5TH GENERATION END-TO-END NETWORK, EXPERIMENTATION, SYSTEM INTEGRATION, AND SHOWCASING

[H2020 - Grant Agreement No. 815178]

Deliverable D3.5

# Monitoring and Analytics (Release A)

Editor Giuseppe Caso (SRL), Özgü Alay (SRL)

Contributors KAU (KARLSTADS UNIVERSITET), LMI (L.M. ERICSSON IRELAND), NCSRD (NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"), UMA (UNIVERSIDAD DE MALAGA), FhG (FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V.), ATOS (ATOS SPAIN SA), INF(INFOLYSIS P.C.), ECM (EURECOM), FON (FON TECHNOLOGY SL), HU (HUMBOLDT-UNIVERSITÄT ZU BERLIN), IHP (IHP GMBH – INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ-INSTITUT FUER INNOVATIVE MIKROELEKTRONIK), PLC (PRIMETEL PLC)

Version 1.0

Date October 15th, 2019

Distribution PUBLIC (PU)

# List of Authors

| | |
|---|---|
| SRL | SIMULA METROPOLITAN CENTER FOR DIGITAL ENGINEERING |
| G. Caso, Ö. Alay | |
| KAU | KARLSTADS UNIVERSITET |
| A. Brunstrom, M. Rajiullah, J. Karlsson, K.-J. Grinnemo | |
| LMI | L.M. ERICSSON IRELAND |
| E. Aumayr, A.-M. Bosneag | |
| NCSRD | NATIONAL CENTER FOR SCIENTIFIC RESEARCH “DEMOKRITOS” |
| G. Xilouris, A. Oikonomakis, T. Anagnostopoulos, H. Koumaras | |
| UMA | UNIVERSIDAD DE MALAGA |
| A. Dias-Zayas , B. Garcia | |
| FhG | FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. |
| M. Emmelmann, F. Eichhorn, T. Briedigkeit, S. Kumar Rajaguru, A. Prakash | |
| ATOS | ATOS SPAIN SA |
| E. Jimeno | |
| INF | INFOLYSIS P.C. |
| C. Sakkas | |
| ECM | EURECOM |
| P. Matzakos | |
| FON | FON TECHNOLOGY SL |
| I. Pretel | |
| HU | HUMBOLDT-UNIVERSITÄT ZU BERLIN |
| L. Reichert, P. Schoppmann | |
| IHP | IHP GMBH – INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ-INSTITUT FUER INNOVATIVE MIKROELEKTRONIK |
| J. Teran Gutierrez | |
| PLC | PRIMETEL PLC |
| A. Phinikarides | |

# Disclaimer

The information, documentation and figures available in this deliverable are written by the 5GENESIS Consortium partners under EC co-financing (project H2020-ICT-815178) and do not necessarily reflect the view of the European Commission.
The 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 reader uses the information at his/her sole risk and liability.

# Copyright

The 5GENESIS Consortium consists of:

| | |
|---|---|
| NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS" | Greece |
| AIRBUS DS SLC | France |
| ATHONET SRL | Italy |
| ATOS SPAIN SA | Spain |
| AVANTI HYLAS 2 CYPRUS LIMITED | Cyprus |
| AYUNTAMIENTO DE MALAGA | Spain |
| COSMOTE KINITES TILEPIKOINONIES AE | Greece |
| EURECOM | France |
| FOGUS INNOVATIONS & SERVICES P.C. | Greece |
| FON TECHNOLOGY SL | Spain |
| FRAUNHOFER GESELLSCHAFT ZUR FOERDERUNG DER ANGEWANDTEN FORSCHUNG E.V. | Germany |
| IHP GMBH – INNOVATIONS FOR HIGH PERFORMANCE MICROELECTRONICS/LEIBNIZ-INSTITUT FUER INNOVATIVE MIKROELEKTRONIK | Germany |
| INFOLYSIS P.C. | Greece |
| INSTITUTO DE TELECOMUNICACOES | Portugal |
| INTEL DEUTSCHLAND GMBH | Germany |
| KARLSTADS UNIVERSITET | Sweden |
| L.M. ERICSSON LIMITED | Ireland |
| MARAN (UK) LIMITED | UK |
| MUNICIPALITY OF EGALEO | Greece |
| NEMERGENT SOLUTIONS S.L. | Spain |
| ONEACCESS | France |
| PRIMETEL PLC | Cyprus |
| RUNEL NGMT LTD | Israel |
| SIMULA RESEARCH LABORATORY AS | Norway |
| SPACE HELLAS (CYPRUS) LTD | Cyprus |
| TELEFONICA INVESTIGACION Y DESARROLLO SA | Spain |
| UNIVERSIDAD DE MALAGA | Spain |
| UNIVERSITAT POLITECNICA DE VALENCIA | Spain |
| UNIVERSITY OF SURREY | UK |

# Version History

| Rev. N | Description | Author | Date |
|---|---|---|---|
| 1.0 | Release of D3.5 | G. Caso, O. Alay (SRL), Almudena Díaz (UMA) | 15.10.2019 |

# LIST OF ACRONYMS

| Acronym | Meaning |
|---|---|
| (e)DRX | (EXTENDED) DISCONTINUOUS RECEPTION |
| (E)GPRS | (ENHANCED) GENERAL PACKET RADIO SERVICE |
| (G)UI | (GRAPHICAL) USER INTERFACE |
| (H)ARQ | (HYBRID) AUTOMATIC REPEAT REQUEST |
| (S)ARIMA | (SEASONAL) AUTO-REGRESSIVE INTEGRATED MOVING AVERAGE |
| (W-)CDMA | (WIDEBAND-)CODE DIVISION MULTIPLE ACCESS |
| 3GPP | 3rd GENERATION PARTNERSHIP PROJECT |
| 5GC | 5G CORE |
| ADB | ANDROID DEBUG BRIDGE |
| AF | APPLICATION FUNCTION |
| AI | ARTIFICIAL INTELLIGENCE |
| AP | ACCESS POINT |
| API | APPLICATION PROGRAMMING INTERFACE |
| ASU | ARBITRARY STRENGHT UNIT |
| CA | CONSORTIUM AGREEMENT |
| CDP | CISCO DISCOVERY PROTOCOL |
| CoAP | CONSTRAINED APPLICATION PROTOCOL |
| CPU | CENTRAL PROCESSING UNIT |
| CQI | CHANNEL QUALITY INDICATOR |
| CSV | COMMA-SEPARATED VALUES |
| DL | DOWNLINK |
| DNS | DOMAIN NAME SYSTEM |
| DT | DECISION TREE |
| DTLS | DATAGRAM TLS |
| DTW | DYNAMIC TIME WARPING |
| e/gNB | EVOLVED/NEXT-GENERATION NODE B |
| E2E | END-TO-END |
| ECM | EPC CONNECTION MANAGEMENT |
| ELCM | EXPERIMENTAL LIFE CYCLE MANAGER |
| eMBB | EVOLVED MOBILE BROADBAND |
| EMM | EPC MOBILITY MANAGEMENT |
| EPC | EVOLVED PACKET CORE |
| ETSI | EUROPEAN TELECOMMUNICATION STANDARDS INSTITUTE |
| FTP | FILE TRANSFER PROTOCOL |
| GA | GRANT AGREEMENT |
| GPS | GLOBAL POSITIONING SYSTEM |
| GPU | GRAPHICS PROCESSING UNIT |
| GSM | GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS |
| HSPA | HIGH SPEED PACKET ACCESS |
| HSS | HOME SUBSCRIBER SERVER |
| HTTP | HYPERTEXT TRANSPORT PROTOCOL |
| HW/SW | HARDWARE/SOFTWARE |
| I/O | INPUT/OUTPUT |
| ICMP | INTERNET CONTROL MESSAGE PROTOCOL |
| IETF | INTERNET ENGINEERING TASK FORCE |

| | |
|---|---|
| IM | INFRASTRUCTURE MONITORING |
| IoT | INTERNET OF THINGS |
| IPFIX | INTERNET PROTOCOL FLOW INFORMATION EXPORT |
| JSON | JAVASCRIPT OBJECT NOTATION |
| KPI | KEY PERFORMANCE INDICATOR |
| LAC | LOCATION AREA CODE |
| LLDP | LINK LAYER DISCOVERY PROTOCOL |
| LTE(-A) | LONG-TERM EVOLUTION(-ADVANCED) |
| LwM2M | LIGHTWEIGHT MACHINE TO MACHINE |
| M&A | MONITORING AND ANALYTICS |
| MAC (LAYER) | MEDIUM ACCESS CONTROL LAYER |
| MAD | MEDIAN ABSOLUTE DEVIATION |
| MANO | MANAGEMENT AND ORCHESTRATION |
| MCS | MODULATION (AND) CODING SCHEME |
| MDAF | MANAGEMENT DATA ANALYTIC FUNCTION |
| MDAS | MANAGEMENT DATA ANALYTIC SERVICE |
| MIMO | MULTIPLE INPUT MULTIPLE OUTPUT |
| ML | MACHINE LEARNING |
| MME | MOBILE MANAGEMENT ENTITY |
| mMTC | MASSIVE MACHINE TYPE COMMUNICATION |
| MQTT | MESSAGE QUEUING TELEMETRY TRANSPORT |
| NAS | NON-ACCESS STRATUM |
| NB-IoT | NARROWBAND-IOT |
| NF | NETWORK FUNCTION |
| NFV | NETWORK FUNCTION VIRTUALIZATION |
| NFVI | NFV INFRASTRUCTURE |
| NR | NEW RADIO |
| NSSF | NETWORK SLICE SELECTION FUNCTION |
| NTP | NETWORK TIME PROTOCOL |
| NWDAF | NETWORK DATA ANALYTICS FUNCTION |
| OAI | OPEN AIR INTERFACE |
| OPEX | OPERATIONAL COSTS |
| OS | OPERATING SYSTEM |
| OSM | OPEN SOURCE MANO |
| OWAMP | ONE-WAY PING |
| PCF | POLICY CONTROL FUNCTION |
| PDCP | PACKET DATA CONVERGENCE PROTOCOL |
| PGW | PACKET DATA NETWORK GATEWAY |
| PHY (LAYER) | PHYSICAL LAYER |
| PM | PERFORMANCE MONITORING |
| POSIX | PORTABLE OPERATING SYSTEM INTERFACE |
| PSC | PRIMARY SITE CONTROLLER |
| PSM | POWER SAVING MODE |
| PTW | PAGING TIME WINDOW |
| QoE | QUALITY OF EXPERIENCE |
| QoS | QUALITY OF SERVICE |
| RAM | RANDOM ACCESS MEMORY |
| RAN | RADIO ACCESS NETWORK |
| REST | REPRESENTATIONAL STATE TRANSFER |

| | |
|---|---|
| RF | RANDOM FOREST |
| RI | RANK INDICATOR |
| RLC (LAYER) | RADIO LINK CONTROL LAYER |
| RNN | RECURRENT NEURAL NETWORK |
| RRC | RADIO RESOURCE CONTROL |
| RSRP | REFERENCE SIGNAL RECEIVED POWER |
| RSRQ | REFERENCE SIGNAL RECEIVED QUALITY |
| RSSI | RECEIVED SIGNAL STRENGTH INDICATOR |
| RSSNR | REFERENCE SIGNAL SNR |
| RTT | ROUND TRIP TIME |
| S(I)NR | SIGNAL TO (INTERFERENCE PLUS) NOISE RATIO |
| SBA | SERVICE-BASED ARCHITECTURE |
| SBI | SERVICE-BASED INTERFACE |
| SDN | SOFTWARE DEFINED NETWORKING |
| sFLOW | SAMPLED FLOW |
| SGW | SERVING GATEWAY |
| SLA | SERVICE LEVEL AGREEMENT |
| SNMP | SIMPLE NETWORK MANAGEMENT PROTOCOL |
| SON | SELF-ORGANIZING NETWORK |
| SSL/TLS | SECURE SOCKETS LAYER / TRANSPORT LAYER SECURITY |
| SVM | SUPPORT VECTOR MACHINE |
| T/U/AM | TRANSPARENT / UNACKNOWLEDGED / ACKNOWLEDGED MODE |
| TAP | TEST AUTOMATION PLATFORM |
| TAU | TRACKING AREA UPDATE |
| TCP | TRANSMISSION CONTROL PROTOCOL |
| TD-SCDMA | TIME DIVISION-SYNCHRONOUS CDMA |
| TTI | TRANSMISSION TIME INTERVAL |
| UDP | USER DATAGRAM PROTOCOL |
| UE | USER EQUIPMENT |
| URLLC | ULTRA RELIABLE LOW LATENCY COMMUNICATION |
| VIM | VIRTUALIZED INFRASTRUCTURE MANAGER |
| VM | VIRTUAL MACHINE |
| VN | VIRTUAL NODE |
| VNF | VIRTUAL NETWORK FUNCTION |
| W(L)AN / WLAN | WIDE (LOCAL) AREA NETWORK / WIRELESS LOCAL AREA NETWORK |
| WiFi | WIRELESS FIDELITY |
| WSMP | WIFI SERVICE MANAGEMENT PLATFORM |

# Executive Summary

This document describes the design and implementation of the 5GENESIS Monitoring & Analytics (M&A) framework (Release A), developed within Task T3.3 of the Project work plan. Figure 1 shows the distributed approach of the M&A framework as part to the 5GENESIS architecture.

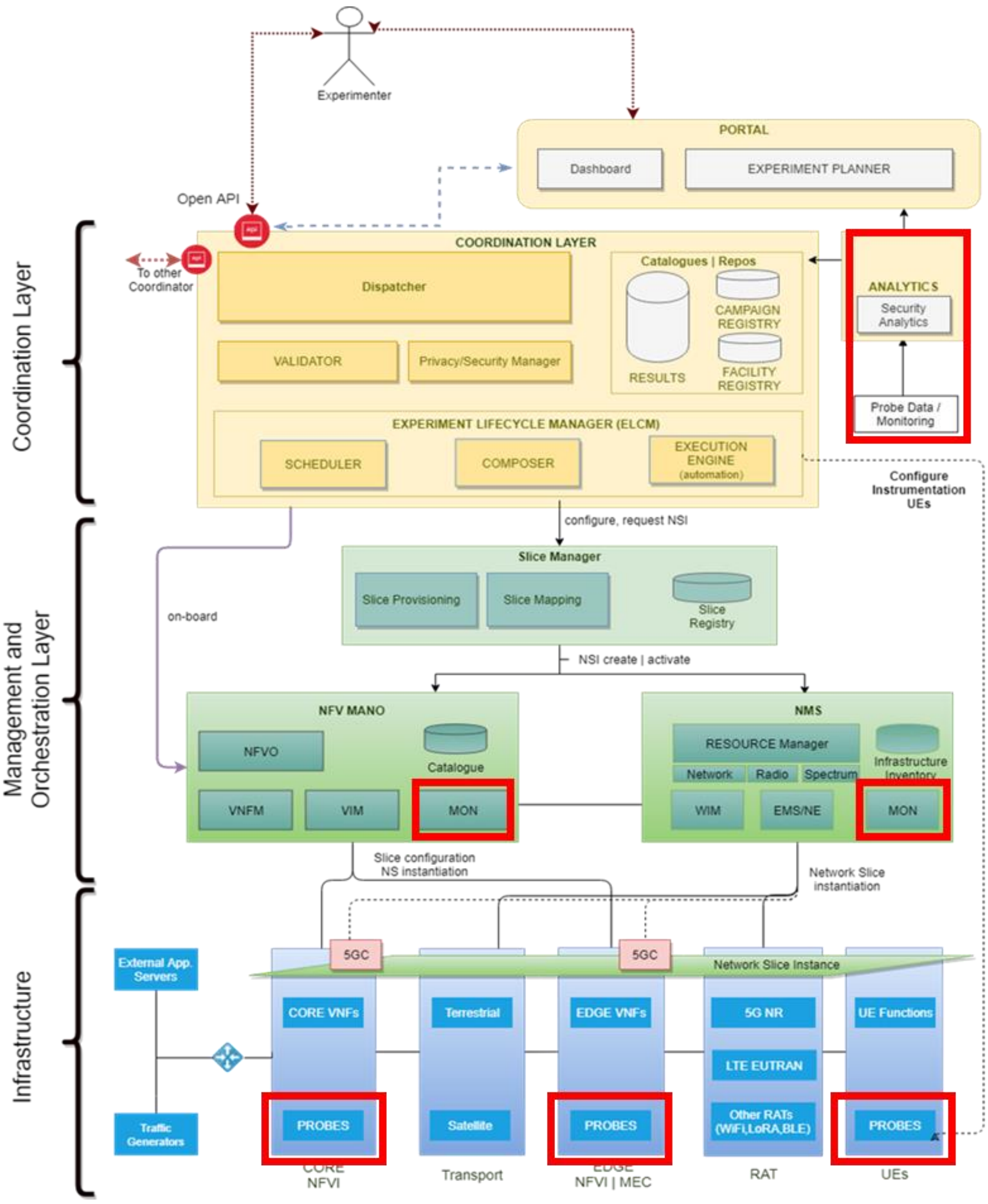


Figure 1 5GENESIS reference architecture: red lines highlight M&A components

The instantiation of a high-performing M&A framework is crucial for a modern communication system, and 5G cellular networks exacerbate such requirement. In particular, this is due to the fact that the services provided by 5G systems have to comply with Service Level Agreements (SLAs), which state the end-to-end (E2E) performance that has to be guaranteed to end-users and verticals, leading to the need of careful management and monitoring of the instantiated resources. A reliable and efficient M&A framework should ultimately consider both end-users' and operators' perspectives, aiming to satisfy and improve user's Quality of Service and Experience (QoS/QoE) and operator's management and operational costs.

Within the above context, the 5GENESIS M&A framework thus includes Monitoring tools and advanced Machine Learning (ML)-oriented Analytics, devoted to the collection and analysis of the heterogeneous data produced during the usage of the 5GENESIS Platform. The ultimate goal, within the Project scope, is to verify the status of the infrastructure components during the execution of experiments for the validation of 5G Key Performance Indicators (KPIs).

In its Release A, the 5GENESIS M&A framework is designed and implemented in 3 main interoperable functional blocks:

- *Infrastructure Monitoring (IM)*, which focuses on the collection of data that synthesize the status of architectural components, e.g., end-user devices, radio access and networking systems, computing and storage distributed units;
- *Performance Monitoring (PM)*, which is devoted to the active measure of E2E QoS/QoE indicators.
- *Storage and ML Analytics*, which enables efficient management of large sets of heterogeneous data, and drives the discovery of hidden values and correlation among them.

The parallel use of IM and PM tools, along with ML Analytics, allows a full and reliable assessment of the KPIs, possibly pinpointing issues leading to performance losses, and ultimately triggering the use of improved network policies and configurations during next experiment executions.

Both framework design and implementation have been carried out considering the 5GENESIS common reference architecture, as well as both commonalities and peculiarities of the 5GENESIS platforms.

The definition and implementation of M&A Release A will serve as a basis for the development and full assessment of the Release B, and will be presented in the next version of this document, D3.6 "Monitoring and Analytics" (Release B), and will represent the final version of the M&A framework operating within the 5GENESIS architecture.

# Table of Contents

# 1. Introduction

## 1.1. Document scope

Embedding a complete Monitoring and Analytics (M&A) framework is key for the design and implementation of a communication system, and 5G cellular networks exacerbate such requirement [1]–[4]. With regards to ***Monitoring***, several techniques, methodologies, and protocols exist; among those, an important functional difference can be highlighted between Infrastructure Monitoring (IM) and Performance Monitoring (PM) [5][6]. On the one hand, IM focuses on the collection of data that synthesize the status of architectural components, e.g., end-user devices, radio access and networking systems, computing and storage distributed units, to mention a few, and also includes passive monitoring of traffic over network interfaces. On the other hand, PM is devoted to actively measure the end-to-end (E2E) performance indicators, in order to highlight the end-users' perspective in terms of Quality of Service (QoS) and Quality of Experience (QoE). Considering ***Analytics***, traditional schemes are based on statistical approaches, which identify the system behavior by statistically analyzing the data collected by the monitoring probes. Nowadays, such approaches are being complemented by Machine Learning (ML)-based schemes, that seem to better cope with the exponential growth of collectable data, and also implicitly trigger the application of Artificial Intelligence (AI) to several network functionalities, towards automation and self-organization [7]–[10]. Analytics based on ML and big data enables efficient management of large sets of heterogeneous data, and drives the discovery of hidden values and correlation among them.

Within the 5G context, where the 5GENESIS Platform is a main actor towards 5G KPI validation and showcasing, the M&A framework faces increased network complexity, heterogeneity, dynamicity, and performance demands. Complexity and dynamicity are due to, in particular, the increasing heterogeneity and reconfigurability of the Radio Access Network (RAN), as well as the introduction of Software Defined Networking (SDN), Network Function Virtualization (NFV), and network slicing paradigms, instantiated on top of networking, computing, and storage resources placed in the cloud or at network edges [1][11]–[14]. Moreover, heterogeneity and increased performance demands are a consequence of the extension of use cases envisioned for 5G with respect to previous generations, and find a clear expression in the definition of evolved Mobile BroadBand (eMBB), Ultra Reliable Low Latency Communication (URLLC), and massive Machine Type Communication (mMTC) services, which ultimately require a sliced network architecture. The instantiation of a high-performing M&A framework is key from a user performance perspective, since it allows to directly consider a user-centric, QoS/QoE-based perspective in the E2E network optimization schemes. Moreover, it is also extremely valuable from the point of view of operators and technology providers [3]. On the one hand, in order to reduce operational costs (OPEX), 5G systems have to pair traditional, mostly human-driven and reactive maintenance mechanisms with autonomous, no human-driven and proactive reconfigurations, which are enabled by a ML/AI-oriented M&A framework [10]. On the other hand, the assurance of a unified and homogeneous service across the same type of users, e.g., belonging to the same slice, is a complex problem whose solution depends on a large amount of factors. In a 5G system, solving this challenge becomes extremely important, considering that provided services have to comply with Service Level Agreements

(SLAs), stating the E2E performance that has to be guaranteed to end-users and verticals. Conventional solutions based on resource overprovisioning increase the network costs and are thus inefficient, and for this reason the design and usage of M&A-based resource allocation schemes are being strongly pursued by mobile network operators [4].

A 5G M&A solution should coherently and complementarily embed IM, PM, and Analytics, in order to collect metrics reporting the status of the system components and QoS/QoE KPIs, and analyze such metrics, in order to find how performances and costs are affected by the system status, triggering reconfiguration and optimization when needed [5][6].

In the 5GENESIS project, Task T3.3 focuses on the design and the implementation of a reliable, efficient, and unified Monitoring and Analytics (M&A) framework across the 5GENESIS platforms. The M&A framework enables to monitor and analyze heterogeneous data, such as infrastructure parameters, traffic, and performance indicators, in order to verify the status of the Platform during its operation, thus allowing a reliable assessment of the KPIs, and possibly pinpointing issues leading to performance losses, which would require the use of improved network policies and configurations.

In this document, we summarize the main activities carried out within Task T3.3 during the 5GENESIS first development phase. Our main goal is to describe the Release A of the M&A framework design and implementation in the 5GENESIS. We further report initial results on the usage of the framework in order to drive the extension of Release A towards the deployment and integration of the final Release B.

## 1.2. Document structure

Section 2 of this document assesses the state-of-the-art for both Monitoring and Analytics solutions, focusing in particular on 5G-oriented solutions. Section 3 presents the Release A of the 5GENESIS M&A framework, discussing the main components, as well as the interfaces with the rest of the 5GENESIS reference architecture. Both platform-agnostic and platform-specific IM and PM tools are described and discussed in Sections 4 and 5, respectively; Section 6 deals with the Analytics main components, focusing in particular on the description of statistical and ML-oriented functionalities under development and integration. Sections 7 report initial use cases. Section 8 summarizes the features of Release A and plans for extending the framework during the next Project phase into its final version, Release B. Conclusions are drawn in Section 9. Finally the annexes are provided at the end of the document.

# 2. Monitoring & Analytics State of the Art

In this section, we provide an overview of the state of the art in Monitoring and Analytics approaches that have been considered during the design of the 5GENESIS M&A framework.

## 2.1. Monitoring State of the Art

As mentioned above, IM and PM tools have to work in parallel in a 5G M&A solution, in order to collect data in a nearly-synchronized manner. Moreover, an important aspect related to PM is to complement traditional E2E measurements with *in-network* counterparts. For example, 5G networks require the Monitoring system, with the help of Analytics, to promptly pinpoint and identify performance bottlenecks caused by in-network malfunctions, that hinders the compliance with SLAs. A plethora of IM and PM tools has been deployed over the years, and some of these tools are being adapted to work in 5G systems [11][12][15]–[19]. A complete description of such tools is out of the scope of this document; a comprehensive list (continuously updated) of IM and PM monitoring solutions can be found in [20], and a good comparison of IM tools is also provided in [21]. However, in order to emphasize current limitations, and in turn highlight the motivation behind the M&A framework deployed within the 5GENESIS project, IM and PM general characteristics and functionalities are discussed in the following, mentioning as a reference the open-source tools embedded in the 5GENESIS Platform. Finally, a taxonomy of IM and PM data and parameters that is possible to collect across a 5G system is also provided.

***Infrastructure Monitoring:*** IM tools aim to provide an overview of the status of the infrastructure by scraping metrics and parameters from the underlying architectural components, in particular via passive mechanisms that do not inject any traffic. These tools usually adopt distributed probes retrieving large amounts of heterogeneous metrics that are exposed by network management protocols, e.g., Simple Network Management Protocol (SNMP), or cloud/edge SDN/NFV instances. High-layer, generic IM tools, such as Prometheus [22] and Zabbix [23], are often used as a centralized solution that interoperates with low-layer, dedicated IM tools and probes, which are devoted to the collection of specific parameters. For example, Prometheus and Zabbix are able to scrape metrics from SDN controllers based on OpenFlow [24], the de-facto SDN standard enabler, as well as from ETSI-compliant NFV components, i.e., OpenStack [25] as Virtualized Infrastructure Manager (VIM), and Open Source MANO (OSM) [26] or Open Baton [27] as Management and Orchestration (MANO). Among others, metrics from OpenStack, mainly collected via a dedicated IM tool called Ceilometer [28], can be forwarded to Prometheus and Zabbix, and specific plugins allow these latter to also collect metrics on the status and operations of Virtual Machines (VMs) and Virtual Network Functions (VNFs) managed by Open Source MANO (OSM) or Open Baton (e.g., see [29]). The collected metrics are often redirected to a centralized entity, e.g., a Prometheus or Zabbix server, which provides a global, infrastructure-level overview.

***Performance Monitoring***: PM includes a large variety of tools, depending on the QoS/QoE metric being monitored [19]. Overall, state-of-the-art focuses on E2E PM, mostly carried out via active probing, i.e., by generating traffic. However, different measurement methodologies exist for each KPI. For example, considering the most common network KPI, the throughput, there are several probing methodologies, leading to a large amount of open/closed-source *speedtests*. This is mainly due to the lack of a general consensus on the methodology to adopt, but also considering that the same KPI can be actually analyzed at different network layers. An important aspect to consider is that the adoption of a particular tool for monitoring a given KPI implicitly affects the measured values [30]. Hence, in large and distributed experimental facilities, such as the one under development in 5GENESIS, it is important to define and report the tools and methodologies adopted for PM, and converge to uniform and common procedures whenever possible, in order to allow experiment reproducibility and result comparability across different platforms. In light of these aspects and within the scope of WP6, the 5GENESIS Consortium worked during the 1st year towards the definition of common testing procedures across the platforms, leading up to shared experiment and test case templates, used to report and assess KPI measurements. Such activities and related outcomes are largely documented in Deliverable D6.1 [31].

*Data Taxonomy*: As mentioned above, a challenge but also an opportunity for 5G Monitoring is the large amount of heterogeneous data that can be collected, stored, analyzed, and ultimately used to track, understand, and optimize system behavior and performance. A preliminary taxonomy of such data, based on domain knowledge, can thus help to identify specific (classes of) parameters that require particular attention and should be used as features for more advanced, ML-oriented analysis.

- *IM data* range across all physical and virtual components of a 5G system. They cover User and User Equipment (*UE)-related data*, such as experienced radio conditions when connected to the network, in terms of Reference Signal Received Power (RSRP), Received Signal Strength Indicator (RSSI), Reference Signal Received Quality (RSRQ), Signal to Interference plus Noise Ratio (SINR), and Channel Quality Indicator (CQI), as well as device power consumption and constraints, users' mobility patterns, and usage profiles at application level. Such parameters can significantly help to setup efficient network configurations and optimized service composition, particularly in the context of a 5G sliced architecture with SLA assurance. On the other side of the system, the collection of *Network-related data*, including RAN/transport parameters on average conditions in terms of resource availability, per-interface I/O traffic loads, per-UE adopted settings, and backhaul/fronthaul type and topology, is key for SLA assurance, while promptly pinpointing unexpected infrastructure behaviors.

  *Core*, *Cloud/Edge*, and *SDN/NFV-related data* provide further observation degrees within the IM context; on the one hand, 5G Core (5GC) monitoring allows the collection of parameters on processing loads of particular core functions, as well as the observation of logs on active bearers and session timeouts, and of agglomerate statistics on successful vs. failed interface setups and UE attaches. On the other, Cloud/edge and SDN/NFV monitoring provides information on processing parameters related to availability and utilization of computational resources, including power and CPU consumption, RAM load, and Disk utilization, to mention a few.

  Finally, when it comes to *Traffic-related data*, specialized industry standards such as sFlow and NetFlow, as well as the IETF protocol IP Flow Information Export (IPFIX), are largely adopted nowadays, and will be likely adopted in 5G systems as well [32]. sFlow samples the observed traffic and provides statistics on the observed protocols. An sFlow record contains Ethernet frame samples and captures the first 128 bytes of each frame, thus including IPv4 and transport layer headers, and tens of TCP and UDP payload bytes. Introduced by Cisco, NetFlow does not capture payloads but only IP and protocols information. IPFIX is a standard protocol taking roots from NetFlow.

- ***PM data*** primarily include traditional QoS/QoE performance KPIs, such as throughput and latency metrics from PHY to application layers, since these KPIs are tailored to assess eMBB and, to some extent, URLLC performance. However, the heterogeneity of 5G use cases and verticals, which include, among others, mMTC, vehicular communications, mission critical and location-based services, requires to extend the pool of PM data towards novel and more specific KPIs. For example, mMTC requires to monitor metrics related to the density of supported and successfully connected devices, as well as the energy efficiency of these latter and the infrastructure in performing the required operations. As part of URLLC, mission critical services require the system to be always aware and thus collect indicators related to service reliability, from connectivity to prompt service creation and dissemination.

## 2.2. Analytics State of the Art

The main goal of Analytics is to find correlations and causalities between system status and network KPIs, in order to validate the network KPIs as well as provide performance improvements by detecting and resolving the identified bottlenecks and system malfunctions. The heterogeneous data collected by the monitoring probes are exposed to the Analytics as a *post*, *on-the-fly*, or o*n-demand* process [4][7][33]. In the first case, Analytics is performed on a days-to-seconds time scale, and applies to certain types of service, e.g., QoS and mobility management, as defined by the O-RAN Alliance [34], but also as a tool to investigate the possibility of enabling long-term system changes, e.g., the introduction of new RAN components, further edge computing units, or advanced security systems. On-the-fly Analytics is arguably more challenging since it acts on a more stringent time scale (around milliseconds), and thus requires additional processing capabilities to the system, which overhead may affect the overall performance. Finally, the on-demand scenario includes the cases when operators or even verticals require Analytics operations as a service for certain areas or time slots. It is then clear that, similarly to Monitoring and corresponding heterogeneous operations, Analytics has also to address multiple tasks, embed multiple functionalities, and possibly work on different time scales.

The concept of Analytics has also evolved over the years and continues to do so [3][35]. The starting point can be identified as the *descriptive analytics*, which is essentially a way to get insights on what happened in the past, in terms of network status and performance, in particular through ad-hoc visualization tools. The majority of M&A frameworks currently adopted by mobile operators provides descriptive analytics, and hence the visualized raw data have to be correlated and modeled for future use in successive, often human-driven steps, becoming extremely challenging in a 5G architecture [36]. To this end, the need for *diagnostic analytics*, which automatizes data correlations, modeling, and classification towards discovery and understanding of network behaviors and anomalies, is significantly increasing as 5G is reaching commercialization and usage. In parallel, *predictive analytics* is also becoming extremely popular and represents a significant add-on to network development and maintenance, since it enables predictions and forecasting about what might occur, based on real-time and/or stored data. Both diagnostic and predictive analytics are envisioned to make large use of machine and deep learning, data mining, and time-series analysis methodologies, as well as modeling approaches, e.g., based on game-theoretic analyses. Finally, *prescriptive analytics* will exploit the three steps above and enable AI-oriented decision-making, being able to suggest options for reconfigurations and policy changes, or even automatically actuate one of them, considering operators and system constraints.

In quest for a complete framework embedding all the above Analytics, a preliminary functional block dedicated to long-term data storage is required. Such block has to possibly exploit heterogeneous types of databases, e.g., relational, non-relational, and based on time-series, since they match differently with Analytics heterogeneous functionalities, from classification (relational and non-relational databases) to prediction and anomaly detection (time-series based databases). A multi-faceted storage block also enables a nearly straightforward deployment of some of the most important and basic Analytics operations, including data filtering, aggregation, and ordering. Such operations essentially result in a pre-processing step with respect to the following functional blocks, which are in charge of executing deeper data analysis, including cross-correlation, modeling, validation, and ML-oriented regression, classification, prediction, and detection [32][37]. When supported by high-performing and optimized usage of computational resources, the above framework automatically enables a *bottom-up* approach, which is finding increasing interest in operators and research community. This approach adopts a nearly unbiased exploration of massive amounts of data while looking for the discovery of relevant insights, in contrast to the *top-down* approach, in which the targets of the exploration are a-priori fixed, together with possible issues to be solved, thus leading to narrow down the amount and nature of explored data, which in turn possibly hinders the discovery of more profound hidden values.

Several activities are being pursued within both academia and industry aiming to develop advanced network analytics platforms, to be potentially used also within 5G systems. When focusing to the most recent proposals and implementations, it can be observed that *distributed* solutions are increasingly preferred with respect to more traditional *centralized* schemes, since these latter are hardly scalable by nature. Among several solutions, data storage and processing engines under the Apache Software Foundation umbrella have been proposed as starting implementation point [38]–[41]. To mention a few examples of network analytics platforms proposed in the literature, the open-source platform Datix [42] focuses on network traffic analysis, proposing a distributed and scalable architecture to handle in a timely manner extremely large amount of NetFlow and sFlow records, similar to the Hadoop/Hive-based solutions proposed in [43][44]. Similarly to Datix, the Blockmon platform also focuses on traffic analysis [45]. Aiming to extend Analytics functionalities beyond traffic analysis, and also to embed advanced ML and Deep Learning frameworks, such as Python-based scikit-learn, Torch, and TensorFlow libraries [46]–[48], NEC Labs Europe introduced the Net2Vec platform [37], which is a platform allowing the use of deep learning algorithms for several tasks, and also leveraging processing acceleration provided by hardware such as GPUs, thus following implementation trends in recent years. Net2Vec has been shown to implement a system creating users' profiles in a timely manner, using traces coming from a real network, unveiling that the use of deep learning techniques can outperform baseline methods, both in terms of accuracy and performance.

There also exist several platforms for Networks Data Analytics, and a short overview of some of the most popular ones is provided in the following. PNDA is an open source platform inspired by modern big data architectures [49]. It can store the data in the rawest form possible, for as long as possible, in a resilient, distributed file system. It also provides the tools to process near real-time streaming data, and to perform in-depth batch analysis on massive datasets. Another platform is Elastic Stack, formed by Elasticsearch, Kibana, Beats, and Logstash components, altogether known as the ELK Stack [50]. It allows reliable and secure data gathering from any source and format via Elasticsearch, and enables real-time visualization via Kibana. This latter also includes advanced applications such as Canvas, which allows the creation of custom dynamic infographics, and Elastic Maps, for visualizing geospatial data.

The InfluxData platform [51] provides a complete system that consists of four components:

- Telegraf is the monitoring collector, with 200+ plugins to retrieve metrics directly from the system it is running on, but also to pull metrics from third-party APIs. Among others, Telegraf supports protocols such as ICMP Ping, SNMP, NetFlow, sFlow, and syslog;
- InfluxDB is the database and storage engine, built to handle time series data;
- Chronograf is the visualization tool with predefined dashboards and a dedicated language to query InfluxDB data;
- Kapacitor is a rules engine for processing, monitoring, and alerting.

In order to complement the provided state-of-the-art analysis, we further report and describe the relevant ongoing standardization efforts involving M&A aspects, which target in particular network automation and self-reconfiguration capabilities, driven by near real-time M&A processes in Annex 1.

# 3. Monitoring & Analytics Release A Framework

This Section describes the Release A of the 5GENESIS M&A framework, and discusses how it integrates with the common reference architecture, depicted in Figure 1 and detailed in Deliverables D2.2 and D2.3 [52][53]. Furthermore, specific implementations of framework components across 5GENESIS platforms are introduced, and references to the corresponding Sections, dedicated to a more detailed description of M&A implementation, integration, and usage, are provided.

As highlighted in Figure 1, the M&A framework spans across all layers of the 5GENESIS architecture, from Infrastructure to Coordination, via MANO. In particular, IM and PM probes mainly lie at the Infrastructure layer, in order to fulfill the requirement of tracking the status of components and application performance, and thus collecting large amounts of heterogeneous parameters and data. Then, a management instance of the Monitoring system can be functionally placed at the MANO layer; the parameters scraped from the infrastructure components (i.e., physical or virtual hosts), referred as *vantage points*, are in fact redirected to a high-level monitoring tool, e.g., a Prometheus or Zabbix server, in order to undergo a first process of centralization. The Coordination layer hosts the storage utilities and the Analytics functionalities. M&A is also connected to the 5GENESIS Portal, given that the results of KPI evaluation and ML-based analyses are redirected to the interested experimenters and shown in dedicated dashboards. The process is similar to what happens for the most relevant *raw* data, which however, for visualization purposes, do not go through Analytics, and are directly exposed to the Portal.

Figure 2 depicts on a high level the Release A of the 5GENESIS M&A framework, which mainly comprises IM and PM blocks, storage utilities, and Analytics functionalities; two colors are used to highlight platform-agnostic vs. platform-specific implementations of some of the functional blocks. As illustrated in Figure 2, the main connection point with the overall architecture is the Experimental Life Cycle Manager (ELCM), developed in Task T3.8, whose main functionalities are the scheduling, composition, and supervision of experimental test cases in the platforms, as detailed in Deliverables D2.3 [53] and D3.15. On the one hand, the *Activation Plugins* represent a first ELCM-M&A interface, and allow the ELCM to activate on-demand IM and PM tools and probes across the platform, at the vantage points involved in the specific test case to be executed, e.g., the network components forming a slice. On the other hand, the *Results Collectors* are a second ELCM-M&A interface, and aim to automatize, via the ELCM, both formatting and long-term storage of the data collected during the execution of test cases.

As will be also explained in the following Sections, some of the IM tools have the ability of short-term storage, and may enable a direct connection to long-term storage utilities, without going through the ELCM *Results Collectors*. This is the main reason for the presence of the *Long-Term Storage Plugin*, which enables a further, direct link between the IM block, or part of it, and the storage utilities. Moreover, the *Raw data Visualization Plugin* enables a direct link between high-level IM tools, e.g., Prometheus or Zabbix, and the visualization software embedded into the 5GENESIS Portal, e.g., Grafana [54]. This is useful in particular for a prompt visualization of raw data captured during the experiments at some relevant vantage points. Finally, Analytics mainly retrieves data from the storage utilities as well as directly from IM and PM tools through dedicated *APIs*. This makes it possible to run advanced statistical and ML analyses, and visualize the results in the Portal.

In the following subsections, specific tools and probes adopted across the 5GENESIS Platform, and forming the aforementioned general framework, are introduced, providing thus a link to the following Sections, in which the specific components are described in detail.

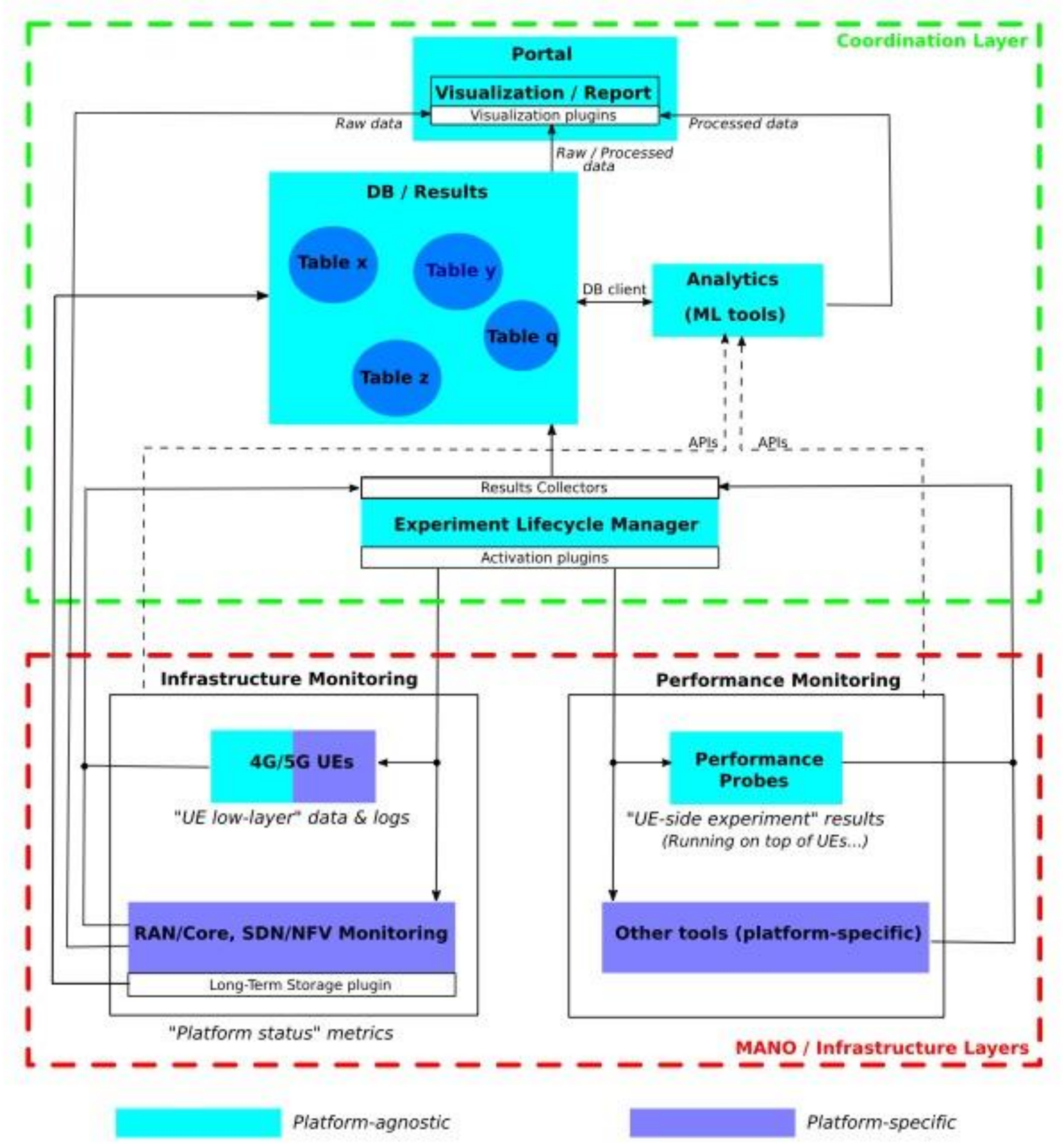


Figure 2. 5GENESIS M&A framework (Release A)

Before moving to the specific M&A components, a brief mention to the 5GENESIS ELCM is given, in order to shed light on its active participation to M&A operations.

Within 5GENESIS, the *Keysight Test Automation Platform* (TAP) software [55] deals with most of the ELCM state changes. In particular, TAP triggers and manages the instantiation of resources for a given test case to be executed, e.g., as selected by the experimenters in the Portal. Moreover, it also assists the deployment of monitoring probes for the collection of raw data, as well as the forwarding of the data to the storage utilities from where Analytics queries the data for further analyses. For this reason, as shown in Figure 3, the 5GENESIS *Activation Plugins* are in essence *TAP Plugins*. Moreover, the *Result Collectors* are *TAP Result Listeners*, which allow a lightweight data formatting and forwarding towards specific storage utilities.

In its Release A, the 5GENESIS M&A framework includes the TAP plugins for most of the IM and PM tools adopted in the platforms, i.e., Prometheus and Zabbix as high-level IM tools, and MONROE Virtual Node (VN) and Remote Agents as platform-agnostic PM probes. Moreover, a TAP Result Listener specifically designed for the 5GENESIS common-format data repository, based on the InfluxDB paradigm [56], has also been deployed. As a backup solution for the first experimentation cycle, a TAP Result Listener devoted to the creation of csv files, in which the collected metrics are stored as well, has also been developed and integrated in the framework.

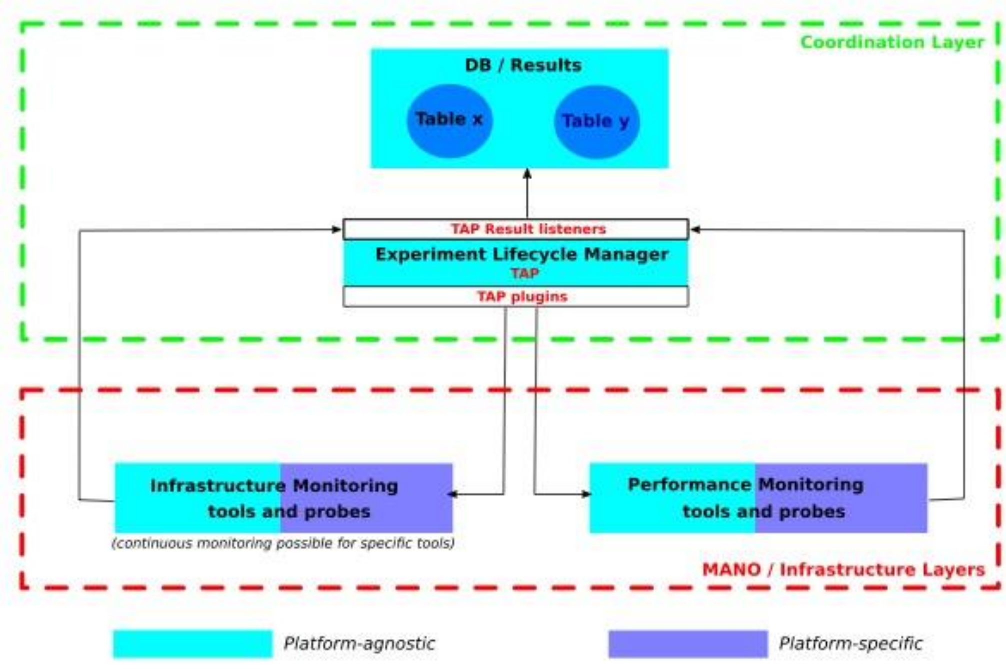

Coordination Layer
DB / Results
Table x
Table y
TAP Result listeners
Experiment Lifecycle Manager
TAP
TAP plugins
Infrastructure Monitoring tools and probes
(continuous monitoring possible for specific tools)
Performance Monitoring tools and probes
MANO / Infrastructure Layers
Platform-agnostic
Platform-specific

Figure 3. 5GENESIS M&A (Release A): Interfaces with ELCM

## 3.1. Infrastructure Monitoring

With regards to the adopted IM tools and probes, four out of five platforms, i.e., Athens, Malaga, Surrey, and Limassol opted for the use of Prometheus as high-level IM tool, with deployments and configurations described throughout Section 4.1. In the Berlin platform, Prometheus is replaced by Zabbix, due to its lightweight integration with Open Baton, the NFV MANO used in this platform; deployment and configuration of Zabbix in Berlin are described in Section 4.2. Overall, the above tools cover the monitoring of SDN/NFV instances, as well as RAN and Core/edge units, since they scrape metrics from network monitoring protocols such as

SNMP, and directly integrate on top of dedicate IM tools, such as Ceilometer, which monitors the status of OpenStack virtual environments instantiated on top of distributed physical hosts.

As regards the UEs, more specific tools are required, in particular for the collection of radio parameters, such as RSRP, SNR, RSRQ, and CQI values, experienced upon connection to the e/gNBs, as well as information on specific settings, such as the adopted Modulation and Coding Scheme (MCS). On one hand, the UEs provided by ECM under OpenAirInterface (OAI) Software Alliance [57], and being enhanced towards 5G NR within Task T3.6, will be adopted in the 5GENESIS platforms. Then, keeping this in mind, the OAI monitoring tool called T-Tracer [58], which has been originally developed for e/gNBs monitoring, is being enhanced in order to be adopted as UE monitoring tool during the next experimental cycles. Initial description of T-Tracer is thus provided in Section 8, which reports some of the activities planned for extending the M&A framework during the next development phase. On the other hand, in order to accommodate experimentations with heterogeneous devices, the platforms also use commercial 4G and 5G UEs, the latter upon availability in the near-future. Monitoring of these devices follows a different approach; in particular, a dedicated Android-based application has been used in the Athens platform during the first experimentation cycle; similarly, the Malaga platform has developed an Android Resource Agent. Both applications will be made available to the entire 5GENESIS Consortium for the next phases, and details on their preliminary development and usage are given in Sections 4.3 and 4.4, respectively. Finally, platform-specific IM extensions, such as the use of LibreNMS [59] in Athens and Limassol platforms, as well as the monitoring procedures for non-3GPP access components, such as WiFi Access Points (APs), currently under integration in the Surrey platform to demonstrate 5G multi-connectivity use cases, are given in Annex 2.

The IM components mentioned above are reported altogether in the left side blocks of Figure 4, together with the tools and probes adopted for PM (right side blocks), which are introduced in the next subsection.

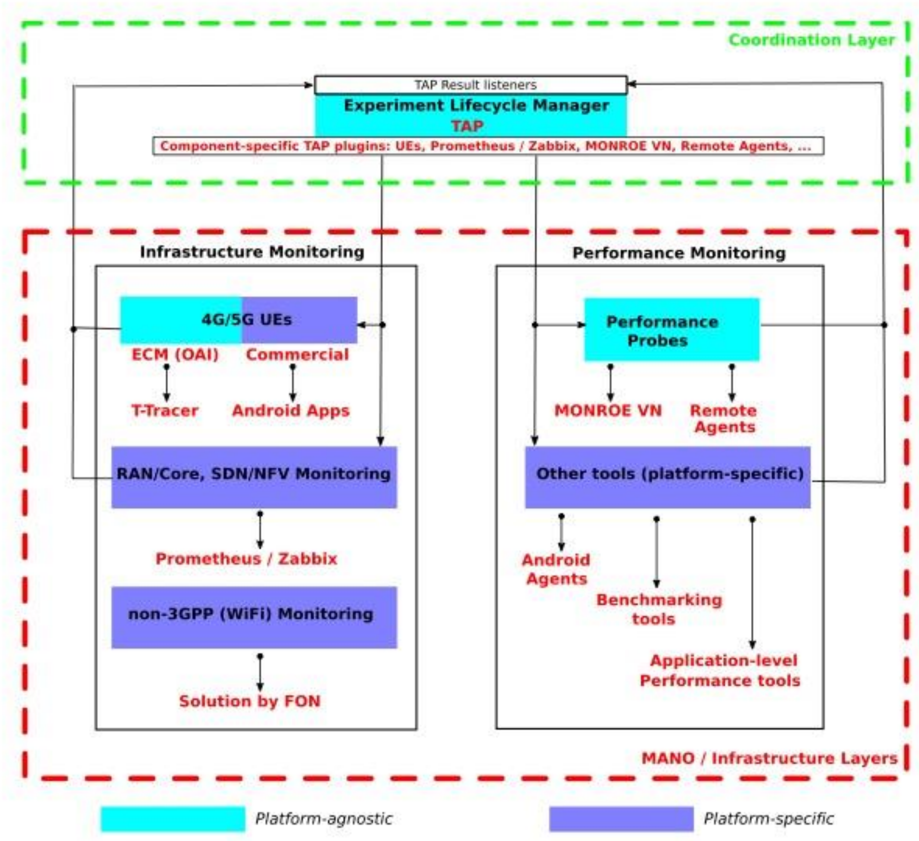

Coordination Layer
TAP Result listeners
Experiment Lifecycle Manager
TAP
Component-specific TAP plugins: UEs, Prometheus / Zabbix, MONROE VN, Remote Agents, ...
Infrastructure Monitoring
4G/5G UEs
ECM (OAI)
Commercial
T-Tracer
Android Apps
RAN/Core, SDN/NFV Monitoring
Prometheus / Zabbix
non-3GPP (WiFi) Monitoring
Solution by FON
Performance Monitoring
Performance Probes
MONROE VN
Remote Agents
Other tools (platform-specific)
Android Agents
Benchmarking tools
Application-level Performance tools
MANO / Infrastructure Layers
Platform-agnostic
Platform-specific

Figure 4. 5GENESIS M&A (Release A): IM and PM tools

## 3.2. Performance Monitoring

Focusing on PM tools and probes, several platform-agnostic instruments have been designed and developed within the scope of the 5GENESIS M&A framework. In particular, MONROE Virtual Node (VN) has been implemented as the platform-agnostic PM tool. MONROE VN takes its roots and expands the results obtained within the EU Project MONROE which is the first transnational platform for large-scale, E2E experimentation in commercial MBB networks [60][61]. MONROE platform currently is operated and maintained by the MONROE Alliance [62]. In order to comply with the 5GENESIS M&A framework, MONROE native physical nodes are being reshaped into virtual monitoring probes, thus leading to MONROE VN, which allows both in-network and E2E, hardware-transparent and on-demand PM. To support controlling MONROE VN through Keysight TAP, a TAP *agent* has been embedded into MONROE VN, in order to deploy, start, and post-process MONROE VN probes. The TAP agent exposes a REST API that can be used by the dedicated TAP plugin to provide the configurations for the specific probe to run, as detailed in Section 5.1. Moreover, further TAP-compliant probes, referred to as *Remote Agents*, have been developed and, similarly to MONROE VN, can be installed on any computer of the 5GENESIS Platform and remotely controlled via the exposed REST APIs, as summarized in Section 5.2. The initial implementation of these probes has led to the creation of monitoring solutions for latency and throughput KPIs, thus making the probes available for the first experimentation cycle, as extensively reported in Deliverable D6.1 [31]. The extension and enhancement of functionalities, e.g., for the measurement of further KPIs during specific 5GENESIS test and use cases, are targeted for the next Project cycles.

Besides MONROE VN and Remote Agents, 5GENESIS platforms currently use other PM tools for specific tests and use cases. Among others, further TAP-compliant probes tailored for the PM of Android-based devices have been developed (Section 5.3). Moreover, as regards extension planned for the next phases, the One-way Ping (OWAMP) [63] open-source client is under integration in the Athens platform, targeting the measurements of one-way latencies between hosts, as discussed in Section 8.

With regards to specific benchmarking and emulation/simulation tools, the IxChariot platform by Keysight [64] is used in the Athens platform to assess network infrastructures and deployments, as well as to simulate heterogeneous data traffic across the platform. Furthermore, the Open 5GCore Benchmarking tool[1] is used in the Berlin platform to evaluate non-functional aspects of the FhG Open5GCore, such as the emulation of users and traffic realistic behaviors. Such platform-specific PM extensions are described in Annex 3. Finally, in order to benchmark the energy efficiency KPI, which is planned to be experimentally assessed in the Surrey platform during massive IoT connectivity test cases, a simulator for the estimation of the energy consumption of NarrowBand-IoT (NB-IoT) devices is developed whose details are described in Annex 4.

## 3.3. Storage and Analytics

As briefly introduced at the beginning of this Section, the 5GENESIS Consortium has agreed on the use of InfluxDB as common tool for the creation of platform-specific instances of a long-term storage utility. InfluxDB is the open-source storage engine provided within the InfluxData framework, and handles in particular time series data. Several motivations have triggered its use in the 5GENESIS M&A framework. Among others:

- InfluxDB provides a lightweight integration with both Prometheus and Zabbix, as well as with Grafana, which is used in the 5GENESIS Portal as a core software for data visualization. The integration with Prometheus and Zabbix allows the definition of a hybrid *"pull and push"* monitoring framework. For example, while Prometheus and Zabbix servers periodically poll the metrics from their probes, i.e., Prometheus *exporters* and Zabbix *agents*, enabling in this way short-term data storage; InfluxDB works instead in a push-based fashion, so that the data are redirected towards long-term dedicated databases, that can be then easily accessed and queried at any time. InfluxDB natively supports a remote read/write protocol for Prometheus [65], and similar solutions can be also found for Zabbix [66]. Overall, such solutions replace the generic *Long-Term Storage Plugin* introduced in Figure 2, and are more specifically referred to as *Prometheus (Zabbix) / InfluxDB Plugin* in Figure 5. Regarding Grafana, a pre-existing plugin can be used so that data stored in InfluxDB instances can be directly queried and visualized in Grafana dashboards (*InfluxDB / Grafana Plugin* in Figure 5) [67]. Moreover, a second option is also possible, and is tailored for raw data visualization: it comprises the use of pre-existing and dedicated *Prometheus (Zabbix) /*

[1] https://gitlab.fokus.fraunhofer.de/5genesis/berlin-platform/tree/develop/tap-plugins/5GCore_benchmarking_tools

*Grafana Plugins* for a direct visualization, thus replacing the generic *Raw data Visualization Plugin* in Figure 2.

- InfluxDB is a key component of the overall InfluxData platform, as briefly described in Section 2.2. Among several other components, the platform also comprises the Telegraf library [68]. The latter includes a large amount of IM and PM probes, and for this reason is being considered for possible integration in the next M&A Release, in order to further diversify 5GENESIS monitoring capabilities. The Malaga platform has already initiated this integration, and some of the Telegraf plugins, focusing on memory and CPU consumption monitoring across distributed platform components, are already in use at specific vantage points of the infrastructure.

In order to create the interface between TAP and InfluxDB instances running within a 5GENESIS platform, an *InfluxDB TAP Result Listener* has been developed and integrated, allowing TAP to act as central entity so to retrieve IM and PM metrics collected at different vantage points, and redirect them at specific InfluxDB measurement tables within the database.

When compared with the possible direct connection between Prometheus (Zabbix) and InfluxDB mentioned above, the "*TAP in-the-loop*" approach allows to select, during an experiment and via predefined settings, the IM metrics to be stored into dedicated InfluxDB tables, thus focusing on IM monitoring during the experiment lifetime avoiding excessive storage of all the monitoring parameters and making the entire framework more scalable.

With regards to the Analytics, its main functionalities are based on Python, at least for this initial Project phase, considering its widespread use as data analysis tool thanks to the large number of heterogeneous libraries, available for both statistical and ML-based analytics, as well as for visualization and reporting. Furthermore, the entire 5GENESIS Portal is also being developed in Python, hence allowing a smoother integration, which is targeted for the next development cycle. The connection to the InfluxDB storage utilities is achieved through the use of a pre-existing *InfluxDB-Python client* [69], which allows both read/write connections to remote InfluxDB instances from within Python. InfluxDB data can be thus queried by Analytics via the client, with this latter basically reproducing in Python the InfluxDB native querying language, referred to as InfluxQL [70].

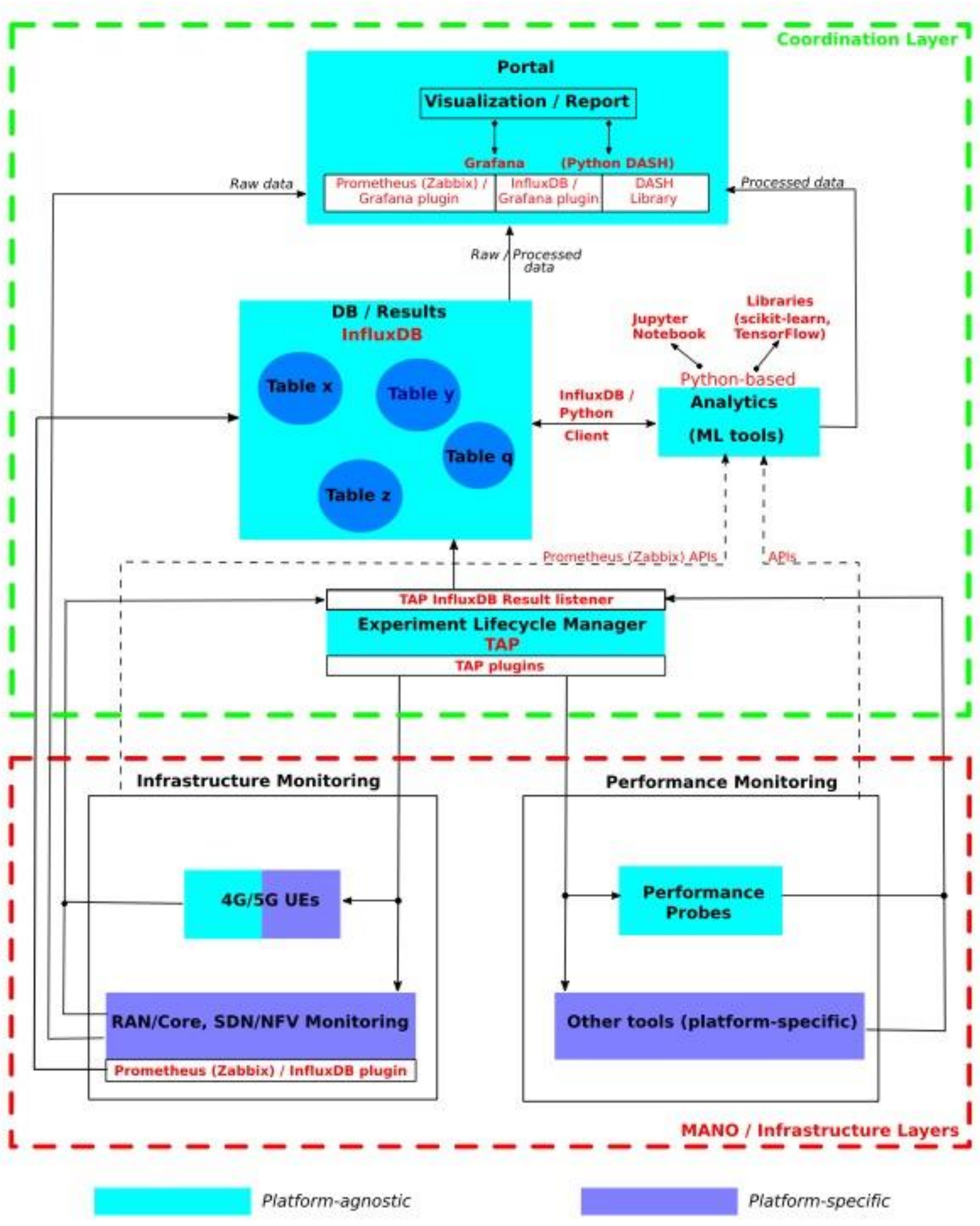


Figure 5. 5GENESIS M&A (Release A): Storage and Analytics functionalities

Once the data needed for specific analyses are retrieved, they can be managed via Python. In particular, initial analyses on the data collected during the first experimentation cycle are being carried out by means of Python-based Jupyter Notebook which is an open-source web application allowing creation and sharing of scripts which enable disparate analytics functionalities, ranging from data cleaning and manipulation to statistical and ML-based analyses, up to flexible and interactive visualization [71].

The use of the Jupyter framework makes it possible to use specific machine and deep learning Python libraries, such as pandas, scikit-learn, and TensorFlow, among others. Moreover, it also allows several extensions, e.g., towards other programming languages, such as R and Julia, and advanced big data frameworks, such as Apache Spark. Regarding the results visualization, the Python DASH library [72], which enables the creation of web-based and interactive applications for data visualization, is being considered to support the use of Grafana, since this latter is more tailored for time series visualization, while DASH would allow to extend the 5GENESIS visualization capabilities. It can be observed here that the overall definition of 5GENESIS visualization tools is, in particular, within the scope of Task T3.4, which aims at the definition of the interfaces towards experimenters and verticals. However, it is in its essence a clear intersection point for several activities within WP3, including T3.3, and for this reason is being addressed collaboratively at Consortium level. Overall, the source code of implemented Analytics algorithms is accessible to the whole 5GENESIS Consortium[2].

More details on storage and Analytics components are reported throughout Section 6, and initial usage examples are given in Section 7. Furthermore, a particular mention should be given to two possible extensions of the M&A framework in its Release A version, that is, a) the possible need of data anonymization, and b) the introduction of network automation schemes towards prescriptive analytics. As it is clear from the high-level description provided above, the M&A framework targets in particular the analysis of 5GENESIS test cases devoted to 5G KPIs measurement and validation. On the one hand, as regards the first extension (data anonymization), it is clear that its usage can be straightforwardly planned during use cases involving real users. In this case, the Analytics components will be extended in order to derive user-centric QoE KPIs, after proper data anonymization via the specific tools introduced in Annex 5. The final goal is to provide even more useful insights on the user perspective via QoS/QoE correlation analysis and QoE modelling. On the other hand, as regards the second extension (network automation), the combination of the M&A framework with so-called *policy engines*, well represented in 5GENESIS by APEX and NEAT components, is under consideration. Being the use of these two components planned in particular within the Surrey platform, with integration during the next Project phase, a short description of both is reported in Section 8, along with a list of other possible enhancements of the proposed framework towards its Release B.

[2] https://gitlab.fokus.fraunhofer.de/5genesis/analytics

# 4. IMPLEMENTATION OF INFRASTRUCTURE MONITORING

Depending on the infrastructure elements and the type of monitoring interfaces each element exposes, the 5GENESIS platforms have integrated several open-source solutions in order to gather and visualize IM information. This Section summarizes these solutions and the way they are integrated under the M&A framework.

## 4.1. Prometheus

### 4.1.1. General Description

Prometheus is an open-source service monitoring system, based on time series database that implements a highly dimensional data model, where time series are identified by a metric name and a set of key-value pairs [22]. Prometheus offers a flexible query language, allowing post-processing of collected time series data. The capability of creating alerts is also useful in order to capture specific events via filters and drive system response. Moreover, Prometheus provides a well-documented API in order to be integrated with visualization tools such as Grafana. Most importantly, Prometheus provides exporters that allow bridging of third-party data into Prometheus, including *cAdvisor* and *collectd* in a “pull” fashion, but also supports “push” through an already implemented gateway.

### 4.1.2. Integration and Configuration in 5Genesis

Prometheus is the software selected to record the real-time metrics of the virtualized services deployed in four out of five 5GENESIS platforms. The selected deployment architecture is the *Hierarchical federation*. Hierarchical federation allows Prometheus to scale to environments with tens of data centres and millions of nodes. In this use case, the federation topology resembles a tree, with higher-level Prometheus servers collecting aggregated time series data from a larger number of subordinated servers. It can be used to take measurements from any device on the platform by creating custom exporters that use the SNMP protocol. For example, a setup might consist of many per-datacentre Prometheus servers that collect data in high detail (instance-level drill-down), and a set of global Prometheus servers which collect and store only aggregated data (job-level drill-down) from those local servers. This provides an aggregate global view and detailed local views. Figure  shows the federation topology currently used in the Athens platform.

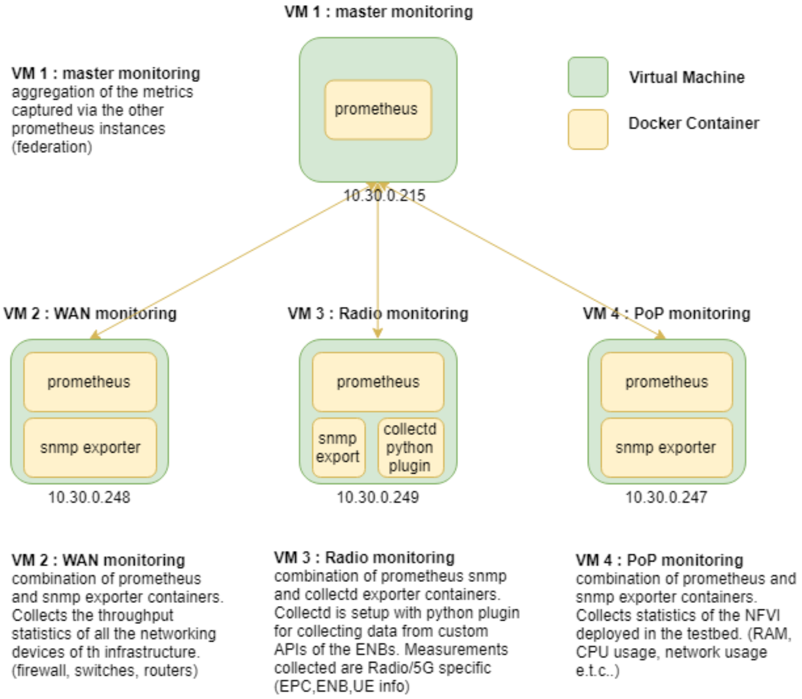
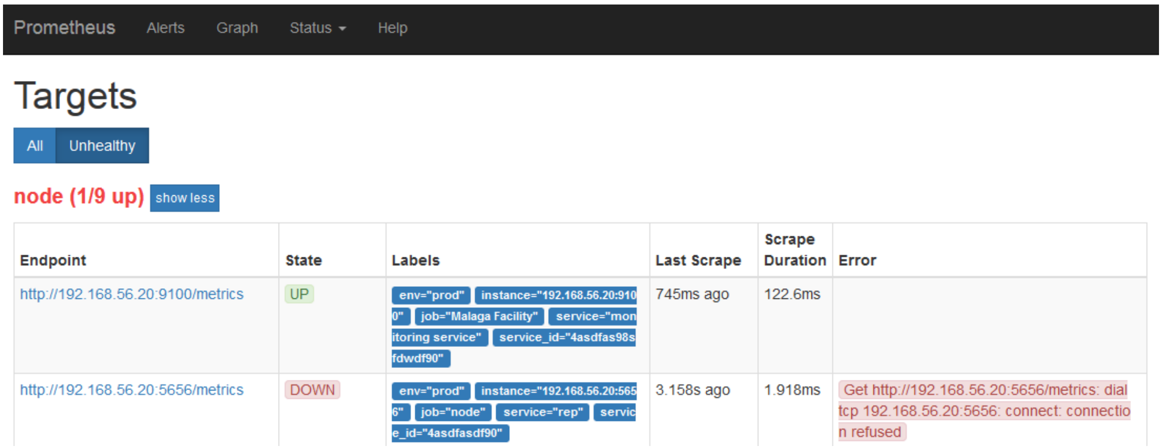

VM 1 : master monitoring
VM 1 : master monitoring
aggregation of the metrics captured via the other prometheus instances (federation)
prometheus
10.30.0.215
Virtual Machine
Docker Container
VM 2 : WAN monitoring
prometheus
snmp exporter
10.30.0.248
VM 3 : Radio monitoring
prometheus
snmp export
collectd python plugin
10.30.0.249
VM 4 : PoP monitoring
prometheus
snmp exporter
10.30.0.247
VM 2 : WAN monitoring
combination of prometheus and snmp exporter containers. Collects the throughput statistics of all the networking devices of th infrastructure. (firewall, switches, routers)
VM 3 : Radio monitoring
combination of prometheus snmp and collectd exporter containers. Collectd is setup with python plugin for collecting data from custom APIs of the ENBs. Measurements collected are Radio/5G specific (EPC,ENB,UE info)
VM 4 : PoP monitoring
combination of prometheus and snmp exporter containers. Collects statistics of the NFVI deployed in the testbed. (RAM, CPU usage, network usage e.t.c..)

Figure 6. Example of Prometheus hierarchical deployment in Athens platform

In order to start the infrastructure monitoring functionality, it is required to register in the Prometheus server the *endpoints* where the probes are deployed. The targets are listed in the Prometheus server, where all the registered endpoints are shown. Figure 7 presents the main interface of the server and the targets monitored in the infrastructure.



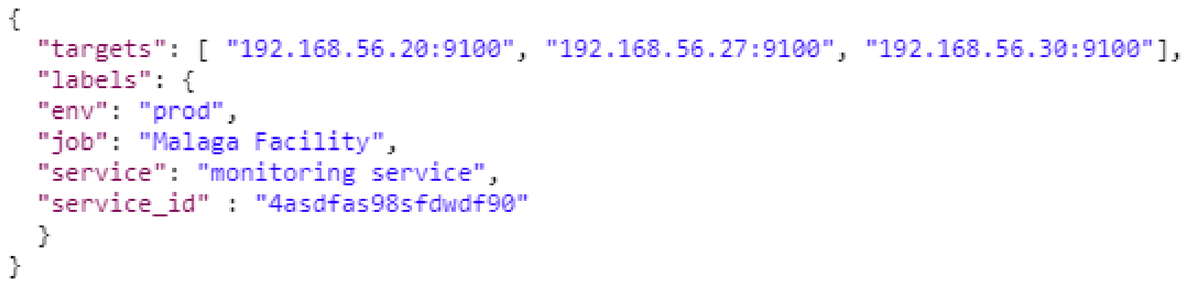
Targets

node (1/9 up)

| Endpoint | State | Labels | Last Scrape | Scrape Duration | Error |
|---|---|---|---|---|---|
| http://192.168.56.20:9100/metrics | UP | env="prod" instance="192.168.56.20:9100" job="Malaga Facility" service="monitoring service" service_id="4asdfas98sfdwdf90" | 745ms ago | 122.6ms | |
| http://192.168.56.20:5656/metrics | DOWN | env="prod" instance="192.168.56.20:5656" job="node" service="rep" service_id="4asdfasdf90" | 3.158s ago | 1.918ms | Get http://192.168.56.20:5656/metrics: dial tcp 192.168.56.20:5656: connect: connection refused |

Figure 7. Example of Prometheus server interface

The targets are registered in the system tagged with several information that will allow to filter and operate with the metrics related to the deployed services. Additional monitoring queries to conduct operations of the range vector that composes the service can be performed to analyse a list of deployed VNFs. Such functions are considered to be queried based on the *service_id* that is included in the registration process. Figure 8 presents the JSON model that includes the elements needed to register the virtual service in the Prometheus server; it is also shown that the targets can contain the IP addresses where the different VNFs composing a NS are deployed.



```
{
  "targets": [ "192.168.56.20:9100", "192.168.56.27:9100", "192.168.56.30:9100"],
  "labels": {
  "env": "prod",
  "job": "Malaga Facility",
  "service": "monitoring service",
  "service_id" : "4asdfas98sfdwdf90"
  }
}
```

Figure 8. Example of JSON model to register a virtual service in the Prometheus server

By means of *exporters*, Prometheus is able to collect heterogeneous metrics related to the infrastructure in which it is deployed. In particular, the *Node exporter* is an executable file that exposes machine resources of the physical or virtual infrastructure where it has been deployed, allowing to monitor in a distributed way a cloud-native environment [73]. The inclusion of the Node exporter in the service is performed in the deployment process, the executable file is installed in the virtual machine and launched to start the monitoring of the resources. Once the Node exporter is started, it is required to register the target (IP and port) with all the other elements to identify the service. Through the Node exporter, Prometheus read and store metrics in the internal short-term time-series database based on PromQL, for more long-term storage can be extended into a remote gateway to InfluxDB, that will allow the analysis of the infrastructure and services for a better optimization in the management of the virtualize infrastructure.

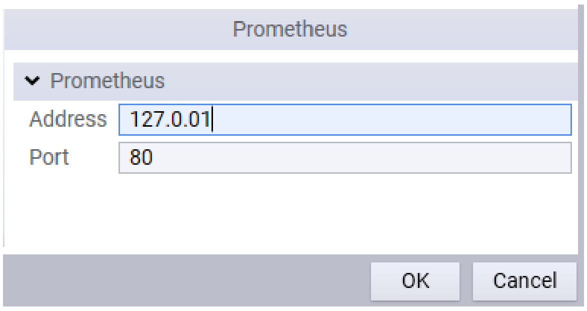
The identified parameters to measure the performance of the infrastructure that are extracted from Node exporter are listed below:

- CPU (system, user, nice, iowait, steal, idle, irq, softirq, guest): Total time the cpus spends in each mode [sec]
- Memory Load: Amount of total memory available in the system in bytes
- Disk Space Used in percent: Total Swap memory being used
- Disk Utilization per Device: Free disk usage in bytes and total size of disk
- Disk IOS per device (read, write)
- Disk Throughput per Device (read, write)
- Context Switches
- Network Traffic (In, Out): Number or sent/received bytes for each eth device
- Netstat (Established)
- UDP stats (InDatagrams, InErrors, OutDatagrams, NoPorts)

### 4.1.3. TAP Plugin

The Prometheus TAP Plugin makes use of the HTTP API in order to retrieve results from the configured instances based on a customizable PromQL query. The results obtained are published as TAP results, and thus, can be received by all of the configured TAP result listeners for further processing.
The Plugin contains two main components:

- The "Prometheus" TAP Instrument, shown in Figure 9, that encapsulates all the configuration values required for connecting with a specific Prometheus instance, as well as the basic functionality for sending requests and retrieving results using PromQL.



Prometheus

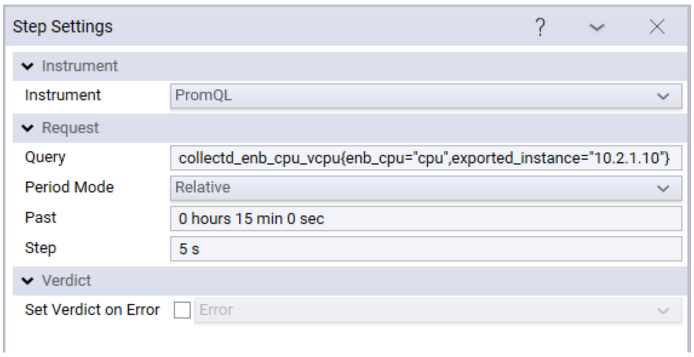
Prometheus

Address 127.0.01

Port 80

Figure 9. "Prometheus" TAP Instrument

- The "Publish Prometheus results" TAP step, show in Figure 10, that provides a way for performing requests to the Prometheus instance available to the platform administrator.



Step Settings

Instrument

Instrument PromQL

Request

Query collectd_enb_cpu_vcpu{enb_cpu="cpu",exported_instance="10.2.1.10"}

Period Mode Relative

Past 0 hours 15 min 0 sec

Step 5 s

Verdict

Set Verdict on Error □ Error

Figure 10. "Publish Prometheus results" TAP Step

### 4.1.4. Visualization

Although Prometheus embeds a tool capable of visualizing the data from multiple time-series, the 5GENESIS M&A requires more elaborate data visualization. For this reason, Grafana is exploited [54]. Grafana is an open-source visualization platform that allows to query, visualize, and alert metrics no matter where they are stored. It additionally allows to create, explore, and share dashboards that can be created dynamically based on the required information. The integration with Prometheus is fairly straightforward and all is needed is to create a dashboard providing the required information, this is useful for the operator of the platform, who can create custom dashboards with in-deep information about the platform, and for the experimenter, who have access to automatically generated dashboards with the results obtained during an experiment execution. Within 5GENESIS, different versions of these dashboards are going to be created, depending on the actual user and/or domain of the metrics (i.e., Radio, Network, etc.). Figure 11 presents an example dashboard for the Athens platform Wide Area Network (WAN) domain real-time monitoring. In particular, one can observe the traffic in/out in each core switch interface that interconnect various segment of Athens platform infrastructure. Figure 12, on the other hand, shows an automatically generated dashboard that is provided through the Portal to experimenters. This dashboard displays the results obtained during one experiment execution.

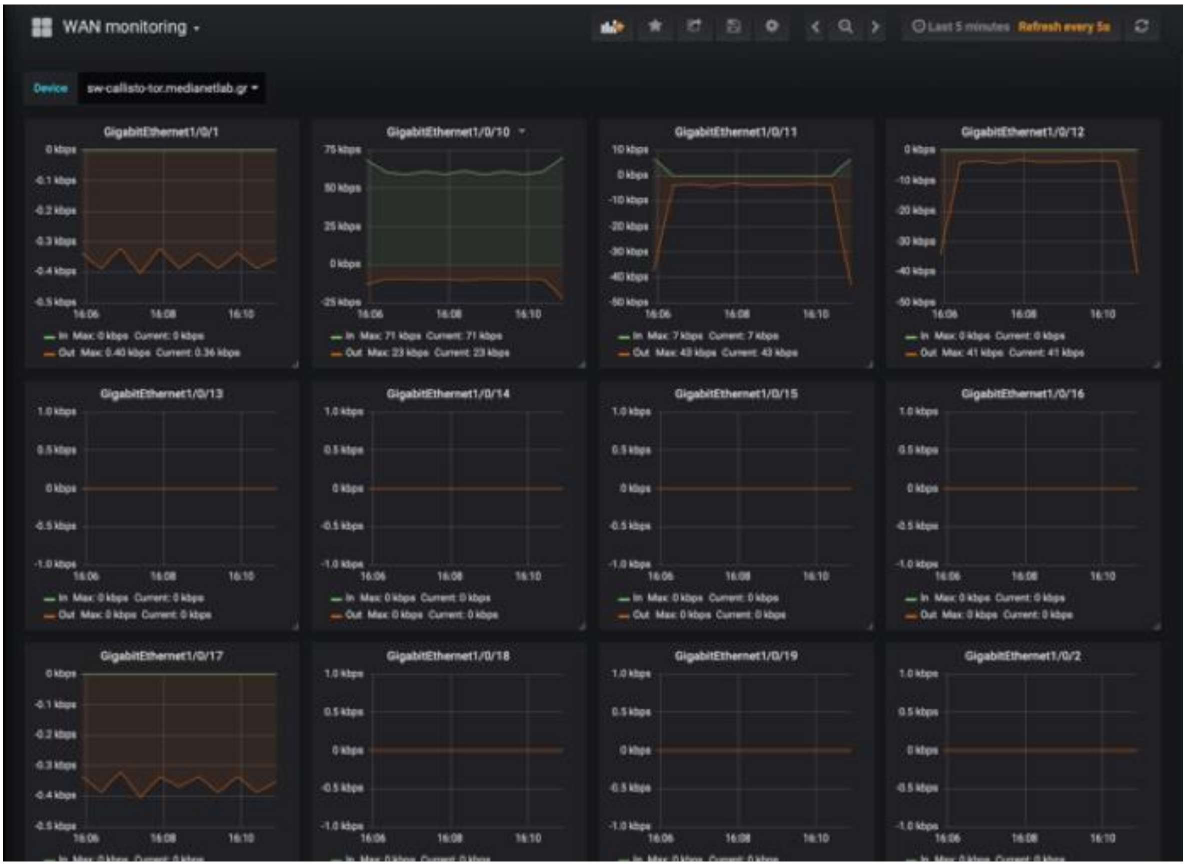
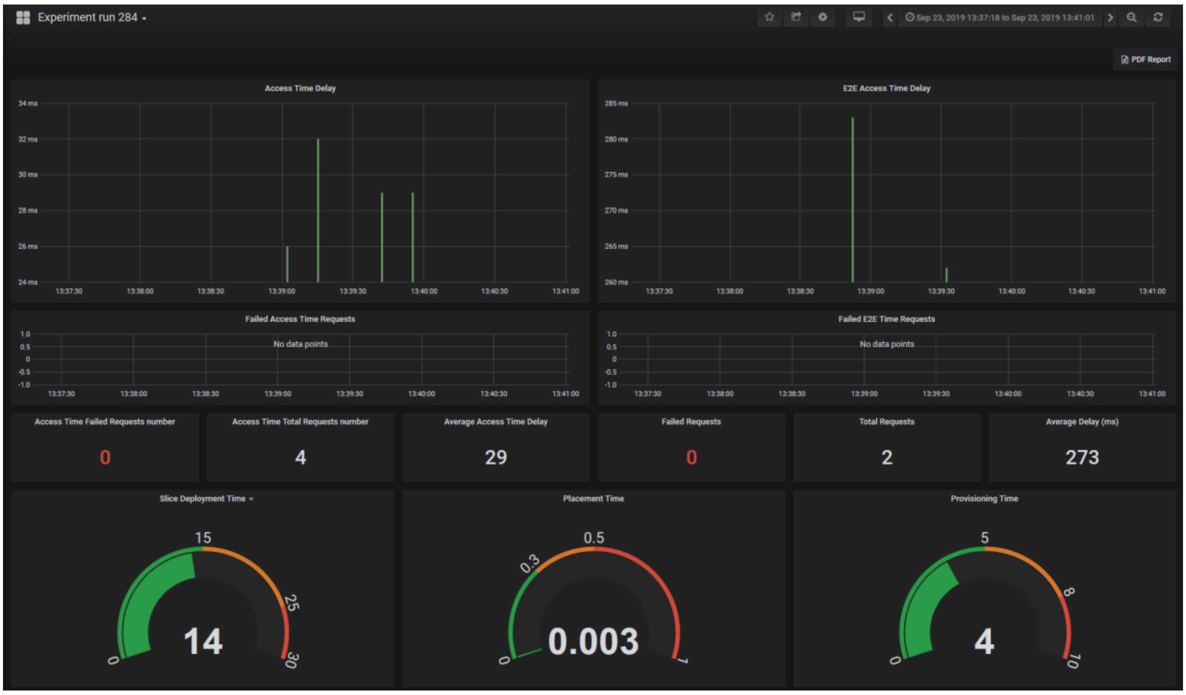


Figure 11. Example of Grafana Dashboard for WAN monitoring in Athens platform



Access Time Failed Requests number 0

Access Time Total Requests number 4

Average Access Time Delay 29

Failed Requests 0

Total Requests 2

Average Delay (ms) 273

Slice Deployment Time 14

Placement Time 0.003

Provisioning Time 4

Figure 12. Example of Grafana Dashboard generated using the results of an experiment execution

# 4.2. Zabbix

## 4.2.1. General Description

The Berlin Platform employs Zabbix to monitor network parameters, as well as data related to health and integrity of compute nodes [23]. As Zabbix supports both polling and trapping modes, it is suitable to gather information on the current status of the system as well as on configuration parameters. To gather monitoring information, Zabbix interacts with Open Baton as well as with the Open5GCore Benchmarking tools (Annex 3), as detailed in the following.

## 4.2.2. Integration and Configuration in 5Genesis

Open Baton provides a monitoring driver towards Zabbix as shown in the Figure 13. As the Open5GCore (5GC) is provided as a fully virtualized deployment, this interface allows to assess, for example, CPU consumption and memory load of the 5GC, as well as of any other deployed network function. In order to obtain additional metrics reflecting the performance of the 5GC itself, the Berlin Platform uses the Open5GCore Benchmarking tool which can expose status and performance parameters towards Zabbix. The parameters that can be monitored include number of connected users, Attachment/Detachment time, and handover execution time, to mention a few.

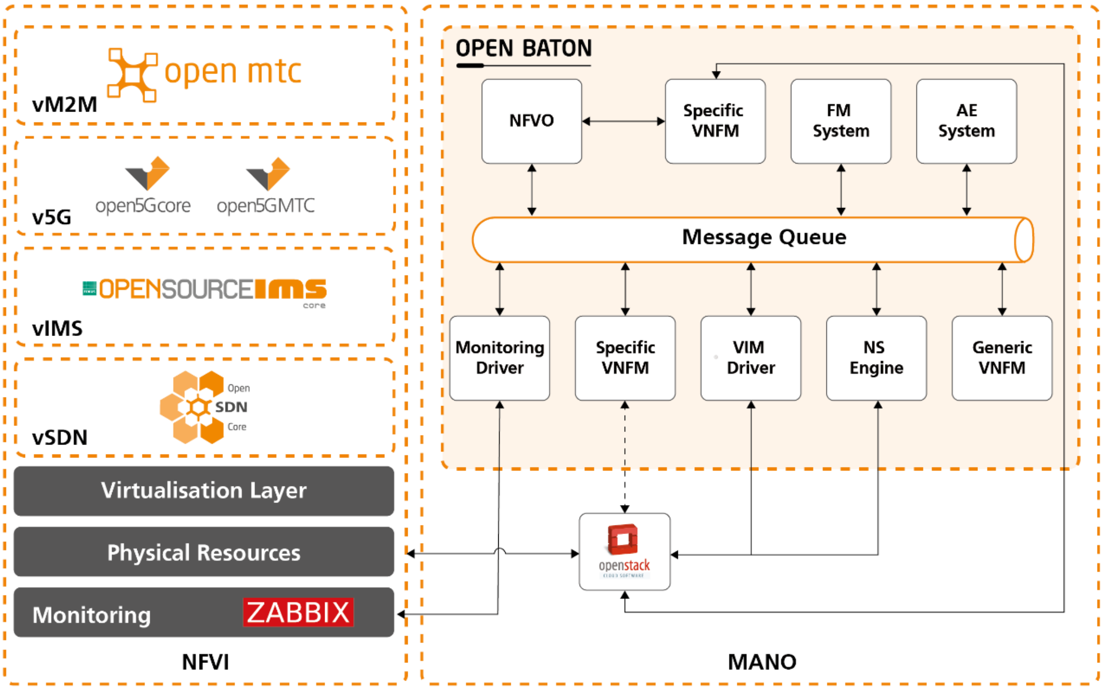


Figure 13. Zabbix interaction with Open Baton

Zabbix can also obtain information on the 5GC performance under various stress tests of the system as executed by the benchmarking tools. Figure 14 provides an example of Zabbix monitoring showing the CPU load of the system as a function of attaching 15000 subscribers to the Core at an attachment rate of 20 attachments per second.

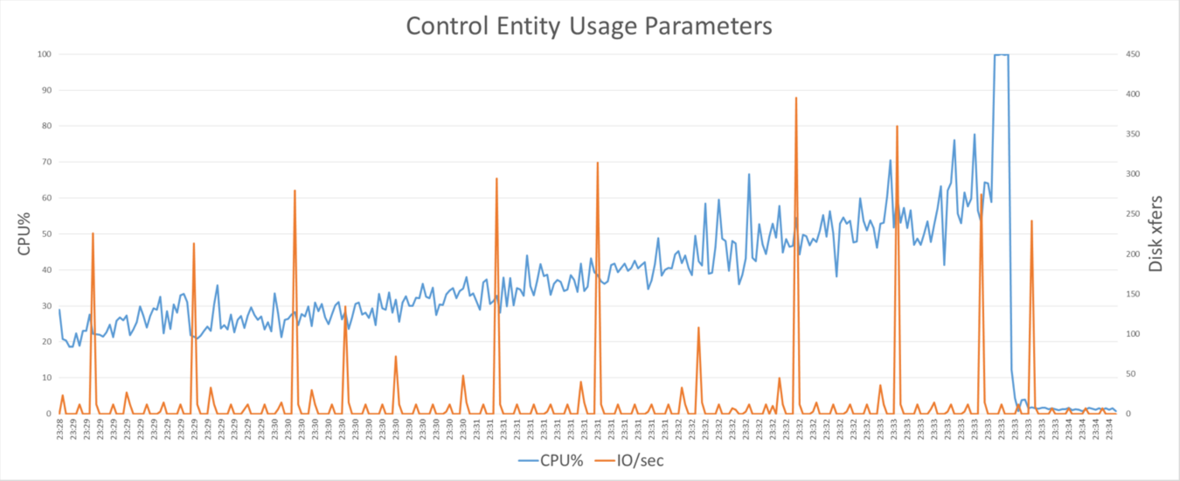


Figure 14. CPU load of Open5GsCore as a function of attaching subscribers (20 attachments per second, up to 15000 subscribers)

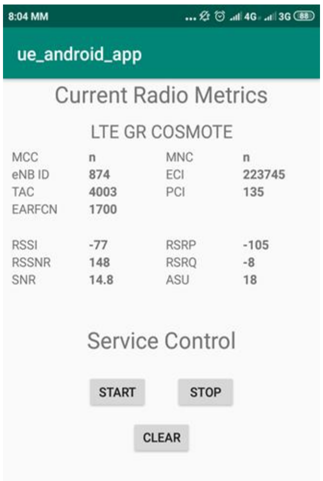
## 4.3. Android App for UE Monitoring

### 4.3.1. General Description

*UE_android_app* is a custom build Android application which provides the functionality of retrieving and storing UE monitored radio metrics. The application is built to run on any Android Operating System (OS) device and exploit OS capabilities and APIs. The purpose of this application is to provide the experimented capability to request Radio metric that are monitored and logged at the UE.

### 4.3.2. Integration and Configuration in 5Genesis

As will be discussed later in more details, 5GENESIS M&A exploits InfluxDB in order to retrieve and store monitored metrics that are produced during experiments. UE_android_app is capable to export the locally (i.e., at the UE) stored monitoring metrics to time series database such as InfluxDB, upon request or automatically. Retrieving radio metrics form the UE operating system is achieved by exploiting the telephony library, provided by Android API [74]. For local storage at the UE, utilized during the experiments, storing in local files is used instead of a native database. The main reason is simplicity in implementation and quick storage of massive amount of monitoring data. This app also has an InfluxDB client that exploits the Web API of InfluxDB sending HTTP requests in order to store any metric from the local files for further processing. The initiation and termination of the logging process is supported dynamically and may be controlled either from the experimenter with manual interaction with the UE or via the provided API, from the Coordination Layer component (i.e., ELCM). Figure 15 shows the UE_android_app UI, which provides to the user real-time radio metrics and a control panel. This control panel can be used for several operations, such as START and STOP, to start/stop recording metrics, and CLEAR, to clear all local files.



8:04 MM

ue_android_app

Current Radio Metrics

LTE GR COSMOTE

|  |  |  |  |
|---|---|---|---|
| MCC | n | MNC | n |
| eNB ID | 874 | ECI | 223745 |
| TAC | 4003 | PCI | 135 |
| EARFCN | 1700 |  |  |
| RSSI | -77 | RSRP | -105 |
| RSSNR | 148 | RSRQ | -8 |
| SNR | 14.8 | ASU | 18 |

Service Control

Figure 15. UE_android_app GUI

The preliminary version of UE_android_app is configured to expose a small subset of radio monitoring metrics which are possible to be logged via Android OS. The list of selected metrics includes RSSI, RSRQ, RSRP, SNR, CQI, introduced above, as well as Reference Signal SNR (RSSNR), Level (an abstract level value for the overall Signal Quality), Signal Strength, and AsuLevel (RSRP in Arbitrary Strength Unit (ASU), where ASU is calculated based on RSRP). Note that metrics such as CQI, Level and Signal strength in dBm are not visible through the UI, but they are collected and stored to the metric records. Each time a user starts recording metrics from the UI or API, a local file is created with a name indicating the date and time the recording started, e.g., *29_07_2019_19_45_58*, standing for date (*dd_MM_yyyy*) and time (*HH_mm_ss*). This approach offers the user the ability to execute experiments and classify them in time limits without executing timestamp queries. Each time a user makes an API call for storing to InfluxDB, a database is created with a name similar to the local file it refers. Moreover, UE_android_app API also supports the storage of every file in a single InfluxDB database (All_UE_metrics). Table 1 shows all API calls exposed by UE_android_app.

Table 1. UE_android_app API calls

| API Call | Action |
|---|---|
| http://ue_ip:8080/start | Start recording |
| http://ue_ip:8080/stop | Stop recording |
| http://ue_ip:8080/list | List all local files with logged metrics |
| http://ue_ip:8080/store_to_influxDB?fileName="<file>\|All" | Store specific file or all files to InfluxDB |
| http://ue_ip:8080/metrics_logs?fileName="file" | Get specific file in JSON format |
| ue_ip:8080/remove?fileName="file\|All" | Delete specific file or all files |

# 4.4. Android Resource Usage Agent

## 4.4.1. General Description

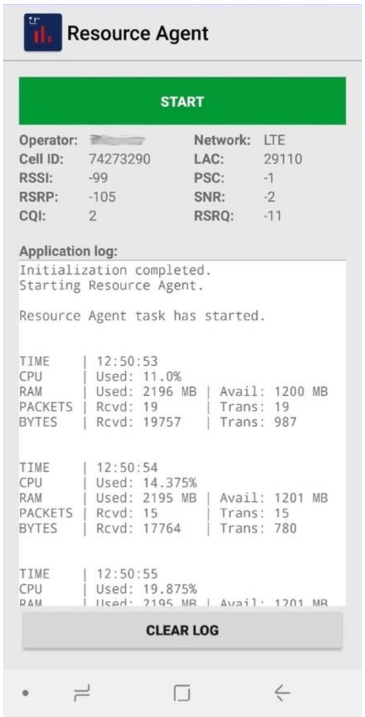
With similar goals with respect to the UE_android_app, a Resource Agent has been also developed, using Android Studio in the Java language. It can be installed on any Android device compatible with API 19 (Android KitKat). Additionally, a TAP Plugin for controlling and retrieving the results from the Resource Usage Agent has already been developed. The plugin is common to the Ping Agent, developed for PM, and for this reason is described in Section 5.2.2, after the introduction of the Ping Agent itself.

## 4.4.2. Integration and Configuration in 5Genesis

Figure 16 shows the UI of the Resource Usage Agent; the application acts as a device and network monitoring application on Android devices, and it is composed by two main parts:

- *Network interface monitoring,* keeping track of Operator, Network, Cell ID, Location Area Code (LAC), Primary Site Controller (PSC), RSSI, RSRP, SNR, CQI and RSRQ parameters;
- *Device monitoring*, keeping track of CPU usage, used and available RAM, and received and transmitted packets/bytes.

The network monitoring is updated at network change, while the device information is updated periodically, once the user presses the UI "Start" button. The update rate is between one and three seconds, depending on the device. Once the device monitoring is initiated, the logs, which include all the device and network information, are updated and tagged with POSIX timestamps. The elapsed time since the last collected information is also shown. The application requires special Android permissions that must be granted on the phone the first time the application is launched.



| | | | |
|---|---|---|---|
| Operator: | [redacted] | Network: | LTE |
| Cell ID: | 74273290 | LAC: | 29110 |
| RSSI: | -99 | PSC: | -1 |
| RSRP: | -105 | SNR: | -2 |
| CQI: | 2 | RSRQ: | -11 |

Application log:

```
Initialization completed.
Starting Resource Agent.

Resource Agent task has started.


TIME    | 12:50:53
CPU     | Used: 11.0%
RAM     | Used: 2196 MB | Avail: 1200 MB
PACKETS | Rcvd: 19      | Trans: 19
BYTES   | Rcvd: 19757   | Trans: 987


TIME    | 12:50:54
CPU     | Used: 14.375%
RAM     | Used: 2195 MB | Avail: 1201 MB
PACKETS | Rcvd: 15      | Trans: 15
BYTES   | Rcvd: 17764   | Trans: 780


TIME    | 12:50:55
CPU     | Used: 19.875%
RAM     | Used: 2195 MB | Avail: 1201 MB
```

Figure 16. Resource Usage Agent UI

# 5. Implementation of Performance Monitoring

Within the 5GENESIS M&A framework, various PM solutions have been already deployed and integrated across the platforms. Among them, MONROE VN and Remote Agents are the platform-agnostic PM probes developed for the entire 5GENESIS Platform during the first Project cycle. Thanks to their virtualized design, these probes allow to collect accurate and precise measurements while being deployed on-demand at different vantage points of the infrastructure. In particular, by doing so, they provide in-network active measurements directly comparable with E2E counterparts, given that they can be configured so to adopt the same, open-source measurement methodology at each vantage point in a given 5GENESIS platform. Furthermore, Android Agents for iPerf, Ping, and FTP performance monitoring are developed that are envisioned to be homogenized within the overall M&A framework during the next development phase. Finally, given the large number of experimental targets, several other tools are being planned for integration and usage in the overall Platform. This Section summarizes the adopted solutions and the way they are being integrated under the M&A framework.

## 5.1. MONROE VN

Several modifications have been actuated on the MONROE core SW, developed during the MONROE project, in order to integrate it within the 5GENESIS experimentation Platform.
To simplify the installation of MONROE on the heterogeneous HW platforms in 5GENESIS project, the client-side MONROE SW packages (the open source part) has been modularized and the code is rewritten so to adhere to FOSS standards and remove all dependencies on proprietary or closed source code components. Moreover, the management system has been moved to an apt/deb package (https://github.com/MONROE-PROJECT/apt-repo) from the original ansible-based system used in MONROE. The scheduling component (i.e., a TAP agent) are integrated as a natural (modular) extension into the platform. These changes allow to install MONROE on both virtual and hardware-based platforms and manage and control them via the same interface (the TAP agent).

To carry out the experiments defined in 5GENESIS, two MONROE measurement probes have been designed so far, the *ping* and *throughput containers* detailed below. The MONROE TAP agent offers the interface used to start the probes.

### 5.1.1. Ping Container

The Ping container[3] measures IP RTT by continuously sending ping packets to a configurable server (default 8.8.8.8, Google public DNS). In 5GENESIS, this container measures latency between where the MONROE VN is deployed and any other end point. The container sends one Echo Request (ICMP type 8) packet per second to a server (end point) over a specified interface until aborted. RTT is measured as the time between the Echo request and the Echo reply (ICMP type 0) is received from the server. The container is designed to run as a Docker container and does not attempt to do any active network configuration. If the interface selected for the

[3] https://github.com/MONROE-PROJECT/Experiments/tree/master/experiments/ping

experiment does not exist (i.e., it is switched off) when the experiment starts, the process will immediately exit.

The default input values are:

```
{
    “server”: “8.8.8.8”,  # ping target
    “interval”: 1000,  # time in milliseconds between successive packets
    “nodeid”: “fake.nodeid”,
    “size”: 56,
    “dataid”: “MONROE.EXP.PING”,
    “meta_grace”: 120,  # Grace period to wait for interface metadata
    “ifup_interval_check”: 5,  # Interval to check if interface is up
    “export_interval”: 5.0,
    “verbosity”: 2,  # 0 = “Mute”, 1=error, 2=Information, 3=verbose
    “resultdir”: “/40onroe/results/”,
    “modeminterfacename”: “InternalInterface”,
    “interfacename”: “eth0”  # Interface to run the experiment on
    “interfaces_without_metadata”: “eth0”  # Manual metadata on these interface
}
```

The “size” parameter corresponds to the payload of the ICMP packet. All debug/error information is printed on *stdout* depending on the “verbosity” variable. The container executes a statement similar to running *fping* like the following:

```
fping -I eth0 -D -p 1000 -l 8.8.8.8
```

The container produces a single line JSON object similar to the following (pretty printed and added comments here for readability)

### Successful reply

```
{
  "Guid": "313.123213.123123.123123", # exp_config['guid']
  "Timestamp": 23123.1212, # time.time()
  "Iccid": 2332323, # meta_info["ICCID"]
  "Operator": "Telia", # meta_info["Operator"]
  "NodeId" : "9", # exp_config['nodeid']
  "DataId": "MONROE.EXP.PING",
  "DataVersion": 2,
  "SequenceNumber": 70,
  "Rtt": 6.47,
  "Bytes": 84,
  "Host": "8.8.8.8",
 }
```

### No reply (lost interface or network issues)

```
{
  "Guid": "313.123213.123123.123123", # exp_config['guid']
  "Timestamp": 23123.1212, # time.time()
  "Iccid": 2332323, # meta_info["ICCID"]
  "Operator": "Telia", # meta_info["Operator"]
  "NodeId" : "9", # exp_config['nodeid']
  "DataId": "MONROE.EXP.PING",
  "DataVersion": 2,
  "SequenceNumber": 70,
  "Host": "8.8.8.8",
}
```

## 5.1.2. Throughput Container

The throughput container[4] uses the *iPerf* tool for active measurements of the maximum achievable bandwidth between two endpoints on IP networks. This container is designed to run on MONROE VN. The container can use either TCP or UDP as the transport protocol. The default input values are:

```
{
      "zmqport": "tcp://172.17.0.1:5556",
      "guid": "fake.guid",  # Need to be overriden
      "nodeid": "virtual",
      "metadata_topic": "MONROE.META",
      "dataid": "5GENESIS.EXP.IPERF",
      "verbosity": 2,  # 0 = "Mute", 1=error, 2=Information, 3=verbose
      "resultdir": "/monroe/results/",
      "server": "130.243.27.222",
      "protocol": "tcp",
      "interfaces": [ "eth0" ],
      "iperfversion": 3 # 2 = "iperf", 3=iperf3
}
```

The iPerf container produces a JSON object (file) per interface (and IP) configured in input "interfaces". An example of produced output is provided in Annex 6.

## 5.1.3. MONROE TAP Agent

The MONROE TAP agent is responsible to deploy and start a container in a MONROE virtual node. The agent exposes a push-based REST API for scheduling and retrieving experiment results on a MONROE node, as illustrated in Figure 27. The scheduler/agent listen by default on port 8080 on all interfaces with a self-signed certificate. To deploy, start, stop and get the experiment results a valid API key is needed.

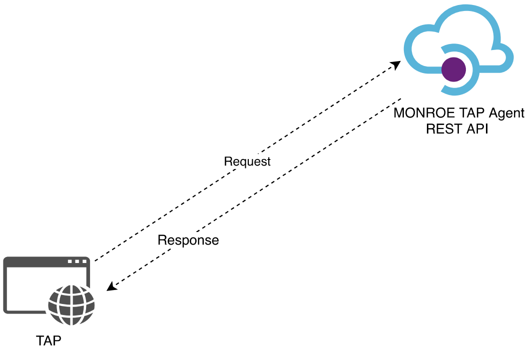


Figure 27 REST API from the TAP agent that TAP can use to control an experiment

[4] https://github.com/MONROE-PROJECT/Experiments/tree/monroe-virtual/experiments/iperf

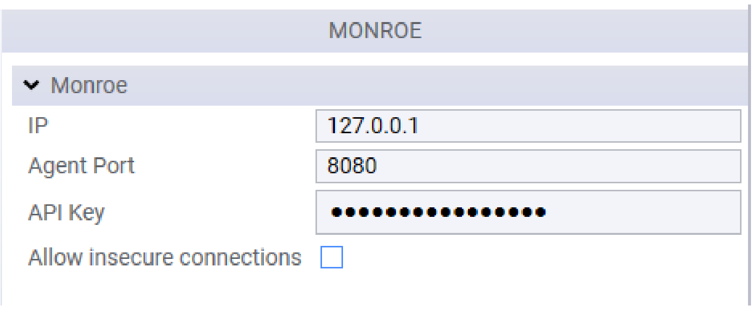
The following APIs are provided by the MONROE TAP agent. Each API is described with an endpoint and the action

| Endpoint | Action |
|---|---|
| 1. /api/v1.0/experiment/<string:schedid>/start', methods=['POST'] | Will deploy and start an experiment, e.g.:<br>curl --insecure -H 'x-api-key: $3cr3t_Pa$$w0rd!' -d '{ "script": "jonakarl/nodetest"}' -H "Content-Type: application/json" -X POST https://<URL>:8080/api/v1.0/experiment/test1/start |
| 2. /api/v1.0/experiment/<string:schedid>/stop', methods=['POST'] | Will stop (i.e. delete an experiment) and retrieve the results (as a zip file), e.g.:<br>curl --insecure -H 'x-api-key: $3cr3t_Pa$$w0rd!' -X POST https://<URL>:8080/api/v1.0/experiment/test1/stop -o test1.zip |
| 3. /api/v1.0/experiment/<string:schedid>, methods=['GET'] | Will retrieve status of a given experiment, e.g.:<br>curl https://<URL>:8080/api/v1.0/experiment/test1<br>* HTTP_200_OK --- experiment is still running<br>* HTTP_428_PRECONDITION_REQUIRED --- experiment is deployed but not running (either stopped or has not started yet)<br>* HTTP_404_NOT_FOUND --- experiment is not deployed (i.e. does not exist) |
| 4. /api/v1.0/experiment', methods=['GET'] | Will return currently running and deployed experiments, e.g.:<br>curl https://<URL>:8080/api/v1.0/experiment |
| 5. /api/v1.0/experiment/<string:schedid>', methods=['POST'] | Deploys an experiment, e.g.:<br>curl --insecure -H 'x-api-key: $3cr3t_Pa$$w0rd!' -d '{ "script": "jonakarl/nodetest"}' -H "Content-Type: application/json" -X POST https://<URL>:8080/api/v1.0/experiment/test1 |
| 6. /api/v1.0/experiment/<string:schedid>', methods=['PUT'] | Starts an experiment, e.g.:<br>curl --insecure -H 'x-api-key: $3cr3t_Pa$$w0rd!' -X PUT https://<URL>:8080/api/v1.0/experiment/test1 |
| 7. /api/v1.0/experiment/<string:schedid>', methods=['DELETE'] | Will stop aka delete an experiment, e.g.:<br>curl --insecure -H 'x-api-key: $3cr3t_Pa$$w0rd!' -X DELETE https://<URL>:8080/api/v1.0/experiment/test1 |
| 8. /api/v1.0/experiment/<string:schedid>/results', methods=['GET'] | Will sync and retrieve the current results of a experiment (as a zip file), e.g.:<br>curl --insecure -H 'x-api-key: $3cr3t_Pa$$w0rd!' https://<URL>:8080/api/v1.0/experiment/test1/results -o test1.zip |

### 5.1.4. MONROE TAP Plugin

The MONROE TAP Plugin provides the functionality of starting and stopping MONROE VN experiments, as well as of retrieving results from these experiments, publishing them as TAP results that can be further processed by the available TAP result listeners.

The MONROE plugin provides an Instrument, as shown in Figure 18, that, from the point of view of the end user, stores all the required configuration for connecting with the MONROE VN, and also encapsulates all the required logic for connecting with the REST API exposed by the TAP Agent running in the MONROE VN, described in Section 5.1.3



MONROE

Monroe

IP: 127.0.0.1

Agent Port: 8080

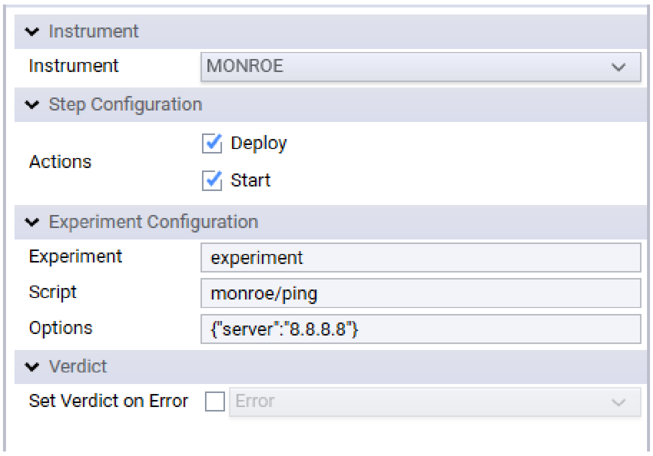
API Key: ●●●●●●●●●●●●●●●●

Allow insecure connections ☐

Figure 18. “MONROE” TAP Instrument

The plugin includes five different TAP Test steps:

- “Start Experiment”: This step is able to deploy and/or start an experiment in the MONROE VN (Figure 19).



Instrument

Instrument: MONROE

Step Configuration

Actions: ☑ Deploy ☑ Start

Experiment Configuration

Experiment: experiment

Script: monroe/ping

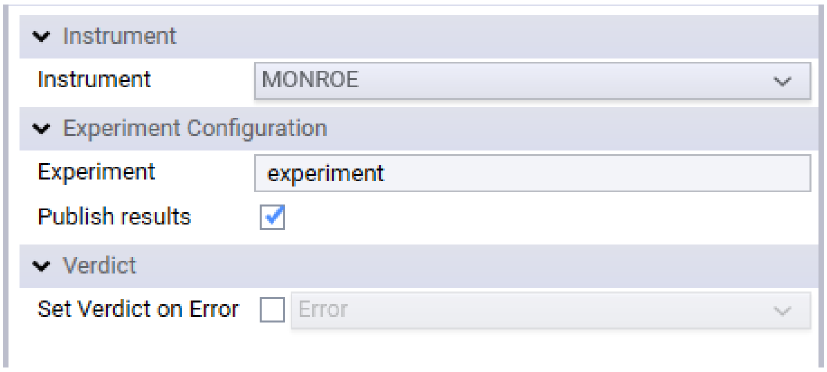
Options: {"server":"8.8.8.8"}

Verdict

Set Verdict on Error ☐ Error

Figure 19. “MONROE Start Experiment” TAP step

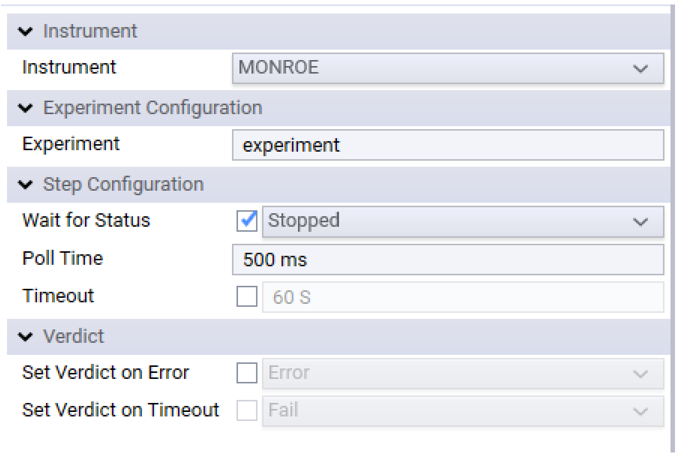
- “Stop Experiment”: This step can finalize the execution of an experiment, optionally retrieving the results generated (Figure 20).



| | | |
|---|---|---|
| Instrument | | |
| Instrument | | MONROE |
| Experiment Configuration | | |
| Experiment | | experiment |
| Publish results | ☑ | |
| Verdict | | |
| Set Verdict on Error | □ | Error |

Figure 20. “MONROE Stop Experiment” TAP step

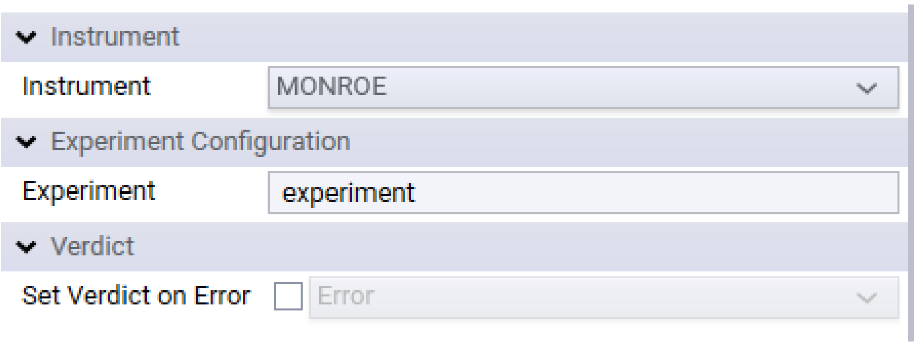
- “Experiment Status Check”: This step can be used for checking the current status of a MONROE experiment, optionally waiting until the experiment reach a particular status Figure 21).



| | | |
|---|---|---|
| Instrument | | |
| Instrument | | MONROE |
| Experiment Configuration | | |
| Experiment | | experiment |
| Step Configuration | | |
| Wait for Status | ☑ | Stopped |
| Poll Time | | 500 ms |
| Timeout | □ | 60 S |
| Verdict | | |
| Set Verdict on Error | □ | Error |
| Set Verdict on Timeout | □ | Fail |

Figure 21. “MONROE Experiment Status Check” TAP step

- “List Experiments”: This step retrieves a list of the running and scheduled experiments in the MONROE VN, sorted by status.
- “Retrieve Experiment Results”: This step can be used in order to retrieve the results from a MONROE experiment. It is possible to retrieve part of the results while the experiment is still running (Figure 22).



| | | |
|---|---|---|
| Instrument | | |
| Instrument | | MONROE |
| Experiment Configuration | | |
| Experiment | | experiment |
| Verdict | | |
| Set Verdict on Error | □ | Error |

Figure 22. “MONROE Retrieve Experiment Results” TAP step

## 5.2. Remote Agents

Complementary to MONROE VN, light-weight Remote iPerf and Pings Agents have been developed. These remote agents can be installed on any computer of the 5GENESIS Platform and be remotely controlled via the exposed REST API. Additionally, the TAP Plugin associated to the iPerf agent has been developed, and the implementation of the one for the Ping agent is currently being finalized. The TAP plugin is used to control the activation and retrieve the results through TAP. The iPerf and Pings Agents have been developed using Python 3.7 as a Flask application. In both cases, the server remains active on the computer and listens to any request received through the exposed REST API. Once a request is received the Agents start an iPerf or Ping instance using the parameters included in the request, and stores all the results generated by the iPerf or Ping processes for later retrieval by the client.

Table 2 and Table 3 show the endpoints exposed by the iPerf and Ping Agents, respectively.

Table 2. iPerf Agent endpoints

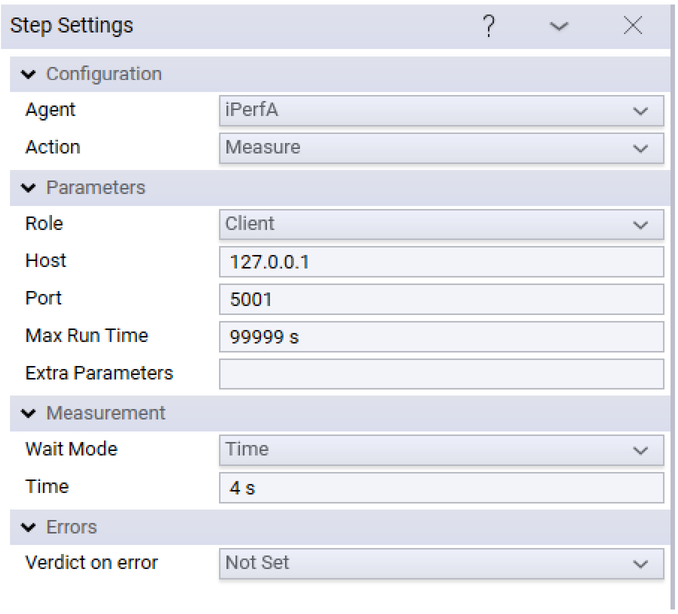
| Endpoint | Method | Description |
|---|---|---|
| /Iperf/<parameters> | GET | Starts an iPerf instance using the parameters specified. The parameters are written as a comma separated list. |
| /Iperf | POST | Starts an iPerf instance using the parameters sent as a Dictionary in the POST body. |
| /Close | GET | Stops the running iPerf instance. |
| /LastRawResult | GET | Retrieves the results from the previous execution. The returned JSON contains a list of all the text lines from the iPerf output. |
| /LastJsonResult | GET | Retrieves the results from the previous execution. The returned JSON contains a list of dictionaries with the following values:<br>• timestamp: POSIX timestamp of the instant when the measurement was taken<br>• throughput: Measured throughput in Mbps<br>• jitter<br>• packetLoss: percentage of lost packets |
| /LastError | GET | Returns a list of strings indicating the error messages from the previous execution (if any). |
| /StartDateTime | GET | Returns the time and date when the previous instance of iPerf was started. |
| /IsRunning | GET | Returns a message indicating if there is an active iPerf instance. |

Table 3. Ping Agent endpoints

| Endpoint | Method | Description |
|---|---|---|
| /Ping/<address> | GET | Starts a Ping instance using the address specified. |
| /Ping/<address>/Size/<packetSize> | GET | Starts a Ping instance using the address given with the specific number of data bytes to be sent. |
| /Close | GET | Stops the running Ping instance. |
| /LastJsonResult | GET | Retrieves the results from the previous execution. The returned JSON contains a dictionary with the following values:<br>• total: Total ping requests<br>• success: Number of requests that received a successful response<br>• icmp_replies: List of dictionaries with a POSIX timestamp, icmp sequence number, TTL, the round trip time and if it is duplicated |
| /StartDateTime | GET | Returns the time and date when the previous instance of Ping was started. |
| /IsRunning | GET | Returns a message indicating if there is an active Ping instance. |

Both iPerf and Ping Agents can be used directly by accessing the exposed REST API; however, in order to ease the integration in the different 5GENESIS platforms, dedicated TAP plugins for controlling the Agents is also being developed. In the case of iPerf, the plugin contains an instrument, referred to as *iPerfAgentInstrument*, that encapsulates all the configuration values and basic logic for connecting with a running iPerf Agent, as well as a TAP step that gives access to the user to all the functionality exposed by the Agent REST API, as shown in Figure 23.

The TAP step provides access to all the endpoints defined in Table 2, and, additionally, provides a “Measure” action. Using this action, it is possible to automate the activation and termination of the iPerf instance, as well as the retrieval of the results by using a single step. The step can perform the measurement for a specified time, or as long as the steps nested under it are running. A similar TAP Plugin for the Ping Agent is being finalized.



Agent: iPerfA
Action: Measure
Role: Client
Host: 127.0.0.1
Port: 5001
Max Run Time: 99999 s
Extra Parameters:
Wait Mode: Time
Time: 4 s
Verdict on error: Not Set

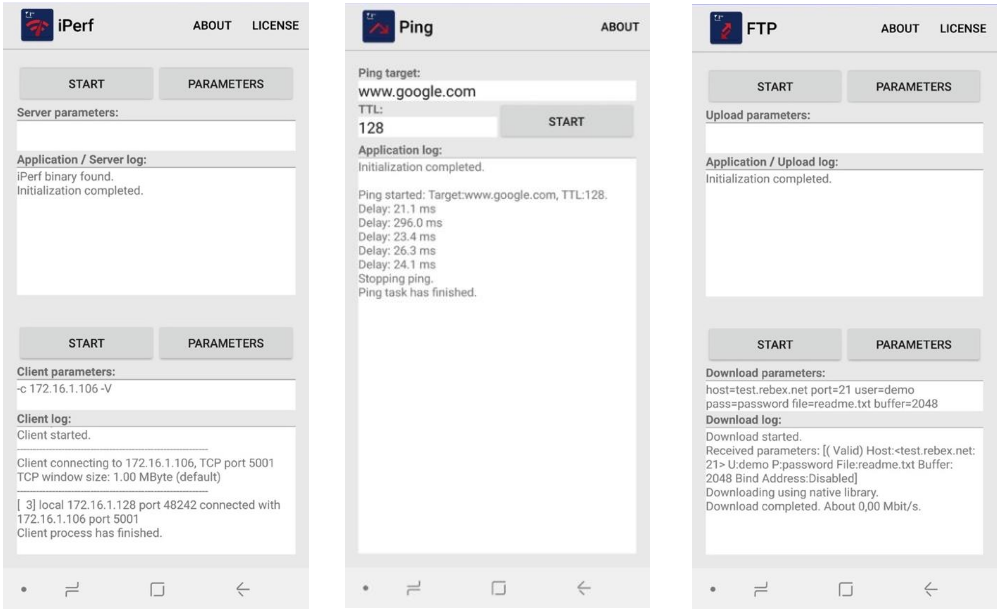
Figure 23. “iPerf Agent” TAP step

## 5.3. PM Agents for Android devices

Besides the Resource Usage Agent for UE monitoring (Section 4.4), three more Android Agents for iPerf, Ping, and FTP performance monitoring has been developed. These agents can be installed on any Android device compatible with API 15 (Android Ice Cream Sandwich), and are envisioned to be homogenized within the overall M&A framework during the next development phase. Additionally, TAP Plugins for controlling and retrieving results from the Ping Agent has already been developed while the support for iPerf and FTP Agents is planned for cycle 2.

### 5.3.1. iPerf, Ping, and FTP Android TAP Agents

The Android Agents have been developed using Android Studio in the Java language and the implemented applications act as wrappers for the probes. The applications are intended to be used as traffic generators during network performance testing. They can be used directly from the UI or remotely, making it ideal for automated testing More precisely, in the case of iPerf and FTP Agents, the applications can act as client or server, and no files are saved on the device in the case of downloads, while random data is sent towards a configured server in the case of uploads. Figures 24 shows the UI for the three Agents. The iPerf and FTP UI is divided into two parts, enabling the use of these probes for UL or DL instances, respectively. Both instances can be configured and executed by pressing the “Parameters” and the “Start” buttons, respectively. The Ping UI has text fields for configuring the Ping parameters, and it can be executed and stopped pressing the “Start” and “Stop” buttons, respectively. During the execution, the corresponding logs are updated and tagged with POSIX timestamps. Moreover, the ICMP sequence number and delay are shown for the Ping Agent, while the transfer direction is visualized for the FTP Agent. In this latter case, the application also displays information about the elapsed time and the download/upload speed, and utilizes a native library written in C language, in order to achieve the maximum transfer speeds that the device can reach.



Server parameters:

Application / Server log:

```
iPerf binary found.
Initialization completed.
```

Client parameters:

-c 172.16.1.106 -V

Client log:

```
Client started.
------------------------------------------------------------
Client connecting to 172.16.1.106, TCP port 5001
TCP window size: 1.00 MByte (default)
------------------------------------------------------------
[  3] local 172.16.1.128 port 48242 connected with
172.16.1.106 port 5001
Client process has finished.
```

(a)

Ping target:

www.google.com

TTL:

128

Application log:

```
Initialization completed.

Ping started: Target:www.google.com, TTL:128.
Delay: 21.1 ms
Delay: 296.0 ms
Delay: 23.4 ms
Delay: 26.3 ms
Delay: 24.1 ms
Stopping ping.
Ping task has finished.
```

(b)

Upload parameters:

Application / Upload log:

```
Initialization completed.
```

Download parameters:

host=test.rebex.net port=21 user=demo pass=password file=readme.txt buffer=2048

Download log:

```
Download started.
Received parameters: [( Valid) Host:<test.rebex.net:
21> U:demo P:password File:readme.txt Buffer:
2048 Bind Address:Disabled]
Downloading using native library.
Download completed. About 0,00 Mbit/s.
```

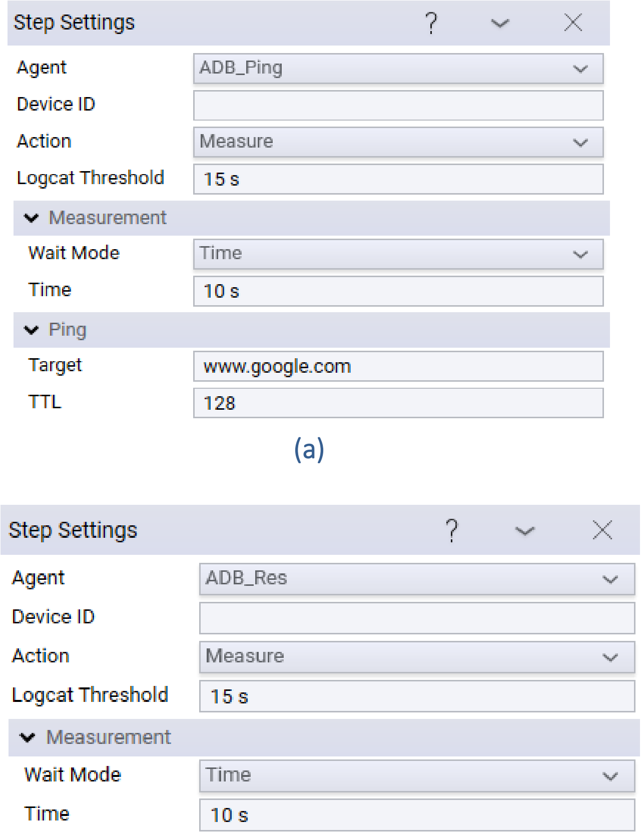
(c)

Figure 24. iPerf (a), Ping (b), and FTP (c) Agents UI

As mentioned above, all the developed applications accept remote commands for starting and stopping the probes. The output is then sent to the system log and to the application UI (if open). The output is formatted so that it can be easily read by external tools, such as the corresponding TAP instruments, as it will be detailed in the next subsection. When a remote command is received, all the applications check the configuration parameters (if these are required) before starting the probes. They also check if a probing instance is already running or not, depending on if the given command is "Start" or "Stop". Once the command is fully processed, a background task to execute the probes is initiated.

### 5.3.2. Android Agents TAP Plugin

The Android Agents described in the previous subsection can be controlled through the Android Debug Bridge (ADB) command line tool, while the measurements generated can be retrieved from Logcat, the Android logging tool, for further processing. However, in order to facilitate the usage of the Agents, a dedicated TAP Plugins that include TAP instruments and steps, for each of the Agents, is being developed. A common Plugin is currently able to control both Ping and Resource Usage Agents, and the support for iPerf and FTP Agents is expected to be released in Cycle 2. The plugin encapsulates the interaction with the different Agents, eliminating the need of using commands, and automatically retrieves all the generated results so that they can be processed by the available Result Listener. The instruments for the Ping and Resource Usage Agents, which "Settings" Steps are shown in Figures 25 (a) and (b), respectively, include a setting where the user selects the ADB Instrument to use. The ADB Instrument provides the basic functionality for controlling Android devices.



Step Settings

| | |
|---|---|
| Agent | ADB_Ping |
| Device ID | |
| Action | Measure |
| Logcat Threshold | 15 s |
| **Measurement** | |
| Wait Mode | Time |
| Time | 10 s |
| **Ping** | |
| Target | www.google.com |
| TTL | 128 |

(a)

Step Settings

| | |
|---|---|
| Agent | ADB_Res |
| Device ID | |
| Action | Measure |
| Logcat Threshold | 15 s |
| **Measurement** | |
| Wait Mode | Time |
| Time | 10 s |

(b)

Figure 25. Ping (a) and Resource Usage (b) ADB Agent TAP steps

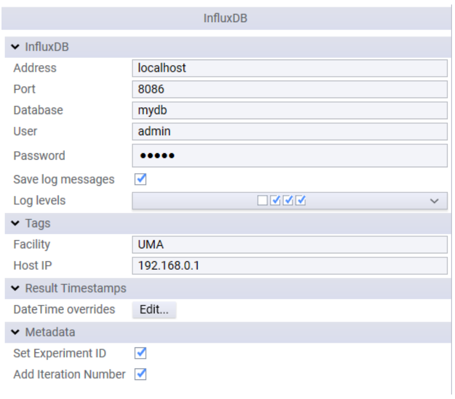
# 6. IMPLEMENTATION OF ANALYTICS

## 6.1. Data Format and Database

As anticipated in Section 3.3, several functionalities of Storage and Analytics components of the M&A framework have been designed and developed during the first 5GENESIS integration phase, and will be extended in the next ones.
Regarding the long-term storage, an InfluxDB TAP Result Listener has been developed. It allows TAP to retrieve IM and PM metrics, in the form of time series, and automatically redirect them to specific InfluxDB measurement tables within a database instance. InfluxDB time series are organized in *points*, which are composed by a time index, at least one key-value *field*, e.g., the measured values (RTT or Throughput values), and zero to many key-value *tags*, containing any possibly useful *metadata* about the field (e.g., "host = Server–01"). Points are written using the InfluxDB Line Protocol [75], which follows the following format:

```
<measurement>[,<tag-key>=<tag-value>...] <field-key>=<field-value>[,<field2-key>=<field2-value>...] [unix-nano-timestamp]
```

The InfluxDB Result Listener is able to automatically process results generated by any TAP step, converting them to the format required by InfluxDB, and sending them to the database over the network by using the Line Protocol. Some of the *tag* values can be set directly on the configuration of the Result Listener, while others (like the TAP version) will be obtained at runtime without user intervention. The Result Listener is also able to extract the timestamp in which the results were generated by parsing the "Timestamp" field, if available. If this is not the case, the user can specify for each result, which fields contain the time information of the results and in which format. Other metadata useful for the Analytics functionalities, like the experiment execution ID or the iteration number, can be easily added to the results by using this Result Listener jointly with two special test steps, as shown in Figure 26. More information about the design and functionality of the InfluxDB Result Listener are reported in the dedicated Deliverable D3.15.



InfluxDB

InfluxDB
Address localhost
Port 8086
Database mydb
User admin
Password •••••
Save log messages ☑
Log levels □☑☑☑

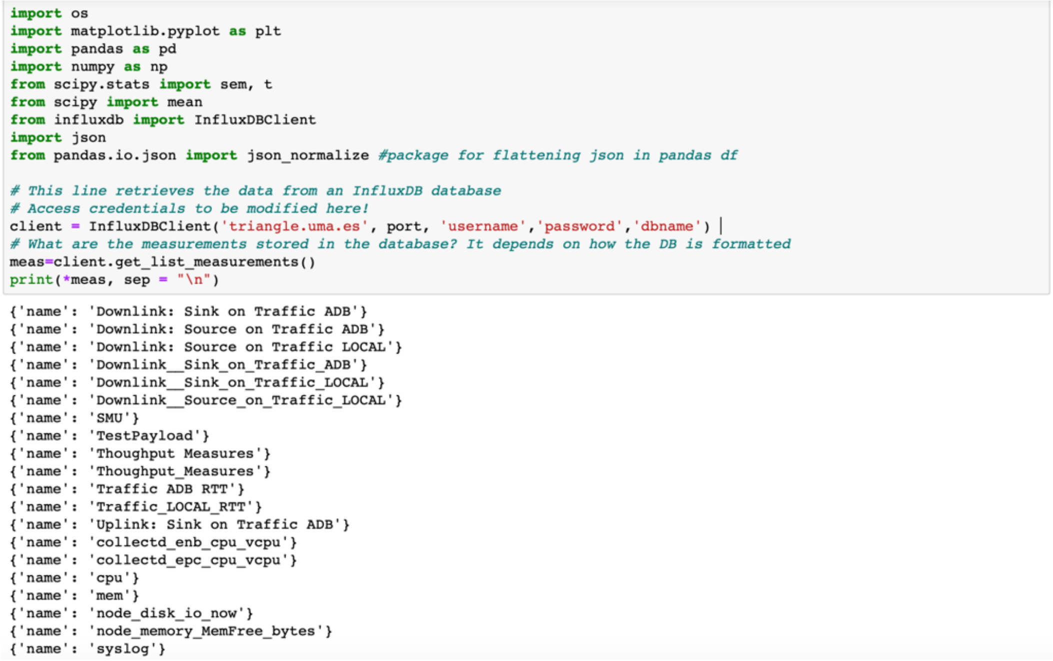
Tags
Facility UMA
Host IP 192.168.0.1

Result Timestamps
DateTime overrides Edit...

Metadata
Set Experiment ID ☑
Add Iteration Number ☑

Figure 26. InfluxDB Result Listener settings

Analytics can then query and read (but also write, if needed) the data needed for analyses via the open source *InfluxDB-Python client* [69]. As visual examples, Figure 27 shows how to enable via the client a remote connection to the InfluxDB instance running in the Malaga platform, and query for the database composition, in terms of existing measurement tables.



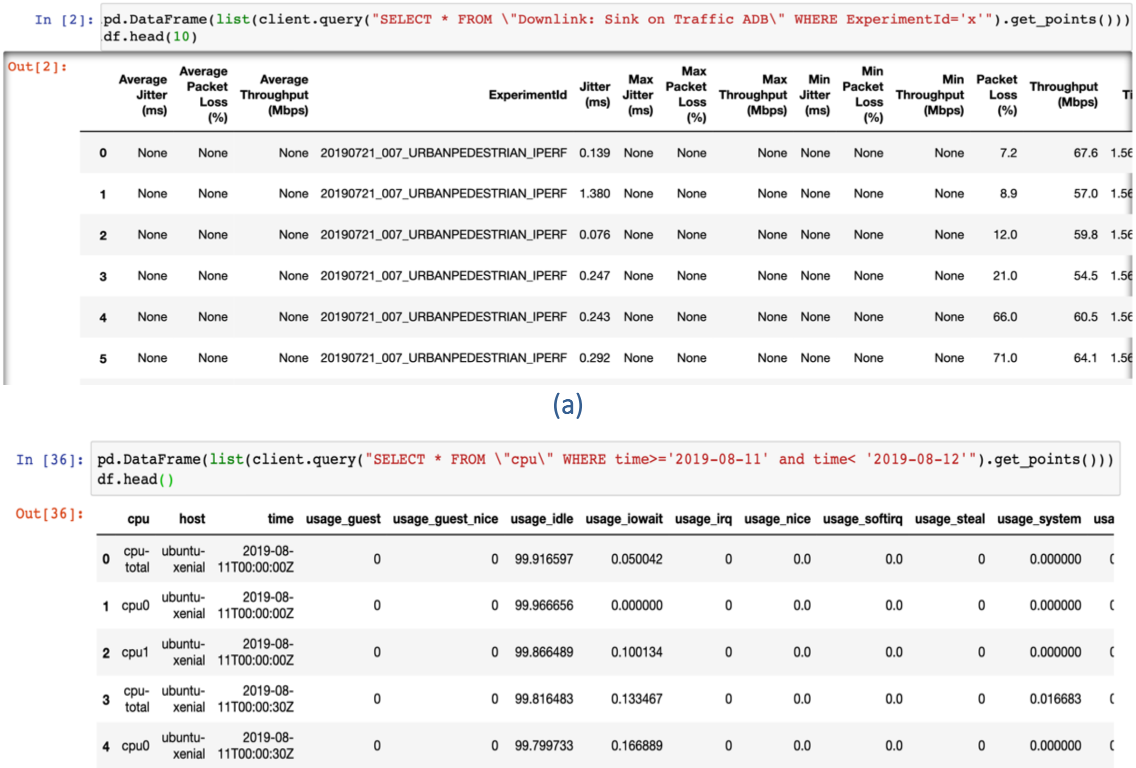
```
import os
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from scipy.stats import sem, t
from scipy import mean
from influxdb import InfluxDBClient
import json
from pandas.io.json import json_normalize #package for flattening json in pandas df

# This line retrieves the data from an InfluxDB database
# Access credentials to be modified here!
client = InfluxDBClient('triangle.uma.es', port, 'username','password','dbname') |
# What are the measurements stored in the database? It depends on how the DB is formatted
meas=client.get_list_measurements()
print(*meas, sep = "\n")
```

```
{'name': 'Downlink: Sink on Traffic ADB'}
{'name': 'Downlink: Source on Traffic ADB'}
{'name': 'Downlink: Source on Traffic LOCAL'}
{'name': 'Downlink__Sink_on_Traffic_ADB'}
{'name': 'Downlink__Sink_on_Traffic_LOCAL'}
{'name': 'Downlink__Source_on_Traffic_LOCAL'}
{'name': 'SMU'}
{'name': 'TestPayload'}
{'name': 'Thoughput Measures'}
{'name': 'Thoughput_Measures'}
{'name': 'Traffic ADB RTT'}
{'name': 'Traffic_LOCAL_RTT'}
{'name': 'Uplink: Sink on Traffic ADB'}
{'name': 'collectd_enb_cpu_vcpu'}
{'name': 'collectd_epc_cpu_vcpu'}
{'name': 'cpu'}
{'name': 'mem'}
{'name': 'node_disk_io_now'}
{'name': 'node_memory_MemFree_bytes'}
{'name': 'syslog'}
```

Figure 27. Example of remote connection to the Malaga platform InfluxDB instance

Figure 28 shows two examples of queries for specific data within a particular measurement table. As shown, data can be queried by Experiment ID (Figure 28 (a)) or by time (Figure 28 (b)), identifying in this latter case a particular observation interval to be further investigated. Other methods are available, following InfluxQL querying rules.



```
In [2]: pd.DataFrame(list(client.query("SELECT * FROM \"Downlink: Sink on Traffic ADB\" WHERE ExperimentId='x'").get_points()))
        df.head(10)
```

Out[2]:

| | Average Jitter (ms) | Average Packet Loss (%) | Average Throughput (Mbps) | ExperimentId | Jitter (ms) | Max Jitter (ms) | Max Packet Loss (%) | Max Throughput (Mbps) | Min Jitter (ms) | Min Packet Loss (%) | Min Throughput (Mbps) | Packet Loss (%) | Throughput (Mbps) | Ti |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | None | None | None | 20190721_007_URBANPEDESTRIAN_IPERF | 0.139 | None | None | None | None | None | None | 7.2 | 67.6 | 1.56 |
| 1 | None | None | None | 20190721_007_URBANPEDESTRIAN_IPERF | 1.380 | None | None | None | None | None | None | 8.9 | 57.0 | 1.56 |
| 2 | None | None | None | 20190721_007_URBANPEDESTRIAN_IPERF | 0.076 | None | None | None | None | None | None | 12.0 | 59.8 | 1.56 |
| 3 | None | None | None | 20190721_007_URBANPEDESTRIAN_IPERF | 0.247 | None | None | None | None | None | None | 21.0 | 54.5 | 1.56 |
| 4 | None | None | None | 20190721_007_URBANPEDESTRIAN_IPERF | 0.243 | None | None | None | None | None | None | 66.0 | 60.5 | 1.56 |
| 5 | None | None | None | 20190721_007_URBANPEDESTRIAN_IPERF | 0.292 | None | None | None | None | None | None | 71.0 | 64.1 | 1.56 |

(a)

```
In [36]: pd.DataFrame(list(client.query("SELECT * FROM \"cpu\" WHERE time>='2019-08-11' and time< '2019-08-12'").get_points()))
         df.head()
```

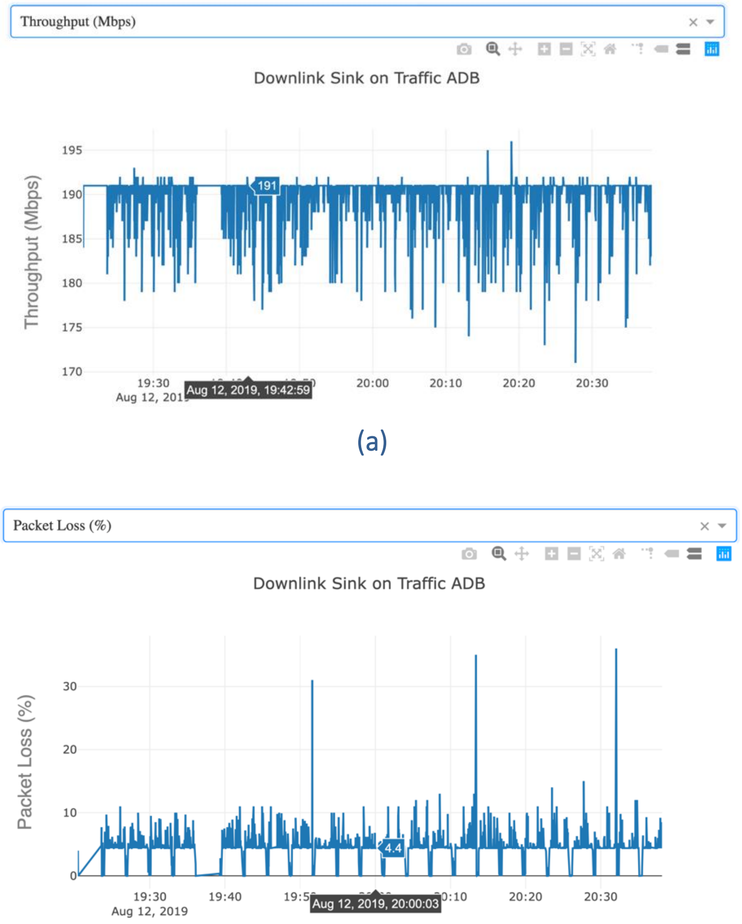
Out[36]:

| | cpu | host | time | usage_guest | usage_guest_nice | usage_idle | usage_iowait | usage_irq | usage_nice | usage_softirq | usage_steal | usage_system | usa |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | cpu-total | ubuntu-xenial | 2019-08-11T00:00:00Z | 0 | 0 | 99.916597 | 0.050042 | 0 | 0.0 | 0.0 | 0 | 0.000000 | ( |
| 1 | cpu0 | ubuntu-xenial | 2019-08-11T00:00:00Z | 0 | 0 | 99.966656 | 0.000000 | 0 | 0.0 | 0.0 | 0 | 0.000000 | ( |
| 2 | cpu1 | ubuntu-xenial | 2019-08-11T00:00:00Z | 0 | 0 | 99.866489 | 0.100134 | 0 | 0.0 | 0.0 | 0 | 0.000000 | ( |
| 3 | cpu-total | ubuntu-xenial | 2019-08-11T00:00:30Z | 0 | 0 | 99.816483 | 0.133467 | 0 | 0.0 | 0.0 | 0 | 0.016683 | ( |
| 4 | cpu0 | ubuntu-xenial | 2019-08-11T00:00:30Z | 0 | 0 | 99.799733 | 0.166889 | 0 | 0.0 | 0.0 | 0 | 0.000000 | ( |

(b)

Figure 28. Examples of querying methods available via the InfluxDB-Python client: by Experiment ID (a) and by time interval (b)

As clear from the above Figures, data retrieved from the database are directly converted to pandas *Dataframes*, so that they are ready for further processing and visualization via Python ML libraries. As an example, Figure 29 shows how some of the retrieved data look in a Python DASH-based web application, which is planned to be extended and integrated in the 5GENESIS Portal.


Throughput (Mbps)

(a)

Packet Loss (%)

(b)

Figure 29. Examples of data time series visualization via Python DASH Library

## 6.2. Statistical Analysis of KPIs

The analysis of the 5G KPIs is based on the results collected in the conducted experiments. An experiment consists of one or more test cases, which contain multiple iterations of a single test. The statistical properties of a single test are calculated from the measurements collected in the test. The statistical properties of a complete test case are obtained by taking the average of the corresponding properties of the test iterations in the test case. This results in a collection of normally distributed test case results whose averages will be close to the real value of the statistical property. Furthermore, it allows to specify confidence intervals for them using the Student-T distribution. A more detailed description of the methodology used for the analysis can be found in Deliverable D6.1 [31] and the code to calculate the statistical properties are listed in 5GENESIS Gitlab repository[5,6].

[5] https://gitlab.fokus.fraunhofer.de/5genesis/berlin-platform/tree/develop/experiments/phase1/scripts

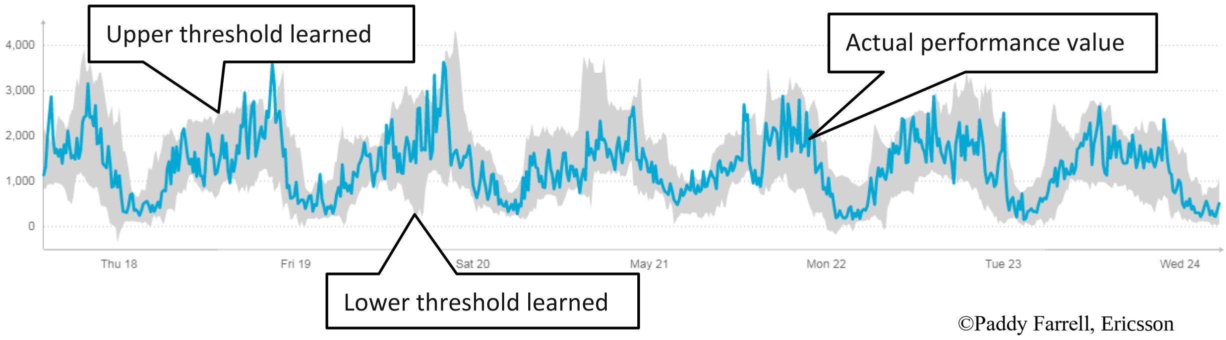
[6] https://gitlab.fokus.fraunhofer.de/5genesis/analytics

## 6.3. Machine Learning based Analytics

In order to provide more advanced analytics for 5GENESIS platforms, the M&A framework integrates several ML-based analytics for several use cases, including anomaly detection, KPI performance modeling and prediction, as well as analysis of correlations and causalities between the monitoring metrics and KPIs. In the following, a brief description of the state-of-the-art algorithms that are planned to investigate for the different use cases is provided.

### 6.3.1. Anomaly Detection

Monitored KPIs, such as latency, throughput and energy efficiency, might occasionally drop (or peak) outside of the expected limits, which could indicate issues in the network or previously unseen user behavior. Using anomaly detection to recognize when a monitored KPI is out of the expected range is important for fast root cause analysis and mediation or adaptation. Classic methods that are suitable for outlier detection on time series are based on Median Absolute Deviation (MAD) and Auto-Regressive Integrated Moving Average (ARIMA).



©Paddy Farrell, Ericsson

Figure 30. Time series forecasting algorithms can be used to learn the likely upper and lower bounds for each recurring time step (hourly, daily, weekly, …)

*Median Absolute Deviation (MAD)*

The Median Absolute Deviation calculates how much a given data point deviates from the median of observed values. Here, the median and double median are used instead of the mean and standard deviation to describe sample distribution statistics. The latter is affected by outliers, whereas the median-based approach is a robust approach for outlier detection. The MAD can be efficiently calculated by determining the double median of a distribution, i.e. the median of the absolute deviations from a distribution's median. The MAD can then be used to determine the range that contains most of the observed values. A value that lies outside that range is considered an outlier. Using the data points from past observations, the MAD can be used to determine the upper and lower thresholds of expected values for every hour, day or week, as shown in Figure 30.

*Seasonal Auto-Regressive Integrated Moving Average (SARIMA)*

SARIMA is a method to forecast seasonal time series, such as hourly, daily or weekly time periods. SARIMA uses past observations to predict the normal range of the next data points. The seasonal aspect requires that the algorithm has seen numerous examples of complete periods. For example, in the case of weekly anomaly detection, the algorithm needs to be trained on several complete weeks' worth of data. It then predicts the temporal behavior of the next time period (e.g. week) as exemplified in Figure 30. The confidence of the prediction can be used as the bounds for the anomaly detection aspect: a data point outside of the learnt limits represents an anomaly.

*Recurrent Neural Network (RNN)*

Depending on the volume of training data, Deep Learning algorithms are also being considered, in particular for anomaly detection on time series data. However, for Deep Learning algorithms to be effective, they require to be trained on large volumes of data, which may not be in the scope of 5GENESIS test cases. Recurrent Neural Networks (RNN) are a class of neural networks in the area of Deep Learning. Contrary to many traditional neural networks, RNNs do not require fixed-sized input and output vectors or a fixed number of computational steps. Therefore, they can process sequences of vectors, as in the case of time series analysis for anomaly detection. The RNN can be used to build a prediction model from current and past values in order to predict the next steps in a time series. The prediction error between actual data point and predicted data point can then be used as an indication of anomaly because the actual data point is not near the expected value. Alternatively to the method above, which does not require a priori knowledge of what constitutes abnormal values, RNNs can also be used in a supervised fashion to learn a classification between normal and abnormal values. But this only works if there is training data available that contains known (i.e., labelled) anomalies.

### 6.3.2. Prediction of Network Requirements and KPI Degradation

The focus here is on predicting how a change in the system might influence a variable's behavior. For example, a use case for prediction can be to estimate the required slice specifications (e.g., adding or removing network elements) to guarantee acceptable throughput, latency and capacity values. Other potential use cases include prediction of KPI degradation upon network element (re-)configuration and prediction of the effect of UE speed for efficient beam forming. For the prediction task, traditional Machine Learning algorithms such as Decision Trees, Random Forests and Support Vector Machines (SVM) are considered.

*Decision Tree (DT)*

Intuitively, Decision Tree learning tries to divide data so that every branch in the tree represents a split in the data according to learnt thresholds in the given features. The algorithm creates the tree recursively by learning which features hold the most information about the observed data. In a simple example, a DT algorithm might have learnt from the observed data that a network configuration was in most cases optimal (or "good") where the latency KPI was under 10ms and the capacity KPI was over 1000 UEs. Even if a latency of under 10ms was measured, if there were less than 1000 UEs present, the network configuration was labelled as "bad". While Decision Trees are not as powerful as some other algorithms (see below), they have the advantage that the learnt model can be visually presented in a human-readable format as shown in the example in Figure 31.

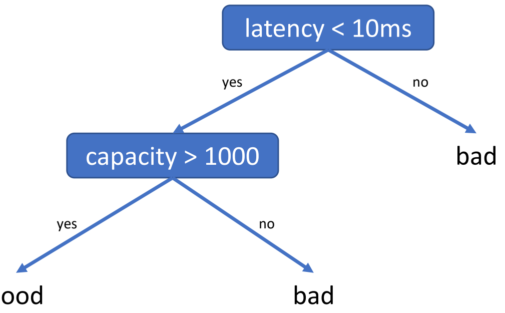


Figure 31. An example of Decision Tree that learnt to classify "good" and "bad" network configurations using the two KPIs latency and capacity

*Random Forest (RF)*

A Random Forest is an extension of the Decision Tree learning method. A (reasonably large) number of decision trees are learnt on different features and different portions of the data. This is called bootstrap aggregation, or "bagging". The results of the individual trees are then combined in a majority vote fashion to achieve better accuracy and less overfitting compared to simpler approaches like the Decision Trees. The tradeoff is a more complex model that cannot be visualized and takes longer to learn.

*Support Vector Machine (SVM)*

A Support Vector Machine model tries to separate observed data points in a high-dimensional space so that they can be assigned to different classes based on which side of the separation they are. This is a powerful learning model that can handle linear but also non-linear classification (and regression) tasks, based on the selected kernel, or similarity function.

### 6.3.3. Correlation and Causality Analysis for KPI Dependence

Correlation is a statistical association between observed variables. Correlation techniques can reveal similar, or in extreme cases identical, behavior between KPIs or other monitored variables. A high positive correlation is the most intuitive case, where two variables exhibit the same nature of change (increase and decrease). However, if two variables show opposing change (one always increases while the other decreases), they are also similar. This is called negative correlation. In a practical example, correlation analysis allows to test how strongly energy efficiency and throughput are depending on the density of users. A low correlation score might indicate that factors other than the monitored variables should be considered to determine energy efficiency and throughput. In the 5GENESIS context, several correlation and time series comparison techniques are under investigation, including Pearson and Spearman correlation, Dynamic Time Warping and clustering such as K-means and graph clustering.

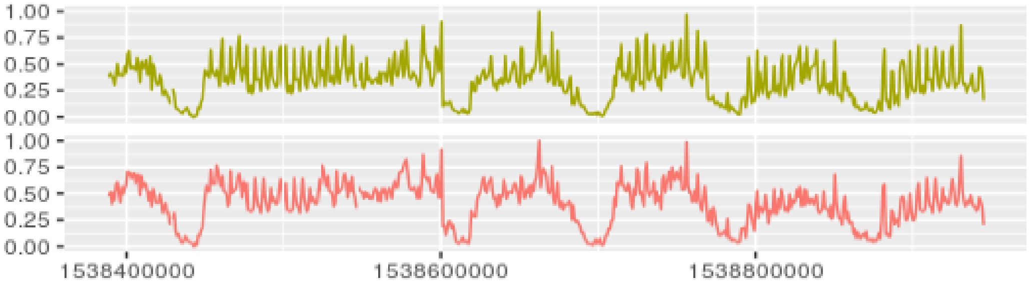


Figure 32. Two KPI time series that exhibit a high positive correlation

*(Linear) correlation on time series*

Algorithms such as Pearson’s correlation and Spearman’s rank correlation are bi-variate analysis tools that measure the strength of association between two variables. Here, the variables are time series, e.g. of monitored KPIs. As exemplified in Figure 32, two time series will have a high positive correlation score if they are almost identical, i.e. one always increases when the other increase and decreases as the other decreases.

*Dynamic Time Warping (DTW)*

While correlation methods are fast and efficient to compute, they are susceptible to lag, warp and shift in the data. In the example in Figure 32, the linear correlation between the two-time series would be lost if one of them would be stretched out or be delayed by even a small amount of time. Dynamic Time Warping is a method to handle these situations. DTW tries to find the matching increases and decreases even if they do not always occur at the exact same time, as long as the shapes of the time series match. The tradeoff is a higher computational complexity and the need to define a good warping window size as well as distance metric.

*Clustering of similar KPIs*

Correlation approaches such as the ones described above are usually bi-variate, i.e. they compare two variables. This pairwise comparison can then be used to create a clustering of the variables/KPIs in order to determine which KPIs are similar to each other and different from other groups of KPIs. State-of-the-art approaches for clustering include k-Means, Agglomerative Hierarchical Clustering and Graph Clustering. These approaches use the correlation score between every pair of KPIs and a pre-define distance metric to create the clusters. K-Means and Agglomerative Hierarchical Clustering typically use the Euclidean distance to determine in which cluster a data point belongs. Graph clustering approaches create a graph where the vertices represent the KPIs and the edge weights between the vertices hold the correlation coefficient between each pair of KPIs. Then, Community Detection algorithms are used to maximize the separation between clusters of vertices.

*Granger causality*

A step further is to investigate which variable causes the change in the other variable(s). A potential use case for this is to investigate which factors are the causes for under-performing KPIs. Correlation alone is not sufficient to show causality as it only shows (temporal) similarity between variables. The Granger causality test is an approach that looks for causality in dependent variables. It uses a method based on linear regression for testing whether one time series can be used for predicting another time series.

### 6.3.4. Comparison of KPI Trends across Platforms for Common Experiments and Validation in a Multi-Platform Setting

The use cases described above (anomaly detection, prediction and dependency analysis) are usually performed within the individual platforms. In addition, the possibility of comparing the behavior of the measured KPIs and network settings in a multi-platform setting is under investigation, in order to compare the platforms and their environments. For example, the validation of KPI values and network configurations on new platforms can be performed based on a consensus of the majority of existing platforms and test cases.

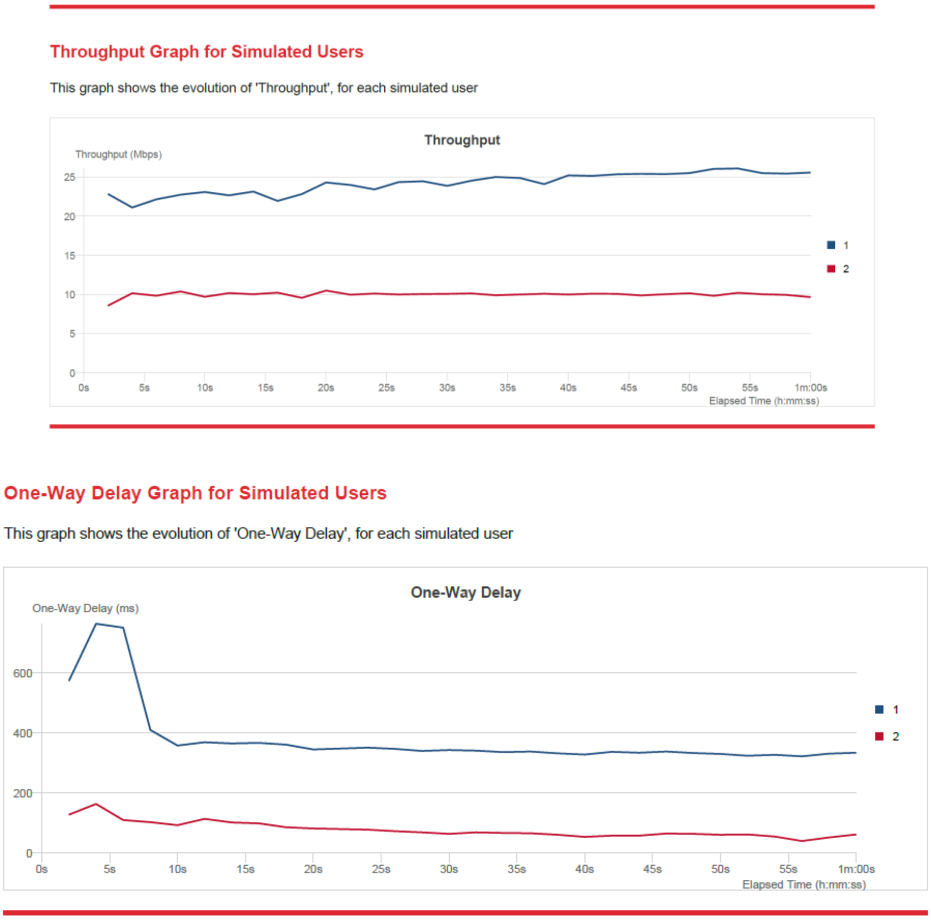
# 7. RESULTS FROM 1ST YEAR EXPERIMENTS AND ANALYTICS

In this Section some examples of results obtained from an initial exploitation of the 5GENESIS M&A functionalities are reported. It should be however highlighted that full usage and integration across platforms of the framework are expected within the next Project phases.

Section 7.1 illustrates the achievement of parallel IM and PM of the cellular architecture, with corresponding real-time visualization in Athens platform. Section 7.2 shows an initial full-chain M&A workflow; in this case, TAP-triggered experiments in the Malaga platform produce different time series for several IM and PM data, which are in turn automatically stored in dedicated InfluxDB tables. Analytics can then pull such data, appropriately rearrange them over time via a time-matching algorithm, and finally execute a linear correlation analysis between all collected parameters. The same workflow is currently under extension towards the inclusion of other significant ML-based analyses, including anomaly detection and prediction, as mentioned in Section 6.3. Furthermore, in order to complement this Section, an initial usage of the NB-IoT simulator is shown and discussed in the dedicated Annex 4; the simulator will be key for benchmarking the experimental results of IoT uses cases targeted in the Surrey platform. Finally, capabilities of the IoT vGW provided for the Surrey platform are described, tested and visualized in a dedicated environment which is presented in Annex 7.

## 7.1. Joint IM & PM Monitoring and Visualization

The Athens platform utilizes Ixria's IxChariot Traffic Generator for conducting measurements and validating the infrastructure. IxChariot simulates a variety of real-world traffic profiles and applications, including Adaptive HTTP Streaming, Video over RTP, YouTube and Netflix (see Annex 3 for more details). It also provides the capability of mixing different applications and simulating the number of users, while reporting several statistics per user or traffic profile. The IxChariot Server is hosted on a Linux Server and is accessed from a web interface, while multiple Software Performance Endpoints (SW EP) have been deployed throughout the infrastructure, including Android mobile devices, end-user PCs and various servers. Several test cases for assessing the QoS/QoE of a variety of scenarios has been defined and deployed in Athens platform. As an indicative example, the QoE of two simulated users using different Video-over-RTP traffic profiles over a selected deployed mobile network is evaluated, by assessing various metrics per user and per traffic profile, such as Throughput, One-way Delay and Media Loss Rate, as shown in Figure 33.



Throughput Graph for Simulated Users

This graph shows the evolution of 'Throughput', for each simulated user

One-Way Delay Graph for Simulated Users

This graph shows the evolution of 'One-Way Delay', for each simulated user

Figure 33. Statistics per a simulated user in IxChariot

At the same time, several statistics of the eNB, EPC and UEs, including SG-i and S1 throughput, DL/UL MCS, CQI, PUCCH/PUSCH SNRs, are recorded using Prometheus and visualized in Grafana, as shown in Figure 34. In this way, end-user QoE results and underlying mobile network conditions during the experiment can be visualized in parallel and correlated via Analytics.

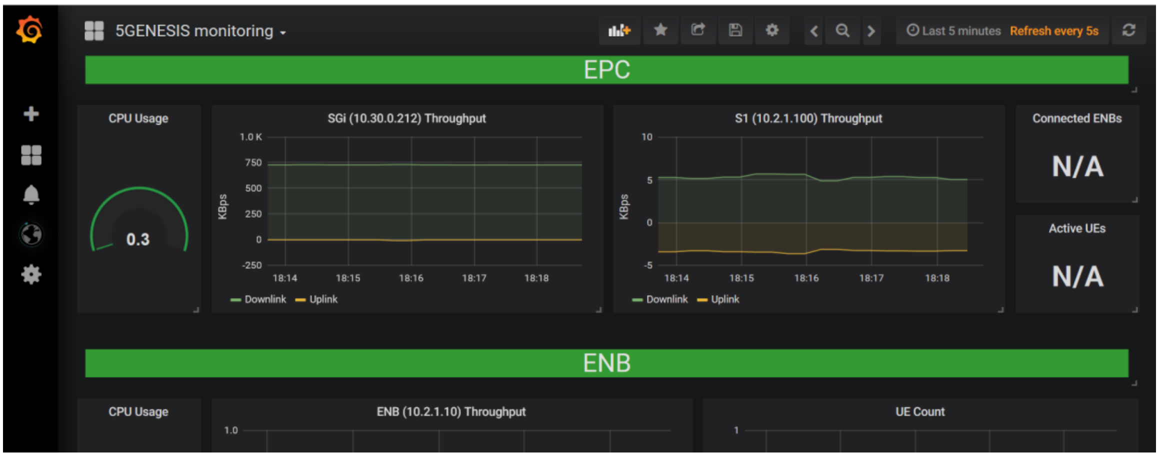


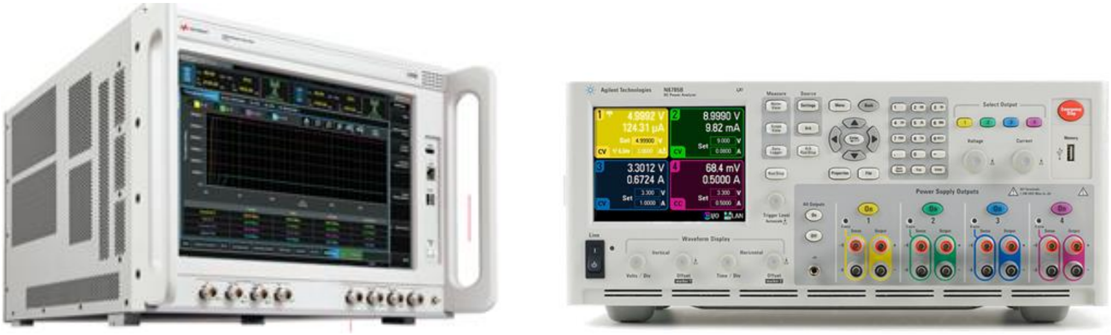
Figure 34. Grafana Visualization of Mobile Network Statistics

## 7.2. Initial Results for M&A Framework and ML-based Analysis

During the experimentation cycle targeting initial KPI validation, whose outcomes are reported in Deliverable D6.1, the Malaga platform has collected further IM and PM data in order to investigate how a KPI such as the throughput is related to other parameters, e.g., radio configurations and device power consumption. Such tests have been performed under different radio scenarios.

### 7.2.1. Setup

The setup used to run the experiments is based on the TRIANGLE setup [77] and it is adapted to 5GENESIS specific test cases. The most relevant components of this setup for collecting radio and power measurements are the UXM and the DC power analyzer, depicted in Figure 35.



(a) (b)

Figure 35. UXM Wireless Test Set (eNB emulator) (a) and DC power analyzer (b) used for experimentation in the Malaga platform

The UXM is an extremely versatile instrument. It is able to emulate multiple base stations with different radio access technologies, including LTE/LTE-A, W-CDMA/HSPA+, GSM/GPRS/EGPRS and TD-SCDMA/HSPA. Additionally, it can also operate simultaneously as radio channel emulator, noise and arbitrary waveform generator, to generate impairments, and signal analyzer. The radio communication between the eNB emulator (UXM) and the UE is not transmitted over the air but conducted through accurately calibrated cabling to ensure that the experienced (emulated) radio conditions (multipath, noise, etc.) are the same that those configured. For testing purposes, most phones typically contain small antenna connectors than may be even hidden, these antennas are used for the connection of the RF cables. Despite the use of cables, the UE observes a totally normal mobile network. For testing purposes, the UXM instrument provides a number of additional useful capabilities, such as a detailed logging. Initial experiments were focused on the following radio measurements:

- **Downlink ACK count:** Ratio of Transport Blocks (TBs) that have been ACKed with respect to the total number of transmitted ones.
- **Downlink NACK count:** Ratio of TBs that have been NACKed with respect to the total number of transmitted ones
- **Downlink StatDTX count:** Ratio of TBs that have neither been ACKed or NACKed with respect to the total number of transmitted ones
- **CQI:** Channel Quality Indicator

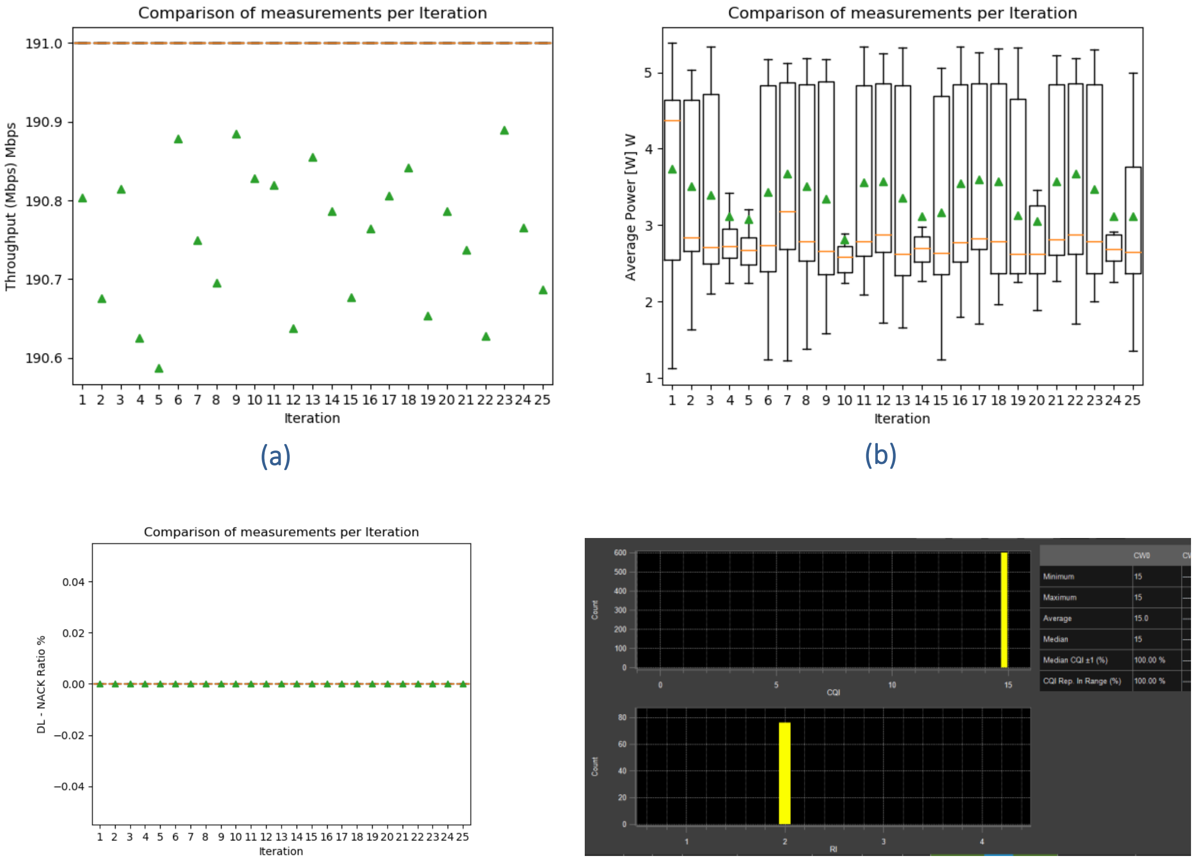
- **RI:** Rank Indicator

On the other side, the battery pins of the mobile device are connected to a DC power analyzer, an N6715B with an N6781A-ATO unit from Keysight. This instrument enables to measure the power consumption in mobile devices with high accuracy. This device can act as a 2-quadrant DC voltage source capable of generating arbitrary waveforms and/or an oscilloscope with data capturing capabilities. As a source, it can provide up to 20 V, up to 3 A. Power rating is 20 W. As a measurement tool, it is capable of measuring values down to nA and uV at a rate of 5.12 us/sample for one parameter.

Three different scenarios have been considered during the experiments:

- **Scenario 1 (ideal)** emulates ideal radio conditions
- **Scenario 2 (urban pedestrian)** emulates the behavior of a channel while the mobile user is walking in a city (EPA5channel model, 20 dB SNR)
- **Scenario 3 (urban driving)** represents a mobile user travelling by car in a city (EVA70 channel model, 15 dB SNR)

The configuration of the experiment includes 4G LTE cells with two transmitting antennas, resulting in a 2x2 MIMO matrix as all LTE devices have minimum two receiving antennas. The user device reports the observed CQI and the rank indicator (RI), the later representing the number of independent data flows that can be transmitted on the channel.

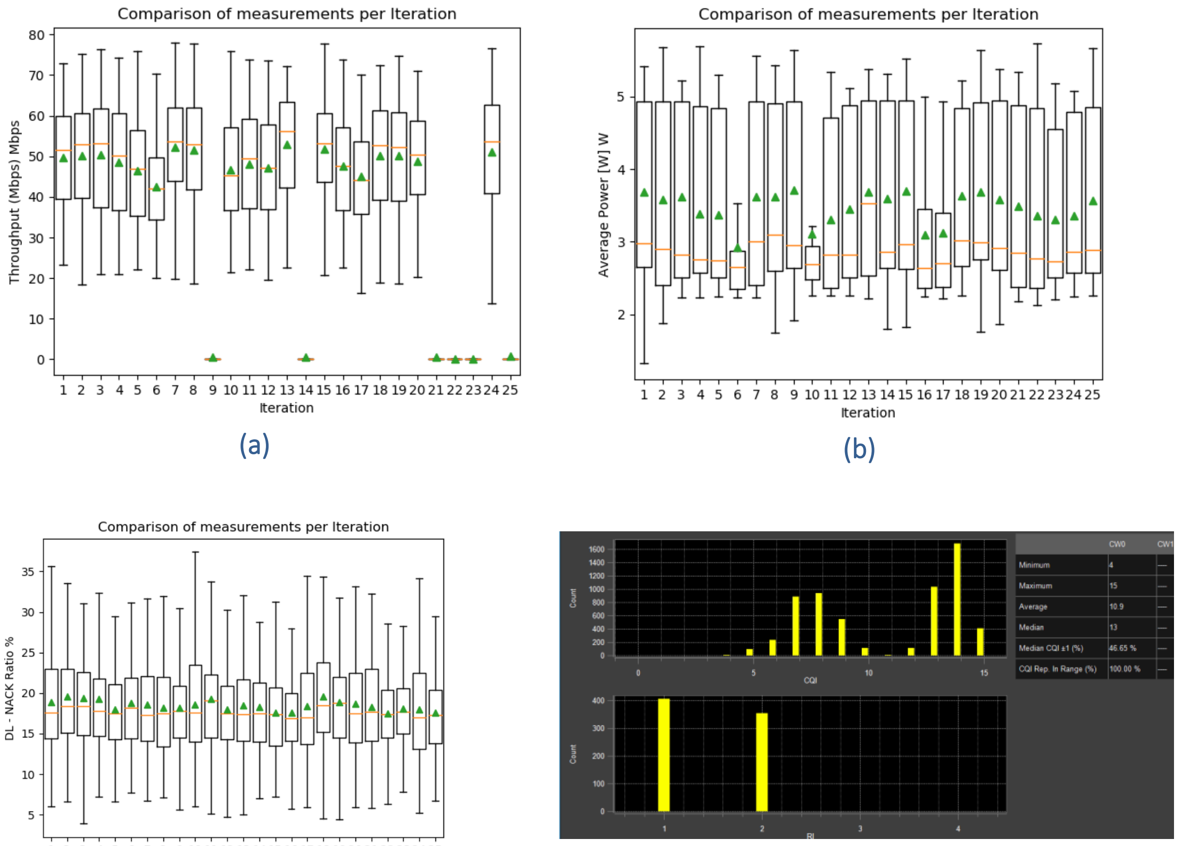


| | CW0 | CV |
|---|---|---|
| Minimum | 15 | — |
| Maximum | 15 | — |
| Average | 15.0 | — |
| Median | 15 | — |
| Median CQI ±1 (%) | 100.00 % | — |
| CQI Rep. In Range (%) | 100.00 % | — |

(a) (b)

(c) (d)

Figure 36. Results in Scenario 1 (ideal): Average DL Throughput (a), Average Power consumption (b), NACK Ratio (c), CQI and RI (d)



|  | CW0 | CW1 |
|---|---|---|
| Minimum | 4 | — |
| Maximum | 15 | — |
| Average | 10.9 | — |
| Median | 13 | — |
| Median CQI ±1 (%) | 46.65 % | — |
| CQI Rep. In Range (%) | 100.00 % | — |

(a) (b)

(c) (d)

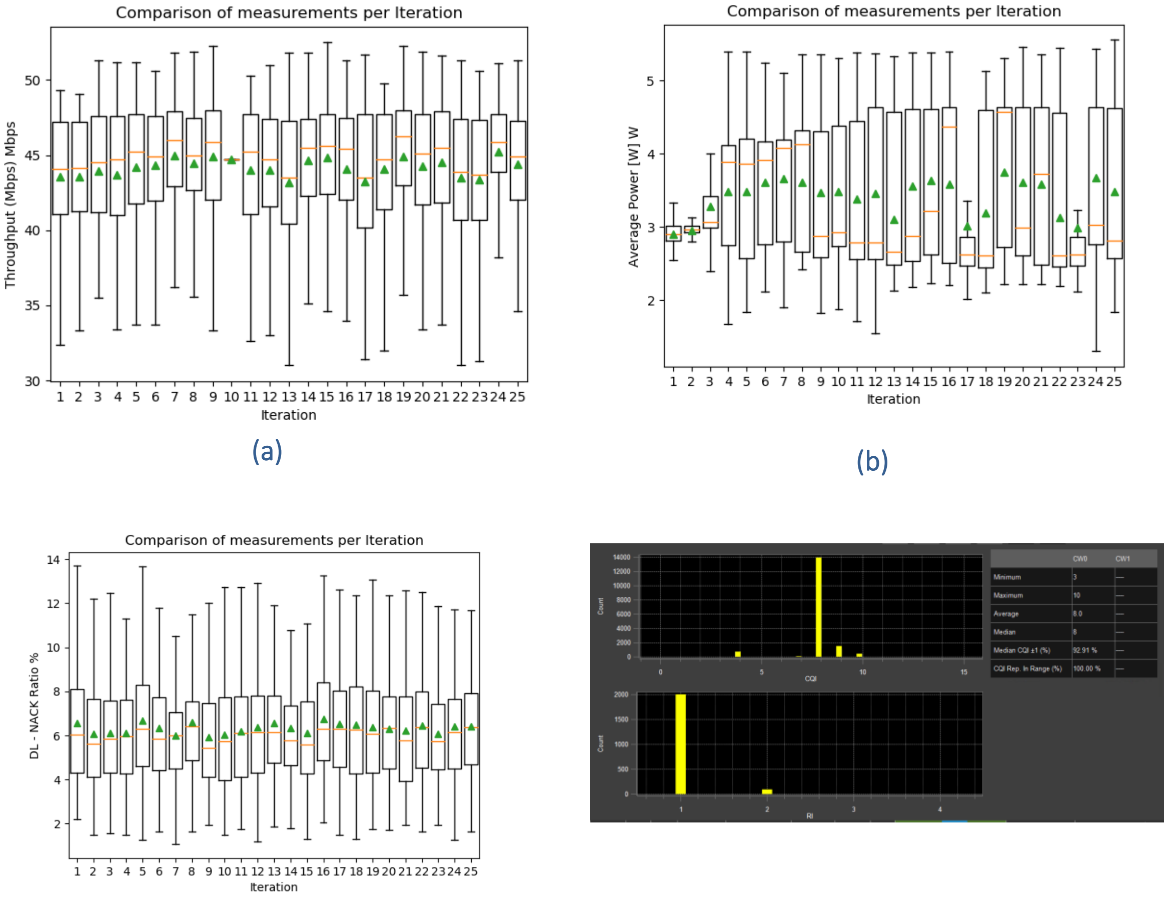
Figure 37. Results in Scenario 2 (urban pedestrian): Average DL Throughput (a), Average Power consumption (b), NACK Ratio (c), CQI and RI (d)

### 7.2.2. Initial Results

Figure 38 shows the results obtained in Scenario 1, 2, and 3, respectively.
Through the CQI value, shown in Figures 36-38 (d), the UE reports an indication of the channel quality, so that the eNB can use it and decide which MCS should be used in DL, in order to limit the block error (3GPP TS 36.213).
A value equal to 15 represents the best channel quality, and as expected, this CQI is always reported under the ideal conditions emulated in Scenario 1. When the channel is not ideal, e.g., under Scenarios 2 and 3, values lower than 15 are used, thus reducing the maximum achievable throughput, as observed by comparing Figures 36-38 (a).

In terms of RI, the UE in Scenario 1 always reports Rank 2 (CW0), while both Rank 1 and Rank 2 (CW1) are reported in Scenarios 2 and 3. Lower ranks also imply lower capacity, being an indication of the number of independent data flows transmitted (e.g., only one independent flow is transmitted with Rank 1).
Further performance decrease is observed in Scenario 3, where a lower SNR, equal to 15 dB, has been chosen to emulate lower signal strengths, as at the UE within the vehicle. In this case, the channel is also more challenging, due to higher delay spread and multipath. As shown in Figure 38, the majority of the recorded RIs indicate single layer (non-MIMO) transmissions

(Rank 1), so that the transmission will most of the time use a single data flow. Additionally, the CQI is lower than in Scenario 2, implying further throughput decrease.



|  | CW0 | CW1 |
|---|---|---|
| Minimum | 3 | --- |
| Maximum | 10 | --- |
| Average | 8.0 | --- |
| Median | 8 | --- |
| Median CQI ±1 (%) | 92.91 % | --- |
| CQI Rep. In Range (%) | 100.00 % | --- |

(a) (b)

(c) (d)

Figure 38. Results in Scenario 3 (urban driving): Average DL Throughput (a), Average Power consumption (b), NACK Ratio (c), CQI and RI (d)

### 7.2.3. Advanced Analytics

Advanced analytics methods allow to obtain more detailed information about the experiment behavior beyond the first-line statistics, e.g., outlier detection and correlation analysis. Before directly working on the measurement data, Analytics must address the challenge of synchronizing data that comes from different measurement sources within the same experiment. During an experiment, the measurements from different probes and devices may come in at slightly different times, often with a few milliseconds difference.

In order to compare the measurements from different sources within the same experiment (or even across experiments), the measurements must be aligned so that the reported values from one source match with the reported value of another source at a specific point in time. Two of the various methods to achieve this are shown in Figure 39. As exemplified in Figure 39, each measurement reports data points at different points in time, with varying intervals in between.

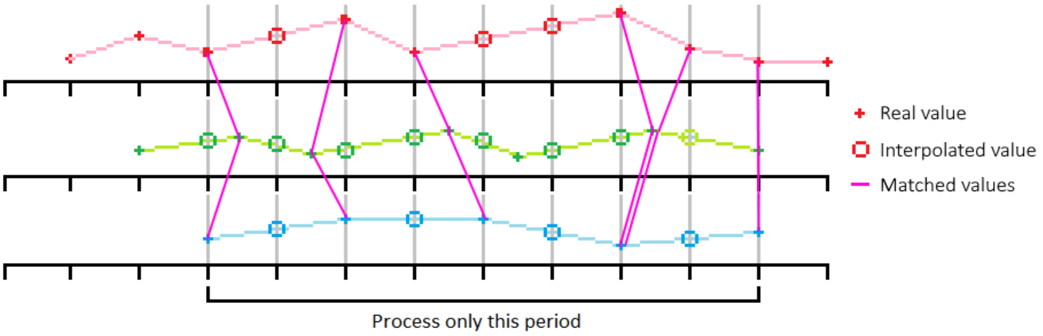


Figure 39. Synchronization between data points. Individual measurement points (red/green/blue pluses) can be linked by interpolation (red/green/blue circles) or various matching approaches (pink lines)

*Interpolation* allows to compare the red, green, and blue measurements at exact time intervals (e.g., once a second) by estimating the values at those time intervals (the circles in Figure 39) given the real values that may occur at random times (the pluses in Figure 39). There exist multiple methods to interpolate values between known data points, such as linear (straight line estimation between two real data points) and various non-linear interpolation methods (e.g., polynomial). Another method to align data points across measurements is to find the data points that are closest together in time, irrespective of the exact time when they were recorded. This is a *synchronization* approach. In Figure 39, the synchronization is demonstrated with the pink lines that connect real data points from the red, green, and blue measurements. The second method is currently implemented in the Analytics component and thus also adopted in the present case, as the time difference between data points across measurements is usually much smaller (often in the order of milliseconds) than the time difference to the neighboring data points in the same measurement (often in the order of seconds), which allows for accurate matching of data points across measurements. Another advantage of this method is that it uses the real values that are reported from the measurement probes and devices. In the current implementation, a time window can be specified (e.g. one second) to only consider data points close together if they lie within this time window. Another challenge to address by Analytics is the possible occurrence of outliers, which are values not lying in the typical range of a variable. In the present cases of the Malaga experiments, the outliers were represented, for example, by download throughput of 9.91e+34 Kbps as shown in Table 4. This clearly unreasonable recorded number suggests an error in the recording or reporting.

Table 4. Detected outlier. Some measurements will report invalid data points, which are outside the range of possible values

| Time (ns) | UL EARFCN | DL - MaximumThput (Kbps) | DL - Meas Time |
|---|---|---|---|
| 1559201270000000000 | 19570 | 308.32 | 0.001 |
| 1559202367000000000 | 19570 | 308.32 | 0.001 |
| 1559202643000000000 | 19570 | 924.96 | 0.005 |
| 1559202978000000000 | 19570 | 308.32 | 0.001 |
| 1559203168000000000 | 19570 | 255.12 | 0.001 |
| 1559203579000000000 | 19570 | 9.91E+34 | 0 |
| 1559203849000000000 | 19570 | 308.32 | 10 |
| 1559204224000000000 | 19570 | 296.28 | 0.001 |
| 1559204426000000000 | 19570 | 296.28 | 0.001 |

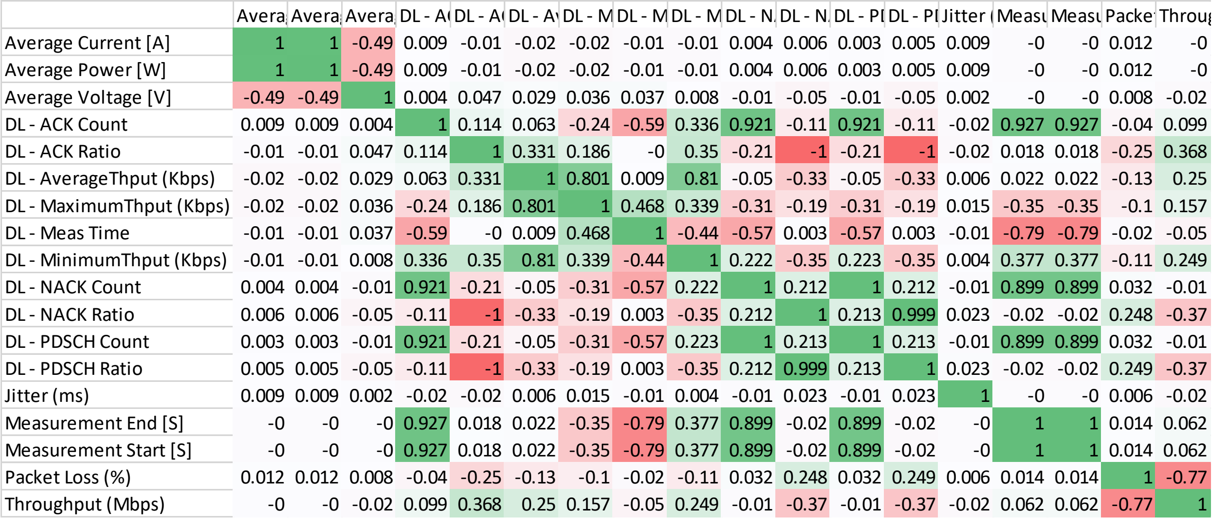
The outlier detection method that is currently implemented uses the standard deviation of the given distribution of recorded data points (per variable). Any value that is at least three standard deviations away from the mean is considered an outlier and will be removed before proceeding to apply analytical methods. The standard-deviation based outlier detection is focused on performance and can be applied in near real-time. More sophisticated methods for outlier detection exist and are described in Section 2. One or more of these methods will likely be offered to the experimenter at a later stage as well.

The experimenter may be interested in the similarity of temporal behavior between recorded variables. A correlation approach is a fast and cheap (in computation resources) way to gain insight into the temporal similarity between variables. In the current implementation, the experimenter can specify the type of correlation they want to perform, choosing from Pearson, Spearman, and Kendall, with the option to provide a custom correlation function. These algorithms are provided by the Pandas package for Python.
The correlation module currently provides two types of correlation use cases:

1. Cross-correlation of fields in the same experiment
2. Correlation of fields across different experiments

The first use case allows the experimenter to compare variables in the same experiment. For example, Table 5 shows the correlation between several recorded variables for an Urban Pedestrian iPerf experiment, including the average power consumption (Average Power [W]) and several download throughput metrics (e.g. DL AverageThput [Kbps]). Equipped with this information, the experimenter can see that there is no linear correlation between these specific variables, which means that the used correlation method (Pearson in this case) was not able to establish a connection between power consumption and download throughput for this specific experiment.

Table 5. Correlation matrix between recorded fields within the same experiment



| | Avera | Avera | Avera | DL - A | DL - A | DL - A | DL - M | DL - M | DL - M | DL - N | DL - N | DL - P | DL - P | Jitter ( | Measu | Measu | Packe | Throu |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Average Current [A] | 1 | 1 | -0.49 | 0.009 | -0.01 | -0.02 | -0.02 | -0.01 | -0.01 | 0.004 | 0.006 | 0.003 | 0.005 | 0.009 | -0 | -0 | 0.012 | -0 |
| Average Power [W] | 1 | 1 | -0.49 | 0.009 | -0.01 | -0.02 | -0.02 | -0.01 | -0.01 | 0.004 | 0.006 | 0.003 | 0.005 | 0.009 | -0 | -0 | 0.012 | -0 |
| Average Voltage [V] | -0.49 | -0.49 | 1 | 0.004 | 0.047 | 0.029 | 0.036 | 0.037 | 0.008 | -0.01 | -0.05 | -0.01 | -0.05 | 0.002 | -0 | -0 | 0.008 | -0.02 |
| DL - ACK Count | 0.009 | 0.009 | 0.004 | 1 | 0.114 | 0.063 | -0.24 | -0.59 | 0.336 | 0.921 | -0.11 | 0.921 | -0.11 | -0.02 | 0.927 | 0.927 | -0.04 | 0.099 |
| DL - ACK Ratio | -0.01 | -0.01 | 0.047 | 0.114 | 1 | 0.331 | 0.186 | -0 | 0.35 | -0.21 | -1 | -0.21 | -1 | -0.02 | 0.018 | 0.018 | -0.25 | 0.368 |
| DL - AverageThput (Kbps) | -0.02 | -0.02 | 0.029 | 0.063 | 0.331 | 1 | 0.801 | 0.009 | 0.81 | -0.05 | -0.33 | -0.05 | -0.33 | 0.006 | 0.022 | 0.022 | -0.13 | 0.25 |
| DL - MaximumThput (Kbps) | -0.02 | -0.02 | 0.036 | -0.24 | 0.186 | 0.801 | 1 | 0.468 | 0.339 | -0.31 | -0.19 | -0.31 | -0.19 | 0.015 | -0.35 | -0.35 | -0.1 | 0.157 |
| DL - Meas Time | -0.01 | -0.01 | 0.037 | -0.59 | -0 | 0.009 | 0.468 | 1 | -0.44 | -0.57 | 0.003 | -0.57 | 0.003 | -0.01 | -0.79 | -0.79 | -0.02 | -0.05 |
| DL - MinimumThput (Kbps) | -0.01 | -0.01 | 0.008 | 0.336 | 0.35 | 0.81 | 0.339 | -0.44 | 1 | 0.222 | -0.35 | 0.223 | -0.35 | 0.004 | 0.377 | 0.377 | -0.11 | 0.249 |
| DL - NACK Count | 0.004 | 0.004 | -0.01 | 0.921 | -0.21 | -0.05 | -0.31 | -0.57 | 0.222 | 1 | 0.212 | 1 | 0.212 | -0.01 | 0.899 | 0.899 | 0.032 | -0.01 |
| DL - NACK Ratio | 0.006 | 0.006 | -0.05 | -0.11 | -1 | -0.33 | -0.19 | 0.003 | -0.35 | 0.212 | 1 | 0.213 | 0.999 | 0.023 | -0.02 | -0.02 | 0.248 | -0.37 |
| DL - PDSCH Count | 0.003 | 0.003 | -0.01 | 0.921 | -0.21 | -0.05 | -0.31 | -0.57 | 0.223 | 1 | 0.213 | 1 | 0.213 | -0.01 | 0.899 | 0.899 | 0.032 | -0.01 |
| DL - PDSCH Ratio | 0.005 | 0.005 | -0.05 | -0.11 | -1 | -0.33 | -0.19 | 0.003 | -0.35 | 0.212 | 0.999 | 0.213 | 1 | 0.023 | -0.02 | -0.02 | 0.249 | -0.37 |
| Jitter (ms) | 0.009 | 0.009 | 0.002 | -0.02 | -0.02 | 0.006 | 0.015 | -0.01 | 0.004 | -0.01 | 0.023 | -0.01 | 0.023 | 1 | -0 | -0 | 0.006 | -0.02 |
| Measurement End [S] | -0 | -0 | -0 | 0.927 | 0.018 | 0.022 | -0.35 | -0.79 | 0.377 | 0.899 | -0.02 | 0.899 | -0.02 | -0 | 1 | 1 | 0.014 | 0.062 |
| Measurement Start [S] | -0 | -0 | -0 | 0.927 | 0.018 | 0.022 | -0.35 | -0.79 | 0.377 | 0.899 | -0.02 | 0.899 | -0.02 | -0 | 1 | 1 | 0.014 | 0.062 |
| Packet Loss (%) | 0.012 | 0.012 | 0.008 | -0.04 | -0.25 | -0.13 | -0.1 | -0.02 | -0.11 | 0.032 | 0.248 | 0.032 | 0.249 | 0.006 | 0.014 | 0.014 | 1 | -0.77 |
| Throughput (Mbps) | -0 | -0 | -0.02 | 0.099 | 0.368 | 0.25 | 0.157 | -0.05 | 0.249 | -0.01 | -0.37 | -0.01 | -0.37 | -0.02 | 0.062 | 0.062 | -0.77 | 1 |

On the other hand, the correlation matrix shows relations that confirm expected dependencies, such as clear relations between the various DL metrics, as well as a relatively strong negative correlation between Packet Loss and Throughput. A negative correlation indicates an antiproportional behavior between the two variables, i.e., a high throughput exhibits a low packet loss and vice versa.

In the second correlation use case (correlation of fields across different experiments), the experimenter may wish to compare the measurement variables of an Urban Pedestrian iPerf experiment with an Ideal iPerf experiment. For this comparison, the two experiments first have to be synchronized with the data point matching method described above, where the temporal information of the data points is transformed into delta values to allow a synchronization between experiments that were likely run at different times. The transformation and synchronization are currently performed automatically.

Table 6 shows the correlation values for each of the measured variables that are common to both compared experiments, in this case the Urban Pedestrian iPerf and the Ideal iPerf. From the first glance, it is apparent that there is little to no correlation between the experiments in comparison. The only exception here is the TimeStamp variable, which of course increases in all experiments as the experiments progresses. At the very least, the TimeStamp correlation serves as verification of implementation correctness.

Table 6. Correlation values between two iPerf experiments, e.g., Urban Pedestrian and Ideal scenarios

| Variable | Correlation |
|---|---|
| Average Current [A] | 0.0346 |
| Average Power [W] | 0.0346 |
| Average Voltage [V] | 0.0665 |
| DL - ACK Count | 0.0176 |
| DL - ACK Ratio | -0.0287 |
| DL - AverageThput (Kbps) | 0.0620 |
| DL - MaximumThput (Kbps) | 0.0781 |
| DL - Meas Time | 0.0036 |
| DL - MinimumThput (Kbps) | 0.0169 |
| DL - NACK Count | -0.0614 |
| DL - NACK Ratio | -0.0287 |
| DL - PDSCH Count | -0.0619 |
| DL - PDSCH Ratio | -0.0290 |
| Jitter (ms) | -0.0217 |
| Measurement End [S] | -0.0003 |
| Measurement Start [S] | -0.0003 |
| Packet Loss (%) | 0.0202 |
| Throughput (Mbps) | 0.0123 |
| TimeStamp | 0.9985 |

From here, the experimenter may wish to investigate further as to why there is no (linear) correlation between the experiments. This is where other analytical methods and visualization can provide additional information. Taking visualization as an example, the power consumption in the two plots for Ideal and Urban Pedestrian iPerf (Figure 40 and Figure 41) show that both metrics exhibit comparable periodic temporal behavior. However, the individual periods are different. The power consumption in each period of the Ideal iPerf experiment stays longer at or near the peak than the power consumption in the Urban Pedestrian iPerf experiment, which decreases immediately all the way down after reaching the peak.

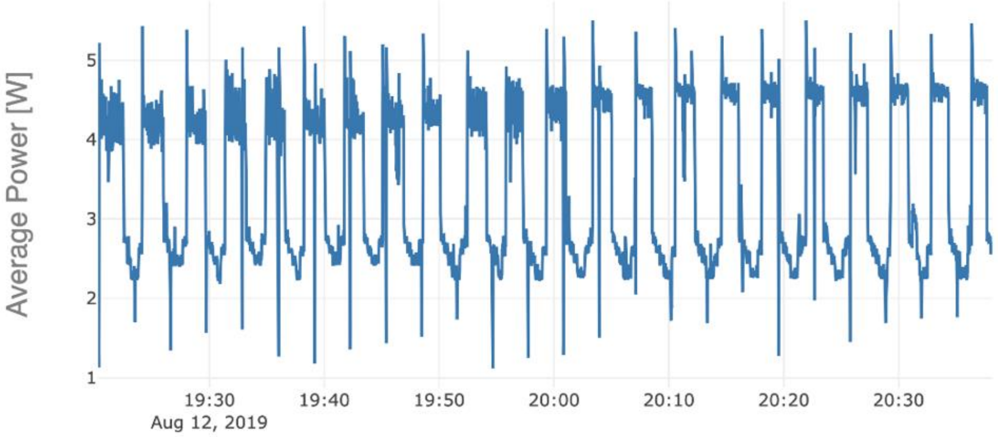


Figure 40. The Ideal iPerf power consumption shows periodic behaviour

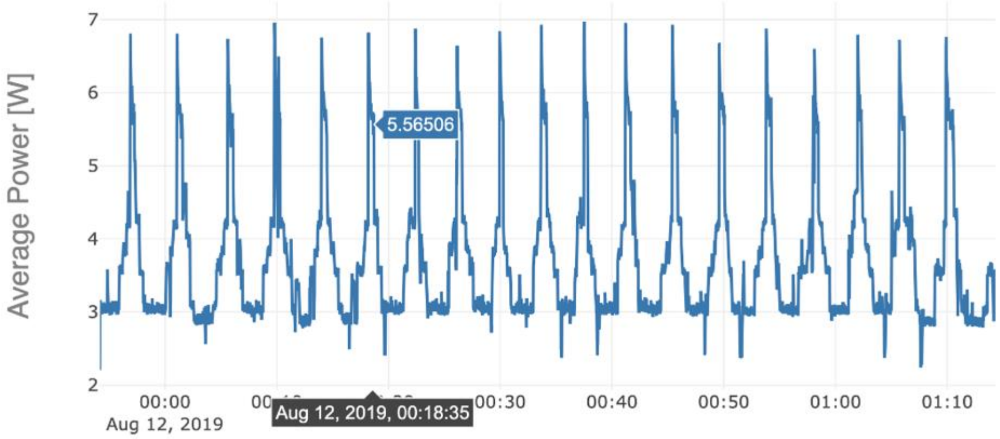


Figure 41. The Urban Pedestrian iPerf power consumption also shows a periodic behavior (compared to the Ideal iPerf), but the characteristic of the individual periods is very different

From the visual analysis it is apparent that there is a relation between the power consumption in both experiments although the correlation analysis did not pick that up. There are potentially multiple reasons why the correlation analysis can fail to reveal such a relation, such as non-linearity, lag and warp of the compared signals. The correlation analysis that is implemented in the Analytics module would struggle to find similarity under these conditions, and more sophisticated time series correlation techniques (planned for future releases, see Section 8) should be employed.

# 8. Release A Summary and Future Plans

## 8.1. M&A Release A

During the first experimental cycle, the 5GENESIS Consortium has executed a large amount of measurements, aiming to initial collection and validation of several KPIs, such as RTT and throughput. The experimental cycle has been performed in parallel across all 5GENESIS platforms towards a full 5G network chain. The results are extensively reported in the dedicated Deliverable D6.1 [31].

The general workflow adopted during the experimentation is reported in Figure 42.

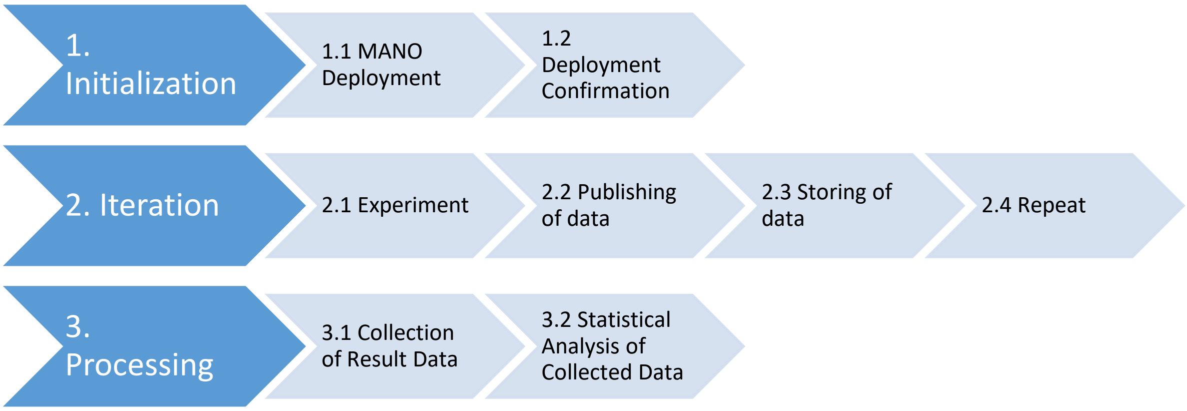


Figure 42. 5GENESIS Workflow for Experimentation Cycle 1

The steps can be grouped into *Initialization*, *Iteration/Experimentation*, and *Processing* phases.

1. *Initialization*
   1.1. Measurement probes are deployed through a MANO
   1.2. The probe deployment is confirmed
2. *Iteration*
   2.1. The experiment is executed
   2.2. Upon conclusion of the experiment, any collected data is published
   2.3. The results are stored in a dedicated storage utility on the experimenter system
   2.4. Steps 2.1-2.3 are repeated
3. *Processing*
   3.1. The data from the experiments is collected for post processing
   3.2. Statistical analysis of the data, as defined in Deliverable D6.1, is conducted using dedicated scripts

Within the above context, the Release A of the M&A framework has contributed providing basic yet fundamental functionalities, i.e., PM probes and statistical analysis. We have produced dedicated scripts to process the collected samples and provide a summary of the experiments in terms of statistical indicators. However, by its nature, the M&A framework enables a deeper 5G analysis beyond KPI statistical validation, as it embeds, since Release A, both infrastructure monitoring and advanced analytics.

*Connections and Interfaces*

As observed in Section 3, the framework has been designed in order to be fully integrated within the 5GENESIS reference architecture. In particular, the M&A Release A includes all the main connections and interfaces enabling an automated experiment execution and data storage, in the form of TAP plugins and result listeners. Moreover, the connections between Analytics and Storage, as well as between Analytics and 5GENESIS Portal have been identified and also preliminary tested, as reported in Sections 6 and 7. Table 7 summarizes the connections between M&A Release A and other blocks of the 5GENESIS reference architecture.

Table 7. M&A Release A: Summary of connections and interfaces

| Connections | | Release A Interfaces | |
|---|---|---|---|
| | | Main Interfaces | Further Interfaces |
| Monitoring ←→ ELCM | | Tool-specific TAP plugins | - |
| Storage ←→ ELCM | | TAP InfluxDB Result Listener | TAP csv Result Listener |
| M&A ←→ Other components | Storage | InfluxDB–Python client | Tool-specific existing plugins and APIs |
| | Portal | Grafana | Python DASH |

*Tools and Components*

With regards to monitoring tools and probes, M&A Release A includes key components for both infrastructure and performance monitoring, thus enabling a nearly-synchronized collection of heterogeneous parameters and KPIs across 5GENESIS platforms. Considering the Storage functionality, a straightforward integration and usage has been achieved and preliminary tested, ruled by TAP (towards the monitoring tools) and a pre-existing Python client (towards Analytics). Finally, initial implementation and usage of Analytics scripts has been also achieved, including key scripts for statistical analysis, as well as for time series synchronization, and intra/inter-experiment correlation. Table 8 summarizes the tools and components included in M&A Release A.

Table 8. M&A Release A: Summary of functionalities and components

| Function | | Release A Components | |
|---|---|---|---|
| | | Main Tools | Further Tools |
| Infrastructure Monitoring | RAN, Core, SDN/NFV | Prometheus,<br>Zabbix | LibreNMS,<br>WiFi Monitoring (non-3GPP Access) |
| | UE | UE_android_app,<br>Android Resource Agent | - |

| | | |
|---|---|---|
| Performance Monitoring | MONROE VN (Ping, iPerf containers),<br>Remote Agents (Ping, iPerf, FTP),<br>Android PM Agents (Ping, iPerf, FTP) | IxChariot,<br>Open 5GCore Benchmarking,<br>NB-IoT simulator |
| Storage | InfluxDB | csv files |
| Analytics | Python-based: Statistical analysis, Time series matching, Linear correlation | - |

## 8.2. Towards M&A Release B

A tight integration of M&A functionalities in the 5GENESIS Platform has been originally planned for the next integration phase, in parallel to the enhancement of the framework from Release A towards Release B. On this line, the first experimentation cycle and the initial usage of Release A has driven several observations to keep in mind for Release B deployment and smooth integration. Among others, the following points can be highlighted:

- In order to support the M&A framework, the TAP-based ELCM should allow the execution of both experiments for statistical analysis of raw data as well as automatic tests for a full system evaluation. This would enable both platform-independent analyses as well as specific system validation tests, all encapsulated in TAP test cases.
- The TAP result listeners should support the export of experiment annotations, e.g., unique experiment and iteration IDs, to be matched with the collected parameters.
- For initial experiments, the parameters were mainly exported as csv files, but for future experiments the export towards InfluxDB long-term storage via the TAP listener will be pursued, since this allows the Analytics functionalities to directly query for the needed inputs in the appropriate database instance.
- The formats used for data export should be further unified across platforms, in order to facilitate the processing and potentially enable cross-platform Analytics. The use of InfluxDB is key for achieving this goal. Moreover, the Consortium is currently working on finding cross-platform agreements on the terminology to be used to identify the parameters collected at specific vantage points. This would allow the Analytics functionalities to be developed and used in a platform-agnostic fashion.
- The automation of Analytics scripts under the TAP-based ELCM would result in the encapsulation of advanced analysis within TAP test cases, enabling their execution on demand, e.g., as required by the experimenter.

M&A Release B will thus target the enhancement of Release A components and functionalities, as well as the final integration in the 5GENESIS Platform.

Table 9 summarizes some of the planned activities towards these goals, and anticipates the following sections, in which specific enhancements and improvements targeted by M&A Release B are described with more detail.

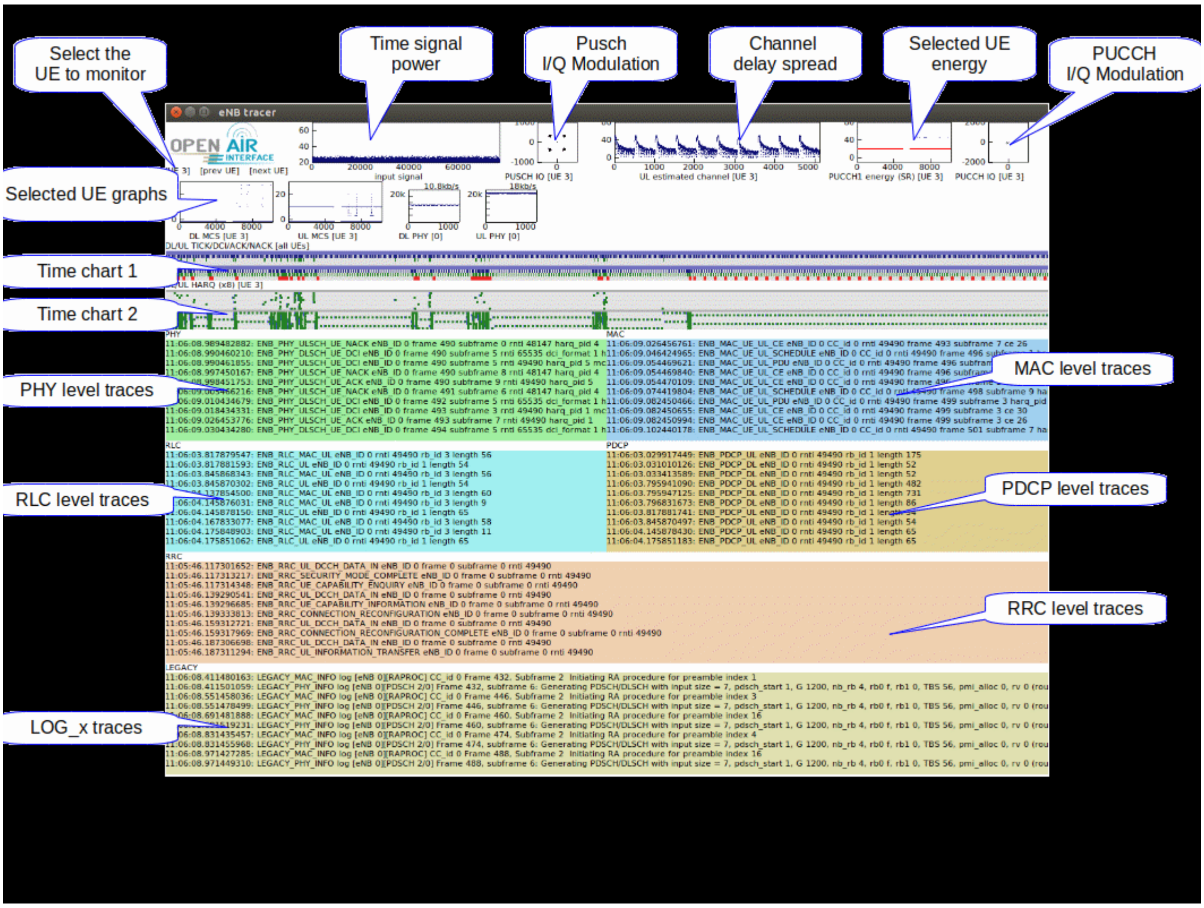
Table 9. M&A Release B planned activities

| Function & Connections | | Release B (Planned Activities) | |
|---|---|---|---|
| | | Main Tools/Interfaces | Further Tools/Interfaces |
| Infrastructure Monitoring | RAN, Core, SDN/NFV | Enhance the configurations of Prometheus and Zabbix | Explore possible use of Telegraf |
| | UE | Enhance the configurations of UE_android_app and Android Resource Agent<br>Integrate T-Tracer | - |
| Performance Monitoring | | Create and integrate new QoS/QoE containers for MONROE VN<br>Create and integrate new Remote/Android Agents | Integrate OWAMP*<br>*planned in Athens platform |
| Storage | | Strengthen the use of InfluxDB | Minimize the use of csv files |
| Analytics | | Include advanced ML functionalities (as in Section 6.3) | Explore possible extension towards big data |
| | | | |
| Monitoring ←→ ELCM | | Create and integrate new TAP plugins as new monitoring tools are integrated | - |
| Storage ←→ ELCM | | Strengthen the use of TAP InfluxDB Result Listener (agreement on format) | Minimize the use of TAP csv Result Listener |
| M&A ←→ Other components | ELCM | Automatize Analytics scripts under TAP | - |
| | MANO layer | Explore integration with policy engines (APEX and NEAT)*<br>*planned in Surrey platform | |

## 8.3. Infrastructure and Performance Monitoring Enhancements

### 8.3.1. T-Tracer

M&A Release B targets the integration of T-Tracer for UE monitoring. T-Tracer is a monitoring tool for OAI radio components, integrating an event/data collector and a set of software to receive, record, display, and analyze the events/data from the collector. In particular, a GUI makes it possible to show several heterogeneous parameters, ranging from signal levels (e.g., power levels) to events at different layers of the RAN protocol stack, such as Physical (PHY) and Medium Access Control (MAC) layers, Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP), and Radio Resource Control (RRC). Figure 43 shows an example of the T-Tracer GUI, in which the visualization of data gathered during an experiment execution can be activated/de-activated dynamically, and the data is grouped into different categories. Besides the GUI, the *textlog tracer* can be also used, so that T-Traces data can timestamped and agglomerated into log files, in particular in *.txt* format, for further post-processing.



PHY
```
11:06:08.989482882: ENB_PHY_ULSCH_UE_NACK eNB_ID 0 frame 490 subframe 0 rnti 48147 harq_pid 4
11:06:08.990460210: ENB_PHY_DLSCH_UE_DCI eNB_ID 0 frame 490 subframe 5 rnti 65535 dci_format 1 h
11:06:08.990461855: ENB_PHY_ULSCH_UE_DCI eNB_ID 0 frame 490 subframe 5 rnti 49490 harq_pid 5 mc
11:06:08.997450167: ENB_PHY_ULSCH_UE_NACK eNB_ID 0 frame 490 subframe 8 rnti 48147 harq_pid 4
08.998451753: ENB_PHY_ULSCH_UE_ACK eNB_ID 0 frame 490 subframe 9 rnti 49490 harq_pid 5
06:09.005466216: ENB_PHY_ULSCH_UE_NACK eNB_ID 0 frame 491 subframe 6 rnti 48147 harq_pid 4
06:09.010434679: ENB_PHY_DLSCH_UE_DCI eNB_ID 0 frame 492 subframe 5 rnti 65535 dci_format 1 h
11:06:09.018434331: ENB_PHY_ULSCH_UE_DCI eNB_ID 0 frame 493 subframe 3 rnti 49490 harq_pid 1 mc
11:06:09.026453776: ENB_PHY_ULSCH_UE_ACK eNB_ID 0 frame 493 subframe 7 rnti 49490 harq_pid 1
11:06:09.030434280: ENB_PHY_DLSCH_UE_DCI eNB_ID 0 frame 494 subframe 5 rnti 65535 dci_format 1 h
```

MAC
```
11:06:09.026456761: ENB_MAC_UE_UL_CE eNB_ID 0 CC_id 0 rnti 49490 frame 493 subframe 7 ce 26
11:06:09.046424965: ENB_MAC_UE_UL_SCHEDULE eNB_ID 0 CC_id 0 rnti 49490 frame 496 s
11:06:09.054469621: ENB_MAC_UE_UL_PDU eNB_ID 0 CC_id 0 rnti 49490 frame 496 subfram
11:06:09.054469840: ENB_MAC_UE_UL_CE eNB_ID 0 CC_id 0 rnti 49490 frame 496 subfra
11:06:09.054470109: ENB_MAC_UE_UL_CE eNB_ID 0 CC_id 0 rnti 49490 frame 49
11:06:09.074419804: ENB_MAC_UE_UL_SCHEDULE eNB_ID 0 CC_id 0 rnti 49490 frame 498 subframe 9 ha
11:06:09.082450466: ENB_MAC_UE_UL_PDU eNB_ID 0 CC_id 0 rnti 49490 frame 499 subframe 3 harq_pid
11:06:09.082450655: ENB_MAC_UE_UL_CE eNB_ID 0 CC_id 0 rnti 49490 frame 499 subframe 3 ce 30
11:06:09.082450994: ENB_MAC_UE_UL_CE eNB_ID 0 CC_id 0 rnti 49490 frame 499 subframe 3 ce 26
11:06:09.102440178: ENB_MAC_UE_UL_SCHEDULE eNB_ID 0 CC_id 0 rnti 49490 frame 501 subframe 7 ha
```

RLC
```
11:06:03.817879547: ENB_RLC_MAC_UL eNB_ID 0 rnti 49490 rb_id 3 length 56
11:06:03.817881593: ENB_RLC_UL eNB_ID 0 rnti 49490 rb_id 1 length 54
11:06:03.845868343: ENB_RLC_MAC_UL eNB_ID 0 rnti 49490 rb_id 3 length 56
11:06:03.845870302: ENB_RLC_UL eNB_ID 0 rnti 49490 rb_id 1 length 54
04.137854500: ENB_RLC_MAC_UL eNB_ID 0 rnti 49490 rb_id 3 length 60
06:04.145876031: ENB_RLC_MAC_UL eNB_ID 0 rnti 49490 rb_id 3 length 9
06:04.145878150: ENB_RLC_UL eNB_ID 0 rnti 49490 rb_id 1 length 65
11:06:04.167833077: ENB_RLC_MAC_UL eNB_ID 0 rnti 49490 rb_id 3 length 58
11:06:04.175848903: ENB_RLC_MAC_UL eNB_ID 0 rnti 49490 rb_id 3 length 11
11:06:04.175851062: ENB_RLC_UL eNB_ID 0 rnti 49490 rb_id 1 length 65
```

PDCP
```
11:06:03.029917449: ENB_PDCP_UL eNB_ID 0 rnti 49490 rb_id 1 length 175
11:06:03.031010126: ENB_PDCP_DL eNB_ID 0 rnti 49490 rb_id 1 length 52
11:06:03.033413589: ENB_PDCP_DL eNB_ID 0 rnti 49490 rb_id 1 length 52
11:06:03.795941090: ENB_PDCP_DL eNB_ID 0 rnti 49490 rb_id 1 length 482
11:06:03.795947125: ENB_PDCP_DL eNB_ID 0 rnti 49490 rb_id 1 length 731
11:06:03.796831673: ENB_PDCP_DL eNB_ID 0 rnti 49490 rb_id 1 length 86
11:06:03.817881741: ENB_PDCP_UL eNB_ID 0 rnti 49490 rb_id 1 len
11:06:03.845870497: ENB_PDCP_UL eNB_ID 0 rnti 49490 rb_id 1 length 54
11:06:04.145878430: ENB_PDCP_UL eNB_ID 0 rnti 49490 rb_id 1 length 65
11:06:04.175851183: ENB_PDCP_UL eNB_ID 0 rnti 49490 rb_id 1 length 65
```

RRC
```
11:05:46.117301652: ENB_RRC_UL_DCCH_DATA_IN eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.117313217: ENB_RRC_SECURITY_MODE_COMPLETE eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.117314348: ENB_RRC_UE_CAPABILITY_ENQUIRY eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.139290541: ENB_RRC_UL_DCCH_DATA_IN eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.139296685: ENB_RRC_UE_CAPABILITY_INFORMATION eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.139333813: ENB_RRC_CONNECTION_RECONFIGURATION eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.159312721: ENB_RRC_UL_DCCH_DATA_IN eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.159317969: ENB_RRC_CONNECTION_RECONFIGURATION_COMPLETE eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.187306698: ENB_RRC_UL_DCCH_DATA_IN eNB_ID 0 frame 0 subframe 0 rnti 49490
11:05:46.187311294: ENB_RRC_UL_INFORMATION_TRANSFER eNB_ID 0 frame 0 subframe 0 rnti 49490
```

LEGACY
```
11:06:08.411480163: LEGACY_MAC_INFO log [eNB 0][RAPROC] CC_id 0 Frame 432. Subframe 2  Initiating RA procedure for preamble index 1
11:06:08.411501059: LEGACY_PHY_INFO log [eNB 0][PDSCH 2/0] Frame 432, subframe 6: Generating PDSCH/DLSCH with input size = 7, pdsch_start 1, G 1200, nb_rb 4, rb0 f, rb1 0, TBS 56, pmi_alloc 0, rv 0 (rou
11:06:08.551458036: LEGACY_MAC_INFO log [eNB 0][RAPROC] CC_id 0 Frame 446. Subframe 2  Initiating RA procedure for preamble index 3
11:06:08.551478499: LEGACY_PHY_INFO log [eNB 0][PDSCH 2/0] Frame 446, subframe 6: Generating PDSCH/DLSCH with input size = 7, pdsch_start 1, G 1200, nb_rb 4, rb0 f, rb1 0, TBS 56, pmi_alloc 0, rv 0 (rou
06:08.691481888: LEGACY_MAC_INFO log [eNB 0][RAPROC] CC_id 0 Frame 460. Subframe 2  Initiating RA procedure for preamble index 16
519231: LEGACY_PHY_INFO log [eNB 0][PDSCH 2/0] Frame 460, subframe 6: Generating PDSCH/DLSCH with input size = 7, pdsch_start 1, G 1200, nb_rb 4, rb0 f, rb1 0, TBS 56, pmi_alloc 0, rv 0 (rou
06:08.831435457: LEGACY_MAC_INFO log [eNB 0][RAPROC] CC_id 0 Frame 474. Subframe 2  Initiating RA procedure for preamble index 4
11:06:08.831455968: LEGACY_PHY_INFO log [eNB 0][PDSCH 2/0] Frame 474, subframe 6: Generating PDSCH/DLSCH with input size = 7, pdsch_start 1, G 1200, nb_rb 4, rb0 f, rb1 0, TBS 56, pmi_alloc 0, rv 0 (rou
11:06:08.971427285: LEGACY_MAC_INFO log [eNB 0][RAPROC] CC_id 0 Frame 488. Subframe 2  Initiating RA procedure for preamble index 16
11:06:08.971449310: LEGACY_PHY_INFO log [eNB 0][PDSCH 2/0] Frame 488, subframe 6: Generating PDSCH/DLSCH with input size = 7, pdsch_start 1, G 1200, nb_rb 4, rb0 f, rb1 0, TBS 56, pmi_alloc 0, rv 0 (rou
```

Figure 43. An example of OAI T-Tracer GUI: data visualization

T-Tracer enables the collection of a complete list of radio parameters [78]. Within the scope of 5GENESIS M&A, the following UE-side log will be considered:

```
{
    D = UE_PHY_MEAS
    "DESC": UE PHY measurements
    "GROUP": ALL:PHY:GRAPHIC:HEAVY:UE
    "FORMAT": int, eNB_ID : int, frame : int, subframe : int, rsrp : int, rssi : int, snr : int, rx_power : int,
noise_power : int, w_cqi : int, freq_offset
}
```

The log provides in fact several physical layer parameters, which can be collected during the experiments and stored in dedicated databases, so that Analytics can access them for post-experiment analysis, e.g., correlation with experienced KPIs such as throughput and latency. T-Tracer is mainly used in 4G OAI RAN at the eNB side, but is currently being integrated with 5G NR at the UE side. A full integration within the 5GENESIS M&A framework is targeted during the next development phase.

### 8.3.2. OWAMP

OWAMP is an open-source command line client application and a policy daemon used to determine one-way latencies between hosts. It is an implementation of the OWAMP protocol [79]. The tipping point of this protocol and therefore its implementation as a software tool is that it is capable of measuring one-way latency thus it can be used for detecting congestion or link capabilities. The tool is used in conjunction with precision time sources (i.e., GPS clocks, Network Time Protocol (NTP), and so on) that allow network hosts and probes to timestamp packets with typical errors that are substantially smaller than the delays seen on real non-LAN networks.

In this sense by using OWAMP, it is possible to collect passive and active measurement data sufficient to determine a broad class of singleton characteristics (e.g., loss probability, median delay, jitter, 90th percentile of delay). Non-singleton characteristics, such as the expected inter-arrival gap of packets that were sent back-to-back, can be measured as well. It should be noted that all measurements are done with synthetic traffic, and application simulation is outside of the scope of OWAMP. The protocol is not designed to be able to send a packet as soon as a response to the previous packet arrives but can send on any predetermined schedule (including immediately after the last packet was sent).

The OWAMP client will possibly be installed in various locations inside the infrastructure both physical and virtual (although virtualized instances create accuracy issues in one-way latency calculation) and will be consistently used for One Way Latency KPI calculations.

Some caveats of using OWAMP in the infrastructure are:

- Require NTP for synchronization (at least 4 NTP clocks are needed for each probe)
- Power management on the probes should be disabled in order to keep the CPU clock stable

For the next integration phase, it is planned to use the OWAMP implementation and API in order to be able to configure the probes according to the experiment and to collect the measurements as soon as the experiment has ended, retrieving the log file and storing the results in the M&A framework database.

### 8.3.3. MONROE VN Enhancements

Development of both common and use case and platform specific performance monitoring tools will continue during the second year of the project. In relation to the MONROE VN, two main development strands are planned.

- One limitation of TAP is that it can only operate in a push-based mode. In situations where the UE lacks a public IP-address and has no other direct network connections to the ELCM, this means that TAP cannot control the monitoring probes at the UE. This is for instance an issue for performing controlled experiments during the festival of lights event in Berlin, where the network connectivity will be public WiFi. To resolve this issue, a TAP agent will be developed for the MONROE scheduler, allowing the MONROE scheduler to act as an intermediary between TAP and the MONROE VN when needed.
- Additional measurement probes will be developed to support performance monitoring of additional metrics and KPIs as defined in WP6.

Planned use case specific performance monitoring enhancements during year two include development of suitable measurement probes and supporting tools to measure energy efficiency for the NB-IoT use case in Surrey. Here, the simulation results presented in Annex 4 will aid the configuration of the measurements and the measurement results will also be used as input in the simulations to cover scenarios beyond what can be covered by the measurements.

## 8.4. Extending Machine Learning Analytics

The ML tools reviewed in Section 6.3 are under development and integration in the overall M&A framework, as exemplified by the result reported in Section 7.2, which showcases an anomaly detection plus linear correlation analysis between experimental data. In particular, as described in Section 8.2, the Consortium is planning to investigate multi-platform Analytics techniques when the platforms will support multi-platform operation. Since these scenarios will directly lead to significant increases of the volume of collected data, the use of deep learning solutions for these particular analyses is under consideration.

## 8.5. APEX and NEAT Integration

In the context of 5G network automation, the APEX (Adaptive Policy Execution) [80] policy engine by LMI presents a possible tool for automated decision making. It can handle adaptive policies, i.e., policies that can modify their behavior based on system and network conditions, including decision making at runtime rather than at policy definition time, and the ability to use context information that was not provided in the incoming event or request.

APEX is a versatile policy engine and can adopt various roles in the 5GENESIS project. In particular, LMI is investigating the following two use cases:

- To provide intelligence resource allocation at slice creation time, APEX can make real-time decisions to influence the slice creation through the slice manager, considering external context information such as business requirements (SLA) and monitored data from the network. For this use case, APEX is considered to interface with the slice manager.

- During runtime of 5GENESIS test cases, APEX can make decisions about the course of the experiment (e.g., related to the measuring process), considering experiment parameters such as target KPIs and monitoring/analytics information from the test scenario. For this use case, APEX is considered to interface with the Analytics module.

As also reported in Deliverable D3.1, Apex will sit in the 5GENESIS MANO layer, and can take in data from the monitoring/analytics components in order to make decisions with respects to the network configuration and test case setup. The integration is in particular targeted for the Surrey platform.

In addition to the APEX policy engine described above, the NEAT policy framework will be integrated during the second development cycle, enabling policy management at the UE. As discussed in Deliverable D4.10, NEAT runs on end hosts and decouples the application from the transport protocol used, allowing the protocol and its configuration to be dynamically selected at run-time based on application preferences, network status and system policy. We plan to explore how the NEAT policy system can make use of monitoring/analytics information in its decisions. For example, information about the quality of different paths can be used as input for interface selection. Monitoring and analytics information can be provided to a NEAT-enabled UE either directly from the M&A component or through configuration information pushed to the UE from the slice manager. Again, the integration is in particular targeted for the Surrey platform.

## 8.6. QoE Modelling and Evaluation

The designed M&A framework allows a nearly straightforward extension and usage for the use cases planned in the platforms involving real users. In this latter case, the Analytics components will be extended in order to derive user-centric QoE KPIs, after proper data anonymization via the tools reported in Annex 5. The final goal is to provide useful insights on the user perspective, while also performing QoS/QoE correlation analysis and QoE modelling.

A clear example of such M&A extension is the Dense Urban Streaming use case targeted by the Berlin platform. In this scenario, a 360° video streaming will be produced and made available to the final users of the 2020 Festival of Lights event. The streaming will be available via a temporary nomadic connectivity island at Humboldt University, which includes 5G mmWave backhaul links to the main 5GENESIS Berlin platform [81, Section 4.5.4.1]. The focus of the use case is to evaluate the KPIs related to the 5G core performance, particularly in terms of Capacity, Latency, Reliability, Service Creation Time, which is in the full scope of the M&A framework. However, the same use case allows to perform QoE video Analytics, thanks to the possibility of collecting feedbacks from end-users in a crowdsourced manner, and embedding the Analytics component with dedicated functionalities [82]. By doing so, several insights can be derived on the performance of different encoding parameters, tiling strategies, video players, adaptation algorithms, and network technologies, and their direct impact on QoE.

# CONCLUSION

This document provided a detailed description of design, implementation, and initial usage of the 5GENESIS M&A framework in its Release A.

Based on 5GENESIS requirements and goals, and in light of state-of-the-art network monitoring and analytics functionalities, the framework positions itself as a key enabler for a complete validation of 5G KPIs. This is due to its capability of parallel collection of infrastructure and performance parameters, and the use of state-of-the-art ML tools for deep and reliable data storage and analysis. Moreover, its design choices and implementation solutions enable a nearly-transparent instantiation across the 5GENESIS Platform, which directly facilitate experiment reproducibility and result comparability within and across the 5GENESIS platforms.

The next Project phase will focus on the full integration of the framework in the 5GENESIS Platform, also moving forward the enhancement of embedded tools and functionalities. In particular, from the Monitoring perspective, it is planned the deployment and integration of further probes, in order to provide an even more extensive view of infrastructure and E2E QoS/QoE parameters and KPIs, while still preserving scalability and avoiding excessive resource usage and data storage. In this view, from the Analytics perspective, it is in particular targeted the investigation of data correlation/causality as well as the introduction of prediction mechanisms, which would help to provide reliable data analysis in an efficient manner.

# References

[1] A. H. Celdrán, M. G. Pérez, F. J. G. Clemente, and G. M. Pérez, "Automatic monitoring management for 5G mobile networks", Procedia Computer Science, vol. 110, 2017, pp. 328–335.

[2] I. Angelopoulos, E. Trouva, and G. Xilouris, "A monitoring framework for 5G service deployments", Proc. IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'17), Lund, Sweden, June 2017, pp. 1–6.

[3] M. G. Kibria et al., "Big Data Analytics, Machine Learning, and Artificial Intelligence in Next-Generation Wireless Networks", IEEE Access, vol. 6, 2018, pp. 32328–32338.

[4] E. Pateromichelakis et al., "End-to-End Data Analytics Framework for 5G Architecture", IEEE Access, vol. 7, 2019, pp. 40295–40312.

[5] L. M. Contreras, P. Doolan, H. Lønsethagen, and D. R. López, "Operational, organizational and business challenges for network operators in the context of SDN and NFV", Computer Networks, vol. 92, 2015, pp. 211–217.

[6] C. Tselios and G. Tsolis, "On QoE-awareness through virtualized probes in 5G networks", Proc. IEEE 21st International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'16), Toronto, Canada, ON, Oct. 2016, pp. 159–164.

[7] Z. Liao, Q. Yin, Y. Huang, and L. Sheng, "Management and application of mobile big data", International Journal of Embedded Systems, vol. 7, no. 1, 2015, pp. 63–70.

[8] Y. Demchenko, P. Grosso, C. de Laat, and P. Membrey, "Addressing big data issues in scientific data infrastructure", Proc. IEEE International Conference on Collaboration Technologies and Systems (CTS '13), San Diego, CA, USA, May 2013, pp. 48–55.

[9] X. Ge, H. Cheng, M. Guizani, and T. Han, "5G wireless backhaul networks: challenges and research advances", IEEE Network, vol. 28, no. 6, 2014, pp. 6–11.

[10] J. Xie et al., "A Survey on Machine Learning-Based Mobile Big Data Analysis: Challenges and Applications", Wireless Communications and Mobile Computing, vol. 2018, 2018, Article ID 8738613.

[11] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: A survey", Computer Networks, vol. 57, no. 9, 2013, pp. 2093–2115.

[12] G. Gardikis et al., "An integrating framework for efficient NFV monitoring", Proc. IEEE Conference on Networking Softwarization (NetSoft'16), Seoul, South Korea, June 2016, pp. 1–5.

[13] E. Chirivella-Perez, J. Gutiérrez-Aguado, J. M. Alcaraz-Calero, and Q. Wang, "NFVMon: Enabling Multioperator Flow Monitoring in 5G Mobile Edge Computing", Wireless Communications and Mobile Computing, vol. 2018, 2018, Article ID 2860452.

[14] Ö. Alay et al., "End to End 5G Measurements with MONROE: Challenges and Opportunities", Proc. IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI'18), Palermo, Italy, Sept. 2018, pp. 1–6.

[15] M. Abderrahim et al., "A holistic monitoring service for fog/edge infrastructures: a foresight study", Proc. IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud'17) , Prague, Czech Republic, Aug. 2017, pp. 337–344.

[16] M. Liyanage et al., "Software Defined Monitoring (SDM) for 5g mobile backhaul networks", Proc. IEEE International Symposium on Local and Metropolitan Area Networks. (LANMAN'17), Osaka, Japan, June 2017, pp. 1–6.

[17] P. Trakadas et al., "Scalable monitoring for multiple virtualized infrastructures for 5G services", Proc. International Symposium on Advances in Software Defined Networking and Network Functions Virtualization (SoftNetworking'18), Athens, Greece, Apr. 2018, pp. 1–4.

[18] Á. Brandón, M. S. Pérez, J. Montes, and A. Sanchez, "FMonE: A Flexible Monitoring Solution at the Edge", Wireless Communications and Mobile Computing, vol. 2018, 2018, Article ID 2068278.

[19] U. Goel, M. P. Wittie, K. C. Claffy, and A. Le, "Survey of end-to-end mobile network measurement testbeds, tools, and services", IEEE Communications Surveys and Tutorials, vol. 18, no. 1, 2016, pp. 105–123.

[20] "Network Monitoring Tools" [Online], http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html, Accessed: Sept. 2019.

[21] "Comparison of network monitoring systems" [Online], https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems, Accessed: Sept. 2019.

[22] "Prometheus" [Online], https://prometheus.io, Accessed: Sept. 2019.

[23] "Zabbix" [Online], https://www.zabbix.com, Accessed: Sept. 2019.

[24] "ONF" [Online], https://www.opennetworking.org, Accessed: Sept. 2019.

[25] "OpenStack" [Online], https://www.openstack.org, Accessed: Sept. 2019.

[26] "Open Source MANO" [Online], https://osm.etsi.org, Accessed: Sept. 2019.

[27] "Open Baton" [Online], http://openbaton.github.io, Accessed: Sept. 2019.

[28] “OpenStack – Ceilometer” [Online], https://docs.openstack.org/ceilometer/latest/, Accessed: Sept. 2019.

[29] “Open Baton – Zabbix Plugin” [Online], https://openbaton.github.io/documentation/zabbix-plugin/, Accessed: Sept. 2019.

[30] C. Midoglu at al., “MONROE-Nettest: A configurable tool for dissecting speed measurements in mobile broadband networks”, Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS’2018), Honolulu, HI, USA, Apr. 2018, pp. 342–347.

[31] 5GENESIS, Deliverable D6.1 “Trials and Experimentations - Cycle 1” [Online], https://5genesis.eu/wp-content/uploads/2019/08/5GENESIS_D6.1_v1.00.pdf, Accessed: Sept. 2019.

[32] R. Hofsted et al., “Flow Monitoring Explained: From Packet Capture to Data Analysis with NetFlow and IPFIX”, IEEE Communications Surveys and Tutorials, vol. 15, no. 4, 2014, pp. 2037–2064.

[33] A. Imran, A. Zoha, and A. Abu-Dayya, “Challenges in 5G: How to Empower SON with Big data for Enabling 5G”, IEEE Network, vol. 28, no. 6, 2014, pp. 27–33.

[34] “O-RAN Alliance” [Online], https://www.o-ran.org/resources/, Accessed: Sept. 2019.

[35] J. Hagerty, R. L. Sallam, and J. Richardson, “Magic Quadrant for Business Intelligence Platforms.” [Online], http://www.microstrategy.com/download/files/whitepapers/open/gartnermagic-quadrant-for-bi-platforms-2012.pdf, Accessed: Sept. 2019.

[36] “Demistifying Network Analytics” [Online], https://www.networkworld.com/article/3190857/demystifying-network-analytics.html, Accessed: Sept. 2019.

[37] R. Gonzalez et al., “Net2Vec: Deep Learning for the Network”, Proc. ACM Workshop on Big Data Analytics and Machine Learning for Data Communication Networks (Bi-DAMA’17), Los Angeles, CA, USA, Aug. 2017, pp. 13–18.

[38] “Apache Kafka” [Online], https://kafka.apache.org/, Accessed: Sept. 2019.

[39] “Apache Spark” [Online], http://spark.apache.org/streaming/, Accessed: Sept. 2019.

[40] “Apache Storm” [Online], https://storm.apache.org, Accessed: Sept. 2019.

[41] P. Carbone et al., “Apache flink: Stream and batch processing in a single engine”, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 36, no. 4, 2015.

[42] D. Sarlis et al., “Datix: A system for scalable network analytics”, ACM SIGCOMM Computer Communication Review, vol. 45, no. 5, 2015, pp. 21–28.

[43] Y. Lee and Y. Lee, “Toward scalable internet traffic measurement and analysis with Hadoop”, ACM SIGCOMM Computer Communication Review, vol. 43, no. 1, 2013, pp. 5–13.

[44] V. K. Bumgardner and V. W. Marek, “Scalable Hybrid Stream and Hadoop network analysis system”, Proc. ACM/SPEC 5th International Conference on Performance Engineering (ICPE’14), Dublin, Ireland, March 2014, pp. 219–224.

[45] D. Simoncelli, M. Dusi, F. Gringoli, and S. Niccolini, “Stream-monitoring with BlockMon: convergence of network measurements and data analytics platforms”, ACM SIGCOMM Computer Communication Review, vol. 43, no. 2, 2013, pp. 29–36.

[46] “scikit-learn” [Online], https://scikit-learn.org/, Accessed: Sept. 2019.

[47] “PyTorch” [Online], https://pytorch.org, Accessed: Sept. 2019.

[48] “TensorFlow” [Online], https://www.tensorflow.org, Accessed: Sept. 2019.

[49] “pnda” [Online], http://pnda.io, Accessed: Sept. 2019.

[50] “Elastic Stack” [Online], https://www.elastic.co/products/elastic-stack, Accessed: Sept. 2019.

[51] “InfluxData” [Online], https://www.influxdata.com/products/, Accessed: Sept. 2019.

[52] 5GENESIS, Deliverable D2.2 “5GENESIS Overall Platform Design and Specifications ” [Online], https://5genesis.eu/wp-content/uploads/2018/12/5GENESIS_D2.2_v1.0.pdf, Accessed: Sept. 2019.

[53] 5GENESIS, Deliverable D2.3 “Initial Planning of Tests and Experimentations” [Online], https://5genesis.eu/wp-content/uploads/2019/02/5GENESIS_D2.3_v1.0.pdf, Accessed: Sept. 2019

[54] “Grafana Labs” [Online], https://grafana.com, Accessed: Sept. 2019.

[55] “Test Automation Platform” [Online], https://www.keysight.com/en/pc-2873415/test-automation-platform-tap, Accessed: Sept. 2019.

[56] “InfluxDB” [Online], https://www.influxdata.com/products/influxdb-overview/, Accessed: Sept. 2019.

[57] “OpenAirInterface” [Online], https://www.openairinterface.org, Accessed: Sept. 2019.

[58] "OpenAirInterface5G – T-Tracer" [Online], https://gitlab.eurecom.fr/oai/openairinterface5g/tree/develop-nr/common/utils/T, Accessed: Sept. 2019.

[59] "LibreNMS" [Online], https://www.librenms.org, Accessed: Sept. 2019.

[60] "MONROE Alliance" [Online], https://www.monroe-project.eu, Accessed: Sept. 2019.

[61] Ö. Alay et al., "Experience: An open platform for experimentation with commercial mobile broadband networks", Proc. ACM 23rd Annual International Conference on Mobile Computing and Networking (MobiCom'17), Snowbird, UT, USA, Oct. 2017, pp. 70–78.

[62] V. Mancuso et al., "Results from running an experiment as a service platform for mobile broadband networks in Europe", Computer Communications, vol. 133, 2019, pp. 89–101.

[63] "Internet 2 - One-way Ping (OWAMP)" [Online], https://software.internet2.edu/owamp/, Accessed: Sept. 2019.

[64] "IXIA A Keysight business – IxChariot" [Online], https://www.ixiacom.com/products/ixchariot, Accessed: Sept. 2019.

[65] P. Dix, "InfluxDB now supports Prometheus remote Read & Write natively" [Online], https://www.influxdata.com/blog/influxdb-now-supports-prometheus-remote-read-write-natively/, Accessed: Sept. 2019.

[66] "Zabbix + InfluxDB" [Online], https://www.zabbix.com/integrations/influxdb, Accessed: Sept. 2019.

[67] "Using InfluxDB in Grafana" [Online], https://grafana.com/docs/features/datasources/influxdb/, Accessed: Sept. 2019.

[68] "Telegraf 1.11 documentation" [Online], https://docs.influxdata.com/telegraf/v1.11/, Accessed: Sept. 2019.

[69] "Python client for InfluxDB" [Online], https://github.com/influxdata/influxdb-python, documentation at https://influxdb-python.readthedocs.io/en/latest/, Accessed: Sept. 2019.

[70] "Influx Query Language (InfluxQL)" [Online], https://docs.influxdata.com/influxdb/v1.7/query_language/, Accessed: Sept. 2019.

[71] "Jupyter"[Online], https://jupyter.org, Accessed: Sept. 2019.

[72] "Plotly|Dash" [Online], https://plot.ly/dash/, Accessed: Sept. 2019.

[73] "Prometheus – Node Exporter" [Online], https://github.com/prometheus/node_exporter, documentation at: https://prometheus.io/docs/instrumenting/exporters/, Accessed: Sept. 2019.

[74] “Developers – CellSignalStrengthLTE” [Online], https://developer.android.com/reference/android/telephony/CellSignalStrengthLte, Accessed: Sept. 2019.

[75] “InfluxDB line protocol reference” [Online], https://docs.influxdata.com/influxdb/v1.7/write_protocols/line_protocol_reference/#syntax, Accessed: Sept. 2019.

[76] “Crontab – Quick Reference” [Online], https://www.adminschoice.com/crontab-quick-reference, Accessed: Sept. 2019.

[77] “TRIANGLE Project – 5G Applications and Devices Benchmarking” [Online], https://www.triangle-project.eu, Accessed: Sept. 2019.

[78] “OpenAirInterface5G – T_messages” [Online], https://gitlab.eurecom.fr/oai/openairinterface5g/blob/develop-nr/common/utils/T/T_messages.txt, Accessed: Sept. 2019.

[79] “A One-way Active Measurement Protocol (OWAMP)” [Online], http://www.rfc-editor.org/rfc/rfc4656.txt, Accessed: Sept. 2019.

[80] “Welcome to APEX – The Adaptive Policy eXecution (Engine)” [Online], https://ericsson.github.io/apex-docs/, Accessed: Sept. 2019.

[81] 5GENESIS, Deliverable D2.1 “Requirements of the Platform” [Online], https://5genesis.eu/wp-content/uploads/2018/11/5GENESIS_D2.1_v1.0.pdf, Accessed: Sept. 2019.

[82] C. Midoglu, Ö, Alay, and C. Griwodz, “Evaluation Framework for Real-Time Adaptive 360-Degree Video Streaming over 5G Networks”, Proc. ACM 25th Annual International Conference on Mobile Computing and Networking (MobiCom’19), Wireless of the Students, by the Students, and for the Students 2019 Workshop (S3 ’19), Los Cabos, Mexico, Oct. 2019

# ANNEX 1 – M&A STANDARDIZATION ACTIVITIES

The need for a M&A framework is also highlighted by the ongoing standardization efforts related to 5G. Among others, ETSI has recently created an industry specification group called Experimental Network Intelligence (ENI), that defines an AI-oriented management architecture based on monitoring-enabled context- awareness, to support network operators in automating their systems [A1_1]. Moreover, 3GPP continues to promote several activities towards the introduction of M&A functionalities within the Service-Based Architecture (SBA) envisioned for 5GC, and standardized since Release 15 [A1_2]. SBA defines different Network Functions (NFs) communicating with each other as originators or consumers of services, via so-called Service-Based Interfaces (SBIs); in the direction of introducing a complete M&A instance, the Network Data Analytics Function (NWDAF) has been integrated in SBA in Release 15 (TS 23.501), aiming to provide data and related analytics to multiple system domains, including slices, NFs, and also Application Functions (AFs). NWDAF enables NFs to access data and analytics for different purposes, including optimized slice selection and control, and reconfiguration of system policies, being in these cases the originator of services consumed by the Network Slice Selection Function (NSSF) and the Policy Control Function (PCF), respectively [4]. However, while NWDAF appears mainly focused on network-related services, the Management Data Analytic Function (MDAF) has also been proposed by 3GPP SA5 (Telecom Management) as originator of services (MDASs) consumed at the MANO layer of the architecture [A1_3, Section 2.2.4].

In the context of the incoming Release 16, 3GPP SA2 (Architecture) has started a new study item referred to as FS_eNA, which analyzes how to enable “Network Automation for 5G”, and thus further clarify the usage of data analytics at the network layer [A1_4]. As mentioned above, the main role of NWDAF is to deliver relevant analytics for slice management and traffic steering/splitting, reusing similar service exposure mechanisms as other NFs [A1_5]; on top of this, FS_eNA envisions further use cases, including but not limited to NWDA-assisted QoS provision/adjustment, customized mobility management, edge computing, load balancing, and network performance prediction [3][4]. Moreover, since NWDAF services are also exposed towards AFs, they can be used in more specific scenarios, such as mMTC management and security-sensitive use cases. FS_eNA is also discussing so-called “UE-driven analytics sharing”, being the data coming from UEs the main enablers for UE-centric analytics, which can possibly trigger further performance optimization, at both user and network sides. On similar aspects, 3GPP has defined two further study items in SA5 and RAN3 respectively, namely “Study on SON for 5G” [A1_6] and “RAN-Centric Data Collection and Utilization for Long-Term Evolution (LTE) and NR” [A1_7], which mainly address data analytics applied to access network optimization.

## References

[A1_1] “New ETSI group on improving operator experience using Artificial Intelligence” [Online], http://www.etsi.org/news-events/news/1171-2017-02-new-etsi-group-on-improving-operator-experience-usin, Accessed: Sept. 2019.

[A1_2] “System Architecture for the 5G System”, v15.3.0, TS 23.501, 3GPP, Sep. 2018.

[A1_3] “5G Mobile Network Architecture for diverse services, use cases, and applications in 5G and beyond” - 5G-MONARCH, Deliverable D2.3 “Final overall architecture”, Apr. 2019

[A1_4] “Study of Enablers for Network Automation for 5G”, v2.0.0, TR 23.791, 3GPP, Dec. 2018.

[A1_5] “Study on Access Traffic Steering, Switching and Splitting Supporting the 5G System Architecture”, v0.1.0, TR 23.793, 3GPP, Aug. 2017.

[A1_6] “New Study on Self-Organizing Networks (SON) for 5G”, v0.2.0, TR 28.861, 3GPP, Dec. 2018.

[A1_7] “Study on RAN-Centric Data Collection and Utilization for LTE and NR”, RP-182105, 3GPP, Sep. 2018.

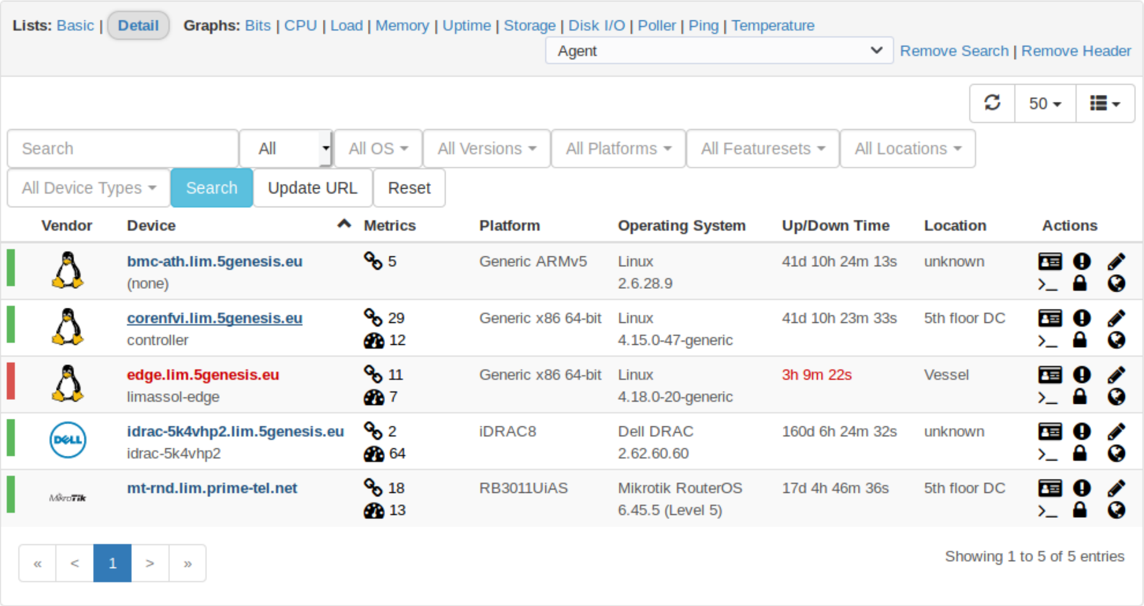
# ANNEX 2 – FURTHER IM TOOLS

## LibreNMS

LibreNMS is a full featured network management system [59]. LibreNMS is an autodiscovering PHP/MySQL/SNMP based network monitoring suite which includes support for a wide range of network hardware and operating systems. It provides listeners that either via automatic or manual configuration collect metrics from the infrastructure elements. Then, specific dashboards similar to Grafana may be created, allowing the visualization of the collected metrics. The software platform beside visualization of the metrics allows collection of Link Layer Discovery Protocol (LLDP) and Cisco Discovery Protocol (CDP) messages in order to form the physical topology of the infrastructure elements and provide node and link information. Moreover, it provides the ability to create alerts and trigger actions based on specific filters.

LibreNMS is used to gather and store measurement data and syslog from the network, the operating systems and the hardware devices used in 5GENESIS Limassol platform (Figure A2_F1).

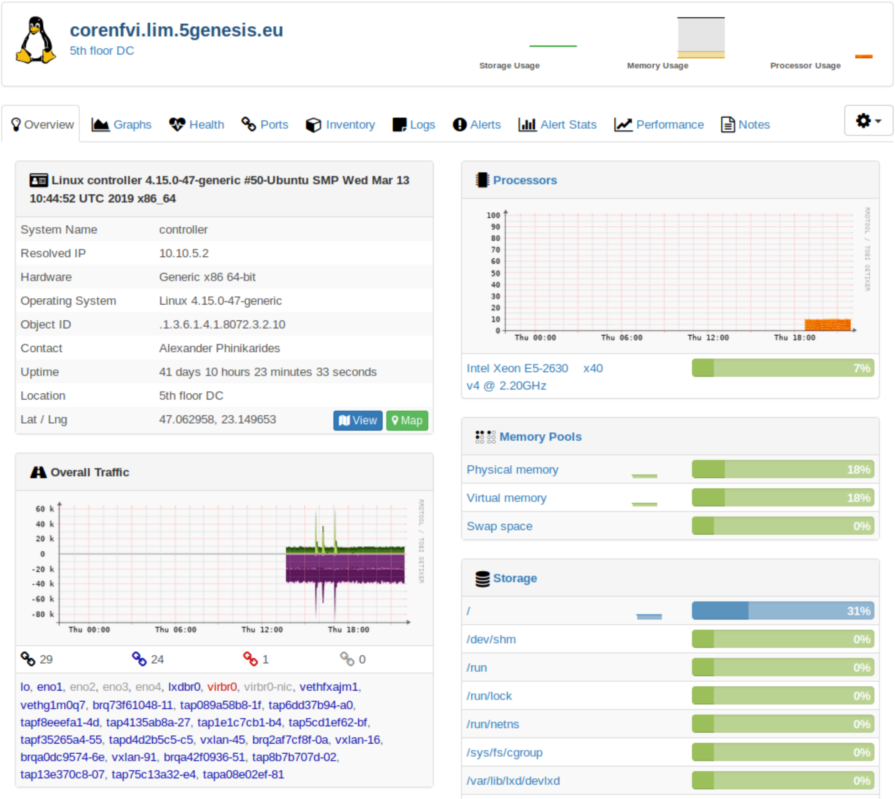
The parameters that are measured are derived from metrics exposed through SNMP. The measured parameters include network interface metrics (such as packets/bytes per second, packet errors, IP / TCP / UDP / ICMP statistics), hardware metrics (such as CPU / RAM / disk usage, disk I/O and temperatures) and operating system metrics (such as running processes, load averages and general system I/O) (Figures A2_F2 and A2_F3).



| | Vendor | Device | Metrics | Platform | Operating System | Up/Down Time | Location | Actions |
|---|---|---|---|---|---|---|---|---|
| 🟩 | | bmc-ath.lim.5genesis.eu<br>(none) | 5 | Generic ARMv5 | Linux<br>2.6.28.9 | 41d 10h 24m 13s | unknown | |
| 🟩 | | corenfvi.lim.5genesis.eu<br>controller | 29<br>12 | Generic x86 64-bit | Linux<br>4.15.0-47-generic | 41d 10h 23m 33s | 5th floor DC | |
| 🟥 | | edge.lim.5genesis.eu<br>limassol-edge | 11<br>7 | Generic x86 64-bit | Linux<br>4.18.0-20-generic | 3h 9m 22s | Vessel | |
| 🟩 | DELL | idrac-5k4vhp2.lim.5genesis.eu<br>idrac-5k4vhp2 | 2<br>64 | iDRAC8 | Dell DRAC<br>2.62.60.60 | 160d 6h 24m 32s | unknown | |
| 🟩 | MikroTik | mt-rnd.lim.prime-tel.net | 18<br>13 | RB3011UiAS | Mikrotik RouterOS<br>6.45.5 (Level 5) | 17d 4h 46m 36s | 5th floor DC | |

Showing 1 to 5 of 5 entries

Figure A2_F1. LibreNMS-based monitoring of 5GENESIS devices in Limassol platform



corenfvi.lim.5genesis.eu
5th floor DC

Linux controller 4.15.0-47-generic #50-Ubuntu SMP Wed Mar 13 10:44:52 UTC 2019 x86_64

| | |
|---|---|
| System Name | controller |
| Resolved IP | 10.10.5.2 |
| Hardware | Generic x86 64-bit |
| Operating System | Linux 4.15.0-47-generic |
| Object ID | .1.3.6.1.4.1.8072.3.2.10 |
| Contact | Alexander Phinikarides |
| Uptime | 41 days 10 hours 23 minutes 33 seconds |
| Location | 5th floor DC |
| Lat / Lng | 47.062958, 23.149653 |

29 24 1 0

lo, eno1, eno2, eno3, eno4, lxdbr0, virbr0, virbr0-nic, vethfxajm1, vethg1m0q7, brq73f61048-11, tap089a58b8-1f, tap6dd37b94-a0, tapf8eeefa1-4d, tap4135ab8a-27, tap1e1c7cb1-b4, tap5cd1ef62-bf, tapf35265a4-55, tapd4d2b5c5-c5, vxlan-45, brq2af7cf8f-0a, vxlan-16, brqa0dc9574-6e, vxlan-91, brqa42f0936-51, tap8b7b707d-02, tap13e370c8-07, tap75c13a32-e4, tapa08e02ef-81

| | | |
|---|---|---|
| Intel Xeon E5-2630 v4 @ 2.20GHz | x40 | 7% |

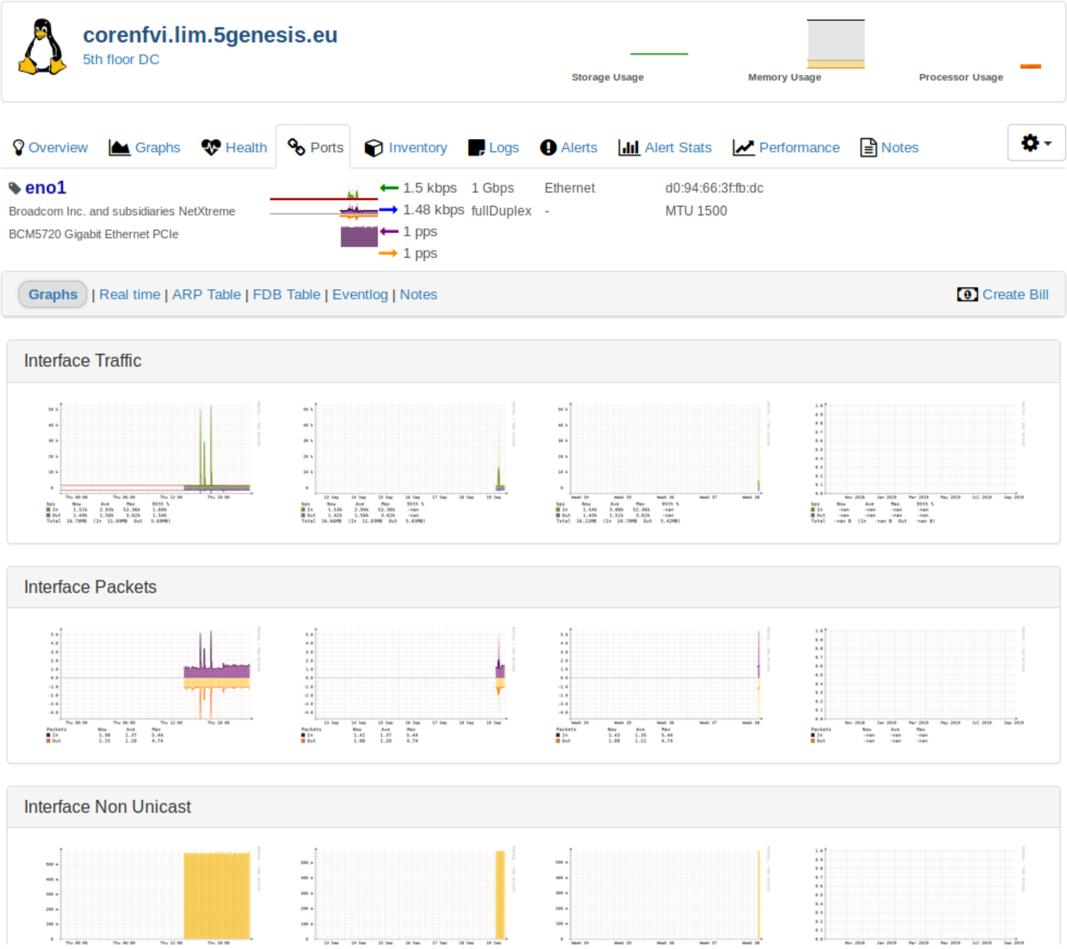
Memory Pools

| | |
|---|---|
| Physical memory | 18% |
| Virtual memory | 18% |
| Swap space | 0% |

Storage

| | |
|---|---|
| / | 31% |
| /dev/shm | 0% |
| /run | 0% |
| /run/lock | 0% |
| /run/netns | 0% |
| /sys/fs/cgroup | 0% |
| /var/lib/lxd/devlxd | 0% |

Figure A2_F2. LibreNMS-based monitoring of the core NFVI in Limassol platform



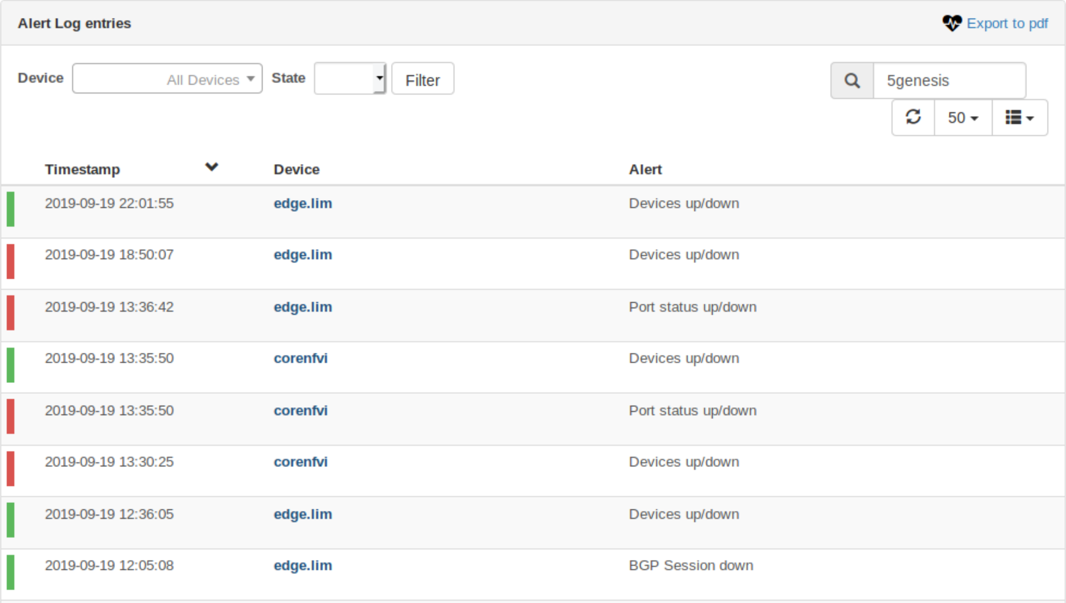
corenfvi.lim.5genesis.eu
5th floor DC

eno1
Broadcom Inc. and subsidiaries NetXtreme BCM5720 Gigabit Ethernet PCIe

| | | | |
|---|---|---|---|
| ← 1.5 kbps | 1 Gbps | Ethernet | d0:94:66:3f:fb:dc |
| → 1.48 kbps | fullDuplex | - | MTU 1500 |
| ← 1 pps | | | |
| → 1 pps | | | |

Figure A2_F3. Core NFVI network interface monitoring in Limassol platform

Besides monitoring, LibreNMS has a rich alerting system that can send real-time notifications when an alert is triggered (Figure A2_F4) and real-time network interface bandwidth polling.



Alert Log entries

| | Timestamp | Device | Alert |
|---|---|---|---|
| 🟩 | 2019-09-19 22:01:55 | edge.lim | Devices up/down |
| 🟥 | 2019-09-19 18:50:07 | edge.lim | Devices up/down |
| 🟥 | 2019-09-19 13:36:42 | edge.lim | Port status up/down |
| 🟩 | 2019-09-19 13:35:50 | corenfvi | Devices up/down |
| 🟥 | 2019-09-19 13:35:50 | corenfvi | Port status up/down |
| 🟥 | 2019-09-19 13:30:25 | corenfvi | Devices up/down |
| 🟩 | 2019-09-19 12:36:05 | edge.lim | Devices up/down |
| 🟩 | 2019-09-19 12:05:08 | edge.lim | BGP Session down |

Figure A2_F4. Alerts triggered in Limassol platform

LibreNMS also has a plugin system which allows the integration of external services. In the Limassol platform, these plugins provide network mapping capabilities (Figure A2_F5) and automated configuration backup for network devices (Oxidized).

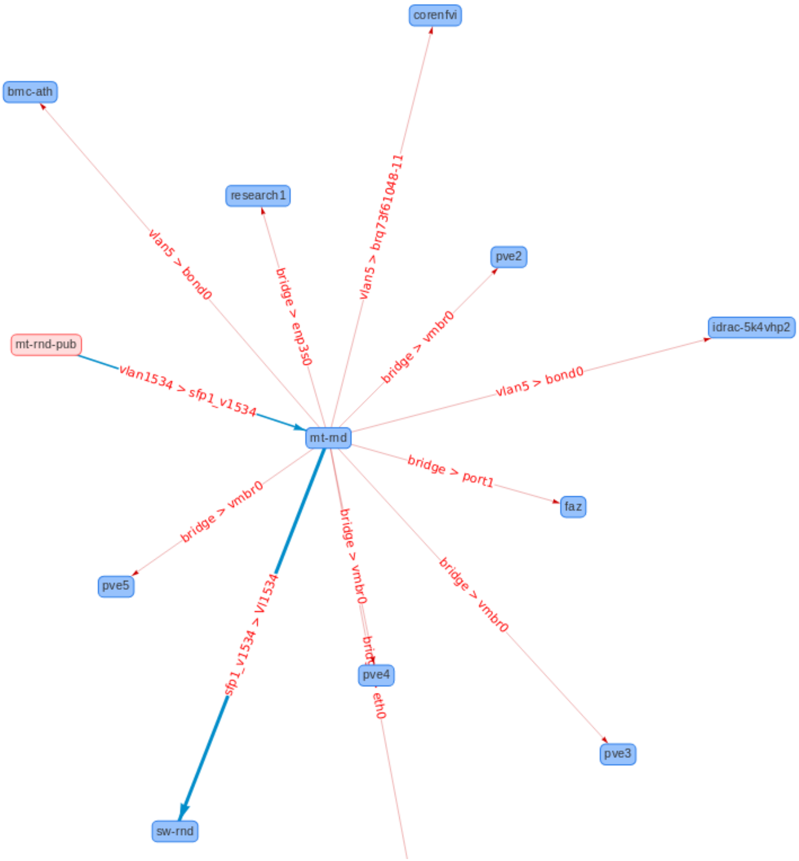


Figure A2_F5. Limassol platform network map as seen by LibreNMS

## WiFi Monitoring

There are many factors that can affect the quality of a WiFi network connection. To get a better understanding about how this connection is behaving over a period of time, the reporting section of the WiFi Service Management Platform (WSMP) is considered. WSMP is a product by FON used to manage, operate, and monetize WiFi services from a single platform. It includes multiple authentication methods (Captive portal, 801.2x, EAP SIM, AKA, AKA', TTLS, WISPr), available as cloud-based, on premises, or hybrid. Thanks to its REST API-based approach, it can be easily integrated and adapted. Through the console component, WSMP provides end-to-end visibility of the network hierarchy, offering several metrics and insights of the managed network. One of the main features is the reporting section. It is designed to give a visual and interactive information about the platform usage.

Several parameters of a specific network or node can be assessed, selecting the metrics to show over a period of time. The metrics groups below are currently supported by WSMP and are available:

- Overview: Summary of the network topology in Nodes and APs, unique users, sessions, and traffic.
- User sessions: User sessions, time, traffic stats, and demographic information.
- Login metrics: Login attempt statistics categorized by type, success, failures, errors returned, etc.
- Access points: Access point and session statistics
- User metrics: User connections and registration statistics
- Network hierarchy: Network topology statistics at the Node level
- Pass sales: Pass sale and usage statistics
- Device metrics: Breakdown of the number of sessions based on the device type, operating system and browser used by the end customer.

By default, the generated report includes an overall view of the Network's/Node's most important metrics, which includes:

- Service usage metrics: User sessions, new registered users, average session time, average session traffic
- Sales metrics: Total sales, refunds, sale products.
- Demographic summary: device types, user gender, age, languages
- Most used AP rank: 10 APs with the most sessions in a Network

In order to integrate WSMP at least one access controller must be created and configured. One per hardware manufacturer is needed. In order to do so, each device needs to go through a AAA process and then be associated to a specific AP. FON has already created the access controllers for the main pre-integrated hardware vendors. These can easily be selected from a list of pre-created controllers. WSMP lets administrators configure secure connections with external endpoints such as radius platforms or other integrated systems. This ensures that if customers are required to encrypt traffic flowing between their backend and WSMP, they will be able to configure connections as required. With WSMP, administrators can define IPSec tunnels and secure associations within those tunnels, as well as define the necessary parameters for the secure connection. In terms of 5GENESIS, the definition of a REST API-based interface exposing the available metrics is being assessed.

# ANNEX 3 – FURTHER PM TOOLS

## IxChariot

IxChariot is a commercial product from Keysight (after IXIA was acquired) used to assess network infrastructures and deployments [A3_R1]. IxChariot uses software agents called Performance Endpoints to simulate application traffic and deliver key performance metrics to a central console for easy management. IxChariot confidently assesses the performance and reliability of networks and applications running on a wide variety of transport interfaces, including wired, wireless, and virtual data centers.

In 5GENESIS, IxChariot will be used for simulating video, audio, and data traffic across the Athens Platform. The provided set of Performance Endpoints includes major OS vendors and terminal types ranging from Linux, MacOS and Windows to Android and iOS as well as virtualized instances to be used in data centers. The output of each experiment with IxChariot provides a report like document with all the metrics that are calculated per endpoint and per traffic flow. In addition, the raw measurements may be exported as xls or csv file in order to be further processed. Integrating IxChariot in the 5GENESIS M&A framework will include efforts for collecting raw measurements and delivering them at the InfluxDB database via a dedicated TAP Plugin.

## Open 5GCore Benchmarking Tool

Benchmarking in general aims to validate non-functional properties of a system (e.g. performance, efficiency, stability or resilience).

The Open 5GCore developed by FhG includes a benchmarking tool[7] (BT), which can be used to evaluate various non-functional aspects of the 5GC. There are several requirements for the Open 5GCore benchmarking tool: in particular, it has to emulate realistic user group behavior, including signaling and data traffic, and also monitor and measure the performance metrics for the system under test. For this reason, the Open 5GCore benchmarking tool is capable of performing UE attachment, detachment, service de/activation and handover operations for evaluating the system. Furthermore, it can emulate network traffic according to a predefined template. In is possible to define different scenarios for the benchmarking tool, which are then loaded from a configuration file or a database. The scenarios define the type (attachment, detachment, etc.), rate (number of operations per second), and ratio (e.g., 75% attachments, 25% handovers) of UE operations to be performed.

To incorporate the benchmarking tool into the 5GENESIS platform, FhG developed a TAP plugin which handles the automation of the benchmarking process. It creates two TAP steps, one for sending a request to start the benchmarking tool, and another step to retrieve the collected statistics. The steps are called ‘Start BT Test’ and ‘Get BT Statistics’. Table  summarizes the parameters that can be passed in the ‘Start BT Test’ step in order to specify which operations the benchmarking tool shall execute. If the request to start the benchmarking tool is successful, the TAP steps wait for the duration it estimates the benchmark to take.

[7] https://gitlab.fokus.fraunhofer.de/5genesis/berlin-platform/tree/develop/tap-plugins/5GCore_benchmarking_tools

Table A3_T1: Benchmarking Tool TAP Plugin Start Step Parameters

| Parameter | Type | Description |
|---|---|---|
| Number of attachments | Positive integer | How many simulated UEs shall be attached during the benchmark. Cannot exceed the number of preconfigured unregistered UEs available to the BT. |
| Number of detachments | Positive integer | How many UEs shall be detached during the benchmark.<br>Cannot exceed the number of UEs attached during the benchmark plus the number of already registered UEs. |
| Number of handovers | Positive integer | How many handovers between virtual cells should be performed. |
| Service deactivation | Positive integer | The number of service deactivation requests (UE idle mode) to send during the benchmark. |
| Service activation | Positive integer | The number of service activation requests (UE active mode) to send during the benchmark. |
| Operations per second | Positive integer | Determines the number of operations per second. |

The “Get BT Statistics” TAP step sends a request to the benchmarking tool and receives the benchmarking results as a response.

## References

[A3_R1] “IXIA A Keysight business – IxChariot Server and Desktop Editions (Data Sheet)” [Online], https://www.ixiacom.com/sites/default/files/2017-08/Ixia-T-DS-IxChariot.pdf, Accessed: Sept. 2019.

# ANNEX 4 – NB-IOT SIMULATOR

An important KPI to be investigated in the Surrey platform is IoT energy efficiency. To better understand the current impact of different system parameters and help guide future 5G measurements, KAU has developed a simulator in the OMNeT++ framework [A4_R1] that facilitates the estimation of the energy consumption of NarrowBand-IoT (NB-IoT) devices in a LTE network. NB-IoT network components covered by the simulator are shown in Figure A4_F1.

The NB-IoT device together with the eNB constitute the Evolved UTRAN (E-UTRAN), i.e., the LTE RAN. The eNB hosts functions such as dynamic resource allocation and radio resource management. The components between the S1-MME and SGi interfaces comprise the Evolved Packet Core (EPC). As follows, the EPC consists of the Mobility Management Entity (MME), the Home Subscriber Server (HSS), the Serving Gateway (SGW), and the Packet Data Network Gateway (PGW). The MME manages user equipment context, identification, and authorization; the HSS holds subscription profiles and security-related information for each registered subscriber; the SGW terminates the downlink data path for user equipment in the LTE Idle state, and initiates paging to the MME when downlink data arrive for the user equipment, e.g., an NB-IoT device. Finally, the PGW is the connection between the EPC and external packet data networks, such as the Internet, over the SGi interface. It is responsible for the allocation of user equipment IP addresses. The PGW also acts as the user-plane anchor for mobility to/from IP-access networks.

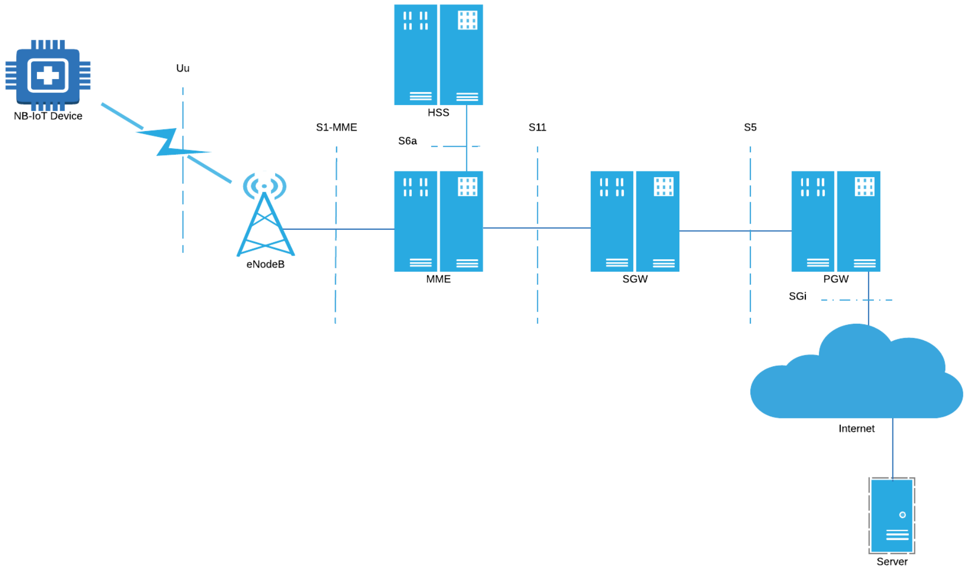


Figure A4_F1. The components of an NB-IoT network covered by the KAU NB-IoT simulator

Although the KAU NB-IoT simulator essentially covers all parts of an NB-IoT communication scenario, it only models the Uu interface (the radio interface) in some detail. Figure A4_F2 depicts the Uu interface together with the implemented IoT protocol stack.

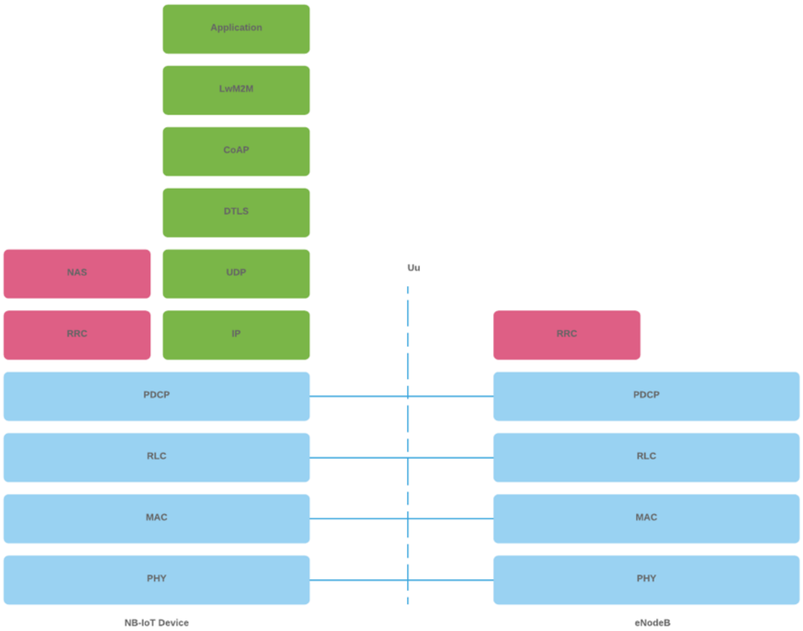


Figure A4_F2. The NB-IoT protocol stack over the Uu interface. The green-colored protocols are parts of the IoT protocol stack; the red-colored protocols comprise the control plane of the LTE radio interface protocol stack, and the blue-colored protocols are common to both the user and control planes of the LTE radio interface protocol stack

The IoT protocol stack comprises the green-colored protocols. The Lightweight Machine to Machine (LwM2M) protocol is an application protocol that provides a link between a NB-IoT device and an LwM2M-enabled server. The LwM2M protocol lets users remotely perform tasks, run diagnostics, and perform application and device management.

The LwM2M runs atop the Constrained Application Protocol (CoAP) [A4_R2], an application protocol that offers a Web-like interface towards NB-IoT devices. Similar to HyperText Transport Protocol (HTTP), CoAP is a document transfer protocol, however, in contrast to HTTP, CoAP specifically targets constrained devices such as NB-IoT devices. CoAP offers a reliable transport service to LwM2M, and employs a retransmission-based recovery mechanism to realize this service. Since CoAP runs on top of UDP, not TCP, SSL/TSL are not available to provide security. Instead, CoAP makes use of DTLS.

The KAU NB-IoT simulator implements the basic parts of the LwM2M, DTLS, UDP, and IP layers, however, implements the CoAP layer in some detail. Particularly, since it is believed to have a significant effect on the energy consumption, the CoAP retransmission-based recovery mechanism is modeled in detail.

The LTE Uu interface protocol stack is divided into control plane (red-colored protocols) and user plane protocols, both of which runs atop a common set of radio interface protocols (blue-colored protocols). The control plane protocols correspond to the non-access stratum for communication between a NB-IoT device and MME, while the user plane protocols are part of the access stratum that transports data across the radio interface.

The topmost part of the control plane comprises the Non-Access Stratum (NAS) protocols of which the EPC Mobility Management (EMM) and EPC Connection Management (ECM) are the two most important ones. EMM manages device mobility, while ECM is responsible for the establishment of a control signaling connection between a NB-IoT device and MME.

The RRC layer is responsible for all layer three control signaling exchange between a NB-IoT device and eNB. For example, RRC manages radio bearers, mobility, and NB-IoT device measurement reporting. It is also responsible for broadcasting system information.

The radio interface protocols encompass PDCP, RLC protocol, and MAC protocols. Packets are delivered to the NB-IoT device from eNB using PDCP. The PDCP protocol is responsible for header compression, access stratum security, and ordered delivery of user plane packets during handover.

The RLC protocol operates in three modes: Transparent Mode (TM), Unacknowledged Mode (UM), and Acknowledged Mode (AM). It is responsible for error correction of transmitted packets through ARQ (only in AM mode), segmentation, and in-order delivery. The RLC protocol runs over the MAC protocol, the main functions of which are to perform radio resource allocation through dynamic scheduling and to handle error correction through HARQ.

The KAU NB-IoT simulator fairly accurately implement all radio interface protocols. In particular, it accurately models the buffer management taking place at the PDCP layer and the ARQ and HARQ mechanisms in the RLC and MAC layers.

Since the KAU NB-IoT simulator targets the estimation of the energy consumption of NB-IoT devices, this is a key part of the simulator. The major energy consuming activity for the NB-IoT device is to keep the radio interface up; especially the transmission of packets is very energy consuming. As a way to reduce the energy consumption of the radio interface, NB-IoT introduces two energy-saving mechanisms, the Power Saving Mode (PSM) and the Discontinuous Reception (DRX), both of which requires a RRC connection between the NB-IoT device and eNB. An RRC connection operates in either of two states: Connected and Idle. A NB-IoT device in RRC Connected has an active RRC connection, and thus a packet transfer may take place, which is not the case in RRC Idle. Each time a packet is transmitted, an inactivity timer is started, and when the inactivity timer expires, the RRC connection enters the RRC Idle state. The energy consumption in RRC Idle is dictated by a timer, the Activity timer, T3324; the expiration of the Active timer triggers PSM.

PSM permits a NB-IoT device in RRC Idle state to enter deep sleep, and basically go down to a base energy consumption. A NB-IoT device in PSM is unreachable from the mobile network and do no become reachable until an application on the NB-IoT device wants to transmit data or a timer, the Tracking Area Update (TAU) timer, T3412, which periodically notifies the availability of the NB-IoT device to the mobile network, expires.

DRX is the mechanism through which the mobile network and a NB-IoT device negotiate when the device can sleep and works in both the RRC Connected and Idle states. In both states, DRX makes the device periodically sleep and periodically listen for data receptions, however, during RRC Idle the sleeping periods are much longer. Particularly, during RRC Idle we talk about an extended DRX (eDRX), which comprises a long sleeping period followed by a so-called Paging Time Window (PTW), during which the length of the sleeping periods is the same as during RRC Connected. Figure A4_F3 illustrates the energy consumption of a NB-IoT device in the two RRC states and how it is minimized by PSM and DRX.

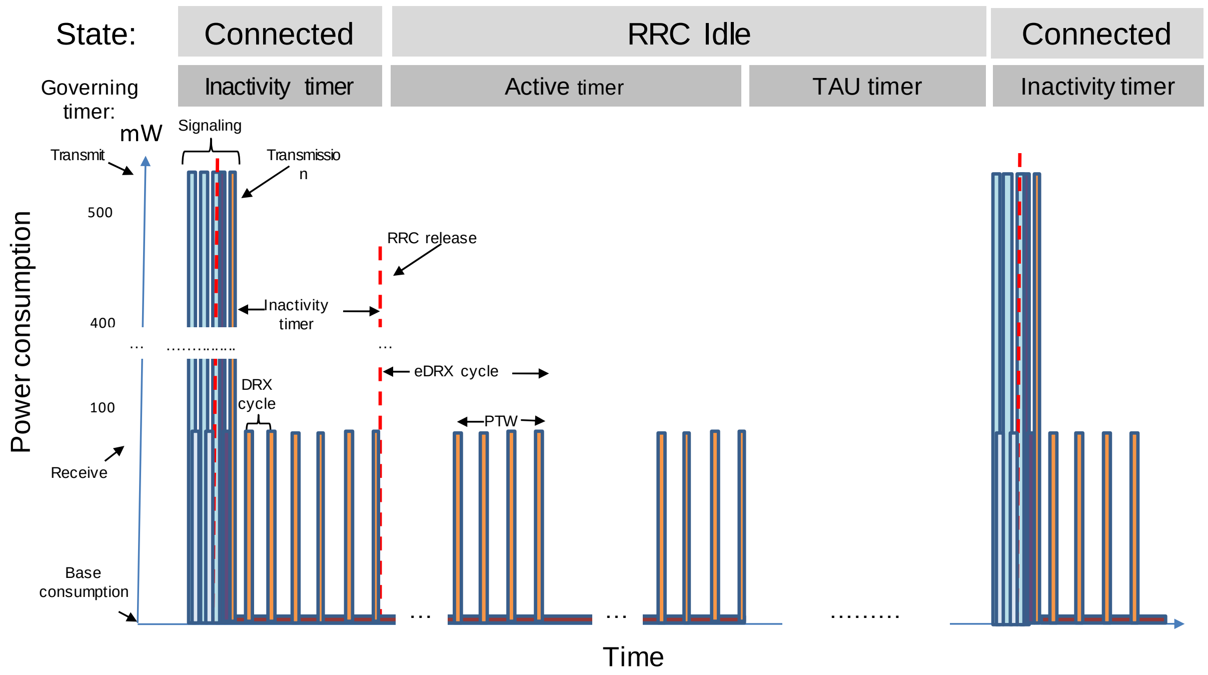


Figure A4_F3. The energy consumption of a NB-IoT device over time

## Case Study: Evaluation of Energy Consumption

An initial evaluation of the energy consumption of NB-IoT devices in a LTE network has been conducted by KAU using its purpose-built NB-IoT simulator. The study focused on a stationary NB-IoT device in a loss-free radio environment that transmits data according to a pattern captured from a trial NB-IoT deployment. Four energy consumption scenarios could be identified (Figure A4_F4):

- Ideal: The CoAP ACK returns before the RRC connection enters the Idle state.
- Narrow Connected (N_C): The CoAP ACK does not return before the RRC connection enters the Idle state and thus a reeastablishment of the RRC connection needs to take place before eNB can transmit the CoAP ACK to the NB-IoT device
- CoAP Short (CoAP_S): The CoAP retransmission timer is too tighly configured which results in a timeout and a spurious retransmission. Every transmission and retransmission result in an acknowledgement, which may arrive after the RRC connection has been released, and may result in another signaling phase to reestablish the radio connection
- CoAP Between (CoAP_B): The CoAP timer expires while the RRC connection is in the Idle state, resulting in an additional signaling phase triggered by the NB-IoT device. Additional signaling sessions may be triggered by the network to facilitate the reception of acknowledgements; one for each transmission/retransmission

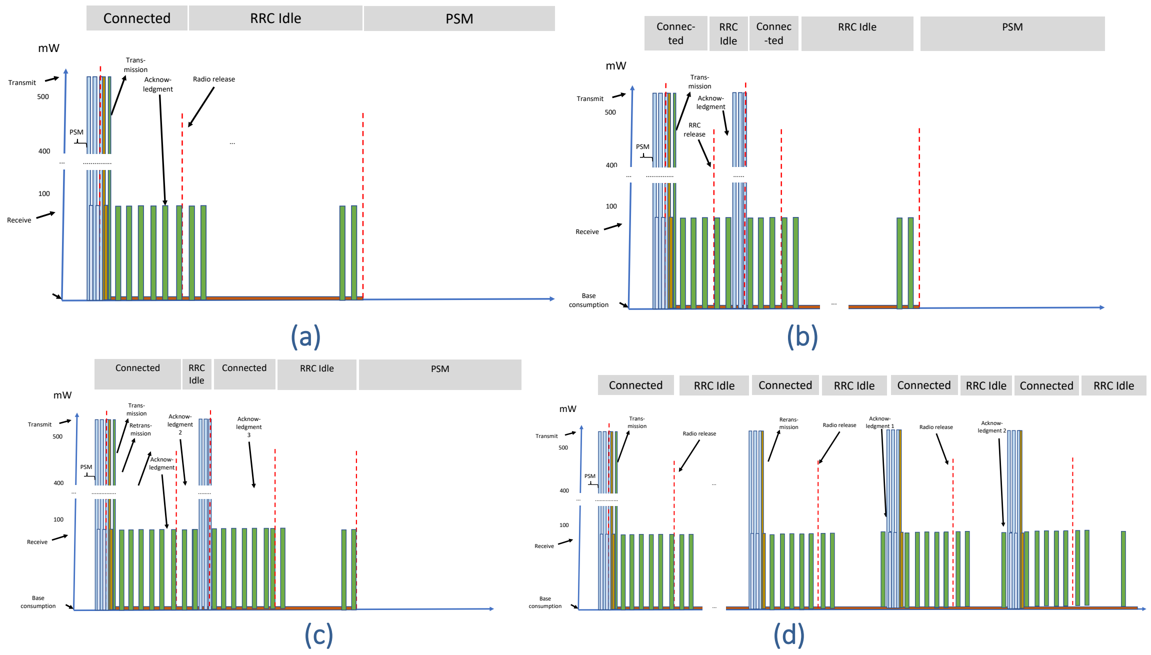


Figure A4_F4. The four NB-IoT energy consumption scenarios studied: Ideal (a), Narrow Connected (N_C) (b), CoAP Short (CoAP_S) (c), and CoAP Between (CoAP_B) (d)

Figure A4_F5 illustrates the outcome of the initial energy consumption evaluation. It can be observed that additional RRC connection establishments have a substantial effect on the energy consumption of an NB-IoT device, and that the proper configuration of timers in the IoT- and radio interface-protocol stacks, e.g., the CoAP recovery and DRX timers, is key for the NB-IoT battery to last 10 years, which is the expected NB-IoT battery lifetime.

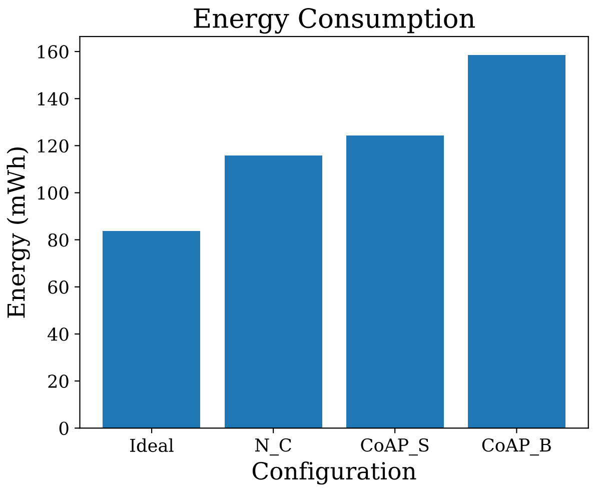


Figure A4_F5. Energy consumption in the four studied NB-IoT scenarios.

## References

[A4_R1] A. Virdis and M. Kirsche (Eds.), “Recent Advances in Network Simulation - The OMNeT++ Environment and its Ecosystem”, Springer Nature Switzerland AG. 2019.

[A4_R2] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP)”, Internet Request for Comments. RFC 7252. June 2014.

# ANNEX 5 – DATA PRIVACY AND ANONYMIZATION

As mentioned above, the usage of the M&A framework can be easily extended to 5GENESIS use cases involving real users. During these experiments, e.g., the Festival of Lights use case in the Berlin platform, personal data are collected and analyzed, and thus privacy aspects have to be considered. A solution to this problem is anonymization.

In a setup with a centralized database, as mostly adopted in 5GENESIS, the database is assumed to be trusted and secure, while analysts are semi-trusted and are not allowed to learn private data. Each database row may represent a single individual, and some fields may hold sensible data to be protected, as they allow the identification of an individual, e.g., a telephone number. Even ordinary information that is not by its nature sensitive can reveal information if it is for example collected over time.

Data anonymization can be achieved for example by suppressing or generalizing information. In the past, various approaches to anonymizing whole datasets were propose, such as k-anonymity, l-diversity and t-closeness, but they all had weaknesses, allowing attacker to break security. The problems approach often lie in the homogeneity of the data, which is difficult to obscure without making the data useless for analysis. For example, researchers at the University of Texas at Austin were able to identify users in an anonymized dataset published by Netflix containing movie ranking of their costumers [A5_R1], by using another public movie database as source for background knowledge.

Since it is not possible to efficiently anonymize data itself, research focused on how to answer database queries without revealing information. The current state of the art for database anonymization is differential privacy [A5_R2], which adds noise to query responses. The idea is to make it impossible for an attacker to find out if a specific individual is in the database or was part of a query response. This technique is used for example by Google [A5_R3] and Apple [A5_R4] for reporting user statistics. For each query, noise is deduced depending on the maximal influence an individual can have on the result. This impact is also called sensitivity. An attacker can then not distinguish original and perturbed data.

Differential privacy has many advantages against other approaches, such as:

- It minimizes the harm for people whose data is stored.
- Unlike other techniques, it does not leak information about atypical individuals.
- It mitigates linkage attacks, e.g., the one on the Netflix database, since noise hinders the identification.
- It is immune to post-processing, since an analyst querying the database without additional knowledge cannot increase privacy loss.

In differential privacy, the noise is created from either Laplace or Gaussian distributions. The second approach is simpler since the impact of Gaussian noise is more easily understandable in the statistical context and can therefore be better corrected. On the other hand, it only allows relaxed versions of differential privacy. Moreover, it should be considered that the fewer the records in a database response, the more the noise that has to be added. However, with more noise, the data richness decreases and it becomes difficult to draw meaningful results.

In the context of 5GENESIS, e.g., for the Festival of Lights in Berlin, the data collected about people connecting to the system are complex. The pattern of users' movement is recorded by

collecting GPS locations, and trajectories are assigned unique IDs. Moreover, for each data point, throughput, packet loss rate, power signal level, provider network and WiFi BSSID are also documented.
Connection length, throughput, packet loss, and density of users are the main 5G KPIs for this use case, in particular from a 5GC perspective. In order to fulfill such KPIs, users may be redirected to dedicated network slices and thus assigned to groups. This means that the KPI analysis might involve only subsets of records belonging to certain groups, which sizes have to be established. Since the analyses are mainly on aggregates and statistical evaluation, the basic conditions for the use of differential privacy are met. For example, determining the size of groups is a counting query, which a standard example for differential privacy. Similarly straightforward is the calculation of KPIs average values. In the context of the Festival of Lights, the use of differential privacy for more fine grained evaluations will be explored. Further evaluations on real data will examine which approach is more suitable for the data to be protected.

## References

[A5_R1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large datasets (how to break anonymity of the Netflix prize dataset)", University of Texas at Austin, 2008.

[A5_R2] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", Foundations and Trends® in Theoretical Computer Science, vol. 9, no. 3–4, 2014, pp. 211–407.

[A5_R3] "Google RAPPOR" [Online], www.github.com/google/rappor, Accessed: Sept. 2019.

[A5_R4] "Apple Privacy" [Online], www.apple.com/lae/privacy/approach-to-privacy, Accessed: Sept. 2019.

# ANNEX 6 – MONROE VN

## Example of outputs from the Throughput container

### "iPerf2"

```
{
 "DataId": "5GENESIS.EXP.IPERF",
 "Protocol": "tcp",
 "DataVersion": 2,
 "Interface": "eth0",
 "Timestamp": 1551446332.883225,
 "Guid": "sha256:a4b55ff5a8893c2e267394fd6481a7908e0a7dd9a48d6a29458104b411712ff9.test-iperf2.7.1",
 "NodeId": "7",
 "Results": {
  "transferID": "3",
  "transferred_bytes": "1012662272",
  "source_port": "52977",
  "timestamp": "20190301131852.883",
  "destination_address": "192.168.100.13",
  "interval": "0.0-10.0",
  "source_address": "172.18.3.2",
  "destination_port": "5001",
  "bits_per_second": "809884422"
 }
}
```

### "iPerf3"

```
{
 "DataId": "5GENESIS.EXP.IPERF",
 "Protocol": "tcp",
 "DataVersion": 3,
 "Interface": "eth0",
 "Timestamp": 1551446053.787472,
 "Guid": "sha256:a4b55ff5a8893c2e267394fd6481a7908e0a7dd9a48d6a29458104b411712ff9.test-iperf3.7.1",
 "NodeId": "7",
 "Results": {
  "start": {
   "connecting_to": {
    "host": "192.168.100.13",
    "port": 5201
   },
   "timestamp": {
    "timesecs": 1551446043,
    "time": "Fri, 01 Mar 2019 13:14:03 GMT"
   },
   "test_start": {
    "protocol": "TCP",
    "num_streams": 1,
    "omit": 0,
    "bytes": 0,
    "blksize": 131072,
    "duration": 10,
      "blocks": 0,
      "reverse": 0
    },
    "system_info": "Linux 90aa79fc96fb 4.9.0-8-amd64 #1 SMP Debian 4.9.110-3+deb9u6 (2018-10-08) x86_64",
    "version": "iperf 3.1.3",
    "connected": [
      {
        "local_host": "172.18.3.2",
        "remote_port": 5201,
        "remote_host": "192.168.100.13",
        "socket": 4,
        "local_port": 36717
      }
    ],
    "cookie": "90aa79fc96fb.1551446043.597668.7500f",
    "tcp_mss_default": 1398
  },
  "intervals": [
    {
      "sum": {
        "end": 1.000105,
        "seconds": 1.000105,
        "bits_per_second": 817688301.083527,
        "bytes": 102221760,
        "start": 0,
        "retransmits": 2,
        "omitted": false
      },
      "streams": [
        {
          "end": 1.000105,
          "socket": 4,
          "rtt": 3960,
          "seconds": 1.000105,
          "bits_per_second": 817688301.083527,
          "bytes": 102221760,
          "start": 0,
          "retransmits": 2,
          "omitted": false,
          "snd_cwnd": 426390
        }
      ]
    },
    {
      "sum": {
        "end": 2.000106,
        "seconds": 1.000001,
        "bits_per_second": 819540175.03198,
        "bytes": 102442644,
        "start": 1.000105,
        "retransmits": 0,
        "omitted": false
      },
      "streams": [
        {
          "end": 2.000106,
          "socket": 4,
          "rtt": 5492,
          "seconds": 1.000001,
          "bits_per_second": 819540175.03198,
          "bytes": 102442644,
          "start": 1.000105,
          "retransmits": 0,
          "omitted": false,
          "snd_cwnd": 575976
        }
      ]
    },
    {
      Interval 3 to 9 deleted for brevity
    },
    {
      "sum": {
        "end": 10.000223,
        "seconds": 1.000061,
        "bits_per_second": 816107503.326209,
        "bytes": 102019640,
        "start": 9.000162,
        "retransmits": 0,
        "omitted": false
      },
      "streams": [
        {
          "end": 10.000223,
          "socket": 4,
          "rtt": 11932,
          "seconds": 1.000061,
          "bits_per_second": 816107503.326209,
          "bytes": 102019640,
          "start": 9.000162,
          "retransmits": 0,
          "omitted": false,
          "snd_cwnd": 1234434
        }
      ]
    }
  ],
  "end": {
    "sum_received": {
      "seconds": 10.000223,
      "start": 0,
      "bytes": 1013134794,
      "end": 10.000223,
      "bits_per_second": 810489767.650944
    },
    "streams": [
      {
        "sender": {
          "end": 10.000223,
          "socket": 4,
          "max_rtt": 11932,
          "seconds": 10.000223,
      "bits_per_second": 812731463.278758,
      "bytes": 1015936976,
      "max_snd_cwnd": 1234434,
      "min_rtt": 3960,
      "start": 0,
      "retransmits": 5,
      "mean_rtt": 8559
     },
     "receiver": {
      "end": 10.000223,
      "socket": 4,
      "seconds": 10.000223,
      "bits_per_second": 810489767.650944,
      "bytes": 1013134794,
      "start": 0
     }
    }
   ],
   "cpu_utilization_percent": {
    "remote_user": 9.5e-05,
    "remote_system": 0.000918,
    "remote_total": 0.001014,
    "host_system": 1.269514,
    "host_total": 1.376223,
    "host_user": 0.119017
   },
   "sum_sent": {
    "end": 10.000223,
    "seconds": 10.000223,
    "bits_per_second": 812731463.278758,
    "bytes": 1015936976,
    "start": 0,
    "retransmits": 5
   }
  }
 }
}
```

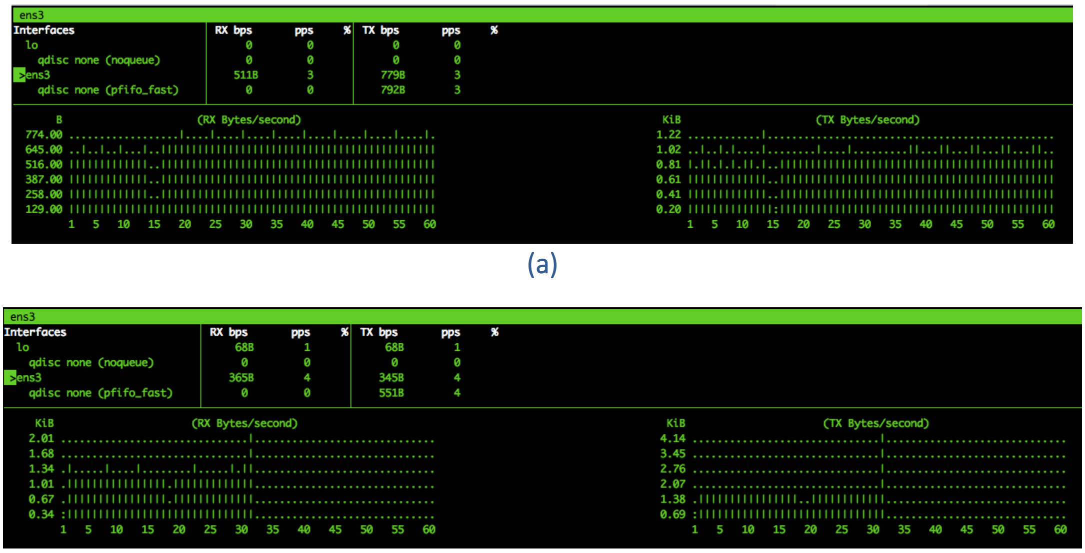
# ANNEX 7 – IOT INTEROPERABILITY

The Surrey platform has IoT interoperability capabilities using the INFOLYSiS vGW, which works by using SDN and NFV. IoT data using different transport protocols like HTTP, CoAP, and Message Queuing Telemetry Transport (MQTT) reach the system and they are being mapped in a plain UDP, and they are gathered and stored in INFOLYSIS IoT vGW. Additionally, there is the possibility of sending the UDP stream of IoT data to different destinations, using different IP and port combinations, for further processing and in order to perform data analytics.
The INFOLYSiS system consists of the following components:

- IoT Proxy: Entry point of the virtual system that all IoT data from external sources target. The data that arrive in this VNF are transported to the relevant mapper VNF in order to be made interoperable.
- HTTP / CoAP / MQTT Map VNFs: These VNFs are responsible for converting HTTP, CoAP, and MQTT IoT data to the interoperable format of the system, i.e., UDP. After being converted, all data are sent to the IoT vGW.
- IoT vGW: The final destination of all data in the INFOLYSiS system. Interoperable UDP data are stored in a MySQL database and are being available through a graphical dashboard. This VNF is also responsible for forwarding multiple UDP streams of IoT data to different recipients to use them.

## IoT monitoring and interoperability of HTTP and MQTT sensors

An initial deployment of the IoT interoperable GW has been successfully deployed in Surrey platform using two different IoT protocols, namely HTTP and MQTT. For validation purposes, data originating from traffic generators were used. The data passed through the appropriate mapping VNFs  and were stored in the MySQL database of the INFOLYSiS IoT vGW.
Figures A7_F1 (a)(b) depict the network traffic in HTTP and MQTT Map VNFs, respectively. All the packets go in and out of the single interface of the VNF (ens3). The bitrate and number of packets of received and transmitted packets are also shown in the Figure.



```
ens3
Interfaces                    RX bps       pps     %  TX bps       pps     %
  lo                               0         0             0         0
    qdisc none (noqueue)           0         0             0         0
>ens3                           511B         3          779B         3
    qdisc none (pfifo_fast)        0         0          792B         3
```

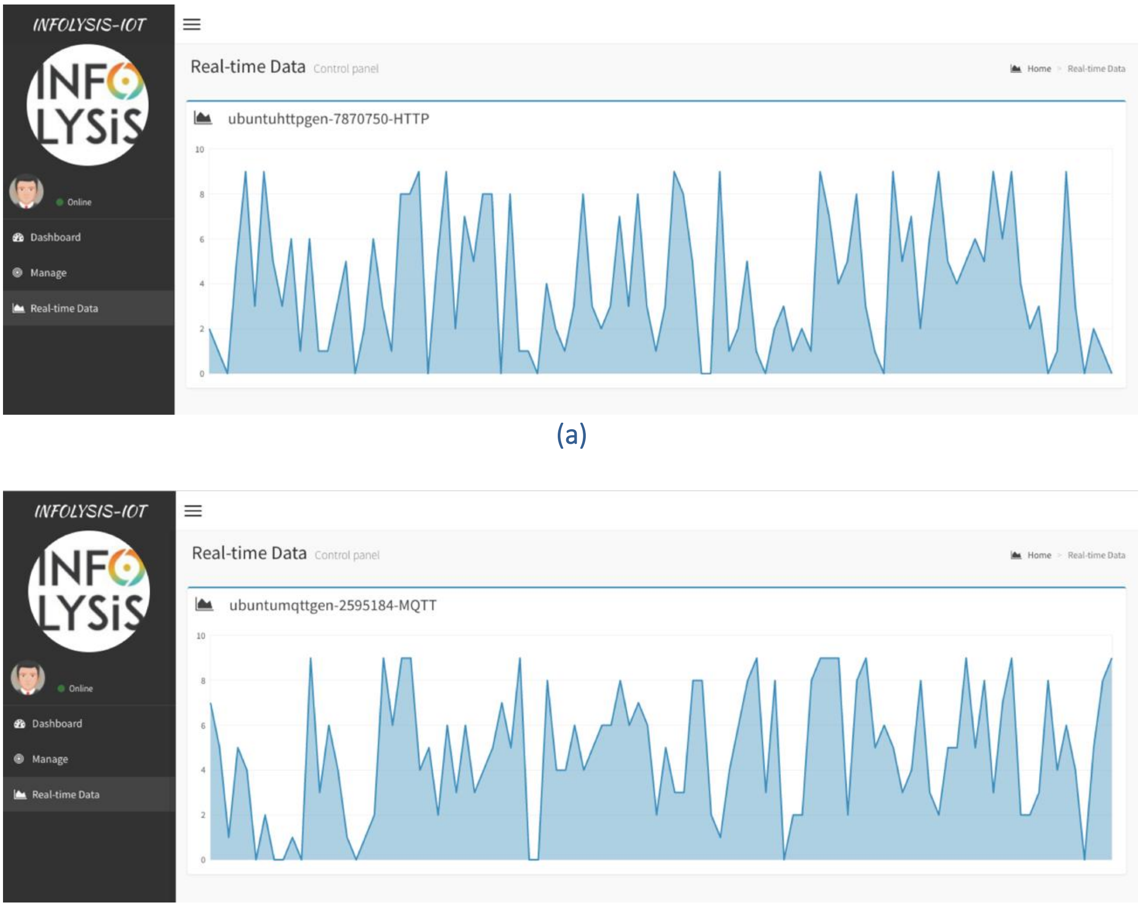
(a)

```
ens3
Interfaces                    RX bps       pps     %  TX bps       pps     %
  lo                             68B         1           68B         1
    qdisc none (noqueue)           0         0             0         0
>ens3                           365B         4          345B         4
    qdisc none (pfifo_fast)        0         0          551B         4
```

(b)

Figure A7_F1. Network traffic in HTTP (a) and MQTT Map VNFs, showing HTTP and MQTT inputs and UDP output in Bytes

Moreover, Figure A7_F2 shows possible views within the INFOLYSiS vGW dashboard. A user can login to the dashboard and observe IoT data-streams in real-time,  by choosing an IoT sensor of interest which is sending data to the vGW.



(a)

(b)

Figure A7_F2. INFOLYSiS vGW Dashboard: real-time data from two sensors HTTP (a) and MQTT (b)