



H2020 5G-Crosshaul project Grant No. 671598

D5.2: Report on validation and demonstration results

Abstract

This deliverable presents the evaluation results for the experiments done in the four demonstrations of the project. It covers the setup for the experiments, the experimentation procedure and discusses the results. The main result is the confirmation of the feasibility of the concepts proposed in the project and their contribution to meet the project objectives and 5G key performance indicators. More specifically, by integrating the architectural components designed in WP2, 3, and 4, we show: how fronthaul (FH) and backhaul (BH) traffic can coexist through integration of several transport technologies, such as those present in the networks of mobile operators today and foreseen for 5G (Demo 4); that the 5G-Crosshaul network can be controlled in a scalable way through software-define networking (SDN) techniques despite its complexity and dimension (Demo 3); that the 5G Crosshaul architecture offers the required flexibility and performance for coping with the needs of the most demanding use cases associated with video transmission (Demo 2); and that efficient energy management of the 5G-Crosshaul network (involving information technology and network resources) is possible (Demo 1).

Document Properties

Document Number:	D5.2
Document Title:	Report on validation and demonstration results
Document Responsible:	CTTC
Document Editor:	Josep Manges
Editorial Team:	Josep Manges
Target Dissemination Level:	Public
Status of the Document:	Final
Version:	2.0
Reviewers:	Thomas Deiß, Andres Garcia-Saavedra, Antonio de la Oliva, Charles Turyagyenda, Jaime Garcia, Alberto Diez, Maurizio Crozzoli

Disclaimer:

This document has been produced in the context of the 5G-Crosshaul Project. The research leading to these results has received funding from the European Community's H2020 Programme under grant agreement N° H2020-671598.

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

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

Table of Content

Table of Content	3
List of Contributors	6
List of Tables	7
List of Figures	9
List of Acronyms	13
Executive Summary	17
Key Achievements	19
1 Introduction	23
2 Experimental evaluation results for energy management of the 5G-Crosshaul (Demo 1)	24
2.1 Demo 1 setup	25
2.1.1 Building blocks from other work packages	25
2.1.2 Additional building blocks	28
2.2 Experiments on monitoring of power consumption in an XPFE domain	29
2.2.1 Experiment #1: Power Consumption Monitoring for XPFE physical nodes	29
2.2.2 Experiment #2: Power consumption monitoring for single network paths and tenants	37
2.3 Experiments on energy-oriented network resource management	41
2.3.1 Experiment #3: Energy-oriented network resource management in RoF domains	41
2.3.2 Experiment #4: Energy-oriented network resource management in XPFE domains	50
2.4 Experiments on energy-oriented provisioning of virtual infrastructures	59
2.4.1 Experiment #5: Energy-oriented virtual infrastructure management	59
2.5 Experiments on energy-oriented network resource management for fronthaul and backhaul traffic	69
2.5.1 Experiment #6: Energy-oriented network resource management in XPFE domains for on-demand provisioning of connections dedicated to fronthaul and backhaul traffic	69
2.5.2 Experiment #7: EMMA resource management over mmWave mesh	80
3 Experimental evaluation results for media distribution exploiting the services of the Crosshaul (Demo 2)	88
3.1 Demo 2 setup	89
3.1.1 Building blocks from other work packages	90

3.1.2	Additional building blocks	92
3.2	Experiments on media distribution	93
3.2.1	Experiment #8: Virtual CDN service on the 5G-Crosshaul infrastructure	93
3.2.2	Experiment #9: Multicast TV service provisioning	101
3.2.3	Experiment #10: Path reconfiguration when QoS degradation	105
3.2.4	Experiment #11: Distribution of live content through the vCDN infrastructure on the 5G-Crosshaul infrastructure	110
4	Experimental evaluation results for hierarchical multi-domain resource management of the crosshaul (Demo 3)	115
4.1	Demo 3 setup	116
4.1.1	Building blocks from other work packages.....	118
4.1.2	Additional building blocks	120
4.2	Experiments on single domain path provisioning and restoration.....	121
4.2.1	Experiment #12: Assessment of the SDN-based control of the Optical Transport Network (optical domain).....	121
4.2.2	Experiment #13: Assessment of the SDN-based control and data plane for the mmWave/Wi-Fi mesh domain.....	126
4.3	Experiments on multi-domain service provisioning and restoration	132
4.3.1	Experiment #14: End-to-end characterization. Network Orchestration across multiple heterogeneous domains: control and data plane characterization	133
5	Experimental evaluation results for crosshaul fulfilment of RAN split requirements (Demo 4).....	156
5.1	Demo 4 setup	157
5.1.1	Building blocks from other work packages.....	158
5.1.2	Additional building blocks	159
5.2	Experiments on 5G-Crosshaul traffic over integrated data plane technologies	160
5.2.1	Experiment #15: Backhaul and Fronthaul services integration through an SDN WS-WDM-PON transport network and XPFEs + Radio-over-Fibre	160
5.2.2	Experiment #16: Evaluation of mixed Digital/Analogue Radio-over-Fibre Implementation.....	168
5.2.3	Experiment #17: Integration of XCSE, XPFE, mmWave and CPRI compression data plane solutions	172

5.2.4 Experiment #18: Evaluation of integrated packet-based FH/BH combining
wired XPFEs and hybrid mmWave/optical wireless links 189

6 Components from technical work packages demonstrated in WP5 experiments 195

6.1 Work package 2: Physical and link layer of 5G-Crosshaul 195

6.2 Work package 3: 5G-Crosshaul Control and Data planes 196

6.3 Work package 4: Enabled innovations through 5G-Crosshaul 197

7 References 198

8 Appendix 1: Current generic network provisioning procedure in a telco provider
infrastructure 201

List of Contributors

Partner No.	Partner Short Name	Contributor's name
P01	UC3M	Jaime Garcia-Reinoso
P02	NEC	Andres Garcia Saavedra, Josep Xavier Salvat, Xi Li
P03	EAB	Chenguang Lu, Daniel Cederholm, Miguel Berg
P04	TEI	Stefano Stracca
P05	ATOS	José Enrique González Blázquez, Beatriz López Herraiz
P06	NOK-N	Thomas Deiß
P07	IDCC	Charles Turyagyenda
P08	TID	Luis M. Contreras
P09	TI	Maurizio Crozzoli
P10	ORANGE	Luiz Anet Neto
P11	VISIONA	Carlos Navarro Martínez, Nuria Sánchez
P13	NXW	Marco Capitani, Elian Kraja, Giada Landi, Francesca Moscatelli
P14	CND	Alberto Diez
P15	TELNET	Enrique Masgrau
P16	FhG-HHI	Jens Pils, Dominic Schulz, Anagnostis Paraskevopoulos
P17	CTTC	Josep Mangués-Bafalluy, Jorge Baranda, José Núñez-Martínez, Ramon Casellas, Arturo Mayoral, Ricard Vilalta, Iñaki Pascual, Juan Luis de la Cruz
P18	CREATE-NET	Raihana Ferdous, Damu Ding
P19	POLITO	Claudio Casetti
P20	ULUND	Kaan Bür
P21	ITRI	Shahzoob Bilal Chundrigar, Samer Talat, Sean Chang

List of Tables

Table 1: Details of the paths installed for EMMA tests on XPFEs domain	31
Table 2: Average power consumption measurement for physical nodes (Watt)	32
Table 3: Power consumption elaborated by EMMA application for virtual entities (paths and tenants) – 1 Gbps of traffic.....	40
Table 4: Details of the energy consumption of RoF nodes	44
Table 5: Packet size of AMQP messages.....	48
Table 6: Details of the paths installed for EMMA tests on XPFEs domain.....	53
Table 7: Average connection setup time (ms)	56
Table 8: Network topologies and number of path for XCI scalability test	58
Table 9: Network connections between VMs established during the vEPC NS provisioning	66
Table 10: Network Service provisioning time components (in seconds)	66
Table 11: Network Service provisioning time components in 5G-Crosshaul testbed (in seconds).....	67
Table 12: Connections details as requested on the Provisioning Manager NBI.....	73
Table 13: Connection details as computed by the Path Computation Manager	73
Table 14: Power measurements and savings in mmWave mesh	86
Table 15: Setup delay. Parent ABNO [seconds].....	138
Table 16: Packet distribution of COP protocol message exchange during previous experiment.	139
Table 17: Setup delay hierarchical XCI: Parent ABNO vs RMA [seconds].....	141
Table 18: Service setup time seen from the different control plane elements [seconds].....	146
Table 19: Service deletion time seen from the different control plane elements [seconds].....	146
Table 20: Packet size distribution (in bytes) of the communication between pABNO and RMA.....	148
Table 21: Local service recovery control plane delay [seconds]	150
Table 22: Centralized service recovery control plane delay [seconds]	151
Table 23: Multi-domain/multi-technology hierarchical XCI data plane performance	152
Table 24: Multi-site XCI transport-level performance	153
Table 25: Multi-site service performance between users for case a)	154
Table 26: Multi-site service performance between users for case b)	154
Table 27: Multi-site service RTT performance [milliseconds]	155
Table 28: Wavelength plan	163
Table 29: Latency at low throughput.....	165
Table 30: Latency at target throughput, separate FH and BH traffic	165
Table 31: Latency at target throughput, simultaneous FH and BH traffic.....	166

Table 32: EVM results (average over 20ms).....	167
Table 33: Throughput scenario 17.1.1 (average and standard deviation).....	178
Table 34: Throughput scenario 17.1.2.....	178
Table 35: Throughput scenario 17.1.3 (average and standard deviation).....	179
Table 36: Throughput scenario 17.1.4 (average and standard deviation).....	179
Table 37: RTT scenario 17.1.1 (average and standard deviation)	180
Table 38: RTT scenario 17.1.2 (average and standard deviation)	180
Table 39: RTT scenario 17.1.3	181
Table 40: RTT scenario 17.1.4 (average and standard deviation)	181
Table 41: Measurement results of latency and packet loss rates for both Normal and FF modes under different traffic profiles	185
Table 42: End-to-end measurement results for EdgeLink	186
Table 43: Latency for directly connected XPFEs	191
Table 44: Latency for XPFEs connected via mmWave link	192
Table 45: Latency for XPFEs connected via optical link	192
Table 46: Latency for XPFEs connected via hybrid wireless link.....	193
Table 47: Physical and link layer components in experiments.....	195
Table 48: Use of XCI components in experiments	196
Table 49: Use of dataplane components in experiments	196
Table 50: Use of applications in demos and experiments.....	197

List of Figures

Figure 1. Representation of the four integrated demonstrations of the project and the 5G-Crosshaul architectural building blocks involved in each of them.....	17
Figure 2. Test scenario and instantiated paths for EMMA experiments on XPFEs domain.....	31
Figure 3. SNMP message between controller and XPFE to retrieve a power consumption measurement.....	32
Figure 4. XPFEs and XPU's in sleeping mode (EMMA GUI).....	33
Figure 5. Network power consumption: all XPFEs and XPU's in sleeping mode (EMMA GUI).....	34
Figure 6. Active XPFEs and XPU's for Path 1 (EMMA GUI).....	34
Figure 7. Power consumption slightly increases with traffic on path 1 (Wasp to Ironman)	35
Figure 8. Active XPFEs for Path 1 and Path 2 (EMMA GUI)	36
Figure 9. Evolution of network power consumption during setup of Path 1 and Path 2 (EMMA GUI)	36
Figure 10. Power consumption with traffic on single XPFEs (EMMA GUI)	37
Figure 11. Power consumption associated to tenant 1, with split for Path 1 and Path 3	41
Figure 12. System under Test.....	42
Figure 13. RoF deployment in the High-Speed Train testbed	43
Figure 14. Energy consumption comparison chart.....	46
Figure 15. Log processing time for IPC	46
Figure 16. Round Trip Time to send context information from moving train to the controller	47
Figure 17. Round Trip time to Switch ON/OFF RoF nodes	47
Figure 18. EMMA GUI to track current location of the high-speed train	49
Figure 19. EMMA interface for the live energy monitoring of RoF nodes	49
Figure 20. Total setup time for each network path	53
Figure 21. Time used by the Provisioning Manager module to elaborate connection setup requests	54
Figure 22. Time used by the Path Computation Manager to compute the requested path.....	54
Figure 23. Time used by the Provisioning Manager to configure the flow rules through the SDN Controller SBI	55
Figure 24. Time used by the State Manager component to switch on the needed nodes	55
Figure 25. Evolution of network power consumption during setup of Path 1 and Path 2 (EMMA GUI)	56
Figure 26. Power consumption saving comparison between EMMA and SPF.....	57
Figure 27. Network topology with 51 nodes	58
Figure 28. Total provisioning time in emulated networks of different size.	59
Figure 29. NXW testbed environment.....	63

Figure 30. 5TONIC Demo1 topology	63
Figure 31. vEPC details	64
Figure 32. EMMA in XPFEs + XPU domains: average (dots), max and min values (line edges) of NS setup time and its components (NXW lab)	67
Figure 33. EMMA in XPFEs + XPU domains: average (dots), max and min values (line edges) of NS setup time and its components (5G-Crosshaul testbed)	68
Figure 34. Fronthaul and backhaul traffic integration environment with virtual networks details	72
Figure 35. Interaction between FH VMs and OpenEPC VMs	72
Figure 36. Interaction between BH VMs and OpenEPC VMs	72
Figure 37. FHRRUtoOpenEPCBBU and BHBBUtoOpenEPCMME connections view in Dlux Web GUI	74
Figure 38. FH traffic encapsulated in table 11 in XPFE 3	75
Figure 39. FH traffic forwarded through queue 5 in table 253 in XPFE 3	75
Figure 40. Background traffic encapsulated in table 11 in XPFE 3	76
Figure 41. Background traffic forwarded through queue 1 in table 253 in XPFE 3	76
Figure 42. BH traffic encapsulated in table 11 in XPFE 1	77
Figure 43. BH traffic forwarded in table 253 in XPFE 1	78
Figure 44. Background traffic encapsulated in table 11 in XPFE 1	78
Figure 45. Background traffic forwarded in table 253 in XPFE 1	79
Figure 46. 110 Mbps of traffic over FHRRUtoEPCBBU connection	79
Figure 47. 50 Mbps of traffic over a priority 1 connection	80
Figure 48. Iperf server output for traffic over FHRRUtoEPCBBU and parallel connection	80
Figure 49. mmWave Controller architecture	83
Figure 50. mmWave topology in XCI GUI	85
Figure 51. mmWave controller topology after EMMA path configuration	85
Figure 52. mmWave mesh power consumption during high traffic configuration	87
Figure 53. Setup of demo 2	89
Figure 54. GUI for TVBA showing the environment	90
Figure 55. GUI for CDNMA showing the service configuration and the CDN topology	90
Figure 56. Deployment time for four vCDN	98
Figure 57. Throughput comparison between a close replica server and a replica server located in an external network	99
Figure 58. CDNMA - Orchestration of new resources due to network failure detection	100
Figure 59. Average and standard deviation of metrics for KPIs	104
Figure 60. Average and standard deviation of each time involved in the metrics (zoom in of Figure 61)	104

Figure 61. Average and standard deviation of the longest times involved in the metrics (zoom out of Figure 60)	105
Figure 62. Average and standard deviation of metric for KPI	108
Figure 63. Average and standard deviation of each time involved in the metric (zoom in of Figure 64)	109
Figure 64. Average and standard deviation of the longest times involved in the metric (zoom out of Figure 63)	109
Figure 65. Total provisioning time for live content distribution through a vCDN with 2 replica servers	113
Figure 66. Self-healing time for live content distribution through a vCDN	114
Figure 67. Demo 3 architecture	116
Figure 68. Demo 3 testbed overview	117
Figure 69. Demo 3 testbed facilities in Barcelona site	117
Figure 70. Graphical User Interface of a) RMA, b) pABNO, and c) Wireless SDN Controller_1, application and control plane elements of the Demo3 testbed	118
Figure 71. Setup time histogram and CDF as seen at different reference points: from the RSVP-TE connection controllers, from the AS-PCE and from the ABNO network orchestrator.	124
Figure 72. Blocking Probability obtained for the 4-node Optical Transport Network requesting one wavelength from source to destination node, for 3 seconds avg. inter-arrival time and varying holding time. X-axis corresponds to traffic load.....	126
Figure 73. Path Computation (or Setup) results for the mmWave/Wi-Fi mesh domain	130
Figure 74. Data Plane Throughput Results for an mmWave/Wi-Fi mesh domain	131
Figure 75. Path Restoration Results in the mmWave/Wi-Fi domain	132
Figure 76. Network topologies as deployed (bottom) and as aggregated by the pABNO controller (top).	138
Figure 77. Histogram and CDF of the setup delay seen by the ABNO orchestrator for the considered topology.....	138
Figure 78. Bit rate of COP protocol message exchange during the previous experiment	139
Figure 79. Latency measured for basic RMA operations for 5 flows	141
Figure 80. RMA. Characterization of the calls during the “Start” phase.....	142
Figure 81. RMA. Characterization of the calls during the FindRoutes operation	142
Figure 82. RMA. Characterization of the calls during the Install Paths operation	143
Figure 83. RMA. Characterization of the calls during the Delete Flows operation.....	143
Figure 84. RMA. Characterization of the calls during the HTTP Client Requests.....	144
Figure 85. Normalized histogram and CDF of the service setup time seen by the RMA.....	145
Figure 86. Normalized histogram and CDF of the service delete time seen by the RMA.....	145
Figure 87. Instantaneous bit rate of control messages exchanged for service setup and deletion between RMA and pABNO (blue: pABNO-to-RMA, red: RMA-to-pABNO)	148

Figure 88. Local recovery histogram (left) and CDF (right) after a link failure/recovery (local down and local up times respectively) in the mmWave/WiFi domain.....	149
Figure 89. Normalized Histogram and CDF of centralized recovery operation seen by the RMA.....	151
Figure 90. passive optical filters above ONUs for separation of RoF and WDM signals	167
Figure 91. EVM result screen shoot after integration	168
Figure 92. EVM (%) to mean received optical Power (dBm).....	171
Figure 93. EVM (%) to number of bands	171
Figure 94. EVM (%) and Power (dBm) to frequencies (MHz)	172
Figure 95. Picture showing some of the components used in experiment #17.....	176
Figure 96. Ethernet bitrate [Mbps] over time	182
Figure 97. EVM [%] over time with average and 2 standard deviations.....	183
Figure 98. EVM PDF histogram.....	183
Figure 99. Frequency error [ppm] over time with average and 2 standard deviations	184
Figure 100. Frequency error PDF histogram.....	184
Figure 101. Average RTT between the LL UE and the LL DU	187
Figure 102. Maximum RTT between the LL UE and the LL DU	187
Figure 103. Average RTT between the HL RU and the HL DU	188
Figure 104. Maximum RTT between the HL RU and the HL DU	189

List of Acronyms

Acronym	Description
ABNO	Applications Based network Operation
API	Application Programming Interface
AS-PCE	Active Stateful PCE
BBU	Base-Band Unit
BH	Backhaul
BP	Blocking Probability
C&M	Control and Management
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
CDNMA	Content Delivery Network Management application
COP	Control Orchestration Protocol
CPE	Customer Premises Equipment
CPFH	Compressed and Packetized Fronthaul
CPRI	Common Radio Public Interface
DML	Directly Modulated Laser
DMM	Distributed Mobility Management
EMAN	Energy MANAGEMENT
EMMA	Energy Monitoring and Management Application
eNB	eNodeB
EPC	Evolved Packet Core
EVM	Error Vector Magnitude
FF	Fast Forwarding
FH	Fronthaul
FHU	Fronthaul Unit
GMPLS	Generalized Multi-Protocol Label Switching
GPON	Gigabit Passive Optical Network
GUI	Graphical User Interface
HD	High Definition
HL	Higher-Layer split
IPC	Industrial Personal Computer
I/Q	In-phase and Quadrature
IT	Information Technology

KPI	Key Performance Indicator
LL	Lower-Layer split
LSP	Label Switched Path
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MIB	Management Information Block
MMA	Mobility Management Application
MME	Mobility Management Entity
MW	Microwave
NFV	Network Functions Virtualization
NFVO	NFV Orchestrator
NS	Network Service
NSD	Network Service Descriptor
NTP	Network Time Protocol
OAI	OpenAirInterface
OAM	Operations, Administration, and Maintenance
ODL	OpenDaylight
OF	OpenFlow
OIF	Optical Interworking Forum
OVS	Open Virtual Switch
OWC	Optical Wireless Communication
OXC	Optical Cross-connect
PCE	Path Computation Element
PCI	Physical Cell Identifier
PDCP	Packet Data Convergence Protocol
PDU	Power Distribution Unit
P-GW	Packet Data Network Gateway
PHY	Physical layer
PM	Provisioning Manager
PoC	Proof-of-Concept
PON	Passive Optical Network
PSTN	Public Switched Telephone Network
PTP	Precision Time Protocol
QoS	Quality of Service
QoE	Quality of Experience

RH	Radio Head
RLC	Radio Link Control
RoF	Radio over Fibre
RMA	Resource Management Application
ROADM	Reconfigurable Optical Add Drop Multiplexer.
RRU	Remote Radio Unit
RTMP	Real-Time Messaging Protocol
RTT	Round-Trip Time
SAP	Service Access Point
SBI	Southbound Interface
SDN	Software Defined Network
S-GW	Serving Gateway
SNMP	Simple Network Management Protocol
SOTA	State of the Art
SPF	Shortest Path First
TCO	Total Cost of Ownership
TDM	Time-Domain Multiplexing
TVBA	TV Broadcasting application
TE	Traffic Engineering
TED	TE database
TM	Topology Manager
UE	User Equipment
vCDN	Virtual Content Delivery Network
vEPC	Virtual Evolved Packet Core
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
VNTM	Virtual Network Topology Manager
VoD	Video on Demand
WSO	Wavelength Switched Optical Networks
WS-WDM	Wavelength Selected Wavelength Division Multiplexing
WS-WDM-PON	WS-WDM Passive Optical Network
WS-WDM-OLT	WS-WDM Optical Line Termination
WS-WDM-ONU	WS-WDM Optical Network Unit

XCI	5G-Crosshaul Control Infrastructure
XCF	5G-Crosshaul Common Frame
XCSE	5G-Crosshaul Circuit Switching Element
xDPd	OpenFlow eXtensible DataPath daemon
XFE	5G-Crosshaul Forwarding Element
XPFE	5G-Crosshaul Packet Forwarding Element
XPU	5G-Crosshaul Processing Unit

Executive Summary

This document defines the experiments in the 5G-Crosshaul project and discusses the experimental results obtained. A catalogue of experiments for the four demonstrations in the project are defined.

The four demonstrations of the 5G-Crosshaul network show: how fronthaul (FH) and backhaul (BH) traffic can coexist through integration of several transport technologies, such as those present in the networks of mobile operators today and foreseen for 5G (Demo 4); that the 5G-Crosshaul network can be controlled in a scalable way through software-define networking (SDN) techniques despite its complexity and dimension (Demo 3); that the 5G Crosshaul architecture offers the required flexibility and performance for coping with the needs of the most demanding use cases associated with video transmission (Demo 2); and that efficient energy management of the 5G-Crosshaul network (involving information technology and network resources) is possible (Demo 1).

Figure 1 provides a high-level view of the architectural components involved in each demonstration. A more detailed explanation of each demo is provided in the following sections.

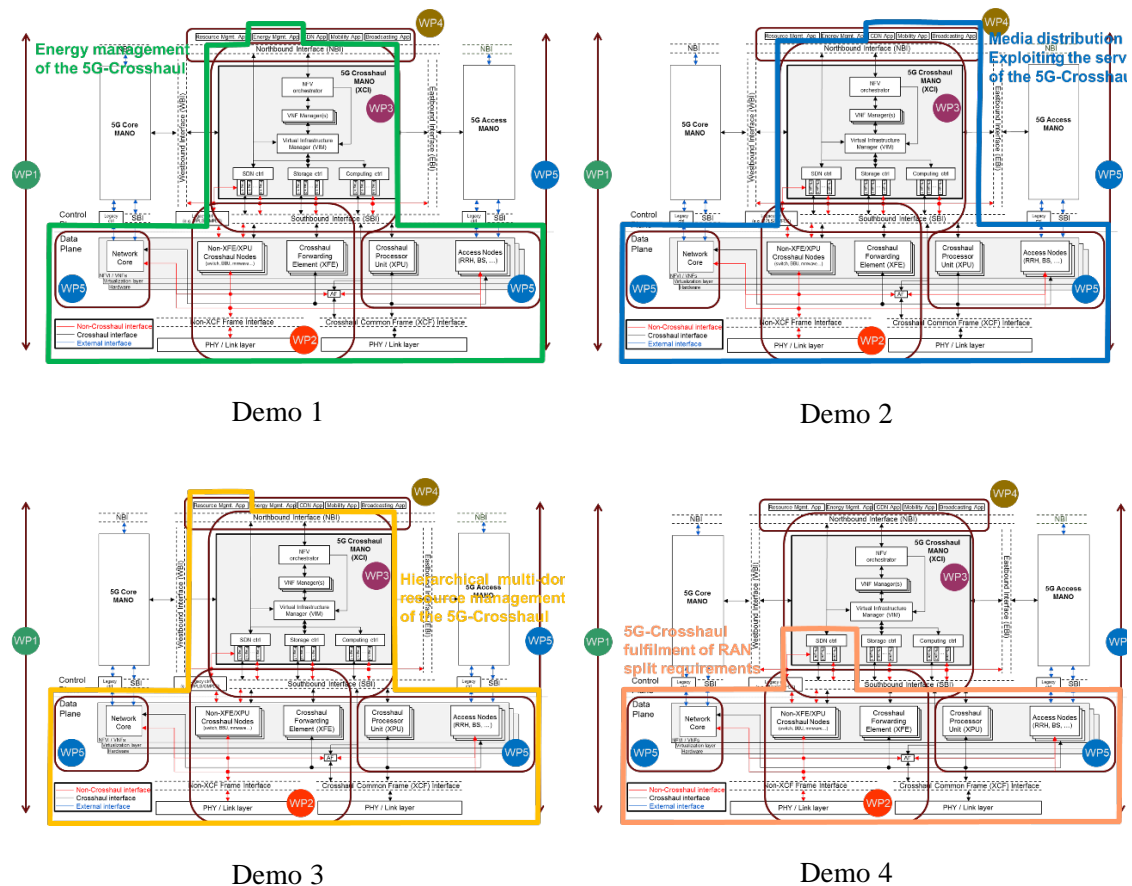


Figure 1. Representation of the four integrated demonstrations of the project and the 5G-Crosshaul architectural building blocks involved in each of them

The catalogue of experiments described in this document run in the 5G-Crosshaul testbed across different locations integrating diverse technologies and functions and serve as proof that the concepts proposed in the project meet the 5G Key Performance Indicators (KPIs) and the

objectives of the project. All experiments have been defined to provide quantifiable measurements aligned with those KPIs and objectives. Therefore, the results in this deliverable show the value of the research done in the project. This is the basis for dissemination and exploitation activities done in the project.

A big effort was put on the definition of experiments towards integration of as much conceptually-related technologies as possible in a single experiment setup. Therefore, the concepts developed in the technical work packages (WP2, 3, 4) are validated and evaluated in an integrated and more general setup in the context of WP5. This integration was done by always bearing in mind the practical constraints imposed by the physical deployments in place.

Notice also that various experimental setups covering some aspects of the four demonstrations have been presented in multiple events, including Mobile World Congress 2016 and 2017, EUCNC 2016 and 2017, the second Global 5G event (2016), WWRF39, and IEEE NFV-SDN 2017, and various papers including experimental evaluations of technologies of the technical WPs have been presented in international conferences (e.g., EUCNC'17).

Key Achievements

A summary of the main results obtained for each demonstration is presented below.

Demo 1 focuses on demonstrating the applicability of the **5G-Crosshaul energy efficiency** concepts through different types of network technologies. Furthermore, it has demonstrated the feasibility of **power consumption monitoring** not only for physical infrastructure elements (XPFEs, XPU), but also virtual or service-level entities like network paths, virtual network functions (VNFs) or tenants. The results have shown that the power consumption of a node can be decomposed into a constant baseline component and a traffic load dependent component. For the system under evaluation, the constant component is around 21 W in sleeping mode and 35 W in active mode, while the variable component due to the traffic load is very low, in the order of hundreds of mW (e.g., around 500mW for 1Gbps traffic for the devices under test). It has also been demonstrated the **on-demand provisioning of energy-efficient network connections** over the 5G-Crosshaul Packet Forwarding Engine (XPFE) and Radio over Fibre (RoF) domains, with automated regulation of device's power states (i.e., sleeping mode, active mode with low traffic, active mode with high traffic). In both domains, the results achieved have provided measurable energy savings. For instance, in the XPFE scenario, without connections, we reach a power saving of 84 W when compared with the 210W of the "always on" approach (around 40%), while with a single path established, the power saving is of 42 W (around 20% reduction). **Scalability** of the energy saving algorithms was evaluated by emulating in mininet realistic topologies (including a 51-node regional network in the North-West of Italy and networks of up to 250 nodes). Under worst case traffic conditions (i.e., maximum traffic load) a global energy saving of 12% can be achieved (due to switching off 6 nodes). This saving must be added to those studied in WP1 (of up to 70%). Additionally, energy savings of 78.6%, which is equivalent to energy savings of approx. 17257 KJ per day, are achieved for the RoF system in the high-speed train site by appropriately switching on and off the RoF nodes. In the same way, substantial energy savings were obtained in the mmWave mesh topology under evaluation. These savings increase around 20% for each additional node that can be switched off as a function of traffic demand.

One additional key aspect to consider is that of service provisioning time, including the switching between different energy management states (when applicable). For the scenario under evaluation, connection setup time for network paths crossing three nodes is in the order of 3 seconds when an XPFE power state change is required. It is less than 1 second otherwise. Scalability tests were run with realistic topologies ranging from 50 to 250 nodes, the average connection setup time remains under 1s (i.e., 0.78s), though it may reach up to 7 seconds for the 250-node network if many nodes must change power state.

Additionally, when **IT resources** are involved (i.e., with automatic activation of the servers where vEPC virtual network functions are deployed), the average network service provisioning time is nearly 5 minutes, out of which the main component is virtual network function (VNF) configuration triggered and orchestrated at the network function virtualisation (NFV) Orchestrator and executed within each VNF followed by the virtual VNF virtual machine (VM) creation. Notice that the deployed VNFs result in fronthaul and backhaul traffic handled by the 5G-Crosshaul Control Infrastructure (XCI). These measurements demonstrate how network function virtualization and smart resource orchestration can effectively reduce the delivery time up to just a few minutes, even for complex end-to-end services, while guaranteeing an energy-efficient sharing of the infrastructure.

In summary, these results contribute in fulfilling the project and 5G PPP KPIs and objectives of reducing the energy consumption by 30% (and more) through energy management, and enabling the provisioning of 5G-Crosshaul services in the order of magnitude of minutes (instead of current values in the order of hours or days).

In the experiments of *demo 2*, we have shown that the TV broadcasting and Video on Demand (VoD) services can be easily deployed on the 5G-Crosshaul network using the 5G-Crosshaul Control Infrastructure (XCI) and the 5G-Crosshaul applications. The results obtained prove the feasibility of deploying virtual media (vCDN and TV broadcasting) infrastructure by exploiting the services offered by the XCI. From the XCI services perspective, the provisioning time and the self-healing capabilities have been evaluated. For Virtual Content Distribution Network (vCDN) services, the provisioning time varies from 157 s for a vCDN with one replica server to 217 s for a vCDN with four replica servers. The biggest contributor to this process is the time needed by the virtual machines to boot and complete the network configurations, ranging from 118 s to 135 s for vCDNs with 1 to 4 replica servers. The self-healing worst case scenario would require a new replica server due to a detected network failure. This new instantiation takes 45.641 s where 30 s of them correspond to the time needed by the VM to boot and apply network configurations. Regarding the load balancing offered by the vCDN, the results have shown that the values obtained for latency and throughput are better when the user is assigned to a replica server closer to the user's location (0.771 ms for latency and 749 Kbits/s for throughput) than when the user is assigned to a server located in different subnetworks (20.888 ms for latency and 354 Kbits/s for throughput).

In the same way, for the TV Broadcasting Application (TVBA), the average and standard deviation of the multicast provisioning time when the TVBA quality probe (TVBAQP) was not previously deployed are $92.72 \text{ s} \pm 4.54 \text{ s}$ ($8.58 \text{ s} \pm 0.37 \text{ s}$ when it was pre-deployed) and it consists of: 1) average user's provisioning time of $1.47 \text{ s} \pm 0.41 \text{ s}$, 2) average Quality Probe's deployment time of $83.62 \text{ s} \pm 4.47 \text{ s}$, 3) average Quality Probe's starting time of $0.33 \text{ s} \pm 0.13 \text{ s}$, and 4) average Quality Probe's analysing time of $5.72 \text{ s} \pm 0.46 \text{ s}$. On the other hand, the TVBA can self-heal the system in an average time of $41.67 \text{ s} \pm 7.28 \text{ s}$ consisting of: 1) average reaction time of $15.88 \text{ s} \pm 9.10 \text{ s}$, 2) average TVBA's decision time of $0.16 \text{ s} \pm 0 \text{ s}$, 3) average solution time of $1.72 \text{ s} \pm 0.39 \text{ s}$ and 4) same Quality Probe's times as before.

Finally, the measures obtained for the Live Content Distribution through a vCDN are populated from the two previous set of results. The total provisioning service time is composed of the instantiation time of a vCDN plus the TV Multicast service deployment time. For a vCDN initially configured with 2 replica servers, the total provisioning time is 249.72 s. Additionally, the self-healing time depends on whether a new replica server needs to be instantiated due to monitoring alerts detected by the CDNMA. If the CDNMA has to deploy a new CDN node the total self-healing time is 87.27 s. However, if only the TVBA needs to configure a new path, this time is set to 41.67 s.

In summary, this demo contributed in fulfilling the project and 5G PPP KPIs and objectives on service deployment time of hours (minutes in this case, for media services) and orchestration of 5G-Crosshaul resources based on traffic load variations.

In *demo 3*, we have shown that a complex multi-domain and multi-technology transport network can be controlled through a hierarchy of SDN controllers that expose the appropriate application programming interfaces (APIs) to the resource management application (RMA). These transport networks consist of heterogeneous technologies that need end-to-end orchestration. In the project, we evaluate a hierarchical controller for wireless/optical resources as seen from an RMA. More specifically, we deploy a hierarchical XCI for which a parent-child relationship is established between contiguous layers. At the top of the hierarchy, the parent controller is based on the IETF Application-Based Network operations (ABNO) architecture [27]. It offers to the RMA the appropriate abstraction level (end-to-end view) and orchestrates the different child controllers, which deal with the specificities of each underlying technology. To eventually understand the end-to-end behaviour related with service setup, we evaluate each network segment (wireless and optical), each plane (application and control planes), and each layer of the hierarchy inside the XCI. The results confirm the benefits of a hierarchy of controllers in which some technology-specific local decisions can be taken by the closest controller in the hierarchy, hence saving

processing and propagation time. In our case, and depending on the domain, there may be differences in path setup/restoration values observed at child vs. parent controller ranging from around 500ms to one order of magnitude (up to units of seconds). For instance, the path restoration time values in the wireless domain (hundreds of ms) and the path setup time at the child controllers (around 2.5s.) is smaller than that observed by the parent ABNO (3.349s.) and RMA (3.971s.). For the scenario under study, the main components of the path setup delay are: 1) RMA processing, 2) RMA-to-parent ABNO latency (about 60ms of RTT) for each message exchange, 3) bidirectional multilayer connection setup in the optical network (around 2.5s.), and 4) wireless domain (tens of ms) assuming a wireless control channel. Overall, experimental results provided show an average of 3.971 seconds for end-to-end (E2E) path setup delay, hence contributing to the 5G target KPI of lowering the service deployment time, in this case, multi-domain path setup, from months to minutes. In fact, these setup times are much lower than those obtained with current (manual) practice, in which values in the order of hours (or even days) are obtained.

From a service management perspective, demo 3 also evaluates the time it takes to deploy all the network resources required to deploy an LTE mobile network service featuring both fronthaul and backhaul (10.5s on average). A total of 16 unidirectional flows were set up through the multi-domain transport network between eNodeBs, remote radio units (RRU), and baseband units (BBU) at the radio access network, and with the serving/PDN gateway (SPGW) and mobility management entity (MME) at the core network.

The flexibility provided by the hierarchical XCI in the processing of a service recovery triggered by a link down event was also evaluated. This recovery process can be done either locally at the child SDN controller level (the lowest level of the control hierarchy) or centralized at the RMA level (the uppermost level of the control hierarchy). The former allows much lower recovery times (0.299s on average) at the cost of potentially suboptimal resulting paths, unlike the latter, which takes 6.652s on average for re-establishing an optimal multi-domain path.

For the sake of completeness, the deployed setup of demo3 has been also analysed from the data plane perspective. The assessed transport setup offers around 153 Mbps end-to-end (E2E), which is mainly limited by a virtual private network (VPN) connection between remote sites, and where emulated users using the previously mentioned LTE mobile network service and placed at both edges of the setup achieve up to 140 Mbps.

In summary, the work carried out in demo 3 contributes in fulfilling the project and 5GPP KPIs objectives of enabling the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of minutes (seconds in this case). It offers a scalable management framework through its hierarchical deployment, and enables the deployment of novel applications and the orchestration of 5G-Crosshaul resources reducing the network management Operational Expenditure (OPEX) due to the flexibility and programmability offered by the XCI in a multi-domain multi-technology setup, which will be the norm in 5G network deployments.

Finally, **demo 4** combines a variety of data plane technologies and evaluates what radio access network (RAN) split options (see [25] Table 5 and [49]) can be supported by each combination of technologies. The split options range between option 8 – split PHY – and option 1 – RLC-RRC. In this way, it is clearer for operators how to build the integrated fronthaul and backhaul at the data plane level to comply with given transport requirements.

We have integrated both backhaul and fronthaul traffic support over the same integrated infrastructure combining wavelength-selected wavelength division multiplexing passive optical network (WS-WDM-PON) that multiplexes traffic coming from 5G-Crosshaul Packet Forwarding Elements (XPFEs) and Radio-over-Fibre (RoF) technology. Data rates and latency measurements of WS-WDM-PON guarantee up to RAN split option 6 (symmetric 10Gbps per point-to-point dense WDM link). The data rates and latencies requirements up to RAN split option 6 are also supported in combination with an XPFE. In addition, Error Vector Magnitude (EVM) results with/without RoF integration are included, which are always below 2.5%. Therefore, it

was experimentally validated that WS-WDM-PON+XPFE and RoF can coexist without any performance degradation. To support lower-layer splits in an efficient manner, innovative techniques were also evaluated. In fact, the mixed digital/analogue radio over fibre implementation allows sending up to 11.05 Gb/s (9 x CPRI 2) CPRI-equivalent bit-rate using less than 200 MHz bandwidth of an off-the-shelf optical transponder, which represents a remarkable spectral efficiency improvement compared to conventional CPRI. The system also allows reaching 36 km distance while still complying with the 3GPP EVM requirements.

We have integrated several transport technologies to a network transporting simultaneously backhaul and fronthaul traffic for both upper (option 2) and lower layer functional splits (options 6 and 8). They include mmWave, 5G-Crosshaul Circuit Switching Element (XCSE), and XPFE. In this setup, we demonstrated that the XCSE guarantees the required rates for all flows, independently of the functional split used (including the most demanding option 8 split). The compressed and packetized fronthaul traffic was also tested and transported over the XCSE, which generated fronthaul traffic of 262.4 Mbps. We also observed that the Fast Forward mmWave together with the XCSE can satisfy the one-way delay constraints imposed by the MAC-PHY split (the average round trip time (RTT) is 0.26 ms). This is also the case with high-load UDP traffic, where the RTT average value is 0.44 ms. The maximum RTT obtained with background traffic suggests that constraints are also satisfied if we take as reference the documents of the small cell forum, because the average of the maximum results is 1.27 ms, with a maximum of 1.5 ms. Other more stringent implementations and recommendations may not be fulfilled in all cases. In that case, prioritization of the fronthaul traffic over the backhaul traffic would reduce its impact, and so, lower layer splits would also be supported, as explained above for the WS-WDM-PON+XPFE setup.

Other functional tests done between the UE all the way through the mobile network (and the underlying 5G-Crosshaul transport) and up to an Iperf server resulted in the following measurements. The rates of the higher layer split under evaluation (option 2) is 18.28 Mbps in the downlink and 25.73 Mbps in the uplink; in other lower layer splits under evaluation (option 6), the rate is 67.65 Mbps in the downlink and 17.46 Mbps in the uplink; for the backhaul traffic, the resulting rate is 8.71 Mbps in the downlink and 10.37 Mbps in the uplink. The user equipment (UE) connected to the option 8 split remote radio unit (RRU) results in 45.504 Mbps in the downlink. These rates are achieved independently from whether there is simultaneous traffic of other UEs in the setup and show that the different functional splits are actually supported.

In a more focused setup, we demonstrated for a combination of mmWave and optical links (hybrid link) the feasibility to support higher-layer functional splits (option 2). The average latency values, depending on link technology of 0.5 ms to 1.5 ms are not suitable for lower layer splits, i.e., split options 5 to 8. These splits are within the hybrid automated repeat request (HARQ) loop and require shorter latencies. The split of the RLC, i.e., split option 4, is possible from a latency perspective, but requires higher bandwidth. Although the hybrid link provides more bandwidth than the individual ones, this is traded off by a latency increase. Therefore, these technologies can be used for splits between RLC and PDCP and for BH traffic only, i.e., split options 1 to 3 are supported.

In summary, the results confirm that the variety of technologies combined with their specific characteristics allow serving the needs of all RAN split options and contribute to fulfil the project and 5GPP KPIs and objectives. More specifically, they contribute to increase the number of devices handled by the network, to reduce its CAPEX and OPEX, to find the appropriate settings in the quest to reduce latencies below 1ms and to increase the global throughput of the network.

1 Introduction

Previous documents delivered by WP5 include extensive studies defining the integrated demonstrations of the 5G-Crosshaul project, which use the building blocks defined in WP2 (data planes), WP3 (control plane) and WP4 (application plane). Due to the high number of heterogeneous technologies and components to integrate in this work package, the integration plan defined four different demos:

- Demo 1: Energy management of Crosshaul
- Demo 2: Media distribution over the Crosshaul services
- Demo 3: Hierarchical multi-domain resource management of the Crosshaul
- Demo 4: Crosshaul fulfilment of the requirements of multiple RAN splits

These demos are deployed in the 5G-Crosshaul testbed, which is composed of four different sites: 5G-Berlin, 5TONIC-Madrid, CTTC-Barcelona and Taiwan. Additionally, other specific setups were carried out with other geographically distributed sites. Most of the demos are deployed in more than one site, and some of them use components available in different interconnected sites. For this reason, along this document we use the term 5G-Crosshaul testbed to identify the global platform where all the demos are deployed.

This document presents results of the four integrated demos defined in the project, running on top of the 5G-Crosshaul testbed. In every demo, we have identified several experiments, with the aim of providing feedback to all work packages of the project about the interaction between the different building blocks defined in such work packages (during project lifetime) and to evaluate their performance as far as relevant KPIs are concerned.

Each demo is presented in different sections. The introduction of each demo also provides the main results and conclusions of all the experiments run for that specific demonstration, hence offering the highlights of all the findings. Then, the demo setup is presented (including involved building blocks from other work packages and additional building blocks). After that, all demo subsections describe the experiments carried out in that demo. We have defined a common structure for all experiments, where the most important items are: a short paragraph describing the experiment, the related state of the art, improvement beyond the state of the art, related use case, measured KPIs, measurement procedure, and main results. All above items are provided in the form of a table at the beginning of each experiment. Finally, each experiment also has a subsection entitled *discussion of results* in which the results of extensive experimental evaluation campaigns for that specific experiment are presented in the form of tables and figures. The reasons behind the obtained results are discussed in detail.

2 Experimental evaluation results for energy management of the 5G-Crosshaul (Demo 1)

Demo 1 shows the 5G-Crosshaul features for (1) the unified monitoring of power consumption and (2) the optimization of energy efficiency in heterogeneous fronthaul/backhaul environments composed of XFE and XPU elements. The demo runs on different types of network technologies in order to demonstrate the applicability of the 5G-Crosshaul energy efficiency concepts to a wide set of infrastructure environments.

Demo 1 focuses on the following three aspects:

- Unified platform, based on NFV and SDN architecture, for power consumption monitoring of physical infrastructure composed of heterogeneous devices (on both network and computing domains), each of them exposing its own management interface. The monitoring features are also extended to virtual resources and virtual infrastructures, on a per-tenant basis
- Energy efficient resource allocation, in heterogeneous network domains (i.e. based on software based XPFEs, mmWave links and analogue RoF technologies), with automated selection of network nodes to be activated for establishing network connectivity and their power states regulation depending on the traffic load.
- Energy efficient provisioning of virtual infrastructures with VNFs Service Function Chains deployed on XPUs in the 5G-Crosshaul domain, with automated selection of target XPUs for VMs placement and related configuration of servers' power states.

The 5G-Crosshaul energy efficiency and power monitoring features over heterogeneous resources are implemented in the Energy Management and Monitoring Application (EMMA, see [50]), operating over the 5G-Crosshaul XCI as an enabler for the management and monitoring of different data plane technologies. In order to demonstrate the applicability of the 5G-Crosshaul EMMA solution over the XCI, the demonstration is evaluated in three different network technological domains: XPFEs, mmWave and RoF network data planes.

- **In the XPFE's network domain**, with reference to the analytical model [48], the power consumption of each node can be decomposed into a constant baseline component, strictly related to the hardware used, and a variable component, which depends on the traffic processed by the node. In particular, in the case of the 5G-Crosshaul testbed, for network devices under tests, experiments shown that the idle value is around 21W for the node's sleeping mode and 35W for the active mode. Results demonstrate how, in a limited network topology, with a total of 6 XPFEs, the on-demand provisioning of energy efficient network connections, i.e. with the automated regulation of the network nodes' power state, allows to save a total of 84W (when compared with the 210W of the "always on" approach) without connections (around 40% of power saving) and 42W (20%) when a first path is configured, activating three nodes. The variable component for 1Gbps traffic along the configured path is estimated around 500 mW, not heavily impacting the overall power consumption. The experiments evaluate also the EMMA capability in processing and aggregating monitoring data, giving the power consumption view for services and tenants.

The scalability of the EMMA algorithm is validated in an emulated reference environment that represents a real regional network in the North-West of Italy. The size of the emulated network is scaled out from 51 nodes up to 250 nodes, increasing the number of antennas connected to edge nodes up to 7485. With a maximum traffic loads, just 6 nodes can be maintained in sleeping mode, with a global power consumption saving of 12%, while the provisioning time average remains under the second with a maximum value of nearly 7 seconds (6.83 s).

In the case of the XPFEs network domain, we also extended the data plane to IT resources (XPU) for the energy efficient provisioning of Network Services. In this case, the on-demand provisioning of a vEPC NS, with the automated regulation of both XPFEs and XPU's power states and the setup of the end to end service connectivity, takes nearly 5 minutes, where the VNFs creation and configuration are the most demanding steps in terms of processing time.

- **In the RoF network domain**, where 12 nodes are deployed in a high-speed train testbed, the capability of the EMMA application of managing the nodes' power state brings a power savings of 78.6%, which is equivalent to power savings of approx. 17257 KJ per day. This is the case where the best results are achieved in terms of power consumption saving, since the particular scope of the high-speed train testbed foresees that RoF nodes are activated just the time for serving the passing train, which is around 130 secs for each trip.
- **In the mmWave mesh network domain**, a topology composed of 4 mmWave mesh nodes, with 2 WiGig modules each, is deployed for serving backhaul traffic. In line with results obtained in the XPFEs' domain, experiments show how the EMMA algorithms allows to save nearly 20% of power consumption for each mmWave node that can be kept in sleeping mode, i.e. disabling the power distribution to the node port.

2.1 Demo 1 setup

Demo 1 operates on a mixed physical infrastructure composed of XPU and three kinds of network technological domains, distributed across different geographical locations of the 5G-Crosshaul testbed. In particular, this demonstration targets the following three types of technologies developed in the context of WP2 and T3.1: (i) XPFEs software switch based on Lagopus¹, (ii) mmWave networks and (iii) Radio-over-Fibre (RoF) nodes deployed in the context of a railway infrastructure with high-speed trains.

Each technology is controlled through a dedicated SDN controller, included in 5G-Crosshaul XCI, acting as a child controller and being responsible for the configuration and management of a single technology domain. Demo 1 controllers follow the general architecture defined in [48] and implement the features to enable power consumption monitoring and management, like southbound plugins for energy data collection, analytics algorithms for computation of per-tenant and per-service power consumption, power state management components to switch on/off devices and energy efficient routing algorithms. The EMMA application, developed in WP4, runs on top of the controller providing the logic for resource allocation, service coordination and energy optimization, offering also a web-based graphical user interface for the system administrator. The detailed description of the building blocks is provided in the following subsection following a per-plane splitting.

2.1.1 Building blocks from other work packages

Data plane building blocks

WP2/ WP3 (T3.1)	
Component / Entity	Description

¹ <http://www.lagopus.org/>

Software based XPFEs	XPFEs based on Lagopus software (v0.29) installed on standard servers. The XPFEs are deployed in a ring topology of three nodes. They implement the XCF defined in T3.1 and support multi-tenancy through a dedicated OpenFlow pipeline. Moreover, they support monitoring of power consumption and configuration of three power states via Simple Network Management Protocol (SNMP).
mmWave mesh nodes	mmWave mesh nodes span up to three point-to-point connections to other nodes. In this way, Gbit/s wireless communications are offered. This is achieved by connecting the mmWave WiGig USB devices to an embedded minnow board and by bridging the interfaces within standard Linux Kernel in software. Each node consists of up to three Ethernet interfaces which can be controlled with either OpenvSwitch or any OpenFlow capable additional entity.
RoF nodes	Analogue-RoF (A-RoF) nodes are installed along the railway track, typically in the tunnels to overcome the signal blockage by the tunnels. A-RoF supports up to 16 optical channels on a single fibre, using the CWDM grid with 20 nm channel spacing in the 1270 – 1610 nm wavelength range. Typical bit rate spans from 1 Gbit/s to 100 Gbit/s per optical channel. Hence, the aggregate capacity of CWDM is 1600 Gbit/s. Management and monitoring of RoF nodes is supported via SNMP.

Control and application plane building blocks

WP3	
Component / Entity	Description
XCI MANO platform	MANO platform, with the VIM based on OpenStack (Mitaka version), extended to control the provisioning of vEPC instances in an energy efficient manner. The MANO component includes the following modules: <ul style="list-style-type: none"> - <i>NFVO</i>, in charge of orchestrating the provisioning of vEPC instances and related cloud resources. The NFVO has been developed from scratch and includes features for energy-aware resource allocation algorithms and automated power state configuration. - <i>vEPC VNFM</i>, developed from scratch, in charge of allocating vEPC VNFs based on the VM placement computed by the EMMA application and configuring the OpenAirInterface (OAI) EPC components. - <i>VIM</i>, in charge of managing the XPU virtual resources. It is based on OpenStack Mitaka and it is extended with features for XPU power consumption monitoring and power states changes (depending on hardware capabilities). OpenStack includes the following major services: Nova, Keystone, Horizon, Neutron, Ceilometer, Glance.
XPFE SDN controller	SDN controller based on OpenDaylight (ODL) and extended to control XPFE software switches and support energy monitoring and management

	<p>features. In particular, it includes the following software modules, developed as new bundles in ODL:</p> <ul style="list-style-type: none"> - <i>Provisioning Manager</i>, in charge of provisioning end-to-end connections assigned to different tenants over an XPFE domain. It instantiates flows based on the paths computed by a dedicated Path Computation module, triggers the automated re-configuration of the XPFEs power states and configures the flow rules interacting with the OpenFlow plugin. - <i>Path Computation Manager</i>, in charge of computing end-to-end paths based on energy efficient routing algorithms and the current configuration of power states in active XPFEs. - <i>Power State Manager</i>, responsible for the actual configuration of the XPFEs power states via SNMP protocol. - <i>Analytics Manager</i>, responsible of collecting power consumption information via SNMP protocol and computing per-tenant and per-service power consumption values based on traffic statistics and existing connection services.
mmWave controller	The mmWave controller is a standard ODL instance controlling all traffic flow, managing and providing a northbound interface to the application. EMMA handles directly the energy management features.
RoF controller	SDN controller based on standard ODL and extended to control RoF nodes to support the energy management features. An existing SNMP plugin of the ODL is extended to support the requirements of EMMA to communicate with the RoF nodes. The additional context information module, developed in the project, is also integrated with the ODL. It gathers the information related to train mobility to determine real-time location of the train. This information is retrieved by the EMMA application to determine when to switch ON/OFF the RoF nodes.
WP4	
Component / Entity	Description
EMMA for XPFEs	<p>SDN application providing a graphical interface and the logic to manage requests for energy efficient paths and visualize power consumption data per physical and virtual entities.</p> <p>In cooperation with the MANO orchestrator, the EMMA will also select the optimal placement of VNFs and trigger the automated change of XPU's power states.</p>
EMMA for mmWave	In mmWave domain, EMMA initializes all nodes, calculating and creating data paths through ODL and controlling power switches per node directly.
EMMA for RoF	In RoF domain, EMMA provides the logic to manage the Power Status of the RoF nodes. In particular, EMMA uses two modules:

	<ul style="list-style-type: none"> • <i>Statistics Module</i>, in charge of fast and reliable storage and retrieval of context information to provide it to the XCI (SDN controller). It also allows XCI to periodically retrieve new records from the database and updates XCI regarding the current location of the train. • <i>Management Module</i>, responsible for the actual control of RoF nodes, it decides whether to switch it ON or OFF via SNMP based on the received context information
--	--

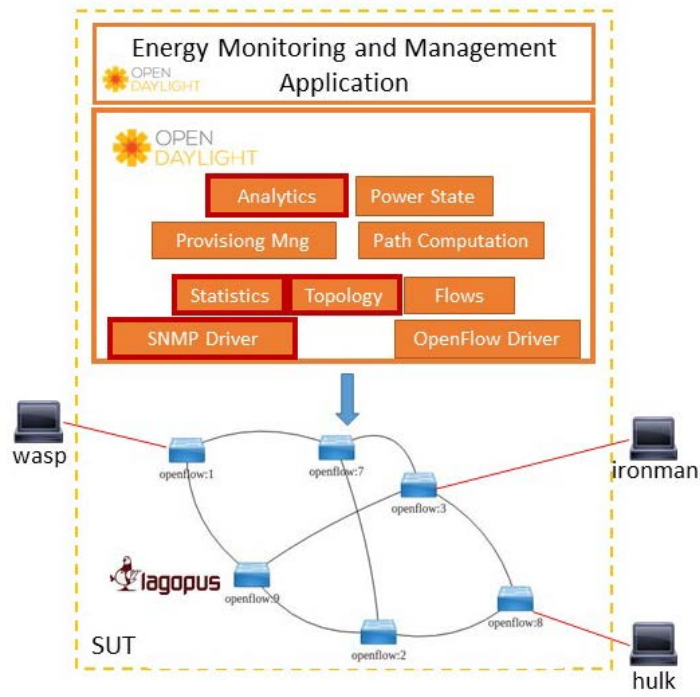
2.1.2 Additional building blocks

In addition to the previous elements, the following building blocks, which are not within the scope of technical development defined in 5G-Crosshaul activities, are required to support Demo 1.

Component/Entity	Description
Mobile UE	Customer Premise Equipment (CPE) is installed on train as the outbound gateway for train-to-ground communications. For RoF EMMA demonstration, the CPE is de facto the mobile UE to be served, and the status of RoF is determined based on CPE's real time mobility information.
Industrial PC (IPC)	IPC is installed on train to provide additional computation and storage capability. It stores all sorts of log data from CPE and parses and extracts the mobility information from the logs to push it into a cloud database.
Macro eNB	Macro eNBs are installed along the rail track to provide radio access. Each eNB is composed of a Baseband Unit (BBU) and a Remote Radio Unit (RRU).
Core Network	The Core Network of High-Speed Rail site is provided by the Evolved Packet Core (EPC) Agent server which possesses the function of MME, S-GW and P-GW.
Cloud database	The cloud database gathers real time mobility information from all running trains, and communicates with RoF EMMA to provide input to the RoF EMMA algorithm.
Traffic generator	Deployed in XPU nodes, attached to edge XPFEs and used to generate testing traffic on the XPFE domain (e.g. using traffic generator tools like Ostinato).
OpenEPC and FH/BH traffic emulator	Deployed in XPU nodes, FH/BH traffic emulator (i.e. respectively UE + RRU or UE + RRU +BBU) is used to generate the target traffic to be handled by the OpenEPC components (MME, BBU, SPGW, EPC-enabler).

2.2 Experiments on monitoring of power consumption in an XPFE domain

2.2.1 Experiment #1: Power Consumption Monitoring for XPFE physical nodes

Power Consumption Monitoring for XPFE physical nodes	
Description	This test shows the capability of the EMMA graphical interface to show the power state of XPFEs and their power consumption based on current traffic through a unified web-based platform. In this test, the power consumption of network nodes changes based on the active traffic flows since the power state is automatically regulated based on the current allocated network resources.
State of the art	Power consumption monitoring is currently available for physical devices through proprietary management systems and interfaces.
Improvement from State of the art	This demonstration shows a unified platform, based on an SDN controller, for power consumption monitoring of infrastructures composed of multiple devices, each of them exposing their own management interface. In this experiment we adopt the SNMP protocol and an EMAN (Energy MANAGEMENT) MIB (Management Information Base) to collect monitoring data from each XPFEs.
5G-Crosshaul Use Cases	Dense Urban Society
System under Test	 <p>The following components of the XCI are relevant for this test:</p> <ul style="list-style-type: none"> - SNMP driver - Statistics Manager

	<ul style="list-style-type: none"> - Topology Manager - Analytics Manager
Project Objectives / 5GPPP KPIs addressed	Obj. 7: Reduce energy consumption in the Crosshaul by 30% through energy management (the function shown in this experiment is an enabler for this objective).
Measured KPIs	Monitoring and accounting: type of monitoring information collected at different architectural layers (functional). In this case monitoring data is power consumption of physical nodes and entire physical infrastructures.
Measurement tools	<p>Wireshark² to check the control plane traffic (SNMP messages) and verify its consistency with the information visualized in the GUI.</p> <p>Ostinato³ network traffic generator to transmit 1Gbps of traffic customized with the proper classifiers, as expected on the configured paths.</p>
Measurement procedure	<ol style="list-style-type: none"> 1. Visualize power consumption data for network nodes from EMMA GUI (not active nodes). 2. Configure static flow rules in Lagopus to enable a path between two servers. 3. Visualize power consumption data for network nodes from EMMA GUI (active nodes without traffic). 4. Generate traffic between the servers. 5. Visualize power consumption data for network nodes from EMMA GUI (active nodes with traffic). 6. In order to verify the exchange of control plane messages between data plane, controller and EMMA application, capture packet with Wireshark and check the consistency with the graphs shown in the GUI.
Constraints	N.A.
Main results	Power consumption monitoring with XPFE in sleeping and active states. The results show that the power consumption of a node can be decomposed into (i) a constant baseline component and (ii) a traffic load dependent component. For the specific XPFE used, the constant component is around 21W in sleeping mode and 35W in active mode, while the variable component due to the traffic load is very low, in the order of hundreds of mW.

² <http://www.wireshark.org/>

³ <http://ostinato.org/>

Discussion of results

The objective of this test is to verify the power consumption monitoring functionalities of the EMMA platform in the 5G-Crosshaul testbed environment, composed of 6 XPFEs nodes. During the test, we progressively instantiate three network connections, shown in Figure 2. These connections are detailed in Table 1, with information about the network path traversed by each of them and the tenant they are associated to. In order to verify the sensitivity of power consumption to the traffic load, we generate traffic between the hosts at the edge of the network using the Ostinato network traffic generator.

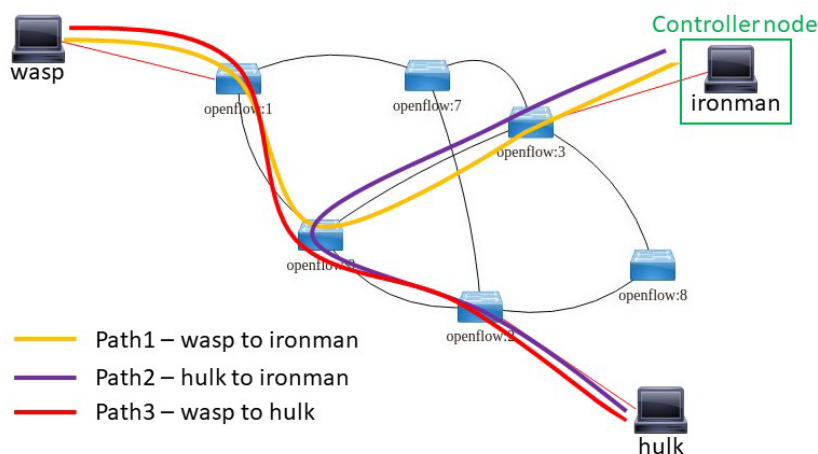


Figure 2. Test scenario and instantiated paths for EMMA experiments on XPFEs domain

Table 1: Details of the paths installed for EMMA tests on XPFEs domain

Path	Source	Destination	Traversed nodes	Tenant
Path 1	wasp	ironman	XPFE 1 – XPFE 9 – XPFE 3	Tenant 1
Path 2	hulk	ironman	XPFE 2 – XPFE 9 – XPFE 3	Tenant 2
Path 3	wasp	hulk	XPFE 1 – XPFE 9 – XPFE 2	Tenant 1

We retrieve the values of power consumption for single nodes and for the global infrastructure using the REST APIs of the Analytics module developed as part of the EMMA application. An example of JSON message received as reply to a single node power consumption query is shown in Example 1.

```
{ "2017-11-21T16:08:00.995Z": 35.01922753065372,
  "2017-11-21T16:08:10.978Z": 35.025438005480986,
  "2017-11-21T16:08:20.977Z": 35.00241015902309,
  "2017-11-21T16:08:30.996Z": 34.98861553979634,
  "2017-11-21T16:08:41.044Z": 34.96322543104032,
```

```
"2017-11-21T16:08:50.938Z": 35.02874351765131,
"2017-11-21T16:09:00.982Z": 34.97283306917892,
"2017-11-21T16:09:11.078Z": 34.97873491686769,
"2017-11-21T16:09:20.992Z": 34.964742608864874,
"2017-11-21T16:09:30.982Z": 35.009160443663134 }
```

Example 1. JSON content with power consumption data from XPFE 1

As we can see, the EMMA application, through the Analytics component, is able to abstract the specific underlying protocols used at the SDN controller to interact with the devices. In fact, the Analytics module exposes a unified REST-based interface which allows other upper layer applications to retrieve power consumption measurements without the need to implement protocol specific drivers.

In our case, the SouthBound Interface (SBI) protocol used within the controller to retrieve power consumption data directly from the XPFEs is the SNMP protocol, using the EMAN MIB [1]. An example of SNMP message exchanged between controller and XPFE to retrieve a single measurement is shown in Figure 3 (the message is captured using the Wireshark tool).

```
▶ Frame 128: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits) on interface 0
▶ Ethernet II, Src: JetwayIn_c9:ef:cf (00:30:18:c9:ef:cf), Dst: RealtekU_41:fe:45 (52:54:00:41:fe:45)
▶ Internet Protocol Version 4, Src: 10.5.1.21, Dst: 10.5.1.51
▶ User Datagram Protocol, Src Port: 161, Dst Port: 43461
▼ Simple Network Management Protocol
  version: v2c (1)
  community: private
  ▼ data: get-response (2)
    ▼ get-response
      request-id: 2084013689
      error-status: noError (0)
      error-index: 0
      ▼ variable-bindings: 3 items
        ▶ 1.3.6.1.2.1.229.1.2.1.1.1: 35495393
        ▶ 1.3.6.1.2.1.229.1.2.1.3.1: -6
        ▶ 1.3.6.1.2.1.229.1.2.1.9.1: 1036
```

Figure 3. SNMP message between controller and XPFE to retrieve a power consumption measurement

Table 2 reports the power consumption data collected during the test execution. For each status of the infrastructure, we have considered 10 measurements taken every 3 seconds and computed the average. The whole experiment has been repeated 10 times, extracting average values from these series of data. In the following, we also present some screenshots from the EMMA application, which simplify the analysis of the data for the user.

Table 2: Average power consumption measurement for physical nodes (Watt)

Average power consumption in Watt for physical nodes and network							
Scenarios	XPFE 1	XPFE 2	XPFE 3	XPFE 7	XPFE 8	XPFE 9	Overall network
All nodes off	21.0299	21.0281	21.02183	20.9614	20.9678	21.0173	125.9841
Path 1 on	35.0349	20.9888	35.0096	21.0168	20.9670	34.9968	167.9023

No traffic							
Path 1 on With traffic	35.4876	21.0455	35.6104	20.9996	21.0180	35.4558	168.2835
Path 1 & 2 on No traffic	35.0309	34.9947	34.9646	20.9988	20.9869	34.9846	181.9242
Path 1 & 2 on With traffic	35.2771	35.2135	35.6386	21.0171	20.9685	35.2783	182.1381
Path 1, 2 & 3 on No traffic	34.9826	34.9894	34.9645	20.9861	20.9693	35.0037	181.9220
Path 1, 2 & 3 on With traffic	35.6249	35.7221	35.5753	21.0042	20.9898	35.5849	183.6007

At the beginning of the test, all the nodes are in sleeping mode (Figure 4), and the power consumption of each node is around 21W, with the whole network consuming around 126W (the power consumption of the whole infrastructure is around 286W considering also two XPU in sleeping mode, i.e., consuming 80W each, see Figure 4 and Figure 5). When Path 1 is established, XPFE 1, XPFE 9 and XPFE 3 are automatically moved to the active power state (Figure 6) and their power consumption reaches around 35W each.

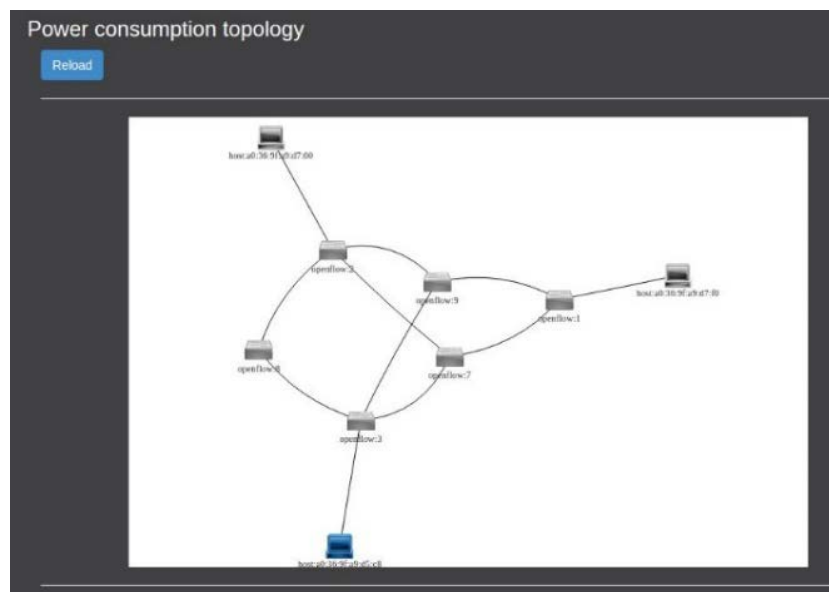


Figure 4. XPFEs and XPUs in sleeping mode (EMMA GUI)

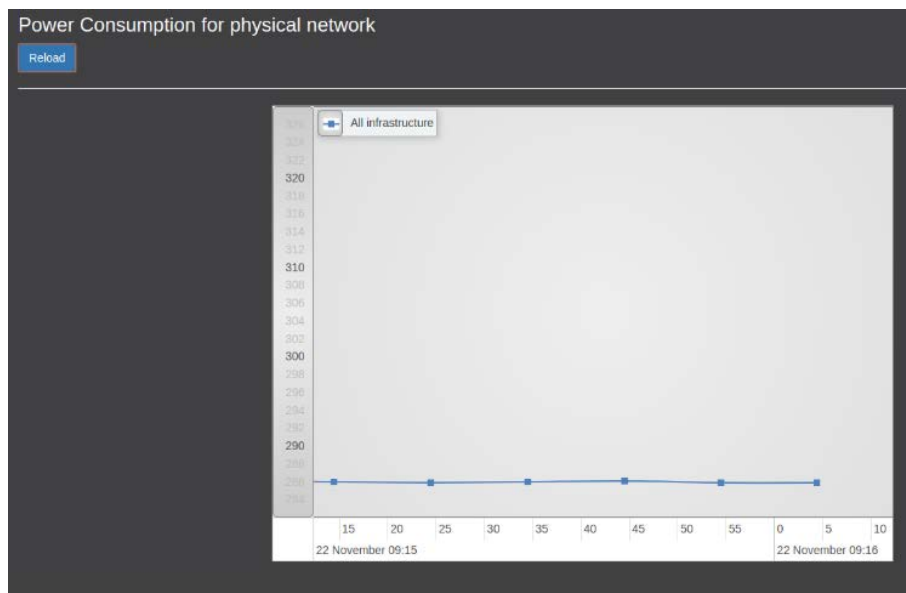


Figure 5. Network power consumption: all XPFEs and XPU in sleeping mode (EMMA GUI)

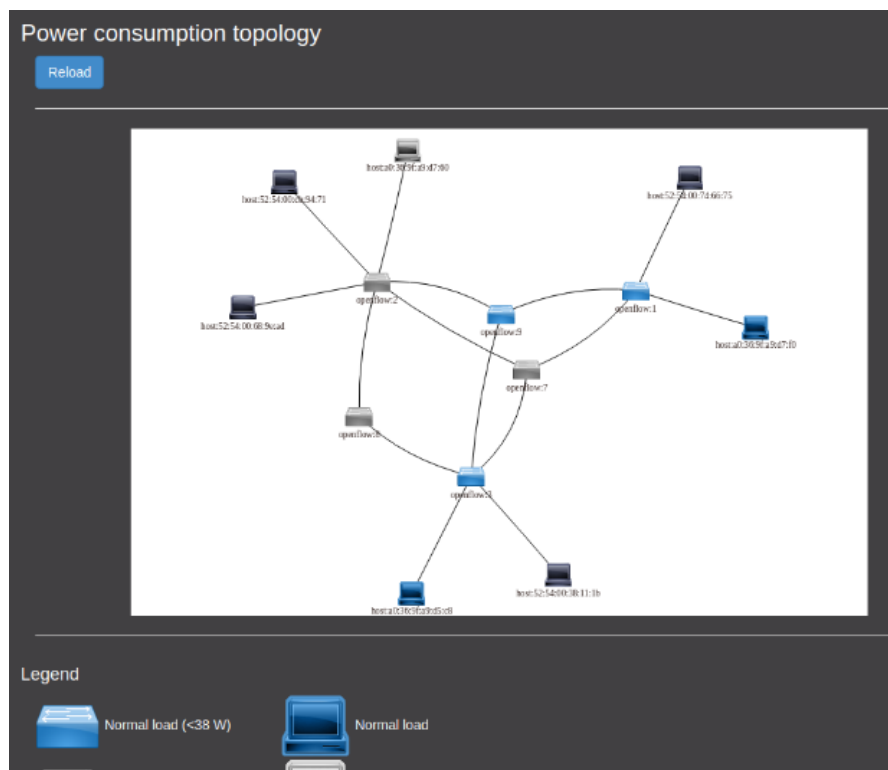


Figure 6. Active XPFEs and XPU for Path 1 (EMMA GUI)

Generating traffic on path 1 (in the order of 1 Gbps), power consumption slightly increases (see Figure 7). With reference to the analytical model we are assuming (see [48] for further details), where the power consumption of a node can be decomposed into (i) a constant baseline component and (ii) a traffic load dependent component, we can conclude that the sensitivity of power consumption to the real-time traffic load is very low. In other terms, at least for the medium amounts of traffic we are able to generate on our testbed, the variable component of power

consumption, i.e. the one associated only to traffic load and thus to tenants' usage of resources, can be considered nearly negligible if compared to its static component. It should be noted that this particular proportion between variable and static component of the node's power consumption is specific of the network technology we are considering - in our case an XPFE is a software switch installed in a general purpose machine - and may be different for other types of packet switching devices. However, since the major component of power consumption is the static one and this is determined by the specific power state of the network node, an efficient way to achieve a good level of power saving is to automatically adapt the nodes' power state to the real-time traffic demands. This is one of the features implemented in EMMA and it will be validated in the following experiments (see Experiment #4 and Experiment #5).

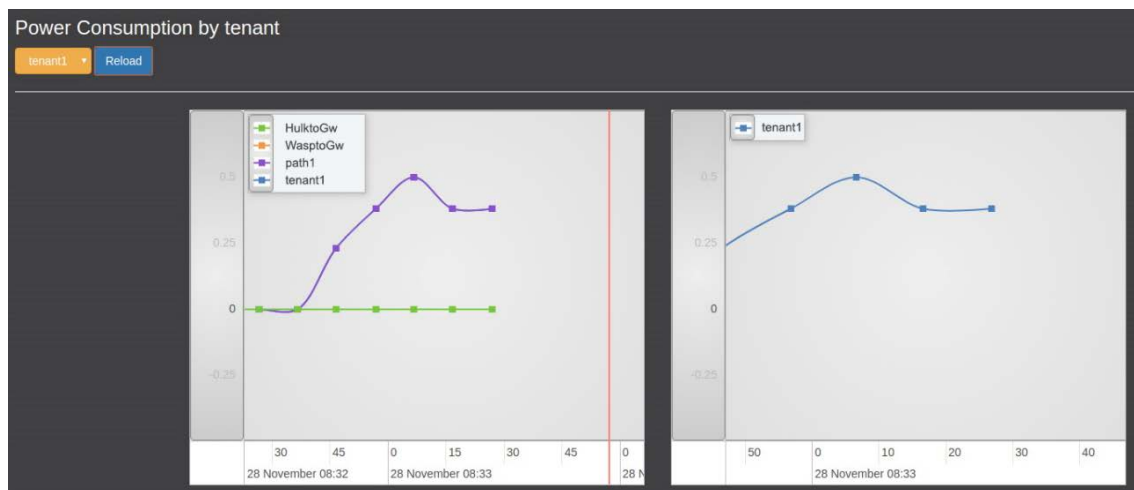


Figure 7. Power consumption slightly increases with traffic on path 1 (Wasp to Ironman)

Path 2 setup triggers the activation of a further XPFE (i.e. XPFE 2 – see Figure 8). For this reason, when both Path 1 and Path 2 are established, the power consumption of the whole network increases to around 182W (the whole infrastructure power consumption is around 382W considering also two active XPU, see Figure 8 and Figure 9) due to the change in the constant power consumption component on XPFE 2 from around 21W to around 35W (these values depend on the specific hardware where the Lagopus software is installed). Again, the generation of traffic does not increase heavily the power consumption. The power consumption of the XPFEs is independent of the actual traffic amount due to the Lagopus software actively polling for received packets. Figure 10 shows the power consumption for XPFE 1 in active mode while traffic generated by tenant1 is traversing the node: the traffic component associated with the tenant is in the order of 400 mW and doesn't increase heavily the overall nodes power consumption with respect to the node's active status.

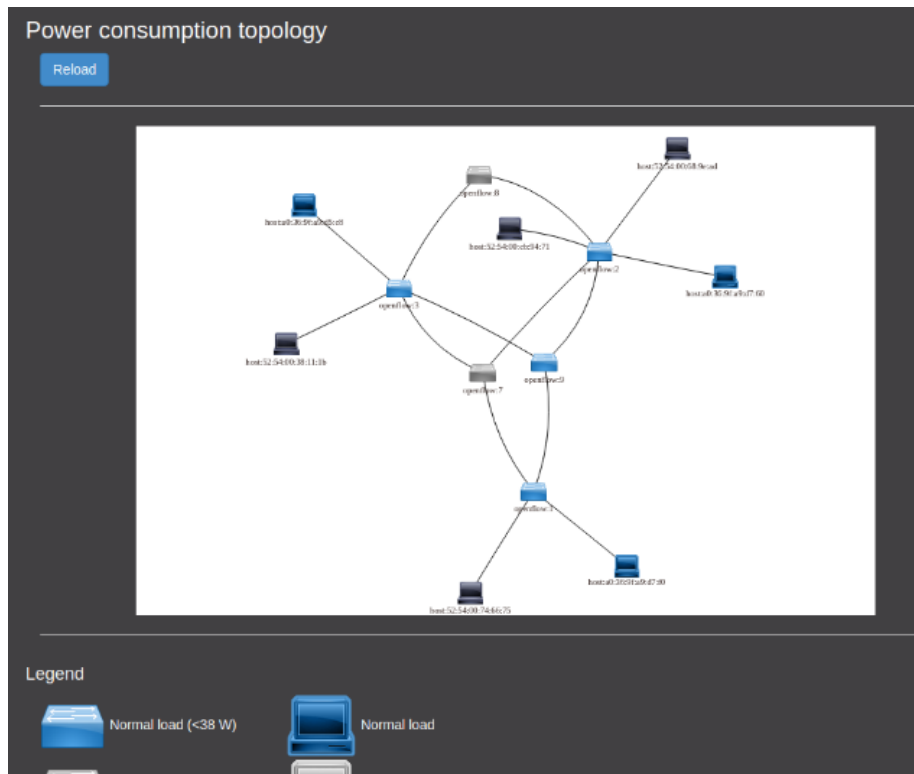


Figure 8. Active XPFEs for Path 1 and Path 2 (EMMA GUI)

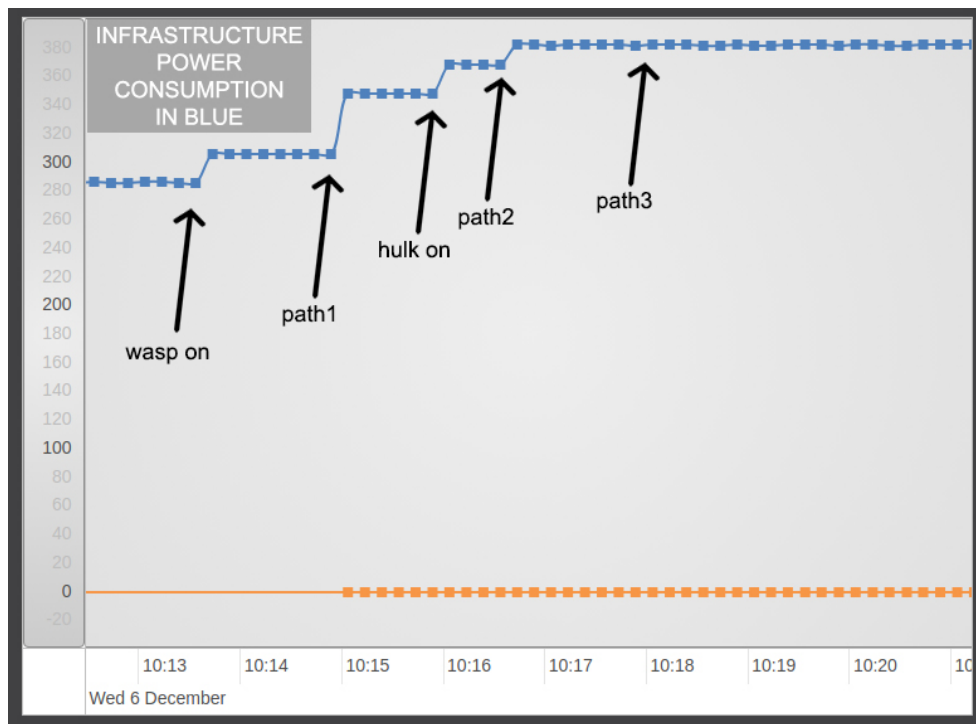


Figure 9. Evolution of network power consumption during setup of Path 1 and Path 2 (EMMA GUI)

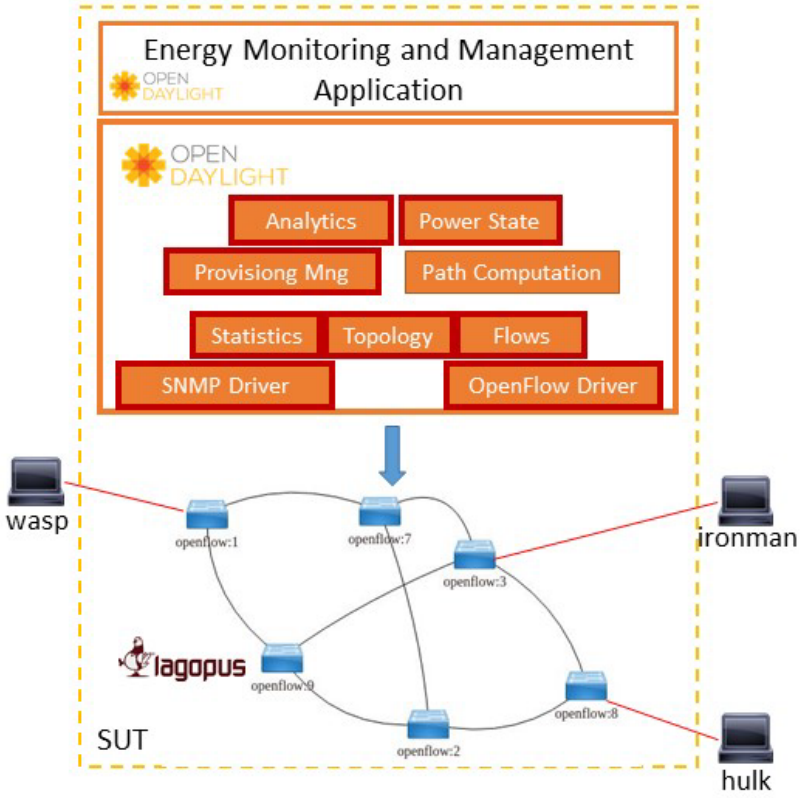


Figure 10. Power consumption with traffic on single XPFEs (EMMA GUI)

Finally, at 10:18:01.514 (as reported in the SDN Controller log) we set up also path 3. In this case, since all the XPFEs are already up and running, the provisioning procedure does not trigger any further node activation and power consumption does not increase, as shown in Figure 9. As expected, in a network with high traffic when all or most of the network nodes are processing traffic, there is low margin for power saving since it is not possible to de-activate nodes.

2.2.2 Experiment #2: Power consumption monitoring for single network paths and tenants

Power Consumption Monitoring for single network paths and tenants	
Description	This test shows the capability of the EMMA application to compute the quota of power consumption associated to specific network paths and to the connection services provisioned for a given tenant. These quotas are computed based on a mathematical model which takes into account the current traffic load of connection-specific flows.
State of the art	Power consumption monitoring is currently available for physical devices through proprietary management systems and interfaces.
Improvement from State of the art	This demonstration extends power consumption monitoring to virtualized entities on a per-tenant basis.
5G-Crosshaul Use Cases	Dense Urban Society

<p>System under Test</p>	 <p>The following components of the XCI are relevant for this test:</p> <ul style="list-style-type: none"> - SNMP driver - OpenFlow driver - Statistics Manager - Topology Manager - Flow Manager - Analytics Manager - Provisioning Manager - Power State Manager
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 7: Reduce energy consumption in the crosshaul by 30% through energy management (the function shown in this experiment is an enabler for this objective)</p>
<p>Measured KPIs</p>	<p>Monitoring and accounting: type of monitoring information collected at different architectural layers (functional). In this case monitoring data are power consumption of physical nodes and entire physical infrastructures.</p> <p>Infrastructure virtualization: monitoring of power consumption per virtual infrastructure (functional).</p> <p>Mapping of virtual to physical resource synchronization: mapping of physical resource power consumption on virtual resource power consumption based on per-tenant resource usage (functional).</p>

Measurement tools	Wireshark Ostinato
Measurement procedure	<ol style="list-style-type: none"> 1. Visualize power consumption data for network nodes from EMMA GUI (not active nodes, no connections established). 2. From EMMA GUI, setup a new connection between two servers for tenant 1. 3. Visualize power consumption data for network nodes, for network connections and for tenants from EMMA GUI (active nodes without traffic). 4. Generate traffic between the servers. 5. Visualize power consumption data for network nodes, for network connections and for tenants from EMMA GUI (active nodes with traffic). 6. Repeat the previous steps for a new network connection associated to tenant 1 and an additional network connection associated to tenant 2. 7. In order to verify the exchange of control plane messages between data plane, controller and EMMA application, capture packet with Wireshark and check the consistency with the graphs shown in the GUI.
Constraints	The accuracy of theoretical energy models should be further validated with real data.
Main Results	The variable component of power consumption for a network path crossing two nodes (i.e. the component due to the traffic load) is in the order of 450 – 500 mW for 1 Gbps traffic. This component is very low compared with the constant component due to power state of the XPFE (around 21 and 35W in sleeping and active mode respectively). It should be noted that, in these experiments, these values for the network paths are computed by the analytics algorithm of the EMMA application based on the traffic statistics and the established path, while in the previous experiment power consumption data was collected directly from the network devices and referred to physical entities, without any classification in terms of paths or tenants.

Discussion of results

As analysed in the previous experiment, the power consumption of network nodes can be split in a constant component due to the power state of the node and a variable traffic load-dependent component which depends on the actual usage of the allocated resources from the tenant who has requested the connection. While in the previous experiment we have analysed the EMMA capability to monitor power consumption for physical nodes and infrastructures, in this experiment the objective is to verify the EMMA analytical processing for computing the variable component of power consumption taking into account the traffic statistics collected at the controller. In particular, this variable component is mapped to specific connections and tenants at the Analytics module of the EMMA application. As for the previous experiments, monitoring

data are retrieved using the Analytics REST APIs and we show some graphs from the EMMA web interface.

The execution of the experiment follows the same procedure of the previous one: we establish the three paths defined in Table 1 and, once they are up and running, we generate traffic between the hosts using the Ostinato tool. This time, however, we are interested in the power consumption associated to single connections (Path 1, Path 2 and Path 3) and single tenants (Tenant 1 and Tenant 2), i.e. the variable component dependent on the traffic load.

Table 3 reports the power consumption data collected during the test execution, following the same measurement procedure described for the previous experiment.

Table 3: Power consumption elaborated by EMMA application for virtual entities (paths and tenants) – 1 Gbps of traffic

Infrastructure status	Power consumption (mW)				
	Path 1	Path 2	Path 3	Tenant 1	Tenant 2
All nodes off	0	0	0	0	0
Path 1 on No traffic	0	0	0	0	0
Path 1 on With traffic	459.5	0	0	459.5	0
Path 1 & 2 on No traffic	0	0	0	0	0
Path 1 & 2 on With traffic	483.4	464.8	0	483.4	464.8
Path 1, 2 & 3 on No traffic	0	0	0	0	0
Path 1, 2 & 3 on With traffic	461.3	495.4	452.2	913.5	495.4

As expected, when the network nodes are in sleeping mode or when no traffic is exchanged among the hosts, the power consumption component associated to the different paths or tenants is zero. Once a path is activated, the associated power consumption is elaborated at the Analytics component: the resulting values are based on the traffic statistics received at the controller's OpenFlow plugin for the specific flows and assuming a linear relationship between the two variables (this model is in line with the measurements performed during the previous experiment).

The power consumption associated to each path, with 1 Gbps of nominal traffic generated through Ostinato, is in the order of 450-500 mW, and the EMMA application is able to map this variable component to the right tenant. Modifying the throughput of the generated traffic, power consumption varies accordingly. For example, in Figure 11 we show the power consumption for tenant 1, as reported in the EMMA web interface, when its two paths are up and carrying a different amount of traffic. In particular, the traffic generator throughput is initially set to 100

Mbps for both Path 1 and Path 3 and then increased up to 300 Mbps for Path 3. At the beginning the power consumption for Path 1 and 3 is around 50mW for each of them and when traffic is increased on Path 3, it scales up to around 100 mW. At the tenant level, the power consumption is given by the sum of the contributions of the single connections.

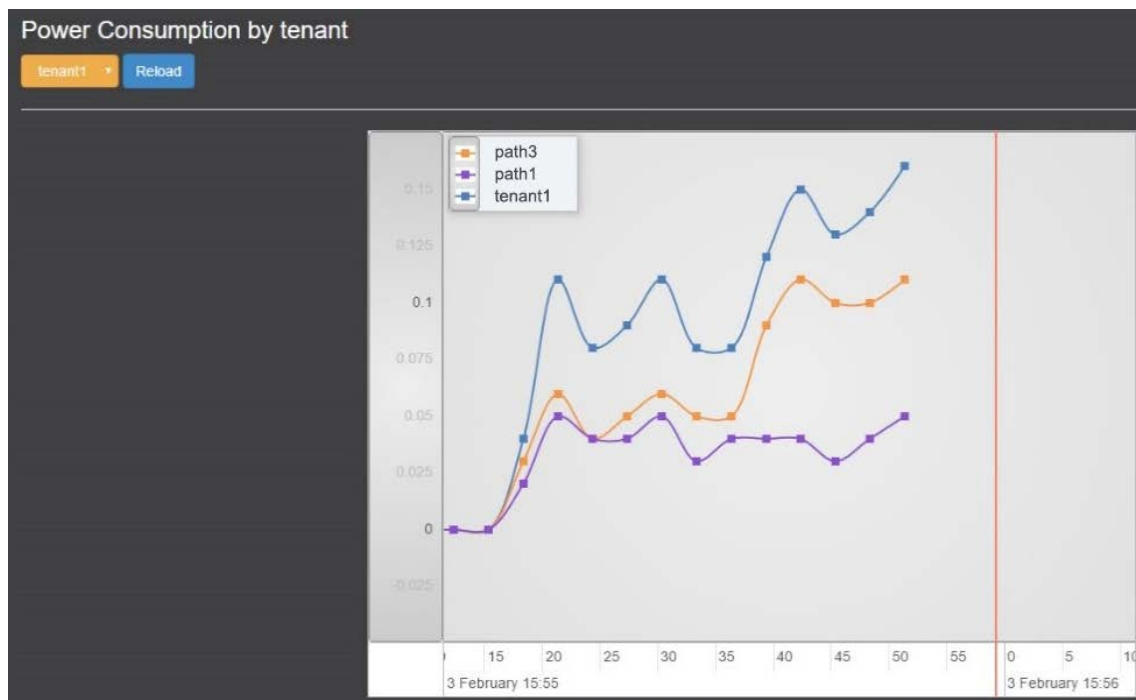
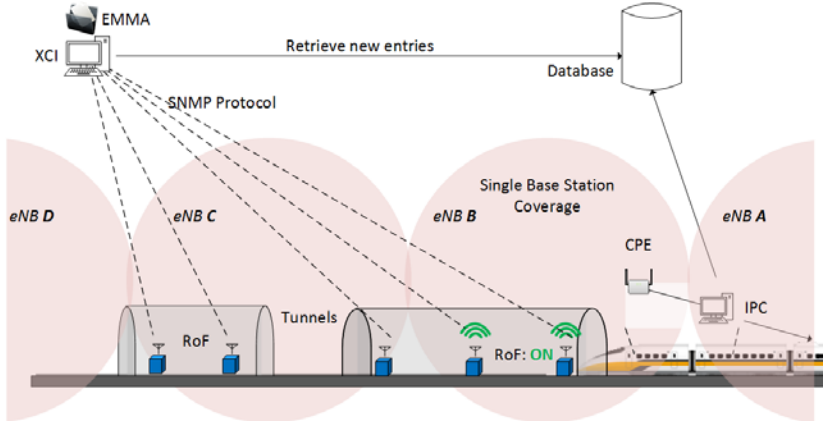


Figure 11. Power consumption associated to tenant 1, with split for Path 1 and Path 3

2.3 Experiments on energy-oriented network resource management

2.3.1 Experiment #3: Energy-oriented network resource management in RoF domains

Energy oriented network resource management in RoF domains	
Description	This experiment shows the capability of the EMMA to save the power of RoF nodes deployed in high-speed rail testbed via the unified XCI. Also, it indicates the capability of the EMMA graphical interface to show the current location of the High-Speed Train in the testbed and, in parallel, it tracks the current energy consumption per RoF node. In this experiment, EMMA switches ON/OFF the power status of deployed RoF nodes based on the train's mobility. An ODL application is developed to act as an SNMP manager and interacts with RoF nodes. In summary, EMMA utilizes the integrated fronthaul/backhaul architecture of Crosshaul to save the energy efficiently, which is otherwise constantly consumed by the deployed RoF nodes along the high-speed rail.
State of the art	In operational environments, RoF nodes are always active to give permanent coverage for the tunnels whether trains are present or not in the tunnel. The current trend of network research is moving toward an "always available" approach in place of "always on" approach for improved energy efficiency [13].

<p>Improvement from State of the art</p>	<p>Minimize the energy consumption of the overall physical network devices, subject to the communication QoS constraints, by selecting the RoF nodes to be turned on or turned off based on the high-speed train mobility (location, direction, speed, etc.)</p>
<p>5G-Crosshaul Use Cases</p>	<p>High-speed Train</p>
<p>System under Test</p>	<p>The following components are used in the system test, as shown in :</p> <ul style="list-style-type: none"> • EMMA application consisting of Management, Statistics and Context information module, sitting on top of ODL-based SDN Controller within the XCI, and using the XCI’s REST APIs • Extended ODL-based SDN Controller integrating SNMP Driver • RoF nodes <p>To complete the test, we also need the following components to provide context information, i.e. physical cell ID (PCI) to represent train mobility:</p> <ul style="list-style-type: none"> • CPE (customer premises equipment) which accesses roadside macro eNBs. • IPC (industrial PC) which can be any computer connecting to CPE’s control interface to save and process the logs from CPE. After processing, it pushes the PCI information extracted to the cloud database.  <p style="text-align: center;">Figure 12. System under Test</p>
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Energy efficiency – 30% reduction Obj. 8: Infrastructure reconfiguration for energy efficiency</p>
<p>Measured KPIs</p>	<p>Energy reduction using EMMA application</p>

<p>Measurement tools</p>	<ul style="list-style-type: none"> • Python scripts are used to compute the results related to energy consumption of RoF nodes. • OpenDaylight Dlux graphical user interface to monitor the current status of the RoF nodes. • Wireshark to check the packet size • Wireshark to check the latency of the packets sent from high-speed train to the controller
<p>Measurement procedure</p>	<p>For KPIs related to energy reduction, monitoring and measurement data are collected from the field trial. EMMA measures the runtime data related to the time where RoF nodes are switched on and provides the current reading of energy consumed by the nodes. Data, w.r.t energy consumption of the RoF nodes, are collected from each single trip within a whole day. Then, the collected information is further analysed to plot the overall EMMA chart.</p>
<p>Constraints</p>	<p>Meaningful results related to the energy savings of the RoF node can be obtained by evaluating the amount of time RoF nodes are switched OFF, then converting the values into energy indicator.</p>
<p>Main results</p>	<p>Energy consumption saving of 78.6%, which is equivalent to energy savings of approx. 17257 KJ per day.</p>

Discussion of results

The experiment demonstrates the reduction in energy consumption of the RoF nodes deployed along the high-speed train testbed. In the testbed, 12 A-RoF nodes are deployed along the railway track between Hsinchu station and Tayouan station, especially in the two tunnels (Litoushan tunnel and Hukou tunnel), where they provide better coverage to the passengers of the train as shown in Figure 12, where the highlighted circles represent different cells.

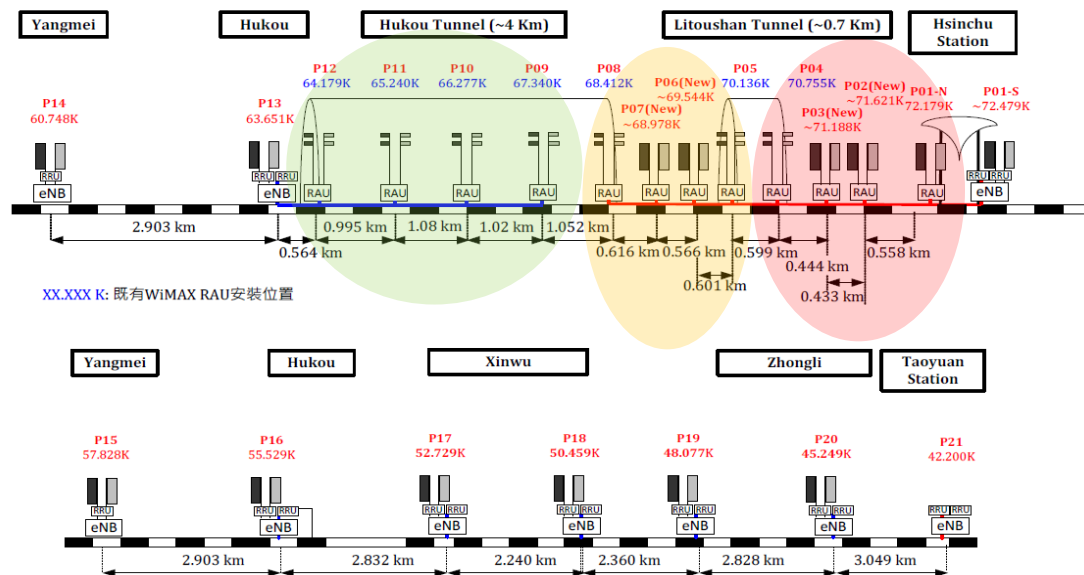


Figure 13. RoF deployment in the High-Speed Train testbed

The Field Trial experiment evaluates the performance of the EMMA application in terms of reduction in RoF nodes energy consumption.

EMMA on-board module is deployed in a single train, and then the experiments were performed repeatedly to quantify the obtained results. In the calculation of energy consumption of the experiments, both the power consumption of RoF node and supporting Cubie Board are considered. The Cubie Board is a single-board computer, currently using Linux operating system. Following are the power consumption values for A-RoF and Cubie Board.

- A-RoF 2x2 MIMO: 19.8 watt
- Cubie Board: 1.375 watt

In this experiment, our intention is to show the reduction in energy of A-RoF node without switching off the Cubie Board, that acts as an SNMP client and manages the A-RoF nodes based on the received signals. The Cubie Board is always on in our tests and always consumes energy.

The high-speed train operates daily between 7 AM and 11 PM. RoF nodes are always switched on without using EMMA, regardless of operational time of the train. Figure 14 illustrates the comparison of energy consumption of RoF nodes with and without EMMA only in the operational time of the train. The testbed comprises of 12 RoF nodes, out of which only 9 RoF nodes are in operational state in the Field Trial. Figure 14 illustrates the energy comparison chart with and without EMMA. The X-axis and Y-axis represent the time of day by hour increments and the energy consumption per hour in percentage, respectively. With EMMA, RoF nodes are switched on only to serve the high-speed train when it is approaching, saving significant energy. In the Field Trial, we observed that it takes approx. 130 seconds for RoF nodes to serve a moving train, which is equivalent to the energy consumption of 24.8 KJ, based on the calculation $E=P \cdot t$ (Energy = power x time). Obtained results highlighted in Table 4 demonstrate the improvement in energy by 78.6%, which is the percentage of total energy consumed with and without EMMA. 78.6% is equivalent to energy savings of 17257 KJ per day, which is the total of energy consumption with EMMA as highlighted below.

Table 4: Details of the energy consumption of RoF nodes

Number of Train Trips (Southbound + Northbound)	Time	Energy consumption without EMMA (KJ)	Energy consumption with EMMA (KJ)
12	7:00	914.76	297.30
12	8:00	914.76	297.30
10	9:00	914.76	247.75
11	10:00	914.76	272.52
10	11:00	914.76	247.75
8	12:00	914.76	198.20
7	13:00	914.76	173.42
9	14:00	914.76	223.00

10	15:00	914.76	247.75
11	16:00	914.76	272.52
13	17:00	914.76	322.07
12	18:00	914.76	297.30
13	19:00	914.76	322.07
12	20:00	914.76	297.30
11	21:00	914.76	272.52
10	22:00	914.76	247.75
6	23:00	914.76	148.65
0	24:00	914.76	44.5
0	0:00	914.76	44.5
0	01:00	914.76	44.5
0	02:00	914.76	44.5
0	03:00	914.76	44.5
0	04:00	914.76	44.5
0	05:00	914.76	44.5
0	06:00	914.76	44.5

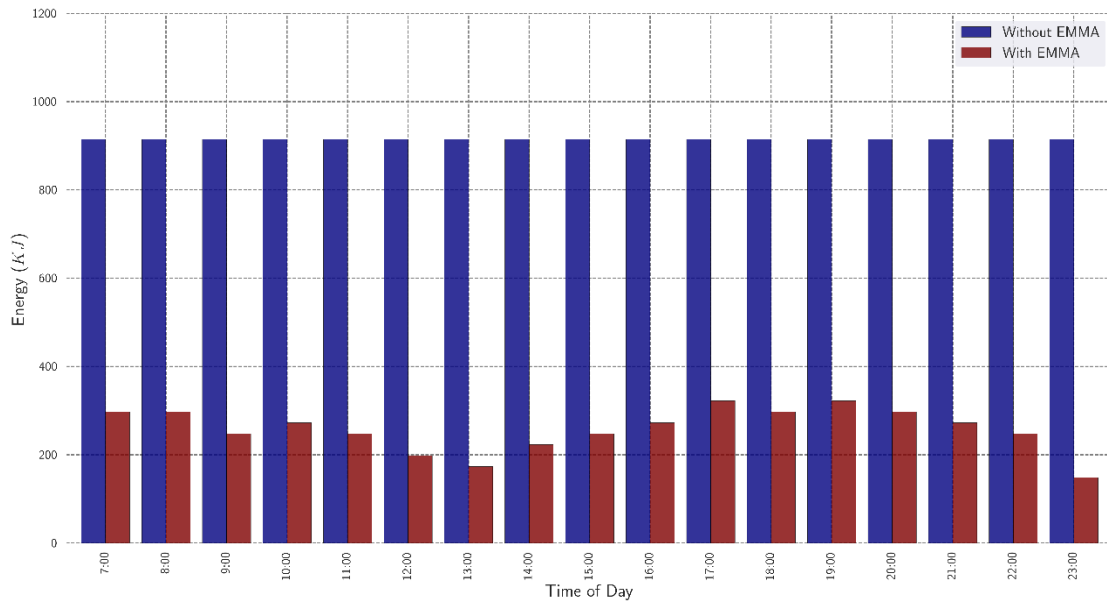


Figure 14. Energy consumption comparison chart

Two more additional issues have been considered while implementing EMMA, namely (1) latency to modify the status of network, (2) bandwidth consumption for related propagation in the system. The latency can be broken into the following components:

- t₁: IPC processing time
- t₂: propagation time from train to the XCI
- t₃: time taken from the XCI ODL-based SDN Controller in sending commands to RoF nodes, until the command is executed

In Figure 15, t₁ (IPC processing time), which is mostly around 1 ms and no more than 5 ms is shown.

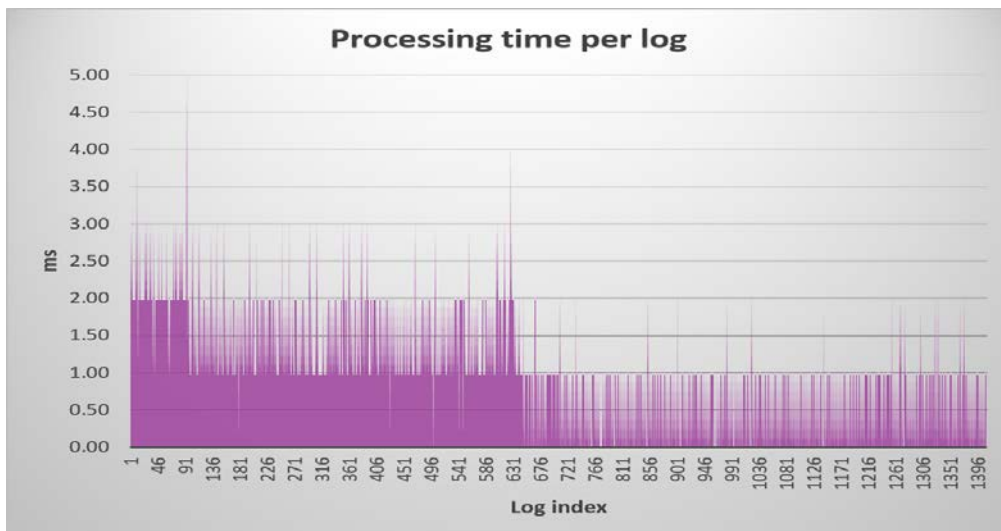


Figure 15. Log processing time for IPC

Figure 16 (t₂) presents the Round Trip time to send the context information from the moving high-speed train to the controller in the lab. It takes on average of 8 ms, and a variant ±9 ms.

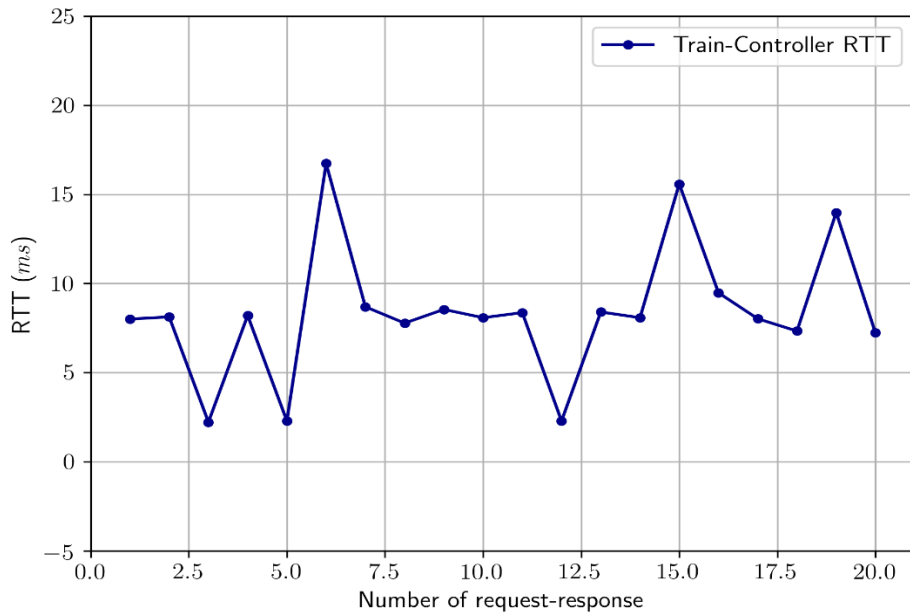


Figure 16. Round Trip Time to send context information from moving train to the controller

Figure 17 (t_3) presents the Round Trip time to switch ON/OFF the RoF nodes, where it can be seen that it takes, on average, 1.5 ms and no more than 4 ms. It is important to note that peaks do not represent network delay but are combined with the processing delay in few of the RoF nodes due to the faulty SNMP agent.

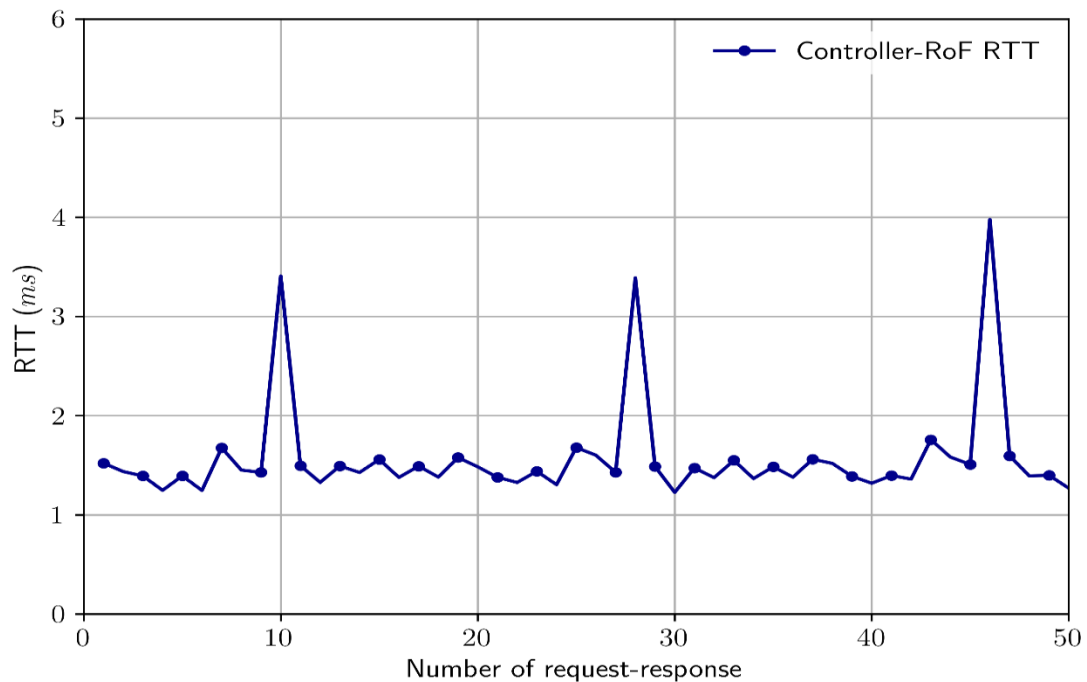


Figure 17. Round Trip time to Switch ON/OFF RoF nodes

For bandwidth consumption, we consider the packets sent via Advanced Messaging Queuing Protocol (AMQP), that consumes part of the bandwidth of the moving backhaul from train to ground. From Table 5 around 1.2 KB and 0.8 KB are transmitted from producer and consumer

respectively during initialization of AMQP connection, and afterwards only 0.2 KB are required to push packets regularly from Producer (i.e. IPC) to Consumer (i.e. DB). On average 8 packets/seconds are sent on a 60 Mbps link

Table 5: Packet size of AMQP messages

Packet and size from Producer (bytes)		(direction)	Packet and size from Consumer (bytes)		Remarks
Connection Start	548	→			Connection Setup
		←	Connection Start Ack	350	Connection Setup
Connection Tune	74	→			Connection Setup
		←	Connection Tune Ack	74	Connection Setup
Connection Open	73	→			Connection Setup
		←	Connection Open Ack	67	Connection Setup
Channel open	67	→			Channel Setup
		←	Channel Open Ack	70	Channel Setup
Queue Declare	84	→			Queue Setup
		←	Queue Declare OK	85	Queue Setup
Basic Publish	81	→			Regular
Content Header	76	→			Regular
Content Body	85	→			Regular
Channel close	88	→			Channel teardown
		←	Channel Close OK	66	Channel teardown

Connection Close	88	→			Connection teardown
		←	Connection Close Ack	66	Connection teardown

EMMA provides a graphical user interface utilizing the DLUX module of ODL to track the current location of the train w.r.t testbed, as illustrated in Figure 18. Details of the train tracking are included in deliverable D4.2 [50] (section 6.4).

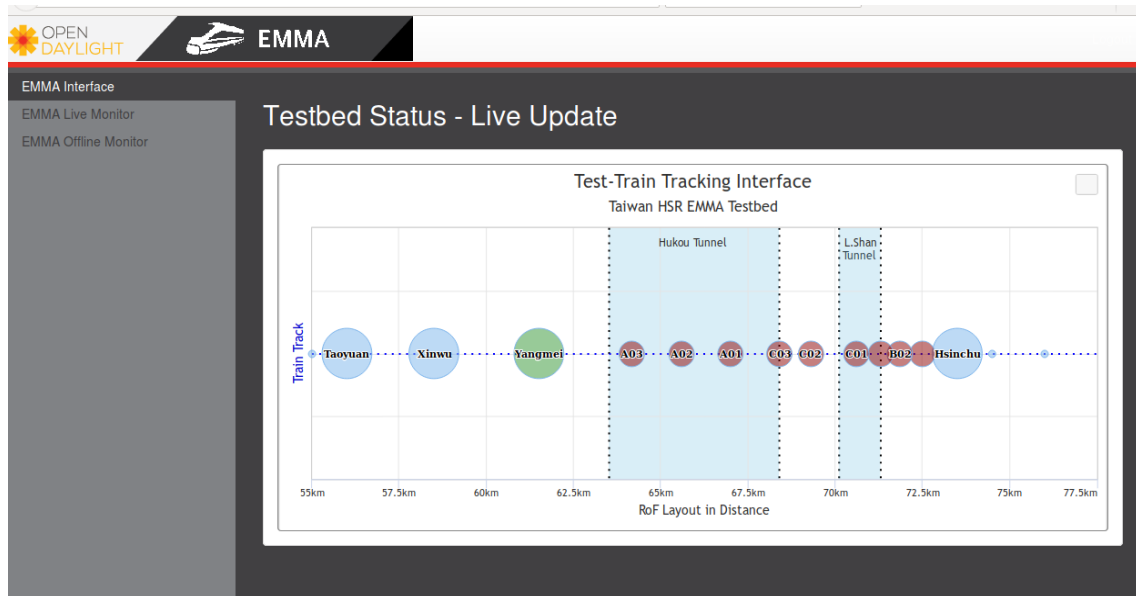


Figure 18. EMMA GUI to track current location of the high-speed train

Figure 19 represents the EMMA live monitor to record the current energy consumption of the RoF nodes.

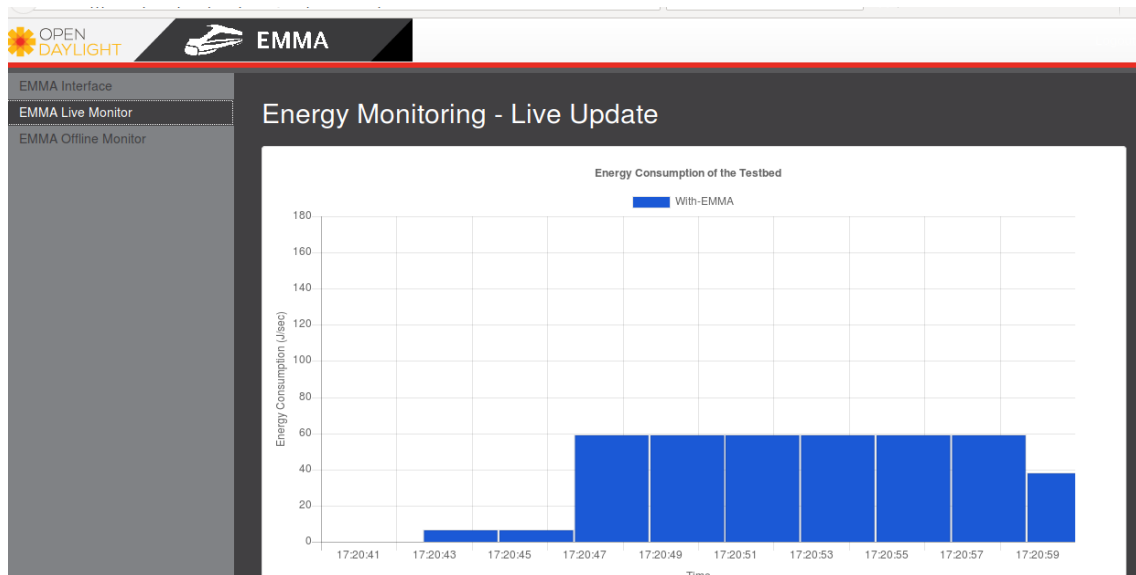
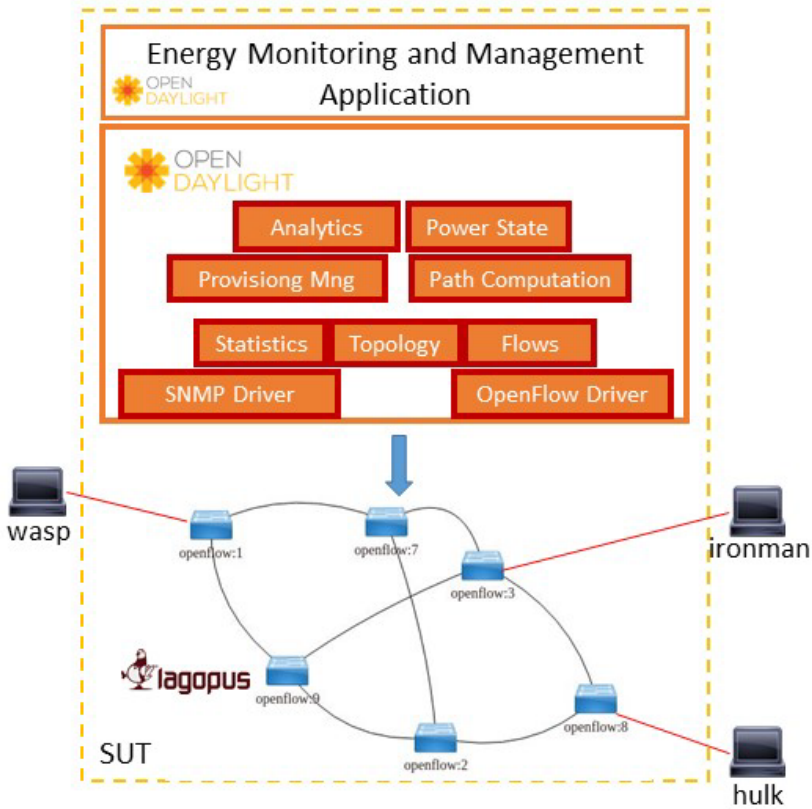


Figure 19. EMMA interface for the live energy monitoring of RoF nodes

2.3.2 Experiment #4: Energy-oriented network resource management in XPFE domains

On-demand provisioning of network connections	
Description	This experiment shows on-demand provisioning of energy-efficient network connections over XPFEs, with automated regulation of XPFEs' power states (i.e. sleeping mode, active mode).
State of the art	In operational environments, network nodes are always active; connections are based on static pre-provisioned paths or dynamically computed using routing algorithms mostly based on SPF (Shortest Path First) mixed with semi-static policy and TE (Traffic Engineering) driven decisions. The current trend in research is moving from an «always on» to an «always available» approach, considering mostly two device power states (on/off) [3].
Improvement from State of the art	Reduction of power consumption selecting the network paths (in XPFEs infrastructures) that minimize the power consumption of the overall physical network, based on the current network resource allocation needs and traffic characterization.
5G-Crosshaul Use Cases	Dense Urban Society
System Test under	 <p>The following components of the XCI are relevant for this test:</p> <ul style="list-style-type: none"> - SNMP driver

	<ul style="list-style-type: none"> - OpenFlow driver - Statistics Manager - Topology Manager - Flow Manager - Analytics Manager - Provisioning Manager - Power State Manager - Path Computation Manager <p>The scalability of the XCI (SDN controller and EMMA network algorithms) has been tested, using Mininet⁴, in emulated reference environments, up to 250 network nodes.</p>
Project Objectives / 5GPPP KPIs addressed	<p>Obj. 2: Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours</p> <p>Obj. 6: Scalable management framework: algorithms that can support 10 times increased node density (only addressed in scalability tests)</p> <p>Obj. 7: Reduce energy consumption in the 5G-Crosshaul by 30% through energy management (the function shown in this experiment is an enabler for this objective)</p>
Measured KPIs	<p>Infrastructure virtualization: time required for setting up a path, including (i) time for path computation; (ii) time for change of device power states and (iii) time for configuration of flow tables.</p> <p>On-demand adaptation; Resource efficiency: service acceptance rate for network connection requests (this has been validated in the context of WP4 using an emulated environment for scalability reasons).</p> <p>Energy efficiency: reduction of power consumption adopting the EMMA application (compared with “always on” approach)</p>
Measurement tools	<p>Wireshark to capture control plane messages.</p> <p>Python scripts to compute results about service acceptance rate.</p> <p>Python scripts to compute results from power consumption data collected via SNMP (test in real environment) or computed based on a theoretical model (test in emulated environment). In this case the first option is adopted.</p>
Measurement procedure	<p>For KPIs related to timings, the Wireshark pcap traces of control plane messages as well as control plane component logs are analysed and processed in order to compute the execution time of the different provisioning steps.</p> <p>For KPIs related to service acceptance rate, python scripts are used to emulate provisioning and termination requests and process their results.</p>

⁴ <http://mininet.org/>

	For KPIs related to power savings, the monitoring data collected from the XPFEs (or processed using power consumption models in case of emulated environment) when EMMA app is running are compared with scenarios where EMMA is not running and traditional algorithms are used (shortest path first and all devices in active mode).
Constraints	<p>The accuracy of theoretical energy models should be further validated with real data.</p> <p>Meaningful results about service acceptance rate and power savings can be obtained only emulating a network with a significant size and topology. The tests on the testbed are used to demonstrate the feasibility of the solution.</p>
Main results	<p>Power consumption saving: without connections, we reach a power saving of 84W when compared with the 210W of the “always on” approach (around 40%), while with a single path established the power saving is reduced to 42W (around 20%).</p> <p>Scalability tests: the efficiency of the power saving algorithms in larger topologies was evaluated by emulating in Mininet realistic topologies (including a 51-node regional network in the North-West of Italy and networks of up to 250 nodes). Under worst case traffic conditions (i.e., maximum traffic load) a global power saving of 12% can be achieved (due to switching off 6 nodes). This saving must be added to those studied in WP1 (up to 70%).</p> <p>Connection setup time: for network paths crossing three nodes, the average setup time is in the order of 3 seconds when an XPFE power state change is required, it is less than 1 second if the power state change is not required.</p> <p>Scalability tests: For realistic topologies ranging from 50 to 250 nodes, the average connection setup time remains under 1s (i.e., 0.78s), though it may reach up to 7 seconds for the 250-node network if many nodes must change power state.</p>

Discussion of results

This experiment evaluates the performance of the EMMA application in terms of connection setup time and reduction of power consumption. The environment is the same used for experiment 1 and 2, with 6 XPFEs and three servers attached to the edges of the network. During the experiment, we instantiate three connections, as defined in Table 6, and for each of them we measure the time required for the provisioning of the paths, identifying the contributions of the different procedures:

- *Coordination of path provisioning procedures*, including the elaboration of the connection request and the translation of the path in the equivalent set of flow rules to be configured on the XPFEs.
- *Path computation*, i.e. the algorithm in charge of selecting the network resources which minimize the global power consumption.
- *Adjustment of XPFEs' power state* in order to activate and deactivate the data plane forwarding in the Lagopus switches.
- *Flow rules configuration* on the data plane XPFEs via OpenFlow protocol.

The analysis is performed analysing the timestamps in the logs of the EMMA components within the ODL controller and the timestamps of the packets exchanged between controller and XPFE devices.

The details of the established paths are reported in the following table. It should be noted that Path 1 setup requires activating XPFE 1, XPFE 9 and XPFE3, while Path 2 setup requires the additional activation of XPFE 2. On the other hand, when Path 3 is established all the XPFEs are already active and no further changes in the XPFEs' power state is needed.

Table 6: Details of the paths installed for EMMA tests on XPFEs domain

Path	Source	Destination	Traversed nodes	Tenant
Path 1	Wasp	Ironman	XPFE 1 – XPFE 9 – XPFE 3	Tenant 1
Path 2	Hulk	Ironman	XPFE 2 – XPFE 9 - XPFE 3	Tenant 2
Path 3	Wasp	Hulk	XPFE 1 – XPFE 9 - XPFE 2	Tenant 1

A consistent reduction of power consumption can be hardly demonstrated in this limited scenario with just 6 XPFEs, due to the low number of network nodes that can be kept in sleeping mode (just 2 after the paths configuration) and of alternative paths that can be used to connect two hosts. For this reason, we use this experiment just to demonstrate the feasibility of the EMMA solution in a real environment, while we refer on wider emulated networks for a more meaningful performance analysis (see Table 8).

Table 7 shows the average connection setup times and their components for the three different paths, in ms, while the range of values of global setup time and its component is shown in following figures. We can see that the total setup time for Path 1 and Path 2, which includes the activation of the XPFEs, is much longer than the time required to establish Path 3, since all the XPFEs are already active (see Figure 20).

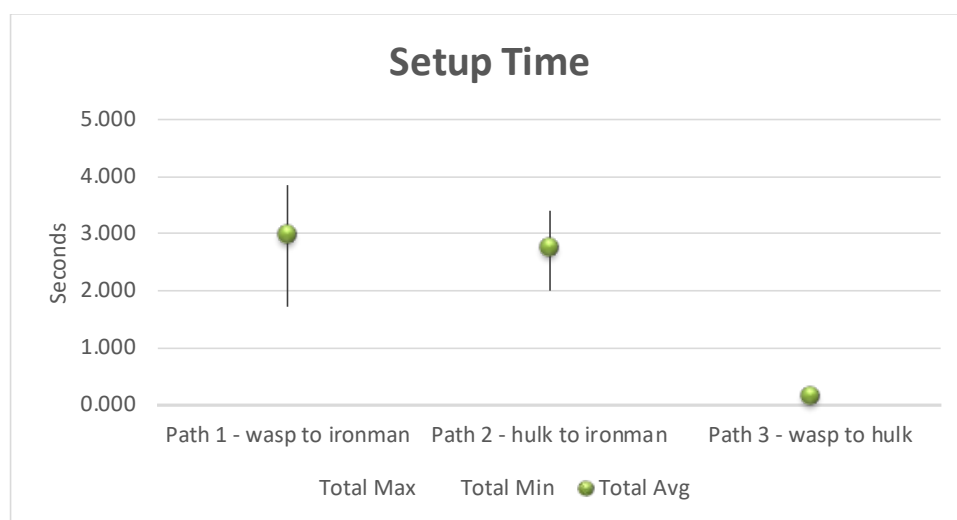


Figure 20. Total setup time for each network path

It is clear that the configuration of the power states of the devices introduces the bigger delay, as well as the major variability in the setup time. While the internal procedures (i.e. provisioning

coordination and path computation, respectively shown in Figure 21 and Figure 22) takes just few milliseconds, the actions that require an interaction between controller and devices are in the order of hundreds of milliseconds for the configuration of OpenFlow flow rules (with a maximum peak of 1.5 seconds, see Figure 23) and few seconds for the configuration of the power state (with a maximum peak of nearly 5 seconds).

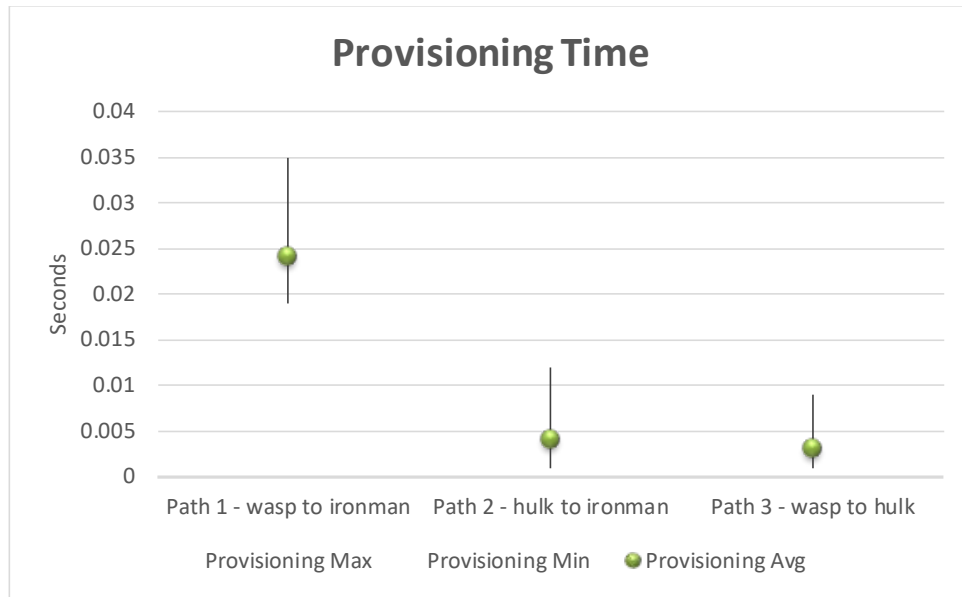


Figure 21. Time used by the Provisioning Manager module to elaborate connection setup requests

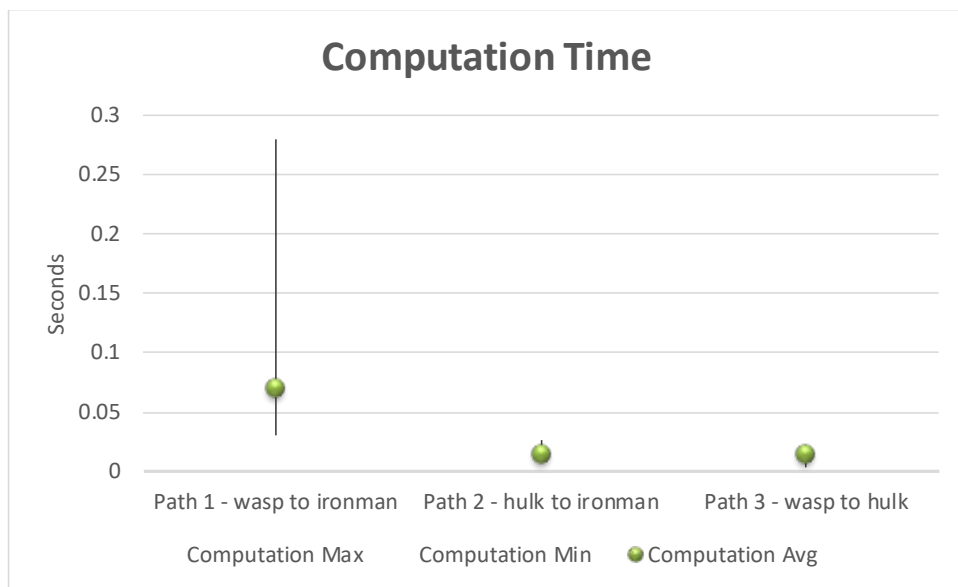


Figure 22. Time used by the Path Computation Manager to compute the requested path

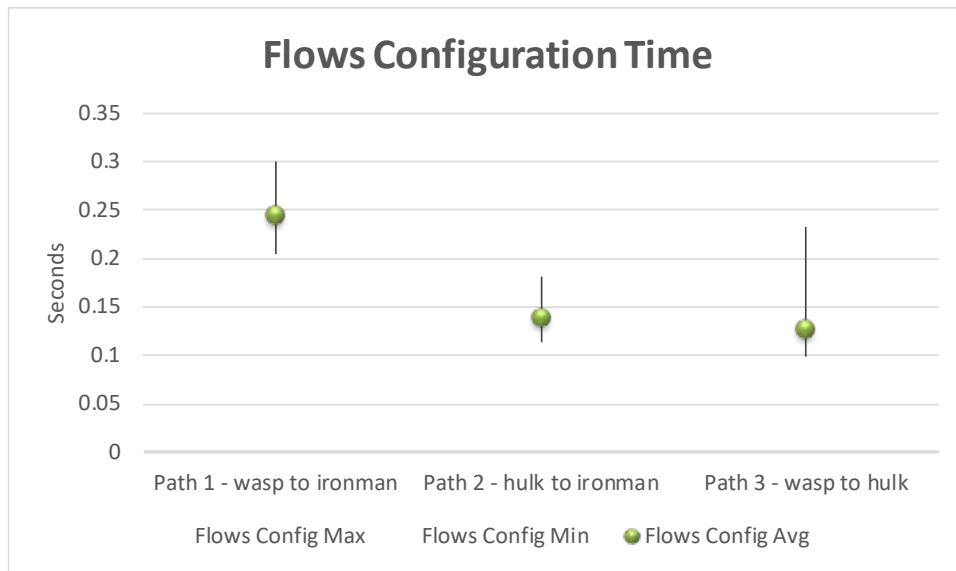


Figure 23. Time used by the Provisioning Manager to configure the flow rules through the SDN Controller SBI

It should also be noted that the configuration of the power state constitutes the main difference in the provisioning procedure, if compared with the traditional connection setup using a shortest path approach, and it brings a further component of the end-to-end provisioning time (see Figure 24).

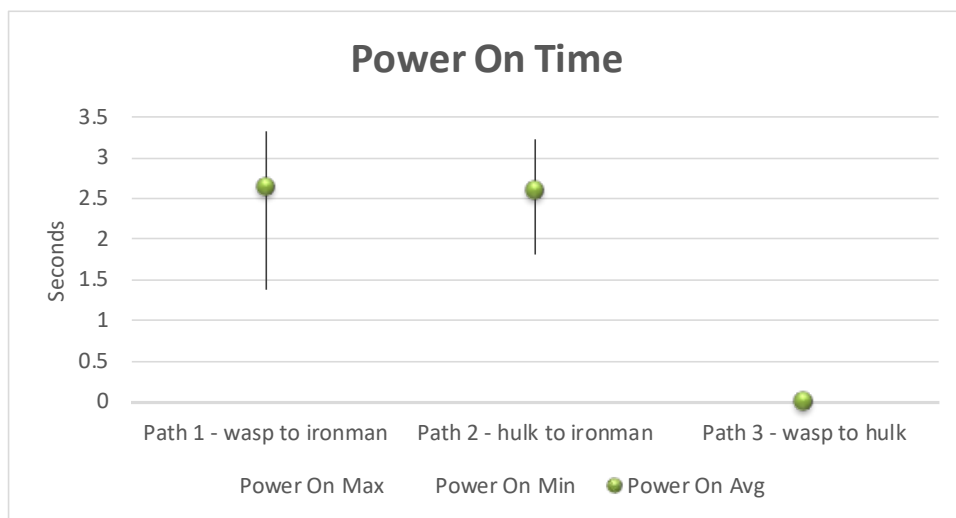


Figure 24. Time used by the State Manager component to switch on the needed nodes

The average value and the variability of this additional time component is due to the type of interaction between the controller and the XPFE for the configuration of the power state via SNMP protocol (following the procedure defined through the standard EMAN MIB⁵): when needed (e.g. for the setup of path 1 and path 2) the controller sends an SNMP command to set the configurational power state and verifies that the command has been correctly executed using an SNMP query on the operational power state of the involved devices.

⁵ <https://tools.ietf.org/html/rfc7460>

However, since the SNMP queries are implemented through an asynchronous polling mechanism in the software, this delay does not increase with the number of nodes to be switched on but varies in a range that depends on the (configurable) polling frequency. This means that in wider networks, where EMMA can reach its best performance due to the high number of nodes that can be maintained in sleeping mode, this delay can be kept in a reasonable range independently on the increasing number of XPFEs to be activated. A further improvement could be achieved by implementing a mechanism based on asynchronous notification via SNMP traps issued by the device as soon as the power state configuration has been actuated, avoiding the delay introduced by the polling procedure.

It should be also noted that the internal processing for the setup of the first path (provisioning coordination, computation and translation of flow rules) takes in general more time than the next ones. This is only due to some initialization processing within the software and it affects the first setup only.

Table 7: Average connection setup time (ms)

Path	Provisioning	Computation	Power State Config	Flow Rules Config	Total Setup Time
Path 1	24 ms	70 ms	2623 ms	244 ms	2982 ms
Path 2	4 ms	14 ms	2593 ms	139 ms	2750 ms
Path 3	3 ms	12 ms	0 ms	127 ms	144 ms

In terms of power consumption saving, we can refer to Figure 25 which shows the evolution of the power consumption of the entire physical network when no paths are established, after the instantiation of path 1 and after the instantiation of path 2. The further instantiation of path 3 does not change anything since all the nodes are already active after path 2 setup.

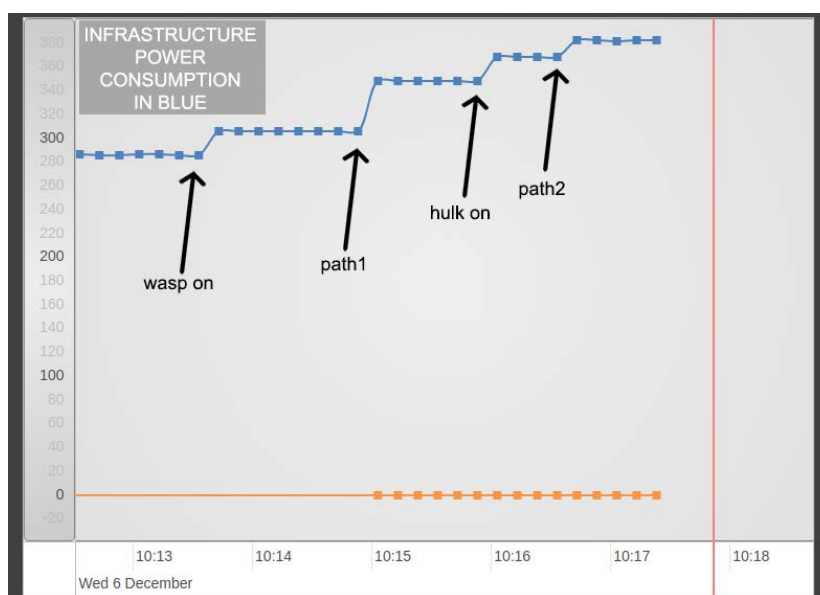


Figure 25. Evolution of network power consumption during setup of Path 1 and Path 2 (EMMA GUI)

In Figure 26 we present a graph that compares the average power consumption with a different number of established paths when EMMA is adopted (blue) and using a shortest path first

algorithm without EMMA, i.e. keeping always all the nodes in full active mode (red). We compare the values in a no traffic condition since, as highlighted in Experiment #1, the traffic would just introduce a very low variable component that would be the same in both cases.

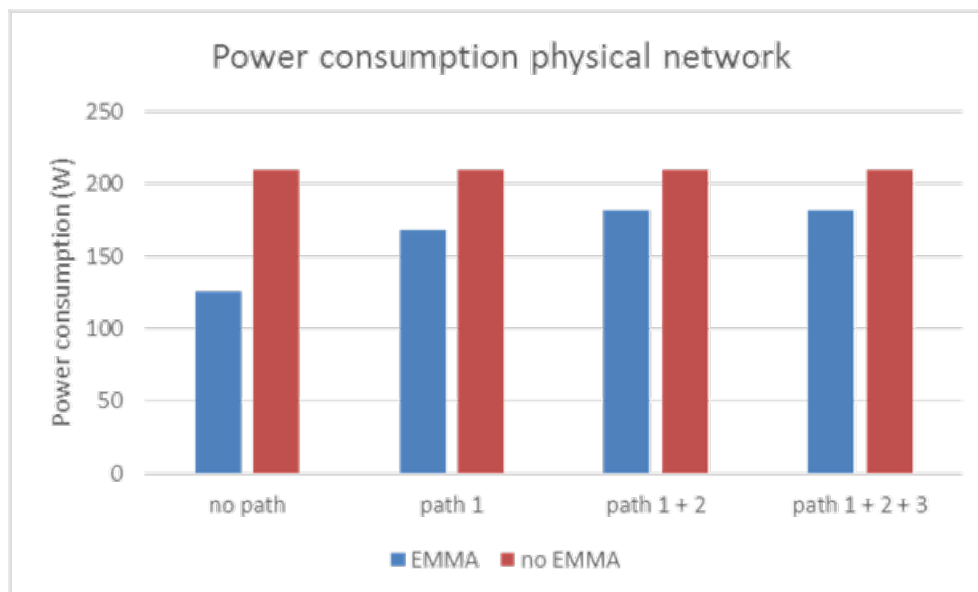


Figure 26. Power consumption saving comparison between EMMA and SPF

As expected, in this particular scenario, the EMMA approach brings benefits just when zero or one path is established (40% and 20% of power saving, respectively), since keeping some nodes in sleeping mode reduces the power consumption of the global network. From the second path on, all the needed nodes are already in full active mode and no further node state change is required, saving overall 28W.

On the contrary, in a larger network, if the traffic is not too high, a large percentage of the nodes can be maintained in sleeping mode since the EMMA routing algorithms tend to concentrate the traffic on a limited number of XPFEs, thus reaching higher values of power saving. This has been demonstrated in the context of WP4, over an emulated network (results are reported in [2]). In this real environment the objective is just to demonstrate the feasibility of the proposed solution and the overhead it brings in terms of provisioning time.

In order to evaluate how this overhead scales when adopted on larger networks, we have also run the XCI SDN controller with the EMMA algorithms over different network topologies, emulated with mininet, which reflect the reference architecture design as foreseen by a network operator in the consortium for regional areas of different dimensions and user densities. The original topology, used also for initial tests reported in [48], serves 1497 antennas, includes a total of 51 network nodes and 61 links, with 4 BBUs, two of them acting as gateway; the topology structure is based on several rings interconnected through some specific nodes acting as BBU and/or gateways, as shown in Figure 27. Network topology with 51 nodes. This is a realistic model for a regional network in the North-West of Italy. Considering the traffic values for the maximum load, as provided by the operator, the EMMA algorithm is able to switch off 6 nodes out of 51, reaching a global power saving of 12% in the worst condition, i.e. with maximum traffic, and assuming the deployment of nodes consuming minimum power in the sleeping mode. As analysed in WP1, this is the contribution of the EMMA algorithms only, that needs to be added to the 70% of power saving enabled by the introduction of 5G-Crosshaul data plane technologies and by the adoption of multi-tenancy.

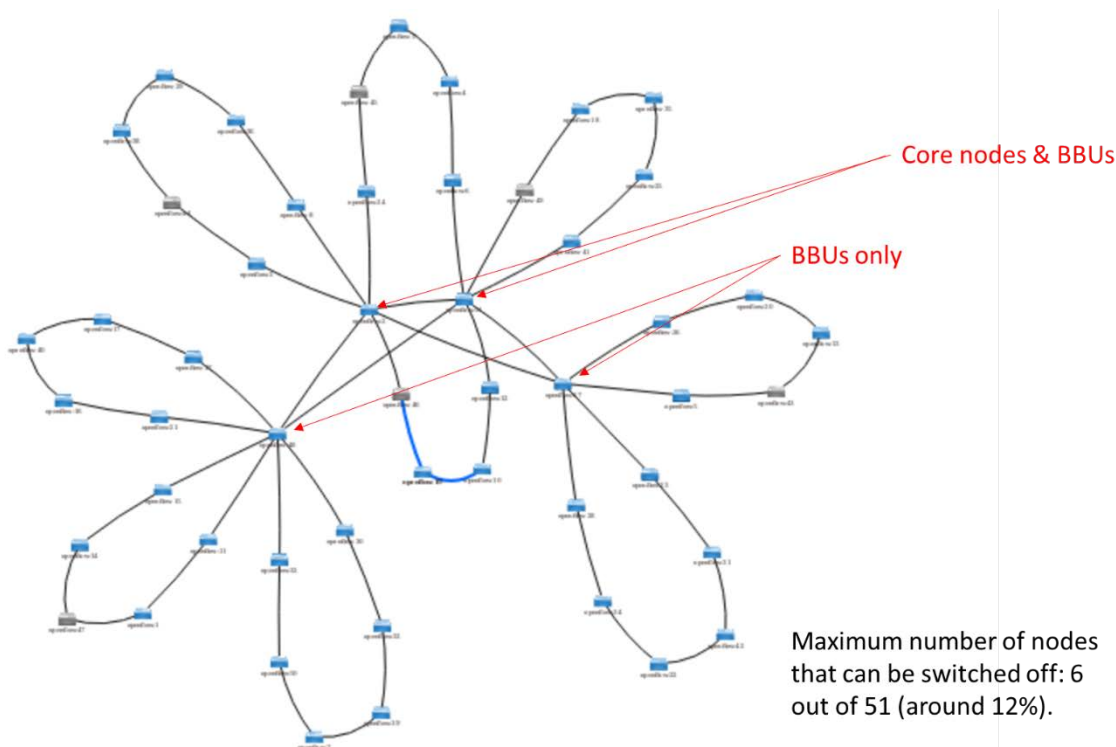


Figure 27. Network topology with 51 nodes

Following the network operator’s guidelines, this topology may scale up to 100, 150, 200, 250 nodes, depending on its geographical placement. Typical deployments are based on a central ring of BBUs and/or gateways, where each of these nodes are then interconnected to a certain number of peripheral rings with edge nodes interconnecting the antennas. The reference topologies are summarized in the table below.

Table 8: Network topologies and number of path for XCI scalability test

# network nodes	# antennas	# BBUs	# gateways	# central ring	# peripheral rings	# bi-directional paths
51	1497	4	2	1	8	2994
100	2994	6	2	1	16	5988
150	4491	8	2	1	24	8982
200	5988	10	2	1	32	11976
250	7485	12	2	1	40	14970

The scalability of the XCI (SDN controller and EMMA network algorithms) has been tested by establishing bi-directional network paths from each edge node to the gateways, one path for each of the antennas attached to the edge nodes up to one of the gateways (for protection issues). Under this assumption, in the worst case condition, the XCI SDN controller has to manage a topology with 250 network nodes and a total of 14970 bi-directional network paths. It should be noted that, in order to support this increasing load, the implementation of the SDN controller has migrated from ODL Beryllium SR2 (used in the test reported in [48] with up to 51 nodes) to the latest

version ODL Nitrogen SR1, which guarantees an overall improvement of the performance. In the following graph (Figure 28), we present the average, maximum and minimum time for the provisioning of the network paths in the reference network topologies. The average remains under the second (0.78 s with a low standard deviation of 0.97 for the worst case of 250 nodes), with a maximum value of nearly 7 seconds (6.83 s) which is due to the time needed to switch on and off nodes when a power status change is needed.

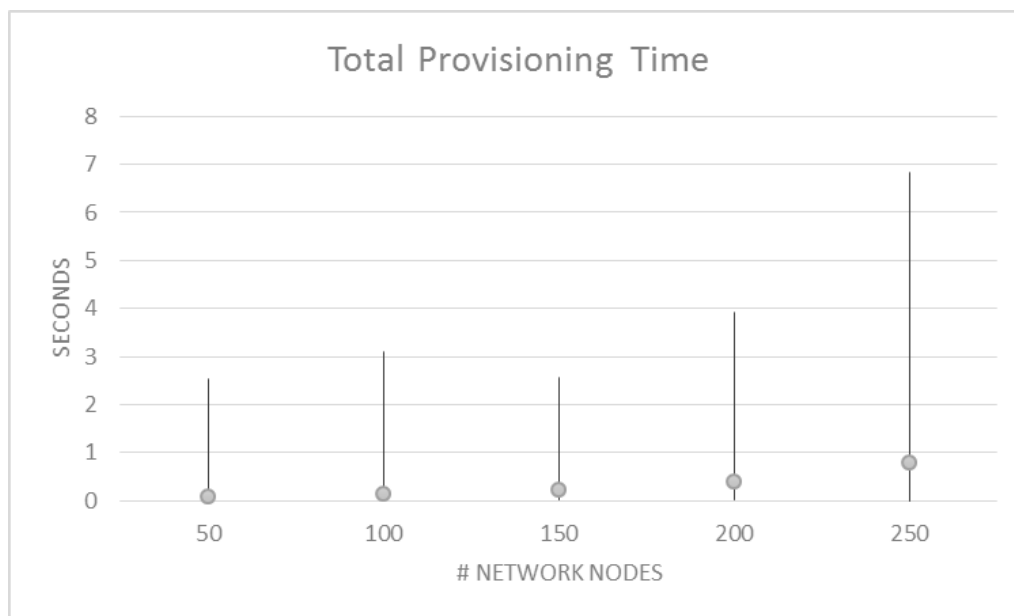


Figure 28. Total provisioning time in emulated networks of different size.

2.4 Experiments on energy-oriented provisioning of virtual infrastructures

2.4.1 Experiment #5: Energy-oriented virtual infrastructure management

Energy-oriented virtual infrastructure management	
Description	This test shows the capability of the EMMA application to provision energy-efficient vEPC service instances following an on-demand approach, selecting the most suitable XPU's for the placement of the vEPC VNFs and automatically regulating the XPU's power states.

State of the art	<p>In operational environments, both network nodes and servers are always active; connections are based on static pre-provisioned paths or dynamically computed using routing algorithms mostly based on SPF mixed with semi-static policy and TE driven decisions. The computation of network and IT allocation is usually disjoint, with VM placement managed by cloud platforms and network allocation managed by the network control plane. In terms of energy management, the current trend of the research is in fact moving from «always on» to «always available» approaches, with mostly two device power status (on/off) (See [3]).</p> <p>In the literature, no experimental works about energy-efficient resource allocation for VNF allocation combined with network paths have been found, but only emulation and simulation results. This kind of state of the art approaches have been considered as benchmark in WP4 activities, where the EMMA algorithms have been evaluated in emulated environments to verify their performance in scalable infrastructures and with different traffic loads (see [50]). They are considered out of scope for WP5, since here the objective is to verify the feasibility of the solution in a real testbed.</p> <p>However, some examples of state of the art work are the following:</p> <ul style="list-style-type: none"> - The solution proposed in [31] seeks to jointly solve the problems of (i) VNF placement, (ii) VNF scheduling, and (iii) traffic routing. The resulting problem is too hard, so the authors resort to heuristics where placement and scheduling/routing are solved jointly; in particular, each VNF in the chain is always placed at a one-hop neighbour node of the preceding one (re-using existing instances if possible). The cost model includes CAPEX, OPEX, and (traffic-dependent) link costs. - The solution proposed in [32] adopts an “energy-aware” approach to VNF placement and scheduling; however, their algorithm actually optimizes the server utilization, i.e., try to use as few servers as possible and placing the others in standby mode. The consumption coming from network elements such as switches and links is not taken into account; indeed, link capacities are neglected altogether and the traffic steering dimension of the problem is left for “future work”.
Improvement from State of the art	<p>Reduction of power consumption selecting the combination of XPU and XPFEs resources that minimizes the power consumption of the overall physical infrastructure based on the current Service Function Chain instances, with automated regulation of the XPU and XPFEs power status, based on current resource allocation needs and traffic characterization. In this work, we verify the feasibility of the solution on a real testbed, evaluating the impact of the power state changes in the service provisioning time.</p>
5G-Crosshaul Use Cases	Dense Urban Society

<p>System under Test</p>	
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 2: Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours</p> <p>Obj. 7: Reduce energy consumption in the crosshaul by 30% through energy management.</p>
<p>Measured KPIs</p>	<p>Infrastructure virtualization: time required for setting up a vEPC instance, including (i) time resource allocation elaboration at EMMA; (ii) time for change of device power states at XPUs and XPFEs and (iii) time for resource allocation in terms of VMs instantiation, VNFs configuration and setup of network flow on XPFEs.</p> <p>On-demand adaptation; Resource efficiency: service acceptance rate for vEPC instances requests. This aspect is validated in WP4, using an emulated environment for scalability reasons; the results are reported in [50].</p> <p>Energy efficiency: reduction of power consumption adopting the EMMA application (compared with “always on” approach). This aspect is validated in WP4, using an emulated environment for scalability reasons; the results are reported in D4.2.</p>
<p>Measurement tools</p>	<p>Wireshark to capture control plane messages.</p> <p>Python scripts to compute results about service acceptance rate.</p>

	Python scripts to compute results from power consumption data collected via SNMP (test in real environment) or computed based on a theoretical model (test in emulated environment).
Measurement procedure	<p>For KPIs related to timings, the Wireshark pcap traces of control plane messages as well as control plane component logs will be analysed and processed in order to compute the execution time of the different computing, provisioning and configuration steps.</p> <p>For KPIs related to service acceptance rate, python scripts will be used to emulate provisioning and termination requests and process their results.</p> <p>For KPIs related to power savings, the monitoring data collected from the XPU and XPFEs (or processed using power consumption models in case of emulated environment) when EMMA app is running will be compared with scenarios where EMMA is not running and traditional algorithms are used (shortest path first, OpenStack Nova scheduler and all devices in active mode).</p>
Constraints	<p>The accuracy of theoretical energy models should be further validated with real data.</p> <p>Meaningful results about service acceptance rate and power savings can be obtained only emulating a network with a significant size and topology. The tests on the testbed are used to demonstrate the feasibility of the solution.</p> <p>Power consumption monitoring and programmable change of power consumption state must be supported at the hardware level on the XPU servers.</p>
Main results	Network Service (NS) provisioning time: the average time for the provisioning of a vEPC NS, including the vEPC components configuration time, is nearly 5 minutes, where the impact of changing the power state of XPFEs and XPUs dynamically (in order to save power when the resources are not used) is minimum, in the order of tens of milliseconds. In preliminary tests, where the VNF applications configuration was not performed, the main time component was the VMs creation at the VIM. After introducing this final step for the NS provisioning, the longest time component is the VNFs configuration triggered and orchestrated at the NFVO and executed within each VNF.

Discussion of results

This experiment evaluates the performance of the EMMA application in terms of vEPC Network Service (NS) setup time. The reduction of power consumption cannot be really appreciated in this experimental environment, due to the limited size of the topology. This aspect has been thus analysed in details in WP4, using an emulated and scalable environment. The related results are reported in [50]. Instead, the objective of this experiment is to validate the whole NS provisioning procedure from a functional point of view and measure the NS provisioning time in a real testbed, evaluating the impact of the time needed to change the power states of XPUs and XPFEs in the global workflow. It should also be noted that this experiment refers to a simple vEPC service instance, with one single VNF for each functional component. The scalability of the vEPC with more VM instances and the placement of the components closer to the user (e.g. in a Multi-Access Edge Computing environment) are out of scope for this experiment.

Initial functional tests were performed in a test environment composed of 3 virtualised XPFEs connected in a ring and two XPUs (i.e. OpenStack Compute Nodes) attached respectively to

XPFE 1 and XPFE 3. The OpenStack network node was connected with the XPFE 2⁶ (see Figure 29).

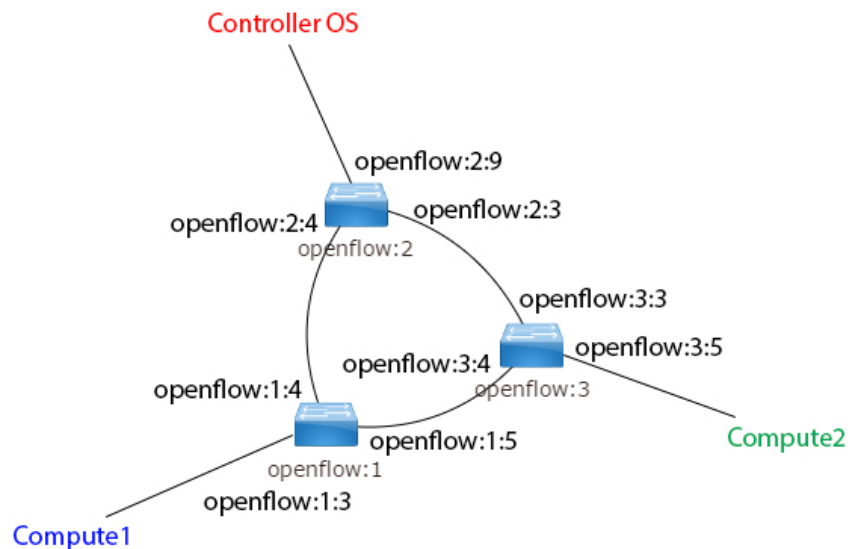


Figure 29. NXW testbed environment

Then, tests have been repeated with a wider topology in the 5G-Crosshaul testbed, composed of 6 XPFEs and 3 XPU, as depicted in Figure 30.

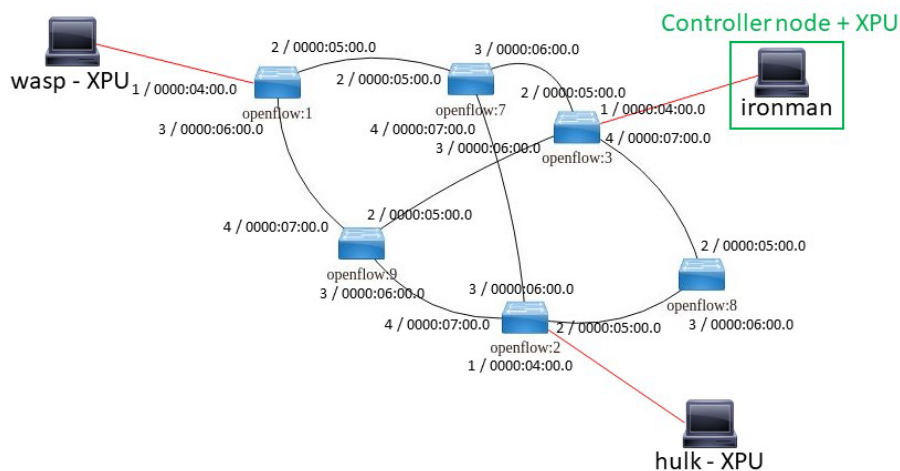


Figure 30. 5TONIC Demo1 topology

During the experiment, we instantiate a vEPC service and we measure the time required for the provisioning of the whole NS, analysing its composition. The vEPC NS is based on the Open Air Interface (OAI) software and includes four VMs (HSS, MME, SPGW and OAISIM). The VMs are interconnected through virtual links (i.e. virtual networks) modelling the reference points between the functional entities of the EPC architecture. Moreover, a further virtual link with a

⁶ In this specific test, the test environment is composed of VMs (2 vCPUs, 8 GB RAM) deployed in NXW lab.

Service Access Point (SAP) interconnecting to external networks has been added for management issues. The details of the resulting vEPC NS Descriptor (NSD) are reported in Figure 31:

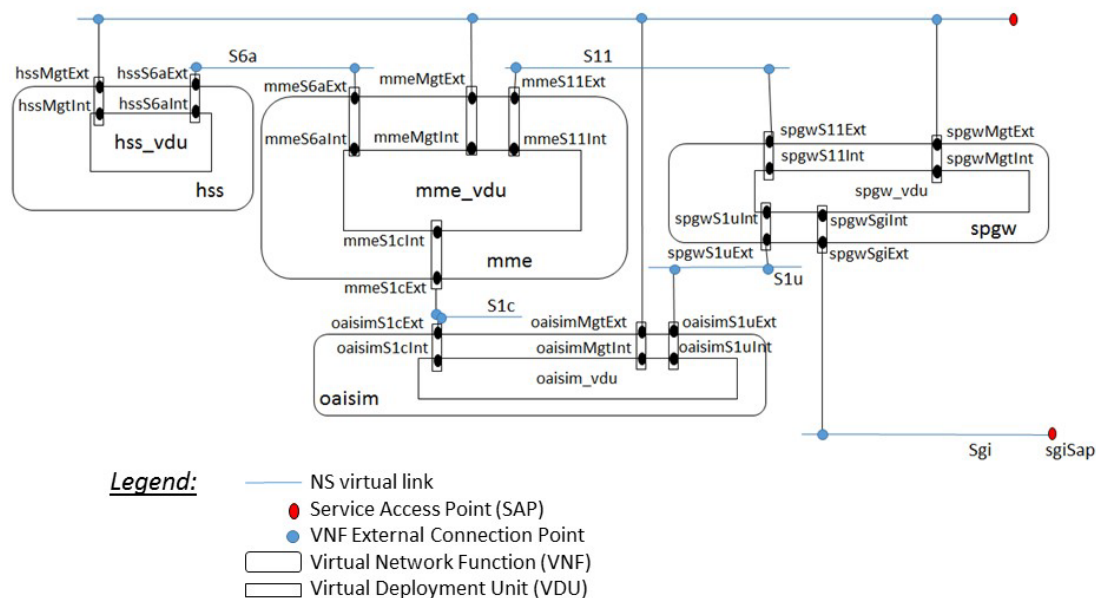


Figure 31. vEPC details

The provisioning of the vEPC NS is fully automated and triggered through the GUI of the NFVO. The workflow includes the following steps (see [48] for a more detailed description of the procedure and the role of each component):

1. Analysis and validation of the NS request at the NFVO
2. Computation of the resources that needs to be allocated to instantiate the NS. This computation is performed by the EMMA algorithm, which decides jointly where the VNFs must be deployed (XPU selection) and which network connections must be established between the selected XPUs, including their paths (path computation). One network connection is created for each virtual link that interconnects VMs placed in different XPUs or for each virtual link that interconnects an SAP (in this case the edge is the VIM network node, since it provides access to the external networks). The objective function of the algorithm is the minimization of the global power consumption of the physical infrastructure.
3. Change of the power state of XPUs and XPFEs, to activate the devices selected in the resource computation step and previously kept in sleeping mode for power saving issues. The change of the power state is triggered by the NFVO and executed by the VIM for the XPUs and by the SDN controller for the XPFEs. The interaction with the devices is based on the SNMP protocol.
4. Creation of the VIM virtual resources associated to the virtual links. In particular, this step is coordinated by the NFVO, which sends the VIM requests to create in the order:
 - a. For each NSD SAP, one virtual port attached to an external router;
 - b. For each NSD virtual link, one virtual network;
 - c. For each virtual network created in the previous step, one subnet.

5. Creation of the VMs associated to the VNFs specified in the NSD. This step is again coordinated by the NFVO, but it is delegated to the vEPC VNF Manager (VNFM), which is responsible of the lifecycle of each single VNF. The VNFM, in turn, sends the VIM the requests to create the VMs and the related virtual ports, attached to the virtual networks created in the previous step. NFVO, VNFM and VIM implements mechanisms to specify the placement of the VMs, in order to apply the decisions taken by the EMMA algorithms in step 2.
6. Setup of the network connections between the XPU, required to provide the underlying connectivity to carry the VMs traffic over the Crosshaul network infrastructure. In this step the NFVO sends requests for new network paths to the SDN controller, which establishes the paths according to the Lagopus XPFEs pipeline.
7. Configuration of NS VNFs. In this final step, the NFVO completes the configuration scripts for each VNF, with the proper values received after the VMs creation (e.g. IP address). Then, through the management network, the NFVO sends a POST request to a REST server hosted by each VNF. The REST server will execute the received script, configuring the VNF.

During the experiment, we measure the following component of the whole NS provisioning time:

- *Coordination of vEPC provisioning procedures*, including the elaboration of the instantiation request and the triggering of the different component in charge of configure the target resources, i.e. the OpenStack VIM for XPU and the ODL-based SDN controller for XPFEs.
- *EMMA computation*, i.e. the algorithm in charge of selecting IT and network resources which minimize the global power consumption.
- *Adjustment of XPU's power state* in order to activate the XPU via VIM (the initial state is sleeping mode for all the compute nodes).
- *Adjustment of XPFEs' power state* in order to activate the data plane forwarding in the Lagopus switches via SDN controller (the initial state is sleeping mode for all the XPFEs).
- *Instantiation time for all the VNFs* in the XPU nodes, based on the VM status monitoring from OpenStack.
- *Flow rules configuration* on the network data plane, including the processing at the SDN controller and the interaction with the XPFEs via OpenFlow protocol.
- *Configuration time for all the VNFs* in the XPU nodes, this procedure is started by the NFVO once the network connectivity is established and includes the orchestration time at the NFVO as well as the time for the configuration procedure which takes place at each VNF.

The analysis is performed processing the timestamps in the logs of the EMMA components, including the messages exchanged between the different modules in the EMMA NFVO, the interaction between the NFVO and the OpenStack VIM, the interaction between the NFVO and ODL controller, the processing timestamps of the Path Provisioning Manager module within the controller and finally the timestamps of the packets exchanged between controller and XPFE devices. In NXW facilities the experiment is repeated 10 times, while in the 5G-Crosshaul testbed up to 50 times, measuring average, maximum and minimum time. In NXW laboratory, in order to have a meaningful test that includes the creation of the network path, the resources advertised by the VIM are adjusted so that the EMMA algorithms are forced to arrange the allocation of MME and SPGW VNFs in an XPU (compute 2) and the allocation of OAISIM and HSS VNFs

in the other XPU (compute 1). This scenario constitutes the most challenging condition, where all the XPUs and XPFEs will need to be switched on, then the provisioning time results can be thus considered an upper bound limit for the given physical infrastructure, since we are testing the worst case where the maximum number of nodes must be activated and the maximum number of connections must be established. The list of connections is reported in Table 9, including the connections established towards the OpenStack network node for the virtual links connected to the management and the SGI SAP. Each connection is obviously bidirectional.

Table 9: Network connections between VMs established during the vEPC NS provisioning

Path	Source	Destination	Traversed nodes	Tenant
mme to oasim	Compute 2	Compute 1	XPFE 3 – XPFE 1	Tenant 1
mme to hss	Compute 2	Compute 1	XPFE 3 – XPFE 1	Tenant 1
spgw to oasim	Compute 2	Compute 1	XPFE 3 – XPFE 1	Tenant 1
oasim to mme	Compute 1	Compute 2	XPFE 1 – XPFE 3	Tenant 1
oasim hss mgt	Compute 1	Network node	XPFE 1 – XPFE 2	Tenant 1
mme spgw mgt	Compute 2	Network node	XPFE 3 – XPFE 2	Tenant 1
spgw sgi	Compute 2	Network node	XPFE 3 – XPFE 2	Tenant 1

Table 10 shows the maximum, minimum and average time for the provisioning of a vEPC NS and the time components within the NS instantiation workflow. The same data are graphically represented in Figure 32.

Table 10: Network Service provisioning time components (in seconds)

Time Component	Maximum	Minimum	Average
XPU power on time	0.021	0.010	0.016
XPFE power on time	0.039	0.019	0.026
Virtual Links creation	60.241	43.227	52.623
VNFs creation	82.066	62.356	64.710
Single VNF creation	60.442	40.222	54.559
Network paths setup	4.246	3.556	3.786
Total NS instantiation time	154.531	116.216	128.408

Observing Table 10, we can see that the whole NS provisioning time is in the order of 2 minutes. The steps with longest duration are, as expected, the deployment of the VNF VMs and the instantiation of all the network related resources (virtual networks, subnets and ports) in the VIM. However, considering that the average time required to instantiate a single VNF is in the order of

55 seconds and the time required to instantiate all the four VNFs raises to an average of 65 seconds, we can expect that the whole provisioning time will not grow too much even with more complex NSs including an increasing number of VMs. This is mostly due to the parallelization performed at the NFVO and VNFM level for the creation of VMs.

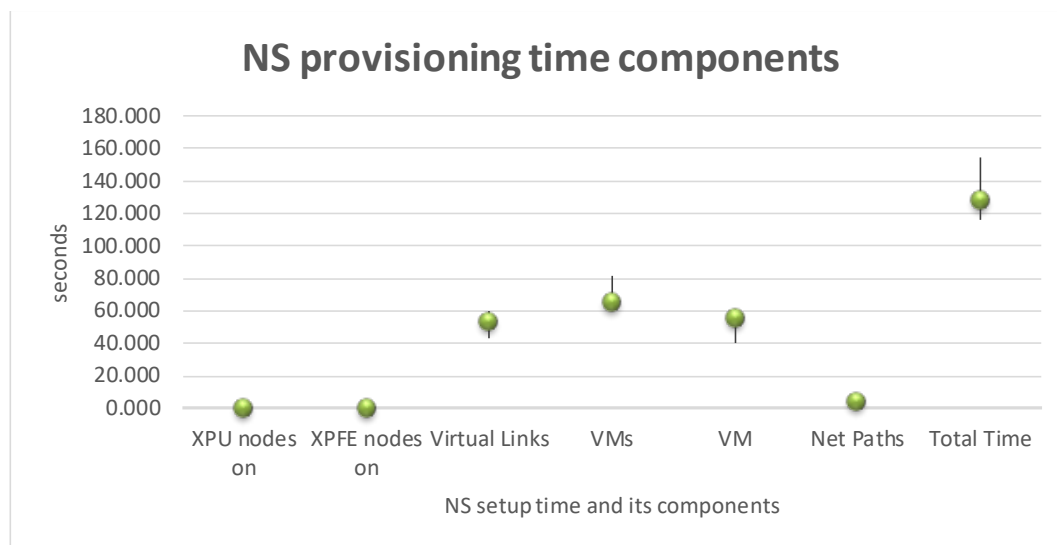


Figure 32. EMMA in XPFEs + XPU domains: average (dots), max and min values (line edges) of NS setup time and its components (NXW lab)

In the 5G-Crosshaul testbed, the experiment is repeated without any constraint on the availability of the XPU resources. The EMMA algorithm arbitrarily decides where the VNFs should be placed and then which XPUs and XPFEs should be eventually switched to *active mode*. Table 11 shows the maximum, minimum and average time for the vEPC NS provisioning time components, where we have taken into consideration also the algorithm computation time and the VNFs applications configuration time.

Table 11: Network Service provisioning time components in 5G-Crosshaul testbed (in seconds)

Time Component	Maximum	Minimum	Average
Computation	15.117	6.144	9.729
XPU power on time	0.111	0.068	0.086
XPFE power on time	2.660	0.871	1.824
Virtual Links creation	31.797	21.953	27.369
VNFs creation	49.738	31.883	38.384
Single VNF creation	38.823	30.235	31.476
Network paths setup	0.065	0.050	0.056
VNFs configuration	316.992	86.885	216.591
Single VNF configuration	187.199	53.854	105.860

Total NS instantiation time	404.868	164.036	299.522
-----------------------------	---------	---------	---------

With respect to the preliminary tests in NXW facilities, we can notice how the VNFs configuration time, as a final step of the workflow, will increase the whole NS setup time up to nearly 7 minutes (5 minutes in average), while the VNFs creation time remains in the order of tens of seconds. Table 11 data are also reported in Figure 33, highlighting how the VNFs configuration step is the most demanding in terms of time, while all the other steps are performed in less than 2 minutes.

In the whole workflow, the time impact to activate the selected physical resources for allocating the VMs and configuring the underlying connectivity is minimum, in the order of tens of millisecond and few seconds respectively. Also in this case, the different requests to activate the resources are sent in parallel at the NFVO. This allows maintaining this time in a reasonable range even with an increasing number of devices to be activated. In fact, if compared with the measurements taken in Experiment #4, the time required to change the power state of the devices decreases from around 2-3 seconds in the previous case to tens of milliseconds in this case. This is mostly due to the optimization of the workflow. The adjustment of the power states is now handled by the NFVO through parallel messages sent to the devices before initiating the resource provisioning and configuration procedures. On the contrary, in Experiment #4, all these procedures were handled by the SDN controller at the same time, thus increasing the time required for each single action. Similarly, the time required at the SDN controller level for configuring the XPFE pipeline flow rules has a minimum impact on the overall provisioning time.

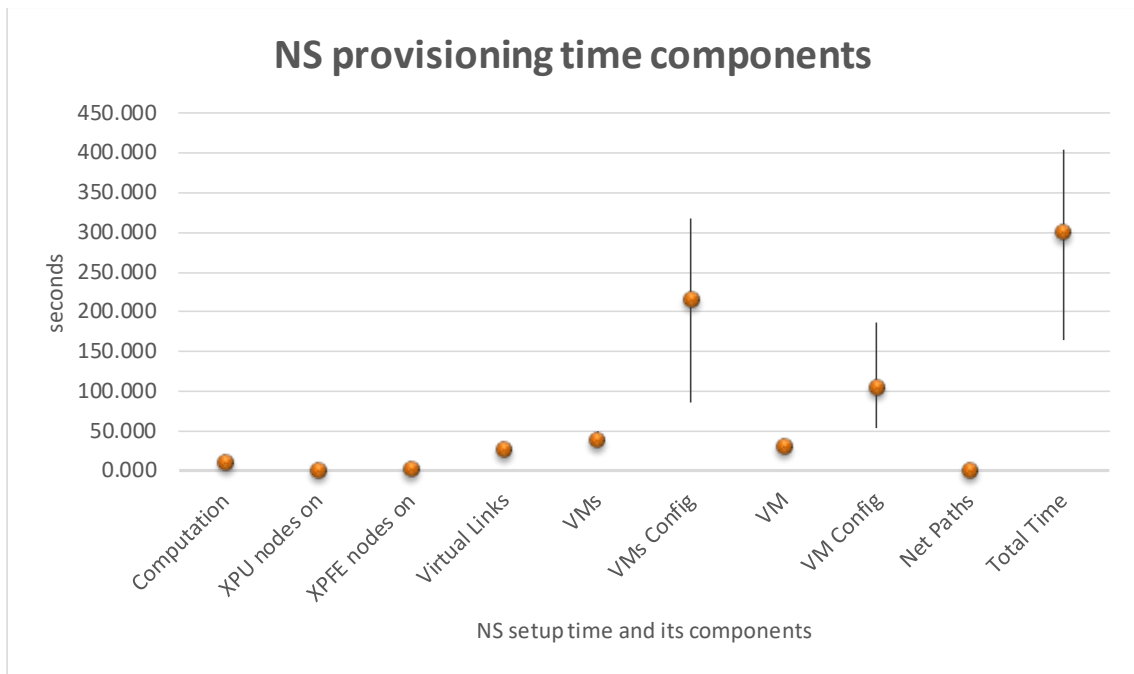


Figure 33. EMMA in XPFEs + XPUs domains: average (dots), max and min values (line edges) of NS setup time and its components (5G-Crosshaul testbed)

2.5 Experiments on energy-oriented network resource management for fronthaul and backhaul traffic

2.5.1 Experiment #6: Energy-oriented network resource management in XPFE domains for on-demand provisioning of connections dedicated to fronthaul and backhaul traffic

On-demand provisioning of network connections for fronthaul and backhaul traffic	
Description	<p>This experiment shows the on-demand provisioning of energy-oriented network connections in XPFEs domains in support of fronthaul and backhaul traffic. In particular, this demonstration shows the setup of connections with different QoS parameters, depending on the target traffic, and how the traffic produced from FH/BH traffic generators are mapped and carried out in these connections.</p>
State of the art	<p>In state of the art solutions fronthaul and backhaul traffic are delivered using different network technologies: the unified fronthaul/backhaul infrastructure is the main innovation of the project.</p>
Improvement from State of the art	<p>The XCI allows creating network paths compliant with the requirements of fronthaul and backhaul traffic over the same network infrastructure, in this case composed of XPFEs. Moreover, the EMMA application optimizes the energy consumption regulating the power status of the devices depending on their current usage.</p>
5G-Crosshaul Use Cases	Dense Urban Society
System under Test	

	<p>The following components of the XCI are relevant for this test:</p> <ul style="list-style-type: none"> - SNMP driver - OpenFlow driver - Statistics Manager - Topology Manager - Flow Manager - Analytics Manager - Provisioning Manager - Power State Manager - Path Computation Manager <p>Moreover, traffic generators for FH and BH traffic will be preliminarily deployed on the XPU's in order to produce the experiment traffic.</p>
Project Objectives / 5GPPP KPIs addressed	<p>Obj. 2: Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours</p> <p>Obj. 7: Reduce energy consumption in the crosshaul by 30% through energy management.</p>
Measured KPIs	<p>Time required for setting up a path, including (i) time for path computation; (ii) time for change of device power state; (iii) time for configuration of flow tables.</p> <p>Energy efficiency: reduction of power consumption adopting EMMA application (compared with “always on” approach).</p>
Measurement tools	<p>Wireshark to capture control plane messages.</p> <p>FH/BH traffic generator.</p> <p>Lagopus management tools to observe packets through flows.</p>
Measurement procedure	<ol style="list-style-type: none"> 1. Preliminary deployment of FH/BH traffic generator in XPU's. 2. Run-time request for a network connection based on the FH traffic constraints through the XCI NBI APIs (latency: 500 μs, throughput: 102 Mbps in DL and 22 Mbps in UL). This triggers the creation of the network paths for the FH traffic. 3. Run-time request for a network connection based on the BH traffic constraints through the extended SDN controller NBI APIs (throughput: 50 Mbps in DL). This triggers the creation of the network paths for the BH traffic. 4. Generation of FH/BH traffic. 5. Verification of the paths where the generated traffic is forwarded. 6. Measurement of the traffic QoS
Constraints	N/A
Main results	<p>The forwarding of the FH traffic is performed in core nodes, as well as in the first hop of the path, through a highest priority queue than the one configured for the BH traffic. After the connections setup requests on the</p>

	<p>Provisioning Manager NBI, the Path Computation Manager, on the base of the specified QoS for the target traffic (i.e. FH and BH traffic), computes the network paths. For each computed path, ingress and core nodes along it, the proper output queue is returned. The Provisioning Manager configures the XPFE pipeline, through the ODL Controller SBI, according to the Path Computation output. Observing encapsulation and forwarding flow tables along each computed path, we can verify that FH traffic is forwarded through a highest priority queue, with respect to the ones computed for the BH traffic.</p> <p>From the power consumption saving point of view, in this limited scenario composed of 6 XPFEs and 2 XPUs, we are able to keep in sleeping mode just 2 network nodes, with a total power consumption saving of nearly 13% with respect to the all-on approach.</p>
--	--

Discussion of results

This experiment validates the EMMA capability of provisioning on-demand network connections in XPFEs domain in support of fronthaul and backhaul traffic. In particular, this demonstration shows the setup of three different connections configured with different QoS parameters, depending on the target traffic, i.e. with a highest prioritisation of the fronthaul traffic handled at the network level through the use of prioritised queues. Note that the queues scheduling is managed at the XPFE level by the switch's software. The test environment, as depicted in Figure 34, is the following:

- 1 VM hosting the CND vEPC (i.e. OpenEPC, composed by MME VM, BBU VM, SPGW VM and EPC-enabler VM) placed in Hulk (XPU).
- 1 VM hosting the CND fronthaul traffic generator (UE VM + RRU VM) placed in Ironman (XPU + Openstack Controller Node).
- 1 VM hosting the CND backhaul traffic generator (UE VM + RRU VM + BBU VM) placed in Wasp (XPU).

In order to allow the communication between the different VMs in the fronthaul/backhaul scenario, the following virtual networks are created in Openstack:

- 1 management network which connects all the VMs to the Controller Node default gateway, configured with vlan 121.
- 1 data network for the fronthaul traffic between the FH VM and the OpenEPC VM, configured with vlan 120.
- 1 data network for the backhaul traffic between the BH VM and the OpenEPC VM, configured with vlan 113.

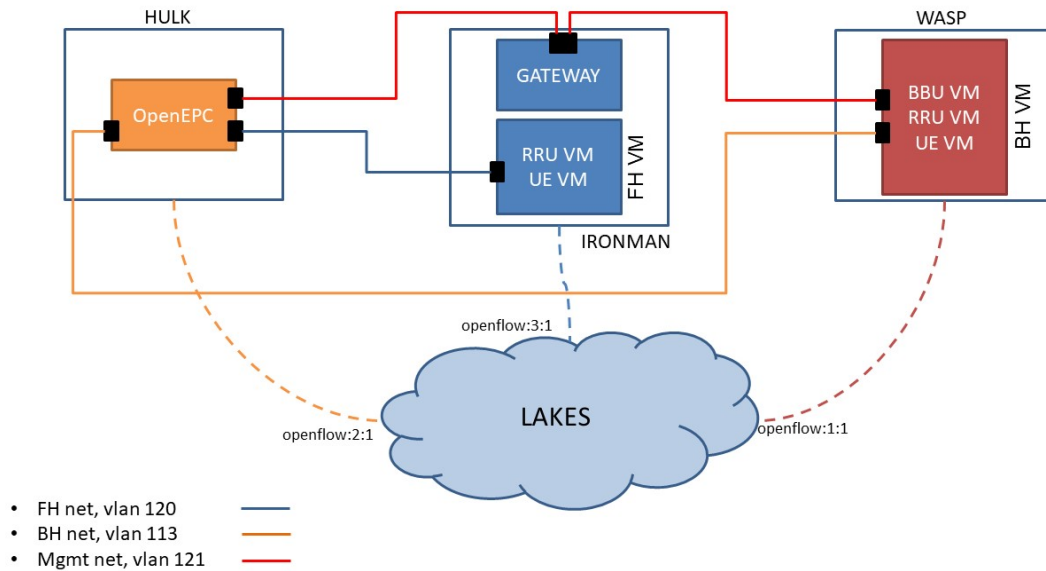


Figure 34. Fronthaul and backhaul traffic integration environment with virtual networks details

The interaction between the different OpenEPC components and the ones used to generate respectively fronthaul and backhaul traffic is depicted in Figure 35 and Figure 36.

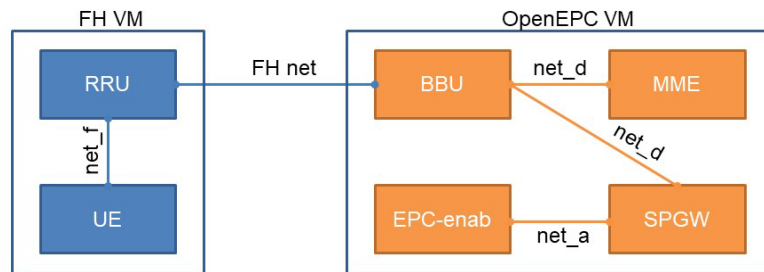


Figure 35. Interaction between FH VMs and OpenEPC VMs

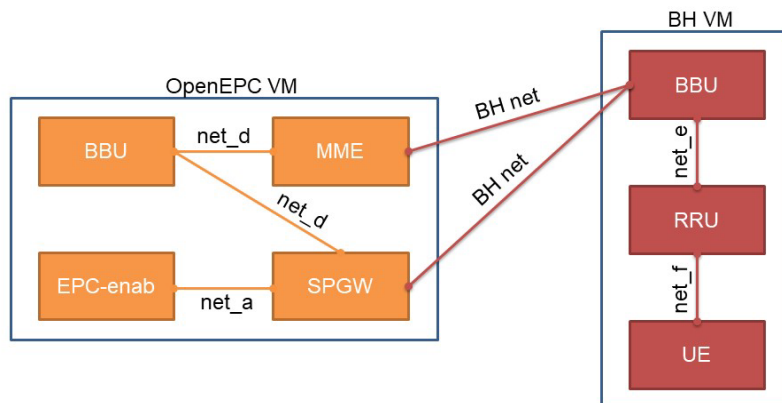


Figure 36. Interaction between BH VMs and OpenEPC VMs

With respect to the OpenEPC workflow, the Provisioning Manager NBI is used to request the creation and configuration of three different network connections, according to the FH and BH virtual networks created in Openstack. The setup requests for the network connections are

performed specifying the proper QoS parameters for handling FH and BH traffic. The details of these connections, as requested to the Provisioning Manager, are listed in Table 12.

Table 12: Connections details as requested on the Provisioning Manager NBI

Connection	Source	Destination	Classifiers	QoS
FH RRU VM to OpenEPC BBU VM	FH RRU VM	OpenEPC BBU VM	Src MAC: fa:16:3e:01:f8:bd Dst MAC: fa:16:3e:3a:70:ce Vlan id: 120	Latency: 500 μ s Throughput: 102 Mbps in DL 22 Mbps in UL
BH BBU VM to OpenEPC MME VM	BH BBU VM	OpenEPC MME VM	Src MAC: fa:16:3e:e6:ef:7e Dst MAC: fa:16:3e:51:e6:64 Vlan id: 113	Throughput: 50 Mbps in DL
BH BBU VM to OpenEPC SPGW VM	BH BBU VM	OpenEPC SPGW VM	Src MAC: fa:16:3e:e6:ef:7e Dst MAC: fa:16:3e:b9:5a:57 Vlan id: 113	Throughput: 50 Mbps in DL

Note that, in addition to these three connections, the Provisioning Manager has been used also for establishing two further connections on the management virtual network, in order to reach VMs via Controller Node (i.e. OpenEPC to gateway and BH VM to gateway).

Once the Provisioning Manager has received the requests for the setup of the FH and BH connections, it delegates the paths calculation to the Path Computation module within the ODL-based SDN Controller. The EMMA algorithm, implemented within the Path Computation module, elaborates the Provisioning Manager's requests according to the specified QoS and returns the computed paths with steps details (i.e. ingress port, egress port and queue to be configured for each node along the path). Then, through the SDN Controller SBI, the Provisioning Manager installs for each path, and steps within it, the proper encapsulation / matching / decapsulation XPFE's pipeline rules, actuating the prioritisation of the traffic by the configuration of the proper output queues as determined by the EMMA algorithm.

The computed paths are represented in Table 13.

Table 13: Connection details as computed by the Path Computation Manager

Connection	Path	Queue
FHRRUtoEPCBBU	XPFE 3 – XPFE 9 – XPFE 2	5
BHBBUtoEPCMME	XPFE 1 – XPFE 9 – XPFE 2	2
BHBBUtoEPCSPGW	XPFE 1 – XPFE 9 – XPFE 2	2

Id	Tenant	Priority	Source
FHRRUtoOpenEPCBBU	tenant1	5	in-port: openflow:3:1 vlan: 120 destination address: fa:16:3e:3a:70:ce
BHBBUtoOpenEPCMM rev	tenant1	2	in-port: openflow:2:1 vlan: 113 destination address: fa:16:3e:e6:ef:7e
FHRRUtoOpenEPCBBU_rev	tenant1	5	in-port: openflow:2:1 vlan: 120 destination address: fa:16:3e:01:f8:bd
WasptoGw	tenant1	1	in-port: openflow:1:1 vlan: 121 destination address: fa:16:3e:b5:d8:09
BHBBUtoOpenEPCMM	tenant1	2	in-port: openflow:1:1 vlan: 113 destination address: fa:16:3e:51:e6:64

Figure 37. FHRRUtoOpenEPCBBU and BHBBUtoOpenEPCMM connections view in Dlux Web GUI

In order to demonstrate the different prioritisation of the FH and BH traffic, we proceed in the experiment as follows:

- FH traffic is generated from the UE VM in the FH VM to the BBU VM in the OpenEPC VM.
- Background traffic is generated from the FH VM to the OpenEPC VM.

According to the XPFE pipeline specification as reported in [48], the “set-queue” operation is applied in table 253 in the first edge node and in core nodes along the path. Observing flow table 11 (Figure 38), where the encapsulation is performed, and flow table 253 (Figure 39) in XPFE 3, along the “FHRRUtoEPCBBU” path, we can see how the traffic directed to OpenEPC BBU VM (i.e. with destination MAC address fa:16:3e:3a:70:ce) is forwarded through queue 5, while the background traffic (i.e. with destination MAC address fa:16:3e:fa:d8:15) is forwarded through a lower priority queue. The following figures present the pipeline details for table 11 and table 253 as captured from the XPFE 3 CLI. Matches against classifiers, as well as relevant encapsulation information, are highlighted.

```
{
  "table": 11,
  "flows": [
    {
      "priority": 10,
      "idle_timeout": 0,
      "hard_timeout": 0,
      "cookie": 0,
      "dl_dst": "fa:16:3e:3a:70:ce",
      "tunnel_id": "1201",
      "actions": [
        {
          "apply_actions": [
            {
              "push_pbb": 35047
            },
            {
              "pbb_isid": 1120
            },
            {
              "push_vlan": 33024
            },
            {
              "vlan_vid": 5216
            },
            {
              "dl_src": "00:30:18:c8:d5:66"
            },
            {
              "dl_dst": "00:30:18:cf:4c:39"
            }
          ]
        }
      ],
      "goto_table": 252
    },
  ],
  "stats": {
    "packet_count": 2405509,
    "byte_count": 3646751644
  }
},
```

Figure 38. FH traffic encapsulated in table 11 in XPFE 3

```
"table": 253,
"flows": [
  {
    "priority": 10,
    "idle_timeout": 0,
    "hard_timeout": 0,
    "cookie": 0,
    "metadata": "5/0x00000000000000ff",
    "dl_dst": "00:30:18:cf:4c:39",
    "pbb_isid": "1120",
    "actions": [
      {
        "apply_actions": [
          {
            "output": 3
          },
          {
            "set_queue": 5
          }
        ]
      }
    ],
    "stats": {
      "packet_count": 2405418,
      "byte_count": 3699532884
    }
  }
],
```

Figure 39. FH traffic forwarded through queue 5 in table 253 in XPFE 3

```
{
  "priority": 10,
  "idle_timeout": 0,
  "hard_timeout": 0,
  "cookie": 0,
  "dl_dst": "fa:16:3e:fa:d8:15",
  "tunnel_id": "1211",
  "actions": [
    {
      "apply_actions": [
        {
          "push_pbb": 35047
        },
        {
          "pbb_isid": 1124
        },
        {
          "push_vlan": 33024
        },
        {
          "vlan_vid": 5220
        },
        {
          "dl_src": "00:30:18:c8:d5:66"
        },
        {
          "dl_dst": "00:30:18:cf:4c:39"
        }
      ]
    },
    {
      "goto_table": 252
    }
  ],
  "stats": {
    "packet_count": 7,
    "byte_count": 622
  }
}
```

Figure 40. Background traffic encapsulated in table 11 in XPFE 3

```
{
  "priority": 10,
  "idle_timeout": 0,
  "hard_timeout": 0,
  "cookie": 0,
  "metadata": "1/0x0000000000000000ff",
  "dl_dst": "00:30:18:cf:4c:39",
  "pbb_isid": "1124",
  "actions": [
    {
      "apply_actions": [
        {
          "output": 3
        },
        {
          "set_queue": 1
        }
      ]
    }
  ],
  "stats": {
    "packet_count": 7,
    "byte_count": 776
  }
}
```

Figure 41. Background traffic forwarded through queue 1 in table 253 in XPFE 3

The same procedure is applied in the BH case:

- BH traffic is generated from the UE VM in the BH VM to the MME VM in the OpenEPC VM.
- Background traffic is generated from the BH VM to the OpenEPC VM.

As in the previous case, we can observe that the background traffic is forwarded through queue 1, while the BH traffic with destination OpenEPC BBU VM (i.e. MAC address fa:16:3e:51:e6:64) is forwarded through queue 2, then with lowest priorities with respect to the FH traffic forwarding (see Figure 42, Figure 43, Figure 44 and Figure 45).

```
{
  "table": 11,
  "flows": [
    {
      "priority": 10,
      "idle_timeout": 0,
      "hard_timeout": 0,
      "cookie": 0,
      "dl_dst": "fa:16:3e:51:e6:64",
      "tunnel_id": "1131",
      "actions": [
        {
          "apply_actions": [
            {
              "push_pbb": 35047
            },
            {
              "pbb_isid": 1115
            },
            {
              "push_vlan": 33024
            },
            {
              "vlan_vid": 5211
            },
            {
              "dl_src": "00:30:18:c8:d1:36"
            },
            {
              "dl_dst": "00:30:18:cf:4c:3b"
            }
          ]
        },
        {
          "goto_table": 252
        }
      ]
    },
    {
      "stats": {
        "packet_count": 344628,
        "byte_count": 522456048
      }
    }
  ]
},
```

Figure 42. BH traffic encapsulated in table 11 in XPFE 1

```
{
  "table": 253,
  "flows": [
    {
      "priority": 10,
      "idle_timeout": 0,
      "hard_timeout": 0,
      "cookie": 0,
      "metadata": "2/0x00000000000000ff",
      "dl_dst": "00:30:18:cf:4c:3b",
      "pbb_isid": "1115",
      "actions": [
        {
          "apply_actions": [
            {
              "output": 3
            },
            {
              "set_queue": 2
            }
          ]
        }
      ],
      "stats": {
        "packet_count": 344628,
        "byte_count": 530037864
      }
    }
  ],
}
```

Figure 43. BH traffic forwarded in table 253 in XPFE 1

```
{
  "priority": 10,
  "idle_timeout": 0,
  "hard_timeout": 0,
  "cookie": 0,
  "dl_dst": "fa:16:3e:b5:d8:09",
  "tunnel_id": "1211",
  "actions": [
    {
      "apply_actions": [
        {
          "push_pbb": 35047
        },
        {
          "pbb_isid": 1125
        },
        {
          "push_vlan": 33024
        },
        {
          "vlan_vid": 5221
        },
        {
          "dl_src": "00:30:18:c8:d1:36"
        },
        {
          "dl_dst": "00:30:18:cf:4c:3b"
        }
      ]
    }
  ],
  "goto_table": 252
},
"stats": {
  "packet_count": 110,
  "byte_count": 16475
}
```

Figure 44. Background traffic encapsulated in table 11 in XPFE 1

```

{
  "priority": 10,
  "idle_timeout": 0,
  "hard_timeout": 0,
  "cookie": 0,
  "metadata": "1/0x0000000000000000ff",
  "dl_dst": "00:30:18:cf:4c:3b",
  "pbb_isid": "1125",
  "actions": [
    {
      "apply_actions": [
        {
          "output": 3
        },
        {
          "set_queue": 1
        }
      ]
    }
  ],
  "stats": {
    "packet_count": 110,
    "byte_count": 18895
  }
}

```

Figure 45. Background traffic forwarded in table 253 in XPFE 1

End to end latency for the FH traffic is measured using the ping tool and generating traffic between the FH RRU VM and the OpenEPC BBU VM, the average of 100 sample values results in 373 μ s, compliant with the 500 μ s specified in Table 12. Furthermore, also the capability of handling GTP traffic is tested on the same path (FHRRUtoEPCBBU). The tcpreplay utility is used for forwarding a GTP pcap trace, verifying that the packets are properly handled by the XPFEs pipeline and forwarded through the proper queue (i.e. queue 5 as configured by the Provisioning Manager). The expected throughput for the FH traffic (see Table 12) is verified using the iperf tool and generating 110 Mbps of traffic along the FHRRUtoEPCBBU path and 50Mbps of traffic on a parallel path traversing the same network nodes and forwarded through queue 1. Iperf's output shown how the configured connections are capable of handling respectively 110 Mbps and 50Mbps of traffic with a minimum packets' loss percentage (see Figure 46, Figure 47 and Figure 48).

```

-----
Client connecting to 172.16.5.25, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 172.16.5.28 port 52558 connected with 172.16.5.25 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  132 MBytes  111 Mbits/sec
[ 3] Sent 94340 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  132 MBytes  110 Mbits/sec  0.025 ms  416/94339 (0.44%)
[ 3] 0.0-10.0 sec  2 datagrams received out-of-order

```

Figure 46. 110 Mbps of traffic over FHRRUtoEPCBBU connection

```

-----
Client connecting to 172.16.13.124, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 172.16.13.121 port 54229 connected with 172.16.13.124 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-15.0 sec  89.5 MBytes 50.0 Mbits/sec
[ 3] Sent 63831 datagrams
[ 3] Server Report:
[ 3] 0.0-15.0 sec  89.5 MBytes 50.0 Mbits/sec  0.050 ms  9/63830 (0.014%)
[ 3] 0.0-15.0 sec  10 datagrams received out-of-order

```

Figure 47. 50 Mbps of traffic over a priority 1 connection

```

[ 4] local 172.16.13.124 port 5001 connected with 172.16.13.121 port 54229
[ 3] local 172.16.5.25 port 5001 connected with 172.16.5.28 port 52558
[ 3] 0.0-10.0 sec  132 MBytes 110 Mbits/sec  0.026 ms  416/94339 (0.44%)
[ 3] 0.0-10.0 sec  2 datagrams received out-of-order
[ 4] 0.0-15.0 sec  89.5 MBytes 50.0 Mbits/sec  0.051 ms  9/63830 (0.014%)
[ 4] 0.0-15.0 sec  10 datagrams received out-of-order

```

Figure 48. Iperf server output for traffic over FHRRUtoEPCBBU and parallel connection

Finally, from the power consumption saving point of view, in this limited scenario composed of 6 XPFEs and 2 XPU, we are able to keep in sleeping mode just 2 network nodes (see Table 13), with a total power consumption saving of nearly 13% without traffic and with respect to the all-on approach.

2.5.2 Experiment #7: EMMA resource management over mmWave mesh

EMMA resource management over mmWave mesh	
Description	This experiment demonstrates the functionality of the EMMA algorithm described in D4.2. The EMMA interfaces with an ODL SDN Controller to reconfigure traffic flows and enable/disable mmWave links/nodes based on injected traffic conditions. Traffic parameters will be generated within the EMMA and pushed to the nodes to emulate the usage of the network by virtual users. The ODL SDN Controller will forward the power control messages to the power control unit as well as the traffic flow configurations to the OpenFlow switches respectively. At any time, the user traffic demands have to be fulfilled and the overall network power consumption is optimized as described in [50].
State of the art	Current wireless meshed networks are operated in an always on nature to provide connectivity at all time. Green HetNets are discussed for 5G under many aspects [34] as well as dynamic switching of access nodes under power optimisation constraints [35][37]. For outdoor operation within the mmWave bands, the channel model is under research and some experiments are already conducted [36].
Improvement from State of the art	As the proposed HetNet structure in [33] has the overlaying always on network already covered, the mmWave mesh can be power optimised under the defined constraints. The proposed EMMA algorithm in [33] will optimise the network power consumption while maintaining user satisfaction for any given traffic

	demand. Nodes/Links can enter and leave the network in an automated way and routing is always chosen in an optimised way. The combination of traffic and energy management mechanism is beyond SoTA.
5G-Crosshaul Use Cases	Dense Urban Society
System under Test	<p>The diagram illustrates a 5G-Crosshaul system architecture. At the top, an 'Energy-Management and Monitoring Application' block contains a 'virtual traffic demand' component. This application is connected to an 'SDN Controller' block. Below the SDN Controller is a 'Power Distribution Unit'. The network consists of four nodes: Node1, Node2, Node3, and Node4. Each node contains an 'OpenV Switch' and is connected to two antennas. The nodes are interconnected via mmWave links (represented by blue lines with antenna icons). The SDN Controller and Power Distribution Unit are connected to all nodes via control and power lines. A legend at the bottom right identifies the connection types: power (red line), ethernet (green line), mmWave (blue line with antenna icons), and control (black line).</p>
Project Objectives / 5GPPP KPIs addressed	<p>Obj. 7: Reduce energy consumption in the 5G-Crosshaul by 30% through energy management.</p> <p>Obj. 8: Orchestration of 5G-Crosshaul resources based on traffic load variations: event-driven capacity surge, broadcast services and high-speed trains.</p>
Measured KPIs	Energy consumption of nodes

Measurement tools	LEUNIG PowerSwitch 12+ Power Distribution Unit (PDU) has internal power meters per socket
Measurement procedure	<p>System will be setup and configured accordingly to “System under Test”. Measurement will focus on functionality on EMMA. Control over traffic flows as well as power states of the Nodes/links is the key focus.</p> <ol style="list-style-type: none"> 1. All links are active and configured 2. EMMA calculates optimised traffic flows 3. EMMA sets traffic flows / disables not used flows by sending instructions to SDN Controller 4. SDN Controller reconfigures flows in OpenFlow switches 5. EMMA disables links/nodes by instructing SDN Controller 6. SDN Controller sends power off/on commands to power control unit 7. Power control unit cuts power to advised control boards, thus disabling the link/nodes completely
Constraints	Traffic capacity of links are limited by forwarding and rewriting in the Control Boards as well as selected modulation and coding scheme (MCS) level on mmWave radios.
Main results	<p>Functional operation of EMMA for mmWave mesh algorithm presented in D4.2 on real network consisting of 4 physical nodes with 4 links arranged in a rectangle. Implementation of specific mmWave functions for real world operation and configuration of physical network to measure and demonstrate power savings up to 70%.</p> <p>Establishing operating mmWave mesh network with traffic capabilities for backhaul multihop data routing.</p>

We set up a mesh network composed of 4 nodes as shown in “system under test”. Each node is attached with 2 WiGig modules, so that 4 WiGig links are configured in this network. For minimizing the effect of interference, Channel 2 (frequency 60.48 GHz, bandwidth 2.16 GHz) is used for Node1-Node2 link and Node3-Node4 link, and Channel 3 (frequency 62.64 GHz, bandwidth 2.16 GHz) is used for the others. All links are set up with modulation coding scheme 9 according to the WiGig standard.

The SDN controller as well as the EMMA application are running on separate machines, which are connected together via the control network. The power distribution unit is also part of this control network. This completes the control-plane for this setup.

The EMMA application, also called optimizer, is implemented in MATLAB as discussed in D4.2. We consider it as an over-the-top application which uses a translation program written in python to convert the commands to REST API calls towards the controller.

The Controller for this demo is composed of various modules as depicted in Figure 49. The main goal for this controller is the management of the mmWave mesh network, configuring the network nodes regarding its traffic forwarding and mmWave link configurations.

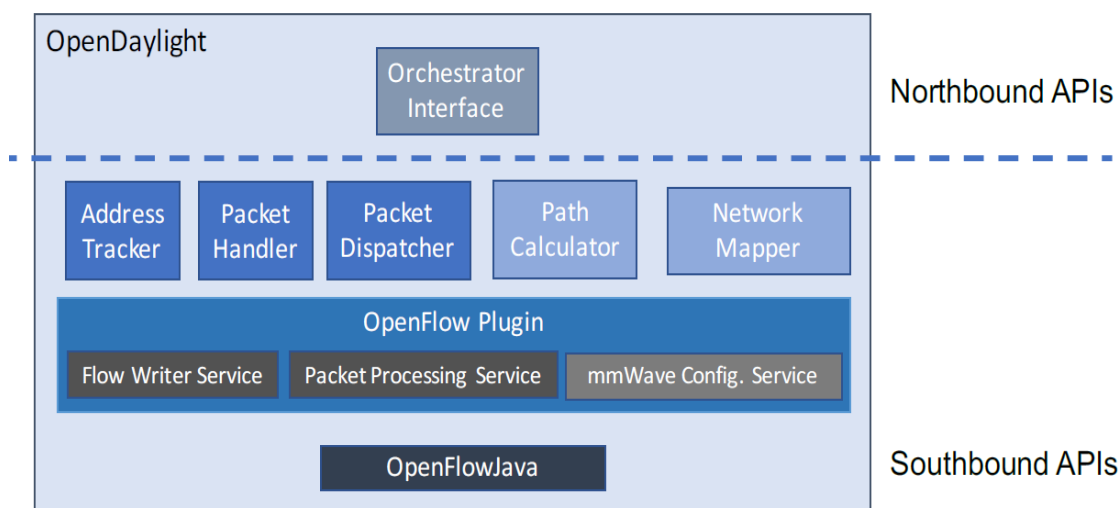


Figure 49. mmWave Controller architecture

Modules

- Packet Handler

The Packet Handler processes OF PacketIn messages and parses the payload of the received packets, translating them into the respective L3/L4/L5 packet datastructure (e.g. IPv4, TCP or UDP packets). Upon such datastructure creation, new notifications are generated, allowing other modules to access the new created data packet.

- Address Tracker

The Address Tracker module is responsible for the mapping of the observed network MAC and IP addresses to the corresponding OF switches. Whenever an OF PacketIn message is received from an address observed for the first time, a datastructure containing the new addresses is created and associated with the OF port where the packet was received. This mapping aids L2 and L3 routing in the managed network by making possible the translation of a source/destination address to a OF node.

- Packet Dispatcher

The Packet Dispatcher sends packets to OF switches through OF PacketOut messages, by using the Packet Processing Service. This module is used whenever a PacketIn message is received and the message payload is sent to the destination OF switch.

- Network Mapper

A network graph is maintained by the Network Mapper module, having the OF switches as vertices and the existing links as edges. This management is made by listening to changes (creation or deletion) of Link datastructures, triggered by an internal Topology Manager service, present in OpenflowPlugin that uses the Link layer discovery protocol (LLDP) for performing Link monitoring in the managed network.

- Path Calculator

Based on the data provided by the Network Mapper, the Path Calculator calculates routes whenever forwarding rules need to be installed for an unmatched flow. Whenever a path

is calculated, the respective forwarding rules (based on a source/destination IP address match) are installed in all the nodes among the path, through the Flow Writer Service.

- **OpenFlow Plugin**

The OpenflowPlugin provides multiple services that allow an ODL application to communicate with the OF switches with OF messages. Examples of these services include flow management, statistics management and packet delivery. The following subsections list the most important components used in this application.

- **Flow Writer Service**

The Flow Writer Service is responsible for installing new forwarding rules in OF switches and sending the corresponding FlowMod messages. These messages include the flow match rules (e.g. L2 address, TCP/UDP ports) and the actions to take, such as dropping the packet, modifying some of its header fields or sending it to a different port.

- **Packet Processing Service**

The Packet Processing Service creates PacketOut messages and sends them to network nodes upon request by the application. These PacketOut messages can contain a payload from a previous PacketIn message or data generated by the controller.

- **mmWave Configuration Service**

The mmWave Configuration Service (mmWCS) was introduced with the goal of configuring the links of mmWave mesh nodes. This configuration is done by sending newly created OF messages for this purpose. Currently, the supported configuration messages can configure an IEEE 802.11ad link by specifying its SSID, security options and beam options (MCS and Beam-ID).

- **Orchestrator Interface**

The Orchestrator Interface provides a REST Northbound API that can be used by other entities, such as the Optimizer to execute network configuration commands. The current interface API supports the following operations:

- **mmWave link configuration**

The configuration of a mmWave link can be issued by providing the input data used by the mmWCS to make this type of configuration.

- **Single forwarding rule installation**

A single L3-match rule is installed in the input provided OF switch. Additionally, the instructions of the created forwarding rule can include the modification of the source and destination MAC addresses, due to the compatibility in the wireless environment. Multiple calls to this method can be used to set up forwarding rules across a network path.

- **Virtual User routing**

This method allows the SDN controller to receive a list of Virtual Users (VU) and their used IP addresses and location in the network (which mmWave mesh node they are connected to), and the routes among them. These routes (ordered

sequence of OF nodes with a source and destination VU) are then used to write forwarding rules among the involved nodes. This method is used to pro-actively prepare the network state when traffic generation occurs.

- OpenFlowJava

The OpenflowJava (OFJ) module contains the specification of all the OF messages, including special functions for serializing and deserializing (convert java objects to byte stream and vice versa), as well as all the necessary classes to establish OF communication channels. Besides the messages in the original OF 1.3 specification, the additional message types used by the mmWave Configuration Service were also implemented in the OFJ.

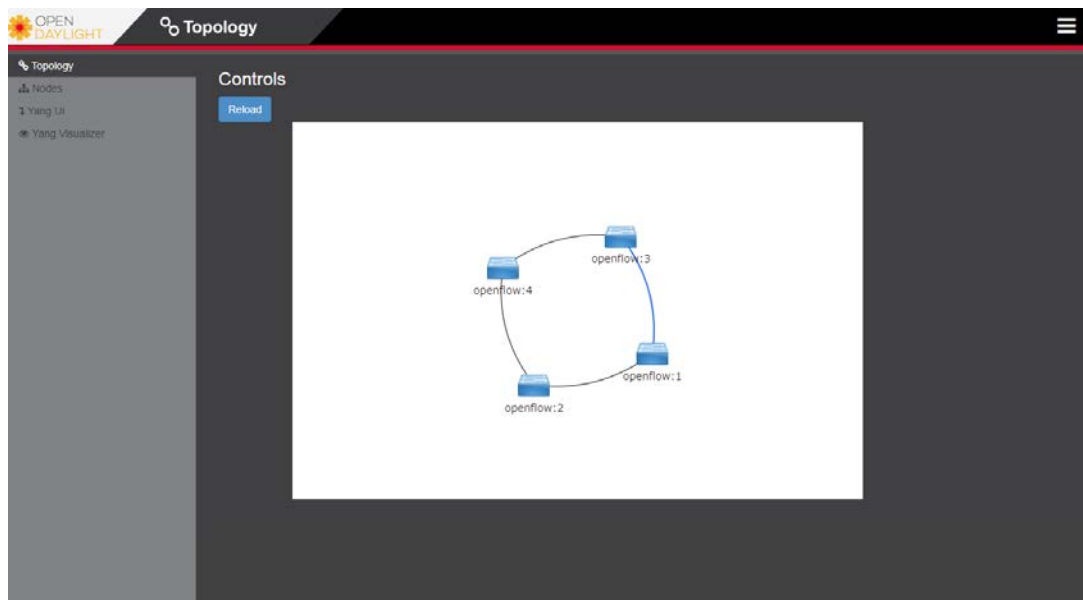


Figure 50. mmWave topology in XCI GUI

As shown in Figure 50, the topology of the mmWave mesh after setup is visible and known within the XCI. After the definition of traffic demand in the optimizer module, the EMMA algorithm will calculate the most suitable network configuration and set the flows accordingly.

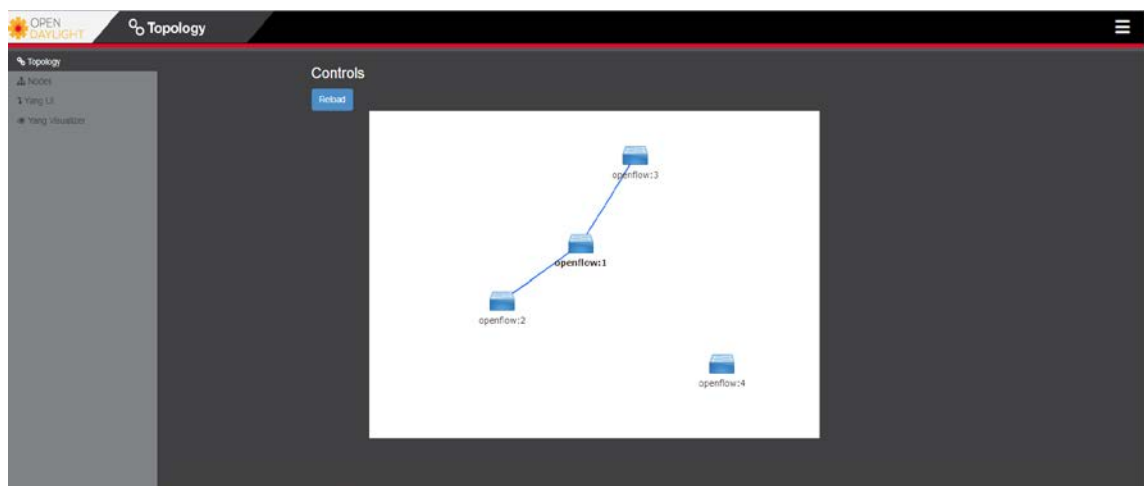


Figure 51. mmWave controller topology after EMMA path configuration

An example of a configured network, with path configured through node 1,2 and 3 is depicted in Figure 51. As one can see, node 4 is not handling any traffic routes anymore. So the power to node 4 can be disconnected via the Power Distributions Unit (PDU). This is the second step after all the traffic routes are set. The EMMA sends a power disable command to the python translator, which generates the corresponding SNMP packet to disable the associated port within the PDU for disabling the power distribution to node 4. Since we are not implemented a graceful shutdown yet, the disabling of node4 is more or less instantaneous.

The power consumption is measured in the following table as well as the savings of gradually disabling nodes from the network. All nodes have identical hardware and software configuration. The different baseline power consumptions between Node1, Node2 compared to Node3 and Node4 can be explained to slightly different workload of the operating system (ssh sessions, background tasks). Also some inaccuracy of the power meter in low load scenarios have to be taken into account as well as the rounding towards wattage within the PDU based on ampere measurements on 230VAC.

Table 14: Power measurements and savings in mmWave mesh

Power Node1 [W]	Power Node2 [W]	Power Node3 [W]	Power Node4 [W]	Power Sum [W]	Saving
12±1	12±1	11±1	11±1	46	0%
13±1	12±1	11±1	0	36	21%
12±1	12±1	0	0	24	47%
12±1	0	0	0	12	74%

The power consumption readings within the table are during idle operation of the mmWave mesh nodes. When traffic is present, the power consumption of the nodes increases due to higher system load. As stated above each packet needs to be processed to rewrite MAC addresses for passing through an mmWave link. Figure 52 shows a typical power reading over time while all nodes within the mmWave mesh handles real IP-based traffic with BH traffic characteristics.

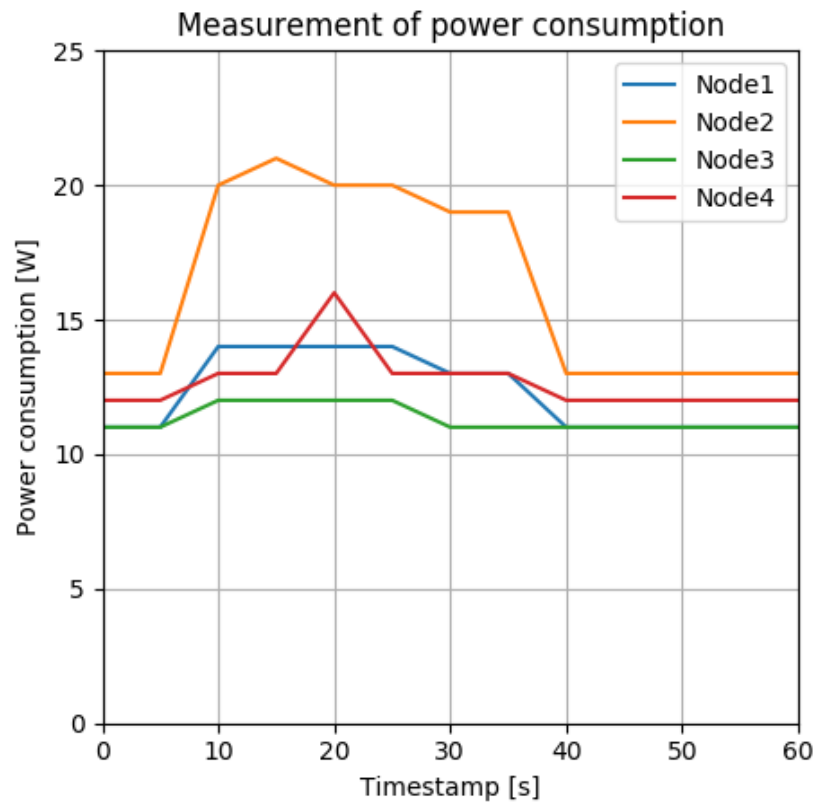


Figure 52. mmWave mesh power consumption during high traffic configuration

The traffic is generated locally within the nodes and routed via the mmWave interfaces to their respective destination. To clarify the traffic is only single-hop in this scenario, meaning each node talks to the connected nodes as the topology in Figure 50. It is visible that during high load of traffic the mmWave node2 doubles in power consumption easily while node3 is almost operating at idle consumption level. The different power levels correspond to the different workload of the nodes during traffic. The main focus here is the functional EMMA and further investigation of the power-behaviour during high traffic load is needed. The power readings are collected via the used LEUNIG PowerSwitch 12+ PDU.

3 Experimental evaluation results for media distribution exploiting the services of the Crosshaul (Demo 2)

Demo 2 is focused on the distribution of media content over the 5G-Crosshaul architecture. In this sense, this demo shows the deployment of Video on Demand (VoD) and TV broadcasting for efficient media distribution in a dynamic and automatic way exploiting the main capabilities of 5G-Crosshaul Control infrastructure (XCI) and the 5G-Crosshaul applications. The main goal is to demonstrate that the 5G-Crosshaul network is suitable for media distribution, does an efficient and coherent use of the network resources and facilitates the improvement of the quality of service (QoS). These features will allow providing services of Video on Demand (VoD) and live video on the same infrastructure.

From the XCI services perspective, the provisioning time and the self-healing capabilities have been evaluated. For Virtual Content Distribution Network (vCDN) services, the provisioning time varies from 157 s for a vCDN with one replica server to 217 s for a vCDN with four replica servers. The biggest contributor to this process is the time needed by the virtual machines to boot and complete the network configurations, ranging from 118 s to 135 s for vCDNs with 1 to 4 replica servers. The self-healing worst case scenario would require a new replica server due to a detected network failure. This new instantiation takes 45.641 s where 30 s of them correspond to the time needed by the VM to boot and apply network configurations. Regarding the load balancing offered by the vCDN, the results have shown that the values obtained for latency and throughput are better when the user is assigned to a replica server closer to the user's location (0.771 ms for latency and 749 Kbits/s for throughput) than when the user is assigned to a server located in different subnetworks (20.888 ms for latency and 354 Kbits/s for throughput).

In the same way, for the TV Broadcasting Application (TVBA), the average and standard deviation of the multicast provisioning time when the TVBA quality probe (TVBAQP) was not previously deployed are $92.72 \text{ s} \pm 4.54 \text{ s}$ ($8.58 \text{ s} \pm 0.37 \text{ s}$ when it was pre-deployed) and it consist of: 1) average user's provisioning time of $1.47 \text{ s} \pm 0.41 \text{ s}$, 2) average Quality Probe's deployment time of $83.62 \text{ s} \pm 4.47 \text{ s}$, 3) average Quality Probe's starting time of $0.33 \text{ s} \pm 0.13 \text{ s}$, and 4) average Quality Probe's analysing time of $5.72 \text{ s} \pm 0.46 \text{ s}$. On the other hand, the TVBA can self-heal the system in an average time of $41.67 \text{ s} \pm 7.28 \text{ s}$ consisting of: 1) average reaction time of $15.88 \text{ s} \pm 9.10 \text{ s}$, 2) average TVBA's decision time of $0.16 \text{ s} \pm 0 \text{ s}$, 3) average solution time of $1.72 \text{ s} \pm 0.39 \text{ s}$ and 4) same Quality Probe's times as before.

Finally, the measures obtained for the Live Content Distribution through a vCDN are populated from the two previous set of results:

- The total provisioning service time is composed of the instantiation time of a vCDN plus the TV Multicast service deployment time. For a vCDN initially configured with 2 replica servers, the total provisioning time is 249.72 s.
- The self-healing time depends on whether a new replica server needs to be instantiated due to monitoring alerts detected by the CDNMA. If the CDNMA has to deploy a new CDN node, the total self-healing time is 87.27 s. However, if only the TVBA needs to configure a new path, this time is 41.67 s.

In summary, this demo contributes to fulfil the project and 5G PPP KPIs; in particular, it demonstrates the ability of 5G-Crosshaul to meet the objectives concerning service deployment time in hours (minutes in this case, for media services) and orchestration of 5G-Crosshaul resources based on traffic load variations.

3.1 Demo 2 setup

This demo uses several building blocks and interfaces defined in the different planes of the 5G-Crosshaul architecture. Figure 53 shows both the building blocks involved in the demo and the control and the data plane interfaces between the different blocks.

The application plane includes four applications: TV Broadcasting application (TVBA), Mobility Management application (MMA), Resource Management application (RMA) and Content Delivery Network Management application (CDNMA). The control plane includes the elements from the Crosshaul Control Infrastructure (XCI), namely: the SDN controller, the VIM, the Virtual Network Function Manager (VNFM) and the NFV orchestrator. All of these applications and the XCI elements run in different virtual machines deployed on the servers installed in the testbed.

The data plane consists of:

- 6 XPFE (PCs) using Lagopus.
- 3 XPU (high-performance servers) where the CDN nodes (origin and replica servers), the TVBA Quality Probes (TVBAQPs) and the Video Headend are installed as virtual machines (VMs).
- 1 or 2 Wi-Fi access point
- 1 or 2 User Equipment (UE) (a laptop).

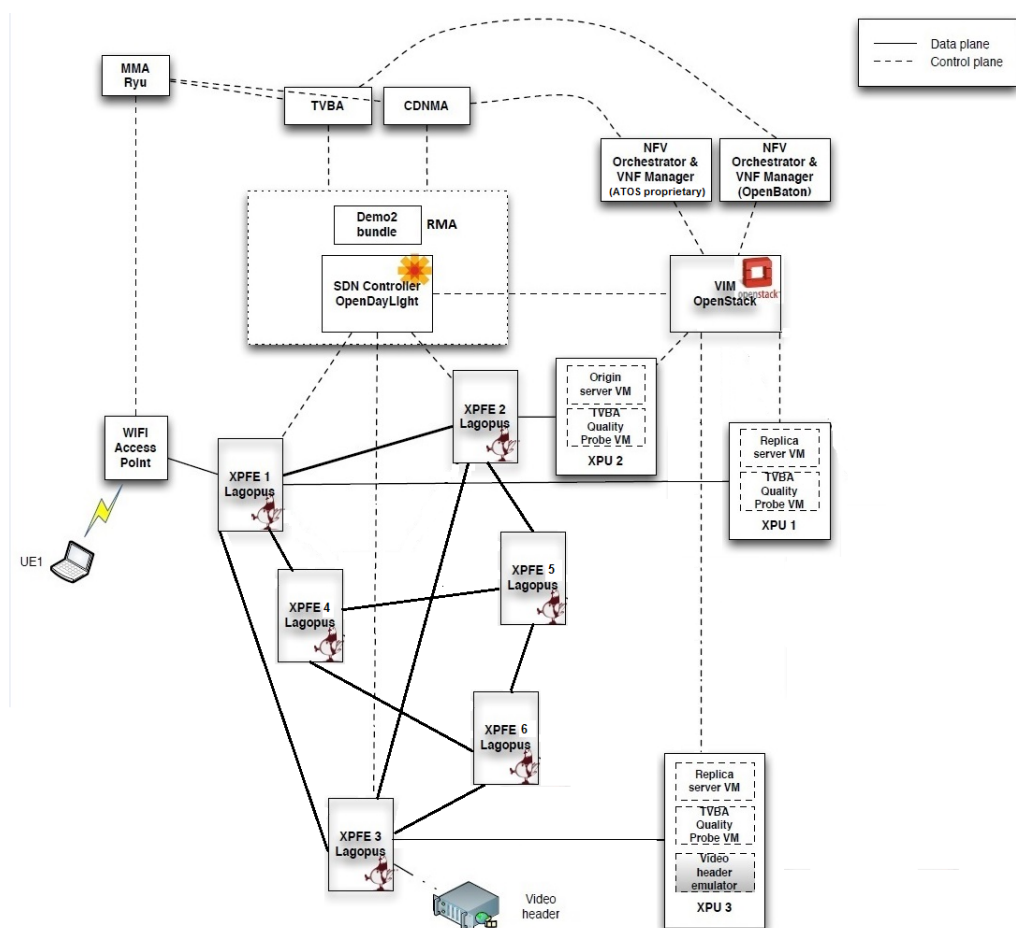


Figure 53. Setup of demo 2

The XPFEs are connected in a mesh network, and three of them are connected with an XPU. It is important to highlight that the content of the replica servers connected to the XPUs 1 and 3 is synchronized through the data network with the origin server that is connected to the XPFE number 2. The Wi-Fi access point, which provides connectivity to the UE1, is connected to the XPFE 1.

TVBA and CDNMA feature their own GUI to be used by the operator having full visibility of the environment.

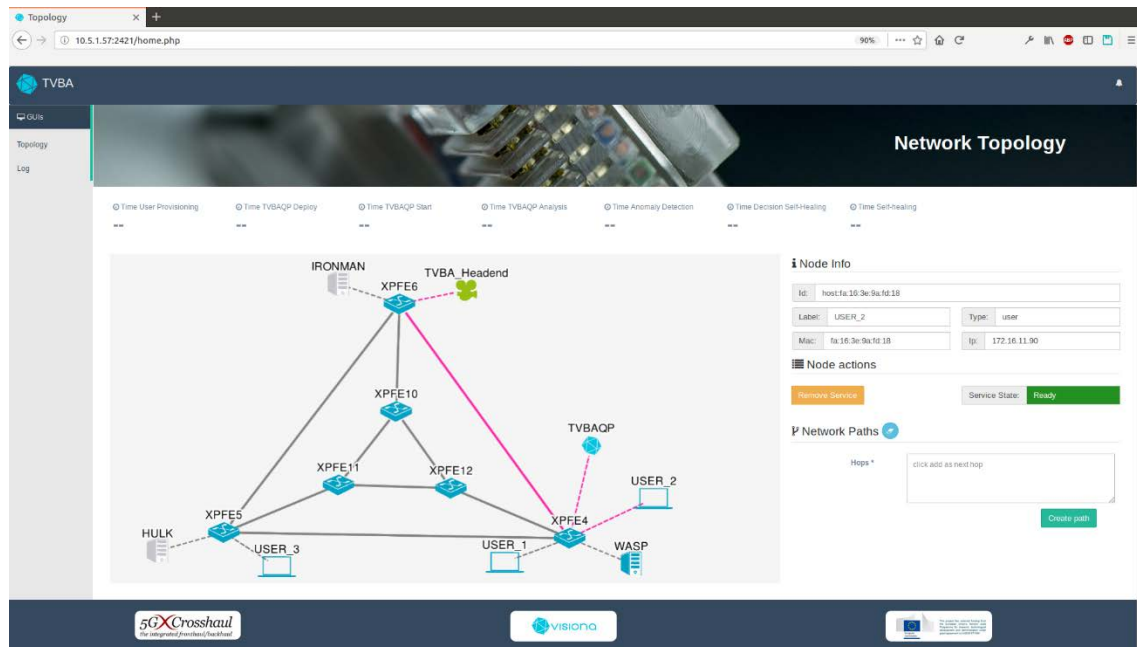


Figure 54. GUI for TVBA showing the environment

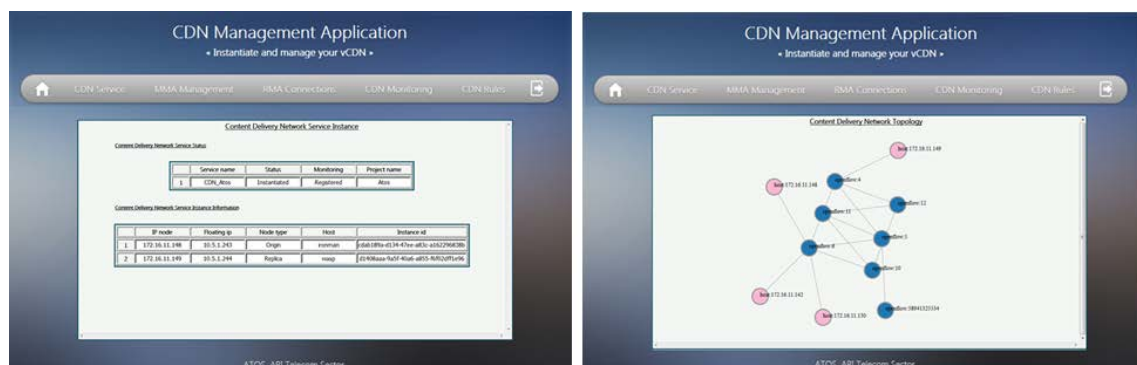


Figure 55. GUI for CDNMA showing the service configuration and the CDN topology

3.1.1 Building blocks from other work packages

Demo 2 focuses mainly on the application and control plane, operating on the network topology to configure media distribution services in an optimal way taking into account the network infrastructure and its capacity, as developed in WP2 and WP3. Several building blocks from WP3 and WP4 also allow orchestrating the resources needed to launch the services and routines of

analysis and control, computing optimal paths and controlling the network flows through the SDN Controller.

Data Plane building blocks

WP2 / WP3 (T3.1)	
Component / Entity	Description
Software based XPFES	XPFES based on Lagopus software (v0.2.10). The XPFES are deployed in a mesh topology of six nodes in the testbed. They implement the XCF defined in [48] and support multi-tenancy through a dedicated OpenFlow pipeline.
XPU	Standard IT servers. Several virtual machines run on these servers supporting the CDN nodes and TVBA Quality Probes.

Control and Application plane building blocks

WP3	
Component / Entity	Description
SDN Controller	This module controls the underlying network elements following the SDN paradigm. Within the scope of Demo2, there is a SDN controller based on OpenDaylight, extended by the RMA for Path Computation and Provisioning and with rules to control the XPFES .
NFV-MANO	These modules manage and orchestrate the network resources for the over-the-top applications. They instantiate, configure, start, update, stop, release and manage the lifecycle of the VMs and their VNFs through the NFVO, VNFM and VIM components. OpenStack is used as VIM while a proprietary algorithm and OpenBaton are used as NFVO and VNF Manager for CDNMA and TVBA respectively.

WP4	
Component / Entity	Description
TVBA	TV Broadcasting Application is responsible for defining and provisioning the live video service that will allow the broadcasting of multimedia content over the network and reach the user with the desired QoS and QoE. It uses the SDN Controller and the RMA to establish the necessary network rules and to create a broadcasting-tree; and the NFV-MANO to manage the TVBA Quality Probes, which are VNFs deployed in the XPU to perform a media quality analysis in real time to ensure QoS and QoE near the user.

CDNMA	The Content Delivery Network Management application is a Web application composed of two algorithms that allow the CDN service management and implementation over the 5G-Crosshaul network. The first algorithm is in charge of the vCDN infrastructure instantiation, and the second one is oriented towards the control and management of the service during its lifetime. The CDNMA application shows the service provider/CDN operator a graphical user interface (GUI) for management and operation actions. The CDNMA GUI shows a graph of the network topology, with the nodes, hosts and links as well as monitoring and configuration information about the vCDN infrastructure. The application allows the service provider/CDN operator to control and manage the service instantiation and the service lifecycle management.
RMA	The Resource Management Application through its own algorithms and the XCI, computes and provisions via the SDN controller the optimal paths followed by the mobile users' traffic to support the VoD and live streaming services provided by the CDN and TVB Applications. Initial and end nodes in the 5G-Crosshaul network, service bandwidth, latency and optionally priority among other in-parallel-deployed services are provided to the application as input parameters.
MMA	The Mobility Management Application is Distributed Mobility Management (DMM) based and its main goal is to optimize traffic offloading for media distribution like CDN and TV Broadcasting. The MMA application provides information about the point of attachment (PoA) of the user to the network, such as the PoA's MAC address and the user's MAC and IP addresses, that allows optimizing the path followed by mobile users' traffic.

3.1.2 Additional building blocks

Component / Entity	Description
TVBA Headend	TVBA headend manages audio and video encoding for TV services and injects the traffic into the 5G network. The headend service includes video coding for MPEG-4 AVC/H.264 services in HD (High Definition).
High Definition TV	Full HD TV for service monitoring
TVBA Quality Probe (TVBAQP)	This tool is deployed dynamically on the XPU's of the network to analyse the quality of video broadcasting experienced by final users. The TVBA Control Panel uses this information for making relevant decisions regarding the broadcast-tree deployed to deliver the content.
User video player	Web player that shows the video from the selected replica server based on the content delivery rules.

3.2 Experiments on media distribution

3.2.1 Experiment #8: Virtual CDN service on the 5G-Crosshaul infrastructure

Virtual CDN service on the 5G-Crosshaul infrastructure	
Description	<p>Deployment of a virtual CDN service across the 5G-Crosshaul infrastructure in which the CDN functions (CDN nodes and load balancer) are dynamically allocated and managed through the CDNMA functionalities. There are two main workflows followed to carry out the service deployment.</p> <p>In the first one, the CDNMA interacts with the XCI NBI, SDN controller and NFV MANO, in particular the NFVO (Orchestrator), in order to deploy the vCDN infrastructure where the CDN functions are instantiated on virtual machines, deployed within the XPU's located close to the end users.</p> <p>In the second one, the CDNMA interacts with the MMA and RMA to manage the vCDN service, providing an efficient and flexible way of delivering content, especially (VoD), and dealing with the massive content requests from the end users, controlling the load balancing over several CDN nodes strategically placed at various locations. The CDNMA receives the point of attachment of the user to the network from the MMA and based on this information, the monitoring of the different CDN nodes and the policies defined by the CDN operator, it decides the best CDN node to serve the user request. The RMA will compute and provision through the SDN controller the route or path followed by the mobile users' traffic between the CDN node assigned and the point of attachment of the user to the network.</p>
State of the art	<p>Content Delivery Networks (CDNs) replicate contents over several mirrored servers placed at various locations in order to deal with heavy load. The entire network in CDNs is managed by a single authoritative entity. CDN cache servers are hardware appliances. The deployment is static and the scale in/out is a difficult process to manage.</p> <p>Nowadays, in parallel with the development of this project, institutions and companies outside this project have been working and making developments with CDNs that take advantage of the virtualization paradigm to provide on-demand content (vCDNs) using current networks. Further information about CDN state of the art can be found in [23]</p>
Improvement from State of the art	<p>The dynamic vCDN infrastructure exploits the virtualization capabilities and leverages NFV platforms to provide an efficient and flexible way of delivering content to the end users, especially (VoD). The virtualization capabilities are leveraged by deploying replica servers as VNFs instead of hardware appliances. Furthermore, as well as the static infrastructure, it is capable of dealing with massive content requests, controlling the load balancing over several replica servers strategically placed at various locations aiming to reduce the global resource consumption in the network, improving the QoS for the end user and making easier the scale in/out processes.</p> <p>Dynamicity has been added in terms of recovery from network failures. The designed vCDN is able to monitor the network status and, automatically, assign the user to another existing replica server or even instantiate another</p>

	<p>replica server if there is none in use that can provide service in case the service fails due to network problems.</p> <p>Furthermore, the TVBA can be integrated so that it sends its content not to the end user but to the different CDNMA replica servers. These servers can distribute the content to the end user, making a more efficient use of the network in terms of resources and consumed energy, improving the user experience.</p> <p>The vCDN developed in the project makes use of the architecture developed for 5G Crosshaul in terms of Data Plane (use of the XPFEs and XPUs), Control Plane (use of the XCI) and the Application Plane (use of RMA and MMA).</p> <p>The 5G-Crosshaul vCDN exploits the in-network path computation provided by the RMA, whereas the MMA is capable of providing CDNMA with the user location in order to calculate the closest available replica server to provide the video streaming service to the final user.</p>
<p>5G-Crosshaul Use Cases</p>	<p>Media distribution. Video on Demand (VoD).</p>
<p>System under Test</p>	
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj.2: Specify the XCI's northbound (NBI) and southbound (SBI) interfaces</p> <ul style="list-style-type: none"> (5GPP KPI) Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours <p>Obj.8: 5G-Crosshaul key concept validation and proof of concept</p> <ul style="list-style-type: none"> (5GPP KPI) Orchestration of 5G-Crosshaul resources based on traffic load variations
<p>Measured KPIs</p>	<p>Provisioning of vCDN infrastructure (minutes) - The time required to deploy a vCDN infrastructure: deployment of VMs and configuration of the monitoring process for the replica servers.</p>

	<p>Streaming service performance:</p> <ul style="list-style-type: none"> • Throughput/Bandwidth capacity (Mbps) - Maximum reliable transmission rate that a path can provide. • Latency (ms) - Network latency is the time taken by data to transmit from one designated point (5G Point of Attachments...) to another (Core, MEC server...) • Number of retransmitted packets <p>Time of resources orchestration (minutes) - Orchestration of resources based on replica servers and network monitoring alerts.</p>
Measurement tools	<p>Wireshark to analyse the RTMP messages associated with the media content.</p> <p>Wowza Streaming Engine media server software.</p> <p>VNF probes based on Prometheus monitoring system.</p> <p>CDNMA control panel, user interface (Web app) with Video service, network topology and metrics.</p>
Measurement procedure	<ol style="list-style-type: none"> 1. CDNMA requests the network topology to SDN controller. 2. CDNMA requests the deployment of a vCDN infrastructure to NFVO, providing the Network Service Descriptor (NSD). 3. The CDN Operator checks the new vCDN infrastructure deployment. CDN nodes are instantiated in the right places and working based on the configuration defined. 4. CDNMA checks specific monitoring information from the different CDN nodes (VNFs) deployed in the network. 5. CDNMA defines the content delivery rules and updates the video player. 6. CDNMA interacts with MMA, gets the user network entry point and decides the best CDN node to serve the user request. 7. RMA operates on the 5G-Crosshaul domain (i.e. on the XPFEs) computing the path between the XPFE to which the XPU hosting the CDN node is connected and the XPFE connected to the user entry point (i.e. the point of attachment, or Wi-Fi access point in this case) based on the information provided by the CDNMA. 8. Video player serves the video from the assigned CDN node, i.e. it delivers the video from the most suitable server. 9. Scale out the vCDN infrastructure. Deployment of a new replica server and balancing the traffic between the current servers and the new server.
Constraints	<p>In a lab environment, it is not possible to show big differences in delays and throughputs.</p> <p>The connection between the replica server 3 and the other nodes is provided by a VPN connection using the Internet to establish the communication.</p> <p>Real-Time Messaging Protocol (RTMP) is the protocol used for the video streaming between the web video player and the replica servers. RTMP is a</p>

	<p>proprietary protocol owned by Adobe which released an incomplete specification of it for public use. We need to take into account that the communication latency between the player and the replica servers depends on the selected streaming protocol as well as on the network load and other network components, such as XPFEs and XPU. However, RTMP allows low-latency communications and this is a key feature for the vCDN service in the 5G-Crosshaul infrastructure.</p> <p>The dynamicity provided by the CDNMA (scale out) has been forced by the simulation of two scenarios that Prometheus monitoring system is able to detect and manage:</p> <ul style="list-style-type: none"> • There is a CPU load increase in the replica server which is streaming video • Network failure by detecting an issue in the closest XPFEs to the replica server which is streaming the video 																				
<p>Main results</p>	<p>1. vCDN Infrastructure deployment: This test is composed of four vCDN instantiations with one, two, three and four replica servers, respectively. The final demo has three replica servers at most: two of them are instantiated in the vCDN deployment scenario and the third one is created when the monitoring service detects a network failure and notifies the CDNMA to instantiate a new server to provide the video streaming. The purpose of testing the deployment of four vCDNs with a different number of replica servers is to show the behaviour of the CDNM application in terms of instantiation time.</p> <table border="1" data-bbox="440 1211 1356 1675"> <thead> <tr> <th></th> <th>XCI (NFVO- VNFM+VIM) (s)</th> <th>VMs booting & network configs (s)</th> <th>Total (s)</th> </tr> </thead> <tbody> <tr> <td>1 replica server</td> <td>39</td> <td>118</td> <td>157</td> </tr> <tr> <td>2 replica servers</td> <td>54</td> <td>123</td> <td>177</td> </tr> <tr> <td>3 replica servers</td> <td>68</td> <td>127</td> <td>195</td> </tr> <tr> <td>4 replica servers</td> <td>82</td> <td>135</td> <td>217</td> </tr> </tbody> </table> <p>The main result is that, compared with traditional CDNs, we have reduced the infrastructure deployment time from hours to minutes.</p> <p>2. CDNMA Streaming Service performance. This test is based on a vCDN infrastructure with two replica servers: one replica server located closer to the UE (replica server 1) than the other one (replica server 3), which belongs to an external network.</p>		XCI (NFVO- VNFM+VIM) (s)	VMs booting & network configs (s)	Total (s)	1 replica server	39	118	157	2 replica servers	54	123	177	3 replica servers	68	127	195	4 replica servers	82	135	217
	XCI (NFVO- VNFM+VIM) (s)	VMs booting & network configs (s)	Total (s)																		
1 replica server	39	118	157																		
2 replica servers	54	123	177																		
3 replica servers	68	127	195																		
4 replica servers	82	135	217																		

	Replica Server 1	Replica Server 3
Average Latency	0.771 ms	20.888 ms
Average Throughput	748.789 Kbps	353.841 Kbps
Total n° Retransmitted packets	0 packets	21 packets

The main result here is that the user experience is greatly enhanced thanks to the optimum placement and dynamic reallocation of the replica servers that provide the video streaming to the end users.

3. Orchestration of new resources when an alert from the Prometheus monitoring service is detected by the CDNMA. This test is based on a vCDN with one replica server which is streaming video to a user. The Prometheus monitoring services detect a failure in the network in the XPFE closest to the replica server. The CDNMA reacts and instantiates a new replica server in another XPU to guarantee the streaming to the user.

NEW REPLICA SERVER DEPLOYMENT	
Alert Analysis (s)	0.045
NFVO+Openstack (s)	13.323
CDNMA configs (s)	32.189
MMA+RMA (s)	0.084
Total (s)	45.641

The main result here is to show that CDNMA is capable of reacting to a network failure and reallocating the users to a new replica server in a magnitude of seconds.

What we have shown here is that the 5G Crosshaul network is capable of dealing with the needs of the most demanding use cases associated with video transmission and their contribution to meet the project objectives and 5G KPIs.

Discussion of results

These tests cover the 5GPPP KPIs addressed for the “Virtual CDN service on the 5G-Crosshaul infrastructure” experiment.

“vCDN Infrastructure Deployment” test

The objective of this test is to verify that the deployment of a vCDN on the 5G-Crosshaul infrastructure can be performed in the magnitude of minutes in the scenario detailed in the experiment description.

In order to test the behaviour of the CDNMA application, we have instantiated four vCDNs, each of them with one origin server and the number of replica servers varying from one to four, respectively.

The hardware features of the VMs, which host the VNFs for the origin and the replica servers, are represented by the out-of-the-box Openstack flavour “m1.small”. We have selected this flavour because gives us enough resources for the experiment requirements and besides we do not overload the testbed with heavier VMs, as the testbed resources are limited:

- 1 vCPU
- 2 GB RAM
- 20 GB Root Disk

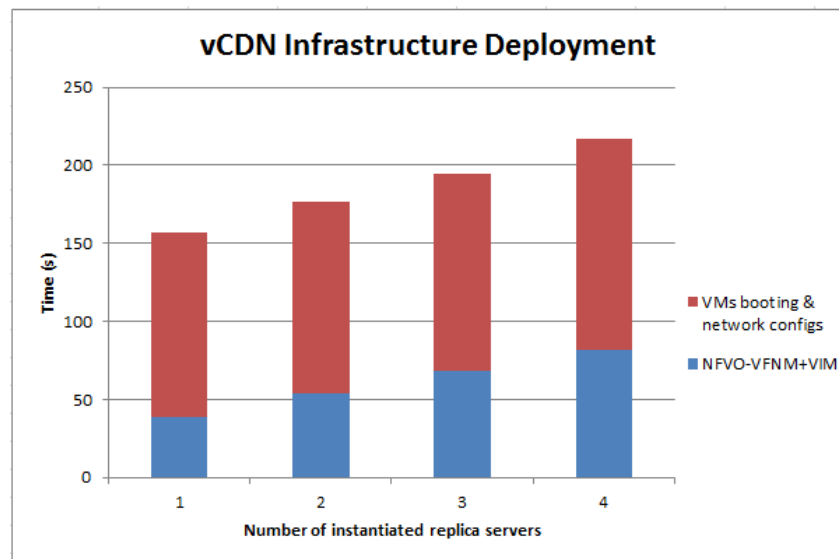


Figure 56. Deployment time for four vCDN

Figure 56 shows the time that the deployment process takes to instantiate the different vCDNs. The total time is composed of two values:

- The time that the proprietary NFVO and the VIM (Openstack) need to instantiate the different VMs (NFVO+VIM). This time ranges from 39 s to 82s for vCDN with 1 to 4 replica servers.
- The time it takes for the new VMs (origin and replica servers) to boot and apply the networks configurations. Once these actions are successfully completed, the VMs are available to provide the CDN service and interact with the CDNMA. This is the biggest contributor to this process, ranging from 118s to 135s for vCDNs with 1 to 4 replica servers.

As the number of replica servers increases the instantiation time also increases following mainly a linear trend. Therefore, knowing the deployment time for a vCDN with few replica servers will let us infer how long it would take to instantiate a vCDN with a high number of replica servers.

“Streaming Service Performance” test

This test shows the streaming performance measured from a replica server located close to the user entry point and a replica server simulated to be instantiated in an external network.

The following graph shows the comparison between the bitrate achieved during the video reproduction by both servers. The results show that the closer replica server provides better outcomes and the bitrate is clearly better comparing it to the external replica’s bitrate.

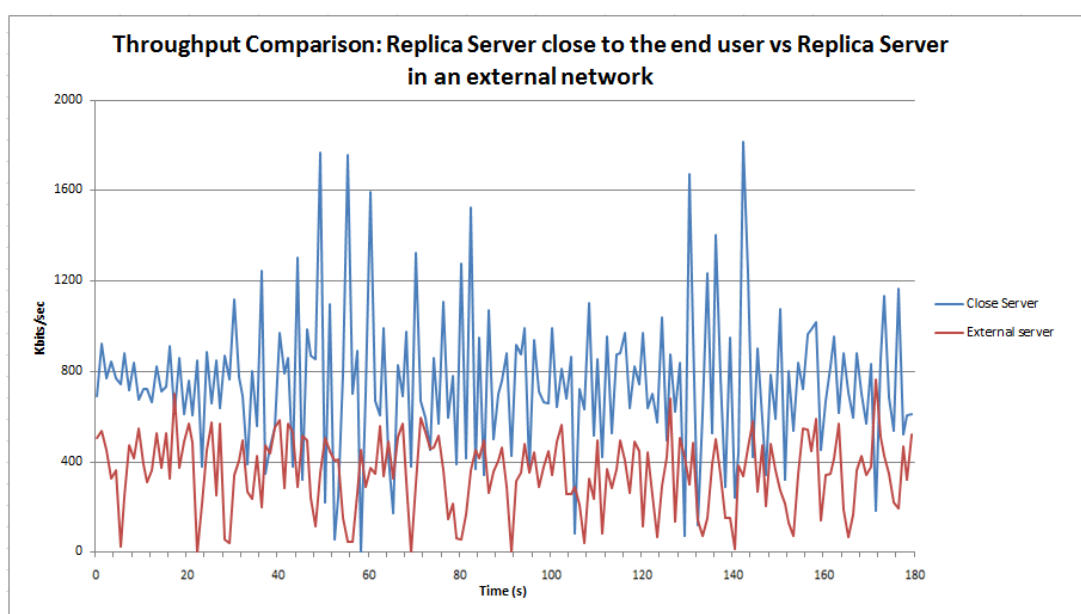


Figure 57. Throughput comparison between a close replica server and a replica server located in an external network

Taken the throughput average of the two streams, we obtain that the closer replica server (749 Kbits/s) offers a throughput 47% greater than the server located in an external network (354 Kbits/s). This is because the replica node is located in a network different from the one the user is connected to and several leaps among different networks are needed to reach the final user. Furthermore, the throughput highly depends on the external line rate, as well. So both features, location and line rate clearly impact in the obtained throughput.

“Orchestration of New Resources” test

The objective of this test is to show the dynamicity that CDNMA, supported by a monitoring system, can provide to the vCDN service on the 5G-Crosshaul infrastructure. This experiment measures the time it takes to scale out of the vCDN infrastructure, i.e., adding a new replica server when the Prometheus monitoring system detects and triggers an alert of an incident in one of the monitored parameters, and there is not any available instantiated replica server to provide the streaming.

For this test, we have simulated a network failure in the XPFE connected to the XPU where the replica server that is streaming the video to the user is running. The management of this incident is described as follows:

- The monitoring system detects this issue and sends an alert to the CDN application.
- The CDNMA receives and analyses the alert, extracting the replica server where the problem is located.
- The CDNMA checks if the impacted replica server is currently providing video streaming to users.
- The CDNMA looks for an instantiated replica server in a different XPU which meets the rules for providing the streaming service (CPU and memory load and number of connected users).
- If there is not any available replica server, the CDNMA requests to the NFVO the deployment of a new replica server.
- The CDNMA notifies the MMA the new streaming server for the pair PoA-User.
- The CDNMA requests the RMA to delete the old path and create a new one from the PoA to the XPU where the new replica server has been deployed.

Figure 58 shows the duration of this whole process, detailing the time taken to complete the different stages that are needed for the new replica instantiation. The total process consumes 45.64 s, divided in 13.32 s for the “NFVO-VFNM+VIM” phase, 32.19 s for the “VMs booting & Network configs” stage and finally, 0.084 s for MMA notification and RMA path computation. The “Alert Analysis” stage does not appear at the end of the bar because the time is so small that is not significant for the total instantiation time.

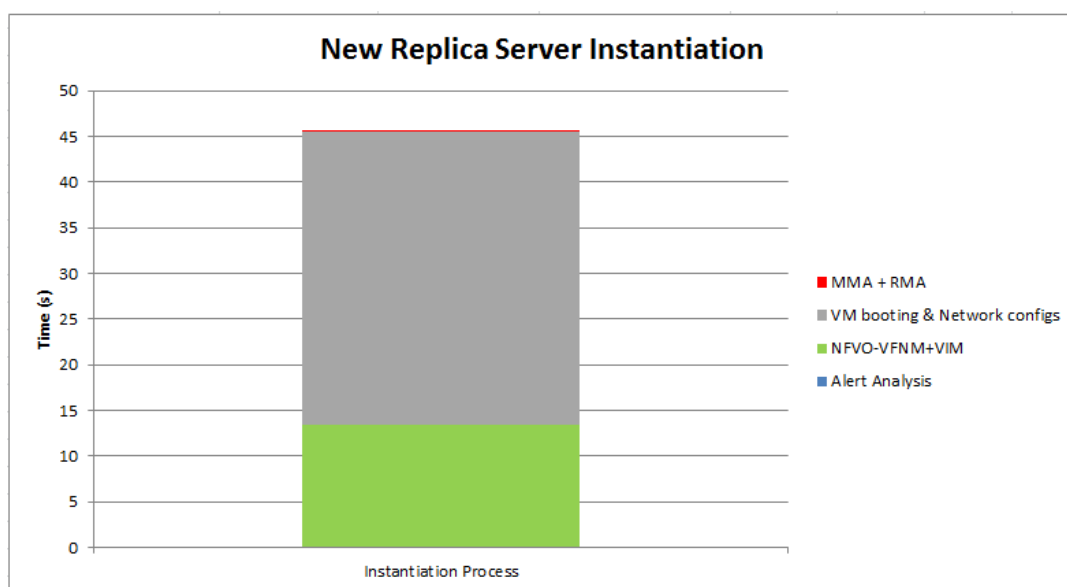


Figure 58. CDNMA - Orchestration of new resources due to network failure detection

These measures show the worst orchestration scenario, i.e., the CDNMA application needs to instantiate a new replica server because there is not any other available to provide the video to the user.

If the vCDN has available replica servers, the time that it would take to reallocate the user to the new server would be Alert Analysis+MMA+RMA.

However, if a new replica server needs to be instantiated the video streaming will be stopped for 45 seconds. Nevertheless, in a SoA CDN, the service will never be recovered meanwhile in a 5G-Crosshaul vCDN the service will be automatically reallocated.

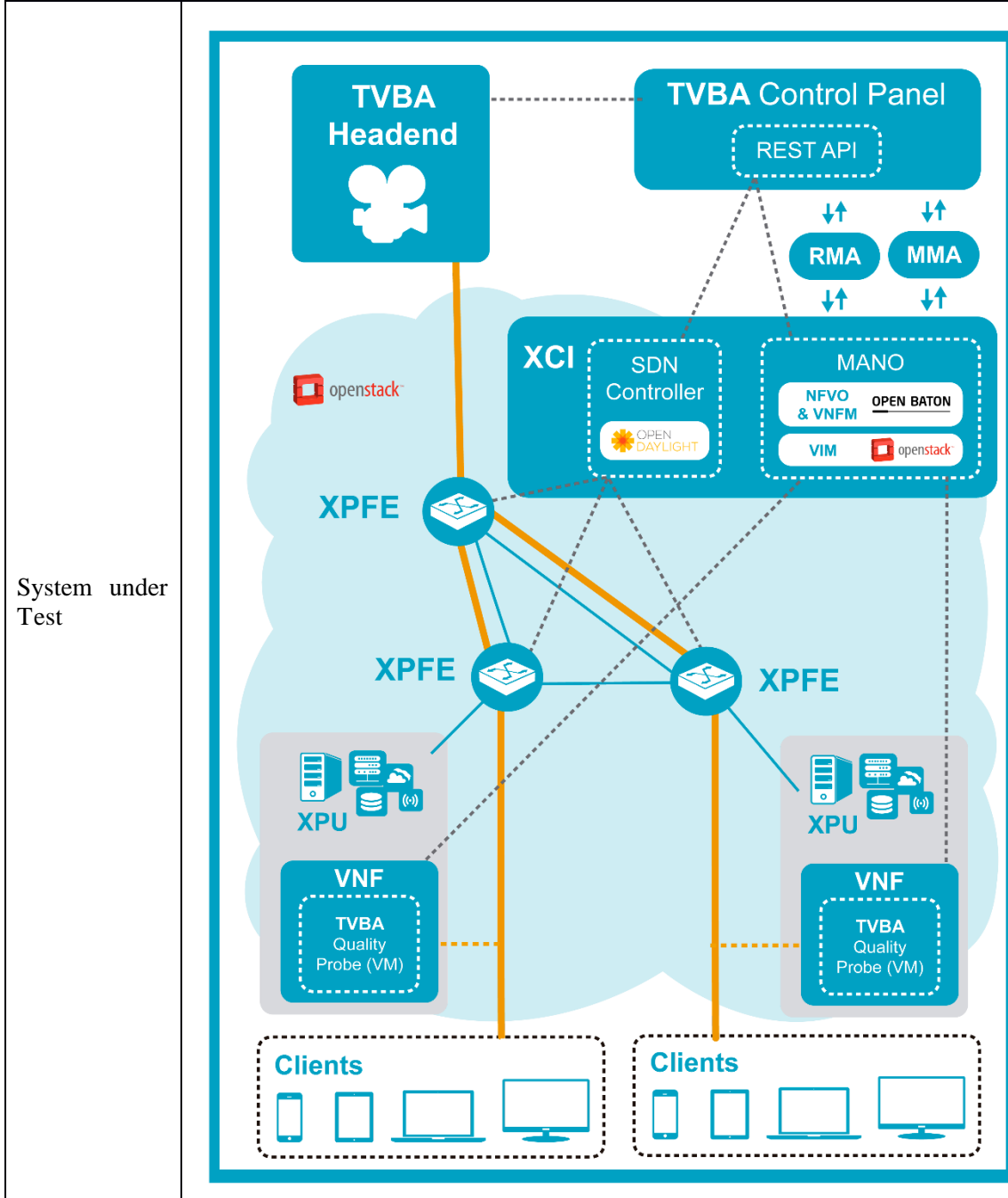
3.2.2 Experiment #9: Multicast TV service provisioning

Multicast TV service provisioning	
Description	<p>Deployment of a multicast TV service using XCI components. With the TVBA Headend streaming content to the network, a user connects to the network and it is announced by the MMA to the TVBA Control Panel. The TVBA requests a path to the RMA (which uses the SDN Controller) to generate a broadcast-tree over the network so that the content reaches the user's equipment (UE). Then [the TVBA] (using the NFV-MANO) deploys a TVBA Quality Probe (a VNF) in a XPU connected to the closest XPFE to the user in order to check the quality (in terms of non-referenced QoE and QoS) of the received media content as close as possible to the user and from then on, continuously sends the results of the quality check to the TVBA to take decisions when needed.</p> <p>From this point, the experiment is divided in 2 sub-experiments: 1) a new user is registered in the network in the same XPFE as the previous one; 2) a new user is registered in the network in a different XPFE. For the first one the TVBAQP is already deployed so service is immediately provided to the user together with the remaining part of the path. For the second one the TVBA requests a new path to the RMA to update the broadcast-tree and a new TVBAQP is deployed in other XPU closer to the new user.</p> <p>In any case there is always a single broadcast-tree.</p>
State of the art	<p>Transmitting video over IP networks is not a novel idea and nowadays several solutions are working and growing. IPTV, which was developed and deployed mainly by large telecommunication providers to compete with digital and satellite services and to fit their network constrains, is one of those solutions and it uses IP multicast.</p> <p>Although current solutions can guarantee some fixed QoS when the service is deployed over private networks as the ones of telecommunication providers, they are unable to provide a non-referenced QoE [21]; that is, a QoE metric which does not need a sample of the source to compare to. Non-referenced QoE, when achieved, gets measures that are definitely more accurate since the source, when used as reference for the comparison, could contain errors. On the other hand, even when QoS is provided for those networks, it cannot be verified without proprietary and customized set-top boxes distributed to the users, that is, limited-life hardware, which increases costs for telecommunication providers [22].</p> <p>Lately telecommunication providers have started to do trials for the deployment of Virtual Set-Top Boxes (vSTB) [51] so that this limited-life hardware will be soon virtualized in the edge of the network.</p>
Improvement from State of the art	<p>The TVBA takes advantage of the 5G-Crosshaul capabilities, based on Software-Defined Networks (SDN) and Network Functions Virtualization (NFV), and makes it extremely easy to deploy software-based Quality Probes on demand which not only verify QoS delivered to the users but also performs</p>

non-referenced QoE analysis, ensuring users' satisfaction and saving costs for telecommunication providers.

A QoS and QoE analysis using 5G-Crosshaul services would provide a flexible, reliable and universal solution for TV Broadcasting. In combination with the future vSTBs, the TVBA will drastically decrease deployment and maintenance costs not needing to send anybody to user's home at any point.

5G-Crosshaul Use Cases Media distribution.



Project Objectives / Obj. 2. Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours.

5GPPP KPIs addressed	
Measured KPIs	<p>Provisioning time (seconds) — Time required to instantiate a TVBA service:</p> <ul style="list-style-type: none"> • Provisioning time when a TVBAQP is already deployed in the targeted XPU (connected to the closest XPFE to the user). • Provisioning time when there is not any TVBAQP deployed in the targeted XPU (connected to the closest XPFE to the user).
Measurement tools	<p>Timestamps coded for every relevant process.</p> <p>TVBA logs</p> <p>TVBAQP itself: QoE and QoS information obtained to take decisions.</p> <p>2 laptops with VLC as users playing the streamed content.</p>
Measurement procedure	<p>Timestamps are placed on every process (e.g., VMs deployment, TVBAQP processing time, etc.) obtaining the total time it takes to deploy a TVBA service. Following the description of the experiment, time is measured from the moment a user is announced by the MMA to the TVBA until the TVBA receives the results from the TVBAQP. Measures are taken for both sub-experiments: when a TVBAQP is already deployed and when it has to be done.</p> <p>Direct observation of the received content on the laptops as users and TVBAQP's quality reports are observed to validate establishment of service.</p>
Constraints	<p>In order to simplify the TVBA Headend implementation and to focus on the experiment objective, the TVBA Headend has been virtualized as a VNF.</p>
Main results	<p>Average provisioning time and standard deviation when TVBAQP was not previously deployed: $92.72 \text{ s} \pm 4.54 \text{ s}$</p> <p>Average provisioning time and standard deviation when TVBAQP was previously deployed: $8.58 \text{ s} \pm 0.37 \text{ s}$</p> <p>Where:</p> <ul style="list-style-type: none"> - Average User's provisioning time: $1.47 \text{ s} \pm 0.41 \text{ s}$ - Quality Probe's deployment time: $83.62 \text{ s} \pm 4.47 \text{ s}$ - Quality Probe's starting time: $0.33 \text{ s} \pm 0.13 \text{ s}$ - Quality Probe's analysing time: $5.72 \text{ s} \pm 0.46 \text{ s}$ <p>TVBA is able to deploy a multicast service in a matter of minutes and even seconds providing non-referenced QoE and QoS monitoring services OTT improving the SOTA as expected; complementing and preparing the ground for vSTB.</p>

Discussion of results

Service provisioning time is defined as: $U_{at} + QP_{dt} + QP_{st} + QP_{at}$. Where:

- U_{at} is the *User's provisioning time*, that is, the time to establish the path from the TVBA Headend to the users using the RMA.

- QP_{dt} is the *Quality Probe's deployment time*, that is, the time a TVBAQP (a VNF) needs to be instantiated. This time is zero if the TVBAQP was already created (requested by other user) in the XPU connected to the closest XPFE to the user.
- QP_{st} is the *Quality Probe's starting time*, that is, the time a TVBAQP needs to be started once it has been instantiated.
- QP_{at} is the *Quality Probe's analysing time*, that is, the time a TVBAQP needs to analyse the media transmission looking for QoS and QoE degradation if any.

Figure 59 shows the average and standard deviation of the previous metrics for both sub-experiments previously mentioned: 1) a new user is registered in the network in the same XPFE as the first one; 2) a new user is registered in the network in a different XPFE.

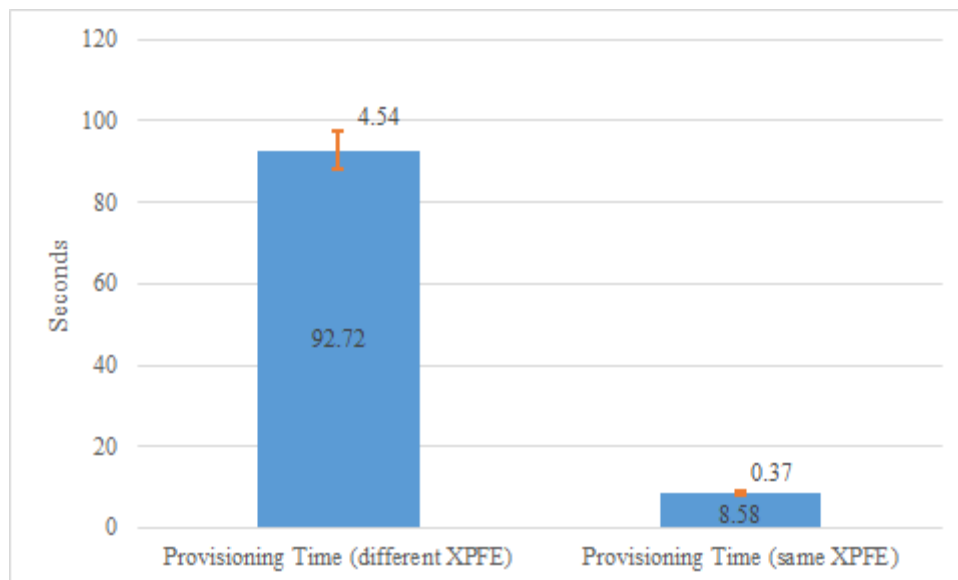


Figure 59. Average and standard deviation of metrics for KPIs

Figure 60 and Figure 61 show an average and standard deviation of each one of the components of the metrics so we can clearly conclude that QP_{dt} and QP_{at} are the most important ones. QP_{dt} would be zero in sub-experiment 2.

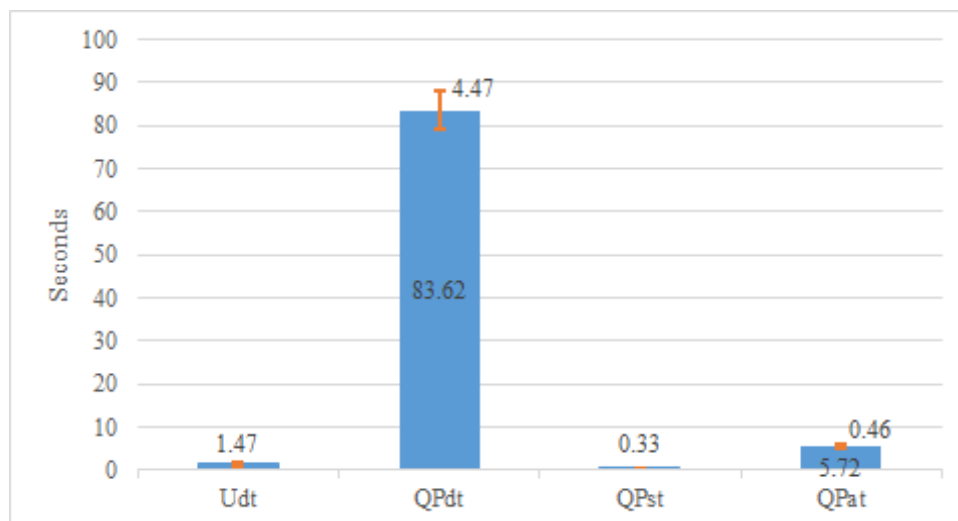


Figure 60. Average and standard deviation of each time involved in the metrics (zoom in of Figure 61)

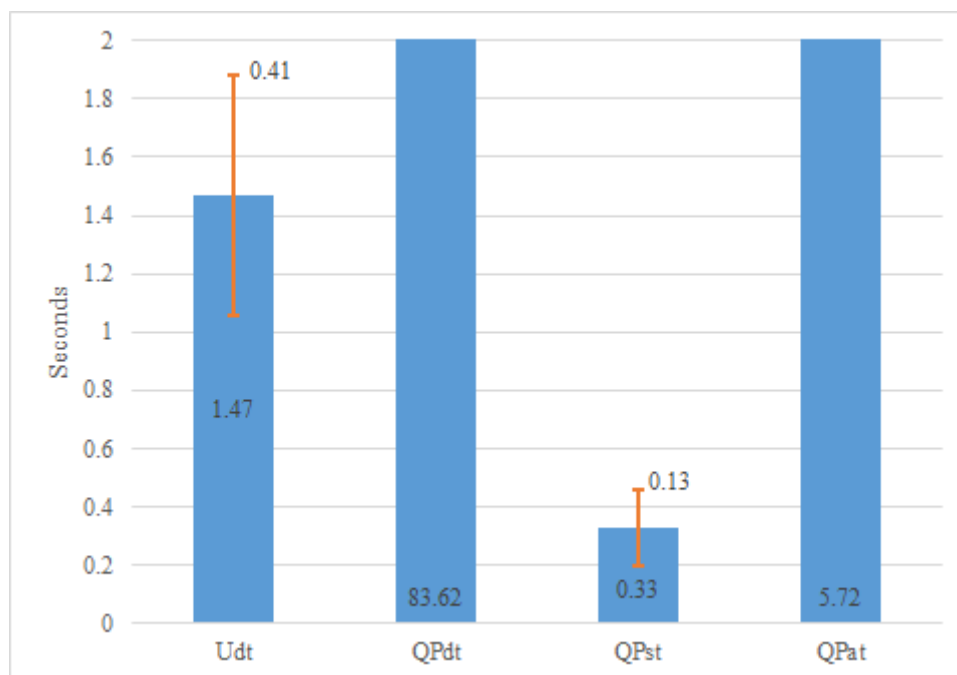


Figure 61. Average and standard deviation of the longest times involved in the metrics (zoom out of Figure 60)

Analyzing each of them:

- QP_{at} depends on the cloud image used as VNF Component for the TVBAQP and on the orchestration software installed by the NFVO in the VNF. A faster-booting image could reduce even more this time. When the users' density increases in the network this time tends to zero since more TVBAQPs have been deployed to monitor other users.
- QP_{at} depends on the buffering time of the TVBAQP and on the time to analyze each frame of the buffered video. Both times depend in turn on the number of frames to analyze. The more frames the more accurate the analysis but the more time to perform it. For TVBA, after an initial benchmark, 250 are used to analyze quality of transmitted media as an acceptable trade-off (spending around 6 seconds in the analysis), although it can be selected by the Operator at any time.

3.2.3 Experiment #10: Path reconfiguration when QoS degradation

Path reconfiguration when QoS degradation	
Description	<p>This experiment takes as starting point the already-provisioned scenario of Experiment #9: Multicast TV service provisioning before the second user is provisioned.</p> <p>An anomaly in the network is triggered (decrement of bandwidth) so that the video reception for the user degrades. The TVBAQP is scheduled to perform analysis regularly and is able to detect this issue with the quality tests reporting to the TVBA Control Panel which starts the healing process for that user. The</p>

	<p>TVBA requests a new path to the RMA and request again the TVBAQP to check the quality of the new service.</p> <p>If still there are problems, the TVBA repeats the process. If no more paths are available, an error is displayed to the operator and the service is stopped.</p>
<p>State of the art</p>	<p>A previous experiment has demonstrated the improvement over the SOTA that is offered by the dynamic deployment of non-referenced QoE and QoS tests near the users.</p> <p>Current IPTV systems are also unable to self-heal themselves when the issue goes further than just the user’s set-top box, at network level, and they need an operator to do it or even a technician to fix issues locally at the nodes.</p>
<p>Improvement from State of the art</p>	<p>5G-Crosshaul services, through path restoration capabilities, allow self-healing capabilities and the TVBA is able to reconfigure the broadcasting-tree to ensure the expected QoS/QoE with minimum impact to the service.</p>
<p>5G-Crosshaul Use Cases</p>	<p>Media distribution.</p>
<p>System under Test</p>	

Project Objectives / 5GPPP KPIs addressed	Obj. 2. Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours. Obj. 8. Self-healing mechanisms for unexpected 5G-Crosshaul link failures through alternative path routing in mesh topologies.
Measured KPIs	Self-healing time (seconds) — Time required to self-heal the system
Measurement tools	Timestamps coded for every relevant process. TVBA logs TVBAQP itself: QoE and QoS information obtained to take decisions. Laptop with VLC as user playing the streamed content.
Measurement procedure	Timestamps will be placed on every process (e.g., VMs deployment, TVBAQP processing time, etc.) obtaining the total time it takes to self-heal a TVBA service. For this experiment the time will be measured from a random time point after provisioning of service, when the anomaly is triggered, until the TVBA receives the new results from the TVBAQP for the new path. Direct observation of the received content on the laptop as user and TVBAQP's quality reports are observed to validate problems in the reception during anomaly and reestablishment of service when the system self-heals.
Constraints	In order to simplify the TVBA Headend implementation and to focus on the experiment objective, the TVBA Headend has been virtualized as a VNF. The TVBAQP is not always running tests since it needs to buffer the received frames to analyse them. It can be configured to run one test after the other or to schedule the tests periodically saving resources at expense of detected issues.
Main results	The full self-healing process takes an average time of 41.67 seconds. Its standard deviation is 7.28 seconds. Where: <ul style="list-style-type: none"> - Reaction time: $15.88 \text{ s} \pm 9.10 \text{ s}$ - Quality Probe's starting time: $0.33 \text{ s} \pm 0.13 \text{ s}$ - Quality Probe's analysing time: $5.72 \text{ s} \pm 0.46 \text{ s}$ - TVBA's decision time: $0.16 \text{ s} \pm 0 \text{ s}$ - Solution time: $1.72 \text{ s} \pm 0.39 \text{ s}$ TVBA is able to self-heal the TV service in a matter of seconds providing non-referenced QoE and QoS monitoring services OTT improving the SOTA as expected; complementing and preparing the ground for vSTB.

Discussion of results

Self-healing time is defined as: $R_t + QP_{st} + QP_{at} + TVBA_{at} + S_t + QP_{st} + QP_{at}$. Where:

- R_t is the *Reaction time*, that is, the remaining time from the moment the anomaly happens to the next scheduled starting time of the Quality Probe. The time between starting events ($Sleep_t$, sleeping time) is defined in the scheduler by the Operator (for testing purposes 2 minutes are configured). The R_t will be always $QP_{at} < R_t < Sleep_t + QP_{at}$. On the other hand, the more the frequency (the less the sleeping time) the more resources are going to be used by the XPU hosting the TVBAQP. A trade-off could be achieved applying this constrain to the algorithms of Energy-Saving Applications.
- QP_{st} is the *Quality Probe's starting time*, that is, the time a TVBAQP needs to be started once it has been instantiated.
- QP_{at} is the *Quality Probe's analysing time*, that is, the time a TVBAQP needs to analyse the media transmission looking for QoS and QoE degradation if any.
- $TVBA_{dt}$ is the *TVBA's decision time*, that is, the time the TVBA needs to take a decision about how to solve the quality issue including all quality information collection.
- S_t is the *Solution time*, that is, the time to apply the solution that for the case of the experiment is the time to provision a new path with constrains, avoiding the links causing the quality degradation.

Figure 62 shows the average and standard deviation of the metric:

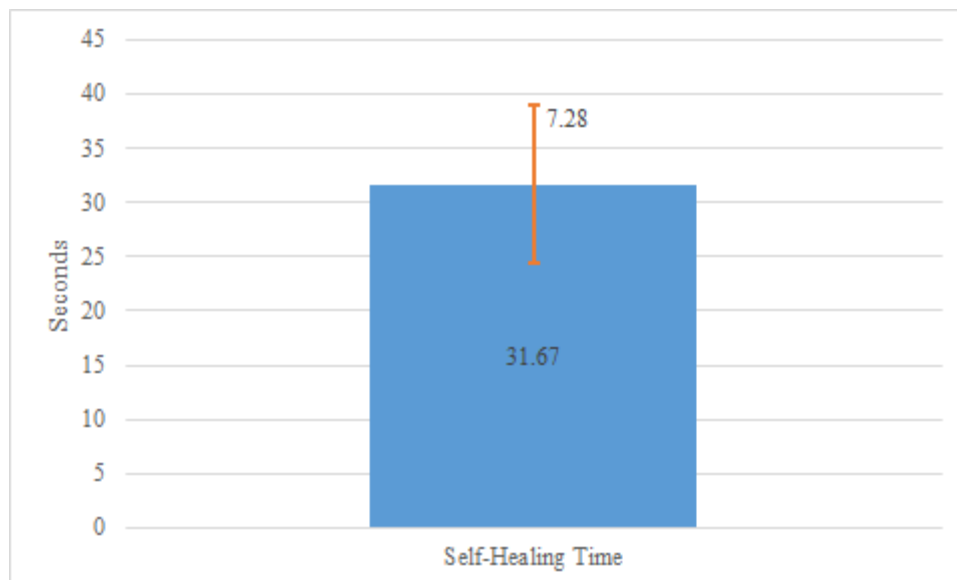


Figure 62. Average and standard deviation of metric for KPI

Figure 63 and Figure 64 show an average and standard deviation of each one of the components of the metrics so we can clearly conclude that QP_{at} and R_t are the most important ones.

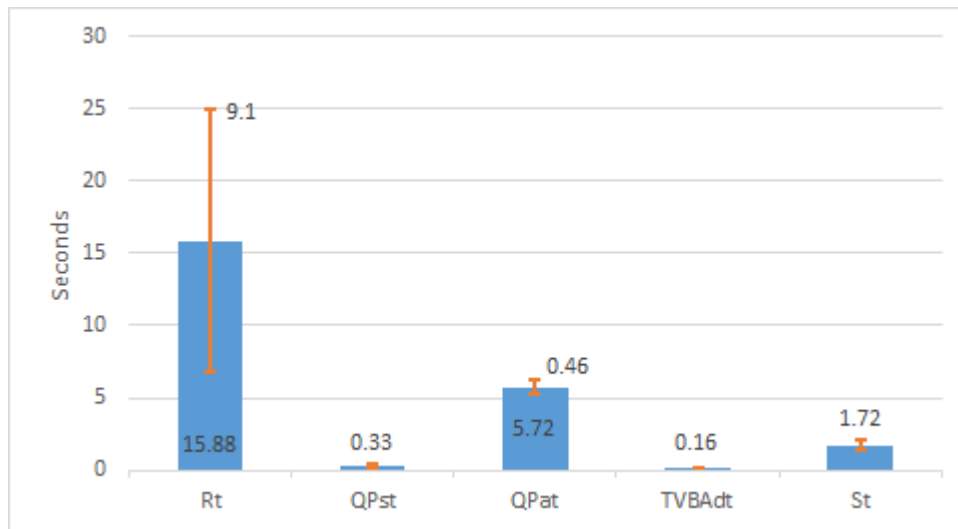


Figure 63. Average and standard deviation of each time involved in the metric (zoom in of Figure 64)

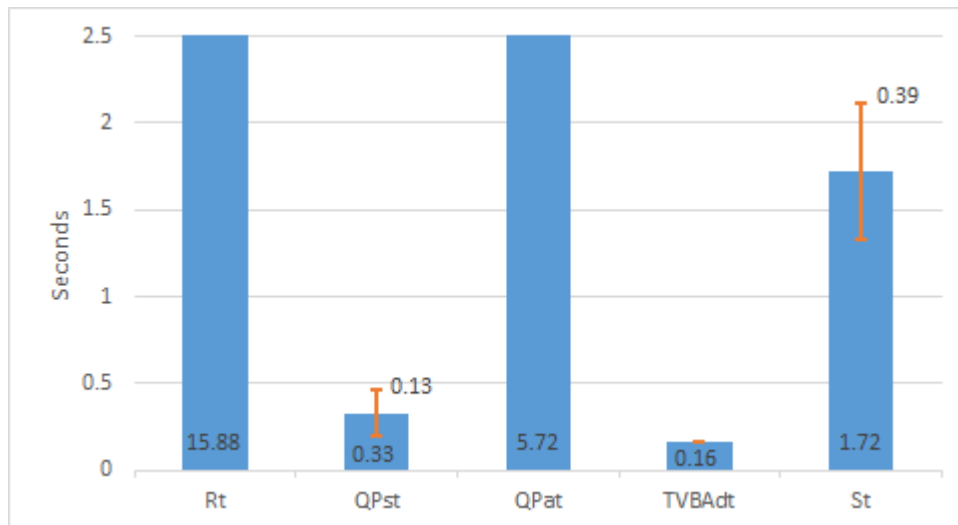


Figure 64. Average and standard deviation of the longest times involved in the metric (zoom out of Figure 63)

Analyzing each of them:

- QP_{at} depends on the buffering time of the TVBAQP and on the time to analyze each frame of the buffered video. It has been analyzed in the previous experiment
- R_t depends on the TVBAQP starting frequency established in the scheduler. Between TVBAQP starting events the traffic is not analyzed and therefore the triggered anomaly cannot be detected until the TVBAQP starts again. Performing tests one after each other can be configured in order to reduce this time to the minimum.

3.2.4 Experiment #11: Distribution of live content through the vCDN infrastructure on the 5G-Crosshaul infrastructure

Distribution of live content through the vCDN infrastructure on the 5G-Crosshaul infrastructure	
Description	<p>This experiment is based on the previous Experiment #8: Virtual CDN service on the 5G-Crosshaul infrastructure, Experiment #9: Multicast TV service provisioning and Experiment #10: Path reconfiguration when QoS degradation. A vCDN infrastructure managed by the CDNMA and a TV Service provided by the TVBA are connected to the 5G-Crosshaul infrastructure to work together.</p> <p>The objective is to provide an integrated platform for the media content distribution service, managing both, VoD and live content in a seamless and dynamic way. The live content transmitted by the TV Headend is provisioned on a multicast way by the TVBA through the 5G-Crosshaul network towards the replica servers which are set up to capture such a traffic. Then the replica servers offer this content to the users located in their respective zones. In this way the users receive the content from the replica server assigned to their zone instead of receiving it directly from the TV Headend saving resources in the backbone while offering added services to the users by means of segregation.</p> <p>Firstly, the replica servers provided by the vCDN join the TVBA network in a similar way as a user does using the MMA. Then the TVBA provides a service to the servers making use of the RMA and monitoring the connection with the TVBAQPs.</p> <p>The content distributed from the TV Headend is received by the server configured in the CDN nodes and streamed to the connected users. The CDNMA manages and monitors the status of the replica servers.</p> <p>In case any QoE or QoS degradation is detected at the CDN nodes, the TVBA self-heals the transmission avoiding interruption of services. While if there is interruption of service between the CDN nodes and the users, the CDNMA scales out and instantiates a new replica server, requesting also a new multicast service for it to the TVBA.</p>
State of the art	<p>Nowadays multimedia content is delivered via broadcast networks, the Internet, IPTV, and mobile. Delivery methods include broadcast, unicast, multicast and peer-to-peer, but the experience is seldom homogeneous and seamless for the customers. That is why most in the content providing industry point out the need of interoperability and cross-platform solutions for media distribution that really satisfy content providers and users' needs (QoS and QoE)[24].</p> <p>Large events as football matches require an important amount of resources and processing work “on-the-fly” so the combination between TV multicast services and CDNs offers useful services to the providers.</p>
Improvement from State of the art	<p>The network flexibility provided by NFV and SDN platforms is used by the vCDN and TVBA infrastructure to offer live content making better use of the network capabilities. The deployment of the replica servers and the TVBA Quality Probes as VNFs instead of hardware appliances allows facilitating and</p>

	<p>reducing the configuration, management and scaling processes, and provides a homogeneous experience aiming to reduce the global resource consumption in the network QoS and QoE is maintained for the end users.</p>
<p>5G-Crosshaul Use Cases</p>	<p>Media distribution.</p>
<p>System under Test</p>	<p>Legend:</p> <ul style="list-style-type: none"> XCI - Crosshaul Control Infrastructure XFE- Crosshaul Forwarding Element XPU- Crosshaul Processing Unit — CDN flow (Best content server) — TV Broadcasting flow (broadcast tree)
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 2: Specify the XCI’s northbound (NBI) and southbound (SBI) interfaces</p> <ul style="list-style-type: none"> • Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours (Media distribution). <p>Obj. 8: 5G-Crosshaul key concept validation and proof of concept</p> <ul style="list-style-type: none"> • Orchestration of 5G-Crosshaul resources based on traffic load variations • Self-healing mechanisms for unexpected 5G-Crosshaul link failures through alternative path routing in mesh topologies.
<p>Measured KPIs</p>	<p>Provisioning of services (secs) — The time it takes to deploy a TVBA and CDN service until the user (end-to-end service)</p> <p>Self-healing — Time for path-restoration given the capacity to relocate paths and resources to deal with network issues related to QoE/QoS.</p>
<p>Measurement tools</p>	<p>Wireshark is used to analyse messages associated with the media content.</p> <p>VNF probes, monitoring system.</p>

	<p>CDNMA & TVBA control panels.</p> <p>QoE and QoS information obtained from TVBAQP to take decisions.</p> <p>Laptop with VLC as user playing the streamed content.</p>
Measurement procedure	<ol style="list-style-type: none"> 1. Checking the vCDN infrastructure deployment. CDN nodes are instantiated in the right places and working based on the configuration defined in the CDNMA. 2. The TV Headend is streaming and the TVBA interacts with it. 3. The TVBA requests to the RMA the provisioning of a path between the TV Headend and the CDN nodes. 4. The RMA establishes the flows through the SDN Controller. 5. The TVBA deploys a TVBAQP (a VNF) as close as possible to the CDN node and starts the monitoring of the transmitted media to self-heal the system when needed. 6. The CDNMA checks the specific monitoring information from the different CDN nodes (VNFs) deployed in the network. 7. The assigned CDN node serves the streaming to the users' video players. 8. If there is a quality degradation at the CDN nodes, the TVBA self-heals the system. If it is between the CDN nodes and the users, the CDNMA instantiates a new server for those users.
Constraints	<p>In order to simplify the TVBA Headend implementation and to focus on the experiment objective, the TVBA Headend has been virtualized as a VNF.</p> <p>The TVBAQP is not always running tests since it needs to buffer the received frames to analyse them. It can be configured to run one test after the other or to schedule the tests periodically saving resources at expense of detected issues</p>
Main results	<p>Average provisioning time of the service: approximately 249 s.</p> <p>Where:</p> <ul style="list-style-type: none"> - vCDN deployment time with 2 replica servers: 157 s - TVBA provisioning time: 92 s <p>Average self-healing time (considering 2 minutes as frequency to monitor the quality of the service):</p> <ul style="list-style-type: none"> • Quality degradation between TV Headend and CDN node (there is no need to deploy a new CDN node to provide the live streaming): 41 s • Service interruption between CDN node and users (a new replica server needs to be instantiated and a new TVBA service has to be provisioned): approximately 137 s <p>Where:</p> <ul style="list-style-type: none"> ○ TVBA self-healing process: 41 s ○ TVBA new provisioning: 92s ○ New CDN node deployment: 45 s

Discussion of results

The TVBA, in order to provide the distribution of live content through a vCDN, needs the vCDN infrastructure to be instantiated before the TV Multicast service is provisioned. This requirement implies that the total time for provisioning a live content distribution exploiting the vCDN capabilities is the sum of the CDN deployment and the TV service provisioning times. These values are obtained from the configurations and results in Experiment #8: Virtual CDN service on the 5G-Crosshaul infrastructure and Experiment #9: Multicast TV service provisioning.

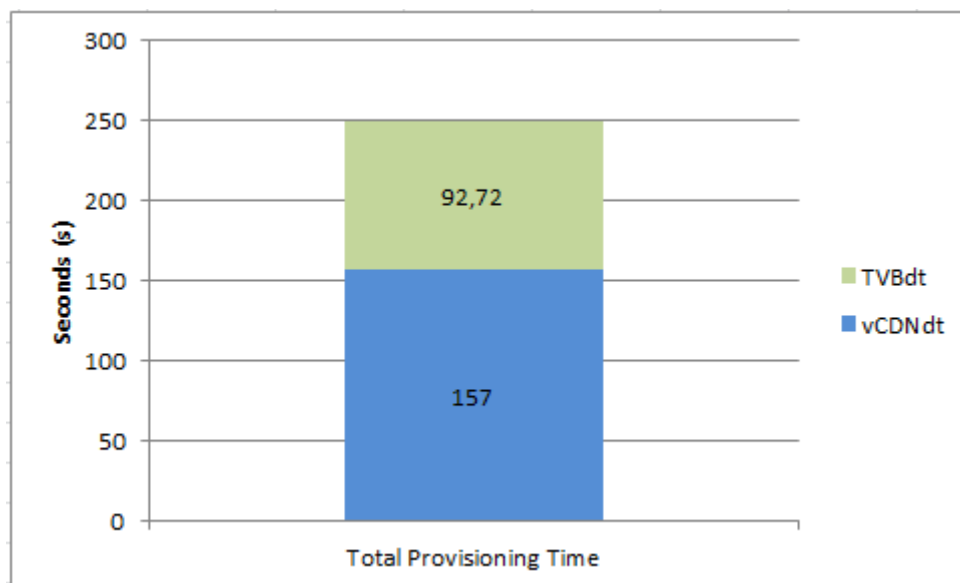


Figure 65. Total provisioning time for live content distribution through a vCDN with 2 replica servers

Figure 65 shows the time required to deploy the two infrastructures involved in the service:

- vCDNdt is the deployment time of a CDN with 2 replica servers located in two different XPU as it was described in the System Under Test section.
- TVBdt is the provisioning time of the TV Broadcasting service.

Self-healing time for live content distribution through a vCDN presents two scenarios which are clearly represented in Figure 66:

- The TVBA self-heals the connection between the TV Headend and the CDN nodes (SHt) so there is no need to instantiate a new replica server somewhere else.
- The CDNMA detects an interruption of service between the CDN node and the users. The CDNMA has to scale out the vCDN, adding a new replica server because the monitoring service has detected issues in the assigned CDN node (RSdt). In this situation, TVBA waits for the new server to be completely available and then provisions the new service (TVBdt).

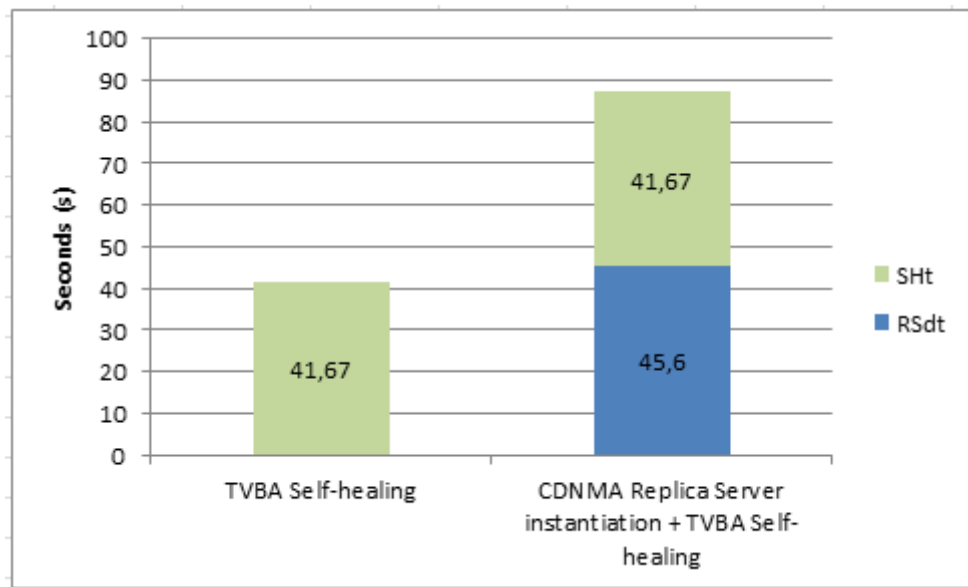


Figure 66. Self-healing time for live content distribution through a vCDN

These results, as well as it was explained for the provisioning time measures, are consequence of the measures obtained in Experiment #8: Virtual CDN service on the 5G-Crosshaul infrastructure and Experiment #10: Path reconfiguration when QoS degradation.

4 Experimental evaluation results for hierarchical multi-domain resource management of the crosshaul (Demo 3)

Demo 3 deals with network orchestration; that is, the ability to provision services in a transport network characterized by having multiple technologies. The demo targets the experimental validation and applicability of the Crosshaul XCI, which, as far as control of network resources is concerned, deploys a hierarchy of controllers. More specifically, we deploy a hierarchical XCI for which a *parent-child* relationship is established between contiguous layers. At the top of the hierarchy, the parent ABNO controller coordinates the operation of multiple technology-specific/multiple domain child SDN Controllers. ABNO stands for Application-based Network Operations and is structured around the functionalities defined by IETF and explained in D5.1 [25].

Demo 3 showcases the use of a REST/JSON-based protocol framework (the Control Orchestration Protocol, or COP), as developed within WP3 between the ABNO controller and the underlying child SDN controllers. COP is also used between the Resource Management Application and the parent ABNO on the northbound of the XCI, hence allowing recursive and scalable deployments. In particular, Demo 3 illustrates the associated procedures (retrieving the topology, abstracting different domains from child controllers and provisioning connectivity services across the domains), focusing on the wireless/wired combination that reflects realistic integrated multi-domain transport networks. We consider a domain as a set of data plane elements under the control of a technology specific control plane, assuming a deployment model of a single SDN controller per domain.

Parts of results presented in this section have been presented in a EUCNC17 paper [26] and have been demonstrated at EUCNC17 [39] and IEEE NFV-SDN conferences [40]. In summary, the results confirm the benefits of having a hierarchy of controllers to perform the orchestration of network resources in a transport network composed by multiple domains using different communication technologies at the data plane level. The flexibility of the proposed solution allows that some technology-specific local decisions or reactions in front of network events such as a link failure can be taken by the closest controller in the hierarchy, hence saving processing and propagation time. In our case, and depending on the domain, there may be differences in path setup/restoration values observed at child vs. parent controller ranging from around 500ms to one order of magnitude (up to units of seconds). For instance, the path restoration time values in the wireless domain (100s of ms) and the path setup time at the child controllers (around 2.5s.) is smaller than that observed by the parent ABNO (3.349s) and RMA (3.971s).

In addition to this, the assessed hierarchy of controllers allows the introduction of a complete E2E LTE mobile network service involving multiple bidirectional flow setups between the multiple mobile network entities (i.e., RRH, BBU, eNB, MME, SPGW) across geographically distributed control plane entities and transport network domains. Values obtained are in the order of seconds (on average 10.5 seconds), hence contributing to the 5G target of lowering the service deployment time (in this case, multi-domain path setup, from months to minutes or even, seconds). In fact, current (manual) setup times in production networks are in the order of hours, or even days (Appendix 1).

The flexibility provided by the hierarchical XCI in the processing of a service recovery triggered by a link down event was also evaluated. This recovery process can be done, either locally at the child SDN controller level (the low level of the control hierarchy) or centralized at the RMA level (the upper level of the control hierarchy). The former allows much lower recovery times (0.299s on average) at the cost of potentially suboptimal resulting paths, unlike the latter, which takes 6.652s on average for re-establishing an optimal multi-domain path.

Finally, from the data plane perspective, the assessed transport setup offers around 153 Mbps E2E, which is mainly limited by a VPN connection between remote sites, and where emulated users using the previously mentioned LTE mobile network service and placed at both edges of the setup achieve up to 140 Mbps.

In summary, the work carried out in demo 3 contributes in fulfilling the project and 5GPP KPIs and objectives of enabling the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours (seconds in this case). It offers a scalable management framework through its hierarchical deployment, and enables the deployment of novel applications and the orchestration of 5G-Crosshaul resources reducing the network management Operational Expenditure (OPEX) due to the flexibility and programmability offered by the XCI in a multi-domain, multi-technology setup, which will be the norm in 5G network deployments.

4.1 Demo 3 setup

At the top of Figure 67, the RMA is in charge of computing optimal routes between endpoints in the multi-domain data plane relying on a graph-based abstracted view of the underlying topology, provided by the XCI. The XCI SDN component is hierarchical, with a parent ABNO controller orchestrating one SDN controller per domain, the so-called child controllers.

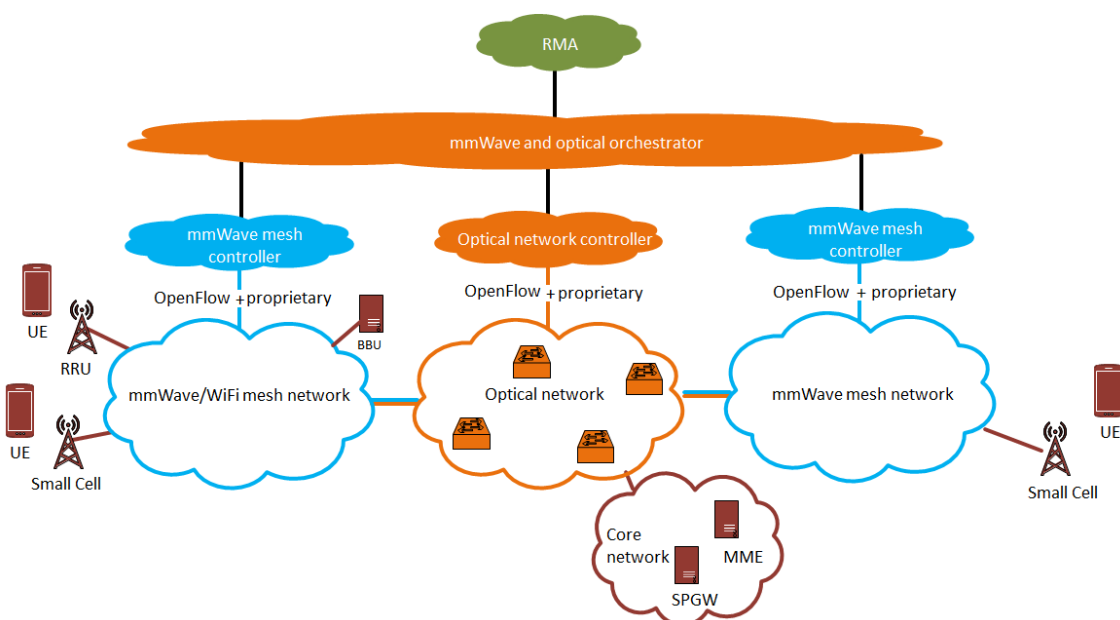


Figure 67. Demo 3 architecture

Each child controller uses a different protocol for inventory, management and monitoring of the respective hardware (e.g. a REST-based interface in one of the mmWave domains and a proprietary protocol in the other one). In order to expose this multi-domain information in a homogeneous manner to the RMA, the parent controller interacts with the child controllers via Control Orchestration Protocol (COP), and so does the RMA with the parent controller. The child controllers can make local decisions at a shorter time-scale than the RMA and the parent controller. For instance, the mmWave mesh controller can configure paths locally within the mmWave mesh network in case of link failures and so, ensure connectivity while RMA calculates and triggers the request for a new path provisioning.

Demo 3 testbed is depicted below (see Figure 68). As detailed next, it involves several components that are explained below in sections 4.1.1 and 4.1.2.

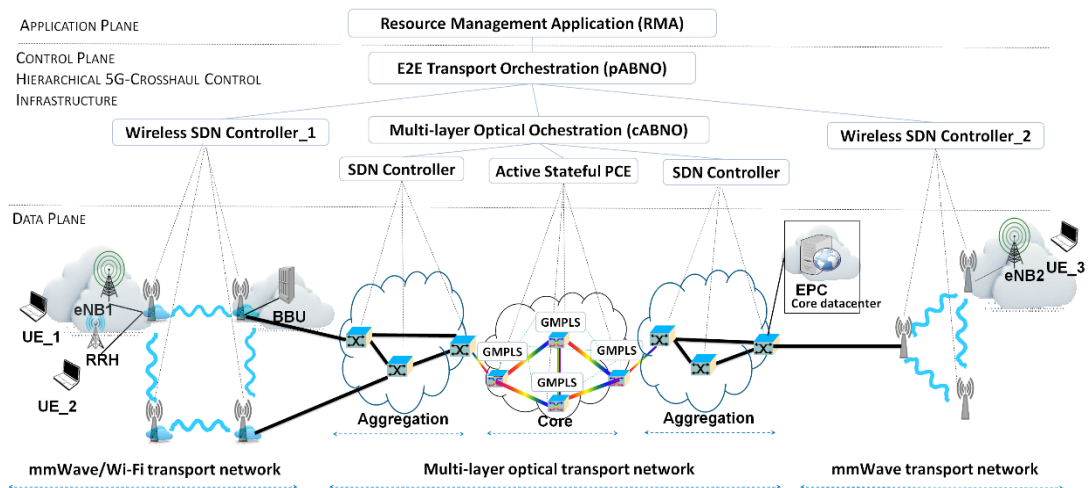


Figure 68. Demo 3 testbed overview

These components are distributed geographically among different partners involved in 5G-Crosshaul project. In particular, the parent ABNO and the child controllers (cABNO and Wireless SDN Controller_1) associated to the mmWave/Wi-Fi transport network and the multi-layer optical transport network are located in a site in Barcelona (Spain), as depicted in Figure 69. The other mmWave transport network is placed in London (UK) and the RMA is placed in Heidelberg (Germany). The elements of the application/control plane count with a Graphical User Interface (GUI), which show useful information for Demo3 purposes. Figure 70 shows some of them. The GUI of the RMA (Figure 70a) shows the abstracted view of the transport network and the currently installed paths in the system. The GUI of parent ABNO (Figure 70b) shows the abstracted representation of the underlying transport network and the set of installed service calls according to COP models. The GUI of Wireless SDN Controller_1 of the mmWave/Wi-Fi transport network (Figure 70c), shows the configuration of the wireless interfaces, the wireless transport topology and a dynamic representation of the relative load of the wireless transport nodes and its associated links. To interconnect the different sites, several different virtual private network (VPN) tunnels for both data and control planes have been established. Thanks to the mentioned COP protocol, the RMA is able to establish end-to-end connections between different sites to make data flow through the tunnels between any pair of endpoints present in the deployment of Figure 68.



Figure 69. Demo 3 testbed facilities in Barcelona site

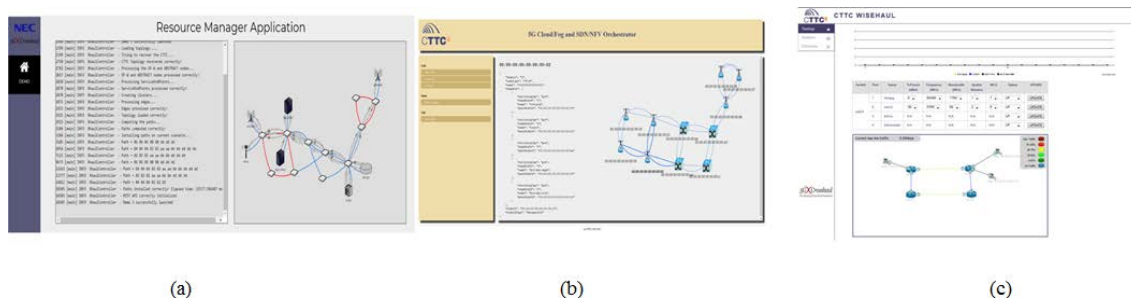


Figure 70. Graphical User Interface of a) RMA, b) pABNO, and c) Wireless SDN Controller_1, application and control plane elements of the Demo3 testbed

From the point of view of the measurements, the tools deployed in the testbed allow obtaining KPIs such as setup delays or associated control plane overhead and others, as described below.

4.1.1 Building blocks from other work packages

Demo 3 encompasses several building blocks from work-packages WP2, WP3 and WP4, as shown in the tables below.

Data Plane building blocks

WP2	
Component / Entity	Description
mmWave/Wi-Fi nodes	A set of OpenFlow-based packet-switched nodes communicating by means of wireless interfaces using IEEE 802.11ac (Wi-Fi) or 802.11ad (mmWave) protocols. These nodes include an instance of open source software switch, such as xDPd and/or OVS for packet forwarding, and the developed RestConf agent to configure and monitor the wireless interfaces.
Wired Packet switching nodes	A set of OpenFlow-based packet-switched nodes. Common-Off-The-Shelf high-end Intel PCs are used to implement the nodes, along with software such as DPDK and/or OVS for the kernel/user-based forwarding function.
ROADMs	4 Reconfigurable Optical Add Drop Multiplexers, arranged in a diamond-shaped topology, and which are able to provision data services (e.g., 10 Gbps lightpaths) by means of a dedicated Generalized Multi-Protocol Label Switching (MPLS) / Path Computation Element (PCE) control plane.

Control and Applications Plane building blocks

WP3	
Component / Entity	Description
ABNO (see D3.2, section 3.3)	The ABNO controller covers the requirements of multi-domain network orchestration; that is, the ability to coordinate service

	provisioning (e.g., flow and connection provisioning) across multiple (technology) domains.
Topology Manager	<p>The Topology Manager (TM) maintains Traffic Engineering Database's (TED) where all the relevant topology information is stored. The TM composes a complete multi-domain topology and a separated topology, built by filtering the whole topology based on the TE information of the links, for each control domain technology. In the TM, a dedicated plugin is implemented for each domain SDN controller NBI API (usually REST based on HTTP/JSON) for topology discovery purposes.</p> <p>The TM is able to aggregate the topology of the underlying domains.</p>
Path Computation Element (PCE)	It is responsible for path computation in a (abstracted) multi-domain topology.
Virtual Network Topology Manager (VNTM)	It is the responsible for the multi-layer management. The VNTM is in charge of satisfying upper layer's connectivity demands by spawning lower layer connections (e.g., optical DWDM channels) represented into the TED as Virtual Links.
Provisioning Manager (PM)	The PM is the module which translates the connectivity requests. The PM implements a provisioning plugin for each different network controller connected to it. In the proposed architecture, it implements the SDN controller provisioning REST API and the PCEP with Stateful and PCE-initiated LSP Setup extensions, for the communication with the AS-PCE.
SDN domains	The selected scenario includes multiple domains, in where a single, centralized SDN controller is responsible for topology, connection and service provisioning within its domain.
SDN Controller	The SDN Controller is domain and partner specific (e.g., Ryu for wireless and wired packet-switched domains) and it is a functional requirement of Demo 3 to have a common NBI for network orchestration (<i>the Control Orchestration Protocol, or COP</i>).
WSON	GMPLS/PCE controlled Wavelength Switched Optical Network.
AS-PCE	An Active-Stateful PCE is used as the entry point to the GMPLS controlled WSON.
GMPLS Control plane	The actual provisioning of lightpaths within the domain is ultimately carried out by the mature, proven GMPLS architecture.
WP4	
Component / Entity	Description
RMA (see D4.1, section 3.3)	The Resource Management Application (RMA) addresses the centralized and automated management of Crosshaul resources, in order to promptly provision and recover transport services with an

	adequate quality while ensuring that Crosshaul resources are efficiently utilized. This involves the RMA application directly consuming the services of the ABNO that performs network orchestration.
--	---

4.1.2 Additional building blocks

In addition to the previous elements, the following building blocks, which are not within the scope of the technical development defined in 5G-Crosshaul activities, are required to support Demo 3.

Component/Entity	
Access Nodes	Nodes connected to the edge transport nodes (mmWave/Wi-Fi nodes) generating the corresponding backhaul/fronthaul traffic to the 5G-Crosshaul network. Servers running an instance of eNodeB (for backhaul traffic) or RRU-BBU (for different fronthaul traffic splits such as PDPC-RLC, MAC-PHY) prototypes (see D5.1, section 9.3) are used for this purpose. In addition to this, if possible, backhaul traffic could be also generated with ns-3 LENA in emulation node [38].
Network Core Node	Node running a software instance of a complete 3GPP standard compliant Evolved Packet Core providing mobile network, such as OpenEPC software. If possible, network core can be also emulated with ns-3 LENA.
Mobile User	Entity connected to the access node, either the eNodeB or the RRU entity, emulating an end user.
Traffic Generators	Open source applications such as Iperf ⁷ or Netperf ⁸ can run in the mobile user to generate different traffic profiles (transport protocol, rate). Other kind of software, such as VideoLAN (VLC), can be used for demo purposes to generate video streaming traffic between endpoints of the considered transport network.
Measuring/Debugging tools	Iperf, Netperf and ping applications provide measuring capabilities to get network performance metrics, such as throughput, latency, jitter, packet loss ratio. Wireshark ⁹ software is used for debugging purposes to check the correct exchange/workflow of control messages between different control/application layer entities, such as the communication between the RMA and the ABNO or the communication between the ABNO and its associated SDN domain controllers.

⁷ iperf.fr

⁸ www.netperf.org

⁹ www.wireshark.org

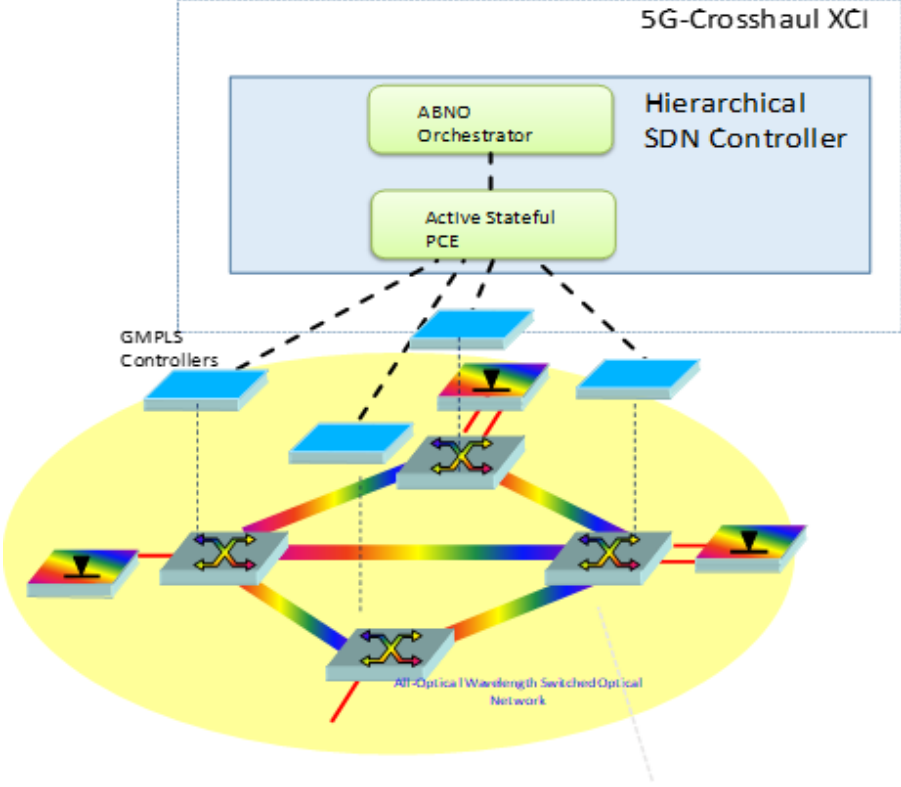
4.2 Experiments on single domain path provisioning and restoration

The following experiments present the initial characterization of each individual transport domain, mainly from the control plane perspective, to perform partial validations of the deployed system with the ultimate objective of better understand the contribution (in terms of time) of each domain when setting up end-to-end connections.

4.2.1 Experiment #12: Assessment of the SDN-based control of the Optical Transport Network (optical domain)

The table below shows the experiment defined to evaluate the capability of the Multi-Layer Optical Transport Network to support the establishment of packet-switched connections in the multi-domain network configuration that the testbed setup is enabling.

Assessment of the SDN-based control of the Optical Transport Network	
Description	<p>The experiment demonstrates and evaluates the performance of the provisioning service within the 5G-Crosshaul XCI. In particular, it evaluates the resource-efficient connection request in the optical domain, considering requests that involve the <i>ABNO-based network orchestration</i> and the <i>Active Stateful PCE</i>.</p> <p>Note that this experiment is defined to support more integrated experiments that are detailed in the following section 4.3. Since the optical network remains a transport network for the aggregation and grooming of packet-switched traffic, its initial characterization is required to have a basic framework of performance.</p>
State of the art	<p>The state of art on the control and management of Optical Transport networks is characterized by the following, see for example [4]:</p> <ul style="list-style-type: none"> • Per-domain provisioning using distributed control plane or a centralized management system, with very little integration into orchestration systems. • Very limited deployments of SDN control for transport networks. Protocols are proprietary or based on legacy systems such as SNMP. • Protocol extensions to support transport networks in SDN-based configuration still undefined or not standard (there is ongoing work and the ONF/OIF and IETF to work on a set of interfaces, collectively known as “transport API”). • Per-vendor or per-technology transport domains each with its isolated control with proprietary extensions, but no end-to-end global management of transport network.
Improvement from State of the art	<p>The following are the main improvements with respect to the state of the art:</p> <ul style="list-style-type: none"> • Possibility of orchestrating multiple heterogeneous domains under the umbrella of a hierarchical system by means of a parent controller based on ABNO. • Use of a dynamic provisioning system based on SDN principles applied to transport networks. • Use of an Active Stateful PCE and a centralized entry point for the automation of the provisioning within a GMPLS network, and putting this PCE under the control of a network orchestration.

	<ul style="list-style-type: none"> • Use of open and standard protocols for service provisioning across multiple heterogeneous technologies and domains.
<p>5G-Crosshaul Use Cases</p>	<p>Transversal to multiple 5G Use cases, related to the automation of network connectivity provisioning by using the 5G-Crosshaul XCI. In particular:</p> <ul style="list-style-type: none"> • Dense urban information society (efficient end-to-end resource management allows serving more traffic with the same resources enabling a cost-effective dense deployment). • Media distribution flows as one demanding example for which efficient resource management is particularly required.
<p>System under Test</p>	
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Design of the Crosshaul Control Infrastructure</p> <p>Obj. 2: Specify the XCI's northbound (NBI) and southbound (SBI) interfaces</p> <p>Obj. 6: Design scalable algorithms for efficient Crosshaul resource orchestration</p> <p>Obj. 7: Design essential Crosshaul-integrated (control/planning) applications</p> <p>Obj. 8: Crosshaul key concept validation and proof of concept</p> <p>Contributes to the following 5GPPP KPIs: Increase the number of connected devices per area, Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours, Scalable management framework, Enable deployment of novel applications reducing the network management Operational Expenditure, Orchestration of 5G-Crosshaul resources, Self-healing mechanisms.</p>

Measured KPIs	<p>Path Setup time defined as the time it takes to provision a service (in this context, the establishment of an optical lightpath). This value can be measured from different reference points (the ingress node responsible for triggering the setup of the lightpath using the GMPLS protocol framework, the AS-PCE, the ABNO system) in order to evaluate the effect of each system in the overall architecture.</p> <p>Blocking Probability in optical networks specific configurations and scenarios. This is defined as the ratio of failed requests over total number of requests. In the scenarios considered, the blocking probability is mostly due to the lack of resources and, specially, due to constraints such as the wavelength continuity constraint.</p>
Measurement tools	<p>Network analysers, packet capturing software.</p> <p>Adapted Request generators: these are software tools that automate the creation of a set or connection requests based on statistical and probability models.</p> <p>Timestamps of Wireshark/Pcap traces at different points of the network of NTP synchronized machines.</p>
Measurement procedure	<ol style="list-style-type: none"> 1. Configure packet capture software to get traces about the exchange of messages and involved protocol. 2. Configure control plane elements to produce traces with timestamps that allow the post-processing of the results for obtaining meaningful values. 3. Trigger the request of a service using the ABNO Northbound Interface and observe the behaviour of the system at difference points. 4. Post-process the traces to obtain the desired measurements.
Constraints	<p>Although not being a real deployment, the scenario is realistic because:</p> <ul style="list-style-type: none"> • It includes control and data plane elements, including a 4 node Optical Transport Network (4 ROADMs/ OXC) • Uses an implemented and deployed control plane based on GMPLS protocol suite • Augments the GMPLS control plane with an Active Stateful PCE, which is an entity defined in the IETF with strong support from industry • It relies on the ABNO (also based on an IETF published architecture [27]) for network orchestration.
Main results	<p>Path Setup time in the optical domain (lightpath setup only):</p> <ul style="list-style-type: none"> • Of the order of tens of milliseconds when measured at the lowest layer of the hierarchical control plane for the optical domain; • Of the order of hundreds of milliseconds when instantiated by the ABNO controller supervising only one domain. <p>Blocking Probability in the optical domain depends on offered traffic and specific scenario. For a reference benchmark consider less than 20 Erlangs to obtain a Blocking Probability (BP) < 0.03%.</p>

Discussion of results

After the initial set of tests, the following sections detail the measurements that have been performed and the key performance indicators that have been obtained. Experiments of 10000 requests have been carried out, where connections arrive following a Poisson point process with negative exponential holding times, fixing inter-arrival average time and increasing holding times depending on the offered traffic in Erlangs. (e.g. one connection arriving on average every 3 seconds, and lasting on average 60 seconds for an offered traffic of 20 Erlangs). For the evaluation, we consider that requests are being uniformly distributed among the four nodes within the network scenario. Each link is characterized for having 8 bidirectional wavelengths supporting 10Gbps client rates. Requests go from one node or optical crossconnects (OXC) to a destination node or OXC and do not involve client transceivers. This is due to hardware limitations that would constraint the results, since only a limited set of transceivers are available, well below the theoretical maximum supported by the optical network.

4.2.1.1 Path setup time in the optical domain

The first measured KPI is the path setup time, including decomposition of path setup time from multiple functional entities (ABNO, Active Stateful PCE (AS-PCE), GMPLS ingress controller). As it can be seen in the diagram of the system under test, its control infrastructure presents a hierarchical structure consisting of a GMPLS control plane for the optical network together with an AS-PCE that enables centralized path computation, connection establishment delegation. The AS-PCE is later part of a wider network orchestration system integrating multiple domains, the ABNO orchestrator. It is worth noting that path setup is affected by the hardware configuration delay: this delay involves configuring the cross-connect and it is vendor dependent. In our specific testbed, we have performed different measurements with hardware configuration and without hardware configuration. Path setup time has been measured in several iterations, in order to obtain the histogram and/or the CDF of the setup delay introduced by the different components of the system, during the processing of the service requests in the experiment.

4.2.1.1.1 Without optical hardware configuration

In order to focus on the control plane behaviour, we configure the hardware optical nodes in emulation mode, requiring a negligible configuration time.

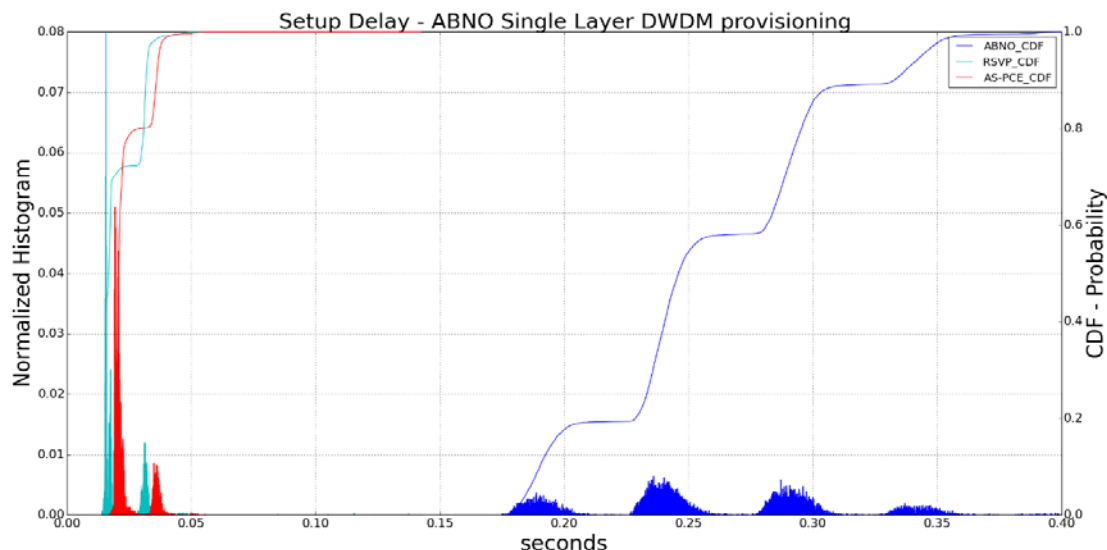


Figure 71. Setup time histogram and CDF as seen at different reference points: from the RSVP-TE connection controllers, from the AS-PCE and from the ABNO network orchestrator.

From the picture above (Figure 71), we get the histogram and CDF of the setup time as follows:

- From the GMPLS control plane (denoted as RSVP in Figure 71): this means the setup delay considering the signalling process, from the ingress node to the egress node, and roughly corresponding to the RTT of the signalling messages with forward Path and reverse Resv messages across the different transit nodes. We see that on average, this shows two peaks, roughly corresponding to whether 2 or 3 optical nodes have been involved in the provisioning. Avg.= 20.72ms.
- From the AS-PCE: the AS-PCE adds a small component to the setup time, associated to the processing of requests from the ABNO and dispatching of requests to the corresponding head-end node. Avg.= 23.80ms.
- Finally, we see the setup time from ABNO orchestrator, which adds an additional time due to the COP protocol and the use of text-based REST interfaces. The multiple peaks in the histogram reflect an implementation artefact through which the ABNO (child and parent) wakes up every 50ms and processes the received requests during that period. Average setup delay seen from ABNO is 259.01ms.

4.2.1.1.2 With optical hardware configuration

In order to take into account the hardware configuration, an additional set of 100 requests has been performed, obtaining the following average values:

- From the ingress node RSVP ~215ms
- From the AS-PCE: ~220 ms
- From the ABNO ~450 ms

Macroscopically, the hardware configuration adds an additional 200 ms to the provisioning time. As a summarizing guideline, the setup delay for the optical domain, seen from the ABNO controller can be probabilistically bound to below 500 milliseconds (sub-second provisioning delay in our considered scenario).

4.2.1.2 Blocking Probability in the optical domain

We consider a system with dynamic arrivals of connection requests, where each connection requires the establishment of an optical lightpath between two client ports (similar to the Public Switched Telephone Network, or PSTN). If a connection cannot be served (for example, due to no more wavelengths being available that can be provisioned to the service), we consider the connection blocked. Blocking probability (for the selected service of lightpath establishment) has been experimentally evaluated as a proportional ratio of successful establishment over offered ones, assuming service request arrival statistics. Likewise, the plot below (Figure 72) has been obtained empirically by performing random setup requests and a Dijkstra-based path computation algorithm ensuring wavelength continuity. We have used several data points from 20 to 50 Erlangs (further experiments are possible, but each experiment is long lasting, with 10000 connection requests each).

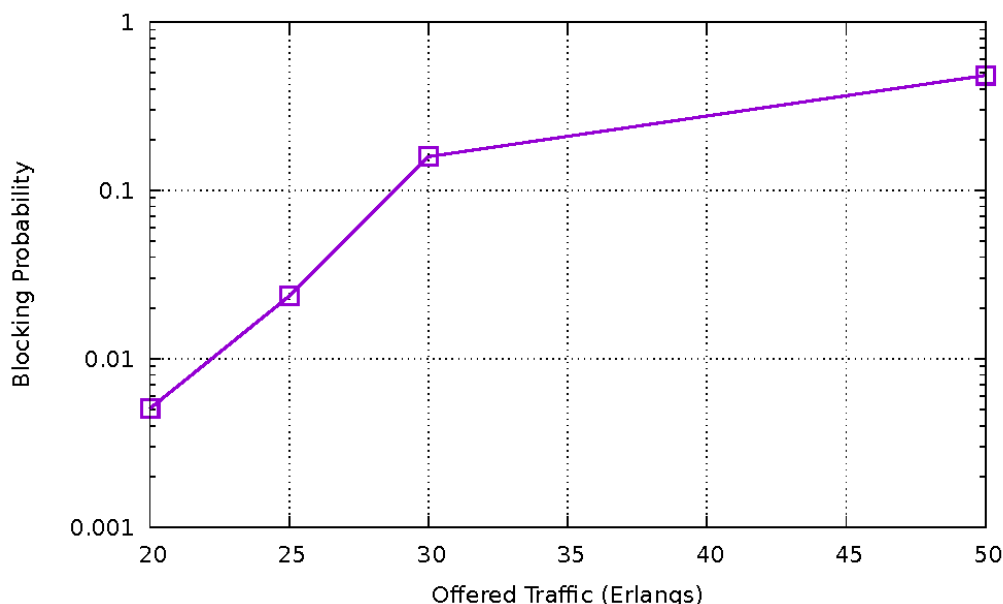


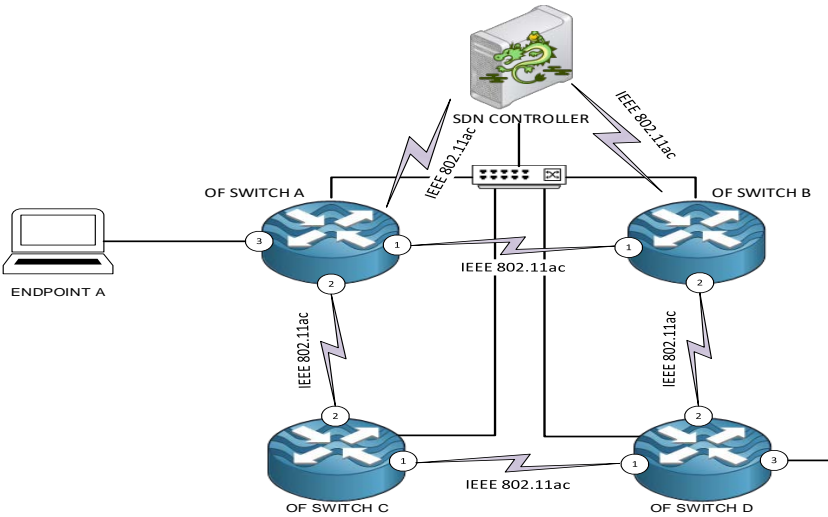
Figure 72. Blocking Probability obtained for the 4-node Optical Transport Network requesting one wavelength from source to destination node, for 3 seconds avg. inter-arrival time and varying holding time. X-axis corresponds to traffic load

The blocking probability (BP) ranges from 0.05% at 20 Erlangs to 48% at 50 Erlangs. This test has been done to stress the system and evaluate the BP trend; in real operation scenarios, it is expected to dimension the network in an operating range in order to keep the BP down to acceptable levels.

4.2.2 Experiment #13: Assessment of the SDN-based control and data plane for the mmWave/Wi-Fi mesh domain

The table below shows the experiment defined to evaluate the capability of the mmWave/Wi-Fi mesh transport network to support the establishment of packet-switched connections in the multi-domain network configuration that demo 3 is enabling.

Assessment of the SDN-based control and data plane for the mmWave/Wi-Fi mesh domain	
Description	<p>The experiment demonstrates and evaluates the performance when provisioning services in a wireless domain built using mmWave and Wi-Fi technologies.</p> <p>The first goal is to evaluate Path Setup Time vs number of hops by analysing the Path Setup Time response for a UDP flow traversing 1, 2, and 3 hops. As backhaul/fronthaul topology, we will use a ring of 4 data plane nodes featuring IEEE 802.11ac standard and the XCI component managing 5G-Crosshaul wireless transport infrastructures.</p> <p>The second goal of this experiment is to evaluate Throughput vs number of hops by analysing the Data Plane Throughput response for a UDP flow traversing 1, 2, and 3 hops. As backhaul/fronthaul topology, we will use a ring of 4 data plane nodes featuring IEEE 802.11ac standard and the wireless SDN controller.</p>
State of the art	SDN technologies have been hitherto constrained to the presence of a wired infrastructure in both the data plane and the control plane

	<p>communication channels, such as the wired data centre or campus networks [28]. The adoption of SDN for a distributed and dense wireless (mmWave or microwave) backhaul/fronthaul domain is currently under discussion and raises some concerns. The decentralization of the control plane yields the deployment reliable and offers low-delay control plane channels at the expense of increasing operator costs. The appropriate trade-off centralization-distribution must be found.</p>
<p>Improvement from State of the art</p>	<p>The specific novelty that goes beyond SOTA in this experiment is the assessment of whether the required service level can be satisfied with low-cost wireless technologies (e.g., Wi-Fi) or not. In particular, we are interested in revealing what common set of functions (or services) that any SDN controller will provide (e.g., establishment of paths between endpoints) can be suitable with a wireless control plane, cheaper in terms of CAPEX compared to usual wired deployments. In this way, not only the data plane will exploit wireless technologies but also the control plane, hence reducing even more the deployment cost of the whole network.</p> <p>Our benchmark is the out-of-band wired control plane scenario in which a specific network only handling control traffic is used to reach a centralized controller. The goal is to analyse to which extent the control and data plane metrics are equivalent no matter the channel used (wired or wireless) for the control plane.</p>
<p>5G-Crosshaul Use Cases</p>	<p>Transversal to multiple 5G Use cases, related to the automation of network connectivity provisioning by using the 5G-Crosshaul XCI. In particular:</p> <ul style="list-style-type: none"> • Dense urban information society (resource management in cost-effective wireless edge transport networks) • Media distribution flows as one demanding example for which efficient resource management is particularly required.
<p>System under Test</p>	 <p>The diagram illustrates the system under test. It features a central SDN Controller connected to four OpenFlow (OF) switches: OF SWITCH A, OF SWITCH B, OF SWITCH C, and OF SWITCH D. Each switch is connected to an IEEE 802.11ac access point. Switch A is connected to Endpoint A. The switches are interconnected via IEEE 802.11ac links. The SDN Controller is connected to each switch via IEEE 802.11ac links.</p>
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Design of the Crosshaul Control Infrastructure</p>

	<p>Obj. 2: Specify the XCI's northbound (NBI) and southbound (SBI) interfaces</p> <p>Obj. 8: Crosshaul key concept validation and proof of concept</p> <p>Contributes to the following 5GPPP KPIs: Increase the number of connected devices per area, Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours, Scalable management framework, Enable deployment of novel applications reducing the network management Operational Expenditure, Orchestration of 5G-Crosshaul resources, Self-healing mechanisms.</p>
Measured KPIs	<p>Path Setup Time: Time taken since the path request arrives to the controller to the installation of the last OpenFlow (OF) FLOW_MOD in the node. We will gather timestamps from the mmWave/Wi-Fi SDN controller and the mmWave/Wi-Fi agents to do so.</p> <p>Throughput: Rate at which packets are received at destination service end-point.</p> <p>Path Restoration Time: Time elapsed from the instant in which the link goes down and the OF PORT_STATUS message is generated to the installation of the last OF FLOW_MOD for the new path in the nodes in response to the link-down event.</p>
Measurement tools	<p>Wireshark used for validation purposes and further analysis and debugging of the experiment.</p> <p>We will trace the necessary timestamps in both the mmWave/Wi-Fi SDN controller and the mmWave/Wi-Fi agents. Path setup time results gathered using Python scripts. It is important to note that the controller and agents involved in the experiment are synchronized by means of NTP and PTP for obtaining precise delay results.</p> <p>Iperf tool used to send and receive traffic from/to service endpoints.</p>
Measurement procedure	<ol style="list-style-type: none"> 1. Insert trace points in the code of the agents and controller to register the timestamp of the desired events. 2. Trigger the request to generate results of the corresponding experiment. Each experiment is conducted at least 10 times for revealing significant statistical results (average and confidence interval values) for all the KPIs under evaluation. 3. Post-process the collected data logs from the agents and controller with external Python scripts to compute the path setup time.
Constraints	<p>In this case, the path setup request will be generated by triggering a TCP traffic flow traversing the mmWave/Wi-Fi mesh domain amongst two service endpoints that, in turn, will generate an OF PACKET_IN when received at the transport node.</p> <p>The number of hops of the connection is initially subject to a maximum of three hops.</p>

	The system under test will only count with a mmWave link to perform path restoration measurements due to limitations presented by the experimental IEEE 802.11ad mmWave devices in terms of configuration of multiples interfaces in a single transport node.
Main results	<p>Path Setup Time vs Number of Hops of the mmWave/Wi-Fi traversed path.</p> <ul style="list-style-type: none"> Higher variability with a wireless control plane but within the order of tens of ms (less than 30ms). <p>Data Plane Throughput Vs Number of Hops of the mmWave/Wi-Fi traversed path.</p> <ul style="list-style-type: none"> Regardless of the number of hops, throughput degradation with wireless out-of-band control plane is similar to the one experienced with wired control plane. <p>Path Restoration Time vs Number of Hops of the mmWave/Wi-Fi traversed path</p> <ul style="list-style-type: none"> Higher variability with wireless control plane, yet path restoration time within the order of hundreds of ms (less than 300ms).

Discussion of results

The results presented focus only on two technologies, that is, 802.11ac and 802.11ad. This means that wireless transport nodes operate in both the 5GHz and 60GHz frequency bands.

In what follows, we characterize both the data and control plane of the single wireless domain by quantifying the time and the achieved throughput at the wireless data plane in the Barcelona site. As for the control plane, we focus on path setup/computation and restoration times, whereas for data plane, we measure achieved data plane throughput. Additionally, the control plane results were crosschecked against the performance of EdgeLink that is also based on the 802.11ad standard. A detailed description of EdgeLink is presented in [47]. We conducted experiments in a Wi-Fi/mmWave mesh backhaul with all data plane links configured to orthogonal channels.

In the following, we evaluate the reactive operation of the wireless domain; that is, when detecting a certain event (e.g., packet arrival, link failure), the management reaction is handled inside this domain, without reaching the controller higher up in the hierarchy, but only the wireless SDN controller. When evaluating the multi-domain scenario (in sections below), we characterize the proactive behaviour, i.e., when the wireless SDN controller receives requests from the network management application through the controller one level up in the hierarchy. In this way, we assess the advantages of taking local decisions confined to a single domain to quickly react to unexpected events before a decision can be taken from a more global point of view.

4.2.2.1 Path Setup (or computation) Time in mmWave/Wi-Fi mesh domain

As for path setup (or computation) time, we can observe that their values increase with the number of data plane hops. Furthermore, with a wireless control plane we can also observe in Figure 73 both increasing path computation time variability with the number wireless hops. The wireless control plane channel is the main reason for these increased latencies in terms of path computation time.

OF PACKET_IN are messages aiming at asking the controller how to treat a packet and OF FLOW_MODs are messages encapsulating the answer triggered by OF PACKET_IN. An OF PACKET_IN is sent to the controller when a data packet arrives to a transport node hosting an OpenFlow switch and there is no flow rule matching that packet. In this way, each OF PACKET_IN generated in the first hop generates an OF FLOW_MOD for all switches in the data plane path (2 switches for 1 hop, 3 switches for 2 hops and 4 switches for 3 hops). If OF FLOW_MOD arrives late to the second or third switches new OF PACKET_IN will be generated. As the number of hops of the wireless backhaul path increase, more OF PACKET_IN can be generated if data packets traversing a switch do not find a flow entry that matches with the flow. Thus, the new OF FLOW_MODs may not have arrived at all the nodes as data packets traverse the path, which in turn triggers more OF FLOW_MODS. Thus, the longer the path, the more likely to exhibit high Path Computation Times (see Figure 73): around 20ms for 2 hops and 30ms for 3 hops in the worst cases). In contrast, a wired control plane experiences similar values regardless of the number of wireless hops. As expected, the values reported for the wired control plane are almost constant with an average computation time of around 5ms irrespective of the number of hops of the computed path. The obtained results using a wired control plane are comparable to the EdgeLink performance when using also a wired control plane, i.e. average path set up times of 3ms, 6.5ms and 9.8ms for one hop, two hops and three hops respectively.

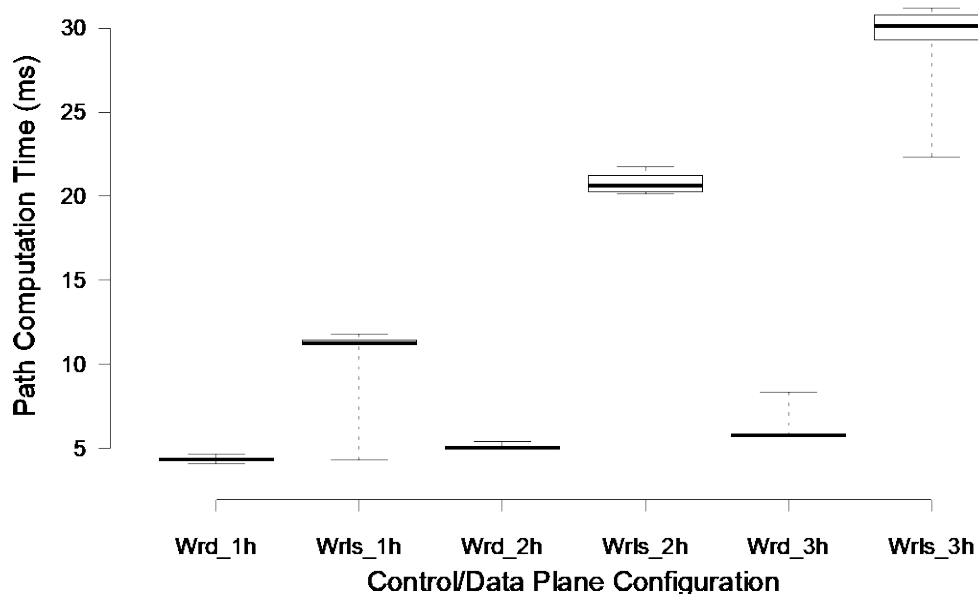


Figure 73. Path Computation (or Setup) results for the mmWave/Wi-Fi mesh domain

As mentioned before, the benchmark in this case, corresponds to an out-of-band wired control channel with the controller, which is the norm for the most common scenarios in which SDN has initially been deployed (e.g., data centres, campus networks [28]).

4.2.2.2 Data Plane Throughput in mmWave/Wi-Fi mesh domain

The specific novelty that goes beyond SOTA in this experiment is the assessment of whether the desired service degree required by a control plane channel can be offered with low-cost wireless technologies (e.g., Wi-Fi). In particular, we are interested in revealing what common set of functions (or services) that any SDN controller provides (e.g., establishment of paths between endpoints) is suitable with a low-cost control plane. In this way, CAPEX savings can even reach not only the data plane but the control plane. Again, the benchmark is the out-of-band wired control plane.

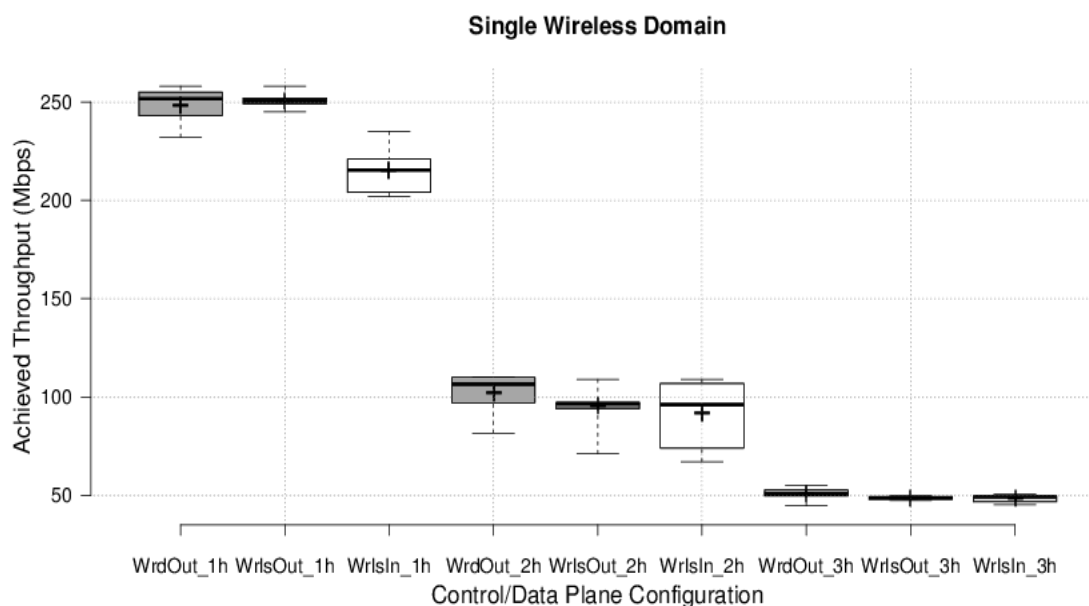


Figure 74. Data Plane Throughput Results for an mmWave/Wi-Fi mesh domain

The Figure above reports the boxplot characterizing the range of values of attained throughput of a single TCP Cubic flow amongst two endpoints in three different setups: i) the control plane entails an out-of-band wired control plane labelled as WrdOut, ii) the control plane uses an out-of-band wireless frequency band labelled as WrIsOut, and iii) the control plane is wireless and is configured to a frequency band shared with the data plane, which is labelled as WrIsIn. These three different setups are evaluated over a data plane of 1, 2, and 3 wireless hops.

First, we can observe a decreasing throughput by at least a factor of 2 with the number of hops, irrespective of the Control/Data plane configuration. The reason for this decrease is different for each 1-hop increase. In the 2-hop case, despite the use of 80MHz orthogonal channels (5180MHz and 5780MHz), we observe a decrease of throughput (around a 2.5x decrease) because the second wireless backhaul link is non line of sight (NLOS). With three hops, we experience a degradation of throughput by a factor of 2 due to frequency reuse (5180MHz) configured in the third hop.

Results reveal that, as expected, the setup exhibiting maximum throughput values is the wired control plane, closely followed by an out-of-band wireless control plane, whereas the one that suffers more from throughput degradation is the wireless in-band configuration control plane. We can observe that for 1 and 2 hops, the wired control plane case exhibits a noticeable variability. We hypothesize that for one hop, the factor determining this variability is the variable external interference of devices in the frequency band of 5GHz. For two hops, the predominant factor for the variability is the path diversity exhibited and the external interference. Interestingly, for a configured three-hop data plane, we observe how the size of the boxplots stretches for all the three potential configurations compared to their 2-hop counterpart. This is mainly attributed to the fixed path followed for all the traffic flows under evaluation, whereas for paths of 2-hops there are diverse options. Also values are quite similar due to the predominant contention and losses exhibited by the wireless data plane. Here external interference does not present a huge influence.

4.2.2.3 Path Restoration Time in mmWave/Wi-Fi mesh domain

What has been measured is whether the system is able to, within an acceptable time (i.e., a few seconds), maintain the backhaul/fronthaul path amongst two end points after a failure in the current established path. Both wired (benchmark) and wireless control channels are compared.

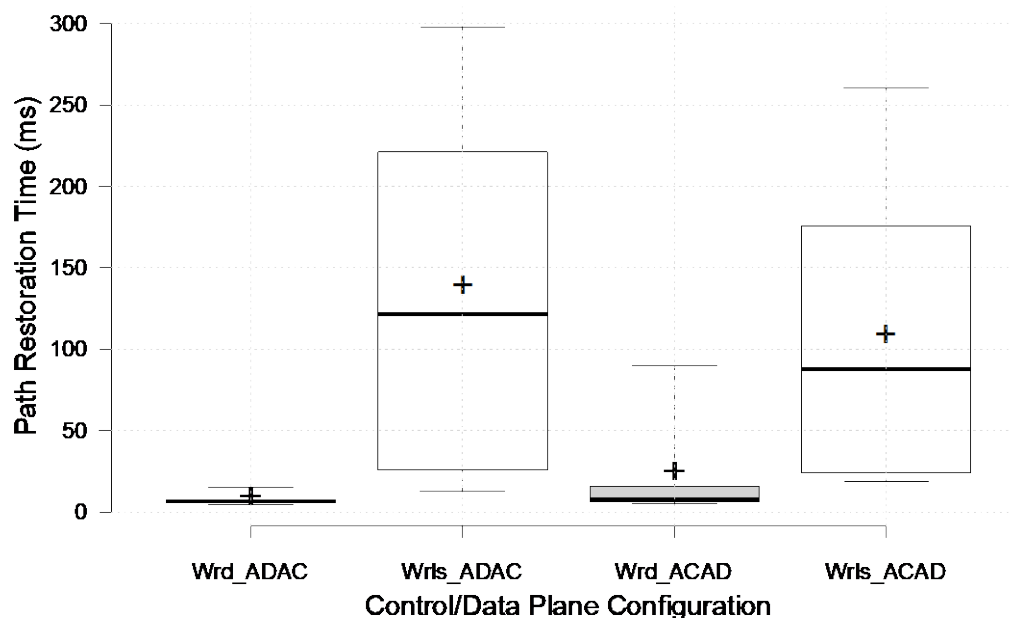


Figure 75. Path Restoration Results in the mmWave/Wi-Fi domain

In particular, we conduct ten repetitions for each experiment analysing restoration properties of the mmWave/Wi-Fi mesh. In each experiment, we trigger a failure in one of the wireless links used to route the traffic for the ongoing flow traversing a three-hop wireless path. A failure in one of the mmWave wireless backhaul links forming part of the path causes a reroute of the ongoing traffic flow using the 802.11ac link. Figure 75 shows the path restoration time for a change in data plane from 802.11ac to 802.11ad and from 802.11ad to 802.11ac for i) a wired control plane (*wrd_ACAD* and *wrd_ADAC*) and ii) wireless control plane (*wrlss_ACAD* and *wrlss_ADAC*). Wireless control channel results reveal a higher degree of variability than those observed with a wired control plane. Yet, the exhibited results with a wireless control plane observed during the ten repetitions are within the order of hundreds of ms (less than 300ms). This degradation of results compared to wired control is caused by the lower stability of the wireless environment. It is worth noting that the wired control plane results are comparable to the EdgeLink performance i.e. average path restoration time of 11.7 ms when performing a change between two of its 802.11ad interfaces.

4.3 Experiments on multi-domain service provisioning and restoration

The previous section 4.2 presents the individual characterization of the multiple domains in the setup. The system insights and the obtained results serve as a guide to perform the following end-to-end system characterization and to better understand the new obtained results. With respect to the previous section, the following experimentation also provides an analysis of the system from

a service level perspective with the deployment of a virtualized LTE mobile network service in the described system.

4.3.1 Experiment #14: End-to-end characterization. Network Orchestration across multiple heterogeneous domains: control and data plane characterization

This experiment integrates the previous ones and conceptually extends Experiment #12, related to the assessment of the SDN-based control of the Optical Transport Network, to a setting characterized by having multiple domains combining different technologies. In addition to this, data plane measurements will be performed after the hierarchical orchestration of network resources provides the required connectivity.

Network Orchestration across multiple heterogeneous domains: control and plane characterization	
Description	<p>The experiment demonstrates and evaluates the performance of the provisioning service within the 5G-Crosshaul XCI.</p> <p>In particular, from the control plane perspective, it evaluates the resource-efficient connection request (coming from RMA) and path provisioning through a multi-technology multi-layer optical/wireless network. From the data plane perspective, it characterizes the data plane performance of the considered multi-domain scenario, both at the transport level and from the point of view of the end-to-end service deployed over the transport network.</p> <p>In summary, the overall objective of this experiment is to validate the ability of the 5G-Crosshaul system to deploy a network service in a heterogeneous multi-domain network in an automated way by exploiting the defined APIs and to show that by so doing, provisioning time is substantially reduced with respect to state-of-the-art practice based on manual configuration of each domain.</p>
State of the art	<p>From the control plane perspective:</p> <ul style="list-style-type: none"> • Per-domain provisioning using distributed control plane or a centralized management system, very limited deployments of SDN control for transport networks. • Per-vendor or per-technology transport domains, each with its isolated control plane with proprietary extensions, but no end-to-end global management of transport network. • Lack of network orchestration across multiple domains and technologies beyond very basic proof-of-concepts. Reduced number of controllers, and in well-defined scenarios and scopes (e.g. IP over optical). Very limited deployments and usually in homogeneous networks, lack of integration of transport segments, mostly covering Layer 2 and Layer 3. • No integration of 5G technologies such as mmWave in the overall overarching control and orchestration. • No integrated edge wireless transport and multi-layer optical network composed of packet-switched wired aggregation and optical core. <p>For a more detailed analysis, please see [29] and [30].</p>
Improvement from State of the art	<p>In summary, the hierarchical XCI system we present in this demonstration offers more scalability, modularity and security over state of the art solutions because an ABNO-based parent controller abstracts information of the heterogeneous domains to let RMA make optimal centralized resource</p>

	<p>management decisions. Furthermore, hierarchical setups enable various levels of reaction. At the lowest level, fast reactions can be provided by the technology/domain-specific controller. More global reactions can be handled by the parent controller (and network management apps exploiting the parent’s services) up in the hierarchy.</p> <p>Furthermore, service setup times are substantially reduced by exploiting the APIs exposed by the XCI, which go down from manual setup times of the order of hours or days (values may range from 20 hours to 6 days, as detailed in Appendix 1: Current generic network provisioning procedure in a telco provider infrastructure) to seconds, hence contributing to reduce service deployment time from months to minutes, as initially envisioned as one of the key 5G goals.</p>
<p>5G-Crosshaul Use Cases</p>	<p>Transversal to multiple 5G-Use cases, related to the automation of network connectivity provisioning using the 5G-Crosshaul XCI. In particular:</p> <ul style="list-style-type: none"> • Dense urban information society (efficient resource management end-to-end allows serving more traffic with the same resources enabling a cost-effective dense deployment); • Media distribution flows as one demanding example for which resource efficient management is particularly required.
<p>System under Test</p>	
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Design of the Crosshaul Control Infrastructure</p> <p>Obj. 2: Specify the XCI’s northbound (NBI) and southbound (SBI) interfaces</p> <p>Obj. 6: Design scalable algorithms for efficient Crosshaul resource orchestration</p> <p>Obj.j 7: Design essential Crosshaul-integrated (control/planning) applications</p> <p>Obj. 8: Crosshaul key concept validation and proof of concept</p> <p>Contribution to the following 5GPPP KPIs: Increase the number of connected devices per area, Enable the introduction/provisioning of new 5G-Crosshaul services in the order of magnitude of hours, Scalable management framework, Enable deployment of novel applications reducing the network management Operational Expenditure, Orchestration of 5G-Crosshaul resources.</p>

Measured KPIs	<p>Path Setup time, evaluated from the parent ABNO and/or the RMA. It is defined as the time it takes to provision an end-to-end multi-domain path through the mmWave/Wi-Fi and multi-layer optical domains.</p> <p>Control plane characterization of involved protocols (control overhead rate and packet size distribution).</p> <p>CPU profiling of RMA application (relative execution time of various functions run by the RMA).</p> <p>Service setup time, evaluated from the parent ABNO, the RMA and the different involved child SDN controllers. It is defined as the time it takes to provision all the network resources required to interconnect the building blocks conforming a complete mobile network (including EPC and LTE access network). The mobile network under test features both fronthaul (PDCP/RLC RAN split) and backhaul links.</p> <p>Local recovery time, evaluated from the wireless child SDN controller. It is defined as the time it takes once a link failure/link recovery is detected by the corresponding transport node until all the affected flows of the LTE service are reconfigured by the wireless child SDN controller at its corresponding domain.</p> <p>Centralized recovery time, evaluated from the RMA. It is defined as the time it takes to reconfigure the flows once a link failure is detected by the corresponding transport node until all the affected flows of the LTE service are reconfigured by the RMA in the whole multi-domain, multi-technology setup.</p> <p>Data plane performance: Throughput measured between a) two service end-points of the transport network and b) two different end-users of an LTE service traversing fronthaul or backhaul transport links. The throughput measurement will be determined as the maximum achievable bandwidth with a packet loss lower than 1%. Furthermore, measurements of jitter, packet loss and RTT are also considered.</p>
Measurement tools	<p>Network analysers, packet capturing software, such as wireshark and tcpdump.</p> <p>Ad hoc request generators.</p> <p>Timestamps of wireshark/pcap traces at different points of the network of NTP/PTP synchronized machines.</p> <p>perf for CPU profiling.</p> <p>Iperf tool as traffic generator and to gather data plane performance metrics.</p> <p>Ping network utility to measure the round-time time (RTT).</p>
Measurement procedure	<ol style="list-style-type: none"> 1. Trigger the request of a path or a whole service using the RMA dedicated interface. 2. Serve the request by the XCI, involving the parent ABNO controller and the child SDN controllers down in the hierarchy. 3. Wait for the confirmation of the establishment of the connection. 4. Parse logs, traces and related data to obtain KPIs as path setup delay, service setup time, local/centralized recovery time and data plane performance.

Constraints	<p>Obtained results, both from the control and data plane perspective, are conditioned by the use of VPN tunnels to interconnect different elements of the setup.</p> <p>The number of wireless hops of the connection is initially subject to one wireless backhaul/fronthaul hops to test end-to-end performance.</p> <p>The injected traffic to perform data plane measurements will consist of UDP traffic.</p>
Main results	<p>From the control plane perspective:</p> <ul style="list-style-type: none"> • Average path setup time of tens or hundreds of milliseconds in the wireless and single-layer optical domains as seen from the child controllers. • Average path setup delays in the order of seconds (for one bidirectional path), as seen from the RMA (3.971s on average). This is in contrast with the values in the order of hours, or even days, of the current manual setup. • The assessed XCI deployment allows the introduction of a complete E2E network service in the order of seconds (on average 10.5 seconds) across geographically distributed control plane entities and transport network domains. • It is also shown and assessed the flexibility provided by the XCI in the process of a service recovery triggered by a link down event. This recovery process can be done, either locally at the child SDN controller level (the low level of the control hierarchy) or centralized at the RMA level (the upper level of the control hierarchy). The former allows much lower recovery (0.299s on average) at the cost of potentially suboptimal resulting paths, unlike the latter, which takes 6.652s on average for re-establishing an optimal multi-domain path. <p>From the data plane perspective:</p> <ul style="list-style-type: none"> • Data plane throughput of the traversed E2E multi-domain path is around 153 Mbps, mainly limited by a VPN connection between remote sites. • A virtualized environment deploying an LTE mobile network service is able to achieve up to 140Mbps between users placed E2E of the setup over a backhaul connection and 116Mbps between users over a fronthaul connection in the scenario under evaluation. • Round-trip time and jitter of the data plane of the traversed multi-domain path is around 56.601ms and 50μs, respectively. Jitter increases up to an average of 190 μs when evaluating it E2E over the virtualized LTE mobile network service in the scenario under test. On the other hand, the RTT maintains similar values.

Discussion of results

Demo 3 involves several components deployed in diverse geographical locations as mentioned before. Notably, the RMA, hierarchical SDN network orchestration and the transport networks

can be deployed in different locations and their interconnection are carried over dedicated (secure) tunnels over the Internet (based on IPSec and using software such as Racoon¹⁰ or OpenVPN¹¹). These tunnels are important in order to fully characterize the control plane and data plane KPIs mentioned previously. Consequently, these links will affect not only setup times as perceived by the element on top of the control hierarchy (RMA application) but also performance at the data plane level.

In order to evaluate the impact of the established tunnels in the setup, this section presents an assessment of control plane and data plane performance when progressively adding to the scenario setup the different components which are after a dedicated VPN tunnel, namely the RMA and the mmWave transport network domain and its associated child SDN controller (Wireless SDN Controller_2).

4.3.1.1 Control Plane performance assessment

4.3.1.1.1 Multi-domain/multi-technology hierarchical XCI

This subsection presents the control plane assessment analysis of the Demo3 setup including the hierarchical control plane deployment (pABNO and corresponding child SDN controllers) to orchestrate network resources in the mmWave/WiFi and the multi-layer optical transport network, which are placed in the Barcelona 5G-Crosshaul Testbed site. The following figure shows the actual network topology for the tests and the abstracted topology as presented to the E2E transport orchestrator (parent ABNO).

The following figure shows the actual network topology for the tests and the abstracted topology as presented to the RMA application. The topology shown for the wireless segment corresponds to the physical deployment. However, the nodes belonging to the multi-layer optical network (the ones labelled as *OpenFlow nodes* and *ROADM* at the bottom part of the figure), have been abstracted as a ring of four elements, showing only its connection to other domains. Hence, the specificities of this domain are hidden to the pABNO and management applications at the application plane.

From the point of view of the pABNO, the figure below (Figure 77) shows the histogram and CDF for the end-to-end setup delay, from the reception of the request in the pABNO NBI to the completion of the operations. The multiple peaks in the histogram reflect an implementation artefact through which the ABNO (child and parent) wakes up every 50ms and processes the received requests during that period. However, the most remarkable observation is the increase of the average setup delay from tens or hundreds of milliseconds in the wireless and single-layer optical domains as seen from the child controllers (see results of Experiment #12: Assessment of the SDN-based control of the Optical Transport Network (optical domain) and Experiment #13: Assessment of the SDN-based control and data plane for the mmWave/Wi-Fi mesh domain) to seconds as seen from the parent (see Table 15). This is due to various factors. First, previous sections only presented unidirectional single-layer values. On the other hand, this section presents values for multi-layer (i.e., Ethernet over wavelength-switched optical network) and bidirectional connection setup, out of which an average of 2.867s. are spent in the multi-layer optical network. Second, there is the interaction and message processing between the parent ABNO and child controllers. Third, there is the sequential handling of some of the messages to set up the E2E path. The table below provides some statistics of a path setup delay as seen from the pABNO.

¹⁰ <https://www.netbsd.org/docs/network/ipsec/rasvpn.html>

¹¹ <https://openvpn.net/>

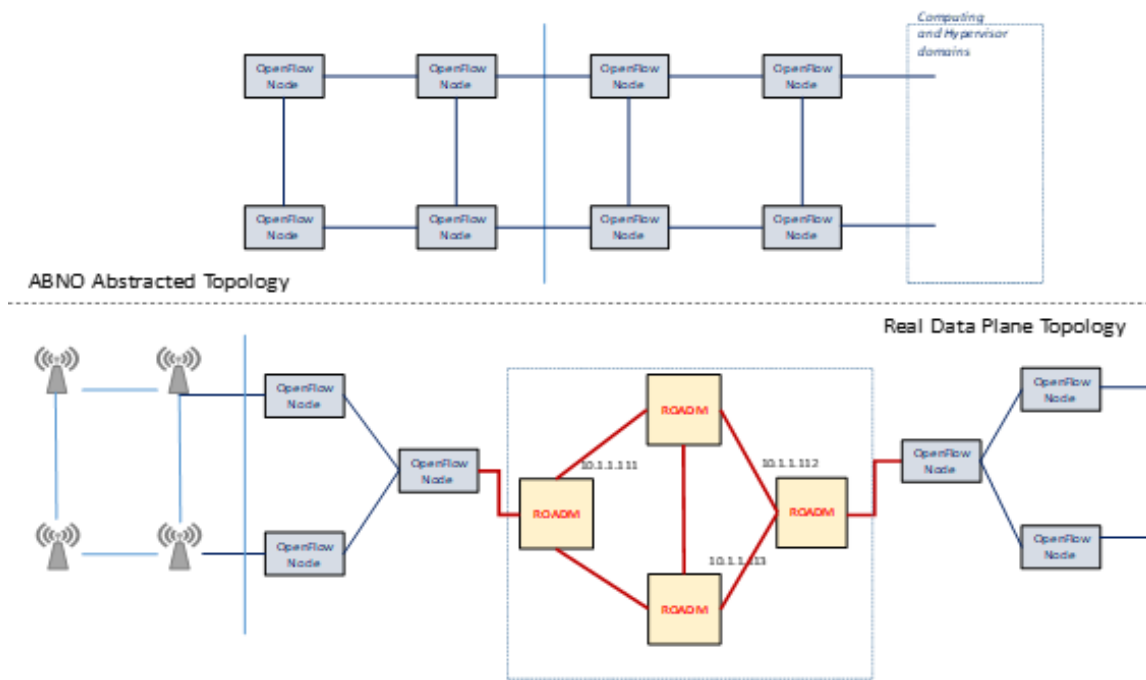


Figure 76. Network topologies as deployed (bottom) and as aggregated by the pABNO controller (top).

Table 15: Setup delay. Parent ABNO [seconds].

	Average	Min.	25-percentile	75-percentile	Max.
pABNO	3.349	3.092	3.294	3.398	3.693

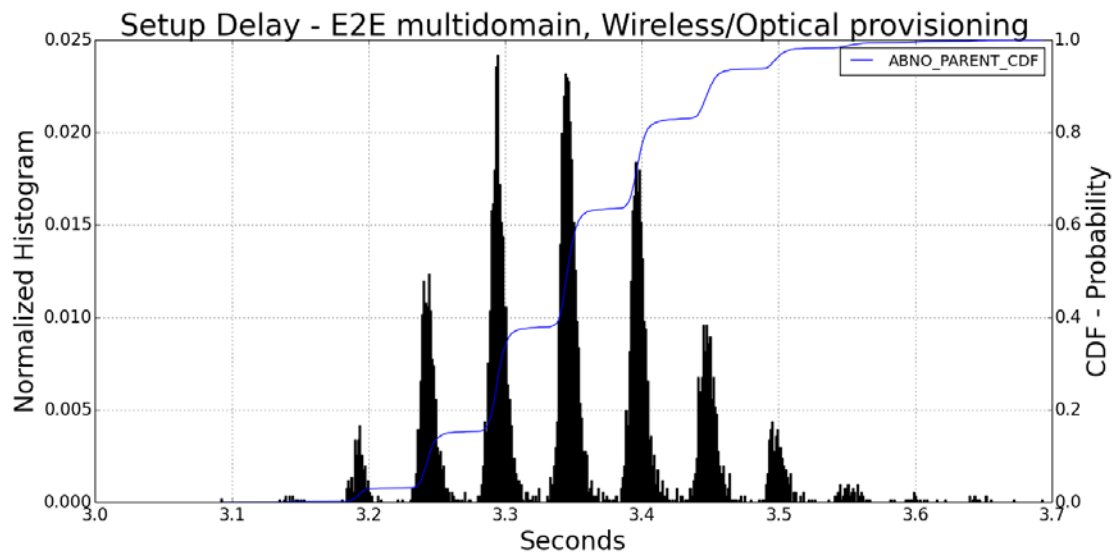


Figure 77. Histogram and CDF of the setup delay seen by the ABNO orchestrator for the considered topology

To retain an order of magnitude and to validate the OPEX savings related to control plane and service provisioning automation, we can state, within reasonable confidence that, on average, the setup delay of provisioning the connectivity is 3.25 seconds (order of units of seconds), which is much lower than that of current manual setup (in the order of hours or days).

4.3.1.1.1.1 Control plane interface characterization

Regarding the control plane message exchange characterization and overhead (which is interface-specific), we evaluate parameters such as average message size (bytes), message rate and flows (messages per second, which are, in turn dependent on the supporting control channels). The following results represent the control plane overhead of the COP protocol between the parent ABNO and the domain controllers (child ABNO and Wireless SDN controller). The carried out experiment consisted on 100 E2E service requests creations and deletions over a period of 300s. The request inter-arrival follows an exponential distribution with a time between requests of 1s and the holding time in the system is 1s as well.

In the Figure 78, it is represented the total captured message I/O in terms of bit rate through the COP interface during the experiment.

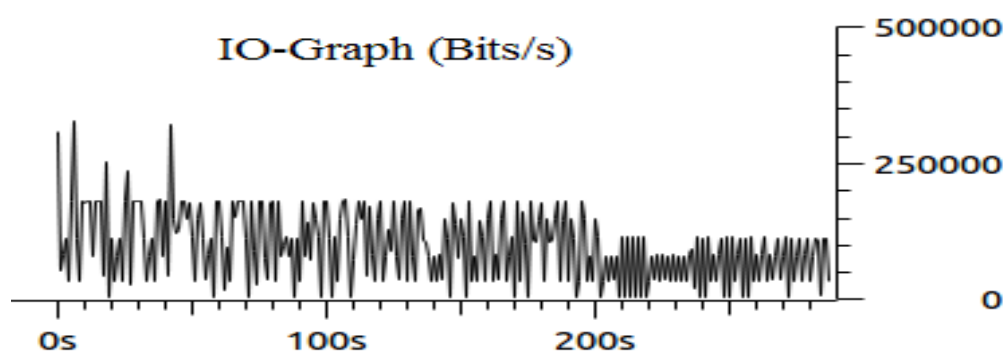


Figure 78. Bit rate of COP protocol message exchange during the previous experiment

The following table includes the packet size distribution of the captured traffic.

Table 16: Packet distribution of COP protocol message exchange during previous experiment.

Topic / Item	Count	Average(Bytes)	Min val(Bytes)	Max val(Bytes)	Rate (ms)	Percent	Burst rate	Burst start
Packet Lengths	15447	447.96	66	2018	0.0536	100%	0.9000	42.917
0-19	0	-	-	-	0.0000	0.00%	-	-
20-39	0	-	-	-	0.0000	0.00%	-	-
40-79	11374	67.89	66	74	0.0395	73.63%	0.6500	42.917
80-159	1065	93.95	83	159	0.0037	6.89%	0.1000	0.971
160-319	548	220.35	187	290	0.0019	3.55%	0.0500	6.916
320-639	924	458.57	356	601	0.0032	5.98%	0.0600	0.000
640-1279	384	741.47	740	744	0.0013	2.49%	0.0200	0.000
1280-2559	1152	1526.37	1367	2018	0.0040	7.46%	0.0600	0.000
2560-5119	0	-	-	-	0.0000	0.00%	-	-
5120-4294967295	0	-	-	-	0.0000	0.00%	-	-

It is worth mentioning that the overhead introduced by COP packets will be mainly determined by the amount of domains in the transport network and the topology offered by child SDN controllers towards the pABNO. For instance, the packet length containing the orchestrated path will be different depending on the number of hops it contains.

4.3.1.1.2 Multi-domain/multi-technology hierarchical XCI and application plane (RMA)

As depicted in Figure 68, the RMA runs on top of the ABNO parent controller, enabling us to dynamically adapt flow paths to current network constraints. The main goal of RMA is to set Crosshaul flows between RRHs and XPU and select the functional split between RRHs and said XPU. The goal is to find a combination of paths/splits that is feasible and maximizes the amount of eNB functionality centralized in XPU. The RMA is developed in Java and comprises different classes that provide the desired functionality. We use a REST interface to communicate with the ABNO controller.

The RMA initializes by recovering the current topology (and abstracted view of it) from the controller and computes the optimal paths/functional splits between XPU and RRHs in the topology. In the event of a link failure, or in case a new link is added to the topology, the application updates the current network data structures and searches for a new solution. The different classes that serve as key architectural components are the following ones:

- **HTTP Client class:** The HTTP Client class is used to create the different HTTP request objects that will be used to communicate with the parent controller's REST API. It allows creating objects that encapsulate common HTTP methods such as GET, POST, DELETE and PUT. The bodies of the HTTP methods are filled out with JSON data objects that the controller understands.
- **Communication manager class:** The communication manager class processes all the HTTP objects regardless of its specific methods. That is, all the HTTP objects are queued in a FIFO queue while several threads process them. Furthermore, the set of threads use different callback objects to pass the results of the different HTTP processed objects back to the class that created them.
- **Network class:** The network class provides different data structures and algorithms that are needed to save the network state and compute the different paths. Specifically, it provides graph data structures that let us save a snapshot of the network. Thus, the data structures can be easily accessed and used to run different graph algorithms. Furthermore, it also contains different path computation algorithms that will be used to calculate the best routes.
- **Manager class:** The manager class glues all other classes to provide the desired functionality. It contains the start and shutdown routines. Plus, the algorithms used to create the request to install a path using an HTTP POST request and delete a path using an HTTP DELETE method. Besides, this class detects a link loss and recovery (upon the notification of the underlying controller) so that the data structures can be updated accordingly so that the path-finding algorithms can produce a better result.

We start the evaluation by comparing the setup times for one bidirectional communication as seen from the pABNO (Table 15) and from the RMA (Table 17). As far as the application plane is concerned, recall that there is a tunnel set up between the Barcelona site and Heidelberg with an approx. RTT of 60ms. However, the difference is of 600ms approximately, which is due to the processing carried out at the RMA. With respect to Figure 79, the installation of n flows is not n times the values in because the most expensive operation is the creation of the initial lightpath. Once it is established, the rest of call requests can be processed faster, as they reuse the same already established lightpath, which is handled at the optical child SDN controller.

Table 17: Setup delay hierarchical XCI: Parent ABNO vs RMA [seconds].

	Average	Min.	25-percentile	75-percentile	Max.
pABNO	3.349	3.092	3.294	3.398	3.693
RMA	3.971	3.667	3.804	4.046	5.281

In the following, we evaluate the performance of the RMA application and its interface with the ABNO controller. Figure 79 depicts box and whiskers for latency measurements incurred in performing 4 basic operations:

1. Request the underlying topology to the ABNO controller. This stage is labelled as “start”.
2. Compute the optimal path for 5 flows of 5Mbps each between 2 random pair of nodes. This stage is labelled as “find”.
3. Request the ABNO controller for installation of the computed routes. This stage is labelled as “install”.
4. Request the ABNO controller to clear the installed routes for the 5 flows. This stage is labelled as “shutdown”.

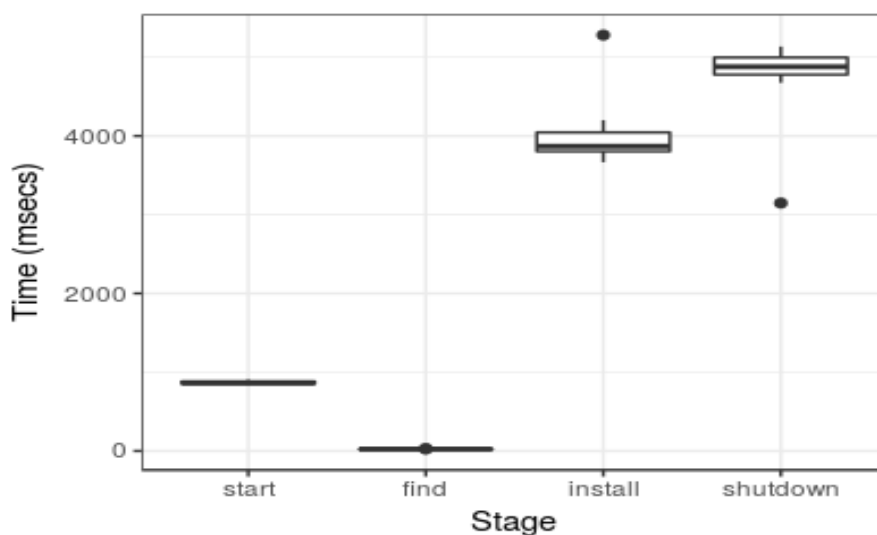


Figure 79. Latency measured for basic RMA operations for 5 flows

In any case, this is still in the order of seconds, which is much lower than current manual setup times in which multiple technological domains are involved (in the order of hours or days) (see Appendix 1). Next, we make a CPU profiling of the RMA application at each of the four stages defined above. Figure 80, Figure 81, Figure 82, Figure 83 show, respectively CPU profile information for stages “start”, “find”, “install” and “shutdown”. Each figure illustrates the relative amount of CPU time consumed (x axis) on each software function (y axis) described above.

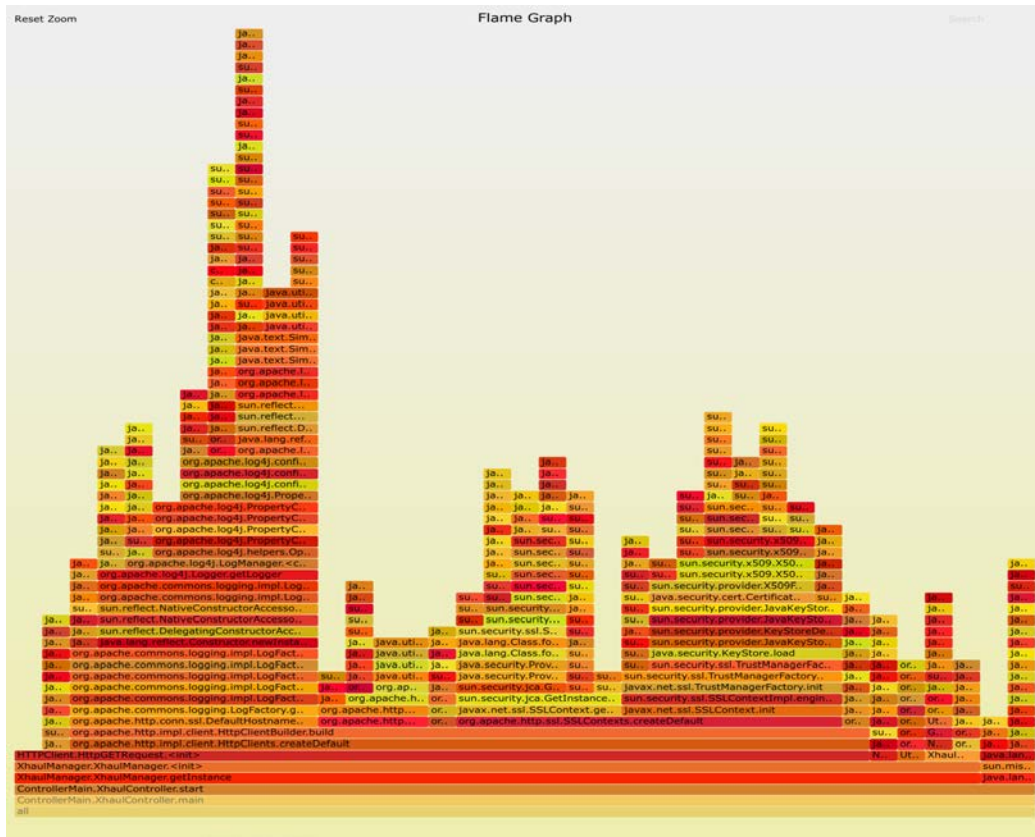


Figure 80. RMA. Characterization of the calls during the “Start” phase

The *start* routine, profiled in Figure 80, may be divided into three different parts. First of all, it loads the necessary classes we need to operate (*XhaulController.start()*). In particular, the HTTP client class is one of the main functions loaded at this stage. As we can see from the figure, initializing the HTTP client requires calling many different library functions. These are stacked on top of function *HttpClient.createDefault()*. These functions encapsulate all the functionality that we require to send HTTP messages. Secondly, it sends an HTTP GET request to the controller to recover the topology, and creates the necessary data structures to save the current network state. Third, it computes the paths using the recovered topology and creates the necessary JSON objects that will be sending inside the HTTP POST requests to install each path.

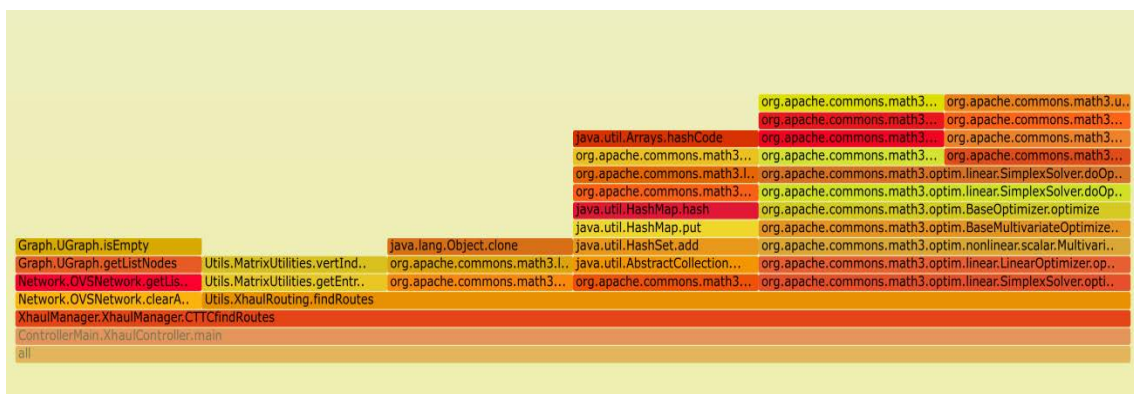


Figure 81. RMA. Characterization of the calls during the FindRoutes operation

The *FindRoutes* routine depicted in Figure 81 belongs to the Manager class (*XhaulManager*). In essence, this function tries to find the different paths between each source and each destination

according to the best centralization degree. As we can see in the figure, this routine starts building the necessary data structures to run different optimization methods. Mathematical functions and data structures are built on top of the apache math3 commons library. Next, data structures are passed on as inputs to the optimization functions. The result is passed back to the manager class so that it can continue with the installation routine to install the paths in the network. From the figure we can see that our Simplex solver consumes roughly around 1/3 of the total CPU time of this stage.



Figure 82. RMA. Characterization of the calls during the Install Paths operation

Whenever the manager class calls the *installPaths* function, detailed in Figure 82, it also passes on the paths that the *findPaths* routine has found. Then, the installation routine loops through the solution and creates the JSON objects that will be inserted in the HTTP POST methods. The HTTP POST request are sent to the communication manager class and executed so that the controller interprets the JSON objects and install the paths in the network.

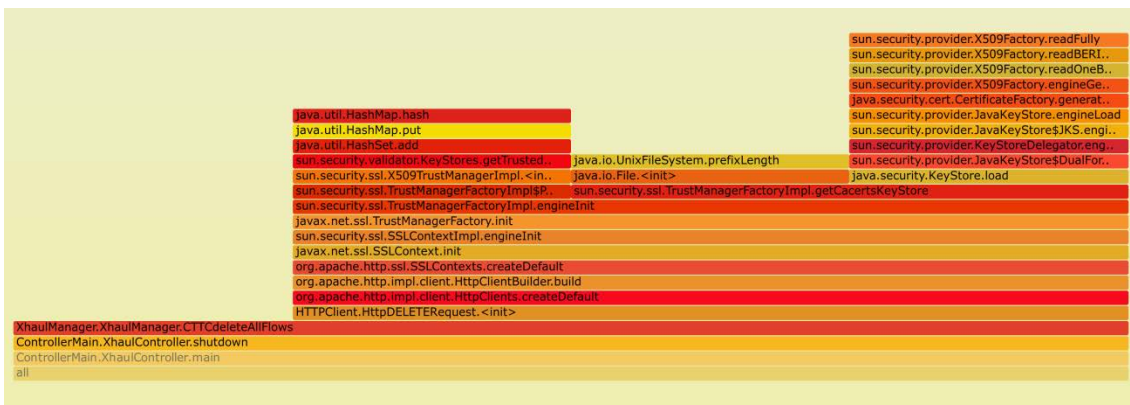


Figure 83. RMA. Characterization of the calls during the Delete Flows operation

The shutdown routine, detailed in Figure 83, essentially unloads the classes loaded in the start routine and deletes the current installed paths. It creates a HTTP DELETE object for each current path installed and sends them to the controller so that it erases all the current forwarding rules installed. After that, it exits the application.



Figure 84. RMA. Characterization of the calls during the HTTP Client Requests

Finally, as mentioned above, communication with the parent control is done through a FIFO queue which is processed in independent threads. The CPU profile of the thread in charge of the communication of this experiment is shown in Figure 84. The figure illustrates the relative CPU time incurred in processing the DELETE, POST, and GET HTTP messages towards the ABNO controller. When an HTTP request has to be processed, the HTTP object created by the HTTP client class is passed on to the communication manager class. The HTTP object is queued and further processed by different threads. In Figure 84 we can see how HTTP DELETE, GET and POST requests are processed by a java Thread. Furthermore, as we are using the apache HTTP commons library to implement our HTTP client, after the thread processes the object, the library functions are called. We can see how the same thread can process different HTTP request methods.

4.3.1.1.3 Multi-site XCI and application plane deployment (RMA)

This section presents a time analysis of the control plane performance when deploying a complete service in the depicted system under test (Figure 68). This analysis will focus on the service setup time and the service recovery time under two strategies: a) local recovery procedure performed by a child SDN controller and b) centralized recovery procedure performed at the RMA. These two recovery strategies illustrate the various levels of reaction that a hierarchical setup enables.

The deployed service is an LTE mobile network consisting of two eNodeBs placed in the different edge wireless transport domains and an RRU-BBU pair featuring PDCP/RLC split placed in the mmWave/WiFi domain, which connect to the different EPC entities (MME, SPGW), which, in turn, is connected to the multi-layer transport network. Hence, the RMA will determine the appropriate paths among different mobile network entities for the fronthaul and backhaul traffic profiles.

In order to establish the LTE mobile network service, eight REST calls (16 flows in order to provide bidirectionality to the different calls) are required. This analysis is different from previous sections since in this case the set of created flows is not arbitrary, but has a joint purpose (i.e. mobile network service deployment). The eight required paths are: from RRU to BBU, from BBU to MME, from BBU to SPGW, from enodeB1 to MME, from enodeB1 to SPGW, from enodeB2 to MME, from enodeB2 to SPGW, from SPGW to MME.

4.3.1.1.3.1 Service Setup Time analysis

Regarding the service setup time analysis, we evaluate the time it takes to provision the complete LTE mobile network service in the presented setup. The measured time interval starts when the RMA computes the required paths and ends when it receives the confirmation from the pABNO that all the requested paths have been installed at the different domains controlled by their corresponding child SDN controllers. The following results have been obtained when performing 100 LTE mobile network service transport flow creations and deletions over a period of 100 minutes (the service is created and deleted every 30s).

The following figures show the histogram and the CDF (seen from the RMA) showing the control plane latency to establish and delete the described LTE mobile network service.

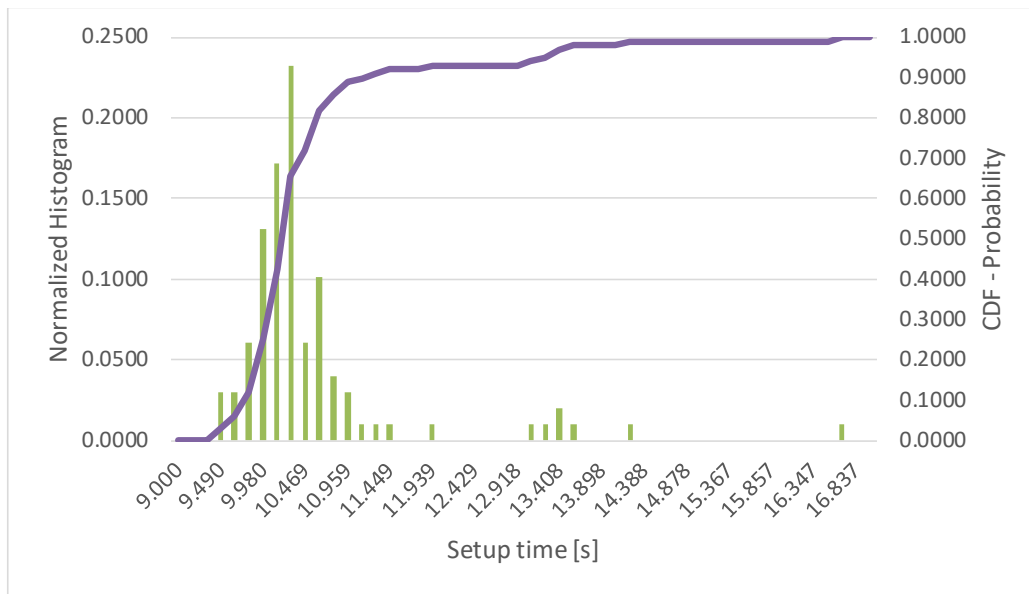


Figure 85. Normalized histogram and CDF of the service setup time seen by the RMA

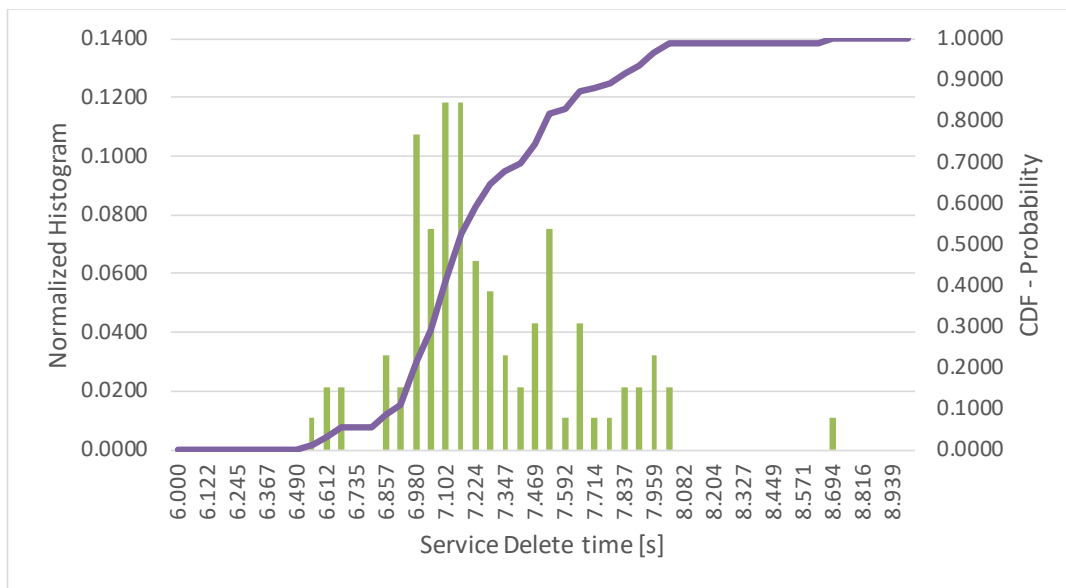


Figure 86. Normalized histogram and CDF of the service delete time seen by the RMA

Table 18 and Table 19 present different statistical measurements seen from the different elements in the hierarchical control plane for its associated service setup and deletion time, respectively.

Table 18: Service setup time seen from the different control plane elements [seconds]

Control entity	Average	Min.	25-percentile	75-percentile	Max.
RMA	10.467	9.343	9.962	10.544	16.658
pABNO	10.112	8.839	9.506	10.150	16.443
Child ABNO	9.794	8.507	9.147	9.808	16.160
Wireless SDN Controller_1	7.855	6.721	7.335	7.879	15.794
Wireless SDN Controller_2	7.659	5.649	7.332	7.814	10.889

Table 19: Service deletion time seen from the different control plane elements [seconds]

Control entity	Average	Min.	25-percentile	75-percentile	Max.
RMA	7.240	6.505	7.011	7.490	8.668
pABNO	6.921	6.033	6.699	7.094	8.339
Child ABNO	6.572	5.684	6.371	6.754	7.975
Wireless SDN Controller_1	5.130	4.362	4.961	5.260	6.537
Wireless SDN Controller_2	5.330	3.063	5.233	5.570	5.933

From the previous figures and tables, we can extract that in 90% of the samples, the time it takes the RMA to calculate and validate the installation of the whole set of call requests to interconnect the different deployed mobile network entities is lower than 11 seconds, with an average value of around 10.5 seconds and a minimum value of 9.3 seconds. Hence, the presented system verifies that the automated orchestration capabilities of a 5G-Crosshaul network could satisfy the 5GPPP KPIs of enabling the introduction/provisioning of new services in the order of magnitude of minutes/hours, or even seconds in this case. This constitutes a big improvement with respect to current network provisioning times in a telco provider infrastructure, as presented in Appendix 1: Current generic network provisioning procedure in a telco provider infrastructure. According to the procedure presented in Appendix 1, current network provisioning times could span between 18 hours (just documentation and sequential execution of network configuration operations) and up to 6 days (when waiting to scheduled network operation windows).

Turning back to the measured values, we can see from the previous tables how these values increase at the different stages of the hierarchical control plane due to the processing operation at each stage, the message passing time between control entities in the hierarchy (RMA-pABNO and pABNO-child controllers) and the delay introduced by the VPN connections (RMA-pABNO and pABNO-mmWave mesh domain). This last factor, the higher variance of latency experienced in the VPN connections, could explain the maximum values experienced in the service setup time. From a deeper analysis of the measured values at the pABNO to set up each call, the most expensive calls in terms of setup time are those requests which involve setting up a bidirectional

multi-domain path traversing the multi-layer optical network due to the set of actions which have to be done at its associated control entities and in the set of network elements hidden under the abstraction provided by the multi-layer optical child SDN controller to the RMA. In particular, the first of such calls is always the one which requires more time, around 2.9 seconds, coinciding with the values stated in previous section *Multi-domain/multi-technology hierarchical XCI*, due to the need to set up the initial lightpath and the Ethernet service on top. After this initial one, all the rest reuse the same lightpath.

Finally, it is worth mentioning the reason why the mmWave/Wi-Fi SDN controller (Wireless SDN Controller_1) presents these measured values. The pABNO receives the different calls from the RMA, decomposes them in different calls for each network domain, which are then sent to the underlying child controllers. Then, all the operations for this call at the different domains are done and a validation is sent to the pABNO, which then acknowledges the RMA. As the most time consuming call setup operation times are done at the multi-layer optical domain and almost all call requests for the mmWave/WiFi transport domain traverse the optical domain, the child controller at this domain has to wait until the pABNO receives the acknowledge from the optical domain to process the following call request. Hence, the time experienced at the mmWave/WiFi domain is mostly due to the pace at which calls are sent to be processed rather than the time itself to setup the different switches present in the mmWave/WiFi domain. In relation to this, it can be added the fact that the calls are not ordered in time with respect to the involved domains. For instance, not all the calls directed to the same set of transport domains, (i.e., mmWave/WiFi domain and multi-layer optical domain with its calls from enodeB1 and BBU towards the EPC entities MME and SPGW) are processed consecutively and are mixed with other calls involving other domains (i.e., mmWave domain and multi-layer optical domain with its calls from enodeB2 to MME and SPGW), increasing the time to consider that the service is setup. This fact modifies the experienced setup time distribution at the different stages/domains, as we can see if we analyse the times experienced by the mmWave transport domain. In this domain, only two calls have to be established (from enodeB2 to MME and from enodeB2 to SPGW). However, in the time between those calls are received at the Wireless SDN Controller_2 child controller, the RMA has sent other calls to the Wireless SDN Controller_1 to establish the path between entities in the mmWave/WiFi domain and the optical transport domain.

With respect to the service deletion time in Table 19, the values present similar trends to the setup time, such as the fact that the values increase as we go up in the hierarchical control plane and delete requests for a pair of domains are not sent in a row by the RMA. In particular, the installed mobile network service is completely deleted in more than 90% of cases in less than 7.8 seconds, with a minimum value of 6.5 seconds and a mean value of 7.2 seconds. In this case, the values are lower with respect to the setup time due to multiple factors. First, there is no processing time devoted to path computation (RMA and optical child SDN controller). Second, the message passing between pABNO and RMA is lighter since the call DELETE operation requests are smaller than call POST requests. However, as before, when repeating the analysis of the measured values at the pABNO for serving the delete operation, we see that those involving the optical domain require more time, due to the set of actions done in its associated control entities and in the set of network elements hidden under the abstraction provided by the multi-layer optical SDN controller, as pointed out previously.

4.3.1.1.3.1.1.1 Control plane interface characterization

The following paragraphs present the control plane interface characterization of the overhead introduced by COP protocol between pABNO and RMA, as done in previous subsection *Multi-domain/multi-technology hierarchical XCI*, for the described service setup experiment.

Figure 87 represents the captured message I/O in terms of bandwidth through the COP interface during a part of the experiment. The blue profile corresponds to the messages exchanged from

pABNO to RMA and the red profile corresponds to the messages exchanged from RMA to pABNO.

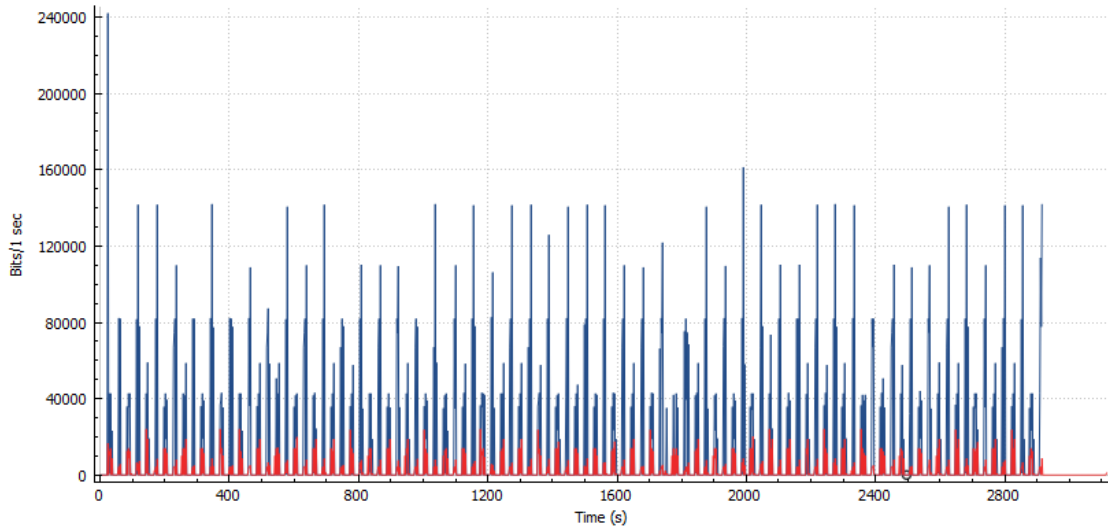


Figure 87. Instantaneous bit rate of control messages exchanged for service setup and deletion between RMA and pABNO (blue: pABNO-to-RMA, red: RMA-to-pABNO)

In Figure 87, we see different spikes in the communication between pABNO and RMA. These spikes correspond to the process of setting up and deleting the LTE mobile network service. The initial spike corresponds to the topology request response of pABNO to RMA. From this figure, we can observe that pABNO-to-RMA message exchange consumes more bandwidth than the RMA-pABNO communication. This is due to the fact that RMA makes requests and pABNO confirms these requests with all the information related to the actions done in each of the transport network domains. In terms of required bandwidth, the peak throughput in Figure 87 is around 240Kbits/s, hence the VPN tunnel connection should not represent a bottleneck from the control plane perspective. In fact, the traffic exchanged between RMA and pABNO is only generated when requesting the creation of a path, the network topology or to notify updates of topology or paths after a network event like a link failure. The following table includes the packet size distribution of the captured traffic for the whole experiment, confirming the trend observed in Figure 87 on the asymmetry in the communication between pABNO and RMA.

Table 20: Packet size distribution (in bytes) of the communication between pABNO and RMA

	RMA to pABNO					pABNO to RMA				
	Count	Average	Min val	Max val	Percent	Count	Average	Min val	Max val	Percent
Packet Lengths	12465	128,960	40	1408	100,000%	13613	744,939	40	1408	100,000%
0-19	0	-	-	-	0,000%	0	-	-	-	0,000%
20-39	0	-	-	-	0,000%	0	-	-	-	0,000%
40-79	10655	41,845	40	52	85,479%	5741	45,914	40	76	42,173%
80-159	201	94,000	94	94	1,613%	0	-	-	-	0,000%
160-319	808	214,260	206	255	6,482%	212	217,065	200	218	1,557%
320-639	0	-	-	-	0,000%	504	544,035	464	590	3,702%
640-1279	401	1059,294	877	1169	3,217%	1108	948,095	674	1130	8,139%
1280-2559	400	1362,000	1316	1408	3,209%	6048	1406,505	1363	1408	44,428%
2560-5119	0	-	-	-	0,000%	0	-	-	-	0,000%
5120 and greater	0	-	-	-	0,000%	0	-	-	-	0,000%

4.3.1.1.3.2 Local Service Recovery Time analysis

As mentioned before, a hierarchical control plane setup enables the possibility of having various levels of reaction to network events. In this subsection, we present a time analysis of the reaction times provided by the technology/domain-specific child controller to recover the established LTE mobile network service in the event of a link failure.

In the occasion of such event, the local recovery procedure is as follows: the software switch detects the link failure and informs the controller with the corresponding OF PORT_STATUS message [42]. The corresponding child SDN controller detects this message and sends an update topology message up the control hierarchy (which will eventually arrive to the application plane, RMA) to inform of the event and to inform that the recovery process will be done locally. Since the child SDN controller maintains a database with all the installed flows, based on the current “domain” network state it checks which flows have been affected and proceeds to calculate an alternative path. Once the child SDN controller has received confirmation that the alternative flows have been correctly installed in the software switch, the child SDN controller notifies the RMA through the pABNO of the alternate paths in the network. Thus, the RMA can update its flow database to have the current network vision and to provide the appropriate paths for subsequent service setup requests.

Additionally, the child SDN controller keeps records of the affected flows and its initial paths, so in the case of a link recovery, the child SDN controller will re-establish the initial paths (if possible) for those previously affected flows. Both the topological change and the changes in the flow paths are notified up the control plane hierarchy to maintain a correct/updated view of the network state.

In the following graphs, we can see the histogram and the CDF of performing one hundred operations of mobile network service local recovery in the mmWave/WiFi domain after a link failure and a link recovery event in case of using a wired control plane. The event occurs in the link between the transport node where the eNodeB and the RRU are attached and the transport node where the BBU is attached, which is a point of connection to the Multi-layer transport domain. In the case of the mentioned link events (failure/recovery), in order to re-establish the mobile network service, three COP calls (six unidirectional flows) have to be re-established: from eNodeB to MME, from eNodeB to SPGW and from RRU to BBU. All those flows have been setup through this link because it is a link using mmWave technology, and hence with higher performance than the other interface in this transport node, which is using IEEE 802.11ac technology.

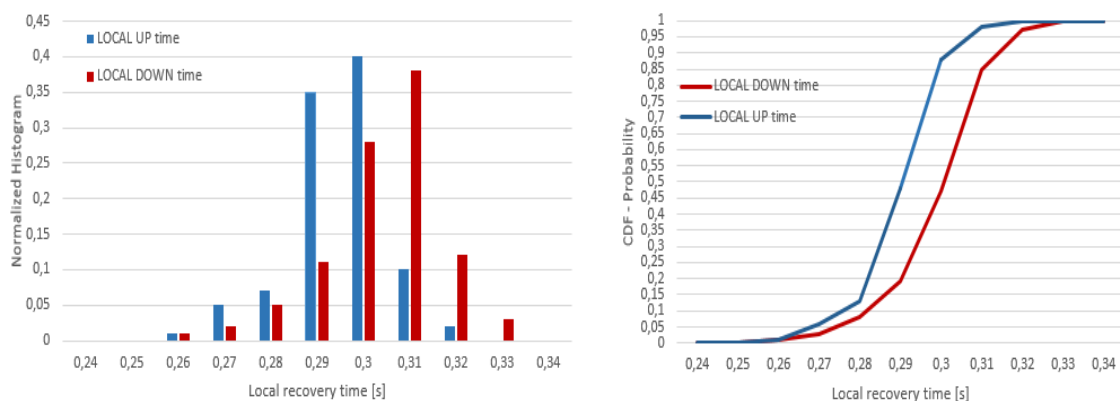


Figure 88. Local recovery histogram (left) and CDF (right) after a link failure/recovery (local down and local up times respectively) in the mmWave/WiFi domain

The values shown in Figure 88 cover the elapsed time between the software switch reports the link event until the child SDN controller has verified that all affected flows have been correctly re-established. In this case, the values labelled as LOCAL DOWN correspond to the recovery times after a link failure, and the values labelled as LOCAL UP correspond to the recovery times after this link becomes operative again, respectively. As mentioned before, after the local recovery process has finished, the child SDN controller notifies upper layers in the hierarchical control setup of the performed changes in the flow provisioning. Table 21 presents the statistical distribution of the local recovery operation performed at the Wireless SDN Controller_1.

Table 21: Local service recovery control plane delay [seconds]

Network Event	Average	Min.	25-percentile	75-percentile	Max.
Link failure	0.299	0.257	0.2957	0.306	0.329
Link recovery	0.289	0.250	0.284	0.296	0.319

As we can see in the Figure 88 and Table 21, the local recovery process after a link failure is slightly slower (around 10ms) compared to the recovery after a link recovery. This is mostly due to the different set of actions after the link event. In the case of a link failure, the three affected COP calls (6 flows) have to be deleted in two switches and installed in the four switches. However, in the case of link recovery, the process is the opposite, the rules have to be deleted in four switches and installed in the initial two switches. In this case, the operation of installing flow rules implies more time than deleting them, not only at the software switch but also at the control plane to transmit the OF FLOW_MOD rules. In addition to this, the processing at the Wireless SDN controller_1 changes. In the case of a link failure, the Wireless SDN Controller_1 checks all the installed COP calls. However, for the case of a link recovery, this checking is only done among the set of previously affected calls. In addition to this, we confirm what was stated in the previous subsection: most of the time it takes to the Wireless SDN Controller_1 to setup the mobile network service is due to the pace at which it receives the different call requests.

The reported recovery values for the link failure event are similar to those reported in [43] when using Open Flow Fast Failover (FF) group tables and using Bidirectional Forwarding Detection (BFD) protocol [44], even when repairing several flows. By using the BFD plus OpenFlow FF approach, a link down event can be detected and re-routing decisions can be triggered locally without the need to involve the controller (if backup paths have been configured in advance). Additionally, that paper does not consider the link recovery case, as in the Demo3 setup described in this section, for which the BFD plus OpenFlow Fast Failover is not a suitable solution. In fact, unlike in our procedure, the RMA and other control-plane entities (parent or child controllers) would not be aware of current network state, both from the topology perspective and the traffic distribution, since it is just handled by FF locally and not propagated. Hence, subsequent service requests would not be handled appropriately, i.e., orchestrating network resources to create a path where some link would not be currently available. In addition to this, the FF failover solution can potentially present more drawbacks, such as:

- a short BFD monitoring interval could lead to increased traffic and processing overhead or triggering false link transitions, which could affect the link performance for data transmission [46];
- the requirement of installing several backup rules (the denser the transport network deployment, the more are needed) and the required rules to perform crankback forwarding to send packets back towards a transport node which has an alternative active path towards the destination [45][44], potentially having large impact on the latency.

4.3.1.1.3.3 Centralized Service Recovery Time characterization

In this case, the recovery procedure is as follows: after the link failure event, the child SDN controller notifies up the hierarchy the change in the topology. The RMA receives the update in the topology and notices that no local recovery is going to be performed. The RMA detects which are the affected flows and sends back to the pABNO the corrective actions: first, deletes those calls with affected paths, and then, re-installs the affected calls. In the case of a link recovery event, the RMA only updates its topology view based on the received notification. At the path computation time, the RMA calculates the path that fulfils the requested conditions in terms of QoS (throughput and latency), otherwise the path is not valid. Hence, after a link failure event, the new calculated path must continue fulfilling these conditions and there is no need to revert to the previous path in case a link recovery event occurs for this same link.

In the following, we can see the histogram and the CDF of performing one hundred operations of centralized recovery at the RMA after a link failure in the same link of the previous section and when using a wired control plane in the mmWave/WiFi domain. The measured time interval starts when the software switch reports the link event and ends when the RMA has verified that all affected flows have been correctly re-established.

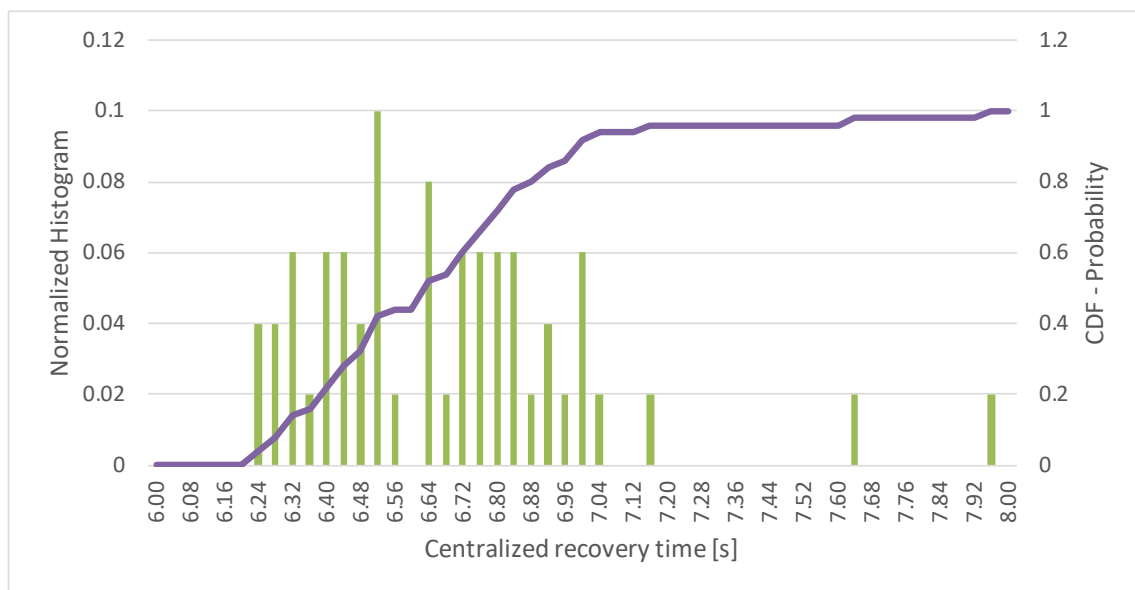


Figure 89. Normalized Histogram and CDF of centralized recovery operation seen by the RMA

Table 22: Centralized service recovery control plane delay [seconds]

Network Event	Average	Min.	25-percentile	75-percentile	Max.
Link failure	6.652	6.010	6.434	6.821	7.957

As observed, the measured time increases to the order of seconds compared to the local recovery case presented in the previous section and this is due to several factors. First, climbing up and down in the hierarchy introduces delay, which is increased also due to the VPN connections. Second, the processing time at the RMA to update the topology and check the affected rules will be higher than in lower control level entities. Due to its position up in the hierarchy, RMA will potentially have to check the integrity of more calls than just the checking process that a child

controller should do. Third, and most relevant, the corrective actions the RMA orders. Due to its global view, the corrective actions provided by the RMA involve deleting and creating actions in multiple domains; in particular, the mmWave/WiFi and the multi-layer optical domain. In the explained case of local recovery, actions only involve the mmWave/WiFi domain. As pointed out in the previous *Service Setup Time Analysis* subsection, actions performed in the multi-layer optical transport domain (both for creating and deleting) may potentially require more time due to the bigger amount of operations at its associated control entities and the underlying involved network elements. Nonetheless, attending to the average values presented in Table 18 and Table 19, the centralized recovery strategy is on average faster than deleting and setting up the whole service again (17.66 seconds versus 6.754 seconds) and keeps the availability of the service in other parts of the transport network which are not affected by the link failure, for instance, the other mmWave transport domain.

4.3.1.2 Data Plane performance assessment

4.3.1.2.1 Multi-domain/multi-technology hierarchical XCI

In this subsection, we conducted an assessment of the multi-domain data plane performance in the setup presented in Figure 68, when only counting with the pABNO and the child SDN controllers associated with the mmWave/WiFi and the multi-layer optical transport network domains, without any VPN tunnel connection in the setup. In particular, we injected a UDP traffic (packet length 1350 bytes) flow originated from an endpoint (linux namespace) attached to a transport node at the edge of the wireless domain (composed of one wireless hop) towards an endpoint (linux namespace) connected to the other edge of the optical domain. Table 23 below reports the statistics of the measured UDP throughput, jitter, and packet loss for twenty iperf sessions of ten seconds each.

Table 23: Multi-domain/multi-technology hierarchical XCI data plane performance

	Average	Min.	25-percentile	75-percentile	Max.
Throughput (Mbit/s)	335	335	335	335	335
Jitter (μ sec)	51.2	43	45.7	54.8	65
Packet Loss (%)	0.0377	0	0.011	0.055	0.12
RTT (ms)	3.26	2.98	3.12	3.23	5.46

The most notable observation in these tests is the constant throughput exhibited of 335Mbps. This value is bigger than the reported value in Experiment #13: Assessment of the SDN-based control and data plane for the mmWave/Wi-Fi mesh domain (section 4.2.2.2) for one hop, because in this case the wireless hop consisted of a mmWave link which has higher throughput than the WiFi IEEE 802.11ac one presented in Experiment #13: Assessment of the SDN-based control and data plane for the mmWave/Wi-Fi mesh domain, whose focus was on measuring data plane performance versus the number of hops in the wireless transport network.

Beside the use of the mmWave link, several optimizations were done in the setup to improve the data plane performance. In particular, some OVS software switches were running in user space due to MPLS label stacking process at the multi-layer optical transport network domain. In order to run OVS instances in kernel space, we changed the setup to work with only a single MPLS label, without stacking, so all OVS operations were done at the kernel space. Another optimization performed at the software switch level was the deactivation of CRC checksums at the interface level.

In addition to this, it is worth mentioning that potential better performance could have been achieved if increasing the maximum available transmission unit (MTU). As the setup is mixing several types of technologies in the mmWave/WiFi transport domain, in this case, the maximum MTU value (1500 bytes) of IEEE 802.11ac interfaces is limiting the performance of the mmWave/WiFi domain because OVS is configured to work with the most restricting value. However, the mmWave interface allows working with bigger values of MTU (7192 bytes). A similar conclusion is stated in [41], where an analysis of the performance of different modulation and coding schemes of an IEEE 802.11ad link as a function of the used MTU is presented.

In addition to throughput, we also gathered jitter and packet loss metrics. As mentioned previously, the maximum achievable throughput performance was determined when achieving a packet loss rate lower than 1%. While performing the measurements, we noticed stability and low variance of both values during repetitions. Finally, we also measured RTT with one hundred samples in one hundred seconds for the tested setup. The average measured value is 3.26ms with a minimum value of 2.98ms, a maximum value of 5.46ms and a standard deviation of 0.539ms.

4.3.1.2.2 Multi-site XCI transport-level performance

In this subsection, we conducted the data plane performance assessment of the whole multi-domain multi-site setup presented in Figure 68, including both mmWave network domains and the multi-layer optical transport network domain. It is worth mentioning that the mmWave transport network and its associated Wireless SDN Controller_2 are connected to the other domains by its corresponding control/data plane VPN tunnels.

To perform this assessment, we injected a UDP traffic flow (packet length 1350 bytes) originated from an endpoint (linux namespace) attached to a transport node at the edge of the mmWave/WiFi domain towards an endpoint (linux namespace) connected to the edge of the other mmWave domain, hence traversing the transport network end-to-end. The table below reports the statistics about the measurements of the attained UDP throughput, jitter, and packet loss for twenty iperf sessions of ten seconds each. The aim of this evaluation is to quantify the impact of the dedicated data plane VPN tunnel to connect the optical domain with one of the mmWave domains.

Table 24: Multi-site XCI transport-level performance

	Average	Min.	25-percentile	75-percentile	Max.
Throughput (Mbit/s)	153.31	152	153	154	154
Jitter (μ sec)	49.63	25	39.25	57.75	80
Packet Loss (%)	0.664	0.260	0.5175	0.795	1.5
RTT (ms)	56.601	52.6	53.1	56.75	74.2

With respect to the results presented in the previous section, the main observed metric that has been affected is the throughput, which has been reduced by a factor of two. Interestingly, jitter values present similar values with respect to the previous case, indicating that the VPN connection does not introduce additional time variations when packets are delivered at destination. With respect to RTT values, the average measured value over 100 samples is 56.601ms with a minimum value of 52.6ms, a maximum value of 74.2ms and a standard deviation of 8.7ms. Based on these values and the previous ones, we can state that the VPN tunnel connection between Barcelona site and London introduces an additional RTT of around 53ms. This value has also been confirmed when doing a ping between the SGPW VM placed at the edge of the multi-layer optical

transport network domain and the eNodeB VM placed at the edge of the mmWave domain at the other side of the VPN tunnel, that is, when only traversing the VPN connection.

4.3.1.2.3 Multi-site XCI service performance

Finally, in this subsection we present the data plane performance perceived by the VM's emulating UEs using the deployed LTE mobile network service under two situations: a) when UE_2 in Figure 68 is attached to a VM working as an RRU featuring PDCP/RLC split placed at one edge of the transport network and UE_3 in Figure 68 is attached to the VM working as enodeB2 placed at the other edge of the transport network, and b) when UE_1 in Figure 68 is attached to the VM working as enodeB1 at one edge of the transport network and UE_3 in Figure 68 is attached to the other VM working as enodeB2 deployed at the other edge of the transport network.

Tables below report the statistics of the measurements for UDP throughput, jitter, and packet loss for twenty iperf sessions of ten seconds each. The aim of this evaluation is to check the end-to-end service performance for final users of the virtualized deployed LTE mobile service. With respect to the results of the previous section, we want to measure how the EPC/LTE traffic encapsulation and the performance of the experimental virtualized deployment affect their perceived performance with respect to the one the transport network is able to offer.

Table 25: Multi-site service performance between users for case a)

	Average	Min.	25-percentile	75-percentile	Max.
Throughput (Mbit/s)	116.76	116	116.75	117	117
Jitter (μ sec)	187.85	53	104	186.75	971
Packet Loss (%)	0.21	0.064	0.1075	0.21	0.73

Table 26: Multi-site service performance between users for case b)

	Average	Min.	25-percentile	75-percentile	Max.
Throughput (Mbit/s)	140.72	138	141	141	142
Jitter (μ sec)	196	52	66	228	896
Packet Loss (%)	0.27	0.071	0.07	0.4	0.9

According to the values in Table 25 and Table 26, there is a performance degradation with respect to the values reported in Table 24 of previous section. This degradation can be explained by the introduction of an environment of VMs virtualizing the mobile service and the overhead of the GPRS Tunnelling Protocol (GTP) involved in EPC/LTE communications.

When comparing the results in Table 25 and Table 26, we see that the case a) shows worse performance than case b) because this communication involves more entities/hops in the communication chain, since UE_2 reaches the EPC through a fronthaul featuring PDCP/RLC split between the RRU and the BBU and UE_1 reaches the EPC directly from the enodeB1.

Finally, Table 27 presents RTT values obtained with 100 samples for the presented cases a) and b). As it can be seen, these values are in line with the ones reported in the previous section and they are mainly due to the VPN connection between Barcelona site and London. In particular,

case b) (enodeB1-UE1) presents slightly lower values than case a) (RRH-UE2). This can be explained again by the way both UEs arrive at the EPC. As commented before, case a) experiences an extra processing at the BBU.

Table 27: Multi-site service RTT performance [milliseconds]

	Min.	Avg.	Max	Std Dev
Case a)	54.55	57.28	76.45	4.14
Case b)	54.04	56.431	70.21	3.62

5 Experimental evaluation results for crosshaul fulfilment of RAN split requirements (Demo 4)

This demo evaluates the fulfilment of RAN split requirements. In 3GPP, new RAN splits are discussed [49] (see also [25] Table 5) as well as a shift towards packet-based fronthaul. The demo evaluates transport technologies to be suitable and fulfil the requirements for fronthaul, backhaul and new RAN split points. Depending on the RAN split the requirements differ in data rate, latency and jitter. Therefore, not all combinations of transport technologies can fulfil all RAN splits. The split options range between option 8 – split PHY – and option 1 – RLC-RRC. In this way, it is clearer for operators how to build the integrated fronthaul and backhaul at the data plane level to comply with given transport requirements.

Overall, multiple technological categories are evaluated by transport technologies (wired, wireless) and structural definition (packed-based, circuit-switched). Thus, combining these dimensions covers a broad set of new functional splits in addition to legacy interfaces, such as CPRI. To experimentally evaluate this technological variety, we have integrated both backhaul and fronthaul traffic support over the same integrated infrastructure combining wavelength-selected wavelength division multiplexing passive optical network (WS-WDM-PON) that multiplexes traffic coming from 5G-Crosshaul Packet Forwarding Elements (XPFEs) and Radio-over-Fibre (RoF) technology. Data rates and latency measurements of WS-WDM-PON guarantee up to RAN split option 6 (symmetric 10Gbps per point-to-point dense WDM link). The data rates and latencies requirements up to RAN split option 6 are also supported in combination with an XPFE. In addition, Error Vector Magnitude (EVM) results with/without RoF integration are included, which are always below 2.5%. Therefore, it was experimentally validated that WS-WDM-PON+XPFE and RoF can coexist without any performance degradation. To support lower-layer splits in an efficient manner, innovative techniques were also evaluated. In fact, the mixed digital/analogue radio over fibre implementation allows sending up to 11.05 Gb/s (9 x CPRI 2) CPRI-equivalent bit-rate using less than 200 MHz bandwidth of an off-the-shelf optical transponder, which represents a remarkable spectral efficiency improvement compared to conventional CPRI. The system also allows reaching 36 km distance while still complying with the 3GPP EVM requirements.

We have integrated several transport technologies to a network transporting simultaneously backhaul and fronthaul traffic for both upper (option 2) and lower layer functional splits (options 6 and 8). They include mmWave, 5G-Crosshaul Circuit Switching Element (XCSE), and XPFE. In this setup, we demonstrated that the XCSE guarantees the required rates for all flows, independently of the functional split used (including the most demanding option 8 split). The compressed and packetized fronthaul traffic is also tested and transported over the XCSE, which generates fronthaul traffic of 262.4 Mbps. We also observed that the Fast Forward mmWave together with the XCSE can satisfy the one-way delay constraints imposed by the MAC-PHY split (the average RTT is 0.26 ms). This is also the case with high-load UDP traffic, where the RTT average value is 0.44 ms. The maximum RTT obtained with background traffic suggests that constraints are also satisfied if we take as reference the documents of the small cell forum, because the average of the maximum results is 1.27 ms, with a maximum of 1.5 ms. Other more stringent implementations and recommendations may not be fulfilled in all cases. In that case, prioritization of the fronthaul traffic over the backhaul traffic would reduce its impact, and so, lower layer splits would also be supported, as explained above for the WS-WDM-PON+XPFE setup.

Other functional tests done between the UE all the way through the mobile network (and the underlying 5G-Crosshaul transport) and up to an Iperf server resulted in the following measurements. The rates of the higher layer split under evaluation (option 2) is 18.28 Mbps in the downlink and 25.73 Mbps in the uplink; in other lower layer splits under evaluation (option 6),

the rate is 67.65 Mbps in the downlink and 17.46 Mbps in the uplink; for the backhaul traffic, the resulting rate is 8.71 Mbps in the downlink and 10.37 Mbps in the uplink. The user equipment (UE) connected to the option 8 split remote radio unit (RRU) results in 45.504 Mbps in the downlink. These rates are achieved independently from whether there is simultaneous traffic of other UEs in the setup and show that the different functional splits are actually supported.

In a more focused setup, we demonstrated for a mmWave link, and optical link, and their combination to a hybrid link the feasibility to support higher-layer functional splits. The average latency values, depending on link technology, of 0.5 ms to 1.5 ms are not suitable for lower layer splits, i.e. split options 5 to 8. These splits are within the hybrid automated repeat request (HARQ) loop and require shorter latencies. The split of the RLC, i.e. split option 4, is possible from a latency perspective, but requires higher bandwidth. Although the hybrid link provides more bandwidth than the individual ones, this is traded off by a latency increase. Therefore, these technologies can be used for splits between RLC and PDCP and for BH traffic only, i.e. split options 1 to 3 are supported.

In summary, the results confirm that the variety of technologies combined with their specific characteristics allow serving the needs of all RAN split options and contribute to fulfil the project and 5GPPP KPIs and objectives. More specifically, they contribute to increase the number of devices handled by the network, to reduce its CAPEX and OPEX, to find the appropriate settings in the quest to reduce latencies below 1ms and to increase the global throughput of the network.

5.1 Demo 4 setup

The Demo 4 setup contains the following transport technologies: mmWave, XPFE, Wavelength Selected Wavelength Division Multiplexing Passive Optical Network (WS-WDM PON), Analog / Digital Radio over Fibre (A/D-RoF), 5G-Crosshaul circuit switching element (XCSE) and Optical Wireless Communication (OWC). They can support possible new RAN splits, as stated in D5.1.

We focus on measuring low-level KPIs on the data-plane for evaluation and only use the control plane where needed for operation. The measurement device is either explicitly stated in the experiment description or an outside entity connected to the system under test. We aim to verify the combinations of different transport network technologies as solutions for data plane connectivity between conventional RRU and BBU as well as new functional split points. Therefore, we evaluate the technical KPIs in all the experiments and define them as follows:

- Throughput
 - The achieved data rate through the system-under-test (SUT). This is an end-to-end assessment throughout the combined system in the experiments. The traffic will be generated outside with State of the art (SoTA) traffic generators.
- Packet Loss Rate
 - The Rate of packet loss during the testing period. The traffic for packet loss rate assessment will be outside the SUT and understood as an end to end measurement for the SUT.
- Latency
 - End to end latency throughout the combined transport technologies.
- Jitter
 - The deviation of latency measurements over time. Statistical data of latency measurements have to be collected to assess the jitter derivation.

- EVM
 - Error Vector Magnitude of constellation points in the physical layer of the radio access signal. This will be used as complement for the other KPIs, where appropriate (e.g., RoF).

The experiments in this section combine copper packet-based with optical packet-based transport technologies as well as packet-based wired with wireless solutions.

For evaluation, data plane traffic will be injected into the systems either by SOTA traffic generators and/or new prototype devices generating traffic at specific split points. This enables Demo 4 to evaluate the performance of the technologies in a more practical and more real environment.

The setup of demo 4 comprises multiple sites, creating a realistic environment in the testbed. This section defines further the building blocks from other work packages. The rest of sections are devoted to the description of the experiments integrating multiple data plane technologies and discussed on the results obtained.

5.1.1 Building blocks from other work packages

Data plane building blocks

WP2	
Component / Entity	Description
XCSE multi-layer switch	Used for circuit-switched traffic and TDM frames
XPFE switch	OpenFlow software switches based on extended version of Lagopus
mmWave mesh nodes	IEEE 802.11ad-based mmWave mesh solution – Used to connect radio access nodes (e.g., small cell and RRU). Required for testing over short distances and/or stability testing, and installation in testbed. Commercial EdgeLink nodes are based on Freescale 1043A boards that host a 64-bit ARM-based quad-core processor for embedded networking with a fanless design. Such boards provide two 10 Gbit/s Ethernet ports supporting PoE (Power-over-Ethernet) and three USB3.0 ports used to connect the mmWave IEEE 802.11ad wireless card and antennas.
mmWave mesh network Gateway node	IEEE 802.11ad-based mmWave mesh solution – Used to connect transport network Crosshaul Network and Core. Required for testing over short distances and/or stability testing, and installation in testbed
WS-WDM-PON	Transparent WDM transport solution compatible with SDN-based control plane implementation – Used to connect core and aggregation Crosshaul network with access Crosshaul devices and client premises (RRHs, small cells, distribution point units, optical terminations...)
WS-WDM-OLT	Equipment based on a L2 Ethernet OpenvSwitch (NXP LS2088ARDB evaluation board) and the corresponding optical modules needed to adapt signals from metro network to the WDM-PON.

WS-WDM-ONUs	Tuneable 10G WDM clients – Optical WDM termination that represents an optical distribution point unit for allocating the available resources among Crosshaul devices and client premises
-------------	--

Control and application plane building blocks

WP3	
Component / Entity	Description
SDN controller	OpenDaylight instances controlling the XPFE switches and the mmWave mesh For the WS-WDM-PON system a Ryu framework-based controller to control and manage forwarding and physical issues
Debug/Test controller	A set of Linux scripts responsible for testing the mmWave link status, traffic injection and mmWave link performance measurement, i.e., latency and throughput.

5.1.2 Additional building blocks

In addition to the previous elements, the following building blocks are required to support Demo 4.

Component/Entity	Description
Access Nodes	A server running an instance of eNodeB emulator (for backhaul traffic) or two or more servers with RRU-BBU (for different fronthaul traffic splits such as PDCP-RLC, MAC-PHY) prototypes (see D5.1, Section 9.3) and a commercial LTE small cell.
Network Core Node	Node running a software instance of a complete 3GPP standard compliant Evolved Packet Core providing mobile network. This demo uses OpenEPC.
Mobile User	Entity connected to the access node, either the eNodeB or the RRU entity, emulating an end user and commercial LTE USB dongle connected to the LTE small cell.
Optical power meter	To validate the information provided by the WDM SDN Controller regarding physical events, such as optical power drops
Portable Optical Spectrum Analyser	To validate and to demonstrate the performance of the WDM SDN Controller in terms of monitoring and management of the elements that form the PON (laser tuning, dynamic channel allocation, quality link monitoring...)

Traffic generator	A server running software (moonGen, iperf) to generate a traffic load and test traffic and to measure received traffic and latency of test traffic.
-------------------	---

5.2 Experiments on 5G-Crosshaul traffic over integrated data plane technologies

5.2.1 Experiment #15: Backhaul and Fronthaul services integration through an SDN WS-WDM-PON transport network and XPFEs + Radio-over-Fibre

Backhaul and Fronthaul services integration through an SDN WS-WDM-PON transport network and XPFEs + Radio-over-Fibre	
Description	<p>This experiment demonstrates both BH and FH traffic support over a network composed of a WS-WDM-PON technology and of XPFEs, validating the proposed solution for FH services according to different functional split options for C-RAN architectures.</p> <p>The physical infrastructure in this network is a passive optical network (PON). It connects a head-end-unit (HEU) and a remote antenna unit (RAU) as well as RUs and DUs. Traffic generators take the role of the RU and DU in this setup.</p> <p>The HEU and RAU exchange radio over fibre signals with CWDM on the optical network. The RoF system contains two control signals carried over 1350 and 1370nm wavelength, whilst radio signals are carried over 1410, 1430, 1450, and 1470nm. The control and signal wavelengths are combined at the HEU and after the fibre network divided again at the RAU.</p> <p>The RUs and DUs, respectively the corresponding traffic generators, exchange packetized FH and BH traffic over the same optical network, using WS-WDM-PON technology. The WS-WDM-PON system uses different wavelengths, it operates at C-Band.</p> <p>An XPFE is used to connect the traffic generators to the PON OLT. An SDN controller implements flow deployment to and operation of the WS-WDM-PON system.</p> <p>This experiment demonstrates both BH and FH traffic support over a network composed of a WS-WDM-PON technology and of XPFEs, validating the proposed solution for FH services according to different functional split options for C-RAN architectures. The coexistence of both WDM technologies must guarantee a certain signal quality and attenuation over the same fibre for operating correctly as expected, i.e., correct WDM filtering at Central Office and at client premises must be done to avoid signal degradation.</p>
State of the art	<p>10G tuneable WS-WDM-PON systems is a technology under development, non-commercial products are ready at the moment. Deploying this solution in existing legacy PONs allows telco operators to upgrade their access infrastructure to reach 5G requirements. CPRI-like traffic, being a TDM bitstream, is not compatible with the proposed packet-based WS-WDM-PON system. Up to now coexistence scenarios based on combining RoF and PONs have not been put into practice.</p>

<p>Improvement from State of the art</p>	<p>As a first improvement, packetized fronthaul traffic can be carried across the network, showing that even low-layer functional splits can be supported. Thus, backhaul and fronthaul services converge over the same fibre infrastructure. In addition, through BBU virtualization and optimal placement it is possible to configure different split options to guarantee FH requirements. As a second improvement, the proposed WDM solution allows automated control by an SDN controller. The solution allows to separate different traffic flows in an OpenFlow compatible way in each of XPFE, OLT, and ONUs. As a third improvement, the long-distance fibre infrastructure can be used by both the RoF and the PON technologies. This could increase the coverage and infrastructure efficiency using the same fibre link while maintaining the same link quality.</p>
<p>5G-Crosshaul Use Cases</p>	<p>Dense Urban Society High-speed Train</p>
<p>System under Test</p>	<p>The diagram illustrates a network architecture for testing 5G crosshaul. It starts with a Traffic Generator (BH) connected to XPFE (Lagopus) via UNI. From XPFE, traffic flows through W-OLT and Cex to W-ONU2 and W-ONU#1. W-ONU2 is connected to TrafficGen (FH, sink) via UNI and FH over Eth. W-ONU#1 is connected to TrafficGen (BH) via UNI. The network also includes RRH, RoF (HEU), RoF (RAU), and VSA/ANT. A WS-WDM-PON controller is connected to the W-ONU components. Traffic flows are color-coded: BH Traffic (green) and FH Traffic (blue). Encapsulation/Decapsulation points for BH/FH to XCF are marked in red.</p>
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Number of devices increase by factor 10 Obj. 3: CAPEX and OPEX reductions Obj. 4: Latency below 1ms Obj. 6: Increase in throughput of 20%</p>
<p>Measured KPIs</p>	<p>Throughput Data rate Packet Loss Rate Latency Jitter EVM</p>
<p>Measurement tools</p>	<p>Packet based FH and BH: FH/BH traffic generator and tester (servers with moonGen) PON-RoF Integration: Ettus X310 with external amplifier to generate LTE signals Rohde & Schwarz FSW13 with LTE Option (analyse LTE signals)</p>
<p>Measurement procedure</p>	<ol style="list-style-type: none"> 1. Deploy the WS-WDM-PON 2. Connect FH traffic source and sink to the network 3. Connect BH traffic sources and sinks to the network 4. Interconnect the WS-WDM-OLT to the metro network via XPFE 5. Configure XPFE and ONU to provide XCF encapsulation 6. Configure a certain functional split option for FH traffic corresponding to an RU

	<ol style="list-style-type: none"> 7. Configure BH traffic 8. Start traffic transmission 9. Validate BH support 10. Evaluate Packet Loss Rate for both BH and FH links 11. Evaluate throughput and latency for both BH and FH links 12. Check whether requirements for 5G functional splits are satisfied 13. Create source signal, which represent BBU signal, with different modulation scheme such as 64QAM. 14. Pass the signal through RoF system only and measure EVM values 15. Pass the signal through the integrated RoF/PON and measure EVM values 16. Compare EVM results from last two steps, to evaluate the signal degradation due technologies coexistence
Constraints	<p>The current state of the technology presents certain limitations, tuneable 10G optics could be considered as an immature technology (still in phase of constant evolution and improvements),</p> <ul style="list-style-type: none"> - Side-Mode-Suppression-Ratio (SMSR) affects directly to both optical launch ad sensitivity power, implying low optical budgets - Tuning speed (Of the order of seconds) - Frequency accuracy ($\pm 2.5\text{GHz}$) - Extinction ratio - Insertion losses (tuneable filter in ONUs) about 3dB - Noise figures and gain ranges for optical amplifiers (up to 1:64 PON topologies can be achieved with the current amplification range) - Rx optical filters (WS-WDM-ONUs) calibration to avoid wavelength deviations <p>Control plane for managing WDM links is needed to remove crosstalk issues</p> <p>In RoF integration, the following limitations exist:</p> <p>In general, the differences between the technologies in terms of their optical budgets (RoF is much lower than WDM), which imply an important constraint in design and deployment phases. It's necessary to assure the correct optical power levels at both systems</p> <p>The RoF system is designed for specific RF power level. The used RRH cannot support such power-levels without degrading the signal so a lower power level was chosen to support operation. Additional RF amplifiers are needed for high power output. PON split ratio involves an important attenuation contribution that can reduce the quality of the received signal, affecting directly the performance of RoF system. 1:32 split ratio could be too high for RoF purposes</p> <p>WDM filtering supposes an essential point during the performance of the experiment due to the need of removing interferences between the different technologies. However, current filters are far from the ideal, then, signal degradation could appear</p>
Main results	<p>Data rates and latency measurements of WS-WDM-PON guarantee up to RAN split option 6 (symmetric 10Gbps per PtP DWDM link), the data rates and latencies requirements up to RAN split option 6 are also supported in</p>

	<p>combination with an XPFE, 7.5 Gbps of traffic with a latency 99th percentile of about 180µs are supported in combination with BH traffic.</p> <p>Validation of PONs coexistence to allow Telco operators to deploy different PON technologies over the same fibre infrastructure</p> <p>Validation of SDN controller performance</p> <p>In addition, EVM results with/without RoF integration are included. Besides, it was proven that WS-WDM-PON+XPFE and RoF can coexists without any performance degradation. EVM values are always below 2.5% for the evaluated setups</p>
--	---

Additional details of the PON setup:

The measurements described below have been performed over a power-splitter based PON, the most extended topology in access optical networks. The initial setup considers only the integration of XPFEs and the PON network, deploying a 1:32 PON split ratio which means a fibre infrastructure with 32 optical terminations for customer premises. Here, only two of those optical lines are used to connect both FH and BH devices to the PON through the WDM ONUs. The proposed WS WDM PON systems operates in C-Band: a total of 40 channels (100GHz grid) where 20 of them are used for downlinks and the remaining 20 for uplinks. Hence, the colourless ONUs (tuneable transmitters) can be tuned to any of the following downstream C-Band channels.

Table 28: Wavelength plan

WS-WDM-PON					
Upstream			Downstream		
λ_o [nm]	f_o [THz]	# Channel	λ_o [nm]	f_o [THz]	# Channel
1532,68	195,6	1	1548,51	193,6	21
1533,47	195,5	2	1549,32	193,5	22
1534,25	195,4	3	1550,12	193,4	23
1535,04	195,3	4	1550,92	193,3	24
1535,82	195,2	5	1551,72	193,2	25
1536,61	195,1	6	1552,52	193,1	26
1537,4	195	7	1553,33	193	27
1538,19	194,9	8	1554,13	192,9	28
1538,98	194,8	9	1554,94	192,8	29
1539,77	194,7	10	1555,75	192,7	30
1540,56	194,6	11	1556,55	192,6	31
1541,35	194,5	12	1557,36	192,5	32
1542,14	194,4	13	1558,17	192,4	33
1542,94	194,3	14	1558,98	192,3	34
1543,73	194,2	15	1559,79	192,2	35
1544,53	194,1	16	1560,61	192,1	36
1545,32	194	17	1561,42	192	37
1546,12	193,9	18	1562,23	191,9	38
1546,92	193,8	19	1563,05	191,8	39
1547,72	193,7	20	1563,86	191,7	40

For simplicity, there are four different optical interfaces mounted at the WDM OLT:

- Crosshaul interface. Multimode 10GbE optical link to interconnect the crosshaul network to the access PON via XPFEs
- PON interfaces. 3x different C-Band ptp DWDM links (10GbE)
 - o Backhaul path
 - o Fronthaul path
 - o Free link (for tuning purposes)

To achieve a complete colourless solution, ONUs also require a tuneable reception stage which is implemented by the integration of tuneable 100 GHz grid optical filters. Each filter needs to be correctly calibrated before ONU start-up and must be controlled in real-time to remove frequency deviations due to temperature changes. In this way, the SDN system performs a continuously link quality monitoring via in-band. Furthermore, in the case of the Backhaul path, a 10G/1G Ethernet switch converter is mounted at the LAN interface of the corresponding ONU for the connectivity with BH traffic generator.

We evaluated latency and jitter for the FH and the BH path with both low and high throughputs as well as with separate FH and BH and simultaneous FH and BH traffic flows. As FH traffic, we have used unidirectional traffic of 7.5 Gbps with a packet size of 1284 B. 7.5 Gbps correspond to the CPRI traffic of three LTE 20 MHz 2x2MIMO cells. The FH traffic was carried over 10G links. As BH traffic we have used bidirectional traffic of 150 Mbps per direction with a packet size of 512 B. 150 Mbps correspond to the peak throughput of a single LTE 20 MHz 2x2 MIMO cell, we have used medium sized packets as a (coarse) approximation of traffic profiles in the field. BH traffic was carried over 10G links over the PON system, but 1G links have been used at the traffic generator.

As low throughput, we have used 100 Mbps for both FH and BH and as high throughput we have used the targeted bitrates.

In the setup 16.5 km of fibre have been deployed between ONU and OLT, which contributes approximately 80.85 μ s to the latency (0.0049 μ s/m @1550 nm). The rest of the latency is due to the XPFE, ONU, and OLT as well as the serialization delay on the links. For a measured latency of 100 μ s, 19.15 μ s are contributed by XPFE, ONU, OLT, and link serializations.

The RoF system was deployed over the same fibre infrastructure. To achieve compatibility between the different technologies several modifications were needed to adapt the original setup to the new specifications defined by the RoF system. The main change in the setup entails the introduction of a WDM filter as an element of coexistence between the two solutions for combining both optical signals over the deployed PON. This optical filter has been designed to works as follows,

- Port A. Pass band filter at (1520-1600 nm)
- Port B. Pass band filter at (1300-1500 nm)
- Port C. Combines Port A and Port B signals

The deployment of the coexistence filter only involves a low additional attenuation (~0.5 dB) which can be offset thanks to the variable optical amplification stage that is integrated at the WDM OLT. The second main modification of the original setup is the need to reduce the PON split ratio from 1:32 (15dB) to 1:4 (6dB) due to the optical budget of the RoF system (lower than WDM ones). In this way, to avoid possible power problems with optical transceivers within the WDM system (optical saturation from -8 dBm), additional attenuation is included before ONUs. On the RoF side of this experiment, the CWDM is flexible and can be modified based on the demands of the fibre wavelength with minor modification on the laser diode parts. Also, it is advisable to have the input around 20 dBm for the best EVM results, i.e. minimum signal distortion.

Discussion of results

Packet processing:

Each measurement was executed three times with a duration of 60 seconds per run. We report average latency, standard deviation, and 99th percentile for each measurement. The latency at low throughput is shown in Table 29.

Table 29: Latency at low throughput

Throughput	#	Latency [μ s]	StdDev[μ s]	99% [μ s]
FH, 100Mbps, unidirectional, 1284 B	1	99.758	2.391	101.206
	2	99.687	2.168	101.088
	3	99.663	2.324	101.027
BH, 100Mbps, bidirectional, 512 B	1, DL	102.835	3.178	104.184
	1, UL	103.282	2.908	103.912
	2, DL	102.842	3.152	104.168
	2, UL	103.311	2.662	104.064
	3, DL	102.795	2.680	104.160
	3, UL	103.310	3.038	104.160

At target bitrates the latency increases slightly for FH, about 1 μ s, or keeps approximately the same for BH traffic, when FH and BH are measured separately, see Table 3030.

Table 30: Latency at target throughput, separate FH and BH traffic

Throughput	#	Latency [μ s]	StdDev[μ s]	99% [μ s]
FH, 7500Mbps, unidirectional, 1284 B	1	100.668	6.837	102.428
	2	100.738	9.852	102.646
	3	100.819	8.185	102.563
BH, 150Mbps, bidirectional, 512 B	1, DL	102.757	2.382	104.136
	1, UL	103.298	3.474	104.152
	2, DL	102.786	2.871	104.136
	2, UL	103.248	2.599	104.120
	3, DL	102.742	2.206	104.096
	3, UL	103.312	3.318	104.216

When sending FH and BH traffic simultaneously, the latency for throughput for both FH and BH traffic increases, see Table 3131. The latency increase for FH is about 1.9 μ s, whereas for BH the increase is 5.6 μ s (DL) and 3.4 μ s (UL). The standard deviation for latency increases also much more BH traffic than for FH traffic, showing that the internal prioritization of the XPFE packet processing is working. On the downside, the 99th percentile of latency values increases to about 180 μ s for both FH and BH traffic. Correspondingly large buffers to compensate jitter would be

required as dropping packets due to too large latency in the order of 1% of packets is not possible, neither for FH nor for BH traffic.

Table 31: Latency at target throughput, simultaneous FH and BH traffic

Throughput	#	Latency [μ s]	StdDev [μ s]	99% [μ s]
FH, 7500Mbps, unidirectional, 1284 B	1	102.626	12.698	186.924
	2	102.662	12.787	187.437
	3	102.565	12.563	185.760
BH, 150Mbps, bidirectional, 512 B	1, DL	107.600	26.704	176.328
	1, UL	106.405	11.664	179.720
	2, DL	108.949	37.006	188.656
	2, UL	106.518	12.036	180.040
	3, DL	108.587	34.468	181.472
	3, UL	106.481	12.077	180.232

Based on the measurements, it is possible to support both FH and BH traffic on the same link in this combination of XPFE and PON. But due to the high 99th percentile values for latency, using this setup for split options 7 (High-Phy – Low-Phy) and 8 (Phy – RF) is critical. These options require a latency below 250 μ s, see [25] Table 5. The measured 99th percentile is well below this requirement, but the observed jitter requires large buffers for compensation which are not expected to be practical. Split options 6 (MAC-PHY) and higher in the protocol stack can be supported without problems.

PON-RoF Integration:

For integrating the RoF system into the PON system, passive optical filters are deployed for multiplexing and de-multiplexing the analogue RoF signal from the WDM system as seen in Figure 84.

Basically, the EVM results are below 2.5% for up to 16 km fibre length with different modulation schemes used in the setup. An exemplary screenshot from the vector spectrum analyser (VSA) can be found in Figure 91. Table 32 shows the original source signal EVM values, then the EVM pass through RoF system. Finally, the signal passes through RoF which coexists with different WDM systems without any degradation in EVM values.

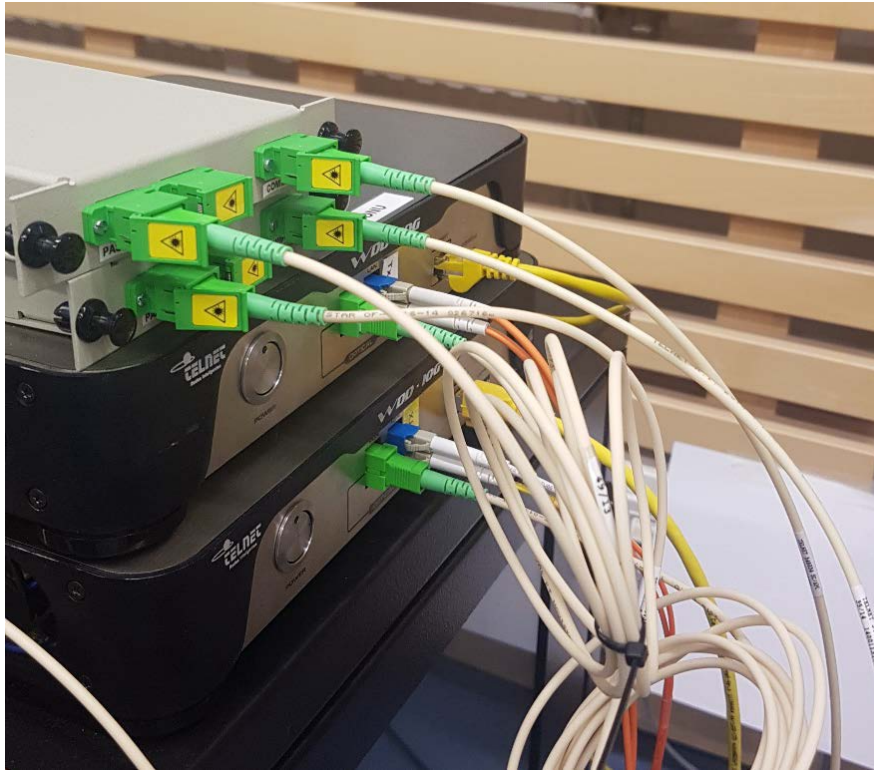


Figure 90. passive optical filters above ONUs for separation of RoF and WDM signals

Table 32: EVM results (average over 20ms)

System setting	BW (PRB)	Gain (dB)	Power (dBm)	EVM (QPSK)	EVM (16QAM)	EVM (64QAM)
Source measurement	25	7.00	10.81	1.32	1.32	1.32
After RoF	25	7.00	10.81	1.96	1.96	1.96
After RoF integration	25	7.00	10.81	1.96	1.96	1.96

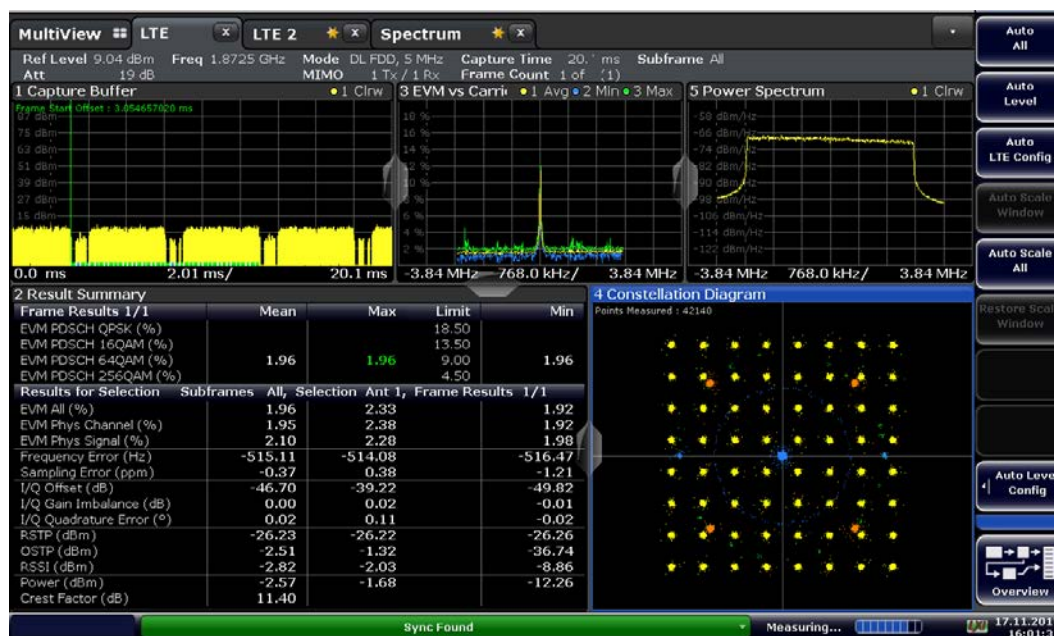


Figure 91. EVM result screen shoot after integration

The latency for FH and BH traffic, as shown in Table 29, Table 30, and Table 31 before RoF integration, does not change by the RoF integration. Repeated measurements after the RoF integration showed similar results.

5.2.2 Experiment #16: Evaluation of mixed Digital/Analogue Radio-over-Fibre Implementation

Mixed Digital/Analogue Radio-over-Fibre Implementation	
Description	Demonstration of a mixed A/D-RoF architecture with up to 3 x 10 MHz MIMO 2x2 downlink LTE signals corresponding to an equivalent fronthaul bit-rate of 3.68 Gb/s (3 x CPRI 2) and using less than 70 MHz of an off-the-shelf Directly Modulated Laser (DML). This solution architecture allows leveraging pre-existing Distributed Antenna Systems (DAS) infrastructures by pushing back the need of a migration into a WDM solution thanks to a more spectrally efficient transmission while guaranteeing, at the same, time full compatibility with currently deployed C-RAN.
State of the art	<p>C-RAN benefited from the low loss of optical fibres and the maturity of commercially available small-form factor pluggable (SFP/XFP), ON/OFF keying transceivers used for metro/access applications. While D-RoF provides increased robustness to optical noise and thus better receiver sensitivities, it is known for its inherently poor spectral efficiency. Also, bit rates scale with the number of sectors, antennas (if spatial diversity is adopted), radio access technologies and frequency bands of a mobile site.</p> <p>Whereas the use of wavelength division multiplexing (WDM) can surely alleviate the bit-rate burden of C-RAN [7], it will surely not be enough to enable the capacities expected in 5G if the current D-RoF approach is maintained. Several solutions are identified to tackle this</p>

	<p>issue, such as the compression of the fronthaul signal thanks to intelligent quantization algorithms [8], the transposition of some radio protocol layers to the RRU with different functional splits [8] and the transport of mobile signals in their native modulation format or other non-binary modulation scheme through analogue radio over fibre (A-RoF). All of such solutions could coexist in a new fronthaul interface in the future.</p> <p>A-RoF in particular has been studied for the past decades but actual deployments have targeted a restricted number of applications such as indoor scenarios (buildings, stadiums, subway stations, etc.) [12]. Analogue DAS is a typical example of such application. While analogue DAS has the merit of enabling a neutral (multi-carrier, multi-frequency) indoor distribution infrastructure, it faces, however, the same challenges as any A-RoF transmission, namely i) the conditioning of the signals needed to avoid degradations during RF aggregation; ii) the limits imposed by optical noise and its impact on the receiver sensitivity; iii) the potential nonlinear conversions on the transmission chain, which could give rise to in-band and out-of-band noise and finally iv) the RF power constraints of electrical/optical devices and the limitations they impose to the signal-to-noise ratio (SNR) of the system as we increase the bandwidth of the transmitted signal. Offline A-RoF RAN schemes with more or less complicated optics have been reported in [9], [11] and a CPRI-compatible, real-time system capable of transporting 53 Gb/s CPRI-equivalent data was recently shown in [12].</p>
<p>Improvement from State of the art</p>	<p>Real-time A-RoF demonstration, contrary to [9], [11] Twice the spectral efficiency of [12] Telco agnostic architecture Allows per-operator management/evolution of the radio front.</p> <p>Contrary to [13], the A-RoF signal is created from scratch, i.e., the system will not suffer from any degradation related to conditioning of received signals during aggregation. This also allows RF power to be more easily set to enable linear operation of the DML.</p>
<p>5G-CrossHaul Use Cases</p>	<p>Dense Urban Society</p>
<p>System under Test</p>	<p>Example use-case:</p> <p>Experimental setup:</p>

<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Increase number of devices by a factor of 10 Obj. 4: Dynamic (re-)allocation of physical resources (band frequencies and powers) Obj. 6: Increase throughput by 20%</p>
<p>Measured KPIs</p>	<p>EVM with respect to the optical loss of the link EVM with respect to the number of A-RoF bands Realistic spectral efficiency gains with respect to D-RoF Spectra and noise spectral density</p>
<p>Measurement tools</p>	<p>Electrical power meters Optical power meters LTE signal analyser</p>
<p>Measurement procedure</p>	<p>A first campaign of experiments has been realized to characterize the linear operating ranges of both electric (DAC & ADC) and optical (laser) devices of the chain. These were done respectively in electrical and optical back-to-back configurations by varying the transmitted and received RF power on the transmission chain. The main objective of this step is to obtain optimal operation values in terms of RF power for different numbers of bands in the system.</p> <p>The main measurements consist of using those optimal values and then vary the input power at the photodiode and get the mean EVM of each of the six bands. With this curve, we would be able to calculate the limits in terms of optical noise that would still allow EVM constraints to be held. Also, this allows us to calculate the penalties in terms of optical budget as we add more bandwidths. The EVM is obtained from an LTE signal analyser and the optical power with an optical power meter. The RF powers are set thanks to electrical power meters.</p>
<p>Constrains</p>	<p>For the moment, we couldn't provide any latency assessment. Most of the methods currently used are based on the CPRI C&M information. However, the C&M channel is transmitted apart from the data channel. Also, the C&M channel is changed it in their proprietary approach.</p> <p>A critical point if coexistence is tested with other splits through wavelength multiplexing is the wavelengths grid. Since we are using an off-the-shelf laser for the A-RoF, we cannot tune its wavelengths. The de/multiplexer channel must be then aligned to it.</p>
<p>Main results</p>	<p>Up to 11.05 Gb/s (9 x CPRI 2) CPRI-equivalent bit-rate using less than 200 MHz bandwidth of an off-the-shelf optical transponder</p>

	<ul style="list-style-type: none"> Up to ~36 km standard single mode fibre (SSMF) transmission distance still compliant with 3GPP in terms of EVM
--	--

Discussion of results

We have investigated the robustness of the system to optical noise and assessed the compromise between SNR and the bandwidth of the A-RoF signal. This ultimately allows us to provide a realistic estimative of the maximum number of CPRI signals that could be A-RoF transmitted with a particular laser while respecting specific end-to-end EVM constraints (results obtained thanks to offline emulation of 12 extra bands apart from the real-time signals):

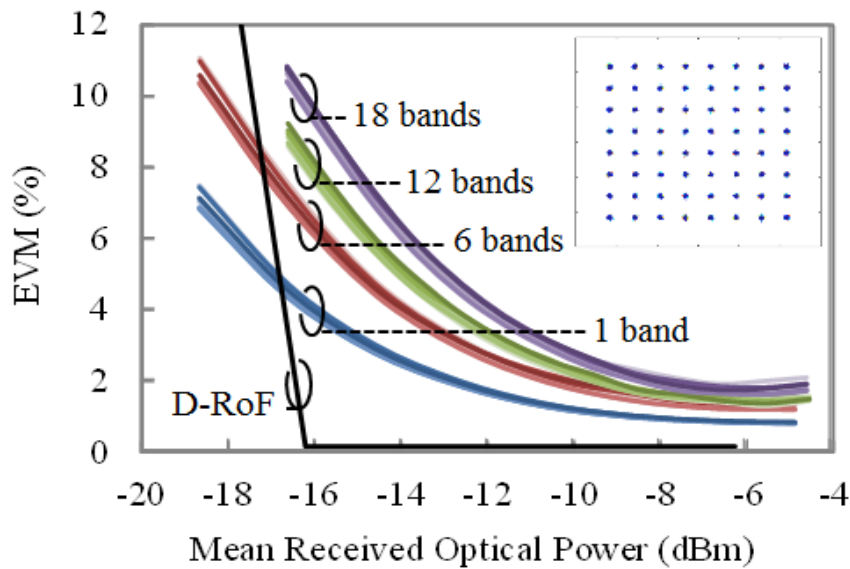


Figure 92. EVM (%) to mean received optical Power (dBm)

Estimation of the maximum allowed number of bands for three different optical budgets:

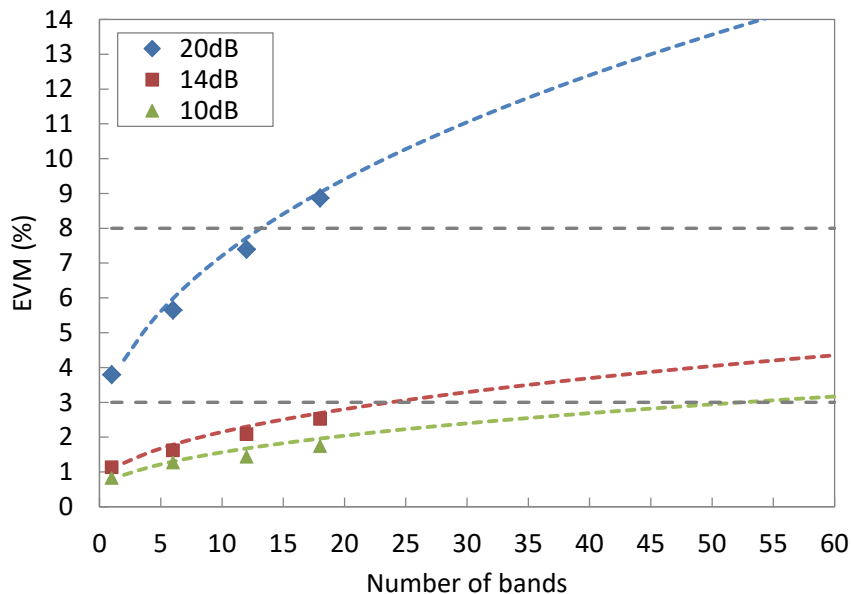


Figure 93. EVM (%) to number of bands

Spectra and EVMs for different RF power levels at the input of the laser:

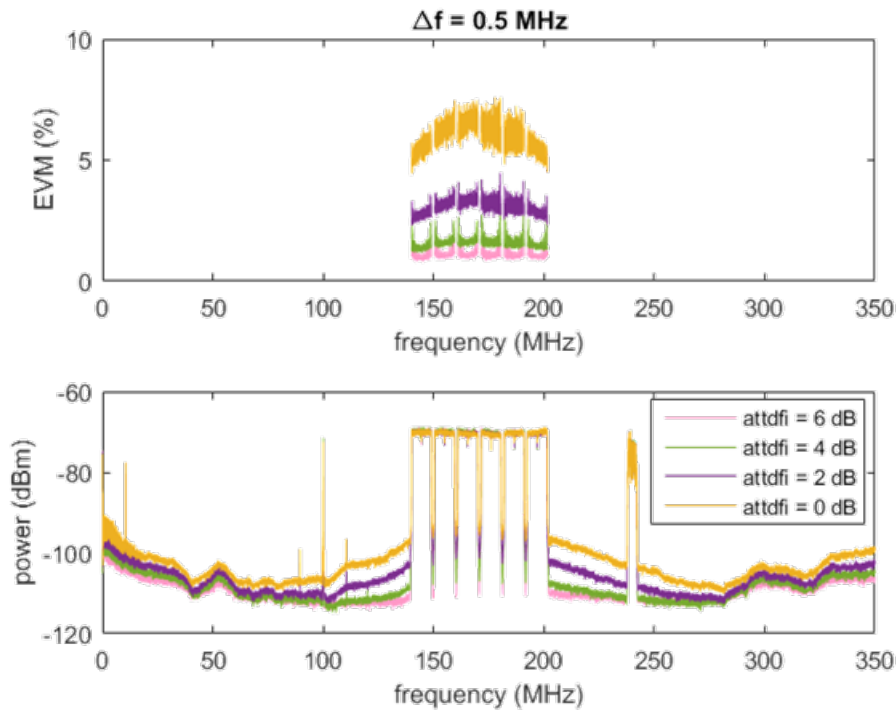


Figure 94. EVM (%) and Power (dBm) to frequencies (MHz)

5.2.3 Experiment #17: Integration of XCSE, XPFE, mmWave and CPRI compression data plane solutions

Integration of XCSE, XPFE, mmWave and CPRI compression data plane solutions	
Description	<p>This experiment showcased three transport technologies developed in 5G-Crosshaul, namely:</p> <ul style="list-style-type: none"> • EdgeLink – mmWave-based wireless transport. • Fast Forwarding (FF) – Low Latency mmWave-based wireless transport. • Compressed and Packetized Fronthaul (CPFH) <p>The considered transport profiles will include the following:</p> <ul style="list-style-type: none"> • Backhaul • Fronthaul with Higher-Layer (HL) Split - between PDCP and RLC, corresponding to Option 2 in 3GPP • Fronthaul with Lower-Layer (LL) Split - between MAC and PHY, corresponding to Option 6 in 3GPP

	<ul style="list-style-type: none"> • Fronthaul with CPRI-like Split - between PHY and RF, corresponding to Option 8 in 3GPP <p>The three transport technologies were applied to different transport profiles in accordance to their latency requirements. In particular, both backhaul and HL being more delay-tolerant traffic profiles were transported by EdgeLink while the LL split that is more delay-sensitive was transported by FF. Finally, the PHY-RF CPRI-like split traffic that has the strictest latency requirement was transported CPFH.</p> <p>All the backhaul and fronthaul traffic was injected into the XCSE, an integrated switch, that further relayed the traffic to EPC (for BH) or DUs (for FH with different splits).</p>
State of the art	<p>The comparable state of the art data-plane transport technologies include:</p> <ul style="list-style-type: none"> • Mesh-based mmWave transport: A mmWave transport network with mesh topology where blockage of a non-LoS path is avoided by SDN-controlled routing and relaying. • CPRI: Transporting in-phase and quadrature (IQ) samples of time-domain radio signals from/to distributed units. • eCPRI: The enhanced version of CPRI targeting at intra-PHY split. <p>Note that 3GPP has specified PDCP-RLC split and intra-PHY split, however, a transport network integrates mixed types of fronthaul traffic profiles and backhaul traffic has not been conceived. Additionally, eCPRI has been specified but is yet to be implemented.</p>
Improvement from State of the art	<p>The key novelty of this demonstration is an integrated transport network that integrates heterogeneous fronthaul traffic profiles (including HL, LL, and CPRI-like split), as well as backhaul. This is enabled by several new solutions including:</p> <ul style="list-style-type: none"> • mmWave transport with fast forwarding: Reduced latency attributed to processing delay in each relaying element, to support the lower-layer split. • CPFH: FH compression to reduce the bandwidth requirement of PHY-RF fronthaul split and subsequently transported over Ethernet. • Multi-Layer XFE: XCSE in the first tier, and XPFE (with XCF) in the second tier. <p>This demonstration exhibited how different traffic profiles can be carried over the transport network comprising various technologies developed by 5G-Crosshaul.</p>
5G-Crosshaul Use Cases	Dense urban society
System under Test	

<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Number of Devices increased of factor 10</p> <p>Obj. 4: 10x factor of frequency reuse</p> <p>Obj. 4: Develop physical and link-layer technologies;</p> <p>Obj. 4: Latency of < 1ms between 5G Point of Attachment (PoA) and mobile core</p> <p>Obj. 5: Deployment of outdoor and indoor small cells</p>
<p>Measured KPIs</p>	<p>Throughput</p> <p>Packet Loss Rate</p> <p>Latency</p> <p>Jitter</p> <p>EVM</p>

Measurement tools	FH/BH traffic generator and tester Laptops Lab instruments like spectrum analyser etc.
Measurement procedure	<ol style="list-style-type: none"> 1. Deployment of source nodes for different traffic profiles. In particular, backhaul traffic for LTE eNB, and fronthaul traffic profiles with PDCP-RLC split (option 2), MAC-PHY split (option 6), and PHY-RF split (option 8) were considered. 2. Transport interfaces were set up between XCSE/integrated switch and remote units with different functional splits (including PDCP-RLC split, MAC-PHY split, and PHY-RF split), as well as a standalone LTE eNB. These included the conversion between Ethernet Copper and Optical media. 3. RAN split hosts were connected to the XPFE and a verification of traffic flows / connection through the combined network conducted. 4. Evaluated throughput, latency, jitter through multiplexed links (i.e. the links that multiplex backhaul and fronthaul traffic). Note that the multiplexed links are marked in red in the system under test diagram.
Constraints	Traffic throughput for functional split limited to chosen configuration, multiplexed link latency has to be tested for fronthaul splits.
Main results	<p>The main results of this experiment show the feasibility to use different inter-connected transport technologies (mmWave, XCSE and XPFE) to simultaneously transport backhaul and fronthaul traffic (both upper and lower layer functional splits). In the first set of experiments we have functionally shown that the XCSE guarantees the bandwidth for all flows, independently of the functional split used. This was done by generating flows between UEs all the way through the mobile network to an Iperf server. For example, the bandwidth of the higher layer split is 18.28 Mbps in the downlink and 25.73 Mbps in the uplink; in the lower layer split, the bandwidth is 67.65 Mbps in the downlink and 17.46 Mbps in the uplink; for the backhaul traffic, the resulting bandwidth is 8.71 Mbps in the downlink and 10.37 Mbps in the uplink; the UE connected to the option 8 split RRU provides a bandwidth of 45.504 Mbps in the downlink and the compressed fronthaul 262.4 Mbps.</p> <p>In the first sub-experiment, the compressed and packetized fronthaul traffic was transported over the XCSE. In the second sub-experiment, all transport technologies were used together to transport the upper and lower layer split fronthaul traffic. It was shown that the Fast Forward mmWave together with the XCSE can satisfy the one-way delay constraints imposed by the MAC-PHY split (the average RTT is 0.26 ms). This is also the case even with high-load UDP traffic, where the RTT average value is 0.44 ms. The maximum RTT obtained with background traffic suggests that constraints are also satisfied if we take as reference the documents of the small cell forum [52], because the average of the maximum results is 1.27 ms, with a maximum of 1.5 ms. Other more stringent implementations and recommendations may not be fulfilled in all cases.</p>

Discussion of results

Two different experiments were conducted over the system under test (Figure 95 shows a picture of some of the components), in order to differentiate the different results depending on the equipment under study.

The first experiment (#17.1) utilized all components described in the system under test except the mmWave island. Considering the strict delay requirements imposed by the lower-layer splits, the goal of this experiment was to test the feasibility of using different functional splits, backhaul and fronthaul traffic, as well as compressed/packetized fronthaul over a multi-layered Crosshaul transport (XPFEs and XCSEs).

The second experiment (#17.2) was focused on determining the feasibility of utilizing the mmWave island (alone and in combination with the XCSE and XPFEs) to transport backhaul and fronthaul traffic generated by both lower-layer and upper-layer splits.

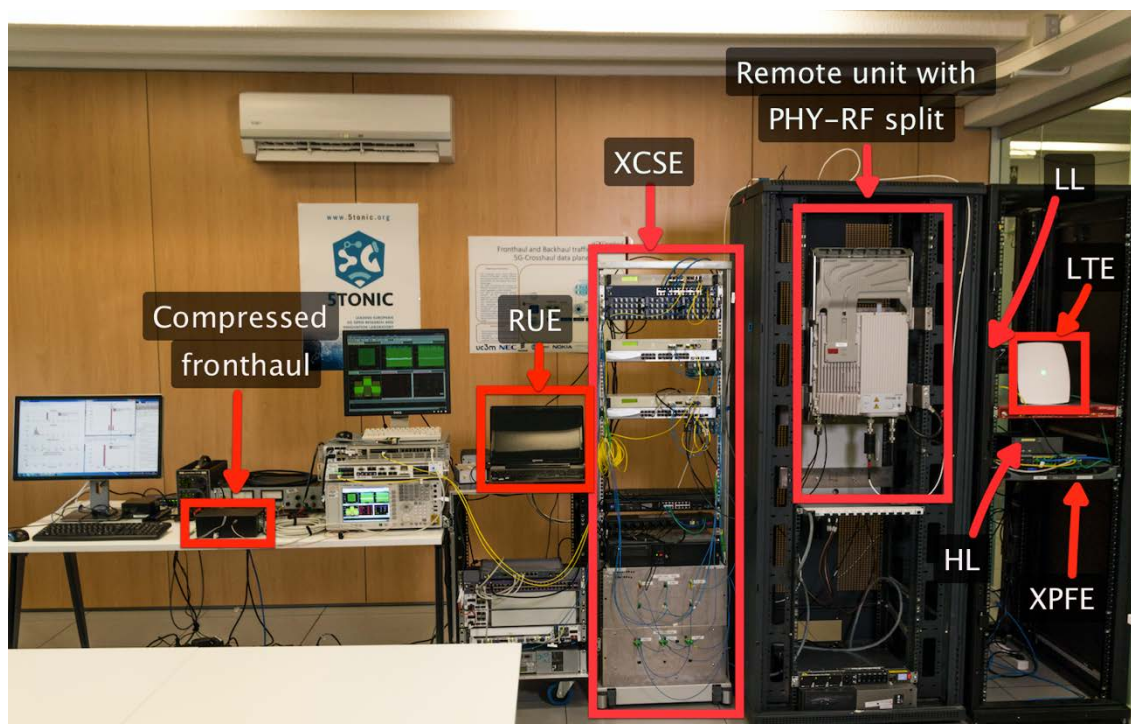


Figure 95. Picture showing some of the components used in experiment #17.

5.2.3.1 Experiment #17.1

After setting up the testbed, several tests were conducted in order to collect information about the performance of the different components involved in this experiment. While the Remote Radio Unit (RU) with Compressed/Packetized Fronthaul was receiving traffic, other user equipment (UE) devices were configured to start and stop sending and receiving traffic. The four UEs used in these tests were connected: 1) to the MAC/PHY RU (LL UE or LL in short), 2) to the PDCP/RLC RU (HL UE or HL in short), 3) to the LTE Small Cell (LTE UE or LTE in short), and 4) to the RU CPRI fronthaul PHY/RF (RU UE or RUE in short). All UEs could transmit and receive traffic towards the application server (AS) except the RUE, which could only receive traffic in the downlink.

Depending on when the UEs start, four scenarios were deployed to test both the throughput and the round-trip time (RTT) between different UEs and the AS. For example, Table 33 shows the results of the throughput of the first scenario (17.1.1) for both downlink (DL) and uplink (UL) in Mbps¹². In this configuration, the LL starts sending and receiving traffic during four minutes; after the first minute, the HL starts sending and receiving traffic to the AS but only during three minutes; after two minutes, the LTE UE starts sending traffic for two minutes and, finally, the RU sends traffic during the last minute.

In the next scenario (17.1.2), we shifted down the UEs from the first scenario, so RUE starts the transmission, followed by the LL, then the HL and finishing with the LTE device. In the remaining scenarios (17.1.3 and 17.1.4) the procedure is similar, so at the end, we have results for all UEs transmitting alone.

After analysing the results of throughput (in Mbps) presented in tables from Table 33 to Table 36 and the results of the RTT (in seconds) shown in tables from Table 37 to Table 40, we can conclude:

- It is feasible to transport backhaul, lower-layer split, upper-layer split, CPRI and compressed/packetized traffic over the multi-layered transport defined in 5G-Crosshaul project.
- For every UE, the performance is very similar in all scenarios, no matter if the UE is sharing the path with other UEs. This result shows the guaranteed low latency and high bandwidth for packet provided by the XCSE.
 - The RUE presents a high standard deviation, so it is difficult to extract conclusions from these results. For example, in the second scenario shown in Table 34 and Table 38, the average throughput of the RUE decreases when other UEs start their activity, but it is important to notice that the standard deviation is high in all results. In all these experiments, the EPC serving the RUE is located in Sweden, while the RUE and AS were deployed in the 5TONIC site, thus the high RTT.
- The throughput is different for different UEs, but it mainly depends on the RTT between each UE and the AS, where UEs share links (HL, LL and LTE). As shown in the system under test figures, the HL device has to pass through the DU which is connected to the third XPFE, which increases the RTT, while the LL UE and its DU are both directly connected to the same XCSE.

¹² All values are calculated as the average (top value in every cell) and standard deviation (bottom) of twenty independent executions.

Table 33: Throughput scenario 17.1.1 (average and standard deviation)

UE	0-60 s		60-120 s		120-180 s		180-240 s	
	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)
LL	67.655 2.6964	17.458 3.5424	66.756 3.4551	18.906 4.8847	66.978 2.1321	16.539 5.1344	68.359 2.1059	14.193 5.8967
HL			22.233 11.477	20.809 11.795	26.722 11.274	17.184 11.451	25.342 11.708	18.312 11.412
LTE					8.7630 0.8553	8.5713 0.9239	8.9170 0.6435	9.0860 0.9583
RUE							43.706 12.63	

Table 34: Throughput scenario 17.1.2

UE	0-60 s		60-120 s		120-180 s		180-240 s	
	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)
RUE	45.504 11.811		44.294 12.426		38.466 11.222		38.082 13.136	
LL			66.611 5.5568	21.402 7.1005	66.217 12.2222	17.107 13.1974	68.150 3.4373	15.592 6.9124
HL					23.154 12.149	20.519 11.921	20.775 12.159	22.246 11.843
LTE							8.8136 0.7258	8.9948 1.1360

Table 35: Throughput scenario 17.1.3 (average and standard deviation)

UE	0-60 s		60-120 s		120-180 s		180-240 s	
	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)
LTE	8.7150 0.45932	10.3752 1.21533	8.8718 0.54203	9.4646 0.81638	9.3120 0.49413	8.7791 0.63563	9.0376 0.52677	9.4227 0.86345
RUE			42.986 13.9515		43.784 8.6886		32.740 8.1168	
LL					66.292 12.7449	16.441 13.3461	68.363 2.8215	14.822 5.9060
HL							28.039 10.9506	15.331 9.9634

Table 36: Throughput scenario 17.1.4 (average and standard deviation)

UE	0-60 s		60-120 s		120-180 s		180-240 s	
	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)	DL (Mbps)	UL (Mbps)
HL	18.276 11.791	25.732 11.962	21.870 12.116	21.946 12.174	26.056 11.961	18.226 11.702	20.860 12.326	23.013 11.970
LTE			8.4461 0.5359	10.180 0.697	8.7205 1.0984	9.2049 1.4456	8.8823 0.5673	9.5396 0.8930
RUE					49.793 15.305		44.113 14.603	
LL							66.735 11.130	13.748 13.052

Table 37: RTT scenario 17.1.1 (average and standard deviation)

UE	0-60 s			60-120 s			120-180 s			180-240 s		
	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max
LL	6.65	16.40 5.77	26.14	6.69	16.44 5.74	25.91	6.79	16.46 5.74	25.96	6.79	16.55 5.74	25.90
HL				2.56	3.39 0.84	6.75	2.67	3.25 0.57	5.64	2.71	3.07 0.31	4.51
LTE							25.7	32.6 10.29	98.8	26.2	31.6 2.71	37.5
RUE										139.8	145.26 3.28	154.1

Table 38: RTT scenario 17.1.2 (average and standard deviation)

UE	0-60 s			60-120 s			120-180 s			180-240 s		
	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max
RUE	139.3	145.1 3.1	152.2	139.9	145.3 3.4	156.1	139.6	145.4 3.4	154.4	139.9	145.3 3.5	156.8
LL				6.73	16.2 5.8	25.9	6.75	16.4 5.8	25.9	6.86	16.6 5.8	26.3
HL							2.73	3.51 0.82	7.31	2.86	3.23 0.36	4.95
LTE										26.6	33.2 12.7	114.4

Table 39: RTT scenario 17.1.3

UE	0-60 s			60-120 s			120-180 s			180-240 s		
	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max
LTE	26.7	33.8 15.9	144.6	26.8	31.9 2.74	37.29	26.3	31.9 2.82	37.73	27.1	32.8 2.8	38.58
RUE				139.2	145.1 3.1	151.7	139.9	145.3 3.2	153.8	139.8	145.3 3	151.9
LL							6.8	16.3 5.76	25.9	6.75	16.4 5.74	25.6
HL										2.62	3.29 0.66	5.98

Table 40: RTT scenario 17.1.4 (average and standard deviation)

UE	0-60 s			60-120 s			120-180 s			180-240 s		
	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max	Min	Avg Std	Max
HL	2.77	3.45 1.94	17.6	2.62	4.76 10.23	81.9	2.58	4.39 5.51	37.5	2.53	3.61 2.4	17.1
LTE				25.4	33.3 15.1	138	26.1	31.5 2.7	37.25	26.3	31.7 2.71	37.5
RUE							139.8	145.2 3.18	152.3	140.1	145.3 3.01	151.8
LL										6.92	16.3 5.74	25.9

On the other hand, in this experiment we also show that fronthaul can be packetized and transported over an Ethernet network (as in eCPRI) and that In-phase and Quadrature (I/Q) data compression can be applied to substantially decrease the fronthaul bitrate requirements. This is based on the results provided by the fifth flow between the CRAN-BBU (cCPRI-PHY/RF) and the RU (Compressed/Packetized Fronthaul-PHY/RF). This is implemented based on a CRAN architecture that is integrated with the XCSE transport, as shown in the figure presented in the system under test. A baseband emulator (CRAN – DU node) is used to generate a 20 MHz 256-QAM LTE baseband signal and encapsulate this into a 9.8G CPRI stream. 30 bits are used per I/Q pair and the I/Q data bit rate then corresponds to ~922 Mbps. CPRI overhead, line coding and

OAM data is also added and approximately 1.2 Gbps of the CPRI stream is needed to transport the carrier. The CPRI stream is then sent through the XCSE transport to a compression and packetization node.

The compression and packetization node terminates the CPRI stream and then applies compression algorithms that utilize the signal redundancy as well as entropy gain to decrease the I/Q data bitrate. The compression is based on resampling, re-quantization and a modified Huffman code, which results in a lossy compression, and thus, a trade-off between compression ratio and radio performance.

Finally, the compressed I/Q data are encapsulated into Ethernet frames and sent to a radio head (RH) where they are decompressed. The signal is then upconverted to radio frequency (LTE band 3) through the Analogue Front End (AFE). The antenna port of the RH is connected with a coaxial cable to an LTE analyser so that radio performance can be measured.

All the nodes in the setup are synchronized. The baseband emulator is synchronized with a 10 MHz signal from the XCSE transport hub node, which utilizes a recovered clock from a Digital Unit. The compression and packetization node is synchronized via the CPRI link. This node then uses SyncE to provide synchronization with the RU. Finally, the LTE analyser is synchronized by a 10 MHz signal over a coaxial cable from the baseband emulator.

The purpose of this proof-of-concept testbed is to focus on the FH interface and test how well the FH compression fulfils the 3GPP requirements. Therefore, the system is evaluated by measuring the Ethernet bitrate to indicate the compression ratio, Error Vector Magnitude (EVM) to show the radio performance and frequency errors to show the frequency synchronization performance.

Measurement data were collected over three days to verify the testbed performance and stability over time. The results are shown in the Figures below and the data are visualized with both time series plots as well as Probability Density Functions (PDF) histograms.

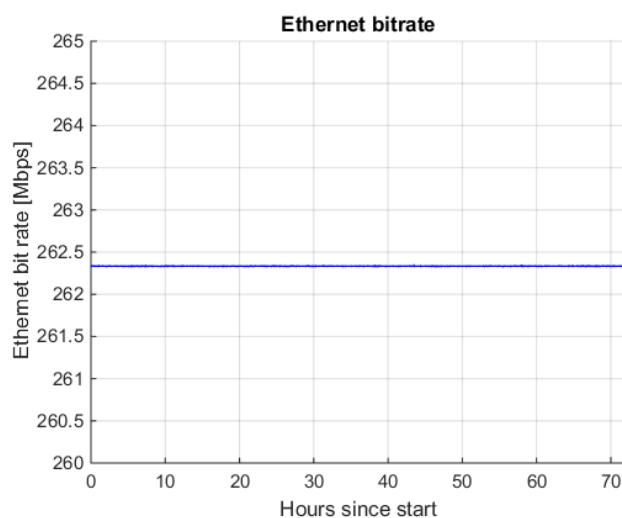


Figure 96. Ethernet bitrate [Mbps] over time

Figure 96 shows that the measured Ethernet bitrate of the compressed FH data is stable over time and hence no PDF is needed to visualize the results. The bitrate is about 262.3 Mbps, which corresponds to an I/Q data compression ratio of ~ 3.7 (neglecting the CPRI overhead). If the bitrate is compared with the corresponding CPRI stream bitrate the bit rate requirement is ~ 4.7 times lower when using the packetized and compressed fronthaul. Note though that no OAM channel is implemented in this prototype (configuration is done directly on the RU).

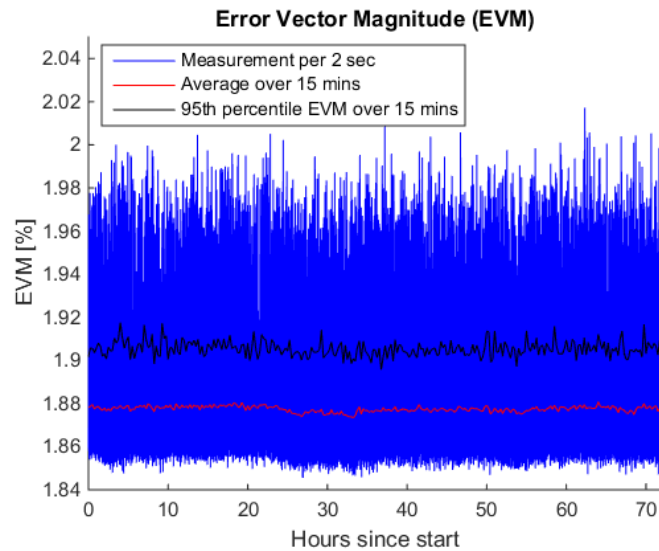


Figure 97. EVM [%] over time with average and 2 standard deviations

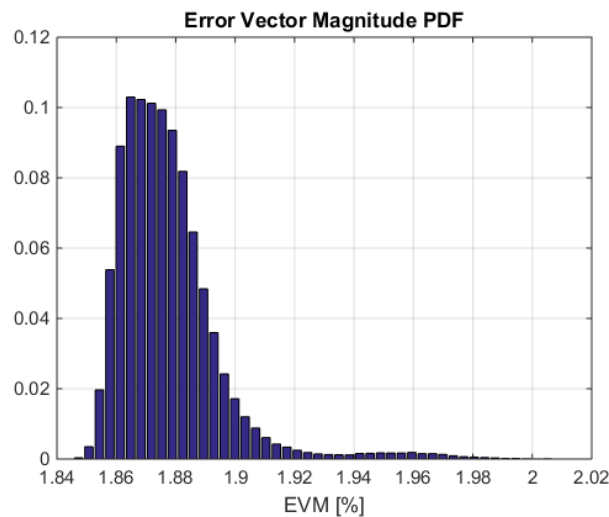


Figure 98. EVM PDF histogram

Figure 97 and Figure 98 show the EVM as both a time series plot (with each sample, average and 95th percentile EVM displayed) as well as a PDF histogram. The results confirm that also the EVM is stable over time with an average value of ~1.88 %, 95th percentile value below 1.92% and the maximum value about 2%. This indicates the radio performance is kept well below the 3GPP requirements (TS 36.104 specifies EVM < 3.5 % for 256 QAM).

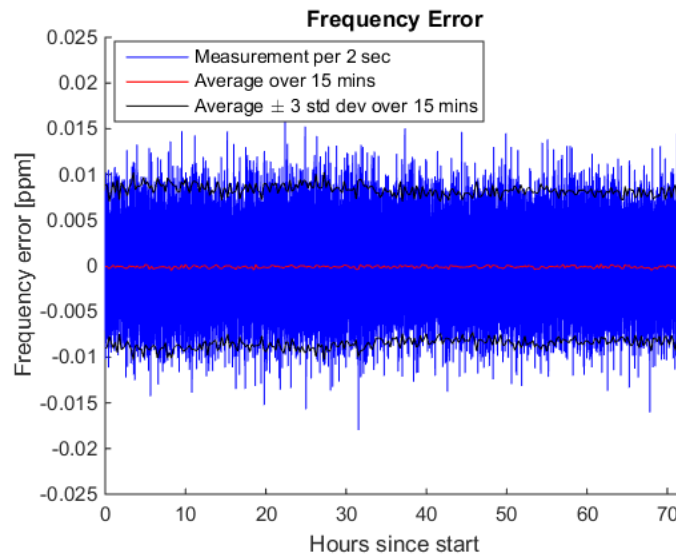


Figure 99. Frequency error [ppm] over time with average and 2 standard deviations

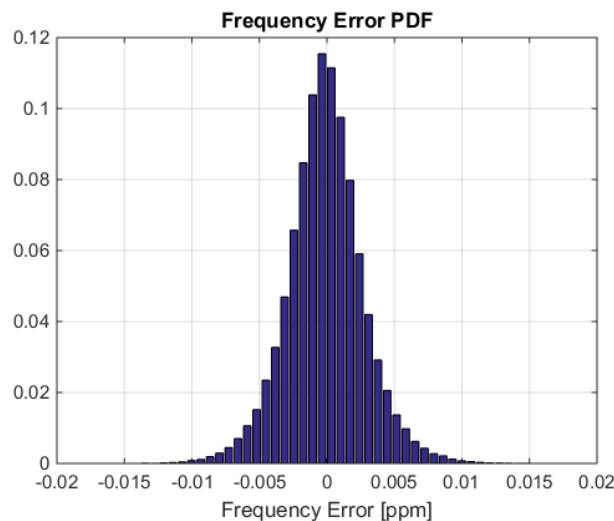


Figure 100. Frequency error PDF histogram

The measured frequency error is displayed in Figure 99 and Figure 100. These figures show that the frequency sync is stable with the variation between ± 0.015 ppm, which are well below 3GPP requirement (TS 36.104 specifies frequency error ± 0.1 ppm for a local area base station) thus indicating that the synchronization implementation is adequate.

Overall, the tests show the feasibility to implement a high-efficiency FH compression and send the packetized and compressed FH data over Ethernet, while it fulfils the 3GPP requirements in radio and synchronization performance.

5.2.3.2 Experiment #17.2

The goal of this sub-set of experiments was to analyse the performance of the different transport technologies used in the system under test, namely the mmWave Fast Forward and Edge Link,

the XCSE, and the XPFE. This experiment presents three different results: the mmWave technologies isolated, the mmWave and the XCSE, and, finally, the three technologies together.

5.2.3.2.1 mmWave

A set of in-lab measurements were taken on the mmWave-based forwarding elements prior to integration with other data plane solutions, i.e., XCSE, XPFE, and CPRI compression. A new mechanism called “Fast Forwarding” (FF) was proposed in WP2 to curtail processing delay for cases with stricter latency requirements (e.g., lower-layer split). It is worth pointing out that Fast Forward operates at 70GHz with frequency-division duplex (FDD), where UL and DL occupy different bands.

mmWave links based on Fast Forwarding, achieved over-the-air (OTA) link capacities of 392 Mbps and 510+ Mbps, under MCS 9 and MCS 11, respectively. The results on throughput were captured via *iperf* traffic generators with packet sizes of 1512 bytes, and a user payload of 1470 bytes was considered for throughput calculation (i.e., the header of 42 bytes was not considered).

The latency and packet loss rates with both Normal (Norm) and Fast Forwarding (FF) modes was examined under different traffic patterns. The measurement results are summarised in Table 41.

Table 41: Measurement results of latency and packet loss rates for both Normal and FF modes under different traffic profiles

Traffic Profiles	One-Way Latency	Packet Loss Rates
Throughput: 378Mbps (MCS 9)	Norm: UL - 99μs; DL - 113μs FF: UL - 83μs; DL - 96μs	Norm: UL – 0.0035%; DL – 0.0083% FF: UL – 0.012%; DL – 0.06%
DL Throughput: 153 Mbps UL Throughput: 51 Mbps (MCS 2)	Norm: UL - 138μs; DL - 153μs FF: UL - 93μs; DL - 110μs	Norm: UL – 0%; DL – 0% FF: UL – 0.000023%; DL – 0.0053%
DL Throughput: 61 Mbps UL Throughput: 5 Mbps (MCS 2)	Norm: UL - 138μs; DL - 147μs FF: UL - 93μs; DL - 103μs	Norm: UL – 0%; DL – 0% FF: UL – 0%; DL – 0.0083%

From the tabulated results, it was evident that FF is able to provide benefits in terms of latency reduction. Specifically, when the FF technique is applied, the latency can be decreased by around 30% to 33%.

Additionally, further measurements were conducted to inspect round trip delays for point-to-point (P2P) EdgeLink transmissions. The latency was measured between traffic generators (instead of node-to-node latency). More importantly, the measurement was undertaken based on the traffic profile involving both backhaul and fronthaul with higher-layer split (UL and DL were 50 and 150 Mbps respectively). The measurement results corresponding to this scenario are summarised in the Table 42.

Table 42: End-to-end measurement results for EdgeLink

Direction	Latency	Packet Loss Rate (UDP)
Uplink	1.341ms	0.00039%
Downlink	1.263ms	0.00018%

5.2.3.2.2 mmWave and XCSE

This experiment was designed to validate the feasibility to use the Fast Forward mmWave and the XCSE together to carry lower layer (MAC-PHY) split fronthaul traffic. In order to collect results to validate the Round-Trip Time (RTT), we used the *ping* command to generate packets between the LL UE (LL) and the LL DU (request and reply). In the test, the ping command was executed during one minute, transmitting packets every one second. The output of this execution provides the minimum, average and maximum RTTs. This was repeated 20 times to collect a statistically significant set of results. This process was conducted in two different scenarios: only the ping process generating traffic (LL no traffic) and another scenario where the ping process shares the channel with a background traffic (LL with traffic). The background traffic was generated using the *iperf* tool, transmitting a 1 Gbps simultaneous bidirectional UDP traffic between the LL UE and the application server (AS).

Figure 101 shows a boxplot representing the main results of the average RTT. For each experiment (with and without traffic), the boxplot shows (from bottom to top) the minimum, first quartile, median, third quartile and maximum of all average RTT values obtained in the tests. The average values, even the highest ones, are below the delay constraints imposed by this lower layer split, i.e., 4 ms [52]. Other sources and specific implementations may require different values for this split.

Figure 102 shows the boxplot of the maximum RTTs obtained in these tests, both with and without traffic (circles represent outliers). These results show that the highest RTTs without background traffic are below the delay constraint imposed by the LL split, i.e., 4ms. Even with background traffic present the maximum RTT values are below this limit. It is worth noting that the average UDP traffic in the uplink direction was 28.38 Mbps.

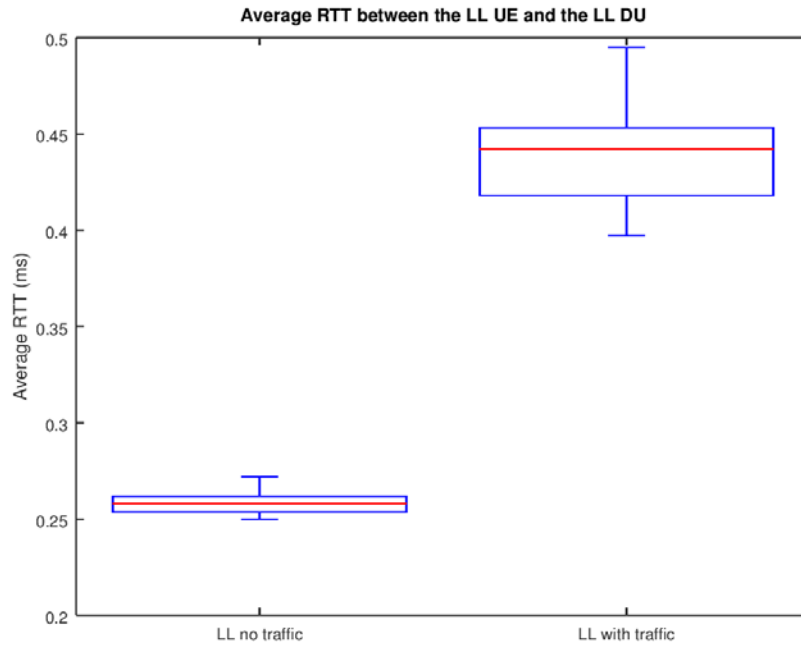


Figure 101. Average RTT between the LL UE and the LL DU

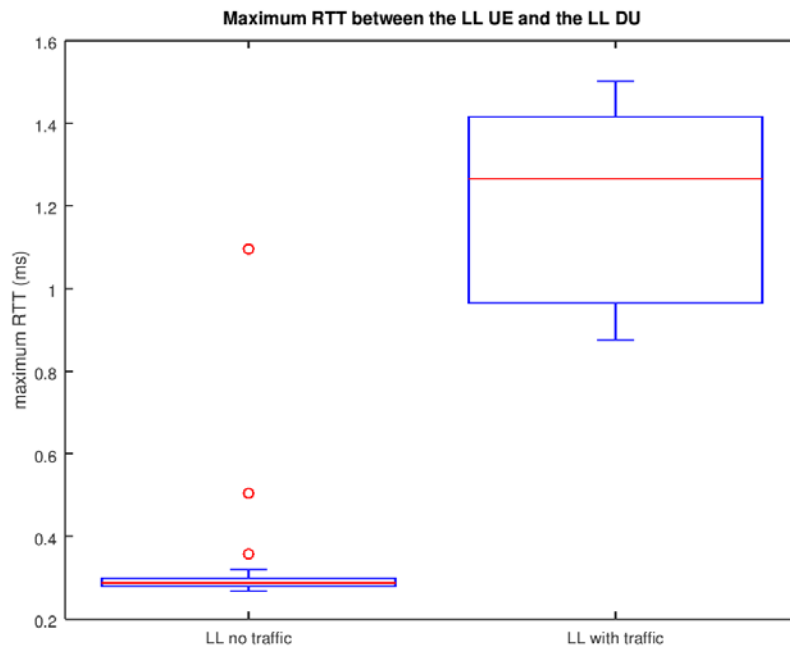


Figure 102. Maximum RTT between the LL UE and the LL DU

5.2.3.2.3 mmWave, XCSE and XPFE

The goal of this experiment was to collect results related with the aggregate delay between the Edge Link mmWave, the XCSE, and the XPFE path. Similar to the previous experiment, the *ping* command was used to obtain the RTT. This time, the ping was executed between the RU for the PDCP-RLC (HL) split and the DU for that split. It is important to notice that there are three XPFEs involved in this communication exchange. Again, these experiments were conducted with and without background traffic using the *iperf* tool. The 1 Gbps simultaneous bidirectional UDP traffic was generated between the UL UE and the application server.

Figure 103 shows the boxplot of the average RTT values obtained in the twenty executions for both tests, with and without background traffic (the plus sign below the boxplot of UL with traffic represents an outlier value). The UL split does not impose any restriction in the one-way delay, but it is important to notice the increase in the average RTT when the background traffic was present.

Finally, Figure 104 presents the boxplot of the maximum RTT values obtained in these experiments involving the HL split RU and DU devices.

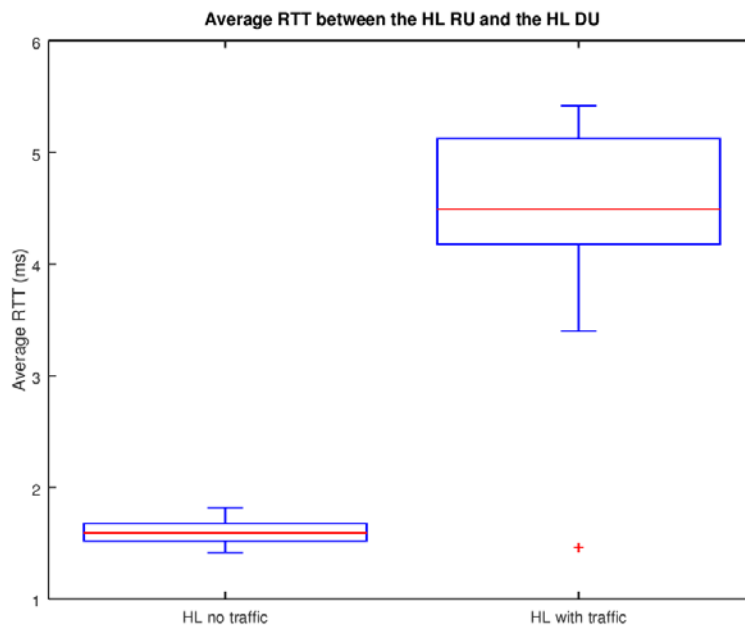


Figure 103. Average RTT between the HL RU and the HL DU

After presenting all these results, it was concluded that all these transport technologies together can be used to simultaneously transport fronthaul (both upper and lower layer splits) and backhaul. We can expect some performance degradation after heavy UDP traffic load, but even in such scenarios, for the average of the traffic, the main constraints are satisfied.

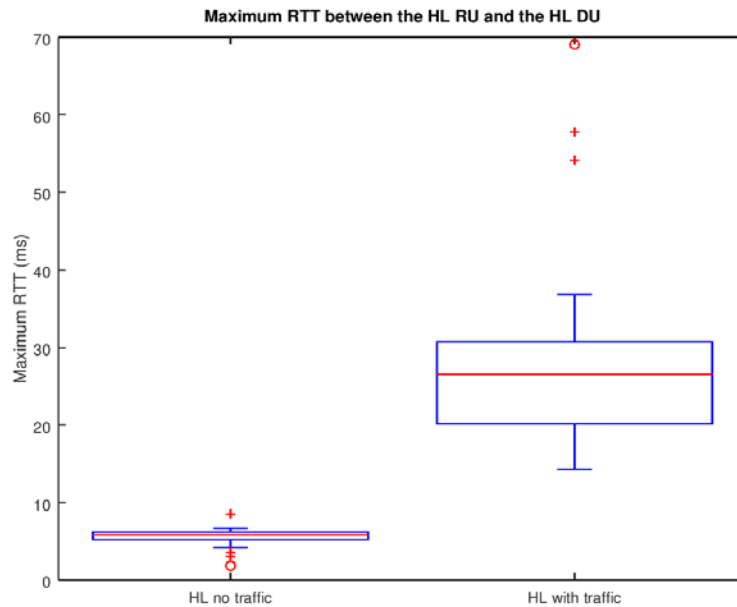
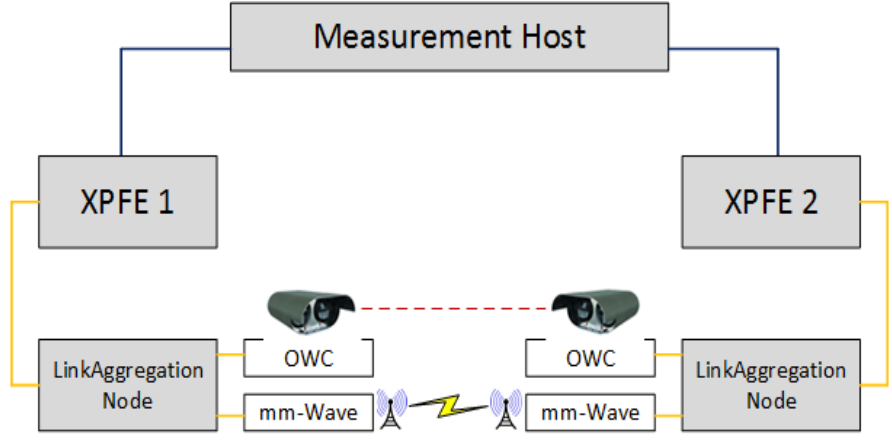


Figure 104. Maximum RTT between the HLRU and the HL DU

5.2.4 Experiment #18: Evaluation of integrated packet-based FH/BH combining wired XPFEs and hybrid mmWave/optical wireless links

Evaluation of integrated packet-based FH/BH combining wired XPFEs and hybrid mmWave/optical wireless links	
Description	This experiment evaluates whether XPFEs connected with the hybrid optical wireless / mmWave system can satisfy the requirements for higher layer functional splits.
State of the art	mmWave: the used commercial proprietary Etherhaul 600TX solution by siklu is operating in the 60 GHz band and uses TDD. It uses 500 MHz bandwidth for up to 500 Mbit/s bidirectional communication. Optical Wireless Communication (OWC): system is state of the art for optical wireless transmission in free space based on low budget LEDs. Other SoTA systems are based on Laser modulation. XPFE: SDN based soft-switches with packet acceleration in user space.
Improvement from State of the art	Integrate OWC and mmWave system into combined, hybrid solution for either link-failure compensation and/or link-aggregation for throughput improvements. XPFE: reduced latency compared to soft switches in user space.
5G-Crosshaul Use Cases	Dense urban society

<p>System under Test</p>	
<p>Project Objectives / 5GPPP KPIs addressed</p>	<p>Obj. 1: Number of Devices increased of factor 10 Obj. 4: 10x factor of frequency reuse Obj. 4: Develop physical and link-layer technologies; Obj. 5: Deployment of outdoor and indoor small cells</p>
<p>Measured KPIs</p>	<p>Throughput Packet Loss Rate Latency Jitter</p>
<p>Measurement tools</p>	<p>FH/BH traffic generator and tester Laptops</p>
<p>Measurement procedure</p>	<ol style="list-style-type: none"> 1. Connect XPFEs with cable as reference. 2. Evaluate throughput, latency, jitter for directly connected XPFEs. 3. Deployment of mmWave link as cable replacement 4. Evaluate throughput, latency, jitter through mmWave link 5. Deployment of optical link as cable replacement 6. Evaluate throughput, latency, jitter through optical link 7. Deployment of hybrid link + aggregation node 8. Connection between aggregation node and XPFE and verification of traffic flows 9. Evaluate throughput, latency, jitter through aggregated link
<p>Constraints</p>	<p>Traffic throughput for functional split limited to chosen configuration Packet size in aggregation mode cannot exceed 1490 bytes</p>
<p>Main results</p>	<p>Latency of mmWave and optical wireless differs but both link technologies can support backhaul operation. Due to their proprietary nature, configuration and adaptation no smaller average latency values than 0.5 ms to 1.5 ms, depending on link type, could be achieved. 99th percentile values were as large as 4.5 ms. Therefore, lower layer splits, i.e., split options 4 to 8 are not supported. Higher layer splits within or above RLC, i.e., split options 1 to 3 are supported.</p>

Back-to-back test

Both XPFEs in the setup have been connected directly by a cable to perform reference measurements. The latency through both XPFEs is approximately 22 μ s, with a standard deviation of about 8 μ s and a 99th percentile of 51 μ s, see Table 43. 500 Mbps were generated in both directions with a packet size of 512B.

Table 43: Latency for directly connected XPFEs

Throughput	#	Latency [μ s]	StdDev[μ s]	99% [μ s]
BH, 500Mbps, bidirectional, 512 B	1, DL	21.968	8.158	51.392
	1, UL	22.116	7.916	51.704
	2, DL	22.015	8.030	51.488
	2, UL	21.984	7.085	51.696
	3, DL	21.979	7.903	51.304
	3, UL	21.926	7.486	51.696

mmWave wireless link test

The mmWave link has a capacity of 500 Mbps, latency has been measured with large (1500 B) and small (64 B) packets to investigate the extremes regarding encapsulation overhead and serialization delay. For small packets, the overhead by XCF encapsulation is relatively large, about 60%. As the links have a capacity below 1 Gbps, the serialization delay of large packets is considerably higher than for small packets. For both large and small packets 300 Mbps plain test traffic has been used, implying different link utilization of the optical and mmWave links, respectively.

Latency and jitter increase considerably compared to the reference measurement. Average latency is in the order of 380-400 μ s, see Table 44.

The increase in latency and especially in jitter is due to packet aggregation and up to 3 retransmissions on the link. The latency with small packets is slightly higher than with large packets. This could be related to operating the link close to its capacity. Remember that the relative overhead by XCF encapsulation is larger for small packets, therefore the link utilization for a given rate of test traffic is higher after encapsulation for small packets than for large packets.

The optical link has even less capacity, about 450 Mbps on physical layer, than the mmWave link, and it is adapting quickly to the link condition. Therefore, in practice a capacity of 300 Mbps can be expected. With test traffic of 300 Mbps the link is already overloaded due to the overhead by the XCF encapsulation.

Table 44: Latency for XPFEs connected via mmWave link

Throughput	#	Latency [μ s]	StdDev[μ s]	99% [μ s]
BH, 300Mbps, bidirectional, 64 B	1, DL	398.868	110.464	684.208
	1, UL	396.622	102.359	659.272
	2, DL	397.226	103.682	530.872
	2, UL	418.958	148.964	890.000
	3, DL	404.815	122.371	746.992
	3, UL	401.340	103.585	710.104
BH, 300Mbps, bidirectional, 1500 B	1, DL	378.638	130.903	567.256
	1, UL	379.303	140.446	567.464
	2, DL	380.671	215.790	564.552
	2, UL	379.185	131.005	565.776
	3, DL	378.660	129.491	564.104
	3, UL	379.645	131.713	566.888

The latency and jitter on the optical link is again larger than on the mmWave link. Measurements for 150 Mbps of test traffic with small packets are shown in Table 45.

Table 45: Latency for XPFEs connected via optical link

Throughput	#	Latency [μ s]	StdDev[μ s]	99% [μ s]
BH, 150Mbps, bidirectional, 64 B	1, DL	611.430	160.112	1326.848
	1, UL	621.157	160.927	1309.248
	2, DL	621.183	160.060	1322.992
	2, UL	621.326	161.639	1340.504
	3, DL	616.422	161.007	1319.040
	3, UL	618.231	161.774	1336.808

Hybrid wireless link test

In order to operate in the link aggregation node, we utilized a round-robin scheme for forwarding the arriving packets to the different link technologies. The round-robin scheme was considered as a first approach, in particular when the difference in latency between mmWave and OWC is taken into account. but for optimal aggregation mode a more complex implementation is needed. This is not suitable with the utilized proprietary equipment.

Table 46: Latency for XPFEs connected via hybrid wireless link

Throughput	#	Latency [μ s]	StdDev[μ s]	99% [μ s]
BH, 300Mbps, bidirectional, 64 B	1, DL	796.058	350.157	1839.800
	1, UL	779.935	350.083	2396.760
	2, DL	794.693	348.055	1826.664
	2, UL	761.158	277.402	1663.008
	3, DL	799.160	356.993	1872.192
	3, UL	763.956	287.637	1707.360
BH, 600Mbps, bidirectional, 1490 B	1, DL	1536.851	944.966	4245.416
	1, UL	1423.701	784.861	3297.944
	2, DL	1541.330	949.766	4294.528
	2, UL	1429.154	773.423	3226.704
	3, DL	1531.533	926.376	4180.104
	3, UL	1420.874	762.598	3172.216

As shown in Table 46, the aggregated link can double the throughput based on the initial measurements for mmWave and OWC leading to 600 Mbps data traffic, while for a moderate aggregate data rate of 300 Mbps the measured latencies are comparable to those of the single wireless links, the situation changes dramatically for the higher data traffic case. At 600 Mbps an average latency of c. 1.5 ms with a standard deviation of c. 0.9 ms is observed, the 99-percentile moving towards 4.5 ms. Considering the average latency under 2 ms and the high standard derivation of almost 1 ms the implementation of the link-aggregation is non-optimal. The chosen link aggregation implementation based on Linux kernel bonding drivers offers the possibility to modify, adapt and even implement a new mode for uneven links based on these results.

With a 99th-percentile of the latency of more than 0.5 ms (mmWave), more than 1ms (optical link), and more than 4 ms (hybrid link) these are not suitable for functional splits of the MAC or lower layers, i.e. split options 4 to 8. The HARQ loop is terminated in the MAC and cannot tolerate latencies in this order. Splits of the RLC are possible, to reduce also link utilization it

would be best to use these links for splits between RLC and PDCP and for BH traffic only, i.e. split options 1 to 3 are supported.

Besides this implementation optimizations, the OWC and mmWave technologies are both highly directive and thus allow a frequency reuse in- and outdoor. There is no quantification, but the same optical and mmWave frequencies can be used to connect small cells via backhaul indoor as well as outdoor to densify the network. Additionally, as already shown in [47], a further important advantage of the hybrid link is its robustness under bad weather conditions. Corresponding results of a long-time (>5 months) monitoring have demonstrated that the influence of, e.g., fog or snowfall is completely different, leading, thus, to a guaranteed high data traffic with one of the two link modes. For bad weather cases, a different link aggregation implementation will be necessary. In this case, the link mode showing a low data rate will be switched off.

6 Components from technical work packages demonstrated in WP5 experiments

The following tables summarize the components designed in each of the technical work packages that are demonstrated in the various experiments in WP5 to validate them in an integrated way.

6.1 Work package 2: Physical and link layer of 5G-Crosshaul

Table 47 lists the components developed in WP2 and the experiments of WP5 in which they are integrated and their performance evaluated.

Table 47: Physical and link layer components in experiments

Component	Experiment
Packetized fronthaul and backhaul mmWave mesh wireless solutions	7, 13, 14, 18
Optical wireless technologies	18
Hybrid mmWave+optical link	18
WS WDM-PON	15
Analogue radio over fibre technologies for 5G-Crosshaul	3, 15, 16
XFE: XPFE	1, 2, 4, 5, 6, 8, 9, 10, 11, 17
XFE: XCSE	17
Fronthaul compression	17

6.2 Work package 3: 5G-Crosshaul Control and Data planes

Table 48 and Table 49 lists the control and data plane components, respectively, developed in WP3 and the experiments of WP5 in which they are integrated and their performance evaluated.

Table 48: Use of XCI components in experiments

Component	Experiment
NFVO	5, 8, 9, 10, 11
VNFM	5, 8, 9, 10, 11
VIMaP	5, 9, 10, 11, 12
SDN controller	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
SDN controller/SBI driver	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
SDN controller/network core services	1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14
SDN controller/path computation	4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
SDN controller/path provisioning	2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
SDN controller/network reconfiguration	4, 5, 6, 7, 10, 12, 13, 14
SDN controller/analytics for monitoring	1, 2, 4, 5, 6
SDN controller/ Parent SDN controller	12, 13, 14
SDN controller/analytical algorithms	5

Table 49: Use of dataplane components in experiments

Component	Experiment
XPFE (enhanced Lagopus)	6, 8, 9, 10, 11, 15, 17, 18
XPFE Flow pipeline	6, 8, 9, 10, 11, 15, 17, 18
eNB with different split options	6, 14, 15, 18

6.3 Work package 4: Enabled innovations through 5G-Crosshaul

Table 50 lists the applications developed in WP4 and the experiments of WP5 in which they are integrated and their performance evaluated.

Table 50: Use of applications in demos and experiments

Application	Experiment
RMA for joint path computation/VNF placement	8, 9, and 11
RMA for joint Routing C-RAN functional split	14
EMMA for XPFE/XPU	1, 2, 4, 5 and 6
EMMA for mmWave Mesh Networks	7
EMMA for High Speed Train	3
MMA for Traffic Offloading	8, 9, 10, and 11
CDNMA	8 and 11
TVBA	9, 10 and 11

7 References

- [1] M. Chandramouli et al. “Monitoring and Control MIB for Power and Energy”, IETF RFC 7460, March 2015.
- [2] S. Tadesse, C. Casetti, C.F. Chiasserini, G. Landi, “Energy-efficient Traffic Allocation in SDN-based Backhaul Networks: Theory and Implementation”, IEEE CCNC 2017, Los Angeles, US, January 2017.
- [3] Ericsson Inc., “5G Energy Performance”, White Paper Uen 284 23-3265, April 2016.
- [4] V. López et al., Towards a Transport SDN for Carriers Networks: An Evolutionary Perspective, 21st European Conference on Networks and Optical Communications (NOC), 2016
- [5] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu et al., “B4: Experience with a globally-deployed software defined wan” in ACM SIGCOMM Computer Communication Review, 2013.
- [6] A. Pizzinat et. al. in J. Lightwave Technol., vol. 33, no. 5, pp. 1077-1083, March 2015.
- [7] Yoon Koo Kwon et. al. in Optical Fiber Communications Conference and Exhibition (OFC), paper Th3J.2, 2014.
- [8] L. Anet Neto et al. in European Conference on Optical Communications (ECOC), paper Th.2.P2.SC3.23, 2016.
- [9] “Small cell virtualization functional splits and use cases,” Small Cell Forum, document 159.07.02, 2016.
- [10] J. Zhang et al. in J. Lightwave Technol., DOI 10.1109/JLT.2016.2608138
- [11] X. Liu et al., in Optical Fiber Communications Conference and Exhibition (OFC), paper M2J.2, 2015.
- [12] H. Zeng et al. in European Conference on Optical Communications (ECOC), paper W.1.E.1, 2016.
- [13] B. G. Kim et al. in European Conference on Optical Communications (ECOC), paper W.4.P1.SC7.75, 2016.
- [14] Liu, Chang, Balasubramaniam Natarajan, and Hongxing Xia. "Small cell base station sleep strategies for energy efficiency." IEEE Transactions on Vehicular Technology 65.3: 1652-1661, 2016.
- [15] Common Public Radio Interface industry cooperation, Available at: <http://www.cpri.info/>
- [16] “Standard for Radio Over Ethernet Encapsulations and Mappings”, Available at: <http://sites.ieee.org/sagroups-1914/p1914-3/>
- [17] “Industry leaders agree to develop new CPRI Specification for 5G”, Available at: <http://www.cpri.info/press.html>, September 2016.
- [18] Bin Guo, Wei Cao, An Tao, and Dragan Samardzija, “LTE/LTE-A signal compression on the CPRI interface,” in Bell Labs Technical Journal, vol.18, no.2, pp.117-133, Sept. 2013
- [19] Shinobu Nanba and Akira Agata, “A new IQ data compression scheme for front-haul link in Centralized RAN,” in Personal, Indoor and Mobile Radio Communications (PIMRC Workshops), 2013 IEEE 24th International Symposium on , vol., no., pp.210-214, 8-9 Sept. 2013
- [20] Karl F. Nieman and Brian L. Evans, “Time-domain compression of complex-baseband LTE signals for cloud radio access networks,” Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE, Austin, TX, pp. 1198-1201, 2013.

- [21] DSL Forum, «TR-126, Triple-play Services, Quality of Experience (QoE) Requirements»
- [22] «TelecomKH», <http://www.telecomkh.com/es/internet/television/productos-y-servicios/productos-y-servicios/iptv/imagenio/jdsu/telefonica/3539>
- [23] State-of-the-art of Content Delivery Network, 2014, Department of Computer Science & Engineering, Uttar Pradesh Technical University United College of Engineering & Research, Allahabad, Uttar Pradesh, India
<https://pdfs.semanticscholar.org/dae5/91772c988239b70807315169b3526749d34b.pdf>,
- [24] <http://www.etsi.org/technologies-clusters/technologies/past-work/media-content-distribution>
- [25] 5G-Crosshaul. “Report on validation and demonstration plans”. Deliverable 5.1, October 2016.
- [26] J. Mangues et al. “Experimental Evaluation of Hierarchical Control over Multi-Domain Wireless/Optical Networks” Proc. of EUCNC’17, Oulu, June 2017.
- [27] D. King and A. Farrel. “A PCE-Based Architecture for Application-Based Network Operations”. IETF RFC 7491, March 2015.
- [28] Open Networking Foundation. “SDN in the Campus Environment.” ONF solution brief, September 2013.
- [29] Sterling Perring, principal analyst, “Understanding the SDN-Driven IP& optical Renaissance”, Nokia and Sedona Systems, Ltd, Available at: http://sedonasys.com/images/Downloads/White_Papers/SDN_Driven_Renaissance/SDN-Driven_IP_Optical_Renaissance.pdf
- [30] Carrier SDN: Service Provider Perspectives, Transition Strategis & Use Cases 2016: a heavy Reading multi-clinet study”, June 2016
- [31] L. Wang, Z. Lu, X. Wen, R. Knopp and R. Gupta, "Joint Optimization of Service Function Chaining and Resource Allocation in Network Function Virtualization," IEEE Access, 2016.
- [32] N. E. Khoury, S. Ayoubi and C. Assi, "Energy-Aware Placement and Scheduling of Network Traffic Flows with Deadlines on Virtual Network Functions," in IEEE CloudNet, 2016.
- [33] 5G-Crosshaul, “D4.1: Initial design of 5G-Crosshaul Applications and Algorithms”, Nov. 2016
- [34] M. Olsson, C. Cavdar, P. Frenger, S. Tombaz, D. Sabella and R. Jantti, "5GrEEen: Towards Green 5G mobile networks," 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Lyon, pp. 212-216, 2013.
- [35] G. K. Tran, H. Shimodaira, R. E. Rezagah, K. Sakaguchi and K. Araki, "Dynamic cell activation and user association for green 5G heterogeneous cellular networks," 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Hong Kong, pp. 2364-2368, 2015.
- [36] R. J. Weiler, M. Peter, W. Keusgen, H. Shimodaira, K. T. Gia and K. Sakaguchi, "Outdoor millimeter-wave access for heterogeneous networks — Path loss and system performance," 2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), Washington DC, pp. 2189-2193, 2014.
- [37] G. K. Tran, H. Shimodaira, R. E. Rezagah, K. Sakaguchi and K. Araki, "Practical evaluation of on-demand smallcell ON/OFF based on traffic model for 5G cellular networks," 2016 IEEE Wireless Communications and Networking Conference, Doha, pp. 1-7, 2016.
- [38] ns-3 LTE/EPC network simulator/emulator. Available at: <http://networks.cttc.cat/mobile-networks/software-tools/lena/>

-
- [39] 5G-Crosshaul partners CTTC, NEC and CNL presented the demo: Resource management of the 5G-Crosshaul at EUCNC 2017. Available at: <http://5g-crosshaul.eu/5g-crosshaul-demo-3-at-eucnc2017/>
- [40] J. Baranda, J. Núñez, I. Pascual, J. Manges, A. Mayoral, R. Casellas, R. Martínez, R. Muñoz, J. X. Salvat, A. García-Saavedra, X. Li, J. Kocur, “Resource Management in a Hierarchically Controlled Multi-domain Wireless/Optical Integrated Fronthaul and Backhaul Network” , 2017 IEEE 3rd Conference on Network Function Virtualization and Software Defined Networks, Berlin, Germany, 2017.
- [41] K. Nguyen, M. Golam, K. Ishizu, F. Kojima, “Empirical Investigation of IEEE 802.11ad Network”, 2017 Workshops of IEEE International Conference on Communications, Paris, France, 2017
- [42] "Open Networking Foundation", OpenFlow Switch Specification (Version 1.3.0), June 2012.
- [43] R.Santos, A. Kassler, “Small Cell Wireless Backhaul Reconfiguration using Software-Defined Networking”, 2017 IEEE Wireless Communication and Networking Conference, San Francisco, USA, 2017
- [44] D. Katz, D. Ward, “Bidirectional forwarding detection (BFD)”, 2010.
- [45] N.L.M., van Adrichem, B.J. Van Asten, and F.A. Kuipers, “Fast recovery in software-defined networks”, 2014 3rd European Workshop on Software Defined Networks (EWSDN), Budapest, Hungary, 2014.
- [46] J.Vestin, A.Kassler, “Low frequency assist for mmwave backhaul- the case for sdn resiliency mechanisms”, 2017 Workshops of IEEE International Conference on Communications, Paris, France, 2017.
- [47] 5G-Crosshaul, “D2.2: Integration of physical and link layer technologies in Xhaul network nodes”, October 2017.
- [48] 5G-Crosshaul, “D3.2: Final XFE/XCI design and specification of southbound and northbound interfaces,” Deliverable D3.2, October 2017.
- [49] 3GPP TR 38.801 V14.0.0 “Study on new radio access technology; Radio access architecture and interfaces” 03/2017
- [50] 5G-Crosshaul. “D4.2: Final design of 5G-Crosshaul Applications and Algorithms.” Deliverable D4.2, October 2017.
- [51] A. Mikityuk, JP. Seifert, O. Friedrich, “The virtual Set-Top Box: On the shift of IPTV service execution, service & UI composition into the cloud”, 2013 IEEE 17th International Conference on Intelligence in Next Generation Networks (ICIN)
- [52] Small Cell Forum, “Fronthaul Transport for Virtualized small cells ”, release 8.0, 10/2016.

8 Appendix 1: Current generic network provisioning procedure in a telco provider infrastructure

In this appendix, we present the generic procedure followed by a telco provider to provision network resources for a new connection in its infrastructure. Starting from some assumptions, the procedure will be presented together with some observations focused on quantitatively determining the amount of time required to carry out every step of the procedure. This analysis is the basis for the benchmark values proposed in Section 4.3.1, validating the results obtained for the proposed hierarchical multi-domain resource management of the crosshaul (Demo3).

8.1. Assumptions

- These are the assumptions that apply for setting up a single bidirectional communication when deploying a mobile service:
- Connection of an eNB to a SGW.
- The eNB is to be connected through a microwave (MW) link to a first IP/Eth aggregator device.
- The SGW is to be connected to an IP/Eth aggregator device.
- A new circuit between IP/Eth devices that require underlying optical transmission links has to be setup for providing connectivity.
- Starting point: all the cards are plugged and the cablings and cross-connections at the physical hardware equipment are already in place.

8.2. Procedure

Notice that the most disruptive operations (e.g. configuring new links in network equipment) are carried out during night periods and must be previously approved by a committee. If multiple departments are involved (e.g., optical transmission, IP), multiple such periods may be required. On the other hand, there are tasks that can be done previously to the disruptive configuration (e.g., identification of ports used in equipment involved prior to configuration). These are the steps followed in the production network of an operator to set up a multi-technology path equivalent to that set up in demo 3:

1. The IP department receives an order to setup a new service, e.g., the connection of an eNB with a SGW. *Normally, there is a queue of previous requests and the new request is not processed immediately after received. In this case, we consider 4 hours as an average value to process this request.*
2. The IP department checks the availability of ports to proceed with the link setup
 - 2.1. There is a planning process of identifying access ports (those for eNB and SGW, if not already connected) and network ports (those providing link continuity between the eNB and the SGW). *This identification process lasts for 2-3 hours, since it is not disruptive, it is done during regular working hours together with the associated documentation templates required to ask for permission for carrying out potentially disruptive network configurations.*
 - 2.2. If there are ports available, the ports are reserved for the end-to-end connection. *This reservation is introduced in a database in a timespan of minutes.*
 - 2.3. If there are no ports available, a network build process is launched (e.g., acquisition of a new card and corresponding pluggable components) and the link setup is kept on hold. *For the sake of a fair comparison with conditions of Demo3, this is not considered, as stated in the assumptions. However, this step is kept in the procedure for the sake of completeness.*

3. The IP department sends an order to the Transmission department to set up an optical circuit for supporting the new IP link. *This is the action derived from step 2.1. It takes 2-3 hours. This action will be done during night operative to minimize the impact on network performance upon approval of the network committee. In the best case, the nocturnal action window is scheduled for the same day the request is demanded, but this is not guaranteed. All the network operations are done upon approval of the network committee.*
4. The Transmission department checks where are connected both ends (i.e., both IP routers). *Step 4 and 5 takes last for 2-3 hours and it is done during morning operative together with the associated documentation to request a network action (i.e., configuration).*
5. The Transmission department identifies the route to be followed for connecting those devices, comparing alternatives for determining the best path
 - 5.1. A driver for the decision will be the selection of the best route for the end-to-end connection into consideration
 - 5.2. In case of no feasibility, the department would evaluate other less optimal routes with resources available. *For the sake of fair comparison with the scenario of demo 3, steps 5.2 and 5.3 are not considered for the time analysis. However, this step is kept in the procedure to keep it as complete as possible.*
 - 5.3. If no routes are available, a network build process is launched (i.e., acquisition of new transponders). *This is not considered in our comparison for the sake of fairness.*
6. The Transmission department launches the work orders for setting up the trails connecting the optical equipment for supporting the new IP link. *This is the action derived from steps 4 and 5. For the sake of fair comparison with scenario of demo 3, it is assumed that physical connection of fibres in patch panels are already done, and so, this step does not imply extra time.*
 - 6.1. The links are configured via Network Management System (NMS), hop by hop, testing and verifying they are operational. *This is done in minutes through the GUI of the NMS. However, since it is a potentially disruptive operation, this is also done during programmed night actions.*
7. The Transmission department confirms to the IP department that the transmission path is ready for service. *This is done in the following morning after step 6, together with the required documentation to ask for permission to configure the network. This step spans for 2-3 hours.*
8. The IP department configures the routers (usually via command line Interface (CLI)) for setting up the IP link. *This action is done at night in another scheduled window in about 2-3 hours.*
9. The circuit in the MW device is established via NMS for providing connectivity in the first transport radio hop between the eNB and the first aggregation device
 - 9.1. The circuit is verified. *This process could be simultaneous to other previous operations. However, as this step implies actions in the same network equipment (routers of Step 8), procedures of telco network infrastructure recommend to decouple such operations in a sequential manner to avoid network problems and for troubleshooting reasons. Hence, this is done in another nightly operational window and lasts for 2-3 hours.*
10. The IP department configures the access ports for establishing the end-to-end transport service (this includes the different logical interfaces, as well as the routing and/or L2 / L3 VPN configuration that could be required)
11. The end-to-end connection is verified
12. All the information is documented on inventory and other supporting systems and the link is declared operational. *Steps 10, 11, and 12 are done sequentially in 2-3 hours in another nightly operational window.*

8.3. Time analysis

With respect to the previous procedure, we can observe that the process of setting up a single connection implies several sequential steps involving different departments, which require of the appropriate documentation and the scheduling of operations during different nightly operational windows. For the sake of a fair comparison with results of Demo3, it is worth mentioning that the presented connection in Appendix 1 only assumes the connection between eNB and SGW, but in Demo 3, we presented the result of multiple connections, i.e., eNB connecting to other entities of the EPC, such as the MME. However, we can assume that the required connections can be grouped into the same request and its impact in the final setup time will not be very relevant.

From the above procedure, we can derive two values (a lower bound and an upper bound). The lower bound is an optimistic approximation based upon the time required strictly to perform the different actions and the associated documentation, without considering the extra delay incurred due to the required scheduling of disruptive network operations at night. In this case, the time to provision a connection in the production network of a network provider is around 20 hours.

On the other hand, the upper bound considers the scheduling of operations during nightly operational windows. In this case, we assume that operations are scheduled during the night immediately after the department receives the request, which, depending on the load may not be guaranteed. In this sense, our estimate would even be optimistic. According to this, completing the procedure would take 6 days.