

H2020-ICT-688712



Project: H2020-ICT-688712

Project Name:

5G Applications and Devices Benchmarking (TRIANGLE)

Deliverable D4.1

Report on the networking capabilities provided by the
testbed

Date of delivery: 31/10/2017

Version: 1.0

Start date of Project: 01/01/2016

Duration: 36 months



Deliverable D4.1

Report on the networking capabilities provided by the testbed

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

Document Number:	ICT-688712-TRIANGLE/D4.1
Document Title:	Report on the networking capabilities provided by the testbed
Lead beneficiary:	Keysight Technologies Denmark
Editor(s):	Andrea F. Cattoni (KEYD)
Authors:	German Corrales Madueno (KEYD), Cesar Augusto Garcia (UMA), Alvaro Martin (UMA), Iván González (UMA), Pedro Merino (UMA), Donal Morris (REDZ), Ricardo Figueiredo (REDZ), Janie Baños (AT4), Manuel García (AT4), Carlos Cárdenas (AT4), Oscar Castañeda (AT4)
Dissemination Level:	PU
Contractual Date of Delivery:	31/10/2017
Work Package Leader:	Andrea F. Cattoni (KEYD)
Status:	Final
Version:	1.0
File Name:	TRIANGLE_Deliverable_D4.1_final.docx

Abstract

This deliverable contains a summary of the development activities performed in WP4. The Work Package focuses on implementing network and network testing functionalities in order to expand the capabilities and the use cases that the TRIANGLE testing approach can support.

Keywords

Network testing, S1 Interface, LWIP Interface, LTE, Wi-Fi, Over-the-Air testing, SDN, NB-IoT, Time synchronization



Executive summary

This deliverable is the summary of all the development activities related to the networking part of the testbed. Work Package 4 contains, all the activities focused at the improvement of the testbed capabilities in order to make the network emulation as much realistic as possible.

As in any wireless access architecture, the testbed has three main networking components that needed upgrade to achieve the End-to-End testing goal set by the project: the Radio Access Network (RAN), the Transport Network and the Core network (EPC). The document has three main sections that represent such components.

In Section 2 all the activities for improving the Radio Access Network capabilities are highlighted. Such activities were focused on three main points:

- Inclusion of the new 3GPP NB-IoT standard for machine-type communication in the UXM wireless test set protocol stack.
- Research and evaluation of device through Over-the-Air measurement in anechoic chamber.
- Inclusion of the LTE-WLAN IP interface (LWIP) to enable LTE-Wi-Fi data aggregation, that is one of the main differentiator for 5G heterogeneous access. Together with the network side, a specific UE client had to be developed given the lack of COTS equipment.

In Section 3 all the developments performed for improving the testing capabilities in the packet network side are explained. The activities are subdivided into three main branches:

- S1 interface inclusion in the UXM, in order to transform the wireless test set into a realistic base station (eNodeB) emulator, capable of being connected to a commercial Evolved Packet Core (EPC).
- Development of a Transport Network Emulation system, to include backhaul impairments into the emulated network scenarios.
- Integration of a QoS broker into the system, able to specifically request QoS upgrades to the EPC and the transport.

Section 4 outlines other technologies that are not directly related to traditional network architectures but enable the End-to-End testing approach proposed by TRIANGLE. The subsections contain information about:

- Development of IEEE 1588-based time synchronization for hardware and software testbed components.
- Distributed network measurements for enabling packet-based analysis and packet tracing along the network.
- EPC control framework allowing to dynamically configuring and re-configuring the EPC parameters, together with injection of negative behaviour in the EPC components and triggering specific EPC-initiated procedures.

Finally, Section 5 provides a wrap-up conclusion about the work carried out by the TRIANGLE project in WP4 during the first two years of the project.



Contents

1	Introduction	1
2	RAN Emulation Capabilities	2
2.1	NB-IoT Cat. NB1	2
	Introduction	2
	NB-IoT Basics	2
	NB-IoT Extension	3
2.2	Antenna Characteristics Emulation	4
	OTA techniques to incorporate the antennas effect	4
	Work Plan.....	4
	Phase 1	5
2.3	Introduction to LWIP.....	8
	LWIP Standard Architecture	9
	Architecture Adaptation to Testbed.....	11
3	Networking Capabilities.....	17
3.1	S1 Interface.....	17
	Introduction	17
	UXM S1-MME and S1-U Extensions	18
3.2	Transport Network Emulation	19
	Architecture	19
	Exposed Functionality	20
3.3	QoS Improvements via SDN Entities.....	23
	Deployment details.....	25
	ONOS modifications to enable virtual path slicing	25
4	Technology Enablers.....	27
4.1	High precision time synchronization	27
	High precision time synchronization results	28
4.2	Network measurement framework.....	30
4.3	EPC Control	31
	Architecture	31
	Current Functionality	31
5	Conclusion	34
6	References.....	35
	Appendix 1: S1 Library	36
	S1 Library public API functions and parameters.....	41
	Appendix 2: LWIP header treatment.....	44



List of Figures

Figure 1 - Typical Downlink MIMO MPAC system	5
Figure 2 - Throughput vs RS-EPRE curves	6
Figure 3 - Results for AAD_1 (device 1) P0 with UMa @ 70% outage level	7
Figure 4 - Results for AAD_1 (device 1) P0 with UMa @ 70% outage level	7
Figure 5 - Results for AAD_1 (device 1) L0 with UMa @ 95% outage level	8
Figure 6 – LWIP overall architecture.....	9
Figure 7 – LWIP detailed architecture.....	10
Figure 8 – LWIP protocol stack.....	11
Figure 9 – Adapted LWIP data plane architecture	12
Figure 10 - LWIP data plane architecture optimized for testbed.....	13
Figure 11 - LWIP control plane architecture optimized for testbed	15
Figure 12 - Detailed architecture optimized for testbed.....	16
Figure 13 - "Vanilla" UXM Operations.....	17
Figure 14 - TRIANGLE Modified UXM Operations	17
Figure 15 Architecture of the S1 Interface Extension.	18
Figure 16: Impairment Plugin Architecture.....	20
Figure 17 Slicing in 5G	24
Figure 18 Basic Transport Topology	24
Figure 19 ONOS Northbound API Extension	25
Figure 20 Virtual Path Slice within a topology Slice	26
Figure 21 Ideal scenario where all the elements in the testbed share the same clock	27
Figure 22 Customized TAP Result Viewer for Current Consumption. Filtering options displayed.	29
Figure 23 Possibility to click the graph or the message to see the other domain.	30
Figure 24: EPC SCPI Server Overview.....	31
Figure 25: Default configuration for the EPC network	32
Figure 26 S1-MME Protocol Stack.....	36
Figure 27 S1Library Architecture	36
Figure 28 S1APDissector example	38
Figure 29 – Model of the LWIPEP sublayer	44



List of Tables

Table 1: Example of alias for the impairment plugin.....	20
Table 2: Standard SCPI Commands of the Impairments SCPI Server.....	20
Table 3: Latency SCPI Commands of the Impairments SCPI Server.....	20
Table 4: Link Error SCPI Commands of the Impairments SCPI Server.....	21
Table 5: Packet Duplication SCPI Commands of the Impairments SCPI Server.....	22
Table 6: Packet Reordering SCPI Commands of the Impairments SCPI Server.....	22
Table 7: Supported Standard SCPI Commands for the EPC SCPI Server.....	32
Table 8: EPC Elements Common Commands for the EPC SCPI Server.....	32
Table 9: MME Specific Commands for the EPC SCPI Server.....	33
Table 10: PCRF Specific Commands for the EPC SCPI Server.....	33
Table 11 S1Library API Functions.....	39
Table 12 S1 Library Supported Procedures.....	39
Table 13 S1Library API functions and parameters list.....	41
Table 14 - LWIP Header structure.....	45



List of Abbreviations

3GPP	3rd Generation Partnership Program
ANDSF	Automatic Network Discovery and Selection Function
AP	Access Point
APER	Aligned Packet Encoding Rules
API	Application Programming Interface
ASN	Abstract Syntax Notation
BER	Bit Error Rate
BGP	Border Gateway Protocol
BLER	Block Error Rate
BSE	Base Station Emulator
CAPEX	CApital EXpenditure
CATL	CTIA Authorized Testing Laboratory
COTS	Commercial Off The Shelf
CR	
CSI	Channel State Information
CTIA	Cellular Telecommunications and Internet Association
D2D	Device-to-Device
DL	Downlink
DMRS	Demodulation reference signal
DoNAS	Data over NAS
DRB	Data Radio Bearer
DRX	Discontinuous Reception
DTX	Discontinuous Transmission
DUT	Device Under Test
E2E	End-to-End
eDRX	Enhanced Discontinuous Reception
EIRP	Effective Isotropic Radiated Power
eNodeB	E-UTRAN Node B

EPC	Evolved Packet Core
ePDG	enhanced Packet Data Gateway
EPRE	Energy Per Resource Element
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
EVM	Error Vector Magnitude
EXP	Experimental
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
GTP	GPRS Tunnelling Protocol
HARQ	Hybrid Automatic Repeat Request
HSS	Home Subscriber Server
ICI	Inter-Carrier Interference
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineers
IMSI	International Mobile Subscriber Identity
IMT	International Mobile Communications
IP	Internet Protocol
IPsec	Secure IP
ISO	International Organization for Standardization
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
LTS	Long Term Support
LWIP	LTE-WLAN IP-level interface
LWIPPEP	LWIP Encapsulation Protocol
M2M	Machine to Machine
MAC	Medium-access Control
Mbps	megabits per second
MIB	Master Information Block
MIMO	Multiple-Input Multiple-Output
MME	Mobile Management Entity
MNO	Mobile Network Operator
MPAC	Multi-Probe Anechoic Chamber
MPLS	MultiProtocol Label Switching
MSE	Mean Squared Error
NAS	Non-Access Stratum
NAT	Network Address Translation
NB-IoT	NarrowBand Internet of Things
NLOS	Nonline of Sight
NPBCH	Narrowband Physical Broadcast CHannel
NPDCCH	Narrowband Physical Downlink Control CHannel
NPDSCH	Narrowband Physical Downlink Shared CHannel
NPRACH	Narrowband Physical Random Access CHannel
NPSS	Narrowband Primary Synchronization Signal

NPSSS	Narrowband Secondary Synchronization Signal
NPUSCH	Narrowband Physical Uplink Shared CHannel
NRS	Narrowband Reference Signal
NTP	Network Time Protocol
OFDM	Orthogonal Frequency Division Multiplexing
ONOS	Open Network Operating System
OPEX	Operational Expenditure
OSI	Open System Interconnection
OTA	Over the Air
OVS	Open Virtual Switch
OVSDb	OVS DataBase
PAPR	Peak-to-Average-Power-Ratio
PCAP	Packet CAPture
PCRF	Policy and Charging Rules Function
PDU	Protocol Data Unit
PGW	Packet Gateway
PHY	Physical Layer
PSM	Power Saving Mode
PTP	Precision Time Protocol
QAM	Quadrature Amplitude Modulation
QCI	QoS Class Identifier
QoE	Quality of experience
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network
RAT	Radio Access Technology
RF	Radio Frequency
RRC	Radio Ressource Control
RRM	Radio Resource Management
RTT	Round Trip Time
S1AP	S1 Application Protocol



SCPI	Standard Commands for Programmable Instruments
SCTP	Stream Control Transport Protocol
SDN	Software Defined Network
SDR	Software Defined Radio
SDU	Service Data Unit
SeGW	Secure GateWay
SGW	Service Gateway
SIB	System Information Block
SINR	Signal to Interference and Noise Ratio
SRB	Service Radio Bearer
SRS	Sounding Reference Signal
SSH	Secure SHell
SUT	System Under Test
TAP	Keysight's Test Automation Platform
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TRX	Transmitter
TTI	Transmission Time Interval
UDP	User Datagram Protocol
UE	User Equipment
UE	User Equipment
UL	Uplink
UMa	Urban Macrocell
UMTS	Universal Mobile Telecommunications System
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-everything
VNF	Virtualized Network Function
VPS	Virtual Path Switch
Wi-Fi	Wireless Fidelity

WLAN	Wireless Local Area Network
-------------	-----------------------------



Document: ICT-688712-TRIANGLE/D4.1

Date: 31/10/2017

Dissemination: PU

Status: Final

Version: 1.0



1 Introduction

Until 4G, there have been two independent types of testing. On the one side, there was the protocol and lower layers testing based on independent emulators capable of creating specific testing situations. In this type of testing, the testing lab has full control over every aspect and parameter of the system. On the other side, End-to-End testing was generally performed by Mobile Network Operators (MNOs) using real radio access network (RAN) equipment and production EPC. The MNO testing lab has, in this case, very limited or no control over the system behaviour. This is due to the limited configurations allowed by commercial RAN equipment, together with the impossibility of modifying a production EPC.

The TRIANGLE project was designed to fill this gap, creating an End-to-End testing environment where the testers had full control over all the elements in the chain. During the project, COTS test and measurement equipment has been upgraded to provide a full End-to-End testing environment, connecting them with the same architecture as present in real networks. At the same time, traditional network components have been instrumented to provide additional testing control to the user and allowing the definition of advanced network test cases.

The resulting system is a testbed where it is possible to have the same control offered by traditional protocol and lower layers testing also at higher ISO-OSI layers, up to the application. By creating a testing sandbox for devices and applications, it is possible to have fully repeatable End-to-End tests and guarantee a proper evaluation of the performance and Quality of Experience (QoE) offered by 5G devices, applications, and services.

Work Package 4 was the part of the project devoted to making the needed modifications and integrating the components for the End-to-End approach. The upgrades covered several aspects: RAN, transport and core network, together with technology enablers that allowed “to glue” together elements, providing control and accurate measurements.

One of the first upgrades made was the introduction of the new 3GPP Rel’13 NB-IoT standard within the eNodeB emulator. This upgrade allowed to expand the market offerings that TRIANGLE as a project can have in the 5G arena.

The introduction of S1 and LWIP interfaces allowed the eNodeB emulator to be an integral part of a 4.9G network, including interworking between LTE and Wi-Fi. Once the eNodeB emulator has been coupled with the EPC, the development of Transport network impairments allows for the creation of advanced and realistic tests. At the same time, aligning with 5G networks required the introduction of a network slicing engine, able to create a network functioning chain similar to a real one.

At the same time, the network was transformed into a test and measurement entity by adding dynamic configuration control and distributed measurement points to allow precise packet tracing. To have a precise reference for the measurement, a time synchronization framework, based on the IEEE 1588 standard, has been introduced in the testbed and enabled in the hardware and software components.

Finally, a set of measurements have been performed in an anechoic chamber in order to provide a device antenna characterization. The objective is to be able to replay such impairment in the testbed as part of the test, in order to overcome the real antenna bypass due to the cable-conducted test choice.



2 RAN Emulation Capabilities

2.1 NB-IoT Cat. NB1

Introduction

The Internet of Things (IoT) refers to the paradigm of physical and virtual “things” that communicate and collaborate over the Internet, with or without human intervention. The spectra of things that may be connected within IoT, ranges from complex machines, such as planes or cars, to everyday appliances, such as consumer refrigerators, and very simple devices like humidity sensors. The emphasis of the IoT is on the services, which represent the primary driver for interconnecting things. Examples of IoT services include: micro-climate monitoring of homes, asset tracking during transportation, or, on a larger scale, controlling the power consumption of all the refrigerators in a country depending on the load, etc.

TRIANGLE, focusing on the evolution towards 5G networks, considers IoT as a unique opportunity for learning and testing new requirements. For example, IoT shares some of the expected 5G features such as reliability, transmission and reception of small data packets from large number of devices. The focus of TRIANGLE is on licensed spectrum solutions, where it is expected that developers will expend more efforts optimizing their solutions due to the need for conformance and interoperability testing. Among all these wireless systems, NB-IoT is the leading solution.

NB-IoT Basics

NB-IoT physical layer is LTE based with a total bandwidth of 180 kHz. NB-IoT supports three modes of operation: standalone, where it can be used to replace legacy GSM carriers; guard-band, where it can be deployed in LTE carrier's guard bands; and in-band, where thanks to its similarities to the LTE physical layer can be deployed in resource blocks within the LTE carrier.

NB-IoT architecture is very similar to LTE, also organized in frames, where a frame is also composed by 10 subframes, each with a duration of 1ms. NB-IoT defines the concept of hyper frames, which counts the number of system frames periods, i.e. one hyper frame corresponds to 1024 system frames (approximately 3 hours). Downlink channels present in NB-IoT are like LTE:

- Narrowband primary synchronization signal (NPSS)
- Narrowband secondary synchronization signal (NSSS)
- Narrowband physical broadcast channel (NPBCH)
- Narrowband reference signal (NRS)
- Narrowband physical downlink control channel (NPDCCH)
- Narrowband physical downlink shared channel (NPDSCH)

The main difference between NB-IoT and LTE is that NB-IoT needs to time multiplex downlink physical channels and signals; while in LTE most of these channels are multiplexed in frequency and time. For example, in each frame only one channel is present: NPSS, NSSS, NPBCH, NPDCCH or NPDSCH.

In the uplink, NB-IoT defines two channels: narrowband physical random access channel (NPRACH) and narrowband physical uplink shared channel (NPUSCH). A completely new random access channel was required for NB-IoT as PRACH in LTE requires 1.4 MHz.



NPUSCH has two formats. Format 1 is used for uplink data and Format 2 for signalling HARQ acknowledgement for NPDSCH.

The access reservation procedure to initiate data transmission is similar to LTE. The device transmits one of the available preamble sequences in the NPRACH. Upon detection of the preamble the eNodeB will send a random-access response (RAR) which among other information includes an uplink grant for transmitting the RRC Connection request. The rest of procedures are almost identical to LTE with the main difference that data exchange with the eNodeB is done at RRC level, where the data is piggybacked in the RRC Connection Setup in the downlink and in the RRC connection setup complete in the uplink.

NB-IoT Extension

It was clear, already from the beginning of the project that IoT was playing an important role on the path to 5G. The state of the art and the scenarios defined in WP2 have confirmed such trend. For this reason, a decision was taken to extend the UXM with pre-standard features related to the Cat. M2 category introduced for the Rel'13 of LTE-A.

Despite the numerous similarities between LTE and NB-IoT, there is still a considerable effort required to extend the capabilities of the Keysight E7515 UXM Wireless Test Set to support NB-IoT. The features span from the adjusted characteristics of MAC and physical layer, to the introduction of the new RRC and data transmission procedures. Obviously, the implementation of the entire protocol required too much effort on TRIANGLE. Instead, TRIANGLE helped to jump-start NB-IoT within Keysight by developing the basic framework that will be later extended by Keysight internal resources. Most of the effort focused on L3.

- Layer 3 Modifications
 - o Definition of system information blocks (SIBs)
 - MIB-NB
 - SIBType1-NB
 - SIBType2-NB
 - SIBType3-NB
 - SIBType4-NB
 - SIBType5-NB
 - SIBType14-NB
 - SIBType16-NB
 - o Upgrade RRC states for NB-IoT
 - o Upgrade RRC procedures
 - o Upgrade RRC encoder/decoders for Rel-13 supporting NB-IoT messages.
 - o Security procedures

In addition, due to the lack of NB-IoT devices we also needed to implement a TAP plugin which allowed us to simulate the connection of a device to test the ongoing development. This plugin can attach to L3 interfaces to simulate all the communication with L2 if a real device was connected. This was useful to verify the effective functionality of the developed code. While such plugin is not part of the testbed, it was an essential development tool.

From this initial work the UXM NB-IoT capabilities have been extended to support the following features:



- Data over NAS (DoNAS)
- Power Saving Mode (PSM)
- Connected and IDLE mode for Extended DRX (eDRX)
- Repetitions
- Coverage Zones

2.2 Antenna Characteristics Emulation

To ensure UE test reproducibility (especially during conformance testing) devices are typically cable connected to the network emulator. This allows perfect control of the radio channel between the eNodeB and the UE but it does not consider the effect of the antennas. The goal of Task 4.4 is to expand the capabilities of the testbed with techniques to emulate the effects of the antenna in the communications channel.

OTA techniques to incorporate the antennas effect

MIMO OTA focuses on integrating the antenna propagation effects of the UE connection to the Base Station. This is an integral part of TRIANGLE ability as test bed to emulate 'real live' scenarios which include all effects and impact of the UE capability.

There are several elements required to fully assess the MIMO device performance:

- Full anechoic chamber
- Radio channel emulator
- Network/Base station emulator (BSE)

There are also different methods to assess MIMO performance, such as:

- MPAC (Multi-Probe Anechoic Chamber): Approved by CTIA (see section 1.8.1 in [1])
- 2-Stage Method: Developed by Keysight. This solution makes use of the Keysight E7515A UXM Wireless Test Set (see section 6.3.1.3 in [2])
- Reverberant chamber

MIMO OTA is a hot topic in standardization. Several methods exist and compete against each other. They are very different in approach and associated costs. The TRIANGLE project has been following the standardization effort and testing some possible approaches.

Work Plan

TRIANGLE partners have designed a work plan to achieve goals in Task 4.4 which basically consists on the integration of the 2-stage method into the BSE and expand the UXM capabilities. This work plan includes three phases.

Phase 1: Test Campaign using MPAC system

The MPAC system is the type of system accepted by CTIA for MIMO OTA performance evaluation for LTE 2x2 UEs. DEKRA, as CTIA Authorized Laboratory (CATL), has one MPAC system the TRIANGLE consortium can take advantage of. The consortium agreed to run a test campaign using DEKRA's MPAC system to have a reference for comparison once the TRIANGLE testbed is updated with the 2-stage method.

Phase 2: Integration of the 2-stage method testbed

This phase will deal with the integration of the 2-stage methodology in the BSE already available in DEKRA's MPAC system as well as with the expansion of the UXM capabilities to support operation with this BSE.

Phase 3: Test campaign using TRIANGLE 2-stage method testbed

This will be the final phase of the work plan and it will validate the TRIANGLE testbed 2-stage methodology capabilities. Test results from Phase 1 using DEKRA's MPAC system will be used as reference for comparison.

Phase 1

Phase 1 started with the execution of a test campaign using smartphones provided by KEYD, which were previously tested in other laboratories using MPAC and 2-stage method systems as part of the MIMO performance assessment research activities within 3GPP. This test campaign was run using the MPAC system available at DEKRA.

Tests performed followed the test methodology defined in 3GPP TR 37.977 sub-clause 12.1 [2] to assess the MIMO reference sensitivity of the devices. The output of these tests is throughput versus downlink signal level curves for the Urban Macrocell channel model (UMa) also defined in [2] and for a number of LTE bands. The downlink signal level is expressed in terms of RS-EPRE (dBm/15 kHz) where RS stands for Reference Signals and EPRE stands for Energy Per Resource Element. For each azimuthal position, the signal level RS-EPRE that would result in 95%, 90% and 70% of the maximum throughput is determined. Azimuthal positions under test cover 0° to 330° in 30° steps. The diagram of the MPAC system is illustrated in the following figure.

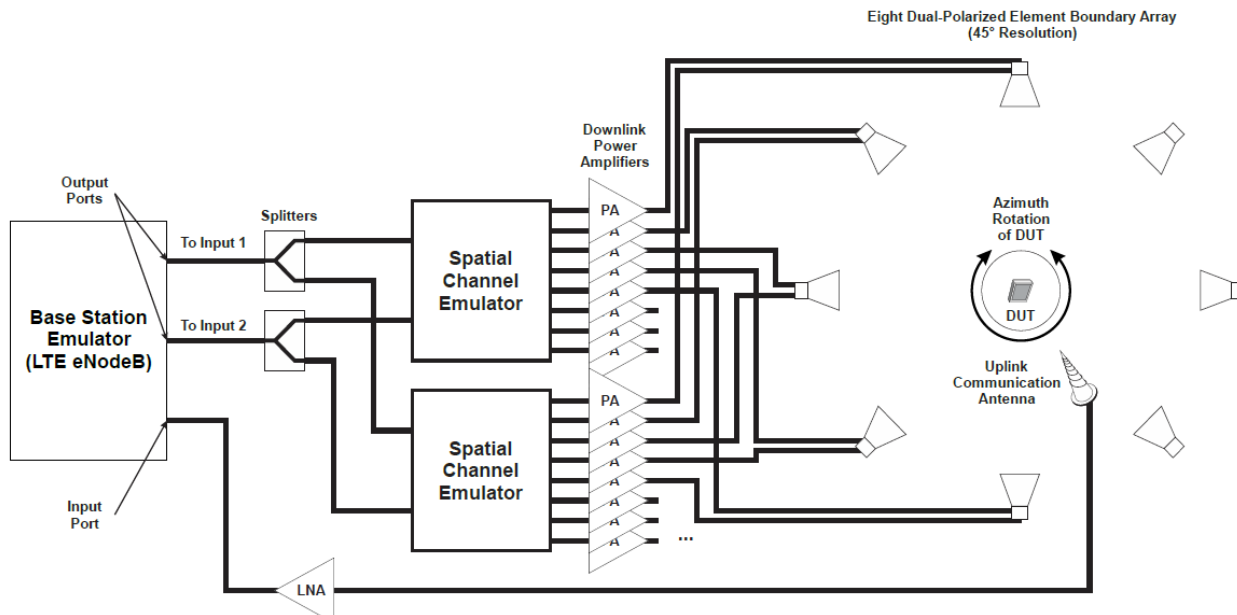


Figure 1 - Typical Downlink MIMO MPAC system

Graphs below are an example of typical Throughput vs RS-EPRE curves obtained by performing these tests.

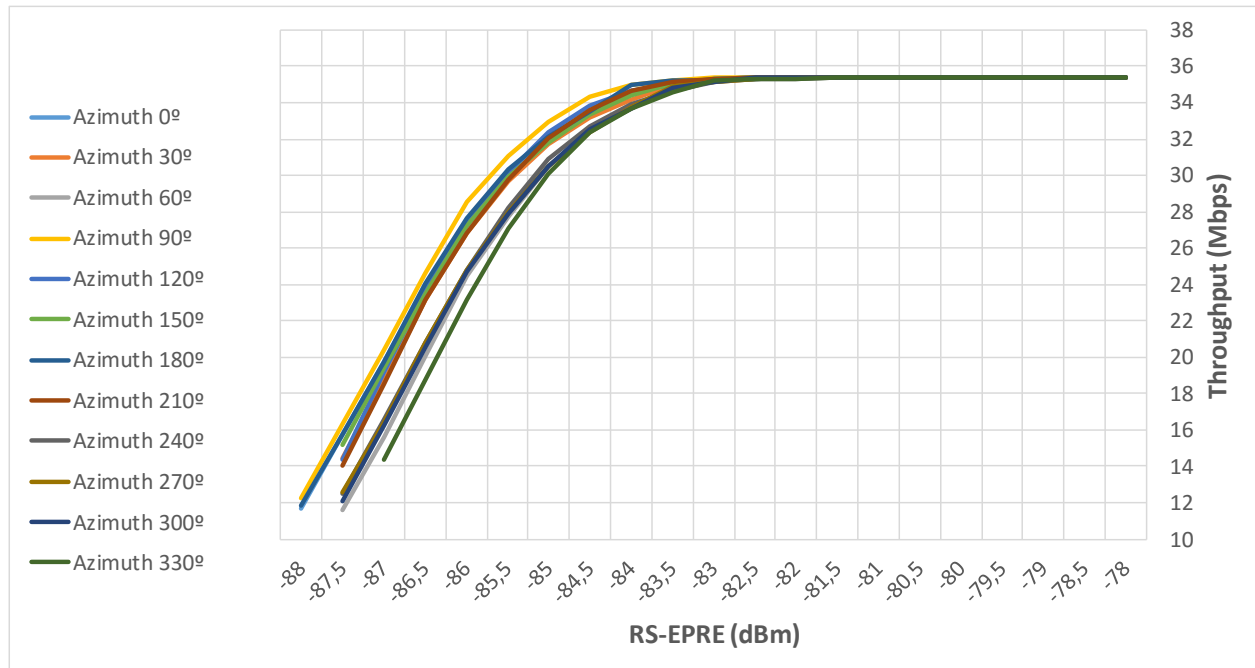


Figure 2 - Throughput vs RS-EPRE curves

Tests covered in test campaign included two (2) different smartphones, two (2) different LTE bands (13 and 7) and three (3) different UE orientations inside the chamber: L0 (Landscape – 0° tilt), P0 (Portrait – 0° tilt) and Face Down.

Results were disseminated on 3GPP at RAN WG4 Meeting #81 in contributions R4-1609332 [3] and R4-1609333 [4]. These contributions presented an inter-comparison study among different laboratories using MPAC systems (lab names kept anonymous to ensure confidentiality). Results led to conclude there were deviations for bands around 700 MHz among the different laboratories. The following figures show the MPAC results for UMa with AAD_1 (device 1) for the P0 70%, L0 70% and L0 95%. The abscissa axis is the azimuth position in degrees. The TRIANGLE laboratory is named “Lab 3” in the study.

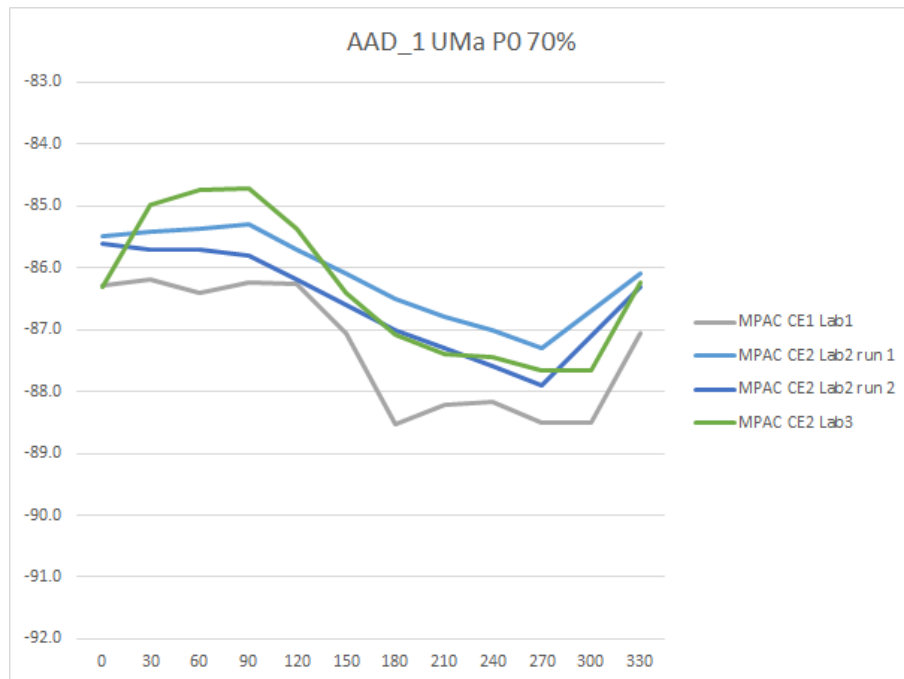


Figure 3 - Results for AAD_1 (device 1) P0 with UMa @ 70% outage level

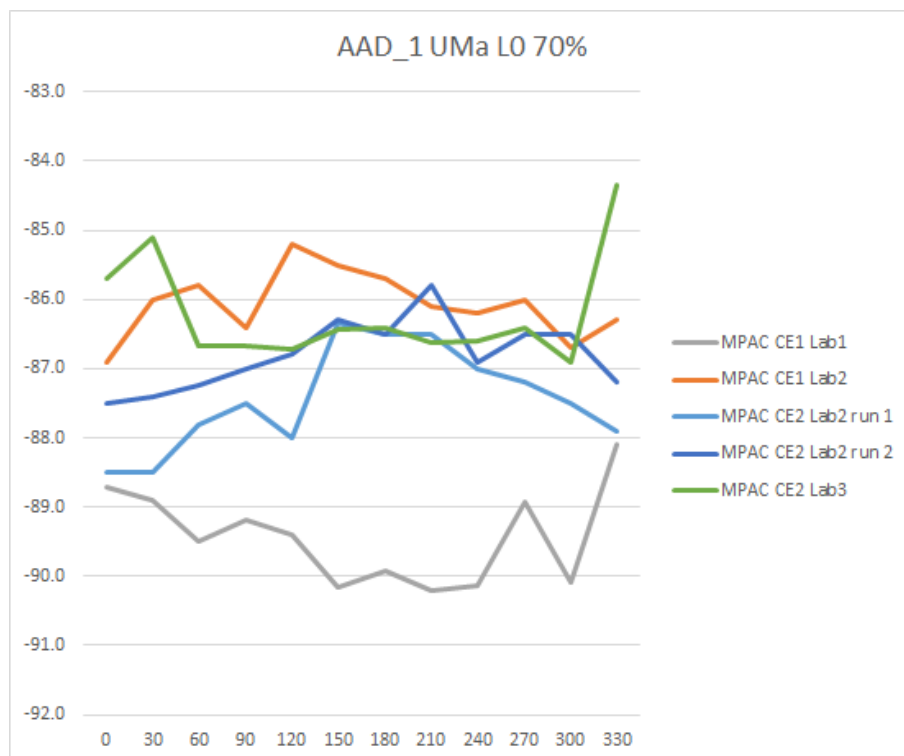


Figure 4 - Results for AAD_1 (device 1) P0 with UMa @ 70% outage level

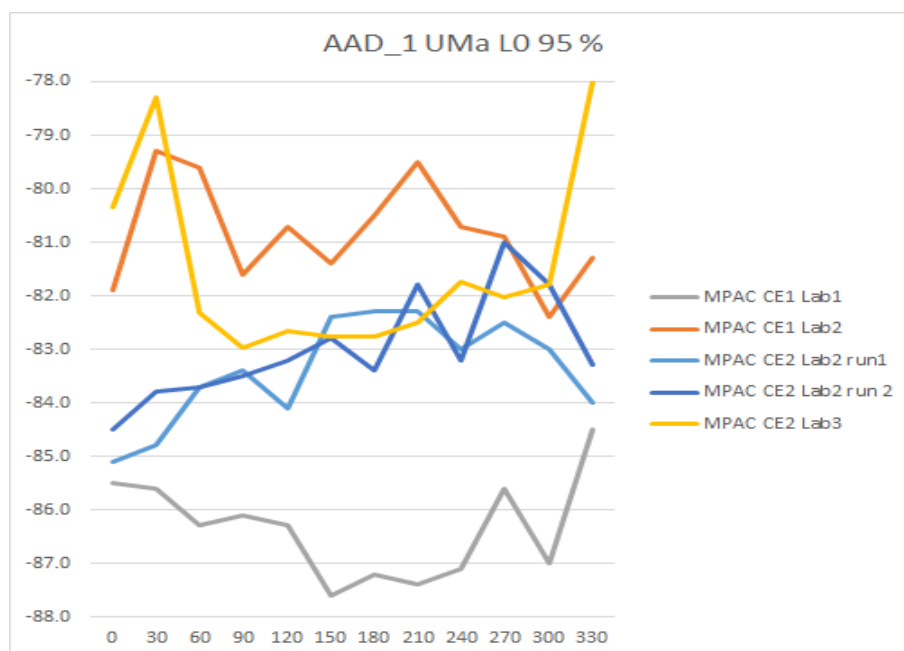


Figure 5 - Results for AAD_1 (device 1) L0 with UMa @ 95% outage level

Allowing for reasonable arbitrary shifts between systems due to different calibration references, orientation specific differences across systems using three different devices range from 3.1 dB to 4.5 dB. Root cause analysis was disseminated in CTIA document MOSG161104 [5]. This document discussed possible reasons for the differences across systems found in the abovementioned analysis including: channel model differences, system calibration issues, device instability (heating effects and uplink power levels, bistable throughput limits due to battery levels, instability near the device failure point, cyclic variations in throughput levels, operational errors). As a conclusion Change Requests (CRs) were proposed to improve the robustness of the test and validation procedures in R4-1609337 [6], R4-1609336 [7] and R4-1610320 [8].

DEKRA has been revising the system calibration together with KEYD and the assistance of the system vendor since then in order to find the reasons behind the deviations found in the abovementioned study. DEKRA has thoroughly repeated all calibration steps with special care put on 700 MHz bands. Samples used for the test campaign had to be delivered again to KEYD because tests within 3GPP had to continue. From that point DEKRA has continued investigating the issues on 700 MHz by testing other devices, still with the assistance of KEYD and the system vendor. During that point DEKRA was audited by CTIA to become CATL (CTIA authorized laboratory) with success. At this point DEKRA is ready to start a new test campaign together with KEYD and 3GPP to understand if deviations have been solved.

Phase 2 and Phase 3 have not been executed.

2.3 Introduction to LWIP

With the growth in the use of data that cellular networks have experimented in the last years, and the forecasts pointing to the fact that such a tendency will continue and even strengthen in the future, network capacity is becoming an important issue to solve. Different strategies have been approached to face this situation, as for example the use of femtocells or

developments to improve transmission. A very interesting approach is the complimentary use of unlicensed spectrum. This can be achieved deploying LTE in unlicensed bands, as those Wi-Fi use. Two technologies use this method of improving capacity, named LTE-U (LTE-Unlicensed) and LAA (Licensed-Assisted Access). A different way of achieving similar benefits is the called LTE-WLAN interworking, consisting in the use of Wi-Fi radio access network to reach LTE core network, besides the usual LTE access. Nowadays two technologies implement this LTE-WLAN interworking: LWA (LTE-WLAN Aggregation) and LWIP (LTE-WLAN RAN level integration using IPSec tunnelling).

LWIP has been chosen as the most appropriate technology to develop and implement due to its advantages in comparison with the other LTE-Wi-Fi interworking solutions. Unlike LWA, LWIP allows to use the alternative WLAN access to route data for both downlink and uplink. Besides that, LWIP can be implemented making use of the legacy Wi-Fi network, as it does not need any modifications or enhancements to the Wi-Fi AP, while LWA does. As LWIP is the most actual of these technologies, it appeared in the 3GPP's Release 13, from March 2016.

LWIP Standard Architecture

One of the key aspects that makes LWIP interesting is its simple architecture. This simplicity implies that all the changes affect exclusively at the RAN level, so the EPC does not know about the use of LWIP. This architecture also allows the use of legacy WLAN infrastructure with no changes. A simple diagram of LWIP architecture can be seen in Figure 6.

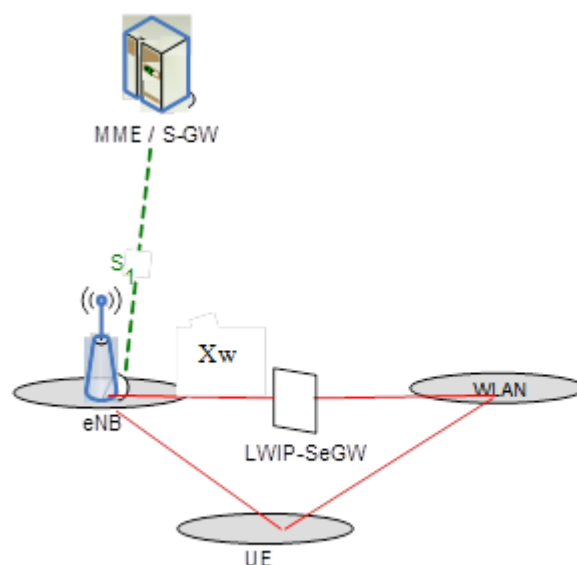
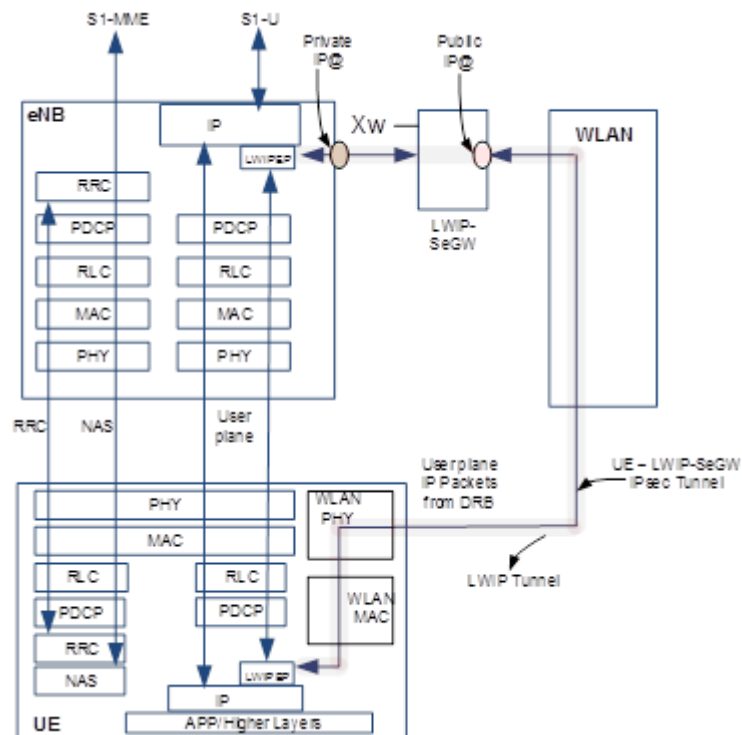


Figure 6 – LWIP overall architecture

All the elements involved in LWIP operation can be seen in the diagram. There is only one new element, LWIP-SeGW, that is connected to eNodeB through a new interface called Xw. This element is related to security, since its function is to be the endpoint of the IPSec tunnel initiated in the UE. This way, the UE does not have direct IP connectivity to eNodeB through WLAN, and the LWIP-SeGW is the interconnection point between WLAN Network and eNodeB. A LWIP-SeGW must connect an UE only with the corresponding interface of the

LWIP is controlled by the eNodeB, in contrast to other technologies that make their operations in the EPC. The eNodeB is the element responsible for activating and deactivating LWIP operation, based on UE measurement reporting, and it can add or eliminate DRBs from LWIP at any time. SRBs use LTE exclusively, and among the different signalling that the eNodeB can send to UE are Mobility sets, bearer information and reconfiguration, or LWIP-SeGW IP address, for example. eNodeB also has to indicate if a DRB used in LWIP includes aggregation, meaning that both LTE and WLAN radio accesses are used simultaneously for that bearer, and hence there is aggregation of two packet flows into a single flow at some point. This can be understood better thanks to Figure 7, where LWIP architecture is represented with the different layers and flows of packets present in this technology and providing much more detail into it.



In the diagram, three user plane flows can be seen: one that uses LTE and does not participate in LWIP, that goes between IP layers of UE and eNodeB directly, another one using LTE access that uses LWIP, since it goes between LWIPEP entity of UE and eNodeB, and a last user plane flow that uses the WLAN access through the IPSec tunnel to reach LWIP-SeGW, and then connects to the LWIPEP entity of eNodeB through Xw interface. Another important point of the diagram is the indication of the IP addresses used to connect to LWIP-SeGW and eNodeB through Xw interface. The first one is a public IP, so the UE can connect to it using WLAN, meanwhile the second is a private IP, and should be reachable only from LWIP-SeGW. The last piece of information that the diagram provides is the

signalling route. For both signalling to and from eNodeB and EPC, only LTE access is used, as it was previously mentioned.

An important issue that LWIP does not solve is reordering. When aggregation is configured for a DRB and both LTE and WLAN accesses are used simultaneously, it is possible that packets do not arrive at destination ordered, due to different throughput or delay between LTE and WLAN. This affects both uplink and downlink. Because of this, in order to allow the use of aggregation, LWIPEP entity or higher layers must implement reordering of LWIP packets. To help on this matter, LWIPEP headers added at the LWIPEP entity showed in Appendix 2: include a field called Sequence number when aggregation is configured. Looking at the sequence number in the headers, the LWIPEP entity or a higher layer could reorder the packets from the two flows belonging to the same DRB. The Figure 8 shows the protocols involved in LWIP WLAN user plane.

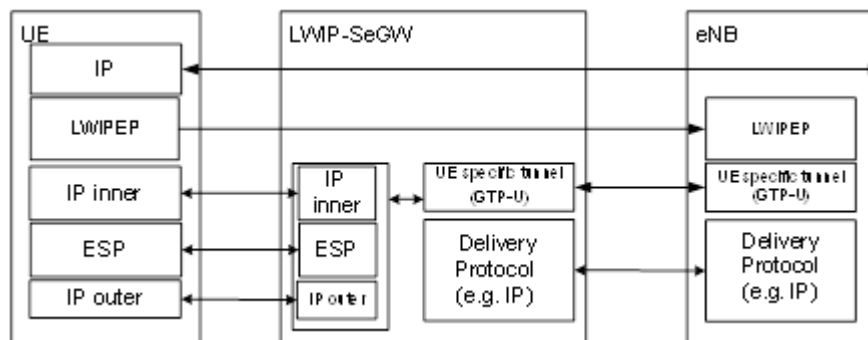


Figure 8 – LWIP protocol stack

Architecture Adaptation to Testbed

The architecture and functions previously detailed must be adapted to our scenario. We will develop the implementation in a UXM, since this equipment allows us to test designs for their validation. This way it will be possible to evaluate the different signaling, interfaces and functions needed in the eNodeB to make LWIP work as intended.

To explain the architecture adapted to our setup, some diagrams have been outlined to provide a better understanding of the scenario. In these diagrams, one can see the different elements involved in the LWIP functions and their relations and communication flows. The first two diagrams relate to the data plane, while the third represents the data plane, and the last one in this section sums up the complete architecture.

In the first of the diagrams, dedicated to the data plane, we can see the LWIP Entity as the center of the scheme. At the upper part of the diagram, in the direction of the EPC, there is only one element, named “S1-EPC” in this document. Its name makes clear its function: it implements the S1-U interface to connect the UXM with an external EPC and provides the corresponding interface to transmit and receive the different data flows from the core network. Continuing downwards, we find L1/L2 layers. Our LWIP Entity will communicate with the corresponding DRB port of this layers, being this the way that the data will use to reach the UE through the LTE radio access. This L1/L2 block represented in the diagram includes the way through all its layers, which are PDCP, RLC, MAC and PHY. As it has been said, UE’s LTE interface is reached from the L1/L2 block.

The rest of the diagram represents the WLAN access, whose ends are the UE’s Wi-Fi interface and the LWIP Entity in the eNodeB. The LWIP Entity implements an interface called

Xw to communicate with the LWIP-SeGW, in order to control UE connectivity and provide extra security. Through this interface, GTP-U tunnels are mounted for the different connections when LWIP is used. LWIP-SeGW is the second element included in our development, and it connects with the UE's WLAN interface through an IPSec tunnel. This gateway is in charge of managing the tunnel establishment, making the communication transparent to the WLAN AP. There must be just one IPSec tunnel per UE, for both uplink and downlink.

The type of packets transmitted through the Xw interface is LWIPEP PDUs. As said in the previous section, those packets include a LWIPEP header and a LWIPEP SDU. If aggregation is enabled, packets sent through the LTE access must be LWIPEP PDUs too, to provide the capability of reordering at the UE. Although it is not present in the diagram for not being important for our development, the UEs shall have a LWIP Entity similar to the one present in the eNodeB.

An interesting detail is that the Xw interface, and hence the GTP-U tunnels, could be implemented in the S1-EPC element instead of the LWIP Entity as the standard says. It would be interesting because in the current scenario S1-EPC provides GTP tunnel implementation, and so we would save reimplementing it in a new element. Anyway, the decision of implementing it in the LWIP Entity will make the element more versatile and usable for different scenarios and architectures, so the work in the reimplementation will be worth it.

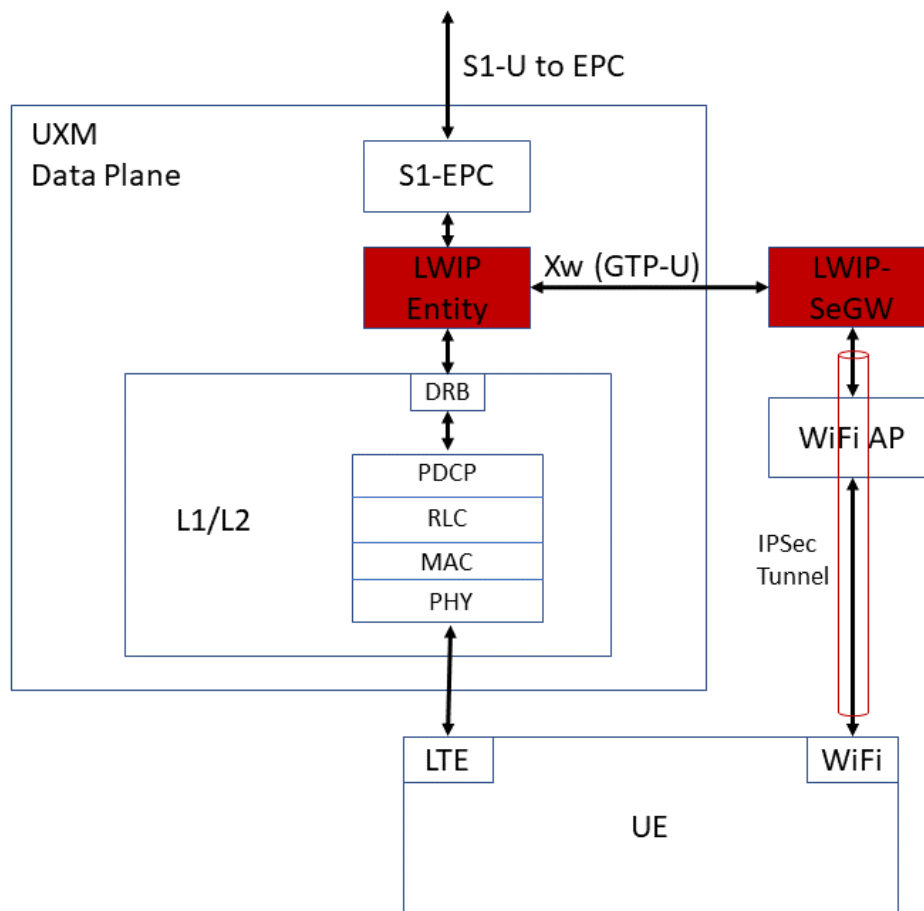


Figure 9 – Adapted LWIP data plane architecture

For the specific case of the testbed, the separation between LWIP Entity and LWIP-SeGW is not necessary, and hence Xw interface would not be needed, simplifying the implementation. This modification can be seen in the next figure. Although both components appear together, a separation must be stated, at least distinguishing both elements functions. This version will be the one that we will implement in the first place.

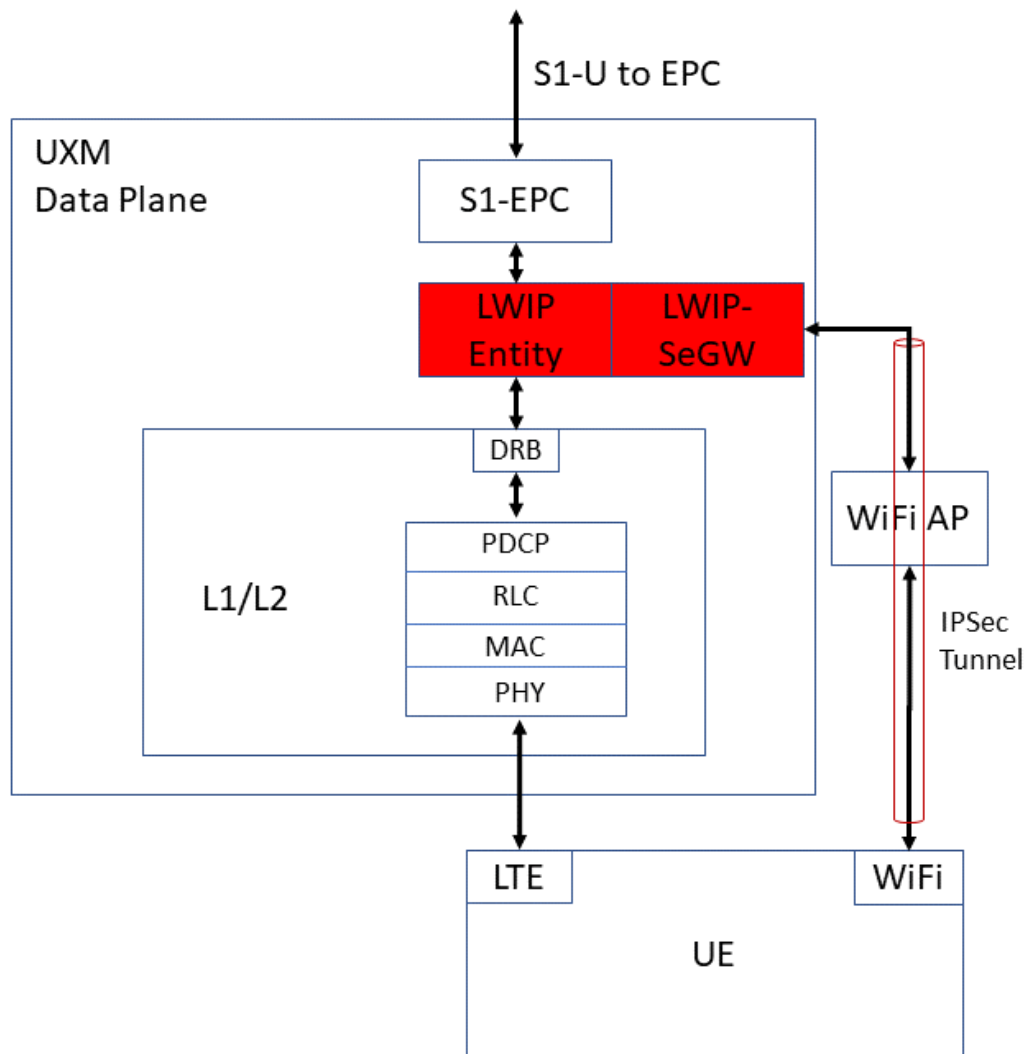


Figure 10 - LWIP data plane architecture optimized for testbed

The next figure shows the elements and relations for the control plane, which transport and process all the signaling messages. In this case a new element adds to the LWIP Entity and LWIP-SeGW as the most important elements: the block named NAS. It represents the NAS layer of the eNodeB, responsible for duties such as mobility management or attach and detach operations. In our scenario, this NAS block is an anchor point for the LWIP signaling, since every control message will travel through it. Lower layers of the eNodeB, the LWIP Entity, and an element labeled “S1 Library” are connected to it.



This “S1 Library”, explained in 3.1 S1 Interface, is an element equivalent to the “S1-EPC” of the previous diagram. It implements the S1-MME interface to provide communication of all the signaling messages to an external EPC, specifically an external MME.

If we continue with the lower layers of the UXM, we find the RRC block, that will communicate with its superior NAS layer, and with the lower L1/L2 layers. The communication with the lower layers is made through the corresponding SRB associated with the UE. This layer is very important for LWIP, as it is in charge of the control of radio resources, and hence it controls the establishment and release of radio bearers, for example. Most of the LWIP operations, as LWIP tunnel and bearers establishment and release, or configuration of UE WLAN measurements, use RRC messages to communicate and configure the UE, as defined in TS 36.300 [1]. After this layer, the signaling is routed through L1/L2 layers to the UE’s LTE interface. It is important to remark that the control plane remains in the LTE access, meaning that no SRB can be configured for LWIP use. All the signaling messages will reach the UE through the LTE access at any moment.

Although all the signaling goes to the UE through the LTE access, there is some control messages that need to reach the LWIP-SeGW: those related to IPsec. The eNodeB may request the LWIP-SeGW to prepare tunnel resources for LWIP operation for a certain UE, and can demand the establishment and release of those IPsec tunnels when needed. In order to do that, the LWIP-SeGW needs some information that must be provided by the eNodeB, as the Initiator ID that the UE will use in the IKEv2 handshake of the IPsec establishment, or the LWIP-PSK key generated in the eNodeB. All this signaling procedures related to the LWIP start and release and the IPsec establishment uses the Xw interface and the XwAP protocol, by means of what is known as XwAP Elementary Procedures.

All that IPsec information is provided to the LWIP-SeGW specifically by the LWIP Entity. Aside from that, the LWIP Entity must ask for control information to the NAS and RRC layers of the eNodeB, in order to know the relation between DRB’s and GTP-U tunnels. In the end the LWIP Entity is the element that must coordinate the LWIP start and release, and ask or deliver all the information needed to achieve it.

As happened with the data plane, in this case there are also some optimizations interesting for our testbed. Apart from joining the LWIP Entity and LWIP-SeGW together, the control plane management should change too. Modifying the control layers as NAS to interact with the LWIP would suppose complicated and unnecessary work. Instead, TAP will be used to interact with all the elements related with the control plane, and all the control management will be performed using it. TAP would be, in certain way, a manager or anchor point for the signaling and control procedures.

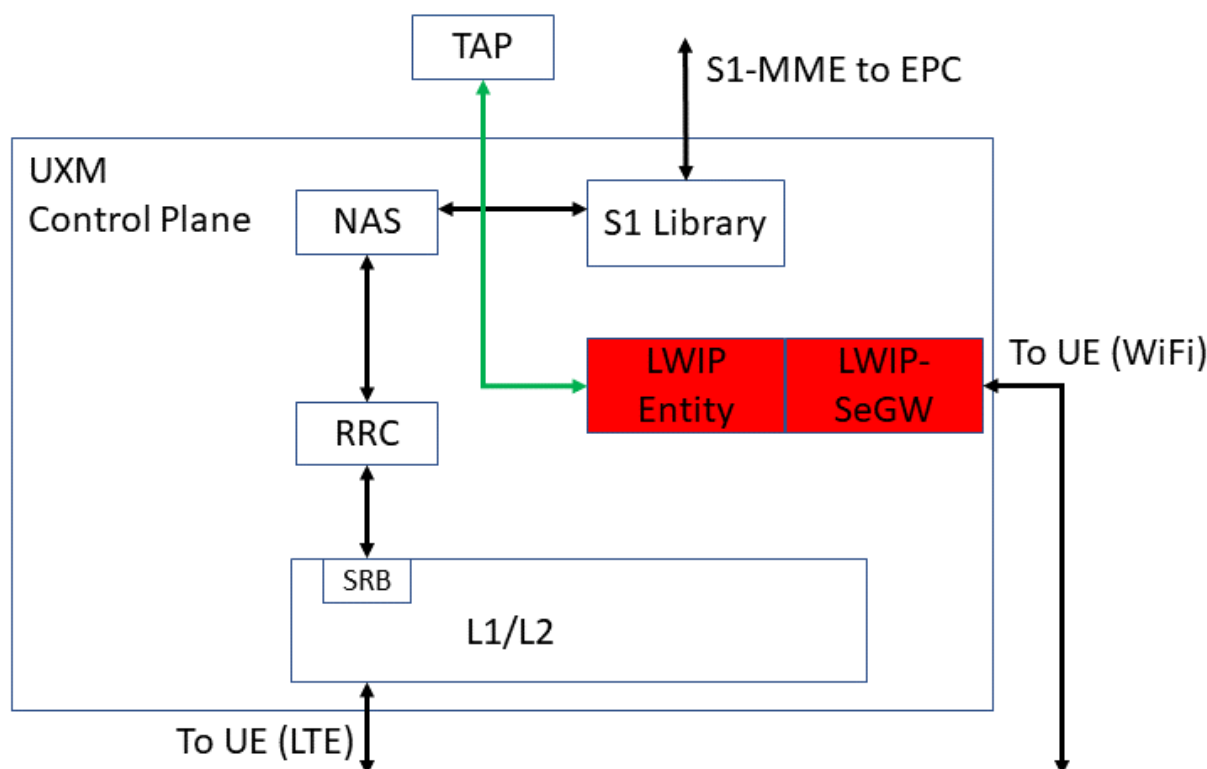


Figure 11 - LWIP control plane architecture optimized for testbed

The last figure related to the specific architecture that will be developed goes deeper in the concepts of LWIP Entity and LWIP-SeGW, the two elements that will be implemented. These elements have been divided in blocks, in a first attempt to organize the different functions that will be needed to implement. The Data plane and the Control plane can be differentiated by the color of the arrows that indicate the communications between blocks.

There are various elements related to both Data and Control plane, whose function is to receive and transmit messages from different interfaces or layers. Those communication blocks allow connection with NAS, S1-EPC, L1/L2 DRB, the UE through IPsec, and between LWIP Entity and LWIP-SeGW through Xw.

The LWIPEP Process block is the only one that receives data packets that is not part of the previously described. Its function is to add or remove the LWIPEP header when needed, depending on the DRB or GTP tunnel source or destination of the packet. Besides that, it must route the packets accordingly to the LTE or Wi-Fi access.

The rest of the blocks are dedicated to the Control plane. There are 3 of these blocks: NAS Control, LWIP Control and IPsec Control. Control blocks functions include managing the IPsec tunnels and the DRBs, as well as storing and providing information needed for the different LWIP procedures.

With the optimizations included in the previous diagrams for our specific testbed, both the LWIP Entity and the LWIP-SeGW architecture outlined would suffer some changes. This final architecture can be seen in Figure 12.

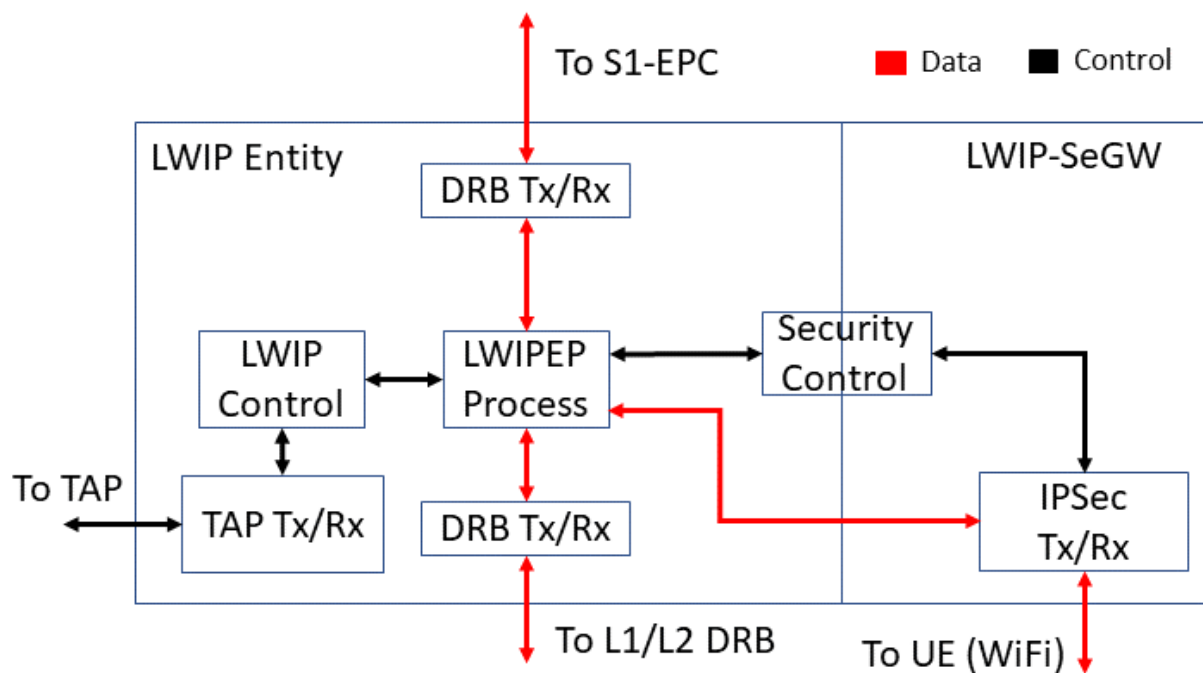


Figure 12 - Detailed architecture optimized for testbed

3 Networking Capabilities

3.1 S1 Interface

Introduction

One of the key elements of the TRIANGLE project testbed is the Keysight E7515A UXM Test Set. The UXM was meant to facilitate functional and RF design of 4.5G and beyond mobile devices. Its manifold features include RF and CA performance testing, maximum data rates up to 1 Gbps1Gbps DL, 100 Mbps UL, or emulation of a wide range of network operations. The UXM emulates all the components of the LTE network, namely E-UTRAN and EPC, providing an end-to-end connectivity to the device under test (UE).



Figure 13 - "Vanilla" UXM Operations

In standard LTE networks, the S1 Interface provides connectivity between the E-UTRAN (eNodeB) and EPC entities (MME and SGW). For control traffic (S1-MME interface), the connection between the eNodeB and the MME utilizes the application signalling protocol S1-AP and the transport protocol SCTP over IP. It is responsible for EPS bearer setup/modify/release procedures, mobility management and NAS transport (direct UE-MME messaging). For the user plane (S1-U interface), data traffic is transported by means of GTP-U tunnels over UDP/IP. A GTP tunnel is established per radio bearer. Every bearer is setup with specific QoS parameters depending on the type of traffic requested.

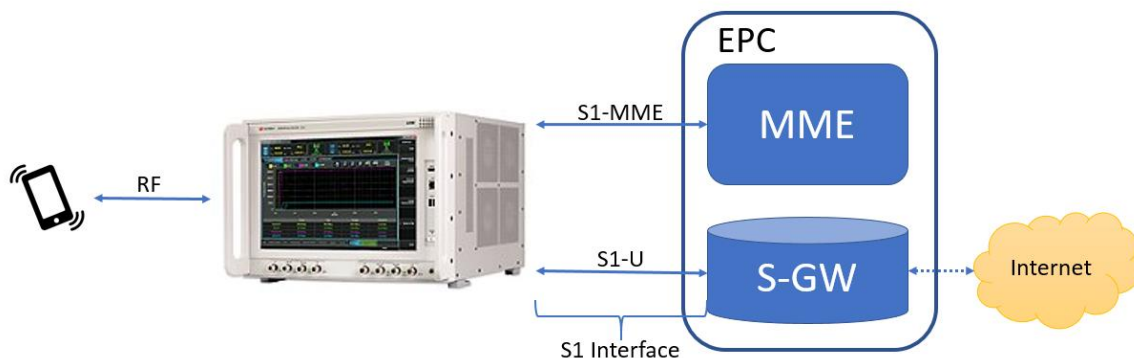


Figure 14 - TRIANGLE Modified UXM Operations

The aim of this extension is to connect the UXM to an external commercial EPC instead of using the internal emulated EPC. Thus, the UXM will perform only eNodeB related tasks. In other words, in TRIANGLE we have added an S1 Interface in the UXM that allows to connect to an EPC. Thus, the testbed users will be able to take advantage not only of the plenty of functionality original in the UXM but also of the new features which involves the S1 Interface extension, for instance monitoring of S1AP messages and user data traffic or the possibility of performing tests using their own EPC (e.g., mobile operators), dramatically boosting the usability and services offered by TRIANGLE.

UXM S1-MME and S1-U Extensions

We modified the UXM's protocol stack to de-couple the internal core network from the rest of the system. In this way, we can either chose to use an external commercial core network or to use the internal core network. The architecture follows the same principals than a real base station, where we added a S1Control block that handles the control plane and a S1User block which encapsulates the data from the user in GTP packets and extract the data from the GTP packets sent to the user by the SGW, see Figure 15.

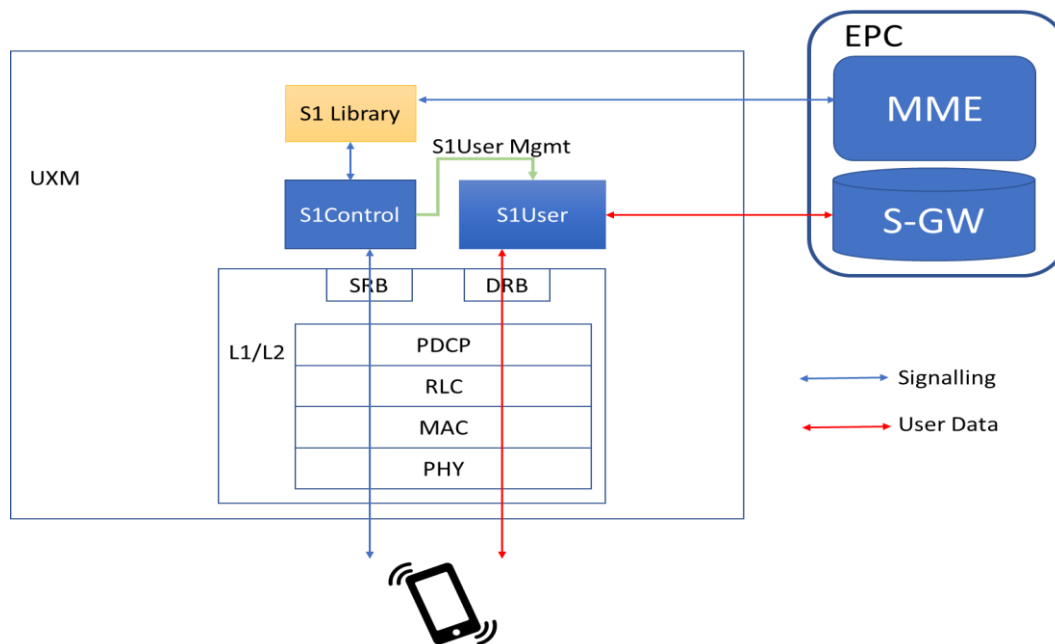


Figure 15 Architecture of the S1 Interface Extension.

The S1-MME (S1AP) is used for exchanging signalling messages between the eNodeB and the MME. The S1AP is defined by 3GPP TS 36.413, which in Release-13 consists on 56 different procedures. These procedures are divided into two different classes: Class 1 & Class 2. Class 1 procedures have an initiating message which can be answered with two possible messages a successful type and a failure type. Class 2 procedures do not require response from the other end. We have implemented all the main procedures that deal with attach process, data transmission, dedicated radio bearers and handovers; which cover all the functionality required by the TRIANGLE testbed. We include a complete list of these procedures:

Supported Class 1 Procedures:

- Handover Preparation
- Handover Resource Allocation
- Handover Cancellation
- E-RAB Setup
- E-RAB Modify
- E-RAB Modification Indication
- E-RAB Release
- Initial Context Setup



- Reset
- S1 Setup
- UE Context Release
- UE Context Modification
- UE Context Modification Indication

Supported Class 2 Procedures:

- Handover Notification
- E-RAB Release Indication
- Paging
- Initial UE Message
- Downlink NAS Transport
- Uplink NAS Transport
- Error Indication
- UE Context Release Request
- UE Capability Info Indication
- eNodeB Status Transfer
- MME Status Transfer

The transport protocol for S1AP messages is SCTP which is sent over IP packets. SCTP is defined in TS 36.412. SCTP was not natively supported by the UXM and therefore it was necessary to add an S1AP library that handles the connection with the MME. The library should also encode and decode the messages before they can be processed at their end. This functionality is also provided by the library. In the next section, we document the different elements of the library and the work done to support the required functionality in TRIANGLE.

We also modified the UXM's protocol stack to add a S1-U interface. This interface is used to send the user data traffic, which is transported by means of GTP-U tunnels over UDP/IP. GTP is defined in 3GP TS 36.414.

3.2 Transport Network Emulation

Architecture

The transport network emulation module is used in the testbed to emulate impairments in any of the Linux machines of the network path and is supported by an impairments plugin SCPI server. The Server is based on the netem module¹, which enables the definition of impairments in a Linux interface. The SCPI Server supports the flexible definition of remote interfaces where the impairments could be configured. An overview of the Server is depicted in the following figure.

¹ <https://wiki.linuxfoundation.org/networking/netem>

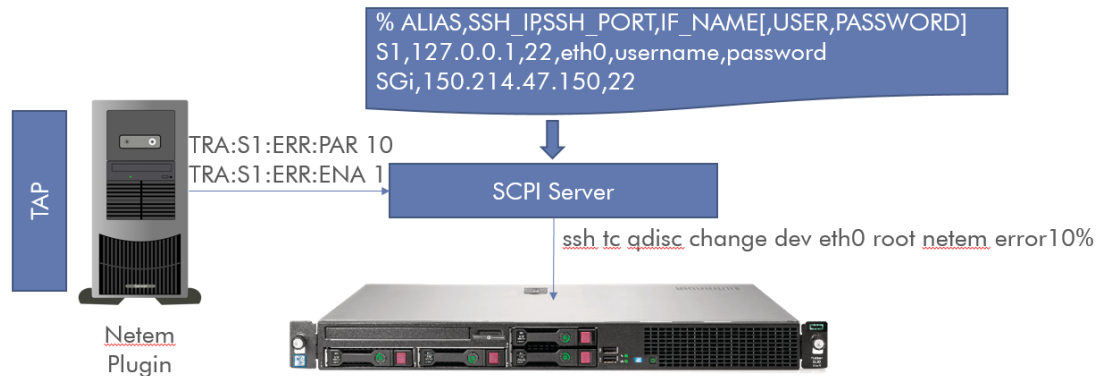


Figure 16: Impairment Plugin Architecture

The server accepts an interface alias, as depicted in the upper block of the figure. The alias is associated with the IP and port that can be accessed using SSH, the interface name and optionally a username and a password for the server (this is not recommended, is better to generate SSH keys between the machines). The set of commands defined by the server accepts an interface alias as an input. Some examples of interfaces are defined in Table 1.

Table 1: Example of alias for the impairment plugin

Alias	Ssh IP:port	Device Name	Comments
S1-MME	8.8.8.8:22001	eth0	S1 control plane interface in the MME
SGi	8.8.8.8:22002	ens32	SGi reference point the PGW
S11	8.8.8.8:22003	lo	S11 interface between the MME and the SGW

Exposed Functionality

The impairments SCPI plugin supports some of the standard SCPI commands (as defined in Table 2) as well as commands to setup different impairments in the interfaces pointed by the alias interface. The server can emulate a latency with a statistical behaviour defined by the user. The commands accepted by the server are defined in Table 3.

Table 2: Standard SCPI Commands of the Impairments SCPI Server

Command	Description
*IDN?	Provides the version of the server
*ERR?	Provide the last error (empty string if none)
*OPC?	Return 1 when the server finishes the last operation
*RST	Remove all the impairments and clean the internal status

Table 3: Latency SCPI Commands of the Impairments SCPI Server

Command	Description
TRA:{ALIAS} :LAT:PAR mean,var,correlation[,distribution]	Set the latency in the specified link alias. The parameters are: <ul style="list-style-type: none"> - mean is the mean latency in ms



	<ul style="list-style-type: none">- var is the variance in ms- correlation is the correlation percentage- distribution is optional and is the statistic distribution which can be: normal, pareto or paretonormal Example: TRA:S1:LAT:PAR 10,2,5,pareto
TRA:{ALIAS}:LAT:PAR?	Return the parameters which are configured in the server (which are not necessarily applied to the server). The order is the same than the one provided in the set method. Example: ->TRA:S1:LAT:PAR? <- 10,2,5,pareto
TRA:{ALIAS}:LAT:ENA enabled	Setting the latency parameters will not be reflected in the interface unless latency is enabled. This parameter set the enabled status (1: enabled or 0: disabled). Example: TRA:SGi:LAT:ENA 1
TRA:{ALIAS}:LAT:ENA?	Return the latency enabled value. Example: ->TRA:SGi:LAT:ENA? <- 1

The SCPI server also enables the definition of link errors in the interface. The smallest possible non-zero value is 0.0000000232%. The command will cause the specified percent of the packets to be randomly dropped.

As specified in the netem documentation a correlation may also be added optionally. This reduces the randomness of the number and can be used to emulate packet burst losses. The commands supported are depicted in Table 4.

Table 4: Link Error SCPI Commands of the Impairments SCPI Server

Command	Description
TRA:{ALIAS} :ERR:PAR error_percentage, error_correlation	Set the error rate in the interface. The parameter accepted is the link error percentage and the correlation percentage. Example: TRA:S1:ERR:PAR 10,2 (set an error rate of 10%)
TRA:{ALIAS}:ERR:PAR?	Return the parameters which are configured in the server (which are not necessarily applied to the server). The order is the same than the one provided in the set method. Example: ->TRA:S1:ERR:PAR? <- 10,1



TRA:{ALIAS}:ERR:ENA enabled	Setting the error rate parameters will not be reflected in the interface unless errors are enabled. This parameter set the enabled status (1: enabled or 0: disabled). Example: TRA:SGi:ERR:ENA 1
TRA:{ALIAS}:ERR:ENA?	Return the errors enabled value. Example: ->TRA:SGi:ERR:ENA? <- 1

Packet duplication can also be defined and provides a way of producing a duplication of random packets. Accepted command are defined in Table 5.

Table 5: Packet Duplication SCPI Commands of the Impairments SCPI Server

Command	Description
TRA:{ALIAS} :DUP:PAR percentage	Set the duplication in the specified link alias. The parameters is the percentage of duplicated packets Example: TRA:S1:DUP:PAR 10
TRA:{ALIAS}: DUP:PAR?	Return the parameters which are configured in the server (which are not necessarily applied to the server). The order is the same than the one provided in the set method. Example: ->TRA:S1: DUP:PAR? <- 10
TRA:{ALIAS}: DUP:ENA enabled	Setting the duplication parameters will not be reflected in the interface unless latency is enabled. This parameter set the enabled status (1: enabled or 0: disabled). Example: TRA:SGi: DUP:ENA 1
TRA:{ALIAS}: DUP:ENA?	Return the latency enabled value. Example: ->TRA:SGi: DUP:ENA? <- 1

Finally, the server also provides commands to configure reordering in the interface. Again, the user can specify the percentage of packets that will be randomly reordered and also optionally a correlation to generate bursts on the interfaces. The commands are defined on Table 6.

Table 6: Packet Reordering SCPI Commands of the Impairments SCPI Server

Command	Description
---------	-------------



TRA:{ALIAS}:REO:PAR percentage,correlation	Set the reordering in the specified link alias. The parameters are: <ul style="list-style-type: none">- percentage is the percentage of reordered packets- correlation is the correlation percentage Example: TRA:S1:REO:PAR 10,2
TRA:{ALIAS}:REO:PAR?	Return the parameters which are configured in the server (which are not necessarily applied to the server). The order is the same than the one provided in the set method. Example: ->TRA:S1:REO:PAR? <- 10,2
TRA:{ALIAS}:REO:ENA enabled	Setting the reorder parameters will not be reflected in the interface unless latency is enabled. This parameter set the enabled status (1: enabled or 0: disabled). Example: TRA:SGi:REO:ENA 1
TRA:{ALIAS}:REO:ENA?	Return the reordering enabled value. Example: ->TRA:SGi:REO:ENA? <- 1

New statistics distributions for the delay based on empirical results can be added to the system. To do so it is necessary to recompile the netem module with ping traces.

3.3 QoS Improvements via SDN Entities

5G is based around slices of network topology with certain characteristics which support a given industry vertical as shown in Figure 17. The link to Quality of Experience in the network arises primarily related to bandwidth and delay characteristics. Quality of Experience related to server and computing processes are considered out of scope as it is a computational issue rather than a network issue.

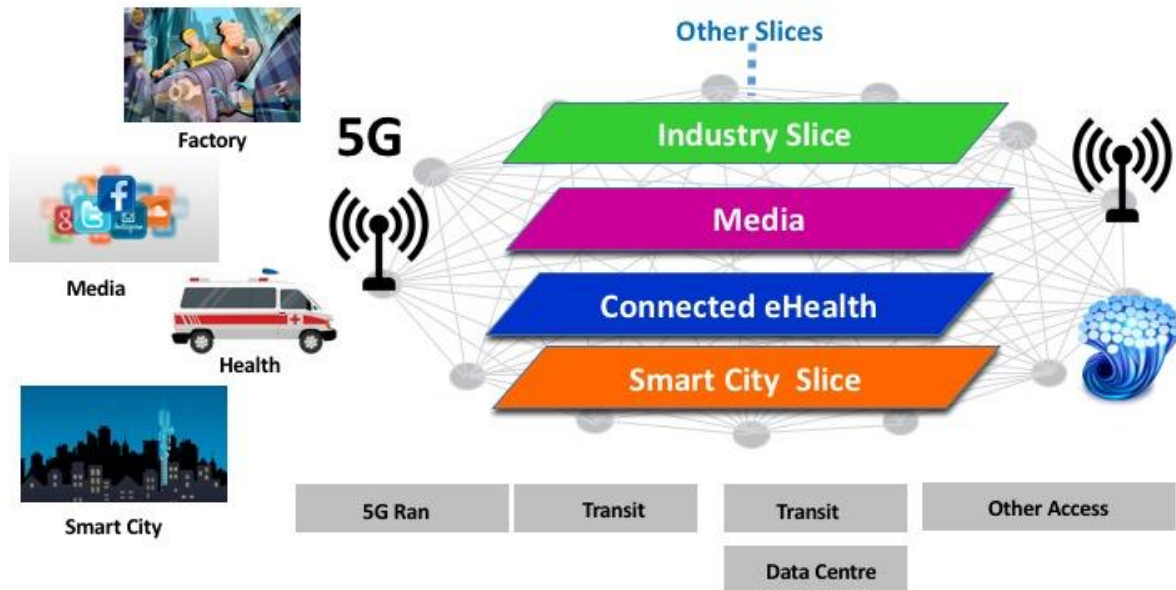


Figure 17 Slicing in 5G

As part of support quality of experience (QoE) in the network we have installed a Virtual Path Slice software defined engine in the testbed in Malaga. Virtual Path slicing allows a quantum of bandwidth to be allocated to a customer event. The software defined network basic scenario is shown in Figure 18. VELOX is the virtual path slice engine. The idea is to deploy a gold/silver/bronze service model which can be used by experimenters to observe different quality of experience levels in their experiment.

The VPS engine can trigger both SDN elements in the network in the testbed as well as radio bearers in the radio network.

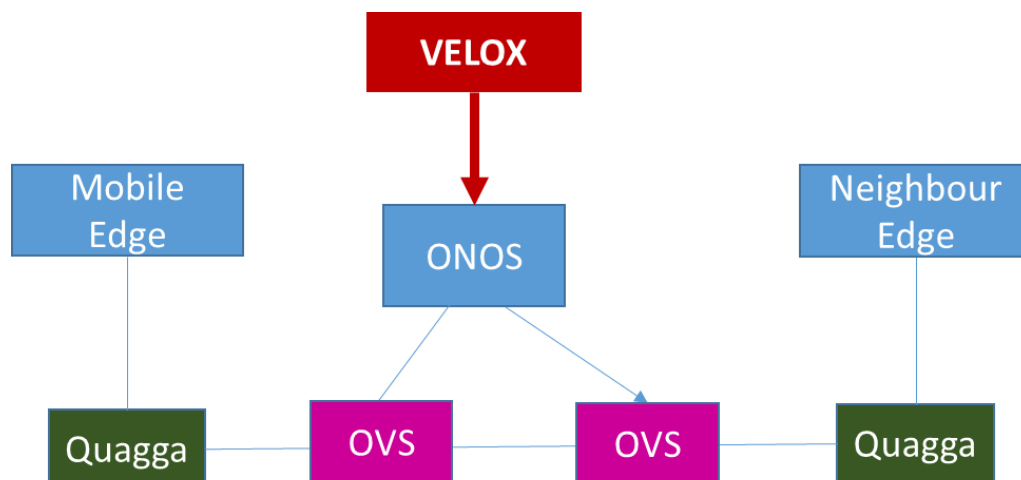


Figure 18 Basic Transport Topology

Deployment details

The deployment was done on Ubuntu server LTS machines and the latest available versions of OVS and ONOS at the time, abstracting transport to a two-node system and an EPC connection for the mobile edge, each part controlled by a different VELOX to support multi-domain bandwidth slices. Quagga was deployed for border router role.

ONOS modifications to enable virtual path slicing

For proper bandwidth slicing in the SDN domain, an extension of ONOS northbound rest API was done to allow for the creation and removal of on demand Queues in the switches via the OVSDB standard. This issue was solved in this manner so that contact with the OVS is always done via the SDN controller rather than directly by VELOX.

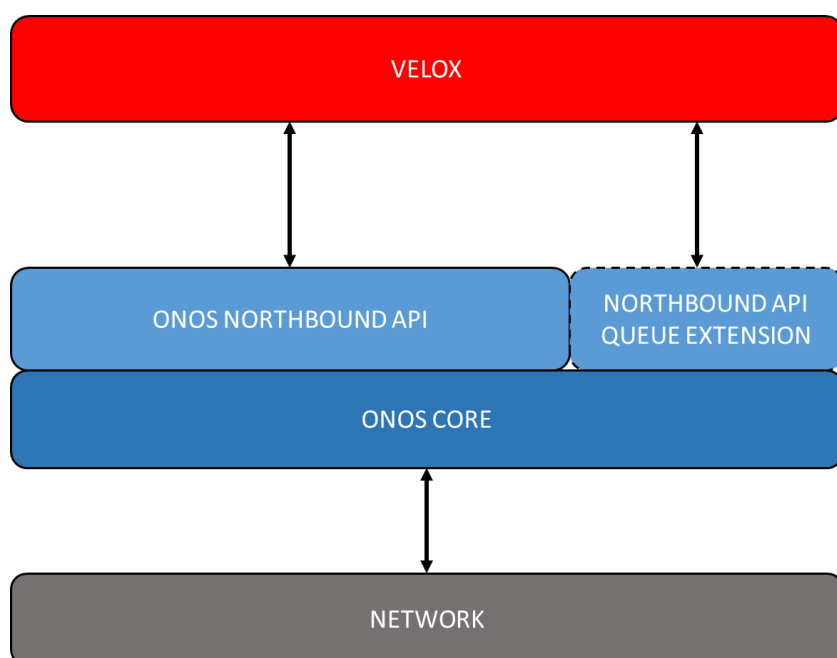


Figure 19 ONOS Northbound API Extension

These queues are rate schedulers which are designed to enforce a target bandwidth for an application. The vertical industry is supported by a topology slice which can be differentiated by several methods (e.g., VLAN ID, DiffservCode points, MPLS EXP bits or BGP VNF address ranges) The model is that the whole SDN topology slice would only allocate bandwidth within in given targets and connection admission control applied across the topology of the slice. As the VPS engine also supports control or radio bearers, by signaling on the Rx interface to the EPC, there is the possibility of end to end control of control of bandwidth allocation which can influence QoE. See Figure 20. Note that it is not possible to be deterministic about radio bandwidth as it depends on power levels and distance from the base station as well as radio factors such as channel interference and fading. For the radio side, it is only possible to provide a target bandwidth with a qualification of 'up to X Mbps' rather than an absolute target of 'X Mbps'.

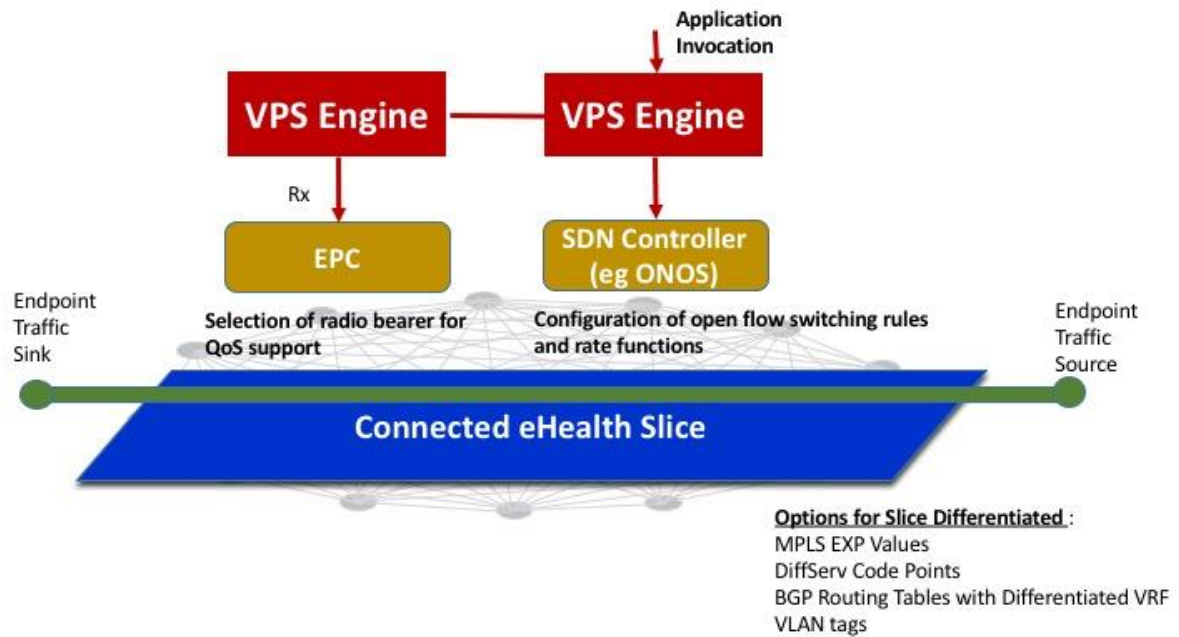


Figure 20 Virtual Path Slice within a topology Slice

4 Technology Enablers

4.1 High precision time synchronization

A common time reference in the testbed is a key enabler to perform cross domain analysis, where measurements in different domains can be observed on the same time axis. An excellent example to illustrate the importance of this analysis is battery consumption; if the energy consumption of a device were to be correlated with protocol messages exchanged with the network, it would allow mobile operators to understand the impact of different protocol operations on energy consumption. It would also enable device makers to test and verify the correct implementation of energy saving mechanisms such as power saving mode or discontinuous transmissions. Finally, App developers could study the impact of the different communication patterns with the remote server.

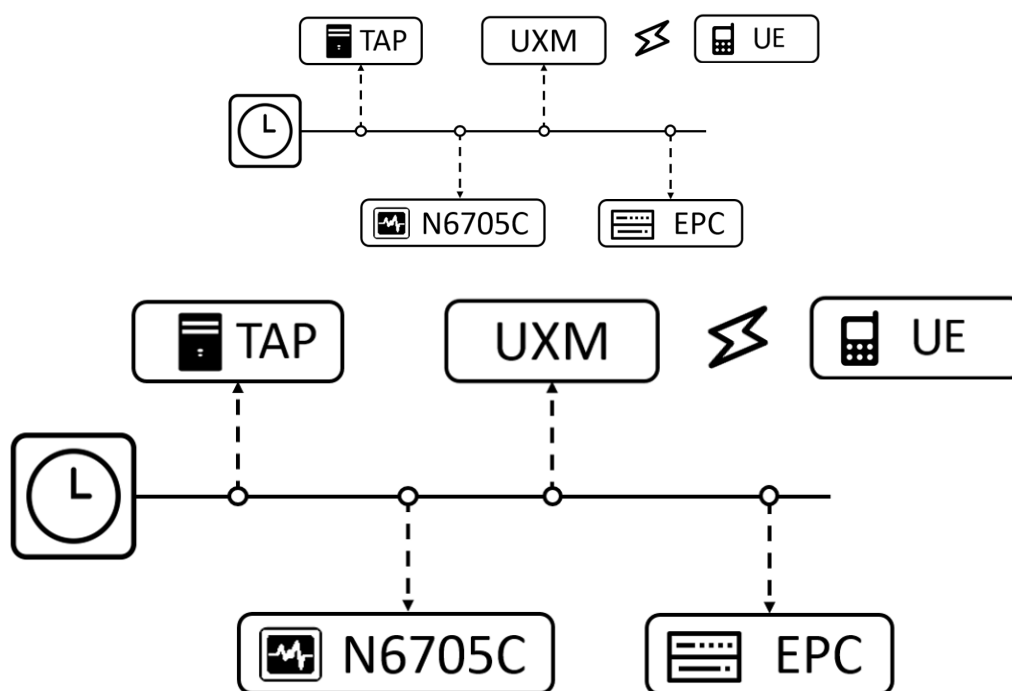


Figure 21 Ideal scenario where all the elements in the testbed share the same clock

TRIANGLE testbed is a complex system where a considerable number of elements are present, see Figure 21. Most of these elements have the capability of taking measurements. Specifically, 5 different system times are coexisting in the testbed:

1. User equipment system time (UE)
2. Windows machine time running TAP (Orchestrator)
3. Power analyser system time (N6705C)
4. UXM system time (Base station emulator)
5. Linux server running the evolved packet core (EPC)



The purpose of this work is to provide high precision time synchronization between different elements in the testbed. To define the precision in time needed, we need to consider the minimum unit of time in the system. In our case, this is given by the Transmission Time Interval (TTI) in LTE. Any unit of information transmitted in the system will take at least one TTI to be transmitted from the UE to the eNodeB. A TTI in LTE corresponds to one millisecond. This requirement implies a synchronization between the different elements in the network in the range of hundreds of microseconds (i.e., one order or magnitude lower). However, not all the elements in the testbed are subject to the same requirement. In fact, requirements of the measurements taken at the application layer such as user interaction or OS events can be relaxed to a few milliseconds.

The protocol chosen for synchronizing the different elements in the testbed is IEEE PTP, also known as IEEE 1588. PTP follows a hierarchical master-slave architecture, where the master clock distributes the clock to the different machines/instruments. There main reason for choosing PTP over other available technologies such as network time protocol (NTP) is that PTP can achieve sub-microsecond accuracy in local area networks (LAN); while NTP precision is around tens of milliseconds in most of the cases.

Aspects to be considered about PTP:

- PTP is susceptible to delays in the packet transmission. Therefore, it is necessary to use a PTP compatible router/switch that will prioritize PTP packets over the rest.
- A PTP clock master node is required. TRIANGLE testbed is using UMA master node.
- PTP requires an Ethernet connection to synchronize the different nodes with the master node. Although using PTP over wireless connections has been investigated, it is not mature enough to be used in the testbed. Therefore, the UE is the only element in the testbed that does not support PTP. Instead, a proprietary synchronization protocol developed by DEKRA is used.
- All the network cards used must support PTP hardware timestamping for optimal performance. Otherwise, the accuracy of PTP is degraded.

The following enhancement were carried out to provide PTP support:

- A new PTP instrument has been added to the UXM. The accuracy exceeds by far the testbed requirement of 1 ms.
- A new PTP instrument has been added to the N6705C. The accuracy exceeds by far the testbed requirement of 1 ms.
- A new PTP plugin has been added to the test automation platform (TAP). This plugin runs in the orchestrator and can supply with PTP time information to the rest of the plugins in the system. Its accuracy depends on the network card used. Results obtained in the worst case (no hardware timestamping) show an average accuracy < 10ms with some outliers of 10-20 ms.

High precision time synchronization results

We performed several tests to verify the proper implementation of the high precision synchronization. The experiment consists of the following. The Device Under Test (DUT) is a LTE device with the battery removed. The DUT's RF port is connected to ana eNodeB emulator (UXM) and is powered by a source/measure unit (N6705c). The UXM and the

N6705c are synchronized using. TAP (KS8400a) is used to collect the measurements from both instruments and intuitively display them.

The experiment results are shown in Figure 22. On the top, the current consumption over time is displayed; while the sequence of messages can be seen in the bottom part. The protocol to be displayed can be filtered per user preferences. It should be noted that it is possible to click on the graph to highlight the corresponding message or to click on a message to highlight the current consumption of the message (see Figure 23).

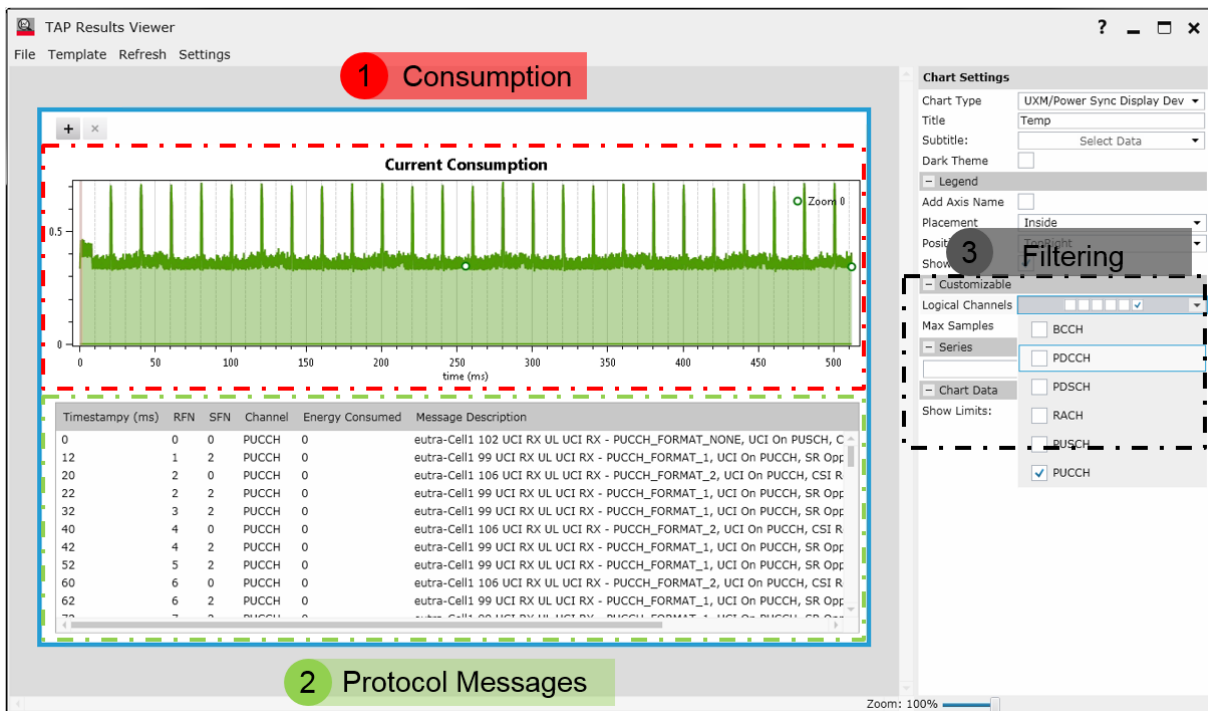


Figure 22 Customized TAP Result Viewer for Current Consumption. Filtering options displayed.

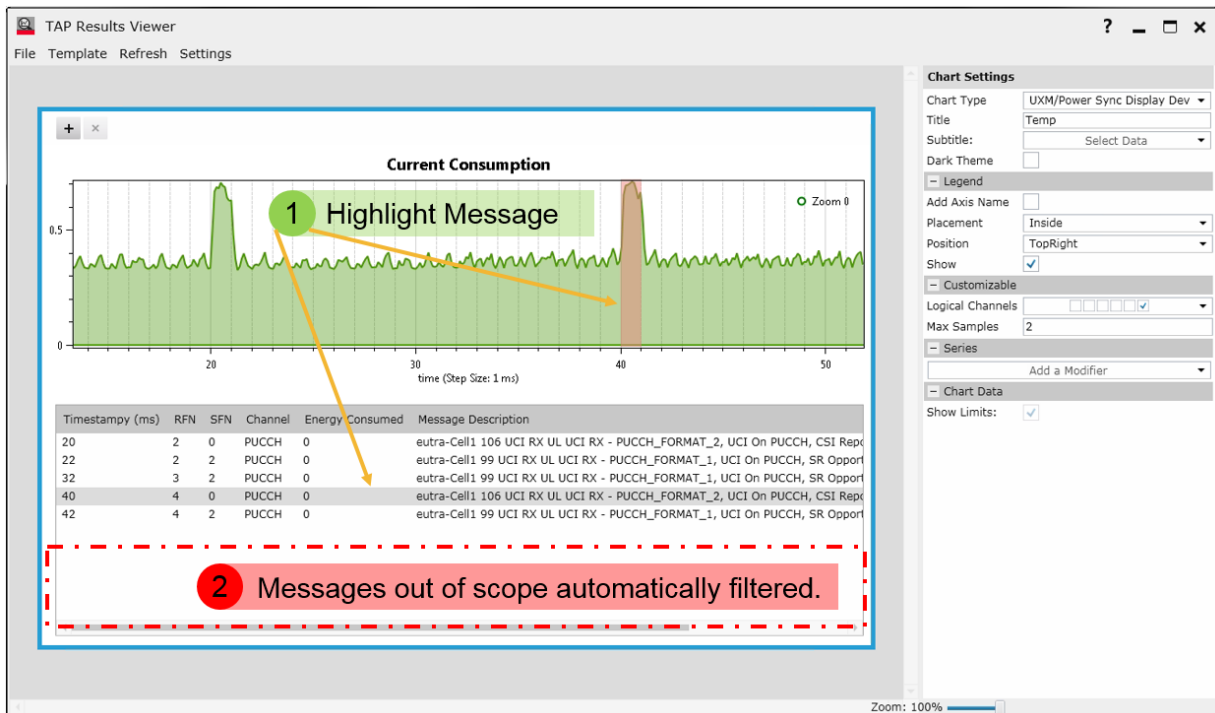


Figure 23 Possibility to click the graph or the message to see the other domain.

This framework allows one to determine the exact energy consumption for every message/data exchanged with the network. For example, the figure illustrates a sudden spike in consumption ($t=20\text{ms}$). The spike is due to a PUCCH request message (i.e., uplink transmission). More specifically a channel state information message sent by the UE to the eNodeB. Therefore, it is possible to estimate the cost of connecting to the network, the cost of transmitting data and the cost of being idle or in energy saving mode. From these measurements, developers can estimate battery lifetime for any given traffic profile.

Finally, we note that the graph starts right at the beginning of a LTE frame and that two spikes are exactly 20ms apart from each other, which is the exact value configured in the UXM. This is a good indication of the correct implementation of the time synchronization instruments.

4.2 Network measurement framework

There are several functionalities that have been introduced in the TRIANGLE testbed to support more measurements.

The main functionality currently available is the remote PCAP server, which is currently deployed in the EPC and enables the capturing of packets from the orchestrator. The packet traces can be used to debug procedures by experimenters and as an input for other measurement elements of the framework.

There are several tools that can be integrated within the ETL framework of the testbed, the main ones are:

- Ping Traces Analyser: it is a tool that can be used to estimate the Round Trip Time of the packets in an interface. The tool support different transport protocols (regular IP, proprietary transports inside the UXM and GTP), which can be useful to have estimates on the latency split on different points on the network.

- FlexIMon: it is a tool that can be used to analyse LTE control plane traces., it was delivered in the FLEX project² and has been extended with new messages. The tool currently generates measurements for the attach, detach, service request and dedicated bearer creation. It can be used to provide insights on the behaviour of the control plane procedures.

4.3 EPC Control

Architecture

The EPC control module is designed to setup the core network of the TRIANGLE testbed as well as to trigger certain procedures which could be use in new scenarios. The EPC SCPI module will read different configuration files and create an EPC instance with the configuration, enabling at the same time the triggering of procedures on real-time.

The EPC control is provided by a SCPI server, which provides functionality to dynamically setup predefined networks and trigger dynamic behaviour on the EPC emulators. The EPC emulators are provided by Polaris Networks and provide standard functionality for the MME, SGW, PGW, PCRF, HSS, ANDSF and ePDG as well as functions to configure and trigger procedures in the EPC. The architecture of the SCPI server is depicted in Figure 24.

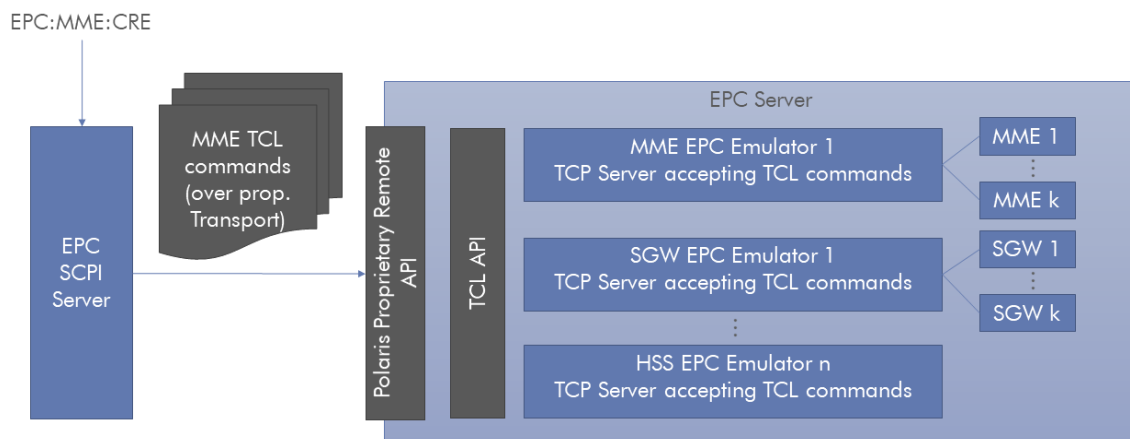


Figure 24: EPC SCPI Server Overview

Figure 24 depicts an overview of the server, which is in charge of translating SCPI commands into the appropriate configuration for the EPC. The communication between the SCPI server and the Polaris Emulators is done employing a proprietary transport interface that will deliver the required TCL commands.

When the SCPI server is connected, it will try to extract the hook of the existing emulators; this hook is the pointer address of the emulator that can be used to send certain commands to the system. The creation of the component is based on txt files that contains the commands that have to be injected into the systems.

Current Functionality

The EPC server supports a set of standard commands; a set of common commands that can be used for any of the components of the core network and specific commands for the MME

² See D5.5 in <http://www.flex-project.eu/project/deliverables>



and PCRF. The standard SCPI commands that are supported by the SCPI server are provided in Table 7.

Table 7: Supported Standard SCPI Commands for the EPC SCPI Server

Command	Description
*IDN?	Provides the version of the server
*ERR?	Provide the last error (empty string if none)
*OPC?	Return 1 when the server finishes the last operation
*RST	Remove all the impairments and clean the internal status

Table 8 provides a list of the commands that are used for all the EPC elements. In the table {EMULATOR} has to be replaced by a valid emulator, currently MME, PCRF, HSS, PGW and SGW are supported.

Table 8: EPC Elements Common Commands for the EPC SCPI Server

Command	Description
EPC:{EMULATOR}:CRE	Creates the default configuration for the emulator. When called, the server will send the commands of the file specified by the configuration. Currently the server support a single set of elements.
EPC:{EMULATOR}:RST	Reset the emulator, this command will remove all the components created in the emulator.

Currently there is a default configuration supported by the server, which is depicted in Figure 25. The S1 interface is exposed to the UXM on the 10.102.81.35 interface, VELOX uses the 10.11.0.35 interface to access to the Rx interface in the PCRF in order to trigger dedicated bearers and the SGI reference point is defined in the 10.10.0.35 interface and NAT with a public IP.

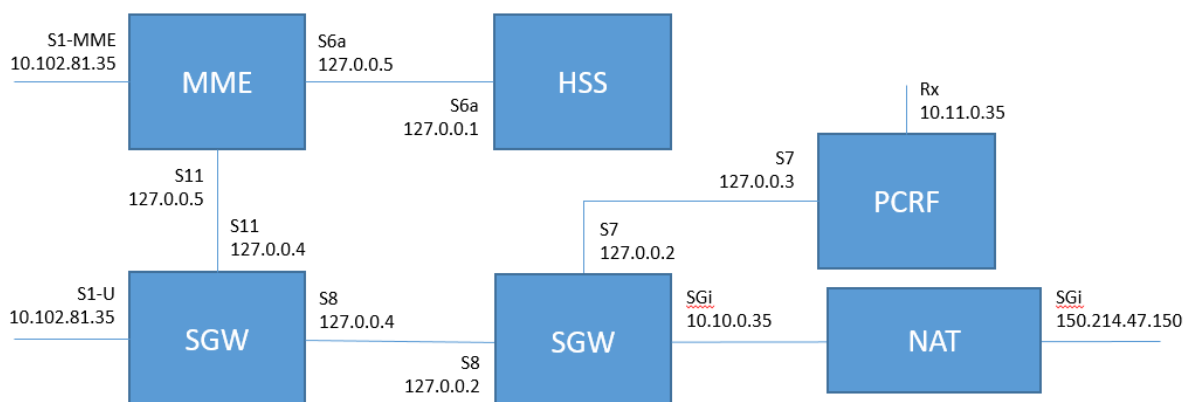


Figure 25: Default configuration for the EPC network

The commands for the MME are launched with a delay of 100 ms. They are used to trigger signalling procedures to extract traces that could be used by the network measurement



framework to produce statistics on the time consumed by different control plane procedures, such as attach, detach, service request and paging.

Table 9: MME Specific Commands for the EPC SCPI Server

Command	Description
EPC:MME:DETA imsi	Trigger a detach procedure for the specified IMSI, the detach message will be of type 1, cause 0 and will indicate that reattach is required.
EPC:MME:REL imsi	Trigger a UE Context Release procedure for the specified IMSI, the message will have cause group 3 and cause value 0.
EPC:MME:PAG imsi	Trigger a paging procedure for the specified IMSI.

The PCRF configuration is mainly used to trigger creation and deletion of dedication bearers. Current version of the commands does not support the specification of a service data template, which will be used to specify which traffic will be sent over the dedicated radio bearer. There will be an additional command to specify the traffic template to be used in the creation commands.

Table 10: PCRF Specific Commands for the EPC SCPI Server

Command	Description
EPC:PCRF:DED:CRE imsi, uelp, qci, mbrDI, mbrUI, gbrUI, gbrDI	This command trigger a dedicated radio bearer in the PCRF, the parameters required are the following: <ul style="list-style-type: none">- imsi is the IMSI of the UE- uelp is the IP of the UE- qci, indicates the required QCI for the DRB.- mbrDI, Maximum BitRate for the downlink- mbrUI, Maximum BitRate for the uplink- gbrDI, Guaranteed BitRate for the downlink- gbrUI, Guaranteed BitRate for the uplink
EPC:PCRF:DED:REL imsi uelp qci	This command trigger a dedicated radio bearer release in the PCRF, the parameters required are the following: <ul style="list-style-type: none">- imsi is the IMSI of the UE- uelp is the IP of the UE- qci, indicates the required QCI of the DRB that is going to be released.



5 Conclusion

During the first two years, the TRIANGLE project has massively expanded the testbed capabilities. Starting from a series of individual components, the project has not only upgraded the individual instruments, but interconnected them together to provide a sandbox for End-to-End testing.

Some of the upgrades have been fully integrated and tested within the testbed architecture. The testbed is now capable of establishing an End-to-End connection between the UE and the Application server through a commercial EPC. Such connection can be controlled and network impairments injected to provide a truly realistic feeling to the customers. The measurement logs are accurately time-synchronized, for having a unique time reference within the testbed. QoS can be activated on demand via VELOX while still under the control of the orchestrator.

The OTA measurements performed by the project have been of valuable help in the standardization process, and they have been presented in standardization for a comparison.

Some upgrades have a slight delay, i.e. the S1 interface and the LWIP, but we assume they will be completed within a few weeks from the delivery of this document. The S1 has only few functionalities in the U-plane that needs to be fully integrated into the UXM, to be compatible with the current software product upgrade.

LWIP has two sides, the network one and the UE client one. While the network side of LWIP is almost completed, the UE part is still under development. This is due to the lack of COTS UE supporting such functionalities, meaning that even the network behaviour, i.e. the traffic forwarding entity/scheduler, requires to be adjusted and tested after UE implementation is completed.



6 References

- [1] CTIA, Test Plan for 2x2 Downlink MIMO and Transmit Diversity Over-the-Air Performance, Version 1.1., August 2016
- [2] 3GPP, 3GPP TR 37.977 V14.5.0 (2017-09), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Universal Terrestrial Radio Access (UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRA); Verification of radiated multi-antenna reception performance of User Equipment (UE) (Release 14).
- [3] R4-1609332, Keysight Technologies, "MPAC alignment measurement results", RAN4#81 Meeting
- [4] R4-1609333, Keysight Technologies, "Root cause analysis for MPAC alignment", RAN4#81 Meeting
- [5] CTIA Certification Program Working Group Contribution, "Root cause analysis for MPAC alignment", November 2016.
- [6] R4-1609337, Keysight Technologies "Draft CR to add V/H spatial correlation validation procedure", RAN4#81 Meeting
- [7] R4-1609336, Keysight Technologies "CR to 37.977 to add noise floor validation procedure", RAN4#81 Meeting
- [8] R4-1610320, Keysight Technologies "Correction of uplink power control setting", RAN4#81 Meeting
- [9] 3GPP, TS 36.413 3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); S1 Application Protocol (S1AP), 9.3
- [10] 3GPP, TS 36.300 Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description
- [11] 3GPP, TS 36.361 LTE/WLAN Radio Level Integration using IPSec Tunnel (LWIP) encapsulation; Protocol specification
- [12] 3GPP, TS 36.331 Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification
- [13] 3GPP, TS 33.401 3GPP System Architecture Evolution; Security architecture, Annex H

Appendix 1: S1 Library

The S1AP Library provides means to create an S1-MME interface with MME entities of remote EPCs. To do so, SCTP associations are established and managed, as well as S1AP (S1 Interface Application Protocol) messages are sent/received over this transport protocol, as shown in the following picture:

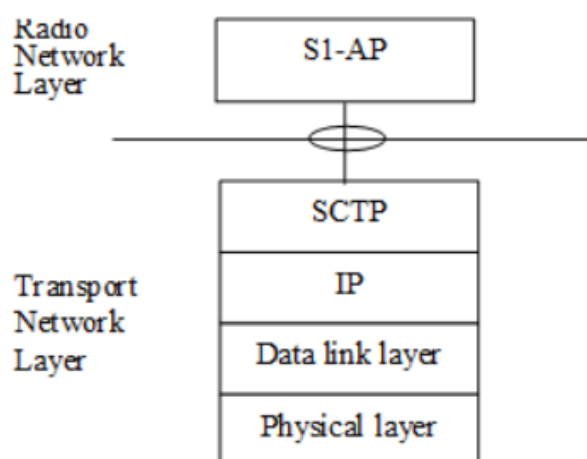


Figure 26 S1-MME Protocol Stack

S1AP Library provides an API which allows user to configure SCTP parameters, set a callback function to receive S1AP messages and send S1AP messages.

S1AP Library Architecture

The figure below briefly depicts the architecture of the S1AP Library:

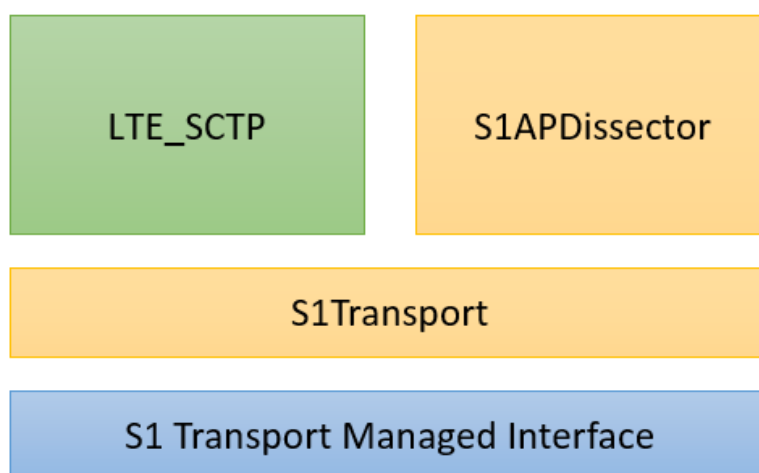


Figure 27 S1Library Architecture

- **S1TransportManagedInterface.** The core of the library has been written in C/C++ language, whereas UXM utilizes C#. For this reason, a specific interface between managed and not managed code was required, provided by this component.
- **S1Transport** performs several tasks:



- Management of SCTP associations.
- Provision of simple functions to generate and send uplink messages (eNodeB to MME direction).
- Provision of a callback functionality to send the received downlink messages (MME to eNodeB direction) in an external sink.

This component uses an S1AP message repository to build new messages to be sent. Users can modify this repository to set some parameters not handled by the library before execution.

- **S1APDissector** is in charge of encoding/decoding S1AP messages. These messages are codified with ASN.1 as per the definitions included in [9].

The class “BasicS1Message” implements a basic set of functions and parameters which are common for all the S1AP messages. For every single message, a class inheriting from BasicS1Message is implemented, handling specific information. The following picture shows an example with the message S1SetupRequest:

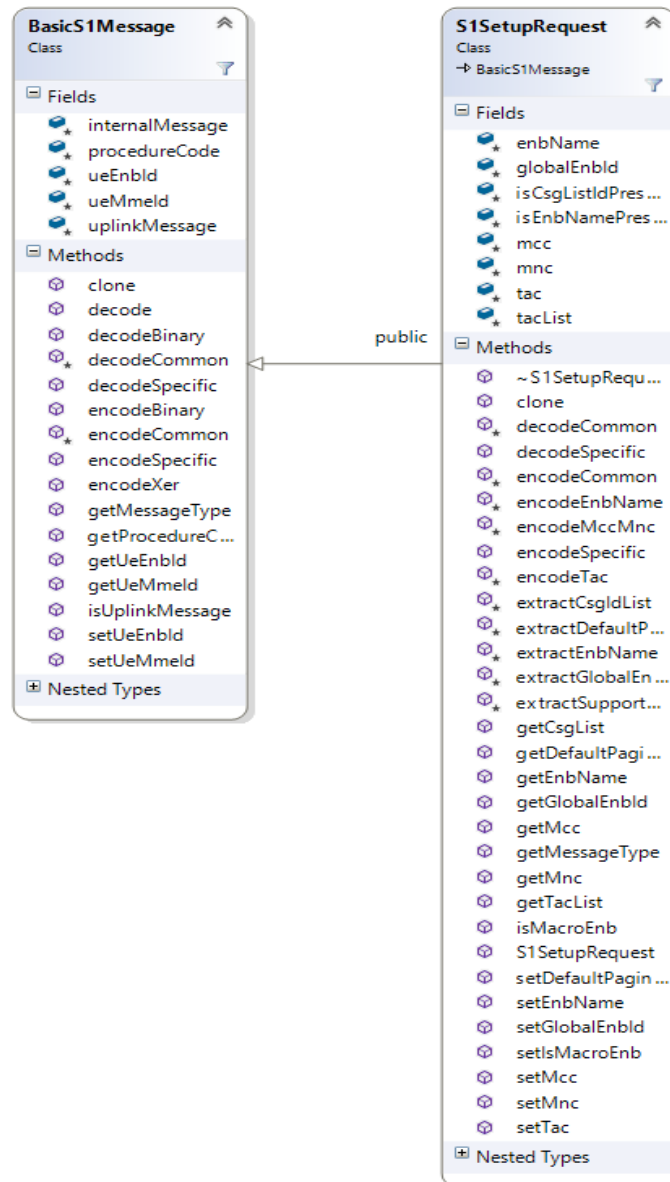


Figure 28 S1APDissector example

Besides the variables storing S1SetupRequest information, the functions “encodeSpecific” and “decodeSpecific”, which implement the encoding/decoding of the IEs (Information Elements) included in this message are relevant.

To have ASN.1 support, the S1AP Library uses the compiler/parser “asn1c”. This compiler has been modified to support APER encoding (Aligned Packet Encoding Rules) as required by the S1MME interface, and also a wrapper has been implemented to support some missing structures contained in the S1AP definitions (mainly Information Object).

- **LTE_SCTP** wraps the library “usrctp”, which supports SCTP on user space. This multiplatform implementation is based on the FreeBSD sctp.

**Public Interface**

The following table enumerates the functions that the public API of the S1AP Library provides:

Table 11 S1Library API Functions

Function	Description
initEmulation	Initiates S1AP Library
terminateEmulation	Terminates S1AP Library
setCallback	Sets the function that will be called when an S1MME message is received. This function has to inherit from "S1TransportMessageSink".
registerBaseStation	Creates an SCTP association with remote peer and sends the S1SetupRequest message
sendInitialUeMessage	Sends message InitialUeMessage
sendNasMessage	Sends message UplinkNasTransport
sendInitialContextSetupResponse	Sends message InitialContextSetupResponse
sendUeCapabilityInfoIndication	Sends message UeCapabilityInfoIndication
sendUeContextReleaseComplete	Sends message UeContextReleaseComplete
sendUeContextReleaseRequest	Sends message eContextReleaseRequest
sendRabSetupResponse	Sends message RabSetupResponse
sendRabReleaseResponse	Sends message RabReleaseResponse
sendResetAcknowledgement	Sends message ResetAcknowledgement
sendHandoverRequired	Sends message HandoverRequired
sendHandoverRequestAck	Sends message HandoverRequestAck
sendEnbStatusTransfer	Sends message EnbStatusTransfer
sendHandoverNotify	Sends message HandoverNotify

The complete list of functions, including parameters, can be found at Appendix 1:

Available Procedures

The following table shows the list of S1AP procedures and messages supported by the S1 Library so far:

Table 12 S1 Library Supported Procedures

S1AP Procedure	S1AP Messages
S1Setup	S1 SETUP REQUEST
	S1 SETUP RESPONSE
	S1 SETUP FAILURE
Initial UE Message	INITIAL UE MESSAGE
Initial Context Setup	INITIAL CONTEXT SETUP REQUEST



	INITIAL CONTEXT SETUP RESPONSE
UE Capability Info Indication	UE CAPABILITY INFO INDICATION
UE Context Release	UE CONTEXT RELEASE COMMAND
	UE CONTEXT RELEASE COMPLETE
UE Context Release Request	UE CONTEXT RELEASE REQUEST
E-RAB Setup	E-RAB SETUP REQUEST
	E-RAB SETUP RESPONSE
E-RAB Release	E-RAB RELEASE COMMAND
	E-RAB RELEASE RESPONSE
Reset	RESET
	RESET ACKNOWLEDGE
Downlink NAS Transport	DOWNLINK NAS TRANSPORT
Uplink NAS Transport	UPLINK NAS TRANSPORT
Paging	PAGING
Error Indication	ERROR INDICATION
Handover Preparation	HANDOVER REQUIRED
	HANDOVER COMMAND
	HANDOVER PREPARATION FAILURE
Handover Resource Allocation	HANDOVER REQUEST
	HANDOVER REQUEST ACKNOWLEDGE
Handover Notification	HANDOVER NOTIFY
eNodeB Status Transfer	eNodeB STATUS TRANSFER
MME Status Transfer	MME STATUS TRANSFER



S1 Library public API functions and parameters

Table 13 S1Library API functions and parameters list

Function	Parameter
bool initEmulation	--
bool terminateEmulation	--
bool registerBaseStation	<code>string</code> mmelpAddress, <code>bool</code> isMacro, <code>uint</code> enbld, <code>ushort</code> tac, <code>byte</code> mcc, <code>byte</code> mnc, <code>string</code> enbName
bool unregisterBaseStation	--
bool sendEnbStatusTransfer	<code>uint</code> ueEnbld, <code>uint</code> ueMmld, <code>byte</code> eRabld, <code>uint</code> ulCountValue_pdcpsn, <code>uint</code> ulCountValue_hfn, <code>uint</code> dlCountValue_pdcpsn, <code>uint</code> dlCountValue_hfn
bool sendHandoverNotify	<code>uint</code> ueEnbld, <code>uint</code> ueMmld, <code>byte</code> eutranCgi_mcc, <code>byte</code> eutranCgi_mnc, <code>uint</code> eutranCgi_cellId, <code>uint</code> tai_tac
bool sendHandoverRequestAck	<code>uint</code> ueEnbld, <code>uint</code> ueMmld, <code>uint</code> eRabAdmittedList_eRabld, <code>string</code> eRabAdmittedList_transportLayerAddress, <code>uint</code> eRabAdmittedList_teid, <code>byte[]</code> ttstc_rrcContainer
bool sendHandoverRequired	<code>uint</code> ueEnbld, <code>uint</code> ueMmld, <code>bool</code> targetId_isMacro, <code>uint</code> targetId_Enbld, <code>ushort</code> targetId_tac, <code>byte</code> targetId_mcc, <code>byte</code> targetId_mnc, <code>bool</code> directForwardingPathAvailability,



	<code>byte[] sttc_rrcContainer,</code> <code>uint sttc_eRabInfo_eRabId,</code> <code>bool sttc_eRabInfo_dIForwarding,</code> <code>uint sttc_eutranCgi_cellId,</code> <code>byte sttc_ueHistoryInfo_CellSize,</code> <code>byte sttc_ueHistoryInfo_eutranCellId_mcc,</code> <code>byte sttc_ueHistoryInfo_eutranCellId_mnc,</code> <code>uint sttc_ueHistoryInfo_eutranCellId_cellId,</code> <code>uint sttc_ueHistoryInfo_timeUeStayedInCell</code>
bool sendInitialContextSetupResponse	<code>uint ueEnbId,</code> <code>uint ueMmeId,</code> <code>uint eRabId,</code> <code>string transportLayerAddress,</code> <code>uint teid</code>
bool sendInitialUeMessage	<code>uint ueEnbId,</code> <code>uint cellId,</code> <code>byte[] nasBuffer,</code> <code>S1TransportEmulation.EstablishmentCause cause,</code> <code>uint m_TMSI,</code> <code>byte mmec</code>
bool sendNasMessage	<code>uint ueEnbId,</code> <code>uint ueMmeId,</code> <code>byte[] nasBuffer</code>
bool sendRabReleaseResponse	<code>uint ueEnbId,</code> <code>uint ueMmeId,</code> <code>uint eRabId</code>
bool sendRabSetupResponse	<code>uint ueEnbId,</code> <code>uint ueMmeId,</code> <code>uint eRabId,</code> <code>string transportLayerAddress,</code> <code>uint teid</code>
bool sendResetAcknowledgment	--
bool sendUeCapabilityInfoIndication	<code>uint ueEnbId,</code> <code>uint ueMmeId,</code> <code>byte[] radioBuffer</code>
bool sendUeContextReleaseComplete	<code>uint ueEnbId, uint ueMmeId</code>
bool sendUeContextReleaseRequest	<code>uint ueEnbId,</code> <code>uint ueMmeId,</code> <code>S1TransportCause cause</code>



Document: ICT-688712-TRIANGLE/D4.1

Date: 31/10/2017

Dissemination: PU

Status: Final

Version: 1.0

void setCallback

[S1TransportMessageSink](#) sink

Appendix 2: LWIPEP header treatment

LWIPEP entity, located at the IP layer of both eNodeB and UE, is the only addition to the protocol stack of the elements present in the RAN, and can be treated as an IP sublayer. This entity is responsible for the LWIPEP header treatment, meaning that it adds the LWIPEP header to IP packets in the UE and removes it in the eNodeB for the uplink, and removes it in the UE and adds it in the eNodeB for the downlink. Besides that, LWIPEP entity can be in charge of reordering packets using the sequence number in the header, previously mentioned, after the removal of the LWIPEP header. In addition to header treatment, the only functions of the LWIPEP entity are the transfer of user plane data, and the identification of the DRB identity to which a LWIPEP SDU belongs.

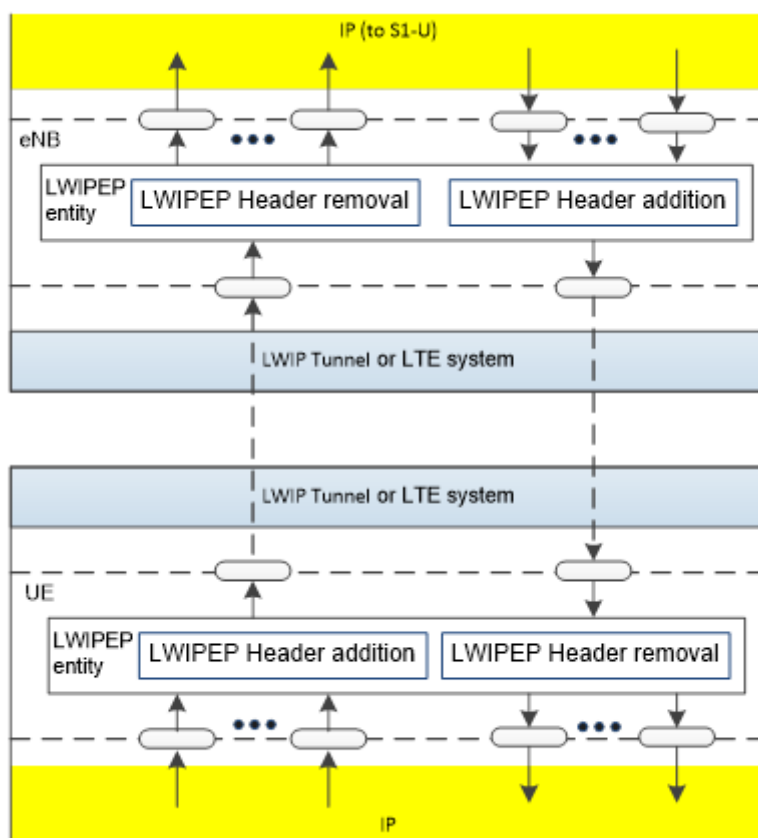


Figure 29 – Model of the LWIPEP sublayer

When LWIPEP entity adds the corresponding header to a packet, it forms a LWIPEP PDU. There is only one type of LWIPEP PDU at this moment, LWIPEP data PDU, and it is made of the header and the LWIPEP SDU. The LWIPEP header is a GRE header with a fix size that can be of eight or twelve bytes. If aggregation is active, both Key and Sequence Number fields are included, and the size is twelve bytes. If aggregation is not active, just the Key field appears in the header, and the size is eight bytes.



Table 14 - LWIP Header structure

LWIP Header

	Byte	0	1	2	3
Offset					
0		GRE Header Fields			
4		Key (DRB Identity)			
8		Sequence Number (Optional)			