



H2020 5Growth Project
Grant No. 856709

D4.4: Final validation and verification report

Abstract

This deliverable constitutes the final report containing the results of the project verification and validation campaigns at the pilot sites, and the findings derived from them.



Document properties

Document number	D4.4
Document title	Final validation and verification report
Document responsible	Paola Iovanna (TEI)
Document editor	Giulio Bottari (TEI)
Editorial team	Carlos Marques (ALB), Carlos Guimarães (UC3M), Giada Landi (NXW), Juan Brenes (NXW), Nicola Venturi (NXW), Hugo Martins (EFACEC_E), Aldo Trindade (EFACEC_E), Paulo Paixão (EFACEC_S), Rui Manuel Antunes (EFACEC_S), Pedro Elísio (EFACEC_S), Rui Antunes (EFACEC_S), Nikolaos Koursioupas (NKUA), Diego Lopez (TID), Matteo Pergolesi (TELCA), Agustín Caparrós (TELCA), Antonio Cobos (TELCA), Teresa Giner (TELCA), Olivier Tilmans (NBL), Chia-Yu Chang (NBL), Luis Bellido (UPM), Ignacio Domínguez (UPM), David Fernández (UPM), Daniel González (UPM), Giulio Bottari (TEI), Fabio Ubaldi (TEI), Stefano Stracca (TEI), Paola Iovanna (TEI), Giorgio Bertinetti (TIM), Andrea Bragagnini (TIM), Mauro Castagno (TIM), Annachiara Pagano (TIM), Andrea Sgambelluri (SSSA), Luca Valcarengi (SSSA), Enrico Forestieri (SSSA), Daniel Corujo (ITAV), Vítor Cunha (ITAV), João Fonseca (ITAV), João Alegria (ITAV), José Quevedo (ITAV), Rui Aguiar (ITAV), Diogo Gomes (ITAV), David Santos (ITAV), António Nogueira (ITAV), Luis Nero (ITAV), Diego San Cristóbal (ERC), Fernando Beltrán (ERC), Josep Mangles-Bafalluy (CTTC), Engin Zeydan (CTTC), Jorge Baranda (CTTC), Luca Vettori (CTTC), Ramon Casellas (CTTC), Manuel Requena (CTTC), Josep M. Fàbrega (CTTC), Claudio Casetti (POLITO), Carla Fabiana Chiasserini (POLITO), Corrado Puligheddu (POLITO), Federico Mungari (POLITO), Francesco Raviglione (POLITO), Jesús Alonso (INNO)
Target dissemination level	PU
Status of the document	Final
Version	1.0
Delivery date	November 30, 2021
Actual delivery date	December 15, 2021 (as agreed with the PO)

Production properties

Reviewers	Ricardo Martinez (CTTC), Josep Xavier Salvat (NEC)
------------------	--

Disclaimer

This document has been produced in the context of the 5Growth Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement N° H2020-856709.

All information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors' view.

Contents

List of Figures.....	5
List of Tables.....	8
List of Acronyms	10
Executive Summary and Key Contributions.....	12
1. Introduction	14
2. Measurement Tooling and Methodology	16
2.1. Data Infrastructure	16
2.1.1. Architecture.....	16
2.1.2. Task Management.....	17
2.1.3. Data Sources.....	19
2.1.4. Data Consumers.....	23
2.1.5. YANG-based drivers	26
2.2. Experiment Catalogue.....	29
2.3. Available Tools.....	29
2.3.1. 5Growth Log Parser and its Integration with SDA and 5Growth Stack.....	29
2.3.2. Telemetry for P4 programmable switches	32
3. Pilot Integration.....	43
3.1. INNOVALIA Pilot.....	43
3.1.1. Technical Requirements and Related KPIs.....	43
3.1.2. ICT-17 and 5Gr Platform Integration	43
3.1.3. Measurement Procedures	44
3.1.4. Innovation Validation and Impact on KPIs.....	47
3.2. COMAU Pilot	52
3.2.1. Technical Requirements and Related KPIs.....	52
3.2.2. ICT-17 and 5Gr Platform Integration	53
3.2.3. Measurement Procedures	53
3.2.4. Innovation Validation and Impact on KPIs.....	56
3.3. EFACEC_S Pilot.....	56
3.3.1. Technical Requirements and Related KPIs.....	56
3.3.2. ICT-17 and 5Gr Platform Integration	57
3.3.3. Measurement Procedures	58

3.3.4. Innovation Validation and Impact on KPIs.....	58
3.4. EFACEC_E Pilot.....	67
3.4.1. Technical Requirements and Related KPIs.....	67
3.4.2. ICT-17 and 5Gr Platform Integration	69
3.4.3. Measurement Procedures	69
3.4.4. Innovation Validation and Impact on KPIs.....	70
4. Report of the Third Validation Campaign	71
4.1. Industry 4.0 pilot – INNOVALIA.....	71
4.1.1. Use Case 1: Connected worker remote operation of quality equipment.....	71
4.1.2. Use Case 2: Connected worker: Augmented Zero Defect Manufacturing (ZDM) Decision Support System (DSS).....	83
4.2. Industry 4.0 pilot – COMAU	92
4.2.1. Use Case 1: Digital twin apps.....	92
4.2.2. Use Case 2: Telemetry/monitoring apps.....	96
4.2.3. Use Case 3: Digital tutorial and remote support	96
4.3. Transportation pilot – EFACEC_S	96
4.3.1. Use Case 1: Safety critical communications.....	97
4.3.2. Use Case 2: Non-safety critical communications.....	102
4.4. Energy pilot – EFACEC_E	110
4.4.1. Use Case 1: Advanced Monitoring and Maintenance Support for Secondary Substation MV/LV Distribution Substation	110
4.4.2. Use Case 2: Advanced critical signal and data exchange across wide smart metering and measurement infrastructures.....	117
5. Conclusions.....	125
6. References	126



List of Figures

Figure 1: Microservice-based architecture of the sda	16
Figure 2: Information Model of a Generic Task.....	18
Figure 3: Information Model of Prometheus Data Source.....	20
Figure 4: Information Model of Network Telemetry Data Source	21
Figure 5: Information Model of Kafka Data Source.....	23
Figure 6: Information Model of Prometheus Exporter Consumer	25
Figure 7: Data Source Driver	27
Figure 8: Data Consumer Driver.....	28
Figure 9: Transformer Driver	29
Figure 10: Global Architecture for 5Growth Log Parser.....	30
Figure 11: Physical Hardware Stages	33
Figure 12: Proof of Concept Scenario for P4 Switch Telemetry.....	35
Figure 13: Openconfig model example.....	36
Figure 14: GNMI response for a get command	37
Figure 15: P4 switches-sda integration data flow	38
Figure 16: Capabilities from P4-stratum switch via gnmic.....	38
Figure 17: TCP iperf between Host1 and Host2.....	39
Figure 18: First response for out-octets query.....	39
Figure 19: Last response for out-octets query	40
Figure 20: Output for Out-octets query before and after use case operations.....	40
Figure 21: Output for Out-discards metric query.....	41
Figure 22: Scenario on Telefonica premises.....	42
Figure 23: Data rate measurements for Test #1	50
Figure 24: Latency measurements for Test #1.....	50
Figure 25: Data Rate measurements for Test #5	51
Figure 26: RTT Latency measurements for Test #5.....	51
Figure 27: Measurement Setup in the COMAU Mobile Infrastructure.....	54
Figure 28: Dynamic Measurement Scenario in COMAU	55
Figure 29: Interdomain+MTD Scenario.....	59

Figure 30: InterDOMAIN+MTD InNovation SERVICE Signalling Diagram.....	60
Figure 31: Delays associated with the operations at the VS level.....	61
Figure 32: 5Gr-VS Instantiation Metrics interDomain Innovation (I).....	62
Figure 33: 5Gr-VS Instantiation Metrics interDomain Innovation (II)	63
Figure 34: Delays of the operations over the VNFs in each domain.....	64
Figure 35: Delays of the operations in the 5g-Vinni Platform(Sonata)	65
Figure 36: delays of the operations in the OSM enabled Domain.....	66
Figure 37: 5Gr-VS instantiation metrics UC1.....	71
Figure 38: 5Gr-SO instantiation metrics UC1 – total instantiation and Core MANO Wrapper Time..	72
Figure 39: 5GR-SO UC1 Time profiling of Service Orchestrator Engine (SOE) operations	73
Figure 40: 5GR-SO UC1 Time profiling of Resource Orchestrator Engine (ROE) operations.....	74
Figure 41: UC1 RTT Latency and Jitter for Scan Results Traffic.....	76
Figure 42: User Data Rate for Scan Results Traffic.....	77
Figure 43: RTT Latency and Jitter for Commands Traffic.....	78
Figure 44: User Data Rate for Commands Traffic.....	80
Figure 45: Video Streaming Latency, Jitter and Data Rate.....	82
Figure 46: 5Gr-VS Instantiation metrics UC2	83
Figure 47: 5GR-SO Instantiation Metrics UC2 – Total Instantiation and Core MANO Wrapper time	84
Figure 48: 5GR-SO UC2 Time Profiling of Service Orchestrator Engine (SOE) operations	85
Figure 49: 5GR-SO UC2 Time profiling of Resource Orchestrator Engine (ROE) operations.....	85
Figure 50: User Data Rate of AGV related Traffic	87
Figure 51: UC2 RTT Latency and Jitter for Scan Result traffic.....	88
Figure 52: UC2 User Data Rate for Scan Result Traffic	89
Figure 53: UC2 RTT Latency and Jitter for traffic	90
Figure 54: UC2 User Data Rate from INNO VM to CMM	91
Figure 55: E2E Latency	93
Figure 56: Virtual Robot superimposed to the real one	94
Figure 57: Grafana screenshot, at the monitoring platform.....	95
Figure 58: Distribution of latency values	95
Figure 59: Histogram of RTT.....	97

Figure 60: Histogram of RTT.....	100
Figure 61: Latency Graph.....	102
Figure 62: Jitter Graph	102
Figure 63: Histogram of RTT for 1200 Bytes Packet Size – static test.....	103
Figure 64: Histogram of RTT for 1200 Bytes Packet Size – dynamic test.....	104
Figure 65: GUI of the Train Driver Console.....	106
Figure 66: Histogram of RTT for 1200 Bytes Packet Size – static test.....	107
Figure 67: Histogram of RTT for 1200 bytes Packet Size – Dynamic test.....	108
Figure 68: Use Case 1 Setup using 5G SA Network with ASOCS RAN and OPEN5GCORE	111
Figure 69: Histogram RTT (1000 Bytes Packet Size)	111
Figure 70: RTT limits for different packet size	112
Figure 71: 5GROWTH Monitoring Platform – Latency in a 15 minute period.....	112
Figure 72: 5GROWTH Monitoring Platform – Packet Loss in a 15 minute period.....	113
Figure 73: Throughput measured for UDP protocol.....	113
Figure 74: Throughput measured for TCP protocol	114
Figure 75: 5GROWTH Monitoring Platform – Throughput in a 15 minutes period	114
Figure 76: 5GROWTH Monitoring Platform – Jitter in a 15 minute period	114
Figure 77: 5GROWTH Monitoring Platform – Packet loss in a 15 minute period.....	115
Figure 78: Histogram of RTT (1000 bytes packet size)	115
Figure 79: RTT limits for different packet sizes	116
Figure 80: Throughput measured for UDP protocol.....	116
Figure 81: Use Case 2 setup using 5G SA Network with ASOCS RAN and OPEN5GCORE.....	118
Figure 82: Histogram of RTT (1000 BYTES PACKET SIZE)	118
Figure 83: RTT limits for different packet sizes	119
Figure 84: 5GROWTH Monitoring Platform – Latency in a 15 minute period.....	119
Figure 85: 5GROWTH Monitoring Platform – Packet loss in a 15 minute period.....	120
Figure 86: Histogram of latency for LAST-GASP in a 10 minute period (UDP TRAP - Application).	121
Figure 87: Histogram of RTT between LVS3 and UA NTP server (1000 bytes packet size)	122
Figure 88: Histogram of jitter between LVS3 and UA NTP server (1000 bytes packet size)	122
Figure 89: Sync Difference between LVS3 in a 60 minute period (UDP TRAP - Application).....	123

List of Tables

Table 1: Data Model Fields of request to anomaly detection module.....	23
Table 2: Data Model Fields of response from anomaly detection module.....	24
Table 3: 5GR-VS Metric parameters	31
Table 4: 5GR-VS instance related Metrics	31
Table 5: 5GR-VS Network slice and network service Metrics	32
Table 6: KPI Requirements for INNOVALIA Use Cases	43
Table 7: Core KPIs Measured in INNOVALIA Pilot	43
Table 8: INNOVALIA UC1 Specific Service KPIs, Core KPIs and Validation Methodology	44
Table 9: INNOVALIA UC2 Specific Service KPIs, Core KPIs and Validation Methodology	45
Table 10: CKPI Measurement Methodology	46
Table 11: Test Plan for the Validation of I8 in INNOVALIA Pilot	47
Table 12: Results for the Validation of I8 in INNOVALIA Pilot	49
Table 13: KPI Requirements for COMAU Use Cases.....	52
Table 14: Core KPIS Validated in COMAU Pilot.....	53
Table 15: KPI Requirements for EFACEC_S Use Cases	56
Table 16: Core KPIS Validated IN EFACEC_S Pilot	57
Table 17: KPI Requirements for EFACEC_E Use Cases	67
Table 18: CORE KPIs validated in EFACEC_E Pilot.....	68
Table 19: INNOVALIA UC1 obtained KPIs.....	82
Table 20: INNOVALIA UC2 AGV Traffic Measurements	91
Table 21: RTT end-to-end in EFACEC_S UC1.....	97
Table 22: Throughputs Measured for UDP Protocol in EFACEC_S UC1	98
Table 23: EFACEC_S UC1 Specific Service KPIs, Core KPIs and Validation Methodology.....	98
Table 24: RTT end-to-end in EFACEC_S UC1 (with E2E secure tunneling).....	100
Table 25: Throughputs Measured for UDP Protocol in EFACEC_S UC1	101
Table 26: EFACEC_S UC1 Specific Service KPIs, Core KPIs and Validation Methodology.....	101
Table 27: Bandwidth for UDP traffic protocol in EFACEC_S UC2.....	103
Table 28: Bandwidth for UDP traffic protocol in EFACEC_S UC2.....	105
Table 29: EFACEC_S UC2 Specific Service KPIs, core KPIs and Validation Methodology	105

Table 30: Bandwidth for UDP traffic protocol in EFACEC_S UC2	108
Table 31: Bandwidth for UDP traffic protocol in EFACEC_S UC2	109
Table 32: EFACEC_S UC2 Specific Service KPIs, core KPIs and Validation Methodology	109
Table 33: EFACEC_E UC1 Specific Service KPIs, Core KPIs and Validation Methodology.....	116
Table 34: EFACEC UC2 Specific Service KPIs, Core KPIs and Validation Methodology	123



List of Acronyms

5Gr - 5Growth

5Gr-VoMS - 5Growth Vertical-oriented Monitoring System

AD – Anomaly Detection Module

AGV – Automated Guided Vehicle

API – Application Programming Interface

AR – Augmented Reality

CKPI – Core 5G KPI

CMM – Coordinate-Measuring Machine

CPE – Customer Premises Equipment

CSMF – Communications Service Management Function

DCM – Data Collection Manager

DL - Downlink

DSS – Decision Support System

E2E – End-to-End

eMBB – enhanced Mobile Broadband

IIoT – Industrial Internet of Things

IM – Information Model

INT - In-band Network Telemetry

Ix – Innovation number x (WP2)

KPI – Key Performance Indicator

LV – Low-Voltage

LX – Level Crossing

mMTC – massive Machine Type Communications

NGSI-LD - Service Interfaces – Linked Data

NS – Network Service

RAN – Radio Access Network

RTT – Round Trip Time

SDA – Semantic Data Aggregator

SKPI – Service KPI

SLA – Service Level Agreement

SO – Service Orchestrator

UC – Use Case

UE – User Equipment

UL - Uplink

URLLC – Ultra-Reliable Low Latency Communication

vEPC – virtual Evolved Packet Core

VIM – Virtualized Infrastructure Management

VM – Virtual Machine

VNF – Virtual Network Function

VPN – Virtual Private Network

VS – Vertical Slicer

ZDM – Zero Defect Manufacturing

Executive Summary and Key Contributions

This deliverable is the final one of WP4 in a series of validation campaign, from D4.2 [5] to D4.4, dedicated to verifying the pilot results conducted at the vertical premises, and aligned with the successive releases of the experimental environments developed in WP3. They include the integration of the innovations designed and developed in WP2. As far as the experimental environments are concerned, this deliverable is aligned with D3.6 [3].

After a first introductory Section, the second Section of this deliverable details the measurement tooling and methodology:

- The **data infrastructure** used to collect, process, and publish the data generated by the validation campaigns of the project.
- The **experiment catalogue** designed to ensure the reproducibility and repeatability of the experimental results.
- The **available tools** that allow the automation of the verification procedures, the processing of the measured data and the reproducibility of the verification results

The third Section presents the **technical requirements** and the **KPIs** under consideration, going into detail about the **measurement procedures**. The Section also addresses the integration of monitoring and other tools from ICT-17 platforms and 5Growth. Finally, the Section ends with the description of the subset of WP2 innovations which are related to the validation procedures.

The fourth Section reports the results of the **third validation campaign** carried out for each pilot, including the constraints imposed, tools used, the measurement results and the analysis of the results obtained.

D4.4 follows the structure defined for D4.2 [5] and D4.3 [6], with special emphasis and detail on the verification results, including the analysis of the applicable innovations for each pilot. It also presents a full snapshot of the latest achievements for methodology and tooling, that have reached a mature stage and the experimental capacity of the different pilots has consolidated.

The main contributions of this deliverable, in terms of top achievements from each of the pilots, can be listed as:

- INNOVALIA pilot
 - Two Use Cases fully integrated in vertical premises at Bilbao, including the full 5Growth platform, and leveraging on the 5G-EVE platform and infrastructure located at Madrid.
 - Vertical applications have been virtualized to work decoupled from the hardware and to allow dynamic orchestration and flexibility to work on different infrastructure setups.
 - 5G technology has been experimented with two alternative deployment options and two separate frequency bands.

- Pilot ("5Growth: Industry 4.0 Remote Operation of Metrology Machinery over 5G") selected among the top ten of the 5GPPP Infrastructure Trials and Pilots brochure No. 3 in 2021.
- COMAU pilot
 - Three Use Cases fully integrated in vertical and operator premises, including the transport network and monitoring platform.
 - KPI validated on the integrated 5G technologies (radio, transport, and cloud) with related E2E orchestration and monitoring (on site and by remote via monitoring platform).
 - Pilot ("5Growth - Industry 4.0: Low-latency on a shared Network") selected among the top ten of the 5GPPP Infrastructure Trials and Pilots brochure No. 2 in 2020.
 - Pilot ("5Growth - E2E Transport-aware Orchestration") selected among the top ten of the 5GPPP Infrastructure Trials and Pilots brochure No. 3 in 2021.
- EFACEC_S pilot
 - Two Use Cases fully integrated in vertical premises, including the monitoring platform.
 - Business, functional and technical validation as well as KPI validation on the integrated 5G technologies and monitoring (on site and by remote via monitoring platform).
 - The 5G level crossing solution is incorporated in the EFACEC product roadmap.
 - Good feedback from main stakeholders such as partners (for instance Frauscher) end-users (for instance infrastructure owners – Administração do Porto de Aveiro/Aveiro Harbour Administration) and private-operators (Medway -freight Operator).
- EFACEC_E pilot
 - Two Use Cases fully integrated in vertical premises, including the monitoring platform.
 - Business, Functional and technical validation as well as KPI validation on the integrated 5G technologies and monitoring (on site and by remote via monitoring platform).
 - Important step ahead considering the development prototyping and PoC of next-generation of EFACEC's products for the secondary distribution substation and the low voltage distribution smart grid.

Additionally, each pilot had at least one selected (WP2) innovation integrated and validated.

1. Introduction

The main objective of WP4 is to evaluate core (network) and service (application) KPIs through the 5Growth field trials (pilots) to validate the applicability of 5G technologies as well as the selected 5Growth innovations designed in WP2 to the different use cases (UC) considered by the various project pilots.

The evaluation was based on the initial definition of measurement and testing methodologies. WP4 provided a detailed description of the said methodologies and a set of tools automate the verification procedures, the processing of the measurement data, and the reproducibility of the results obtained in three specific validation campaigns. These campaigns were carried out in different periods of the project lifetime to have intermediate validation checkpoints during the progress of the pilot deployments.

As a result, WP4 has ensured that the verification of all pilots has been carried out accurately and with a systematic approach to ensure the scientific quality of the project results, and to help ensuring that the findings and tools can be applied beyond the project lifetime by reusing knowledge and/or assets built in the pilots.

The work of WP4 has been and is aligned with the activities of WP2 and WP3 to enforce the innovation, deployment, and validation steps in three cycles during the project lifetime. Specifically, WP4 applies the appropriate releases created by WP3, which include the modules developed in WP2, and selects the proper experimental environment for the corresponding cycle, comprising:

1. Data sources and consumers, including the metadata for them.
2. External tools, available through the project tool's catalogue.
3. Specific measurement methods documented in the deliverables.
4. Pilot use cases are validated as part of successive releases of the experimental environments (updated environments are described in D3.6 [3]) and reported in the associated deliverables, including the environment characteristics and the outcome of the verifications performed during the period.
5. The verification methodology and associated tooling provide a common substrate for the execution of the verification campaigns, to:
 - Enhance the quality of the verification campaigns.
 - Guarantee repeatability and reproducibility as essential features of the experimental reports.
 - Contribute to a better understanding of the mechanisms for aggregating and processing telemetry data for data-driven network management.

Ultimately, the validation campaigns had the scope of assessing the fulfilment of the Key Performance Indicators (KPIs) starting from the Service KPIs, which are high-level abstractions of the business requirements (not directly measurable) and directing the validation towards the related Core KPIs (e.g., latency, throughput, availability, etc.). The Core KPIs were measured by either specific test setups or alternative methodologies.

This deliverable is organized as follows:

- **Section 2** describes the measurement tooling and methodology. This Section includes the data infrastructure responsible for enabling an experimental data processing framework that collects measurements from the monitored network infrastructures, the 5Growth experiment catalogue including the Information Models (IMs) for the 5Growth experiment descriptors, and the new tools added to the toolset to support the pilot validation campaigns.
- **Section 3** reports, for each pilot, the technical requirements and KPIs measured in the pilot validation campaigns, the integration process to make the monitoring data collected through experiments in 5G EVE and/or 5G VINNI, available to the 5Growth Monitoring Platform, and the innovations incorporated in each pilot or use case that are related to performance assessment.
- **Section 4** reports, for each pilot, the results of the third validation campaign in terms of measured values and tested performances.

2. Measurement Tooling and Methodology

2.1. Data Infrastructure

The scope of the data infrastructure, introduced in previous deliverables, enables an experimental data processing framework to collect measurements from the monitored network infrastructures and aggregates the monitoring data. The data infrastructure can deliver aggregated data for further analysis. The Semantic Data Aggregator (SDA) is the monitoring framework that the data infrastructure builds upon. This deliverable introduces a new release of the SDA addressing the evolution of the architecture, the addition of new features, and the integration of data consumers.

2.1.1. Architecture

The incorporation of new features in the SDA framework increases the complexity and coordination of the development tasks. To this end, the SDA moved towards a microservice-based architecture where a combination of lightweight pieces of software provides the framework's features. This approach allows for the developers to implement the business logic at their own pace.

In addition, the SDA embraces cloud-native technologies such as Docker [7] and docker-compose [8]. The microservices that comprise the SDA are deployed as Docker containers, whereas docker-compose is a tool that eases the orchestration and configurations of Docker containers in a single machine.

In this release of the SDA, new building blocks – boxes in bold – are introduced as depicted below:

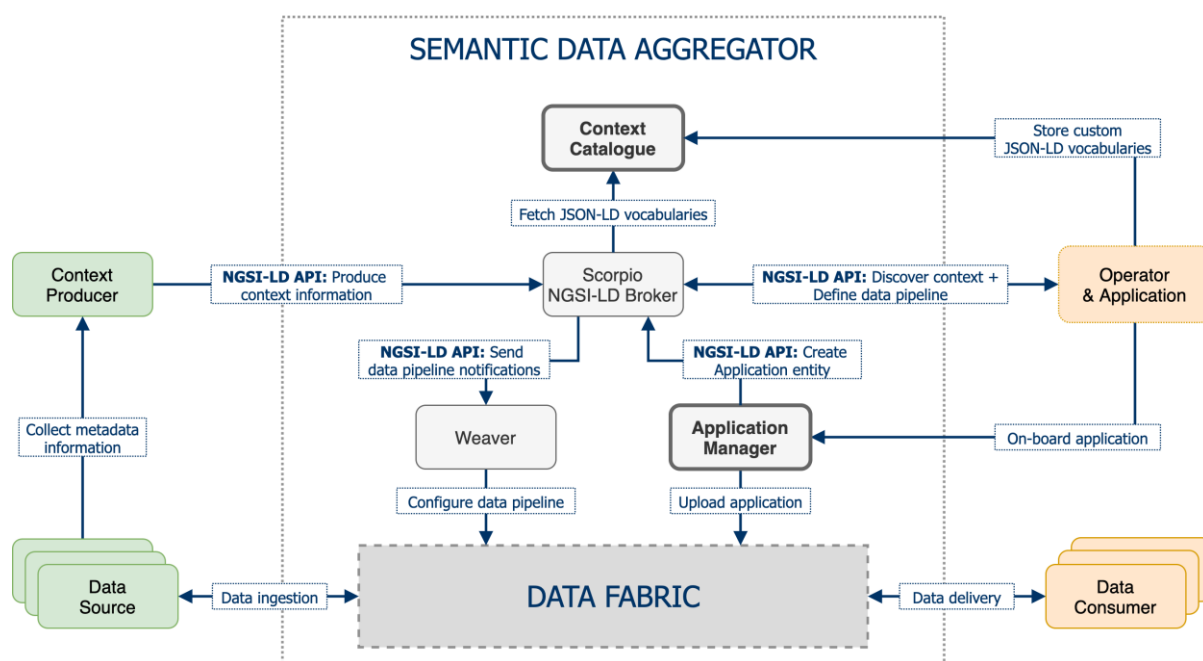


FIGURE 1: MICROSERVICE-BASED ARCHITECTURE OF THE SDA

Figure 1 presents the Data Fabric building block as a black box as it has not been modified since the release reported in D4.3 [6]. Similarly, Scorpio remains as the chosen implementation of a Next Generation Service Interfaces – Linked Data (NGSI-LD) Context Broker. On the other hand, the Weaver component has evolved to consume new kinds of context information and configure the Data Fabric accordingly, though further details will be provided in Section 2.1.2. Therefore, the new contributions introduced in this release are the *Application Manager* and *Context Catalogue* building blocks.

2.1.1.1. Application Manager

The Application Manager service handles the on-boarding of applications into the SDA. The Application Manager leverages the FastAPI [9] framework to implement a REST API through which users can upload custom Apache Flink JARs and Apache NiFi templates. During the on-boarding process, the Application Manager service is the responsible for validating the contents of the application, checking if it already exists, uploading the application into the target platform – either NiFi or Flink – and recording the associated metadata as new context information in the Scorpio Context Broker. Thus, this new service introduces a new layer that abstracts users from the interaction with the platforms running within the Data Fabric. Additionally, the Application Manager establishes the baseline for user control access in the SDA, though the implementation of this mechanism is left for future releases.

Additionally, the Application Manager offers a second feature which is an application catalogue. Once an application has been successfully on-boarded in the SDA, the Application Manager places a copy of the code of the application in a static web server – application catalogue – through which users can find the code of the application. For instance, by leveraging this feature, non-developer users can easily examine the code of an Apache Flink JAR that is currently running in the SDA without any interaction with the Flink platform itself.

2.1.1.2. Context Catalogue

Service that contains collections of JSON-LD vocabularies that can be used upon exchanging information with the NGSI-LD protocol. The Context Catalogue service eases the specification of vocabularies in the requests to the NGSI-LD API by pointing the HTTP Link header to the URL of a particular vocabulary collection in the service. This is a more efficient method as it avoids including the whole used JSON-LD vocabulary as part of the message body in the request.

2.1.2. Task Management

This release of the SDA introduces the *Task Management* feature, which enables building data pipelines in the Data Fabric. This new feature is based on the definition of two new NGSI-LD entities, namely, *Application* and *Task*.

The Application entity captures the metadata associated with what the SDA understands as an application that can be executed in the Data Fabric: a JAR in Apache Flink and a template in Apache

NiFi. The metadata contains relevant information, such as the internal ID in the respective platform or a URL to the Application Manager's catalogue to retrieve the code of the on-boarded application.

The Task entity captures the metadata associated with the execution of an application. Depending on the target platform, a Task could be a job in the case of Apache Flink or a flow in the case of Apache NiFi. The Task entity records relevant context information ranging from the internal ID and the execution arguments to the execution state. Furthermore, the Task entity always includes the *hasInput* and *hasOutput* NGSI-LD relationships. These relationships convey the input and the output of data associated to the execution of a Task. A generic NGSI-LD information model is depicted in Figure 2. Samples of information models illustrating Tasks along with the supported data sources and data consumers are provided in the following subSections.

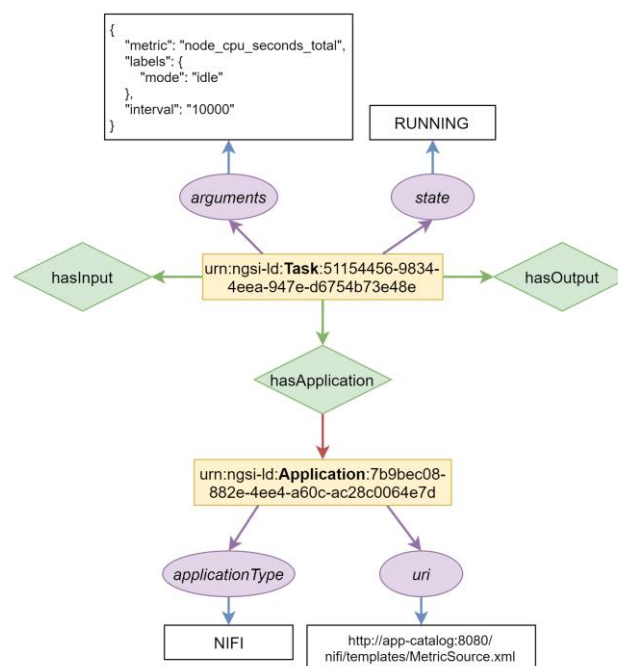


FIGURE 2: INFORMATION MODEL OF A GENERIC TASK

The Task Management feature proposes a top-bottom approach where operators – or applications – create instances of the Task entity where the arguments and the relationships to other data assets are specified. On the other hand, the Weaver component is subscribed to the Scorpio Context Broker to receive notifications about updates on instances of the Task entity – as shown in Figure 1. Upon receiving a notification, the Weaver parses all the associated metadata – i.e., information related to Task and entities from their relationships – and configures the execution of an application in the target platform, e.g., a flow in Apache NiFi. Therefore, operators can easily build data pipelines in the Data Fabric by defining instances of the Task entity as new context information in the Scorpio Broker.

Lastly, as part of the Task Management feature, the Task entity contains NGSI-LD properties of particular interest that receive the name of *action* and *state*. The *action* property allows for operators to declaratively request the desired state of Task instance – i.e., the *state* property. The content of the *action* property is processed by the Weaver component upon every NGSI-LD notification. Currently, the *action* property supports the *START* and *END* values, where the former value triggers

the execution of an application, while the latter actuates the termination of the execution and the eventual deletion of the associated Task instance. When the action is triggered, the *state* property of the Task entity is updated by the Weaver component using the NGSI-LD API. Currently, the state property supports the *RUNNING* and *CLEANED* values to indicate the state of the Task instance in every moment. In summary, by updating the contents of *action* property, the Task Management feature offers a mechanism to manage the life cycle of Task instances, and therefore, the stages in a data pipeline.

2.1.3. Data Sources

As introduced in the previous deliverable D4.3 [6], the ETSI CIM (Context Information Management) standards specify a framework for exchanging context information. To gather data lineage information of the data flows between sources and consumers, the SDA leverages the ETSI CIM standard for the metadata management. The ETSI CIM standards specify the NGSI-LD protocol, which sources and consumers can leverage to exchange metadata information through the SDA's NGSI-LD context broker component.

As part of the specification, ETSI CIM introduces the NGSI-LD information models to represent the context information. This subsection shows NGSI-LD information models that describe the metadata associated with the different data sources of the SDA which were previously identified in D4.3 [6] document.

2.1.3.1. Prometheus

Prometheus [10] has become one of the most relevant monitoring systems for metric collection. In this regard, Prometheus server is one of the main components of the 5Growth Vertical-oriented Monitoring System (5Gr-VoMS) platform to collect metrics from monitored targets. Moreover, Prometheus is the main component in the monitoring system available in ICT-17 platforms such as the SONATA platform of 5G-VINNI project. In fact, the SDA framework is an adaptation building component to enable the integration between 5G-VINNI and 5Growth monitoring platforms, allowing the 5G-VINNI SONATA platform to expose Prometheus metrics to the 5Gr-VoMS platform. Specifically, this integration between 5G-VINNI and 5Growth monitoring platforms is needed for the EFACEC pilot use cases. Therefore, Prometheus is identified as one of the most important data sources considered in the project.

Figure 3 depicts a diagram of the NGSI-LD information model that represents and describes the metadata associated to a data source based on a Prometheus server instance.

In this model, we can identify three main groups of entities that capture the data lineage during the collection stage for a data source of the Prometheus type:

- **Input dataset group:** Prometheus data source allows collecting a set of metrics. Each Metric entity is identified by name and may produce multiple time series for each combination of dimensions – i.e., Label entity. The metrics that share a common family name and type – e.g.,

counter or gauge – belong to the same Metric Family (concept defined by the OpenMetrics [11] initiative).

- **Collection task group:** a Task entity represents the execution of a NiFi flow that subscribes to specific Prometheus metrics with a particular polling interval from the Prometheus server instance – i.e., Prometheus entity – whose endpoint service is accessible via URL. The Task that records this flow runs a NiFi application named *MetricSource*. In addition to the specific Prometheus metric and the polling interval, the Task *expression* argument can be used to filter the time series collected for a metric based on the specified label values.
- **Output dataset group:** the data produced by the *MetricSource* Task are a set of metrics that are stored in a particular Kafka topic in the Kafka broker that runs within the SDA. The metrics are encoded in JSON-IETF format following a particular YANG data model.

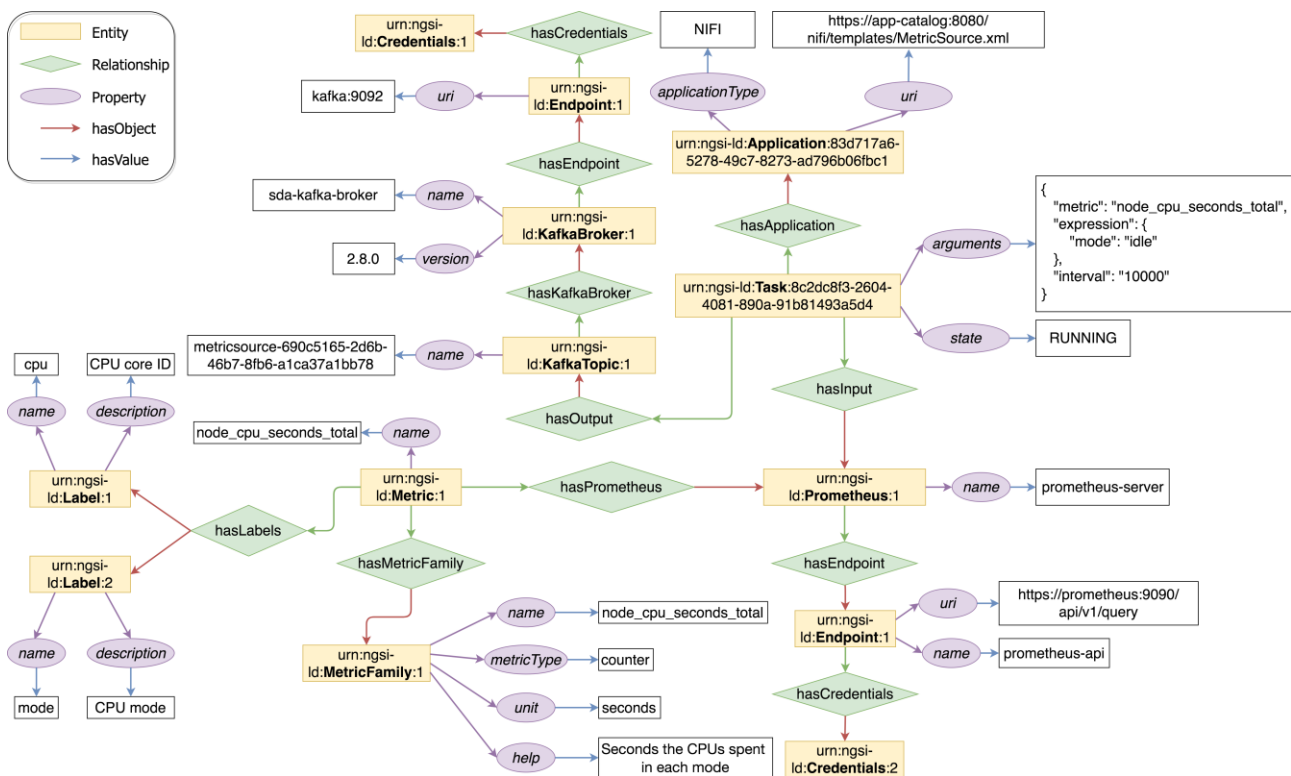


FIGURE 3: INFORMATION MODEL OF PROMETHEUS DATA SOURCE

2.1.3.2. Network Telemetry

An important type of data source identified in the network infrastructure domain is the telemetry-based network devices. Model-based streaming telemetry is gaining attention as a monitoring mechanism for network devices, mainly relying on YANG data models and management protocols like gNMI.

Figure 4 depicts the NGSI-LD information model that records the metadata associated with a data source based on a YANG-modeled network device. This data source is represented by a telemetry

service supported by a network device. The current prototype focuses on gNMI, but other protocols such as NETCONF and RESTCONF can leverage this information model as well.

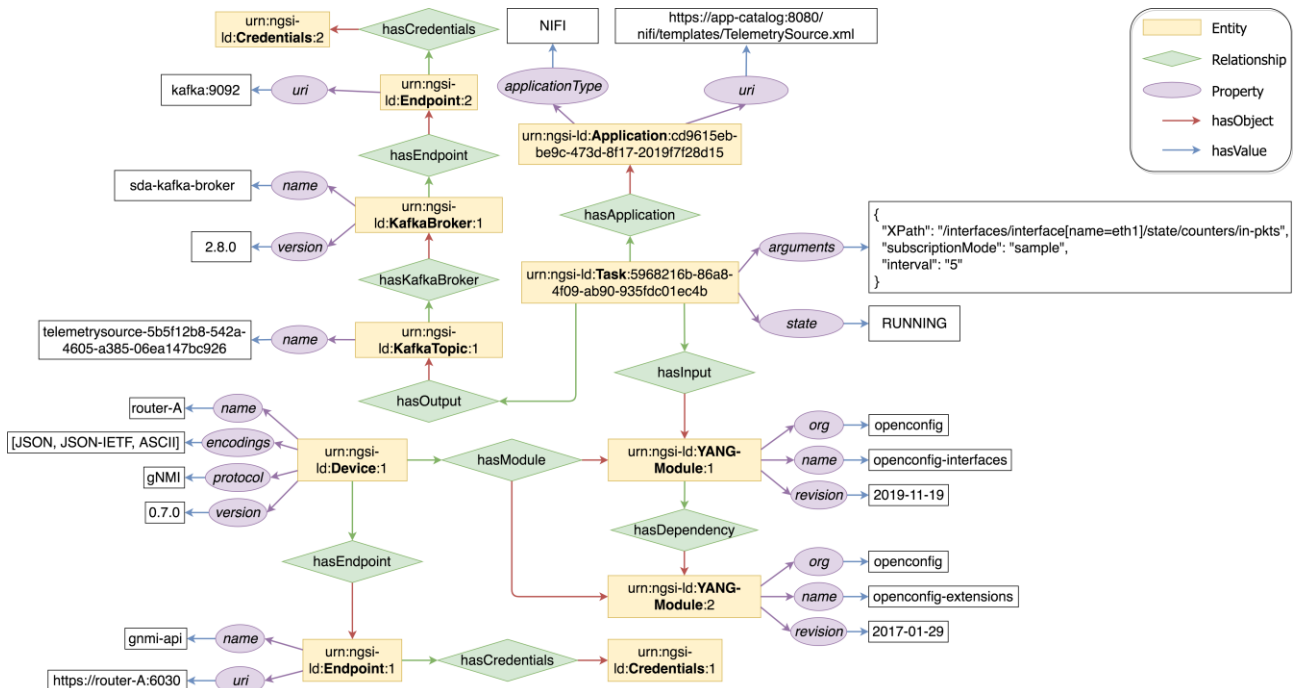


FIGURE 4: INFORMATION MODEL OF NETWORK TELEMETRY DATA SOURCE

In this model, we can identify three main groups of entities that capture the data lineage during the collection stage for a data source of YANG-based network device type:

- **Input dataset group:** a YANG-based network device allows collecting telemetry data according to its supported YANG model(s). Each YANG model is composed of a set of YANG modules identified by module name, revision number, and organization name.
- **Collection task group:** the Task entity represents the execution of a NiFi flow that subscribes to specific YANG-based telemetry data (determined by the *XPath* selector) according to the YANG modules available from a network device – i.e., Device entity – whose endpoint service is managed by gNMI protocol and accessible via URL. The Task that records this flow runs a NiFi application named *TelemetrySource*. The task application requires the specific *XPath* and a subscription mode as arguments. The subscription mode argument can be “on-change” or based on sampling interval. When a subscription is defined to be “on-change”, the data updates are only collected when the value of the telemetry data changes. When the subscription is defined to be a sample, an interval argument in seconds is needed.
- **Output dataset group:** the data produced by the *TelemetrySource* Task are telemetry data records that are stored in a particular Kafka topic in the Data Substrate component of the SDA. The telemetry data are encoded in JSON-IETF format following the YANG data model.

2.1.3.3. Apache Kafka

Apache Kafka has been identified as another important type of data source. In the context of 5Growth project, two main integration use cases based on a Kafka data source were identified:

- **Interaction between the 5G EVE and 5Growth monitoring systems:** the 5G EVE monitoring system – i.e., the Data Collection Manager (DCM) component – is based on Apache Kafka in which the monitoring data is published in topics according to a particular data model based on JSON encoding format. For the integration of both the 5G EVE and 5Growth monitoring platforms, the DCM component was identified as a data source for the 5Growth monitoring platform. To achieve the interconnection process, the usage of the SDA was proposed. This integration case served as a proof of concept to characterize and integrate Kafka-based data source in the SDA.
- **Log Parser tool integration:** the Log Parser is a monitoring tool to analyze the lifecycle management related metrics of both the network services and the network slices instantiated in the 5Growth infrastructure. Such collected information is presented in the form of logs in real-time mode. The SDA framework was proposed to aggregate and normalize these logs to deliver this information to interested data consumers. The processed logs are written in the Log Parser's Kafka cluster component from which SDA can collect the data streams to achieve real-time monitoring.

Figure 5 depicts the NGSI-LD information model that records the metadata associated with a data source based on Kafka broker. In this model, we can identify three main groups of entities that capture the data lineage during the collection stage for a data source of the Kafka type:

- **Input dataset group:** the characterization of Kafka data source is simple as it only describes the Kafka broker, the related topics, and the associated endpoint. The sample information model shows the Kafka cluster of the Log Parser tool that is composed of two different Kafka brokers, each one accessible by its endpoint.
- **Collection task group:** a Task entity represents the execution of a NiFi flow that subscribes to a specific Kafka topic from the Log Parser Kafka cluster, whose endpoint services are accessible via URL. The Task that records this flow runs a NiFi application named *LogParserSource*. In the Task *arguments* property, the *group_id* and the *offset* parameters are specified. The *group_id* allows specifying the name of the consumer group a Kafka consumer belongs to. This is a parameter required in the NiFi application to collect the information from a topic of a Kafka-based data source. The *offset* parameter allows specifying either the topic information is read from the beginning – i.e., *earliest* value – or from the last available record – i.e., *latest* value.
- **Output dataset group:** the data produced by the *LogParserSource* Task application are a set of logs stored in a specific Kafka topic in the Kafka broker that runs within the SDA. The logs are encoded in JSON-IETF format following a particular YANG data model.

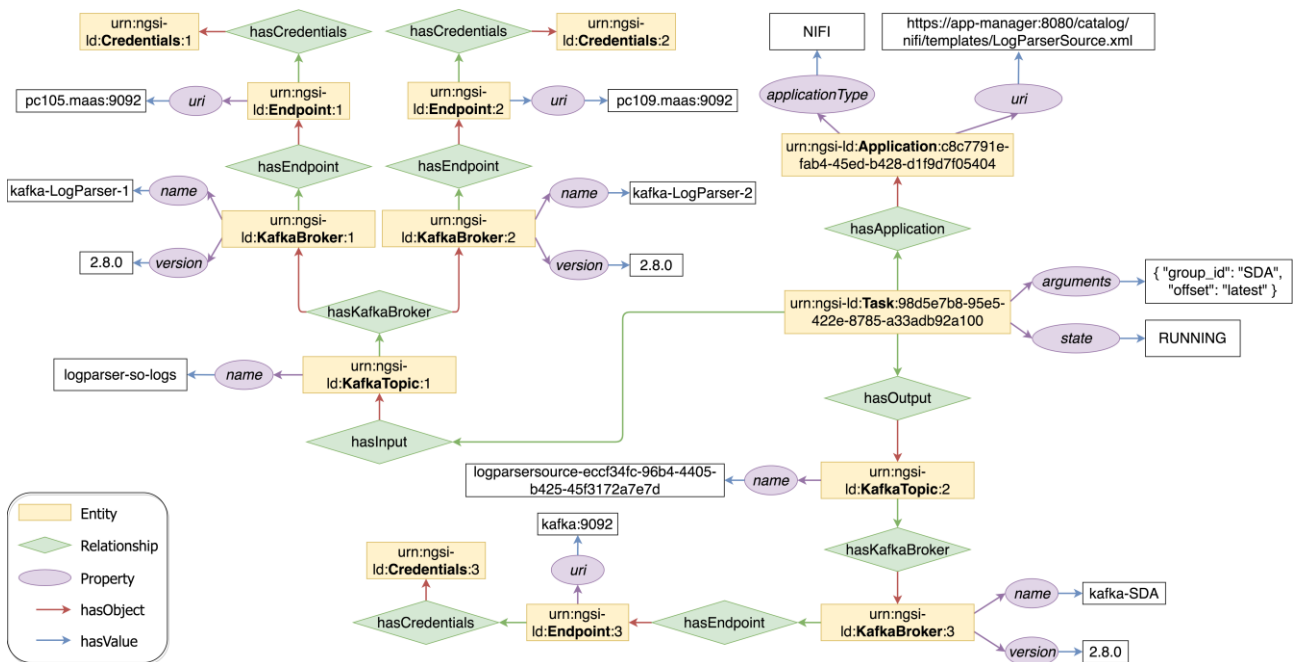


FIGURE 5: INFORMATION MODEL OF KAFKA DATA SOURCE

2.1.4. Data Consumers

This subsection provides an update on the data consumers of the SDA which were previously identified in D4.3 [6] document.

2.1.4.1. Anomaly Detection Module (AD)

The Anomaly Detection (AD) module was an innovation developed within the WP2 that was pinpointed as a potential data consumer of the SDA framework. In this use case, the SDA would aggregate RAN measurements from multiple sources and deliver them to the AD in a suitable format for consumption. As a result, the AD would then process the delivered data and infer anomalies in the monitored RAN network by applying machine learning techniques.

To provide real-time analysis of the network, the AD consumes streams of events containing RAN measurements for each monitored cell. The AD module offers a REST-based service for consuming the stream of events. This already developed service supports the ingestion of events encoded with JSON structured as per a specific data model. The finalized JSON schema is presented in Table 1. Thereafter, the AD analyzes the aggregated measurements and responds with an identifier bundled in JSON format which indicates the behavior based on different severity levels. The JSON schema is illustrated in Table 2.

TABLE 1: DATA MODEL FIELDS OF REQUEST TO ANOMALY DETECTION MODULE

Field Name	Data Type	Description
ul_delay	Float	Uplink delay (ms)
dl_delay	Float	Downlink delay (ms)
lost_packets	Integer	# of lost packets, per service per user

rsrp	Float	RSRP (dB)
transfer_protocol	Boolean	TCP or UDP encoded [0,1]
ulrx_cell	Integer	UE Bytes received from the cell
timestamp	Datetime	Measurement timestamp (datetime format - "yyyy'-MM'-'dd' 'HH':mm:ss")
cell_id	Integer	RAN cell ID

TABLE 2: DATA MODEL FIELDS OF RESPONSE FROM ANOMALY DETECTION MODULE

Field Name	Data Type	Description
behavior	String	The severity of the behavior identified by the AD module. There are 3 identified severity levels: [Normal, Suspicious, Anomalous]
timestamp	Datetime	Measurement timestamp (datetime format - "yyyy'-MM'-'dd' 'HH':mm:ss")

The integration between the SDA and the AD was tested using simulated data sources due to the lack of real data sources from the pilots. The lack of data sources in the pilots that would provide RAN measurements impeded the SDA from implementing the aggregation mechanism that would generate valuable data to be consumed by the AD module.

2.1.4.2. Prometheus Exporter

A Prometheus exporter allows exposing metrics through an HTTP endpoint to be scraped by a job configured in Prometheus. As of this deliverable, the Prometheus exporter is the only type of data consumer integrated with the SDA.

Following the same approach for the supported data sources, we define a new NGSI-LD information model that records the metadata associated with a data consumer based on the Prometheus exporter.

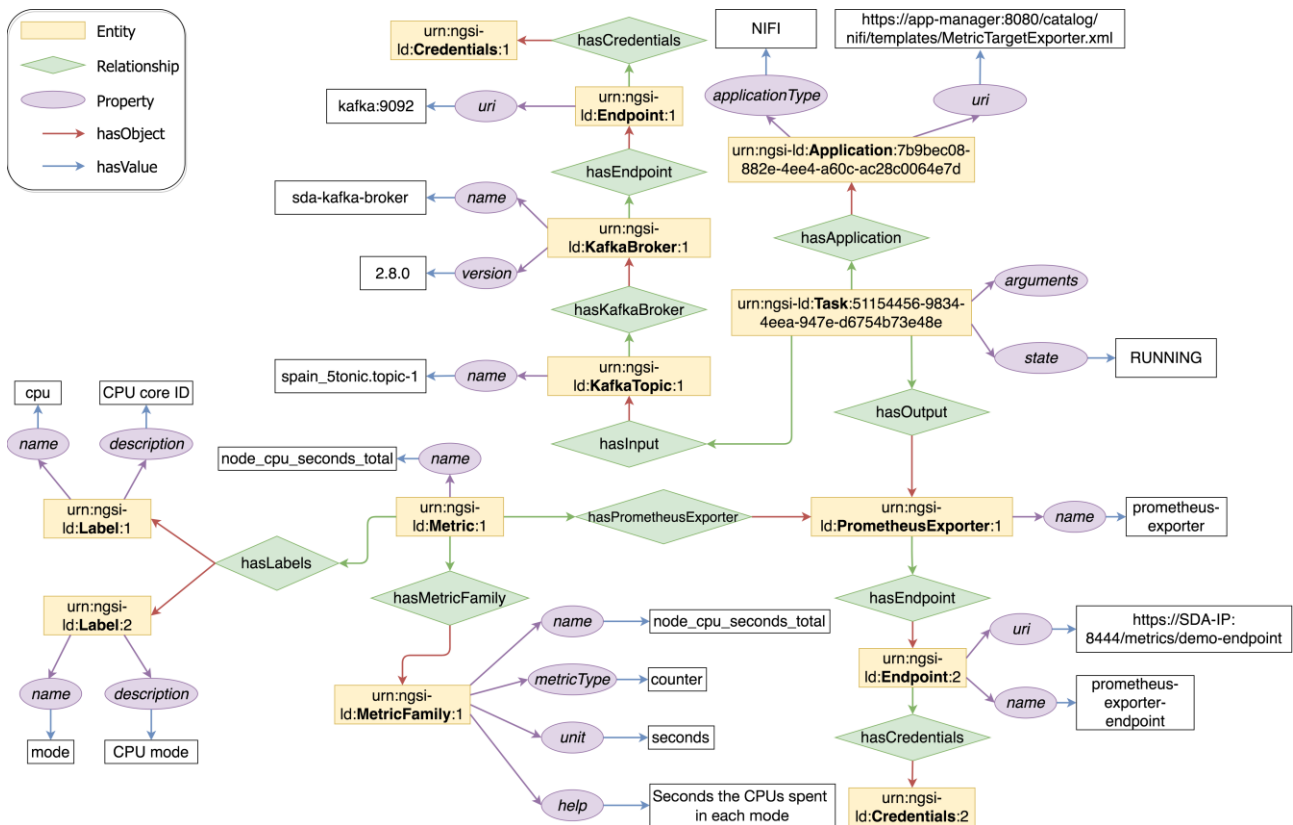


FIGURE 6: INFORMATION MODEL OF PROMETHEUS EXPORTER CONSUMER

Figure 6 illustrates an instance of the NGSI-LD information model of a Prometheus exporter. In this model, we can identify three main groups of entities that capture the data lineage during the dispatch stage for a data consumer of the Prometheus exporter type:

- **Input dataset group:** metrics generated by previous transformations are stored in a specific Kafka topic in the Kafka broker that runs within the SDA.
- **Dispatch task group:** a Task entity represents the execution of a NiFi flow that collects metrics from the specified Kafka topic, adapts the data to the Prometheus text-based format, and exposes the data through a specified HTTP endpoint, i.e., Prometheus Exporter entity. The Task that records this flow runs a NiFi application named *MetricTargetExporter*.
- **Output dataset group:** the data produced by the *MetricTargetExporter* Task are a set of Metric entities which are exposed through the specified Prometheus Exporter. In the same manner, as the model of a Prometheus data source, these Metric entities include additional metadata information such as the Metric Family and a list of Labels.

2.1.4.3. Prometheus Pushgateway

The purpose of the Pushgateway [12] component is caching metrics that have been generated by ephemeral and batch jobs, hence allowing for Prometheus to later scrape them. However, the Prometheus community does not recommend using this mechanism for scraping metrics except for those scenarios where metrics are sent to Prometheus in batch fashion.

From the standpoint of the SDA, there is no distinction between ephemeral (batch) jobs and streaming jobs as the SDA follows the Kappa architecture [13]. The SDA implements the Kappa architecture by writing those metrics generated from the aggregated data into an Apache Kafka topic. Kafka precisely implements a pub-sub mechanism that enables a caching system similar to that intended by the Pushgateway. As a result, by leveraging Kafka, the SDA can always implement a Prometheus Exporter – as shown in previous subsection. Therefore, for these reasons the exposition of new metrics to Prometheus via the Pushgateway mechanism has been deprecated.

2.1.5. YANG-based drivers

In the previous D4.3 [6] document, it was introduced that the integration of data sources within the SDA requires the configuration of collection agents that leverage a special component called drivers which are responsible for transforming the ingested events collected from the sources into a particular YANG data model and then structuring the data using a particular encoding format.

These software components called drivers have been extended to cover the integration of data consumers within its associated dispatch agents to deliver the data to consumers following its own YANG data model.

In addition, there are other type of drivers involved in the transformation process from the data source model to the data consumer model.

Therefore, the combination of these types of drivers allows transforming the resulting data collected from monitoring sources into a format that is suitable for those interested data consumers. This Section provides an update on the implementation of the different types of so-called YANG-based drivers.

2.1.5.1. Data Source Drivers

As introduced, the integration of the data sources with the SDA requires the configuration of collection agents. These agents need to address the adaptation of both the transport mechanism and the encoding format.

Regarding the encoding aspect, the collection agent leverages the data source driver which is responsible for transforming the ingested events into a YANG data model. By "translating" the event into a YANG data model tree, the SDA can structure data using the JSON IETF encoding format.

To facilitate the adoption of new data sources, a methodology for implementing the so-called data source drivers within the collection agents is proposed. Figure 7 depicts the approach to this implementation.

YANG-TOOLS JSON-IETF encoding

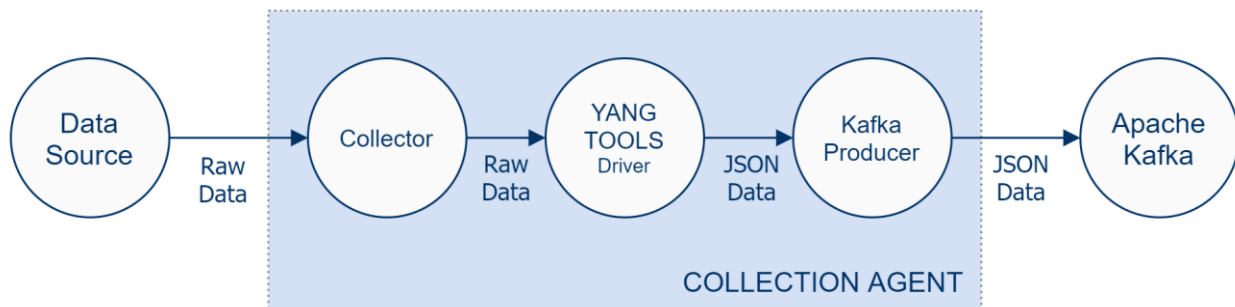


FIGURE 7: DATA SOURCE DRIVER

This approach is agnostic to the chosen tools, but in the case of the current SDA implementation, Apache NiFi, Apache Kafka, and JSON IETF encoding format are the chosen pieces of software for building data source drivers.

The *Collector* is a NiFi processor that allows collecting the data from the *Data Source* according to its transport mechanism – e.g., in the Prometheus particular case, the data source is a Prometheus instance with its own HTTP endpoint and the NiFi processor works as a HTTP client.

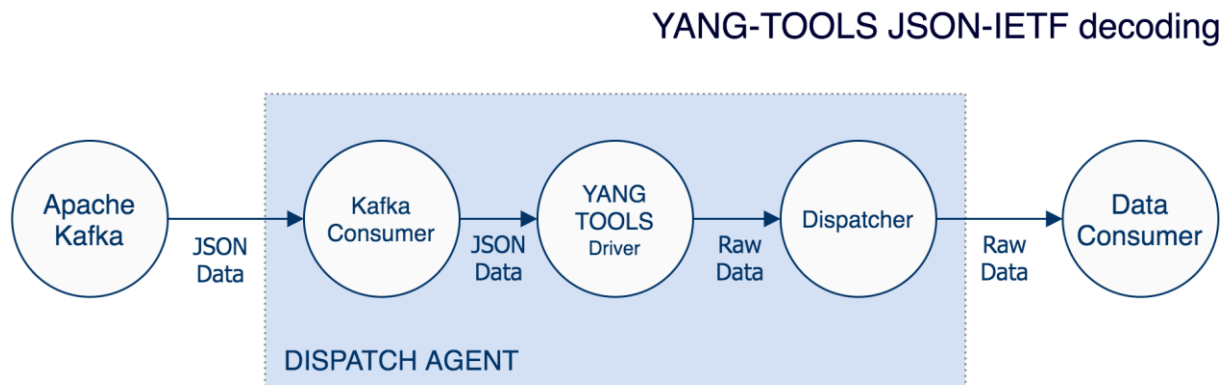
Each data source driver is implemented with the YANG-TOOLS [14] Java library as a custom NiFi processor. These types of drivers are built on Java class bindings that YANG-TOOLS generate for the YANG model(s) associated with each data source. The data source driver is responsible for normalizing the collected data by parsing the raw data from the source and producing data structured according to the associated YANG model. For the sake of interoperability, the produced data are encoded into the JSON IETF format.

Once the driver finishes the job, the collection agent applies the required mechanisms that write the encoded structured data into the *Data Substrate* – i.e., the SDA Kafka broker – making use of *Kafka Producer* as another NiFi processor.

2.1.5.2. Data Consumer Drivers

As with the data sources, there are YANG model(s) associated with each data consumer integrated into the SDA and is required a dispatch agent with its corresponding driver to deliver the information to those data consumers.

To facilitate the adoption of new data consumers, a methodology for implementing the so-called data consumer drivers within the dispatch agents is proposed. Figure 8 depicts the approach to this implementation.

**FIGURE 8: DATA CONSUMER DRIVER**

First, the dispatch agent applies the required mechanisms that read the encoded structured data according to the *Data Consumer* model from the *Data Substrate* – i.e., the SDA Kafka broker, making use of *Kafka Consumer* as another NiFi processor. The structured data according to the YANG model associated with the *Data Consumer* is encoded in JSON IETF format and it is generated from the transcoding process between the source data model and the consumer's own data model (for more information on the transcoding process see the following *Transformer Drivers* subsection).

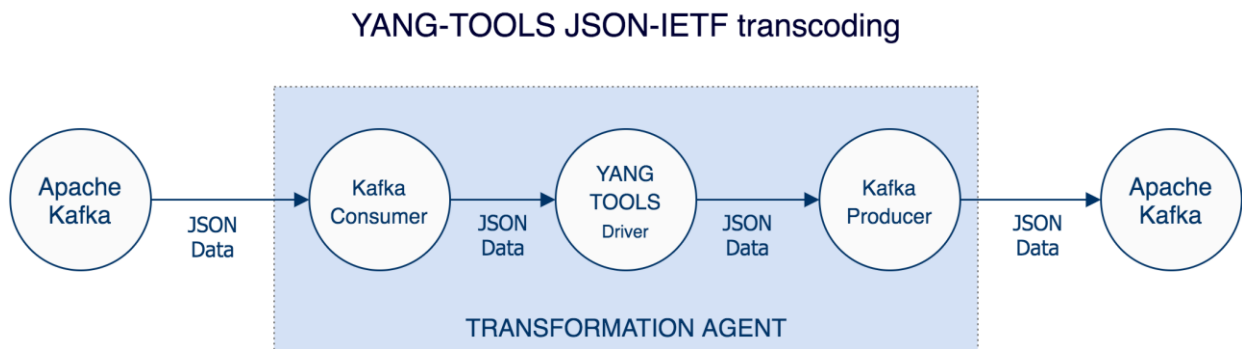
As with the aforementioned data source drivers, each data consumer driver is implemented with the YANG-TOOLS library as a custom NiFi processor. This NiFi processor is responsible for decoding the structured data according to the YANG models associated with the consumer to the real data format expected by the consumer.

Once the driver finishes the job, the dispatch agent applies the required mechanisms – e.g., transport protocol, credentials, additional requirements, etc. – in order to deliver the decoded data to the *Data Consumer* making use of the *Dispatcher* component working as another NiFi processor.

2.1.5.3. Transformer Drivers

As introduced, there is a last type of driver that enables mapping between YANG models. The scope of this type of driver is the transformation from the data source model to the data consumer model, in which transformation agents are involved.

To facilitate the transformation process, a methodology for implementing the so-called data transformer drivers within transformation agents is proposed. Figure 9 depicts the approach to this implementation.

**FIGURE 9: TRANSFORMER DRIVER**

The final goal is that a transformation agent working as NiFi chained flows can receive events encoded according to the YANG model of the *Data Source* and read from a particular *Data Substrate* topic and be able to write the events into another *Data Substrate* topic with the data encoded and structured according to the YANG model of the *Data Consumer*.

This transformer driver is also implemented with the YANG-TOOLS library as a custom NiFi processor. The transformation process is achieved by mapping between the Java bindings associated with the YANG-modeled data from both the source and the consumer. Note that the inputs and outputs of the transformer driver are events encoded in JSON-IETF format, and therefore, the driver performs a JSON-IETF transcoding.

2.2. Experiment Catalogue

The 5Growth experiment catalogue is available online, next to the source repositories of the platform, at the following URL: <https://github.com/5growth/5gr-pilots>.

This repository contains all experiment descriptors, as presented in D4.3 [6] Section 2.4, grouped by pilots and then use-cases. These experiment descriptors contain all the information needed to either reproduce the results in the same pilot site, or to compare the results across sites.

2.3. Available Tools

2.3.1. 5Growth Log Parser and its Integration with SDA and 5Growth Stack

The 5Growth Log Parser can interact and is integrated with the SDA (Semantic Data Aggregator) described in D4.3 [6] Section 2.1.1, and the 5Growth stack (mainly with SO and VS) via VPN connections, as depicted in Figure 10. The 5Growth Log Parser (5Gr-LP) is an extension of the log parser described in D4.3 [6] with the additional functionality to integrate with SDA and VS log parsing, in addition to the already existing SO level log parsing. In practice, this is achieved by running multiple Spark jobs inside the 5Gr-LP. In the architecture, SDA can ingest the output data streams of the 5Gr-LP to achieve real-time monitoring. The steps numbered from #1 to #6 in Figure 10 were already detailed in D4.3 [6] Section 2.2.1 [5]. The only difference in the updated flowchart is that logs

from VS are also handled. To this end, a separate Apache Flume jobs being run in step #2 are created for connecting to VS and SO log folders. In addition, a separate Kafka topic being created in step #3 for ingesting logs from levels SO and VS levels and the processing of logs in step #4 are handled by separate Spark Jobs. For the subscription of SDA, separate Kafka topics are created for VS and SO logs for cleaned data after Spark processing. In general, there are two possible integration options between the 5Gr-LP and SDA as shown in Figure 10. These options are as follows:

1. **Option-1** (marked with green color from point A): the SDA acts as a subscriber to the Kafka topic used by the 5Gr-LP. The SDA simply subscribes to the topic and receives data for each defined metric as it is produced. However, this integration requires SDA to clean the data, which the 5Gr-LP already does for SDA.
2. **Option-2** (marked with orange color from point B): SDA pulls the data from the ElasticSearch server. This is the same approach that SDA takes for the Prometheus data source integration. SDA basically pulls the data from the ELK stack's REST API at certain intervals. This approach is less efficient due to the overhead of obtaining the same data with Kafka but has the advantage that the data comes from ElasticSearch already structured and can be visualized with tools such as Kibana.

For integration, we selected SDA as the subscriber to 5Gr-LP system because it is the most efficient mechanism in terms of integration. In Figure 10, considering 5Gr-LP architecture, SDA is connected to 5Gr-LP as Option-1 marked with Letter-A as well. In this figure, the 5Gr-LP platform can be considered as the data producer for SDA and transmits the relevant metrics collected by VS and SO of the 5Growth's stack and processed in log format.

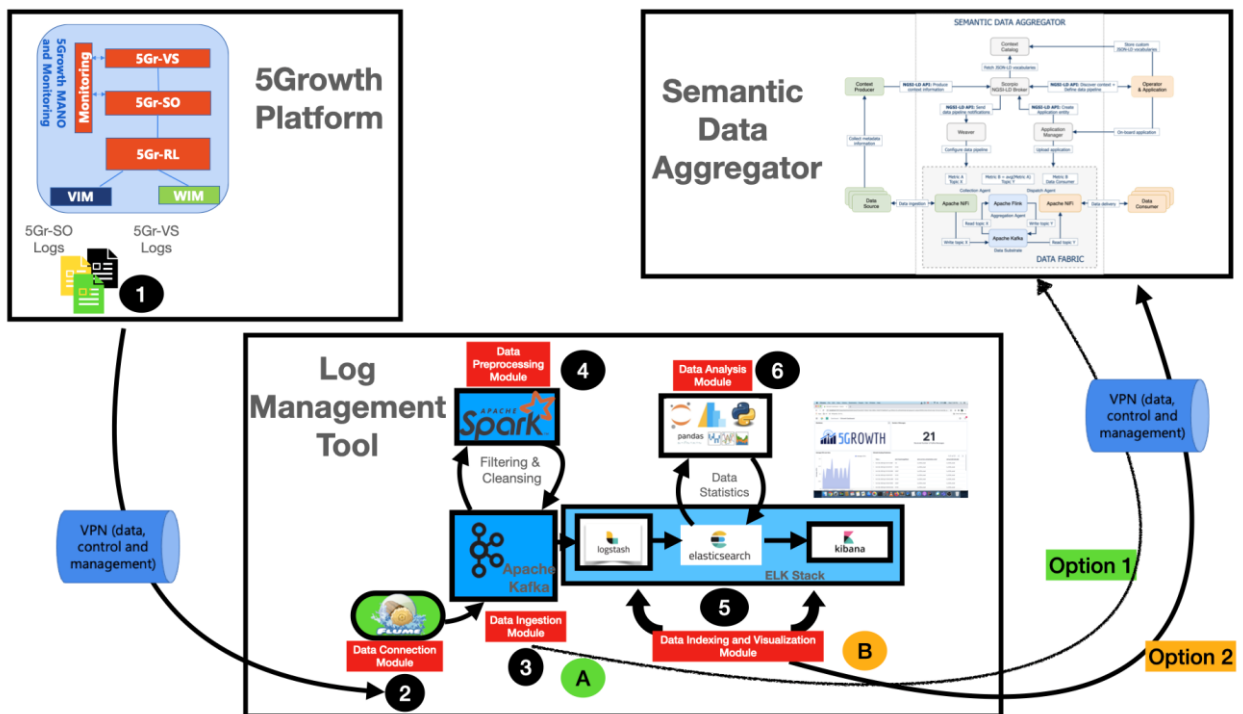


FIGURE 10: GLOBAL ARCHITECTURE FOR 5GROWTH LOG PARSER

The list of available metrics and their corresponding descriptions of 5GR-VS in the 5Gr-LP can be found in Table 4 and Table 5. Table 4 provides communication service level metrics with end-to-end network slice level metrics and Table 5 provides network slice level metrics (with end-to-end network service metrics). These metrics are linked to their slice using the parameters from Table 3. The provided metrics are complemented with information about the specific instance, supporting the statistical aggregation of the results. In addition, all different metrics also record their creation timestamp, allowing statistical analysis in the time domain (e.g., time of day, day of week, etc.).

TABLE 3: 5GR-VS METRIC PARAMETERS

Parameter Name	Data Type	Description
VSI_ID	Long	Vertical Service instance (VSI) ID
VSSI_ID	String	Vertical Sub-Service Instance ID
VS_NAME	String	Vertical Service Blueprint Name
NSI_ID	Long	Network Slice Instance ID
NFV_NSI_ID	Long	NFV Network Service Instance ID

TABLE 4: 5GR-VS INSTANCE RELATED METRICS

Metric Name	Data Type	Associated Parameters	Description
Vertical_Service_Translation_Time	Long	VS_NAME, VSI_ID	The time required to translate a vertical service request into network slice level request. Not available in the communication service level integrations (i.e., 5G EVE), since the translation workflow is not applied in these scenarios.
Vertical_Service_Arbitration_Time	Long	VS_NAME, VSI_ID	The time required to compute the arbitration procedures for a vertical service instance request. As in the previous case, not used in the CSMF level integrations (i.e., 5G EVE).
Vertical_Service_Instatiation/Termination_Time	Long	VSI_ID, VS_NAME	Time measured between the arrival of the vertical service lifecycle management request and the completion of the vertical service instantiation/termination workflow
EVE_VSS_Instatiation/Termination_Time	Long	VSI_ID, VSSI_ID	Time computed at the 5G EVE driver to complete the instantiation/termination of one vertical sub-service
EVE_VSS_Ready_Time	Long	VSI_ID, VSSI_ID	Time the 5Gr-VS waits for a human operator in 5G-EVE platform to accept the requested operation (e.g., experiment) in the 5G EVE portal

In the case of full in-house deployments and NSMF-level integration (i.e., 5G-VINNI), the 5Gr-LP for VS also provides metrics related to the network slices and network services described in Table 5.

TABLE 5: 5GR-VS NETWORK SLICE AND NETWORK SERVICE METRICS

Metric Name	Data Type	Associated Parameters	Description
Vertical_Service_NSI_Instatiation/Termination_Time	Long	NSI_ID	Time required to complete the instantiation/termination of a network slice. Measured at the vertical service lifecycle management module between the moment of issuing the network slice lifecycle management request and the arrival of the network slice status change notification.
Network_Slice_Instatiation/Termination_Time	Long	NSI_ID	Time required to complete the instantiation/termination of a network slice. Measured at the network slice lifecycle management module between the arrival of the request and the completion of the instantiation/termination workflow. Only available for the full 5Growth stack deployments.
Network_Service_Instatiation/Termination_Time	Long	NFV_NSI_ID	Time required to complete the instantiation of a network service. Measured at the network slice lifecycle management module between the moment of issuing the network service lifecycle management request and the moment when the network service instance status is updated at the 5Gr-SO. Only available for the full 5Growth stack deployments.

Note that Table 4 applies to all three of the 5Growth pilots. Table 5 does not apply to INNOVALIA and EFACEC pilots. The results using the vertical service level metrics presented in Table 4 are reported in the INNOVALIA pilot (Section 4.1).

2.3.2. Telemetry for P4 programmable switches

With the emergence of switch data plane programmability, new data plane programming protocol interfaces are being introduced to enable a user to specify packet processing requirements in logical tables (program a flexible pipeline), which are then compiled down to an optimized pipeline of physical hardware stages (depicted in Figure 11). One of these protocol interfaces is P4

(Programming Protocol independent Packet Processor) [15]. P4 provides a switch pipeline abstraction on which a programmer can program the entire set of switch features in one P4 configuration command.

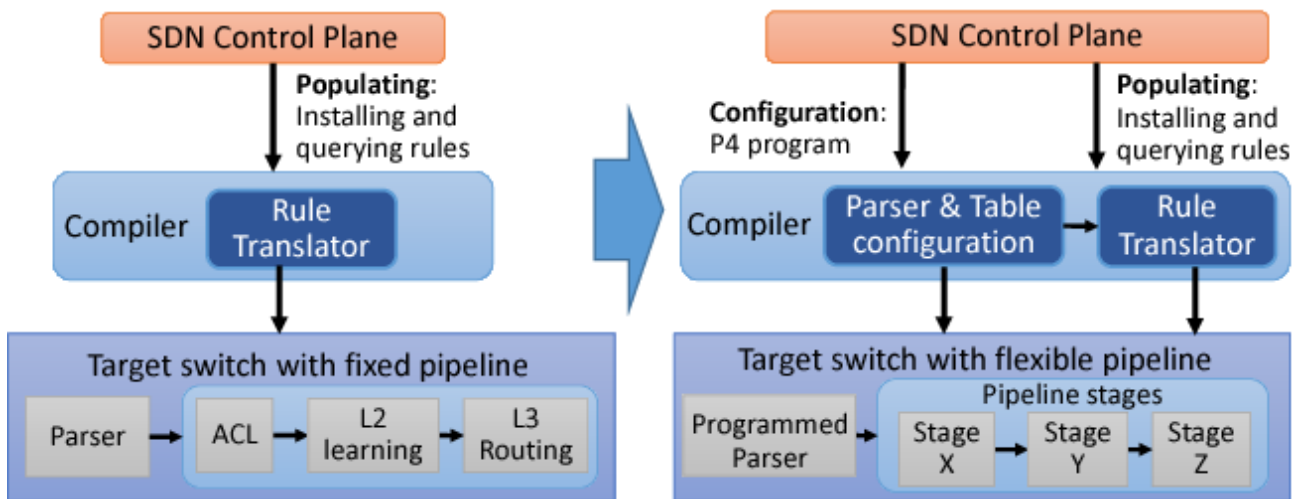


FIGURE 11: PHYSICAL HARDWARE STAGES

With the emergence of data-plane programmability, new approaches to network telemetry are available. One of these approaches, enabled by P4, is the In-band Network Telemetry (INT) [16]. This Section focuses on the telemetry capabilities of this kind of network devices and how they can be integrated in the 5Growth monitoring system. We present here a short comparison between classic telemetry and in-band telemetry. One approach does not exclude the other, they can be combined to deliver more complete and fine-grained information about the device state and its performance.

2.3.2.1. Classic Telemetry

For several years, network monitoring is a crucial activity, mostly targeting the overall health of network infrastructure and the detection of problems. In the past, Simple Network Management Protocol (SNMP) became one of the de-facto network monitoring protocols, providing a common mechanism for network devices (such as routers, switches, servers, firewalls, wireless access points, etc.) to relay management information to a centralized entity (i.e., SNMP manager).

However, the network infrastructures have increased in size and complexity, far beyond the infrastructure capabilities provided by SNMP. Modern network devices and network operating systems moved away from the SNMP protocol and the Management Information Base (MIB) for the sake of telemetry and device configuration. Industry-wide standardization efforts have defined a device-agnostic configuration model, OpenConfig [17], defined with the YANG modelling language [18]. The model is well structured and strongly typed which facilitates the programmability. OpenConfig is officially agnostic from the communication protocol, anyway gNMI (gRPC Network Management Interface) [19] is increasingly popular among manufacturers. The main advantage of gNMI is the capability of streaming data to and from network devices with an efficient binary

encoding format. Furthermore, the protocol offers primitives to the clients to subscribe for stream state changes so as to receive data only when a difference with the previous state is detected.

Recently, flow monitoring appeared as a new category of network monitoring that provides a finer-grained understanding of the traffic structure traversing the network represented by NetFlow, sFlow, and IPFIX.

2.3.2.2. In-Band Network Telemetry (INT)

To advance in the telemetry issues, in-band network telemetry is an emerging representative of the network telemetry, which has received extensive attention in both academia and industry in recent years. Different from the traditional network measurement and software-defined measurement, in-band network telemetry combines both data packet forwarding with network measurement. In-band network telemetry collects the network status by inserting meta-data into the data plane packets. With in-band network telemetry, problems like performance bottlenecks, network failures or misconfigurations are easier to detect since the network administrators have direct visibility of what is happening on the data plane.

2.3.2.3. Matching with 5GROWTH KPIs

The 5GROWTH project has provided a detailed specification of the Service KPIs and their mapping to Core KPIs in D4.1 [4]. Thanks to the telemetry approaches and features presented above, some of the Core KPIs can be measured directly inside the network devices on which the networking infrastructure is built. Metrics and logs that are already collected at the end devices and Virtual Network Functions (VNFs), can be enriched with fine-grained information about the network enabling better fault detection and fault localization, as well as easier resolution of performance problems.

We selected the following Core KPIs that can be measured with network device telemetry:

- CKPI-2 Packet Loss (%): The number of packets that fail to reach their destination, measured in specific interfaces of the use case logical architecture.
- CKPI-3 Guaranteed Data Rate (Mb/s): The data rate is the number of bits per unit of time sent over a specific interface of the use case logical architecture. The guaranteed data rate is the minimum expected data rate for the overall use case to function/operate correctly.
- CKPI-8 Data Volume (Gb): The total quantity of information transferred over a given interface during specific use case operations, measured in bits.

2.3.2.4. Proof-of-Concept Scenario

The proof-of-concept scenario is based on a Mininet network with two virtual hosts and a BMv2 software switch [20]. The BMv2 is a software reference model for P4-enabled programmable switches. The switch is configured with Stratum [21] as its switch operating system. Stratum provides several features to achieve better control on the P4 switch. Stratum provides three interfaces:

- P4Runtime interface: used to control the data plane elements of the switch and to dynamically populate forwarding tables.
- gNMI interface: for configuring the switch and for retrieving information about the switch state.
- gNOI interface: for accessing other operational variables of the switch

In this PoC scenario, the gNMI interface will be used to retrieve telemetry data. The scenario is depicted in Figure 12:

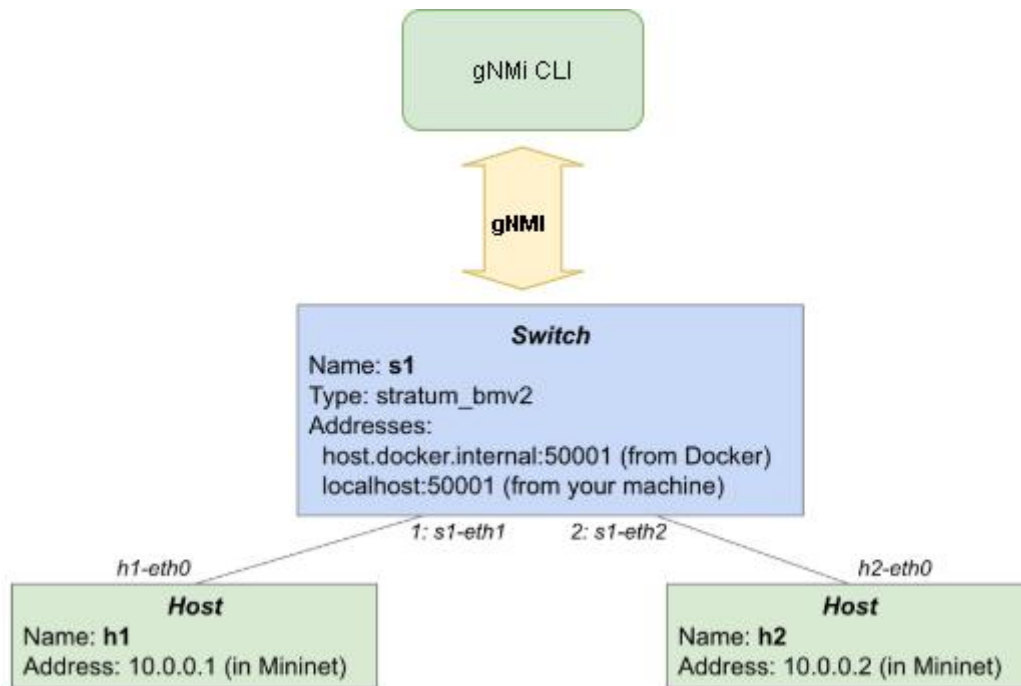


FIGURE 12: PROOF OF CONCEPT SCENARIO FOR P4 SWITCH TELEMETRY

Two virtual hosts are connected to the software switch virtual network interfaces. The hosts are configured to generate traffic to be exchanged between them.

As mentioned before, the gNMI interface can be used for both getting information about the switch state and setting configuration. The available objects to interact with are catalogued in an OpenConfig model, organized hierarchically. An example is shown in Figure 13.

```

Module: openconfig-if-ethernet
  augment /ocif:interfaces/ocif:interface:
    +--rw ethernet
    +--rw config
      |   +--rw mac-address?
      |   +--rw auto-negotiate?
      |   +--rw duplex-mode?
      |   +--rw port-speed?
      |   +--rw enable-flow-control?
    +--ro state
      +--ro mac-address?
      +--ro auto-negotiate?
      +--ro duplex-mode?
      +--ro port-speed?
      +--ro enable-flow-control?
      +--ro hw-mac-address?
      +--ro counters
      ...

```

FIGURE 13: OPENCONFIG MODEL EXAMPLE

For the sake of a proof of concept, we access the gNMI interface through a Python CLI. The “get” primitive can be used to retrieve information from the device. It accepts as a parameter a path on the OpenConfig hierarchy. As an example, we can query the device for the number of unicast packets observed on the interface “s1-eth1”:

```
get --interval 1000 sub-sample /interfaces/interface[name=s1-eth1]/state/counters/in-unicast-pkts
```

The parameters “--interval 1000 sub-sample” are introduced to get a periodicity of one second. Hence, the *counter* will be updated each second. The result is shown in Figure 14.

```
*****
RESPONSE
update {
  timestamp: 1564838158894794047
  update {
    path {
      elem {
        name: "interfaces"
      }
      elem {
        name: "interface"
        key {
          key: "name"
          value: "s1-eth1"
        }
      }
    }
    elem {
      name: "state"
    }
    elem {
      name: "counters"
    }
    elem {
      name: "in-unicast-pkts"
    }
  }
  val {
    uint_val: 585
  }
}
*****
```

FIGURE 14: GNMI RESPONSE FOR A GET COMMAND

The PoC scenario validated the possibility to correctly retrieve data with the gNMI interface and OpenConfig protocol on P4-enabled network devices.

The telemetry data generated by the P4-enabled switches can be integrated with the 5GROWTH monitoring platform through the SDA presented in Section 2.1.1 of D4.3 [6]. The SDA can collect data from multiple sources and subsequently process it to harmonize it in a common data format, in this case compatible with the 5GROWTH monitoring platform.

One of the available sources in the SDA is the telemetry-based data sources, capable of integrating YANG-based network devices. The BMv2 software switch in the proof of concept presented in 2.3.2.4 is integrated as a telemetry-based source. The SDA already provides a client (gNMIc [22]) to retrieve data from network devices exposing a gNMI interface. The data, in OpenConfig format, is then processed by the SDA to convert it into the metrics format of the 5GROWTH Monitoring Platform. Figure 15 shows the data flow from the P4 switch to the 5GROWTH monitoring platform, passing through the SDA.

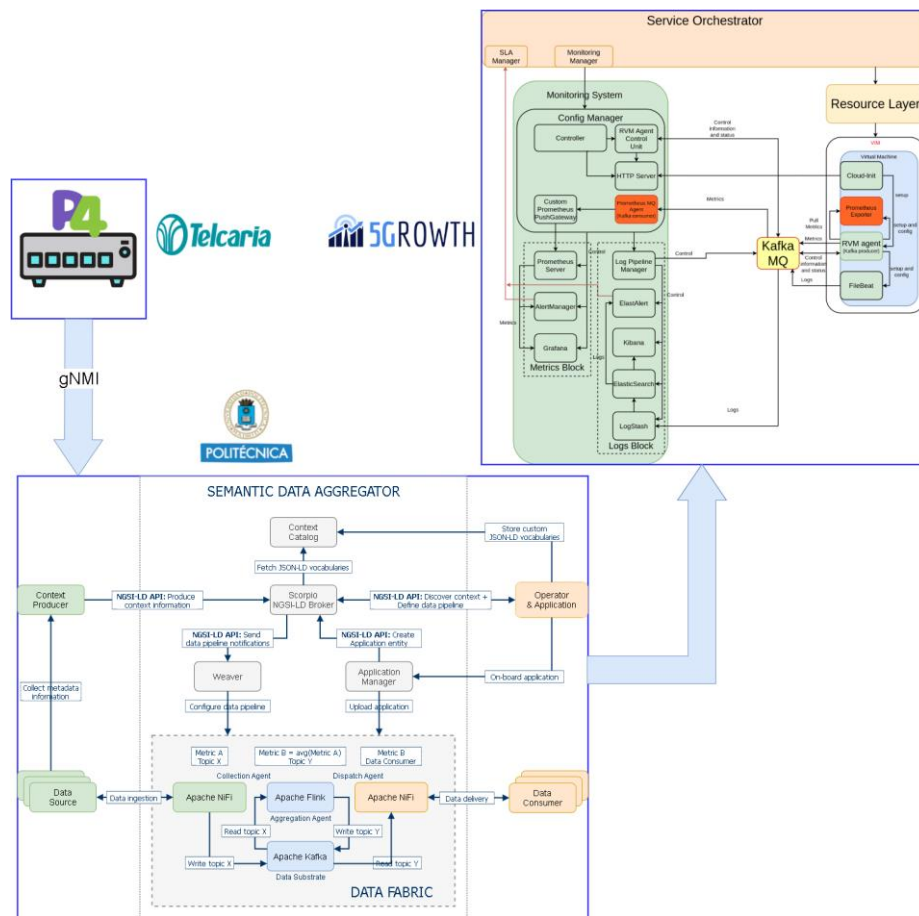


FIGURE 15: P4 SWITCHES-SDA INTEGRATION DATA FLOW

The core KPIs presented in Section 2.3.2.3 must be mapped to the gNMI queries. To know the supported metrics on the gNMI interface, we can send a query to obtain the capabilities of the P4-Stratum switch. Figure 16 shows the response for the capabilities query.

```
stratum@stratum:~/tutorial/basic$ gnmic -a localhost:50001 -u admin -p admin --insecure capabilities
gNMI version: 0.7.0
supported models:
- openconfig-interfaces-stratum, Open Networking Foundation, 0.1.0
- openconfig-lacp-stratum, Open Networking Foundation, 0.1.0
- openconfig-platform-stratum, Open Networking Foundation, 0.1.0
- openconfig-platform-transceiver-stratum, Open Networking Foundation, 0.1.0
- openconfig-qos-interfaces-stratum, Open Networking Foundation, 0.1.0
- openconfig-interfaces, OpenConfig working group, 2.4.1
- openconfig-if-ip, OpenConfig working group, 3.0.0
- openconfig-lacp, OpenConfig working group, 1.1.1
- openconfig-platform, OpenConfig working group, 0.12.2
- openconfig-platform-linecard, OpenConfig working group, 0.1.1
- openconfig-platform-port, OpenConfig working group, 0.3.2
- openconfig-platform-transceiver, OpenConfig working group, 0.7.0
- openconfig-vlan, OpenConfig working group, 3.2.0
- openconfig-system, OpenConfig working group, 0.7.0
- openconfig-hercules-platform-linecard, OpenConfig working group, 0.2.0
- openconfig-hercules-qos, OpenConfig working group, 0.1.0
- openconfig-hercules-platform, OpenConfig working group, 0.2.0
- openconfig-hercules-platform-chassis, OpenConfig working group, 0.2.0
- openconfig-hercules-platform-port, OpenConfig working group, 0.2.0
- openconfig-hercules, OpenConfig working group, 0.2.0
- openconfig-hercules-interfaces, OpenConfig working group, 0.2.0
- openconfig-hercules-platform-node, OpenConfig working group, 0.2.0
```

FIGURE 16: CAPABILITIES FROM P4-STRATUM SWITCH VIA GNMIC

The response reports the gNMI version supported (i.e., 0.7.0) and the OpenConfig objects available. The capabilities are vendor/Network Operating System dependant. In this case, the showed capabilities belong to the Stratum-based switch, but in other commercial devices such as Cisco, Juniper, etc. different capabilities, and hence more metrics, can be obtained due to their proprietary software-hardware pairs. On the other hand, each device that supports Stratum offers a set of common metrics.

According to Figure 12, the interface s1-eth1 is selected a reference to get the data and test the metrics. Furthermore, the "out direction" metrics of the interface will be requested. "In direction" can be acquired as well, but the "out direction" is enough to demonstrate the experiment. The traffic has been generated with the *iperf* tool which sets up a TCP connection between h1 and h2. A typical command is shown in Figure 17.

```
mininet> h1 iperf -s -p 5566 -i 1 &
-----
Server listening on TCP port 5566
TCP window size: 85.3 KByte (default)
-----
mininet> h2 iperf -c 10.0.0.1 -p 5566 -t 15
-----
Client connecting to 10.0.0.1, TCP port 5566
TCP window size: 85.0 KByte (default)
-----
[  3] local 10.0.0.2 port 53374 connected with 10.0.0.1 port 5566
[ ID] Interval      Transfer    Bandwidth
[  3]  0.0-16.1 sec  1.88 MBytes  976 Kbits/sec
```

FIGURE 17: TCP IPERF BETWEEN HOST1 AND HOST2

In the following, it is described the relationship between metrics that can be obtained by gNMIc (with the OpenConfig YANG models supported, shown in Figure 16) and the 5Growth Core KPIs. The model used for all metrics is *openconfig-interfaces*.

The first KPI is CKPI-3 Guaranteed Data Rate. The query required to get this metric is:

```
gnmic -a localhost:50001 -u admin -p admin -e proto --insecure sub --path
interfaces/interface[name=s1-eth1]/state/counters/out-octets --suppress-redundant --
updates-only
```

```
{
  "source": "localhost:50001",
  "subscription-name": "default-1636711294",
  "timestamp": 1636711296040150149,
  "time": "2021-11-12T11:01:36.040150149+01:00",
  "updates": [
    {
      "Path": "interfaces/interface[name=s1-eth1]/state/counters/out-octets",
      "values": {
        "interfaces/interface/state/counters/out-octets": 3979868
      }
    }
  ]
}
```

FIGURE 18: FIRST RESPONSE FOR OUT-OCTETS QUERY

```
{
  "source": "localhost:50001",
  "subscription-name": "default-1636711294",
  "timestamp": 1636711312040162525,
  "time": "2021-11-12T11:01:52.040162525+01:00",
  "updates": [
    {
      "Path": "interfaces/interface[name=s1-eth1]/state/counters/out-octets",
      "values": {
        "interfaces/interface/state/counters/out-octets": 5914522
      }
    }
  ]
}
```

FIGURE 19: LAST RESPONSE FOR OUT-OCTETS QUERY

With this “subscription” query, the information about “out-octets” will be shown only in case there is some update. In this way, when *iperf* starts, the data about out-octets in the s1-eth1 interface will be shown to compare the first response (Figure 18) against the last one (Figure 19). The time and the octets differences will be calculated to generate the data rate.

The results show that there are 1934654 bytes in 16 seconds. Hence, the resulting bandwidth is 967'327Kb/s which is like *iperf* results, see Figure 17.

The second KPI is CKPI-8 Data Volume. The query required to get this metric is:

```
gnmic -a localhost:50001 -u admin -p admin -e proto --insecure get --path
interfaces/interface[name=s1-eth1]/state/counters/out-octets
```

```
[
  {
    "timestamp": 1636713837709355791,
    "time": "2021-11-12T11:43:57.709355791+01:00",
    "updates": [
      {
        "Path": "interfaces/interface[name=s1-eth1]/state/counters/out-octets",
        "values": {
          "interfaces/interface/state/counters/out-octets": 5914522
        }
      }
    ]
  }
]

[
  {
    "timestamp": 1636714036172488949,
    "time": "2021-11-12T11:47:16.172488949+01:00",
    "updates": [
      {
        "Path": "interfaces/interface[name=s1-eth1]/state/counters/out-octets",
        "values": {
          "interfaces/interface/state/counters/out-octets": 7970580
        }
      }
    ]
  }
]
```

FIGURE 20: OUTPUT FOR OUT-OCTETS QUERY BEFORE AND AFTER USE CASE OPERATIONS

With this “get” query, the information about “out-octets” will be shown whenever a specific use case stakeholder needs it. When *iperf* finished, the data about out-octets in the s1-eth1 interface is compared with the data before the specific use case operation started (Figure 20). With this, the octets difference is calculated to generate the data volume in this operation.

The results show that there are 2056058 bytes of difference between them. The data volume is 2'05Mbytes which is like the iperf done in Figure 17.

The third and last KPI is CKPI-2 Packet Loss. The query required to get this metrics is:

```
gnmic -a localhost:50001 -u admin -p admin -e proto --insecure get --path  
interfaces/interface[name=s1-eth1]/state/counters/out-discards
```

```
[  
  {  
    "timestamp": 1636715127306230878,  
    "time": "2021-11-12T12:05:27.306230878+01:00",  
    "updates": [  
      {  
        "Path": "interfaces/interface[name=s1-eth1]/state/counters/out-discards",  
        "values": {  
          "interfaces/interface/state/counters/out-discards": 0  
        }  
      }  
    ]  
  }  
]
```

FIGURE 21: OUTPUT FOR OUT-DISCARDS METRIC QUERY

With this “get” query, the information about “out-discards” will be shown whenever a determinate use case stakeholder wants (Figure 21). With this, the packet loss can be achieved at any moment considering the previous Sections’ queries to elaborate the ratio.

The result shows how there are 0 “out-discards”. Since the packet loss is 0% which is equal to the iperf done in Figure 17.

2.3.2.5. Scenario on Telefonica premises

The proof-of-concept scenario is intended to be replicated at Telefonica premises. The goal is to replace the Bmv2 software switch with a bare-metal P4-enabled switch and replace the virtual hosts with a traffic simulator. A Stordis BF2256X 1T A1F with Tofino chipset integrated has been selected as the programmable switch for telemetry experiments. The new scenario is presented in Figure 22. In this scenario, we have a server that hosts the ONOS controller and the Telemetry collector. The ONOS controller is connected to the Stordis controller port for dynamic forward table configuration. The telemetry collector server is connected to a switch data port to get in-band telemetry data. A Spirent traffic center (receiver and generator) has been connected to two Ethernet ports of the switch, to generate several data flows.

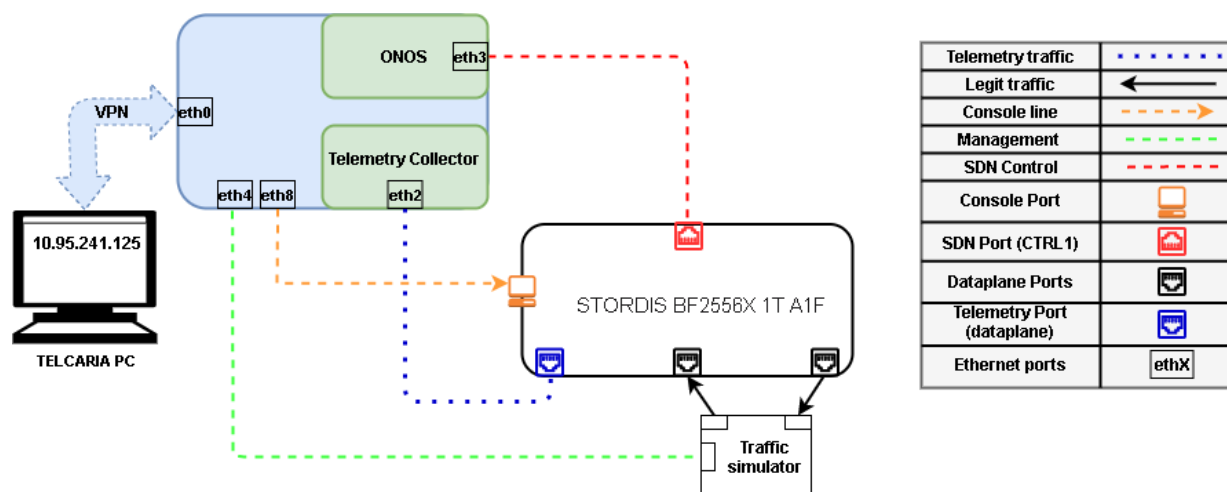


FIGURE 22: SCENARIO ON TELEFONICA PREMISES

In the scenario presented above, we can collect telemetry both with legacy and in-band approaches. The first goal is to replicate the PoC of Section 2.3.2.4, enabling a gNMI interface on the switch for collecting telemetry data in OpenConfig Format. The second goal is to experiment with in-band telemetry, since the Bmv2 software switch does not provide this kind of functionality.

The experimentation campaign includes the investigation of the following aspects:

- Evaluate the resource consumption of telemetry, both classic and in-band, on the switch's performance, especially under conditions of heavy traffic.
- Comparison of the information provided by in-band telemetry with respect to classic telemetry. Combine the two sources to achieve greater visibility on the device state.
- Evaluate traffic overhead due to in-band telemetry and available strategies to control it.

3. Pilot Integration

3.1. INNOVALIA Pilot

3.1.1. Technical Requirements and Related KPIs

As already explained in D4.3 [6] Section 3.1.1, the performance that can be achieved in the use cases depends on the capabilities of the underlying technology (i.e., 5G NR interface), the setup and the available resources. Recapping the information already displayed in that previous deliverable, the document that has been used as a reference for the achievable performance figures of the radio interface is the ITU-R report: "Minimum requirements related to technical performance for IMT-2020 radio interfaces" (<https://www.itu.int/pub/R-REP-M.2410>). Considering this plus the radio conditions and the implemented setup, it was established that the values gathered in the Table below would be reasonable. These targets remain the same for the validation campaign covered in this document as those for previous validation campaigns.

TABLE 6: KPI REQUIREMENTS FOR INNOVALIA USE CASES

	INNOVALIA UC1	INNOVALIA UC2
Network Latency	<20 ms + 0.5 ms/100 km	<20 ms
Data Rate	40 Mbps	40 Mbps
Mobility	static	3-50 km/h
Service Creation Time	<5 min	<5 min

The Mobility requirement is checked by verifying that the communication is not interrupted while the UE moves. The rest of requirements are validated by inspecting the related Core KPIs measured in this pilot, which are summarized in Table 7.

TABLE 7: CORE KPIS MEASURED IN INNOVALIA PILOT

Core KPI	Validation
CKPI-1 End-to-end Latency	Measured
CKPI-3 Guaranteed Data Rate	Measured
CKPI-6 Slice Creation Time	Measured
CKPI-8: Data Volume	Alternative methodology
CKPI-9 Jitter	Alternative methodology

3.1.2. ICT-17 and 5Gr Platform Integration

When deploying the solution distributed along the INNOVALIA and 5TONIC sites, no extra adaptations are needed to what was reported in D4.3 [6] Section 3.1.3. The workflow between the 5G EVE and 5Growth stacks is the same as reported there. The only difference is that the 5Growth stack will be physically located at a farther distance in the current case.

Regarding the metric collection, it is just the 5Probe that is deployed at Bilbao site, on the server that holds the Packet Gateway (PGW). This probe sniffs the user plane traffic and sends the metrics to the usual influxDB located at 5Tonic site. The flow continues all the way to the Kafka broker in the same manner as it did when the probes were deployed at 5Tonic.

3.1.3. Measurement Procedures

The final list of SKPIs that have been evaluated for UC1 *Connected Worker Remote Operation of Quality Equipment* are included in the following Table:

TABLE 8: INNOVALIA UC1 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	Main	Emul/Meas	Validation methodology
P1UC1-SKPI-1: Teleworker- CMM Synchronization (=5GR-SKPI-2)	CKPI-1 End-to-end Latency	Y	Emulated	Expressed as Bad/Acceptable/Good. This will be mapped with the measured Core KPI to establish the range of values that imply a Bad, Acceptable or Good synchronization.
P1UC1-SKPI-2: High-resolution Real-time Video Quality (=5GR-SKPI-4)	CKPI-3 Guaranteed Data Rate	Y	Measured	Expressed in values from 1 to 5, being 1 the lowest and 5 the highest quality perception. This will be mapped with the measured Core KPIs to establish the range that correlates to each QoE value.
	CKPI-9 Jitter	Y	Measured	
P1UC1-SKPI-3: Service Setup Time (=5GR-SKPI-1)	CKPI-6 Slice Creation Time	Y	Measured	Expressed in minutes. It will be measured by using the timestamps of the activity logs. It will be the difference between the timestamp when the service instantiation was requested and the timestamp when the instantiation confirmation is received. As the NS and VNFs need to be instantiated, it can take some minutes.
P1UC1-SKPI-4: Radius of Operation (=5GR-SKPI-5)	CKPI-1 End-to-end Latency	Y	Emulated	Expressed in kilometers. To obtain the distance, first, latency is measured, and artificial extra latency is added to find the maximum value of latency that allows a proper remote operation of the machine. That value is then translated into a distance parameter, assuming that every 100 km a packet has to traverse adds half a millisecond to the one-way latency and thus, 1 ms to the RTT.
P1UC1-SKPI-5: Integrated Multitype Communications (=5GR-SKPI-6)	CKPI-3 Guaranteed Data Rate	Y	Measured	Expressed as Success or Fail. It will be verified whether the guaranteed data rate for each slice is met. In real-life network conditions, there will be other users using the network as well. The

				5G network guarantees that the resources are distributed among the users according to their requirements.
P1UC1-SKPI-6: Extensive Network Coverage in the Factory Premises (=5GR-SKPI-7)	CKPI-1 End-to-end Latency	Y	Emulated	Expressed in m ² . The strategy for testing will be trying the performance at increasingly farther distances from the radio antenna to obtain the maximum distance that enables a good performance of the use case. The area can then be calculated as: $\pi \times \text{distance}^2 \times (\text{antenna beam width}/360)$.
	CKPI-3 Guaranteed Data Rate	N	Measured	

The SKPIs that have been finally quantified for UC2 *Connected Worker Zero Defect Manufacturing Decision Support System* is listed in the Table below:

TABLE 9: INNOVALIA UC2 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	Main	Emul/Meas	Validation methodology
P1UC2-SKPI-1: Service Operation Time (=5GR-SKPI-8)	CKPI-1 End-to-end Latency	N	Measured	Expressed in minutes. The total time for the whole operation described will be calculated, but also the selected CKPIs will be measured to verify the right performance at each stage. 1.AGV->CMM: Sending of piece code (Packet Loss as main KPI, if the command is not received, next action will not be triggered) 2.CMM: Downloading inspection program (Data Rate as main KPI, the download speed will impact the most on time) 3.AGV->CMM: Piece in place for inspection (Packet Loss as main KPI, if the command is not received, next action will not be triggered) 4.CMM: Inspection execution (Data Rate as main KPI, as the results are sent in real-time to the workstation) 5.CMM->AGV: Piece can be taken back (Packet Loss as main KPI, if the command is not received, next action will not be triggered) 6-AGV: Piece ready for unloading (Packet Loss as main KPI, if the command is not received, next action will not be triggered)
	CKPI-3 Guaranteed Data Rate	Y	Measured	
P1UC2-SKPI-2: Service Creation Time (=5GR-SKPI-1)	CKPI-6 Slice Creation / Adaptation Time	Y	Measured	Expressed in minutes. It will be measured by using the timestamps of the activity logs. It will be the difference between the timestamp when the service instantiation was requested and the timestamp when the instantiation confirmation is received. As the

				NS and VNFs need to be instantiated, it can take some minutes.
P1UC2-SKPI-3: AGV-Edge Control Synchronization (=5GR-SKPI-2)	CKPI-1 End-to-end Latency	Y	Emulated	Expressed as Bad/Acceptable/Good. This will be mapped with the measured Core KPIs to establish the range of values that imply a Bad, Acceptable or Good synchronization.
P1UC2-SKPI-4: Network support for user mobility (=5GR-SKPI-3)	CKPI-3 Guaranteed Data Rate	Y	Measured	Expressed as Success/Fail. This will require a test where the end device is moving, so it can be evaluated with the traffic sent by the AGVs while they are moving. The traffic should be stable and not be altered too much while the AGVs move. The packet loss will be increased artificially to see how it impacts the ability of the AGVs to move smoothly and the ability of the 5G network to guarantee the data rate.
P1UC2-SKPI-5: Concurrency of simultaneous users in a given area (=5GR-SKPI-7)	CKPI-1 End-to-end Latency	Y	Emulated	Expressed in m ² . The strategy for testing will be trying the performance of the AGV at increasingly farther distances from the radio antenna to obtain the maximum distance that enables a smooth performance. The area can then be calculated as: $\pi \times \text{distance}^2 \times (\text{antenna beam width}/360)$.
	CKPI-3 Guaranteed Data Rate	N	Measured	

The CKPIs that finally can be measured within this pilot are summarized in Table 10.

TABLE 10: CKPI MEASUREMENT METHODOLOGY

CKPI	Tool	Measurement methodology
CKPI-1 End-to-end Latency	5Probe	The probe calculates the round-trip time it takes to the packet since it leaves the source until the acknowledgement is received in response. It is measured in ms.
CKPI-3 Guaranteed Data Rate	5Probe	The probe measures the number of bytes of every packet that goes through the probe per second. The downlink and uplink traffic are measured independently.
CKPI-6 Slice Creation Time	Vertical Slicer	This CKPI is measured using the logs from the Vertical Slicer, obtaining the difference between the timestamp when the service creation was ordered and the timestamp when the confirmation is received that the service is available.
CKPI-8: Data Volume	Indirect from CKPI-3	Using the data rate metrics obtained with the probe and dumped into the database, the volume of traffic in Gbits will be calculated from the sum of all downlink and uplink throughput from the start of the experiment until it ends.
CKPI-9 Jitter	Indirect from CKPI-1	Having the value of latency of each packet, the delta of latency between packets is calculated and the average jitter over a period is obtained.

3.1.4. Innovation Validation and Impact on KPIs

The Innovation I8 proposes an approach to performance isolation based on network slicing, to retain bandwidth and delay guarantees. Details about how the innovation works and how it was integrated in the INNOVALIA pilot setup at 5Tonic laboratories were presented in D3.6 [3] Section 2.1.2.2.

The Innovation applies QoS policies to the network traffic of the use-case, prioritizing the traffic of the Optical Scanner device over the traffic of the Video Camera device. To achieve this, two network slices are defined to: i) isolate the network traffic of the two devices while traversing, programmable software switch; and ii) apply QoS parameters to the network slices to proportionally and fairly treat both slices based on the request throughput and delay guarantees, rather than just prioritize one slice and starve the other one. The goal of the validation is to show that the innovation can provide acceptable KPI performance for the pilot under conditions of traffic congestion, at least for one of the two slices. As declared in D4.3 [6] Section 3.1.1, the affected KPIs are:

- CKPI-1 End-to-end Latency
- CKPI-3 Guaranteed data rate
- CKPI-9 Jitter

To validate the innovation's features and demonstrate its benefits on the pilot, the test plan presented in Table 11 has been prepared. The video slice generates 30 Mbps, while the scanner slice generates 20 Mbps throughout the complete test plan, both using TCP as transport layer protocol and nonadaptive bit stream. With the QoSlicing application, we can change traffic characteristics for both slices. More specifically, on the one hand we can set QoS parameters to define the traffic profile of each slice in terms of transmission delay (T_DELAY) and latency (C_DELAY). On the other hand, we can set a priority value for each slice (a lower value corresponds to a lower priority) and these different priority values correspond to different physical queues, i.e., isolated also in queueing.

TABLE 11: TEST PLAN FOR THE VALIDATION OF I8 IN INNOVALIA PILOT

Test #	Bottleneck	Video slice priority	Video slice QoS parameters	Scanner slice priority	Scanner slice QoS parameters
0	No	None	None	None	None
1	25 Mbps	None	None	None	None
2	25 Mbps	1 (low)	C_DELAY = 1000 ms T_DELAY = 0	5 (high)	C_DELAY = 1000 ms T_DELAY = 0
3	25 Mbps	1 (low)	C_DELAY = 1000 ms T_DELAY = 0	5 (high)	C_DELAY = 20 ms T_DELAY = 500 ns/bytes
4	25 Mbps	None	C_DELAY = 70 ms T_DELAY = 400 ns/bytes	None	C_DELAY = 20 ms T_DELAY = 500 ns/bytes
5	25 Mbps	1 (low)	C_DELAY = 70 ms T_DELAY = 400 ns/bytes	5 (high)	C_DELAY = 20 ms T_DELAY = 500 ns/bytes

The first test represents the baseline condition, i.e., without introducing I8 innovation into the solution, while the remaining five tests are presented with I8 innovation. The initial test's objective

(Test #0 in Table 11) is to confirm that the Bmv2 switch introduction in the use case scenario neither affects the functionality, nor the performance of the already designed pilot networking setup. Regarding functionality, the new software switch acts as a L2 forwarding device and imposes no changes in L2 or L3 existing networking design. As for performance, the mini-PC running Bmv2 switch proves in this test to have the same forwarding capacity as the hardware switch it replaces. This was confirmed by measuring affected KPIs through dedicated probes for video and for scanner traffic. The following tests were decided to be depicted in a step-by-step manner to assure how this innovation can be properly applied for both slices. In other words, I8 innovation is not a plug-and-play scheme but will need detailed slice information and careful configuration to properly utilize the resources, in terms of queuing model and packet processing. In detail, priorities and QoS profiles were sequentially set and combined to see how the I8 Innovation could make an impact on the previous results.

To emulate the aforementioned a congested network, a bottleneck was created from test #1 to test #5 on one of the switch's interfaces. It is a tool for limiting the interface data rate, hence emulating a network congestion behaviour on the bottleneck link, and activating the innovation that affects both slices of network traffic. The Bmv2 switch allows setting up a bottleneck by configuring the packet rate and packet depth of all its physical queue. A bottleneck of 25 Mbps was implemented to ensure that only one of the slices could achieve its data rate.

Table 12 reports the results and conclusions about the tests execution.



TABLE 12: RESULTS FOR THE VALIDATION OF I8 IN INNOVALIA PILOT

Test #	Latency and Jitter (avg)	Data Rate (avg)	Comments
0	55 ms video RTT 10,22 ms video jitter	30 Mbps video rate	Base delays without any traffic congestion, i.e., both slices have a similar receiving data rate as their transmission data rate.
	38 ms scanner RTT 5,76 ms scanner jitter	20 Mbps scanner rate	
1	800 ms video RTT 96,38 ms video jitter	12 Mbps video rate	Huge delays with traffic limitation up to 25 Mbps. And we can see that both slices suffer a similar amount of delay RTT and receiving data rate.
	800 ms scanner RTT 136,39 ms scanner jitter	12 Mbps scanner rate	
2	1000 ms video RTT 88,36 ms video jitter	3 Mbps video rate	Due to the application of priority (higher priority goes to scanner slice), the scanner traffic is immediately forwarded and manages to i) keep low latency, ii) send data at its maximum rate, and iii) keep low jitter. However, camera slice will severely suffer in data rate and ~1 sec RTT.
	47 ms scanner RTT 0,85 ms scanner jitter	20 Mbps scanner rate	
3	800 ms video RTT 178,10 ms video jitter	10 Mbps video rate	Due to applied QoS policies to the scanner slice, its data rate is limited to 16Mbps with a smaller RTT than in test #2, limited in its physical queue size by the C_DELAY (~20ms). Also, the video slice can get the remaining capacity with improved delay RTT.
	26 ms scanner RTT 0,41 ms scanner jitter	16 Mbps scanner rate	
4	55 ms video RTT 2,87 ms video jitter	20 Mbps video rate	This test shows an opposite behaviour in which the video slice gets all its requested resources and base RTT as in test #0. In contrast, the scanner slice receives remaining resources with abundant RTT.
	66 ms scanner RTT 35,81 ms scanner jitter	3 Mbps scanner rate	
5	58 ms video RTT 9,00 ms video jitter	3-10 Mbps video rate	Thanks to the applied QoS policies to both slices, the scanner slice performs with the requested data rate (same as in test #3), while the video slice can keep its delay nearly to its base delay and jitter in test #0. We acknowledge that the data rate is dropped for the video slice; however, it will only decrease the quality of the video.
	26 ms scanner RTT 0,49 ms scanner jitter	16 Mbps scanner rate	

Test #1 is useful to show the behaviour of the use-case in a situation of traffic congestion on the network. Indeed, when setting a bottleneck of 25 Mbps in Test #1, both slices compete to use the total available throughput, resulting in a fair resource sharing. Therefore, each slice manages to reach an average data rate of 12 Mbps with very high latency RTT values. Figure 23 and Figure 24 show respectively the Data Rate and RTT measurements for Test #1.

Test 1 Data Rate Measurements

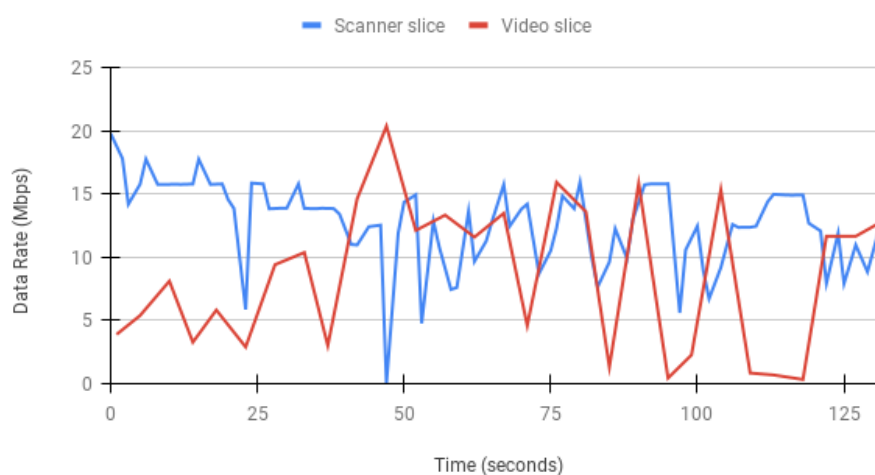


FIGURE 23: DATA RATE MEASUREMENTS FOR TEST #1

Test 1 RTT Measurements

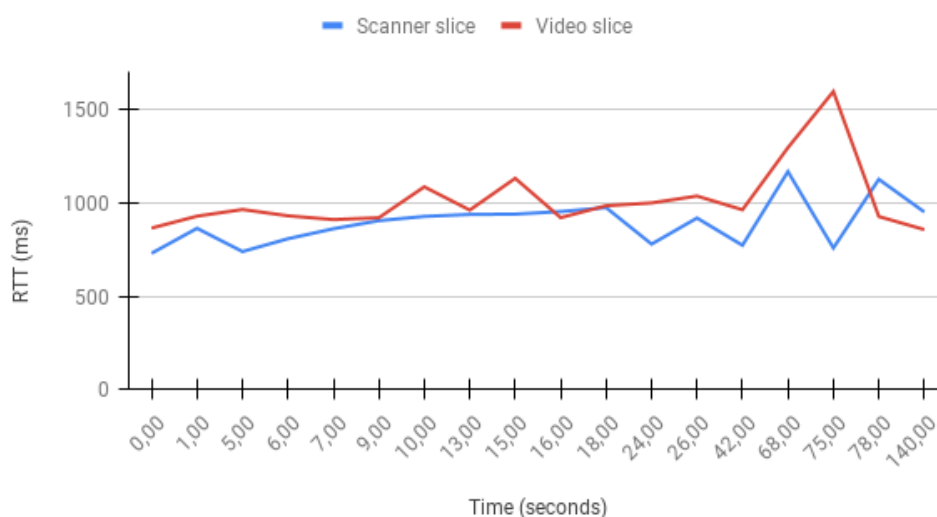


FIGURE 24: LATENCY MEASUREMENTS FOR TEST #1

In this situation of Figure 23, both slices have the same priority level, and thus they fairly share the medium and their aggregated data rate will consume the whole 25Mbps bottleneck capacity. Furthermore, the most critical slice, i.e., the scanner slice, requires low latency and sufficient throughput to provide an acceptable user-experience. Both KPIs cannot be provided under network

congestion without the implementation of the I8 Innovation QoS application. In conclusion, in this situation, the pilot is experiencing bad performance and it is completely unusable.

After modelling each slice traffic by limiting its rate with virtual queues, Test #5 finally proves the I8 Innovation QoS application functionality: even when the exact same bottleneck of 25 Mbps limits the total throughput, the video and the scanner slices manage to keep their latencies nearly to their base values, being able to ensure the latency KPI required together with the guaranteed data rate KPI proposed for the scanner slice. Figure 25 and Figure 26 show the results of Test #5.

Test 5 Data Rate Measurements

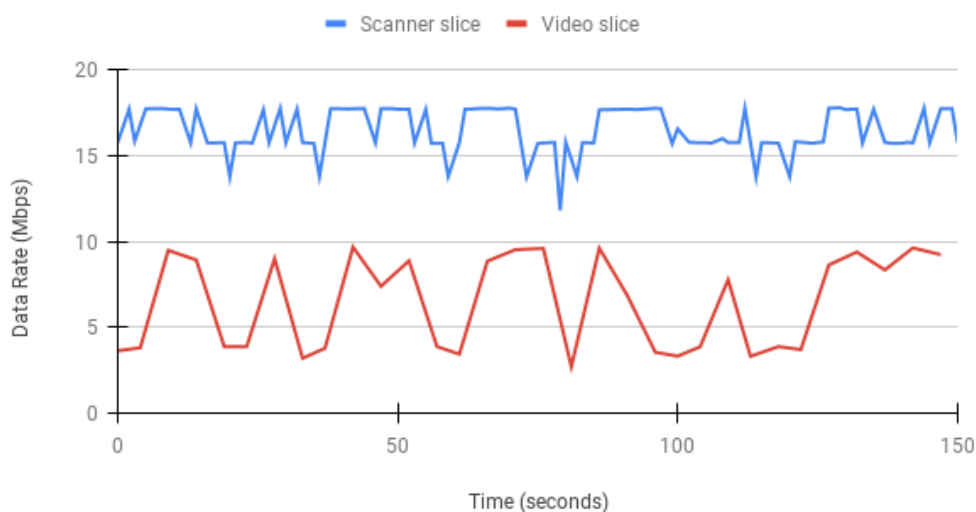


FIGURE 25: DATA RATE MEASUREMENTS FOR TEST #5

Test 5 RTT Measurements

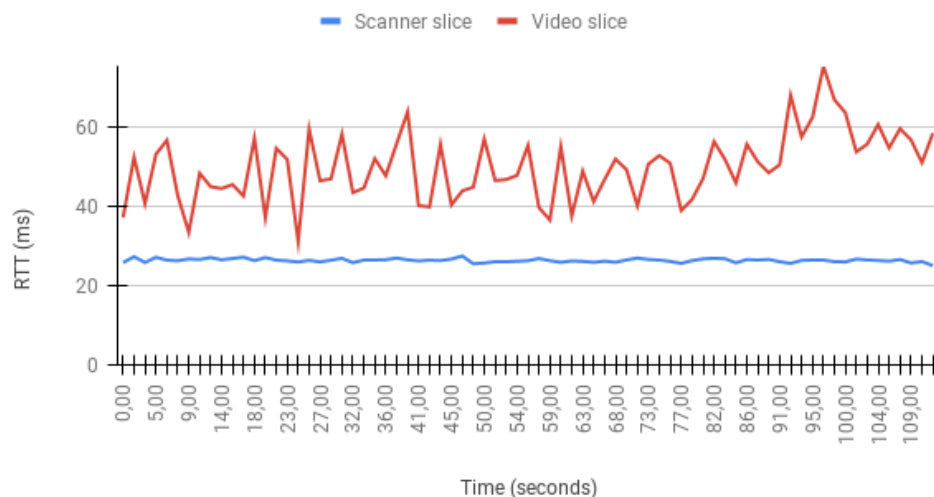


FIGURE 26: RTT LATENCY MEASUREMENTS FOR TEST #5

The scanner slice, whose performance is critical for an acceptable user experience, is given a higher priority and therefore it gets a throughput of 16 Mbps and a RTT latency of 26 ms. Please notice how

the QoS application managed to shape the slice traffic optimally, achieving a lower RTT latency with respect to the baseline seen in Test #0 (38ms) with no traffic congestion and no prioritization. The video slice is given a lower priority and therefore gets a lower data rate (3-10 Mbps) but still gets an acceptable RTT latency of 58 ms on average. Concerning the situation in Test #1, where both slices fight to share the scarce resources of the network without properly setting on QoS policies and priority levels, the video slice achieves an acceptable performance in Test #5 also verified by human inspection of the video stream.

In conclusion, the integration of the innovation has been validated successfully. The test results demonstrated that the innovation can separate the traffic of the two slices, but most importantly, keeping the KPIs of one of the two slices in acceptable value ranges for the correct operation of the pilot. Furthermore, the other slice with low priority is not starved in terms of resources and can still provide sufficient performance.

3.2. COMAU Pilot

3.2.1. Technical Requirements and Related KPIs

The technical requirements for the three use cases of the COMAU pilot have been defined in D1.1 [1]. Table 13 summarizes the target values expected for the three UCs. As reported in D3.6 [3], the challenge for UC1 is to achieve a low latency level, for UC2 is to serve a dense population of UEs, for UC3 to guarantee a data transfer at a high data rate.

TABLE 13: KPI REQUIREMENTS FOR COMAU USE CASES

	COMAU UC1	COMAU UC2	COMAU UC3
Network Latency	<15 ms	<100 ms	<15 ms
Data Rate	Up to 250 Mbps	Up to 250 Mbps	Up to 500 Mbps
Connection density	Up to 5 devices/m ²	Up to 5 devices/m ²	Up to 1 devices/m ²

It's worth noticing that, for UC1, the data rate of "up to 250 Mbps" refers to the data transfer between the computer where the digital twin application runs and the AR headset where the digital twin is presented side by side with the real robot image. This data transfer, and the related target, refer to a local WiFi or cable and not to the 5G connectivity. The challenge for UC1 is related to the latency as explained in detail in Section 4.2.1.

Similarly, for UC3, the target of 15 ms is not related to ensuring this low latency level between the two sites (factory and remote expert location). Instead, this target level is related to the case in which AR glasses are used by the maintenance technician on the shop floor of the factory. For this scope, the connection between the AR headset and the technician's laptop is typically ensured via local WiFi or cable. In essence, for UC3, the target low latency is not to be ensured by 5G while the target of 500 Mbps is asked to the 5G network. For this motivation, UC3 is served by an eMBB slice (and not by a URLLC slice as UC1).

Tables 17/18/19 of D4.2 [5] identify the service KPIs associated with each UC and the related mapping on the Core KPIs. For example, the Service KPI on “Support of industrial protocols over 5G networks” is associated with the Core KPIs of “End-to-End Latency” and “Packet Loss”.

Section 3.2.1 of D4.3 [6] reports the details of the validation methods applied to each of the Core KPIs listed in the table and how they are related to the Service KPIs. Table 14 provides just a recap of the Core-KPIs related to the COMAU pilot and which validation method is used for each one: Core-KPIs 1/2/3/9 are measured on the vertical premises while Core KPI 5/7/10/11 are verified through alternative methodologies.

TABLE 14: CORE KPIS VALIDATED IN COMAU PILOT

Core KPI	Description	Validation
CKPI-1	End-to-End Latency	Measured
CKPI-2	Packet Loss	Measured
CKPI-3	Guaranteed Data Rate	Measured
CKPI-5	Availability	Alternative Methodology
CKPI-7	Connection Density	Alternative Methodology
CKPI-9	Jitter	Measured
CKPI-10	Received Radio Signal Quality	Alternative Methodology
CKPI-11	Buffer Occupancy	Alternative Methodology

3.2.2. ICT-17 and 5Gr Platform Integration

To perform measurements without physically entering the vertical premises, the exploitation of the monitoring platform, which is part of the 5Growth platform, is planned for the first use case (UC1) in the pilot. One monitoring probe is deployed in a VM/container running in a PC connected to the Customer Premises Equipment (CPE) where the user’s terminals (e.g., robots, sensors, AR glasses...) are connected. The other probe is deployed in a VM/container in the APP SERVER. The probes are the ones described in D4.2 [5] and deployable through the 5Growth monitoring platform. They can collect latency, jitter, packet loss, the percentage of packet loss and the total transmitted packets. A common or two separate Prometheus exporters then export the data to the 5Growth platform where the monitoring platform is running. More details on the use of the monitoring platform in the COMAU platform are reported in D3.6 [3], Section 3.1.

3.2.3. Measurement Procedures

As indicated in Table 14, some Core KPIs have been validated through alternative procedures with respect to direct measurements through instruments. As for the measured one, the details on the related measurement setup and procedures are reported in the following.

The measurements in COMAU pilot have been conducted in a testbed located on the shopfloor of COMAU. A photograph of the area and a general scheme of the pilot are reported in D3.6 [3], Section 3. Figure 27 illustrates a schematic of the measurement setup used to assess the performance of the measured Core-KPIs 1/2/3/9.

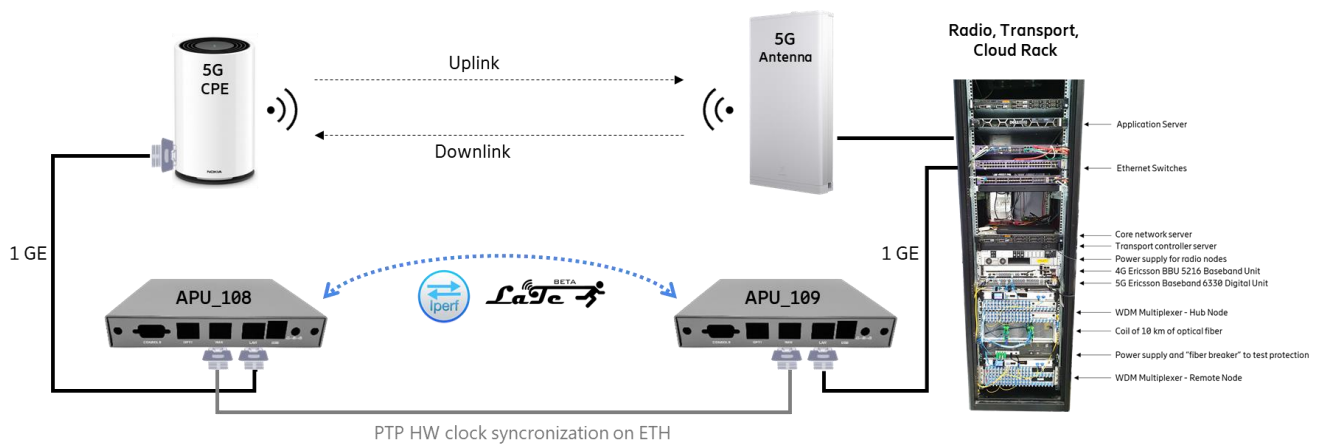


FIGURE 27: MEASUREMENT SETUP IN THE COMAU MOBILE INFRASTRUCTURE

In this lab setup, a 5G CPE communicates with the Ericsson 5G antenna AIR 6488 in the downlink direction (DL), from the antenna to the CPE, and in the uplink direction (from the CPE to the antenna). Two PC Engines APU2C4, labeled APU_108 and APU_109 in the figure, are connected to the two endpoints of the radio network assessing the transmission performances from the 5G CPE to the radio core network server. The measurements evaluate the performance of the entire radio-transport chain as the optical systems and the fiber coil (10 km) are traversed by the eCPRI flow supporting the radio traffic. More details on the measurement setup are available in D4.3 [6] Section 4.2.

With this test setup, three validation campaigns have subsequently assessed the target performances to validate the associated KPIs. A measurements campaign is organized in "sessions", each during at least four days to obtain a significant set of measured values and to intercept possible "spikes" of performances, if any. Specifically:

- The **first validation campaign** (December 2020), which results have been reported in D4.2 [5], has verified the Core KPIs on the radio network before integrating with the transport infrastructure. In this first campaign, the transport layer has been verified as a separate entity in the Ericsson (TEI) labs in Pisa.
- The **second validation campaign** (June 2021), which results have been detailed in D4.3 [6], has been the one that have measured the performances on the radio-transport network as an integrated entity. This measurement campaign has provided important indications tuning the 5G network to achieve the best possible performances (especially in terms of RTT latency). Based on these indications, in September 2021, the network has been further configured by Ericsson technicians that have worked for a week on the COMAU premises in tight coordination with the TIM staff. The scope of the tuning was to optimize the network performances to achieve the best possible latency as all the other parameters was already measured to be on target in this second campaign.
- The **third validation campaign** (September 2021), which results are reported in Section 4.2 of this deliverable, has verified the network performances in relation to the final KPIs targets validated on the fully integrated pilot after the mentioned network tuning. All the parameters (i.e., Core KPIs) was already complying with targets after the second validation campaign

except latency for which a specific tuning has been operated before running this third validation campaign. The measured latency after the tuning is reported in Section 4.2.1 (Figure 55).

As reported in the following Section 3.2.4, the **monitoring platform** is used to perform measurements from remote, without the need to visit the vertical premises. In November 2021, as a complement of the third validation campaign, three days of additional measurements have been performed from the SSSA laboratories located in Pisa, Italy, connected to the monitoring platform located in Turin.

The automated performance analysis, using the 5Growth platform, has been conducted in the testbed previously described with some additional components. Figure 28 shows the test setup. The 5G data-plane includes one CPE communicating with the Ericsson 5G antenna AIR 6488. The 5Growth orchestration platform has been installed in a virtual machine in TIM premises. TIM site is connected to the COMAU pilot network using a dedicated VPN as illustrated in D3.6 [3].

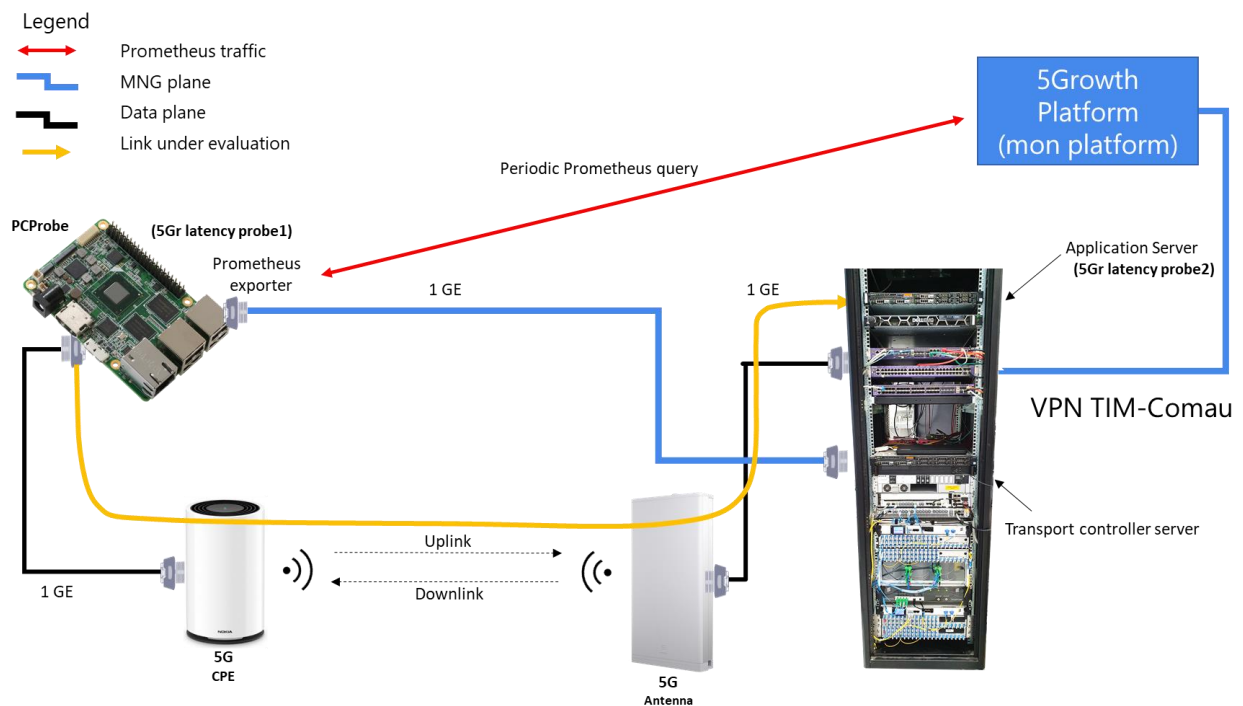


FIGURE 28: DYNAMIC MEASUREMENT SCENARIO IN COMAU

The link latency has been evaluated using two end points: 1) the PCprobe, constituted by an Upboard with Ubuntu 18.04 and Python3.8) installed behind the 5G CP; 2) the application server in the OpenStack cloud environment.

The measurement setup considers the performance evaluation of the full radio-transport chain, the optical system, and the fiber coil (10 km) used to transmit the eCPRI and the cloud abstraction (i.e., virtualization environment) to implement the OpenStack-based server app. Only the uplink latency (i.e., from the PCprobe to the server app) has been evaluated during the tests. The PCprobe and the server app run the 5Gr latency probe, described in Section 2.3.1 of D4.2 [5], pre-onboarded on the machines, because of the limited Internet access in the Comau testbed.

The PCprobe from one side is connected to the 5G CPE to reach the app server (data plane), while from the other it presents a direct connection with the transport server (management plane). In the latter case, acting as proxy (a destination NAT has been configured) towards the 5Growth platform, the transport server allows the transmission of Prometheus queries coming from the monitoring platform over the VPN connection.

3.2.4. Innovation Validation and Impact on KPIs

D2.3 [2], in Section 3, reports the 12 novel innovations aimed at covering the gaps motivated by the project pilot use cases. Among those innovations, Section 3.1.2 of D3.6 [3] reports the relevant ones for the COMAU pilot (i.e., I1, I2, I3, I6) and how such innovations have been used to enforce the three use cases.

I2 and I3 have an impact on the validation and on the related verification of KPIs. More specifically, I2 (i.e., "Support of Verticals. Vertical Service Monitoring") is used in the first use case (UC1) to create and instantiate probes at the endpoints of the use case to collect measurements (jitter, delay, packet loss) that are then conveyed to the monitoring platform. In parallel, I3 ("Monitoring Orchestration") introduces the monitoring platform which 5Gr-SO triggers, when a UC1 instance is created, to collect measurements from the probes and to evaluate the QoS KPI.

3.3. EFACEC_S Pilot

3.3.1. Technical Requirements and Related KPIs

The technical requirements for the two EFACEC_S Pilot use cases have been previously defined in D1.1 [1]. As mentioned before, the main target for Use Case 1 (UC1) is to achieve a low latency allowing the support of safety critical communications, while the main target of Use Case 2 (UC2) is related to guaranteeing a data transfer at a high data rate required for video transmission in scenarios involving mobility. Therefore, the UC1 is supported by ULLRC slice and UC2 is supported by eMBB slice. Table 15 summarizes the target values for the two UCs.

TABLE 15: KPI REQUIREMENTS FOR EFACEC_S USE CASES

	EFACEC_S UC1	EFACEC_S UC2	Notes
Network Latency	<10 ms (Note 1)	<100 ms (video images requirements)	Note 1: The system is configurable to work with network latency less than 50 ms, in compliance with railway signalling safety communications standards EN50159-2 class 77

Data Rate	Up to 10 Mbps (protocolar messages)	Up to 80 Mbps (Note 2)	Note 2: Up to 60 Km/h
Availability	99,99 %	99,99 %	

As reported in D4.2 [5], Tables 20/21 describe the relationship between each UC's service KPIs and the Core KPIs. In the D4.3 [6] Section 3.3.1, also some considerations concerning the KPIs, the measurements and test environment were reported and remain valid.

Table 16 summarizes the considered Core-KPI, regarding the EFACEC_S Pilot and the methodology to verify and validate the requirements.

TABLE 16: CORE KPIS VALIDATED IN EFACEC_S PILOT

Core KPI	Description	Validation
CKPI-1 (UC1)	End-to-End Latency (wheel sensors #1 or #2 to LX Controller)	Measured
CKPI-2 (UC1)	Packet Loss (wheel sensors #1 or #2 to LX Controller)	Measured
CKPI-5(UC1)	Availability (wheel sensors #1 or #2 to LX Controller)	Alternative Methodology
CKPI-1 (UC2)	End-to-End Latency (Video camera <-> Train driver console)	Measured
CKPI-2 (UC2)	Packet Loss (Video camera <-> Train driver console)	Measured
CKPI-3(UC2)	Guaranteed Data Rate (Video camera <-> Train driver console)	Measured
CKPI-9 (UC2)	Jitter (Video camera <-> Train driver console)	Measured
CKPI-5	Availability (Video camera <-> Train driver console)	Alternative Methodology

3.3.2. ICT-17 and 5Gr Platform Integration

The exploitation of the monitoring platform, which is part of the 5Growth platform, is planned for both UCs in the pilot. In the first UC, the measurements involve the communications between the wheel sensors and the Level Crossing Controller. Therefore, the probes were installed both at the sensors level as at the Lx controller level. As described in D3.6 [3], the EFACEC 5G CPEs include a processor unit, and the probes were deployed in these devices. Concerning the second UC, the measurements involve the communications between a Level crossing video camera and the onboard video console display). The probes were installed both at the processor unit (raspberry PI) located at the LX controller cabinet as at the console device. These probes can collect, for each link (wheel sensors to lx controller or video camera to onboard console display device) measurements such as latency, jitter, packet loss, the percentage of packet loss and the total transmitted packets; and can report to the 5G VINNI monitoring platform, where the metrics are collected through the SDA

component and exported to the 5Growth monitoring platform. This monitoring integration is described in D3.6 [3] Section 5.1.3.2, the UC in Section 5.2 and the probe in D4.2 [5] Section 2.3.1.

3.3.3. Measurement Procedures

As described in D4.3 [6], Section 3.3.1, it was adopted a measurement strategy involving validation campaigns at Lab environment level (Validation of Transportation Pilot Use Cases over a real 5G network -5G SA network integrating ASOCS RAN and Fraunhofer Open5GCore at IT Aveiro labs) and at vertical premises level (Aveiro harbor). According to this strategy, the following validation campaigns were performed:

- The **first validation campaign** (December 2020), which results have been reported in D4.2 [5], has verified the performance measurements at IT Aveiro labs (IT2) and some simulators were used to functionally validate the UCs.
- The **second validation campaign** (June 2021), which results have been detailed in D4.3, was also carried out at lab environment (IT2), since the deployment at vertical premises was not completely available. During this campaign it was also collected the improvements, related to the performance, that had been achieved since the first validation campaign. Finally, it were collected measures using the probes and the new applications functionalities.
- The **third validation campaign** (since July 2021), which results are reported in Section 5.2 of this deliverable, has verified the network performances in relation to the final KPIs targets and final use case functionalities, validated on vertical premises (Aveiro harbor). Since July, when the deployment was finished at the vertical premises, several campaigns were performed. Based on the collected results, several improvements were carried out, mainly to the 5G network available bandwidth but also in terms of coverage and stability. Also, some functional requirements had been fulfilled, such as fixed IPs (provided by the 5G network) and adjustments in the Use Case were performed and validated.

In this last validation campaign, three different approaches were performed, involving at first level the procedures described in D4.2 [5], Section 2.2.3, at second level the procedures involving the probes and the export of the measurements to Prometheus and finally the measurements collected at the application level (GUI), performed specifically for the Use Case 2.

3.3.4. Innovation Validation and Impact on KPIs

In this section we validate the usage of the Innovation 6 (I6) "End-to-End Orchestration, Federation and Interdomain" (interdomain component) and Innovation 11 (I11) "Security and auditability" working together over the pilot's infrastructure, as pointed out in D3.6 [3]. Summarily, the security innovation-based service (namely the Moving Target Defense (MTD) mechanism) is being deployed between two domains, the ICT-17 VINNI domain featuring SONATA and a OSM-based domain in the ITAV premises of the pilot, allowing for a proactive defense of network resources.

Furthermore, that the Interdomain innovation mechanism allows for the orchestration of an E2E Network Slice. This E2E Network Slice makes use of two different Network Slices of different Communication Service Providers (CSPs) Domains (i.e., VINNI and ITAV), as Network Slices Subnets. Each of the Network Slices Subnets manages the network resources provided by a different CSP Domain.

Upon being integrated into the 5GR-VS this innovation concept allows for the request of an E2E Vertical Service and it's respective E2E Network Slice by a given Vertical (i.e., EFACEC_S), who has assets in different CSP Domains.

Making use of ETSI Open-Source Mano (OSM) as a Network Slice Manager in both domains, the security innovation solution provides an Wireguard as a VPN Tunnel between the two domains, complemented with the MTD mechanism, which was used to protect these interdomain interfaces against reconnaissance, delivery, and undiscovered exploits in the exposed service stack. The MTD function works alongside the secure tunnel that interconnects the PoPs, leveraging a known Two-Factor Authentication (2FA) protocol to defend the secure tunnel. Additionally, the integrated operation of the interdomain and security innovations was also conceived to allow for the dynamic orchestration of the security service between both domains. This scenario is depicted in Figure 29.

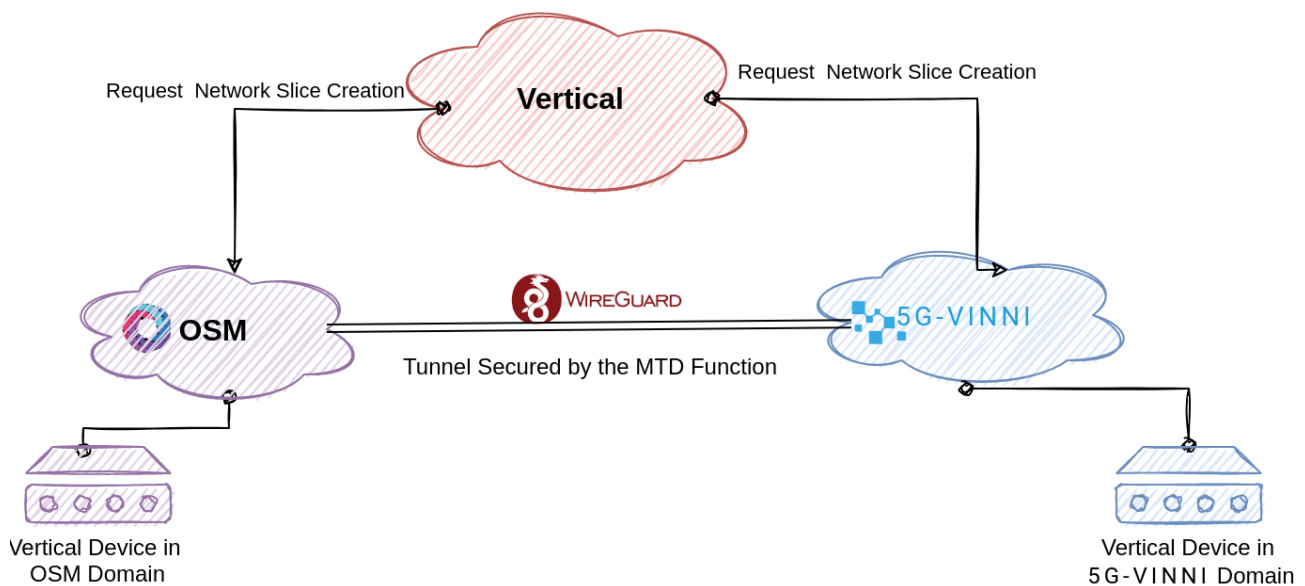


FIGURE 29: INTERDOMAIN+MTD SCENARIO

The signaling diagram of the orchestration of the E2E Interdomain Service and the MTD function, as well as the configuration of the VNFs, in each domain using the 5Growth-Vs is presented in Figure 30.

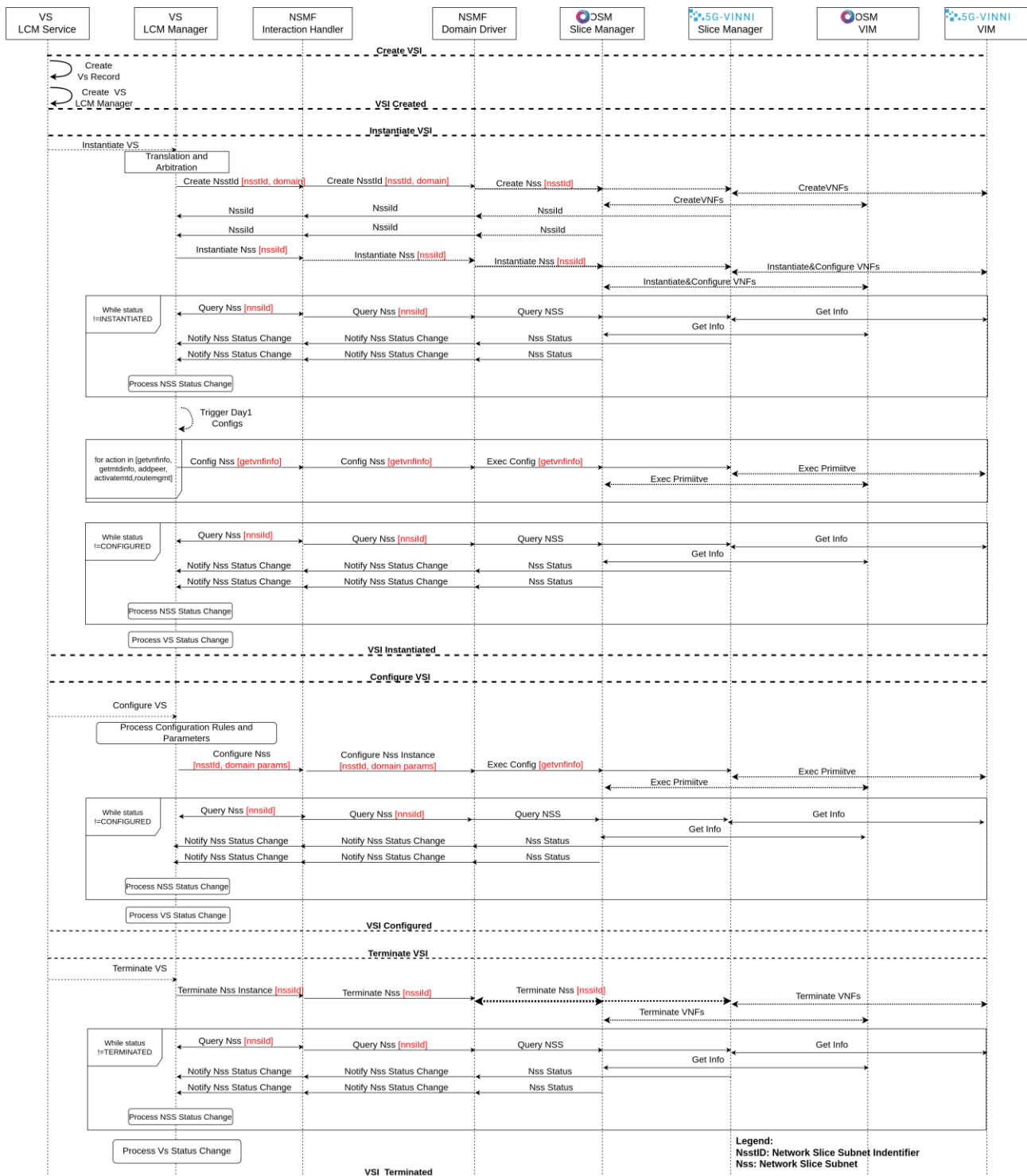


FIGURE 30: INTERDOMAIN + MTD INNOVATION SERVICE SIGNALLING DIAGRAM

The deployment of this service occurs upon establishment of the network service that is offered by the operators to the vertical, and therefore is already in place quite before the actual use cases begin. As a result, the availability of these innovations does not have direct implication in the KPIs associated with the use cases. Nevertheless, we hereby showcase results that reflect the instantiation performance of the interdomain deployment of MTD (considering, of course, that this deployment

is a prototype) in order to showcase the capability of added dynamism and flexibility that 5G networks are able to provide to verticals. The results focus on the average time needed for the instantiation, the configuration, and the termination of the E2E service and associated VNFs in each domain and were collected after 20 runs. These results are separated by each of the 3 main network slicing platforms involved in the process: 5Growth Vertical Slicer, 5G-VINNI (SONATA) and Open Source Mano (OSM).

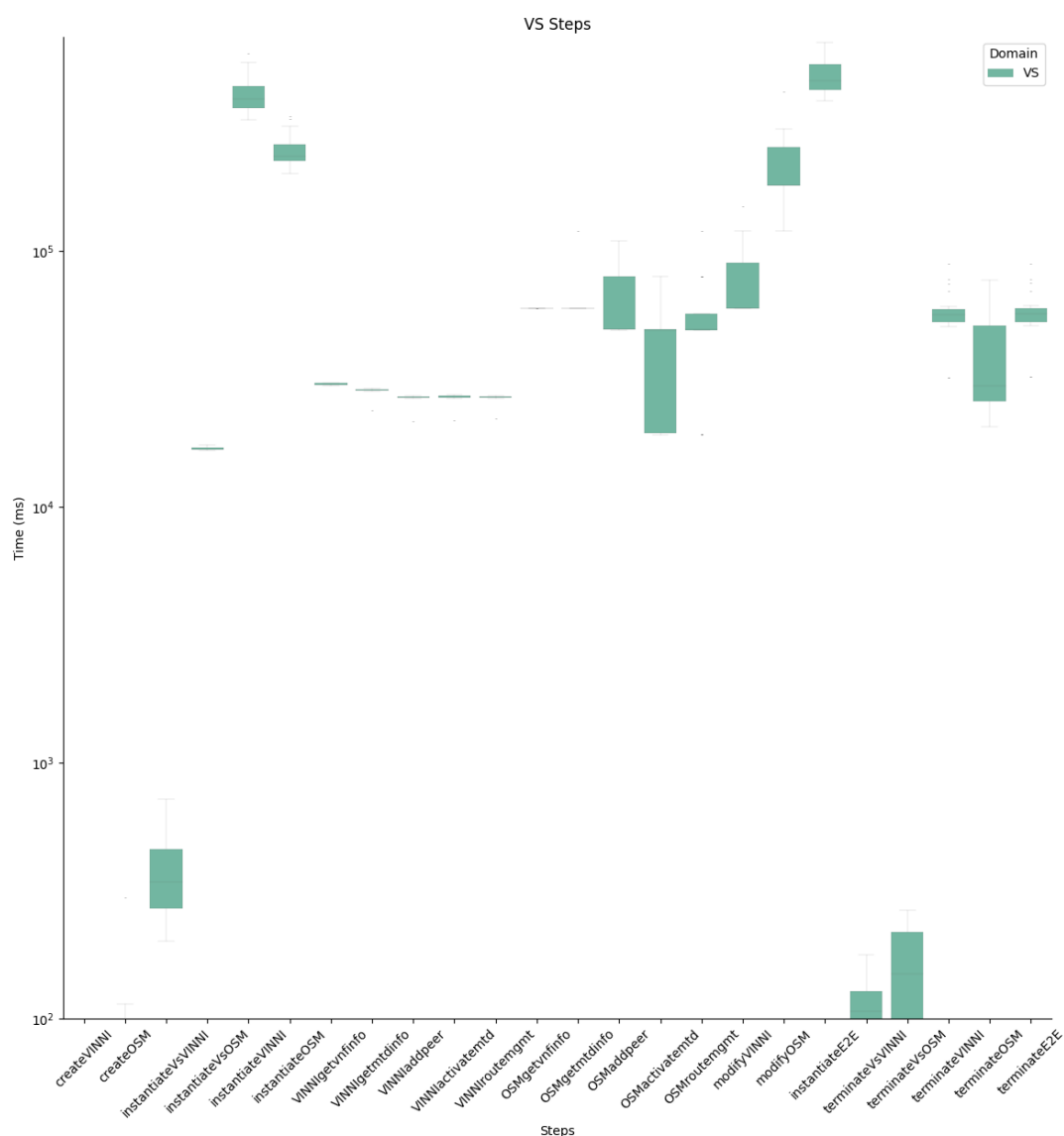


FIGURE 31: DELAYS ASSOCIATED WITH THE OPERATIONS AT THE VS LEVEL

Concerning Figure 31, we can verify that the most impacting instantiation delay is observed in the VS component. The main reason for these delays is related with the instantiation of the VNFs in each domain and afterwards with the modification the VNFs, whilst waiting for the exchange of

information between these domains, as a polling mechanism is being run in order to check if the information is already available.

Following these results, we present an analysis performed with the Log management tool described in Section 2.3.1. This analysis consists of twenty independent repetitions of the vertical instantiation operation involving the interdomain Innovation to extract the time metrics associated to the operations coordinated by the 5Gr-VS during the instantiation operation.

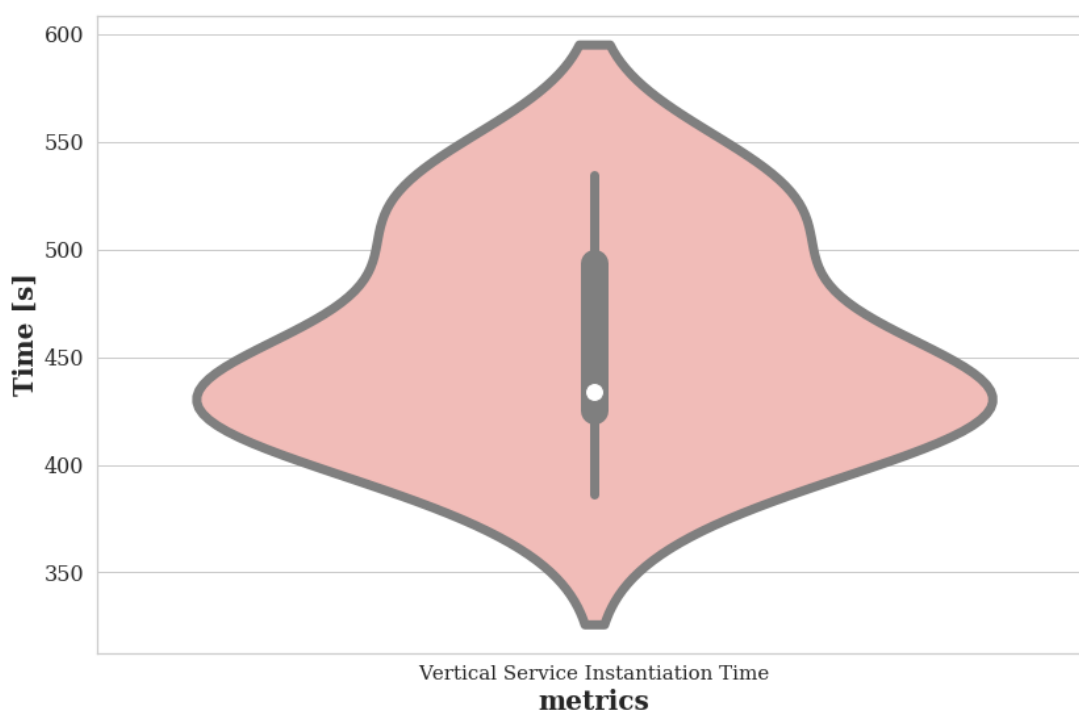


FIGURE 32: 5GR-VS INSTANTIATION METRICS INTERDOMAIN INNOVATION (I)

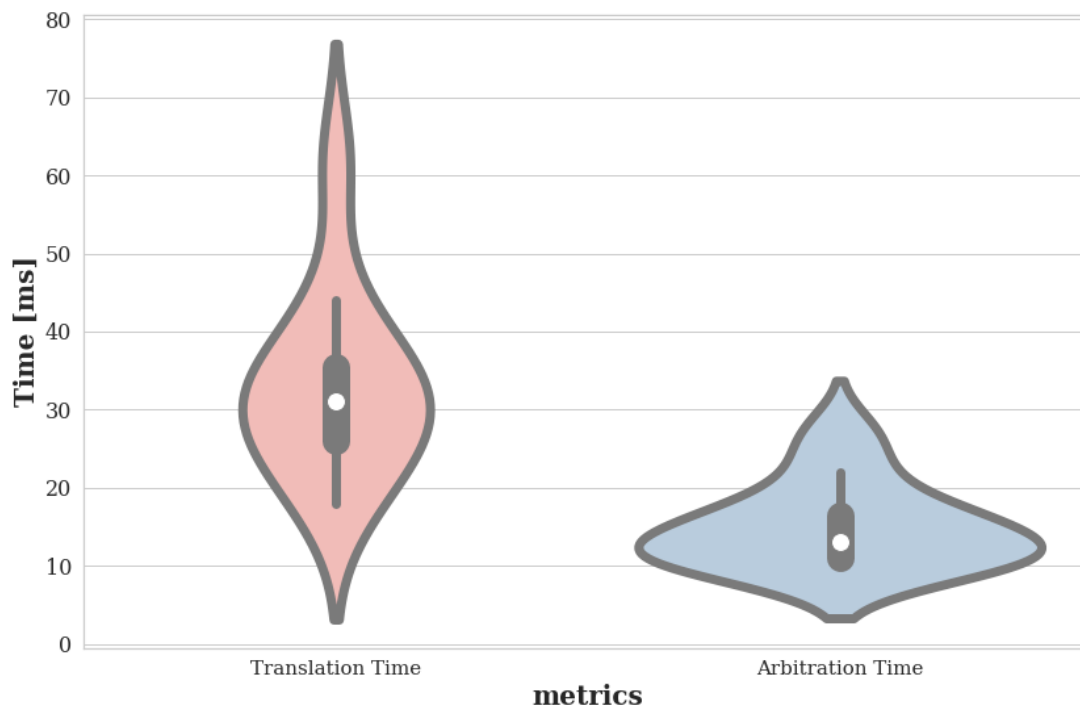


FIGURE 33: 5GR-VS INSTANTIATION METRICS INTERDOMAIN INNOVATION (II)

Figure 32 represents the violin plot¹ of the total vertical instantiation time measured from the 5Gr-VS module for the whole set of experiments evaluating the interdomain innovation. The median value is on the order of 425 seconds, and it is the time required by the 5Gr-VS to coordinate the deployment of a vertical service between the two independent domains. In this case, the 5Gr-VS presents an integration at the Communication Service Management Function (CSMF) with the different involved domains. These results are in line with the service delivery targets for next-generation mobile networks. Other relevant operations performed in the 5Gr-VS part of the instantiation process are translation and arbitration, as depicted in Figure 33.

In the translation operation, the 5Gr-VS performs a mapping between the high-level vertical request and the associated network slice template, the network service, and the domains in which it must be instantiated. The arbitration operation handles the distribution of the resources allocated to the vertical tenant in case there are different concurrent services.

The processing associated with the translation and arbitration operations requires times on the order of the milliseconds, and therefore does not have a large impact on the overall vertical service

¹ A violin plot represents a combination of boxplot and kernel density estimate. It shows the distribution of quantitative data across multiple levels of one (or more) categorical variables so that these distributions can be compared. Unlike a box plot, where all of the components of the plot correspond to actual datapoints, the violin plot contains a kernel density estimation of the underlying distribution. In violin plots, the white dot represents the median, the thick gray bar in the center represents the interquartile range, and the thin gray line represents the rest of the distribution, except for points that are determined to be "outliers" using a method that is a function of the interquartile range.

instantiation time, in the order of hundreds of seconds. As shown in next paragraphs and graphs, the most time-consuming operations are those related to the actual allocation of resources in the underlying infrastructure. This is the extracted conclusion when performing an analysis on the logs of the underlying orchestration blocks.

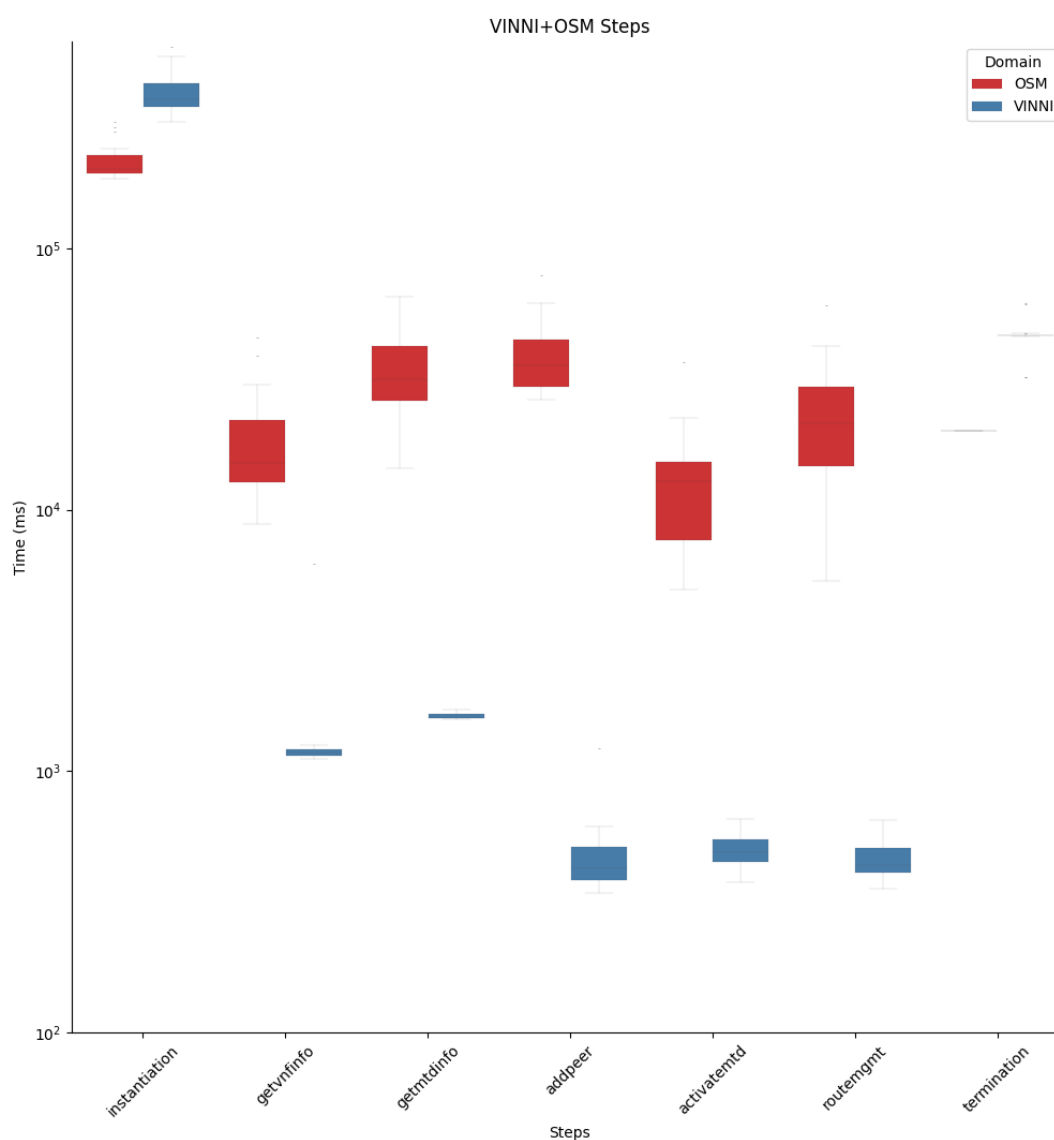


FIGURE 34: DELAYS OF THE OPERATIONS OVER THE VNFS IN EACH DOMAIN

As can be seen in Figure 34, the most time-consuming operation is the instantiation of the VNFS. It is also noticeable a major difference in the delays associated with the configuration of the VNFS in the 5G-Vinni and the OSM enabled domains.

The graphics for 5G-Vinni and the OSM enabled domains are presented in more detail Figure 35 and Figure 36.

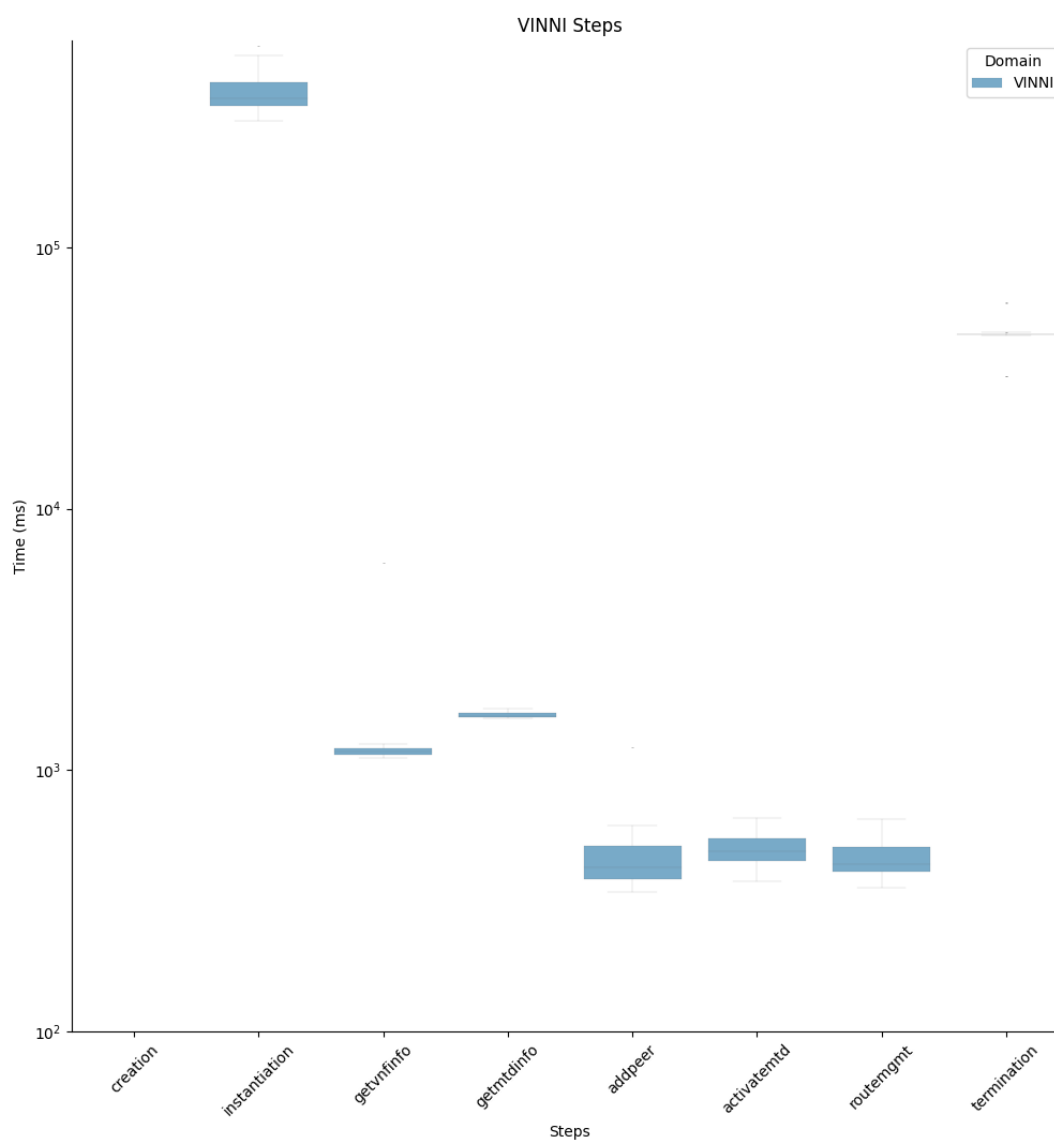


FIGURE 35: DELAYS OF THE OPERATIONS IN THE 5G-VINNI PLATFORM(SONATA)

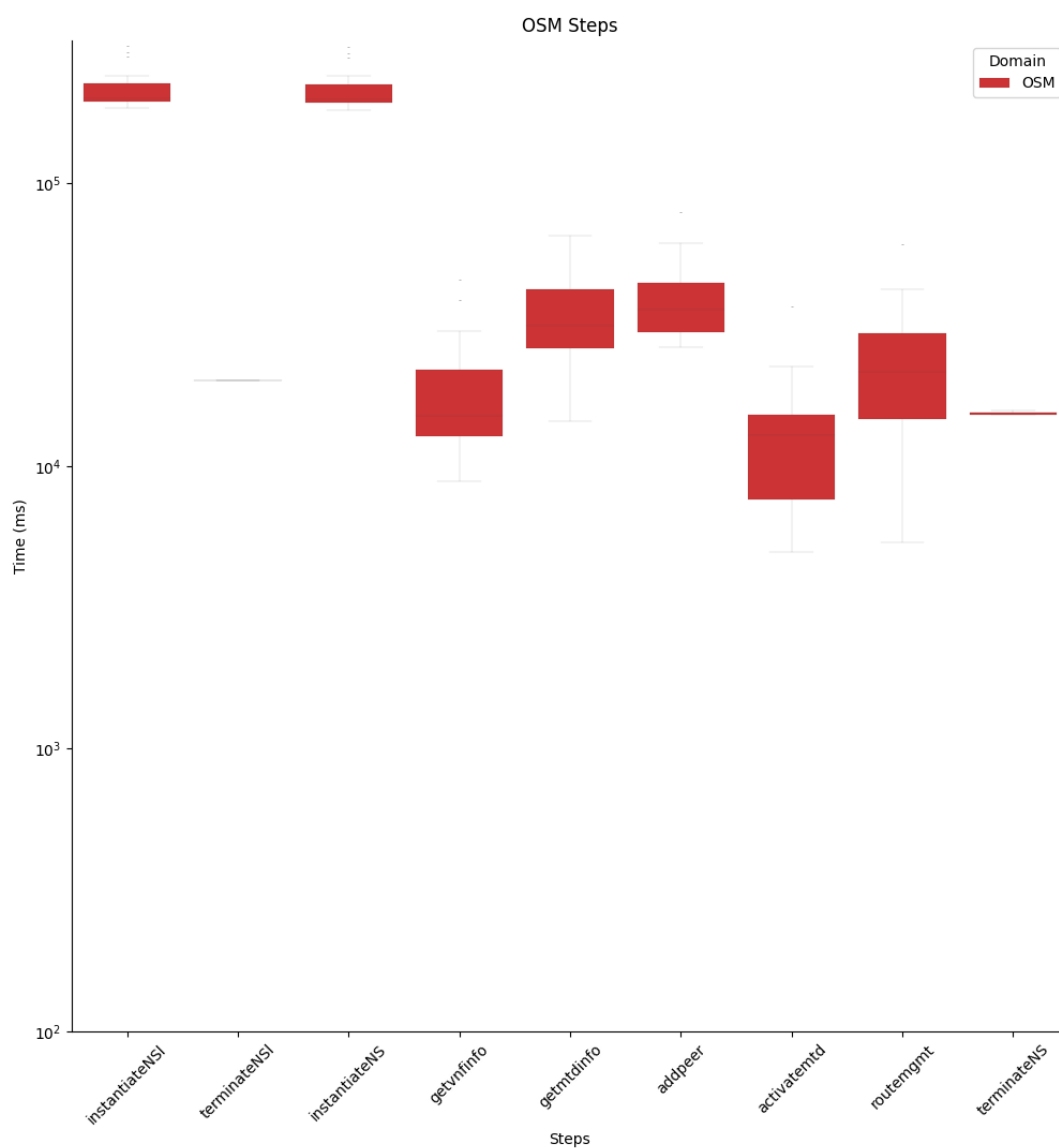


FIGURE 36: DELAYS OF THE OPERATIONS IN THE OSM ENABLED DOMAIN

The VS level descriptors related with this new configuration, templates and blueprints are available in the public GitHub repository dedicated to the 5GR Pilots [24].

The VNF packages, Network Service Descriptors and Network Slices used in the E2E Service are present on the following repository [25].

3.4. EFACEC_E Pilot

3.4.1. Technical Requirements and Related KPIs

The technical requirements for the two EFACEC_E Pilot use cases have been previously defined in D1.1 [1].

The relationship between the Service KPIs associated with each of the use cases and the Core KPIs was established and reported in Tables 22 and 23 in deliverable D4.2 [5].

Section 3.4.1 of D4.3 [6] reports the details of the validation methods applied to each of the Core KPIs listed in the table and how they are related to the Service KPIs.

As mentioned in previous deliverables, there are two main targets for the first use case (EFACEC_E UC1). The first target is to achieve a low latency communication allowing the telemetry information to reach the control center and the mobile crew with no perceived delay associated. The second target is to guarantee a proper bandwidth allowing the delivery of HD video with good quality and no glitches and to enable both remote control center operation and field maintenance team to work effectively.

The main target of the second use case (EFACEC_E UC2) is to guarantee a low latency communication between the low voltage sensors and the secondary substation controller, to support two requirements: to keep the low voltage sensors timely synchronized (synchronization between all sensors equal to or less than 1 ms) and to send the last-gasp capable devices information as soon as it is generated with very low latency ensuring that the device still has power.

Therefore, the UC1 is supported by eMBB slice and UC2 is supported by ULLRC slice. Table 17 summarizes the target values for the two UCs.

TABLE 17: KPI REQUIREMENTS FOR EFACEC_E USE CASES

	EFACEC_E UC1	EFACEC_E UC2	Notes
Network Latency	<100 ms	<10 ms	
Data Rate	Up to 20 Mbps	Up to 1 Mbps	
Availability	99,99 %	99,99 %	It is measured the service availability considering the number of lost packets per total amount of transmitted packages during a transmission period.

The specific considerations concerning the KPIs, the measurements, and test environment that were reported in the D4.3 [6], Section 3.3.1, are confirmed in this deliverable.

Table 18 provides a summary of the considered Core-KPI, regarding the EFACEC_E Pilot and the methodology to verify and validate the requirements.

TABLE 18: CORE KPIS VALIDATED IN EFACEC_E PILOT

Core KPI	Description	Validation
CKPI-1 (UC1)	End-to-End Latency (Gsmart in Secondary Substation to FE in Control Centre)	Measured
CKPI-2 (UC1)	Packet Loss (Gsmart in Secondary Substation to FE in Control Centre)	Measured
CKPI-5(UC1)	Availability (Gsmart in Secondary Substation to FE in Control Centre)	Measured (number of lost packets per total amount of transmitted packages during a transmission period.)
CKPI-3(UC1)	Guaranteed Data Rate (IP camera in Secondary Substation to Workstation in Control Centre)	Measured
CKPI-9 (UC1)	Jitter (IP camera in Secondary Substation to Workstation in Control Centre)	Measured
CKPI-5 (UC1)	Availability (IP camera in Secondary Substation to Workstation in Control Centre)	Measured (number of lost packets per total amount of transmitted packages during a transmission period.)
CKPI-1 (UC1)	End-to-End Latency (WFE in Control Centre to Mobile Device)	Measured
CKPI-2 (UC1)	Packet Loss (WFE in Control Centre to Mobile Device)	Measured
CKPI-5(UC1)	Availability (WFE in Control Centre to Mobile Device)	Measured (number of lost packets per total amount of transmitted packages during a transmission period.)
CKPI-1 (UC2)	End-to-End Latency (LVS3 in IT2 building to Gsmart in Secondary Substation)	Measured
CKPI-2 (UC2)	Packet Loss (LVS3 in IT2 building to Gsmart in Secondary Substation)	Measured
CKPI-5(UC2)	Availability (LVS3 in IT2 building to Gsmart in Secondary Substation)	Measured (number of lost packets per total amount of transmitted packages during a transmission period.)



3.4.2. ICT-17 and 5Gr Platform Integration

To perform measurements without physically entering the vertical equipment, the exploitation of the monitoring platform, which is part of the 5Growth platform, is planned for first use case (EFACEC_E UC1) in the pilot, with external equipment. One monitoring probe is deployed in a Raspberry Pi connected to a CPE which covers the "Secondary Substation" and the other probe is deployed in a VM "Scada" in the "Control Centre", this "Control Centre" is deployed in a Server inside the DC. This probe can collect latency, jitter, packet loss, the percentage of packet loss and the total transmitted packets. The other probes are deployed in all the VMs that belong the "Control Centre". This probe can collect the throughput in each direction of each interface in a VM. The probes report to the 5G VINNI monitoring platform where the metrics are collected through the SDA component and exported to the 5Growth monitoring platform. This monitoring integration is described in D3.6 [3] Section 4.1.3.2, the UC in Section 4.2, the first probe in D4.2 [5] Section 2.3.1 and the second probe in D4.3 [6], Section 2.2.2.

3.4.3. Measurement Procedures

As described in previous deliverables, along the project execution it was adopted a measurement strategy involving running validation campaigns at Lab environment level (Validation of Energy Pilot use cases over a real 5G standalone (5G SA) network integrating ASOCS RAN and Fraunhofer Open5GCore at IT Aveiro labs) and at vertical premises level (Secondary Distribution Substation in Aveiro University premises). According to this strategy, the following validation campaigns were performed:

- The **first validation campaign** (December 2020), which results have been reported in D4.2 [5], has verified the performance measurements at IT Aveiro labs (IT2) over two different setups: 5G network with emulated RAN, and 5G SA network with real RAN and real core.
Due to the impact of Covid-19 pandemic, the first use case was partially tested, and the second use case was not tested (according to Covid-19 project replanning).
- The **second validation campaign** (June 2021), which results have been detailed in D4.3 [6], was also carried out at lab environment (IT2), since the deployment at vertical premises was not ready and the main goal was to extend the validation of the Use Cases and verify the performance improvements since the first validation campaign. Measures were also collected using the 5GROWTH probes.
- The **third validation campaign** (since September 2021), which results are reported in Section 4.4 of this deliverable, has verified the network performance concerning the KPI targets over the final and complete deployment of the two Energy Pilot use cases, validated on vertical premises (Secondary Substation in Aveiro University, and IT building IT2).

Since September 2021, several testing sessions were performed on site, covering the two use cases incrementally, and based on the results several improvements were carried out, mainly

in 5G network bandwidth and stability. Also, some adjustments to the use cases setup were performed and validated.

In the third validation campaign three different approaches were taken to collect network measurements:

- 1) Using third party tools like iperf3 and ping, according to the procedures described in D4.2 [5], Section 2.2.4.
- 2) Using the 5Growth probes developed by SSSA and ITAV for collecting measurements and subsequent export to Prometheus
- 3) Collecting measurements at application level (in Gsmart trace log). This was performed specifically in the Use Case 2.

A general scheme of the pilot, showing the probes location is reported in D3.6 [3], Section 4.1.3.2.

3.4.4. Innovation Validation and Impact on KPIs

We refer the reader to Section 3.3.4 to the analysis of the multi-domain innovation and its impact in the vertical service instantiation operation driven by the 5Gr-VS.

4. Report of the Third Validation Campaign

4.1. Industry 4.0 pilot – INNOVALIA

This section includes the results obtained in the tests performed on the AIC site, where the UCs have been deployed for this last validation campaign, for the INNOVALIA pilot. All the details regarding the solution design, the infrastructure implementation, the components deployment, and the UCs setup are presented in Section 2 of D3.6 [3].

4.1.1. Use Case 1: Connected worker remote operation of quality equipment

4.1.1.1. Service deployment time

The following graphs present the time metrics extracted with the 5Growth Log Parser Tool described in Section 2.3.1 for ten repetitions of the automated deployment of Use Case 1. Figure 37 shows the violin plots of the evaluated 5Gr-VS metrics described in Section 2.3.1.

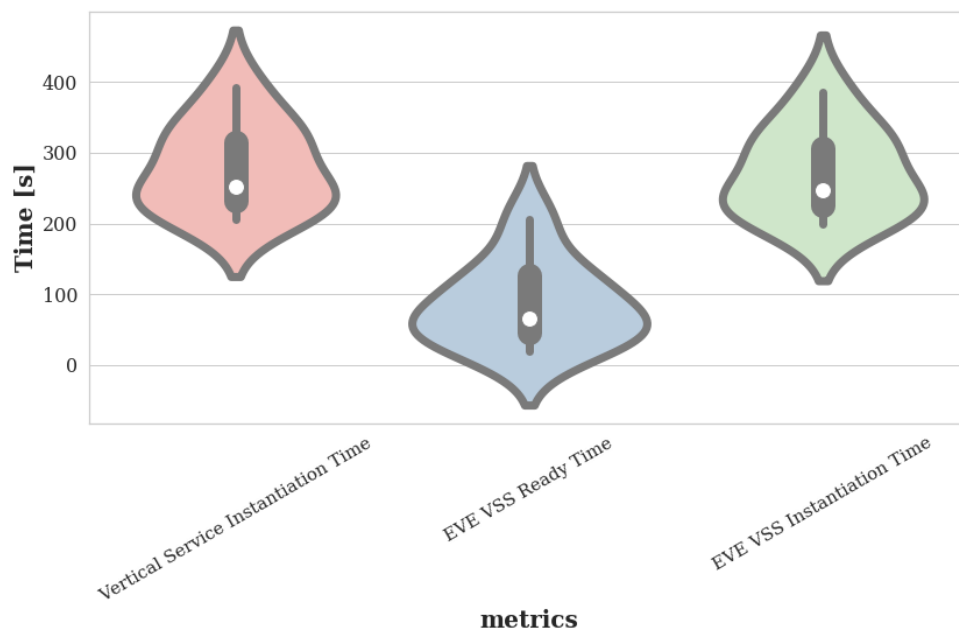


FIGURE 37: 5GR-VS INSTANTIATION METRICS UC1

As it can be observed, the median experienced time to deploy Use Case 1 (i.e., *Vertical Service Instantiation Time* metric) is in the order of 250 seconds. This value is in line with the target set in Section 3.3.1. This target is aligned with requirements for 5G networks, reducing the service creation time to the order of minutes. This time presents high variability mainly due to two factors: (i) the impact of the *EVE VSS Ready time* and (ii) the presence of several polling operations between 5Growth and 5G EVE platform. Regarding (i), this metric accounts for the time it passes since the 5Gr-VS requests the deployment until the human operator of the 5G EVE platform manually accepts the

request. Regarding (ii), there are three polling operation between 5Growth and 5G EVE in the whole deployment process: (a) the verification of acceptance of 5Gr-VS instantiation request in 5G EVE, (b) the polling the 5Gr-VS does to check that the instantiation is ready at the 5GEVE level, and (c) the polling the interworking layer (IWL) driver of 5G EVE does to the 5Gr-SO to check that the NFV-NS is in instantiated state.

The difference in time between *Vertical Service Instantiation Time* and *EVE VSS Instantiation time* metrics accounts the time required by 5Gr-VS to perform the required translation and onboarding operations from 5Gr-VS descriptor format to 5Gr-EVE descriptor format to make the instantiation request. These operations take less than 10 seconds.

Next, the following graphs present the time profiling obtained from the parsing of the 5Gr-SO logs for the deployment of UC1 after the request of the 5GEVE IWL driver. A complete description of the analysed metrics can be found in Section 2.2.1 of D4.3[6].

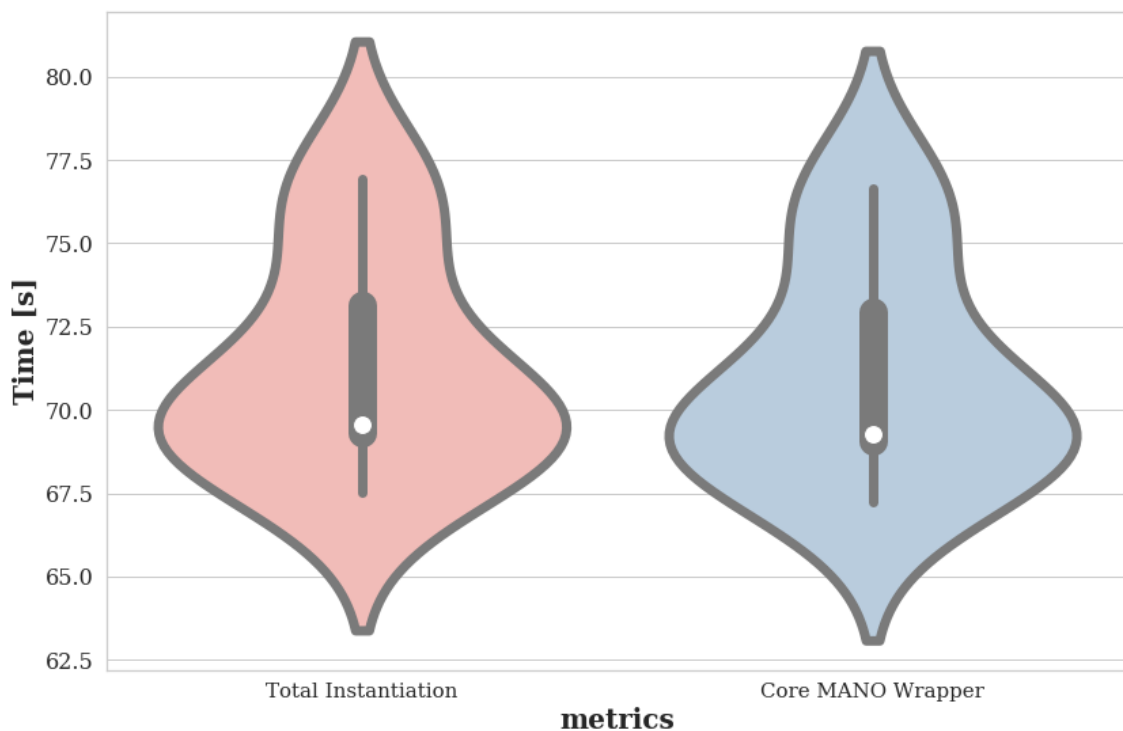


FIGURE 38: 5GR-SO INSTANTIATION METRICS UC1 – TOTAL INSTANTIATION AND CORE MANO WRAPPER TIME

Figure 37 shows the Violin plots for the Total Instantiation time and the Core MANO Wrapper time. Both values are set next to each other to assess the impact of the operations performed by the Core Mano Wrapper module of the 5Gr-SO in the Total Instantiation time. As it can be observed, they are practically the same, meaning that the operations performed by the Core Mano Wrapper interacting with the associated Cloudify CORE MANO platform are the most time-consuming operation in the whole instantiation process held at the 5Gr-SO. These operations are the creation of the necessary virtual networks at the corresponding edge datacentre to then attach the virtual machine (VMs)

implementing the VNF and the creation of such VM. A median value around 70 seconds is required to instantiate the NFV-NS at the 5Gr-SO level.

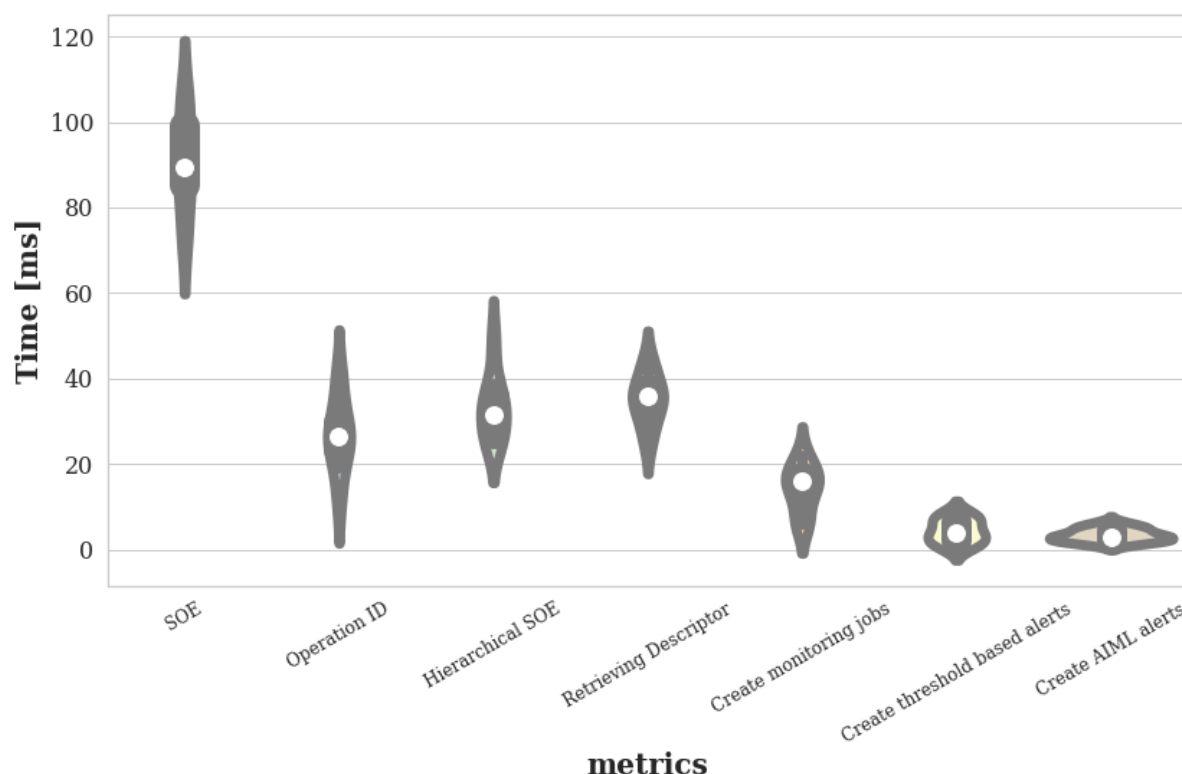


FIGURE 39: 5GR-SO UC1 TIME PROFILING OF SERVICE ORCHESTRATOR ENGINE (SOE) OPERATIONS

As it can be observed from Figure 39, the service orchestration operations performed by the SOE module of the 5Gr-SO only contribute around 90 ms, as median value, to the total experienced instantiation time. From these operations the most time-consuming one are the ones preparing and coordinating the following operations of the instantiation process (e.g., Operation ID, Hierarchical SOE dispatching (SOEp-SOEc), Retrieving descriptor from Catalogue DBs). The remaining metrics relate to the interaction with the modules configuring monitoring and alerts. They present a low value, devoted to analysing the network service descriptor to check if there is the need of configuring monitoring jobs or alerts. In UC1, there is no need for such configurations. Nonetheless, the required time would be in the order of few hundreds of milliseconds, as maximum, which represent a limited impact in the experienced total instantiation time.

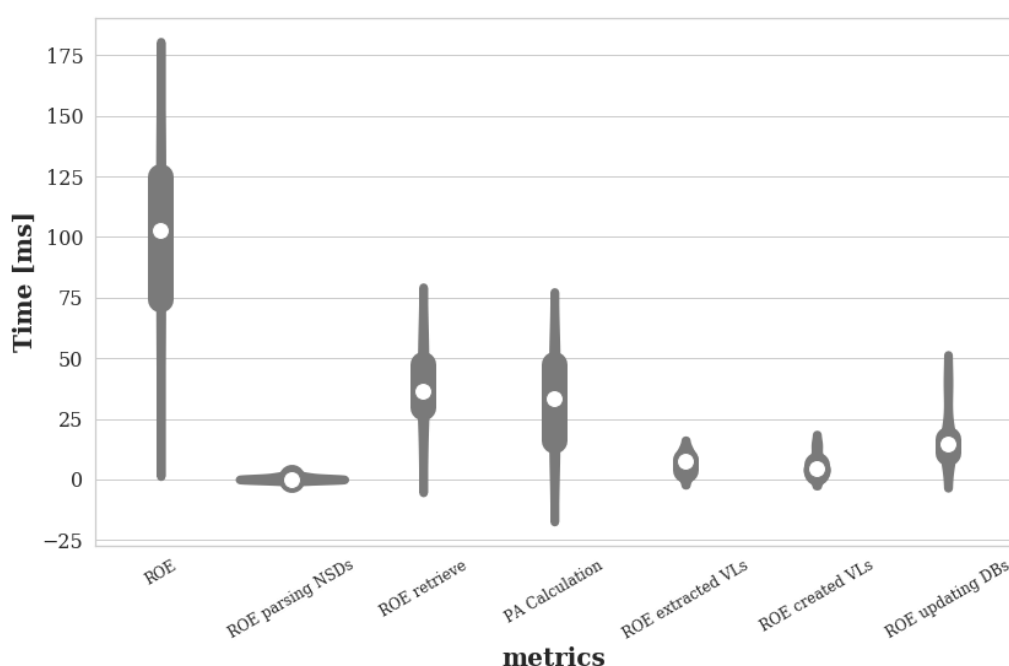


FIGURE 40: 5GR-SO UC1 TIME PROFILING OF RESOURCE ORCHESTRATOR ENGINE (ROE) OPERATIONS

From Figure 39, we can observe that the contribution to the total instantiation time of the resource orchestration operations performed by the ROE module is, on average, very similar to the one of the SOE. The most time-consuming operations, namely ROE Retrieve RL resources and PA calculation, which are in the order of tens of milliseconds. These operations are part of the process required to select the appropriate resources at the underlying infrastructure for the network service deployment and are just executed before the operation of the Core MANO wrapper. In this case the time devoted to extract and create virtual links is low because there is no need to establish a logical link between the single VNF of the UC1. The experienced time is due to the time to verify if such logical links are needed before its determination and establishment.

4.1.1.2. Network infrastructure throughput

To measure the maximum throughput achieved in the network under the existing radio conditions at AIC premises, some iperfs were performed.

Iperfs were run using mid-band frequency for both Innovalia and Automotive Smart Factory (ASF) rooms. Having the iperf client on the M3 PC and the iperf server on the INNO VM, running three parallel threads for a few minutes, provided an average result of 36 Mbps. Running the operation in reverse, with the same options, lead to an outcome of 34 Mbps. Then, the test was done with the iperf client on the INNO VM and the iperf server on a PC connected on the camera and CMM side. The value achieved for this test was 14 Mbps. For the execution in reverse, the result was 23 Mbps.

4.1.1.3. Use Case 1 Operation

The setup implemented for use case 1 at vertical premises is documented in D3.6 [3] Section 2.2. With such setup in place, the use case was run, and the results observed are documented below.

Regarding the CMM and M3 application traffic flows, they produce the subsequent metric graphs. First, it is shown the traffic pattern produced by the scan results sent from the VM towards the M3. The probe is measuring the traffic that goes through the EPG, so what will show significant results is the RTT Latency and the Data Rate between the probe and the M3, which includes one RAN segment path. The following graphs are showing traffic on TCP ports 51294-51298 and 1027 and with host IP 10.3.200.36 (the M3) involved.



FIGURE 41: UC1 RTT LATENCY AND JITTER FOR SCAN RESULTS TRAFFIC

The lines in Figure 41 show the result is an average of 140 ms RTT latency when there is traffic.



FIGURE 42: USER DATA RATE FOR SCAN RESULTS TRAFFIC

Figure 42 illustrates that the peak data rates demanded by the scan results go up to 10 Mbps.

Next, to see the metrics of the commands traffic including the RAN segment between the INNO VM and the CMM, the probe measures the commands travelling across the EPG from the INNO VM towards the CMM. The resulting RTT Latency is the RTT between EPG and CMM. The graphs include the TCP traffic on ports 51294-51298 and 1027 with destination IP 10.3.200.34 (the CMM IP).



FIGURE 43: RTT LATENCY AND JITTER FOR COMMANDS TRAFFIC

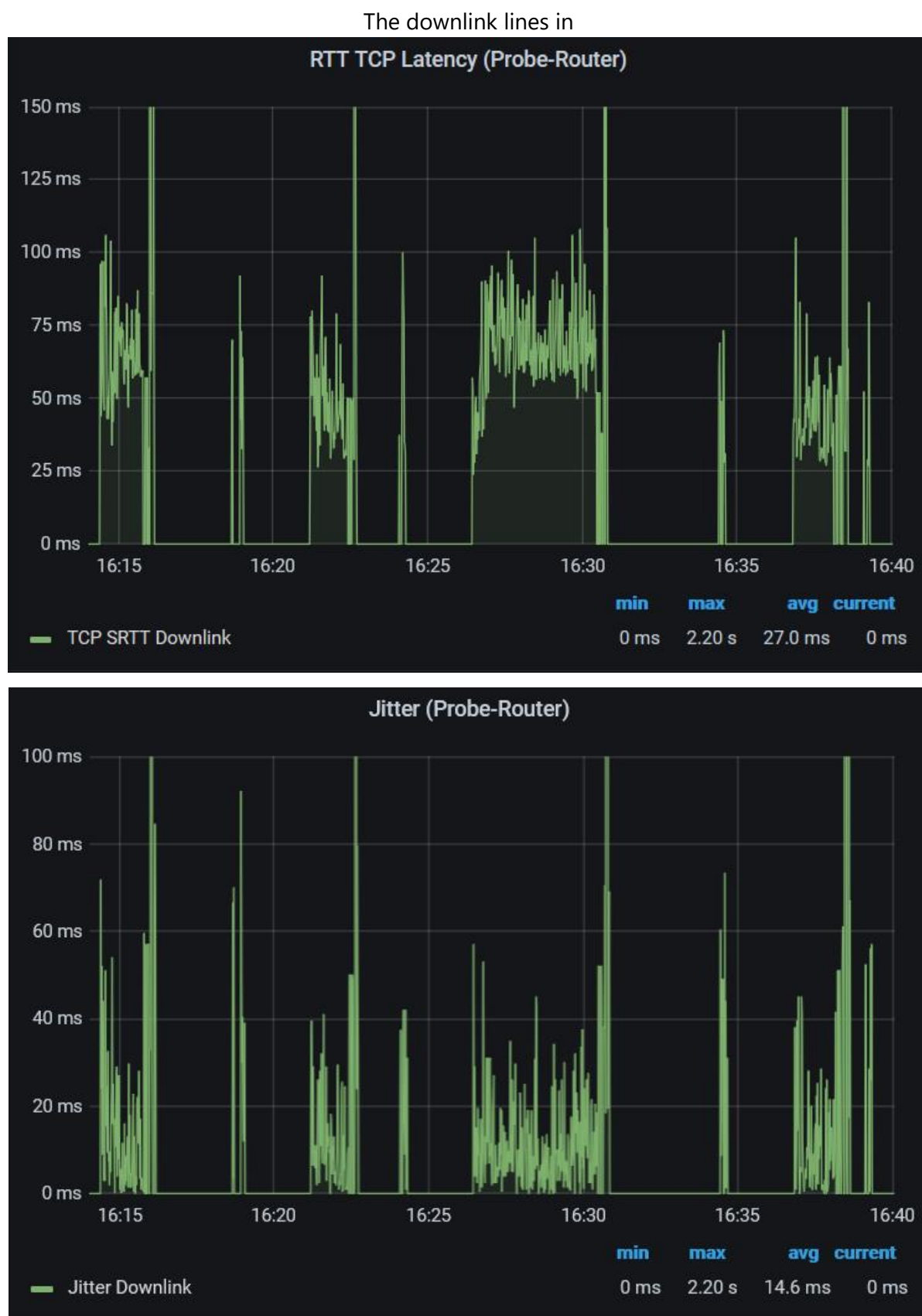


Figure 43 show the outcome is an average of 70 ms latency when there is traffic.

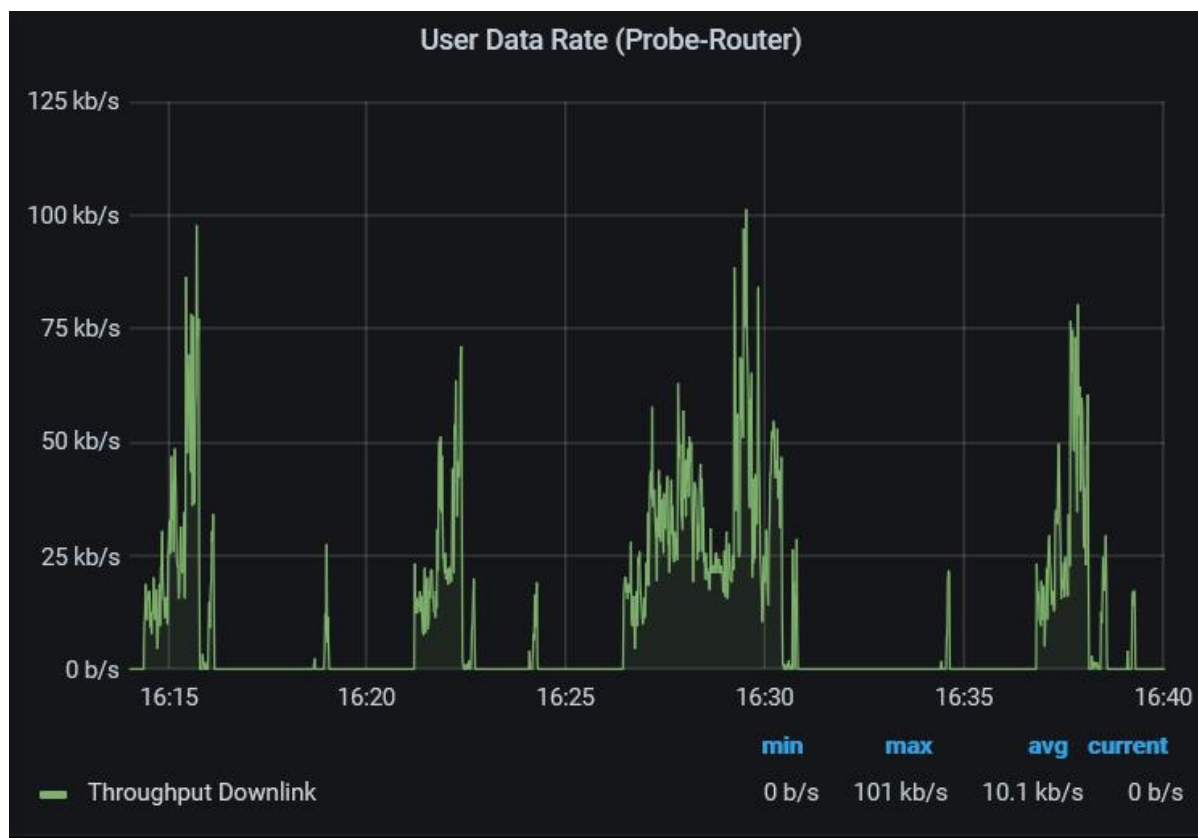
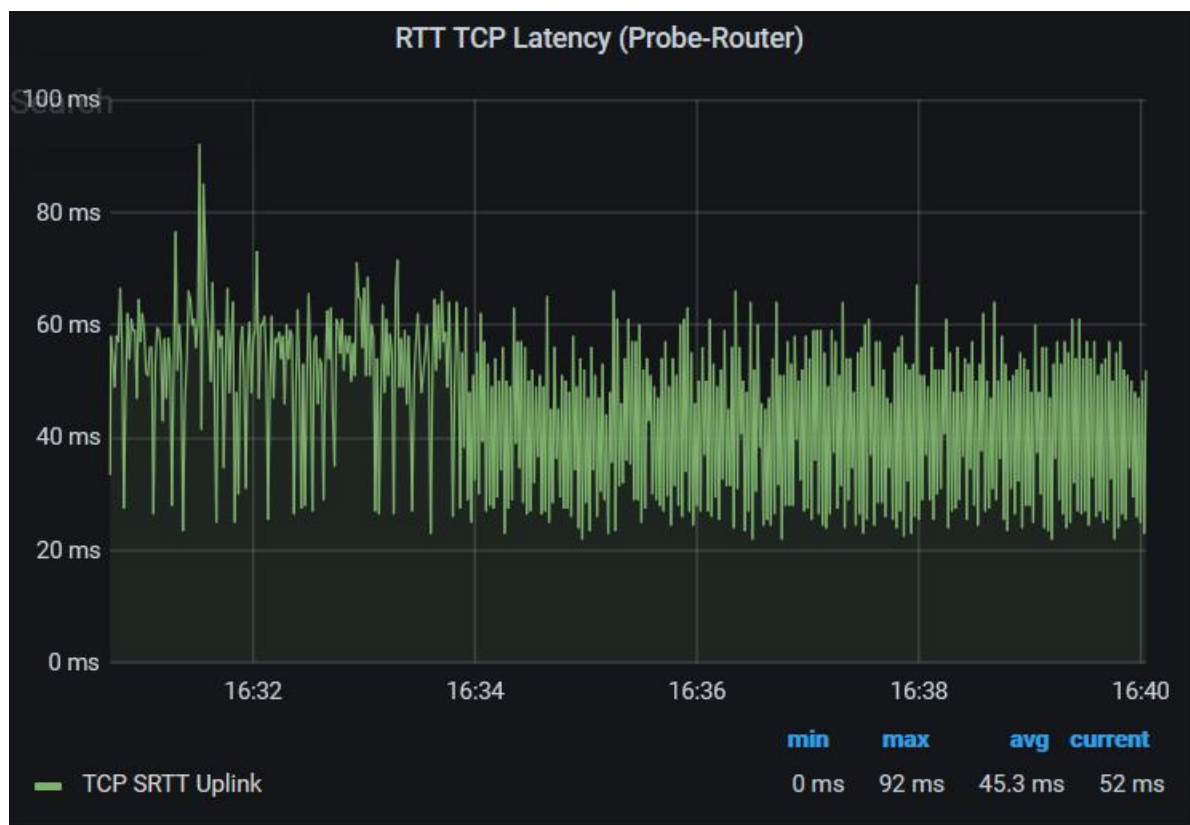


FIGURE 44: USER DATA RATE FOR COMMANDS TRAFFIC

For this kind of traffic, low data rates are demanded, rarely exceeding 80 kbps.

The video streaming produces the following graphs:





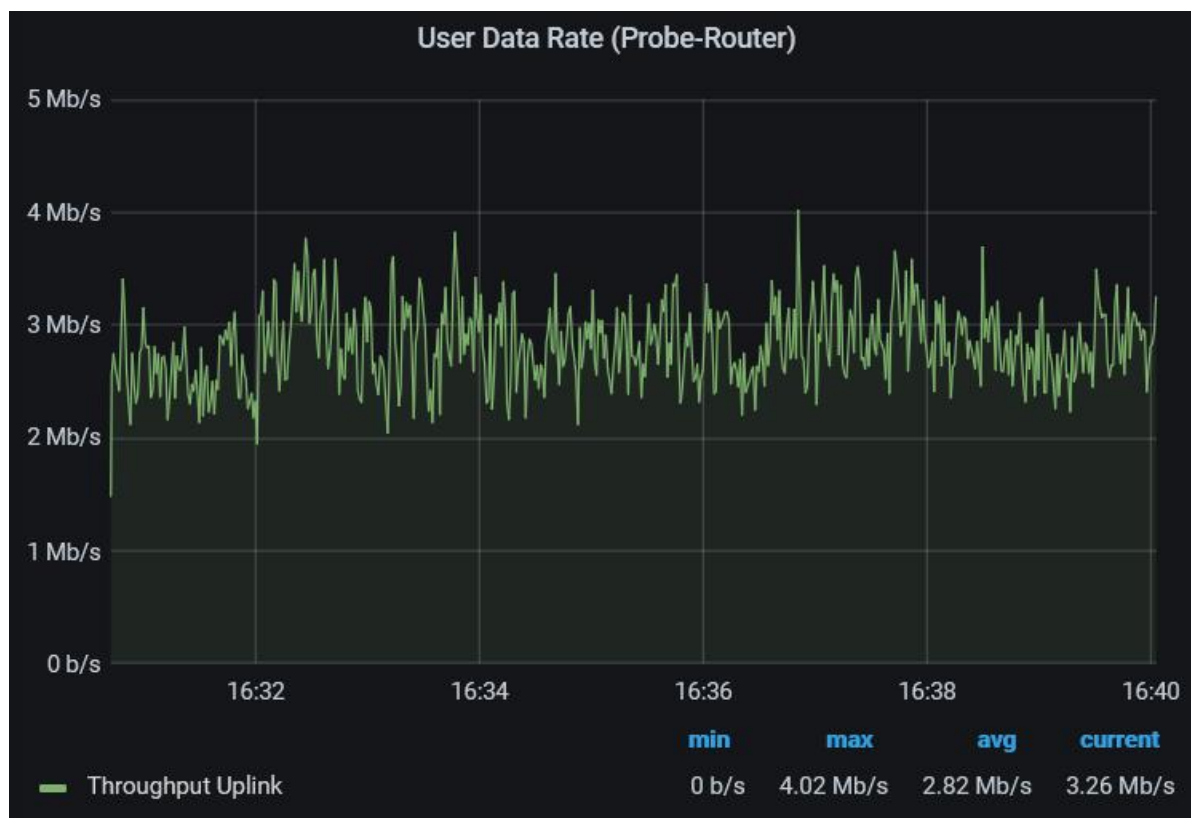


FIGURE 45: VIDEO STREAMING LATENCY, JITTER AND DATA RATE

Figure 45 shows the RTT TCP Latency of the video streaming measured, averaging 45 ms. User Data Rate of the video streaming is stable at around 3 Mbps.

The Service KPI results achieved are summarized in Table 19.

TABLE 19: INNOVALIA UC1 OBTAINED KPIS

SKPI	CKPI	CKPI result	SKPI result
P1UC1-SKPI-1: Teleworker-CMM Synchronization (=5GR-SKPI-2)	CKPI-1 End-to-end Latency	M3: 70 ms + 140 ms = 210 ms Video: 45 ms	Acceptable.
P1UC1-SKPI-2: High-resolution Real-time Video Quality (=5GR-SKPI-4)	CKPI-3 Guaranteed Data Rate	3 Mbps	4
	CKPI-9 Jitter	21 ms	
P1UC1-SKPI-3: Service Setup Time (=5GR-SKPI-1)	CKPI-6 Slice Creation Time	250 s	4.17 minutes
P1UC1-SKPI-5: Integrated Multitype Communications (=5GR-SKPI-6)	CKPI-3 Guaranteed Data Rate	10 Mbps + 3 Mbps = 13 Mbps	Success
P1UC1-SKPI-6: Extensive Network	CKPI-1 End-to-end Latency	M3: 70 ms + 140 ms = 210 ms Video: 45 ms	100 m ²

Coverage in the Factory Premises (=5GR-SKPI-7)	CKPI-3 Guaranteed Data Rate	10 Mbps	
--	-----------------------------	---------	--

4.1.2. Use Case 2: Connected worker: Augmented Zero Defect Manufacturing (ZDM) Decision Support System (DSS)

4.1.2.1. Service deployment time

This subsection presents the analysis performed with the 5Growth Log Parser Tool for ten repetitions of the automated deployment of Use Case 2. For the sake of simplicity, the comments associated to the following graphs present the observed differential aspects with respect to Use Case 1. For further details, we refer the reader to Section 4.1.1.1.

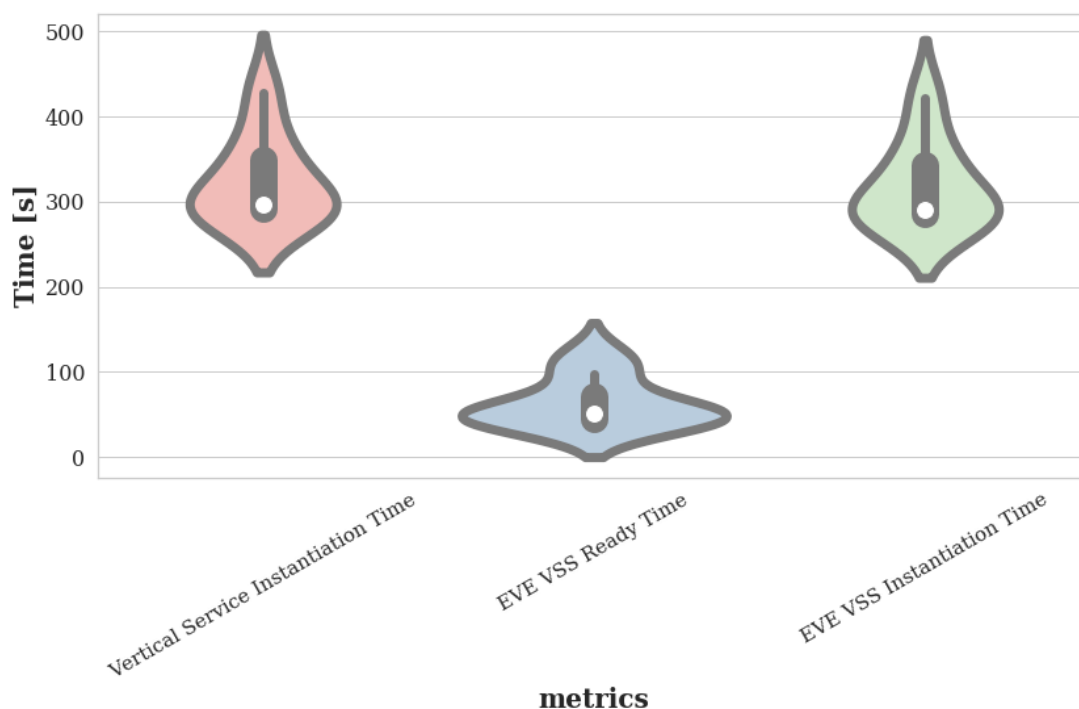


FIGURE 46: 5GR-VS INSTANTIATION METRICS UC2

In this case, the deployment time increases up to around 300s, as median value, but still in line with the fixed targets. In Use Case 2, the network service presents an additional VNF, which translates into additional time to create the associated VM and potentially additional polling loops at the 5Gr-VS and 5GEVE procedures, thus impacting in the experienced deployment time. The former can be clearly observed if comparing 5Gr-SO associated graphs, namely Figure 38 and Figure 47.

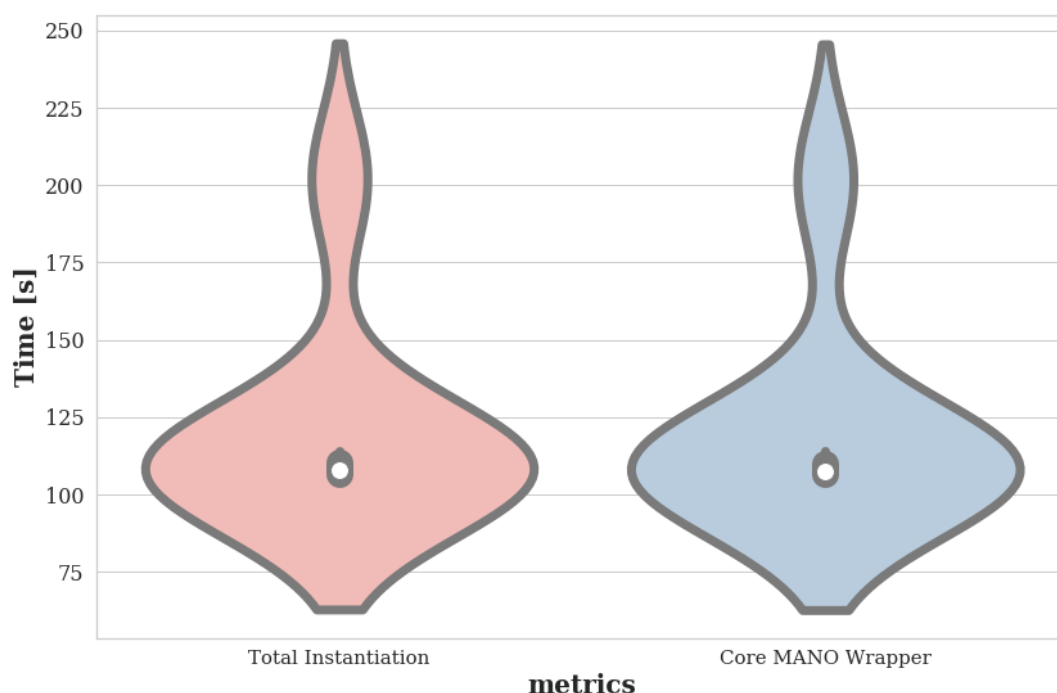


FIGURE 47: 5GR-SO INSTANTIATION METRICS UC2 – TOTAL INSTANTIATION AND CORE MANO WRAPPER TIME

Figure 47 shows that the total instantiation time experienced in 5Gr-SO for Use Case 2 jumps from 70 seconds in Use Case 1 to 105 seconds, as median value. As mentioned before, this increase in time belongs to the time required to allocate resources for the additional VNF controlling the AGV, as observed thanks to the evolution of the Core MANO Wrapper time.

The operations performed at the SOE (Figure 48) and ROE (Figure 49) modules of the 5Gr-SO follows the same trends presented in Section 4.1.1.1. They are in the order of milliseconds, having a very limited impact in the total deployment time experienced in the 5Gr-SO.



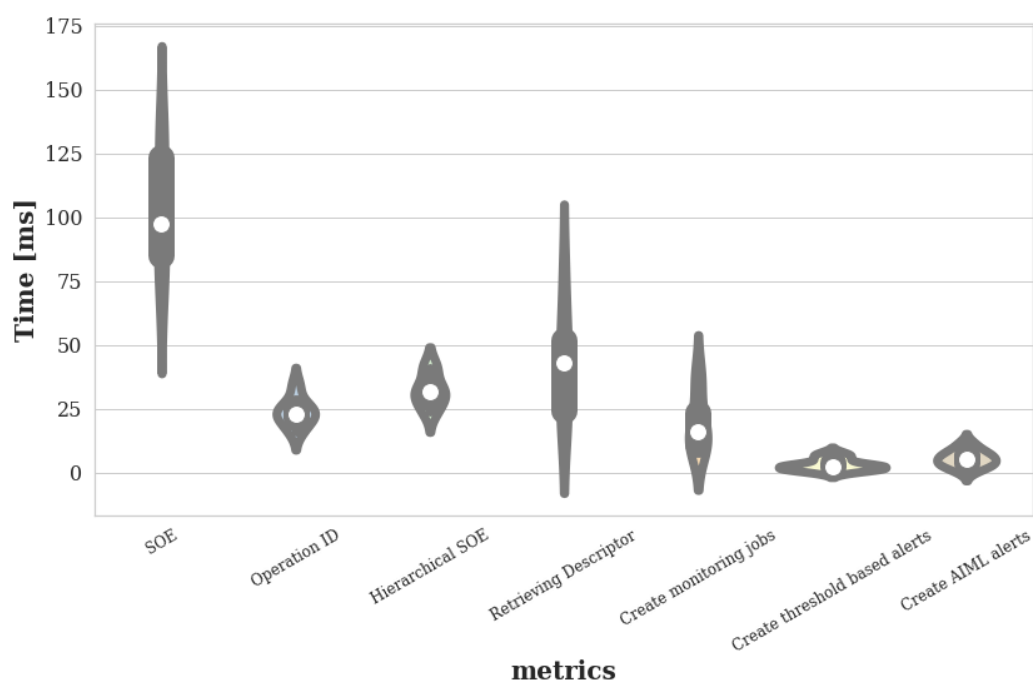


FIGURE 48: 5GR-SO UC2 TIME PROFILING OF SERVICE ORCHESTRATOR ENGINE (SOE) OPERATIONS

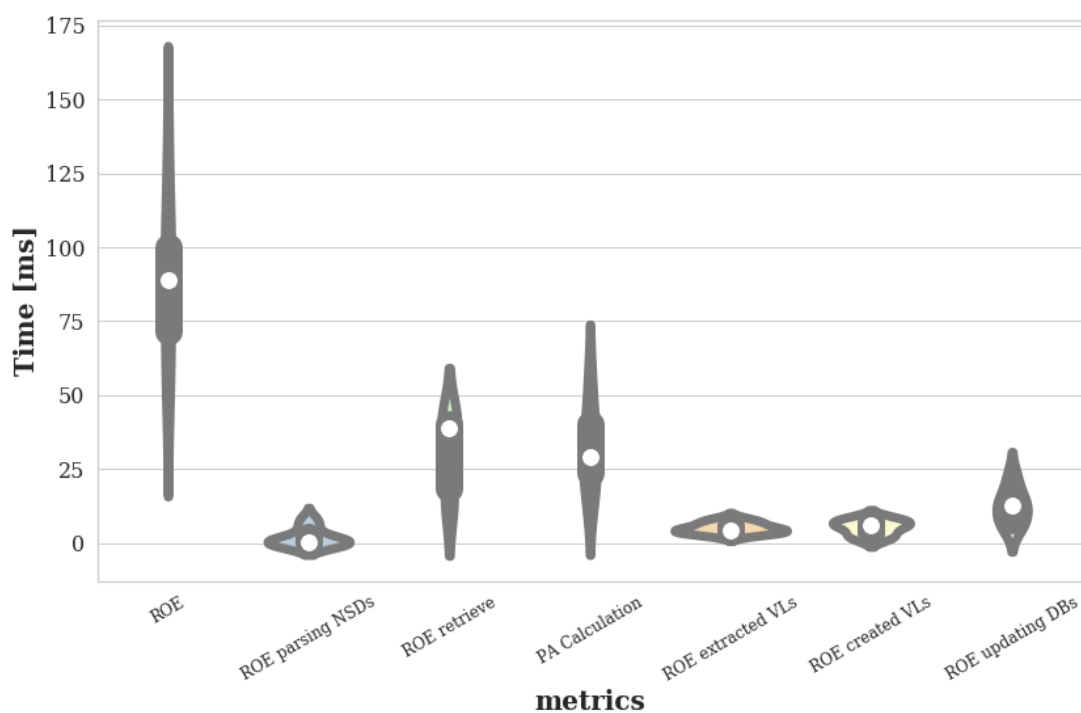


FIGURE 49: 5GR-SO UC2 TIME PROFILING OF RESOURCE ORCHESTRATOR ENGINE (ROE) OPERATIONS

4.1.2.2. Network infrastructure throughput

Previously to the AIC experimentation stage, throughput tests had been done with millimetre wave (mmW) radio at 5Tonic lab. Several tests were done with different conditions. The achieved results were: (i) with carrier aggregation of six 100 MHz carriers, 1 W per cell and standing 3 m away from the antenna: 430 Mbps DL and 40 Mbps UL; (ii) with carrier aggregation of six 100 MHz carriers, 1 W per cell and standing 1 m away from the antenna: 800 Mbps DL and 50 Mbps UL; and (iii) with carrier aggregation of six 100 MHz carriers, 5 W per cell and standing 1 m away from the antenna: 720 Mbps DL and 80 Mbps UL.

In the context of Use Case 2 at Bilbao, iperf tests were done with the CMM device end at ASF connected via millimetre wave (mmW) antennas. At AIC, the radio conditions were again aggregation of six carriers, 5W per cell and standing 5m away from the antenna. Under these conditions, and again running three parallel threads for a few minutes the result obtained for the iperf from a client on the INNO VM to a server on a PC connected close to the camera and CMM, was 13 Mbps. When running the iperf in reverse, the outcome was 27 Mbps.

4.1.2.3. Use Case 2 Operation

This use case was run over the setup that was implemented as explained in D3.6 [3] Section 2.3. The results obtained are gathered in this section.

The traffic between the AGVc VM and the AGV while the AGV is moving is UDP, and it is demanding an average data rate of 270 kbps in uplink (from the AGV towards the AGVc VM) and 100 kbps in downlink (from the AGVc VM towards the AGV).

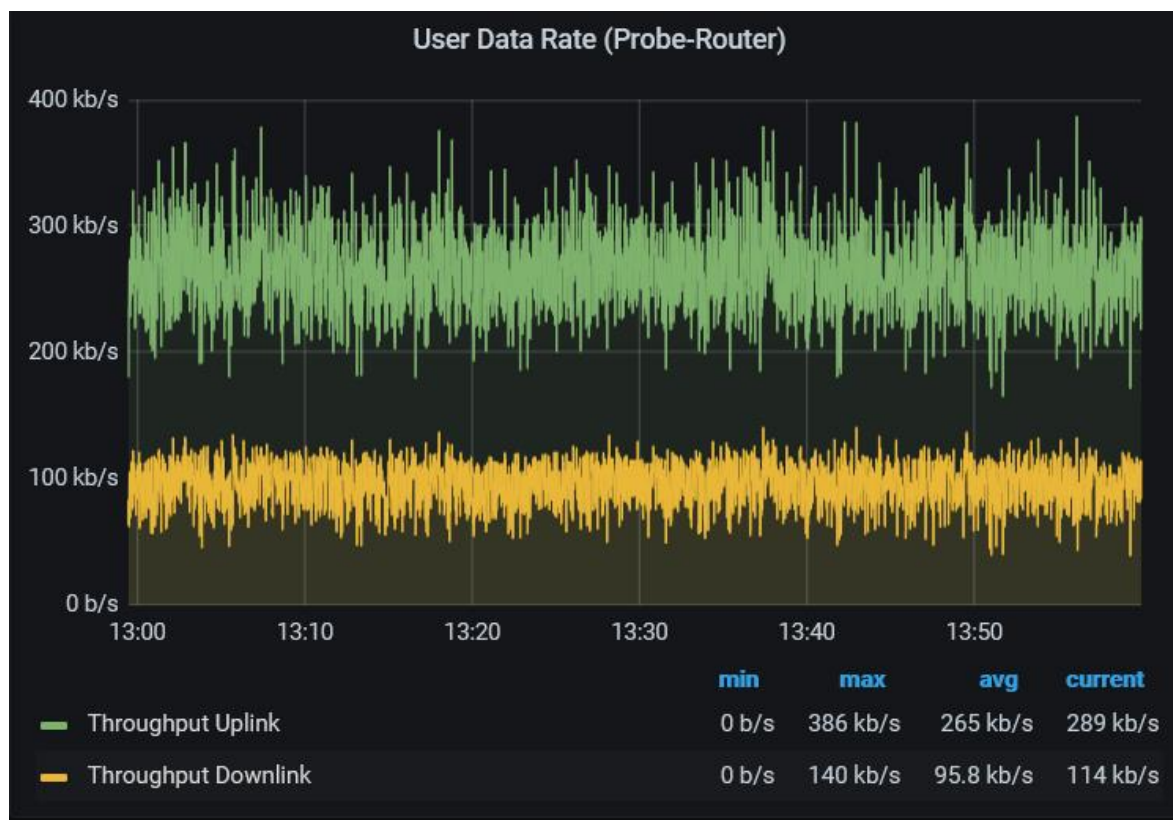


FIGURE 50: USER DATA RATE OF AGV RELATED TRAFFIC

The traffic pattern of the M3 and CMM applications during the execution of the use case using the mmW band network looks like it is shown afterwards. First, the scan results provided by the CMM produce the following figure.





FIGURE 51: UC2 RTT LATENCY AND JITTER FOR SCAN RESULT TRAFFIC

The lines in Figure 51 show that the result is an average of 130 ms RTT latency when there is traffic.

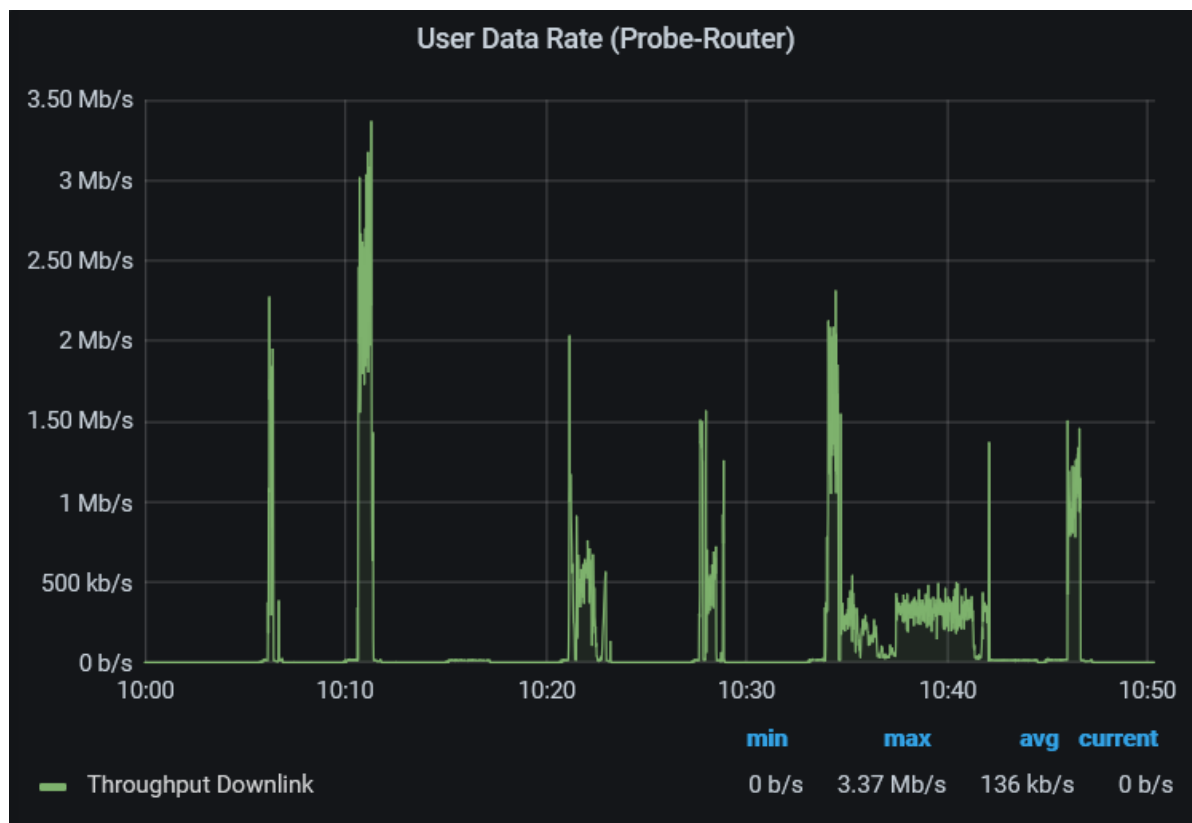


FIGURE 52: UC2 USER DATA RATE FOR SCAN RESULT TRAFFIC

Figure 52 shows that the maximum data rate demanded by the application is 3 Mbps.

Next are the metric values obtained for the commands sent towards the CMM, in the segment between the EPG and the CMM.



FIGURE 53: UC2 RTT LATENCY AND JITTER FOR TRAFFIC

The lines in Figure 53 show the outcome is an average of 65 ms latency when there is traffic.



FIGURE 54: UC2 USER DATA RATE FROM INNO VM TO CMM

The User Data Rates show that the application is demanding values of less than 60 kbps.

The summary of the obtained results is the following:

TABLE 20: INNOVALIA UC2 AGV TRAFFIC MEASUREMENTS

SKPI	CKPI	CKPI result	SKPI result
P1UC2-SKPI-1: Service Operation Time (=5GR-SKPI-8)	CKPI-1 End-to-end Latency	65 ms + 130 ms = 195 ms	5 minutes
	CKPI-3 Guaranteed Data Rate	3 Mbps	
P1UC2-SKPI-2: Service Creation Time (=5GR-SKPI-1)	CKPI-6 Slice Creation / Adaptation Time	300 s	5 minutes
P1UC2-SKPI-3: AGV-Edge Control Synchronization (=5GR-SKPI-2)	CKPI-1 End-to-end Latency	not measurable with the 5Probe for UDP traffic	Good
P1UC2-SKPI-4: Network support for user mobility (=5GR-SKPI-3)	CKPI-3 Guaranteed Data Rate	M3: 3 Mbps AGV: 300 kbps	Success
P1UC2-SKPI-5: Concurrency of	CKPI-1 End-to-end Latency	M3: 65 ms + 130 ms = 195 ms	Has not been possible to test. The

simultaneous users in a given area (=5GR-SKPI-7)		AGV: not measurable with the 5Probe for UDP traffic	infrastructure can allow multiple users at these rates though.
	CKPI-3 Guaranteed Data Rate	M3: 3 Mbps AGV: 300 kbps	

4.2. Industry 4.0 pilot – COMAU

This Section reports the results of the third and final validation campaign performed as described in Section 3.2.3 of the present deliverables.

4.2.1. Use Case 1: Digital twin apps

This use case, whose architecture is described in Section 3.2 of D3.6 [3], has a strict requirement on **latency**. This requirement comes from the need of having the digital twin (i.e., virtual representation) in perfect “visual” alignment with the real robot. In other words, the user looking at the digital twin, on a screen or in AR/VR glasses, shall have the perception of synchronization of the two robots: the real robot and its digital replica.

The requirement posed on E2E RTT latency in D1.1 [1] was 15 ms as maximum. It is useful here to clarify how this delay limit, imposed on the radio link, was estimated at the beginning of the project.

The digital twin animation is a sequence of frames constructed by a specific rendering application running on a computer. Each frame, and the resulting animation, are based on the real-time coordinates of the robot axes positions coming from the (remote) robot controller. These coordinates are transferred across the 5G link. The refresh rate between two consecutive frames is 24 Hz which corresponds to $1/24 \text{ sec} = 41.7 \text{ ms}$ between two consecutive frames.

When the digital twin is represented in VR or AR glasses, the refresh rate inside the glasses is higher than 24 Hz, to avoid the “cybersickness” effect for the user. Like the ones used in the pilot, commercial AR glasses have a refresh rate in the order 80 Hz or more (the higher the better) which corresponds to $1/80 \text{ sec} = 12.5 \text{ ms}$.

In summary, the AR application receives a new frame from the digital twin rendering application, every 41.7 ms but refreshes the image presented in the glasses to the human every 12.5 ms. The new image in the glasses can be a new one, just received from the rendering application, or the replica of the previous one (in case that a new frame is still not arrived at the end of the current 12.5 ms slot).

The time budget of 41.7 ms is composed as follows: transmission over 5G + time waiting for the start of the next processing cycle in the AR program (max 12.5 ms) + processing time of the AR program for refreshing the new internal frame (12.5 ms).

The time for transmission over 5G is then: $41.7 \text{ ms} - 12.5 \text{ ms} - 12.5 \text{ ms (max)} = 16.7 \text{ ms}$.

To account some margin, in D1.1 [1], it has been fixed as KPI a threshold of 15 ms for the transmission latency over 5G, even if with the mentioned 16.7 ms the AR experience is good.

The LTE network installed on the same premises has demonstrated, in past experiments, to ensure a latency greater than 20 ms. For this specific use case, this would lead to a quite visible misalignment between the real robot and its digital replica.

As for the bit rate in data transfer, for the specific needs of UC1, the throughput is not a challenging constraint. In fact, as visible in the UC1 scheme in D3.6 [3], the low latency transfer is limited to conveying the robot's real-time coordinated from the robot from the robot controller to the application that renders the digital twin. Consequently, the validation campaign has assessed the latency by sending 1 KB ping signals from one end to the other.

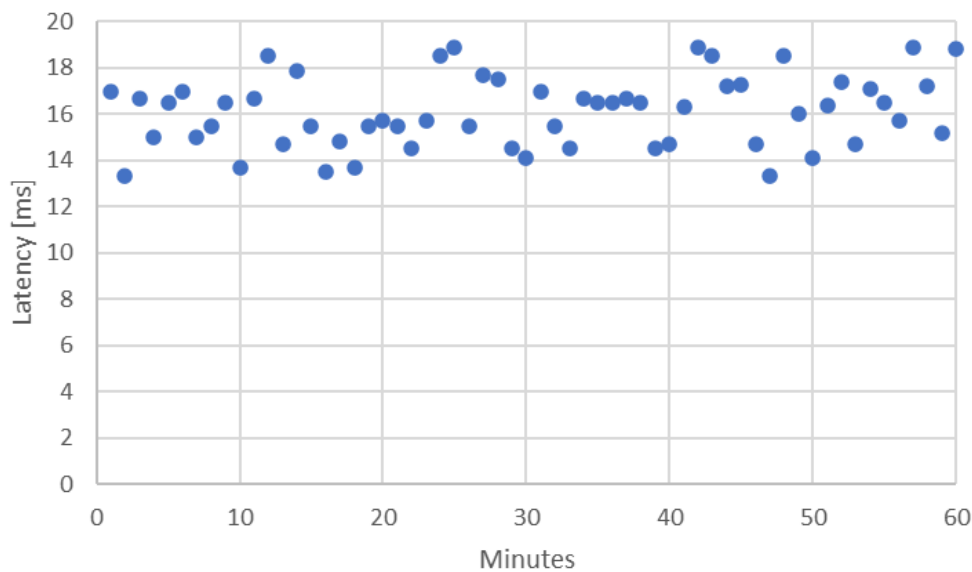


FIGURE 55: E2E LATENCY

On a time window of one hour, in which a ping has been sent every minute to measure the related latency, the validation resulted in values ranging from 13.3 ms to 18.9 ms with an average of 16.1 ms as illustrated in Figure 55. This performance, being in the order of the mentioned 16.7 ms, has practically resulted in a perfect “perceptible” alignment of the real robot with its digital twin inside the AR glasses.

In addition to the latency measurements, a “visual” verification of the use case from the user perspective has been done by posing “side-by-side” the digital reproduction of the real robot and the real robot itself to verify the perception of the real-time alignment between the virtual and the real entity as illustrated in Figure 56.



FIGURE 56: VIRTUAL ROBOT SUPERIMPOSED TO THE REAL ONE

This practical assessment demonstrates a very good synchronization and presents an interesting behavior that results in the virtual replica slightly anticipating the real robot as it, obviously, presents a mechanical inertial delay in the real robot which is not present in the virtual robot. So, to achieve an accurate alignment and compensate for such inertial delay, it is possible to add a small time offset in the movements of the virtual robot.

UC1 has been selected to perform the dynamic evaluation using the **monitoring platform** in the 5Growth stack. Prometheus exporter allows the collection of three parameters: the unidirectional link latency, the jitter, and the percentage of packet loss. Through the Grafana dashboard, it is possible to plot these parameters with time.

The automated measurement procedure, described in Section 3.2.3, has been adopted to evaluate the link latency between the CPE and the server where the application is running (unidirectional latency in uplink) on a measurement time window of three days.

Figure 57 shows the screenshot collected at the Grafana module of the 5Growth monitoring platform, where the data coming from the Prometheus exporter, installed at the PCProbe, is reported.

The figure shows how the latency evolves during the three days. The measured average value is **8.27ms** (min 5.2, max 12.3), as shown in the right side of the plot. As reported above, the third on-site measurement have reported an average latency value of 16.1ms in round trip (DL+UL).

In the measurements done with the monitoring platform, where only one direction is considered and the virtualization environment of the application VM is in place, the obtained average result is about half (i.e., 8.73 ms), showing a good match between the two considered measurement procedures (i.e., on site vs remote). No packets have been lost during the tests. At the same time all the jitter values have been registered in the range -6.8ms and 6.4 ms, with an average value of 0ms.



FIGURE 57: GRAFANA SCREENSHOT, AT THE MONITORING PLATFORM

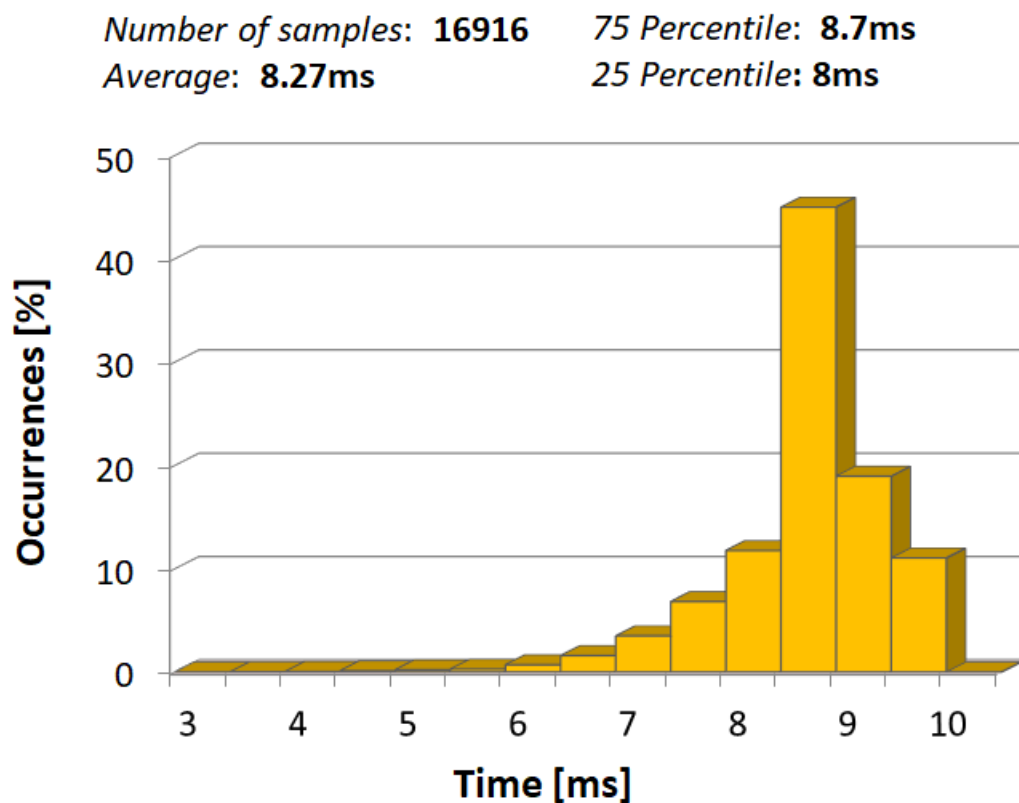


FIGURE 58: DISTRIBUTION OF LATENCY VALUES

4.2.2. Use Case 2: Telemetry/monitoring apps

UC2 is associated with an mMTC traffic profile. The key performance indicator for this UC is then the number of devices that can be connected simultaneously to the network infrastructure without degrading the performance already connected devices. An in-field verification is not possible because a “massive” density of devices is not present in the pilot due to the dimension of the experimental setup (three robots and a conveyor as explained in D3.6 [3]).

However, it is sensible to refer to the value specified for the Ericsson commercial radio products used in the area when the mMTC profile is used. This value is estimated in 5 devices for square meter, on average, in terms of radio support. The underlying transport network is dimensioned not to limit the radio performances so the mentioned number of 5 devices is ensured in the integrated environment.

4.2.3. Use Case 3: Digital tutorial and remote support

UC3 is associated with eMBB performances as the video streaming related to the remote support use case requires the transmission of real-time video. As reported in D4.3 [6], the second validation campaign has already verified the network performances in terms of throughput and packet loss on the integrated (radio, transport, platform) infrastructure.

More specifically, the second validation campaign has measured a TCP average throughput of 789 Mbit/s in DL and 43 Mbit/s in UL. The UDP average throughput has resulted in 855 Mbit/s in DL and 42 Mbit/s in UL. The related graphs are reported in D4.3 [6], Figures 35-38.

After the second validation campaign, as detailed in Section 3.2.3, the network has been tuned for latency optimization while the throughput settings have not been modified.

However, new throughput measurements have been repeated with the following results: TCP 761 Mbit/s DL, 65 Mbit/s UL; UDP 952 Mbit/s DL, 66 Mbit/s UL. Both the DL values are well above the target of 500 Mbit/s identified for UC3.

The packet loss has been measured as lower than 2×10^{-6} achieving the same performance of the second validation campaign.

4.3. Transportation pilot – EFACEC_S

This Section reports the results of the third and final validation campaign performed as described in Section 3.3.3 of the present deliverable. For each Use Case, a diagram of the deployed testbed is updated, and the achieved measurements are shown by tables, histograms, and graphs. This section also reports the results obtained when functional validation tests were performed in the final site premises environment (SAT – Site Acceptance Test). Finally, a measure campaign was carried out in the same vertical site, using the same vertical equipment, and using the same infrastructure but with a different 5G network, with the obtained results are being here reported.

4.3.1. Use Case 1: Safety critical communications

This final validation campaign was performed in the real vertical premises and , this final pilot environment was used for collecting measurements and validation results.

The architecture was described in D3.6 [3], Section 5.2

The results obtained for the UC1 are summarized in the following figures and tables.

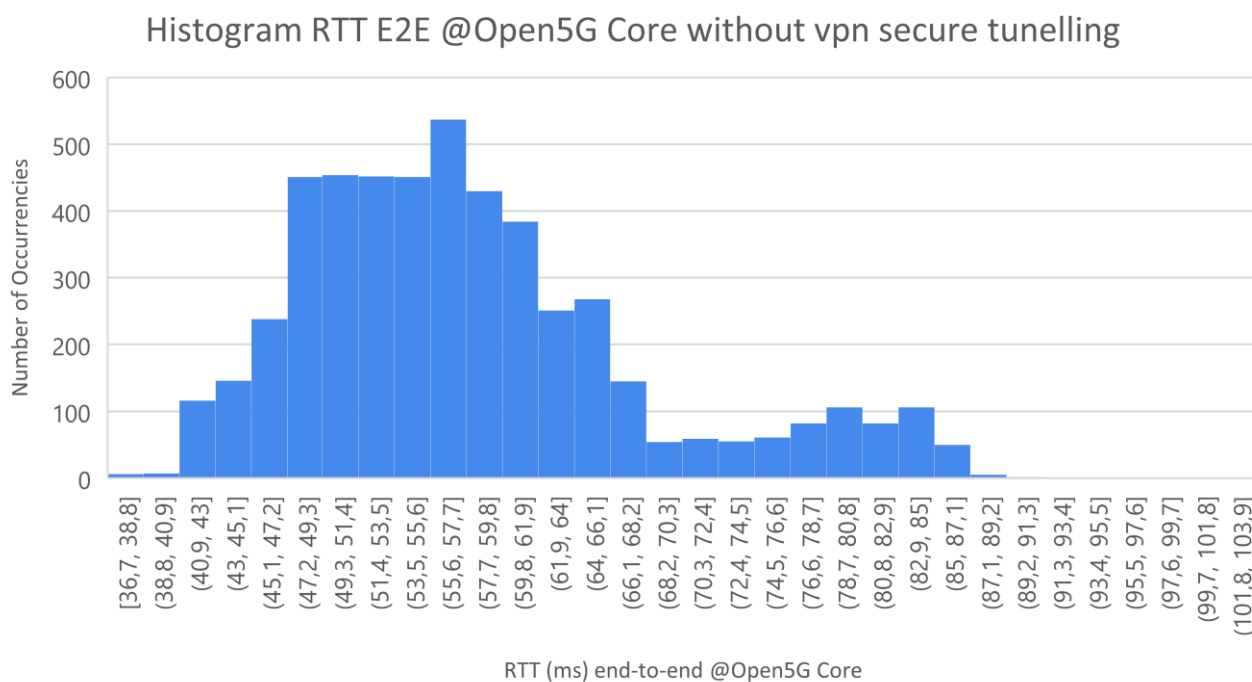


FIGURE 59: HISTOGRAM OF RTT

The following table summarizes the collected results regarding E2E RTT. These measurements were collected in real vertical premises representing the theoretical worst case (level crossing located 650 meters from the 5G Antenna and the wheel sensor #2, located 840 meters from the 5G Antenna). It was also collected, measurements using VPN secure tunnels and with no secure procedures enabled. It was detected no impact in the latency values.

TABLE 21: RTT END-TO-END IN EFACEC_S UC1

Meas. Nbr.	RTT min (ms)	RTT avg (ms)	RTT max (ms)	Jitter (ms)
1	36.687	58.036	103.153	10.193

Therefore, the E2E latency for the UC1 is around 29 ms (average).

Regarding the performance tests, also throughput values were collected, and next tables shows the measurements on the UDP bitrates.

TABLE 22: THROUGHPUTS MEASURED FOR UDP PROTOCOL IN EFACEC_S UC1

Test	Bandwidth	Jitter (ms)	Lost/Total Datagrams	%failure
UDP-250Kb/s	250 Kb/s	6.305	0/697	0%
UDP-500Kb/s	500 Kb/s	8.522	0/1392	0%
UDP-1Mb/s	1,00 Mb/s	9.834	0/2787	0%
UDP-2Mb/s	1,93 Mb/s	29.139	0/1853	0%
UDP-4Mb/s	3,95 Mb/s	4.443	0/3692	0%
UDP-8Mb/s	7,80 Mb/s	3.132	0/7417	0,0%
UDP-16Mb/s	15,6 Mb/s	1.491	0/14832	0,0%
UDP-20Mb/s	19,9 Mb/s	1.113	0/18539	0,0%
UDP-32Mb/s	21,2 Mb/s	0.842	2/21250	0,0094%

For the throughput analysis with the UDP protocol, the results are around 20 Mbps with no significant losses and with a jitter value of 0,842 ms.

When comparing the performance results from the previous campaign (at IT/Altice Labs) it is possible to verify some degradation related to E2E latency and some improvements related to UDP throughput and packet loss. As explain in previous deliverables, the data messages (protocolar messages) related to this Use Case requires low bandwidths, but it requires low latency and ultra-reliable communications.

TABLE 23: EFACEC_S UC1 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	CKPI result	SKPI result	Validation methodology
P3UC1-SKPI-1: Sync between LX detectors and Controller (=5GR-SKPI-2)	CKPI-1 End-to-end Latency	29 ms (average)	Acceptable	Expressed as Bad/Acceptable/Good. This will be mapped with the measured Core KPIs to establish the range of values that imply a Bad, Acceptable or Good synchronization.
	CKPI-2 Packet Loss	0% (related to required bandwidth)	Good	
P3UC1-SKPI-2: Communication Availability between Lx Detectors and Lx Controller (=5GR-SKPI-10)	CKPI-5 Availability	--	Bad (mainly because the functional behaviour of the level crossing is poor motivating	Ability of a product / equipment / system to be in state to perform a required function under given conditions and environment at a given instant of time or over a given interval assuming that the required external resource is provided; In fact, the goal is to measure the time the system will be unavailable. Availability = 1-Unavailability

			false positives related to the train detection)	
--	--	--	---	--

In the scope of the validation campaigns the performance measurements related to latency, jitter, packet loss. Similar graphs have been collected, reflecting the integration monitoring methodology (first probe integration), and have been registered in the 5G monitoring platform. Also, throughput (second probe integration) has been collected, registered and integrated in the 5G monitoring platform.

4.3.1.1. SAT

The Site Acceptance Tests were performed with the real level crossing scenario, with all software and hardware components enabled, interconnected, and running. The major goal of this tests was to realize if communication between the wheel sensors and the level crossing controller are properly established, if the safety and security protocols are working according with the requirements and if the signalling operations are working properly when a train is approaching the level crossing, when the train is passing the level crossing, when the train is leaving the level crossing and when no trains are in the level crossing area. As reported in previous deliverables this scenario is related to safety critical communications the availability of the system is one the major KPI of the Use Case 1 meaning that the level crossing, for instance must be kept in safety conditions when a failure is detected in the communications. i.e., if a communication failure is detected the level crossing starts the safety procedures and the road signals turn to red, and the bells start ringing indicating, this way, that the cars cannot drive over the level crossing (railway announcement). This a safety procedure according the railways standards.

4.3.1.2. SAT Results

During the SAT, it was detected that the network was not stable, and some communication failures were detected causing the level crossing to activate the safety conditions even when the train was not approaching. This is under investigation, but the coverage conditions will be improved, and the network latency will also be improved to improve the overall availability and to reduce oscillations in latency values.

Note: After collecting the final measurements some field activities were initiated and are still on going to improve the results, together with all Pilot partners and suppliers. It was also detected that in the harbour area some obstacles (equipment to be shipped) have been stored and they interfere with the previous coverage conditions. This had a huge impact in the vertical Pilot and further procedures are ongoing to solve this issue.

4.3.1.3. Vertical Pilot results using the 5GAINER 5G network

During these trials, another 5G network became available in the same geographical area of the pilot. This network, entitled *Portugal 5G+IA Networks Reliability Center* (5GAINer), was able to provide connectivity service to the existing assets that EFACEC already had deployed for the trials in 5Growth, but instead using a commercial-graded 5G hardware solution. This was an interesting opportunity that, despite not being planned, would be able to provide insightful results for the project, and this pilot in particular. After contacting the responsables of 5GAINer, we were provided with SIM cards granting our devices with connectivity to this network, allowing us to execute the same set of tests belonging to this validation campaign. The next subsections thus provide an addition to the planned work, by showcasing results using the 5GAINer network.

The results obtained for the UC1 are summarized in the following figures and tables.

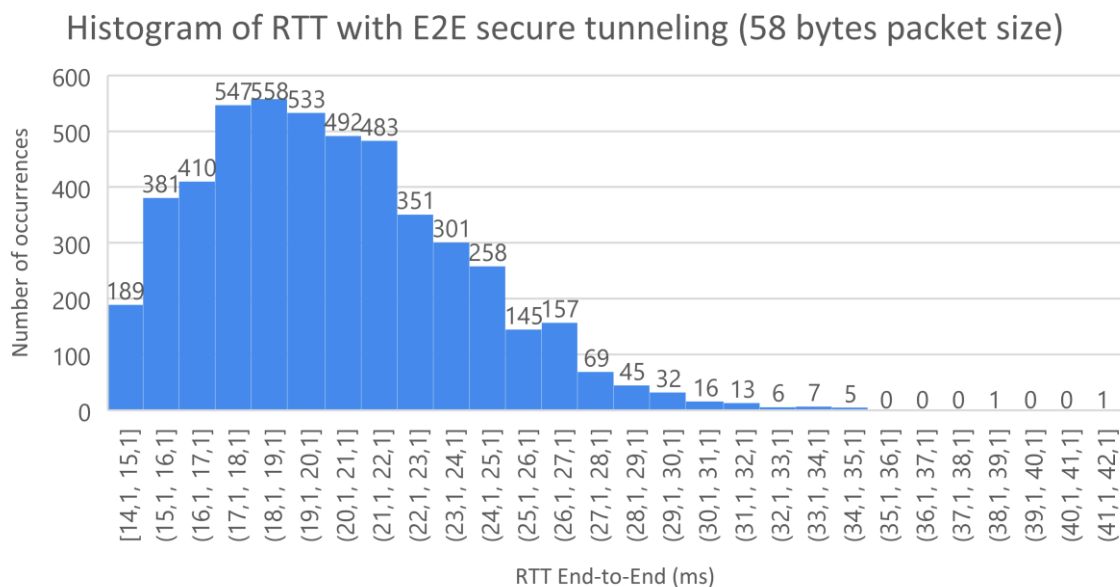


FIGURE 60: HISTOGRAM OF RTT

The following table summarizes the collected results regarding E2E RTT. These measurements were collected in real vertical premises representing the theoretical worst case (level crossing located 650 meters from the 5G Antenna and the wheel sensor #2, located 840 meters from the 5G Antenna). It was also collected, measurements using VPN secure tunnels and with no secure procedures enabled. It was detected no impact in the latency values.

TABLE 24: RTT END-TO-END IN EFACEC_S UC1 (WITH E2E SECURE TUNNELING)

Meas. Nbr.	RTT min (ms)	RTT avg (ms)	RTT max (ms)	Jitter (ms)
1	14,1	16,51	41.9	3,57

Therefore, the E2E latency for the UC1 is around 8,25 ms (average).

Regarding the performance tests, also throughput values were collected, and next tables show the measurements on the UDP bitrates.

TABLE 25: THROUGHPUTS MEASURED FOR UDP PROTOCOL IN EFACEC_S UC1

Protocol	Bitrate (Mbps)	Jitter (ms)	Lost/Total Datagrams	% failure
UDP	92.5	0.052	5 / 76280	0.0066

For the throughput analysis with the UDP protocol, the results indicate maximum values around 92 Mbps with no significant losses and 90 Mbps with no losses and with a jitter value of 0,052 ms.

These results led to the following KPIs (Table 20 herein):

TABLE 26: EFACEC_S UC1 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	CKPI result	SKPI result	Validation methodology
P3UC1-SKPI-1: Sync between LX detectors and Controller (=5GR-SKPI-2)	CKPI-1 End-to-end Latency	8,25 ms (average)	Good	Expressed as Bad/Acceptable/Good. This will be mapped with the measured Core KPIs to establish the range of values that imply a Bad, Acceptable or Good synchronization.
	CKPI-2 Packet Loss	0% (related to required bandwidth)	Good	
P3UC1-SKPI-2: Communication Availability between Lx Detectors and Lx Controller (=5GR-SKPI-10)	CKPI-5 Availability	No failures detected.	Good/always available during the validation campaigns	Ability of a product / equipment / system to be in state to perform a required function under given conditions and environment at a given instant of time or over a given interval assuming that the required external resource is provided; In fact, the goal is to measure the time the system will be unavailable. Availability = 1- Unavailability

In the scope of the validation campaigns the performance measurements related to latency, jitter, packet loss Similar graphs have been collected, reflecting the integration monitoring methodology (first probe integration), and have been registered in the 5G monitoring platform. Also, throughput (second probe integration) has been collected, registered, and integrated.

The following graph shows the latency variation along the time for the UC1 solution using the 5G monitoring platform, for scales of 5 minutes. According to the collected measures it be can be verified that similar results are obtained (using the probe approach methodology) when compared with the previous ones, related to minimum, average and maximum latency values.

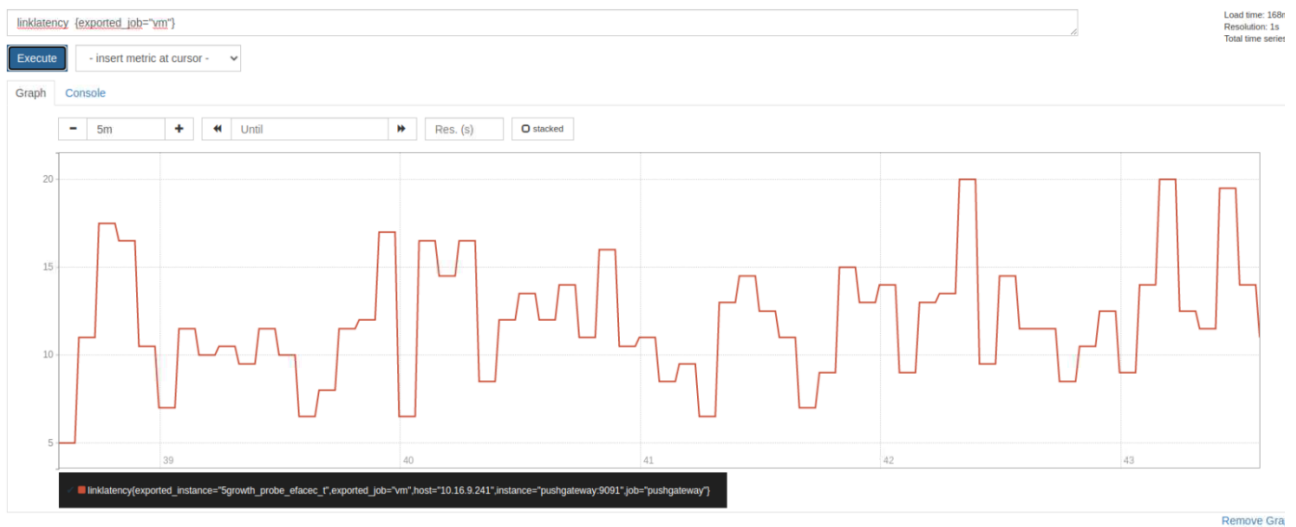


FIGURE 61: LATENCY GRAPH

Similar graphs have been collected and registered in the 5G monitoring platform, regarding jitter, packet loss (first probe integration) and throughput (second probe integration).

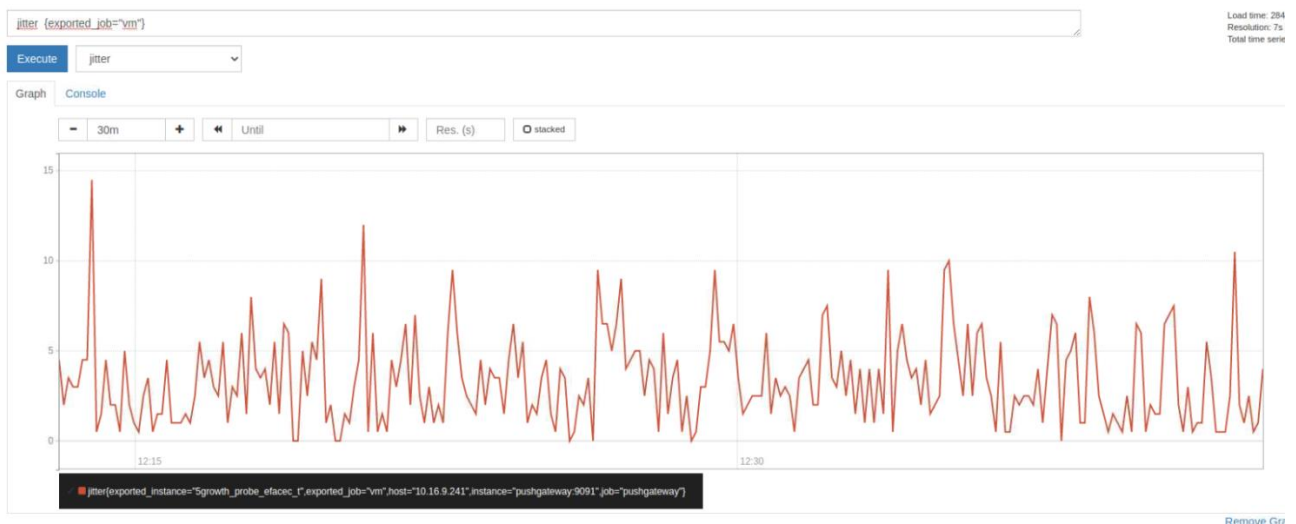


FIGURE 62: JITTER GRAPH

4.3.1.4. SAT Results

During the SAT, using the 5GAINER 5G network, it was achieved good performance values (latency, packet loss, bandwidth, and availability) and it was possible to validate the UC1 functionalities in the real vertical premises, using train simulators or even real freight trains to detect the proper behaviour of the level crossing, supported by 5G technology.

4.3.2. Use Case 2: Non-safety critical communications

This final validation campaign was performed in the real vertical premises and this final pilot environment was used for collecting measurements and validation results.

The architecture was described in D3.6 [3], Section 5.3.

The results obtained for the UC2 are summarized in the following figures and tables

RTT Histogram in 5G SA altice network open5gcore static test over wireguard tunnel

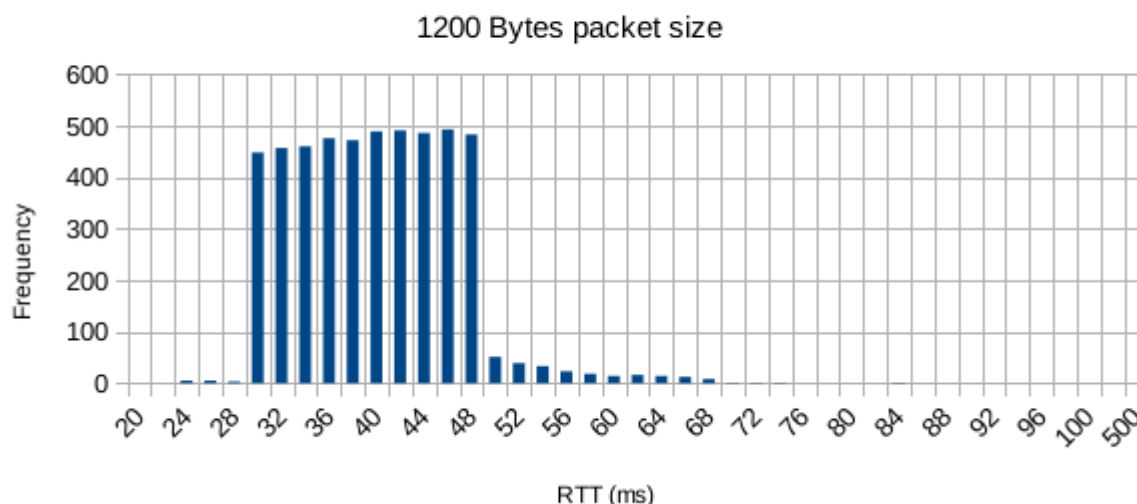


FIGURE 63: HISTOGRAM OF RTT FOR 1200 BYTES PACKET SIZE – STATIC TEST

The RTT and jitter values were measured end-to-end between the Syslogic PC console (train driver console) and the Raspberry pi at the other end, passing through the secure wireguard tunnel. Considering that in the total route, the measured RTT passes through two Up-links and two Down-links, given its symmetry, we can state without doubt that in this case the end-to-end latency is exactly half of the measured RTT. As it is mentioned in the figure, these measurements were collected in static conditions, from the camera located at the level crossing (600 meters from the 5G antenna) to the train driver console (900 meters from level crossing).

The RTT average in this scenario is 39,08 ms and minimum values of 23,631ms was collected. The average latency value for this test scenario = 19,54 ms and minimum values of 12 ms was collected.

Next table shows the bandwidth static test with iperf3 for UDP data traffic.

TABLE 27: BANDWIDTH FOR UDP TRAFFIC PROTOCOL IN EFACEC_S UC2

Test	Bandwidth	Jitter (ms)	Lost/Total Datagrams	%failures
UDP-250Kb/s	250 Kb/s	5.509	0/1634	0%
UDP-500Kb/s	500 Kb/s	5.694	0/3268	0%
UDP-1Mb/s	1,00 Mb/s	8.902	0/6536	0%
UDP-2Mb/s	2,00 Mb/s	7.127	0/13070	0%
UDP-4Mb/s	4,00 Mb/s	4.826	3/26142	0,011%
UDP-8Mb/s	8,00 Mb/s	3,314	66/52267	0,13 %
UDP-16Mb/s	16,0 Mb/s	1,321	29/104597	0,0028%
UDP-20Mb/s	20,0 Mb/s	1,142	34/130758	0,026%

UDP-32Mb/s	32,0 Mb/s	0.852	463/209174	0,22%
UDP-64Mb/s	64 Mb/s	0,566	184257/417031	44%

The performance of the 5G network, in terms of UDP traffic is usable for values around 20 Mbps with 0,026% of lost datagrams within the coverage area of the level crossing allowing the trigger conditions to activate the video transmission to the train. The maximum UDP throughput is around 32 Mbps but the lost datagram is high (around 0,22 %). However, it was detected in several validation campaigns that packets/datagrams were received out of order independently of the bandwidth being tested. It should be noted that the limiting factor of data throughput is the radio uplink. Upstream and downstream throughputs were measured separately, and the results were in the order of 65 Mb/s and 500 Mb/s, respectively. In any case, a more symmetrical result can be obtained through RAN configuration and will be tested in the future.

To achieve the behavior of the system in function with the movement of the train, the previous test was repeated but with a dynamical approach, i.e with the car/train in movement along the railway track in this level crossing area.

Next figure shows the histogram related to the RTT for dynamic tests.

RTT Histogram in 5G SA altice network open5gcore dynamic test over wi-reguard tunnel

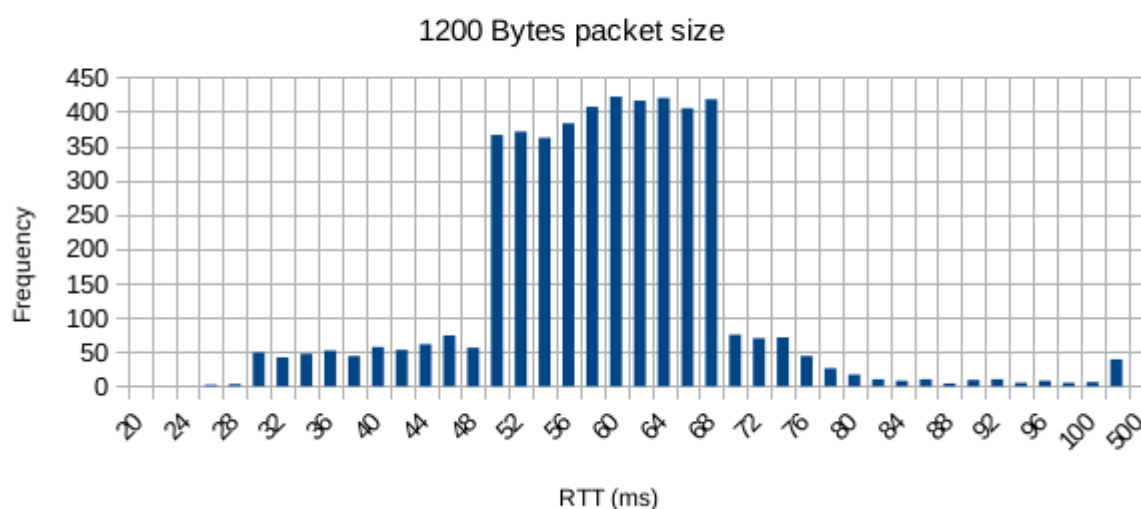


TABLE 28: BANDWIDTH FOR UDP TRAFFIC PROTOCOL IN EFACEC_S UC2

Test	Bandwidth	Jitter (ms)	Lost/Total Datagrams	%failures
UDP-2Mb/s	2,00 Mb/s	9.393	6/13066	0,046%
UDP-4Mb/s	4,00 Mb/s	4.160	3/26136	0,011%
UDP-8Mb/s	8,00 Mb/s	3,314	36/52279	0,069 %
UDP-16Mb/s	16,0 Mb/s	1,286	20/104584	0,019%
UDP-20Mb/s	20,0 Mb/s	1,142	12/130889	0,0092%
UDP-32Mb/s	32,0 Mb/s	0.613	135/209090	0,0065%
UDP-64Mb/s	64 Mb/s	0,529	184406/416695	44%

The performance of the 5G network, in terms of UDP traffic is usable for values around 32 Mbps with 0,0065% of lost datagrams within the coverage area of the level crossing allowing the trigger conditions to activate the video transmission to the train. However, it was detected in several validation campaigns that packets/datagrams were received out of order, independently of the bandwidth being tested.

TABLE 29: EFACEC_S UC2 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	CKPI result	SKPI result	Validation methodology
P3UC2-SKPI-1: High-resolution Real-time Video Quality (=5GR-SKPI-4)	CKPI-2 Packet Loss	0,065%	Fair. The video transmission has intermittent failures	Expressed in values from 1 to 5. <i>Mean Opinion Score (MOS)</i> is a well-known measure of video quality (5-Excellent, 4-Good, 3-Fair, 2-Poor, 1-Bad).
	CKPI-3 Guaranteed Data Rate	32 Mps		
	CKPI-9 Jitter	0,613 ms		
P3UC1-SKPI-2: Real Time sensors (video camera to onboard train console) monitoring latency (=5GR-SKPI-2)	CKPI-1 e2e latency	29,05 ms	Good (dynamic test)	Expressed as Bad/Acceptable/Good. This will be mapped with the measured Core KPIs to establish the range of values that imply a Bad, Acceptable or Good synchronization.
	CKPI-2 packet Loss	0%		
P3UC3-SKPI-3: Real Time Sensors	CKPI-5 Availability	--	Bad (mainly because the functional	Ability of a product / equipment / system to be in state to perform a required function

Monitoring Communication Availability (=5GR-SKPI-10)		behaviour of the video transmission crossing is poor)	under given conditions and environment at a given instant of time or over a given interval assuming that the required external resource is provided; In fact, the goal is to measure the time the system will be unavailable. Availability = 1- Unavailability.
---	--	---	---

As explained in the last deliverable, the train onboard console contains a GUI that allows to observe the real time performance values. Next figure shows an image of the GUI representing in real time the latency (15,316 ms), jitter (2.366 ms) and data rate (555,875 kbps) when the HD video stream is transmitting over the 5G network when the train is at a distance equal to 689 meters to the level crossing area



FIGURE 65: GUI OF THE TRAIN DRIVER CONSOLE

When comparing the measurements results with the previous campaign it is possible to observe some improvements in the performance values, but it also detected the impact of moving from the lab to the field as well as the impact of the mobility issues.

4.3.2.1. SAT

The Site Acceptance Tests were performed with the real level crossing scenario, with all software and hardware components enabled, interconnected, and running. The major goal of this tests was to realize if the video transmission from the level crossing camera to the train or car occurs in good conditions. As reported in previous deliverables this scenario is related to non-safety critical communications and the availability of the system as well the guaranteed bandwidth, are the major KPI of the Use Case 2, meaning good video communication even with dynamic conditions (mobility).

4.3.2.2. SAT Results

During the SAT, it was detected that the network was not stable, and some communication failures were detected causing the video transmission to be interrupted or being transmitted with poor quality. This is under investigation, but the coverage conditions will be improved, and the analysis of the datagrams being received out of order is ongoing in order to improve the overall availability and to reduce oscillations in video transmissions.

4.3.2.3. Vertical Pilot results using the 5GAINER 5G network

Once that during 5G network became available in the same geographical area of the pilot it was very interesting, to execute the same set of tests belonging to this validation campaign. The next subsections thus provide an addition to the planned work, by showcasing results using the 5GAINer network.

RTT Histogram static @1830m over wireguard tunnel

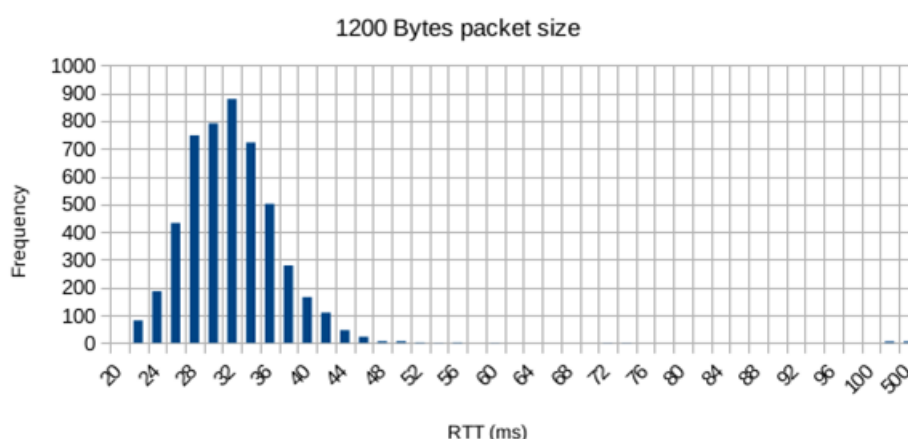


FIGURE 66: HISTOGRAM OF RTT FOR 1200 BYTES PACKET SIZE – STATIC TEST

As it is mentioned in the figure, these measurements were collected in static conditions, from the camera located at the level crossing (687 meters from the 5G antenna) to the train driver console (1143 meters from level crossing).

The RTT average in this scenario is 30,56 ms and minimum values of 22 ms was collected. The average latency value for this test scenario = 15,28 ms and minimum values of 11 ms was collected.

Next table shows the bandwidth static test with iperf3 for UDP data traffic.

TABLE 30: BANDWIDTH FOR UDP TRAFFIC PROTOCOL IN EFACEC_S UC2

Test	Bandwidth	Jitter (ms)	Lost/Total Datagrams	%failures
UDP-2Mb/s	2,00 Mb/s	1.062	0/13072	0%
UDP-4Mb/s	4,00 Mb/s	2,439	0/26139	0%
UDP-8Mb/s	8,00 Mb/s	3,314	0/52277	0%
UDP-16Mb/s	16,0 Mb/s	1,321	0/104556	0%
UDP-32Mb/s	32,0 Mb/s	0.357	64/209142	0,031%
UDP-64Mb/s	64 Mb/s	0,183	206/418374	0,049%
UDP-80Mb/s	75,3 Mb/s	0,136	30323/522048	5,8%

The performance of the 5G network, in terms of UDP traffic is usable for values around 64 Mbps with 0,049% of lost datagrams within the coverage area of the level crossing allowing the trigger conditions to activate the video transmission to the train . The maximum UDP throughput is around 80 Mbps but the lost datagram is high (around 5,8 %).

Next figure shows the histogram related to the RTT for dynamic tests using a wireguard secure tunnel.

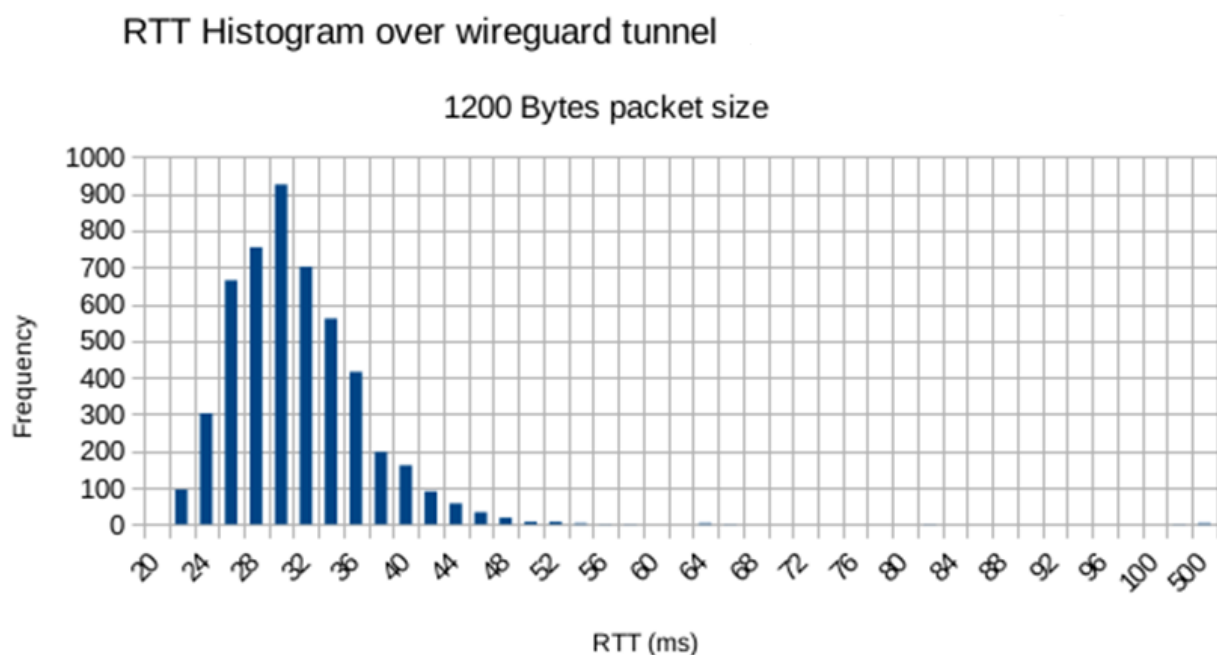


FIGURE 67: HISTOGRAM OF RTT FOR 1200 BYTES PACKET SIZE – DYNAMIC TEST

As it is mentioned in the figure, these measurements were collected in dynamic conditions, from the camera located at the level crossing (687 meters from the 5G antenna) to the train driver console moving along the railway track.

The RTT average in this scenario is 30,3 ms and minimum values of 22 ms was collected. The average latency value for this test scenario = 15,15 ms and minimum values of 11 ms was collected. This result indicates that the latency is not affected by the movement/speed of the car/train.

Next table shows the bandwidth dynamic test with iperf3 for UDP data traffic.

TABLE 31: BANDWIDTH FOR UDP TRAFFIC PROTOCOL IN EFACEC_S UC2

Test	Bandwidth	Jitter (ms)	Lost/Total Datagrams	%failures
UDP-2Mb/s	2,00 Mb/s	2.439	0/13067	0%
UDP-4Mb/s	4,00 Mb/s	2.088	0/26138	0%
UDP-8Mb/s	8,00 Mb/s	1.401	0/52276	0%
UDP-16Mb/s	16,0 Mb/s	0.837	20/104578	0%
UDP-32Mb/s	32,0 Mb/s	0.359	46/209177	0,0022%
UDP-64Mb/s	64 Mb/s	0,165	191/418387	0,046%
UDP-80Mb/s	80 Mb/s	0,171	320/522975	0,061%

The performance of the 5G network, in terms of UDP traffic is usable for values around 80 Mbps with 0,061% of lost datagrams within the coverage area of the level crossing allowing the trigger conditions to activate the video transmission to the train.

TABLE 32: EFACEC_S UC2 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	CKPI result	SKPI result	Validation methodology
P3UC2-SKPI-1: High-resolution Real-time Video Quality (=5GR-SKPI-4)	CKPI-2 Packet Loss	0,061%	Excellent	Expressed in values from 1 to 5. <i>Mean Opinion Score (MOS)</i> is a well-known measure of video quality (5-Excellent, 4-Good, 3-Fair, 2-Poor, 1-Bad).
	CKPI-3 Guaranteed Data Rate	80 Mps		
	CKPI-9 Jitter	0,171 ms		
P3UC1-SKPI-2: Real Time sensors (video camera to onboard train console) monitoring latency (=5GR-SKPI-2)	CKPI-1 e2e latency	15,15 ms	Good (dynamic test)	Expressed as Bad/Acceptable/Good. This will be mapped with the measured Core KPIs to establish the range of values that imply a Bad, Acceptable or Good synchronization.
	CKPI-2 packet Loss	0%		
P3UC3-SKPI-3: Real Time	CKPI-5 Availability	100%	Good (always	Ability of a product / equipment / system to be in state to

Sensors Monitoring Communication Availability (=5GR-SKPI-10)		available in the level crossing area)	perform a required function under given conditions and environment at a given instant of time or over a given interval assuming that the required external resource is provided; In fact, the goal is to measure the time the system will be unavailable. Availability (%) = 1- Unavailability (%)
--	--	---	---

4.3.2.4. SAT Results

During the SAT, using the 5GAINER 5G network, it was achieved good performance values (latency, packet loss, bandwidth, and availability) and it was possible to validate the UC2 functionalities in the real vertical premises, using train simulators or even real freight trains to achieve the proper behaviour of the system supported by 5G technology allowing, this way, to transmit the HD video images with a good quality.

4.4. Energy pilot – EFACEC_E

This section reports the results of the third and final validation campaign performed as described in Section 3.4 of the present deliverable. In the two following sections, for each of the two use cases, a diagram of the deployed testbed is updated, and the achieved measurements are shown in tables, histograms, and graphs.

4.4.1. Use Case 1: Advanced Monitoring and Maintenance Support for Secondary Substation MV/LV Distribution Substation

For the third validation campaign the use case setup is functionally completed and fully deployed in the vertical site (IT datacenter and UA secondary distribution substation), as described in D3.6 [3], section 4.2.

The probes' location is depicted in figure 10 in D36, section 4.1.3.2.



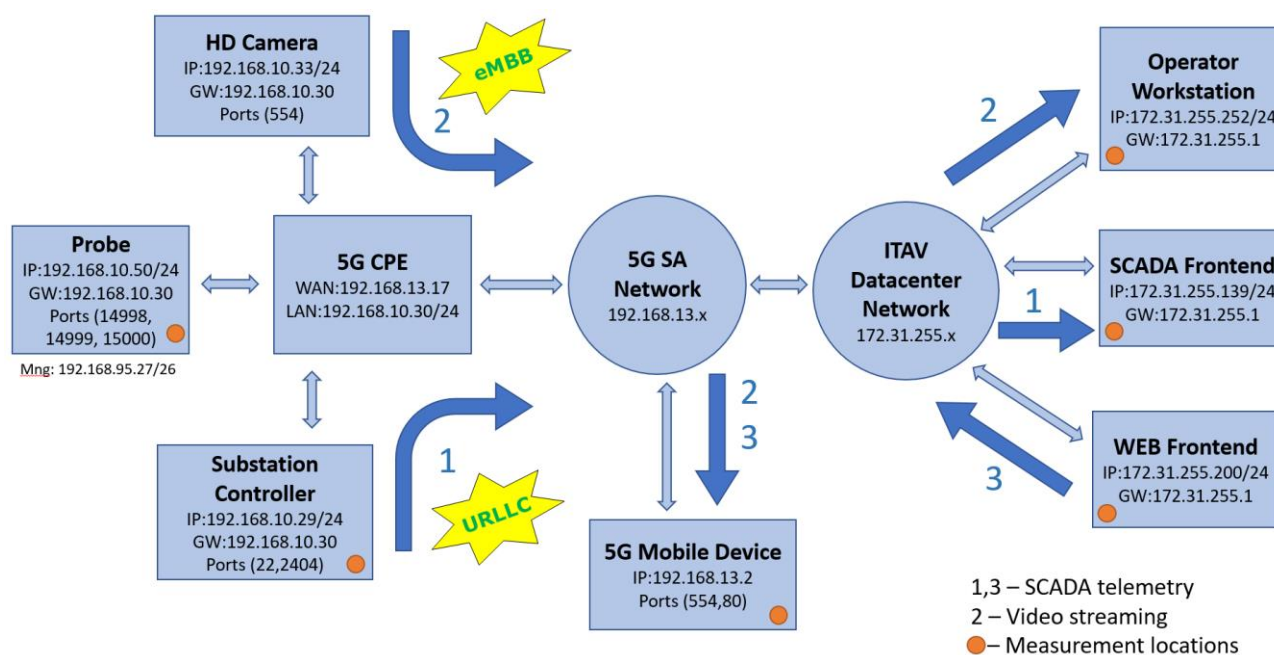


FIGURE 68: USE CASE 1 SETUP USING 5G SA NETWORK WITH ASOCS RAN AND OPEN5GCORE

The following figures summarize the results obtained addressing the latency and availability requirements regarding the telemetry data flow over the 5G network between the Secondary Substation and the Control Center.

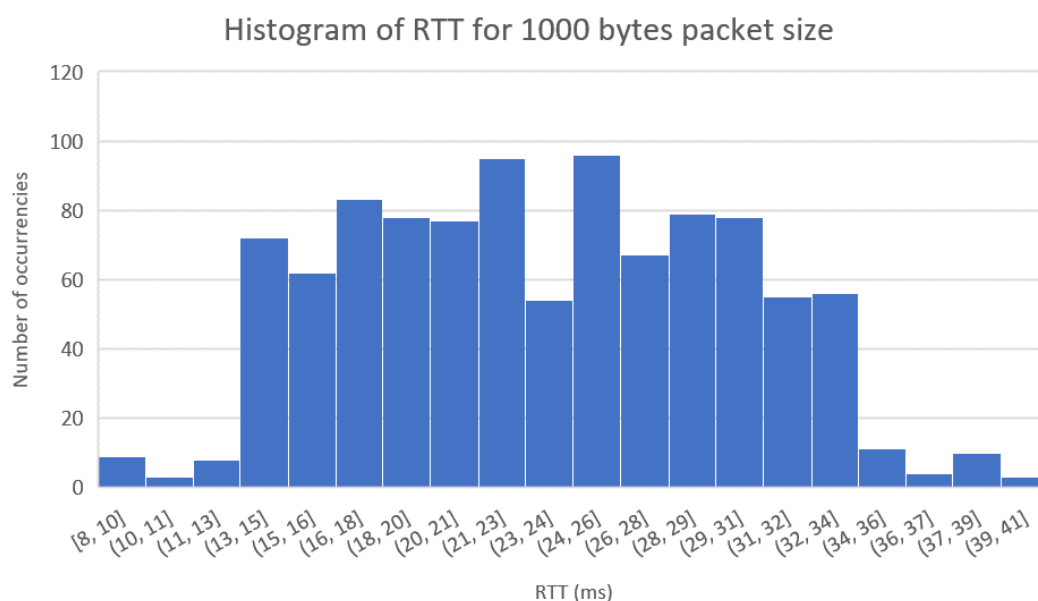


FIGURE 69: HISTOGRAM RTT (1000 BYTES PACKET SIZE)

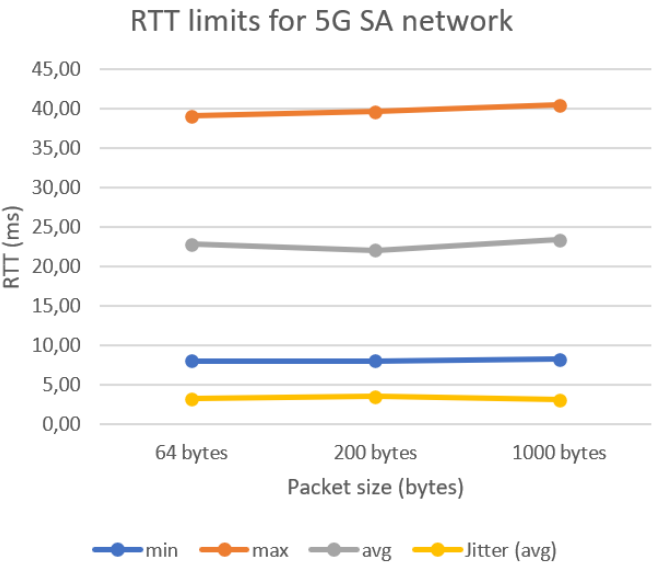


FIGURE 70: RTT LIMITS FOR DIFFERENT PACKET SIZE

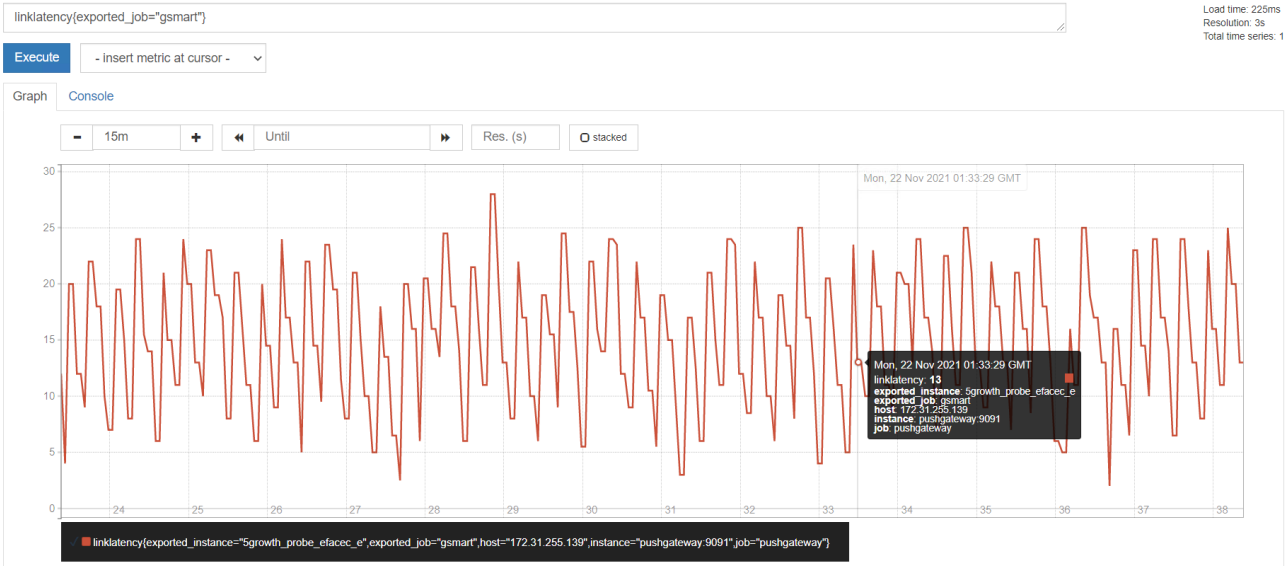


FIGURE 71: 5GROWTH MONITORING PLATFORM – LATENCY IN A 15 MINUTE PERIOD

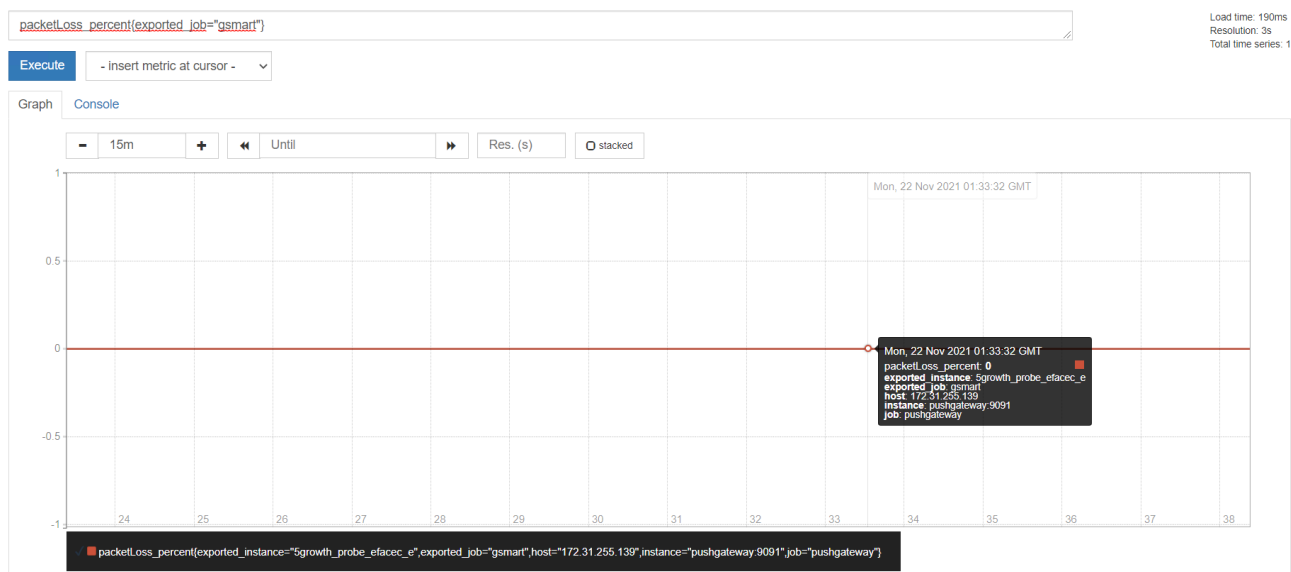


FIGURE 72: 5GROWTH MONITORING PLATFORM – PACKET LOSS IN A 15 MINUTE PERIOD

When comparing with the results of the second campaign obtained in the lab, with the RU placed next to the 5G core, the latency is slightly higher, but the obtained average and maximum latency values are well controlled and under the limit for this scenario (100ms).

As previously stated in D4.3 [6], section 3.4.1, considering the nature of the project and the fact that the 5G network solution used in the Energy pilot is a non-commercial product, it does not make sense trying to measure the network availability. Conversely, it is measured the service availability considering the number of lost packets per total amount of transmitted packages during a transmission period.

No packet loss was detected, at the conditions of this test, during the test period which results in a service availability of 100%.

The following figures and tables summarize the results obtained addressing the data rate and availability requirements regarding the video streaming over the 5G network between the Secondary Substation and the Control Center.

Bitrate	Bitrate receiver	Jitter	Lost Datagrams	Total Datagrams	Lost/Total
1 Mbits/sec	1,0 Mbits/sec	9,992 ms	0	1000	0,00%
10 Mbits/sec	10,0 Mbits/sec	1,647 ms	0	9999	0,00%
20 Mbits/sec	20,0 Mbits/sec	1,083 ms	18	19998	0,09%
30 Mbits/sec	30,0 Mbits/sec	0,410 ms	21	29998	0,07%
40 Mbits/sec	40,0 Mbits/sec	0,295 ms	16	39997	0,04%
50 Mbits/sec	50,0 Mbits/sec	0,175 ms	22	49997	0,04%
60 Mbits/sec	60,0 Mbits/sec	0,180 ms	13	59996	0,02%

FIGURE 73: THROUGHPUT MEASURED FOR UDP PROTOCOL

Interval	Transfer	Throughput	Tx retries	
0,00-60,00 sec	471 MBytes	65,8 Mbits/sec	3 sender	
0,00-60,00 sec	471 MBytes	65,8 Mbits/sec		receiver

FIGURE 74: THROUGHPUT MEASURED FOR TCP PROTOCOL

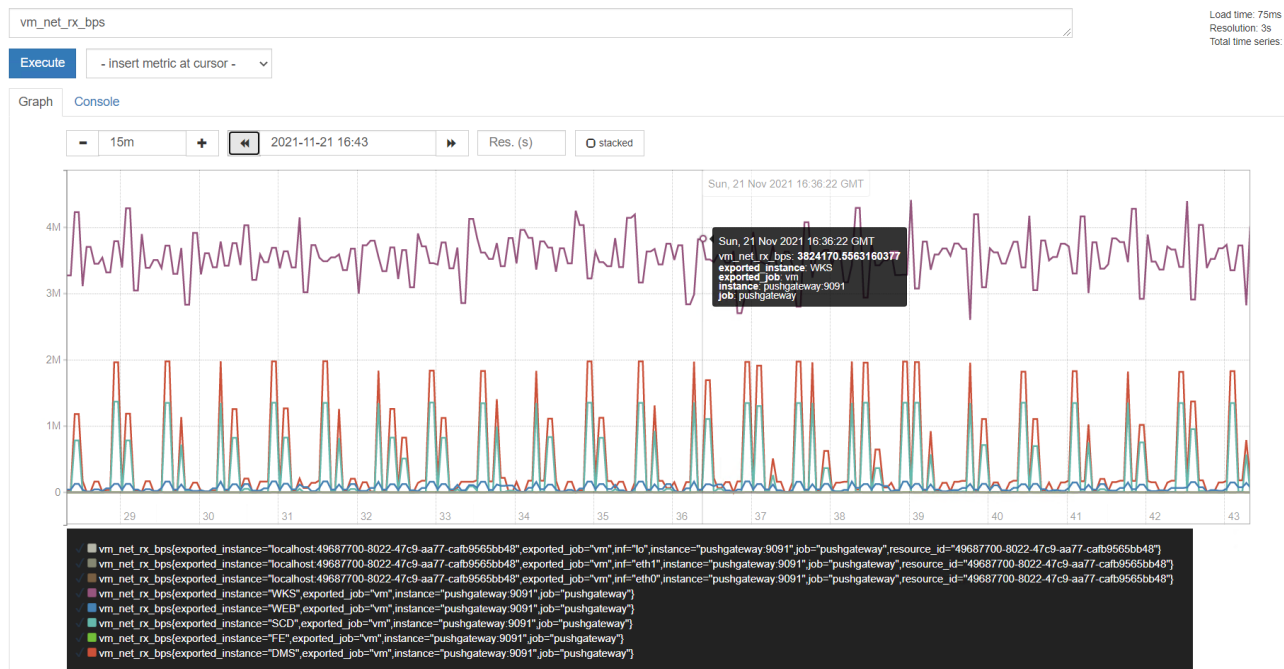


FIGURE 75: 5GROWTH MONITORING PLATFORM – THROUGHPUT IN A 15 MINUTES PERIOD

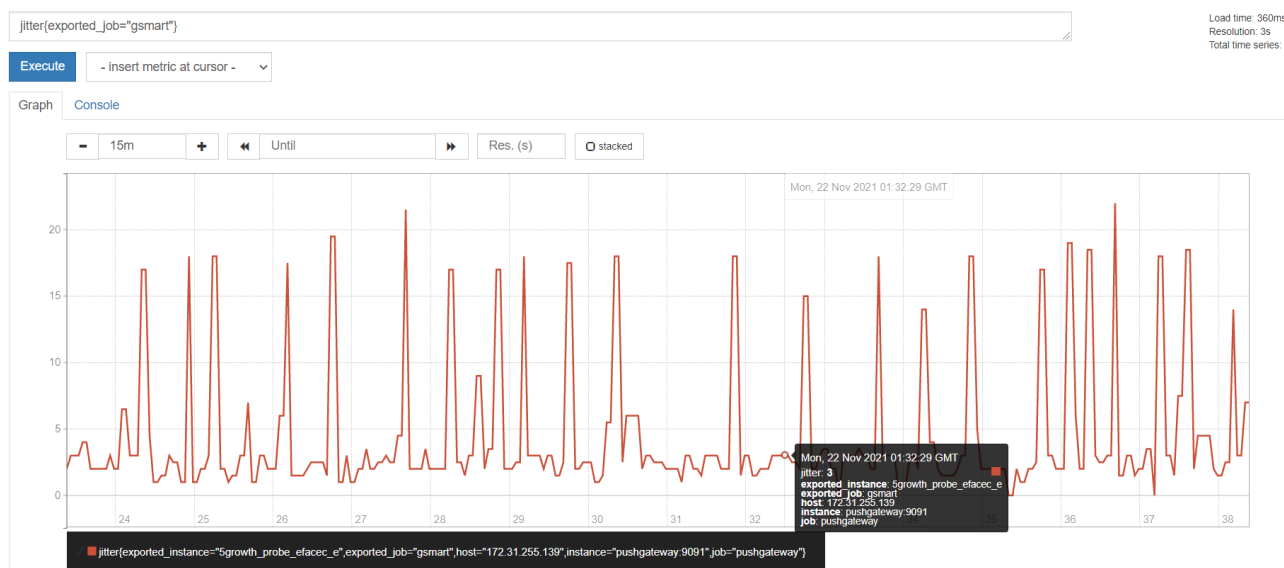


FIGURE 76: 5GROWTH MONITORING PLATFORM – JITTER IN A 15 MINUTE PERIOD



FIGURE 77: 5GROWTH MONITORING PLATFORM – PACKET LOSS IN A 15 MINUTE PERIOD

When comparing with the results of the second campaign obtained in the lab, with the RU placed next to the 5G core, the jitter average is higher, and in the results obtained with the 5Growth probes it is visible the occurrence of jitter spikes over 15 ms. Nevertheless, the experience when watching the video streaming in the Control Center is smooth, detailed and without digital artifacts.

No packet loss was detected, at the conditions of this test, during the test period which results in a service availability of 100%. Even when pushing for higher uplink data rates between 20Mbps and 60 Mbps, the rate of lost packets is low ($< 0.09\%$), and compatible with a good video streaming experience.

The following figures and tables summarize the results obtained addressing the latency and availability requirements regarding the augmented reality scenario between the Control Centre and the Mobile Device.

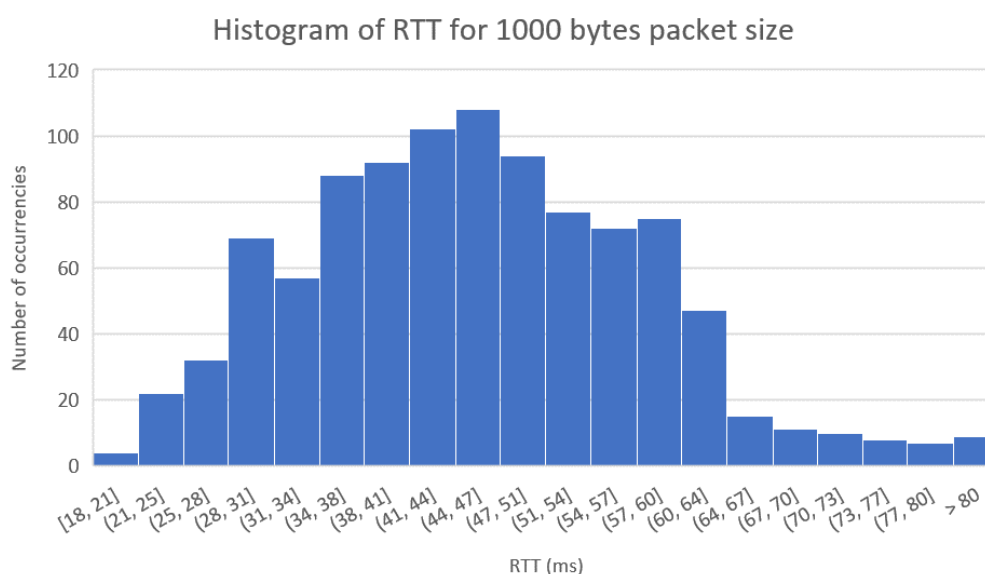
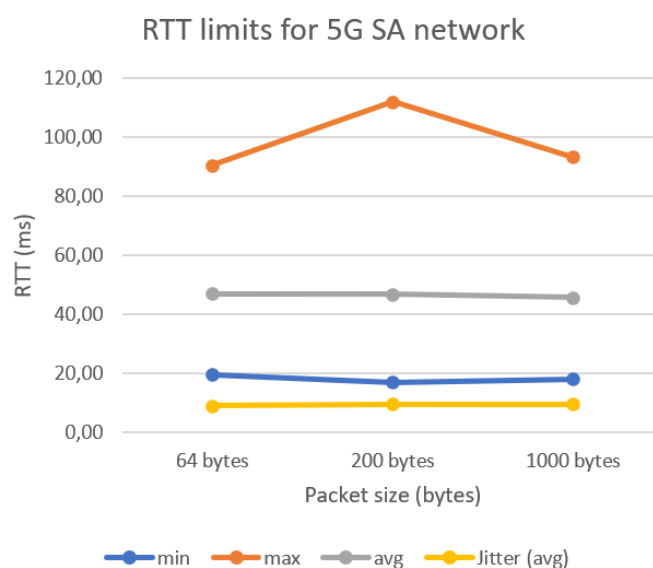


FIGURE 78: HISTOGRAM OF RTT (1000 BYTES PACKET SIZE)

**FIGURE 79: RTT LIMITS FOR DIFFERENT PACKET SIZES**

Bitrate	Bitrate receiver	Jitter	Lost Datagrams	Total Datagrams	Lost/Total
1 Mbits/sec	1,0 Mbits/sec	6,826 ms	0	999	0,00%
10 Mbits/sec	10,0 Mbits/sec	1,487 ms	26	9998	0,26%
20 Mbits/sec	20,0 Mbits/sec	0,725 ms	5	19999	0,03%
30 Mbits/sec	29,4 Mbits/sec	0,652 ms	639	29993	2,13%
40 Mbits/sec	39,9 Mbits/sec	0,311 ms	62	39963	0,16%
50 Mbits/sec	48,8 Mbits/sec	0,272 ms	1165	49980	2,33%
60 Mbits/sec	49,4 Mbits/sec	0,234 ms	10523	59912	17,56%

FIGURE 80: THROUGHPUT MEASURED FOR UDP PROTOCOL

The mobile scenario is available to test for the first time only in the third campaign. The observed latency has average values around 23 ms, but exceeding 100 ms on a few occasions.

On the other hand, a packet loss rate around 0.26% was measured at the conditions of this test, during the test period which results in a service availability of 99.74%, which is considered an average performance considering the scenario requirements.

The following table aggregate the results concerning all the KPIs for Use Case 1.

TABLE 33: EFACEC_E UC1 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	CKPI Result	SKPI Result
P4UC1-SKPI-1: Monitoring Sensors Information Collection and Visualization End- to-end Latency	CKPI-1: E2E latency	11.67 ms	Good

(5GR-SKPI-2)	CKPI-2: Packet Loss (@ ~1Mbit load)	0	
P4UC1-SKPI-2: Monitoring Sensors Information Collection and Visualization Availability (5GR-SKPI-10 & 5GR-SKPI-2)	CKPI-2: Packet Loss (@ ~1Mbit load)	0	Good
	CKPI-5: Availability	100%	
P4UC1-SKPI-3: High-definition Surveillance Video (5GR-SKPI-4)	CKPI-2: Packet Loss (@ ~10Mbit load)	0	Good
	CKPI-3: Guaranteed Data Rate	60.0 Mbits/sec	
	CKPI-9: Jitter (@ ~10Mbit load)	1.647 ms	
P4UC1-SKPI-4: Augmented Reality Information Real-time End-to-end Latency (5GR-SKPI-2)	CKPI-1: E2E latency	22.90 ms	Average
	CKPI-2: Packet Loss (@ ~10Mbit load)	26/9998	
P4UC1-SKPI-5: Augmented Reality information Real-time Availability (5GR-SKPI-10 & 5GR-SKPI-2)	CKPI-2: Packet Loss (@ ~10Mbit load)	26/9998	Average
	CKPI-5: Availability	99.74%	

4.4.2. Use Case 2: Advanced critical signal and data exchange across wide smart metering and measurement infrastructures

For the third validation campaign the use case setup is functionally completed and fully deployed in the vertical site (UA secondary distribution substation, and IT2 building), as described in D3.6 [3], section 4.3.

The probes' location is depicted in Figure 10 in D3.6 [3], Section 4.1.3.2.

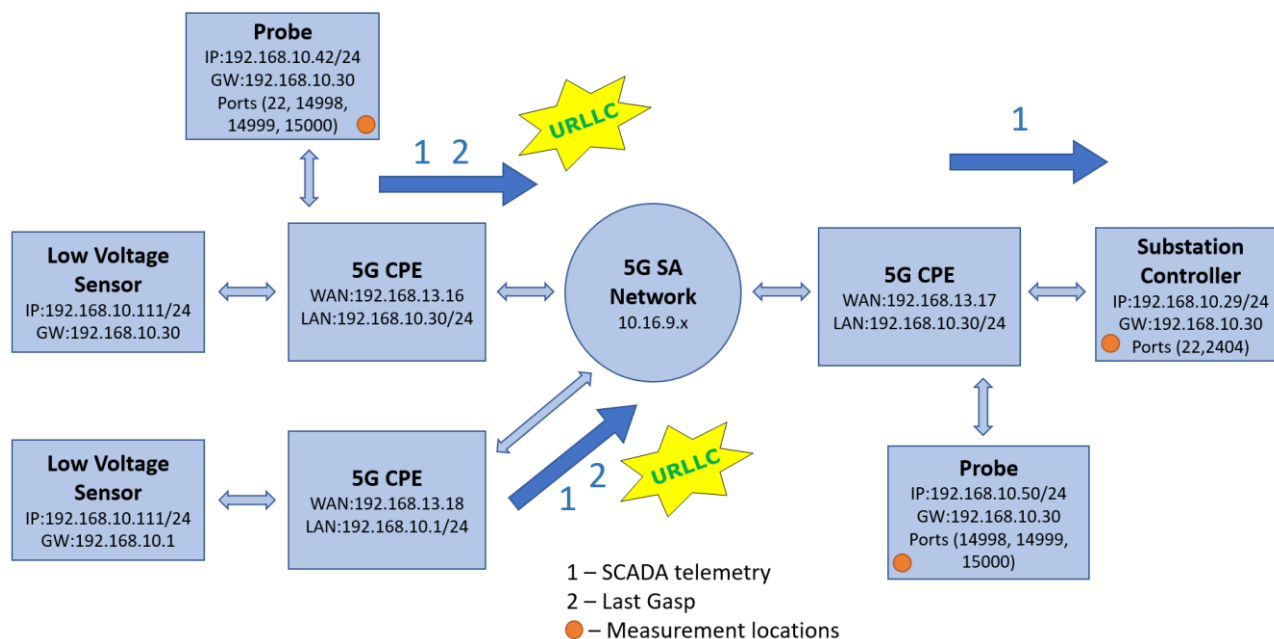


FIGURE 81: USE CASE 2 SETUP USING 5G SA NETWORK WITH ASOCS RAN AND OPEN5GCORE

The following figures summarize the results obtained addressing the latency and availability requirements regarding the telemetry data flow over the 5G network between the Low voltage sensors (LVS3) and the controller inside the secondary distribution substation (Gsmart).

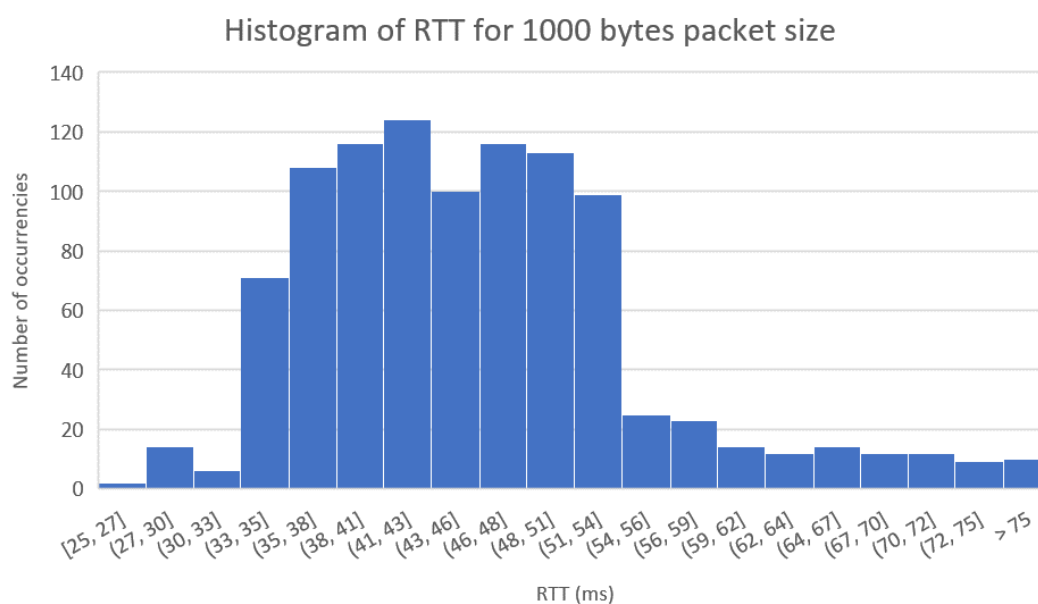
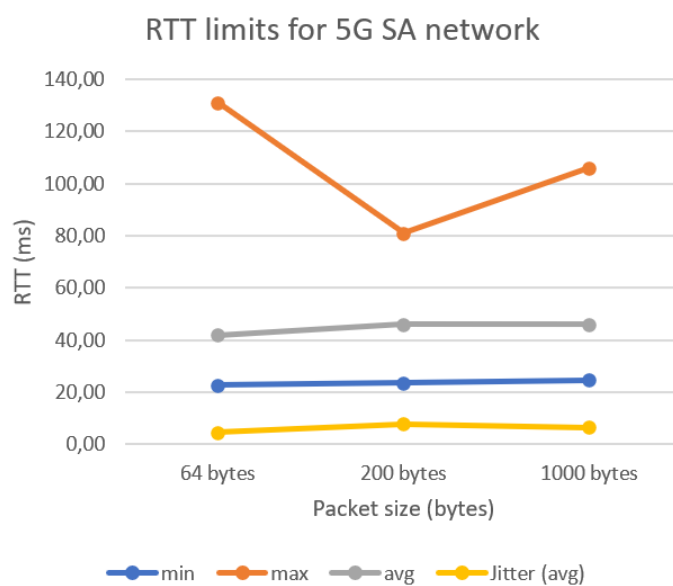


FIGURE 82: HISTOGRAM OF RTT (1000 BYTES PACKET SIZE)

**FIGURE 83: RTT LIMITS FOR DIFFERENT PACKET SIZES****FIGURE 84: 5GROWTH MONITORING PLATFORM – LATENCY IN A 15 MINUTE PERIOD**

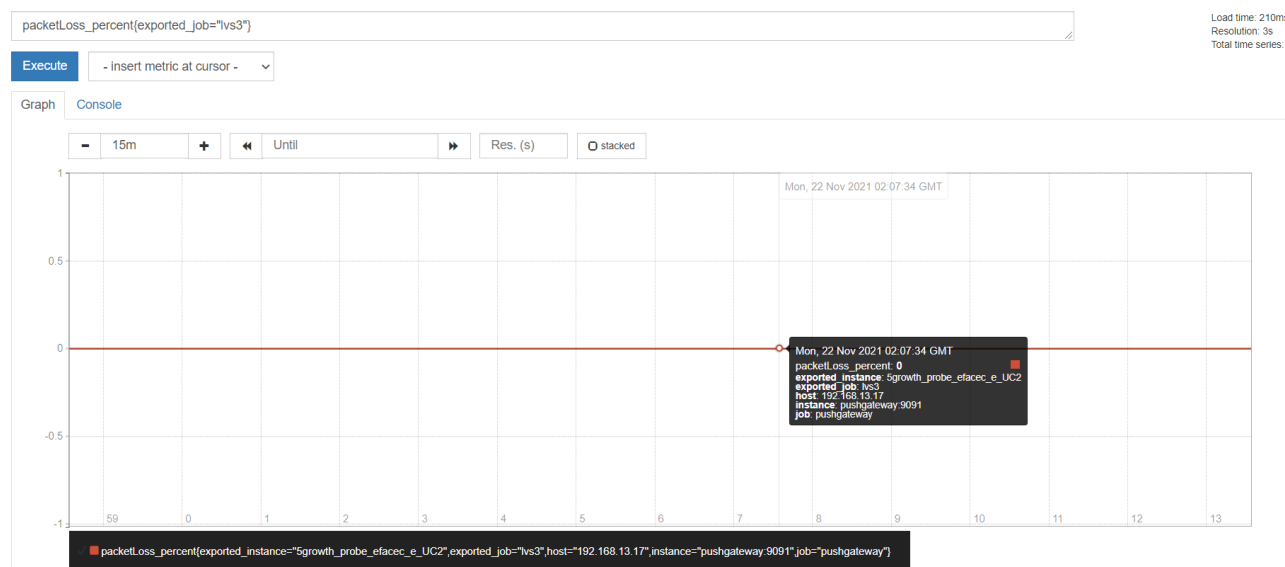


FIGURE 85: 5GROWTH MONITORING PLATFORM – PACKET LOSS IN A 15 MINUTE PERIOD

When comparing with the results of the second campaign obtained in the lab, with all the equipment communicating to just one RU placed next to the 5G core, the average latency is now slightly lower, but a greater variation in the latency range is observed, with maximum latency values sometimes higher than 100ms, which is considered an average performance considering the scenario requirements (latency < 100ms).

No packet loss was detected, at the conditions of this test, during the test period which results in a service availability of 100%.

The following figure summarize the results obtained addressing the latency and availability requirements regarding the last-gasp events over the 5G network between the Low voltage sensors (LVS3) and the controller inside the secondary distribution substation (Gsmart).

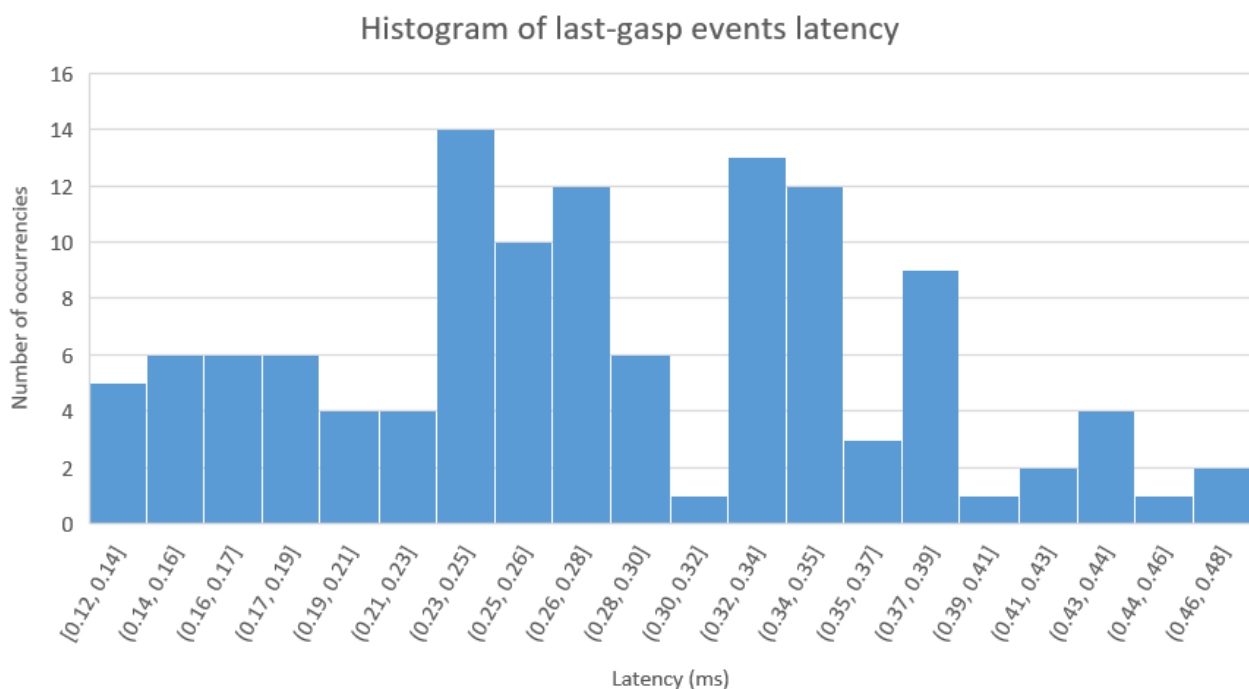
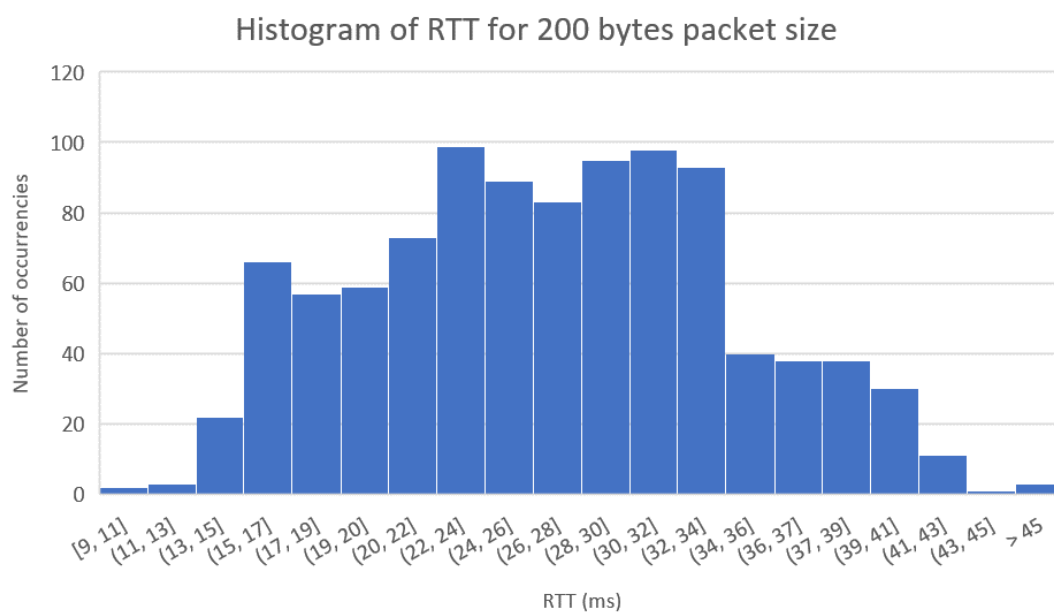
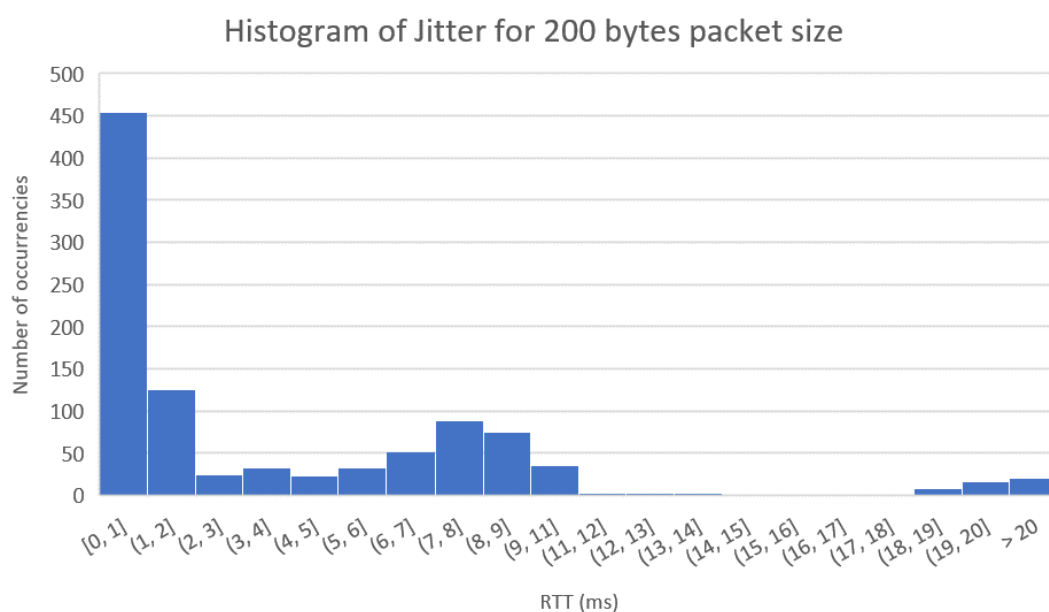


FIGURE 86: HISTOGRAM OF LATENCY FOR LAST-GASP IN A 10 MINUTE PERIOD (UDP TRAP - APPLICATION)

An average latency near 28 ms in the communications of the last-gasp events between the LVS3 installed in the low voltage network and the Gsmart installed inside the secondary substation was measured directly in the application.

No packet loss was observed, with all the last-gasp events created in LVS3 devices being transmitted to the 5G network and received in the Gsmart, which results in a service availability of 100%, in the conditions and period of the test.

The following figures and tables summarize the results obtained addressing the latency and jitter requirements regarding the synchronization scenario between the Low Voltage Sensors.

**FIGURE 87: HISTOGRAM OF RTT BETWEEN LVS3 AND UA NTP SERVER (1000 BYTES PACKET SIZE)****FIGURE 88: HISTOGRAM OF JITTER BETWEEN LVS3 AND UA NTP SERVER (1000 BYTES PACKET SIZE)**

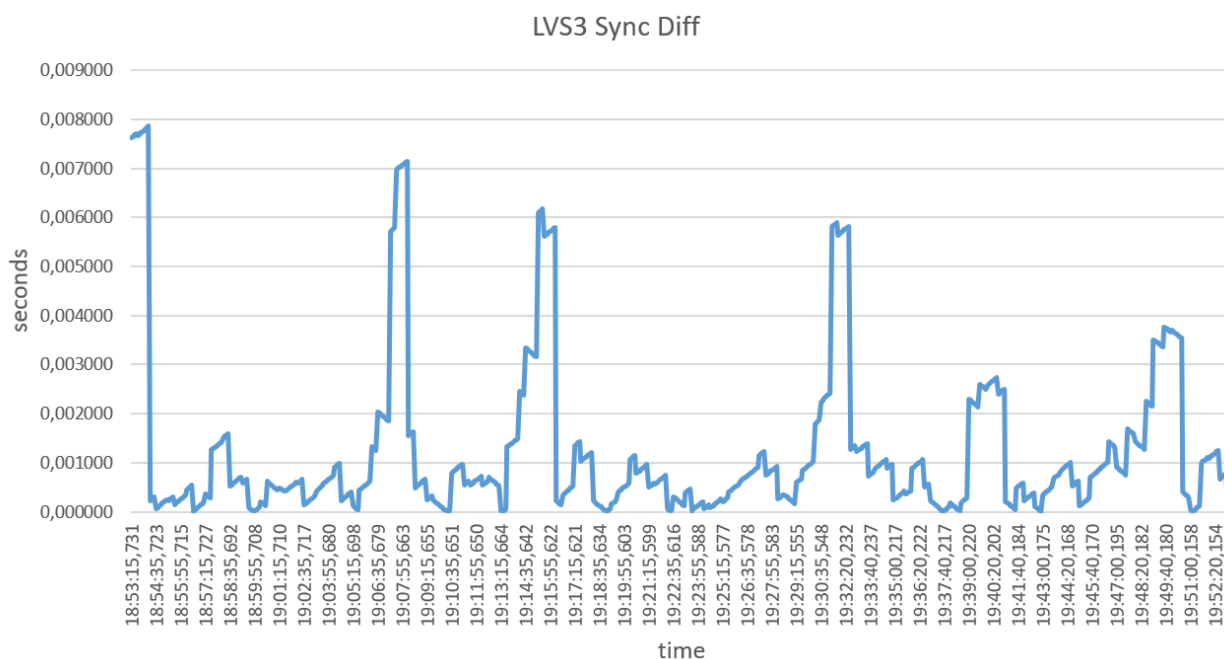


FIGURE 89: SYNC DIFFERENCE BETWEEN LVS3 IN A 60 MINUTE PERIOD (UDP TRAP - APPLICATION)

Considering the time synchronization between the low voltage sensors (LVS3), the average results obtained are good and in line with the target objective for this scenario (~1ms), with an average difference of 1.21 ms.

Some spikes reaching 6 or 7 ms were also observed, nevertheless it is hard to accurately establish responsibilities on this, considering the role of the NTP synchronization process itself running independently on each LVS3 device.

The latency and jitter results in the communication of the LVS3 devices with the NTP server (installed inside UA perimeter) are under control and for sure contribute to the good results obtained in the synchronization between the two LVS3, as described above.

The following table aggregate the results concerning all the KPIs for Use Case 1.

TABLE 34: EFACEC UC2 SPECIFIC SERVICE KPIS, CORE KPIS AND VALIDATION METHODOLOGY

SKPI	CKPI	CKPI Result	SKPI Result
P4UC2-SKPI1: Last-gasp Information End-to-end latency (=5GR-SKPI-2)	CKPI-1: E2E latency	27.94 ms	Average
	CKPI-2: Packet Loss	0/600	
P4UC2-SKPI2: Last-gasp Information Connectivity Availability (=5GR-SKPI-10 & 5GR-SKPI-2)	CKPI-2: Packet Loss	0/121	Good
	CKPI-5: Availability	100%	

P4UC2-SKPI3: Secondary Substation End-to-end Latency (=5GR-SKPI-2)	CKPI-1: E2E latency	22.98 ms	Average
	CKPI-2: Packet Loss (@ ~1Mbit load)	0/1004	
P4UC2-SKPI4: Secondary Substation Availability (=5GR-SKPI-10 & 5GR-SKPI-2)	CKPI-2: Packet Loss (@ ~1Mbit load)	0/1004	Good
	CKPI-5: Availability	100%	
P4UC2-SKPI5: Time Synchronization between Low Voltage Sensors (=5GR-SKPI-2)	CKPI-1: E2E latency	13.40 ms	Good
	CKPI-9: Jitter	4.17 ms	
	Device Synch Diff.	1.21 ms	



5. Conclusions

WP4 leverages the deployment conducted in WP3 to measure pilot-specific business, functional, and technical KPIs. This action was carried out iteratively in three cycles, starting with an initial validation of the non-integrated components of the pilots, and later evaluating the integrated pilots in the vertical industry premises.

Referring to the mature status of the pilots described in D3.6 [3], this document reports on the full maturity of the 5Growth technical solutions, initiated in D4.2 [5] and pursued in D4.3 [6]. It provides details and results of the validation of the features and functionalities available to the verticals after the integration of all the components that make up the use cases: Radio, Transport, Cloud, 5Growth Platform, Vertical Systems, and related applications.

The third validation campaign, reported in this document, verified the fulfilment of the vertical requirements originally defined in D1.1 [1]. This validation has been achieved by measuring or assessing by other methods a set of Core KPIs closely related to a set of Service KPIs according to the specific table included in D4.3 [6].

This deliverable has provided evidence of the work carried out since the previous validation report D4.3 [6], illustrating the evolution of the methodology and tooling and the analyses of the verification results against the target KPIs.

This deliverable has also illustrated the integration process that provides the 5Growth Monitoring Platform with the collected monitoring data from experiments in the 5G EVE and 5G VINNI facilities.

6. References

- [1] 5Growth "Business Model Design", Deliverable 1.1 (D1.1), September 2019.
- [2] 5Growth "Final Design and Evaluation of the innovations of the 5G End-to-End Service Platform", Deliverable 2.3 (D2.3), May 2021.
- [3] 5Growth "Execution report of in-house use cases for Pilots" Deliverable 3.6 (D3.6), November 2021.
- [4] 5Growth "Service and Core KPI Specification", Deliverable 4.1 (D4.1), May 2020
- [5] 5Growth "Verification methodology and tool design", Deliverable 4.2 (D4.2), November 2020.
- [6] 5Growth "5G facility validation and verification report", Deliverable 4.3 (D4.3), June 2021
- [7] Docker - <https://www.docker.com/>
- [8] Docker-compose - <https://docs.docker.com/compose/>
- [9] FastAPI - <https://fastapi.tiangolo.com/>
- [10] Prometheus - Monitoring system & time series database - <https://prometheus.io/>.
- [11] The OpenMetrics project — Creating a standard for exposing metrics data - <https://openmetrics.io/>.
- [12] Prometheus Pushgateway - <https://github.com/prometheus/pushgateway>
- [13] Kappa architecture - <https://www.oreilly.com/radar/questioning-the-lambda-architecture/>
- [14] YANG Tools Developer Guide - OpenDaylight Documentation - <https://docs.opendaylight.org/en/stable-aluminium/developer-guide/yang-tools.html>
- [15] Bosshart, Pat, et al. "P4: Programming protocol-independent packet processors." ACM SIGCOMM Computer Communication Review 44.3 (2014): 87-95.
- [16] Kim, Changhoon, et al. "In-band network telemetry via programmable dataplanes." ACM SIGCOMM. Vol. 15. 2015.
- [17] OpenConfig official website: <https://openconfig.net/> - Visited on 03/11/2021
- [18] YANG official website: www.yang-central.org - Visited on 03/11/2021
- [19] gNMI specification: <https://github.com/openconfig/reference/blob/master/rpc/gnmi/gnmi-specification.md> - Visited on 03/11/2021
- [20] BMv2 The reference P4 software switch - <https://github.com/p4lang/behavioral-model> - Visited on 03/11/2021
- [21] STRATUM. Available at: <https://opennetworking.org/stratum/> - Visited on 22/11/2021
- [22] gNMIc. Available at: <https://gnmic.kmrdev.com/> - Visited on 03/11/2021
- [23] 5G-EVE, "Final version of the experimentation portal", D4.6, May 2021.
- [24] 5growth pilots descriptor repository <https://github.com/5growth/5gr-pilots>
- [25] IT Aveiro OSM VNF Repository <https://github.com/ATNoG/osm-packages/>