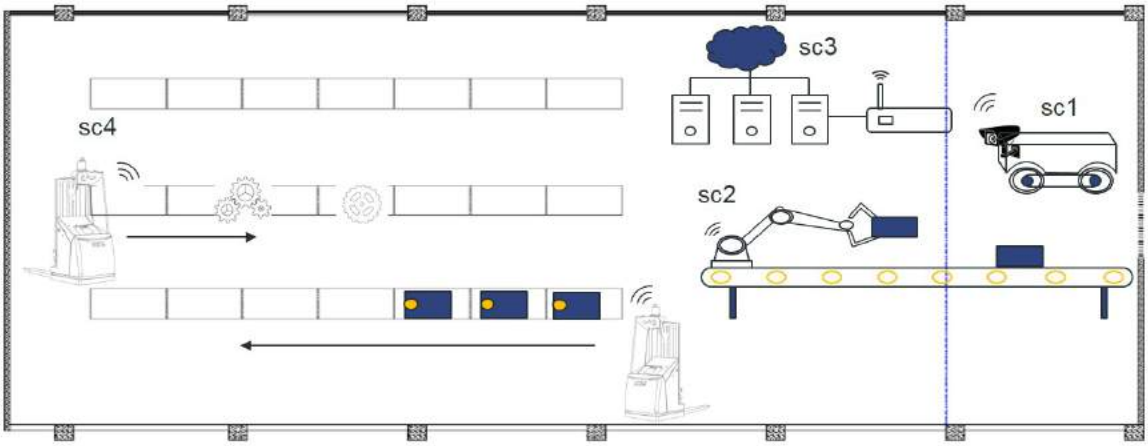

Figure 1 Demo of showcases as part of a future factory at IoT lab IMEC

1.2 Performance attributes

This section introduces the performance attributes of all four showcases.

For showcase 1, the system achieves a moderate throughput in the mmWave uplink and a low-latency and reliable transmission at the downlink for the beam control, providing the capability of dynamic control on the physical layer (PHY).

Showcase 2 focuses on a very low latency (haptic response) of the robotic arm control, and ultra-high reliability, which is achieved through dynamic control and Doppler radar.

Showcase 3 adopts a hierarchical orchestration scheme for managing multiple network segments, enabling the deployment of end-to-end network slices for supporting services with diverging service requirements.

Showcase 4 employs several RATs and therefore provides a high dynamic control capability in order to coordinate multiple RATs. In addition, it also provides reduced latency links for remote control.

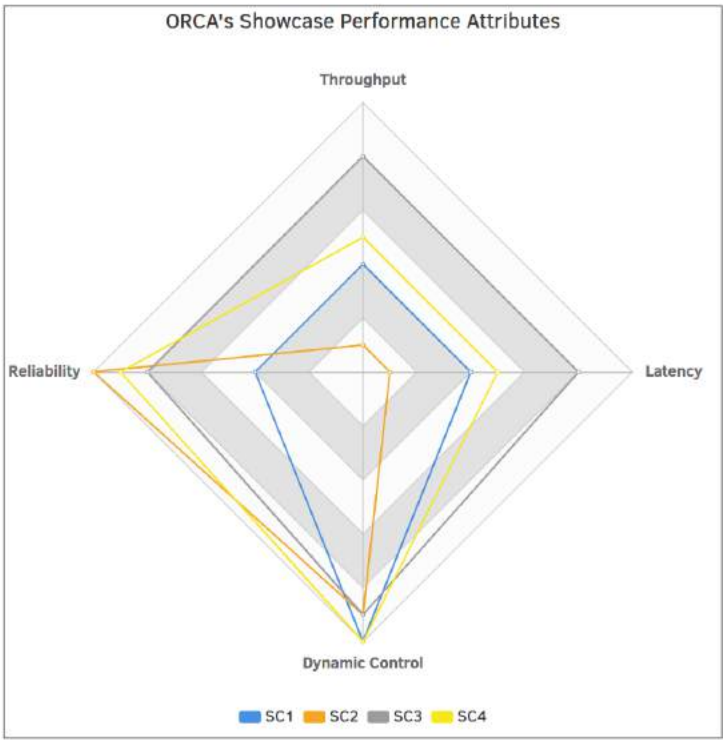

Figure 2 ORCA showcase performance attributes.

The focus areas of ORCA showcases are categorized into four aspects: The *PHY/MAC RAT selection* is the base of common radio functionalities, hence all 4 showcases focus on this area. The *dynamic deployment of intelligent control* is a focus area intended for dedicated control tasks, such as beam-tracking algorithm of showcase 1, channel bandwidth adjustment demonstrated in showcase 2, and network/compute resource optimization in showcase 3. *Hierarchical orchestration* is a global control function, it involves multiple controllers at different levels, such as the hyperstrator, orchestrator and controllers in showcase 3, coordination of sub 6GHz and mmWave as well as parametrization of mmWave beams in showcase 1, control of interworking for multiple RATs in showcase 4, to achieve coherent system level performance. Finally *Hybrid SDR/SDN* is a unique focus area of showcase 3, as it involves network slicing cross network segments, and SDN is utilized for the wired network segments. The focus areas of the showcases are summarized in Table 1.

	PHY/MAC RAT selection	**Dynamic deployment of intelligent control**	**Hierarchical orchestration**	**Hybrid SDR/SDN**
SC1	x	x	x	
SC2	x	x		
SC3	x	x	x	x
SC4	x		x	

Table 1 The focus area of ORCA showcases

2 SHOWCASE 1: 26 GHZ MMWAVE

2.1 Goals

The main goal of this showcase is to demonstrate the real-time beam steering functionality for the 26 GHz mmWave antenna arrays of TUD. As a result of ORCA 3rd year, TUD demonstrates how its mmWave setup can be used as an experimental support for the development of wireless communication systems in an industrial environment. In addition, we also demonstrate the employment of TUD's multi-user system by allowing more than one sub 6 GHz link.

2.2 Challenges

One of the main challenges of this showcase is the real-time beam tracking capability of TUD's solution. First of all, it was necessary to perform the beam steering functionality on the FPGA, in order to guarantee fast beam tracking under the mobility scenario. In addition, it was also necessary to create a control loop with low latency. Moreover, TUD designed the mmWave antenna that can be easily attached to any SDR platform, which facilitated the development of this functionality. The main challenges of this showcase was the real-time beam tracking capability of TUD's solution.

In addition, since we allow more than one sub 6 GHz link we also need to have a multi-user mechanism to coordinate the links.

2.3 Concepts

This showcase emulates a real condition expected to be encountered in an industrial environment. More precisely, we consider a remote-controlled robot moving around the factory hall with a camera attached to it for inspection purposes. The camera is constantly transmitting a live video stream to the AP through the mmWave link. Then, the base station forwarded the video to a factory worker located remotely via a second mmWave link.

Therefore, the mmWave solution is advantageous in two aspects. First, it uses a novel frequency band allowing more capacity of the network. Secondly, the directivity feature of the electromagnetic waves allows a more efficient frequency reuse, in other words, it is possible to use several links close to each other using the same time and frequency with very low interference level. We take advantage of this aspect by employing a second mmWave link as backhaul, where the video is forwarded to the user who is not located in the base station.

In addition, the remote user can steer the robot using a sub 6 GHz link, which makes it possible to thoroughly inspect the production line remotely.

2.4 Results

This showcase demonstrates the fast beam steering algorithm of TUD. Under the mobility scenario, the system is able to successfully track the best beam pair with a maximum delay of 20ms and minimum delay of 2.5 ms when the best TX beam changes by one beam, which is suitable for the video streaming application. In addition, the capacity of the mmWave frequency band is duplicated by reusing a second link with very limited interference.

Additionally, we demonstrate the employment of two sub 6 GHz links operating in TDD mode.

2.5 Innovation

A very relevant innovative aspect of this showcase is related to the 26 GHz antenna arrays, which are easily integrated to the TUD's GFDM implementation for National Instruments' SDRs. Moreover, we demonstrate that a simple beam tracking algorithm can be employed for applications that do not require an ultra-reliable link, e.g., video streaming.

2.6 Demo setup

The demo has two User Devices (UDs), one is mobile and the other is static. The mobile UD is an Automated Guided Vehicle (AGV) and performs a given task in a factory. It sends a video transmission to the AP through the mmWave uplink, this video is then forwarded to the static UD using the backhaul mmWave link, for a video inspection application.

In the uplink between the static UD and the AP, the remote user sends data to steer the robot. This data is then forwarded from the AP do the mobile UD in the sub 6 GHz downlink. In addition, this link is also used to send control information about the beam steering algorithm to the mmWave transmitter. The setup is depicted in Figure 3.

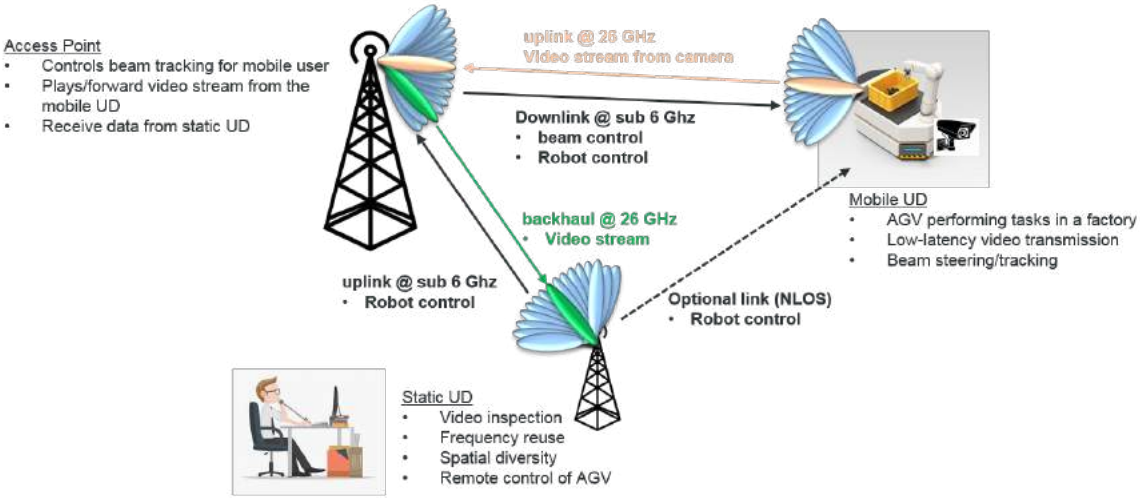

Figure 3 Showcase 1 setup.

2.7 Impact

This showcase demonstrates the mmWave transmission for industrial communication using commercial SDR platforms, which opens a variety of possible extensions for future work, including the experimentation of implementations for the mmWave as well as test of applications actual hardware and over the air transmission.

2.8 Technical developments

To achieve this showcase, TUD developed the 26 GHz mmWave frontends, which are integrated to the TUD GFDM PHY implementation, that provides a low-latency air interface. In addition, the real-time beam tracking was developed in order to cope with mobility condition emulated by the moving robot. In particular, the beam tracking algorithm consists of a constant channel quality measurement at the access point, which changes the transmit or receive beam when the channel quality drops. Finally, TUD has also implemented a TDD based multi-user protocol to allow several sub 6 GHz links.

3 SHOWCASE 2: ZIGBEE NEW RADIO

3.1 Goals

The main goal of this showcase is to demonstrate the low latency, reliable and flexible link control established via SDR. IMEC has implemented multi-channel virtual transceivers, and KUL has integrated Doppler radar with full duplex communication. Both solutions are established upon IEEE 802.15.4 PHY, the former solution aims to achieve better throughput/latency performance as an IoT gateway, whereas the latter aims to increase the reliability of each single control link with full duplex MAC (collision detection and avoidance) and a Doppler radar (environment sensor).

3.2 Challenges

Some challenges are faced during the implementation of this showcase. For instance, the IMEC radio is originally working in TDMA MAC, whereas it is not the case for KUL radio. In order to integrate the two systems, two channels supported by the virtual multi-channel transceiver are set apart for KUL radio. The receivers of multi-channel transceiver implementation share a single radio frontend, which has an Automatic Gain Controller (AGC). However, this AGC applies a global gain setting on all the channels, which causes issues when the transmitters of different channels are located at different distances. In order to overcome this issue, a simple AGC is implemented, it applies gain on each channel individually.

Integration of a configurable Doppler radar system into a communication device, merely by reusing its already-existing waveform and hardware, is also one of the challenges in this showcase. Unlike the traditional radar systems, the Doppler radar in this showcase has to merely use the device's self-transmit signal, while this additional functionality does not affect communication.

3.3 Concepts

The showcase aims to construct a real-life industrial IoT scenario, involving multiple robotic arms and controllers that need to be remotely controlled. In addition, a radar implementation is integrated with KUL's full duplex communication. The radar makes use of signals transmitted for communication to estimate moving objects' speed. This information can help to identify if the target robot is reacting to the command, or if there are external mobile objects present. This feature helps to achieve ultra-high reliability by informing the control unit about the reaction of the devices, or applies the environmental information to implement an explicit response accordingly.

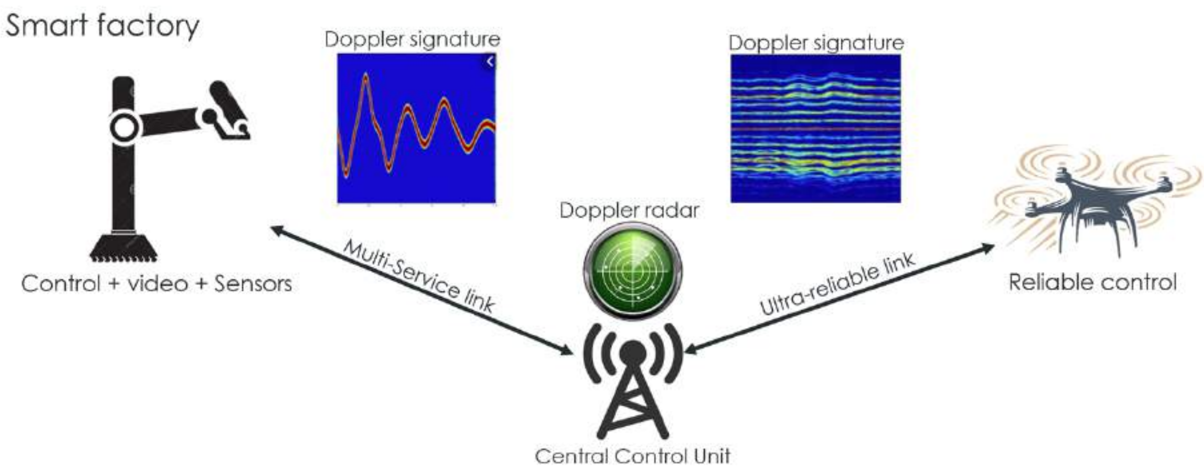

Figure 4 The concept of showcase 2.

3.4 Results

This showcase demonstrates the multi-channel virtual transceiver realized on Zedboard as an IoT gateway, which offers sufficient performance to support control of multiple robotic arms concurrently. In addition, KUL radio demonstrates digital and analog self-interference cancellation and Doppler radar integration with a commercial SDR.

3.5 Innovation

A relevant and innovative aspect of this showcase is related to the construction of multiple virtual radios on the same hardware. The baseband processing modules of the virtual radios are not instantiated multiple times, instead the baseband module runs at a much faster speed, the processing capacity is thus sufficient to support multiple channels with low latency communication requirements. The realized radar-capable in-band full-duplex SDR is also the first of its kind as it performs the environment sensing in a mono-static scheme, where no contribution from another device is required. Besides, it performs this capability opportunistically, without the need for extra power and spectrum.

3.6 Demo setup

The general demo setup is shown in the figure below. A gateway (zedboard) supports a 4x Virtual transceivers. The transceivers operate on 4 channels concurrently. The first 2 channels are used to control 2 robotic arms, the remaining two channels are used by KUL radios, realized on USRP X310. The time and frequency slot used by the system is shown in the upper left corner of Figure 5.

Two additional zedboards are added as wireless interfaces of the robotic arms. It is noted that these zedboards can be replaced by commercial sensor nodes, however it is used here because of the special operating frequency required by KUL radio for full duplex communication. The KUL radio uses microwave signals sent for communication as input of Doppler radar, and it is capable of in-band full-duplex communication. It controls one of the robots through imec-gateway and exploits the communication signal to sense the reaction of the robot opportunistically.

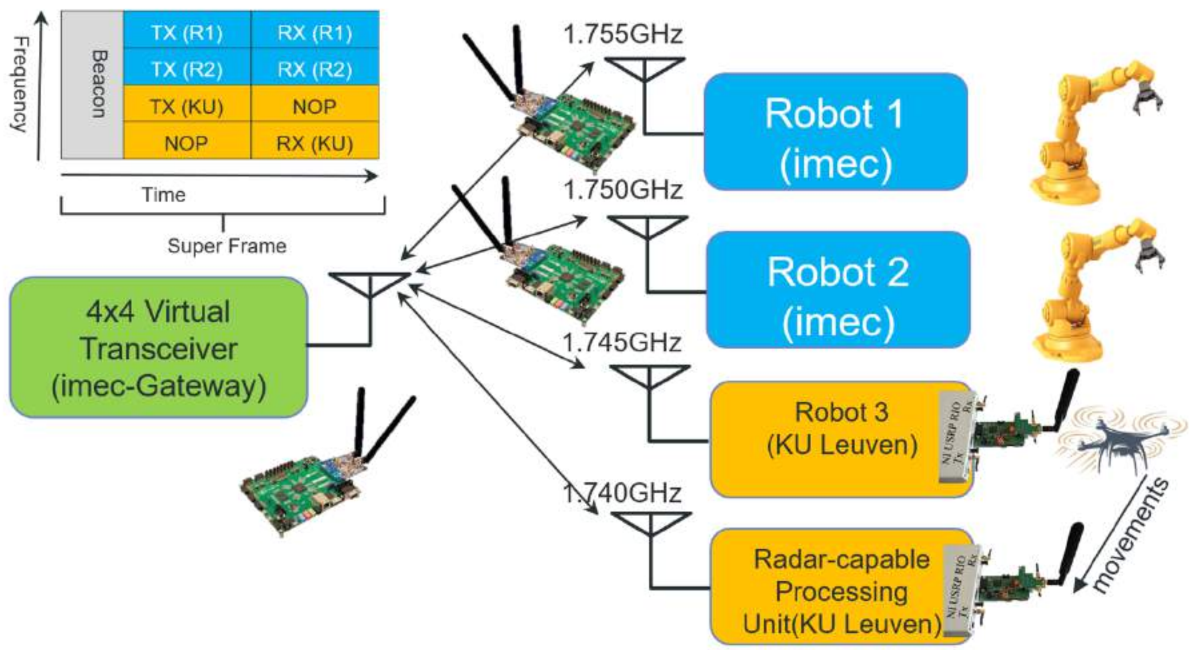

Figure 5 Demo setup of showcase 2

In order to demonstrate the commercial chip compatibility, an additional demo is setup, shown in the left side of Figure 6 below. The gateway is still a zedboard, but the radio nodes close to the industrial entities (sensors/actuators) are replaced by commercial sensor nodes. It is also noted that the gateway can maximum support 8 channels, and it can change the active channels' operating frequency if necessary. At the beginning, the virtual transceiver will continuously monitor the status of Personal Area Network (PAN) 1. Based on the received information from PAN1, the transceiver will turn on/off the critical industrial applications running in different PANs.

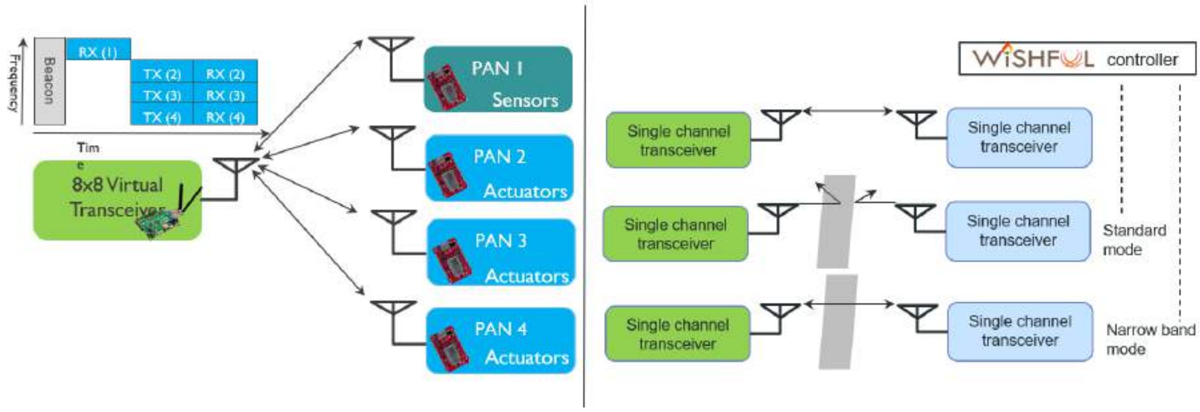

Figure 6 Supplementary demo setup of showcase 2.

In addition, the signal channel transceiver will be used in two different modes: (i) standard mode, which will fail to communicate when a metallic plate is placed in between the transmitter and receiver, and (ii) narrow band mode, which will maintain communication even when the obstacle is present. This setup is shown in right side of Figure 6. The tuning of such kind of parameter is achieved via Wishful controller.

3.7 Impact

The implementation of showcase 2 of previous year is already used by open call experimenters. It is particularly attractive for IoT applications. The fact that SDR solution is compatible with commercial sensor nodes helps to broaden the application of SDR in real-life scenarios. The virtualization of multiple channels is realized in a smart way, to avoid duplication of processing modules. This mechanism has lead to a patent and a new publication. It is also the first demonstration of a reconfigurable radar-communication system in an in-band full-duplex communication context.

3.8 Technical developments

To achieve this showcase, IMEC developed multi-channel virtual transceiver upon IEEE 802.15.4 PHY. The realization of the virtualized transceiver incorporates digital up/down conversion and filter banks, and the over-clocking of the baseband processing chains, and finally a smart mechanism of "context switching" when the baseband processing chain is shared by multiple channels. More details are discussed in D4.5[1] and D3.5 [2]. To perform self-interference rejection, required for both in-band bi-directional communication and radar functionalities, KUL has developed a hybrid tuning algorithm that tunes the digital and analog self-interference modules in such a way that the device achieves optimum communication and radar performance. The implementation is detailed in D4.5 and D5.3.

4 SHOWCASE 3: DISTRIBUTED NETWORK SLICING

4.1 Goals

The main goal of this showcase is to demonstrate the deployment of customised and isolated end-to-end network slices. As a result of ORCA's 3rd year, TCD and IMEC aim to demonstrate in this showcase how their end-to-end network slicing set up can be used to support different types of services with diverging service requirements on top of a shared physical network infrastructure.

4.2 Challenges

One of the main challenges of this showcase is the coordination on the resource allocation among different network segments for deploying end-to-end network slices. First of all, it was necessary to virtualise network segments for creating customised and isolated network segment slices, e.g., a virtual radio access network (RAN), a virtual transport network (TN), and a virtual core network (CN). Then, decomposing the end-to-end network requirements per network segment, allowing the delegation of the resource management to separate specialised orchestrators, tailored for the particularities of each network segment. Finally, achieving a cohesive resource allocation across multiple network segment slices to ensure a consistent end-to-end QoS for the network slices.

4.3 Concepts

This showcase emulates a real network infrastructure that can be encountered in mobile network deployments. More precisely, we consider a scenario whereby the network provider (NP) can use its physical network infrastructure to offer network slices as a service (NSaaS) — in other words, creating network slices on the fly to support different types of communication services and serve service providers (SP). First, we instantiate network slices as a service, reacting to requests from SPs, which contain high-level end-to-end service requirements, e.g., throughput, delay, reliability. Then, our high-level orchestrator, the hyperstrator, maps the high-level end-to-end requirements onto high-level local requirements for the separate network segments, enabling the decentralisation of the decision over the resource allocation and function placement to specialised orchestrators in charge of specific network segments. This way, we simplify the design and development of the individual orchestrators, while allowing most effective resource management directives in each network segment. Moreover, our hyperstrator coordinates the deployment of network segment slices and ensures a cohesive and optimised performance across networks segments to guarantee a consistent end-to-end QoS for fulfilling the given service request.

4.4 Results

This showcase demonstrates the deployment of customised and isolated end-to-end network slices to support communication services with diverging service requirements. We can successfully decouple the resource management per network segment and delegate the resource allocation and function placement to separate specialised orchestrators while ensuring a cohesive performance across network segments and consistent end-to-end QoS.

4.5 Innovation

A relevant and innovative aspect of this showcase is related to the joint orchestration of radio, transport and compute resources for deploying end-to-end communications services. Moreover, we demonstrate how we can apply different models, abstractions and paradigms in each network segment and yet, deliver a consistent end-to-end QoS.

4.6 Demo setup

The demo consists of an experimental end-to-end network infrastructure comprised of three networks segments: a CN, a TN, and a RAN. We use this network infrastructure for supporting three distinct types of services: best-effort remote file storage for notebooks, high-throughput video streaming for handhelds, and low-latency remote vehicle control for autonomous cars; each assigned to a specific end-to-end network slice. The demo setup is depicted in Figure 7.

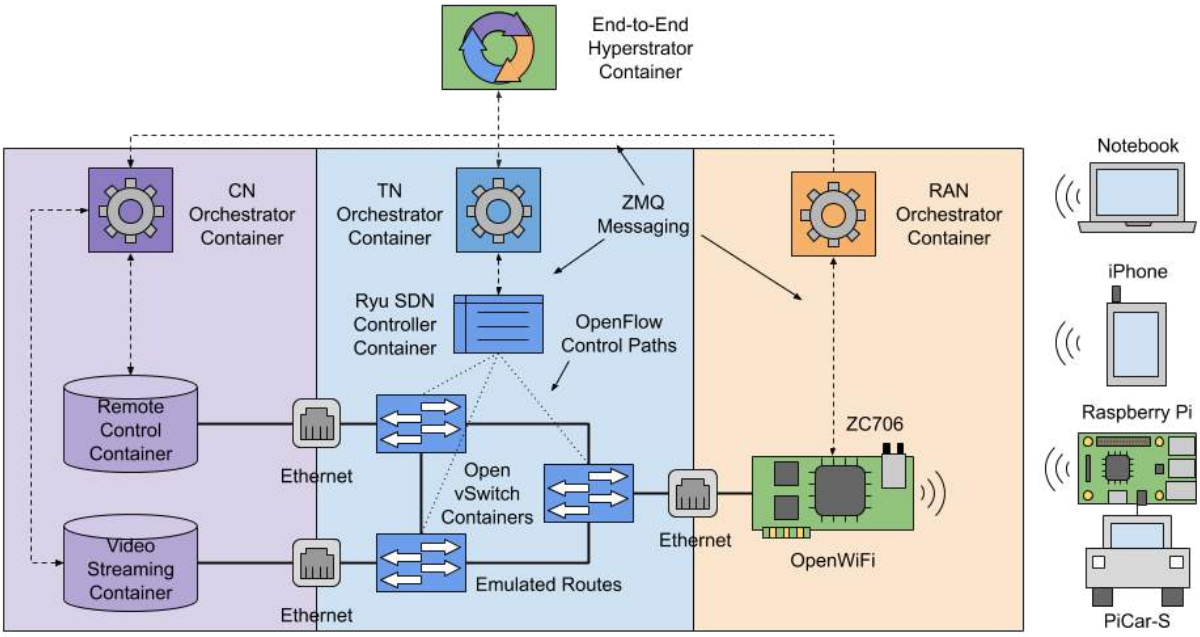

Figure 7: Showcase 3 experimental end-to-end network setup.

4.7 Impact

This showcase demonstrates the coordination across multiple network segments for the deployment of customised and isolated end-to-end network slices, which opens a variety of possible extensions and future works, including modelling and analysis of traffic patterns to leverage statistical multiplexing on the allocation of heterogeneous resources to create network slices. In order to virtualise the RAN segment and create isolated slices that support commercial off-the-shelf devices, we developed a full-stack standard-compliant WiFi radio hypervisor on an FPGA and embedded ARM platform, which resulted in an open-source project called OpenWiFi, which is now publicly available for the research community (https://github.com/open-sdr/openwifi).

4.8 Technical developments

To realise this showcase, TCD and IMEC developed an experimental end-to-end network infrastructure, composed of separate network segments, i.e., a RAN, a TN and a CN, each of which can be independently orchestrated, sliced and combined for creating network slices. In addition, we developed a hierarchical orchestration plane, consisting of specialised orchestrators tailored for each network segment, as well as a higher-level orchestrator, known as the hyperstrator, in charge of coordinating the underlying specialised orchestrators and ensuring a cohesive performance across network segments. More information about the technical developments of Showcase 3 can be found in D4.5 [1], where we detail the development of the experimental setup, and evaluate the network slice provisioning overhead, as well as the performance of slices in the wireless domain.

5 SHOWCASE 4: MULTI-RAT INTERWORKING AND CONTROL

5.1 Goals

With ORCA Showcase 4 (SC4) the concepts for interworking and aggregation of multiple radio access technologies (RAT) will be demonstrated by leveraging the real-time Multi-RAT platform developed within ORCA project by NI, see D3.5 and D4.5 [1,2]. An industry 4.0 application on top of this platform will present a typical scenario for wirelessly connected automated guided vehicles (AGV) integrated into the ORCA factory of the future as main goal for this showcase, see chapter 1. The automated, centralized controlling of those AGVs is abstracted and simplified by using a game controller for steering the vehicle. The abstracted demo use case is shown in Figure 8.

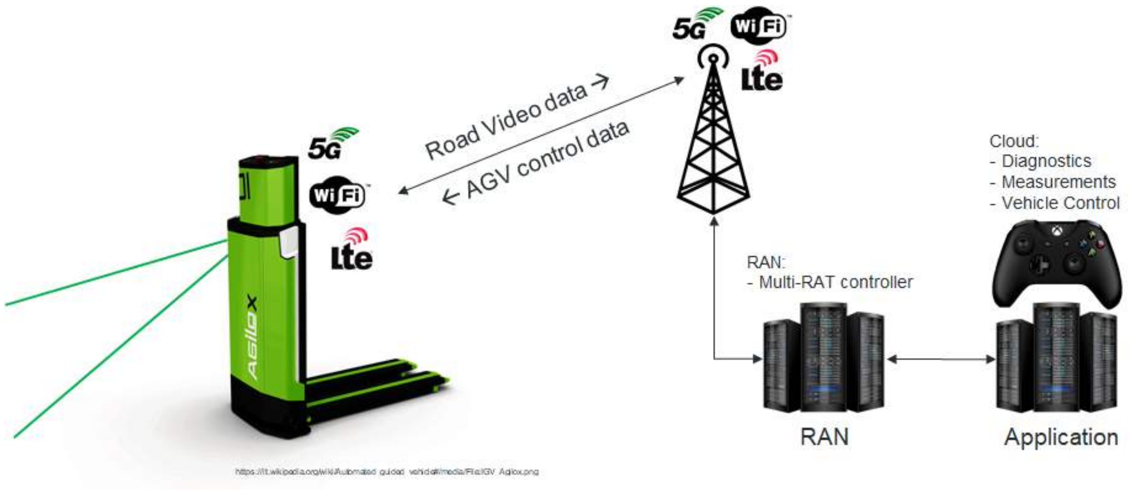

Figure 8: Showcase 4 demo scenario.

5.2 Challenges

Determinism and real-time behaviour are the keys for a reliable wireless system and the main development challenge. NI PXI or USRP 2974 real-time controller hardware with NI Linux RT operating system allows optimization of process scheduling to meet real-time requirements of the higher layer protocols which are represented by ns-3 modules for LTE and WIFI. Those modules where modified by NI to support high performance real-time execution. 5G higher layer stacks where not fully available by the end of this project, see D3.5 [2]. That's why NI integrated an adapted LTE protocol stack towards the 5G flexible numerology physical layer (PHY). The PHY processing for the three different RATs is implemented on FPGA-based NI USRP-RIO SDR. The connection between PHY (on FPGA) and MAC (on CPU) has RAT-dependent throughput and latency requirements. NI L1-L2 API is a way of MAC-PHY interfacing designed with these requirements in mind. For seamless orchestration of RAT technologies, an additional routing application was required to support continuous end-to-end data flow on the application level. Further run-time switching of interworking technologies was made available to support more realistic scenarios.

5.3 Concepts

The main concepts applied for the showcase 4 demo setup are described below with the focus on an industry 4.0 application on top of the Multi-RAT interworking platform.

- Multi-RAT base station and terminal station Software-Defined Radios (SDR) supporting LTE, WIFI and 5G radio access technologies

- RAT interworking technologies such as LTE-WLAN aggregation (LWA) for LTE-WIFI interworking and dual connectivity (DC) for LTE-5G interworking including runtime re-configuration driven by a centralized Multi-RAT controller unit
- All RATs are implemented as full stack solutions supporting end-to-end data transfer
- Variable traffic routing during run-time allows seamless operation on application level
- Robot control application shows capabilities of wireless links in an industry 4.0 environment

5.4 Results

With the end of the ORCA project a full stack Multi-RAT solution running on real-time SDR platform was made available to experimenters through the OWL/TUD testbed [3]. With the robot control application on top, capabilities for an industrial use case were validated. A Multi-RAT controller evaluates link and traffic conditions and allows run-time RAT reconfiguration.

5.5 Innovation

Presented Multi-RAT platform is relevant in scenarios like factory floor automation and communication. In such environments, each machine might come from a different vendor and each vendor might utilize a different RAT. For system critical applications the reliability can be increased by using multiple RATs as redundant links. Aggregating, managing and interworking of RATs is an important topic of ongoing research for the operators of such factories. The innovation of Year 3 is the addition and integration of a flexible 5G link supporting mixed numerology and higher layers described in D3.5 and D4.5 [1,2].

5.6 Demo setup

The full showcase 4 demo setup is visualized in Figure 9. As described above it consists of a Multi-RAT base station and terminal station supporting LTE, WIFI and 5G RATs and their interworking technologies such as LWA/LWIP or DC (blue part). Controllable network gateway applications are used for flexible traffic routing during run-time to and from the robot control application based on decisions which are taken by the Multi-RAT controller (highlighted, orange part).

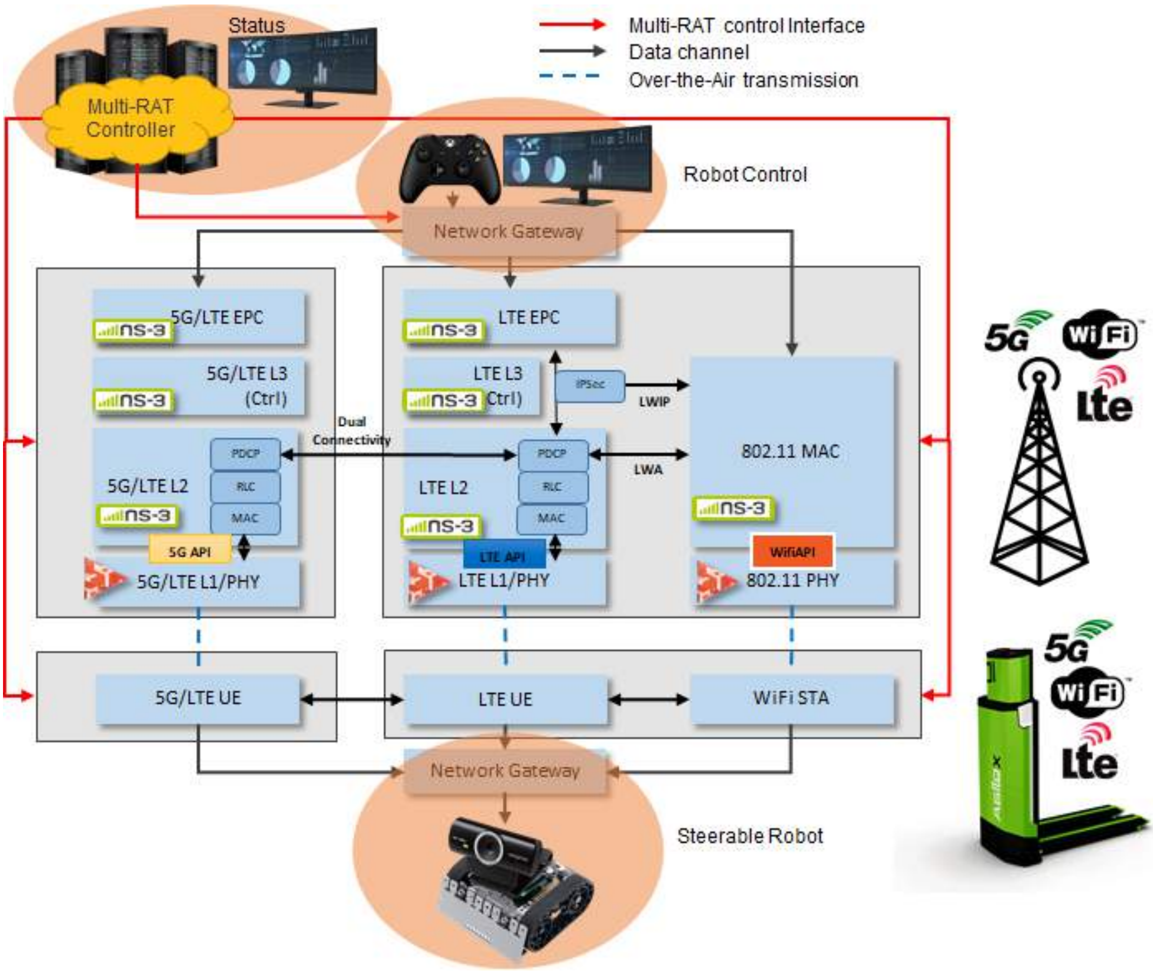

Figure 9 Showcase 4 demo setup

The demo setup focuses on wireless transmission in downlink direction. All three wireless links run in parallel, and LTE can be seen as the master path. Based on commands from a (game) controller the robot control application sends data packets to the network gateway, which forwards the packets to an available wireless link, e.g. LTE. The data will be passed to the complete protocol stack on the base station, transmitted over the air and received by the terminal station. Another network gateway forwards the data from the terminal station to the steerable robot which is the final destination. In parallel the robot provides a video stream from the driving perspective which will be used for steering the robot virtually (as a further enhancement the steering could be done autonomous based on image processing and algorithmic processing of robot control commands in the cloud). The Multi-RAT controller evaluates link and traffic conditions, and achieves RAT run-time reconfiguration by enabling/disabling LWA/LWIP/DC interworking functionality or reconfiguring the network gateways in order to re-route the data to a completely different RAT, e.g. WIFI or 5G. During RAT re-configuration the robot is seamlessly steerable which is the key goal of this showcase and a proof for an industrial application with high reliability constraints. A status display attached to the Multi-RAT controller shows actual run-time parameters. Further manual RAT re-configuration is possible with this graphical interface.

5.7 Impact

Researchers might not have multiple, open and fully modifiable RATs at their disposal. The platform developed in showcase 4 saves time for researchers by providing a head start for RAT interworking experiments across all layers, without the need to invest significant amount of effort in setting up and then integrating the individual PHY links. With this completed platform communication resilience is provided which can be used for experiments on traffic offloading, redundancy data paths and seamless RAT switching. The environment of ns-3 is predestined for simulation but with the interface to real-time SDR implementations, researchers have the possibility to do real experiments as easy as simulations.

5.8 Technical developments

To establish the showcase 4 demo scenario several additions to the Multi-RAT platform described in D3.5 and D4.5 were necessary [1,2]. These technical developments are introduced in the following subsections.

5.8.1 Robot control application

Industry 4.0 augments wireless sensors and connectivity of production equipment, automotive and medical robotics in an industrial scale. These sensors must be monitored at constant rate, which is next to impossible for a large industry, if done manually. Hence, in showcase 4 remote monitoring using UDP video streaming, and using the video stream as input to control the production equipment is applied. In this demo application, an Arduino programmable Pololu robot [4] is used to emulate an industrial production equipment, which is controlled by a gamepad controller remotely. The monitoring is handled from a video stream captured by the robot which is then relayed to a monitoring station (video display). The demo setup is shown in Figure 10.

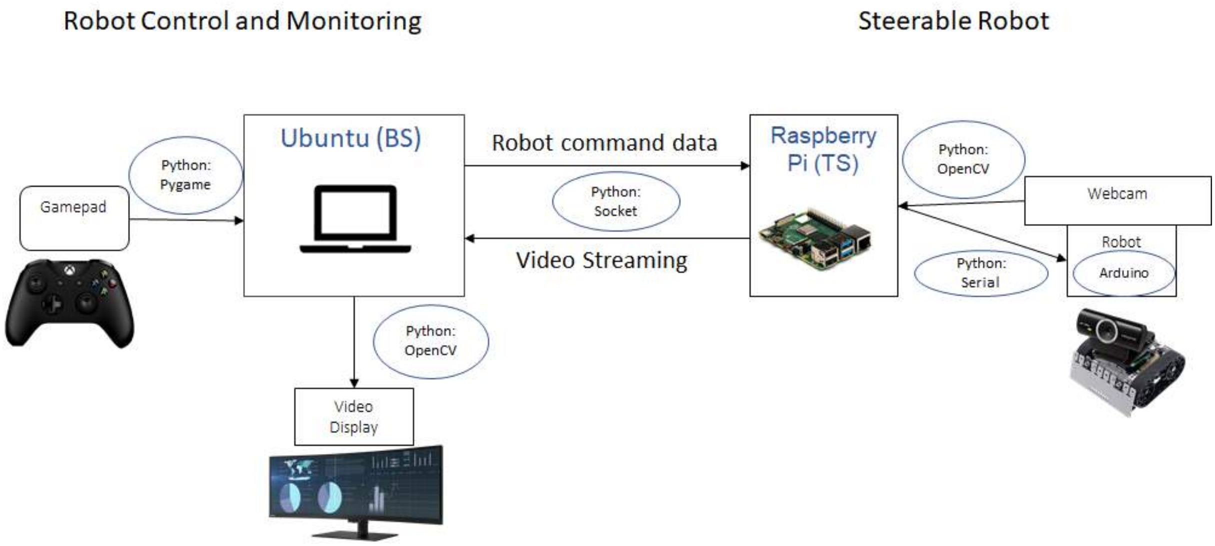

Figure 10 Robot control setup overview

The setup contains a Raspberry Pi with camera module, which streams live video it has captured, to an Ubuntu PC with as low latency as possible. The Ubuntu PC in turn sends control commands to the Raspberry Pi, again with as low latency as possible. For simplicity of explanation, control and video data are transmitted directly over UDP protocol using an ethernet connection. In the final showcase demo the robot command data stream is transmitted over-the-air using the Multi-RAT platform, as shown in Figure 9.

The complete robot control application consists of four software parts leveraging different application programming interfaces (APIs) as listed below:

- Gamepad Interfacing: Pygame API [5]
- Webcam and Video viewing: OpenCV [6]
- Interfacing Zumo robot with Raspberry Pi: serial API

- Ubuntu PC with Raspberry PI: socket API

The Pygame API is a powerful framework for interfacing a gaming controller. The functions are quite easy to understand and well documented. As seen from Figure 11, the steering action can be represented with axis 3 and 4 (4th and 5th column). While 3rd axis signifies steering w.r.t horizontal axis (h), the 4th axis represents steering in vertical axis (v) of cartesian coordinate. Each quadrant of the cartesian coordinate can be easily distinguished because the collective values of vertical and horizontal axis are unique in each of those quadrants.

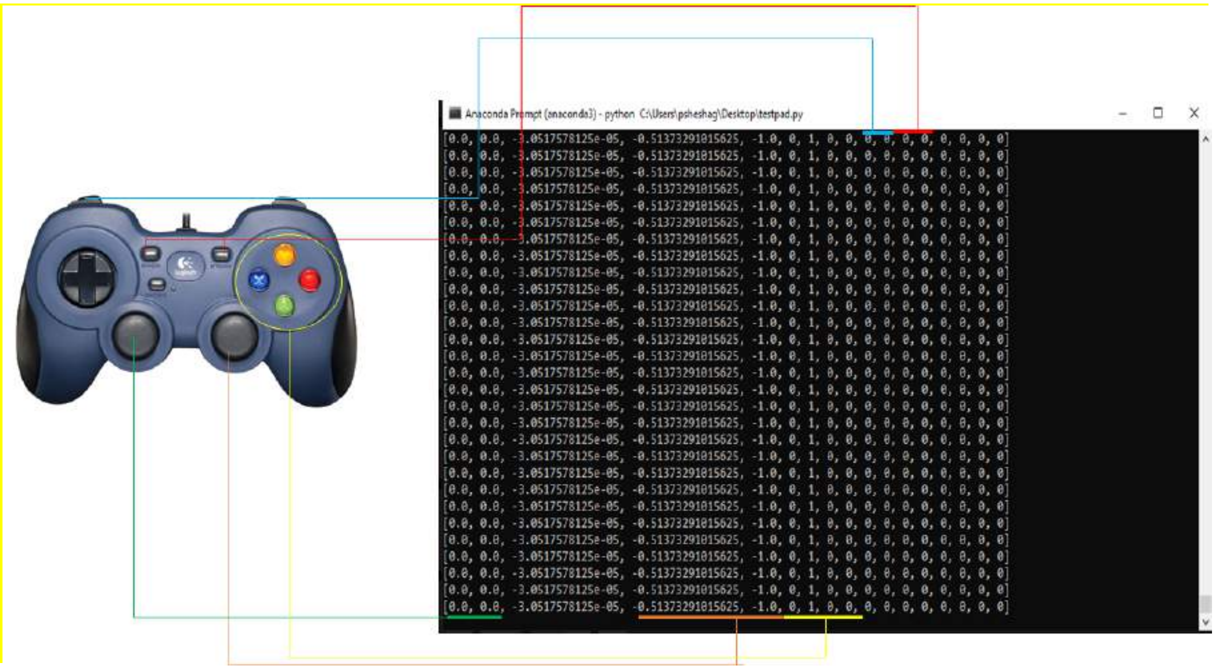

Figure 11: Gamepad controller output

Based on this gamepad controller output the driving angle θ and the speed s are calculated and encoded into a control command string "θ.s". This command string is sent to a server program on the Raspberry Pi through network sockets. Once the Raspberry Pi has received the command, the same command is sent to the robot by using serial API via USB.

To ensure safety operation of the robot, an emergency stop operation was implemented as an interrupt service routine (ISR) with a signal handler. With this a link break or connection loss can be detected very fast in order to stop the robot immediately which is required for a safety operation in a factory environment.

For the video streaming part, the OpenCV framework [6] is used. It's an open source library for image and video processing written in Python and C/C++. It has various APIs and functions for performing image compressions, image analysis and frame encoding. Since the size of each raw frame can be very high (almost 900,000 bytes per frame in SC4), compression techniques are necessary, which are enabled by codecs (enCOder, DECoder). For the application in showcase 4 the JPEG codec is used. The video frames are transferred via the user datagram protocol (UDP). The flowcharts for the client and the server video streaming applications are shown in Figure 12.

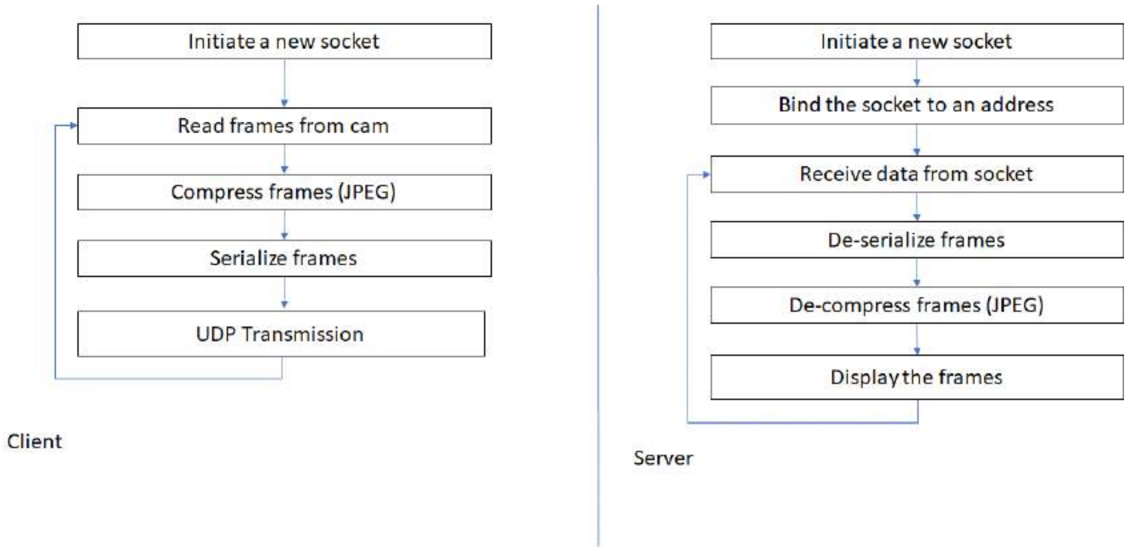

Figure 12 Flowcharts for client and server video streaming applications

5.8.2 Network gateway for variable traffic routing

To achieve dynamic switching between different RATs, a routing application (similar to an OpenFlow-enabled switch) is needed. In the following the routing application design and its implementation is introduced. An overview graph is shown in Figure 13.

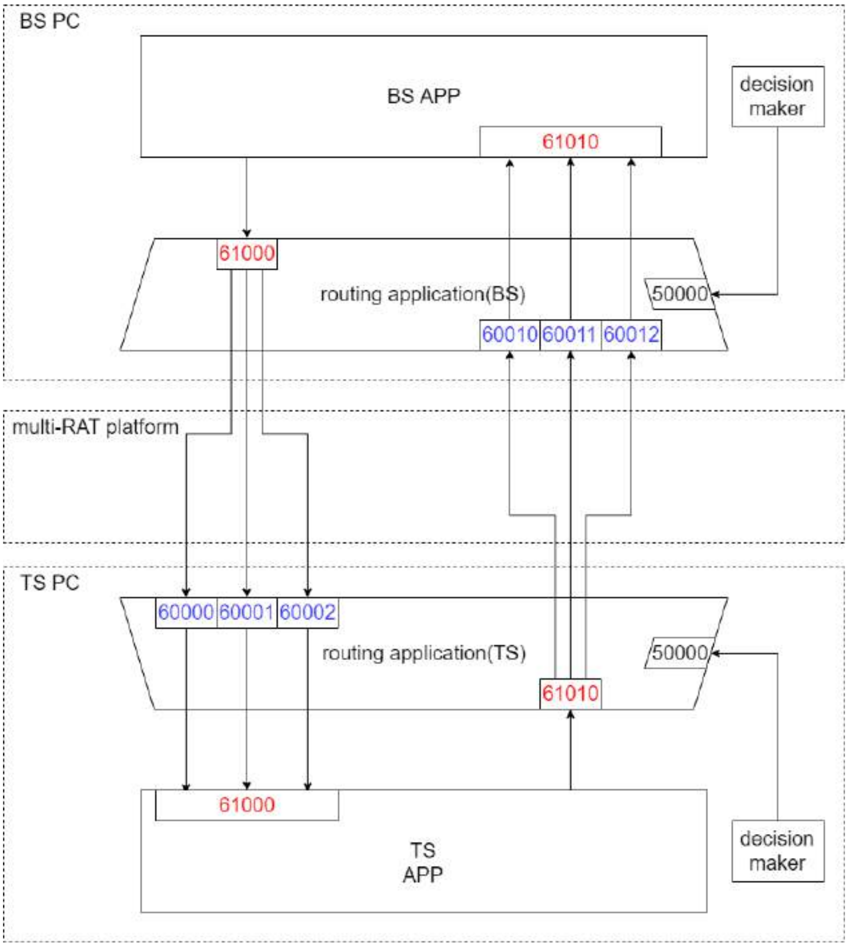

Figure 13 Routing application overview

As shown in the graph above, the routing application has following functionalities:

- Receive data from upper layer application (such as a robot control application) and forward the data to the entry of different RATs of the multi-RAT platform based on the decision.
- Receive data from the multi-RAT platform and forward the data to upper layer application.
- Receive decision from the decision maker and forward the data over the corresponding RAT.

In order to achieve fast data transmission, UDP is chosen to implement the routing application. Multiple UDP sockets are needed to bind or listen to different IP addresses and different ports, as well as forward data to specific ports. Since UDP port number above 50000 is user defined, a specific port numbering scheme is defined to allocate port numbers for the routing application, see Figure 14.

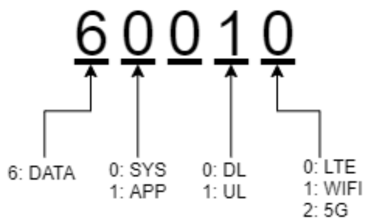

Figure 14 Port numbering scheme

A lookup table in Table 2 shows all the IP addresses and port numbers that are used in this routing application. When the application runs in different modes for base station (BS) or terminal station (TS), the used IP and port configurations are different.

BS		TS	
ipAddr	BS_IP	ipAddr	TS_IP
ip5gAddr	FG_BS_IP	ip5gAddr	FG_TS_IP
listenDataPort	61000	listenDataPort	61010
listenDecisionPort	50000	listenDecisionPort	50000
ltePortTx	60000	ltePortTx	60010
wifiPortTx	60001	wifiPortTx	60011
fgPortTx	60002	fgPortTx	60012
ltePortRx	60010	ltePortRx	60000
wifiPortRx	60011	wifiPortRx	60001
fgPortRx	60012	fgPortRx	60002
rxFwdPort	61010	rxFwdPort	61000

Table 2 Port numbering lookup table

As shown in Figure 13 the decision maker is an instance that decides through which RAT the data should be transferred. The decision maker could be a program (Multi-RAT Controller), a graphical user interface (GUI) or an entity as part of hierarchical orchestration. The decision maker sends a command with a specific format using TUD TestMan protocol [7] to the routing application on port 50000. Thus, the routing application knows which RAT to choose. The possible TestMan commands have been extended to accommodate for this functionality. Two example commands are shown below.

BS0:ROUTING:WRITE:DL_LTE/WIFI/5G_UL_LTE/WIFI/5G

TS0:ROUTING:WRITE:DL_LTE/WIFI/5G_UL_LTE/WIFI/5G

The routing application is implemented in Python. The multi-threading structure of the program is shown in Figure 15 where

- dl data thread: receives data from upper layer application (APP) and forwards the data to the corresponding RAT according to the decision.
- main thread: listens for decision and saves the decision
- ul lte thread: receives data from LTE platform and forwards the data to upper layer APP
- ul wifi thread: receives data from WIFI platform and forwards the data to upper layer APP
- ul 5g thread: receives data from 5G platform and forwards the data to upper layer APP

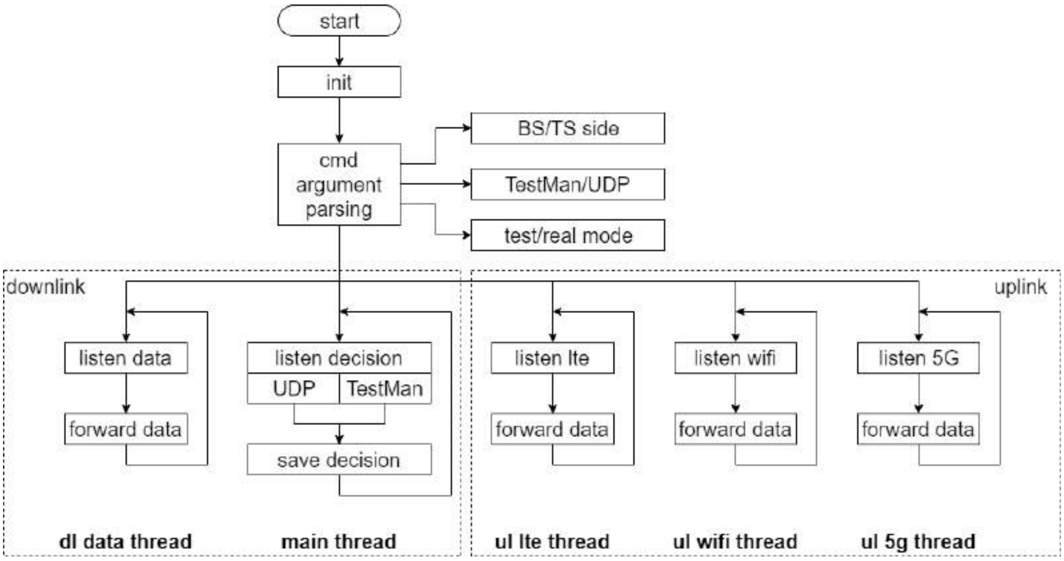

Figure 15 Routing application structure

The application provides 4 command line argument options, which are shown below

```
xiafuning@xiafuning-VirtualBox:~/routing-app-demo$ ./routing-app-demo.py -h
commandline parameters:
argv: []
opts: [('-h', '')]

Usage: [-h --help] [-b --base <number>] [-t --terminal <number>] [--testman] [--testmode]
Options:
        -b --base        run in base station mode                 Default: 0
        -t --terminal    run in terminal station mode             Default: 0
        --testman        enable TestMan server                    Default: False
        --testmode       run in test mode                         Default: False
        -h --help        this help documentation
```

The application can run in base station (BS) or terminal station (TS) mode based on different IP and port configurations. Regarding communication with the decision maker, 2 options are available in this application. The first option is to use a normal UDP socket that listens on port 50000. The second option is to use TestMan, which is a framework that supports flexible inter-program communication. By default, TestMan is disabled and it can be enabled by command line argument shown above. When TestMan is enabled, a TestMan server is established and receives decision commands from other TestMan clients. The integration of both TestMan and UDP interface ensures possible extensions to experimenters needs in the testbed. A test mode is also available for testing and debugging. When test mode is enabled, the application will use an IP and port configuration which is different from the lookup table and the application can be tested on a single machine with two instances running in BS and TS mode separately.

5.8.3 Multi-RAT controller

To enable control plane functionality for the Multi-RAT platform a simple Multi-RAT controller was implemented as described in Figure 16.

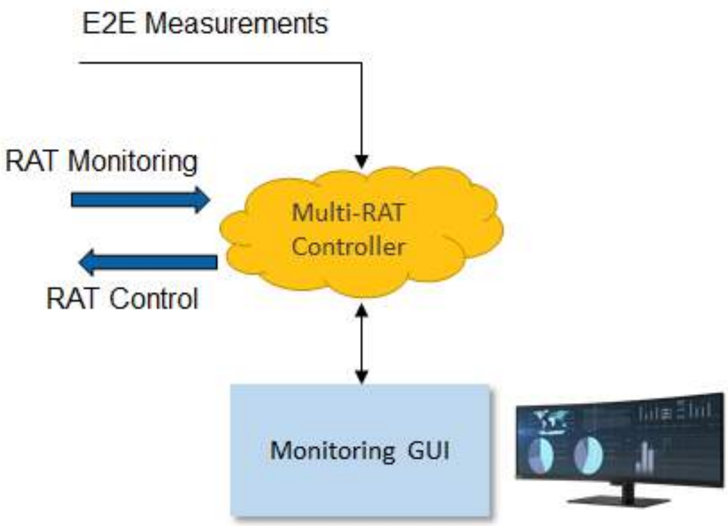

Figure 16 Multi-RAT controller block diagram

The Multi-RAT controller consumes several input parameters such as

- E2E Measurements (e.g. round-trip time)
- RAT Monitoring (e.g. activity status of DC/LWA/LWIP)
- RAT Control (e.g. activation of DC/LWA/LWIP RAT interworking, RAT switching)
- Control input from Monitoring GUI (e.g. manual activation of RAT interworking and RAT switching)

The application outputs status variables to the Monitoring GUI in order to visualize the operation of the whole platform. Based on those in- and outputs a basic control algorithm can be implemented by experimenters to allow dynamic Multi-RAT control which can be seen as a part of hierarchical orchestration of end-to-end networks. With this it is possible to control the different interworking technologies such that different traffic flows can be switched on and off while the whole Multi-RAT setup is running.

6 CONCLUSIONS

This deliverable describes the final ORCA showcases of year 3. In general, the showcases are built in the context of a future factory, which involves low-latency communication for remote robot control, multi-user mmWave with beam tracking to increase the network capacity by frequency reuse, multi-RAT to add flexibility and robustness, and hierarchical orchestration and hybrid SDR/SDN solution to better cope with diverging application requests. These functionalities demonstrate how the ORCA testbeds can be used for future wireless applications, by experimenting novel and flexible full stack implementations.

In particular, showcase 1 demonstrates a multi-user mmWave system with beam tracking, where the targeted application is a remote-controlled robot with video streaming. Showcase 2 brings a low-latency and high reliable control loop for remote control of robots, accompanied by a Doppler radar for increased reliability. Showcase 3 demonstrates the deployment of flexible end-to-end network slices with hybrid SDR/SDN solution, which allows three types of services, namely, best-effort, high throughput and low-latency. Finally, showcase 4 demonstrates a reconfigurable multi-RAT system that employs WiFi, LTE and 5G links, which are used for remote control of a movable robot.

In conclusion, this document described the final set of showcases of ORCA project, which highlighted the achieved SDR functionalities in project, and serves as guidelines for experimenters to use ORCA facilities for future wireless research.

REFERENCES

[1] ORCA Deliverable D4.5, Final operational SDR platforms with end-to-end capabilities, January 2020.
[2] ORCA Deliverable D3.5, Final operational real-time SDR platforms, January 2020
[3] TUD Testbed, http://owl.ifn.et.tu-dresden.de/orca/, December 2019
[4] Pololu Zumo 32U4 Robot User's Guide, https://www.pololu.com/docs/0J63, December 2019
[5] Pygame Documentation, https://www.pygame.org/docs/, December 2019
[6] OpenCV-Python Tutorials, https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html, December 2019
[7] TestMan github repository, https://github.com/vodafone-chair/TestMan.git, December 2019