H2020-ICT-688712

Project: H2020-ICT-688712

Project Name:
5G Applications and Devices Benchmarking (TRIANGLE)

Deliverable D4.3

Report on the QoS management at the application level

Date of delivery:	11/12/2018	Version:	1.0
Start date of Project:	01/01/2016	Duration:	36 months

Deliverable D4.3
Report on the QoS management at the application level

Project Number:	ICT-688712
Project Name:	5G Applications and Devices Benchmarking
Project Acronym	TRIANGLE

Document Number:	ICT-688712-TRIANGLE/D4.3
Document Title:	Report on the QoS management at the application level
Lead beneficiary:	Keysight Technologies Denmark
Editor(s):	Donal Morris (RedZinc)
Authors:	Ricardo Ricardo Figueiredo, Terry O Callaghan, Pilar Rodriguez (RedZinc), German Madueno, Andrea Cattoni (KEYD)
Dissemination Level:	PU
Contractual Date of Delivery:	31/10/2018
Work Package Leader:	German Madueno (KEYD)
Status:	Final
Version:	1.0
File Name:	TRIANGLE_Deliverable_D4.3_FINAL

Abstract

TRIANGLE is a project about 5G Applications and Devices Benchmarking. TRIANGLE enables testing, benchmarking and certification of apps and devices in a pre-5G environment. This deliverable is about supporting quality of service in TRIANGLE. We provide an automated method for testing QoS based services. We discuss the state of the art in the market place for 5G services, slicing and Gold, Silver Bronze differential services. We implement a QoS testing environment with a quality of experience portal, a virtual path slice orchestrator and integration to a tester. From an SDN point of view we deploy an SDN testbed and deploy the virtual machine monitoring. In the deliverable we evaluate the role of QoS in 5G, its current status and the expected evolution

Keywords

QoS, Slicing

Executive summary

The focus in this TRIANGLE deliverable is on bandwidth slices and how it impacts quality of experience (QoE).

In this deliverable we provide a description of the tools for application developers and researchers to mitigate the impact of network conditions by requesting a Quality of Service (QoS) to be configured on the network that suits their application and improves their overall perceived QoE.

Section 2 is about supporting quality of service in TRIANGLE. We discuss the state of the art in QoS techniques and look at the market place for 5G services. We study slicing and differential service based on standardised usage scenarios. A categorization A Gold, Silver Bronze categorization of the quality of experience service model is presented.

Section 3 is about the application of QoS selection methods. We discuss the orchestrator used, the quality of experience portal and integration to the TAP which controls the UXM tester. In addition, we describe the SDN deployment.

In section 4 we describe the virtual machine monitoring implemented.

Finally, in section 5, we evaluate the role of QoS in 5G, its current status and the expected evolution.

Contents

List of Figures

List of Tables

List of Abbreviations

AP	Access Point
API	Application Programming Interface
APNet	Antennas, Propagation and Radio Networking
ASN	Autonomous System Number (used in Border Gateway Protocol for routing to different network administrations)
BER	Bit Error Rate
BGP	Border Gateway Protocol
BLER	Block Error Rate
BS	Base Station
CAPEX	CApital EXpenditure
CDMA	Code Division Multiple Access
CFO	Carrier Frequency Offset
CO	Confidential
CP	Cyclic Prefix
CR	Cognitive Radio
CRS	Cognitive Radio Systems
CSI	Channel State Information
CSMA	Carrier Sense Multiple Access
C2X	Car-to-Anything
D	Deliverables
DL	Downlink
D2D	Device-to-Device
DMRS	Demodulation reference signal
DRX	Discontinuous Reception
DTX	Discontinuous Transmission
EIRP	Effective Isotropic Radiated Power
EIT	European Institute for Innovation and Technology
E2E	End-to-End
eMBB	Enhanced Mobile Broadband
EVM	Error Vector Magnitude
EPC	Evolved Packet Core
FDD	Frequency Division Duplex
FD-MIMO	Full-Dimension MIMO
FEC	Forward Error Correction
FR	Frequency Response
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HARQ	Hybrid Automatic Repeat Request
IaaS	Infrastructure as a Service
ICI	Inter-Carrier Interference
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineers
IMT	International Mobile Communications
IP	Intellectual Property
IPR	Intellectual Property Rights
IR	Internal report
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union-Radio
KPI	Key Performance Indicator
LAN	Local Area Network
LOS	Line of Sight
LTE	Long Term Evolution
LTE-A	Long Term Evolution-Advanced
L2S	Link to System
M	Milestones
Mbps	megabits per second
Mo	Month
MA	Multiple Access
MAC	Medium-access Control
MGT	Management
MIMO	Multiple-Input Multiple-Output
MMC	Massive Machine Communication
mMTC	Massive Machine Type Communications
M2M	Machine to Machine
MSE	Mean Squared Error
NaaS	Network as a Service
NGMN	Next Generation Mobile Networks
NLOS	Nonline of Sight
N5	Interface in the PCF
OFDM	Orthogonal Frequency Division Multiplexing
OPEX	Operational Expenditure
PA	Power Amplifier
PaaS	Platform as a Service
PAPR	Peak-to-Average-Power-Ratio
PC	Project Coordinator
PHY	Physical Layer
PU	Public
PCRF	Policy and charging rules function
PCF	Policy control function
QAM	Quadrature Amplitude Modulation
QAP	Quality Assurance Plan
QMR	Quarterly Management reports
QoE	Quality of Experience
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network
RAT	Radio Access Technology
RF	Radio Frequency
R&D	Research and Development
RRM	Radio Resource Management

RTD	Research and Technological Development
RTT	Round Trip Time
RX	Interface in PCRF/PCF
SDR	Software Defined Radio
SINR	Signal to Interference and Noise Ratio
SRS	Sounding Reference Signal
T	Task
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TRX	Transmitter
TTI	Transmission Time Interval
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
URLLC	Ultra-Reliable low-latency communications
USRP	Universal Software Radio Peripheral
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-anything
VRF	Virtual Routing and Forwarding
WCDMA	Wide Code Division Multiple Access
WLAN	Wireless Local Area Network
WP	Work Package
WPAN	Wireless Personal Area Networks
XaaS	Anything as a Service

1 Introduction

One of the main differentiating factors between applications in the smartphone market is the user Quality of Experience (QoE). QoE depends on several factors such as the quality of service (QoS). The QoS is typically something bound to the network and the way operators configure their network. A business opportunity exists however for operators to differentiate the QoS based on the end user subscription (e.g., standard versus premium). Typically, QoE can only be modified at the application level, e.g., changing video resolution depending on the connection status. In other words, the application developer has always been limited by the network conditions of the end user, but this is no longer the case.

In TRIANGLE we provide the tools for application developers and researchers to mitigate the impact of network conditions by requesting a QoS to be configured on the network that suits their application and improves their overall perceived QoE.

To support this QoS configuration in the network, a set of pre-defined QoS profiles (Gold, Silver and Bronze) were defined. This simplifies and optimises the delivery process, as well as the deployment of a Software Defined Network (SDN) infrastructure and software stack to evaluate the impact while using this type of underlying network.

Several ways to request and configure QoS have been integrated to the testbed, making it easier to interact with this technology regardless of the knowledge level of how it works. This makes QoS real-time configuration available.

To ensure quick action on addressing any issue on QoS, the infrastructure that supports QoS profiling is constantly being monitored. Given the number of moving pieces this is essential to make the experimenters experience smooth. For that end, we provide an automated method for testing QoS based services using tools developed in TRIANGLE.

Before we provide more details on the QoS framework, we briefly describe the current state of the art in QoS techniques and the evolution of the market place for 5G services.

1.1 Brief history of QoS

Quality of Service has been well researched for packet networks. The early IntServ model from the mid 1990s based on deterministic resource reservation using signalling [1] did not scale well with the limitations of processing in routers on the market at that time. The idea of flow state in the internet core was not scalable for high volume of sessions. DiffServ was invented as a stateless mechanism with assured forwarding and expedited forwarding [2] Most internet core routers supported this but DiffServ did not support the concept of end to end flows as it only respected per hop behaviours. MPLS [3] traffic engineering used the earlier resource reservation protocol, coupled with multi-protocol label switching to reserve traffic engineering paths for aggregate flows in core networks. Scheduling would differentiate resources on a path using EXP value in the MPLS headers for different flows. Typically, several different flow profiles are used and widely deployed in most core networks by 2000.

The concept of net neutrality, the explosion of access network capacity and the lack of a business model impeded the deployment of QoS from a customer view point for many years.

The arrival of OpenFlow, software defined networks and virtualisation ushered in the concept of slicing. This would create a whole topology region of virtualised compute, connectivity and storage resources for a dedicated infrastructure or industry vertical or customer. The idea is that each customer or user group would have their own ‘lane’ and that one lane would not interfere with another lane.

Slicing is the first 5G capability mentioned in the 5G Manifesto [4] endorsed by 17 major telecoms industries in Europe. “Demonstration of the concept of 5G network virtualisation (slicing) to accommodate specific needs or business models with enhanced levels of service assurance and guarantees”

2 Supporting Quality of Service in TRIANGLE

2.1 Gap Analysis

We have conducted a gap analysis for QoS in 5G and have the following analysis and it is displayed in Table 1.

Table 1: QoS Gap Analysis

1	**Net Neutrality**	Net neutrality [5] which originated from the IT industry as a movement to inhibit telecoms coms companies from differentiating traffic. This has been adopted by legislators and has inhibited market development of QoS support. It is a political and market regulatory issue.
2	**Business Model**	There is limited understanding how to implement a business model for QoS. In order to implement QoS it is necessary to have some management of capacity and its allocation. Competition has usually implied an all you can eat model.
3	**PCRF**	Operators have defined PCRF [6] (policy rules and charging function) model into telecoms standards enabling differentiation of services using QCI (Quality Class Indicator). But PCRF is typically not available to applications in 4G networks
4	**Communications**	Slicing, with different lanes for different users is a common concept in 5G research and development, but it would appear that it is not well communicated and not known outside the 5G development community.
5	**Over the Top**	OTT – Over the Top, providers (e.g. NetFlix) implement retail services on best effort, but because the telco has no QoS API at the point of traffic insertion there has been very little in development of application which could exploit this.

There are a number of factors which have inhibited QoS take-up so far and which need to be addressed by the 5G community including the exposure of a QoS API and an alignment of slicing terminology.

2.2 QoS benefits

Benefits of QoS for user are:

- Improve user experience
- Cut through congestion for specific applications (e.g. medical)

- Support mission critical applications
- Avoid sense of slow network (if customer prepared to pay extra)

Benefits of QoS for 5G Operator are:

- Matching customer applications requirements to network capacity
- Up Selling so that value added services can be offered
- Incremental revenue and better return on investment, especially related to over the top traffic sources
- Linking investment in network infrastructure to payment
- Implement a sender pays model aligns infrastructure CAPEX with traffic and with sources of revenue.

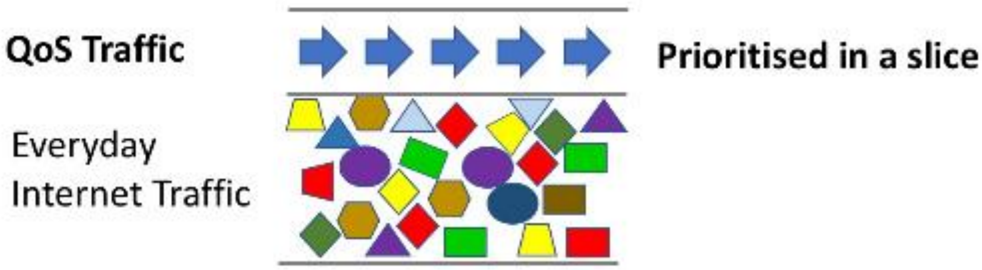


Figure 1: Prioritised QoS Traffic in a slice

Implementation of QoS with 5G specifications will be possible by the use of the N5/Npcf interface as an Application Function (AF) or via an external interaction with the Network Exposure Function (NEF). The QoS-related interface is termed Rx in 4G and N5/Npcf in 5G (depending on use of traditional or SBA architecture). Solutions such as VELOX support this method. VELOX operates as an AF role and uses a Resource Driver for the Rx/N5 interface to request resources in the data plane. The use of VELOX assumes an agreement with the network provider for having direct access to the core functionalities. The reference for Rx/N5 in the 5G architecture can be seen in Release 15 document ETSI TS 129 513 V15.0.0 on policy rules and charging for 5G [7].

2.3 Business Models for 5G Services

The Next Generation Mobile Alliance (NGMN) 5G white paper [8] identifies a wide range of services including:

- Broadband Access Everywhere with 50+ Mbps Everywhere supported by Ultra Low-Cost Networks;
- Higher User Mobility including High Speed Train, Remote Computing, Moving Hot Spots, dynamic and real-time provision of capacity, 3D (three dimensional) Connectivity – Aircrafts.

- Massive Internet of Things (IoT) such as Smart Wearables (Clothes), Sensor Networks and Mobile Video Surveillance; Extreme Real-Time Communications such as in the Tactile Internet;
- Lifeline Communication supporting Natural Disaster; Ultra-reliable Communications for Automated Traffic Control and Driving, Collaborative Robots eHealth and Extreme Life Critical, Remote Object Manipulation: such as in Remote Surgery 3D Connectivity: Drones Public Safety;
- Broadcast-like services.

In order to deliver on these services slicing is considered as a key component. Today the infrastructure of the telecoms network is built in silos. In 5G slicing will move the model from silos of network processes to slices dedicated towards particulate sectors and markets. These markets are called verticals by the 5G community. Figure 2 from Deutsche Telekom shows the model of moving from silos to slices.

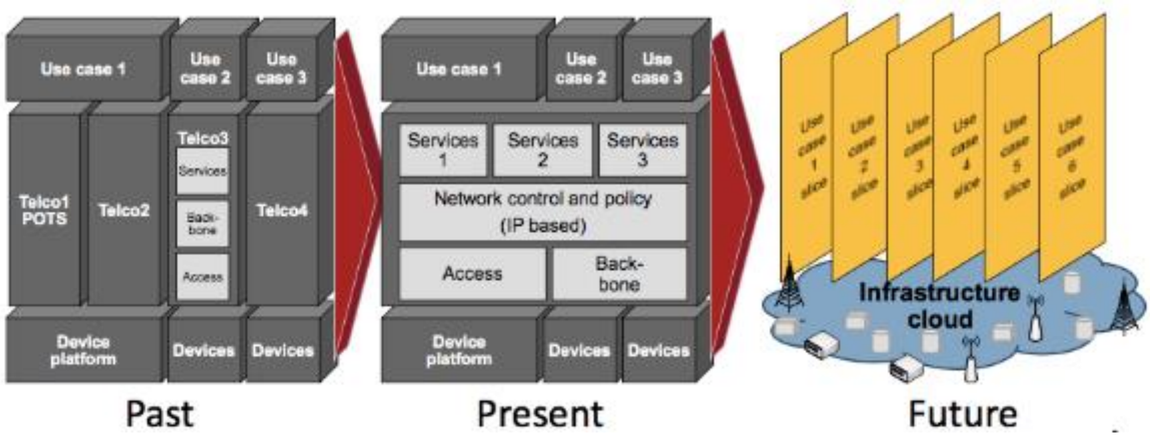


Figure 2: Moving from silos to slices (Source DT) [9]

5G anticipates a range of slices supporting various market place elements. Projects such as SONATA[10], 5GEX [11] and SLICENET [12] are working on the various slicing aspects. At its simplest level ‘the slice is the service’. This can be further be decomposed as a set of resources allocated to a specific application mission and industry sector as shown in Figure 3 provided by TRIANGLE.

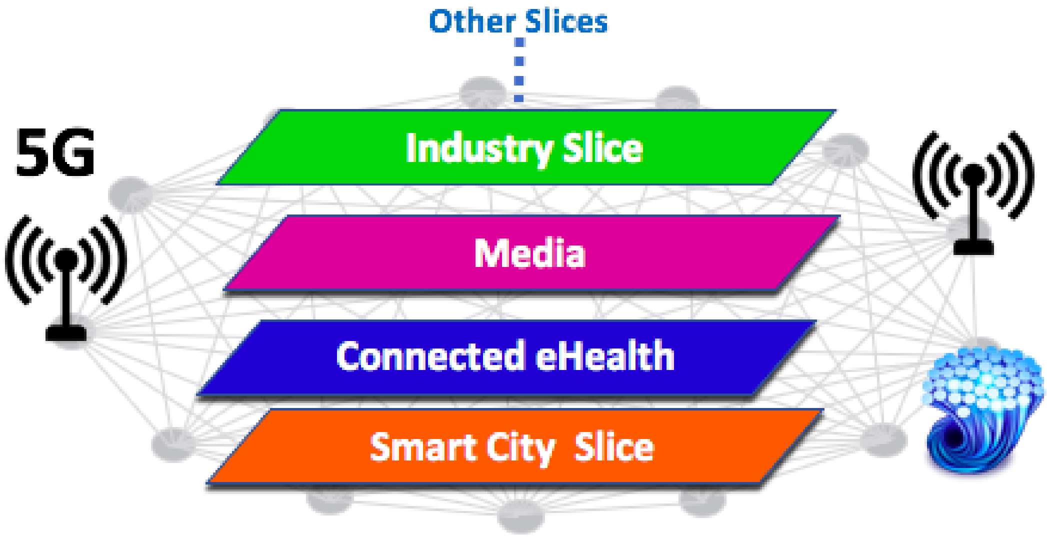


Figure 3: 5G slices focused on particular vertical industry sectors.

2.4 Slicing for M2083 Use Cases

In TRIANGLE we consider and develop a model for differentiated services based on QoS in terms of the M2083 use cases.

In 5G different usage scenarios are anticipated for fast broadband, low latency and also machine communications. ITU-T M2083 [13] shows the usage scenarios for International Mobile Telecommunications in 2020 and beyond. The three usage scenarios expected are: Enhanced mobile broadband (eMBB); Ultra-reliable and low latency communications (URLLC) and Massive machine type communications (mMTC).

Based on the M2083 Model we can anticipate three foundation slices related to URLLC, mMTC and eMBB. Each will have resource management associated with it. Figure 4, developed in TRIANGLE, shows the resource management model based on different regions in the end to end path from user equipment to EPC core network. In the air interface, the resource management can allocate the resource blocks via a scheduler. Then we consider three distinct SDN domains. A fronthaul and C-RAN SDN region consisting of links and virtual network functions. A back haul and SDN region, again consisting of links and virtual network functions and a core with a virtualised EPC on virtualised networks functions in an SDN environment. In order to support slices and provide capacity dimensions links related to each slice are allocated capacity and this can be dynamically adjusted in order the allow the network to 'breath' based on arriving customer traffic demand and growth.

It is anticipated that personalised flows in the future internet give an enhanced tier based on SDN and creates a win-win-win scenario – a win for the user, the application provider and the network provider.

Today, Over The Top (OTT) applications (e,g, NetFlix, Google Drive, Skype, Facetime) effectively use the sockets in IP and TCP/UDP protocols as their application programming interface in order to obtain global reachability. With the current architecture of the internet Application Providers can procure interconnection bandwidth to deliver their apps, but they cannot procure end user bandwidth (e.g. fiber to the home or 4G today radio bandwidth) and so they deliver applications over the top of the internet and singularly rely on layer 3/4 sockets to obtain global reachability. With slicing every application receives its own lane.

The essential idea is to segment the network capacity into lanes. This can be implemented by using an aggregated queue in a gateway network element dimensioned for X% of the capacity for traffic painted as BE - best effort - using diffserv code points. Other A% B%, C% in the SDN domain can be painted EF or AF – expedited forwarding or assured forwarding - using diffserv code points. Painting traffic is implemented by setting the diffserv code point or MPLS 'exp' bit to signify a particular traffic category. Each domain can be sliced using traffic-based connection admission control techniques to give a guarantee to the traffic. This is especially important for inelastic traffic such as live video which is sensitive to TCP back-off common in well loaded best effort networks. Wholesale and business to business customers with multiple applications in the area of Internet of Things (IoT) eHealth, Consumer, Cloud, Media Social can use the slicing mechanism to obtain a slice of bandwidth across multiple autonomous systems and to the consumer connected via optic fiber access or 4G/5G radio access network.

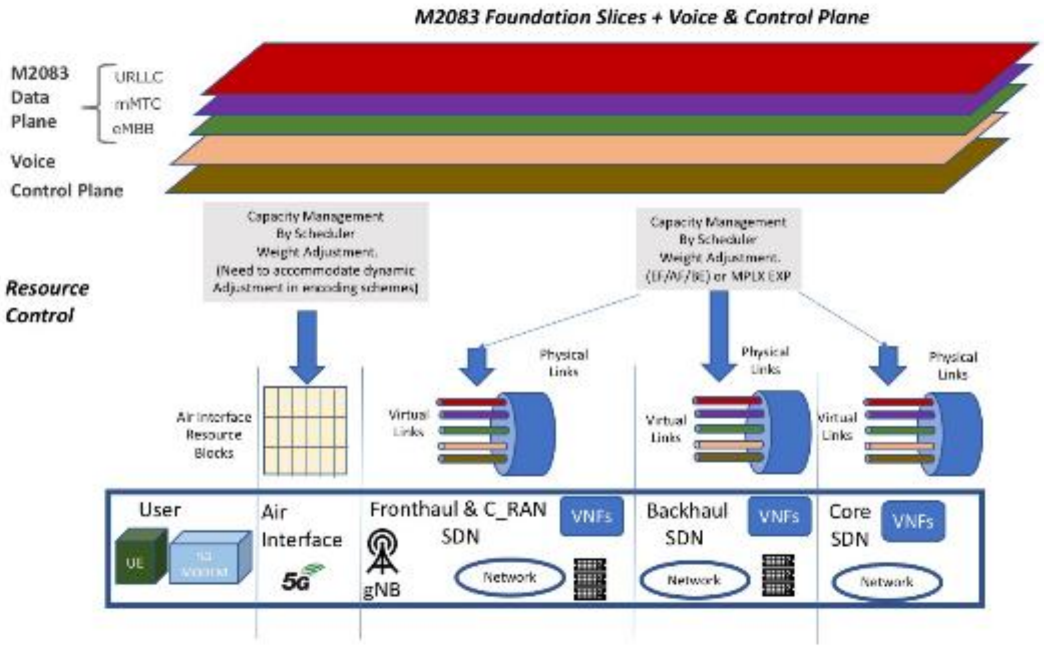


Figure 4: Proposed Resource Management Model

These M2083 foundation slices can be used as building blocks to access higher level slices sold in the market place to industry verticals. Differentiation can be via address management. See the model in Figure 5 as developed in TRIANGLE.

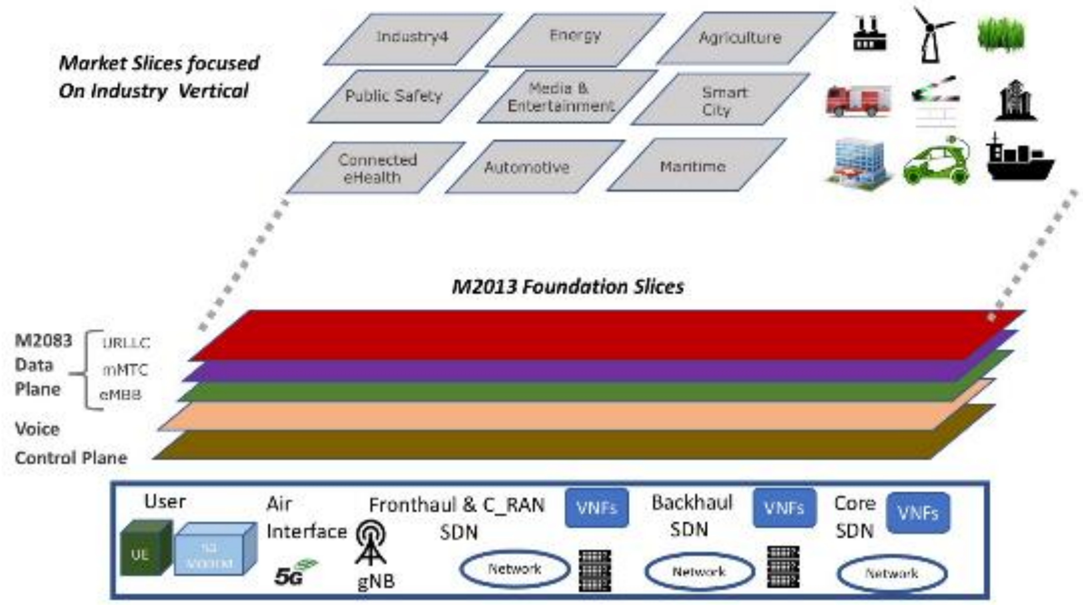


Figure 5: M2083 Foundation Slices mapping to Slice in the market vertical industries

2.5 Quality of Experience Service Model

In order to make differentiated services available to experimenters, in TRIANGLE we defined service model following discussion with one of the experimenters. This is a model which we are proposing. Gold/Silver/Bronzes are not widely used but have the potential to be used in 5G associated with slicing and QoS support.

A Gold/Silver/Bronze service model with target bandwidth was defined. The idea is based around the Gold/Silver/Bronze model of service definition which are mapped to target bandwidth and which, in turn are mapped to resource blocks in the radio side. The scheduler parameters (resource blocks and scheduling availability) are desired maximums, that the scheduler cannot guarantee, but tries to achieve. An example service mapping is shown in Table 2. The details of these calculations are given in Appendix 2. This is based on a cell capacity of 125 Mps from 4G environment. As 5G arrives 400 Mbps is a more realistic target for the calculation using cell capacity. We provide a Platinum model in Table 3 based on a higher target bandwidth for the cell. Other service mappings are possible.

Table 2: Service to Bandwidth Resource Mapping for Gold, Silver Bronze Mapping

Service	Bandwidth Resource Description	Cell Capacity	Downlink LTE Scheduling %
Loaded Network Condition	Unspecified Theoretical Downlink 0.47Mbps	125 Mbps	Frequency domain (DL) 5.0% Time domain (DL) 7.5%
Bronze-2-x	2Mbps downlink Unspecified uplink	125 Mbps	Frequency domain (DL) 21.35% Time domain (DL) 7.5%
Silver-4-x	4Mbps downlink Unspecified uplink	125 Mbps	Frequency domain (DL) 21.35% Time domain (DL) 15.0%
Gold-6-x	6Mbps downlink Unspecified uplink	125 Mbps	Frequency domain (DL) 42.65% Time domain (DL) 15.0%
SuperGold-10-x	10Mbps downlink Unspecified uplink bandwidth	125 Mbps	Frequency domain (DL) 42.65% Time domain (DL) 18.75%

Table 3: Service to Bandwidth Resource Mapping for Platinum Mapping

Service	Bandwidth Resource Description	Cell Capacity	Downlink LTE Scheduling %
Platinum-10-x	10Mbps downlink Unspecified uplink	400Mbps	Frequency domain (DL) 25% Time domain (DL) 10%
Platinum-20-x	20Mbps downlink Unspecified uplink	400Mbps	Frequency domain (DL) 50% Time domain (DL) 10%
Platinum-40-x	40Mbps downlink Unspecified uplink	400Mbps	Frequency domain (DL) 50% Time domain (DL) 20%
Platinum-60-x	60Mbps downlink Unspecified uplink bandwidth	400Mbps	Frequency domain (DL) 50% Time domain (DL) 30%

3 Application QoS selection Methods

3.1 Overview

This section enables an experimenter to experiment with QoS. The user can experiment with Gold/Silver/Bronze services and make adjustments in the target bandwidth to achieve certain performance levels in terms of QoE.

In auto mode an experimental user can select Gold/Silver/Bronze services in the QoE portal. Their device can be connected to the testbed and the user can build an experiment and conduct a measurement campaign where the focus is related to access bandwidth.

In research mode the experimental user can deploy their application backend inside a slice and observe performance of bandwidth and virtual machines. In research mode the user can also build their own Gold/Silver/Bronze services which can be implemented in the EPC.

To enable these experiments, different components were developed around the Virtual Path Slice Orchestrator (VELOX) and the QoE Portal. These are described in detail below. More specially, section 3.2 revisits the basics about VELOX. Section 3.3 describes the VELOX northbound API. Section 3.4 describes the QoE Portal. Finally, Section 3.5 describes the SDN deployment to implement a slice.

3.2 VELOX Virtual Path Slice Orchestrator

VELOX [14] (also referred to as VPS Engine) provides a technology agnostic way for applications to request a quantum of bandwidth and a type of service differentiation via its Northbound API (Section 3.3). This allows application to request an end-to-end virtual path slices across multiple networks administrative domains in a transparent manner. The local Administrative Domain where the client application is a customer and any necessary remote Administrative Domains need to have a VELOX present to install the requested bandwidth. This bandwidth request installation is achieved with an architecture based on Resource Technology specific drivers which provide a VELOX specific Northbound interface and a technology specific southbound interface towards the network control plane, thus allowing VELOX to have an internal technology agnostic core to provide abstracted end-to-end topologies to its customers. The Overall architecture can be seen in Figure 6.

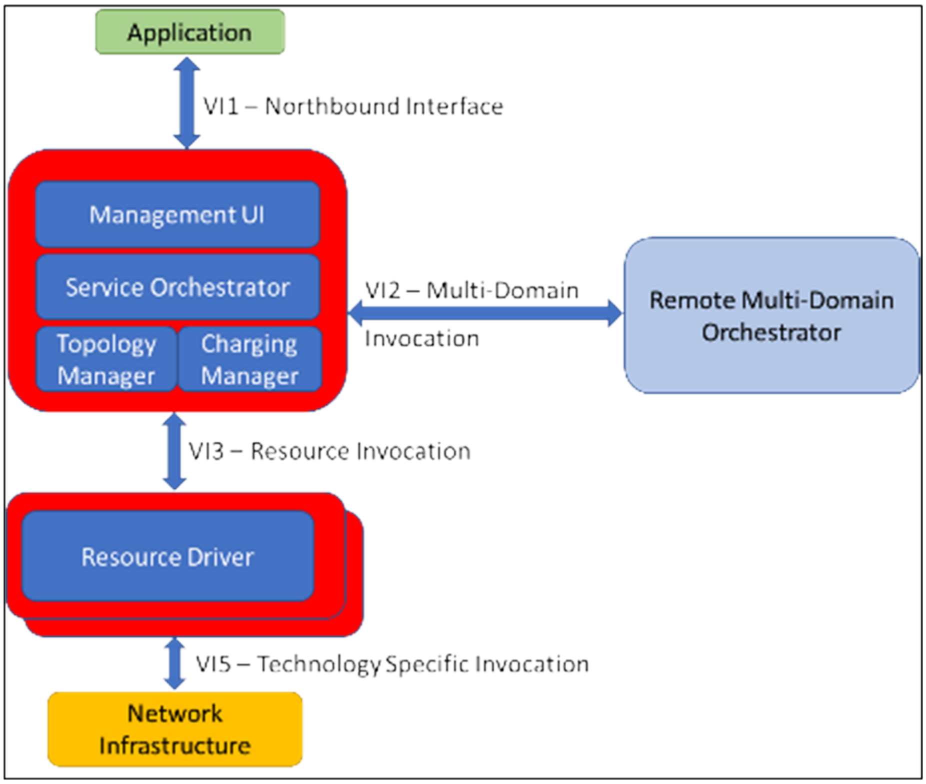


Figure 6: VELOX Architecture

3.3 VELOX Northbound API

VELOX services can be requested via its Northbound API which can be access via simple TCP Sockets and receives requests based on JSON formatted messages. It provides functionality for service activation/deactivation as well as listing. It is based on API Keys, keeping service availability managed per user. A sample usage flow can be seen in Figure 7. This method is used as a quick way to prototype a synchronous interface.

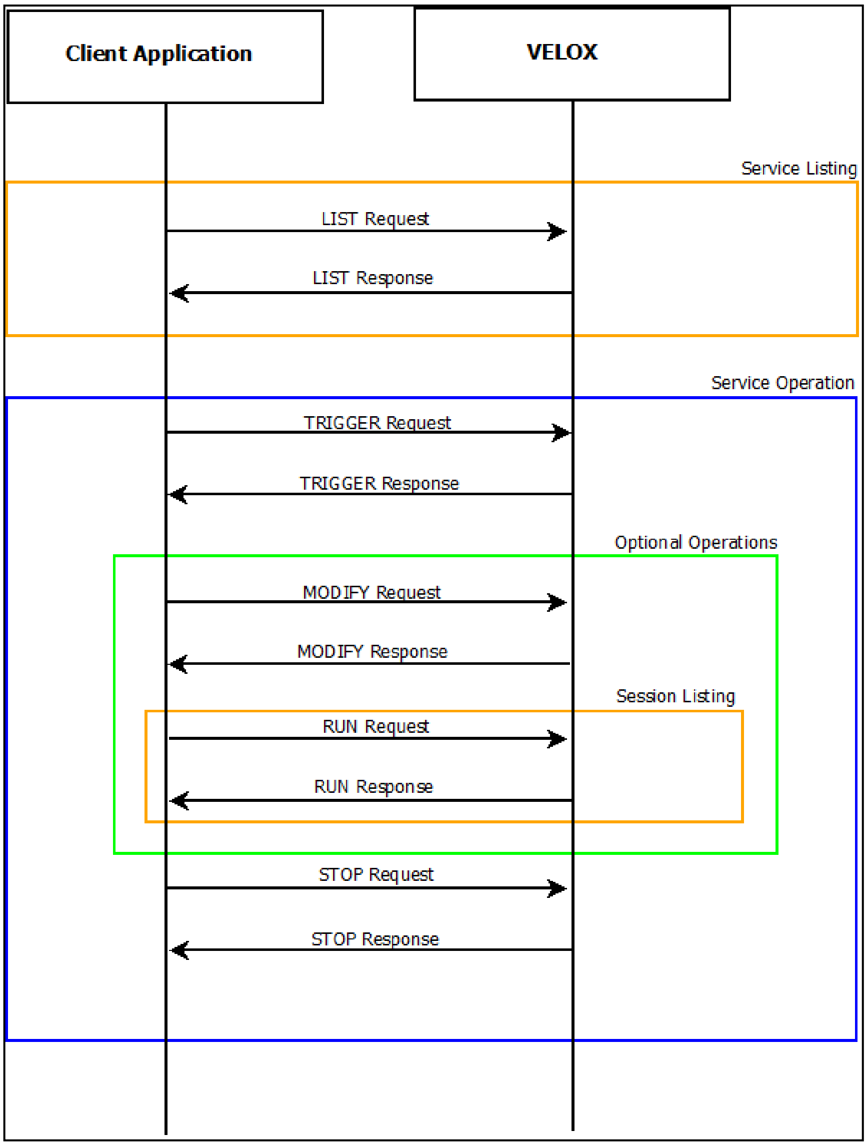


Figure 7: VELOX usage flow

An experimenter application can implement the API to directly request and manage its QoS services in VELOX. The API documentation can be found in Appendix 3.

3.4 Quality of Experience Portal

To facilitate experimentation with VELOX without requiring the implementation of the VELOX Northbound API, a Quality of Experience Portal for experimenters was created. This acts as an implementation of the API but providing it as a Graphical Interface for easy experimenting for researchers or experimenters before committing to it in the TRIANGLE framework tests. The outline of the Quality of Experience Portal in the TRIANGLE testbed is show Figure 8.

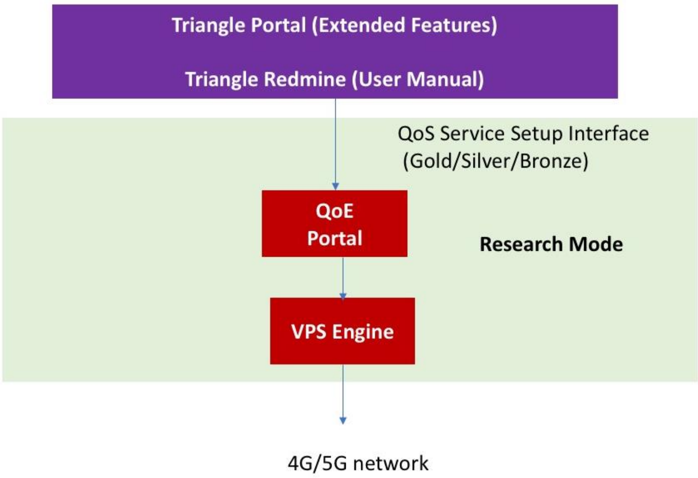


Figure 8: QoE portal

The QoE Portal for VELOX experimentation is composed of two main parts, an experimenter interface and a management interface. Screen shots to illustrate the process of the portal are given in appendix 1.

For experimenters the features included are:

- Allow for user registration
  - Mandatory TRIANGLE ID
  - Allow for typical password recovery
  - Bind a unique VELOX API Key to each user on creation
- Allow activation and deactivation of VELOX services
  - Uses the VELOX API
  - Present available services
  - Present running services
  - Show human readable service results
  - Facilitate source/destination inputs (label IPs, etc)
- Allow History consultation
  - Show services ran
  - Show termination status (ok, not ok, by system clean up)
  - Show time details

For TRIANGLE testbed managers to be able to control and monitor VELOX service users and services the following features are included:

- Display list of users
  - Login times
  - Activity log of service activations/deactivations
  - Activate/deactivate users and respective VELOX API Keys
- Manage VELOX
  - Set VELOX access details such as ip/port
  - Clean-up running services

3.4.1 Quality of Experience Portal Research Mode & Auto Mode

The Portal approach includes a number of ingredients and two modes of operation. The Quality of Experience portal can operate in two modes:

- Research Mode (Manual Mode)
- Auto Mode

In the Research Mode the experimenter can manually select a service on the QoE portal with bandwidth characteristics. In Manual Mode four test profiles are supported corresponding to the Service Model below, Platinum, Gold, Silver, Bronze.

In Research Mode the process steps are:

- User access the QoE portal
- User selects service (Gold, Silver, Bronze)
- The VPS engine is available to invoke a service using QCI 2 in the radio layer.

The research mode is not automated so a preferred “auto mode” was developed as illustrated in Figure 9. This mode simplifies the testing process for an experimenter, which can easily define different service levels quality in terms of bandwidth, for gold silver or bronze. This is invoked into the VPS engine which would normally request this in a standardised approach using the Rx application interface into the EPC. The approach used in TRIANGLE is slight different due to the UXM wireless tester capabilities. In auto mode, the service level is enforced by controlling the UXM via TAP, by uploading and running a quality plan for each service model. Furthermore, in the auto Mode the experimenter can make a new test plan in TAP for running an automatic experiment. In the auto mode the service is mapped into resource blocks in the UXM. The quality plan (gold/silver/bronze) data sent to the UXM is shown in Appendix 2.

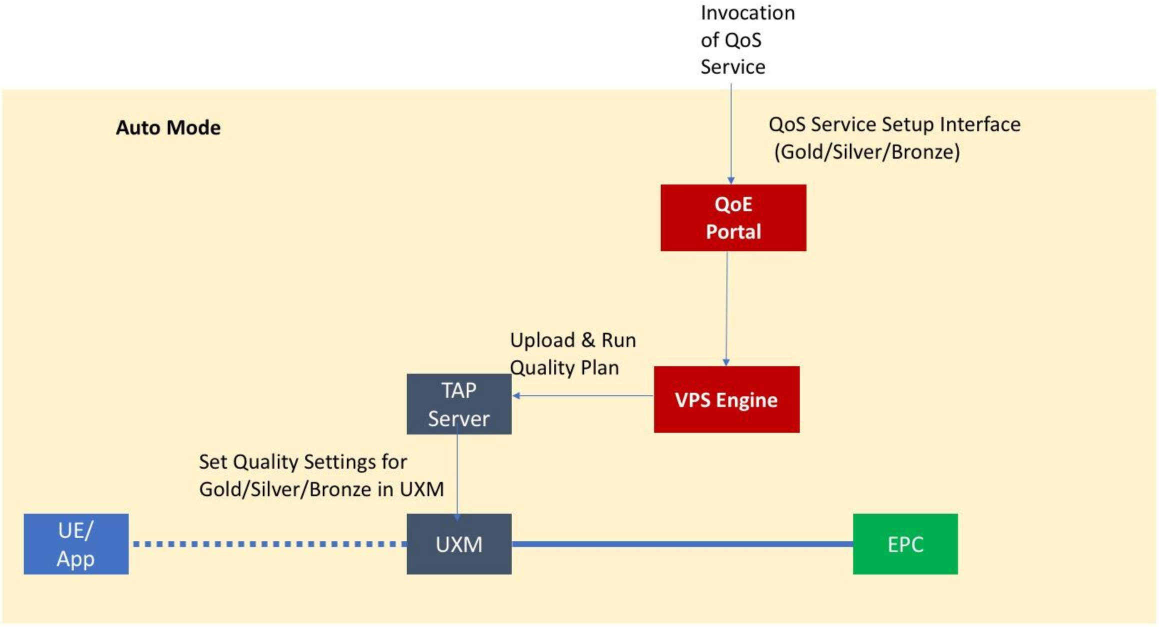


Figure 9: QoE portal and TRIANGLE Testbed Integration

Mean Opinion Score is a standard method in telecommunications of quantifying subjective information. It was originally developed for voice services to compare the subjective quality of experience of voice calls. In TRIANGLE we have expanded the original scope of MOS in order to provide an automated evaluation of different services and user experience areas as compared to the traditional MOS defined by ITU. This is further elaborated in the TRIANGLE Deliverable D2.7 [15].

The QoE Portal enables the experimenter to specify the following details and record the results of an experiment:

Table 4: Process to capture MOS related to QoE Portal

QoS Invocation	1.1	Trigger/Invoke Services
	1.2	Stop Service
MOS capture	2.1	Pop up with “would you like to enter MOS Data”.
	2.2	Drop down with Rating/Label 5 Excellent 4 Good 3 Fair 2 Poor 1 Bad
	2.3	Disregard/Save
	2.4	Would you like to add another MOS view
	2.5	Repeat
	2.6	End. Show average.
	2.7	Record in history

With this feature it is possible to capture, per service activation, an overall picture of the Quality of Experience obtained while using VELOX to request QoS for the application. In this case the experimenter is responsible of generating and collecting the MOS data.

3.4.2 VELOX to UXM Driver

To facilitate the integration of QoS shaping requests with devices (or applications in the devices) being tested in the UXM a driver in VELOX was created to allow the real-time QoS changes to evaluate the impact of these. The way for VELOX requests QoS to the UXM is via the TAP REST API.

The sequence of events leading up to a specific QoS being loaded in the UXM via VELOX can be seen in Figure 10, the mention Quality plans are predefined XML files that loaded into VELOX and matched to the appropriate VELOX Service. These XML can be also run directly in TAP if needed. This works in the following cases:

1. the QoS settings are set BEFORE the execution of an automated experiment;
2. the QoS setting are set during a non-automated experiment. A non-automated experiment has the limitation of no dynamic channel model, no dynamic variations of the RBs, all the parameters on the UXM need to be set by hand or via a mixture of manual control and manual TAP runs.

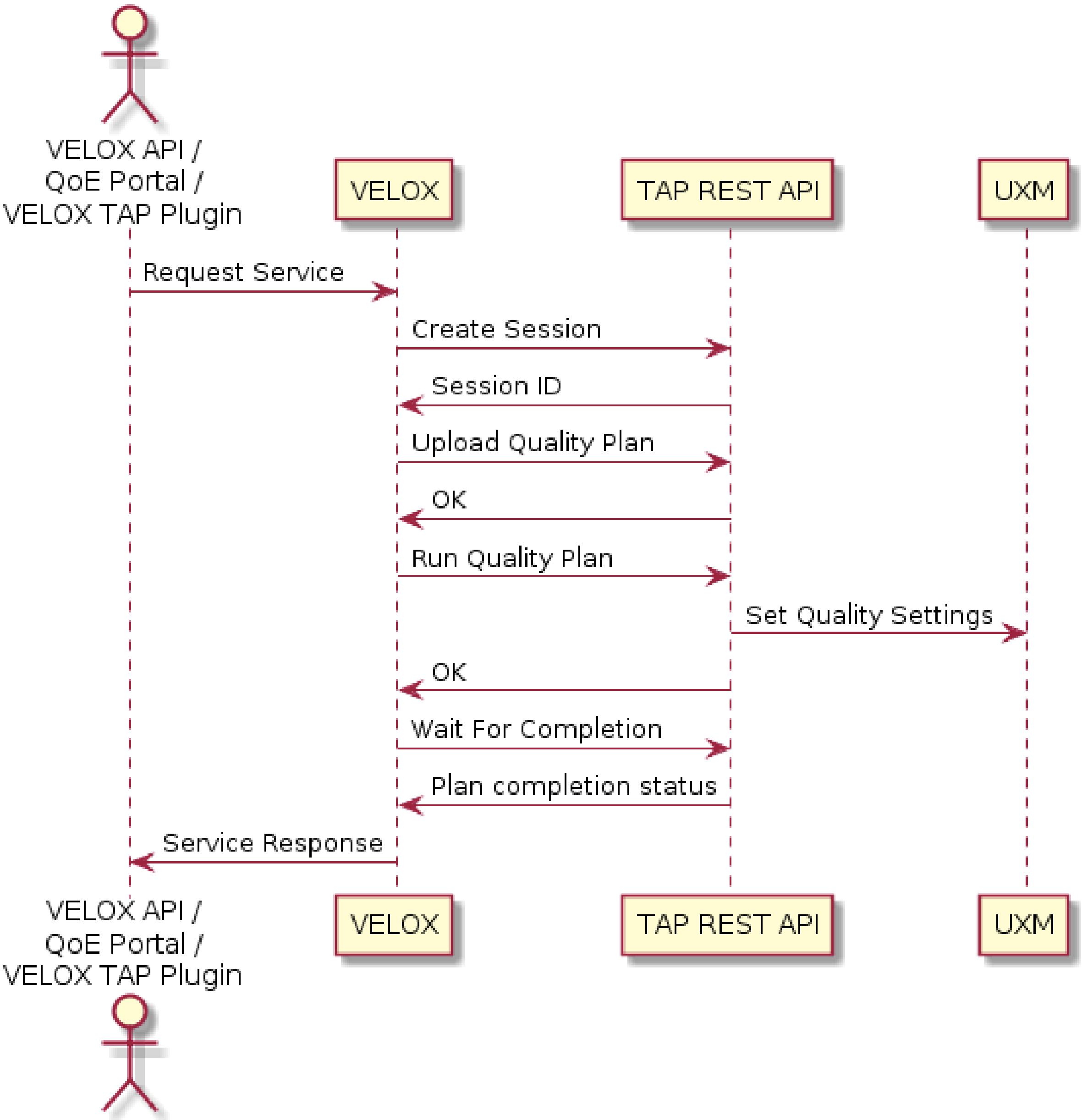


Figure 10: VELOX UXM Driver Request Sequence

This driver can be activated in VELOX in a transparent manner, so no new actions are required, only the use of the VELOX usage methods specified in sections 3.3 and 3.4 above.

VELOX expects response times lower than 2 seconds to be considered real-time and beyond that it returns an error due to timeout. As a results usage response times are high due to the process of session creation in the TAP REST API. On suggested method by which these can be mitigated is the usage of SSD hard-drives in the TAP server.

3.5 SDN Deployment

A small setup of Openflow switches and controller has been deployed for research experimentation.

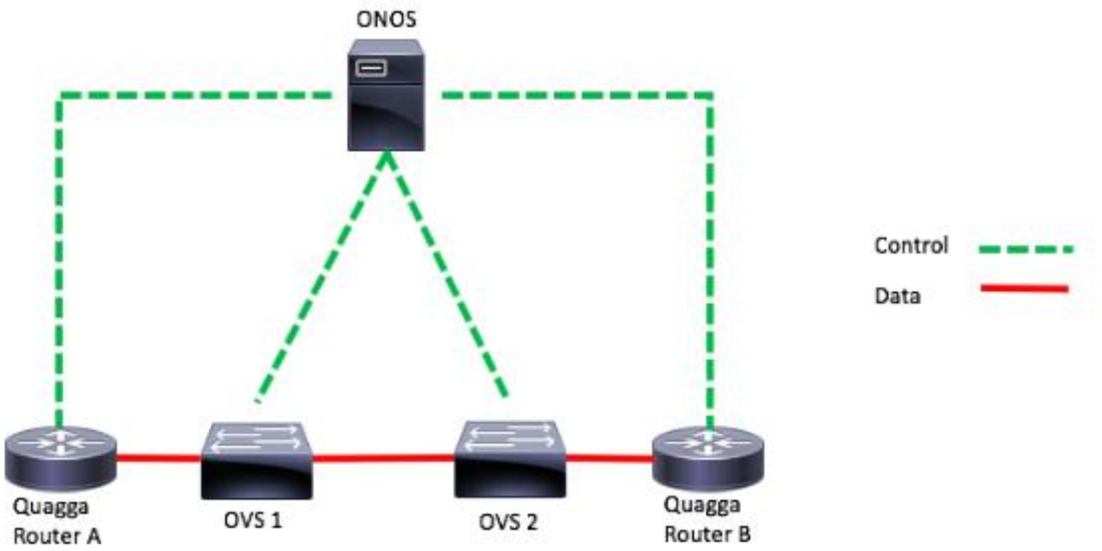


Figure 11: SDN Setup in Triangle

This setup allows for small scale experimentation of Software Defined Networks (SDN). It is comprised of two Open vSwitches (OvS) with pre-installed QoS queues, two Quagga routers and an ONOS SDN controller as seen in Figure 11.

As outlined in D4.1 [14] an SDN testbed was deployed to the TRIANGLE Testbed in Malaga. An elaborated version of this is shown in Figure 12. We consider two domains. One domain is the back-haul domain with a distinct autonomous system number. One domain is the core network including radio access network and EPC. The VPS engine controls the QoS at the radio in this network (ASN 65001). A different VPS engine is signalled in ASN 65002, to allocate a virtual path slice in the back-haul domain. (Note ASN means autonomous system number and defines the number of the network administration).

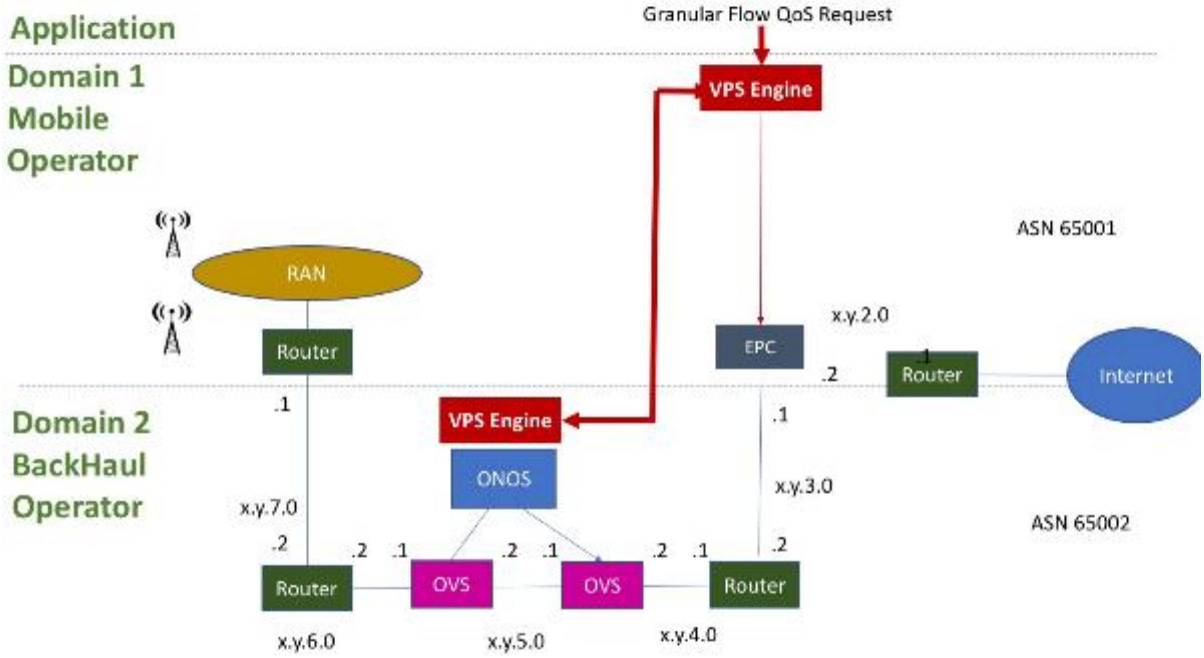


Figure 12: Multidomain SDN deployment model

It is also possible to enable QoS rules in the SDN setup via VELOX. This QoS setting is enabled through a combination of flows and the pre-installed queues in the OvS. This enabling and disabling of QoS is completely transparent to the user and possible with any of the methods already stated for VELOX use.

4 VM Monitoring

4.1 App back end processing in a virtual machine.

App backend processing affects QoE. This can be measured in the virtual machine associated with the app using the Nagios Tool.

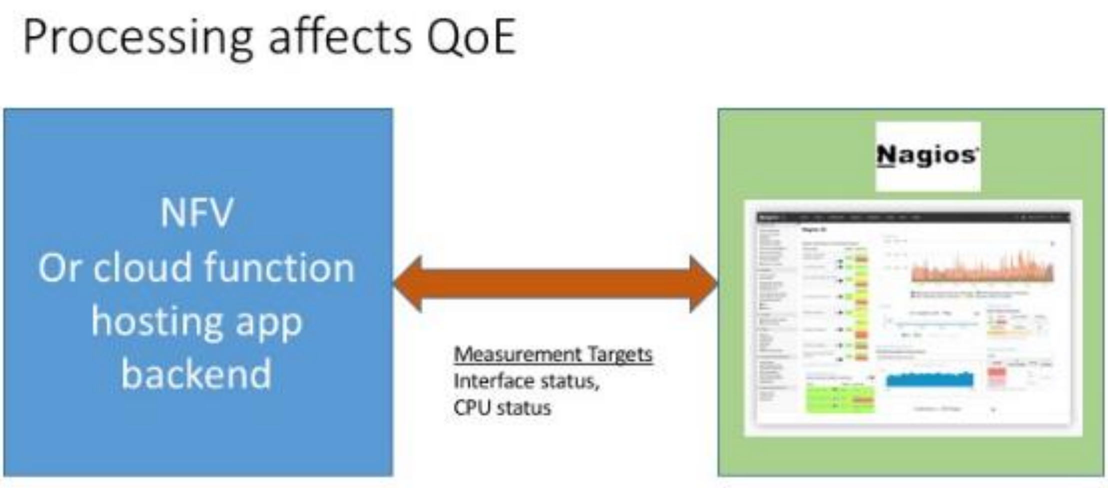


Figure 13: Measuring App Host affecting QoE

4.2 NAGIOS

Nagios is a popular open-source monitoring system. It keeps an inventory of the servers and monitors them so there will be possible to know when critical services are up and running. It is an essential tool for any production environment, because by monitoring uptime, CPU usage, or disk space, it will be possible to head off problems before they occur.

More detailed information is available at the official Nagios page [16].

In TRIANGLE, the Nagios Server was installed in Ubuntu 16.04.3 LTS x86_64 with private networking configured and a web server (Apache) and PHP installed.

The hosts to be monitored are also Ubuntu 16.04.3 LTS x86_64 within the private network of the Nagios Server.

Figure 14: Nagios Map

The recollected data will be shown in the web server installed in the address: http://172.30.1.23/nagios/

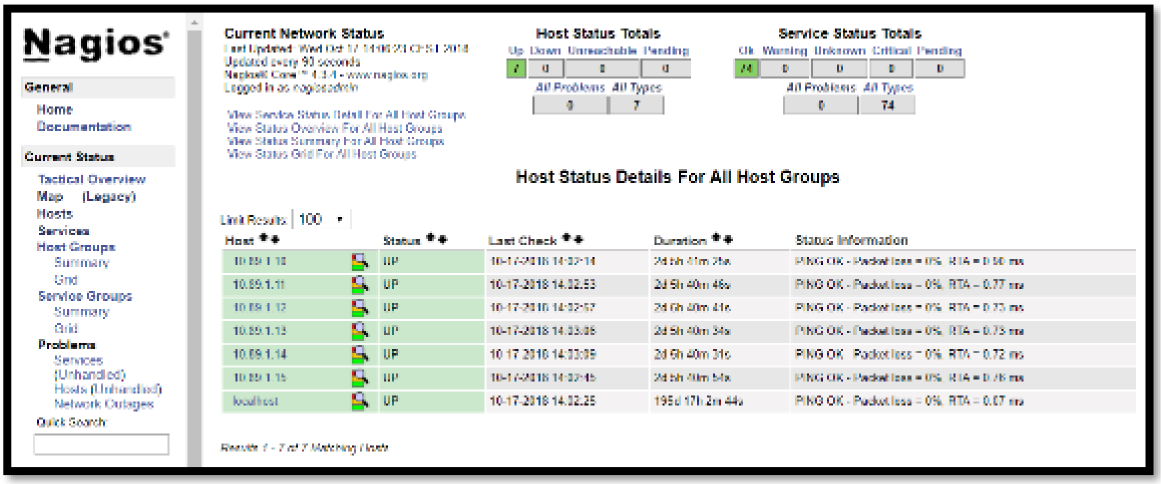
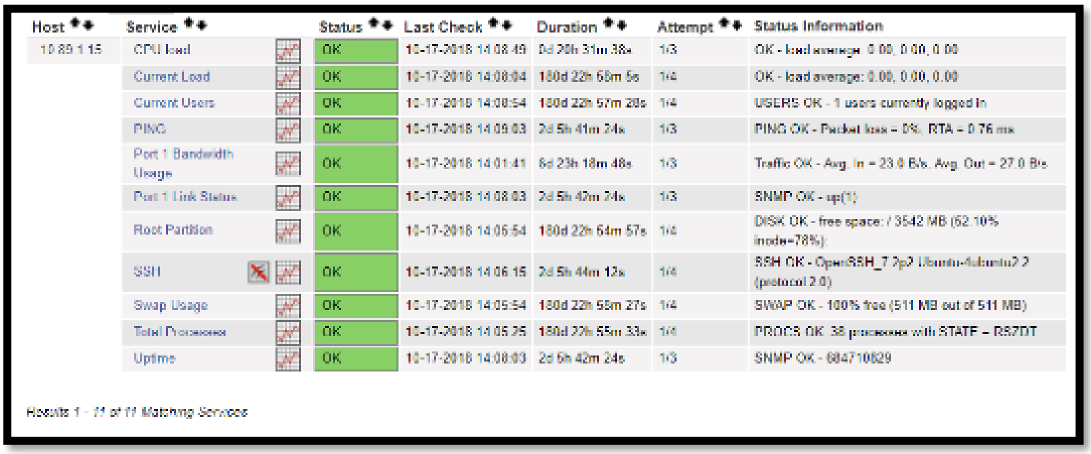


Figure 15: Monitored Nagios Host

In the server, it was installed the MRTG software for monitoring and measuring the traffic load on network links. It allows the user to see traffic load on a network over time in graphical form. It was also installed the Nagiosgraph tool that will allow us to collect a graphic output of the collected data in the last day, week, month and year.

A monitored data sample is shown in Figure 16. CPU Load is shown in Figure 17 and Figure 18.



Host	Service	Status	Last Check	Duration	Attempt	Status Information
10.89.1.15	CPU load	OK	10-17-2018 14:08:49	0d 20h 31m 38s	1/3	OK - load average: 0.00, 0.00, 0.00
	Current Load	OK	10-17-2018 14:08:04	180d 22h 58m 5s	1/4	OK - load average: 0.00, 0.00, 0.00
	Current Users	OK	10-17-2018 14:08:54	180d 22h 57m 28s	1/4	USERS OK - 1 users currently logged in
	PING	OK	10-17-2018 14:09:03	2d 5h 41m 24s	1/3	PING OK - Packet loss = 0%, RTA = 0.76 ms
	Port 1 Bandwidth Usage	OK	10-17-2018 14:01:41	8d 23h 18m 48s	1/3	Traffic OK - Avg. In = 23.0 B/s, Avg. Out = 27.0 B/s
	Port 1 Link Status	OK	10-17-2018 14:08:03	2d 5h 42m 24s	1/3	SNMP OK - up(1)
	Root Partition	OK	10-17-2018 14:05:54	180d 22h 54m 57s	1/4	DISK OK - free space: / 3542 MB (62.10% inode=78%):
	SSH	OK	10-17-2018 14:06:15	2d 5h 44m 12s	1/4	SSH OK - OpenSSH_7.2p2 Ubuntu-4ubuntu2.2 (protocol 2.0)
	Swap Usage	OK	10-17-2018 14:05:54	180d 22h 55m 27s	1/4	SWAP OK - 100% free (511 MB out of 511 MB)
	Total Processes	OK	10-17-2018 14:05:25	180d 22h 55m 33s	1/4	PROCS OK: 38 processes with STATE = RSZDT
	Uptime	OK	10-17-2018 14:08:03	2d 5h 42m 24s	1/3	SNMP OK - 684710829

Results 1 - 11 of 11 Matching Services

Figure 16: Monitored Data in 10.89.1.15 Host

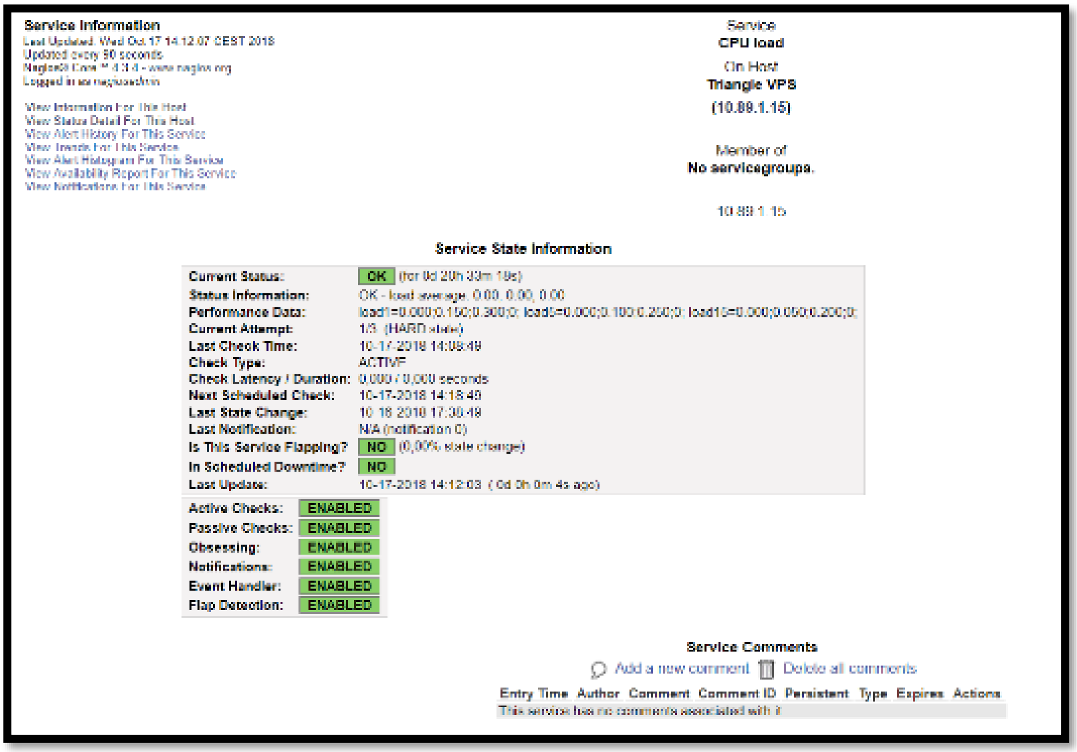


Figure 17: CPU Load Data

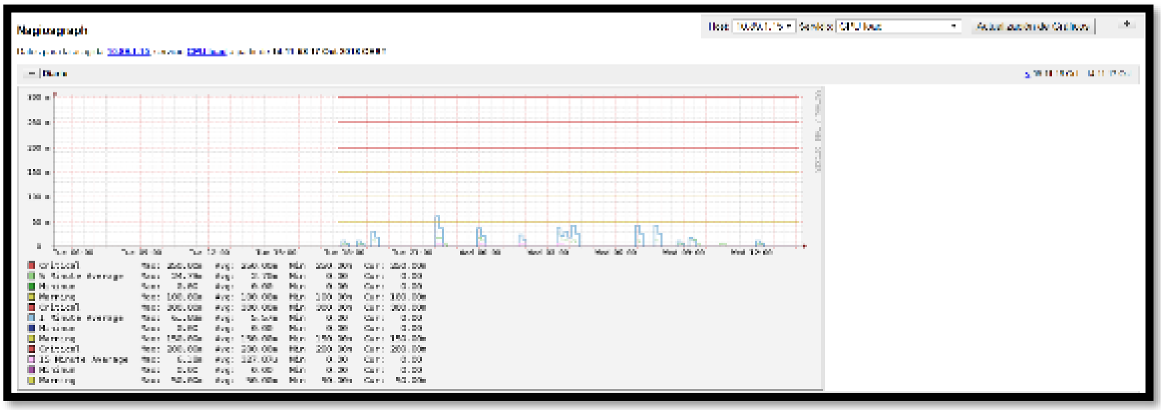


Figure 18: CPU Load Daily Graph

Current Load:

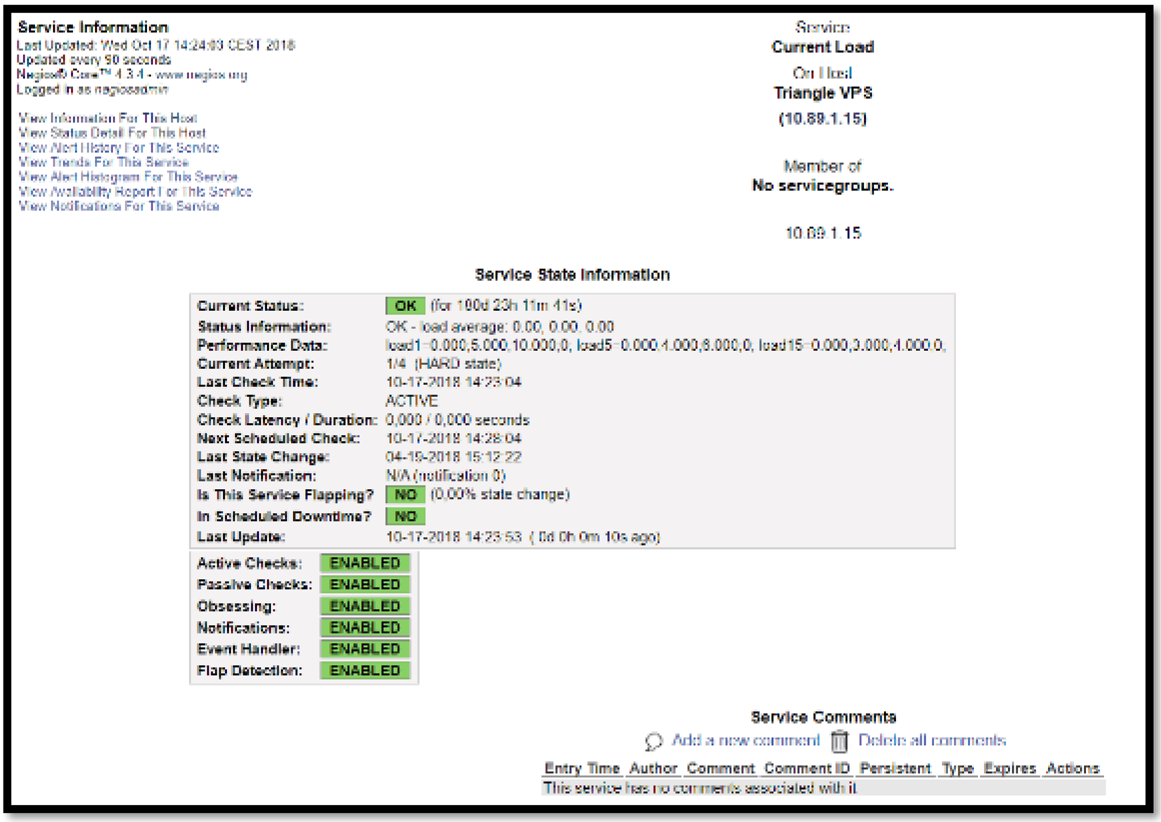


Figure 19: Current Load Data

4.3 Addressing QoE Hot Spots

Naigos can provide insight to resolving hot spots in performance measurements related to apps. We can use the Nagios measurements to obtain learning about the slice bandwidth capacity between the virtual machines in the slice and the performance of the VMs in terms of memory and CPU performance.

Figure 20 shows that the application owner can make adjustments in the capacity at three levels:

(i) virtual machine parameters,
(ii) air interface or
(iii) slice capacity to address a hot spot in the quality of experience.

Adjusting virtual machine parameters (CPU cores, RAM) can improve app performance if the hot spot is in the virtual machines. If the hot spot is bandwidth related, then capacity can be increased. Also, adjusting Gold/Silver/Bronze parameters such as resource blocks can improve app performance if the hot spot is in the radio access. This measurement is out of scope of Naigos and would typically be inside the application.

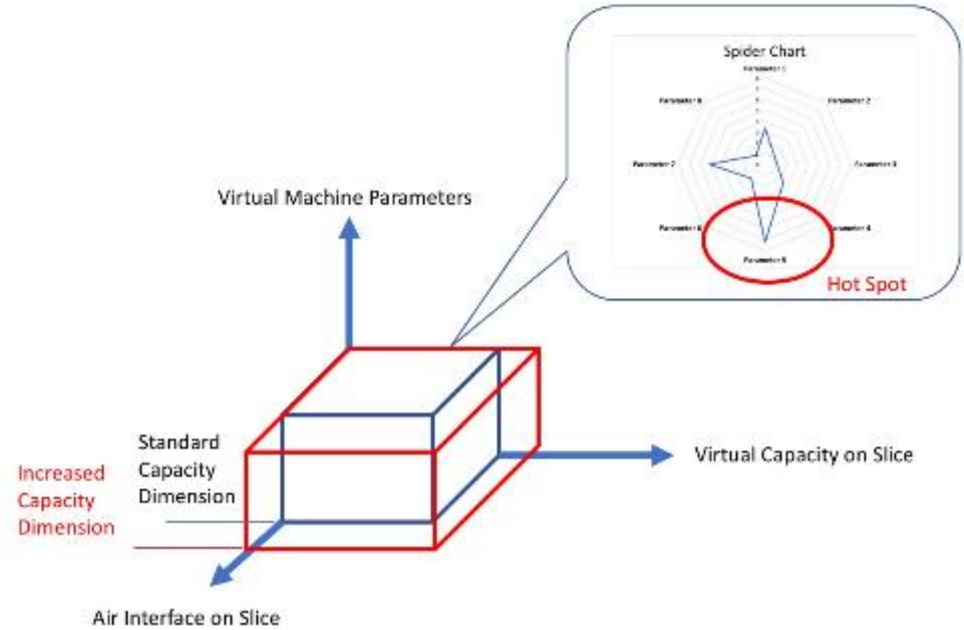


Figure 20: Dimension adjustment to address hot spots

5 Future Models for testing and conclusions

5.1 Future model for implementing QoS Testing with dynamic resource block allocation

We propose a potential model for implementing QoS testing of 5G Applications for testers which have ability to manage schedulers and resource blocks dynamically[1]. An outline model is shown in Figure 22. In this model we consider a tester which is QoS capable. A test control dashboard controls Gold/Silver/Bronze service model into a test automation program (TAP). Resources are configured into the tester. In addition, resources are also configured into the QoS application management function (VELOX). Air interface resource blocks related to the Gold/Silver/Bronze can be implemented by the Tester under the command of the test automation program or via the signals in the channel related to the resource block request.

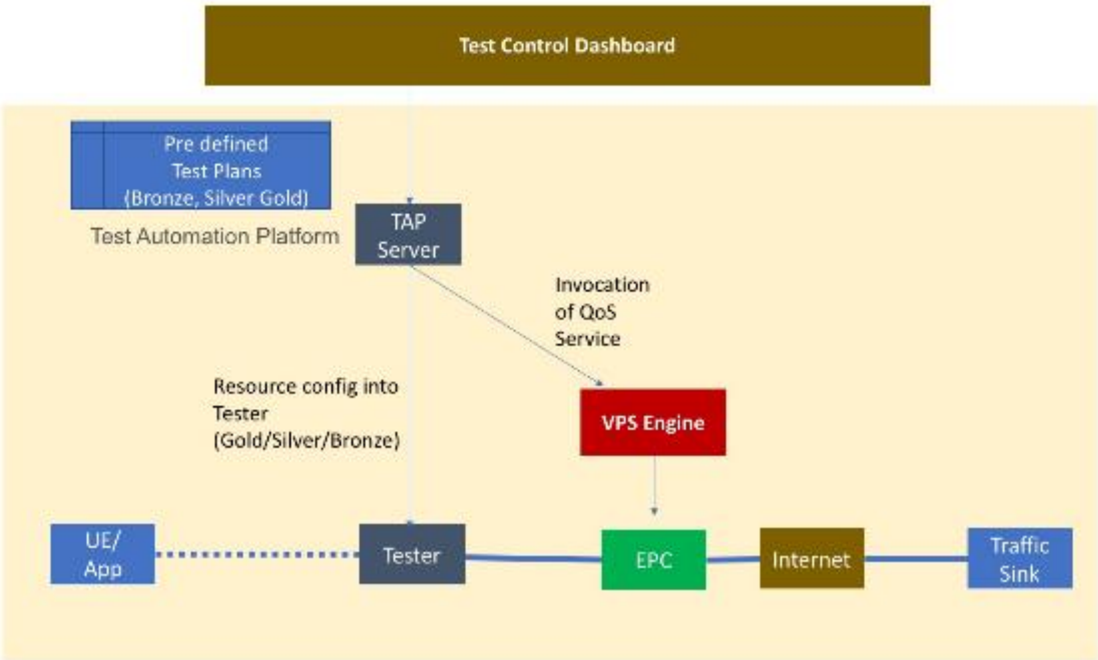


Figure 21: model for a tester with dynamic QoS selection

The detailed steps are given in Figure 22. This process requires a tester which can allocate uplink and down link resources based on QCI Bandwidth profiles. The steps are summarized in the table below. These steps are based on a guaranteed bit rate (GPR) model.

[1] It should be noted this is not in the scope of this project, but a constructive exercise on how to improve the process.

1	Gold, Silver, Bronze Selection	In this step the user of the application under test selects the appropriate service profile (e.g. Gold)
2	DUT Connection	In this step the device under test (DUT) is connected to the tester
3	Test Program Management	In this step the test controller dashboard selects a set of test campaigns in the test automation platform (TAP)
4	Override	(optional) In this step we override the EPC functions so that the tester can directly select resource blocks in uplink and downlink.
5	S1	The tester is connected to the EPC via the S1 interface.
6	Invocation	In this step the test automation program selects (Gold/Silver/Bronze) service as part of an automatic program and invoke its according to the program sequence.
7 8	Manual Invocation	(optional) In this step the dashboard can use the QoE Portal to select a (Gold/Silver/Bronze) service and invoke it manually
9	Application Function Control	In this step the application function interface Rx (new name N5) can select to the PCRF (policy rules charging function) the appropriate QCI (quality class indicator) and uplink/downlink bandwidth.

Table 5: Detailed Steps

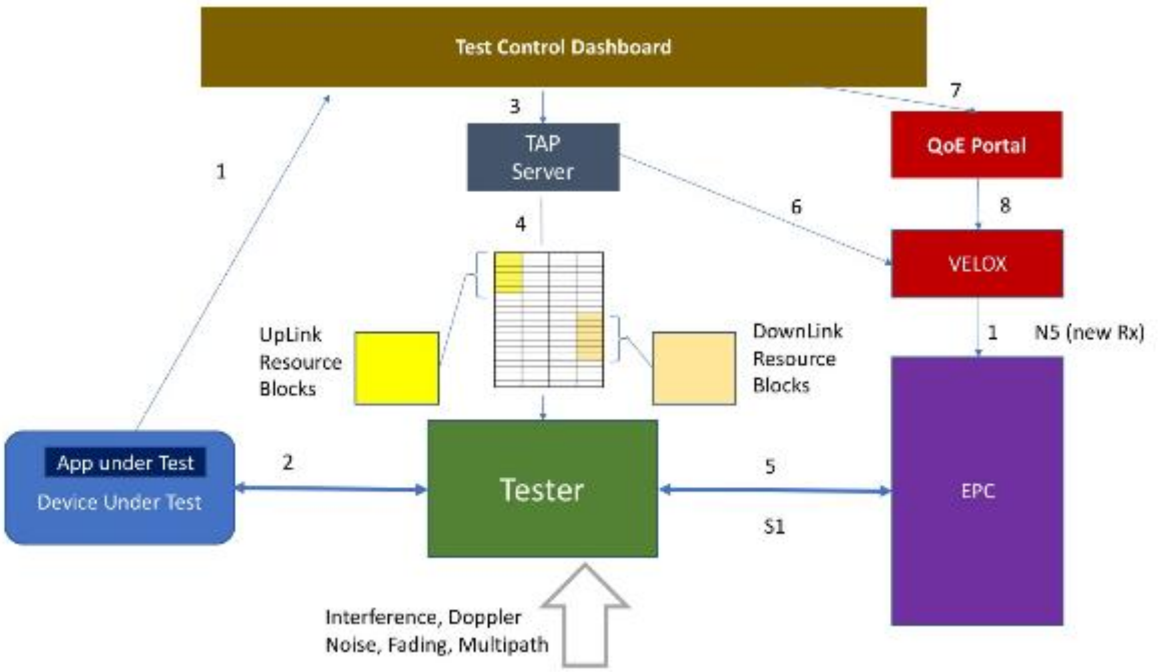


Figure 22: Detailed Steps

5.2 Conclusion

We provide the tools for application developers and researchers to mitigate the impact of network conditions by requesting a Quality of Service (QoS) to be configured on the network that suits their application to research and learn how this can improve their overall perceived Quality of Experience.

In this deliverable we provide methods for automating the process between the user who can select a differentiated (gold/silver/bronze) service and automatically execute it on the TRIANGLE pipeline. We provide an automated method for testing QoS based services using the tools from RedZinc, Keysight and University of Malaga.

This is implemented in the context of slicing, integrating a QoE portal to a virtual path slice engine and to tester which can emulate various radio conditions.

We deployed an SDN environment to the testbed and a method of testing the performance of each virtual machine. This can be used to determine if performance limitations are based on network or back end process or both.

We conducted a gap analysis to regarding the future evolution of testing from the point of view of QoS. and the evolution of the market place for 5G services.

6 Bibliography

[1] IETF, Resource ReSerVation Protocol (RSVP) - RFC 2205.

[2] IETF, Definition of the Differentiated Services Field RFC 2474.

[3] IETF, RSVP-TE: Extensions to RSVP for LSP Tunnels RFC-2309.

[4] 5G Manifesto for timely deployment of 5G in Europe., http://ec.europa.eu/newsroom/dae/document.cfm?action=display&doc_id=16579.

[5] Wikipedia, https://en.wikipedia.org/wiki/Net_neutrality.

[6] Wikipedia, https://en.wikipedia.org/wiki/Policy_and_charging_rules_function.

[7] ETSI, Policy and Charging Control signalling flows and QoS parameter mapping;, https://www.etsi.org/deliver/etsi_ts/129500_129599/129513/15.00.00_60/ts_129513v150000p.pdf (valid 31/10/2018).

[8] 5G White Paper, NGMN Alliance.

[9] DT, NETWORK SLICING IN 5G. HansEinsiedler. Deutsche Telekom. 5G Config Group.

[10] SONATA, http://www.sonata-nfv.eu (valid 31/10/2018).

[11] 5Gex, http://www.5gex.eu (valid 31/10/2018).

[12] SliceNet, https://slicenet.eu (valid 31/10/2018).

[13] ITU, Recommendation ITU-R M.2083-0. (09/2015). IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond. M Series.

[14] TRIANGLE, Deliverable D4.1 Report on the networking capabilities provided by the testbed.

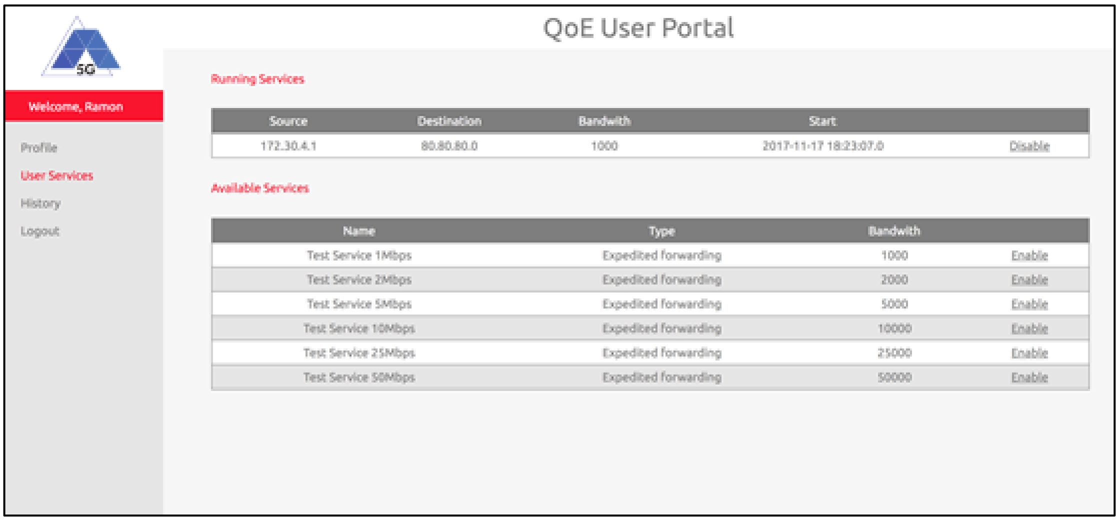
[15] TRIANGLE, Deliverable D2.7 QoE Evaluation: The TRIANGLE Testbed Approach.

[16] Nagios Documentation, https://www.nagios.org/documentation/.

Appendix 1 QoE Portal Overview

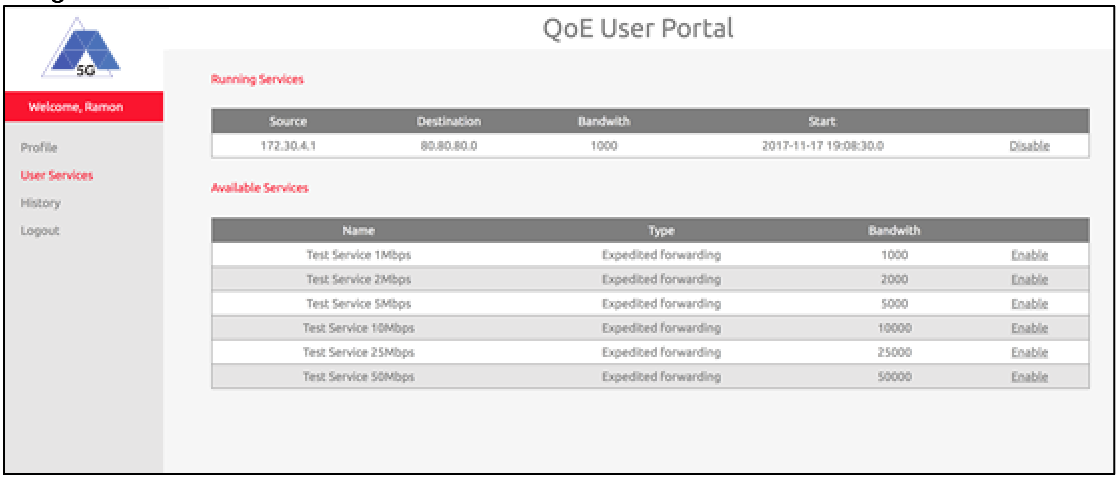
When you click on ‘Create an Account’, the page appears.

There are two profile’s type: User and User Management. First, the User will be shown.

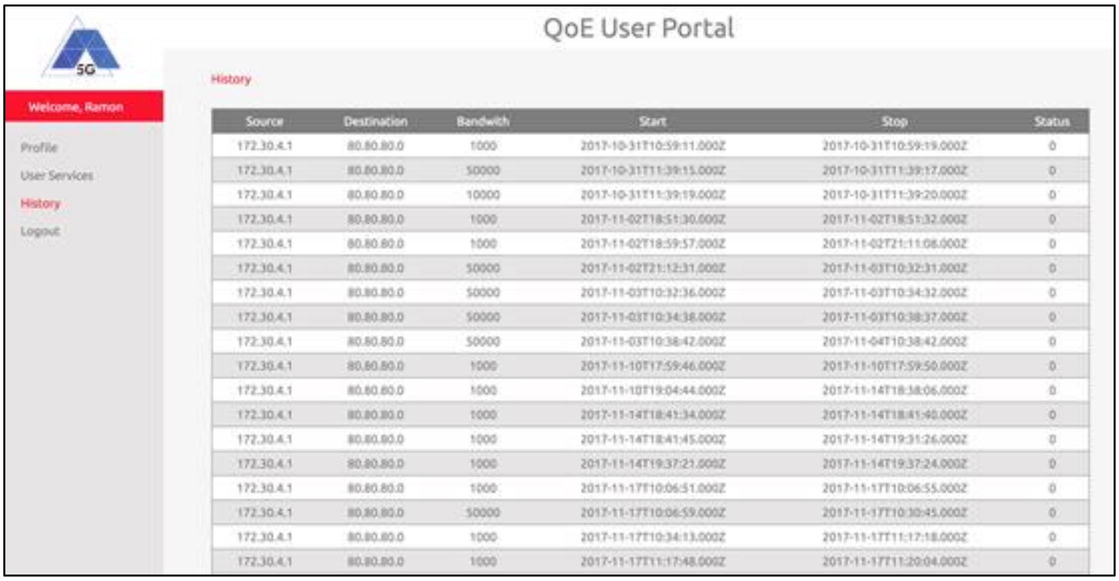


The main idea this feature is to display the ‘Available Services’ and ‘Running Services’. When you click on ‘Enable’ in ‘Available Services’ table, you will be able to active a service.

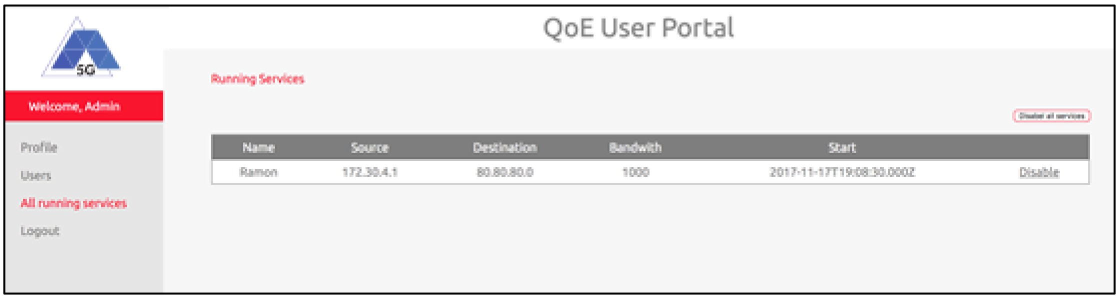
The system request ‘Source’ and ‘Destination’. After this information the service is running.



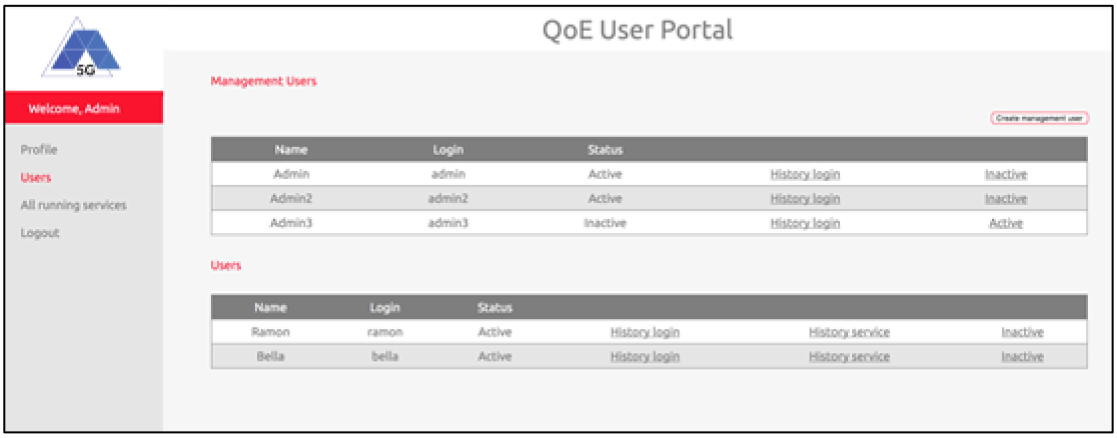
The ‘History’ item, it shows all history user’s services.



Now, User Management will be shown.

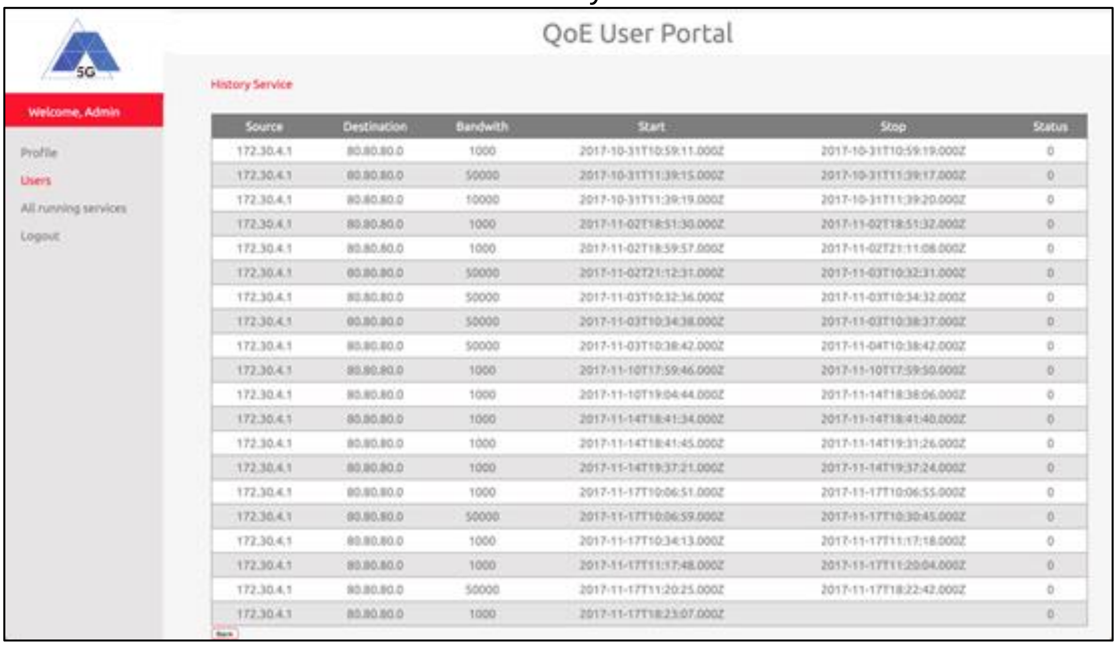


It shows all running services and you can disable one service or all services. The 'Users' item allows, the management about all users.



This feature allows the activation and deactivation of users, create other management user and view history about services and login.

Screen History Service



Appendix 2 Allocation of LTE scheduling capacity to achieve the target bandwidth for the gold, silver and bronze models.

		Urban-Office	Gold, Silver, Bronze Model						Platinum Model				
		UR-OF											
High level scenario description		Business building area - stationary user, good coverage, high number of users	Business building area - stationary user, good coverage, high number of users						Business building area - stationary user, good coverage, high number of users				
Sub-scenario description		**Default working conditions** (average NW & cell load)	**Default working conditions** (average NW & cell load)	**Scenario with high user load and low bandwidth 9e.g. football stadium or congested street)**	**Bronze-2-x**	**Silver-4-x**	**Gold-6-x**	**Super Gold-10-x**	**Default working conditions** (average NW & cell load)	**Platinum -10-x**	**Platinum -20-x**	**Platinum -40-x**	**Platinum -60-x**
Serving cell	RSRP	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm	fixed at -90dBm

	AWGN	Constant AWGN level for average SNR = 25dB	Constant AWGN level for average SNR = 25dB						Constant AWGN level for average SNR = 25dB				
	Channel model	EPA	EPA	EPA	EPA	EPA	EPA	EPA	EPA	EPA	EPA	EPA	EPA
	Channel model Doppler	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz	5 Hz
	Channel model correlation	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium	Medium
LTE scheduling	Frequency domain (DL)	50.00%	50.00%	4.00%	20.00%	20.00%	42.00%	42.00%	25.00%	24.00%	48.00%	48.00%	48.00%
	Time domain (DL)	30.00%	30.00%	7.00%	7.00%	15.00%	15.00%	19.00%	5.00%	10.00%	10.00%	20.00%	30.00%
	Frequency domain (UL)	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%
	Time domain (UL)	30.00%	30.00%	30.00%	30.00%	30.00%	30.00%	30.00%	30.00%	30.00%	30.00%	30.00%	30.00%
Network	Additional each way latency	0ms	0ms	0ms	0ms	0ms	0ms	0ms	0ms	0ms	0ms	0ms	0ms

Comments		Medium correlation due to offices being in skyscrapers/tall buildings	Medium correlation due to offices being in skyscrapers/tall buildings						Medium correlation due to offices being in skyscrapers/tall buildings				
4G Model	Cell DL load (other users)	85.00%	85.00%	99.72%	98.60%	97.00%	93.70%	92.02%					
125	Max capacity (125*(1-CellLoad))	18.75	18.75	0.35	1.75	3.75	7.875	9.975					
	Average capacity (measured)	16											
5G Model	Cell DL load (other users)								98.7500%	97.6000%	95.2000%	90.4000%	85.6000%
400	Max capacity 400*(1-CellLoad))								5	9.6	19.2	38.4	57.6

Average capacity (measured)				

Appendix 3 Application Programming Interface

The VELOX VPS Engine API provides developers with a set of resources that allow the listing and control of VPS services.

The API is a set of JSON encoded requests and responses sent over a TCP connection. All service availability is bound to the API Key used.

In order to use the VELOX API an application must:

1. Create a TCP connection to known IP address/port (provided by local operator)
2. Write Request (as a single text line, new line ends a request)
3. Read Response (sent as a single line)

Connections are terminated on the VELOX side after sending the response

All Requests must use the API key generated by the local operator VELOX.

API Key
The API Key provided will always be a standard UUID in human readable format without dashes. Example:

ECE335024E3E466CA98BF5014D5C7D86

IPv4 vs IPv6
VELOX Supports both IPv4 and IPv6 services, but does not allow IPv4 mixed with IPv6, in requests that have both source and destination addresses, both must be of the same IP version. All versions of IPv6 abbreviation are supported.

Security
Currently the system considers a safe connection already exists between the Operator and the 3rd Party.

Trigger Request
Starts a new session of an available service between two provided IPs. All services are started as unidirectional with an optional argument to make it bidirectional. Used for initiating a service between two IPs.

Returns a dataset containing the response code and the session ID of the running service in case of a successful trigger.

Message format

TRIGGER Request		
Field	**Type**	**Description**
key	String	API Key
type	1	Request type, must be 1

source	String	IPv4/6 of the source point
destination	String	IPv4/6 of the destination point
service	Integer	ID of the service to trigger
bidirectional	Boolean	OPTIONAL: set to “true” for bidirectional service, omitted or set to “false” for unidirectional service.

TRIGGER Request Example

{"key":"ECE335024E3E466CA98BF5014D5C7D86", "type":1, "source": "200.20.10.32", "destination": "193.22.33.55", "service": 16}

TRIGGER Response		
Field	**Type**	**Description**
type	Integer	Is 1 if successful or 0 in case of a generic error
code	Integer	Assumes a value from the Response Codes table
session	String	Alphanumeric string of variable length that uniquely identifies a service session, will be empty in case of error

TRIGGER Response Example

{"type":1, "code":0, "session":"25172.16.14.10172.16.3.1300.403685523078995513327567058 91"}

Stop Request

Stops the running service session provided by ID.

Returns a dataset containing the response code.

Message format:

STOP Request		
Field	**Type**	**Description**
key	String	API Key
type	2	Request type, must be 2
session	String	Alphanumeric string that identifies the service session to stop

STOP Request Example

{"key":"ECE335024E3E466CA98BF5014D5C7D86", "type":2, "session": "25172.16.14.10172.16.3.1300.4036855230789955133275670589 1"}

STOP Response		
Field	**Type**	**Description**
type	Integer	Is 2 if successful or 0 in case of a generic error
code	Integer	Assumes a value from the Response Codes table

STOP Response Example

{"type":2, "code":0}

Modify Request

Modifies a running session provided by ID, changing the base service used.

Returns a dataset containing the response code.

Message format:

Modify Request		
Field	**Type**	**Description**
key	String	API Key
type	3	Request type, must be 3
session	String	Alphanumeric string that identifies the service session to stop
service	Integer	ID of the service to modify the session to

MODIFY Request Example

{"key":"ECE335024E3E466CA98BF5014D5C7D86", "type":3, "session":"25172.16.14.10172.16.3.1300.40368552307899551332756705891", "service": 14}

Modify Response		
Field	**Type**	**Description**
type	Integer	Is 3 if successful or 0 in case of a generic error
code	Integer	Assumes a value from the Response Codes table

MODIFY Response Example

{"type":3, "code":0}

LISTING REQUESTS

List Request
Generates a list of services available for Trigger and Modify Requests.

Returns a dataset containing the service list.

Service Format:

Service		
Field	**Type**	**Description**
id	Integer	Service ID
name	String	Service Name
type	String	Type of Service handling inside the operator (Ex: Expedited Forwarding)
bandwidth	Integer	Service Speed in Kbps

Message format:

List Request		
Field	**Type**	**Description**
key	String	API Key
type	4	Request type, must be 4

LIST Request Example

{"key":"ECE335024E3E466CA98BF5014D5C7D86", "type":4}

List Response		
Field	**Type**	**Description**
type	Integer	Is 4 if successful or 0 in case of a generic error
code	Integer	Assumes a value from the Response Codes table
services	Array of Service	Array of available services (in the specified format), can be and empty array in the event of no available services

LIST Response Example

{"type":4, "code":0,"services":[{"id":14,"name":"Operator Service 5Mbps", "type":"EF", "bandwidth":5000}, {"id":16,"name":"Operator Service 2Mbps", "type":"EF", "bandwidth":2000}]}

Run Request
Generates a list of running service sessions.

Returns a dataset containing the session list.

Session format:

Session		
Field	**Type**	**Description**
id	String	Alphanumeric string that identifies the service session
service	Integer	Base Service ID
bandwidth	Integer	Service Speed in Kbps
source	String	IPv4 of the source point
destination	String	IPv4 of the destination point
start	String	UTC Data and Time of the session start

Message format:

Run Request		
Field	**Type**	**Description**
key	String	API Key
type	5	Request type, must be 5

RUN Request Example

{"key":"ECE335024E3E466CA98BF5014D5C7D86", "type":5}

Run Response		
Field	**Type**	**Description**
type	Integer	Is 5 if successful or 0 in case of a generic error
code	Integer	Assumes a value from the Response Codes table
sessions	Array of Session	Array of running session (in the specified format), can be and empty array in the event of no running session

RUN Response Example

{"type":5, "code":0
"sessions":[{"id":"710.1.3.110.1.1.10.448130380713861241347963976614","service":6, "bandwidth":150, "source":"10.1.3.1","destination":"10.1.1.1","start":"2012-09-18 10:26:18"},{"id":"710.1.3.110.1.1.10.747157442557350613479737382 28","service":7, "bandwidth":100, "source":"10.1.3.1","destination":"10.1.1.1","start":"2012-09-18 13:09:00"}]}

Response Codes

Some codes might not appear for some request responses.

Code	Meaning
0	Request executed successfully
1	Invalid API Key
2	Unknown Request
3	Invalid arguments
4	Invalid service
5	Invalid session
6	Insufficient bandwidth available (on modify it is considered that the service remains running unaltered)
7	No Path between source and destination points with requested service type
8	Unable to execute operation (internal failure)
9	Nothing to Modify (trying to modify to running service)
10	Source-Destination pair already being used in another service

Generic Error Responses

Some errors are generic in nature and do not conform to a specific type and as such will return as type 0 with the error code:

Generic Error Example

```
{"type":0, "code":1}
```