H2020 5G-Coral Project

Grant No. 761586

D4.2 – 5G-CORAL Proof of concept and future directions

Abstract

This deliverable describes the experimentations used for the integration, verification and validation of the 5G-CORAL solutions. Mainly it explains the demonstrations/Proofs-of-Concept (PoCs) for each use case along with their corresponding performance evaluation results, which validates the feasibility of implementing 5G-CORAL concept for each use case. Also, it presents how the PoCs address the objectives of 5G-CORAL. Finally, the conclusions and future directions are outlined.

Document properties

Document number	D4.2
Document title	5G-CORAL proof of concept and future directions
Document responsible	ADLINK Technology (ADLINK)
Document editor	Ivan Paez, Alan Chen.
Target dissemination level	Public
Status of the document	Stable
Version	1.0

Table of contributors

Partner	Contributors
ITRI	Samer Talat, Ibrahiem Osamah
UC3M	Milan Groshev, Luca Cominardi
IDCC	Giovanni Rigazzi, Charles Turyagyenda
AZCOM	Riccardo Ferrari, Giacomo Parmeggiani
SICS	Saptarshi Hazra, Bengt Ahlgren
EAB	Chenguang Lu, Gyanesh Patra, Daniel Cederholm
TIM	Gian Michele Dell'Aera
NCTU	Chi-Yu Li, Hsu Tung Chien, Chung-Hau Lee
ADLINK	Ivan Paez, Alan Chen
TELCA	Aitor Zabala, Pedro Bermúdez

Production properties

Reviewers	Samer Talat, Chenguang Lu, Antonio de la Oliva, Bengt Ahlgren

Document history

Revision	Date	Issued by	Description
1.0	28 August 2019	ADLINK	D4.2 Ready for publication

Disclaimer

This document has been produced in the context of the 5G-Coral Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement N° H2020-761586.

Table of Contents

List of Figures

List of Tables

List of Acronyms

3GPP 3rd Generation Partnership Project
6LoWPAN IPv6 over Low power Wireless Personal Area Networks
AP Access Point
API Application Programming Interface
APP Application
AR Augmented Reality
ARP Address Resolution Protocol
ATM Asynchronous Transfer Mode
AVRCP Audio/Video Remote Control Profile
BLE Bluetooth Low Energy
CD Computing Devices
CPU Central Processing Unit
D2D Device to Device
DC Data Centre
DDS Data Distribution Service
DHCP Dynamic Host Configuration Protocol
DNS Domain Name System
EFS Edge and Fog computing System
EPC Evolved Packet Core
ePDG Evolved Packet Data Gateway
ESS Extended Service Set
ETSI European Telecommunications Standards Institute
ET End Terminal
GPS Global Positioning System
GUI Graphical User Interface
HTTP HyperText Transfer Protocol
HTTPS HyperText Transfer Protocol Secure
IEEE Institute of Electrical and Electronics Engineers
IETF Internet Engineering Task Force
IoT Internet of Things
KPI Key Performance Indicator
KVM Kernel-based Virtual Machine
L2CAP Logical Link Control and Adaptation Protocol
LAN Local Address Network
LTE Long Term Evolution
LXD Next generation system container manager
MAC Media Access Control
MEC Mobile Edge Computing
MQTT Message Queuing Telemetry Transport
NAT Network Address Translation
NFV Network Function Virtualisation
NS Network Service
NSD Network Service Descriptor
OBU On Board Unit
OCS Orchestration and Control System
OI Organisation Identifier
OS Operating System
OSS Operation Support System
P2P Peer to Peer
PoC Proof of Concept
PHY PHYsical layer
QoS Quality of Service
RAM Random Access Memory
RAN Radio Access Network
RAT Radio Access Technologies
ROM Read-Only Memory
ROS Robot Operating System
RSU Road Side Unit
UE User Equipment
UUID Universally Unique IDentifier
V2I Vehicle to Infrastructure
V2N Vehicle to Network
V2V Vehicle to Vehicle
V2X Vehicle to Anything
VM Virtual Machine
VNF Virtual Network Functions
VR Virtual Reality
WP Work Package

Executive Summary

An ultimate goal for 5G-CORAL is to integrate, validate and demonstrate the technological components developed in WP2 and WP3. This final deliverable of 5G-CORAL Work Package 4 focuses on demonstrations/Proofs-of-Concept (PoCs) and performance evaluation for each use case with the developed components. The PoCs tackle different feature of 5G-CORAL platform as shown in Table 1-1. This deliverable is organized to provide a detailed description and evaluation of each PoC in its corresponding section. In each section, it first provides a description of all PoCs along with the performance metrics and the user benefits. Then, it describes the integrated components and the required 5G-CORAL components with physical and logical architectures details. Next, it describes the measurement results of each PoCs in comparison to the state of the Art.

TABLE 1-1: SUMMARY OF KEY FEATURE PER POC

PoCs	Key features
PoC #1 – Augmented Reality Navigation	Cooperative computing among Fog CDs
PoC #2 – Virtual Reality	Multi-tier Hierarchical computing of Fog/Edge
PoC #3 – Fog-assisted Robotics	Intelligent control following the moving robots
PoC #4 – Multi-RAT IoT	Multi-RAT convergence and virtualization
PoC #5 – SD-WAN	On Demand federation of resources
PoC #6 – High-Speed Train	Fog assisted control unit for the handover and service migration of group of users: The core following group of users
PoC #7 – Connected Cars	Moving Fog CD and distributed computing

Finally yet importantly, it presents the achievements and future directions. The future direction contain several unique features in each PoC. In AR PoC, distributed computing will be deployed in large scale commercial train station with new application. In VR PoC, the streaming engine will be replaced with a more optimized streaming solution to reduce a video stream processing time. Besides, the developed VR solutions will be moved the stitching from the 360 camera to the Edge to reduce 360 stitching processing time with a shorter orchestration set up time. In fog assisted robotics, the integration possibility with different RATs and a remote coordinated control will be considered. In addition, different protocols will be investigated to improve the cycle time of the mobile robots. Also, a federation with SD-WAN will be studied to allow Fog-assisted robotics to federate with other domains. In multi-RAT IoT PoC, the integration of orchestrator to provide new features. Besides, machine learning will be investigated to incorporate techniques, capabilities, and exploiting of the Big Data potential of 5G-CORAL. A large-scale testbed will be considered also in this PoC with more computing resources, more radio heads and more devices. In SD-WAN PoC, machine learning techniques will be explored to enhance the host mobility detection service and predict user movements between APs accurately. Also, an enhancement for the orchestration mechanisms will be studied to place EFS functions, applications and services optimally. In High-Speed train PoC, automatic deployment of the vMME functions can be considered as new feature of orchestrator. In Connected Car PoC, investigation for more efficient encoding schemes to further reduce latency will be considered.

Key achievements

The key achievements in this deliverable are highlighted per PoC as below:

- PoC #1 – Augmented Reality Navigation: This PoC focus on the distributed computing of the limited resource Fog nodes. Distributed computing distributes the incoming job or dispatches it to another fog node (available), if the area is crowded with the users. This allows the utilization of the resources in the EFS, increasing in number of handled requests. In AR PoC, user captures the Point of Interest and send it over the Fog Node for the image recognition, then provides navigation to Points of Interests (POIs). If the Node is overloaded, then distributed computing will guarantee the user experience. This PoC minimize computing latency by processing image recognition and navigation tasks at the EFS. Also it overcome Single Point of Failure (SPF) by centralized distributed computing mechanism.

- PoC #2 – Virtual Reality: This PoC focuses on reducing the E2E latency experienced by the VR user by offloading complexity and processing load from both the server and the terminal. In this respect, a large amount of latency has been reduced from 26 seconds, measured during Year-1 trials, down to 8 seconds. Further improvements are anticipated by replacing the streaming engine with a more optimized streaming solution. In addition, 5G-CORAL platform on top of Fog05 contributed into a significant reduction in the time spent by the orchestrator in setting up and taking down the end-to-end video streaming service. Besides, the VR service decomposition into multiple microservices (EFS apps and functions) increases flexibility and unlocks more deployment options, on top of an agile multi-tier EFS stack deployment managed by Fog05.

- PoC #3 – Fog-assisted Robotics: This PoC focus on network assisted D2D function for robot-to-robot communication and the EFS mobility function. Also, localization EFS Service was integrated to tracks the robots estimated 2D position and make it available over the EFS Service platform. These adopted features allow to extend continues connectivity for the mobile robots. On the other hand, OCS is responsible for lifecycle management (e.g. instantiation, query and termination) of the EFS functions. This PoC decreased the end-to-end reaction time by moving the intelligence in the edge, use of the context information in order to predict misbehaviors, react, and coordinate the robot movements by exchanging data over the D2D channel.

- PoC #4 – Multi-RAT IoT: This PoC focus on multi-RAT convergence and virtualization using software defined radio. Implementations of IEEE 802.15.4, LoRa and NB-IoT stacks have been softwarized and separated into radio head and edge parts, where the latter can be instantiated as needed using edge computing. The components are "dockerized" and integrated in a testbed. IQ/Data wideband samples are made available as an EFS service from the 802.15.4 radio head, which is used by an interference analyzer EFS application. This PoC showcase the feasibility of implementing Multi-RAT IoT stacks in EFS, which indicates better scalability for supporting high IoT connection density. Some of the features are adopted in Contiki-NG open-source project.

- PoC #5 – SD-WAN: This PoC focus on provide a simple PoS application with roaming capabilities. The PoS Terminal roaming across domains, while PoS service is uninterrupted and QoE maintained. This PoC showcase static federation mechanism, host mobility service to locate PoS Terminals across domains and traffic delivery optimization by means of offloading. The resource of SD_WAN were integrated to orchestrator, which has the global view of the whole set of resources, either federated or not. The resource orchestrator is able to place EFS applications more precisely where offloading capabilities are required. This PoC improved latency, bandwidth and jitter by moving applications closer to the user/consumers. Also, moving applications closer to user allows service providers to serve more end users in the same area due to the increase capacity and reduced latency at the Edge/Fog. reducing bottlenecks caused when application traffic moves from users terminal to the cloud.

- PoC #6 – High-Speed Train: This PoC focus on improving on-board user experience by reducing the interaction with on-land base stations. Furthermore, edge networks and virtualization technologies are utilized to bring services closer to the traveling users. Adopting 5G-CORAL platform as EFS and OCS contribute to overcome the signaling storm and backhaul latency challenges to maintain a continuous service. In high-speed train POC, we propose a mobile service continuity solution, which includes a group handover and application migration schemes. Our experimental results show that the proposed schemes can reduce the control signals and migration downtime by 50% and 36%, respectively. This PoC showcase the container-based migration and mobility services.

- PoC #7 – Connected Cars: This POC focus on adopting 5G CORAL architecture to guarantee a lower and more stable latency compared to the legacy-centralized architecture. In this POC, it has been seen that the payload size is reduced by 82% compared to state of Art. In the following Figure 8 11 the comparison of the CDF of the latency measurements done of the three protocols are depicted. In this PoC, Latency is low and stable since EFS app service and function are placed on the OBU and RSU. Also, RSUs can be used to serve small area, reducing congestion and bottleneck i.e. in traffic jam situation. Finaly, E2E latency lower bound is represented by the RAT latency, using more suitable RAT, DSRC or C-V2X or 5G NR to contribute toward 1ms target.

In summary, the aforementioned PoCs contributed in fulfilling the project objectives on utilizing different features of 5G-CORAL of EFS and OCS. Extensive experiments show that 5G-CORAL can improve the user experience in various applications with lower latency, better throughput, stability and more.

1 Introduction

One of the key objectives of 5G-CORAL WP4 is to evaluate the merits of 5G-CORAL solution through a comprehensive set of proof-of-concepts (PoCs) featuring high-throughput and low-latency demanding applications at the vicinity of the end-user and in real-world environments. The goal of WP4 is to validate the technology components developed in WP2 and WP3. Therefore, the 5G-Coral platform has been used to experimentally validate all these components together into real world multifunctional testbeds.

The validation of 5G-Coral components have been done by means of implementing a set of Proof of Concepts (PoCs) based on the use cases defined in WP1 [1].

This deliverable D4.2 is built up from the D4.1 [2] presented at the mid-term of the project. The D4.2 introduces all the seven PoCs that were deployed and tested during the project. It also describes the real-world environments. It highlights the technologies and infrastructures to be validated in each site. More importantly, it presents the progress and corresponding actions performed during the Year 2 of the project, the overcome obstacles and future directions.

The rest of the deliverable is organized as follows:

- Section 2 presents the PoC #1: Augmented Reality Navigation developed by ITRI. This PoC focuses on Augmented Reality (AR) Navigation to provide a continuous indoor AR navigation experience for the users in the shopping mall. The objective is to augment the user recorded video frames with a navigation arrow to its desired destination.
- Section 3 presents the PoC #2: Virtual Reality developed by IDCC. This PoC focuses on a 360° video live streaming service delivered by several 360° cameras located in specific points of interest inside a shopping mall. The main motivations for using this technology can range from offering an ultimate experience to users attending a live event, such as celebrity appearances, contests and sporting events, to helping to relieve overcrowded situations that can occur when a live event attracts a significant number of people in a limited space. In such cases, a 360° video live cast can offer the opportunity for everyone inside a shopping mall to watch the live event in panoramic view and limit the crowd management cost.
- Section 4 presents the PoC #3: Fog-assisted robotics developed by UC3M. This PoC focusses on an enhance robot systems with powerful capabilities by leveraging the computing resources available in the Fog, by placing the intelligence of the robots in the Fog (i) to ensure low latency between the robots and their brains due to the short distance, and (ii) to consume context information on the access network in order to optimize the robotics operations
- Section 5 presents the PoC #4: Multi-RAT IoT developed by Ericsson and RISE/SICS. This PoC investigated and showcased the feasibility to cloudify the communication stacks of multiple IoT RATs in EFS of a 5G-CORAL system. The main benefit is that cloudification of multi-RAT functions makes an IoT access system RAT-agnostic, which can improve significantly the system scalability, flexibility, future-proofness and reduce the costs thanks to the possibility of orchestration and automation.
- Section 6 presents the PoC #5: SD-WAN developed by Telcaria. This PoC focused on demonstrate the concept of **federation** and the benefits that it brings to the users in the Shopping Mall scenario. More specifically, the PoC exposes the mechanisms needed to allow different owners of different network domains to share their available resources (with a previous negotiation and agreement, i.e. static federation) in order to optimize the performance of the network and the Quality of Experience (QoE) of the end users.
- Section 7 presents the PoC #6: High Speed Train developed by ITRI. In moving infrastructures such as this PoC, several challenges are faced due to mobility such as complex signal processing, signal penetration loss, frequent handover (HO) and

terrestrial signal blocking. To overcome these challenges a two-hop architecture solution was adopted to provide better signal quality for on-board users.

- Section 8 presents the PoC #7: Connected Cars developed by AZCOM and TI. This PoC focusses on showing how 5G-CORAL technologies can be applied to meet the ever stringent requirements of the vehicular communication. In this scenario, EFS components can run on the edge and fog cloud. The vehicles are equipped with an On Board Unit (OBU) while the infrastructure consists of Road Side Units (RSUs) which are fog nodes. All these devices can collect speed, direction and position data to help drivers and improve the cars safety.

Every PoC have a subsection that describe the executed planning of the PoCs relevant to the use cases defined in WP1. For each PoC we describe the PoC Testbed design, overview of the integrated components, the software and hardware configuration, the PoC's KPIs and ways to measure performance metrics and final results. Especially the PoCs demonstrated at the European Conference on Networks and Communications (EuCNC) 2019 and at the Midterm review in Taiwan.

Section 9 shows the PoCs relation with the project goal and objectives and finally Section 10 concludes this deliverable with the conclusions and a summary of future directions for each PoC.

2 PoC #1: Augmented Reality Navigation

This PoC focuses on Augmented Reality (AR) Navigation to provide a continuous indoor AR navigation experience for the users in the shopping mall. The objective is to augment the user recorded video frames with a navigation arrow to its desired destination. The user will see a guiding line grounded in the real-world image displayed on his/her screen so that it will resemble a real object, i.e. a pointer, to the desired destination. Moreover, users will be able to see shop promotions on their screen whenever they pass by the store. These special offers will enhance the shopping experience for the mall's client.

To realize this scenario, End Terminal (ET), in our case a smartphone, accesses the WiFi access point (connected to a Fog CD) which allows installation of the AR Navigation application. A video of a shopping mall alley or a landmark (e.g. store logo, mall information signs) is captured by the device's camera. Chosen frames from the video are sent to the Image Recognition (IR) application residing at the WiFi gateway. The IR app performs the analysis of the incoming frames in order to detect location of the ET. At the same time, the AR Navigation app transmits beacon signature data that it registered from nearby iBeacons devices distributed around the shopping mall. This data is in turn fed to the Localization function which estimates the location of the ET in parallel to Image Recognition (IR) app. In order to maximize precision of the localization, two modules (IR app and Localization function) interact with each other exchanging location estimates. This process, in addition to improving location evaluation, aims at decreasing computational burden. Indeed, IR application might decrease the size of the database to explore if rough location of the ET is known. It is the location estimated from the beacon signals that allows the IR app to select relevant portion of the database and hence decrease latency but also improve precision. The precise location is then fed back to the localization function which releases it back to the ET. Once the AR Navigation app receives the location of the user, it displays position and direction on the mall map (as part of the application GUI).

In order to ensure that the IR module operates within given latency constraints under various loads (different numbers of client ETs), the workload can be distributed to multiple fog CDs, not necessarily the closest one. This allows to utilize the resources available within the fog system to increase the number of handled user requests and shorten the latency required to serve the user requests. This distributed computing feature collaborates with IR function that might be deployed on the same Fog CD. An AR image computation request is dispatched in form of tasks among the neighboring DCMs/fog nodes (see Figure 2-1). The dispatching is based on real-time load measurements.

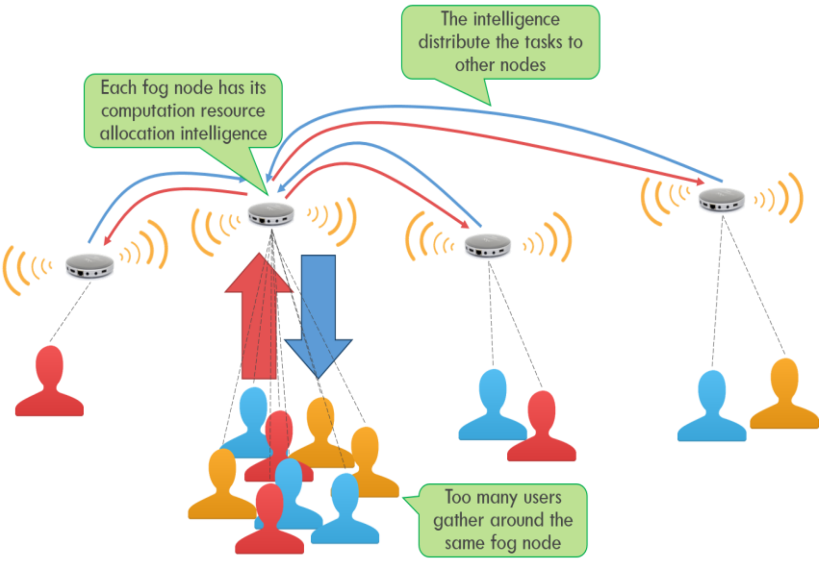


FIGURE 2-1: POC #1 DISTRIBUTED COMPUTING OF AR

The following Table 2-1 introduces the metrics related to user's benefits in the context of the PoC #1.

TABLE 2-1: POC #1 PERFORMANCE METRICS AND USERS BENEFITS

Performance metrics	Description	User Benefits
Latency	Round trip time between issuing a command from an ET and execution from a Fog CD	Minimize computing latency by processing image recognition and navigation tasks at the EFS
Delay	Communication delay between the Fog CDs (for distributed computing)	Increase the capacity of connections by using the distributed computing of the incoming requests. In addition, it overcomes one node failure of centralized distributed computing mechanism.
Computing latency	Time required by a Fog node to complete a job	Increase number of connection by distributing the incoming requests Also, it overcome one node failure of centralized distributed computing mechanism
Localization Precision	Error between true location points and obtain location points	Enhance the user experience by providing services with more accurate proximity.

2.1 PoC Testbed Design

AR navigation system is divided into backend and frontend. On the one hand, the backend mainly constitutes of three components: 1) distributed computing mechanism (DCM), 2) statistics distributor (SD) for data storage and synchronization, and 3) image recognition (IR) server. On the other hand, the frontend consists of a user-friendly augmented reality application. In this PoC, we put a great deal of focus on providing a seamless experience in a multi-user scenario. In order to satisfy the latency constraints under various loads (i.e., different number of simultaneous connected users), the distributed computing mechanism is developed, as part of the backend subsystem, to distribute the computational workload among available fog nodes. This enables the utilization of the available resources within the infrastructure to increase the number of handled users and shorten the latency required to serve each user.

2.1.1 Overview of Integrated Components

AR navigation testbed aims at integrating several technological components which will support the deployment of various experiments to verify and validate the 5G-CORAL solution in the indoor environments. In Y2, the prominent additional feature of the AR navigation PoC is the computational workload distribution capability. The distributed computing mechanism consists of several modules (database, distribution handler, resource allocation optimizer) that work together with the image recognition algorithm towards better user experience. The integration of all these components creates a responsive platform for low-latency and multi-user augmented reality navigation application.

Fundamentally, DCM receives user computation requests. The incoming requests are distributed among the fog nodes in the infrastructure. The distribution is made based on real-time load measurements of the receiving (local) node and the load of neighboring nodes. All the fog nodes in designed distributed environment are aware of the entire system resource status. The system status is synchronized among all the nodes through the centralized statistics distributor. The control plane and data plane of the PoC are detailed in the following logical steps:

Control Plane

Step 1: Setup application connection.

Step 2: Push MySQL commands to SD node.

Step 3: Synchronize the SD node with local database.

Step 4: Read data from local database.

Step 5: Compute the system workload and make distribution decision, then store the result in the local database.

Step 6: Read the distribution result from the local database.

Step 7: Update the SD node with the new result.

Step 8: Setup connection with a selected IR Server (based on the distribution decision).

Data Plane

Step 9: DCM receives image from the user device.

Step 9.1: DCM forwards image to the selected IR server.

Step 9.2: IR server receives the image.

Step 9.3: IR server performs image recognition.

Step 9.4: IR server sends the recognition result back to DCM.

Step 10: DCM sends the result to the user device.

Figure 2-2 shows the testbed design of the AR navigation PoC with distributed computing integration. In the next sections, we map the integrated components to 5G-CORAL building blocks. In addition, we perform measurements for the end-to-end latency and the latency components caused by the control plane as shown in Table 2-5.

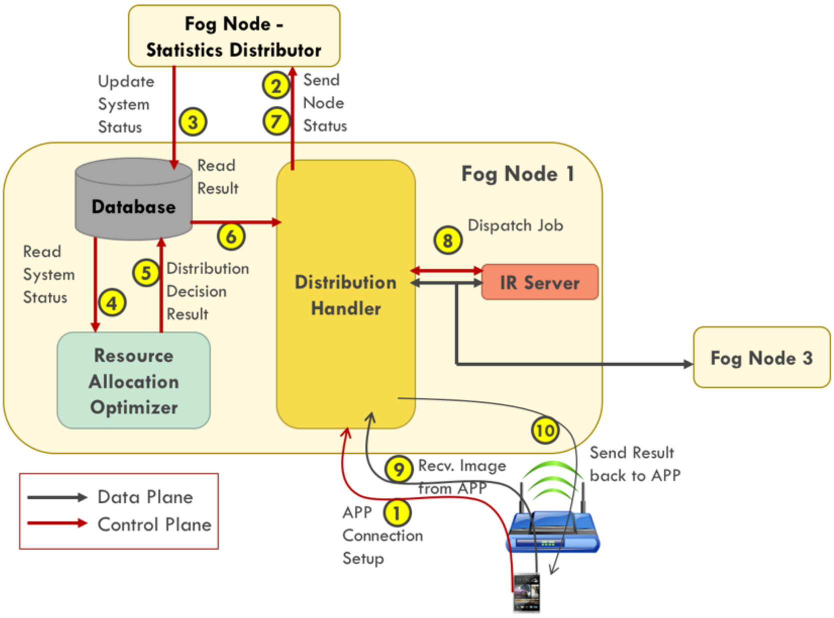


FIGURE 2-2: POC #1 DISTRIBUTED COMPUTING WITH NATIVE AR APPLICATION

2.1.1.1 Hardware Setup and Specifications

The following describes the hardware setup and specifications for both the front and backend of the AR navigation PoC. The hardware specification details can be found in Table 2-2:

- **Fog Node –** Fog nodes are utilized to host the software components including distributed computing modules, image recognition algorithm and the localization modules. Each fog node (i.e., Jetson TX2) supports lightweight virtualization technologies such as LXC/LXD and Docker and connects to a WiFi access point or to another fog node in the infrastructure.
- **Access Point (AP) –** Provides wireless connectivity for the users of the AR navigation application. The access points are connected to the computing infrastructure through Ethernet in which users' tasks are sent for computation.
- **Smartphone –** User devices that are equipped with sufficient computing resources to run the interface of the AR navigation application. Currently, the AR navigation application is developed to run on Android smartphones.
- **iBeacon –** Low-energy Bluetooth devices are deployed in the testbed to detect the presence of users. The deployment of iBeacons helps fog nodes to refine the location of users via transmitting area ID information to the backend.

TABLE 2-2: POC #1 HARDWARE SPECIFICATIONS USED IN THE EXPERIMENTAL SETUP

Hardware Component	Specifications

Fog Node	CPU	Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM® Cortex®-A57 MPCore
	Memory	8GB 128-bit LPDDR4 Memory
	Network	802.11a/b/g/n/ac 2×2 867Mbps WiFi 10/100/1000 BASE-T Ethernet
	GPU	256-core NVIDIA Pascal™
Access Point	Model	TL-WR1043ND v2
	Wireless Adapter	Qualcomm Atheros
	Memory	8MB + (16GB)
	Network	4× Ethernet 2×Wireless
Smartphone	Model	HTC One M9
	CPU	4x1.5 GHz Cortex-A53 & 4x2.0 GHz Cortex-A57
	Memory	3 GB RAM
	Network	HSPA 42.2/21.1 Mbps, LTE-A Cat6 300/50 Mbps 802.11 a/b/g/n/ac, dual-band, Wi-Fi Direct

2.1.1.2 Software Setup and Specifications

The software setup of the PoC includes the operating system and the virtualization software supported in the infrastructure. For the fog nodes, Jetson TX2 does not support LXC/LXD out of the box. Therefore, we run Ubuntu 16.04 with a modified kernel to support virtualization. Among the key modules enabled in the new kernel are Namespaces, Control Groups, FUSE (for use with lxcfs), Checkpoint/Restore support and SQUASH filesystem. As for the access points, we utilized OpenWRT for a fully controlled Linux-based framework. The software specification details can be found in Table 2-3:

TABLE 2-3: POC #1 SOFTWARE SPECIFICATION USED IN THE EXPERIMENTAL SETUP

Software Component		Specifications
Fog Node	OS	Ubuntu 16.04
	Kernel	4.4.38 (modified)
	LXD	3.0.3
Access Point	Framework	OpenWRT (15.05.1)
	Kernel	3.18.23

2.1.2 Required 5G-CORAL Building Blocks

Table 2-4 provides an exhaustive list of 5G-CORAL EFS and OCS building blocks (sub-system) to form a working AR live navigation PoC.

TABLE 2-4: POC #1 REQUIRED 5G-CORAL BUILDING BLOCKS

Sub-system	Component's Name	Description
EFS	Image Recognition	An **Application** to recognize the incoming features from a pre-defined distributed database.
EFS	Location Estimation	A **Function** of iBeacon localization, IMU (Inertial Measurement Unit) and Image Recognition to precisely localize the current position of the user.
EFS	Localization Service	**EFS Service** - The density of visitors supports the system to initiate the demo scenarios and where to operate. The localization service provides significant information for the user to navigate and reach the desired location in the shopping mall.
EFS	Navigation system	**EFS Service** - Navigation system for the robot to navigate through the Shopping Mall
OCS	Instantiation	An **OCS Component** to Instantiate EFS sub-systems from a given catalogue. It also instantiates virtual AP to a Physical AP.
OCS	Monitoring	**OCS Component** reporting monitoring/statistics information (Current Resource usage) of the containers/Neighboring Fog nodes.
Non-EFS	ET AR navigation application	Captures frame of the objects and sends it to the Fog Node in a periodic fashion to navigate itself to the destination.

GPS (Global Positioning System) is a commonly used localization technology nowadays. However, GPS is achieved by satellites, and therefore fails to perform in indoor environments. The drawback motivates this PoC to propose an alternative technology for indoor localization.

Indoor localization services bring several benefits to users. For example, shopping mall can provide location tailored services to the customers. Stores in the shopping mall can push advertisement or coupons to the customers nearby. Furthermore, using indoor localization service we can also navigate customers to their destinations so they won't get lost in the shopping mall.

In order to receive location services timely, a successful indoor technology should be responsive, highly accurate and scalable. We leverage FogCD as our computing node instead of an internet server. In this fog based network architecture, both data storage and data computing are moved to the edge of the network. Localization services can not only decrease its communication delay but computing latency.

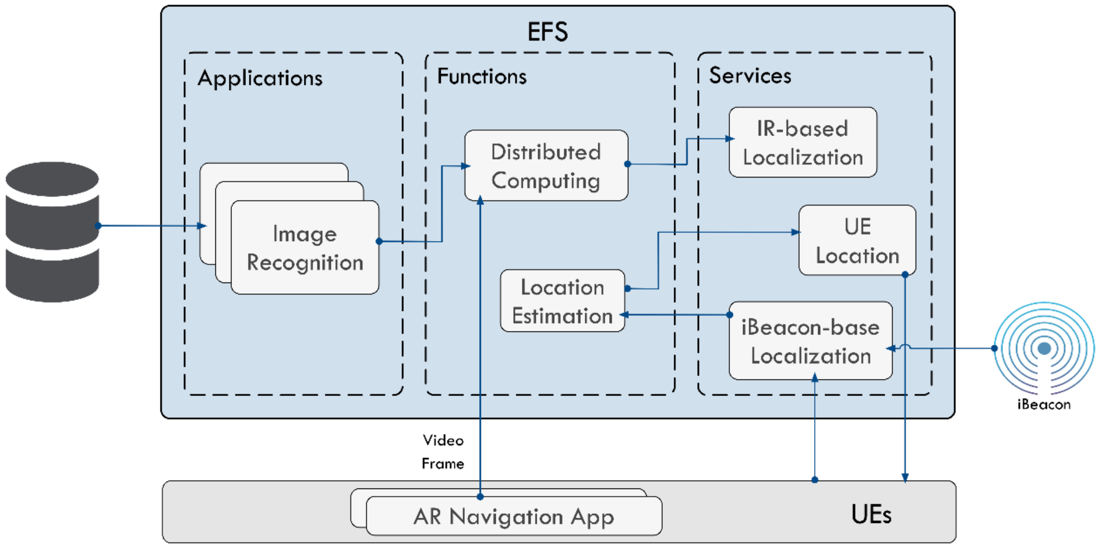


FIGURE 2-3: POC #1 EFS ENTITIES CONFIGURATION FOR AR USE CASE

As depicted in Figure 2-3, this PoC leverages 5G-CORAL EFS to offload the computation tasks on the mobile device to the computing node in the vicinity. In this fog based environment, both data storage and data computing needed by the localization service happens outside of the mobile devices. Localization application on the mobile device are able to leverage fog nodes to perform low latency computing and decrease its energy usage.

2.2 PoC KPIs: Measured Values for the performance metrics

In this section, we explain the experimental setup for AR distributed computing. Then, we present our experimental results for computing latency based on virtualized and Native application. In particular, the total computing latency when number of users increased are highlighted in the following subsections.

2.2.1 Measurement Methodology

The current State-of-the-art for the AR technology such as ARCore released by Google [3] provides AR service on the end-terminal without any external computation support, hence relying on up to 1000 reference images which are stored in a single image database [4]. However, this may not be feasible in big shopping malls scenarios. Many more images may be required to seamlessly support AR based navigation and shop identification simultaneously. By leveraging the concept and the design of distributed computing for AR with segmented reference image database, more computation capacity and reduced communication delay can be provided by the AR Navigation application. With all the aforementioned facts, the objective is to augment the user recorded video frames with a navigation arrow similar to the popular car navigation application. The user will see a guiding line grounded in the real world image displayed on his/her screen so that it will resemble a real object, a pointer, to the desired destination as depicted in Figure 2-4. We ran this experiment for 30 repetitions with standard deviation of 21 with Jetson TX2 as the Fog node which have distributed computing capabilities. Then, the computing latency for AR Navigation is calculated for LXD-based and Native AR application. The computing latency is defined as the time taken by the Fog node to process the image recognition process for the AR application.

FIGURE 2-4: POC #1 AR LIVE NAVIGATION IN SHOPPING MALL

2.2.2 Measurement Results

Table 2-5 shows the performance metrics related to the AR navigation. The tabulated metrics includes the reference measured values from the state-of-the-art and the measured values from the AR Navigation infrastructure. The main performance metrics in this use case are end-to-end latency, inter-fog nodes latency, computing latency and localization precision.

TABLE 2-5: POC #1 MEASURED VALUES FOR THE PERFORMANCE METRICS RELATED TO AR NAVIGATION.

Performance metrics	Description	Reference Values	Measured Results
End-to-end Latency	Round Trip Time (RTT) between issuing a command from an end-terminal and execution at the Fog CD	• 500 *ms* between the end-terminal and the Fog CD [5]. • 850 *ms* between the end-terminal and the Cloud [5].	Average of 122 *ms* between the Fog CD and the end-terminal.
Inter-Fog Nodes Latency	Communication delay between the Fog CDs (for distributed computing)	Latency of 11.8 *s* for a 521 MB file [6].	Latency of 7s for 521MB file
Computing Latency	Time required by a Fog node to complete a job	30 ms at the Cloud and 200 ms at the end-terminal [7].	85 ms to 150 ms to compute a job at Fog CD
Localization Precision	Error between true location points and computed location points	2 to 3 meters via TDOA [8] and triangulation [9]	Average error of 0.56 meters.

The network latency from issuing a job at the end-terminal and the execution of the job at the Fog CD is 122 *ms* on average. Besides, we measured the computing latency which is the time required by the Fog CD to process a frame (i.e., feature extraction and database matching) and return the results. This latency value could vary depending on the job distribution. In the case of local processing (on the Fog CD that receives the recognition job), the measured latency is 85 ms while in the case of distributed processing (job dispatched to a neighbouring Fog CD), the recorded latency is 150 *ms*. The end-to-end latency is the sum of the network latency and the computing latency, which are shown in Figure 2-5 and Figure 2-6 for different deployments. Finally, the localization precision of this PoC has been achieved with an average error rate of 0.56 meters.

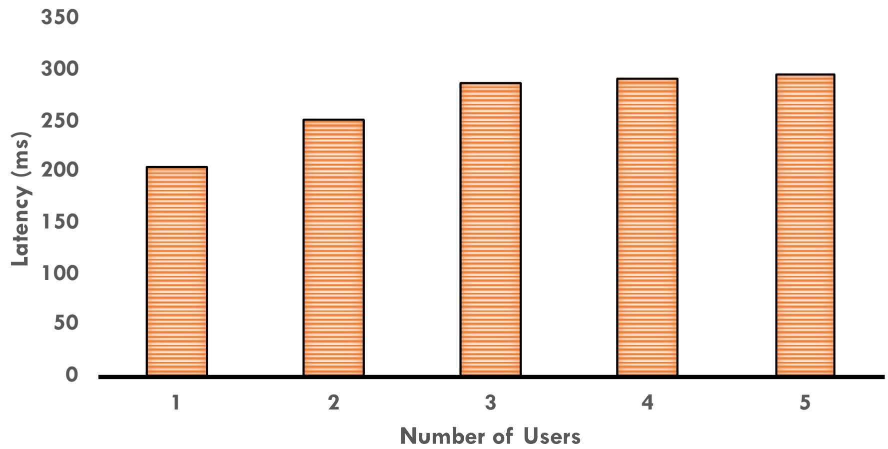


FIGURE 2-5: POC #1 END-TO-END LATENCY OF AR NAVIGATION AS A NATIVE APPLICATION

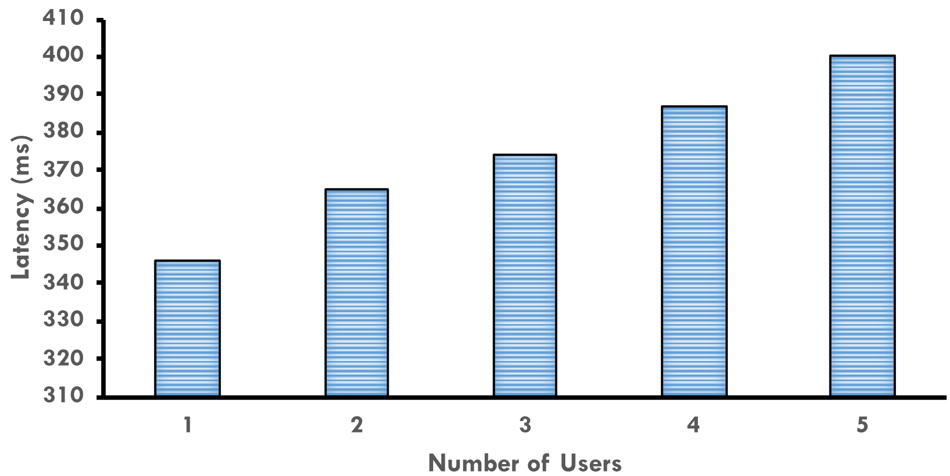


FIGURE 2-6: POC #1 END-TO-END LATENCY OF AR NAVIGATION AS AN LXD CONTAINERIZED APPLICATION

Figure 2-5 and Figure 2-6 show the experimental results for end-to-end latency for different number of users and for native and containerized AR application, respectively. This latency presented in the figures includes end-to-end communication latency, inter-fog latency and computing latency. It is important to note that containerization brings flexibility in deployment and infrastructure management but the virtualization overhead increases the experienced latency by 25% when compared to the native deployment.

2.3 Year 2 Achievements and Future Directions

Over the years, researchers have proposed several indoor localization technologies such as infrared ray, ultrasonic, Millimeter Wave, visible light and Wifi. Although technologies like infrared ray and ultrasonic may achieve high accuracy in short distance, they consume more energy than other technologies. Since indoor positioning system is based on mobile device, low energy consumption is one of the main concerns. Indoor localization should also be based on common technologies. Technologies like Millimeter Wave and visible light may require proprietary hardware. Furthermore, shopping mall is a relatively noisy environment, Wifi based technologies are prone to interference.

Initially, in this PoC we devised a high availability and highly compatible indoor localization mechanism which can achieve sub-meter accuracy in general mobile devices. We chose iBeacon technology for its low interference, energy efficient and low cost. We apply particle-filtering approach to counteract the random noise in the environment. Due to the fact that the users may be carrying diverse devices, we design our proximity calculation algorithm by the relative readings from the iBeacon base station.

In Second year, the main focus is the distributed computing of the limited resource Fog nodes. Distributed computing distributes the incoming job or dispatches it to another fog node (available), if the area is crowded with the users. This allows the utilization of the resources in the EFS, increasing in number of handled requests. In AR with distributed computing, user captures the Point of Interest and send it over the Fog Node for the image recognition, then provides navigation to Points of Interests (POIs). If the Node is overloaded, then distributed Computing will guarantee the User experience. The distributed computing functionality can be adopted in large scale and different application. We aim towards the exploitation of this technology in Taipei main station.

2.3.1 Demonstration activities

Throughout the project lifetime, we carried out a number of demonstrations and exhibitions at relevant venues in Europe and Taiwan. In the following, we describe the main events where we showcased our AR PoC.

- **First 5G-CORAL trial (2018):** In November 2018, Figure 2-7, it was completed the first 5G-CORAL trial at Global Mall Nangang Station store in Taipei, Taiwan. Based on 5G-CORAL platform, AR PoC featured the continuous indoor AR navigation experience for the clients in the Shopping Mall. In this demo, the objective was to augment the user recorded video frames with a navigation arrow. The user senses a guiding line grounded in the real-world image displayed on his screen so that it reminds a real object, i.e. a pointer, to the desired destination.

FIGURE 2-7: POC #1 AR DEMO SHOWCASE DURING TAIWAN TRIALS

- **COMPUTEX 2019:** the AR PoC was showcased in June 2019 during the COMPUTEX in Taipei, Taiwan, Figure 2-8. It was outlined the uniqueness of distributed computing AR Navigation on top of the 5G-CORAL platform as follow
    - **Computation Offloading –** Offloads the computation tasks of the end-devices/central cloud to the network edge
    - **Multi-RAT Localization –** Locates the user using deployed iBeacon and edge image recognition.
    - **Distributed Computing–** Utilizes the distributed nature of the Fog node to offload the load among Fog nodes to facilitate users without affecting the QoS.

FIGURE 2-8: POC #1 DEMO EXHIBITION DURING COMPUTEX IN TAIPEI

3 PoC #2: Virtual Reality

The Virtual Reality (VR) PoC aims at showcasing the benefits of a 360° video live streaming service delivered by several 360° cameras located in specific points of interest inside a shopping mall. The main motivations for using this technology can range from offering an ultimate experience to users attending a live event, such as celebrity appearances, contests and sporting events, to helping to relieve overcrowded situations that can occur when a live event attracts a significant number of people in a limited space. In such cases, a 360° video live cast can offer the opportunity for everyone inside a shopping mall to watch the live event in panoramic view and limit the crowd management cost. Table 3-1gives an overview of the performance metrics assessed in this PoC and describes the main benefits for the end users.

TABLE 3-1: POC #2 PERFORMANCE METRICS AND USER BENEFITS

Performance metrics	Description	User Benefits
Bandwidth	Viewport adaptive streaming approach reduces the bandwidth required to stream 360-degree video while maintaining the same viewing experience	Reduced probability of experiencing video quality degradation (blurry or jumpy picture).
Service Setup & Takedown Time	Service setup time refers to the time it takes for the OCS to instantiate and provision all the VR components, distributed across the EFS, to produce an end-to-end VR service	Quick service instantiation and lower waiting time.
End-to-end Delay	End-to-end application delay among: Fog CDs, Edge server, Cloud Data Center (Cloud DC) and terminal clients.	Lower out-of-sync between the live event and the video delivery
Power consumption	Power consumed by the Graphic Processing Unit (GPU) on the Cloud DC when all the tasks are executed on the same machine or when offloading is employed.	N/A for end user, it may imply benefits for the provider, greener environment and reduced cost.

3.1 PoC Testbed Design

It is important to highlight that 360° video delivery implies high bandwidth consumption and low latency requirements which are hard to achieve in conventional wireless networks. One of the solutions to reduce the bandwidth consumption consists of employing a viewport adaptive streaming technology. This technique relies on the clients' viewing orientation and aims to deliver the portion of the 360° video (e.g., viewport) being watched by the user in high quality/resolution, whereas the rest is delivered in low quality/resolution [10]. As shown in Figure 3-1, three 120° tiles are generated from a panoramic scene and encoded in two different quality level, i.e., high and low quality, while the live streaming server is responsible for choosing the appropriate tile quality combination based on the user orientation. Furthermore, viewport adaptive streaming is currently supported by the latest MPEG VR standard, Omnidirectional Application Format (OMAF), and MPEG video streaming standard Dynamic Adaptive Streaming over HTTP (DASH). Yet, tile-based High Efficiency Video Coding (HEVC) transcoding/encoding adds extra computational complexity to the system, as tiled video streams have to be decoded and re-composed into 360° video frame at the client side, which adds computation into the devices and may not be supported by legacy devices. An illustration of the building blocks employed at the client side is presented in Figure 3-2. A key role in this system is played by the view orientation tracking module, which collects the client orientation view and notifies the DASH server of any change in order to adapt the viewport accordingly. Finally, the HEVC decoder module recomposes the panoramic view and the VR rendering module delivers the 360° video.

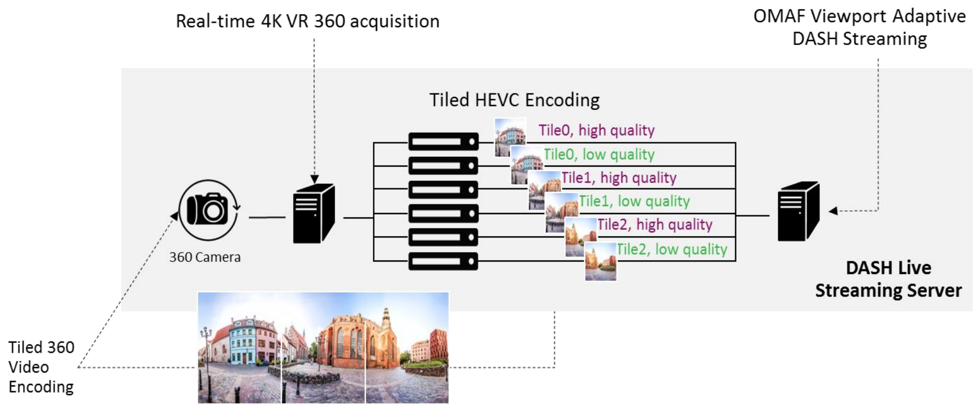


FIGURE 3-1: POC #2 ARCHITECTURE OF THE 360° VIDEO DELIVERY (SERVER SIDE)

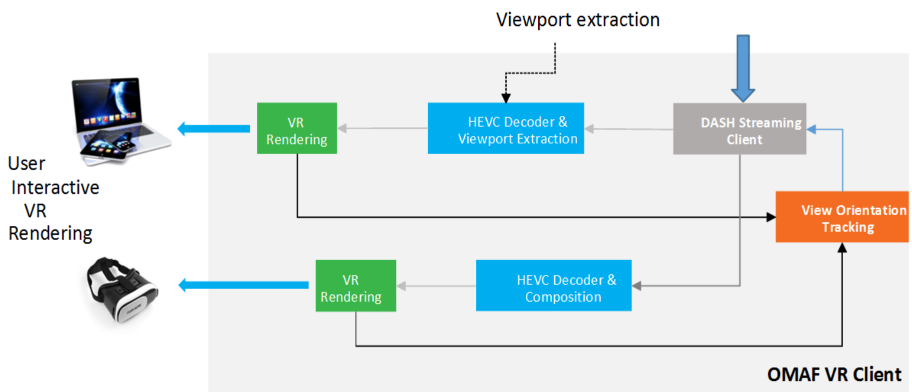


FIGURE 3-2: POC #2 ARCHITECTURE OF THE 360° VIDEO DELIVERY (CLIENT SIDE)

3.1.1 Overview of Integrated Components

Our testbed consists of a multi-tier computing, storage and networking platform capable of conveying multimedia traffic generated by one or more Insta360 Pro cameras to fixed and mobile clients. We assume that each component is equipped with a Gigabit Ethernet network adapter and is connected to the network layer represented by a Gigabit Ethernet switch, whereas the phone terminal is connected to a Wi-Fi IEEE 802.11ac access point. For the sake of clarity, the tasks performed by each layer are listed at the bottom of the diagram. The top layer hosts the orchestration component, which deploys and manages all the entities running the Fog05 agent [11]. This is achieved by using a laptop or any computer equipped with a screen and running the Fog05 agent, in order for the operator to execute scripts and verify the successful function onboarding, see Figure 3-3.

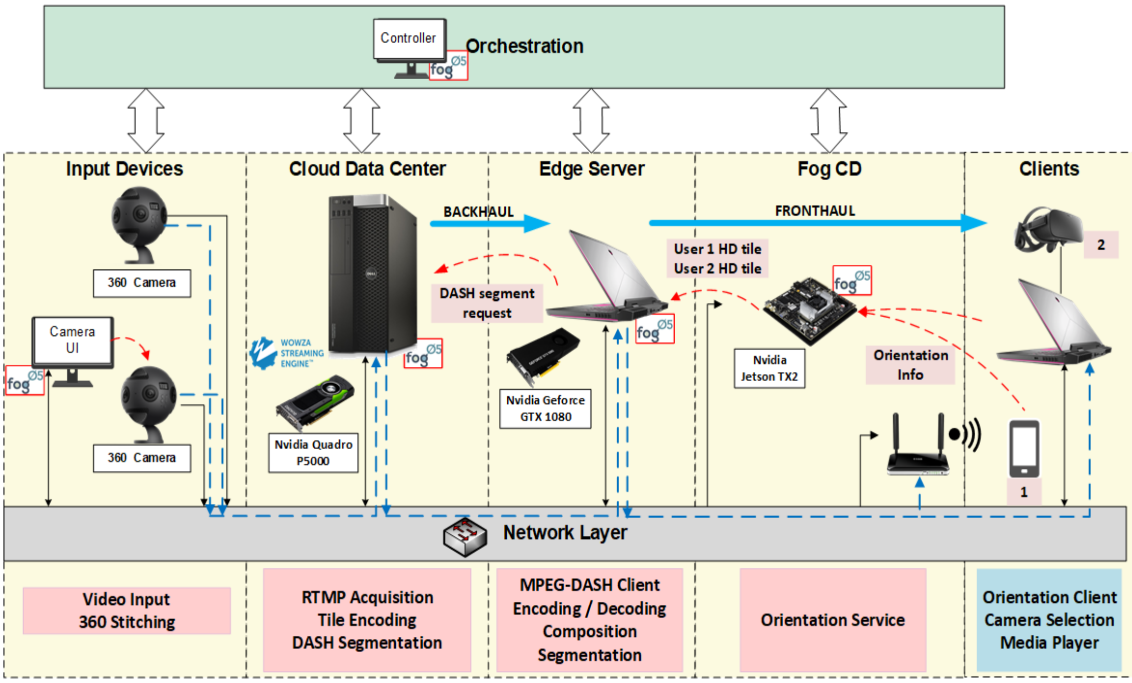


FIGURE 3-3: POC #2 OVERVIEW OF THE VR POC SETUP

3.1.1.1 Hardware Setup and Specifications

As previously described, the 360° live video streaming is initiated by the cameras connected to Wowza Streaming Engine hosted by the cloud data center [12]. To trigger the RTMP session, a computer laptop running the Fog05 agent is also plugged to the camera, thus allowing automatic session instantiation and termination. The Insta360 Pro camera is equipped with 6 fisheye lenses and can perform real-time stitching of 4K video sequences. Next, the data center handles the tile-based HEVC encoding by leveraging computing resources provided by the Nvidia Quadro P5000 GPU.

Our data center consists of a Dell Precision Tower 7810 equipped with an Intel Xeon E5-2670 v4 2.30 GHz CPU, 64 GB RAM and 1 TB HDD storage. In addition, this machine runs a DASH server compliant with the latest MPEG immersive Omnidirectional Media Format (OMAF) standard. The DASH segments are then received by the edge machine represented by a Dell Alienware 17 laptop equipped with an Intel Core i7-8750H CPU, 16 GB RAM, 128 GB SSD and a Nvidia GeForce GTX 1080 graphic card, able to execute complex tasks such as tile decoding and video frame composition. Furthermore, the edge server exploits the orientation information supplied by the fog node. To this end, we use an Nvidia Jetson TX2 development board, consisting of a Jetson TX2 module, which embeds a powerful GPU and two ARM CPUs. It is worth pointing out that although this platform features a high-performance 256-CUDA core graphic processor, we solely rely on the available CPU power, as the user orientation tracking is not meant to use hardware acceleration.

The orientation service computes in real-time the video stream tiles that must be encoded in high definition by processing the orientation info, i.e., yaw, roll and pitch, periodically reported by the terminals according to the orientation report rate system parameter. Also, a count-down timer is associated with each reported tile: the timer can be configured by setting the orientation decay period system parameter and is periodically decremented and reset whenever a new matching orientation is reported.

Finally, the edge server transmits the optimized DASH video stream to the clients. Specifically, we consider two types of video terminals, i.e., a Samsung S9+ Android-based mobile phone and an Oculus Rift VR headset connected to a computer laptop. Also, we conveniently developed an Android app featuring a user media player capable of reporting the orientation info as well as tuning to one of the video streams according to the user choice. Table 3-2 list hardware and software specifications considered in our setup.

TABLE 3-2: POC #2 HARDWARE SPECIFICATIONS USED IN THE EXPERIMENTAL SETUP

Hardware Component		Specifications
Fog Node	CPU	Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM® Cortex®-A57 MPCore
	Memory	8GB 128-bit LPDDR4 Memory
	Network	802.11a/b/g/n/ac 2×2 867Mbps WiFi 10/100/1000 BASE-T Ethernet
	GPU	256-core NVIDIA Pascal™
Access Point	Model	TP-Link TL-WR940N
	Wireless Adapter	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
	Memory	16MB
	Network	4× Ethernet 2×Wireless
Data center	CPU	Intel Xeon E5-2670 v4 2.30 GHz
	Memory	64GB 128-bit LPDDR4 Memory
	Network	10/100/1000 BASE-T Ethernet
	GPU	Nvidia Quadro P5000
Edge server	CPU	Intel Core i7-8750H
	Memory	16GB 128-bit LPDDR4 Memory
	Network	802.11a/b/g/n/ac; Gigabit Ethernet
	GPU	Nvidia GeForce GTX 1080
VR goggles	Model	Oculus Rift
Smartphone	Model	Samsung S9+
Camera	Model	Insta360 Pro

3.1.1.2 Software Setup and Specifications

Table 3-3 list the software specifications considered in our setup.

TABLE 3-3 : POC #2 SOFTWARE SPECIFICATIONS USED IN THE EXPERIMENTAL SETUP

Software Component		Specifications
Fog Node	OS	Ubuntu 16.04
	Kernel	4.4.38 (modified)
	LXD	3.0.3
Access Point	OS	Proprietary
Data center	OS	Windows 10 Professional
Edge server	OS	Windows 10 Professional

3.1.2 Required 5G-CORAL Building Blocks

Figure 3-4 lists all the EFS and OCS entities involved in this use case and shows the mapping between entities and computing substrates.

Video source: one or more 360° cameras are meant to capture the scene and dispatch the live video content to a cloud data center, which represents the EFS entry point. This is achieved by connecting the cameras with a streaming engine, e.g., Wowza Streaming Engine, and establishing a Real Time Multimedia Protocol (RTMP) live stream session, which ensures a TCP-based persistent connection and low-latency communication.

EFS: built upon three different tiers, i.e., cloud data center, hosting powerful processing units and located on cloud provider premises, edge server, located closer to the user equipment (UE) and providing limited capabilities, and fog computing devices (CD), resource-constrained devices operating in the UE proximity, the EFS offers all the essential functions and services to deliver the live video stream. After RTMP acquisition, the data center performs tile-based High-Efficiency Video Coding (HEVC) encoding, thus partitioning each video frame into three tiles (3 x 1 uniform tiling), each capturing a 120° viewing angle, which are encoded at high or low-quality resolution. Next, the DASH streaming server packetizes the bitstream data into multiple chunks, i.e., DASH segments, which are requested by the DASH client running on the edge server. Furthermore, the edge server decodes the tiled video streams and composes the 360° video frame for the UE. A key EFS component of our solution are the fog CDs deployed nearby the UE, whose main task is to collect the viewing orientation from the UE and to provide via HTTP REST API the DASH client with the UE viewport, i.e., the portion of the 360° video that must be delivered at high quality. This way, the DASH client is able to quickly recompose the video frame by decoding the correct tiles, depending on the orientation information sent by the fog CDs. Note that all the EFS software processes are native applications, as virtualization has not been considered in this work.

OCS: orchestration and control are accomplished by means of Fog05 [11]. Specifically, Fog05 processes the request for instantiation of an immersive video app made by the OSS and assigns computing tasks to each EFS component running the Fog05 agent in order to build the end-to-end video streaming service. For instance, Fog05 may instantiate the orientation service into a number of fog CDs as well as instruct a service migration between two fog CDs to support user mobility.

OSS: this module triggers the instantiation of the end-to-end video streaming service by sending a request to the OCS. The OSS may be managed by the multimedia service provider or network operator offering the 360° video streaming.

UE: the user terminal consists of three components, namely, a media player, a camera selector and the orientation client. The first two elements are managed by the user, whereas the latter runs in the background and forwards information on the user orientation to the orientation service located on the fog CD.

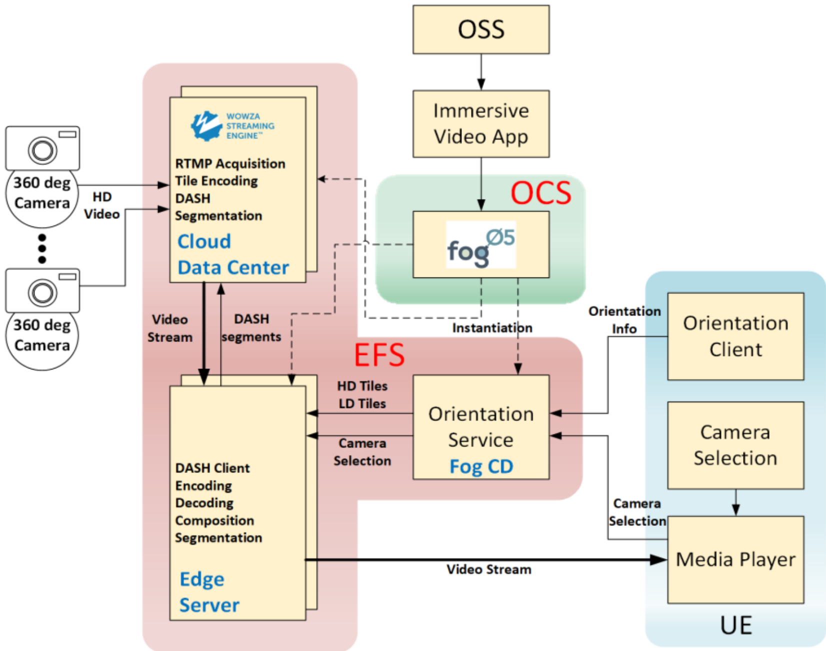


FIGURE 3-4: POC #2 SOFTWARE BUILDING BLOCKS COMPOSING THE E2E VR SERVICE

Table 3-4 presents the detailed list of 5G-CORAL EFS and OCS building blocks that constitute the VR PoC.

TABLE 3-4: POC #2 REQUIRED 5G-CORAL BUILDING BLOCKS

Sub-system	Component's Name	Description
EFS	RTMP acquisition	**EFS application** responsible for performing the RTMP acquisition enabling persistent connections and low-latency communications. It consumes data provided by the camera and sends the output data stream to the tile encoding application.
EFS	Tile encoding	**EFS application** to perform the tiled 360 video encoding. It processes the data stream coming from the RTMP acquisition application and provides the DASH segmentation module with the tiled encoded data stream.
EFS	DASH segmentation	**EFS application** in charge of segmenting the data stream encoded by the tile encoding application through DASH, which is consumed by the DASH client application
EFS	DASH client	**EFS application** to reassemble DASH segments sent by the DASH segmentation application. The output data is then sent to the decoding application. This application also uses the ET orientation information provided by the ET orientation service,

		which provides the user's view angle.
EFS	Decoding	**EFS application** performing the decoding of tiled video streams sent by the DASH client. The decoded video stream is then delivered to the composition EFS application.
EFS	Composition	**EFS application** responsible for re-composing tiled video streams into 360 video frame at the client side. This component receives tiled video streams decoded by the decoding EFS function.
EFS	ET orientation	**EFS service** responsible for selecting which tile has to be sent to the ET based on the orientation information provided by the orientation client. The selected tile is communicated to the DASH client.
OCS	EFS service platform manager	**OCS component** in charge of managing the orientation service deployed on the fog nodes. For this purpose, Fog05 will be used to manage the service lifecycle.
OCS	EFS App/Func Manager	**OCS component** responsible for managing multiple EFS applications required to run the VR streaming. For this purpose, Fog05 will be used to manage the application lifecycle.
Non-EFS	VR App media player	SW component running on the ET and on the laptop connected to the VR headset. It shows the VR multimedia content after being decoded and processed by the appropriate EFS elements.
Non-EFS	Orientation client	SW component responsible for informing the ET orientation service of the current ET orientation status, so that the viewport adaptive streaming service can be efficiently executed.
Non-EFS	Camera streaming	SW component interfacing with the 360 camera. It is mainly employed to instantiate a 360 live video streaming session with the edge data centre server.

3.2 PoC KPIs: Measured Values for the performance metrics

In this section, we report and discuss the results we obtained by measuring the KPIs listed Table 3-5. We first describe the methodology we adopted to conduct the experiments. Next, we present the results obtained and compare them with reference values reported in Table 3-6.

3.2.1 Measurement Methodology

We ran a set of experiments by using a 35-second pre-recorded 360 video sequence stored on the data center, which feeds the Wowza streaming engine. The results are generated by executing the same experiment for 10 repetitions and by recording the metrics every second. A summary of the system parameters employed is reported in Table 3-5. For the sake of simplicity, we only considered a stationary phone terminal and randomly changed its orientation in each repetition. To show the impact of the video processing load on the data center, we retrieved the GPU load, the GPU power consumption and the memory usage by using a free system tool called GPU-Z [13], which supports NVIDIA video cards. The first metric gives an estimation of how active the GPU is in a given interval, whereas the second metric indicates the power in Watt consumed by the GPU, and the last one reports the memory used. Furthermore, we extracted such metrics for three different configurations: idle, i.e., the video streaming service is inactive; split mode, where all the computing tasks, except for the orientation service running on the fog CD, are distributed between the data center and the edge server; no split mode, where all the tasks are executed by the data center.

Moreover, we obtained the bandwidth consumed in downlink by the user and considered three different streaming modes in order to highlight the benefits of the adaptive tile encoding strategy. The information on the bandwidth is obtained by using a network monitoring tool, such as Wireshark, which allows to monitor the bandwidth consumed by distinct applications on the same machine.

Finally, we measured service setup time, takedown time and E2E delay by repeating the same experiment for 30 times. The first two metrics were recorded by tracing the start and finish time in the Linux terminal where the Fog05 controller was located, whereas the E2E delay was obtained by using a 10ms-precision stopwatch and comparing the value shown in real time with the value being observed on the smartphone screen.

TABLE 3-5: PoC #2 SYSTEM PARAMETERS

System Parameter	Description
Video Resolution	4K (3840x1920)
Video Bitrate	30 Mbps
Video Frame Rate	30 fps
Orientation Report Rate	10 Hz
Orientation Decay Period	250 ms

3.2.2 Measurement Results

Figure 3-5 shows the GPU load, power consumption and memory used by the cloud data center measured through GPU-Z. As expected, the GPU computing load distribution between the data center and the edge server results approximately in a 7% reduction of the GPU load, which translates into more processing capacity available for other services running in the data center[1]. Furthermore, the split mode leads to a small reduction of the power consumed by the data center. Specifically, up to 2 Watts can be saved by offloading some GPU processing onto the edge server. This also means that most of data center power is spent executing the tile encoding and the DASH segmentation together with the Wowza streaming engine. Finally, it is worth pointing out that our approach helps to reduce the data center memory occupancy, with a memory saving equal to 600 MB in comparison with the no split mode.

[1] We could not collect the GPU metrics related to the edge server as the video card was not compatible with GPU-Z. Therefore, we could not quantify the potential load and power increase on the edge server side.

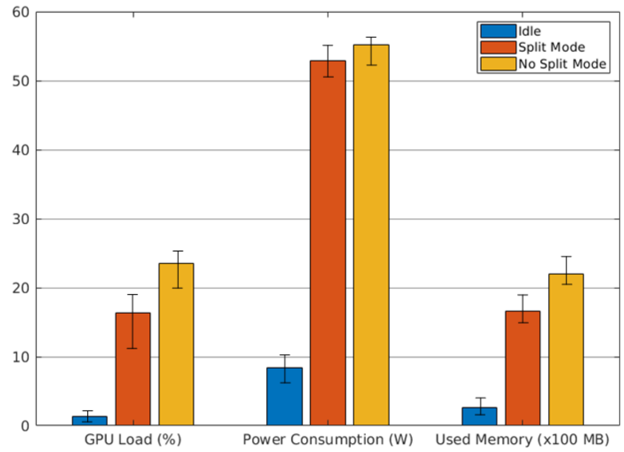


FIGURE 3-5: POC #2 GPU LOAD, POWER CONSUMPTION AND USED MEMORY MEASURED AT THE CLOUD DATA CENTER.

The Empirical CDF of the downlink data rate is shown in Figure 3-6. We note that the adaptive tile-encoding requires roughly a bandwidth equal to 21 Mbps, whereas a non-optimized approach encoding all the tiles with maximum quality leads to a bandwidth consumption equal to 31 Mbps, thus to a 33% increment. Obviously, this is due to the larger size of the DASH segment requested by the edge server, which results in a higher bandwidth consumption over the link between data center and edge server. If compared with the reference value in Table 3-6, we note again the benefit of adopting the tile-based encoding approach in reducing the downlink bandwidth consumption without affecting the user QoE.

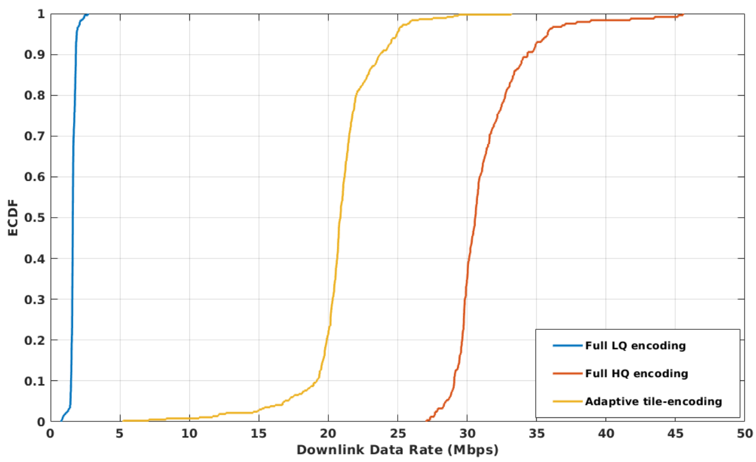


FIGURE 3-6: POC #2 DOWNLINK DATA RATE FOR FULL LQ, FULL HQ AND ADAPTIVE TILE-ENCODING.

Figure 3-7 shows the ECDF of the time taken by the OCS to set up the whole video streaming service. We obtained an average value of 10.75 seconds and a standard deviation equal to 0.35. This result is significantly lower than the value we reported in Year 1 and demonstrates the big enhancement in the OCS achieved in Year 2. Also, the ability of quickly deploying EFS elements hosted by heterogeneous resources makes 5G-CORAL suitable for operators and service providers, who can take advantage of zero-touch deployment solutions. We expect to obtain better performance over the next Fog05 releases, as the project is building momentum and the open-source community can support its development and maintenance.

We also measured the time taken to remove the service and free the resources. As shown in Figure 3-8, the service takedown time falls within the range between 11.8 and 13.2 seconds, with an average value of 12.62 seconds and standard deviation of 0.32.

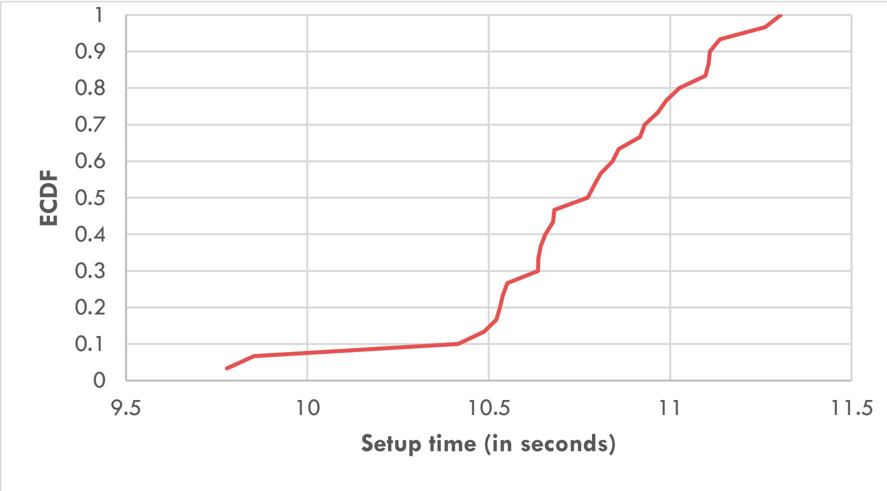


FIGURE 3-7: POC #2 SERVICE SETUP TIME.

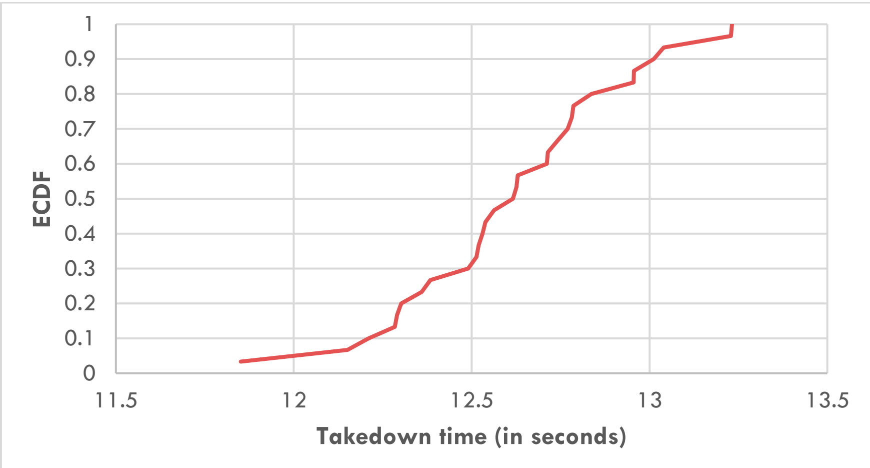


FIGURE 3-8: POC #2 SERVICE TAKEDOWN TIME.

The ECDF of the E2E delay affecting the video streaming service is shown in Figure 3-9. We notice that the delay is mostly localized around a region between 7.2 and 7.4 seconds, with an average value equal to 7.32 seconds and standard deviation of 0.09. Although this results in a high reduction with respect to the value reported in Year 1, such performance cannot ensure a good user QoE, as the spectator would appreciate the time difference between the video streaming and the real-world scene. As mentioned below, we plan on further reducing the E2E delay by considering different approaches, such as offloading the 360 video stitching from the cameras or employing optimized video streaming engine solutions.

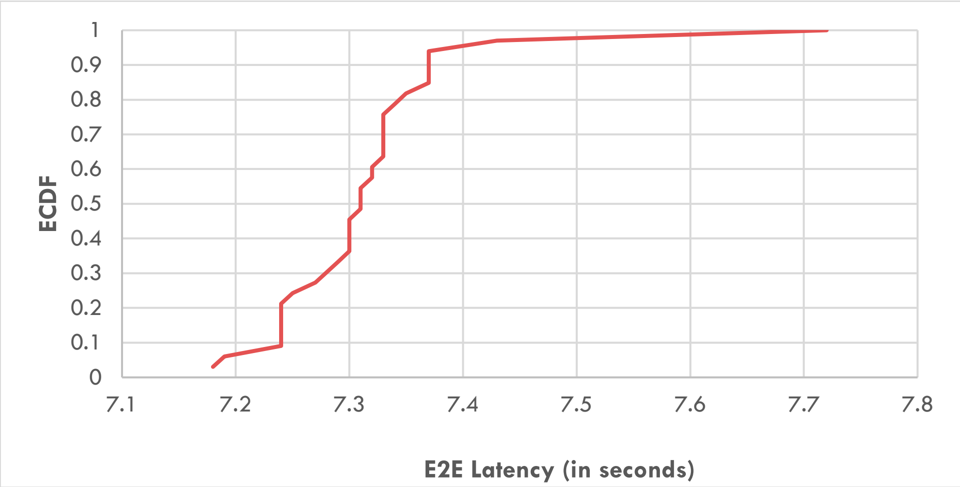


FIGURE 3-9: PoC #2 ECDF OF THE E2E LATENCY.

Table 3-6 shows the initial performance metrics list. Highlighted performance metrics will be measured using the VR PoC.

TABLE 3-6: PoC #2 MEASURED VALUES FOR THE PERFORMANCE METRICS RELATED TO VR

Performance metrics	Description	Reference Values	Measured Results
Bandwidth	Viewport adaptive streaming approach reduces the bandwidth required to stream 360-degree video while maintaining the same viewing experience	**30 Mbps** [14]	**21 Mbps** (average value)
Service Setup & Takedown Time	Service setup and takedown time refers, respectively, to the time it takes for the OCS to instantiate and remove all the VR components, distributed across the EFS.	**117 s** (set up time value obtained during Year-1 trial) Note: the takedown feature was not available during Year-1 trial	**10.75 s** (average setup time) **12.6 s** (average takedown time)
End-to-end Delay	End-to-end application delay among: Fog CDs, Edge server, Cloud Data Center (Cloud DC) and terminal clients.	**25 s** (reference value obtained during Year-1 trial)	**7 s** (average value)

Power consumption	Power consumed by the Graphic Processing Unit (GPU) on the Cloud DC when all the tasks are executed on the same machine or when offloading is employed.	**No reference** (reasonable distribution ratio of power consumption among multiple substrates).	**53 Watts** (average value)

3.3 Year 2 Achievements and Future Directions

In Year 2, our focus has been on reducing the E2E latency experienced by the VR user. In this respect, we extensively profiled the PoC and noticed that a large amount of latency was due to the storage unit on the data center. By replacing the Hard-Drive Disk (HDD) with a high-end Solid-State Disk (SSD), we managed to reduce the latency from 26 seconds down to 8 seconds. Further improvements could be accomplished by reducing the complexity of the system. For example, we observed that the Wowza streaming engine introduces a delay of 6 seconds in every iteration. Although we are not yet able to determine whether this is an inherent limitation of the video coding protocol or an implementation issue, as future work, we plan to replace the streaming engine with a more optimized streaming solution, able to ensure a lower video stream processing time. We also believe that the latency cannot be reduced below 1 second as the 360 camera executes the 360 stitching and introduces a latency higher than 1 second. For this reason, big effort will be directed towards developing solutions to move the stitching from the 360 camera to a high-end computing machine.

Thanks to enhancements to the Fog05 distributed key-value store, we also obtained a significant reduction in the time spent by the orchestrator in setting up and taking down the end-to-end video streaming service. As shown in the previous section, we reported a service setup time more than ten times lower than the one measured during Year-1 trials, and this has been achieved by replacing the DDS Pub/Sub protocol with a new protocol, i.e., Yet Another Key-value Store (YAKS) [15], which allows a faster and more reliable data sharing across the cloud-to-thing continuum. In the future, we expect to obtain a more stable orchestration platform based on YAKS, supported by the Eclipse open source community.

3.4 Demonstration activities

Throughout the project lifetime, we carried out a number of demonstrations and exhibitions at relevant venues in Europe and Taiwan. In the following, we describe the main events where we showcased our VR PoC.

In November 2018, the first 5G-CORAL trial was conducted at Global Mall Nangang Station store in Taipei, Taiwan, see Figure 3-10. Our PoC featured the multi-tier fog/edge/cloud platform presented above, while two 360 cameras were installed in different location of the exhibition area. Also, visitors were encouraged to test both the smartphone and the VR goggles in order to experience the 360 panoramic view and conveniently switch between the two video sources.

FIGURE 3-10: PoC #2 DEMO SHOWCASE DURING THE TAIWAN TRIALS.

Similarly to the first trial, in February 2019 the VR PoC was showcased during the Mobile World Congress 2019 in Barcelona, Spain, see Figure 3-11. We outlined the benefits of micro-services based distributed computing at the extreme edge of the 5G network, emphasizing the cost-efficiency in deploying an end-to-end 360 video streaming solution on top of the 5G-CORAL platform.

FIGURE 3-11: PoC #2 DEMO EXHIBITION DURING MWC'19 IN BARCELONA

At the EuCNC 2019 in Valencia, Spain, a joint demo between 5G-CORAL and 5G-TRANSFORMER was presented, see Figure 3-12. The goal was to demonstrate the benefits of introducing the network slicing component in our fog-edge platform. In this respect, we combined the 360 video service with a low-latency robotic application: a 360 camera was installed on top of a robot with the aim of delivering the video stream to a remote operator, who was able to drive the robot by watching the video coming from the 360 camera. Also, the camera wirelessly streamed the video to a Wi-Fi IEEE 802.11ac AP, which was connected to the data center. Two network slices, i.e., eMBB and URLLC, were instantiated by the 5G-TRANSFORMER vertical slicer module, which triggered the deployment of all the necessary functions and applications within the 5G-CORAL platform.

FIGURE 3-12: POC #2 JOINT DEMO BETWEEN 5G-CORAL AND 5G-TRANSFORMER AT EUCNC'19.

4 PoC #3: Fog-assisted Robotics

Fog-assisted robotics aims to enhance robot systems with powerful capabilities by leveraging the computing resources available in the Fog. To that end, by placing the intelligence of the robots in the Fog it is possible (i) to ensure low latency between the robots and their brains due to the short distance, and (ii) to consume context information on the access network in order to optimize the robotics operations. Consequently, it is possible to design lightweight and low-cost robots.

The Fog-assisted robotics PoC consists of two demo scenarios: cleaning robots and delivery robots. The idea of the cleaning robots scenario is to have robots that autonomously clean an area of the shopping mall. The robots clean periodically based on the density of visitors (people) in a given area of the shopping mall or manually triggered by a shopping mall employee upon an incident e.g., broken glass in some area. For this scenario to be realized, a dedicated virtual network is created for the robots, to isolate their operation and enable a third-party company to deal with the day to day operations. The robots are connected to the (virtual) network with the intelligence of the robots residing in the network. At start the robots are in idle state, connected to the network and out of sight for the shopping mall visitors. The EFS triggers (periodically or event-based) the cleaning of a certain area (e.g., where there are less visitors). For the sake of presentation, this can be manually predetermined area. The robots use the localization service to navigate their way to the cleaning area. Then the robots clean the area and return by navigating back to the initial position. In case the robot goes out of network coverage a new access point is deployed using the 5G CORAL infrastructure. The robots perform a handover to the new access point without interruption of service. The virtual AP migration is performed in order to extend the Wi-Fi coverage of the robots. This demo scenario can be realized with single or multiple robots.

The second demo scenario is based on delivering goods/products from storage rooms to the shops in the shopping mall. Since large items need multiple robots to carry them to a shop, the idea of the demo is to show synchronous cooperation between multiple robots, carrying a single item. Similarly, to the first demo scenario, a dedicated virtual network for the robots are created in which the intelligence of the robots resides. At start the robots are in idle state (maintaining the network connection) located in the storage room. An employee would manually initiate the transportation of a large items from the storage room to a shop. Two robots form a fleet using Wi-Fi and Wi-Fi direct technology. The Wi-Fi is used for infrastructure-to-robot communication assisting the robots for navigation and establishing Wi-Fi direct robot-to-robot connection. The Wi-Fi direct connection is established for robot-to-robot or Device-to-Device (D2D) communication for low-latency connection and maintaining better coordination (e.g., moving in formation). Upon establishing the fleet (formation) the large item is loaded on the robot. The robots deliver the item to the destination while maintaining the formation. Once the item is delivered, the robots break the formation and navigate back to the idle location. The following Table 4-1 shows the PoC user benefits.

TABLE 4-1: POC #3 PERFORMANCE METRICS AND USER BENEFITS

Performance metrics	Description	User Benefits
Delay	Communication delay between the Fog CDs and the robots	Low-delay communication that enables the move of the intelligence of the robot to the network.
Low-latency	The latency between issuing a command remotely (from the robot intelligence	Improved QoE with the user less likely to experience robot stop-and-

	application) and execution of the command by the robot	go movements
AP Deployment time	The time it takes to create the new APs in the Fog CDs using the Hostapd [16].	Easier and more flexible virtual access point deployment based on the context information available in the EFS.
Reaction time	The time between detecting an obstacle and execution of the remote command by the robot.	Faster detection and response time of the robots to environmental changes.
Total migration time	Time it takes to perform the virtual AP migration, due to mobility of the robots.	Extended Wi-Fi range of the mobile robots without interruption of service.
Downtime	Period of time that the robots are without Wi Fi connectivity.	Fast and secure handoff from one AP to another that permits continues connectivity for robots in motion.

4.1 PoC Testbed Design

In this section we describe in detail the Fog-assisted PoC testbed designed in 5TONIC lab [17]. The testbed is designed to support different virtualization techniques (LXD/KVM), native applications and multiple communications channels simultaneously, as shown in Figure 4-1. In the following, we will present an overview of the hardware and software integrated components and describe the 5G-CORAL building blocks.

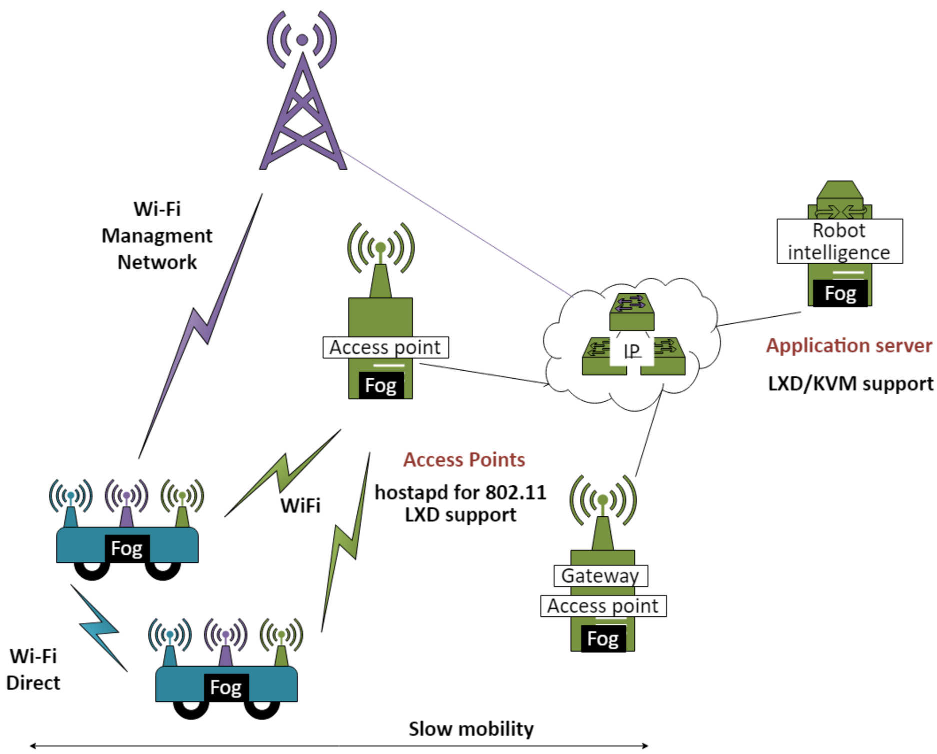


FIGURE 4-1 : POC #3 SETUP OF FOG-ASSISTED ROBOTICS

4.1.1 Overview of Integrated Components

As described in the previous section, in order to realize the Fog-assisted robotics scenario, several components need to be deployed. All EFS and OCS entities are shown in Figure 4-2.

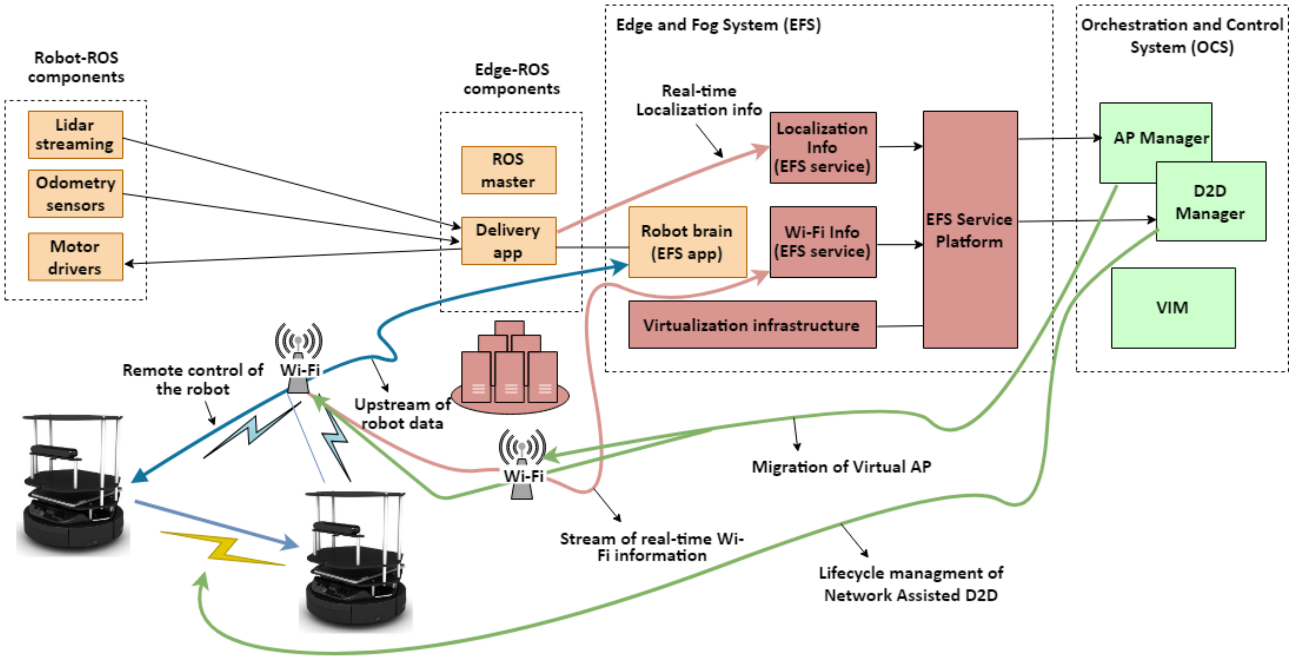


FIGURE 4-2: POC #3 SOFTWARE BUILDING BLOCKS COMPOSING THE FOG-ASSISTED ROBOTICS

The Edge and Fog System implemented in our testbed comprises a set of EFS resource, specifically one Fog CD (Fog 02 in Figure 4-3), two Wi-Fi enable Fog CDs (Fog 01 and Fog02 in Figure 4-3) for providing Wi-Fi connectivity to the robots, and the robot themselves. An EFS function implements the Wi-Fi Access Point (AP) capabilities in both Fog nodes and it is implemented as a container. Therefor the communication between the robots and the EFS applications comprises a Wi-Fi link, from the robots to the AP, and a wired network, from the AP to the Fog CDs. Two EFS services are implemented, namely Wi-Fi Info and Localization Info EFS service. The Wi-Fi info EFS service is deployed on the Fog CDs that have Wi-Fi connectivity and provides real time context information about the robots connected to the Wi-Fi AP. The Localization EFS service tracks the robots estimated 2D position. Both EFS services are publishing their information over the EFS Service platform which is implemented in the form of MQTT broker. The robotics EFS application in our set-up is implemented as various ROS components distributed across the robots and the Fog CD. In the Fog, the ROS components are embedded in a virtual machine, while on the robots the ROS components run as native applications. As illustrated in Figure 4-2, the ROS components on the robots are in charge of (i) reading data from the sensors (e.g., odometry, Lidar) and send to the robot brain and (ii) executing driving commands from the robot brain.

The OCS in our testbed consists of EFS Manager and VIM. The EFS Manager is responsible for lifecycle management (e.g instantiation, query and termination) of the virtual AP migration and Network-assisted D2D. We implemented the EFS Manager based on the description presented in Section 3.1.3 in D3.2 [18]. This section explains the OCS workflow needed to establish the D2D communication and perform the migration. In addition to this, fog05 [11] is used as VIM.

Figure 4-3 shows the exemplary scenario that we implemented on an experimental testbed in 5TONIC [17] laboratory. As described before, it is composed of 3 static Fog CDs (mini PCs), a Gigabit switch, Wi Fi Router (Linksys WRT160NL Router) and 2 mobile robots (Turtlebot 2) with Kobuki mobile base. On 2 static Fog CDs, 802.11ac capable modules are implemented. On the other side, the robots are equipped with lap top and Lidar (RPLIDAR-A2). It is worth mentioning that each robot has three Wi Fi adapters that are used for management, infrastructure-to-robot and robot-to-robot communication consequently.

FIGURE 4-3: POC #3 EXPERIMENTAL TESTBED SETUP (FOG CDS, ETHERNET SWITCH, MANAGEMENT AP AND ROBOTS)

4.1.1.1 Hardware setup

This PoC requires the following hardware devices to run different software components:

- **Robots** – The first demo scenario can be realized with a single robot, whereas for the second scenario (item delivery) at least two robots are needed to present the robot-to-robot coordination and moving in formation. The robots are be pre-built robots (e.g., Robotnik – Turtlebot [11] equipped with Lidar with software component running on top the Robot Operating System (ROS). The Turtlebot robot is shown on Figure 4-4 on the left. On Figure 4-4 on the right are shown the Lego Mindstorm robots used in the demonstration described in Section 4.1

FIGURE 4-4: POC #3 TURTLEBOT ROBOT (LEFT) AND LEGO MINDSTORM EV3 ROBOTS (RIGHT)

- **Access Points (APs)** – Two hosting systems (devices) that can support LXD and have 802.11 capable cards.
- **Intelligence point** – Single host device that support KVM. The robots' intelligence is stored and run on these devices.

The hardware specification details can be found in Table 4-2:

TABLE 4-2: POC #3 HARDWARE SPECIFICATION USED IN THE TESTBED

Hardware Component		Specifications
Robots	Model	Kobuki TurtleBot2
	CPU	Logic Supply Jetway NF9G-QM77 64-Bit CPU Intel Core i5 -3610ME CPU @ 2.70 GHz
	Memory	8GB DDR2 Memory
	Network	10/100/1000 BASE-T Ethernet
Access Points	CPU	Logic Supply Jetway NF9G-QM77 64-Bit CPU Intel Core i5 -3610ME CPU @ 2.70 GHz
	Memory	8GB DDR2 Memory
	Wireless Adapter	Qualcomm Atheros QCA986x/988x 802.11ac
	Network	10/100/1000 BASE-T Ethernet
Intelligence point	CPU	Logic Supply Jetway NF9G-QM77 64-Bit CPU Intel Core i5 -3610ME CPU @ 2.70 GHz
	Memory	8GB DDR2 Memory
	Network	10/100/1000 BASE-T Ethernet

4.1.1.2 Software Setup

- **Access Points (APs)** – The host device would run Hostapd [16] inside LXD containers in order to create new 802.11 access points. The newly created access points use the same (existing) physical interface. The LXD containers are based on the Alpine Linux (lightweight Linux distribution for cloud environments).
- **Gateway** – A DNS forwarder, DHCP server and firewall that runs on top of a device that supports LXD. It would act as a gateway for the robots' end-to-end service.
- **Robot brain** – ROS based software that runs on a device that supports KVM. It acts as ROS master for the robotic system and also hosts the intelligence of the robot, in charge of navigating the robot based on the available information.

The software specification details can be found in Table 4-3:

TABLE 4-3: POC #3 SOFTWARE SPECIFICATIONS USED IN THE TESTBED

Software Component		Specifications
Access Points (APs)	OS	Alpine Linux 3.8.0
	Hostapd	2.6
	LXD	3.14
Gateway	OS	Alpine Linux 3.8.0

	Dnsmasq	2.79
	LXD	3.14
Robot brain	OS	Ubuntu 16.04
	ROS	Kinetic Kame
	KVM	2.11.1

4.1.2 Required 5G-CORAL Building Blocks

Table 4-4 presents description of the 5G-CORAL (OCS and EFS) building blocks through which the Fog-assisted robotics PoC is implemented.

TABLE 4-4: POC #3 REQUIRED 5G-CORAL BUILDING BLOCKS

Sub-system	Component Name	Description
OCS	VIM	**VIM** - Automatic instantiation. Configuration of the WiFI AP, deployment and instantiation of the Robot App into the robot.
OCS	Manager	**EFS Manager** - Responsible for lifecycle management (e.g instantiation, query and termination) of the EFS functions.
EFS	Network assisted D2D	**EFS Function** - Implementation of Wi-Fi Direct P2P connection to establish robot-to-robot communication.
EFS	Virtual AP (Wi-Fi)	**EFS Function** – Software based access point capable of turning normal network interface cards into access points. Used to pride robot-to-infrastructure connectivity.
EFS	Mobility	**EFS Function** - Enables continues connectivity for the mobile robots, with fast and secure handoffs from one virtual AP to another.
EFS	Localization Info	**EFS Service** -The density of visitors supports the system to initiate the demo scenarios and where to operate. The localization service provides significant information for the robot to navigate and reach the desired location in the shopping mall
EFS	Wi-Fi Info	**EFS Service** – Provides real time context information (e.g., signal strength, packet transmissions, packet re-transmissions, connection time etc) about the robots connected to the Wi-Fi AP.
EFS	EFS robot application	**EFS Application** – Implementation of the robotics system as various ROS components distributed across the robots and the Fog CD.

EFS	EFS Service platform	MQTT broker that handles sub/pub for **EFS Services**
EFS	Gateway	**EFS function** – A DNS and DHCP server runs on top and acts as a gateway for the robots' end-to-end service

4.2 PoC KPIs: Measured Values for the performance metrics

In order to explore the benefits and performance of the 5G CORAL platform, we build experimental environment using the testbed described in the previous section. In such environment we have deployed all the components shown in Table 4-4. The objective of the experimental testbed is to evaluate the Fog controlled robotics paradigm. Table 4-1 presents the list of all performance metrics and their user benefits that are measured for the Fog-assisted robotics PoC.

4.2.1 Measurement Methodology

4.2.1.1 Delay

The measurements for the delay KPI were done measuring the TCP communication delay between the robot Intelligence and the robots over a Wi-Fi Channel. In this case a dedicated virtual network is created to enable connectivity between the robots and their intelligence that resides in the network. The robot is connected to this network via the virtual AP. Ping messages from one of the robots to the robot Intelligence are used to measure the RTT communication delay.

4.2.1.2 Low-latency

The Fog-assisted robotics QoE for the end user demands very low latency between issuing a command remotely and execution of the same command by the robot. In this experiment, the mobile robot is controlled by the robot brain application. The robot brain (running in the Fog node) sends movement commands to the robot using ROS messages, published in a specific topic devoted to the movement commands. The movement command is composed of a tuple (speed, timestamp), where speed is the linear velocity of the robot and timestamp is the time when the command has been issued in the robot brain. Once the command is received in the robot, we extract the timestamp and compute the delay in receiving the movement instruction. During the duration of the experiment, all components were synchronized and share the same time reference for accurate measurements. All the data that is exchanged between the robot brain application and the mobile robot is based on TCP.

4.2.1.3 Reaction time

The reaction time KPI measures the amount of time it takes to respond to an obstacle detection. To measure the reaction time, the robot was navigated in a close loop by the Robot brain. The mechanism for exchanging data between the Robot intelligence and the robot ROS components is based on TCP. The closed-loop starts with the robot brain sending speed command to the robot using ROS messages. Upon receiving the speed parameter thought the wireless link, the robot wheels start to move. The loop is then closed by the robot continuously sending-back the Lidar sensor data to the robot Intelligence. The brain analyzes the Lidar data and generates new speed command, which will serve as input to the next turn of the closed-loop. The closed-loop stops with the robot brain sending stop command to the robot. This command is generated when the robot Brain detects that there is an obstacle 1 meter away from the robot.

The experiment runs were performed in a square empty room at 5TONIC laboratory. Each run consists of the robot brain driving the robot on a straight line towards a wall. The starting position of the robot is approximately 1.5 meters away from the wall. Then the robot accelerates from the starting position with speed of 0.1 m/s and drives according to the close-loop mechanism. After arriving approximately 1 meter away from the wall, the robot stops.

To that end, we performed 50 experiment runs. For each experiment run, in the robot measure the reaction time, while in the robot brain we record the obtained Lidar data.

4.2.1.4 AP Deployment time

The AP deployment time KPI is very important to measure the real benefit of having the deployed 5G CORAL platform components for realization of the Fog-assisted robotics PoC. For measuring the AP deployment time, our custom EFS Manager will instantiate an instance of Hostapd [16] in one of the Fog CD by passing a descriptor to fog05 [11]. In our case, as deployment time we consider the time between instantiating the virtual AP from the EFS Manager and the moment when the virtual AP SSID is available.

4.2.1.5 Total migration time

The virtual AP migration feature in Fog-assisted robotics PoC consists of three phases. First, the EFS manager with the help of Wi-FI Info EFS service, detects that virtual AP migration is needed and instantiate a new virtual AP. Then, when the new virtual AP is up and running the EFS manager triggers the handoff from the currently associated virtual AP to the new one. It is important to mention that both virtual AP instances are configured to use IEEE 802.11r [19] in order to permit extended connectivity for the robot in motion, with fast and secure handoffs. The 802.11r IEEE specification offers two options to speed up the authentication exchange: FT over the Air and FT over DS. In our case, we are using FT over DS.

In order to measure the total migration time, we performed 50 experimental runs. For each experimental run in the EFS Manager we measured the total migration time, which is the time elapsed from detecting that virtual AP migration is needed till the handover is performed. Moreover, to normalize the deployment time we already make available a copy of the Hostapd image in the Fog node. By doing this, the EFS Manager does not need to copy the Hostapd image over the network.

4.2.1.6 Downtime

The downtime KPI measures the period that the robots are without Wi-Fi connectivity due to the migration of virtual AP. The measured KPI is important because it provides insight of the robot control loop duration limit in order to have successful roaming without interruption of service.

We performed 50 successful handovers using two Hostapd [16] based virtual AP. Both APs were pre-configured to use 802.11r [19] over DS. For each handover we measured the overall downtime.

4.2.1.7 D2D deployment time

The D2D deployment time KPI is the time required to establish Wi-Fi Direct D2D connectivity between the two robots, using the Wi-Fi as infrastructure-assistance. Our custom EFS Manager is implemented as a container in one of the Fog CDs and follows the work-flow described in Section 3.1.3.2 in D3.2 [18].

In order to measure the D2D deployment time we performed 100 experimental runs. For each experimental run we measured the D2D deployment time, which is the time elapsed from detecting that the D2D connection can be established (in the EFS Manager) till the moment both robots are reachable over the D2D connection. In order to ensure reachability, we used the ping command.

4.2.2 Measurement Results

The obtained data from the Delay, low-latency and reaction time experiments are analyzed and aggregated to generate the results presented in Figure 4-5. In overall, Figure 4-5 presents the Cumulative Density function (CDF) for each set of measurements. From left to right, first the results regarding the communication Delay are presented. In this plot we measure the round trip-time (RTT) from one of the robots to the robot Intelligence and back. The second and third plot reports the results for low-latency and reaction time KPI.

TABLE 4-5: POC #3 DELAY, LOW-LATENCY AND REACTION TIME STATISTICAL CHARACTERISTICS (S)

Experiment	Min	Max	Mean	Median	Std Dev.
Delay	0.001	0.342	0.006	0.001	0.028
Low-latency	0.001	0.163	0.004	0.002	0.006
Reaction time	0.002	0.139	0.023	0.012	0.028

The most important statistical values of the measured performance metrics with good network strength, little noise and no congestion are reported in Table 4-5. Starting by analyzing the Delay experimental scenario, we can notice that the RTT mean value of the distribution is 6 ms. This is reasonable since as we move the robot intelligence to the Fog, the packets don't need to travers one or more networks and we can offer low communication delay. Regarding the low-latency and reaction time performance metrics, these values are directly affected by the RTT delay and ROS as meta-operating environment for controlling robots. The mean value of the low-latency and reaction time distributions are 4ms and 23ms.

In 2018 ETSI performed a study [20] on 5G system requirements for different use cases in Ultra-Reliable Low-Latency Communications. In this study by ETSI, Section 5.3.7 is devoted to mobile robots and their demands. For robot cooperative driving a cycle time of 10-50 ms is set as a potential requirement. Our results show clearly that the 5G CORAL Fog-assisted robotics paradigm supports this cyclic data communication characterized by ETSI and we can enable remote and synchronized control of robots.

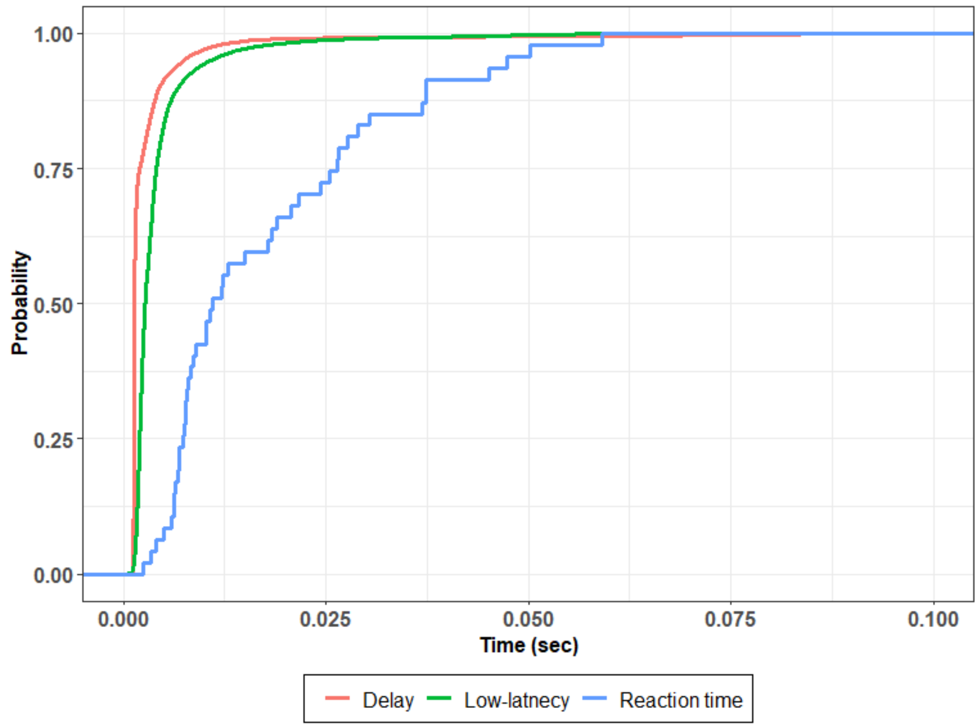


FIGURE 4-5: POC #3 EXPERIMENTAL CDF OF DELAY, LOW-LATENCY AND REACTION TIME KPI

The publish and subscribe mechanism for exchanging messages between the robot and the robot intelligence is based on TCP. This means undesired delay and/or packet loss can be introduced in the system by any interference on the Wi-Fi channel. Additional delays and/or packet loss can result in decreased accuracy and slower detection by the robot intelligence. Moreover, additional delays in the execution of movement commands can degrade the smoothness of the driving.

In Section 5.4 of D3.2 [18], the Network assisted D2D scenario is evaluated and the results show how the latency reduction of the D2D communication channel can help to optimize the coordinated movements of the robots.

Based on this analysis, we performed 2 sets of measurements each containing 30 experiment runs. For the first set of measurements we recorded the jitter between the robot and the robot brain, using the virtual AP. For the second set of measurements we measured the jitter between the two robots using Wi Fi Direct [21].

Figure 4-6 reports the experimental Cumulative Density Function (eCDF) for each set of measurements. The x-axis represents the jitter in milliseconds while the y-axis represents the cumulative percent. Regarding the line presented in the figure: Centralized line shows the eCDF of the jitter between the robot and the Robot brain. Similarly, D2D shows the eCDF of the jitter between the two robots.

TABLE 4-6: POC #3 STATISTICAL CHARACTERIZATION OF JITTER (MS)

Experiment	Min	Max	Mean	Median	Std Dev.
Centralized	0.261	9.564	1.33	0.99	0.920
D2D	0.388	4.897	1.342	1.199	0.614

The breakdown of the jitter along with the most significant statistical parameters is reported in Table 4-6. The results show that there is no significant difference between some of the parameters, such as the minimum and the average. This is understandable since we are using Wi-Fi as radio access technology for both experiments. However, peaks in the delay, decreased network speed, poor signal level and increased jitter between the robot and the robot brain can be experienced due to environmental changes (e.g., obstacles, like windows, metal cabinets) and other external interferences. Accordingly, the network assisted D2D communications helps at avoiding such scenarios.

For this reason, we decided to measure the reaction time of the network assisted D2D. We performed the measurement methodology described in Section 4.2.1.3 but this time the close-loop mechanism for navigating the robot was placed in the second robot. We used the established D2D communication channel to exchange the ROS messages.

Figure 4-7 presents the time (in seconds) needed to respond to an obstacle detection. The box plot includes the comparison between the D2D and Centralized reaction time. We may notice that the Centralized reaction time has periodic peaks of +50ms. Aldo these peaks are not so frequent they enough to cause imprecision in the coordinate driving of the robots. On the other side, the D2D reaction time has a quite low standard deviation with most of the values close to the average. The experiment results demonstrate that D2D connection can improve the robotics movement precision and reaction, Table 4-7.

TABLE 4-7: POC #3 STATISTICAL CHARACTERIZATION OF REACTION TIME (S)

Experiment	Min	Max	Mean	Median	Std Dev.
Centralized	0.002	0.139	0.023	0.012	0.028
D2D	0.002	0.043	0.019	0.015	0.011

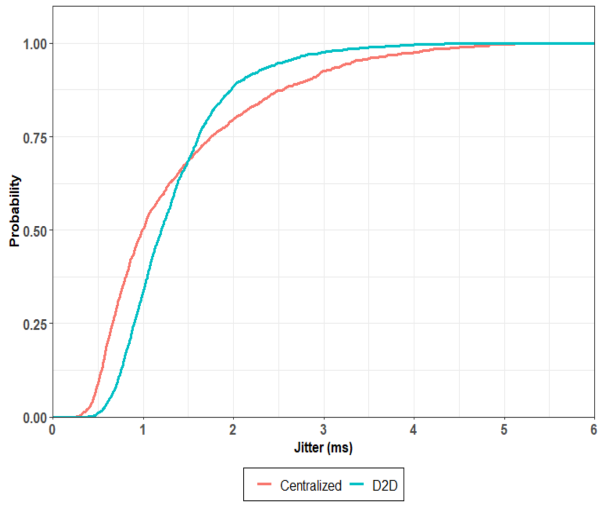


FIGURE 4-6: POC#3 EXPERIMENTAL CDF OF JITTER

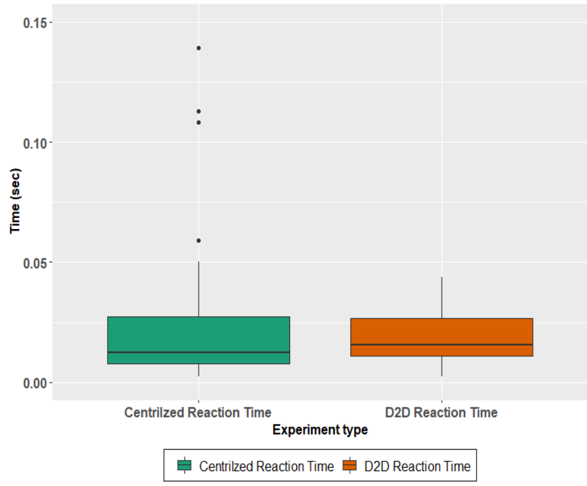


FIGURE 4-7: POC#3 ROBOT REACTION TIME. THE BOX PLOT INCLUDES COMPARISON OF CENTRALIZED VERSUS D2D REACTION TIME

The obtained results regarding the deployment time are based on average values of 100 trails for each presented case. Figure 4-8 shows the comparison between the total deployment time of virtual AP and D2D EFS functions. All the experiment runs were performed using fog05 as virtual infrastructure manager. Table 4-8 on the other side shows the most important statistical parameters of the downtime.

TABLE 4-8: POC #3 STATISTICAL CHARACTERIZATION OF DEPLOYMENT TIME (S)

Experiment	Min	Max	Mean	Median	Std Dev.
Downtime	0.012	0.091	0.038	0.035	0.012

Starting by analyzing the total deployment time, results show that 5.63 seconds are required to establish a successful robot-to-robot communication and 23.91 seconds are required for deployment of virtual AP. These results are strictly bounded to the different implementation procedures of the EFS functions. On one side, we have the D2D EFS function that implements the Wi-Fi Direct procedure [21]. This procedure is quite simple, and it is composed of (i) scanning for available p2p devices and (ii) establishing p2p connection. On the other side, the virtual AP EFS Function is implemented using Hostapd [16] and is deployed in a Linux container. The results show that fog05 need 3.61 seconds to instantiate the virtual AP container using LXD. The rest of AP deployment time (20.3 s) is due to Hostapd [16] Linux boot time and the time needed the robot to detect that the SSID is available. Regarding the time that the robots are without Wi-Fi connectivity we recorded a value of 38 ms.

The obtained results show that by using the context information available in the EFS we can extend the Wi-Fi range of the mobile robots without interruption of service. With a downtime of 38 ms this functionality is enabled for cooperative robotic driving (cycling time < 50ms [20]) and for video operated remote control (cycling time < 100ms [20]). Furthermore, the context information can be also used to deploy the D2D communication for maintaining better synchronization between the robots in cases when there is increased interference on the robot-to-infrastructure Wi-Fi channel.

Table 4-9 presents a summary of the validation results for the performance metrics for the PoC compared with the reference values.

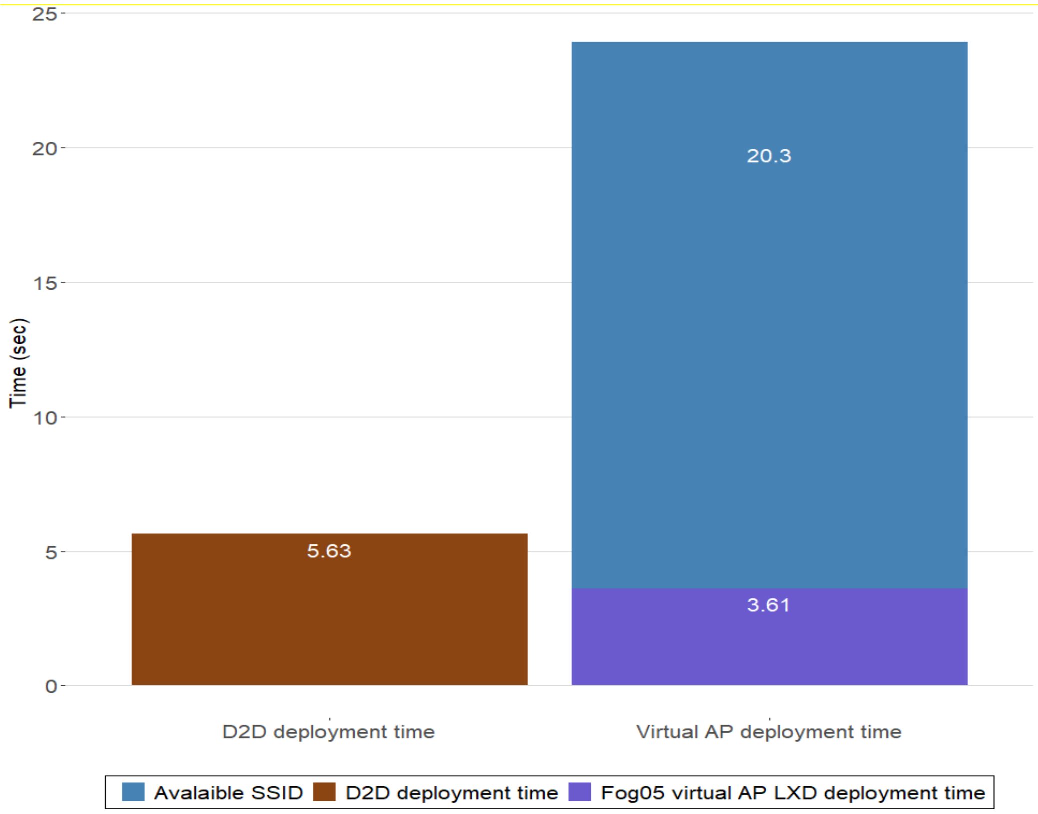


FIGURE 4-8: POC #3 DEPLOYMENT TIME. THE BAR PLOT INCLUDES D2D AND VIRTUAL AP DEPLOYMENT TIME.

TABLE 4-9: POC #3 MEASURED VALUES FOR THE PERFORMANCE METRICS RELATED TO FOG-ASSISTED ROBOTICS.

Performance metrics	Description	Reference Values	Measured Values
Delay	Communication delay between the Fog CDs and the robots	~50 ms RTT between robots and Cloud [22]	6ms
Low-latency	The latency between issuing a command remotely (from the robot intelligence application) and execution of the command by the robot	~3ms to set speed to the robot from Edge server [23] ~6ms to read robots's sensors from Edge server [23] ~35 ms to take a photo from robot's camera from Edge server [23]	4ms

AP Deployment time	The time it takes to create the new APs in the Fog CDs using the Hostapd [16].	6~8s to deploy Docker containers of different sizes [24]	3.61s to deploy the LXD container
Reaction time	The time between detecting an obstacle and execution of the remote command by the robot.	1~50ms to enable cooperative driving [20]	23ms
Total Migration time	Time it takes to perform the virtual AP migration, due to mobility of the robots.	6.3~70 s to migrate LXC containers, (the migration time depends on the container size) [25].	24.29s

4.3 Year 2 Achievements and Future Directions

This section emphasizes the Fog-assisted robotics progress done in the second year and proposes possible directions that may be taken in the future to extend the work that has been done.

From EFS point of view, two new EFS functions were added to the PoC in the second year. The network assisted D2D function, that enables the robot-to-robot communication and the Mobility EFS function that permits extended continues connectivity for the mobile robots. Localization EFS Service was integrated to tracks the robots estimated 2D position and make it available over the EFS Service platform.

Regarding OCS, EFS Manager was implemented to the PoC, responsible for lifecycle management (e.g. instantiation, query and termination) of the EFS functions.

For what concerns future directions several options have been identified. First, we would like to investigate the possibilities to integrate different RATs (LTE, MM wave) for remote coordinated control and understand the advantages and disadvantages of each technology. Second, it may be possible to explore different communication protocols in order to improve the cycle time of the mobile robots. As an additional future direction, the Fog-assisted robotics use case could be integration with the SD-WAN, to demonstrate how the Fog-assisted robotics use case can easily federate with other domains, possibly integrating the cloud as a new computing domain where the vertical will be able to instantiate application resources during off-peak hours, lowering the CAPEX and OPEX costs. The SD-WAN will provide roaming and offloading capabilities to the robot-assisted use case vertical, abstracting the network configuration complexity of the whole process. The vertical will be able to plan accordingly where the robotic brain is capable to roam (roaming strategies) and when the offloading (offloading strategies) will be triggered. The SD-WAN will allow the robotic brain to roam across the fog and the cloud and offload its robotic brain to the cloud when required

4.4 Demonstration activities

During the duration of the project, we have shown public demonstrations in different venues. One version of the Fog-assisted robotics demo was showcased in the first year of the project at Ericsson Innovation Day 2018, Figure 4-9 and European Microwave Week 2018, Figure 4-10. This demonstration consisted of two Lego Mindstorms robots placed on a table inside a marked rectangle (with black ink) area. The robots form a fleet and move within the rectangle border in a coordinated manner. In case one of the robots arrive at the rectangle black border, both robots stop and change the direction of the movement. The idea is to present the process of deploying the robot resources on the fog devices, control the robots remotely by placing the robot brain in the network, assisted by the Wi Fi infrastructure.

The Fog-assisted robotics PoC that is described in the beginning of this section was successfully presented in the Taiwan mid-term trial audit, Figure 4-11. In the Nangang shopping mall in Taipei, Taiwan we publicly demonstrated how to use a fleet of robots in order to have a cooperative delivery of large items.

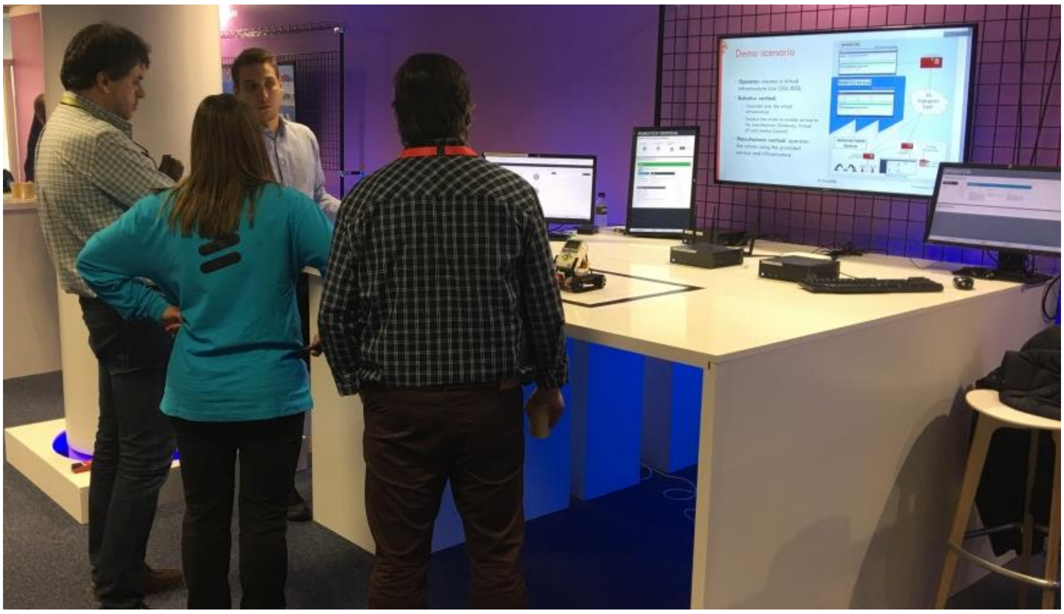


FIGURE 4-9: POC #3 FOG-ASSISTED ROBOTIC DEMO AT ERICSSON INNOVATION DAY

FIGURE 4-10: POC #3 FOG-ASSISTED ROBOTIC DEMO AT EUROPEAN MICROWAVE WEEK

FIGURE 4-11: POC #3 FOG-ASSISTED ROBOTIC DEMO BEING PRESENTED IN THE TAIWAN MID-TERM TRIAL AUDIT

5 PoC #4: Multi-RAT IoT

This PoC testbed is designed for the Multi-RAT IoT use case described in D1.1 [1] to investigate and showcase the feasibility to cloudify the communication stacks of multiple IoT RATs in EFS of a 5G-CORAL system. The main benefit is that cloudification of multi-RAT functions makes an IoT access system RAT-agnostic, which can improve significantly the system scalability, flexibility, future-proofness and reduce the costs thanks to the possibility of orchestration and automation.

In WP2, we mainly developed the communication-stack functions of individual IoT RATs, namely IEEE 802.15.4, NB-IoT and LoRa, and investigate their performances separately (More details can be seen D2.2 [26].). After the WP2 verification on each IoT RAT, in WP4, we focus on the integration of these IoT RATs in one integrated PoC testbed and investigate the performance impacts between RATs when they are running simultaneously in a shared physical infrastructure of the testbed. It also showcases a full EFS design with the three key EFS elements, namely EFS functions, applications and services, as well as using MQTT as the EFS service platform.

In the following, we will provide more details about the PoC design, describe the test methodology, and test results. The results show that having 3 RATs running simultaneously achieves the same performance as the single RAT run individually, which verifies the feasibility of the multi-RAT IoT use case. Further, it shows that the IQ data service can be used to capture and detect BLE interferences, as one example of Interference Analyser application based on the IQ data service. In addition, the 2nd year achievement, future directions and the past demonstration activities are also presented.

5.1 PoC Testbed Design

During this project, we prototyped three IoT access communication stacks (transceiver protocol stacks) of IEEE 802.15.4, NB-IoT and LoRA in software, which are virtualized using Docker containers. In the PoC testbed, we integrate the three RATs in the same setup, as shown in Figure 5-1. The testbed is designed to support IEEE 802.15.4, NB-IoT and LoRa simultaneously. As an example of EFS service, we further added the IQ-data publishing functionality on the radio head side, integrated a MQTT broker as the EFS service platform and developed an Interference Analyzer application which performs IQ data analysis using the received IQ data subscribed via the MQTT broker.

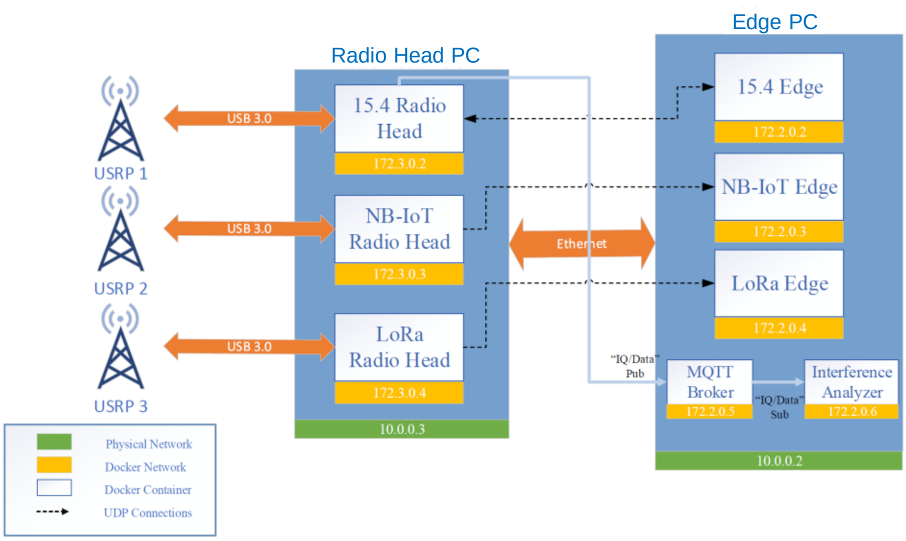


FIGURE 5-1: POC #4 BLOCK DIAGRAM OF THE MULTI-RAT IOT POC DESIGN

5.1.1 Overview of Integrated Components

An end-to-end testbed is designed for PoC purpose. In the following, more details are provided regarding hardware components and software components used and developed in the testbed setup.

5.1.1.1 Hardware Setup

In the hardware setup, there are network/edge side and terminal side. The EFS infrastructure of 5G-CORAL is on the network/edge side. The following provides more information regarding the hardware components used in the testbed, while the hardware specifications are summarized in Table 5-1 as well.

- **Network/edge side**: the edge infrastructure in this use case. The hardware setup is shown in Figure 5-2 with the following hardware components
  - **USRPs**: USRP (Universal Software Radio Peripheral) [27] is a well-known SDR kit which is well supported by open-source communities (e.g. GNU Radio [28]) and widely used in research and prototyping works. We chose to use USRP for the PoC development. In this testbed, three USRP B210 [29] used for three RATs, respectively, which are shown as USRP 1, 2 and 3 in Figure 5-1. These are the SDR kit for RF-baseband conversion. The following presents the current configurations.
    - IEEE 802.15.4: It is configured at 2.4 GHz band with sample rate of15 Msps to cover three 5 MHz channels for multi-channel implementation.
    - NB-IoT: It is configured at 2.4 GHz but non-overlapped with IEEE 802.15.4 carriers. The choice of 2.4 GHz is only for test and demonstration purpose to be able to transmit the RF signal over the air. The sample rate is set to 1.92 Msps for one channel.
    - LoRa: It is configured at the EU 863-870 MHz unlicensed band. The sample rate is set to 1Msps to support up to 2 channels, e.g. at 863.1 and/or 863.3 MHz.
  - **Radio Head PC**: This PC is used to fast prototype and verify the radio head functionalities required for this use case. All functionalities are implemented in software in the PC, instead of in hardware (e.g. in the FPGA in USRP). Although it adds some latency, it is sufficient for the PoC purpose. Basically, this PC is used to bridge between the USRPs having USB interfaces and the Edge PC having an Ethernet interface. As shown in Figure 5-1, three USB ports on the PC are used to connect to the three USRP B210 respectively, while the Ethernet port of the PC is connected to the Edge PC via an Ethernet Switch (not shown in Figure 5-1).
  - **Edge PC**: This PC represents the edge infrastructure, i.e. the EFS infrastructure, in this PoC testbed, to have a minimum setup which is easy to carry for demonstrations. As shown in Figure 5-1, the Edge PC hosts the developed EFS functions (i.e. edge functions for each individual RATs), EFS application (i.e. Interference Analyzer), EFS service platform (i.e. MQTT broker).
  - **Ethernet network**: A Gigabit Ethernet switch is used in the testbed to represent to the transport network between the radio heads and the edge infrastructure. Figure 5-1 also shows the current IP configurations of the Radio Head PC, the Edge PC and the Docker containers hosted in these two PCs.

Figure 5-2: PoC #4 Edge-side hardware setup (without Ethernet switch)

- **Terminal side**: the following three types of IoT devices are used as the terminals/UEs for three supported RATs, respectively, as shown in Figure 5-3.
    - **IEEE 802.15.4**: Zolertia FIREFLY [30]
    - **LoRa**: Pycom FiPy [31]
    - **NB-IoT**: In NB-IoT, we focus on the DL PHY. For the device side, we developed our own NB-IoT receiver emulator using an USRP and a laptop or mini PC, on which we implemented some key components of NB-IoT receiver functionalities using GNU Radio.

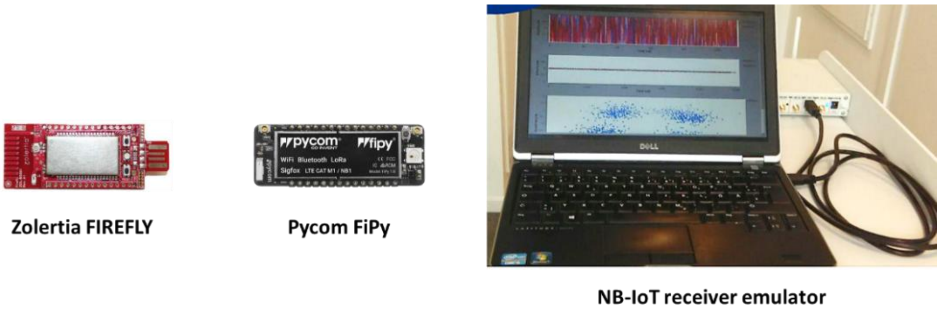


Figure 5-3: PoC #4 Three types of IoT devices used in the PoC

Table 5-1: PoC #4 Hardware Specifications Used In The Experimental Setup

Hardware Component		Specifications
Radio Head PC	Model	Intel NUC i7 RX Vega Hades Canyon VR, NUC8I7HVK
	CPU	Core i7 Quad-Core I7-8705G
	Memory	16GB DDR4 Memory 2400MHz

	Storage	512GB SSD
	Network	2 x LAN (Gigabit Ethernet) - RJ-45
	USB	5 x USB 3.0 - Type A
Edge PC	Model	Intel NUC i7 RX Vega Hades Canyon VR, NUC8I7HVK
	CPU	Core i7 Quad-Core I7-8809G
	Memory	16GB DDR4 Memory 2400MHz
	Network	512GB SSD
IoT Devices	IEEE 802.15.4	Zolertia FIREFLY
	LoRa	Pycom FiPy
	NB-IoT	Emulated by a USRP B210 and a laptop or mini PC
USRP	Model	USRP B210
	Antennas	Sub-GHz antenna for LoRa and 2.4 GHz antenna for IEEE 802.15.4 and NB-IoT

5.1.1.2 Software Setup

The focus of this testbed development is on the network/edge side to develop and virtualize multi-RAT communication stack functions as software components deployed on the Edge. For this purpose, key software components are developed and hosted on the Radio Head PC and Edge PC, respectively. More details regarding the implemented components are described in the following and the specifications of used software development environment and tools are summarized in Table 5-2 as well.

- **Radio head software:** Three radio head software programs are developed using GNU Radio for three RATs, respective, to interface USPR and Ethernet. They are virtualized as individual Docker containers, as shown in Figure 5-1. The main functionalities implemented are described as follows.
    - **USB-Ethernet conversion:** This performs interface adaptation between USB interface and Ethernet interface. For transport protocol, both TCP (using ZeroMQ) and UDP are supported. This is implemented for all three RATs.
    - **UL IQ-sample filtering:** This is implemented for IEEE 802.15.4 and LoRa. The purpose is to make the fronthaul traffic load adaptive to the air-interface traffic load. The radio head software doesn't send any data when there is no UL traffic received. The current method is based on preamble detection. Some RSSI-based methods have been also evaluated. But the performance is not as good as the method based on preamble detection, while the complexity of the preamble-based method is higher. More details about the comparison can be seen in D2.2 [26].
    - **Multi-channel multiplexing/de-multiplexing (DL/UL):** This is implemented for IEEE 802.15.4 (both UL and DL) and LoRa (UL only). In DL, it multiplexes multiple

baseband channels into a wider bandwidth trunk in frequency domain. Then the multiplexed wideband IQ samples are sent to the intended USRP. In UL, wideband IQ samples are received from the USRP. They are de-multiplexed into multiple single-channel IQ samples. Each single-channel IQ samples are sent separately to the intended edge software for demodulation. More details can be seen in D2.2 [26].

  - **Fronthaul compression**: This is implemented in NB-IoT. It significantly reduces the FH bit rate by moving the cyclic-prefix function to the radio head. More details can be seen in D2.2 [26].
  - **IQ-data publishing**: This is implemented for IEEE 802.15.4. The idea is to publish IQ samples as an EFS service. The radio head software program of IEEE 802.15.4 listens to the air interface and publishes the wideband IQ samples as the IQ/Data service to the EFS service subscribers via the MQTT broker using the MQTT protocol.

- **Edge software**: These software programs represent all key components of an EFS system of a 5G-CORAL system. They are all virtualized as Docker containers hosted in the Edge PC. Therefore, this testbed is designed as an EFS PoC for the Multi-RAT IoT use case.
  - **Communication stacks**: These software programs are developed as the EFS functions for different RATs.
    - **IEEE 802.15.4**: The 802.15.4 setup uses the network stack provided by Contiki-NG [32] running in native configuration. CSMA is used for the MAC layer, uIP in combination with 6LoWPAN network adaptation layer is used as the network layer. For our demonstration, we use a LWM2M server as our application layer. The 802.15.4 PHY layer is designed in GNU Radio [33] software framework. The PHY interacts with the Contiki network stack through a UDP socket-based radio driver. We adapted the GNU Radio implementation to the RH-Edge setup and integrated the Contiki process, allowing full stack communication.
    - **LoRa**: We adapted the GNU Radio LoRa implementation [34] to our RH-Edge setup. To minimize the front haul link utilization, preamble detection techniques for different SFs is introduced. Our setup also incorporates multi-channel support together with SF characterization techniques for supporting multi-channel gateway across multiple SFs. The implementation supports L1 and simple application layer sending and receiving short messages.
    - **NB-IoT**: Some key component of the OFDM transmitter specified in NB-IoT DL PHY (i.e. L1) have been developed using GNU radio following the NB-IoT numerology with NPSS, NSSS and NRS implemented. A convolutional code of coding rate ½ is used. For performance evaluation purpose, full-buffer dummy data are transmitted. A chat application is developed for demonstration purpose supporting sending text messages from a mobile phone using Telegram App to the receiver, where the transmitter is hooked up to a Telegram bot to forward the message from the transmitter to the receiver.
  - **MQTT broker**: This software component is used as the EFS service platform. The Multi RAT IoT provides an EFS service in the form of IQ samples which can be used by interference analyzer application. The EFS service publishes data to the EFS service

platform which is implemented as an MQTT broker. We are using MQTT V3.1.1 in our service platform. The interference analyzer application subscribes to this EFS service application through our MQTT broker.

- **Interference Analyzer:** This software program is developed as an EFS application example utilizing IQ data. The Interference Analyzer subscribes the IQ/data service via the MQTT broker using MQTT protocol. After receiving the wideband IQ samples from the IQ/data service, Interference Analyzer program analyze the IQ sample and provide the insights about the air interface noise and interferences.

TABLE 5-2: POC #4 SOFTWARE SPECIFICATION USED IN THE EXPERIMENTAL SETUP

Software Component		Specifications
Radio Head	OS	Ubuntu 18.04
	GNU Radio	3.7.13.5
	USRP Hardware Driver (UHD)	3.15.0
	Docker	19.03.0-rc3
	MQTT client	Mosquitto 1.4.15
Edge	OS	Ubuntu 18.04
	GNU Radio	3.7.13.5
	Docker	19.03.0-rc3
	MQTT broker	Mosquitto 1.4.15

5.1.2 Required 5G-CORAL Building Blocks

As stated before, the focus of the PoC is to develop and showcase the EFS design of a 5G-CORAL system for the Multi-RAT IoT use case. Table 5-3summarizes how the developed software components map to 5G-CORAL EFS building blocks. The current PoC doesn't comprise of any OCS components. But an OCS (e.g. using Kubernetes) can be added later in the system in the future development.

TABLE 5-3: POC #4 MAPPING TO 5G-CORAL EFS-RELATED BUILDING BLOCKS

Area	Component's Name	Description
EFS Function	IEEE 802.15.4 communication stack	IEEE 802.15.4 full stack implementation
EFS Function	LoRa communication stack	LoRa UL L1 implementation
EFS Function	NB-IoT communication stack	NB-IoT DL L1 implementation

EFS Service platform	MQTT broker	Handle sub/pub for EFS services
EFS Service	IQ/Data service	Wideband IQ samples containing the information of the air-interface noise and interferences
EFS Application	Interference Analyzer	Analyze the IQ sample and provide the insights about the air interface noise and interferences
Non-EFS	IEEE 802.15.4 radio head software	USB-Ethernet conversion, UL IQ-sample filtering, multi-channel multiplexing (DL) and demultiplexing (UL), IQ-data publishing
Non-EFS	LoRa radio head software	USB-Ethernet conversion, UL IQ-sample filtering, multi-channel demultiplexing (UL)
Non-EFS	NB-IoT radio head software	USB-Ethernet conversion, fronthaul compression

5.2 PoC KPIs: Measured Values for the Performance Metrics

The key idea of this use case is to move the RAT-specific transceiver functions to the Edge infrastructure, as shown in Figure 5-4. In the legacy design, the radio head functions and transceiver functions are embedded in the same device, e.g. access point or gateway. We refer to this design as standalone setup. In the 5G-CORAL design, we split these two types of functions where radio head functions are kept in radio head and move the transceiver functions to the Edge. In this design, multiple RATs can be simultaneously supported at one shared Edge infrastructure. We refer to this design as split setup.

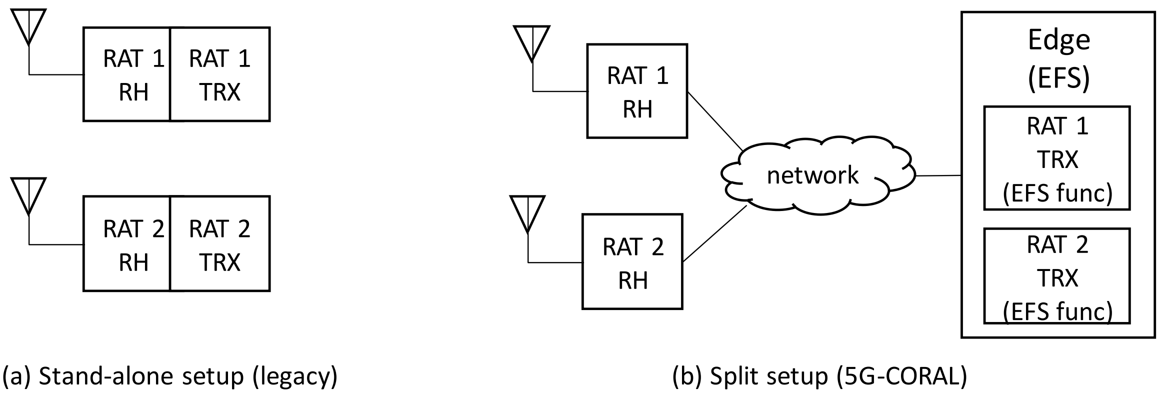


FIGURE 5-4: POC #4 ILLUSTRATION OF THE LEGACY STAND-ALONE SETUP AND THE 5G-CORAL SPLIT SETUP WITH 2 RATS

In our testbed, we can configure the system to operate in a standalone setup or a split setup. In the standalone setup, the Radio Head PC hosts both radio head software and edge software. As presented previously, in the split setup, radio head software programs are hosted in the Radio Head PC while edge software programs are hosted in the Edge PC. In D4.1 [2] and D2.2 [26], we have shown that the split setup achieves similar performance results as the standalone setup for IEEE 802.15.4 and NB-IoT individually. It verifies the feasibility of the split setup for individual IoT RATs.

After the individual RAT development and verification, we focus on the integration of multiple RATs in the split setup in this deliverable in WP4. Furthermore, the testbed is also developed to showcase EFS services and applications. We will show that the IQ/Data service can be used by the Interference Analyzer to help understand more the surrounding radio conditions in terms of noises and interferences, which are important information for interference mitigation and radio resource management. In the following, we will first describe the measurement methodology for the concept verification and then present the measurement results.

5.2.1 Measurement Methodology

The following two types of tests are performed for evaluation and verification of the EFS in 5G-CORAL.

1. Integration test

The first purpose of this testbed is to evaluate and verify the EFS concept that multiple IoT RATs can run simultaneously in the split setup, i.e. in a shared Edge infrastructure. Ideally, the performance of each IoT RAT when running with other IoT RATs simultaneously should be the same or similar as that when it runs itself only in the Edge. To evaluate this, we perform an integration test with the 3 RATs in the following steps.

- Step 1: Initiate IEEE 802.15.4 container, perform IEEE 802.15.4 measurement and record the measurement results. In this step, it is a single-RAT test.
- Step 2: Initiate LoRa container, perform both LoRa and IEEE 802.15.4 measurements simultaneously and record the measurement results. In this step, it is a two-RAT test.
- Step 3: Initiate NB-IoT container and run full DL traffic. Then perform both LoRa and IEEE 802.15.4 measurements simultaneously and record the measurement results. In this step, it is a three-RAT test. In this case, NB-IoT is more used as a background workload.
- Step 4: compare the measurement result of Step 1, 2 and 3.

Table 5-4 shows the metrics that will be measured in each step presented above.

TABLE 5-4: PoC #4 MEASUREMENT METRICS

RAT-specific metrics	Common metrics
○ IEEE 802.15.4: ping delay, packet error/drop rate (PER) ○ LoRa: PER vs SNR	○ CPU usage ○ Memory usage ○ Network/link usage

2. Interference Analyzer test

The second purpose of this testbed is to show an example of Interference Analyzer application and how it can utilize the EFS service of IQ/Data published by the IEEE 802.15.4 radio head software via MQTT. In this test, we will generate certain types of interferences, e.g. WiFi, Bluetooth and show the Interference Analyzer is able to receive the IQ/data via MQTT and perform the analysis.

5.2.2 Measurement Results

The following subsections show the detailed measurement results for the integration test and the Interference Analyzer test.

5.2.2.1 Integration test results

Figure 5-5 represent the ping results in terms of CDF against the RTT values while mentioning the PER findings in each RAT combination. We observe an increase in PER value while including additional RAT. As the execution of IEEE 802.15.4 is highly computing intensive, the packet error rate (PER) increases slightly due to the resource contention at Radio Head PC (shown in Figure 5-7(A)) in case of multi-RAT, which is not part of the Edge. The MAC retransmission timeout for Acknowledgement of successfully transmitted packets is set to 100ms, which explains the knee in the curves around that 100ms point in Figure 5-5. In reality, the radio head functions will be implemented in hardware, e.g. FPGA, DSP etc. This would not be an issue. Therefore, the results show that IEEE 802.15.4 can be realised with same performance in presence of other RATs in our Multi-RAT IoT scenario.

Figure 5-6 shows the PER vs SNR results for LoRa in a multi-RAT scenario. In this experiment, the LoRa received signal power is kept constant at -28dBm and noise is added in the radio head software to for SNR configuration to obtain a specific value of SNR. The Pycom FiPy continuously transmits 500 data packets with SF = 7 and at the edge we count the number of received packets for different SNRs. Figure 5-6 shows that the results are very similar among three scenarios for single, two and three RATs running simultaneously, respectively. It indicates that the performance of our LoRa receiver is not affected by the presence of other RATs with all the combinations showing close to 0% PER when SNR is set greater than -10dB.

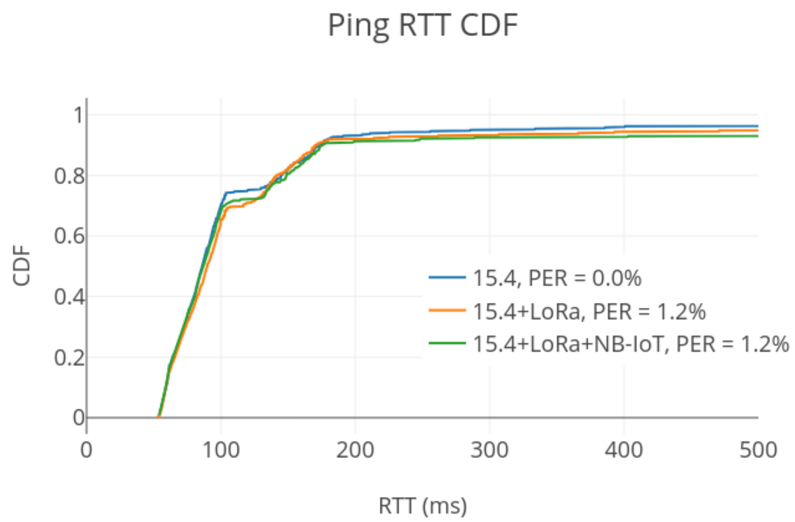


FIGURE 5-5: PoC #4 COMPARISON CDF AND RTT OF PING, AND PER RESULTS FOR IEEE 802.15.4 IN MULTI-RAT SCENARIO

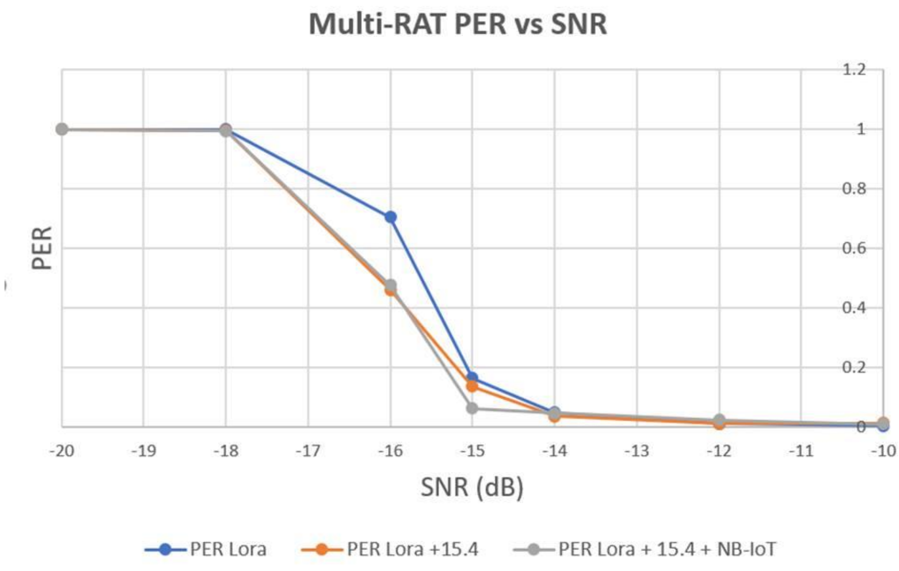
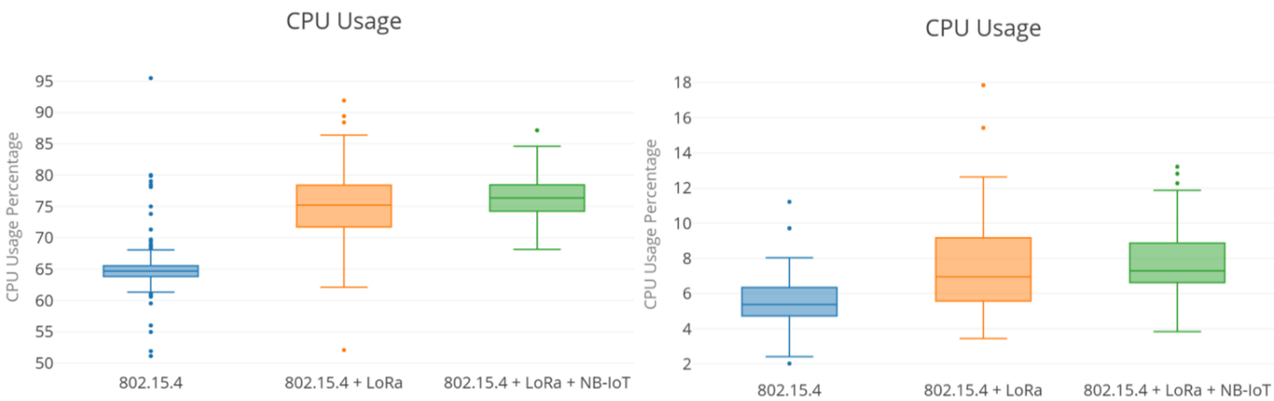


FIGURE 5-6: COMPARISON PER VS SNR FOR LORA IN MULTI-RAT SCENARIO, SF = 7

The CPU usage for the Radio Head PC, shown in Figure 5-7 (A), remains below 80% with all the RATs running simultaneously, while the CPU usage for IEEE 802.15.4 Radio Head can reach more than 60%. Of the three RATs, IEEE 802.15.4 is the most computation intensive mainly because of higher sampling rate coupled with processing intensive filtering mechanisms in the Radio Head for supporting 3 channels simultaneously. We highlight that the CPU usage for Radio Head PC can become the bottleneck in our setup as we increase number of RATs supported. But the CPU usage results at Edge PC is quite different and acceptable as shown in Figure 5-7 (B). The maximum CPU usage remains under 20% with all three RATs running simultaneously, of which IEEE 802.15.4 edge function being most compute intensive with maximum CPU usage of around 8%. This is because the IEEE 802.15.4 edge function instantiates and executes multiple IEEE 802.15.4 PHY and MAC instances for multi-channel operation. We see an increase in the CPU usage with number of RATs deployed simultaneously, which can be explained by the increased number of Docker containers running at Radio Head PC and Edge PC.

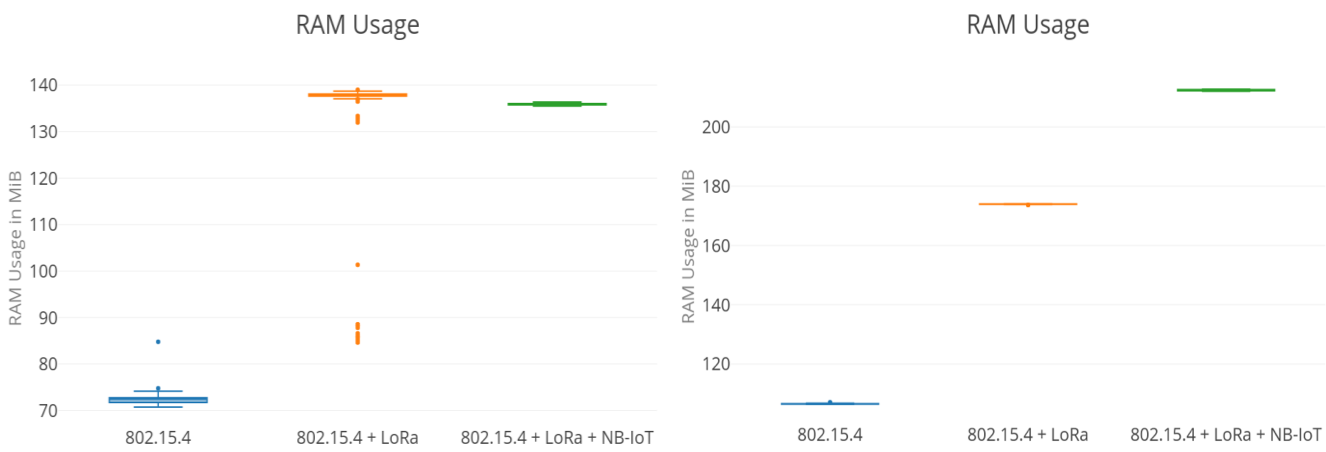


(A) RADIOHEAD PC **(B)** EDGE PC

FIGURE 5-7: POC #4 COMPARISON OF CPU USAGE AT RADIO HEAD & EDGE PC IN MULTI-RAT SCENARIO

The memory overhead of running multiple RATs is negligible as with all three RATs combined consumes less than 140 MB at Radio Head PC as shown in Figure 5-8 (A). We observed that our

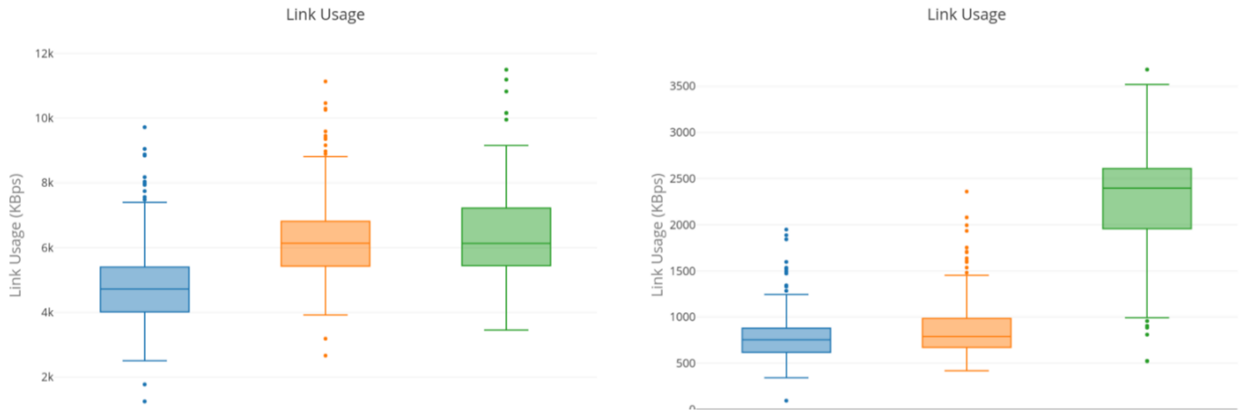
RAM usage doubles with the introduction of the LoRa RAT. This can be attributed to large buffers in the preamble detection algorithm as well as 20 MB of RAM allocated to the LoRa USRP buffer to prevent overflow errors. LoRa radio head in standalone setup utilizes more than 25% CPU resource with the 1MSPS sampling rate. But because of the high processing workload of the IEEE 802.15.4, LoRa enters resource contention in multi-RAT scenario and results in dropped radio samples. The increased buffer size helps us in ensuring negligible dropped samples while introducing a buffering delay of 200ms which is acceptable as per LoRa specifications. Similarly, the RAM usage at Edge PC, shown in Figure 5-8 (B), remains little under 200 MB for all 3 RATs running simultaneously. Our results show that with our current processing and memory resources it is very much possible to deploy multiple RATs in our PoC ensuring reliable performance.



(A) Radio Head PC (B) Edge PC

Figure 5-8: PoC #4 Comparison of RAM Usage at Radio Head & Edge PC in multi-RAT scenario

As explained before, in our experiment, LoRa contributes to uplink, NB-IoT contributes to downlink heavily, and finally, 802.15.4 contributes to both uplink and downlink. We observed, in Figure 5-9 (A), that the uplink link usage remains below 10MBps even though we are processing 62MBps of actual sample data. This is achieved with the use of preamble detection which drastically reduces the uplink link usage. It can be seen that with addition of LoRa RAT to IEEE 802.15.4 the uplink usage increases, whereas the inclusion of NB-IoT has minimal impact being downlink heavy. Similarly, in Figure 5-9 (B) it I evident that LoRa has the minimum impact to the downlink usage whereas with addition of NB-IoT the downlink usage increases rapidly, because NB-IoT continuously streams IQ samples.



(A) Uplink Link Utilization (B) Downlink Link Utilization

Figure 5-9: PoC #4 Link Usage at Edge PC in multi-RAT Scenario

5.2.2.2 Interference analyzer test results

In this setup, we use the MQTT publisher on the Radio Head PC to publish IQ samples to the MQTT broker with MQTT topic IQ/Data. The purpose of this test is to exemplify the possibility to use the IQ samples subscribed to capture the information about interferences.

In this test, we use Bluetooth 4.1 as an example of an interference source to IEEE 802.15.4. Particularly, we create Bluetooth traffic by streaming music from an Android 9.0 smartphone to a wireless headset. At the 802.15.4 RH software, IQ samples are collected for 1 second and publish these samples to the MQTT Broker hosted in the Edge PC. On the Edge PC, we use an MQTT Subscriber/Client to receive the samples and write them to a log file, which will be processed offline for interference analysis purpose.

For the logged samples, we first group the samples into 625 microseconds segments (Bluetooth slot duration) and calculate the PSD (power spectrum density) for the segments. As examples, Figure 5-10 and Figure 5-11show the PSD of slot 1 and slot 2, respectively. The Bluetooth signal and its frequency hopping pattern are clearly captured. It is shown that the signal on each channel has 1 MHz bandwidth (defined by the bandwidth which is higher than the level of 20 dB below the peak). In slot 1, channel 69, 74 and 77 are used, while slot 2 changes to channel 70 and 72. These observations (signal bandwidth and frequency hopping) can help tell the interferences are from Bluetooth. Therefore, these results showcase one use case for using the IQ data service in EFS.

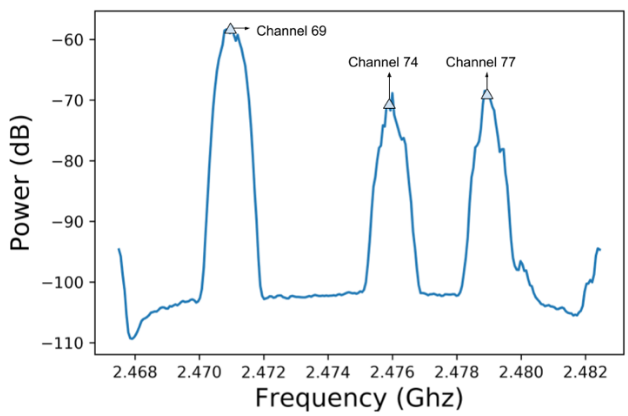


Figure 5-10: PoC #4 PSD for slot 1

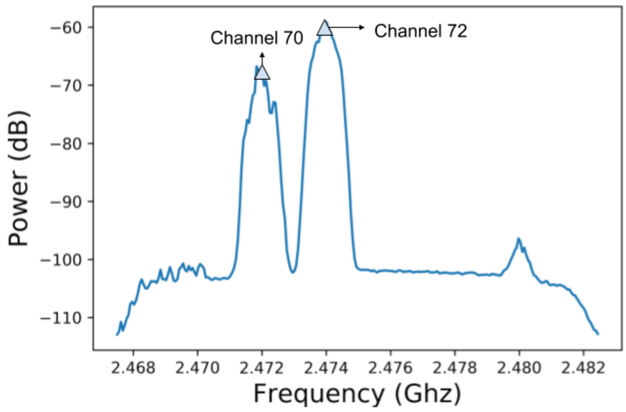


FIGURE 5-11: PoC #4 PSD FOR SLOT 2

5.3 Year 2 Achievements and Future Directions

The following summarizes the achievements regarding the PoC testbed development and tests in Year 2 of the project period.

- LoRa Radio Head and Edge software programs are developed and added into the integrated testbed.
- Multi-channel capability has been implemented in IEEE 802.15.4 Radio Head and Edge software programs.
- IEEE 802.15.4, NB-IoT and LoRa software programs are "dockerized" and integrated in one setup, where they can run simultaneously.
- An MQTT broker is added to the testbed, acting as the EFS service platform.
- IQ/Data service are implemented and integrated in the IEEE 802.15.4 Radio Head software, which can publish wideband IQ samples to the MQTT broker following the MQTT protocol.
- Interference Analyzer application is developed on the Edge. It subscribes the IQ/Data service from the MQTT broker using the MQTT protocol and perform noise and interference analysis.
- Final integration tests and Interference Analyzer tests are performed.

The current PoC development focuses on feasibility study and performance evaluation of the EFS part with a small-scale setup. In the future, the PoC can be further extended with OCS functions and evaluated in a larger setup. The following summarize some possible future directions.

- Orchestration: Add an orchestrator, e.g. Kubernetes, to the testbed. Investigate and implement new orchestration features.
- Machine learning: Investigate possibilities to incorporate machine learning techniques and capabilities into the testbed and explore new applications leveraging the potential of the big amount of data available on a 5G-CROAL system.
- Large scale testbed: Scale up current testbed with more computing resources, more radio heads and more devices, to test the system performance for a large-scale deployment.

5.4 Demonstration activities

During this period of year 2, we have shown one 5G-CORAL internal demo (i.e. Taiwan mid-term trial demo) with invited Taiwanese companies in Taiwan on November 2nd, 2018, see Figure 5-12, and one public demo in ICT 2018 in Vienna on December 5th, 2018, see Figure 5-13. In both demos, we showed the feasibility to virtualize individually IEEE 802.15.4 and NB-IoT on the Edge. The following show the photos of the demo events.

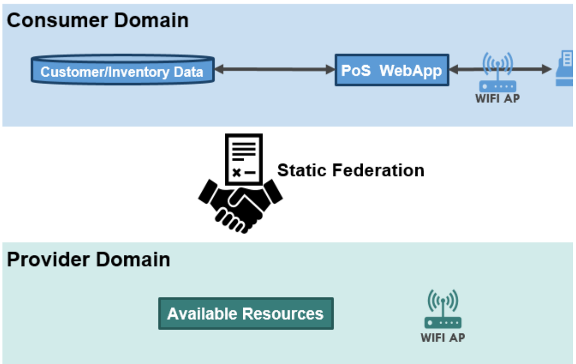
Next demo has been scheduled in the end of October in Taiwan in the final 5G-CORAL trial. In this demo, we will show a complete integration demo with 802.15.4, LoRa and NB-IoT running simultaneously and will also showcase the Interference Analyzer application.

FIGURE 5-12: POC #4 MULTI-RAT IOT DEMO AT NANGANG SHOPPING MALL IN TAIPEI ON NOVEMBER 2ND, 2018

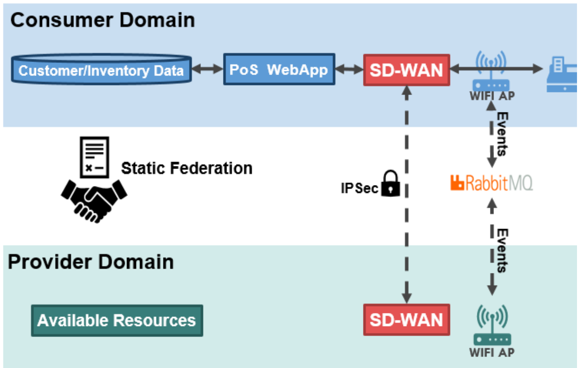
FIGURE 5-13: POC #4 MULTI-RAT IOT DEMO AT ICT 2018 IN VIENNA ON DECEMBER 5TH, 2018

6 PoC #5: SD-WAN

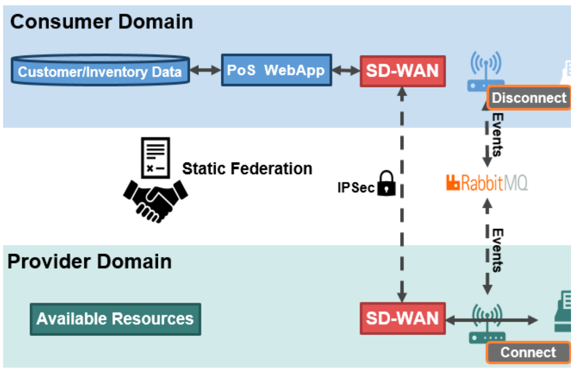
The PoC described in this section focuses on demonstrating the concept of **federation** [18] and the benefits that it brings to the users in the Shopping Mall scenario. More specifically, the PoC exposes the mechanisms needed to allow different owners of different network domains to share their available resources (with a previous negotiation and agreement, i.e. static federation) in order to optimize the performance of the network and the Quality of Experience (QoE) of the end users.



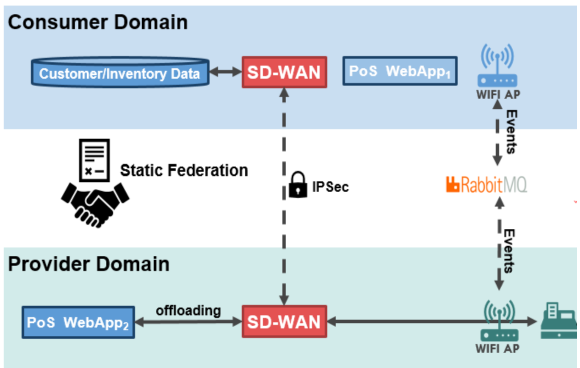
A) PHASE 0: TWO INDEPENDENT EFS DOMAINS.



B) PHASE 1A: INTERCONNECTION BETWEEN BOTH DOMAINS SET-UP



C) PHASE 1B: END USER MOVES FROM CONSUMER DOMAIN TO PROVIDER DOMAIN



D) PHASE 2: OFFLOADING OF POS WEB APPLICATION AND QOE IMPROVEMENT

FIGURE 6-1: POC #5 PHASES

The use case chosen for this demonstration is an on-line shopping application (Point of Sale Service), as it fits inside the requirements of the SD-WAN Federation use case, I) requiring a secure communication path between both domains, II) PoS end users need to switch frequently between domains (Mobility) and III) QoE of end users needs to be maintained to sustain the business case. The initial components of the scenario is shown in Figure 6-2 a). At a high level, two independent domains are represented: the **consumer domain**, which is the home domain for the PoS Service, hosting the whole on-line shopping application, and the **provider domain**, which owns available computing, networking and storage resources. The on-line shopping application is divided in two components: a customer/inventory **database** and a Point of Sell (PoS) **web application**, both running in two different virtual containers in the EFS. Finally, a Wi-Fi access point which is located in the EFS to provide network access to the end users.

As their names suggest, the consumer domain is going to take advantage of the EFS resources granted by the provider domain, according to the conditions agreed in a previous negotiation (static federation agreement), which can be in the form of capital benefits, permissions to use certain resources of the other domain, etc.

Figure 6-1 illustrates the phase of the PoC when both domains have reached the static federation agreement, and therefore the new elements that are necessary to enable the federation process are represented. I) Software-Defined Wide Area Network (SD-WAN) functions are deployed in both domains, controlled by an **SD-WAN manager** instantiated in both domains inside the OCS, which oversees the routing and network configuration between the two domains. II) A **mobility service** is implemented and installed in the OCS of the consumer domain, using the RabbitMQ as publish subscribe data bus. The mobility service listens to connection and disconnection events on the WiFi access points of each network domain, with the purpose of detecting the movement of the end users from one domain to the other. III) An IPSec tunnel is established between both domains connecting both domains over the WAN securely, including the control and data planes. IV) At this point the consumer domain OCS has total control over the virtual resources shared, both the data and control plane are steered towards the consumer domain, enabling the consumer domain to instantiate virtual resources over the provider domain.

In the next phase, depicted in Figure 6-1 c), the user moves from the consumer domain to the provider domain. I) The mobility service detects the roaming and triggers the SD-WAN manager to perform the necessary network configuration so that the communication between the end-user and the PoS web application is not disrupted (roaming). II) The routing rules and DHCP parameters are changed so that the end-user is able to maintain the same IP address as it had in the consumer domain. Therefore, the federation mechanism, along with the SD-WAN orchestration, has the advantage of not disrupting the service consumed by the end user when he moves between different and independent network domains.

The last phase of the PoC is illustrated in Figure 6-1 d). I) The WAN link experiences a downgrade of the network conditions, worsening the Quality of Experience (QoE) of the end user, measured in terms of **response time**, which is defined as the **latency** between the user request and the arrival of its correspondent response from the PoS WebApp. II) When the network conditions of bitrate and latency on the WAN link are constrained and limited, the response times of the web application increase. III) This issue is mitigated by ***offloading*** the PoS WebApp from the consumer domain to the provider domain. That is, the consumer domain leverages the resources negotiated in the federation agreement to instantiate a container to run the web application in the provider domain, bringing it closer to the end-user and thus improving his QoE. IV) the SD-WAN manager modifies the routing so that the service is not disrupted, establishing the new IP connectivity between the end user and the PoS web application, and between the PoS web application and the customer/inventory database.

After concluding the different phases of the PoC, Figure 6-1 maps the performance metrics explained in this section with the user benefits obtained from the federation of resources, which will be further demonstrated and measured in terms of the overall latency, throughput and QoE of the offered service. Furthermore, the network management mechanisms implemented in this PoC demonstrate the capability of offloading an application closer to the end users, which are moving across multiple domains.

TABLE 6-1: POC #5 PERFORMANCE METRICS AND USERS BENEFITS

Performance metrics	Description	User Benefits

Latency	Round trip time latency between the PoS Terminal and the PoS WebApp, and the PoS WebApp and the PoS Customer and Inventory Database.	Maintained latency during roaming and federation scenarios.
Throughput	Maximum throughput available between the PoS Terminal and the PoS WebApp.	Maintained throughput during roaming and federation scenarios.
Quality of Experience (QoE)	The overall experience the end user is experiencing from the PoS Service. Measured as the PoS response time.	Assured QoE during roaming and federation scenarios.

6.1 PoC Testbed Design

6.1.1 Overview of Integrated Components

The testbed setup for this PoC is illustrated in Figure 6-2 . As it can be observed, the consumer domain and the provider domain are hosted by two independent *Dell latitude e5550* computers, connected via LTE to a central cloud which emulates the WAN connection (in yellow) between domains. Inside both *Dell* computers, there is a dedicated OCS and EFS hosted by a *KVM* based virtual machine. The OCS and EFS are connected through a management network (in green) inside a single domain, composed by a Linux virtual bridge and Linux virtual Ethernet interfaces. Additionally, a virtual Wi-Fi access point function is deployed for each EFS in both domains, which has the role of providing the access network and entry point for end users to both consumer and provider domains.

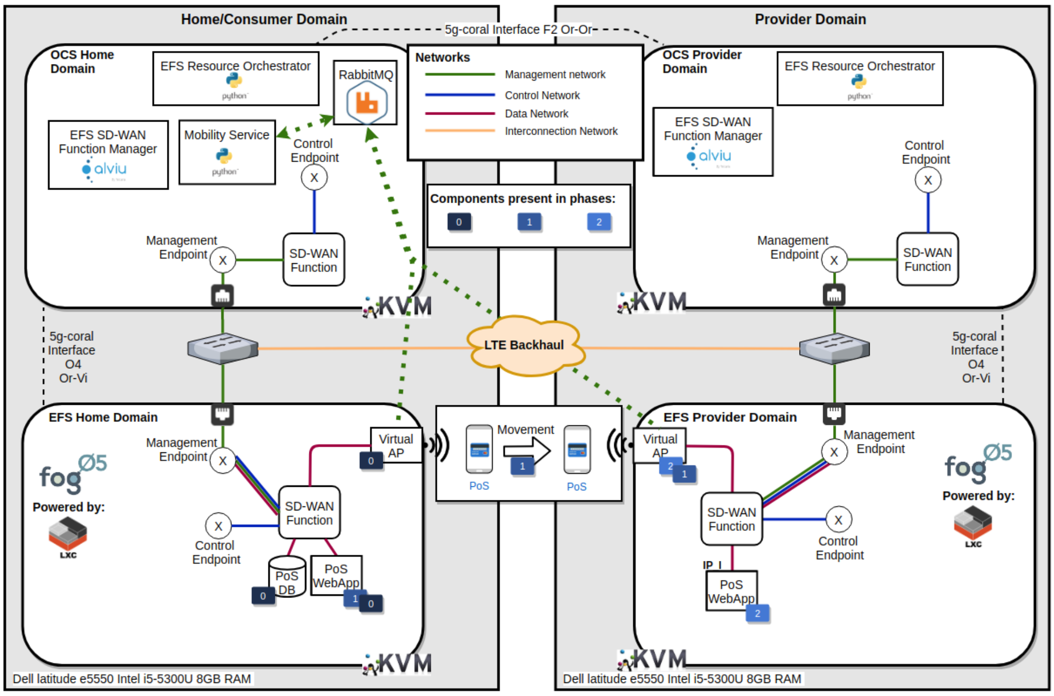


FIGURE 6-2: POC #5 TESTBED DESIGN

The data and control plane between domains are managed by the SD-WAN Functions, which are controlled by the SD-WAN Manager located at the OCS. Furthermore, the SD-WAN Manager also controls the WAN communication between domains.

6.1.1.1 Hardware setup

Table 6-2 describes in detail the hardware components used for this PoS and its specifications.

TABLE 6-2: POC #5 HARDWARE COMPONENTS SPECIFICATIONS

Component	Model	Characteristics
Computer hosting the consumer domain	Dell Latitude E5550	I5-3500U 8GB RAM Ubuntu 16.04
Computer hosting the provider domain	Dell Latitude E5490	i5-8250U 7.7GB RAM Ubuntu 18.04
Wi-Fi card of the consumer domain's virtual AP	Intel Wireless-AC 7265 Dual Band	IEEE 802.11g (2.4 GHz)
Wi-Fi card of the provider domain's virtual AP	Intel Corporation Wireless 8265 / 8275	IEEE 802.11g (2.4 GHz)
LTE USB modems	HUAWEI E3372	LTE FDD: Cat4 DL: 150 Mbps / UL: 50 Mbps @20 M BW

6.1.1.2 Software setup

Inside the EFS, the SD-WAN Function, PoS database and the PoS WebApp are hosted by *LXC* containers. The communication between the PoS database, the PoS WebApp, the Wi-Fi AP and the end user is performed within the data plane network.

The lifecycle of the LXC containers is managed by the EFS Resource Orchestrator located in both OCS, leveraging the Fog05 service as virtual infrastructure manager (VIM). Additionally, a mobility service is located on the OCS of the consumer domain, which detects end user movements between domains, making use of the RabbitMQ publish subscribe data bus connected to the virtual APs.

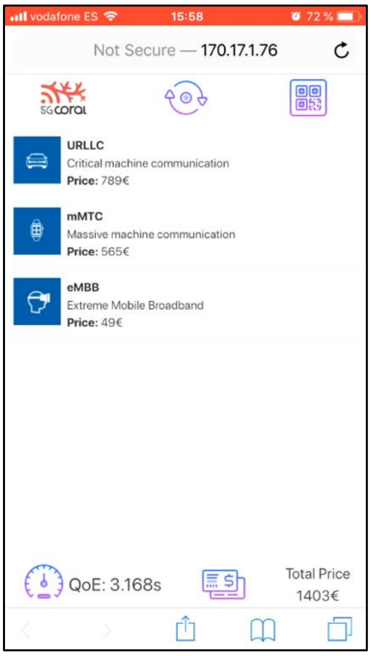


FIGURE 6-3: POC #5 POS WEBAPP DASHBOARD

Finally, the frontend component of the PoS WebApp measures the **response times** of every user interaction and displays it on the bottom left corner of the user screen, see Figure 6-3. In this way, it is possible to gather the average values of the response times in the different phases of the PoC, allowing to quantify the improvements and benefits of the federation and offloading of resources demonstrated in the PoC.

6.1.2 Required 5G-CORAL Building Blocks

The main objective of this PoC is to demonstrate how 5G-CORAL edge and fog computing system is beneficial to the SD-WAN use case and showcase novel technologies which the SD-WAN could bring to the system to enhance its federation interfaces, provide seamless roaming and offloading capabilities, while integrating nicely with 5G-CORAL architecture [26]. Figure 6-4 illustrates how the SD-WAN use case is integrated within the 5G-CORAL architecture. From top to bottom, the OCS Resource Orchestrator has the global view of resources inside its controlled domain, including possible resources from other domains which it could extend for specific use cases in different mobility scenarios. Sequentially, the reader can find the corresponding life cycle managers for the two main services, the SD-WAN and the PoS Service. The necessary virtual resources from the compute, storage and networking domains are managed by the OCS VIM, which is composed by two components; I) Fog05: which manages the storage and computing resources to further insatiate the PoS Applications and II) OCS SD-WAN Function Controller: managing the network virtual resources, such as the virtual switches where the SD-WAN function will be instantiated.

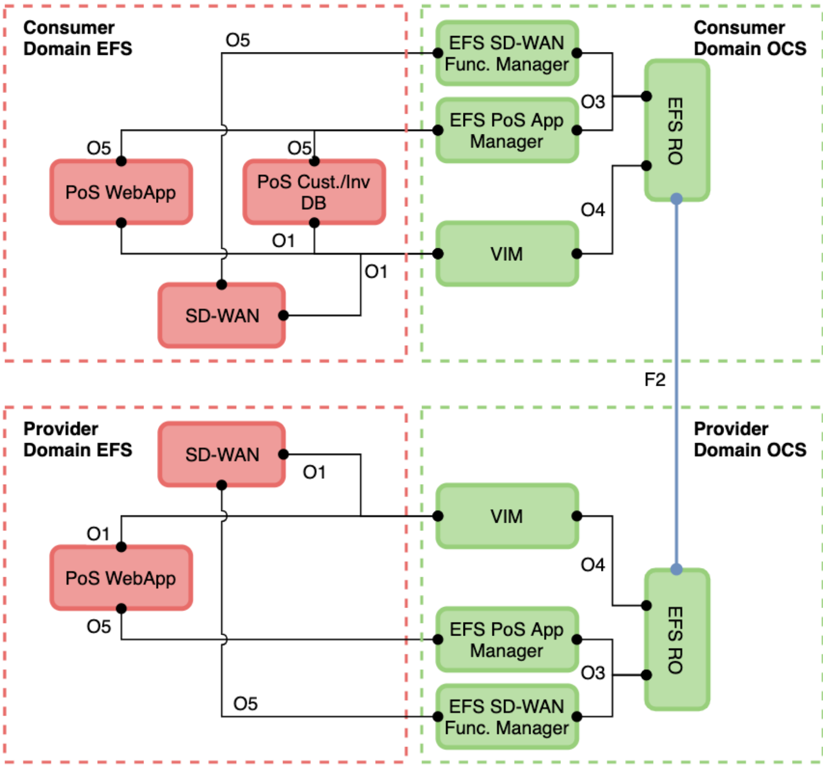


FIGURE 6-4: POC #5 RELATION BETWEEN POC, EFS AND OCS

Table 6-3 lists the necessary 5G-CORAL building blocks for the SD-WAN PoS use case, which includes all components types from both the EFS and OCS, with the exception of the OCS Stack Orchestrator, as the arbitration functionality is very limited in this use case, and can be easily handled by either the OSS/BSS or by the OCS Resource Orchestrator.

TABLE 6-3: POC #5 REQUIRED 5G-CORAL BUILDING BLOCKS

Area	Component's Name	Description
EFS Function	SD-WAN	EFS Function implementing wide range of network services, such as DHCP, ARP, DNS, Layer 4 Firewall, End to End Secure Tunneling. Network services are provided by the SD-WAN Controller.
EFS Function	Virtual AP	EFS Function implementing AP network function, to provide Wi-Fi connectivity to users at range.
EFS Service	Host Mobility Service	EFS Service leveraging the EFS platform, which subscribes to APs raw context events to further analyze and detect when a user has moved from AP, publishing the result into an specific queue from the EFS service platform.
EFS Service Platform	RabbitMQ	EFS Service Platform which leverages on RabbitMQ MQTT data broker, to forward messages from EFS Services to EFS Functions and Applications.

EFS Application	PoS Web Application	EFS Application containing the business logic and static content in the PoS Service.
EFS Application	PoS Customer and Inventory Database	EFS Application containing the Customers and the Inventory data database from the PoS Service.
OCS VIM	Fog05	OCS VIM component which manages the infrastructure and virtual resources of fog nodes deployed for the PoS Service.
OCS VIM	OCS SD-WAN Function Controller	OCS VIM component in charge of managing the set of virtual network resources intra and inter fog nodes, leveraging on the SD-WAN function.
OCS Function Manager	OCS SD-WAN Function Manager	OCS Function Manager in charge of managing the life cycle of the SD-WAN function and the network services provided by the OCS SD_WAN function controller.
OCS Application Manager	OCS PoS Application Manager	OCS Application Manager in charge of managing the life cycle of the point of sale application, which can be further decomposed in two other EFS applications; PoS Web Application and PoS Customer and Inventory Database.
OCS Resource Orchestrator	OCS Resource Orchestrator	OCS Resource Orchestrator has the global view of the virtual or physical resources envisioned by this PoC. Additionally, the OCS Resource Orchestrator deployed at this PoC has federation capabilities at the resource orchestration layer, which allows to share resources on demand across domains.

6.2 PoC KPIs: Measured Values for the performance metrics

The following subsection will describe the KPIs which will be validated and how they will be measured. The PoC #5 will leverage the PoS service and the SD-WAN in order to deploy two full 5G-CORAL domains, where latency, throughput and QoE intra and inter domain KPIs will be measured and validated. Notice that the KPIs will be validated after the correct integration of the use case components as 5G-CORAL components in this PoC.

6.2.1 Measurement Methodology

The measurement methodology adopted to validate the use case in the deployed PoC is described in the following subsections, each one explaining the measurement methodology for each KPI: latency, throughput and Quality of Experience.

- Latency compared with Phase 0 (ideal) conditions KPI, defines the round-trip time between the PoS Terminal and the PoS WebApp. Latency will be measured for each phase by using the ping command with different intervals (1s, 0.5s, 0.1s) and packet sizes (default, 700, 1400). Latency at phase 0 will then be compared to phase 1 and 2, where we should see how results obtained from phase 2 are better than phase 1, close to phase 0 results.

- Throughput compared with Phase 0 (ideal) conditions KPI, defines the maximum throughput available between the front-end components in the PoS service. These components are the PoS Terminal and the PoS Web Application, where the most data intensive workloads can be found. Throughput will be measured by using the TCP and UDP iperf3 [35] tool, varying the UDP rate increasingly until packets are lost (identifying the maximum rate to transmit). Each test will be repeated three times and the average will be taken.

- Quality of Experience (QoE) compared with Phase 0 (ideal) conditions KPI, corresponds to the overall experience the end user obtains through the PoS Service, measured as the PoS response time and compared against a reference scenario which for this use case is Phase 0. The QoE KPI is measured at the PoS Terminal client application as explained at the introduction of section 6. QoE will be measured for all phases 3 times, taking as final result the average.

6.2.2 Measurement Results

In this section, the results obtained from the measurements are analyzed. First, the measurements collected for the latency KPI are presented in Table 6-4. In phase 0, when the PoS Terminal and the PoS Web Application are communicating within the consumer domain, the latency values oscillate around 2 ms. However, in phase 1, when the PoS Terminal is connected to the provider domain and is communicating with the PoS Web Application through the LTE backhaul, the latency values increase until reaching 123 ms, showing how the network conditions are degraded, due to the fact that the communication is performed between two different domains. Subsequently, in phase 2, when the offloading of the PoS Web Application from the consumer domain to the provider domain is realized, the latency values are reduced closer to the phase 0 values, demonstrating the benefits of the federation mechanism.

TABLE 6-4: POC #5 LATENCY RESULTS

Phase	Packet size (bytes)								
	56			700			1400		
	Interval time (ms)			Interval time (ms)			Interval time (ms)		
	1	0.5	0.1	1	0.5	0.1	1	0.5	0.1
0	7.839	2.578	2.171	2.693	2.354	1.744	2.606	2.066	1.836
1	103.167	69.629	65.719	77.988	74.844	73.075	92.675	123.245	90.088
2	6.152	2.374	1.805	2.444	2.636	2.544	3.742	4.259	2.918

Secondly, the results obtained for the throughput KPI are exposed in Table 6-5. In this case, the average throughput in phase 0 is 25.6 Mbps, whereas when the communication is performed through the LTE federation link (phase 1) the average throughput drops to 4.2 Mbps. Once again, the benefits of the federation carried out in phase 2 are shown, due to the average throughput is improved to 13.16 Mbps. Ideally, this value should be closer to the phase 0 results, but this difference may be caused by the fact that we are using different Wi-Fi cards for each domain, in addition to the stochastic nature of the network conditions in wireless connections.

TABLE 6-5: POC #5 THROUGHPUT RESULTS

	Test 1 (Mbps)	Test 2 (Mbps)	Test 3 (Mbps)	Average (Mbps)
Phase 0	32.9	22.1	22.0	25.6
Phase 1	4.29	4.28	4.23	4.2
Phase 2	13.8	12.7	13.0	13.16

Finally, Table 6-6 shows the measurements gathered for the QoE KPI. As explained in subsection **Error! Reference source not found.**6.1 the QoE is directly correlated with the response times of the interactions of the users with the PoS Web Application through the PoS Terminal. These are the values that populate Table 6-6, and it can be observed that, in average, the response time increased from 3.16 seconds to 11.71 seconds when the PoS Terminal moves from the consumer domain to the provider domain, preventing thus the correct operation of the PoS service and degrading the QoE of the user. Afterwards, leveraging the offloading and federation mechanisms, the average response time is reduced back to 3.63 seconds in phase 2, recovering the correct operation of the service and improving the QoE of the user.

TABLE 6-6: POC #5 POS SERVICE RESPONSE TIMES RESULTS

	Test 1(s)	Test 2 (s)	Test 3 (s)	Average (s)
Phase 0	3.1	3.5	2.9	3.16
Phase 1	14.8	8.3	12.04	11.71
Phase 2	4.5	3.6	2.8	3.63

Table 6-7 contains a summary of all the measurements collected, as well as a comparison of the results obtained for the different phases, in order to show the improvements achieved by the mechanisms showcased by this PoC. To do this comparison, the values obtained for phase 0 are taken as a reference (ideal conditions), because all the components are located in the consumer domain. Subsequently, the ratio between the other phases (1 and 2) and the reference phase (0) is calculated, according to the following expression:

$$Ratio = \frac{Phase\ n}{Phase\ 0}\ ;\ \ n = 1{,}2$$

In this way, it can be seen that the latency for phase 1 is in average 35.43 times higher than in phase 0. In addition, Phase 2 – Phase 0 ratio is almost 1, which means that the results of phase 2 are very close to the ideal conditions.

In case of the throughput KPI, the average throughput is 0.16 times the throughput value in the initial phase, whereas in phase 2 the average throughput is approximately half the throughput in the initial phase, as explained above, due to the nature of the Wi-Fi access connection.

Finally, the QoE in phase 1 is degraded with a ratio of 3.7 with respect to phase 0, and calculating the same ratio for phase 2, a result of 1.15 is obtained, getting closer to the targeted value (ratio = 1).

TABLE 6-7: POC #5 MEASURED VALUES FOR THE PERFORMANCE METRICS RELATED TO SD-WAN

Performance metrics	Description	Reference Values	Measured Values
Latency	Round trip time latency between the PoS Terminal and the PoS WebApp.	Phase 0 latency values are taken as a reference.	Phase 1-Phase 0 ratio: 35.43 Phase 2-Phase 0 ratio: 1.14
Throughput	Maximum throughput available between the PoS Terminal and the PoS	Phase 0 throughput values are taken as a reference.	Phase 1-Phase 0 ratio: 0.16 Phase 2-Phase 0 ratio:

	WebApp.		0.54
Quality of Experience (QoE)	The overall experience the end user is experiencing from the PoS Service. Measured as the PoS response time.	Phase 0 QoE values are taken as a reference.	Phase 1-Phase 0 ratio: 3.7 Phase 2-Phase 0 ratio: 1.15

All the measurements exposed and analyzed above, in terms of latency, throughput and QoE KPIs, have allowed to quantify the benefits and advantages that can be provided by the mechanisms and technologies showcased in this PoC, such as federation, offloading and SD-WAN.

6.3 Year 2 Achievements and Future Directions

This section summarizes the PoC #5 second year achievements and proposes several directions that may be taken in the future to continue the work that has been done.

Throughout the project, a number of challenges have been overcome. First, the initial PoS Application used (Wallace PoS) was not suitable for business logic and data functional split. Both components were tightly coupled, and even if the database could be remotely configured, the application was too sensitive to latency and was not able to handle more than 1ms latency between the business logic and database. We addressed this problem by implementing ourselves a simple PoS application, which from scratch was logically divided in two sub components (PoS WebApp and the PoS Customer and Inventory Database). Second, QR scanning was initially unreliable and faulty at the PoS WebApp, however reducing the image resolution before feeding to the QR scanning service and triggering a try harder configuration parameter at QR scanning service, improving the QR recognition.

In addition, when the remote database is located in a different subnet as the PoS Web Application, the PoS Database triggered a DNS reverse lookup, which holds the application until a 10s timeout was triggered. Deactivating the reverse DNS lookup at the PoS Customer and Inventory Database fixed the problem affecting the QoE KPI.

Furthermore, as LTE modems lacked the capability to expose their public IP addresses to the hosting machine, there is an unconfigurable NAT preinstalled at the modem. The solution is to use an intermediate VPN connection to our own private cloud, and route traffic through the cloud, which allows us to have routable IP addresses between domains, while using LTE as RAT.

Roaming capabilities were a challenge as well, as initially roaming across domains could not be handled by the VIM, a new IP address was allocated each time there was a connection from another domain. SD-WAN Function/Controller are based in SDN technologies, where basic network functions (DHCP, routing, DNS, etc.) are centralized at the controller, enabling the controller to maintain the IP address of a certain host in a roaming scenario.

Finally, current offloading capabilities were not integrated into the VIM, so they were integrated into the resource orchestrator, which has the global view of the whole set of resources, either federated or not. The resource orchestrator is able to place EFS applications more precisely where offloading capabilities are required.

In terms of future directions, several options have been identified. For example, it may be possible to explore how machine learning mechanisms could enhance the host mobility detection service to predict user movements between APs accurately. Secondly, another possibility would be to deploy the PoC in a large-scale scenario with more than two domains, while serving real customers, to test the performance metrics defined and validate if they are still held. Thirdly, enhancing the orchestration mechanisms with arbitration algorithms that could place EFS functions, applications and services optimally, in terms of energy consumption, cost, performance and QoE, which will require an advanced monitoring platform to gather and aggregate metrics (even across domains) to further analyze them and feed the orchestration components.

6.4 Demonstration activities

TELCA delivered at Taiwan'18 trial the SD-WAN video demonstration (see Figure 6-5), focusing on the Point of Sale use case, where multiple businesses in a shopping mall use Point of Sale devices to perform credit card payments. This demonstration showcased the use of SD-WAN, Edge and fog computing technologies. Instead of establishing a single VPN connection to the bank network, the PoS devices were connected to a virtual WiFi access point in the EFS that is again connected to the SD-WAN function, which finally establishes the secure WAN link to other business locations, including the bank payment gateway. The main features demonstrated were:

- Integration of the SD-WAN and WiFi AP functions in EFS
- Preliminary demonstration of orchestration and control components such as, Kubernetes, SD-WAN orchestrator and SD-WAN controller.
- Traffic steering of SD-WAN function between two different network interfaces to achieve better network performance.

FIGURE 6-5: POC #5 TAIWAN '18 TRIAL THE SD-WAN VIDEO DEMONSTRATION

TELCA showcased at EUCNC 2019 held in Valencia (see Figure 6-6), the orchestration and federation mechanisms envisioned in 5G-CORAL, demonstrating their correct operation in the SD-WAN PoS use case. The PoC showcased a Point of Sale roaming across domains, while the PoS web application follows it, offloading traffic generated from the PoS to the nearest PoS web application and maintaining all customers and inventory data in the home domain. Additionally, TELCA showcased the static federation protocol designed in WP3, which is implemented inside a preliminary implementation of the OCS Resource Orchestrator, which is mainly focused on the federation and the host mobility service in order to trigger the instantiation (offloading) of the PoS Web Application as a consequence of host migration from one AP in one domain to another.

FIGURE 6-6: POC #5 EUCNC '19 DEMONSTRATION

7 PoC #6: High-Speed Train

In recent years, high-speed rails have been deployed across many countries around the world. For instance, 2000km and 10,000km of train lines are to be deployed by 2020 in France and Spain, respectively [36]. In such moving infrastructures, several challenges are faced due to mobility such as complex signal processing, signal penetration loss, frequent handover (HO) and terrestrial signal blocking. To overcome these challenges, two-hop architecture solution is adopted to provide better signal quality for on-board users [37], [38]. In two-hop architecture, on-board small cells are deployed behind customer-premises equipment (CPE) and connected to the roadside base stations. Consequently, there is no direct interaction between on-board users and roadside base stations, which significantly reduces the number of handover events.

In train networks, users can enjoy a variety of mobile services, such as video conferencing and online gaming. To enhance user experience, multi-access edge computing (MEC) and network function virtualization (NFV) have emerged as solutions where the cloud services are brought closer to the edge of the network. On one hand, MEC virtualizes applications at the edge of the network and reduces the end-to-end delay. On the other hand, NFV decouples the network functions and applications from the underlying hardware and allows them to be implemented as software. Now that mobile services can be deployed independently from the hardware in a distributed fashion, software-define networking (SDN) enables a dynamic and responsive network to new services. SDN complements MEC and NFV especially due to the separation of the control and data planes which simplifies management, provides programmability and enhances scalability and performance.

Although the two-hop architecture and edge networks can enhance user experience, we recognize two issues in regards to mobile service continuity namely, control signaling storm [39], [40] and backhaul latency. Firstly, when a large number of train users transit from on-board a train to a station, they generate a signaling storm due to massive HO events. This massive signaling can lead to connection tear-down thus affecting user's experience. Secondly, also due to mobility, users leave the initial serving edge (i.e., on-board the train) which leads to increased latency due to backhauling. In High-speed train use case, our main contributions are summarized as follows:

- Propose a group HO scheme which exploits NFV and MEC by deploying virtualized mobility management entity (vMME) at the edge to lower signalling storm.
- Develop container pre-copy migration scheme to relocate applications in MEC environment to eliminate backhaul latency. This work outperforms the state-of-art (SoA) container migration scheme [25]

Table 7-1provides an initial list of performance metrics which will be measured through the experiments using high-speed train PoC. Additional metrics are being considered and interim results are being obtained from the performed experiments.

TABLE 7-1: POC #6 PERFORMANCE METRICS AND USERS BENEFITS PERFORMANCE

Performance metrics	Description	User Benefits
Migration time	The time requires to migrate an application from on board Fog CD to on land Fog CD due to the mobility of the user.	Improve the user experience especially when user move from on board to on land. i.e. reduce the down time

Handover Latency	Round trip time between issuing handover command from eNB and Complete of handover	Improve the user experience especially when user move from on board to on land. i.e. reduce the signal storming effect and keep the applications stable in the Handover time.

7.1 PoC Testbed Design

In LTE, mobility management entity (MME) is the main function that handles mobility control signaling. It is responsible for initiating paging and authentication of the user devices. Also, MME retains location information at the tracking area level for each user and then selects the appropriate gateway during the initial registration process. Inter-MME HO is needed especially if MME is adopted in moving networks. Inter-MME HO involves three control stages [41]. First, the source eNB initiates the HO by sending a request message over the S1-MME reference point. Second, the source MME selects the target MME and configures a messaging tunnel over a control interface called S10. Finally, the source MME transfers the configuration message to target eNB over S1 interface.

Several works have been proposed to improve the HO performance [42], [43], [44]. For example, a multicast paging procedure to alleviate the MME signaling load is proposed in [42]. Also, the MME performance with multicast and unicast paging procedures is measured. In [43], SDN is utilized to evaluate the HO performance and to reduce the jitters and packet loss. In [44], SDN is used to improve the HO performance. In [45], shared MEC is proposed to support user mobility and reduce total cost of migration among MEC during HO. However, none of the existing works addressed the signaling reduction of S10 interface during inter-MME HO. For that, we propose an enhanced Inter-MME HO scheme which initiates the HO process for QoS-based classified group before it is required.

There are two types of containerization technologies, namely system-based and application-based containerization. On one hand, system containers behave like a standalone Linux system. That is, the system container, such as Linux Container (LXC) has its own root access, file system, memory, processes, networking and also can be rebooted independently from the host system. On the other hand, the application container isolates an application from other applications running on the same host kernel and operating system. This means that the development of a containerized application with necessary libraries, configurations and dependencies does not affect other applications and also the host system.

Container migration can be classified into stateful and stateless. In stateless migration (also known as cold or offline migration), the state of the container is not preserved when the container is moved to the destination host. In the case of stateful migration (also known as live migration), the state of the container is retained when the container is resumed at the destination host. There are three types of stateful migration schemes in the literature as follows:

- **stop-and-copy**- freezes the container, checkpoints its state, copies the container image and its state to the destination then restores the state from the checkpoint [46].
- **pre-copy** - performs iterative state checkpointing while the container is running till the amount of in-memory change is at minimum, then concludes with a shorter stop-and-copy [47] Iterative checkpointing reduces the size of the final checkpoint which is performed while the container is frozen. This minimizes the time required for the final checkpoint and the time required to copy the checkpoint to destination.

- **post-copy** - performs a short stop-and-copy to move essential state data, then starts the container at the destination and retrieves the rest of the data when required [48]. This type of migration has a very small downtime but containers may suffer from performance degradation due to the time needed to wait for the requested memory pages.

In traditional hypervisor-based virtualization, virtual machine (VM) migration is well investigated [49] and many successful solutions are commercially available. For instance, a pre-copy based VM migration scheme is presented in [47]. An active VM continues to run in the course of in-memory data iterative pre-copying. During a consecutive iteration, only changed memory (dirty pages) are transferred. At last, a final state copy is performed while the VM instance is frozen and then transferred to the destination host. This way, the amount of downtime is greatly reduced when compared to a pure stop-and-copy scheme. Although VM migration is a mature technology, it relies on hypervisors and most of the existing solutions are tailored for data centre environment where network-attached storage (NAS) and specific virtualization technology are utilized. NAS enables all the host machines in a data centre to access a network-shared storage which reduces the time spent during the copying stage. However, in a scenario where migration takes place between MECs (train to land), state and local-disk storage must also migrate over wide area network.

Container migration has lately caught much attention from the research community [47], [50]. Especially, since containerization offers many advantages, in terms of resource efficiency and performance, over traditional hypervisor-based virtualization. This fact enables the instantiation of lightweight containerized applications suitable for IoT services [51]. In reference [50], container migration mechanism is developed for power efficiency optimization in heterogeneous data center. This work assumes that the source and destination hosts have access to a NAS and thus container data is not copied over wide area network. Recently, a framework for migrating containerized applications is presented in [51]. The proposed framework is the first to consider container migration for MEC environment. Fundamentally, the framework is a layered model that aims to reduce the migration downtime. While the presented results show reduction in downtime, the framework relies on stop-and-copy which is inefficient method for containers with large in-memory state. In our proposed solution, we develop a pre-copy migration scheme to relocate both system-based and application-based containers across edge networks.

In 5G-CORAL, we deploy the vMME on-board as illustrated in Figure 7-1. As mentioned, the S10 interface will face large amount of control signals when hundreds of user's devices perform HO simultaneously. Therefore, we enhance the S10 interface to reduce the signaling between on-board and on-land EFS nodes, where the adopted EFS virtualization infrastructure has the MME functionality. It is the totality of the hardware and software components that build up the environment. In particular, the EFS consists of several elements namely applications, functions and services as shown in Figure 7-2. These elements are detailed in the following subsection:

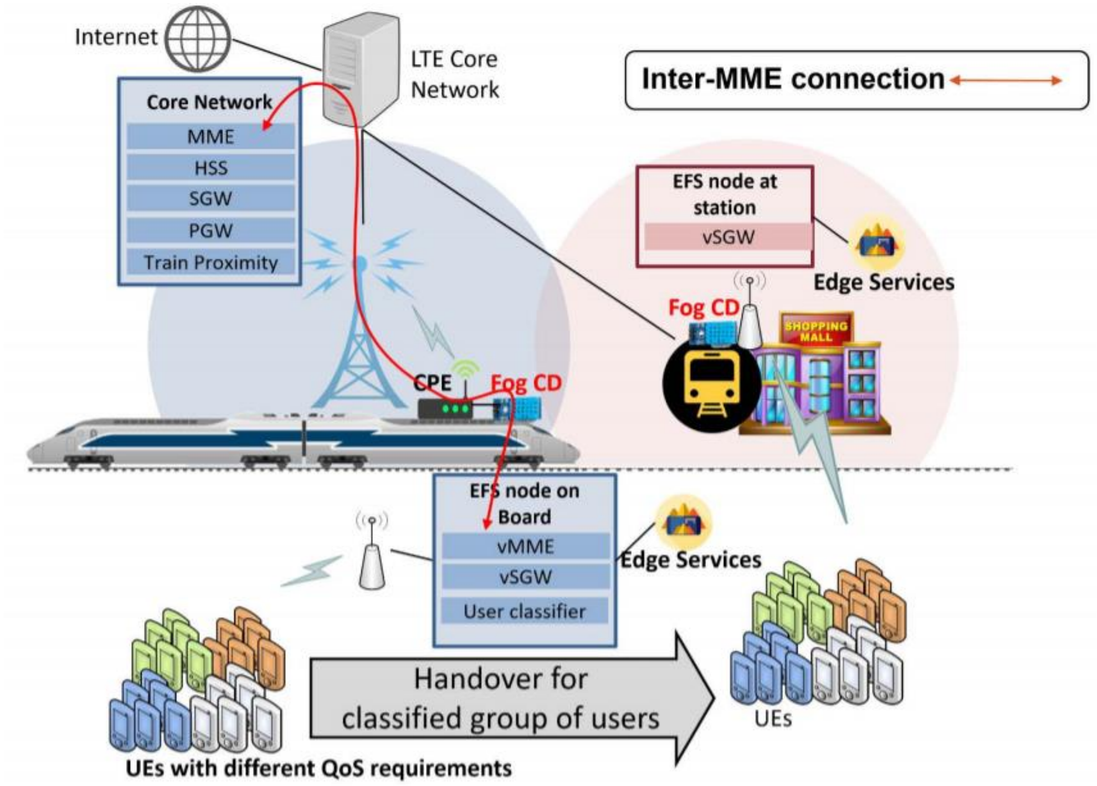


FIGURE 7-1: POC #6 ENHANCED INTER-MME HANDOVER IN TWO HOP ARCHITECTURE

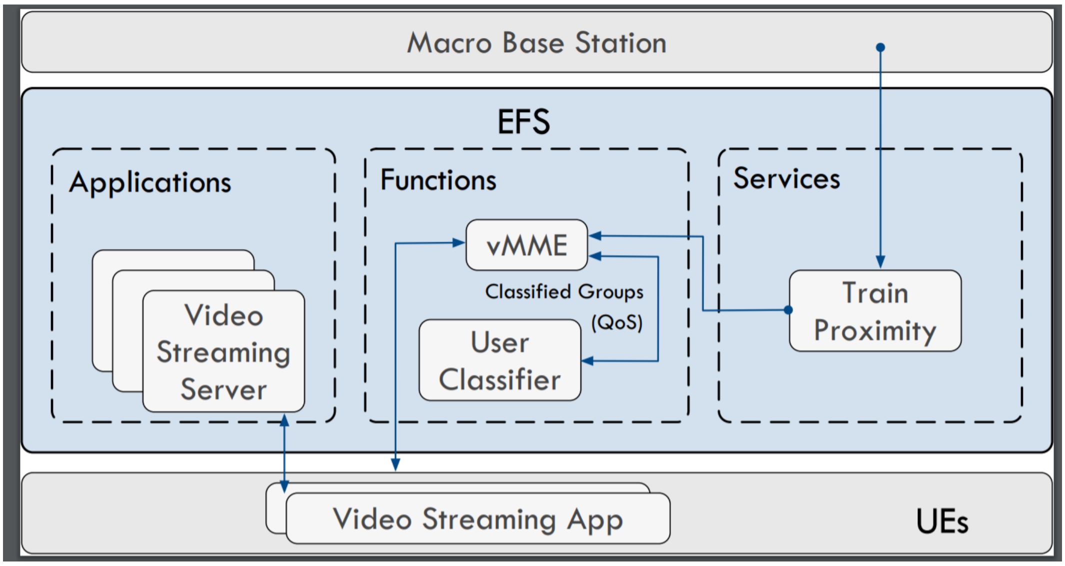


FIGURE 7-2: POC #6 EFS ENTITIES FOR HIGH-SPEED TRAIN USE CASE

7.1.1 Overview of Integrated Components

This PoC uses several software/hardware components as listed below:

7.1.1.1 Hardware setup

The following describes the hardware setup and specifications for high-speed train PoC. The hardware specification details can be found in Table 7-2. It is worth noting that checkpoint and restore functions in user (CRIU) are computationally expensive. The checkpoint function collects a process tree and its resources, freeze the process, then write them to files. The restore function reads the files, resolves shared resources, fork the process tree then restore the process resources. Both functions perform I/O operations which are generally slow especially on rotational block devices such as hard disk drive (HDD). To improve the migration scheme, we include the following enhancements on the EFS nodes:

- **Low-latency computing capabilities:** Linux generic kernels fail to provide time guarantees for time-critical applications. Hence, we incorporate low-latency computing into the EFS nodes. In addition, we scale the CPU performance to avoid latency caused by waking up from idle state.
- **Fast storage:** HDD uses mechanical mechanism to persistently store data in blocks of 512 bytes. As such, I/O operations experience seeking time delays (i.e., the time it takes the disk head to find the target track). Here, we utilize temporary file system ***(tmpfs)*** to enhance the performance as it allows short-term files to be written and read without generating disk I/O.

TABLE 7-2: POC #6 HARDWARE SPECIFICATIONS USED IN THE EXPERIMENTAL SETUP

Hardware Component		Specifications
Fog Node	CPU	Dual-Core NVIDIA Denver 2 64-Bit CPU Quad-Core ARM® Cortex®-A57 MPCore
	Memory	8GB 128-bit LPDDR4 Memory
	Network	802.11a/b/g/n/ac 2×2 867Mbps WiFi 10/100/1000 BASE-T Ethernet
	GPU	256-core NVIDIA Pascal™
Small cells	Model	Wistron NeWeb
	Wireless Adapter	LTE compatible- Rel 9
	Operational frequency	Band 7
	Network	4× Ethernet 2×Wireless
CPE	Model	B315s-607
	Wireless Adapter	LTE compatible
	Operational frequency	Band 7
	Network	4× Ethernet 2×Wireless
Core Network IPC	Model	RJ-45
	CPU	Intel i7 4 core
	Memory	16 GB RAM
	Network	802.11a/b/g/n/ac 2×2 867Mbps WiFi

		10/100/1000 BASE-T Ethernet

7.1.1.2 Software setup

The software specification can be found in Table 7-3 and detailed as follow:

- **User Applications:** Containerized applications are created for UEs based on their QoS. For example, a containerized video streaming application is created for a group of UEs demanding the same QoS requirements.
- **User Classifier function:** The UEs on-board have different QoS requirements. The user classifier function classifies the UEs into groups based on context information such as QoS Class indicator (QCI) and allocation and retention priority (ARP) [52] that are extracted from vMME (i.e. nine different QoS types defined in the legacy system). The classification is accomplished by two steps. In the first step, the user classifier extracts UEs context information and sorts them based on QCI and ARP index. In the second step, the classifier shares the groups with vMME in descending QoS order for service HO triggering (see Figure 7-3). The classification process is very important to reduce the signalling overhead during the inter-MME Handover and maintain the service interruption at the minimum.
- **vMME:** it interacts with target MME, located in the core network, to perform the enhanced Inter-MME HO as shown in Figure 7-1 and Figure 7-4. In step 0 to step 10, the enhanced inter-MME HO is executed ahead of time unlike the legacy system (i.e. inter MME handover) as follow. As the train approaches the station, MME detects it and sends a trigger to vMME. Next, vMME executes group HO based on the user classification. In particular, the vMME transmit essential UEs context information early which will reduce the amount of signalling during the HO process. Then, vMME forwards the relocation UEs information request to target MME and receives its response. The remaining part follows the standard inter-MME HO procedure [53].

Algorithm 1: Service HO Triggering

Input: Active UE list UEs ($UEs = UE1, UE2, ..., UE_i$), each UE_i has QCI_i and ARP_i
Output: the selected UEs H to be handovered
Step 1: Sort UEs into group list G_j, $1 \leq j \leq 9$
for *each* UE_i *in* UEs **do**
    G_{QCI_i}.append(UE_i)
for *each group list* G_j, $1 \leq j \leq 9$ **do**
    SortByARP(G_j)
Step 2: Decide H and triggers
Concatenate G_i, $1 \leq j \leq 9$ to full list G
while G.len() $> \Theta$ **do**
    H = G.deque(Θ)
    SendServiceHOTrigger(H)
H = G.deque(G.len())
SendServiceHOTrigger(H)

FIGURE 7-3: PoC #6 ENHANCED INTER-MME HO ALGORITHM

- **Train Proximity:** The proximity of trains is handled by the MME of the core network. The MME monitors several UEs and PCIDs (eNB ID), and since the route path of the train is known, the approximate location of the train can be estimated. In particular, the MME monitors the CPE on-board connection to eNB in the two-hop architecture. When the train is approaching the station, the CPE will HO to a specific eNB near the station. Thus, MME can estimate the approximate location of train and executes the HO triggering function. In HO triggering function, MME sends the trigger to the vMME ahead of time. Also, the user containerized application will migrate from on-board to on-land EFS nodes. Then, it executes service HO triggering as UEs are moved to specific eNB.

TABLE 7-3: POC #6 SOFTWARE SPECIFICATION USED IN THE EXPERIMENTAL SETUP

Software Component		Specifications
Fog Node	OS	Ubuntu 16.04
	Kernel	4.4.38 (modified)
	LXD	3.0.3
NextEPC	Framework	LTE Rel 13
	interface	vMME, S1/S10
Multiple user Emulator	Framework	LTE Rel 13
	interface	Next EPC user emulator

- **OCS (Service migration):** The proposed migration scheme enabling technologies include 5G-CORAL, Linux containers, CRIU, and remote file synchronization (***rsync***). The orchestrator part of the OCS manages the lifecycle of the containerized applications running on the EFS nodes. It supports management operations such as instantiating, cloning, migrating, scaling and terminating. CRIU is utilized to checkpoint the state of the migrating containers. The local-disk and the state of the containers are copied to the destination node by using ***rsync*** for its remarkable speed and efficiency. To relocate a container between EFS nodes with minimal downtime, we develop a pre-copy migration scheme. Figure 7-4 shows the pre-copy procedure as an integral part of the proposed mobile service continuity solution. The logical steps of the proposed scheme are summarized in following:
    - **Local-disk-copy-** the container base-image is assumed to be available in all edge nodes to reduce traffic overhead and to keep the total migration time to minimal. Local-disk synchronization is performed to copy application related files.
    - **Iterative-pre-copy -** the container state is dumped to the source node storage, and then copied over to the destination while the container continues to run. Next, pre-copy iterations are performed to checkpoint and copy only the memory pages that have changed (dirtied) since the last checkpoint.
    - **Stop-and-copy**- the container gets frozen in this step, and then a final checkpoint and copy round is performed. The downtime observed by the user occurs during this step.
    - **Restore-and-terminate** - the container is restored in the destination EFS node and the frozen container in the source node gets terminated.

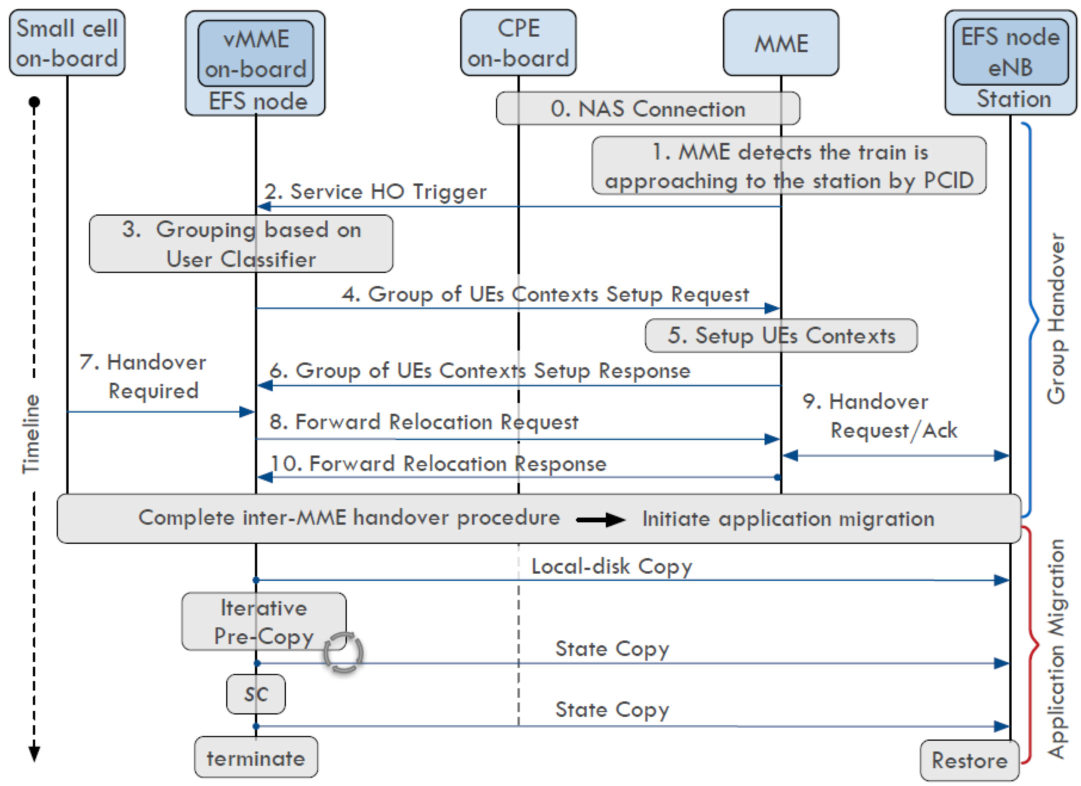


FIGURE 7-4: PoC #6 SERVICE CONTINUITY FLOWCHART FOR HIGH-SPEED TRAIN USE CASE

7.1.2 Required 5G-CORAL Building Blocks

TABLE 7-4: PoC #6 REQUIRED 5G-CORAL BUILDING BLOCKS

Sub-system	Component's Name	Description
EFS	vMME	An **EFS Function** responsible to redirect local traffic from onboard and onland Responsible to handover the classified group of users from onboard to on land
EFS	User Classifier	An **EFS Function** to classify users into groups based on the shared application type and QoS requirements
EFS	Train approximate	An **EFS Service** to report the approximate location of the train based on PCI information of Macro base station deployed along the railway.
EFS	QoS and application type service	An **EFS Service** to report user's context information related to QoS and application type such as data session, allocated resources, bearer service, and network internal routing information.
Non-EFS	LTE Small Cell platform, EPC components	Responsible to provide the connectivity for the end user on the Train and the EFS components

OCS	Service Migration	An **OCS** Component to migrate the end user content from onboard to on land unit once the train approaches the train station.

7.2 PoC KPIs: Measured Values for the performance metrics

In this section, we explain the experimental setup for the two-hop architecture and the EFS deployment on- board and OCS deployment on-land. Then, we present our experimental results for the enhanced inter-MME HO and the edge application migration. In particular, the control signal packet sizes, HO latency, downtime measurements for legacy system and the proposed schemes are highlighted in the following subsections.

7.2.1 Measurement Methodology

In the emulation environment of moving network, we utilize real-time reference signals received power (RSRP) and received signal strength indication (RSSI) from a CPE of Taiwan high-speed rail. The collected signals are taken from a 30 km route at an approximate train speed of 300 km/h. The emulation environment consists of the following components:

- **EFS Node:** Hosts several applications, functions and services such as video streaming application, vMME and user classifier function, and train proximity service, respectively. In our experiment, we utilize NextEPC [53] framework as baseline for vMME. Then, we modify vMME to fit our proposed scheme.
- **Small cell:** A small cell acts as an eNB on-board of the train. Another small cell acts as an on-land Macro base station that has the capability to adjust RSSI and RSRP to emulate the train signal degradation.
- **CPE:** Gateway between on-board and on-land small cells. It can also act as WiFi access point.
- **UE:** On-board user devices are emulated using NextEPC UE emulator. The emulated users have different services requirements.
- **Core Network and OCS:** A server hosts the core network and emulates the train proximity. Also, runs the OCS to enable application migration based on the train mobility.

The experiment is carried out as follows. First, the MME, the small cell and the CPE are powered on. Then, the vMME is powered on and connected to HSS in the core network. Next, four emulated UEs connect to the small cell (on-board). At the core network, the MME connects to the target eNB (on-land). As train mobility is emulated, the MME sends HO trigger to vMME which in turn classifies the users into groups based on the QoS. In this emulated train network, latency and throughput measurements are executed as shown in Table 7-5.

TABLE 7-5: POC #6 EXPERIMENTAL SETUP LATENCY AND THROUGHPUT

Measurements	Average Values
Latency between EFS node and core network	50ms

Ping latency between UE to core network	75ms
Iperf tcp test (EFS node to core network)	35Mbps
Iperf tcp test (UE to core network)	15.3Mbps

7.2.2 Measurement Results

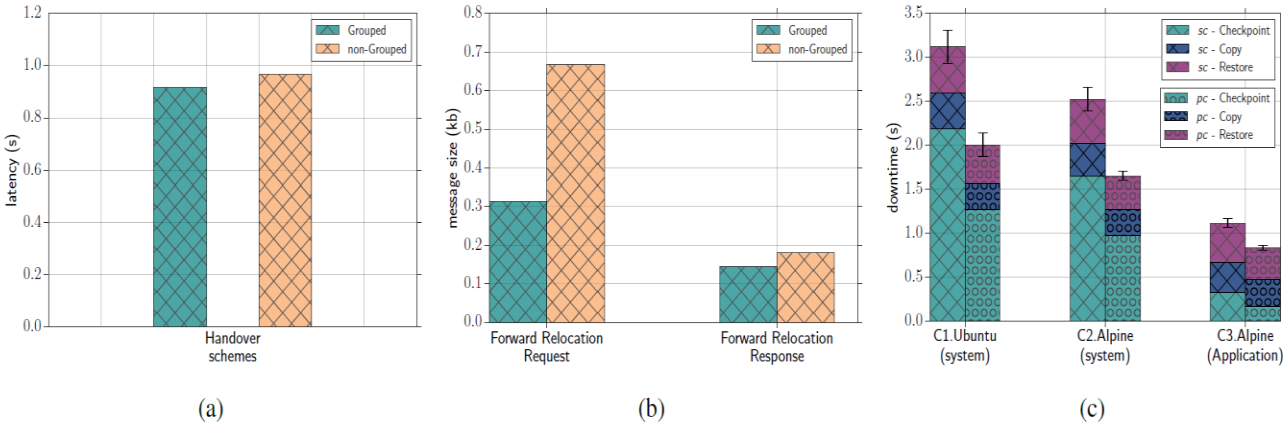


FIGURE 7-5: POC #6 GROUP HANDOVER AND APPLICATION MIGRATION. (A) HANDOVER LATENCY. (B) HANDOVER CONTROL MESSAGES OVERHEAD. (C) CONTAINER MIGRATION DOWNTIME.

Figure 7-5 (a) represents a comparison between the enhanced and the legacy inter-MME HO. The x-axis represents the enhanced inter-MME (Grouped) and legacy inter-MME (non-Grouped) while the y-axis represents the inter-MME HO time. The grouped HO slightly improved the total HO time when compared to non-grouped inter-MME HO. In both cases, the HO time is high due to the emulation environment overhead. In real deployment, the average HO time is approximately 200ms. It is important to note that the latency reduction of HO is not the focus of this work. As such, maintaining the same HO latency highlights our objective to reduce signaling storm.

Figure 7-5 (b) represents a comparison between grouped and non-grouped inter-MME HO control messages. The x-axis represents the forward relocation request and forward relocation response, respectively. On one hand, the forward relocation request contains several context information request such as International mobile subscriber identity (IMSI), target identification, and UE Time. On the other hand, the forward relocation response provides the reply for the requested context information for UEs. The y-axis represents the average control message sizes in bytes. In case of forward relocation request, the grouped inter-MME scheme scaled down the average control messages up to 50% per user in comparison to the non-grouped scheme. That is, the forward relocation request messages size per UE are 690 and 320 bytes for non-grouped and grouped schemes, respectively. As such, a significant signal reduction is achieved when hundreds of UEs are managed by the proposed scheme. For the forward relocation response, the grouped inter-MME HO scheme reduces the overhead by 25% per user in comparison to the non-grouped scheme. The forward relocation response messages are 207 bytes and 149 bytes per user for non-grouped and grouped, respectively. Note that this reduces the signals to core network significantly in large scale scenario and at the same time contributes towards application stability at the end user side.

To benchmark container migration, we implemented the stop-and-copy (sc) scheme to reproduce the results presented in [25] and evaluated its downtime against our proposed pre-copy (pc) migration scheme. The migration experiments were carried out between two EFS nodes. In this experiment, we evaluate the downtime during the migration of a video streaming application running in LXC system containers (Ubuntu and Alpine) and LXC application container (Alpine). The presented results are based on average values of ten trails for each presented case. Figure 7-5 (c) compares the migration downtime of the two schemes.

Since the rate of dirty pages for the video streaming application is minimal, most of the downtime is attributed to the common steps of both migration schemes rather than the amount of data copied. For instance, in the case of C1 (Ubuntu system container), the downtime due to the checkpoint process are 2.19s and 1.27s for the sc and pc, respectively. In the case of C3 (Alpine application container), the observed migration downtime is an average of 0.83s.

Table 7-6 shows the breakdown of the downtime during migration tasks. Overall, the proposed pc migration scheme with the EFS enhancements reduces the downtime by 36% when compared to the SoA container migration.

TABLE 7-6: POC #6 DOWNTIME OF MIGRATION TASKS

Stop-and-copy			Pre-copy		
Checkpoint	**Copy**	**Restore**	**Checkpoint**	**Copy**	**Restore**
C1 – System Container (Ubuntu 4.4.0 – 116)					
2.18s	0.40s	0.53s	1.26s	0.30s	0.43s
C2 – System Container (Alpine 3.7)					
1.65s	0.36s	0.50s	0.97s	0.29s	0.38s
C1 – System Container (OCI - Alpine 3.7)					
0.32s	0.34s	0.44s	0.17s	0.29s	0.36s

TABLE 7-7: POC #6 MEASURED VALUES FOR THE PERFORMANCE METRICS RELATED TO HIGH-SPEED TRAIN.

Performance metrics	Description	Reference Values	Measured Values
Down time	The time requires to migrate an application from on board Fog CD to on land Fog CD due to the mobility of the user.	3s [48], [49], [50], [51].	0.85s- 2s based on container image
Handover Latency	Round trip time between issuing handover command from eNB and Complete of handover	0.9s for a 2-hop architecture implementation in lab [42], [43], [44].	0.9s with 25% signal reduction

7.3 Year 2 Achievements and Future Directions

In moving infrastructure scenarios such as train networks, two-hop architecture is adopted to improve on-board user experience by reducing the interaction with on-land base stations. Furthermore, edge networks and virtualization technologies can be utilized to bring services closer to the traveling users. Nevertheless, when large number of users transit from train to station, a signaling storm and backhaul latency become challenges to maintain a continuous service. In high-speed train PoC, we propose a mobile service continuity solution which includes a group handover and application migration schemes. Our experimental results show that the proposed schemes can reduce the control signals and migration downtime by 50% and 36%, respectively. The vMME finalities can be extended to be deployed by OCS in nodes according to the demands. Also it can be adopted in to reduce the control signaling in very crowded locations.

7.4 Demonstration activities

Throughout the project lifetime, we carried out a number of video demonstrations presented or posted at 5G-CORAL website. In particular, 5G-CORAL trial in November 2018, we presented a video for High-Speed train PoC at 5G-CORAL trial in Taipei, Taiwan. This video targeted a minimum service interruption for passengers on train especially when transiting from one Macro base station to another one along the railway. Especially, it addressed how fog nodes can handle partially core network functionalities at the edge especially at transit from on board to on land as depicted in Figure 7-6.

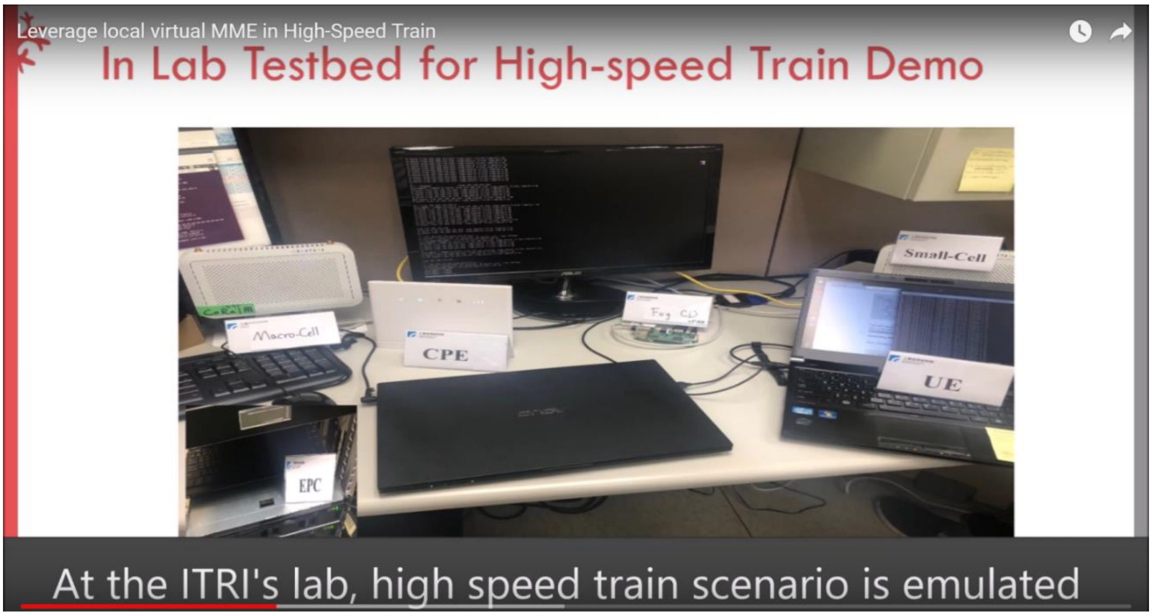


FIGURE 7-6: POC #6 HIGH-SPEED TRAIN AT TAIWAN TRAIL 2018

8 PoC #7: Connected Cars

Vehicular communications are all the kind of communication between vehicles and infrastructure (V2I), vehicles (V2V), network (V2N) and pedestrian (V2P). In the contest of 5G CORAL, the Connected Car scenario aims to show how 5G CORAL technologies can be applied to meet the ever stringent requirements of the vehicular communications [54] and [55], [56]. In this scenario, EFS components can run on the edge and fog cloud as depicted in Figure 8-1. The vehicles are equipped with an On Board Unit (OBU) while the infrastructure consists of Road Side Units (RSUs) which are fog nodes. All these devices can collect speed, direction and position data to help drivers and improve the cars safety.

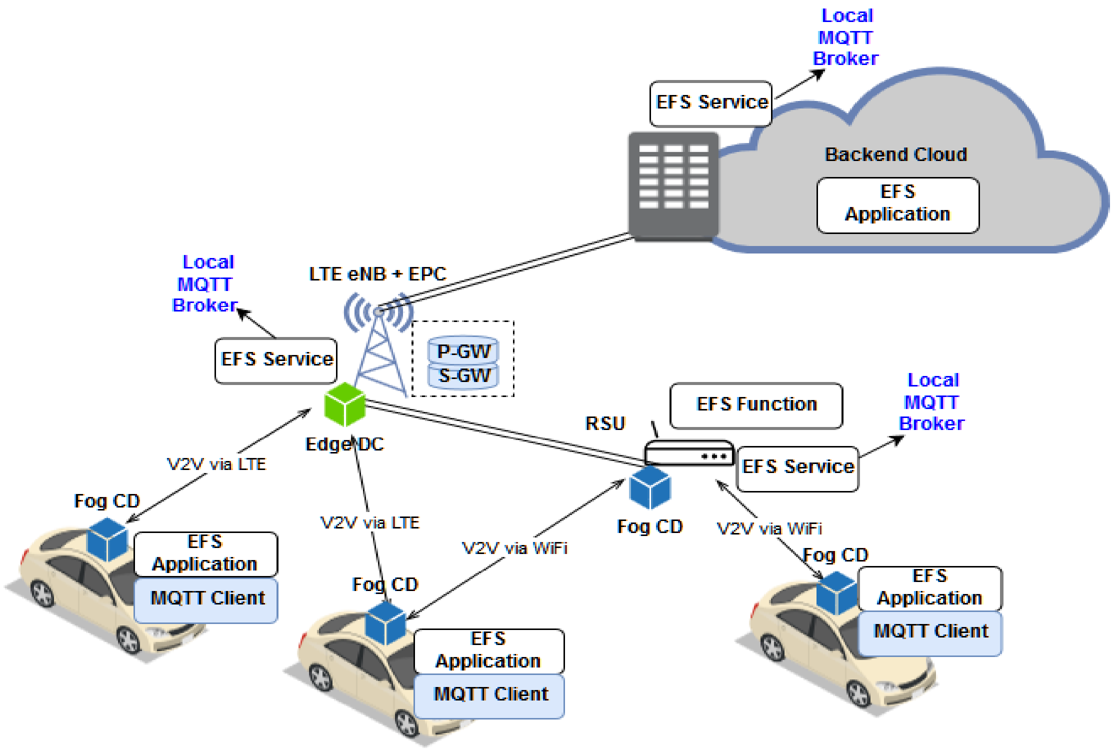


FIGURE 8-1: POC #7 CONNECTED CARS TESTBED ARCHITECTURE

In this testbed, 5G-CORAL technologies, Edge and Fog Computing System (EFS), and Orchestration and Control System (OCS), are integrated with other non-EFS components such as legacy 4G LTE eNBs. Moreover RSU works as a WiFi virtual access point enabling another RAT. The goal is to validate the 5G CORAL multi RAT architecture allowing low, stable and reliable communications, for some use cases, i.e. collision avoidance emergency vehicle approaching and vehicle breakdown notification.

Table 8-1 provides the list of performance metrics measured through the experiments performed into the TI laboratory testbed.

TABLE 8-1: POC #7 PERFORMANCE METRICS AND USER BENEFITS

Performance metrics	Description	User Benefits
Latency	Maximum tolerable elapsed time from the instant a data packet is generated	Low and stable latency

	at the source application to the instant it is received by the destination application.

8.1 PoC Testbed Design

8.1.1 Overview of Integrated Components

First of all, the testbed is developed and tested in the Azcom Labs and then, with the support of TI, is developed in the TI Labs in Turin, as depicted in the following Figure 8-2

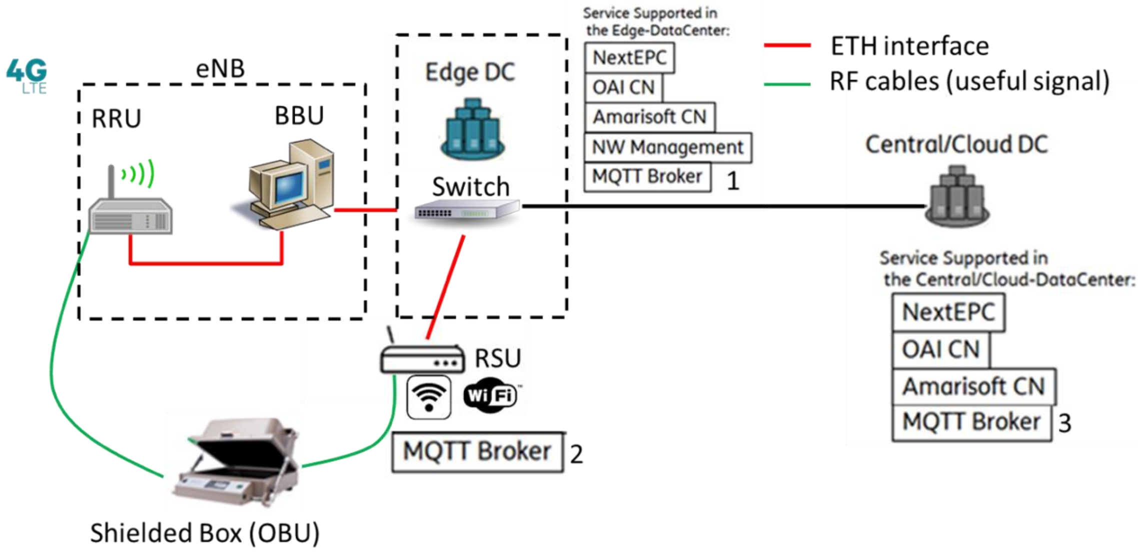


FIGURE 8-2: POC #7 CONNECTED CARS TESTBED ARCHITECTURE IN TI LAB

As described in the previous deliverable D4.1 [2], the TI wireless lab is involved in the integration activities to analyze the impact of radio propagation conditions and the network load on the capability of collision avoidance. The Figure 8-2 shows a generic description of TI wireless lab where both the LTE 4G and WiFi RAT are installed and connected to an edge DC. The MQTT Broker can be instantiated in several domains in order to analyze the impact of network latency to the connected car application performance.

The components used for connected car testbed are described below, summarizing the ones already described in D4.1 [2]:

- **On-board unit (OBU):** the OBU, designed and developed by Azcom, is a vehicular Fog node, which is configured to run safety applications on the vehicles. The OBU is placed inside the shielded box for the LAB test. The vehicle information are simulated from pre-generated files (containing: position information, speed information, alarm log etc…). An external laptop connected to the OBU is used to collect information in order to analyze the connected car system performance.
- **Road-side unit (RSU):** the RSU, provided by Azcom, is a Fog node, which is configured as a WiFi virtual access point, and where is instantiated a MQTT broker. The RSU is a MintBox Mini 2 Pro. (MBM2 Pro) [57], in the following Table 8-2 the most important characteristic are summarized.

TABLE 8-2: POC #7 MINTBOX MINI 2 PRO SPECIFICATIONS

Component	Specification
CPU	Quad core Intel Celeron J3455
RAM	8 GB
ROM	120 GB SSD
WiFi	802.11a/c

- **eNBs**: both open source eNB (exploiting OpenAirInterface (OAI) Software [58]) or commercial nodes are provided by Telecom Italia and can be used in these tests. The eNB is composed of a Radio Remote Unit (RRU) for the implementation of the RF part and the low-L1 LTE protocol and a Base Band unit (BBU) for the higher-L1 and L2/L3 LTE protocol. Commercial nodes have higher bandwidth capacity compared to open source solution, but the latency performance for small packet transmissions are similar. Because of that, both solutions are usable for test purpose when the number of OBU under test is limited.
- **Shielded box**: connected via RF cable to the eNB and to the RSU in order to create a controlled radio environment inside the box.
- **Core Network (EPC)**: preliminary LAB tests use NExtEPC as an open source EPC.
- **End terminal and car emulator**: an external laptop is used for visualizing and collecting the experimental data. Moreover, it can also run the simulation software which can generate additional connected car messages (GPS and OBDII tracks).

8.1.2 Required 5G-CORAL Building Blocks

The PoC proposed by Azcom and Telecom Italia is focused on the "Road Safety" use cases, specifically on a delay sensitive warning service. The PoC is composed of two different scenarios, as described in the D4.1 [2]. The first one is depicted in the following Figure 8-3 shows a legacy configuration where the MQTT broker is instantiated beyond the Core Network, in a distant cloud. See D4.1 [2] for more details on this.

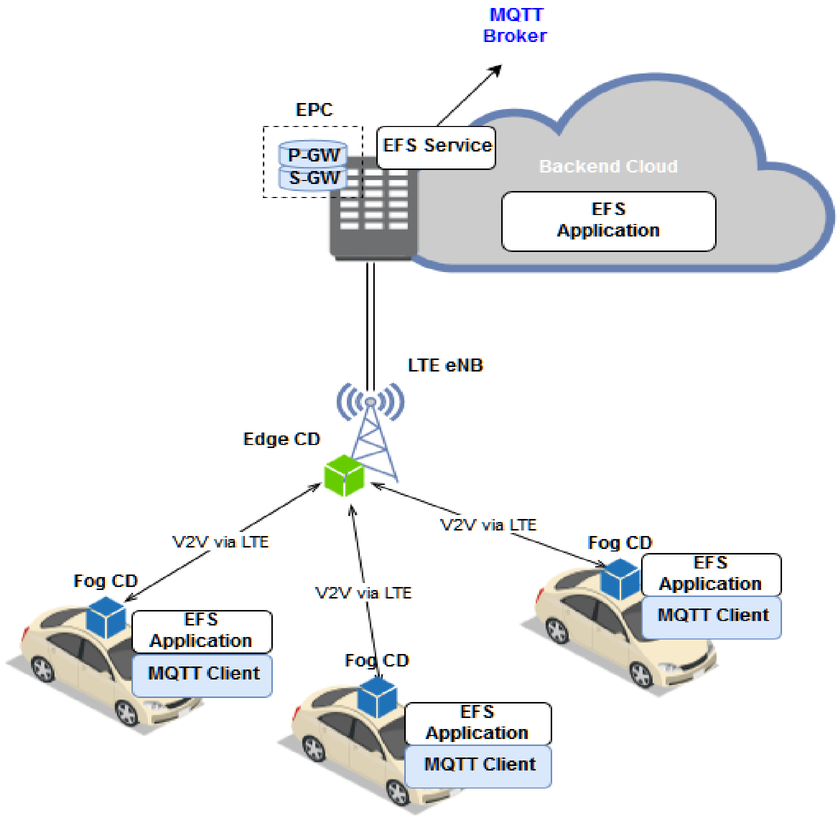


FIGURE 8-3: POC #7 CONNECTED CARS POC ARCHITECTURE: LEGACY SCENARIO

On the other hand the in the 5G CORAL architecture, the 5G CORAL EFS services and applications are configured to run both on edge and fog devices, which are located on the road infrastructure (e.g. RSUs) and in the vehicles (e.g. OBUs). As depicted in Figure 8-4 the 5G CORAL architecture enables the deployment of the MQTT broker closer to the user: not only in the edge cloud on a MEC server near the eNB, where also a local EPC is present D4.1 [2], but also even closer, in a fog CD, i.e. on the RSU. This enables local routing and consequently guarantees a lower and stable latency compared to the legacy scenario.

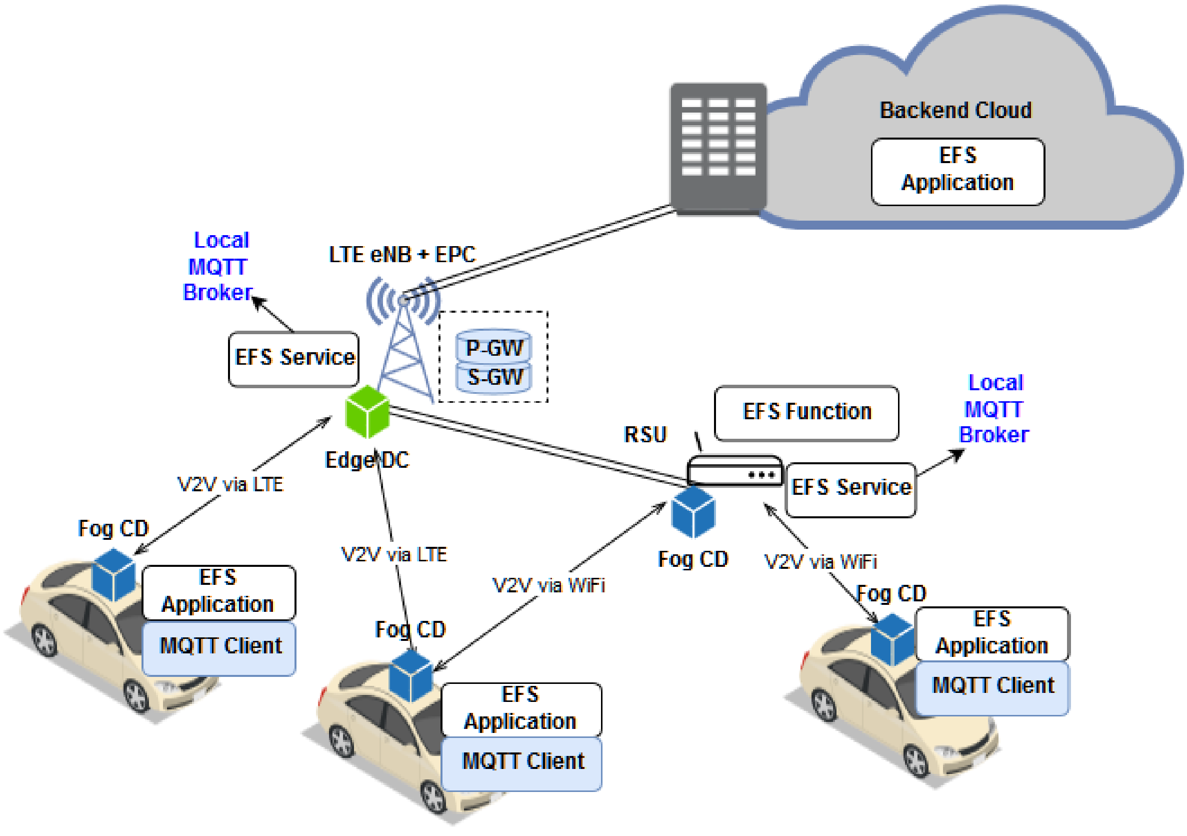


FIGURE 8-4: POC #7 CONNECTED CARS POC ARCHITECTURE –5G CORAL SCENARIO

In this second scenario of the PoC, two instances of the MQTT broker are deployed: one on the edge cloud on a MEC server near the eNB and the second one on the RSU. Moreover, the RSU works as a WiFi virtual access point enabling a secondary RAT. It is worth pointing out that the lower bound of the E2E air latency is still represented by the RAT E2E air interface latency, which, in the PoC, is represented by 4G LTE RAN and the WiFi. The brokers are connected in the MQTT bridge configuration where a central and local broker can be distinguished. In this configuration, when a local broker receives a massage from one of its clients, it publishes it to the central broker which publishes it back to all the other local brokers which in turn publish it to all of their subscribers. This leads to a duplication of messages; however, it results in an increase of the reliability of the whole system since the OBUs can be connected to both the MQTT brokers by means of two different RATs, and if one fails, there is the second one as a backup. Since messages can arrive with different delay, in this configuration, it is fundamental to be able to cope with duplicated messages. The software running on the OBUs keeps track of the received messages and preserves only the freshest ones.

In this PoC the following entities are deployed, as depicted in Figure 8-5:

- EFS Applications:
    - Collision avoidance: warn in case vehicles can collide;
    - Emergency vehicle approaching: warn if an emergency vehicle is approaching;
    - Vehicle breakdown notification: warn if the vehicle is not working properly.
- EFS Services:

- Telemetric information as velocity, position, heading, vehicle type...;
- Warning notification related to a given area and for a certain time.

- EFS Functions:
  - WiFi Virtual access point.
- Non-EFS Function (TI testbed):
  - Next EPC;
  - Phluido eNB.

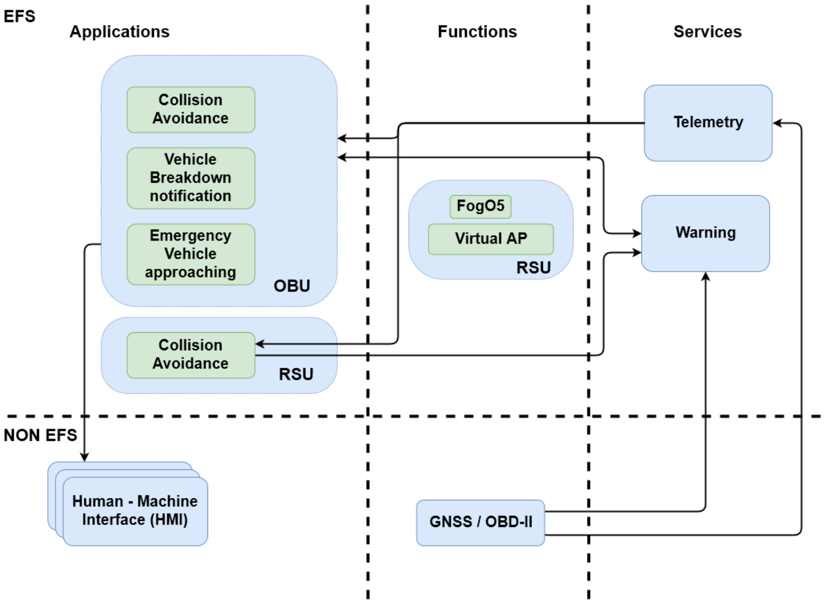


FIGURE 8-5: POC #7 EFS ENTITIES CONFIGURATION FOR CONNECTED CAR USE CASE

The software architecture of the vehicular EFS system depicted for the sake of clarity in Figure 8-6, is the same already described in D4.1 [2]; two different RATs are now available.

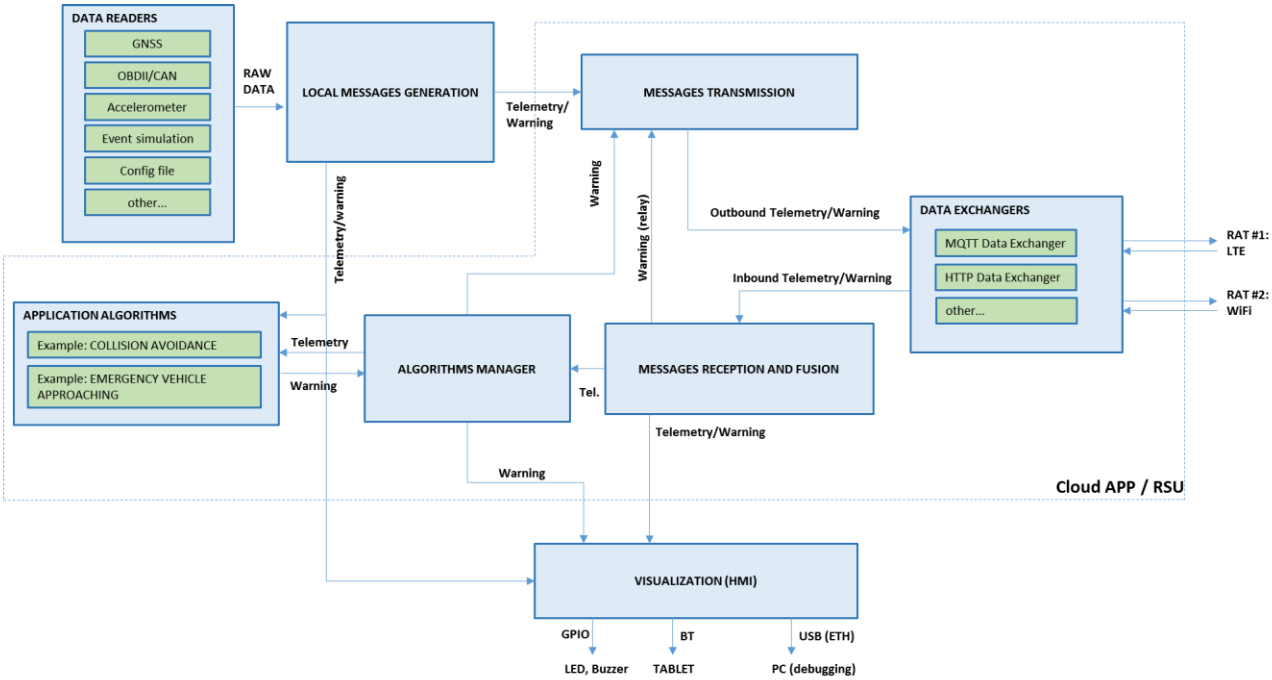


FIGURE 8-6: POC #7 CONNECTED CAR POC SOFTWARE ARCHITECTURE

Below are reported the assumptions considered for the implementation of this PoC:

- aside the LTE, WiFi RAT technology is used in the experimentation in the TI Lab, since this represents a fast and simple way to proof the concept, the architecture proposed is still valid considering different RATs;
- the considered maximum car speed will be limited according to the E2E latency performance of the system.

As described in D4.1 [2] the vehicles trajectory for the experimentation activities in TI testbed are simulated. A web dashboard for the visualization of CAM and DENM messages has also been developed. The experimentation is done with two OBUs connected to two MQTT brokers as described in the following sections. All the measurements are done in TI testbed, in Turin (Italy).

The Connected Cars PoC relays on several 5G-CORAL building blocks: EFS service, EFS application and OCS.

TABLE 8-3: POC #7 REQUIRED 5G-CORAL BUILDING BLOCKS

Sub-system	Component's Name	Description
EFS (Application)	Safety Application	• Collision avoidance: warn in case vehicles can collide; • Emergency vehicle approaching: warn if an emergency vehicle is approaching; • Vehicle breakdown notification: warn if the vehicle is not working properly.
EFS (Service)	Telemetric Service	Telemetric information: • velocity; • position;

		• heading; • vehicle type.
EFS **(Service)**	Warning notification	• Warning notification related to a given area and for a certain time.
EFS **(Service)**	Message Service Platform	• MQTT Broker, provided on a LXD container.
EFS **(Functions)**	Message Fusion and management	• Message Fusion: mandatory for Multi-RAT support, as it selects the freshest messages among duplicates; • Message Management: stores received warning messages to show an alert only when relevant.
Non-EFS	LTE eNB + EPC components	• provide the connectivity for the EFS components; • Deployed at TI test-bed, it consists of a Phluido eNB (running on COTS hardware) and of a Next EPC (running on a VM).
OCS	EFS Application and Service LCM	• Instantiation of the EFS function on RSU.

8.2 PoC KPIs: Measured Values for the performance metrics

8.2.1 Measurement Methodology

The lab setup is described in Figure 8-7 where the eNB and the RSU are connected to the same subnet, this subnet is connected to the edge DC where RAN function can be deployed. In the edge DC several VMs could be instantiated, other VMs could be instantiated in the cloud DC. The VMs running in the edge DC and cloud DC are instantiated in different subnets. A specific network function provided by TIM is used to simulate the backhauling load and latency condition for the connection between edge DC and cloud DC.

Observing Figure 8-7 the MQTT Broker could be instantiated in position 1, 2 and 3, also multiple MQTT simultaneously deployed is supported:

Position 1: MQTT Broker in the edge DC near to 4G RAN and local EPC instantiation

Position 2: MQTT Broker in the RSU (WiFi Connection)

Position 3: MQTT Broker in the cloud DC near to geographic EPC instantiation

RF cables are used to convey the signal from the 4G Remote Radio Unit (RRU) and from the RSU to the shielded box, a Programmable Attenuator is used to analyze the impact of different received signal levels.

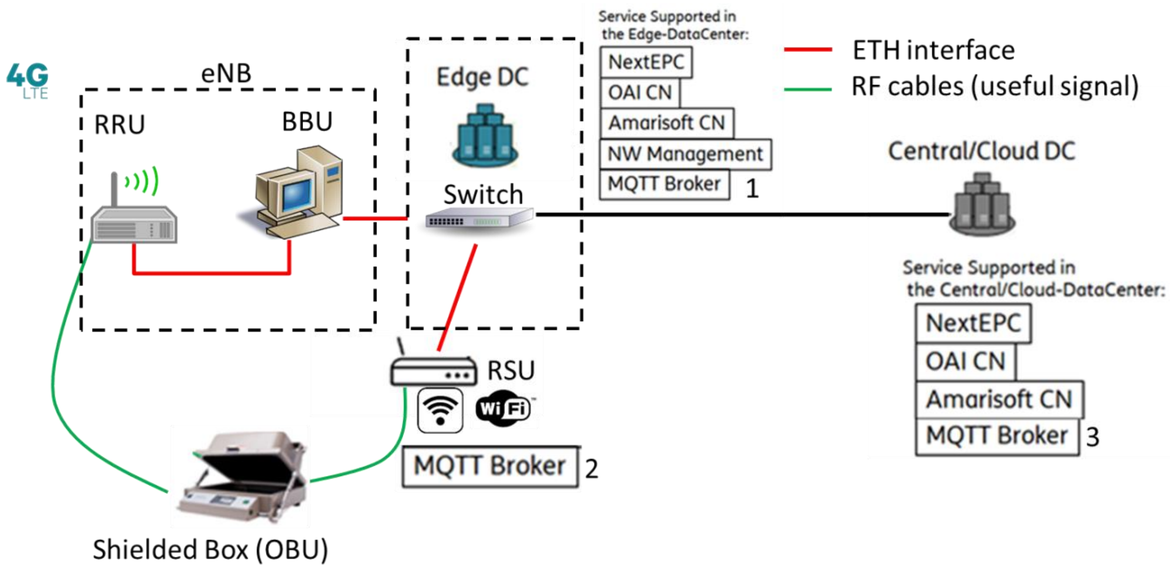


FIGURE 8-7: POC #7 HARDWARE AND SOFTWARE OVERVIEW IN THE TELECOM ITALIA LAB

In Figure 8-8 it is detailed how the RAN (eNB and/or RSU) is connected to the shielded box. The insertion of a split allows propagating part of the signal over the air to connect an additional user to the system. This user can generate DL and UL traffic to load the LTE cell or the WiFi connection in order to test the system with different levels of traffic load (not generated by V2X communication)

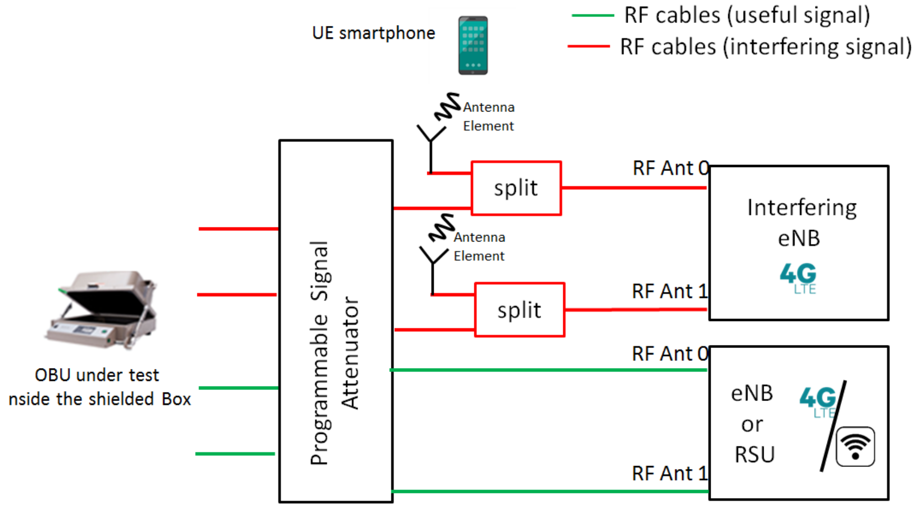


FIGURE 8-8: POC #7 RF CONNECTION CONFIGURATION

As depicted in Figure 8-8, the OBUs can be connected to several MQTT brokers, placed at different distances from the user. Two configurations are tested: the centralized legacy one where only the MQTT broker in the central cloud (position 3 in Figure 8-7) is available; and the 5G CORAL configuration, where Core Network is instantiated in the edge-DC and two MQTT brokers are deployed, one on the edge DC, and one on the RSU, respectively position 1 and 2 in Figure 8-7. The set of measurements are done considering two different traffic load scenarios, one with only OBUs are doing traffic in the LTE cell and another one in a fully loaded LTE cell, where an external LTE user load the network with a not V2X traffic.

8.2.2 Measurement Results

TABLE 8-4: POC #7 MEASURED VALUES FOR THE PERFORMANCE METRICS RELATED TO SAFETY

Performance metrics	Description	Reference Value	Measured Value
Latency	Maximum tolerable elapsed time from the instant a data packet is generated at the source application to the instant it is received by the destination application.	100 ms* [59] 20 ms** [59].	50-60 ms
*** Mutual vehicle awareness use case** **** Pre-crash sensing**			

The EFS service E2E latency is the metric measured for the two different configurations in the two different scenarios described above. After several measurements, the Cumulative Distribution Function (CDF) of the latency is calculated for the centralized and the 5G CORAL architecture. The results are depicted in the following Figure 8-9 on the left for the centralized one and on the right for the 5G CORAL architecture.

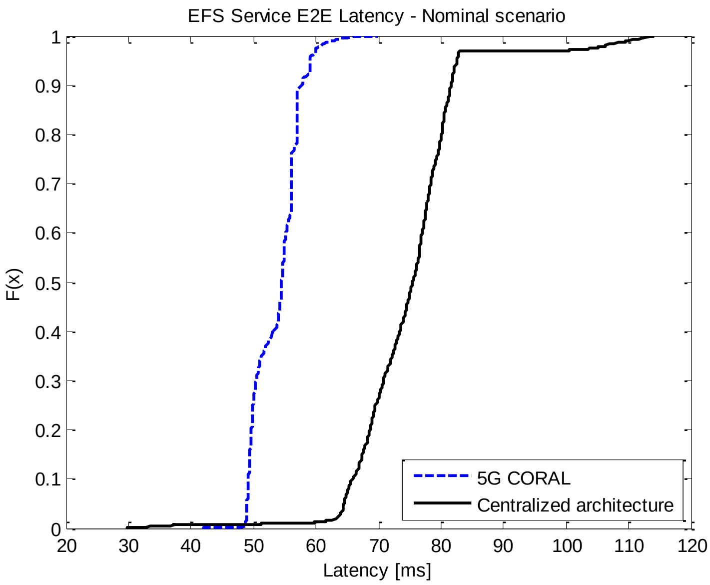


FIGURE 8-9: POC #7 LATENCY MEASURED CDF CENTRALIZED VS 5G CORAL

As depicted in the previous Figure 8-9, the 5G CORAL architecture guarantees a lower and stable latency, the latency values are in the range of 45-65 ms, compared to the widest range of the centralized architecture of 20-180ms. Moreover, the 5G CORAL architecture satisfies the latency requirements of the Mutual Vehicle awareness use case, 100 ms.

In the following Figure 8-10 the PDFs of the measurements in the full loaded network scenario are depicted for the two different scenarios.

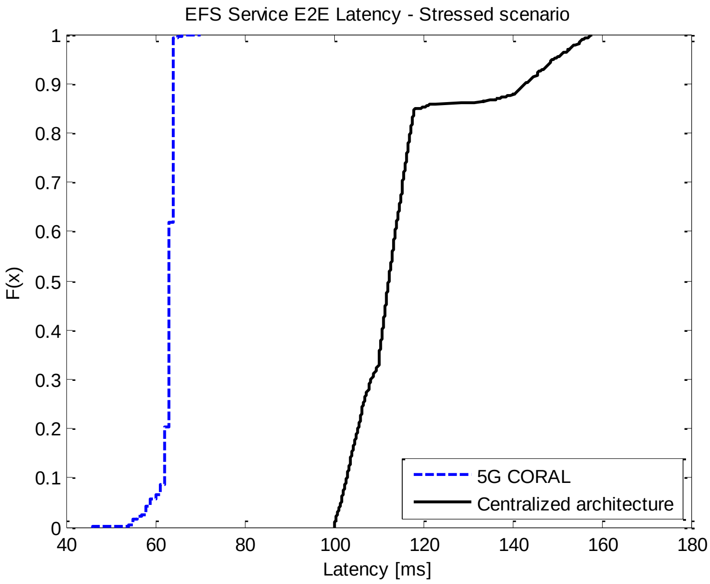


FIGURE 8-10: PoC #7 LATENCY MEASURED PDF CENTRALIZED VS 5G CORAL IN FULLY LOADED SCENARIO

Even in this stressed scenario, the 5G CORAL architecture guarantees a latency in the range of 45-70 ms, which is lower and more stable when compared to the centralized one that is the 100-160 ms range. Moreover, the latency requirements of the Mutual Vehicle awareness use case are still satisfied by the 5G CORAL architecture.

Because of the small dimension of the transmitted packets, the system is well robust to the channel quality reduction (caused by the increasing distance from the serving cell during vehicular movement) in both centralized and 5G Coral architecture. The most important effect taken into account in the previous analysis is the network load, this scenario can be caused by a large number of vehicles connected to the same radio infrastructure. The results of the stressed scenario demonstrated the value of 5G CORAL architecture also when a large number of vehicles are transmitting data to the MQTT Broker.

8.3 Year 2 Achievements and Future Directions

This PoC shows that the 5G CORAL architecture can guarantee a lower and more stable latency compared to the legacy centralized one. This satisfies the Mutual Vehicle awareness use case latency requirement, defined by [59], as the one shown in the PoC. However, considering more stringent use cases as Pre Crash Sensing, see [59], the latency is still too high. Further studies on the latency have been done, revealing that the latency is affected not only by the distance between user and broker, but also by the CAM and DENM messages size exchanged among vehicles. In the PoC the JavaScript Object Notation (JSON) encoding is used since it is easier to process and to debug across multiple platforms, but this format produces a message size of 1900 bytes. Consequently, different data encoding protocols have been studied, to understand how latency is linked to the data size. Two different protocols have been analyzed; namely CBOR (Concise Binary Object Representation, [60]), and Protocol Buffers, [61]. CBOR is a binary data serialization format similar to JSON. It can encode the data producing a smaller message, increasing the processing and transfer speeds. In our PoC it has been seen that it reduces the payload size to 1020 bytes, decreased by 42% compared to JSON data size. Protocol Buffers are a flexible, efficient, automated mechanism for serializing structured data. In our PoC it has been seen that it reduces the payload size to 346 bytes, that corresponds to a reduction of 82%

compared to JSON data. In the following Figure 8-11 the comparison of the CDF of the latency measurements done of the three protocols are depicted.

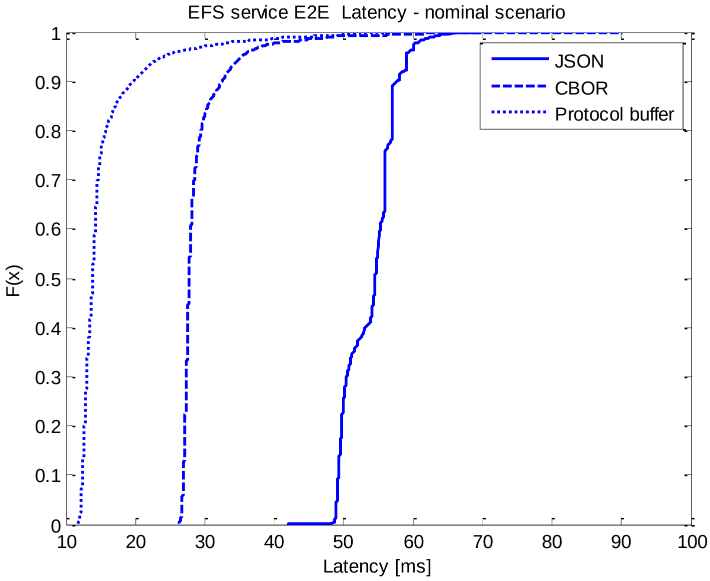


FIGURE 8-11: POC #7 LATENCY MEASURED PDF IN 5G CORAL – CBOR VS PROTOCOL BUFFER

It is very clear that reducing the message size results in a reduction of the latency. Protocol Buffers gives the best results, guaranteeing the lowest latency among the three protocols studied. It is worthy to notice that this kind of protocol is getting closer to the latency requirements of the Pre-Crash Sensing that is 20 ms see [59]. A part from these very good results the study of the encoding scheme can be further carried out.

As defined above, the limit of the latency is defined by the RATs used, and more suitable RAT can be used in placed of the WiFi, i.e. Cellular-V2X (C-V2X) [62] or DSRC [63], without modifying the architecture proposed in the deliverable.

8.4 Demonstration activities

At EUCNC 2019, it was shown the OBUs, the RSU and the web dashboard used to visualize the data exchanged over the PoCs platform, see Figure 8-12. The demonstration was accompanied by a poster and a video showing the complete description of the architectural solution, the results achieved in these two years and a field trial using real car. As described before, OBUs exchange alerts regarding the existence of a safety risk. This can be achieved exploiting the 5G-CORAL EFS Service based on MQTT. At the EUCNC demonstration, two different instances of the MQTT brokers were used. The first one was placed on the RSU in Valencia, which was connected via WiFi to the OBUs. The second in a central cloud hosted in Azcom's Lab in Milan, connected to the OBU by means of an LTE connection. Each OBU was connected to both the MQTT brokers. A simulated path, generated for testing purposes, was uploaded to the OBUs and visualized on the web dashboard. In addition to showing the car's simulated route, the web dashboard was also used to generate simulated warnings which were received in real time by the OBUs. Also, a prototype of a collision avoidance algorithm was running on the OBUs which generated warning messages every time the simulated positions of the two OBUs were coming too close to each other.

FIGURE 8-12: PoC #7 Connected CAR PoC at the EUCNC 2019

9 PoCs Relation and Impact to Projects Objectives

The section presents the summary of the impacts from each PoC work to the relevant R&D topics defined in the project objectives of Objective #2, #3 and #4 in 5G-CORAL Technical Annex, Section 1-3 [64].

9.1 PoCs Relation and impact to Project Objective #2

TABLE 9-1: PROJECT OBJECTIVE#2 AND R&D TOPICS RELEVANT TO POCS

Objective #2: Design virtualised RAN functions, services, and applications for hosting in the 5G-CORAL Edge and Fog computing System (EFS)
R&D Topic 1: Explore the virtualization of RAN functions in the EFS for multiple RATs, develop their requirements, and assess their merits from an access convergence viewpoint.
R&D Topic 2: Specify EFS services for collection, aggregation, publishing, and use of radio and network context information by applications and possibly virtualized functions.
R&D Topic 3.1: Multi-RAT convergence function to optimize the traffic delivery across the multiple RATs (e.g., selection, aggregation, offloading).
R&D Topic 3.2: Authentication functions for facilitating session continuity, aggregation, and offloading across multiple RATs
R&D Topic 3.3: Network-assisted D2D function for discovery and communication of proximity networking and computing resources
R&D Topic 3.4: Network-offloading function that enables device and core network offloading by running some of its functions in the EFS
R&D Topic 4: Develop EFS applications using EFS services from multiple RATs and the transport and core networks to improve network KPIs and user QoE, such as: IoT gateway, augmented reality, user-targeted advertisements, and cars communication.
Verification: Validate selected EFS components through a system verification in the integration testbeds from WP4.

TABLE 9-2: HIGHLIGHTED RELATIONS AND IMPACTS TO PROJECT OBJECTIVE #2

PoC	R&D Topics	Verification
PoC #1 – Augmented Reality Navigation	**R&D Topic 2:** User will be able to offload the computation task (image recognition) to the Fog node available in the vicinity. Tasks are also offloaded to the neighboring Fog nodes if the primary Fog node is not able to support the incoming request.	Components have been integrated and verified in the Shopping Mall Testbed.
	R&D Topic 3: D2D protocols are used over the wireless links between Fog nodes to discover the nodes and distribute the jobs among the multiple nodes.	
PoC #2 – Virtual	**R&D Topic 2:** An EFS service called	The specific EFS

Reality	orientation service collects and publishes the user orientation which will be used to optimize the video streaming. **R&D Topic 3.4:** A number of video processing and streaming tasks are moved from the remote server and end user to the edge server and fog nodes, representing the EFS.	components have been intenerated, tested and validated by means of a proof of concept inside a shopping mall in Taiwan.
PoC #3 – Fog-assisted Robotics	**R&D Topic 3.2:** Demo 2: Wifi direct D2D is authenticated and established via WiFi **R&D Topic 3.3:** Demo 2: Wifi direct D2D used for local control feedback loop between the robots **R&D Topic 3.4:** Demo 1 and 2: Robot intelligence moved in the EFS	Components have been intenerated, tested and validated by means of a proof of concept inside a shopping mall in Taiwan.
PoC #4 – Multi-RAT IoT	**R&D Topic 2:** The virtualized 802.15.4 stack provides an IQ/Data service with the IQ samples from raw radio signal data. The service enables central monitoring of activity and interference on the radio channels. **R&D Topic 3.1:** The PoC is primarily focused on multi-RAT access for IoT. Multiple RATs including NB-IoT, IEEE 802.15.4 and LoRa stacks are softwarized and centralized at one edge node. The communication stacks are virtualized. It shows the IoT access convergence with multi-RAT processing aggregation. Measurement results show the performance is not impacted by running multiple RATs simultaneously on the Edge, showcasing the concept is feasible. **R&D Topic 4:** In the PoC, the IEEE 802.15.4 radio head software publishes the wideband IQ simples as the IQ/Data service to the MQTT broker. The Interference Analyzer application subscribes the IQ/Data service from the MQTT broker and perform noise and interference analysis using the IQ samples received. The benefit is to help understand more the interference issue and thus help improve radio resource management for interference mitigation.	Components have been intenerated, tested and validated by means of a proof of concept inside a shopping mall in Taiwan.

PoC #5 – SD-WAN	**R&D Topic 3.4:** By leveraging SD-WAN function offloading capabilities, traffic from a provider EFS domain is offloaded to its nearest PoS WebApp.	EFS Functions, Applications and services developed have been integrated, and validated by means of a proof of concept inside the shopping mall scenario in Taiwan.
PoC #6 – High-Speed Train	**R&D Topic 3.3:** Context information will be extracted from WiFi and Small Cells and will take part in classification users into groups which contribute into offloading and traffic loads.	Extraction of context information is performed in the emulation high-speed train. Where the QoS of passengers information are obtained successfully in the emulated high speed train in ITRI's lab, Taiwan.
	R&D Topic 4 Using EFS services and functions related to multi-RATs will improve network KPI especially in term of QoE.	
PoC #7 – Connected Cars	**R&D Topic 2:** Integration of the computing resources available at cars in the EFS.	Integration and verification of components in the Connected Car Testbed in Italy

9.2 PoCs Relation and Impact to Project Objective #3

TABLE 9-3: PROJECT OBJECTIVE#3 AND R&D TOPICS RELEVANT TO POCS

Objective #3: Design an Orchestration and Control system (OCS) for dynamic federation and optimized allocation of 5G-CORAL EFS resources
R & D Topic 1: Extend existing industrial frameworks for NFV, MEC, and fog to best suit dynamic environments where EFS resources are volatile.
R & D Topic 2: Develop federation mechanisms for EFS resources belonging to multiple owners and subject to different technical, business, and administrative requirements.
R & D Topic 3: Develop interfaces for automated deployment of EFS functions and applications
R & D Topic 4: Integrate the EFS with central clouds to enable instantiation and migration of virtual functions and applications between the EFS and central clouds.
R & D Topic 5: Develop orchestration and control algorithms for elastic placement and migration of EFS functions and optimized allocation of EFS resources
Verification: Validate selected OCS components through a system verification in the integration testbeds from WP4.

TABLE 9-4: HIGHLIGHTED RELATIONS AND IMPACTS TO PROJECT OBJECTIVE #3

PoCs	R&D Topics	Verification
PoC #1 – Augmented Reality Navigation	**R & D Topic 3:** OCS allows automated deployed of AR application as the user comes into the vicinity of the Fog node.	Components will be integrated and verified in the Shopping Mall Testbed.
PoC #2 – Virtual Reality	**R & D Topic 3:** An orchestration solution will be developed and adopted to conveniently distribute computing tasks between fog nodes and edge data centres as well as migrate the tasks based on traffic load and user mobility.	All the OCS components needed for this use case will be validated through a proof of concept showcased inside a shopping mall in Taiwan.
PoC #3 – Fog-assisted Robotics	**R&D Topic 3:** Demo 1 and 2: all the components composing the robot service are automatically deployed	Components have been integrated and verified in the Shopping Mall Testbed.
	R&D Topic 4: Demo 2: robot intelligence will be instantiated in the cloud when coordinated movements between the robots are not required to demonstrate the cloud to thing continuum.	The Topic 4 is planned to be done in the final technical review.
	R&D Topic 5: Demo 1: main intelligence over WiFi but close to the edge to reduce robot reaction time. Demo 2: main intelligence over WiFi for high-level fleet coordination and local feedback on the robot via Wifi direct D2D	
PoC #5 – SD-WAN	**R&D Topic 2:** This Demo tackle On Demand federation of resources	Components have been integrated and verified in the Shopping Mall Testbed.
PoC #6 – High-Speed Train	**R&D Topic 5:** This demo will optimize resources especially when service migration is needed	Service migration targeting the emulated high speed train is done and show up to 25% reduction in the total migration time
PoC #7 – Connected Cars	**R&D Topic 3:** The EFS function running on the RSU is managed by OCS	Integration and verification of components in the Connected Car Testbed

9.3 PoCs Relation and Impact to Project Objective #4

TABLE 9-5: PROJECT OBJECTIVE#4 AND R&D TOPICS RELEVANT TO POCS

Objective #4: Integrate and demonstrate 5G-CORAL technologies in large-scale testbeds making use of facilities offered by Taiwan, and measure their KPIs
R & D Topic 1: R & D Topic 1: Customise existing testbeds in Taiwan to meet the needs of 5G-CORAL proof-of-concept in large-scale deployments.
R & D Topic 2: Integrate and validate EFS and OCS in large-scale testbeds, such as shopping mall, high-speed train, and connected cars
R & D Topic 3: Demonstrate and trial multi-RAT access convergence and low latency applications, such as augmented reality and car safety, in real-world scenarios involving real users.
R & D Topic 4: Evaluate the performance of 5G-CORAL solution in the field through measurement of relevant KPIs on data rates and latency in low and high mobility environments.
Verification: Proof of concept experiments in Taiwan in a commercial 8,000 square meters shopping mall area with up to 15 people per 100 square meters.

TABLE 9-6: HIGHLIGHTED RELATIONS AND IMPACTS TO PROJECT OBJECTIVE #4

PoC	R&D Topic	Verification
PoC #1 – Augmented Reality Navigation	**R&D Topic 3:** AR Navigation will be demonstrated in the Shopping Mall Scenario.	Demonstration of AR navigation in the Shopping Mall testbed.
	R&D Topic 3: Low latency application "AR navigation" will be demonstrated in the real-world shopping mall testbed.	
PoC #2 – Virtual Reality	**R&D Topic 1:** The VR use case have been demonstrated inside the shopping mall in Taiwan.	The VR demonstration has been carried out in the shopping mall scenario.
	R&D Topic 2: Relevant KPIs has been measured through a demonstration in real-world conditions.	
PoC #3 – Fog-assisted Robotics	**R&D Topic 2:** Both robot demos integrates EFS and OCS and will be demonstrated on the large-scale testbed of the Shopping Mall Scenario	Demonstration of the Robot demo in the Shopping Mall testbed
	R&D Topic 3: Demo 1 and 2 demonstrate multi-RAT access convergence, it's not augmented reality and car safety, but edge robotics is equally valid	
PoC #4 – Multi-RAT IoT	**R&D Topic 2:** The PoC validates all key EFS components involved (i.e. EFS functions, applications, service platform and service), in the shopping mall	The Multi-RAT IoT has been demonstrated in the shopping mall scenario in

	context. The validation is done by lab tests and demonstrations.	Taiwan.
	R&D Topic 3: Demonstrated multi-RAT access convergence for massive IoT applications, i.e., low-power wireless, including NB-IoT, multi-hop mesh with IEEE 802.15.4 and LoRa.	
PoC #5 – SD-WAN	**R&D Topic 1:** The SD-WAN use case will be demonstrated in the shopping mall scenario in Taiwan.	The SD-WAN use case has been demonstrated in the shopping mall scenario in Taiwan.
PoC #6 – High-Speed Train	**R&D Topic 2:** High-speed train demo focusing on validation of EFS and OCS.	Emulation in ITRI's lab for high speed train and the validation results present EFS and OCS features
	R&D Topic 3: High-speed train demo demonstrate multi-RAT access convergence.	
PoC #7 – Connected Cars	**R&D Topic 1:** Demonstration of Enhanced Safety in connected car scenario in the Connected Car Testbed.	Integration and verification of components in the Connected Car Testbed in Turin (Italy) with the support of TI
	R&D Topic 2: Demonstration of delay sensitive application "Enhanced Safety in connected car" in a real-world testbed in Turin (Italy) with the support of TI	
	R&D Topic 4: field trail have been executed in real-world testbed in Turin (Italy)	

10 Conclusions and Future Directions

The work conducted within WP4 during the second year of the 5G-CORAL project has allowed us to test the technologies developed in 5G-CORAL WP2 and WP3 to develop Proof-of-Concepts supporting the various use cases identified in WP1 [1]. In particular, in this deliverable D4.2, we have presented a report with meaningful performance measures and concept validation with developed PoC testbeds, highlighting the added-value of 5G-CORAL solution.

We then presented seven PoCs (at the final stage of the project), namely: Augmented Reality Navigation, Virtual Reality, Fog-assisted Robotics, Multi-RAT IoT, High- Speed Train, and Connected cars. We presented the physical and logical architectures of each PoC, as well as detailed specifications for the integration and deployment of the individual components comprising each PoC. Validation results were also presented. This deliverable also provided a glance at the demonstrations carried out through the second year of the 5G-CORAL project.

Accordingly, within each section corresponding to each PoC has explained the challenges they faced during the second term of the project and how they were overcome. In this deliverable, possible future directions for further improvement and exploitation are also provided on individual PoC basis. Last but not least, we conclude this deliverable with a summary of the future directions per PoC in Table 10-1.

TABLE 10-1: SUMMARY OF FUTURE DIRECTIONS PER POC

PoCs	Future directions
PoC #1 – Augmented Reality Navigation	• Large-scale testbed: test the distributed computing functionality in a large-scale and for new applications. • Commercial trial: aim towards the trial deployment in Taipei main station
PoC #2 – Virtual Reality	• Latency reduction: (1) replace the streaming engine with a more optimized streaming solution to reduce a video stream processing time. (2) Developing solutions to move the stitching from the 360 camera to the Edge to reduce 360 stitching processing time. • Orchestration improvement: improve orchestration platform with short service setup time based on YAKS (a pub/sub protocol) [15].
PoC #3 – Fog-assisted Robotics	• Multi-RAT: investigate the possibilities to integrate different RATs (LTE, NR etc.) for remote coordinated control and understand the advantages and disadvantages of each technology. • Communication protocols: try different protocols to improve the cycle time of the mobile robots. • Federation: integrate with SD-WAN to demonstrate how the Fog-assisted robotics use case can easily federate with other domains.

PoC #4 – Multi-RAT IoT	• OCS: integrate with an orchestrator, e.g. Kubernetes and investigate new orchestration features. • Machine learning: investigate possibilities to incorporate machine learning techniques and capabilities, exploiting the Big Data potential of 5G-CORAL. • Large-scale testbed: scale up with more computing resources, more radio heads and more devices in a large-scale deployment.
PoC #5 – SD-WAN	• Machine learning: explore machine learning techniques to enhance the host mobility detection service to predict user movements between APs accurately. • Large-scale testbed: deploy the PoC in a large-scale scenario with more than two domains and serve real customers. • OCS improvement: enhance the orchestration mechanisms with arbitration algorithms that could place EFS functions, applications and services optimally.
PoC #6 – High-Speed Train	• OCS: Add OCS for automatic deployment of the vMME functions according to the demands.
PoC #7 – Connected Cars	• Latency reduction: investigate more efficient encoding schemes to further reduce latency

11 References

[1] 5G-CORAL, "5G-CORAL D1.1 - Initial system design, use cases, and requirements," 2018. [Online]. Available: http://5g-coral.eu/wp-content/uploads/2018/04/D1.1_final7760.pdf.

[2] 5G-CORAL, "5G-CORAL D4.1 - Testbed definition, integration and demonstration plans," 2018. [Online]. Available: http://5g-coral.eu/wp-content/uploads/2018/09/D4.112820.pdf.

[3] Google, "ARCore, Google," 2018. [Online]. Available: https://developers.google.com/ar/.

[4] Google, "Recognize and Augment Images, Google," 2018. [Online]. Available: https://developers.google.com/ar/develop/c/augmented-images/.

[5] W. Zhang, S. Lin, F. H. Bijarbooneh, H. F. Cheng and P. Hui, "Cloudar: A cloud-based framework for mobile augmented reality," in *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, 2017.

[6] T. Skeie, S. Johannessen and C. Brunner, "Ethernet in substation automation," *IEEE control systems magazine,* vol. 22, pp. 43-51, 2002.

[7] Z. Huang, W. Li, P. Hui and C. Peylo, "CloudRidAR: A cloud-based architecture for mobile augmented reality," in *Proceedings of the 2014 workshop on Mobile augmented reality and robotic technology-based systems*, 2014.

[8] Z. Technology, "WhereNet," 2018. [Online]. Available: https://www.zebra.com/us/en.html.

[9] P. Bahl, V. N. Padmanabhan, V. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," 2000.

[10] M. Graf, C. Timmerer and C. Mueller, "Towards bandwidth efficient adaptive streaming of omnidirectional video over http: Design, implementation, and evaluation," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, 2017.

[11] A. Corsaro and G. Baldoni, "fogØ5: Unifying the computing, networking and storage fabrics end-to-end," in *2018 3rd Cloudification of the Internet of Things (CIoT)*, 2018.

[12] W. M. Service, "Wowza Streaming Engine," 2017. [Online]. Available: https://www.wowza.com/products/streaming-engine.

[13] TechPowerUp, "TechPowerUp GPU-Z v2.23.0," 2019. [Online]. Available: https://www.techpowerup.com/download/techpowerup-gpu-z/.

[14] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *2016 IEEE International Symposium on Multimedia (ISM)*, 2016.

[15] A. Atolab, "YAKS: Yet-Another Key-Value Store," 2018. [Online]. Available: https://pypi.org/project/yaks/. [Accessed July 2019].

[16] G. Linux, "Hostapd," 2019. [Online]. Available: https://wiki.gentoo.org/wiki/Hostapd.

[17] 5TONIC, "AN OPEN RESEARCH AND INNOVATION LABORATORY FOCUSING ON 5G TECHNOLOGIES," 2018. [Online]. Available: https://www.5tonic.org/.

[18] 5G-CORAL, "5G-CORAL D3.2 - Refined design of 5G-CORAL orchestration and control system and future directions," 2019. [Online]. Available: http://5g-coral.eu/wp-content/uploads/2019/06/D3.2.pdf.

[19] I. E. E. E. S. Association and others, "IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std,* vol. 802, p. 2793, 2012.

[20] E. T. S. I. S. Association and others, "Digital Enhanced Cordless Telecommunications (DECT) Study on URLLC use cases of vertical industries for DECT evolution and DECT-2020," *Technical Report,* vol. 1.1.1, pp. 1-104, 2018.

[21] T. Instruments, "OMAP Wireless Connectivity NLCP WiFi Direct Configuration Scripts," 2018. [Online]. Available: http://processors.wiki.ti.com/index.php/OMAP_Wireless_Connectivity_NLCP_WiFi_Direct_Configuration_Scripts.

[22] G. Mohanarajah, D. Hunziker, R. D'Andrea and M. Waibel, "Rapyuta: A cloud robotics platform," *IEEE Transactions on Automation Science and Engineering,* vol. 12, pp. 481-493, 2014.

[23] L. Agostinho, L. Olivi, G. Feliciano, F. Paolieri, D. Rodrigues, E. Cardozo and E. Guimaraes, "A cloud computing environment for supporting networked robotics applications," in *2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, 2011.

[24] H.-J. Hong, P.-H. Tsai and C.-H. Hsu, "Dynamic module deployment in a fog computing platform," in *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2016.

[25] A. Machen, S. Wang, K. K. Leung, B. J. Ko and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications,* vol. 25, pp. 140-147, 2017.

[26] 5G-CORAL, "5G-CORAL D2.2 - Refined design of 5G-CORAL edge and fog computing system and future directions," 2019. [Online]. Available: http://5g-coral.eu/wp-content/uploads/2019/06/D2.2_Final.pdf.

[27] (. National Instruments, "Ettus Research," 2018. [Online]. Available: https://www.ettus.com/.

[28] T. G. N. U. R. Foundation, "GNU Radio," 2018. [Online]. Available: https://www.gnuradio.org/.

[29] (. National Instruments, "USRP-B210," 2018. [Online]. Available: https://www.ettus.com/all-products/ub210-kit/.

[30] Zolertia, "Firefly board," 2018. [Online]. Available: https://zolertia.io/product/firefly/.

[31] Pycom, "Fipy," 2018. [Online]. Available: https://pycom.io/product/fipy/.

[32] Github, "Contiki-NG: The OS for Next Generation IoT Devices," 2018. [Online]. Available: https://github.com/contiki-ng/contiki-ng.

[33] Github, "GNU Radio," 2018. [Online]. Available: https://github.com/gnuradio/gnuradio.

[34] Github, "GR Lora," 2018. [Online]. Available: https://github.com/rpp0/gr-lora.

[35] Ubuntuiperf3, "IPERF3 Manual," [Online]. Available: http://manpages.ubuntu.com/manpages/xenial/man1/iperf3.1.html. [Accessed 06 2019].

[36] R. Vickerman, "High-speed rail in Europe: experience and issues for future development," *The annals of regional science,* vol. 31, pp. 21-38, 1997.

[37] H.-W. Chang, M.-C. Tseng, S.-Y. Chen, M.-H. Cheng and S.-K. Wen, "Field trial results for integrated WiMAX and radio-over-fiber systems on high speed rail," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011.

[38] C.-L. Lai, S. B. Chundrigar, S. T. Talat and H.-W. Chang, "Staying Connected on the Go: Network Sharing Based on Multioperator Traffic Inside Moving Transportation," *IEEE Vehicular Technology Magazine,* vol. 13, pp. 97-104, 2018.

[39] E. Grigoreva, J. Xu and W. Kellerer, "Reducing mobility management signaling for automotive users in LTE advanced," in *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2017.

[40] N. S. Networks, "Signaling is growing 50% faster than data traffic," 2012.

[41] E. U. T. R. A. Network, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," *EUTRA Network,* 2011.

[42] I. Widjaja, P. Bosch and H. La Roche, "Comparison of MME signaling loads for long-term-evolution architectures," in *2009 IEEE 70th Vehicular Technology Conference Fall*, 2009.

[43] H. T. Larasati, F. H. Ilma, B. Nuhamara, A. Mustafa, R. Hakimi and E. Mulyana, "Performance evaluation of handover association mechanisms in SDN-based wireless network," in *2017 3rd International Conference on Wireless and Telematics (ICWT)*, 2017.

[44] F. Laassiri, M. Moughit and N. Idboufker, "An Improvement of Performance Handover in Worldwide Interoperability for Microwave Access using Software Defined Network," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS,* vol. 9, pp. 84-88, 2018.

[45] W. Nasrin and J. Xie, "SharedMEC: Sharing clouds to support user mobility in mobile edge computing," in *2018 IEEE International Conference on Communications (ICC)*, 2018.

[46] C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. S. Lam and M. Rosenblum, "Optimizing the migration of virtual computers," *ACM SIGOPS Operating Systems Review,* vol. 36, pp. 377-390, 2002.

[47] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on*

Networked Systems Design & Implementation-Volume 2, 2005.

[48] M. R. Hines, U. Deshpande and K. Gopalan, "Post-copy live migration of virtual machines," *ACM SIGOPS operating systems review,* vol. 43, pp. 14-26, 2009.

[49] V. Medina and J. M. García, "A survey of migration mechanisms of virtual machines," *ACM Computing Surveys (CSUR),* vol. 46, p. 30, 2014.

[50] J. Nider and M. Rapoport, "Cross-isa container migration," in *Proceedings of the 9th ACM International on Systems and Storage Conference*, 2016.

[51] R. Morabito, I. Farris, A. Iera and T. Taleb, "Evaluating performance of containerized IoT services for clustered devices at the network edge," *IEEE Internet of Things Journal,* vol. 4, pp. 1019-1030, 2017.

[52] E. T. S. I. S. Association and others, "Digital Cellular Telecommunications System (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Policy and charging control architecture," *Technical Specification,* vol. 23.203, pp. 1-221, 2014.

[53] NextEPC, "Open Source," 2019. [Online]. Available: https://nextepc.org/.

[54] E. T. S. I. E. Standard, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.02_60/en_30263702v010302p.pdf.

[55] M. Lauridsen, L. C. Gimenez, I. Rodriguez, T. B. Sørensen and P. E. Mogensen, "From LTE to 5G for Connected Mobility.," *IEEE Communications Magazine,* vol. 55, pp. 156-162, 2017.

[56] 3. P. P. S. 22.885, "Study on LTE support for Vehicle-to-Everything (V2X) services," 2015. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2898.

[57] MintBox, "Linux Mint Store," 2018. [Online]. Available: https://linuxmint.com/store_mintbox.php.

[58] S. Alliance, "OpenAirInterface," 2018. [Online]. Available: https://www.openairinterface.org/.

[59] 3. P. P. S. 22.185, "LTE Service requirements for V2X services," 2015. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/122100_122199/122185/14.03.00_60/ts_122185v140300p.pdf.

[60] IETF, "IETF Tools," [Online]. Available: https://tools.ietf.org/html/rfc7049#page-5.

[61] Google, "Protocol buffer," [Online]. Available: https://developers.google.com/protocol-buffers/.

[62] 5GAA, "V2X Functional andPerformance Test Report;Test Procedures and Results," Nov 2018. [Online]. Available: https://5gaa.org/wp-content/uploads/2018/11/P-180106-V2X-Functional-and-Performance-Test-Report_Final_051118.pdf.

[63] 5. Americas, "Cellular communications V2X towards 5G," March 2018. [Online]. Available: https://www.3g4g.co.uk/5G/5Gtech_6008_2018_03_5G_Americas_White_Paper_Cellular_V2X_Communications_Towards_5G__Final_for_Distribution.pdf.

[64] 5G-Coral consortium., "Project “5G-CORAL”: A 5G Convergent Virtualised Radio Access Network Living at the Edge. Technical Annex, Section 1-3," 2017.