5G Crosshaul
the integrated fronthaul/backhaul

H2020 5G-Crosshaul project
Grant No. 671598

D4.1: Initial design of 5G-Crosshaul Applications and Algorithms

Abstract

The deliverable (D4.1) reports the activities undertaken in Work Package 4 (WP4) in the first year of the project. It describes the definition and design of the applications and algorithms, the Northbound Interface (NBI) requirements exposed by 5G-Crosshaul Control Infrastructure (XCI), the implementation details and the validation and evaluation methodologies used in each application.

Document Properties

Document Number:	D4.1
Document Title:	**Initial Design of 5G-Crosshaul Applications and Algorithms**
Document Responsible:	UC3M
Document Editor:	Sergio Gonzalez
Target Dissemination Level:	Public
Status of the Document:	Final
Version:	1
Reviewers:	Thomas Deiss, Alberto Diez, Luis Contreras

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 Nº H2020-671598.

Table of Content

List of Contributors

Partnet No.	Partner Short Name	Contributor's name
P01	UC3M	Sergio González, Antonio de la Oliva
P02	NEC	Xi Li, Andrés García Saavedra
P05	ATOS	Jorge Rivas Sánchez, José Enrique González Blázquez
P11	VISIONA	David Jiménez, Danny R. Fonseca
P13	NXW	Giada Landi, Francesca Moscatelli, Marco Capitani
P16	FhG-HHI	Kei Sakaguchi
P17	CTTC	Ramón Casellas Ricard Vilalta, José Núñez, Josep Mangues
P18	CREATE-NET	Leonardo Goratti, Domenico Siracusa, Raihana Ferdous
P19	POLITO	Carla Fabiana Chiasserini, Claudio Ettore Casetti
P21	ITRI	Samer Talat, Sean Chang, Shahzoob Bilal Chundrigar, Chia-Lin Lai

List of Figures

List of Tables

List of Acronyms

Acronym	Description
5G	5th Generation
AP	Access Point
API	Application Programming Interface
BER	Bit Error Rate
BS	Base Station
BSS	Business Support Systems
CAPEX	Capital Expenditure
CDN	Content Delivery Network
CDNMA	Content Delivery Network Management Application
CIR	Committed Information Rate
CP	Control Plane
CPU	Central Processing Unit
DB	Data Base
DHCP	Dynamic Host Configuration Protocol
EIR	Extended Information Rate
EMMA	Energy-Management and Monitoring Application
ETSI	European Telecommunications Standards Institute
GUI	Graphical User Interface
GW	Gateway
HD	High Definition
HO	Handover
HST	High Speed Train
ICMP	Internet Control Message Protocol
ID	Identifier

ILP	Integer Linear Programing
INLP	Integer Non-Linear Programming
IR	Internal Report
KPI	Key Performance Indicator
MAC	Media Access Control
MANO	Management and Orchestration
MEC	Mobile Edge Computing
MMA	Mobility Management Application
MOS	Mean Opinion Score
MTA	Multi-Tenancy Application
NBI	Northbound Interface
NFV	Network Function Virtualization
NFVO	Network Function Virtualization Operator
NIC	Network Interface Card
OF	Open Flow
ONOS	Open Network Operating System
OPD	OpenDaylight
OPEX	Operational Expenditure
OS	Operating System
OTT	Over-The-Top
PoA	Point of Access
PoC	Point of Connection
PPP	Public Private Partnership
QoE	Quality of Experience
QoS	Quality of Service

RAN	Radio Access Network
REST	Representational State Transfer
RMA	Resource Management Application
RRH	Remote Radio Head
SDN	Software-Defined Networking
SLA	Service Level Agreement
SP	Service Provider
SQL	Structured Query Language
SUT	System Under Test
TCP	Transmission Control Protocol
TE	Traffic Engineering
TTL	Time to Live
TVBA	TV Broadcasting Application
UE	User Equipment
UP	User Plane
vCDN	Virtual Content Delivery Network
vEPC	Virtual Evolved Packet Core
VI	Virtual Infraestructure
VIM	Virtual Infrastructure Manager
VIMaP	Virtual Infrastructure manager and Planner Application
VLAN	Virtual Local Area Network
VM	Virtual Machine
VN	Virtual Network
VNE	Virtual Network Embedding
VNF	Virtual Network Function

VNFP	Virtual Network Function Provider
VNF-FG	Virtual Network Function Forwarding Graph
VNO	Virtual Network Operator
VNP	Virtual Network Provider
VoD	Video on Demand
WP	Work Package
XCF	5G-Crosshaul Common Frame
XCI	5G-Crosshaul Control Infrastructure
XFE	5G-Crosshaul Packet Forwarding Element
XPU	Crosshaul Processing Unit

Executive Summary

This deliverable shows the work carried out within the scope of Work Package 4 (WP4) of the 5G-Crosshaul project for the first 16 months of the project. This document is focused on the 5G-Crosshaul applications, more specifically in the application algorithms design, development, implementation, interaction and evaluation of each of these applications. Although this document provides a quite complete view of the applications and how do they relate to the rest of the Crosshaul architecture, they will be refined based on the experimentation results in D4.2.

This deliverable starts with a description on how the WP4 work fits in the overall architecture of the project. The applications running on top of the Crosshaul architecture provide the intelligence that is used by Crosshaul to provide added value services, which are not currently available in state of the art transport networks. The applications described in this document provide the following services:

- Resource Management
- Multi-tenancy
- Energy Management and Monitoring
- Content Delivery Networks
- Mobility
- Broadcasting

It is important to highlight that although the functionality provided by these applications is somehow present in current networks, there is currently no architecture supporting all these services on top of transport networks, in a coherent and integrated way.

The main focus of this document can be summarized as follows:

- Description in a clear and schematic way of the main algorithms that the applications will implement as well as the main modules involved for the application performance and the main scenarios considered is provided.
- The different applications need to serve the use cases defined in WP1. To this aim, the mapping of the applications and the WP1 use cases is presented in this report according to the functionality of the applications and the functional requirements of the use cases. This mapping will be used as the baseline reference for the validation and evaluation of the applications for the different use cases.
- Most of the applications have strong relations and interaction between them. One clear example is the Resource Management application, which is the central point for managing the underlying resources and thus requires knowledge of the decisions taken by all other applications to optimize the Crosshaul transport network. We present a detailed analysis of the different interactions between the applications in this document.
- The application design follows the Software Defined Networking paradigm, using a clearly defined set of APIs to interact with the brain of the Crosshaul architecture, the XCI, via its Northbound Interface (NBI). In this deliverable we not only provide a view on how these interactions are performed, but we also provide requirements for the definition of the XCI NBI, which is being agreed with WP3. Note that the XCI NBI design has also implications on the XCI SBI (provided by joint work between WP2 and WP3), since the functionality requested to the XCI must be applicable and implementable at lower layers.
- Regarding interfaces, we also consider that applications may provide services to other tenants or even third party applications that may work on top of Crosshaul “legacy” services. This document details the interfaces that external applications (or other XCI services) may use to interact with each application, describing in detail the abstract

primitives which must be supported by the applications' NBI, the messages, their senders and receivers as well as their parameters. We also include the mapping of these primitives in specific protocol messages.

- One of the key research aspects of the WP4 during the first 16 months has been the definition of an architecture, complying with NFV and SDN concepts, for multiple tenants sharing of the Crosshaul infrastructure. Although the data plane related mechanisms for multi-tenancy are dealt with in WP3, WP4 has taken the lead on the XCI multi-tenancy architecture. The main idea behind the proposed solution, is based on the concept of the recursive SDN control developed within the ONF. The major challenge with this approach is the need of a bookkeeping application that is able to orchestrate the resources available at the different physical and virtual recursive domains. We have addressed this challenge through the development of the Multi-tenancy application, which serves a dual purpose. On the one hand, it is in charge of bookkeeping the resources allocated to each tenant, while on the other hand it provides a copy of the NBI APIs defined for the XCI but considering only the resources allocated to the specific tenant. This architecture has been subject of strong discussions during months and we believe is one of the key achievements of the WP4 during this period. The details of this work have been reported in D1.1 [1], as a key contribution to the Crosshaul Multi-MANO architecture design in support of multi-tenancy.

In addition to the above mentioned aspects, during this period, substantial effort has been devoted to plan for the implementation and validation activities. Specifically, we report on this deliverable the description of the implementation details, the development status, the workflow and the roadmap of each application. We also define a standard structured methodology to perform validation tests over the different applications and algorithms to evaluate the defined several 5G-Crosshaul use cases in WP1 and their related scenarios. This document also contains the detailed evaluation plans and the mapping between the validation and evaluation tests and the project objectives and KPIs. Some initial evaluation results of the energy management application and the resource management application are also provided in this document.

Finally, it is important to note that all information is in line with the work developed in other work packages and aligned with the baseline system architecture defined in the deliverable D1.1 [1].

Key Achievements

The key achievements of WP4 during the first 16 months of the project reported in this document are summarized in the following.

The **first** important **achievement** presented is **an in-depth description of the design of seven Crosshaul applications** defined within WP4. This description focuses on the **definition of the applications** and **the main algorithms** that the applications will implement as well as the main modules involved for the application design and the main scenarios considered.

The **second** key **achievement** is a detailed **analysis of the different interactions between the applications**. The interaction between the applications and the workflows and interfaces to other applications (**applications' NBI**) to be taken into account is a key point that will allow harmonizing the whole application ecosystem.

The **third** key **achievement** is the specification of **initial requirements of the NBI of the XCI** with the agreement with WP3. The requirements are in terms of a list of required services to be consumed by the different Crosshaul applications, which need to be provided by the XCI. For each XCI service, this document provides a brief description and the list of consumers on the application side, and furthermore details the type of interaction and exchanged information, as expected at the north-bound interface of each XCI service. The mapping of XCI services on specific functional components at the control layer is addressed in WP3 [2].

Finally, **the plan for the implementation, validation and evaluation of all applications** under development in WP4 is presented. This deliverable provides a **detailed description of application implementation work** including high-level design, the development status, the workflow and the implementation roadmap of each application. The scope of validation and evaluation activity is to assure the proper behaviour of the functions and algorithms of the applications and more importantly to evaluate the performance of these applications for the validation of the project defined objectives and KPIs, in the different scenarios according to the defined WP1 use cases. This document **provides the detailed evaluation plans and the mapping between the validation and evaluation tests and the project objectives and KPIs**. **Some initial evaluation results** of the energy management application and the resource management application are also provided in this deliverable.

1 Introduction

The objective of this document is to present the initial design of the 5G-Crosshaul applications and algorithms defined in the scope of WP4. Each application is presented detailing the design, the implementation procedure and the evaluation and validation tests that will be performed. This document introduces the interaction between the applications and the interaction with the 5G-Crosshaul Control Infrastructure (XCI). Figure 1 shows the scope of WP4 in the 5G-Crosshaul architecture:

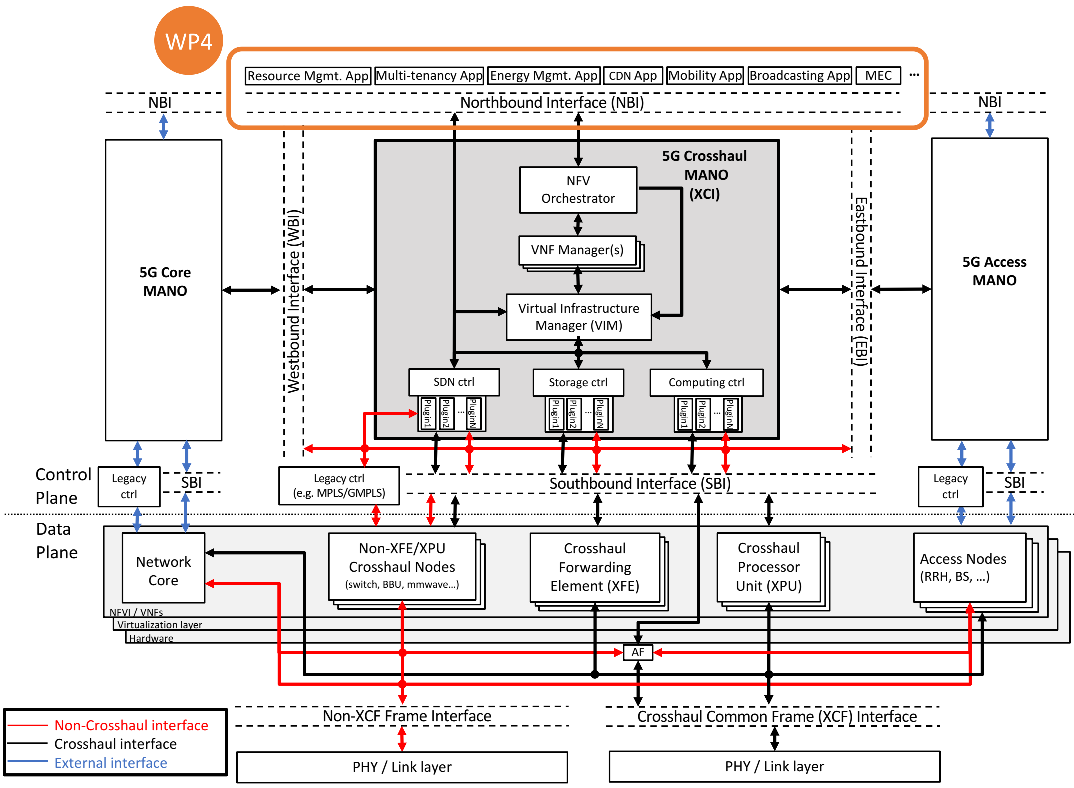


Figure 1: Scope of Work Package 4

This document consists of 7 sections and 5 appendixes organised as follows:

- Section 2 describes an overview of the 5G-Crosshaul architecture and the components.
- Section 3 describes the design of the 5G-Crosshaul applications and the mapping of these applications to the use cases defined in WP1.
- Section 4 explains the interaction between the applications, describing the workflows and the application layering and how the XCI behaves when there are multiple tenants providing services on top of a single 5G-Crosshaul infrastructure operator.
- Section 5 is focused on the interactions between the applications and the XCI including the NBI requirements and the XCI services.
- Section 6 describes the methodology of validation and evaluation, emphasizing the scenarios, the project KPIs and the mapping between each application test and the project objectives.

- Section 7 provides the conclusions.
- Appendix I show the interaction of the 5G-Crosshaul applications through its Northbound Interface (NBI), describing the NBI model and specification.
- Appendix II describes the implementation and development details of each application, the functionalities, a high level design overview and the implementation roadmap.
- Appendix III is focused on the evaluation and validation of the applications, describing the roadmap, the validation procedure and environment and in some applications the first result obtained.
- Appendix IV details the EMMA algorithm for mmWave meshed network.
- Appendix V explains the Centralized Network and Energy Management (CNEM) for Radio-over-Fibre.

2 Overview of the 5G-Crosshaul architecture

5G-Crosshaul provides an adaptive, programmable and cost-efficient substrate to accommodate advanced services on top of it. Different kinds of services may be provided by the owner of the 5G-Crosshaul to its clients based on the degree of control offered to operate the network infrastructure, and the mechanisms used to provide services. Thus, there will be services deployed that will request network resources in an agnostic way, meaning that, the infrastructure owner will have the control of the network and the service provider will just manage the *Service Level Agreements* (SLAs). Vice versa, for other services, the network operator will have limited or even full control of the allocated virtual infrastructure depending on specific situations, and will manage their own virtual resources.

In the context of 5G-Crosshaul, resources encompass communication technologies, storage and compute and hence are of different nature. The 5G-Crosshaul system architecture is shown in Figure 2. The architecture highlights that Crosshaul specific switching and processing elements, respectively the Crosshaul Forwarding Elements (XFE) and the Crosshaul Processing Unit (XPU), are the key components of the network fabric at the user plane level. The information is routed over the Crosshaul transport network using the Crosshaul Common Frame (XCF). An Adaptation Function (AF) is also used to integrate in the new intelligent transport network legacy and non-Crosshaul network elements. Crosshaul resources are controlled by the XCI and are managed by the Management and Orchestration (MANO) layer. The MANO is compliant with the European Telecommunications Standards Institute (ETSI) reference architecture. In particular, a specific MANO is used for the 5G-Crosshaul transport network, whilst dedicated distinct MANOs are in charge of the core and access networks.

The architecture shown in Figure 2 provides high flexibility and possibilities to use the transport network resources, which allows decentralizing specific functions in the XPUs and disseminating intelligence throughout the whole transport system. Doing a parallel with the Cloud Radio Access Network (C-RAN) architecture [3], XPUs include some functions of the Base Band Units (BBUs). Indeed, functions can be dynamically instantiated in the XPUs elements and moreover can be moved at different locations under the management of the MANO. XFEs instead are multi-technology switches in order to meet different application constraints and to suit the different scenarios which have been envisaged for 5G [1]. 5G-Crosshaul applications are used to manage and optimize the network to the greater extent possible. Resources are managed dynamically and planned without resorting to expensive pre-provisioning, while at the same time energy and mobility of users that may cause the sudden appearance of traffic peaks is part of a multi-objective optimization problem. Optimization of specific aspects of the network behavior is done by applications such as the Resource Manager Application (RMA), Mobility Management Application (MMA), Energy Management and Monitoring Application (EMMA), Virtual Infrastructure Manager and Planner Application (VIMaP), Content Delivery Network Application (CDNA) and TV Broadcasting Application (TVBA).

An important feature of the 5G-Crosshaul network is to support multi-tenancy through the Multi-Tenancy Application (MTA). Tenants can allocate network, storage and compute resources within the transport system. An important distinction between the tenants depends on the level of control that each tenant has on the respective resources, as mentioned above. Herein three main service use cases of interest which have been identified are discussed: Over-the-Top (OTT), Mobile Virtual Network Operator (MVNO) and Mobile Network Operator (MNO). Each case allows to study different levels of control on network resources. The mapping of the three different service use cases within the general 5G-Crosshaul architecture is provided in Figure 3.

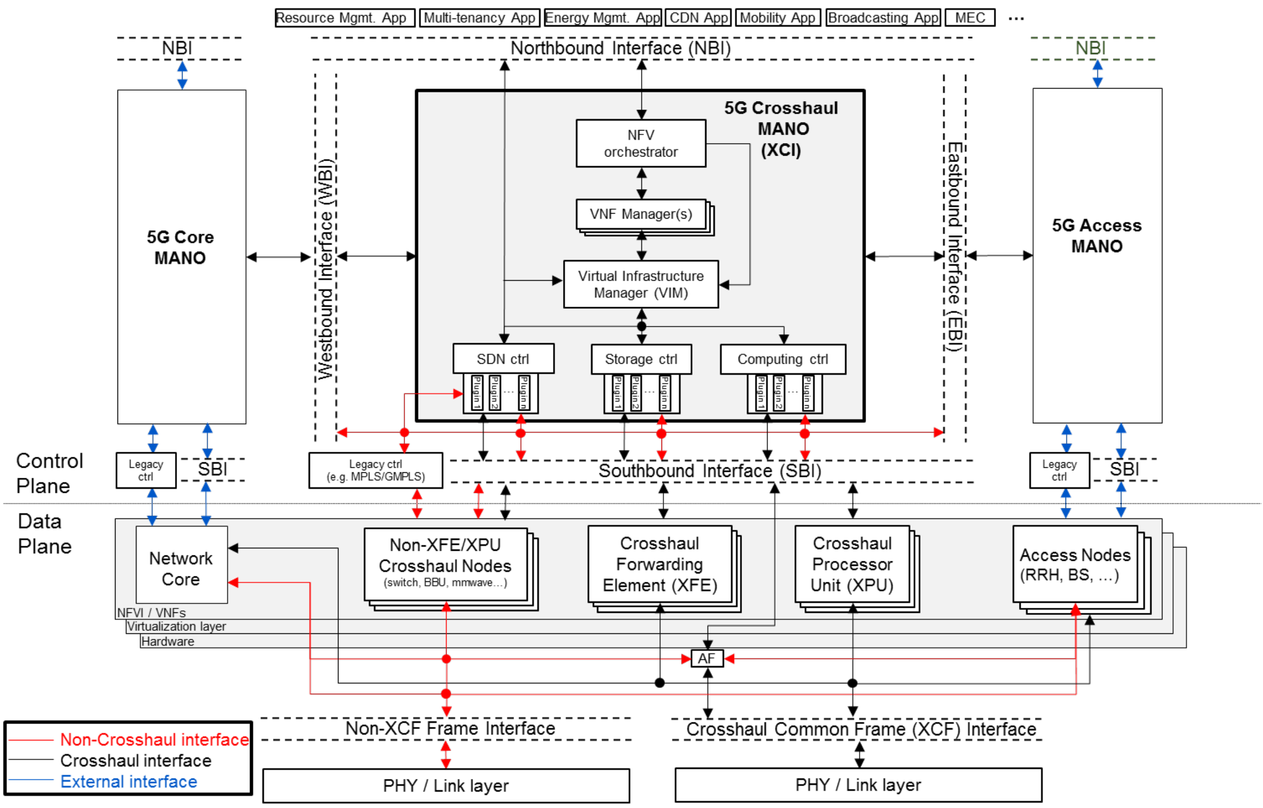


Figure 2: 5G-Crosshaul architecture

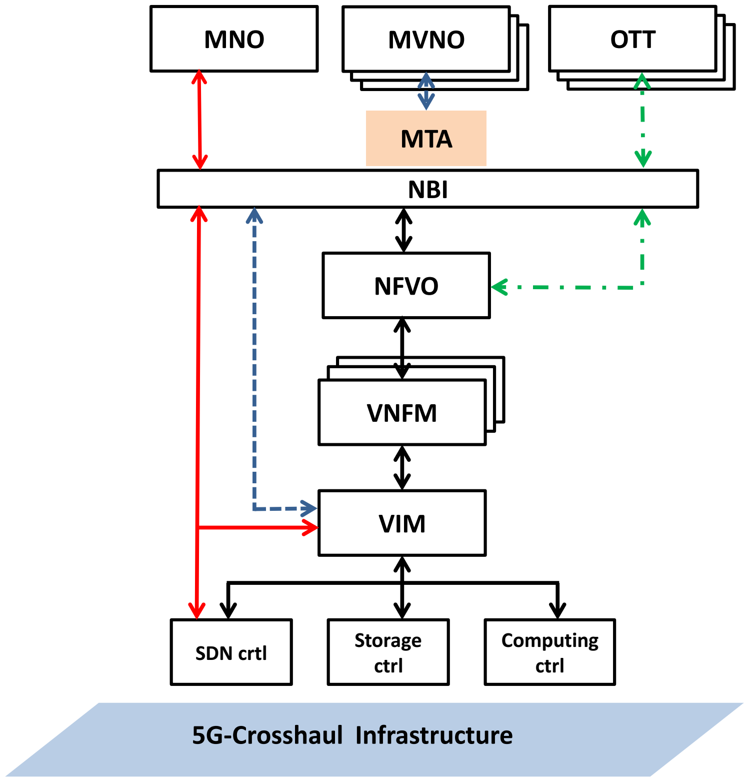


Figure 3: Mapping of three different service use cases (OTT, MVNO and MNO) onto the 5G-Crosshaul system architecture

3 Description and design of 5G-Crosshaul applications

3.1 Multi-Tenancy Application

3.1.1 Description of MTA

Multi-tenancy is a desired feature of 5G-Crosshaul to enable a generalized, flexible sharing of physical infrastructure resources among multiple tenants, e.g., multiple Virtual Network Operators (VNOs) or service providers like OTT operators offering network services like CDN, TV broadcasting. The Multi-Tenancy Application (MTA) is the application on top of the 5G-Crosshaul XCI, which aims to enable flexible sharing of 5G-Crosshaul physical resources (networking, computing, and storage) among different tenants. The main objective of MTA is to reduce the costs (i.e., CAPEX and OPEX) by sharing the infrastructure resources and maximize their utilization in a cost-efficient manner, while at the same maximizing the energy efficiency.

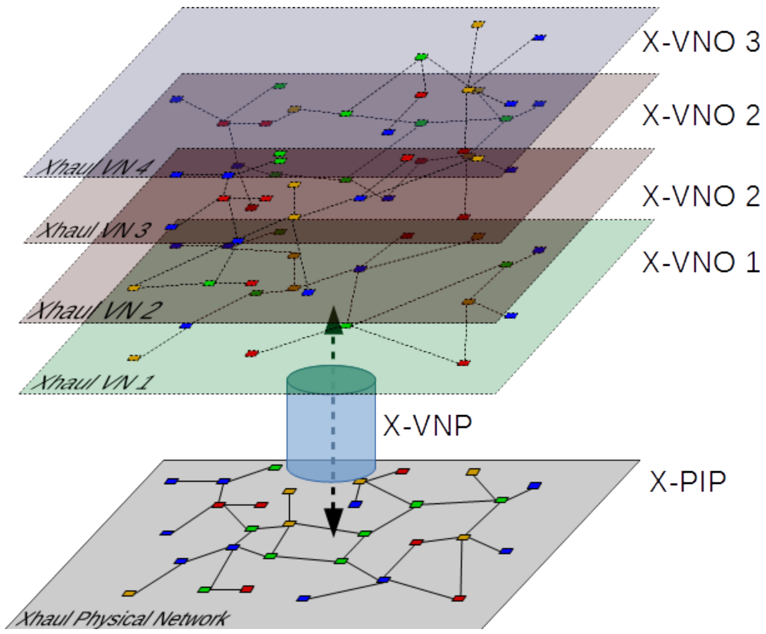
<table>
<tr><td>Title</td><td>Multi-Tenancy Application (MTA)</td></tr>
<tr><td>Description of the Application</td><td>In MTA, a Crosshaul scenario involves the following stakeholders [1]:

Crosshaul Physical Infrastructure Provider (X-PIP), who owns and manages the physical infrastructure including both network and computing resources.
Crosshaul Virtual Network Provider (X-VNP), responsible for assembling virtual resources from one or multiple PIPs into a single virtual topology providing a logical abstraction of the Crosshaul infrastructure.
Virtual Network Operator (VNO), which maintains and configures a virtual network over the virtual topology provided by a VNP according to the needs of a Service Provider (SP).
Service Provider (SP), which uses a virtual network to offer a service (e.g. video delivery). One VNO can set up multiple virtual networks for different services.

A Virtual Network Provider (VNP) often leases resources from one or multiple Crosshaul Physical Infrastructure Providers (PIPs) who owns the physical resources and assembles these physical resources into a virtual network infrastructure. The VNP provides virtual resources over its substrate network to multiple network operators (Virtual Network Operators, VNOs), where each of them owns a slice of the physical resources (computing resources and networking resources). Network virtualization is applied to provision multiple virtual networks (i.e. multiple tenants) over a substrate network for sharing the computing and network resources. A virtual network (VN) is a logical topology composed of a set of virtual nodes (e.g. virtual routers and switches) interconnected by virtual links over the substrate network.

Multi-tenancy application shall provide the following features:

Resources are provided concurrently and seamlessly to the tenants.
Virtual domains are isolated per tenant, i.e. each VNO shall use the complete addressing space, and deploy their choice of network OS and set their own (virtual) topology. In some use cases they may even own their own controllers to manage their own virtual resource inside
</td></tr>
</table>

	each tenant. • Virtual slices of resources are allocated dynamically, allowing the network to scale to multiple tenants without service disruption to existing VNOs.
Picture	 *Figure 4: Virtual network deployments over a common physical substrate*
Business or technique drivers	Multi-tenancy is a key enabler to maximize the degree of exploitation (and thus, revenue) of physical resources, e.g. by allowing the exploitation of networking services by a virtual operator in a location where the original (physical) operator has lower load requirements. In this way, the overhead of physical equipment maintenance is minimized and its utilization is maximized. Besides, both PIP/VNP and VNO are benefited by this application: • PIP/VNP gains the flexibility on provisioning virtual network slicing with different features to VNOs to meet their individual requirements. • VNO’s virtual network can achieve optimal utilization by dynamically scaling up/down feature to meet their dynamic requirements on load, coverage and topology.

Use cases and Scenarios

(1) Use case 1: Dynamic allocation of virtual resources

When receiving a tenant request from a VNO for requiring a slice of Crosshaul resources, MTA computes and allocates virtual slices of resources to VNOs according to their requirement of KPIs (such as CPU/memory capability, link capacity, latency, jitter, loss). When one or more VNOs request slicing updates, or when new tenants join the common substrate (Figure 5), the MTA can reshuffle and/or reconfigure dynamically the virtual resources seeking a global optimum solution. This requires the admission control policy to accept or reject the request on demand.

Figure 5 illustrates the required procedure:

- A VNO sends a request for a network slice to VNP; the request contains the description about the desired Virtual Network (VN), e.g. topology.
- VNP computes the optimal slice to meet VNO requirements.
- VNP allocates and instantiates a slice with the resource required by VNO.
- VNP sends the reply to the VNO with the allocated slice.

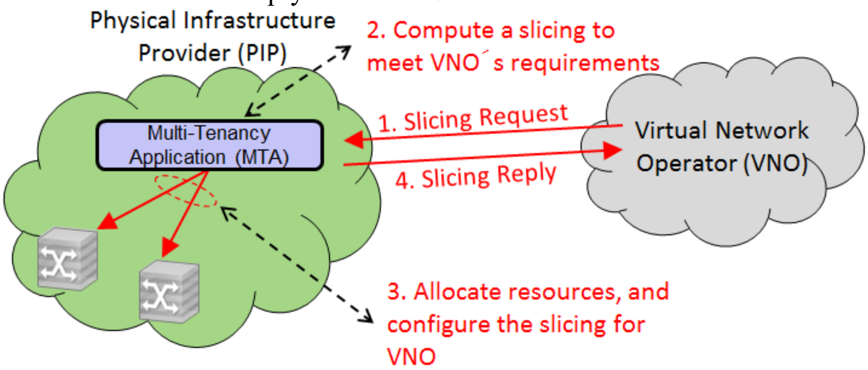


Figure 5: Dynamic update of a VNO slice on-demand

The following functions shall be implemented in MTA:

- **Admission control policy**: when VNP gets a request from a VNO for setting up a virtual network, they need to decide whether to accept it or not, depending on whether the substrate network has enough resources available to satisfy the demand of this tenant.
- **Virtual network deployment** (creating virtual network slices, Figure 4): namely finding a specific subset of nodes (node mapping) and links (link mapping) in the substrate network to create a virtual network for a tenant which satisfies its resource demand, known as the virtual network embedding (VNE) problem [4]. To solve the VNE problem, the following algorithms shall be explored:
    - Efficient routing and load balancing algorithms over the physical infrastructure to guarantee the resource demand per VNO.
    - Scheduling and allocation of transmission, CPU and memory resources.

- Placing virtual network functions: function placement (both BH and FH) should be optimized to favour centralization when possible and maximize the resource utilization efficiency. This requires network function virtualization support from NFVO and VNF manager functions in XCI.
- Joint RAN + Crosshaul resource orchestration: we can extend the multi-tenancy application to go beyond the Crosshaul domain by jointly allocating RAN and Crosshaul resources for each tenant according to their individual requirements. In another word, each tenant is assigned to a slice of RAN and Crosshaul resources with the joint consideration of RAN and Crosshaul domains. The objective is to gain maximum overall resource utilization while considering the load distribution, RAN and Crosshaul performance constraints, as well as the availability of physical resources (e.g. radio and transport resources, computing resources in the cloud nodes).

- **Per-Tenant Monitoring:** monitoring of network QoS (e.g. throughput, delay, loss) and resource usage per tenant.

(2) Use case 2: Own resource management by the tenant

In this use case, VNO is able to manage and optimize the resource usage of its own virtual resources. That means, we allow each tenant to manage their own virtual resources inside each tenant. So we will require a per-tenant controller or per-tenant MANO (XCI) approach. In case of one MANO (XCI) per tenant, this will result in multi-MANO (XCI) architecture. This use case will not be studied in this project.

(3) Use Case 3: Multi-tenancy sharing in high speed train scenario

For moving infrastructure as a high speed train, since end users taking a high speed train belonging to different Virtual Network Operators (VNOs) will move at a very high speed, if end users inside a high speed train are only connected to outdoor macro base stations, they may suffer from bad Quality of Experience (QoE) due to the obstacles like Doppler Effect and penetration loss caused by metallic carriage of a train. Therefore, in order to provide a good coverage and better QoE to the end users, constructing a sharing environment in a high speed train serving the end users from different VNOs will be a good solution. The following description will consider two special cases for multi-tenancy sharing in a high speed train.

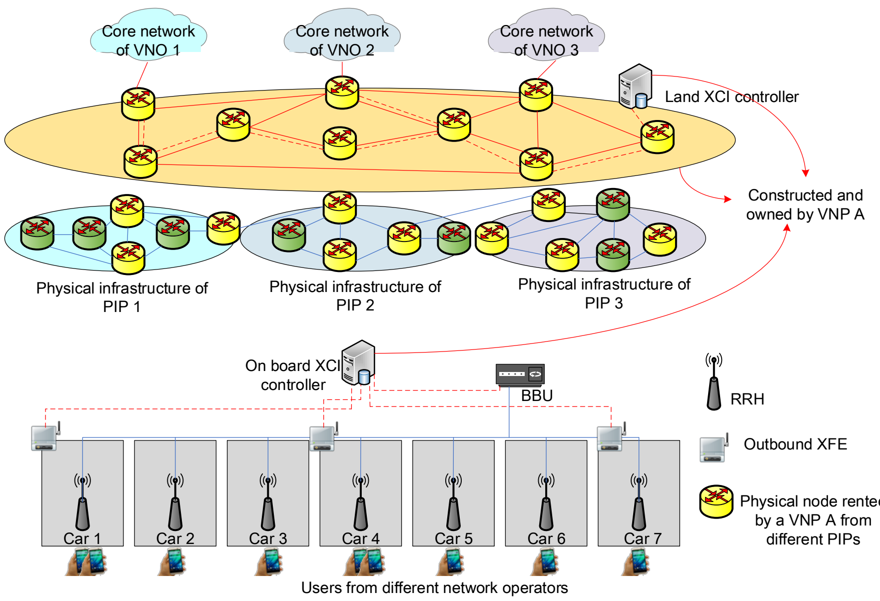
On-board multi-tenant sharing environment:

As shown in Figure 6(a) and Figure 6(b), inside a high speed train, a third party (i.e., VNP A) negotiates with a PIP to deploy the on board multi-tenancy sharing environment such as deploying a BBU and multiple RRHs in each car of a high speed train to construct a sharing Fronthaul (FH) network (i.e., Figure 6(a)) or deploying a small cell in each car to create a sharing 5GPoA access network (i.e., Figure 6(b)) to the end users of different VNOs. No matter which deployment (either Figure 6(a) or Figure 6(b)) to provide multi-tenancy sharing in a high speed train, an on board XCI controller deployed by VNP A will manage and allocate the available network resources on board to different VNOs, i.e., responsible for virtual network deployment for each tenant on board. Furthermore, all traffic from end users to the internet will only be transmitted through the outbound XFE to the Crosshaul Backhaul network using Crosshaul Control Frame (XCF).

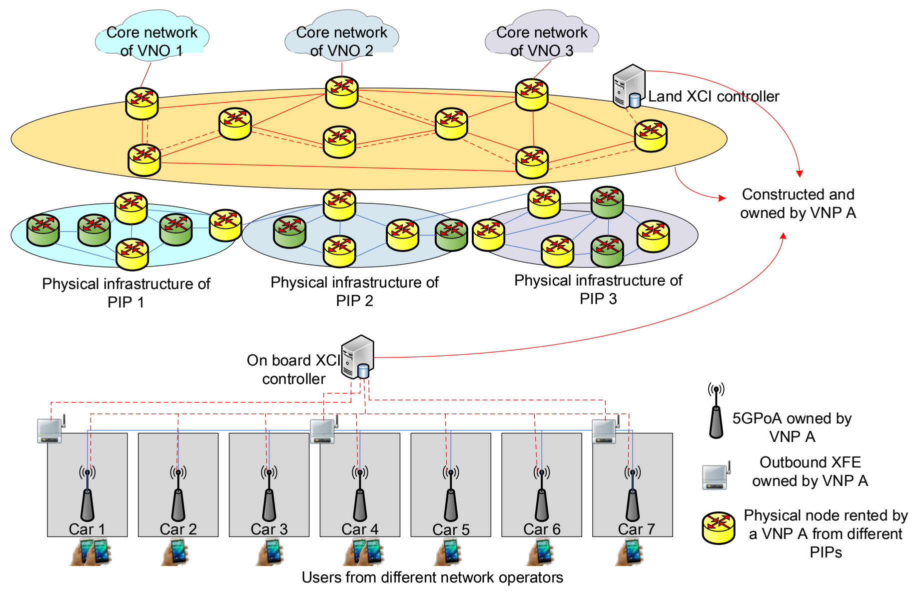
Backhaul network sharing environment:

For the traffic to the Backhaul network (i.e., from outbound XFE to backhaul network), it can be seen that VNP A can negotiate and rent the physical nodes (colored in yellow in Figure 6) from different PIPs. A yellow node (i.e., a physical node) can be regarded as router or switch. Based on this, VNP A constructs a physical substrate with multiple physical nodes rented from different PIPs (as seen the orange-colored part in Figure 6). Both of the traffic from outbound XFE in a high speed train and that transmitted in the physical network follows XCF format. Similar to on board sharing environment, a Land XCI controller is deployed to optimize the utilization of the network resource in backhaul network.

VNP A can use the functions provided by MTA to offer the required resources for each tenant involved in the sharing environment as mentioned in the above description (i.e., admission control policy, virtual network deployment and Per-Tenant Monitoring). In other words, each tenant can get the isolate network slice with requested resources for on-board and backhaul network sharing as exemplified in Figure 5.



(a) Scenario 1: Land backhaul sharing and onboard fronthaul sharing

<table>
<tr><td></td><td colspan="2">


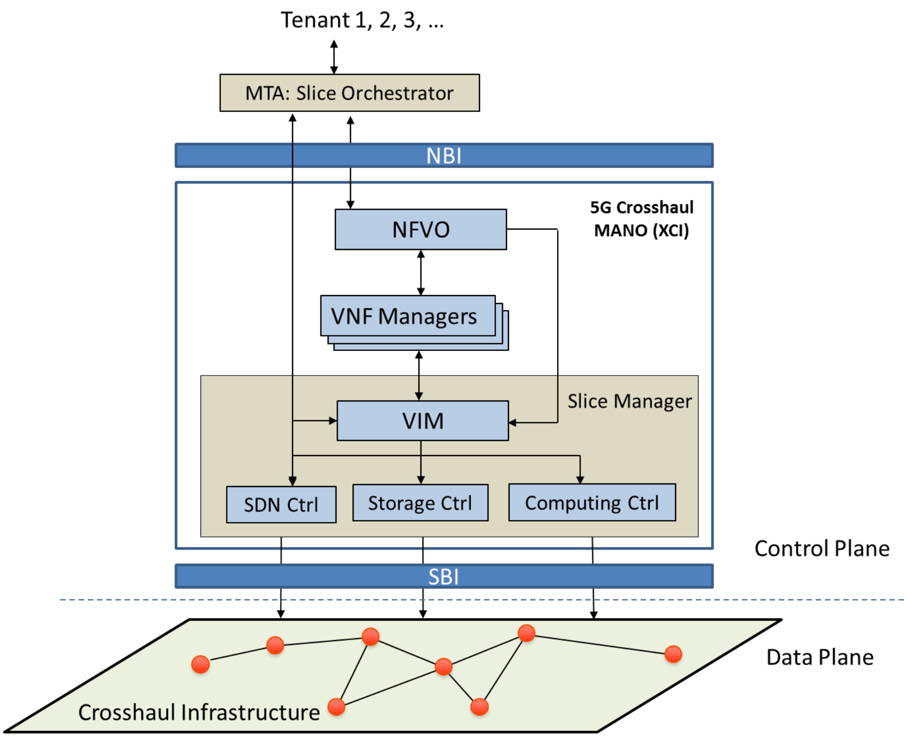
(b) Scenario 2: Land backhaul sharing and onboard small cell sharing

Figure 6: Multi-tenancy sharing for high speed train scenarios
</td></tr>
<tr><td>Technologies to be used or considered</td><td colspan="2">Network virtualization technique needs to be explored to create virtual networks for multiple tenants. Optimization techniques will be considered for resource orchestration.</td></tr>
<tr><td rowspan="2">Information requirements

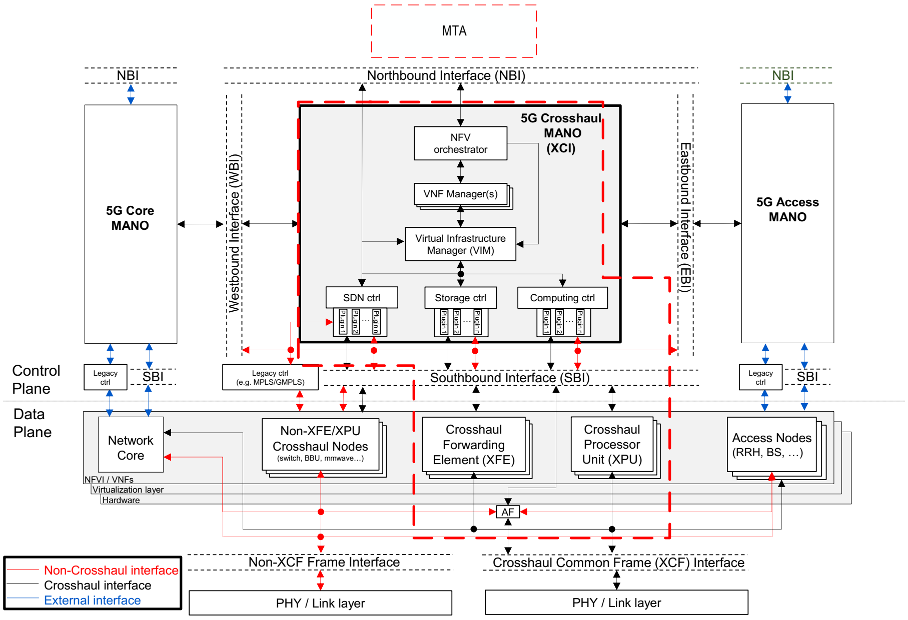
(Inputs)</td><td>What</td><td>Who may provide this</td></tr>
<tr><td>Abstraction of Crosshaul physical/virtual infrastructure:
List of physical Crosshaul nodes.
List of physical Crosshaul links.
Each physical link has associated a profile:
Type of link (and profile associated)
Wireless (microwave, mmWave, etc)
Wired (Ethernet, fiber, etc)
Set of channels
Set of transmission powers
Endpoint nodes and ports [node Id/port Id]
Link capacity [Mb/s]
Estimation on max, min, mean, and variability of link delay [usec]
Estimation on mean packet loss rate [%]</td><td>XCI provides information on physical infrastructure (through NBI)
On board XCI provides information on the physical infrastructure in the train (through NBI)
VIMaP can provide virtual infrastructure information</td></tr>
</table>

	• Each physical node has associated a profile: ○ Type of node (and profile associated): ○ BBU, RRH, Small cell, outbound XFE ○ Router, switch, ROADM (optical nodes), etc. • List of ports ○ Computing capacity [op/s] ○ Memory capacity [bytes] ○ Physical location [GPS coordinates]	
	Abstraction of Virtual Network request: • Virtual node indexes. • Virtual link information is provided as an adjacency matrix of size NxN where N is the number of virtual nodes. An entry equal to 1 indicates the existence of a direct bidirectional link, 0 if there is no direct link. • Each virtual link request has associated a network demand: ○ Max, min, mean and variability of traffic load [Mb/s] ○ Estimation on max, min, mean, and variance of link delay [usec] ○ Max packet loss rate tolerated [%] • Each virtual node has associated a computing demand: ○ Max, min, mean and variability of computing resources required [op/s] ○ Max, min, mean and variability of memory requirement [bytes]	• User of the application (e.g. operator or service provider)
	Monitoring of physical resources: • Estimation on each physical link of: ○ Max, min, mean and variability of throughput load [Mb/s] ○ Max and variability of delay experienced [usec] ○ Mean packet loss rate [%] • Estimation on each physical compute node	• XCI and VIMaP may provide information on the physical infrastructure (through NBI)

<table>
<tr><td></td><td>of:
o Max, min, mean and variability of computing resources used [op/s]
o Max, min, mean and variability of memory used [bytes]</td><td></td></tr>
<tr><td rowspan="2">Outputs</td><td>What</td><td>Who may need this</td></tr>
<tr><td>• Virtual network mapping per VN:
• Mapping virtual to physical node
• Link mapping. A virtual link is mapped into a physical path:
o Routing rules to create paths (virtual links)
o Bandwidth reservation on each physical link [Mb/s]
• Computing resources requirement on physical nodes [op/s]
• Memory requirement on physical nodes [bytes]</td><td>VIMaP is used to configure the mapping between virtual and physical resources
The SDN controller in the XCI may enforce part of this as well, e.g. Service Functions Chaining (SFC)</td></tr>
<tr><td></td><td>Per-Tenant Monitoring: monitoring of network QoS (e.g. throughput, delay, loss) and resource usage per tenant.</td><td>Other applications who requires per tenant status</td></tr>
<tr><td>App-specific functions</td><td colspan="2">• Virtual network orchestration functions
o Node mapping: mapping virtual nodes to physical nodes
o Link mapping via routing, load balancing over physical infrastructure network, this can be computed by VIMAP or RMA applications.
o QoS differentiation, scheduling, traffic shaping
o Optimization of placement of virtual network functions
o Allocation of required networking, as well as CPU and memory resources for each tenant
o Admission control function to admit the request for creating a virtual network (for each tenant)
o Per-tenant monitoring: monitoring of network QoS (e.g. throughput, delay, loss) and resource usage per tenant</td></tr>
<tr><td>Required
(external)</td><td colspan="2">• Virtual Infrastructure Manager and Planner (VIMaP) is needed to provide the abstraction of underlying physical and virtual network infrastructure
• Allocation of physical resources, e.g. through VIMaP application</td></tr>
</table>

functions	○ Bandwidth reservation on each physical link ○ configuration of routing to set up paths for virtual links ○ Enforcement of traffic scheduling, shaping, QoS provisioning, etc. • Management of virtual functions through XCI, for instance ○ Network Functions Virtualization ○ Service Function Chaining • Monitoring of physical network metrics from the XCI: link delay, throughput, jitter, packet loss, consumed computing resource memory resource, etc.
Function Architecture	 *Figure 7: Multi-Tenancy application function architecture* The MTA requires two main components in the function architecture: • **Slice Orchestrator**: Multi-tenancy application is the slice orchestrator, which makes the logical decisions on slicing physical resources to provide virtual networks, each owning a slice of the physical resources. • **Slice Manager**: This function is located inside the XCI to support MTA to realize the multi-tenancy in the control plane, responsible for managing and controlling physical resources (computing, storage and networking resources) to instantiate and configure virtual networks (i.e. mapping the virtual resources onto the physical resources), to execute and enforce the decisions made by the MTA/slice

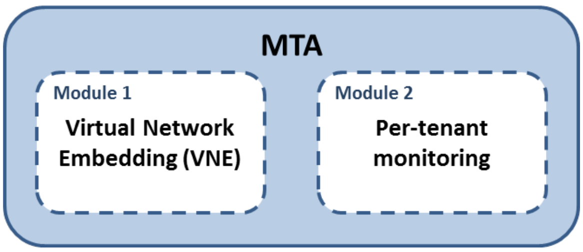
<table>
<tr><td></td><td>orchestrator.</td></tr>
<tr><td>Implementation</td><td>An analytical and simulative approach will be chosen as initial study.</td></tr>
<tr><td>Relation to other applications</td><td> (1) Virtual Infrastructure Manager and Planner (VIMaP) is used for (i) planning physical or virtual infrastructure; (ii) instantiating VMs and Network connectivity services; (iii) allocating Virtual Infrastructures composed of virtual compute nodes (hosts) and virtual XFEs; (iv) provision and configure IT and networking resources for the tenants by providing mapping between the physical resources and the virtual networks. (2) Supporting all other applications which are tenant based (CDNMA, MMA and TVBA) </td></tr>
<tr><td>Application mapping on Crosshaul system architecture (High level view)</td><td> The MTA requires the support from the XCI components to configure the mapping of virtual infrastructure to the physical one and allocate the corresponding computing, networking and storage resources; as well as requires the support from the data plane to identify the tenants.



The 5G-Crosshaul XCI offers two main MTA services:
1. The deployment of Network Services (NS) as defined by the ETSI NFV architecture, i.e. to create a NS with specific virtual network functions (VNFs), somehow interconnected as defined in the VNF-Forwarding Graph. It is ETSI/ NFV specific, and provides an upper level view of a virtual application. In this case, a tenant is an administrative entity that owns one or more Network Services offered through the XCI MANO Interface. In other words, the XCI MANO keeps the state of the different </td></tr>
</table>

	NS instances and which tenant has allocated them. It is thus part of the MANO (from the NFVO to the VIM) to check access rights, assign resource quotas and provide efficient means for the resource partitioning and isolation. 2. The deployment of Virtual Infrastructure (VI), which is defined as a coherent set of heterogeneous network, compute and storage infrastructure, abstracted and virtualized from the underlying physical infrastructure, and which is composed of e.g. virtual hosts interconnected by network slices. In this case, A tenant is an administrative entity that is able to instantiate one or more virtual infrastructures. It is part of the actual service control model to define the degree of control over the virtual infrastructure. In WP4 we are focused on the second service for the design and implementation of Multi-Tenancy Application (MTA) that is above the XCI, while that the first service will be offered by the XCI (more specifically by the NFVO and VIM components).

3.1.2 Design of MTA Algorithms

Definition of basic MTA algorithms	The MTA application has essentially the following tasks: • Solve the Virtual Network Embedding (VNE) problem. VNE takes care of the allocation of virtual resources both in nodes (XFEs, XPUs) and links. A dynamic mapping of virtual resources onto a physical hardware for multiple Virtual Network Operators (VNOs) should lead to maximize the utilization of the underlying substrate. • Admission Control. MTA should be able to reject VNO embedding requests that may lead to violating an SLA or a waste of hardware resources. • Per-Tenant Monitoring: MTA shall provide individualized QoS information such as resources utilization such as load, latencies and losses for each of the VNOs or tenants. • Tenant identification: MTA is responsible for identifying the tenant (e.g. assigning a tenant ID) and providing the mapping of a tenant to a network slice. To meet the above tasks, MTA is composed of two modules: VNE module and Monitoring module.  The following list describes the variables used in the high-level description of the algorithms of each of the modules:

Variable	Description
$G_r(V_r, E_r)$	A graph representing the virtual network requested by the VNO. It contains a set V_r with the vertices of the graph representing forwarding nodes (XFEs) and computing nodes (XPUs) of the virtual network, including the computational resources required for each of those nodes. It may also contain statistics of expected resource utilization. It also contains a set E_r with the edges of the graph representing the links of the virtual network, including the network capacity (throughput load) and latency requested for each of those links. It may also contain statistics of expected utilization and link QoS (delay, losses).
$G_p(V_p, E_p)$	A graph representing the physical 5G-Crosshaul network. It contains a set V_p with the vertices of the graph representing forwarding and computing nodes of the physical network, including the maximum and residual computational capacity of each of those nodes. It also contains a set E_p with the edges of the graph representing the links of the 5G-Crosshaul physical network, including the maximum and residual network capacity and latency of each of those links.
$G_v^i(V_v^i, E_v^i)$	A graph representing the virtual network actually allocated to the VNO i in V. It contains a set V_v^i with the vertices of the graph representing forwarding and computing nodes of the virtual network i, including the statistics of the computational resources utilized by this VNO on each of those nodes, and the mapping to physical nodes. It also contains a set E_v^i with the edges of the graph representing the links of the virtual network i, including the statistics of the network resources utilized by this VNO on each of those links, and the mapping to physical paths (set of physical links).
G	Set containing all virtual graphs G_v^i mapped onto the physical 5G-Crosshaul network.
V	Set of VNO identifiers

- **Module 1. Virtual Network Embedding (VNE)**

Function: Find the optimum mapping between a virtual network graph provided by a VNO or tenant onto the physical network graph abstraction of the 5G-Crosshaul transport network. If the mapping violates the physical capacity constraints or it leads to unacceptable revenue (GetReward() function in the below algorithm), the request is rejected; otherwise the request is accepted.

Algorithm 1 Virtual Network Embedding

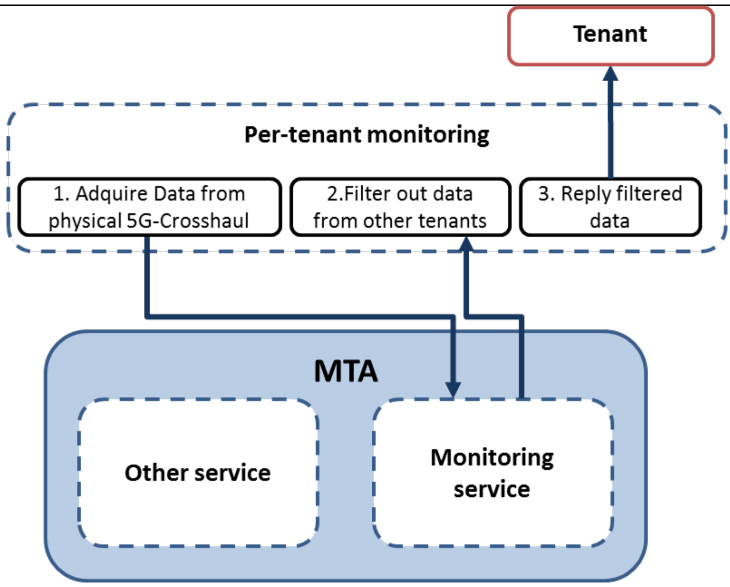
procedure VNE(G_r)
  for each $i \in \mathcal{V}$ **do**
    G_v^i = `MonitorTenant(`i`)`
  end for
  G_p = `UpdateCrosshaul()`
  j = `UniqueID(`$\mathcal{V}$`)`
  V_v^j = `NodeMapping(`$V_r \in G_r$`)`
  E_v^j = `NodeMapping(`$E_r \in G_r$`)`
  `R = GetReward(`$G_v^j(V_v^j, E_v^j)$`)`
  if `AdminCtrl(R) = Accepted then`
    $\mathcal{V} \leftarrow j$
    `Instantiate(`j`)`
    `Return Accepted`
  `else`
    `Return Rejected`
  `end if`
`end procedure`

procedure NodeMapping(V_r)
  for each $n \in V_r$ **do**
    $V_v \leftarrow$ Find best port mapping
  end for
  Return V_v
end procedure

procedure LinkMapping(E_r)
  for each $l \in E_r$ **do**
    $E_v \leftarrow$ Find best path mapping
  end for
  Return E_v
end procedure

- **Module 2. Per-Tenant Monitoring**

Function: MTA shall provide its tenants with individualized monitoring information of the status and utilization of virtual resources. To this aim, MTA shall use the monitoring service of the XCI and take out information that corresponds to the tenant being served at the moment.

	
Definition of MTA algorithms for the high-speed train use case	MTA for high Speed train scenario described in Figure 39 in section 6.2.5 is composed of two basic algorithms. The first algorithm reflects the On-board controller that is in charge of collecting data and pushing to the appropriate outbound gateway whereas the second algorithm reflects the On-Land controller, responsible for dynamically isolated path computation to Virtual Network Operators. <u>Functions</u> Dynamic Allocation of Virtual Infrastructure for Virtual Network Operators <u>Algorithm</u> 1) Module 1 (On-board Controller) ***Begin*** *MTA on board Get Network Topology from XCI* ***For** Each received request* *Calculate the appropriate outbound gateway for this received request* ***If** outbound gateway is not available/load on the gateway exceeds threshold* *Compute second appropriate outbound gateway* *Forwards the traffic into the chosen outbound gateway* *(Dynamic Allocation of most suitable computation path, along with bandwidth while preserving complete isolation)* ***End if*** ***End for***

	Updates Network Topology and inventory *(including available resources)* ***End*** Algorithm 2) Module 2 (On-Land Controller) ***Begin*** *MTA on land Get Network Topology from XCI* ***For*** *Each received request from wireless backhaul* ***If*** *Network resources are available.* *Request a slice from VIMaP.* *Connects end-user to its Core Network of VNO using this requested slice* *(Dynamic Allocation of most suitable computation path, along with bandwidth while preserving complete isolation)* ***End if*** ***End for*** *Updates Network Topology and inventory* *(including available resources)* ***End***
Requirements for XCI NBI	• **VIM NBI** The interface has to provide methods (API REST…) to interact with the VIM, which is responsible to create the virtual links. • **NFVO NBI** The interface is used to forward the requests from the tenants for deploying a network service and instantiation of virtual functions (e.g. L2/L3 virtual switching) according to Forwarding Graphs. • **SDN Controller** For (de)allocating the networking resource, the interaction with XCI is through the SDN Controller to configure the networking resource (e.g. path provisioning).
Requirements for NBI / SBI (Application)	• **NBI** An NBI interface is needed for the virtual tenants to communicate with the MTA about their requirements of the virtual network they wish to instantiate (including network graph and computational and network resources requirements). • **SBI**

	An SBI interface is needed for collecting the abstraction of the substrate infrastructure topology and the monitoring information on the QoS of the physical nodes and links.
Data Plane Requirements	Multi-tenancy requires network virtualisation technologies in the data plane, e.g. VPN technologies such as L2 tunnelling. MTA also requires the multi-tenancy support from the XCF (MAC-in-MAC or MPLS-TE) and XFE

3.2 Mobility Management Application

3.2.1 Description of MMA

Title	*Mobility Management Application (MMA)*
Description of the Application	The Mobility Management Application (MMA) is one of the OTT (Over-The-Top) applications of the 5G-Crosshaul. MMA is Distributed Mobility Management (DMM) based and its main goal is to provide mobility management for mobility scenarios such as vehicle mobility use cases like high speed train scenarios, and also to optimize traffic offloading for media distribution like CDN and TV Broadcasting. The information required by the MMA is the CDN nodes placement as well as the constraints from the tenant (service level agreements, PoCs and XPUs to be used, etc.). This information will be provided by other applications: the CDNMA and the RMA. The MMA uses the services offered by the RMA to provide best paths between the different elements of the network, with the main goal of optimizing the route or path followed by mobile users' traffic towards the Internet or to a core service provided in a datacenter. It will be done by computing the best allocation of services, like PoC, CDN Nodes and BBU allocation. After computing the best set of elements to provide a service to the user, the MMA will request the RMA to find the best path connecting these points based on the network status. The MMA works in close cooperation with the 5G-Crosshaul XCI and RMA; it uses the routing information collected by the XCI to provide best handover optimization between UE and core network.
Business or technique drivers	The main driver of this technology is focused on the reduction of the traffic within the core network. By the selective offload of traffic as near as possible to the RAN, we can reduce the costs of the core network, which requires to transport less amount of traffic. It will also improve the experience of the user by introducing lower delays in the end-to-end communication. Through the use of context-aware handover optimization, the radio link failure probability due to unsuccessful handover operations is reduced, decreasing unnecessary handovers, and improving the selection of the target BS by leveraging the context information.

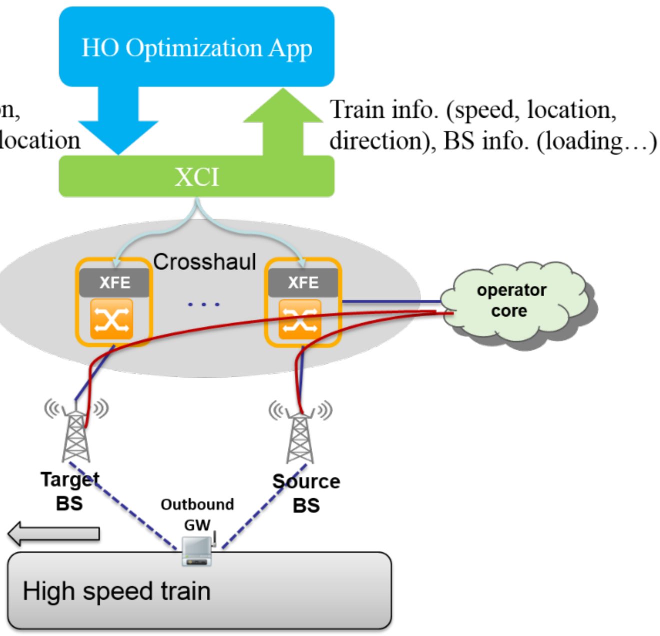
<table>
<tr><td>Use cases and Scenarios</td><td>
For the high speed train use case, handover (HO) is a critical operation in mobility management, and it is even more challenging for high mobility and high data rate scenarios because of more frequent HOs and higher sensitivity to link degradation, respectively. The focus of the proposed MMA algorithm is to exploit the routing information, including train location, speed, and direction, etc., in maintaining the routing path for the possible HO operation. This application will exploit the deterministic trajectory of the train in advance. Consequently, the MMA will reduce the handover time and maintain a high level of successful handover without degrading the UE performance.
The information required by the MMA is collected by XCI as shown in
Figure 8. When the train (as a UE) moves, the handover process initiated according to measurement reports, then T-PoA is provided by a new proactive routing path to the core network. The MMA will work offline to get relevant information to calculate the most appropriate proactive routing path for handover process. Moreover, the required network resource configuration corresponding to the new routing path is achieved by MMA algorithm and incorporation of RMA (in case it supports MMA functionalities). Compared to conventional handover process, the MMA algorithm in 5G-Crosshaul is made more efficient and reliable.


Figure 8: Context-aware handover optimization for high speed train use case
</td></tr>
</table>

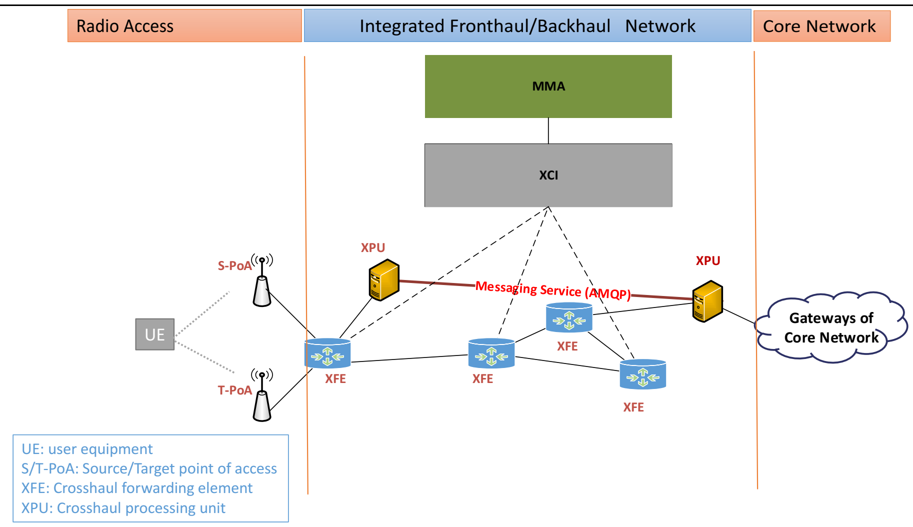
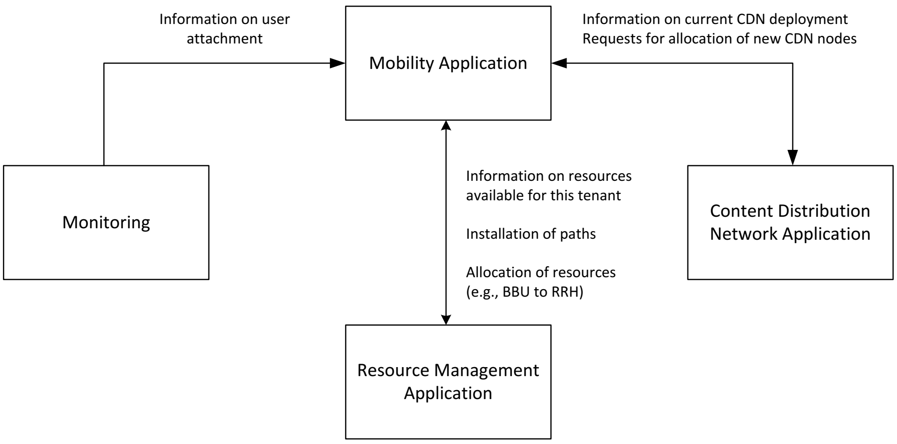
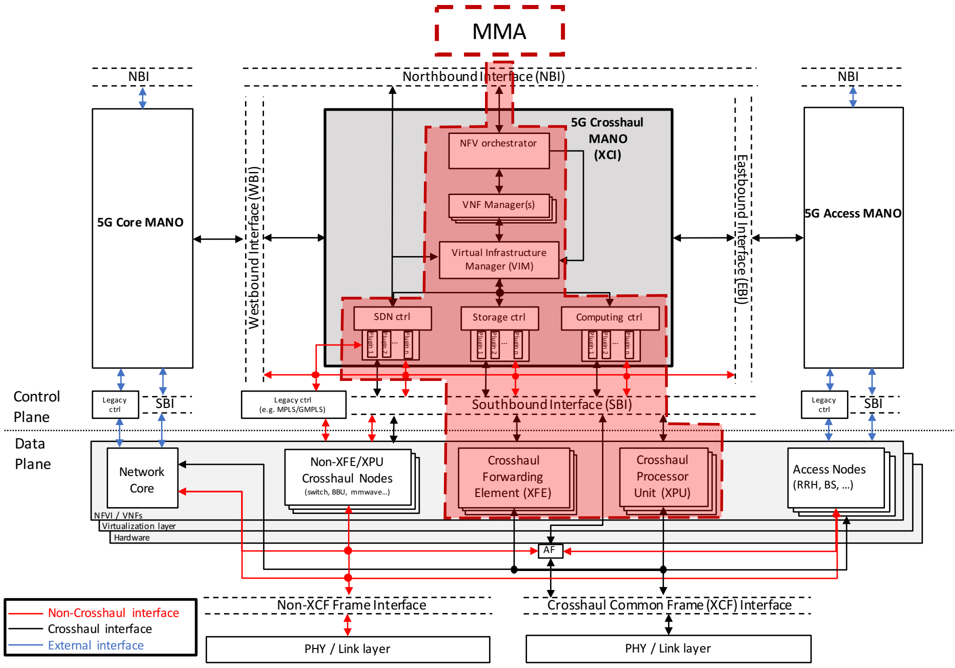


Figure 9: MMA interaction with 5G-Crosshaul for high speed train use case

Technologies to be used or considered	MMA is Distributed Mobility Management (DMM) based and its functionality should be agnostic on the transport technology used. The setup of paths considers that XFEs will be controllable via SDN primitives including the gathering of metrics regarding their capacity, delay, etc.	
Information requirements (Inputs)	***What***	***Who may provide this***
	• Type of traffic (QoS) • Policies • Routing function to query for paths (provided by RMA) • Location update of the outbound gateway(s) on the high speed train	MTA RMA VIMaP EMMA CDNMA
Outputs	***What***	***Who may need this***
	• PoC assigned to a flow • Proactive request for setup of paths and caching nodes for the case of the high mobility	RMA CDNMA
App-specific functions	• Mapping of the different users from each tenant to the available PoCs to be used per tenant. • Maintenance of the communication while moving. • Dynamic allocation of mobility anchors and gateways. • Dynamic pre-provision of paths along the track of the high speed train scenario.	

	• This application will exchange recommendations with the CDNMA to decide on the best placement of CDN nodes.
Required (external) functions	This application depends on the RMA to compute optimal paths between the different elements in the network. This application may need the following functionalities from external applications: • A set of possible routes between a PoA (considering the different functional splits) and a given PoC or CDN/destination. Each route must include the QoS terms that can be provided • A set of possible locations available for the tenant to place CDN nodes. • RouteTrack of the high speed train
Function Architecture	 *Figure 10: Mobility application function architecture.* Figure 10 presents the architecture of the Mobility application and its relation with the rest of applications on the 5G-Crosshaul control plane.
Implementation	The implementation details and the actual status of the MMA is detailed in Section 9.2.
Relation to other applications	(1) **RMA**: MMA needs the functionality provided by the resource management application to compute optimal paths between the different elements of the network. (2) **CDNMA**: MMA also requires interaction with the content delivery network application in order to determine the best placement of cache nodes based on client distribution.

<table>
<tr>
<td>Application mapping on WP1 system architecture (High level view)</td>
<td>The following figure shows the mapping of MMA onto the 5G-Crosshaul system architecture, under the red shape all the elements involved with the MMA (directly through the MMA or indirectly through the RMA) are detailed.


</td>
</tr>
</table>

3.2.2 Design of MMA Algorithms

<table>
<tr>
<td>Application</td>
<td>Mobility Management Application (MMA)</td>
</tr>
<tr>
<td>Definition of basic algorithms</td>
<td>The MMA has basic algorithms for two situations when a new user connects to the network and when a handover occurs:

- Connection of new user:

  Begin

  if packet-in received then:

  if llc/snap header then:

  get_user_mac

  get_AP_IP

  get network topology

  decide best GW/PoC based on heuristic and proximity

  if CDN then:

  notify CDNMA PoA IP to</td>
</tr>
</table>

	CDNMA *(send the PoA IP to CDNMA and get the CDN node assigned by the CDNMA)* get CDN Node from the response **end if** compute all the data send to RMA *(Send the data to the RMA for the creation of the path and get the path from RMA)* check response from RMA **return** **else if** dhcp discovery **then:** send dhcp offer **else if** **if** dhcp request **then:** send dhcp ack **end if** **end if** **End** • Handover: **Begin** determine the target base station find the best proactive path **if** CDN **then**: send new GW to CDNMA *(send new GW to CDNMA and if it is necessary get a new the CDN node)* wait response from CDNMA **end if** recompute all the data send data to RMA *(Send the data to the RMA for the proactive creation of the path and get the path from RMA)* wait response from RMA **End**

Requirements for XCI NBI	• **SDN controller** The XCI NBI has to provide methods to get a vision of the network topology. o Get nodes (XFE and ports) o Get links (connection points and bandwidth) o Get hosts and node connections (XPU and ports)
Requirements for NBI / SBI (Application)	• **OAM (Operation and Management Interface)** o An interface to the operator to obtain the information in which XPUs the GW are instantiated. o An interface to obtain the BS of the train trajectory for proactive path creation. • **NBI** o An interface with the CDNMA with the following primitives: ▪ Add/Delete Notification ▪ Add/Update/Delete User_CDN • **SBI** o An interface to the RMA for creation of paths. • **EBI/WBI** o An interface to the RAN for requiring the information about where the user is connected.
Data Plane Requirements	MMA does not have specific requirements for the Data Plane.

3.3 Resource Management Application

3.3.1 Description of RMA

Title	*Resource Manager Application (RMA)*
Description of the Application	Centralized and automated management of Crosshaul resources, in order to promptly provision transport services with an adequate quality while ensuring that Crossshaul resources are effectively utilized. The three main operational pillars of the application are: • Dynamic resource allocation: for each new demand, the RMA will provide the initial network configuration, including the route(s) and the resources to be provisioned, according to the network graph provided by the XCI (abstract view of the underlying network node

	and link resources). The path computation function will be carried out by a Path Computation Element (PCE), which may be either a service provided by the application itself or an external service. • Dynamic function placement: in addition to the resource allocation, the application will evaluate the functional split to be applied to the RAN, where specific functions can be flexibly (de)centralized in central processing nodes (i.e. XPUs). • Network adaptation: the application will provide an updated Crosshaul configuration (e.g. including new routes, and/or a different configuration of physical parameters) upon triggering events such as changes in traffic demands, interactions with other applications (e.g. with the MTA when a new VNO joins, or with the EMMA), link or node failure, etc. The adaptation may reuse the algorithms proposed for the resource allocation and the function placement. In general, given an initial Crosshaul state and the knowledge of the input traffic (i.e. the distribution of RAN traffic demands) plus some information about the RAN status, the application applies the most appropriate algorithms and returns a potential new Crosshaul state, which is then eventually enforced/applied by means of the XCI. It is worth to remark that the RMA application will not perform any reconfiguration of the RAN; however, the new Crosshaul configuration may have an impact on the RAN performance as well. This application requires a certain degree of knowledge of the underlying Crosshaul network resources. With respect to the Crosshaul, this shall be provided in the form of a graph (i.e. network graph) which abstracts relevant physical layer parameters which pertain to the different technologies (such as capacity, delay, jitter, SINR associated to a wired/wireless link, node capabilities, etc.). Regarding the traffic demands (heterogeneous in terms of volumes and types) that are served by the Crosshaul, these are assumed to be generated and/or provisioned in the radio access network and distributed in different geographical regions. The application can operate in a multi-tenant environment, providing a service to a Virtual Network Operator (VNO), which is assigned a slice of Crosshaul resources via the MTA. After the initial design of the Virtual Network (that is not performed by this application), the VNO has the view of the RAN+Crosshaul slice that it is controlling, and it may request the service of the RMA to serve the tenant (i.e. VNO) traffic. This application is essential for the control and management of the Crosshaul resources (links and nodes), therefore it can be applied to almost all the scenarios foreseen in the project. For instance, it may provide real-time and optimal utilization of Crosshaul resources for dense small cells, where geographical locations (traffic) of users dynamically vary from time to time.

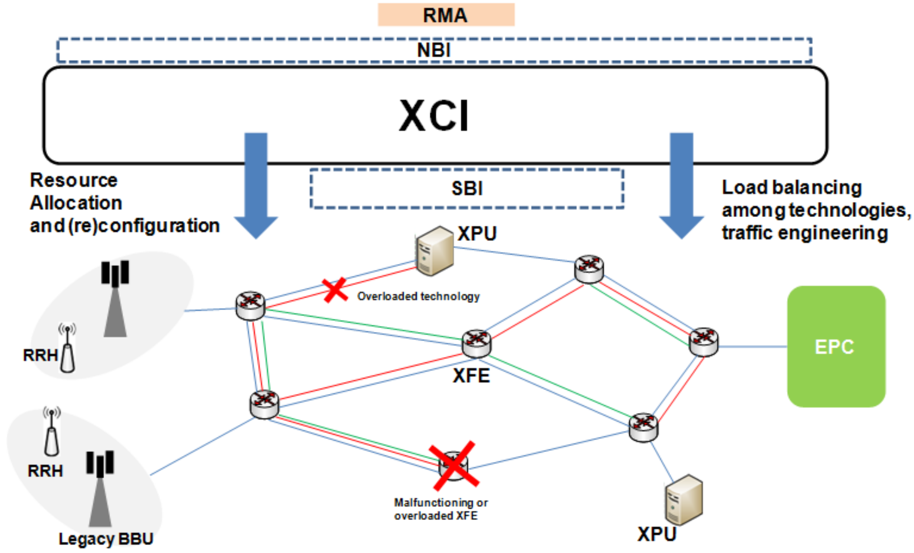
<table>
<tr><td>Picture</td><td>


Figure 11: General scenario illustrating the role of RMA within 5G-Crosshaul
</td></tr>
<tr><td>Business or technique drivers</td><td>
The effective utilization of Crosshaul resources is crucial to ensure that all users, including high-mobility users, receive the requested services with adequate level of QoS (enjoying a satisfactory experience). Over-provisioning of transport resources is not an economic solution for the PIP. Hence, this application is expected to provide basic and advanced configurations of Crosshaul resources in a rational way, thus ensuring that the 5G transport network will be able to serve more traffic or more requests. At the same time, the proposed application can contribute to reduce the energy consumption by using the different network elements and transport technologies in intelligent manner.

Overall, the RMA is expected to be very relevant in order to maintain a low OPEX and to provide resilience in case of network failures. In addition to that, this application creates the link with the access segment of the 5G network, where the traffic served by the Crosshaul is generated and terminated. The proper management of the Crosshaul is not conceivable without taking the RAN into account at least to some extent.
</td></tr>
<tr><td>Use cases and Scenarios</td><td>
The Application has to be exploitable in a number of situations in which users generate requests for different service types. Requests are generated within the RAN where users are camping. Requests are sporadic over time and nomadic as users move geographically. Different services carry traffic-specific requirements, including QoS constraints. Situations have to be distinguished between different traffic types in order to provision the most adequate provisioning of Crosshaul resources. In addition, whenever unbalances occur within the RAN, with areas in which users upload/download large volumes of traffic and nearby areas in which a light amount of traffic is created, Crosshaul resources need to be intelligently allocated/reallocated on the basis of the current situation.

The RMA shall include the following preliminary list of use-cases:

(1) Dynamic Resource allocation

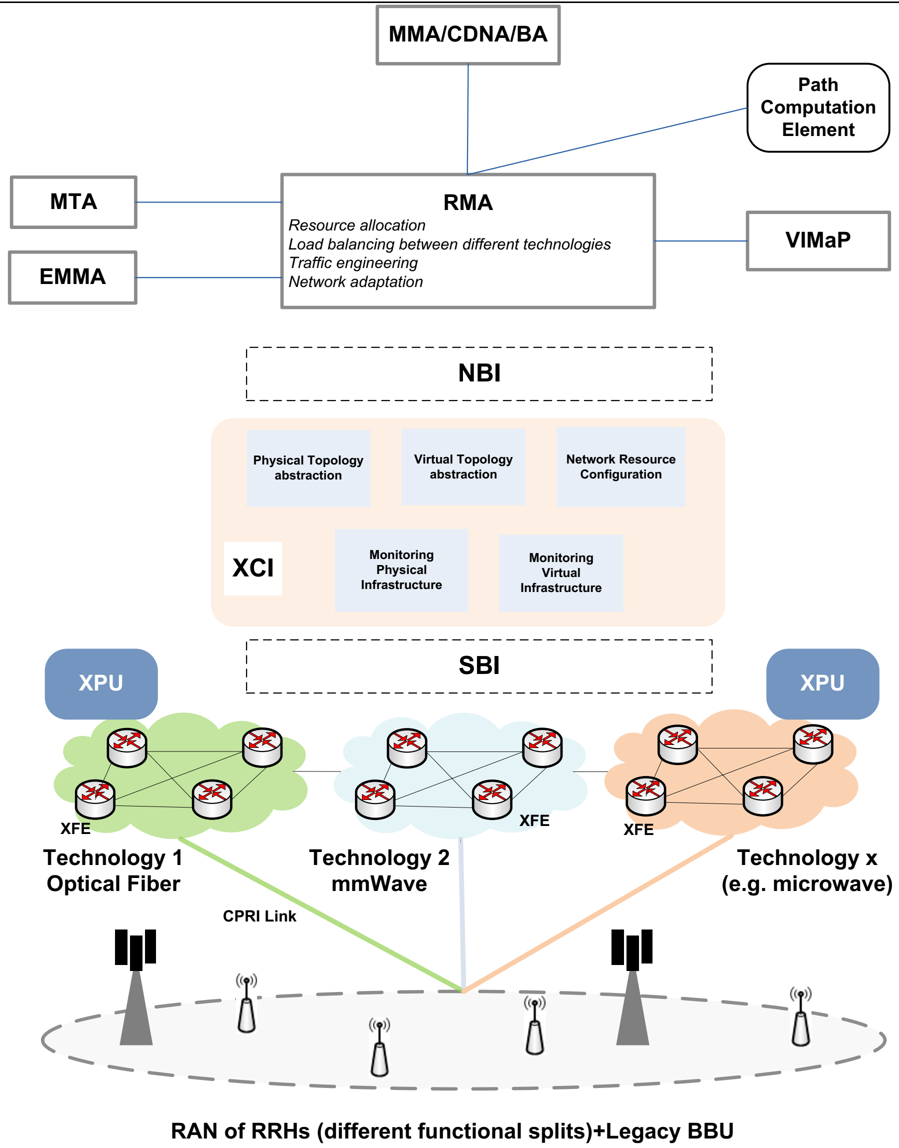
• Selection of the most suitable multi-technology routes between
</td></tr>
</table>

	nodes according to the requested QoS: in the Crosshaul different technologies may serve the input traffic. According to the actual requirements of the request, the application will select the most appropriate technology (e.g. a suitable radio segment or a more capacious optical-based wired segment). • Load balancing and traffic offloading across different paths: traffic engineering techniques in order to ensure that the network is effectively utilized. A specific operation could be the (re)partition of traffic across the RRHs, in which, according to the capabilities of the RRH, it could be possible to select different resources/technologies on the Crosshaul to serve the input traffic. For instance, a RRH with a congested fronthaul link may offload part of the traffic it should serve to another RRH (directly connected with the former) that has a more capable or less congested fronthaul link. • Optimization of Single Frequency Networks (SFNs) for content broadcasting/multicasting. Multicast and broadcast services are provided through simultaneous, synchronized transmission of content by the RRHs that are part of the same SFN. Depending on user demand, transmission data rates and RAN resource availabilities, SFNs have to be carefully designed so as to satisfy the quality of service required by users. (2) Dynamic function placement o Selection of the RAN functional split: specific functions can be flexibly (de)centralized in central processing nodes (i.e. XPUs). (3) Network Adaptation • Failure or degradation on CPRI links: in this case, the 5G-Crosshaul Resource Management application will react to a (sudden) degradation of the CPRI link (wireless e.g. Microwave, mmWave) due to a change in environmental conditions (attenuation of the signal). In this case, the network adaptation function will relieve from: • Failure case: Re-routing through a traditional close-by BS (i.e. full decentralization). • Degradation case: In this case it would be possible to modify the functional split by moving from a fully centralized split (e.g. C-RAN) to a less centralized option with less stringent capacity/delay demands. Please note that the use-case described above can be either applied on a network-wide basis, thus having a potential impact on multiple VNs, or restricted to the resources of a specific VN. Typical scenarios: virtual reality office; shopping mall; stadium; blind spots; open air festival; high-mobility users in small-cell environments, emergency communications; massive deployment of sensors & actuators.

Technologies to be used or considered	Potentially, all the wired and wireless Crosshaul technologies should be considered for this application.	
Information requirements (Inputs)	**What**	**Who may provide this**
	Abstraction of Crosshaul physical infrastructure / tenant infrastructure: • Per-tenant and per-network Topology awareness ○ Node indexes and location given by GPS coordinates. • Link information is provided as an adjacency matrix of size NxN where N is the number of Crosshaul nodes. An entry equal to 1 indicates the existence of a direct bidirectional link, 0 if there is no direct link. • Each link has associated a profile: ○ Total throughput capacity [Mb/s] ○ Estimation on max, min, mean, and variability of link delay [usec] ○ Estimation on mean packet loss rate [%] • Current RAN functional split (per access node i) ○ Set of functions that run in access node i ○ Per XPU j: Set of functions of access node i that run in XPU j	XCI may provide information on physical infrastructure (through NBI) Or another WP4 application with discovering mechanisms may provide related information.
	Abstraction of traffic demands: • Demand characteristics (unicast, multicast, requested content items, required survivability, etc) • Endpoints of the demand • User mobility information (e.g. pattern and average user speed [m/s]) • Max, min, mean and variability of expected load [Mb/s] • Maximum delay tolerated [s] • Max packet loss rate tolerated [%]	User of the application (e.g. operator, service provider) MMA

	Path Computation: • Given the above inputs, calculation of the path (including route and resources to be used) according to specific metrics	This may be provided by the application itself or by a third application
	Monitoring of the status of each demand: • Max, min, mean and variability of traffic load [Mb/s] • Max and variability of delay experienced [usec] • Mean packet loss rate [%]	User of the application (e.g. operator, service provider)
	Monitoring of the infrastructure (both per-tenant and per-network): • Estimation on each link of the SNR/SINR • Information about faulty behaviours for alarm notification • Instantaneous link utilization [b/s][%]	XCI may provide information on physical infrastructure (through NBI) Or another WP4 application with discovering mechanisms may provide related information.
	Status of the nodes: • ON/OFF • power consumption metrics • energy-aware virtual function allocation	EMMA
Outputs	**What**	**Who may need this**
	New network state, including, for each primary and backup path of each demand • Route/s (potentially involving different technologies across one or multiple paths); this shall include switching information, if needed • Bandwidth reserved on each physical link [Mb/s] • Mapping between service and reserved resources (e.g., caching and SFNs configuration) o RAN Functional split (per access node i) o Set of functions run in access node i o Per XPU j: Set of functions of access node i run in XPU j	XCI shall enforce this into the network All the applications (either customer or network applications) that need to instantiate traffic flows in the network

App-specific functions	Resource management functions • Initial Routing and resource allocation • Balancing among different technologies • Traffic engineering • Selection of functional split • Network adaptation in case of change in traffic pattern, signal degradation, failures.
Required (external) functions	• Abstraction of the physical/virtual infrastructure • Path computation • Allocation of networking resources, e.g. through XCI • Bandwidth reservation on each physical link • Enforcement of routing rules • Enforcement of traffic scheduling, shaping, QoS provisioning/differentiation • Monitoring of the physical/virtual infrastructure • Monitoring of the connection status • User demand information (requested content items and their characteristics, including the traffic distribution)
Function Architecture	Initial view of the function architecture:

	 *Figure 12: Resource management application function architecture*
Implementation	An analytical and simulative approach will be chosen as main methodology. Some use-cases will be integrated into the mmWave meshed backhaul & access demonstrator that will be developed in WP5.
Relation to other applications	• **MTA**: the RMA can be called in order to manage the resources of each tenant. In this case, it will act as a server application; • **EMMA**: the RMA is complementary of the energy management and monitoring application where the total energy consumption is minimized while maintaining the QoS of each VNO or end user. • **MMA**: RMA can provide computation of the optimal/sub-optimal path depending on available information and network conditions Further interactions could be with the **CDNMA** for caching of contents.

3.3.2 Design of RMA Algorithms

<table>
<tr>
<td>Definition of basic RMA algorithms</td>
<td>
RMA algorithm for PC-VNFP service
Among others RMA shall provide three types of services: Path Computation (PC) service, Path Computation-Virtual Network Functions Placement (PC-VNFP) service, and Path Computation-BS functional decomposition (PC-BS-split). The first refers to the service provided by this application in computing the optimal path for network flows based on specific inputs such as the ID of a tenant network, the source and destination node IDs, as well as the bandwidth and latency demands of the flows. The PC service computes the optimal path trying to optimize some specific objective function $U(\cdot)$. In this endeavour RMA shall mainly deal with the network resources and technology types available within the 5G-Croshaul network. On the other hand, the PC-VNFP service stands for a more challenging task performed by this application since the RMA performs the optimal path computation together with the placement of the Virtualised Network Functions (VNFs) on XPU nodes. Finally, PC-BS-split service is in charge of jointly maximising the degree of BS centralization while routing flows between RRHs and XPUs. Maximising the degree of BS centralization refers to offloading as much BS functionality into XPUs as possible, given networking constraints.
Besides the inputs already mentioned for the PC service, in this case the RMA necessitates also the list of VNFs to place in XPU nodes and the Forwarding Graph (FG) information which a flow requires in order to fulfil the service chain. RMA shall provide these services to any entity in the 5G-Crosshaul network which is requesting them, including any other application or module of the XCI requesting those services. For the purpose of providing the two services mentioned above, RMA ought to achieve the following objectives:

VNF placement: Placement of services on XPU nodes in such a way that service requests are fulfilled and the objective function $U(\cdot)$ is optimized. To achieve objective function optimisation, each XPU node is equipped with a number of cores, with one core fully dedicated to one virtual machine (VM), and a VM hosting one VNF. In addition, it is also assumed that one core can serve up to a certain flow demand which, if exceeded, would require either VM scale-up or creation of a new VM for that VNF, within the same or in another XPU;
Flow allocation: compute the optimal path for a flow in such a way that the FG is fulfilled, and the objective function $U(\cdot)$ is optimized.

Since 5G-Crosshaul includes the possibility of deploying VNFs on XPU nodes and connecting XPFEs elements through different transmission technologies, the optimization problem was formulated accordingly. In particular, the objective of RMA to perform PC and PC-VNFP services consists in the following minimisation:
$$U = \min\left(C_{VNF} + C_f + C_d\right) ,$$
where C_{VNF} is the cost associated to deploying a VNF over an XPU node;
</td>
</tr>
</table>

	C_f denotes instead a fixed cost associated to a transmission technology and C_d is a dynamic cost associated to that technology. Costs are introduced here as a penalty that the system incurs in case of making specific decisions. Overall cost minimisation clearly yields the optimal solution. All costs are unit-less and serve the purpose of describing the differences between selected transmission technologies. Conceptually, the different cost components find the follow physical explanation: • **VNF deployment cost:** Fixed cost paid when deploying VNFs on XPU nodes. • **Fixed technology cost:** Cost associated to using a transmission technology. Particularly, a wireless transmission technology can reasonably exhibit lower fixed cost, thinking for instance to reflect simpler and quicker installation process. On the other hand, a fixed transmission technology would require larger initial effort for laying down the necessary infrastructure. • **Dynamic technology cost:** Cost associated to the use of a specific transmission technology. This is the unit of cost paid by each flow for using such technology, and a flow with higher bandwidth demand shall incur in larger cost. Moreover, a fixed transmission technology shall exhibit in general a lower dynamic cost, which is an expression of larger available bandwidth and higher reliability than a wireless transmission technology counterpart. The objective of computing the optimal path for each flow seeks to minimise the overall cost U, when the cost of each flow is given by the summation of fixed plus dynamic costs. The objective of minimising the overall cost function is a possible way of approaching the problem of managing network resources in conjunction of compute resources. In this case the objective is that of using resources as efficiently as possible. In other words, the objective translates directly into the possibility of reusing those XPFEs and XPUs which have been already used by flows already allocated by the RMA algorithm. Only when the capacity of used links and nodes is exceeded, other technological edges and computing nodes will be used. This allows a less fragmented, and thus more efficient, use of resources. Resources which are not used can be conveniently put into a low energy state and reactivated when necessary (e.g. to cope with subsequent overloading conditions). ***Description of the RMA algorithm for PC-VNFP service*** The RMA algorithm is formulated as an equivalent Integer Linear Programming (ILP) problem. This formulation not only determines the placement of services and routing of the flows, but also seeks to minimize the resource utilizations. Generically, network nodes are denoted here as u and v. Implicitly, it is assumed that a switching element is connected to an XPU in such a way to enable network wide connectivity (i.e. connecting the XPU to XPFEs). A network flow is a traffic from/to the mobile core network to/from users, depending on whether it is respectively downlink or uplink traffic. Realistically, to each flow is associated bandwidth and latency

demand, which the RMA algorithm must fulfil to be successful. In the remainder of this section the ILP formulation below relies also on the work done in [5] and [6]. The network and flow specific parameters are defined below:

Parameter	Description
$G(V,E)$	Network graph to topology, V is set of nodes (XFEs and XPUs) and E is set of edges.
$w_{(u,v)}$	{0,1}: 1 if there exists an edge between node u and v; 0 otherwise
$c(u,v)$	Capacity of link (u,v) between node u and v
$l(u,v)$	Latency of link (u,v).
$k^c_{(u,v)}$	Fixed cost of using the edge (u,v) i.e. if any amount traffic, greater than zero, passes through the edge (u,v), this is the cost paid
$k^d_{(u,v)}$	Dynamic cost of unit flow that passes through edge (u,v)
h^n_i	Fixed cost of instantiating a VNF of type n on node $i \in V$
O_v	Cores available at node $v \in V$. Each core can support one VNF
U_s	Number of flows service $s \in S$ can support on one core
M^n_i	{0,1}: 1 if service $n \in S$ can be supported at node i, 0 otherwise

Parameter	Description
F	Set of all flows in the network
s^f	Start node of flow $f \in F$
t^f	Destination node of flow $f \in F$
d^f	Capacity demand of flow $f \in F$
l^f	Latency demand of flow $f \in F$. Latency that flow incurs while moving from source to destination should be less than this value.
K	Set of all services (i.e. VNFs) that can be placed on XPU

	nodes
C^f	Service chain of flow $f \in F$. Set of services (i.e. VNFs) that flow $f \in F$ needs to traverse in a specific order. i.e. $n_1 \to n_2 \to \cdots \to n_l$ where $n_i \in K$
C_{st}^f	$[n_{s^f} \to C^f \to n_{t^f}]$ Service chain of flow $f \in F$ which includes s^f and t^f nodes. To ensure that flow starts at node s^f and ends at node t^f, two virtual services n_{s^f} and n_{t^f} are introduced at s^f and t^f, nodes respectively. Since n_{s^f} and n_{t^f} services are only present at s^f and t^f nodes, these nodes are selected as start and end node of the flow path.

Variables which are needed in the ILP problem formulation are provided herein below.

- $x_{(u,v)}^{f(n\to m)}$: {0,1}. 1 if edge (u,v) is used to reach from service n to m in the FG C^f of flow $f \in F$; 0 otherwise.
- $x_{(u,v)}$: {0,1}. 1 if any flow passes over edge (u, v); and 0 otherwise. This is not a decision variable. In specific,

$$x_{(u,v)} = \begin{cases} 1, & \text{if } \sum_{f\in F}\sum_{(n\to n)\in C_{st}^f} x_{(u,v)}^{f(n\to m)} + \sum_{f\in F}\sum_{(n\to n)\in C_{st}^f} x_{(v,u)}^{f(n\to m)} > 0 \\ 0, & \text{otherwise} \end{cases}.$$

The condition shown in the previous equation can also be reorganized in the following way

$$\begin{aligned} & x_{(u,v)} \le \sum_{f\in F}\sum_{(n\to n)\in C_{st}^f} x_{(u,v)}^{f(n\to m)} + \sum_{f\in F}\sum_{(n\to n)\in C_{st}^f} x_{(v,u)}^{f(n\to m)} \\ & x_{(u,v)} \ge x_{(u,v)}^{f(n\to m)}, \forall f \in F, \forall (n\to m)\in C_{st}^f \\ & x_{(u,v)} \ge x_{(v,u)}^{f(n\to m)}, \forall f \in F, \forall (n\to m)\in C_{st}^f \end{aligned}.$$

- S_i^{fn} : {0,1}. 1 if service $n \in C_{st}^f$ is at node i for flow $f \in F$, and 0 otherwise.

- X_{ia}^n : {0,1}. 1 if service $n \in K$ is placed on node i, 0 otherwise. This is not a decision variable. In other words, it holds that

$$x_{ia}^n = \begin{cases} 1, & \text{if } \sum_{f\in F} S_{ia}^{fn} > 1, \forall n \in C^f, \forall i \in V, \forall a \in O_i \\ 0, & \text{otherwise} \end{cases}.$$

Also in this case the above equation can be reorganised in the following manner:

$$x_{ia}^{n} \leq \sum_{f \in F} S_{ia}^{fn} > 1, \forall n \in C^{f}, \forall i \in V, \forall a \in O_{i}$$

$$x_{ia}^{n} \geq S_{ia}^{fn}, \forall n \in C^{f}, \forall i \in V, \forall a \in O_{i}$$

Detailed cost components

The RMA algorithm seeks to find the optimal placement of VNFs that minimizes the fragmentation of resources in the network, provided that the demands of the flows can be satisfied. The different cost components description is provided in detail below.

VNF Deployment Cost: This is the cost to deploy a VNF on a node, with the constraint as follows

$$C_{VNF} = \sum_{i \in V} \sum_{n \in K} \sum_{a \in O_i} h_i^n x_{ia}^n \ .$$

Link Usage Fixed Cost: Cost incurred whenever a link is used by any of the flows. This cost is generally technology dependent.

$$C_f = \sum_{(u,v) \in E} k_{(u,v)}^{c} x_{(u,v)} \ .$$

Link Usage Dynamic Cost: Dynamic cost is based on the amount of link resources used by flows. This is cost per unit of flow going through a link.

$$C_d = \sum_{(u,v) \in E} k_{(u,v)}^{d} \sum_{f \in F} \sum_{(n \to m) \in C_{st}^{f}} d^f x_{(u,v)}^{f(n \to m)} \ .$$

The cost function to minimise was shown in equation (2) in sub-section 3.3.1. Furthermore, the terms edge and link are used interchangeably.

Optimization Constrains

Edge Capacity Constraint: Each edge has a capacity limit that must not be violated. This constraint makes sure that the sum of all the flows passing through an edge never exceeds the capacity of the link.

$$\sum_{f \in F} \sum_{(n \to m) \in C_{st}^{f}} d^f x_{(u,v)}^{f(n \to m)} \leq c(u,v) \qquad \forall (u,v) \in E \ .$$

Flow Latency Constraint: Each flow has a latency constraint that must not be violated. A flow should not experience latency greater than its latency constraint.

$$\sum_{(n\to m)\in C_{st}^f} \sum_{(u,v)\in E} l(u,v) x_{(u,v)}^{f(n\to m)} \le l^f \qquad \forall f \in F \tag{10}$$

Edge Constraint: Only edges which are present in the physical network are used when going from one service to another in the service chain.

$$x_{(u,v)}^{f(n\to m)} \le w_{(u,v)} \qquad \forall (u,v) \in E, \quad (n \to m) \in C_{st}^f$$

Flow Constraint: $\forall_i \in V$, $(n \to m) \in C_{st}^f$ where service m is after service n i the service chain C_{st}^f. Flow constraint ensures that there is a single continuou path between pair of nodes on which service n and m are placed. Here, n_{sf} $n_{tf} \in C_{st}^f$ and they are placed at s^f and t^f, respectively. This ensures tha flow path starts at node s^f and ends at node t^f.

$$\sum_{j\in V} x_{(i,j)}^{f(n\to m)} - \sum_{k\in V} x_{(k,i)}^{f(n\to m)} = \sum_{a\in O_i} S_{ia}^{fn} - \sum_{a\in O_i} S_{ia}^{fm}$$

Service Selection Constraint: Service $n \in C_{st}^f$ can only be selected on nodes that can host service n.

$$S_{ia}^{fn} \le M_i^n \quad \forall n \in C_{st}^f, \quad \forall i \in V, \forall a \in O_i$$

Service Node Selection: Single node is selected to host a service in the service chain C_{st}^f of flow $f \in F$.

$$\sum_{i\in V} \sum_{a\in O_i} S_{ia}^{fn} = 1 \quad \forall n \in C_{st}^f, \quad \forall i \in V$$

Node Capacity Constraint: We can keep track of number of cores being used. Each free core at a node can host a single service. So number of services hosted at node is limited by the number of cores available at that node.

$$\sum_{n\in K} \sum_{a\in O_i} X_{ia}^n \le |O_i| \quad \forall i \in V$$

Service Capacity Constraint: Each service deployed in an XPU node can only serve up to a certain flow demand. For instance, service n can serve 1000 Mbit/s, while service m can service 100 Mbit/s on a single core.

$$\sum_{f\in F} \sum_{n\in C^f} d^f S_{ia}^{fn} \le U_n \quad \forall i \in V, \forall a \in O_i$$

Single VNF per core: This constraint makes sure that there can be only one

	VNF instance per core on a node. $$\sum_{n\in K} X_{ia}^{n} \leq 1 \quad \forall i \in V, \forall a \in O_i$$.
Requirements for XCI NBI	The requirements for the XCI NBI are summarized here in brief: • Abstraction: topology and inventory of the network and XPU (e.g. regarding network links, nodes, capabilities, etc. as well as computing and storage resources) directly from the underlying controllers. • Notifications: notifications from the XCI NBI regarding state changes including, but not limited to, resource failures. • Monitoring: information about the devices of the underlying infrastructure; this includes the ability to query/poll for statistics and performance/monitoring counters.
Requirements for NBI / SBI (Application)	The RMA application will provide a northbound interface towards other applications and modules requesting for the computation (optimization) service. The interface will be used to provide the response that includes the information about the flows that can be installed and the processing and storage resources that can be used.
Data Plane Requirements	No specific requirements for the data plane.

3.4 Virtual Infrastructure Manager and Planner

3.4.1 Description of VIMaP

The Virtual Infrastructure Manager and Planning application (VIMaP) is logically part of the 5G-Crosshaul XCI and stays at the lowest level of the application hierarchy. The VIMaP enables other applications (such as the Multi-Tenant Application, MTA) to request the constrained allocation of physical and virtual Crosshaul resources (i.e., computing and networking resources) represented as an abstracted construct (e.g., a slice), and proceeds to instantiate, deploy and provision them over the Crosshaul infrastructure (see Figure 13).

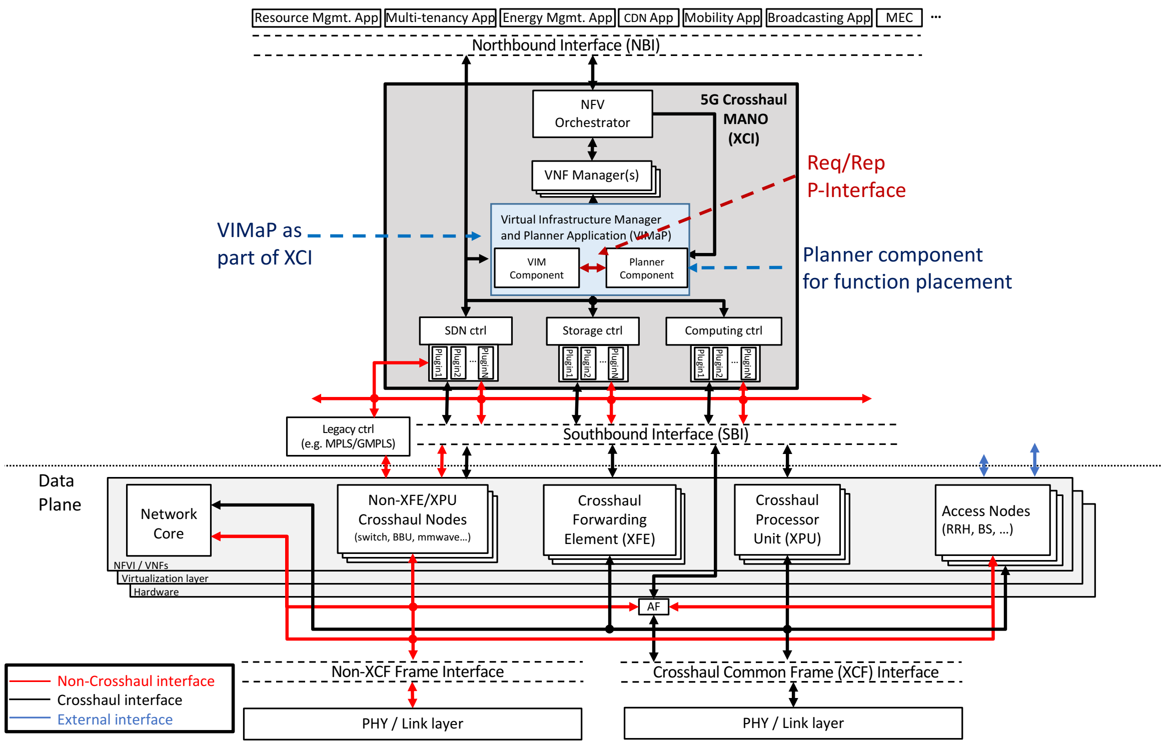


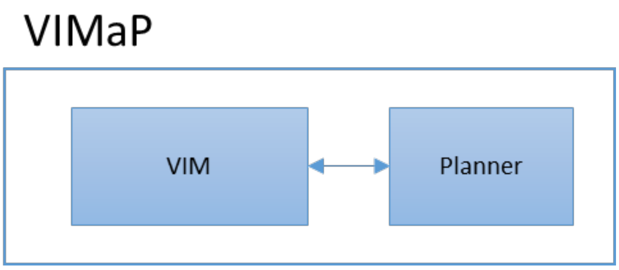
Figure 13: VIMaP component within the 5G-Crosshaul XCI

It acts as a bridge between an application requiring a heterogeneous set of resources and the actual interaction with one or multiple underlying controllers. The VIMaP itself, consequently, exports an API to the aforementioned applications, and this API constitutes its NBI. As a functional entity within the XCI, the VIMaP-NBI is thus part of the “aggregated” XCI-NBI, the latter also composed of the NBI exported by the low level XCI functional elements (notably, the network, cloud and storage controllers). To accomplish its function, the VIMaP imposes a certain number of requirements on the part of the XCI-NBI that it uses.

The goal of this application is to plan and optimize the physical and virtual Crosshaul resources (i.e., computing and networking resources) and to instantiate over the Crosshaul infrastructure the decisions taken. VIMaP consists of two components: the planner and the Virtual Infrastructure Manager (VIM). The planner component is in charge of running the required resource allocation algorithms for the planning and (re-) optimization of Crosshaul resources. The VIM is the component responsible for the dynamic provisioning and instantiation of Crosshaul resources.

The planner runs the resource allocation algorithms with the aim of (re-)optimizing the Crosshaul resources. It is able to perform the constrained allocation of interconnected endpoints (including VMs, VNFs, etc) and network connectivity services.

The VIM is the responsible for handling jointly the different IT and Network resources. Virtualization of IT resources is provided by means of a Cloud Controller, which is able to control the Crosshaul Processing Units (XPU), where the computing and storage resources are allocated. A VM might be requested based on its availability zone, its hardware resources (i.e. flavor), or the disk image/container/… to be loaded. A VM is also allocated inside a network. A second management network is provided by default to the VM to provide management access to the VM by the different applications (e.g. VNF Managers).

Title	*Virtual Infrastructure Manager and Planner (VIMaP)*
Description of the Application Functions	The main functions offered by the VIMaP are: ○ Underlying **resource information** requests, the main information provided is the Request Network and IT resources status, which includes: • Network topology • Available IT resources (CPU, memory, and storage at each server). • Deployed network connectivity services (including VM interconnections). • Status of deployed VMs. • **Perform constrained allocation of interconnected VMs and network connectivity services.** ○ Endpoints (VMs, VNFs, etc) and links layout graph. ○ Expected resource demands (e.g. traffic matrix) or dynamic traffic demands from end-users. ○ Intent-based service request. ○ Receive (re-)planning resource optimization notifications, including: • VM migration • Link reallocation • Resource failures • Enable direct **instantiation**, including: ○ Virtual Computing Instance (VM) (such as virtual machine/container/…) Create, Read, Update, Delete (CRUD) mechanism; ○ Network connectivity services CRUD mechanism.
Picture	 *Figure 14: Virtual infrastructure manager and planner application.*
Business or technique drivers	A resource efficient deployment of Crosshaul resources is fundamental for the realization of SDN/NFV advantages. VIMaP allows provisioning the best possible Crosshaul resources for deploying a given service on top of the operator infrastructure. In turn, this results in more services and users being packed over the same infrastructure, and so, more benefits for the operator. Depending on the stakeholder layout in a given scenario, in addition to its own services, the stakeholder may sell exceeding capacity to other stakeholders so as to increase its sources of revenue.

<table>
<tr><td rowspan="1">Use cases and Scenarios</td><td colspan="2">Network Functions Virtualization Infrastructure as a Service (NFVIaaS). Crosshaul offers an infrastructure so that third parties offer their services on top of it. This is enabled by the capability to create and interconnect several virtual machines (VMs) (or containers) hosting VNFs. These VMs will be located in different XPUs and they will be interconnected through the XFEs, which are also dynamically configured by the VIMaP.

At the coarsest, higher level, the VIMaP API allows a client to perform an allocation request, conveying a set of resources and their interdependencies (e.g., a set of interconnected virtual machines following an arbitrary graph). The planner component of the VIMaP executes resource allocation algorithms and the VIM performs their dynamic provisioning and instantiation. This is without precluding that the VIMaP may offer finer granularity APIs allowing direct access to the resources. A use case for this is the placement of CDN nodes over the virtual infrastructure.</td></tr>
<tr><td>Technologies to be used or considered</td><td colspan="2">VIM: OpenVIM aspect of OpenMANO, Cloud Controllers such as OpenStack.

The VIMaP is not dealing with VNF management or orchestration, which are left to upper levels of the application hierarchy.</td></tr>
<tr><td rowspan="3">Information requirements

(Inputs)</td><td>What</td><td>Who may provide this</td></tr>
<tr><td>A VIMaP allocation requests conveying a high level construct or a low level access for network and IT resources.

A VIMaP allocation request containing a detailed virtual infrastructure description including virtual compute nodes and virtual XFEs (requires MT support from the underlying SDN/Cloud controllers).</td><td>MTA (acting as a VIMaP client)

CDN application</td></tr>
<tr><td>Constrained allocation request of interconnected VMs and network connectivity services.
o Endpoints (VMs, VNFs, etc) and links layout graph.
o Expected resource demands (e.g. traffic matrix) or dynamic traffic demands from end-users.
o Intent-based service request.</td><td>MTA (acting as a VIMaP client)</td></tr>
</table>

	(Re-)planning or (re-)optimization request.	MTA, internal
	Request for instantiation of VMs and/or Network connectivity services.	MTA (acting as a VIMaP client)
	Information about physical network topology	Lower level XCI (SDN controller), through its NBI
	Information about XPU/cloud capabilities.	Lower level XCI (cloud controller), through its NBI
Outputs	**What**	**Who may need this**
	Virtual Machine resource allocation. Network connectivity services resource allocation.	Requests from the higher level apps (MTA) honored by multiple requests delegated to the underlying SDN controller(s), Storage Controller(s), Computing Controller(s)
App-specific functions	Request Network and IT resources status. Perform constrained allocation of interconnected VMs and network connectivity services. Receive (re-)planning resource optimization notifications. Instantiate VMs and network connectivity services. Allocation of Virtual Infrastructures composed of virtual compute nodes (hosts) and virtual XFEs.	
Required (external) functions	At a later stage, the VIMaP application may off-load part of the computation function to an external entity, relying e.g. on an extended PCEP protocol. The interface requirements defined by the VIMaP application to be fulfilled by the XCI NBI are related to the network, storage and cloud controller functional elements and involve, mainly: 1) the ability to query the underlying topology and inventory of the network and XPU (e.g. in terms of network links, nodes, capabilities, etc. as well as computing and storage resources) directly from the underlying controllers, 2) the ability to obtain notifications from the XCI NBI regarding state changes including, but not limited to, resource failures, 3) the ability to query/poll for statistics and performance/monitoring counters, 4) the ability to instantiate, dynamically, virtual machines and storage volumes and to retrieve their assigned parameters, 5) the ability to define network connectivity services by provisioning arbitrary interconnections between the allocated VMs (endpoints) e.g. by configuring the required traffic flows and 6) the ability to allocate virtual infrastructures including network slices.	

	Multi-tenancy support - For advanced users not only assuming a simple overlay on inter-connected VMs, the SDN controller should support the request of more complex virtual network topologies, including explicit description of virtual network nodes and links, similar to the NEMO models or the VTN models (see https://wiki.opendaylight.org/view/NEMO:User_Manual or https://wiki.opendaylight.org/view/Release/Lithium/VTN/User_Guide).
Function Architecture	There are two main components: • The VIM component part, which interacts with the underlying controllers for the actual provisioning. • The Planner part, responsible for the algorithmic aspects. This part can be off-loaded to a third-party entity.
Implementation	CTTC plans to have a test-bed with integrated Cloud and Network resources in the scope of this WP, which will exploit the work carried out in other WPs. The network resources will consist on the interconnection of mmWave, electrical and optical network domains. The test-bed will allow performing measurements of end-to-end connectivity between virtual machines and providing dynamic seamless virtual machine migration. The main focus will be on the VIM component of VIMaP.
Relation to other applications	VIMaP is located at the lowest layer of the application hierarchy. It encapsulates direct access to one or multiple low level controllers and allows other applications (e.g. MTA) to request the allocation of slices. At a later stage the computational aspect may be off-loaded to a resource manager application.

3.4.2 Design of VIMaP Algorithms

Definition of basic algorithms	The VIMaP algorithmic aspects cover different cases: • The VIMaP has a basic placement algorithm that is defined by the underlying placement default algorithms located within the cloud and SDN controllers that are part of the XCI control sublayer • The VIMaP shall allow the direct placement of Virtual Machines as computed by other entities or functional modules. In this case, no algorithm is involved. • The VIMaP may deploy algorithms in its Planning (P-) component as an additional internal interface (e.g. based on REST) • **Default algorithm** Function: algorithm on the placement of Virtual machines (server instances) inter-connected in a given graph across in the 5G-Crosshaul network. Algorithm: Module 1

<table>
<tr><td></td><td>
Begin
Get Infrastructure Topology from XCI
(List of all the switches, hosts and links from the topology manager)
If not XPUs in Network Topology available then
reject
for each VM node type
Select best XPU for deployment (e.g. available)
end for
for each Link in topology for Inter-VM connectivity
Provision connectivity via SDN controller
end for
End
• P-component
<u>Function:</u> algorithm on the placement of Virtual machines (server instances) inter-connected in a given graph across in the 5G-Crosshaul network.
<u>Algorithm:</u> the system has been designed in such a way that improved algorithms can be provisioned later on, by defining an interface between the VIM and P components. This interface is internal and agreed upon in any given implementation.
</td></tr>
<tr><td>Requirements for XCI NBI</td><td>
Inputs: Ability to query the underlying topology and inventory of the network and XPU (e.g. in terms of network links, nodes, capabilities, etc. as well as computing and storage resources)
Output: status of resources

Input: Subscription to SDN controller evens
Output: Ability to obtain notifications from the XCI NBI regarding state changes including, but not limited to, resource failures

Input: Subscription / polling to statistics and performance/monitoring counters,
Output: events / statistics

Input: Instantiate, dynamically, virtual machines and storage volumes
Output: retrieve their assigned parameters
</td></tr>
</table>

<table>
<tr><td></td><td>Input: Ability to define network connectivity services by provisioning arbitrary interconnections between the allocated VMs (endpoints) e.g. by configuring the required traffic flows. Advanced use case with arbitrary topologies
Output: allocation of flows
SDN Controller
The interface has to provide methods (API REST…) to get the network topology, to retrieve a list of all known nodes, hosts and links from the topology manager.
Get nodes and all the links getting out/in from/to the node
XFEs and port connections
Get all link connections between nodes including their properties
Links (connection points, Bandwidth)
Get XPUs
XPUs and port connections
SDN Controller
Methods for the provisioning of flows for Inter-VM connectivity.
Cloud controller
To support VIM functions, for allocation of software images, instantiation of virtual machines, network resource management, etc.</td></tr>
<tr><td>Requirements for NBI / SBI (Application)</td><td>NBI
Defined in [2]
SBI
The VIMaP application will interact with the SDN cloud and storage controllers bound to the underlying infrastructure (part of the XCI control sublayer)</td></tr>
<tr><td>Data Plane Requirements</td><td>No specific requirements for the data plane.</td></tr>
</table>

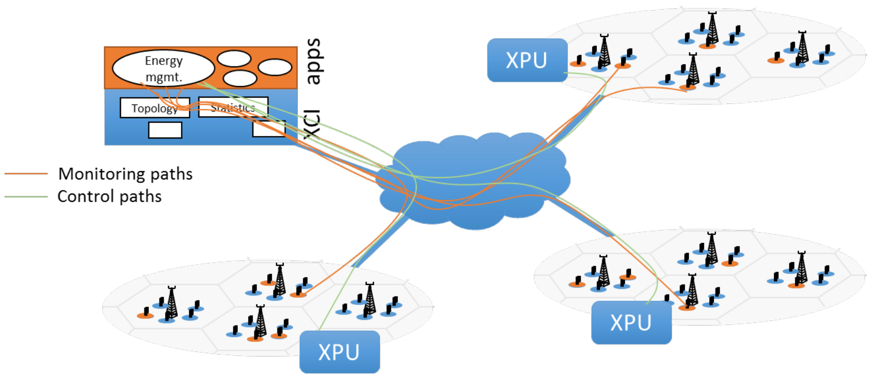
3.5 Energy Management and Monitoring Application

3.5.1 Description of EMMA

The EMMA application provides two main functions:

- Energy monitoring for physical and virtual infrastructures in the integrated fronthaul/backhaul environment, considering both XFE and XPU resources.
- Energy management for power consumption optimization through energy-aware routing and VNFs' placement selection.

The following table provides a detailed description of EMMA.

Title	*Energy-Management and Monitoring Application (EMMA)*
Description of the Application	EMMA is a network application with the main goal of monitoring energy parameters of RAN, fronthaul and backhaul elements, estimating energy consumption and triggering reactions to optimize and minimize the energy footprint of the virtual Crosshaul network while maintaining QoS for each VNO or end user. Together with energy specific parameters like power consumption and CPU loads, EMMA will also collect information about several network aspects such as traffic routing paths, traffic load levels, user throughput and number of sessions, radio coverage, interference of radio resources, and equipment activation intervals. All these data will be used to compute a virtual infrastructure energy budget to be used for subsequent analyses and reactions/optimizations. The application will optimally schedule the power operational states and the levels of power consumption of Crosshaul network nodes, jointly performing load balancing and frequency bandwidth assignment, in a highly heterogeneous environment. Also the re-allocation of virtual functions across the Crosshaul will be done as part of the optimization actions, in order to move fronthaul or backhaul virtual network functions in less power consuming or less loaded servers, thus reducing the overall energy demand from the network.
Picture	 *Figure 15: Energy-management and monitoring application*
Business or technique drivers	The main driver is the reduction of the grid energy consumption across all Crosshaul components, which has a major environmental impact and a sizable cost for operators.

Use cases and Scenarios	***Multi-tier networks, with dense small-cell deployment.*** A scenario of interest features areas served by multi-tier networks, with dense small cell deployment, where the reduced coverage allows for minutely-designed sleep-cycle management algorithms. Relying on the separation of data and control planes in the Crosshaul, signaling and data services can be independently provided by either macro or small cells, and can be implemented as separate, independent services in the same physical equipment while the reconfiguration allowed by SDN controllers will aim at minimizing the control overhead for a given traffic configuration. Wireless fronthaul and backhaul for small cell BSs are an effective way to reduce deployment cost of Crosshaul networks. In the case of wireless Crosshaul, the energy consumption can be reduced by dynamic ON/OFF (sleep) control of wireless links with adaptive routing in accordance with traffic demands of small cell BSs. The use of energy harvesters to supply the small cells to further reduce the network energy cost is also envisaged in the multi-tier network scenario. Control algorithms for traffic offloading and sleep-cycles shall be designed for the Crosshaul architecture to optimally manage the energy inflow and spending. **Re-allocation of VNFs between fronthaul and backhaul to optimize energy footprints.** At virtual network instantiation, the overall energy footprint of the virtual network is computed to decide an energy-efficient allocation of the Crosshaul virtual network topology and VNFs. Energy budget is one of the metrics to minimize when selecting the physical infrastructure elements (computing, storage, network) where to allocate the VNFs and the other virtual network entities. Then, starting from a given virtual fronthaul/backhaul topology with a given instantiation of Crosshaul VNFs, energy and network monitoring parameters are collected and aggregated on different bases (starting from physical infrastructure indicators and mapping/measuring related logical indicators for the virtual infrastructure, per virtual tenant). When configured thresholds are crossed/exceeded, an optimization process starts by re-allocating/moving functions, re-balancing energy loads in the network and consequently switching off physical resources in the Crosshaul data plane layers (where possible, i.e. when no virtual services are insisting on them).	
Technologies to be used or considered	This functionality should be agnostic on the transport technology and be mapped in abstract energy parameters of the Crosshaul infrastructure components, linked to the specific (i.e., per-technology) indicators. IETF EMAN, as well as existing studies in the literature, can be a starting modeling reference for a potential energy-wise data model.	
	What	***Who may provide this***

Information requirements **(Inputs)**	Energy related constraints, metrics and objectives, expressed through operator policies.	Management interface, Multi-Tenancy application
	Energy-related information for physical network devices and XPUs. This info could be either dynamic values, such as traffic load and measured amounts of harvested energy, or static values, such as available power states of a device, including the related available functionalities and power limitations, types and capabilities of harvesting and energy storage, etc. [Novel energy consumption models for Crosshaul components and technologies may be required. Input/interaction with WP2 is expected in order to assess such a need].	XCI, through NBI (SDN controller for network related information)
	Information about physical network topology	XCI (SDN controller), through NBI
	Information about interference level in the RAN	RAN controller, through EastWest interface (not in 5G-Crosshaul scope).
	Information about XPU capabilities.	XCI (VIMaP), through NBI.
	Information about: • Virtual Infrastructures. • Mapping between physical and virtual network resources for the established Virtual Infrastructures.	MTA
Outputs	***What***	***Who may need this***
	Power consumption from grid vs harvested energy whenever applicable: • For Virtual Infrastructure • For Physical Infrastructure (or its segments)	The operator (values available from app APIs) Mobility application EMMA itself, as input for further actions

<table>
<tr><td rowspan="3"></td><td>Specification of ON-OFF status of links/nodes</td><td>EMMA itself, as input for further actions</td></tr>
<tr><td>Specifications of virtual infrastructure re-configuration
New selection and configuration of VMs for VNFs</td><td>VIM, as input to modify the current virtual infrastructure</td></tr>
<tr><td>Commands to regulate the power state/level of the devices</td><td>XCI (SDN controller for network devices)</td></tr>
<tr><td>App-specific functions</td><td colspan="2">Computation of power consumption metrics associated to virtual resources starting from power-related measurements from physical devices, as provided by the SDN controller (network side) and the VIM (IT side).
Computation of global power consumption and fraction of global power consumption from the grid for each virtual infrastructure and for the underlying physical infrastructure (or parts of it).
Automatic detection of inefficient allocation of resources from the power consumption perspective (based on operator policies).
Computation of optimal placement of virtual infrastructures and services over Crosshaul physical infrastructure, considering power consumption related objective functions. As option, this function can be provided by external entities through dedicated, multi-objective algorithms.
Instructions for re-optimization in deploying Crosshaul virtual infrastructures.
Instructions for changing the power state/level of a physical network device or server as a consequence of changes in its traffic/computational load.
Instructions for RAN power settings can be issued with the goal of energy consumption reduction (out of 5G-Crosshaul scope).</td></tr>
<tr><td>Required
(external) functions</td><td colspan="2">Multi-tenancy application (to create the initial Virtual Infrastructures)
Virtual Infrastructure Manager
Statistics and metrics collection related to interference in the RAN (out of 5G-Crosshaul scope), harvested energy, traffic demand [To be developed within WP3].</td></tr>
</table>

Functional Architecture

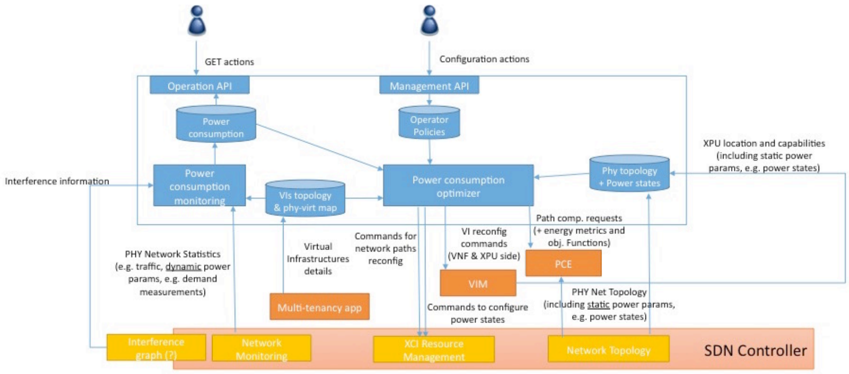


Figure 16: Energy-management and monitoring application function description

Vis topology & phy-virt map: database storing information received from MTA on virtual infrastructure topology and mapping between physical and virtual resources.

- Power consumption monitoring takes as input the above information, as well as physical network statistics related to time-varying parameters (e.g. traffic demand, level of harvested energy, interference level in the RAN – out of 5G- Crosshaul scope). It then computes and updates energy-related metrics, which are stored in the power consumption database. Such metrics can be used by operators to change/refine the collection of their policies.
- Power consumption optimizer devises the optimal physical and virtual network reconfiguration in order to reduce energy consumption of the whole physical infrastructure. It does so by exploiting previously described information, as well as operator's policies and static information on the physical network topology and power states/capabilities of network devices.
- The optimizer provides the envisioned energy-efficient reconfiguration of the physical network devices to the SDN controller, which should act as an arbiter in case of contradicting/overlapping reconfiguration commands generated by different apps (for example, a command to switch off a device which still has active flows should be denied by the SDN controller). The optimizer also provides instructions on virtual infrastructure reconfiguration to the VIMaP. The EMMA should also compute energy-efficient network paths and VNF allocation. However, this function could be delegated to an external Path Computation Element (as in the picture example), exploiting their routing algorithms. In this case, the PCE native objective functions and internal metrics should be feed with suitable energy-related parameters computed by the EMMA.
- Note that the EMMA could be de-/activated depending on the operator policy.

Implementation	An analytical approach supported by software emulation will be chosen as initial methodology. A PoC of the implemented algorithms (i.e. porting of an algorithm in the designated/selected SDN controller framework and/or Virtual Infrastructure orchestrator) The developed EMMA will be integrated into the mmWave meshed backhaul demonstrator that will be developed in WP5.
Relation to other applications	• **VIMaP**: EMMA should make use of the functions exposed by the Virtual Infrastructure Manager for instantiation of VMs and SFCs and for their reconfiguration (triggered during re-optimization procedures). Suitable APIs should be exposed by the VIMaP to enable: • Configuration of monitoring agent at the physical infrastructure and VM level to enable the collection of relevant energy parameters • Programmability and selection of the XPUs where the VNFs will be deployed • **RMA**: EMMA should also interact with the RMA on decisions about: ◦ Power state/level of physical devices ◦ Load balancing and traffic offloading.

3.5.2 Design of Power Consumption Optimizer Algorithms in EMMA application

The Power Consumption Optimizer (PCO) in EMMA aims at minimizing energy consumption by: (i) turning off drivers of links whenever possible, resulting in proportional (possibly non-linear) changes, and (ii) turning off those nodes whose links are inactive.

Both approaches can be studied by building a directed network graph where vertices represent the network nodes and edge correspond to links connecting the nodes. Let us then assume that the network is characterized by the parameters in Table 1 (a) and its status is described by the variables in Table 1 (b). In particular, the power consumption associated with link (i,j), denoted by $P(i,j,t)$, linearly depends on the traffic flowing over the link. Its value, as well as the power consumption of a node in idle state (P_{idle}), can be computed based on some parameters characterizing OpenFlow switches and combining the models presented in [7] and [8]. We refer the reader to the [2] document for further details on the power consumption model we adopted. The total power consumption of a node in active state is then given by Eq. (1). The traffic flowing over a link at a given time instant is expressed in bit/s and it is given by the sum of the traffic associated with all flows that are routed through the link.

Table 1: EMMA Network Parameters (a) and Variables (b)

Parameter	Description
N, L	Number of nodes and links
$\mathcal{N}$, $\mathcal{L}$	Set of nodes and links
$(i,j) \in \mathcal{L}$	Link from node i to node j
$C(i,j)$	Capacity of link (i,j)
$\mathcal{F}(t)$	Set of active traffic flows at time t
$f^{sd} \in \mathcal{F}(t)$	Active flow between source s and destination d
$V(f^{sd})$	Traffic volume of flow f^{sd}
$\boldsymbol{\pi}$	Generic path given by an ordered sequence of links
P_{idle}	Power consumption of an idle node
$P(i,j,t)$	Power consumption associated with link (i,j) at t

Variable	Description
$x_{(i,j)}(t)$	Takes 1 if link (i,j) is on at time t, 0 else
$y_i(t)$	Takes 1 if node i is on at time t, 0 else
$z_{\boldsymbol{\pi},f^{sd}}(t)$	Takes 1 if f^{sd} is routed through path $\boldsymbol{\pi}$ at time t, 0 else
$\tau_{ij}(t)$	Traffic flowing over link (i,j) at time t

The computation of the traffic is reported in Eq. (2).

$$P(i,t) = P_{idle} + \sum_{j\in\mathcal{N}, j\neq i} [P(i,j,t)x_{ij}(t) + P(j,i,t)x_{ji}(t)] \tag{1}$$

$$\tau_{ij}(t) = \sum_{f^{sd}\in\mathcal{F}(t)} \sum_{\boldsymbol{\pi}:(i,j)\in\boldsymbol{\pi}} V(f^{sd}) z_{\boldsymbol{\pi},f^{sd}}(t)\,. \tag{2}$$

Given the previous definitions, we formalize the problem as an optimization problem where the instantaneous power consumption of the network (i.e., the sum of the power consumption due to active nodes and links) is minimized, subject to the system constraints. The formulation is reported in (3)-(7). In particular, we impose the following:

1. flow conservation at every node (4);
2. the traffic flowing over a link cannot exceed the link capacity (5);
3. a link can be active only if both its end nodes are active (6) (in (6) *M* is a large, positive constant);
4. the power consumption at any node cannot exceed its value of peak power (7).

Objective

$$\min \sum_{i \in \mathcal{N}} y_i(t) P_{idle} + \sum_{(i,j) \in \mathcal{L}} x_{ij}(t) P(i,j,t) \qquad (3)$$

Constraints

- Flow conservation constraint, for any $j \in \mathcal{N}$:
  $$\sum_{i \in \mathcal{N}, i \neq j} \tau_{ij}(t) x_{ij}(t) - \sum_{k \in \mathcal{N}, k \neq j} \tau_{jk}(t) x_{jk}(t) = \sum_{f^{sd} \in \mathcal{F}: j=d} V(f^{sd}) - \sum_{f^{sd} \in \mathcal{F}: j=s} V(f^{sd}) \qquad (4)$$
- Capacity constraint, $\forall (i,j) \in \mathcal{L}$:
  $$\tau_{ij}(t) \leq C_{ij} \qquad (5)$$
- Link activation constraint, $\forall i \in \mathcal{N}$:
  $$\sum_{j \in \mathcal{N}, j \neq i} [x_{ij}(t) + x_{ji}(t)] \leq M y_i \qquad (6)$$
- Peak power constraint, $\forall i \in \mathcal{N}$:
  $$\mathrm{P}_{idle} + \sum_{j \in \mathcal{N}, j \neq i} [P(i,j,t) x_{ij}(t) + P(j,i,t) x_{ji}(t)] \leq P_{peak} \qquad (7)$$

Note that, given the variables definition and the fact that *P(i,j,t)* depends on the traffic flowing over a link, which in its turn depends on the variables *z*'s, the problem turns out to be an integer non-linear problem. Due to its complexity, obtaining the optimum solution may not be viable in general cases. In light of this, we design a heuristic algorithm, which ensures an efficient solution in the presence of any network scenario.

To this end, we leverage the First Fit algorithm [Fang] and design a heuristic scheme that efficiently allocates traffic flows, starting with the largest ones, with the aim to minimize the length of the flow path and the energy consumption of the overall network. Given a flow, our heuristic first tries to fit the flow into the existing active network while meeting the flow QoS requirements. It then turns on other links and/or nodes only if no suitable path is found. Also, every time a new link and/or node are added to the "active network", the EMMA PCO looks for a better alternative path for flows that have started a given time ago (again, processing the largest flows first). This time is set to half the duration of the average flow duration. If a more energy-efficient allocation is found, then flows may be diverted on alternative paths, provided that their QoS requirements are still met. Note that the PCO differs from the First Fit algorithm since it tries to find a better path for already allocated flows whenever any change in the active topology occurs.

Definition of basic algorithms	The EMMA PCO is composed of two algorithms. **Algorithm 1** New flow allocation **Require:** Topology, new flow, network power state, traffic load **for** each new flow **do** Compute all shortest paths across active topology **if** suitable path is found **then** % if more than one, select one at random Allocate the flow **else** Compute shortest paths considering the whole network **if** suitable path is found **then** % if more than one, select one at random Turn on the selected links and nodes that are off Allocate the flow installation_time ← current_time Run Algorithm 2 % It moves previous flows to a better path if any

<table>
<tr>
<td></td>
<td>
Algorithm 2 Move flows to a better path
Require: Topology, information on current flows
1: $\mathcal{S} \leftarrow$ {flows s.t. installation_time $\geq T_a$}
2: Order flows in $\mathcal{S}$ with decreasing rate requirements
3: for each flow in $\mathcal{S}$ do
4:     Search a path on the active topology
5:     if suitable path exists $\wedge$ (new path cost $<$ old path cost) then
6:         Move flow from old to new path
7:         installation_time $\leftarrow$ current_time
8: if there are links and/or nodes no longer carrying traffic then
9:     Turn them off

Algorithm 1 presents the sequence of actions to be taken whenever a new flow has to be activated in the network. Input to the algorithm are the topology, the new flow, information about the power state of network switches and the traffic crossing every link (line 1). The computation of the possible paths for the incoming flow will be done in the active network (line 2). Then, if a path is found on the active network and it can support the required QoS, install the flow (lines 3-5). If no path is found on the active network, consider the whole network to find a path with the required QoS for the new flow (line 7). If a path is found, activate the links and nodes that are newly added to the network, install the new flow, record the flow install time and trigger the flow rerouting algorithm (lines 8-14).

Algorithm 2 states the steps followed during rerouting existing traffic. This algorithm is run whenever there is a change in the active network topology, i.e., if nodes and/or links become active while finding a path for a new flow. Input to the algorithm are the topology and flows in the network (line 1). The algorithm then selects all flows that have started at least T_a seconds ago (line 2). For each flow satisfying the start time hysteresis compute a path on the current topology (lines 3 - 4). If the cost of the computed path is less than the cost of the path where the flow is routed initially, move the flow to the new path and update the flow installation time (lines 5 - 7). If the process of moving flows to a different path results in some links and/or nodes being idle, turn off those links and/or nodes (lines 8 - 9).
</td>
</tr>
<tr>
<td>Requirements for XCI NBI</td>
<td>
• SDN Controller
The SDN controller has to provide REST APIs at the NBI in order to:
  • collect information from the XPFEs network topology, network statistics, monitoring and analytics services;
  • configure the power states of XPFEs network devices;
  • establish network paths for inter-VM connectivity.
• NFVO
The NFVO has to provide rest APIs in order to:
  • collect information from the IT infrastructure inventory service;
  • configure the power states of XPUs;
  • provision and manage VNFs and Virtual Network Services (VNSs).
</td>
</tr>
</table>

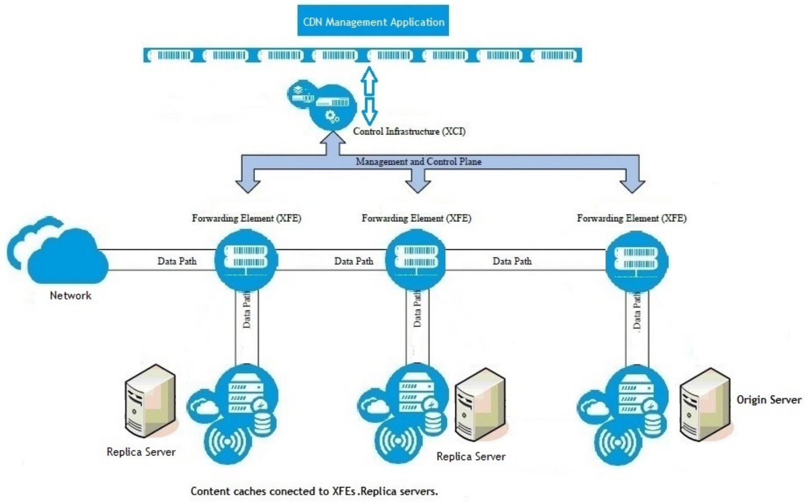
Requirements for NBI / SBI (Application)	• **NBI** Defined in section 8.1.5. • **SBI** The EMMA application will interact with the XCI and with the VIMaP application in order properly actuate its services of re-optimization of resources in the data-plane. In particular, APIs exposed by the VIMaP will be used in order to: • Configure and monitor the physical/virtual infrastructure, collecting energy monitoring data • Select and configure the XPUs in the deploying of VNFs/NSs
Data Plane Requirements	EMMA requires power consumption monitoring capabilities at the data plane. If this kind of capability is not available, power consumption will be estimated based on traffic statistics for XFEs and load for XPUs.

3.6 Content Delivery Network Management Application

3.6.1 Description of CDNMA

The CDNMA is an Over-The-Top (OTT) application related to the distribution of media contents, especially Video on Demand (VoD), that uses the services and APIs offered by other applications and the XCI NBI, SDN controller and NFV MANO, in particular the NFV-O (Orchestrator), to manage the vCDN infrastructure configuration and the content delivery rules on the 5G-Crosshaul network. The XPUs will host the CDN nodes on virtualized server instances.

Title	*CDN Management Application (CDNMA)*
Description of the Application	The CDN management application will control the load balancing over several replica servers strategically placed at various locations in order to deal with massive content requests. The CDNMA will improve the content delivery maximizing bandwidth and improving accessibility through the vCDN infrastructure according the user demands.

Picture	 *Figure 17: vCDN deployment over the 5G-Crosshaul infrastructure*
Business or technique drivers	Content distribution, especially video traffic is expected to be the dominant contributor to the mobile data traffic demand, therefore content media distribution is being more and more present in everyday life communications, anywhere, any time and in end-user multi-device environments. Especially relevant scenarios for content delivery are massive events such as international music festivals or big sport events, such as football World Cup or the Olympic Games, in which high volume of traffic is needed maintaining and acceptable QoS/QoE for the end user. Media distribution optimization is strongly related to resource management.
Use cases and Scenarios	**CDNs (Content Delivery Networks)** A content delivery network (CDN) is an interconnected system of cache servers located strategically for delivering content. It is a combination of a content-delivery infrastructure (in charge of delivering copies of content to end-users), a request-routing infrastructure (which directs client requests to appropriate replica servers) and a distribution infrastructure (responsible for keeping an up-to-date view of the content stored in the CDN replica servers). In 5G-Crosshaul, the content delivery infrastructure will be implemented via networks of content caches -replica servers- which are deployed close to the XFEs across the 5G-Crosshaul topology (please refer to Figure 17). The origin server will be located in the 5G-Crosshaul network too. The request-routing and distribution infrastructure functions will be enforced through optimal content routing and delivery based on information from the 5G-Crosshaul network. A content provider accesses the 5G-Crosshaul framework and requests through the CDN management application a vCDN infrastructure. The CDNMA would receive information from the network controller about the network topology, information from the vCDN infrastructure about the CDN nodes status and information about user´s location. Taking this information as input and applying optimization logic defined by the CDN

<table>
<tr><td></td><td colspan="2">operator based on the CDN metrics and network metrics, the application strategically deploys and optimizes the replica servers in order to optimize resource utilization and deliver the content to the end users/content consumers with maximum possible QoS/QoE.

The CDNMA will take advantage of the 5G-Crosshaul network design because the application will be network-aware, so it will be able to provide a faster service to the end user.

The CDNMA running on the XCI will manage the entire CDN infrastructure and it will evaluate the system performance; the CDNMA will receive monitoring information from the infrastructure, the CDN nodes and other applications (location, latency, packet-loss, bandwidth, server load).

Stakeholders: Content provider, Network provider, CDN operator, End users or clients.

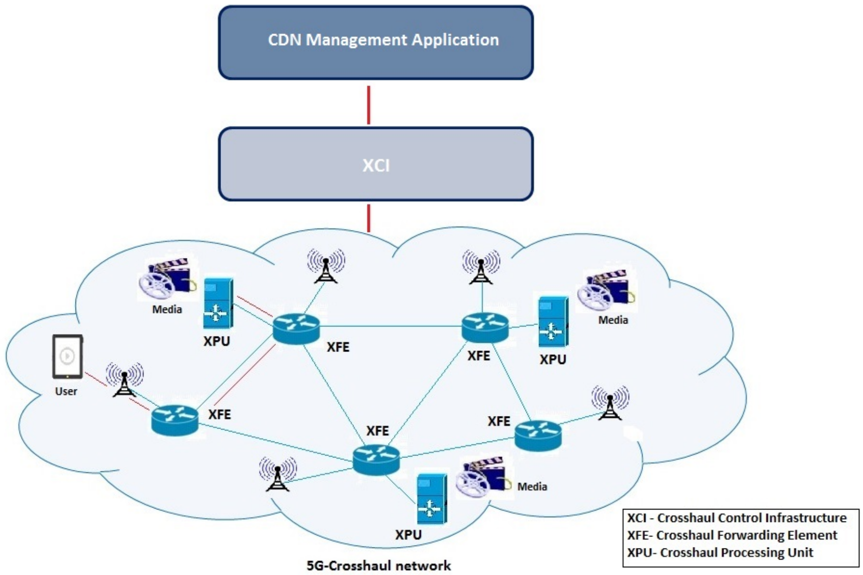


Figure 18: Media distribution scenario: Content Delivery Network (CDN)</td></tr>
<tr><td>Technologies to be used or considered</td><td colspan="2">Cache servers will be used to deliver the media content for the end users. Routing techniques will be considered in order to optimize the resource utilization.

The used transport technologies should be transparent for the application.</td></tr>
<tr><td rowspan="2">Information requirements

(Inputs)</td><td>What</td><td>Who may provide this</td></tr>
<tr><td>Global view of the network infrastructure:

• Nodes (XFEs, XPUs):

• Status (up, down)</td><td>• XCI may provide information on network infrastructure (through NBI)</td></tr>
</table>

	• Connections • Links: • Status (up, down) • Connection points • Bandwidth, rate of data transfer (Mb/s)	• Or another WP4 application with discovering mechanisms may provide the required information.
	Global view of the vCDN infrastructure: • CDN nodes (Origin server, Replica servers): o Status (up, down) o CPU load (%) o Memory load (%) o Number of connected users (n)	• XCI may provide information on vCDN infrastructure (through NBI) • Or specific monitoring mechanisms may provide the required information.

	User information: • Location. User network entry point.	• XCI may provide information on user (through NBI) • Or another WP4 application with discovering mechanisms may provide the required information.
Outputs	**What**	**Who may need this**
	vCDN infrastructure configuration CDN node assigned Content delivery rules	• XCI shall enforce this into the network • Or another WP4 application
App-specific functions	• Content Distribution and management functions ○ Origin and Replica servers activation ○ Content synchronization ○ Content selection and delivery • Request-routing algorithms -> Adaptive, consider the current system condition to select a replica server for content delivery. Metrics: load, congestion, location… • CDN operator policies and constraints • CDN metrics monitoring	
Required (external) functions	○ Joint BH/FH optimization (routing, scheduling, load balancing, etc.) and dynamic network configuration • Monitoring of 5G-Crosshaul network elements	
Function Architecture		

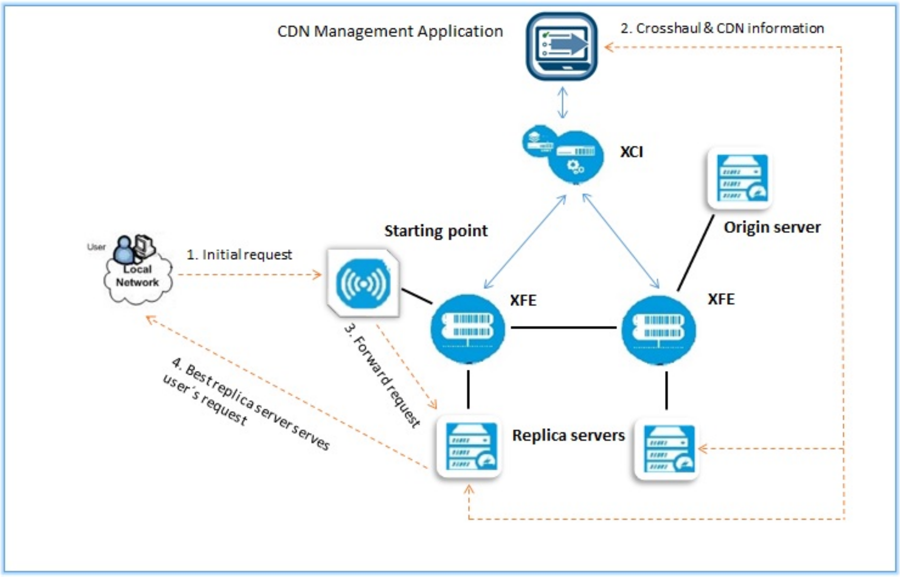
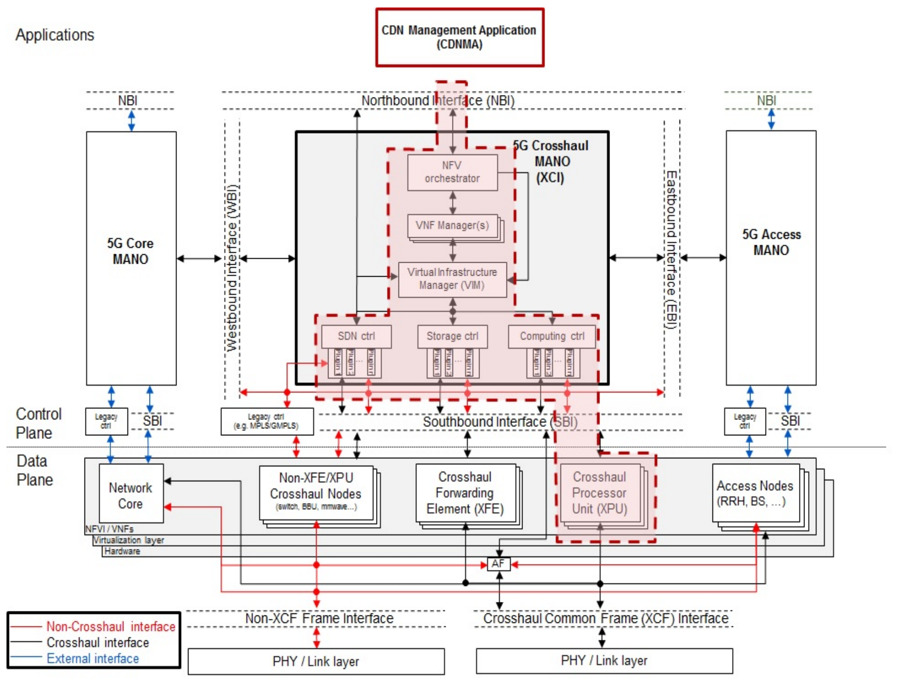
<table>
<tr><td></td><td>

Figure 19: Content delivery network management application function architecture</td></tr>
<tr><td>**Implementation**</td><td>In order to develop a complete virtual CDN infrastructure and test its performance on the 5G-Crosshaul network as one of the main procedures for the media distribution, different tests will be carried out and a proof of concept will be integrated and evaluated on the 5G Berlin and 5TONIC Madrid testbeds.</td></tr>
<tr><td>**Relation to other applications**</td><td>This application could interact with the following applications:
• **Mobility Management Application (MMA):** The MMA would provide the user network entry point to the CDNMA.
• **Resource Manager Application (RMA):** The RMA would mainly deal with the network resources to compute the optimal paths between the user and the CDN server assigned.
• **Multi-tenancy Application (MTA):** MTA would provide individualized information of resources utilization for each tenant, i.e. for each CDN provider.
• **Virtual Infrastructure Manager and Planner (VIMaP):** The VIMaP would place the CDN nodes (server instances) in the location requested and would interconnect them follow a given graph across in the 5G-Crosshaul network.</td></tr>
</table>

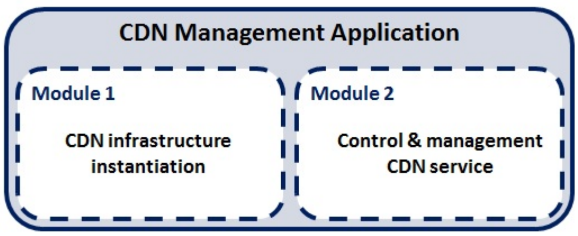
Application mapping on WP1 system architecture (High level view)



3.6.2 Design of CDNMA Algorithms

Definition of basic algorithms

The CDNMA application is a Web application composed of two basic algorithms that allow the CDN service management and implementation over the 5G-Crosshaul network. The first algorithm is in charge of the CDN infrastructure instantiation, and the second one takes care of the control and management of the CDN service during the lifetime.



- **Module 1. CDN infrastructure instantiation**

  Function: Deployment of CDN nodes on virtualised server instances connected through one or more VNF-FGs in the 5G-Crosshaul network.

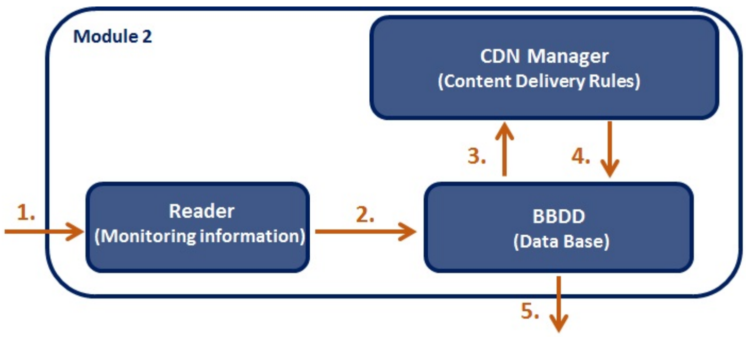
  Algorithm:

  Module 1

  Begin

  Get Network Topology from XCI

  (List of all the switches, hosts and links from the

	topology manager) **if** XPUs in Network Topology available **then** **for** each CDN node type *(Origin node, Replica node)* Select best XPU for deployment *(The most suitable computing nodes to deploy the CDN nodes based on the criteria (geographical location, coverage area, dense urban environments…) defined by the CDN operator)* Select VNF Descriptor *(VNFD_origin, VNFD_replica)* **end for** Set CDN VNF-FG configuration *(Logical description of interconnecting CDN VNFs and traffic flow between them)* Send complete CDN infrastructure description to XCI *(VNFD_Origin, VNFD_Replica, CDN VNF-FG)* **else** error CDN infrastructure instantiation not possible **end if** **End** • **Module 2. Control & Management of the CDN service** Function: Control and management of the content delivery rules over the CDN nodes placed in the network, based on the monitoring information received from the network infrastructure and the logic defined by the CDN operator. 

	Algorithm: Module 2 **Begin** Get Monitoring information from the network *(CDN infrastructure information, user information)* Store the information on the DB Define content delivery rules { **for** each user request Check Network Entry Point and Nearest CDN node **if** Nearest CDN node OK **then** *(Status = up, CPU load (%) < x % , memory load (%) < y %, user number < z)* Associate CDN node **else** Find suitable CDN node *(Status, location, CPU load , memory load, user number)* Associate CDN node **end if** **end for** } Update content delivery rules on the DB Send content delivery rules updated *(CDN node assignments)* **End**

<table>
<tr><td>Requirements for XCI NBI</td><td>

SDN Controller
The interface has to provide methods (API REST…) to get the network topology, to retrieve a list of all known nodes, hosts and links from the topology manager.

Get nodes and all the links getting out/in from/to the node
XFEs and port connections
Get all link connections between nodes including their properties
Links (connection points, Bandwidth)
Get hosts and the node connections
XPUs and port connections

NFVO NBI
The interface has to provide methods (API REST…) to interact with the NFVO.
The NFVO will receive a Network Service Descriptor (NSD) from the CDNMA application to instantiate the CDN service.
The Network Service Descriptor (NSD) will reference all other descriptors which describe components that are part of the CDN service.

CDN Node VNF Descriptor
CDN VNF-FG Descriptor

</td></tr>
<tr><td>Requirements for NBI / SBI (Application)</td><td>

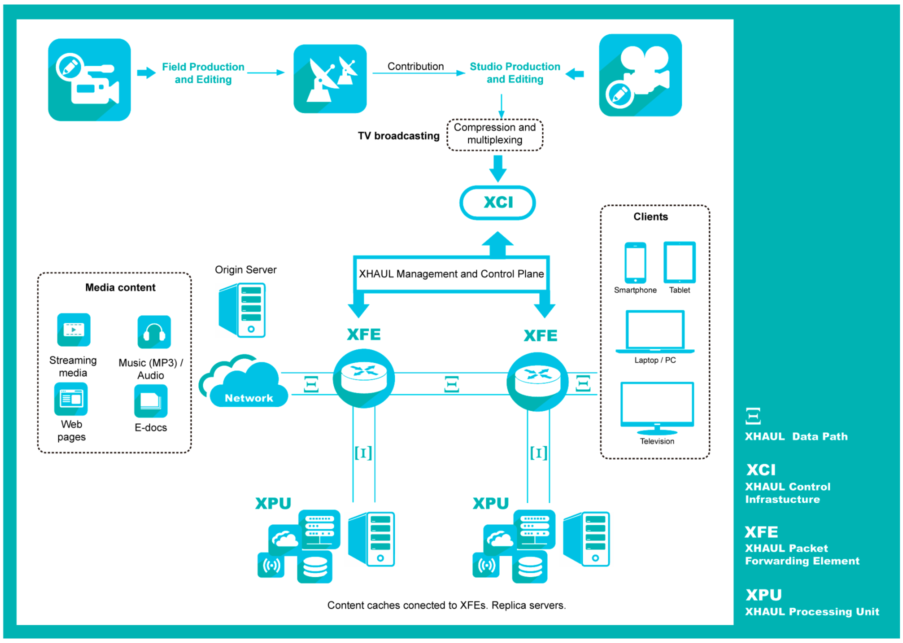
NBI
It is an Over-the-Top (OTT) application. The application will show towards the operator (e.g. the CDN operator) a graphic interface running on a REST API for management and operation actions.
SBI
The CDNMA application will interact with some applications and functions to interchange valuable information to perform the service properly:
Mobility Management Application (MMA)
The CDNMA application will receive the user network entry point from the MMA application and it will decide the best CDN node to service the user request.
The CNDMA application will send back the CDN node assignment to the MMA application to find the best network path to provide the service.
VNFs monitoring information

</td></tr>
</table>

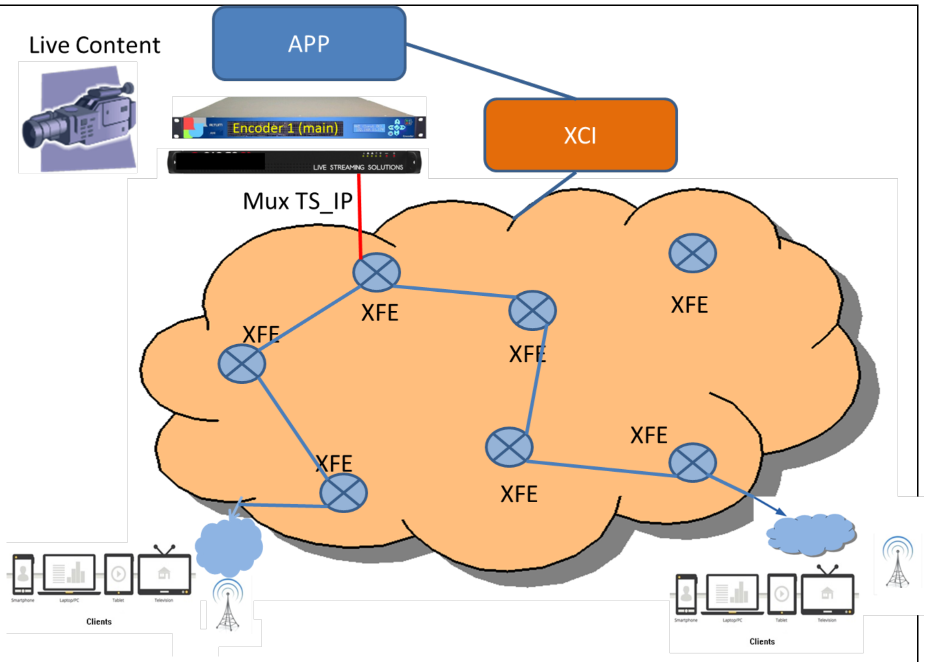
	The CDNMA application will receive specific monitoring information from the different CDN node VNFs placed in the network. The monitored information will be: • Status (up, down) • CPU load (%) • Memory load (%) • Number of connected users (n)
Data Plane Requirements	The CDNMA doesn´t have specific requirements for the data plane. The underlying data frame should be transparent for the application.

3.7 TV Broadcasting Application

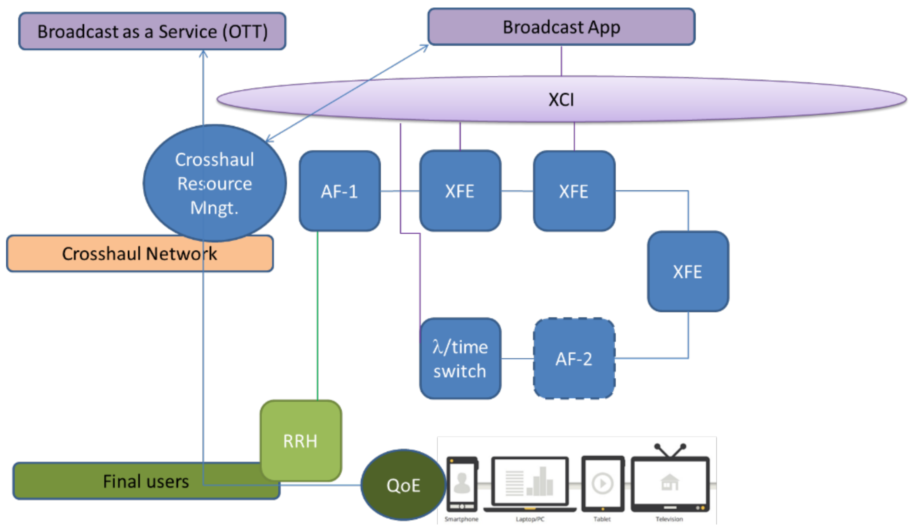
3.7.1 Description of TVBA

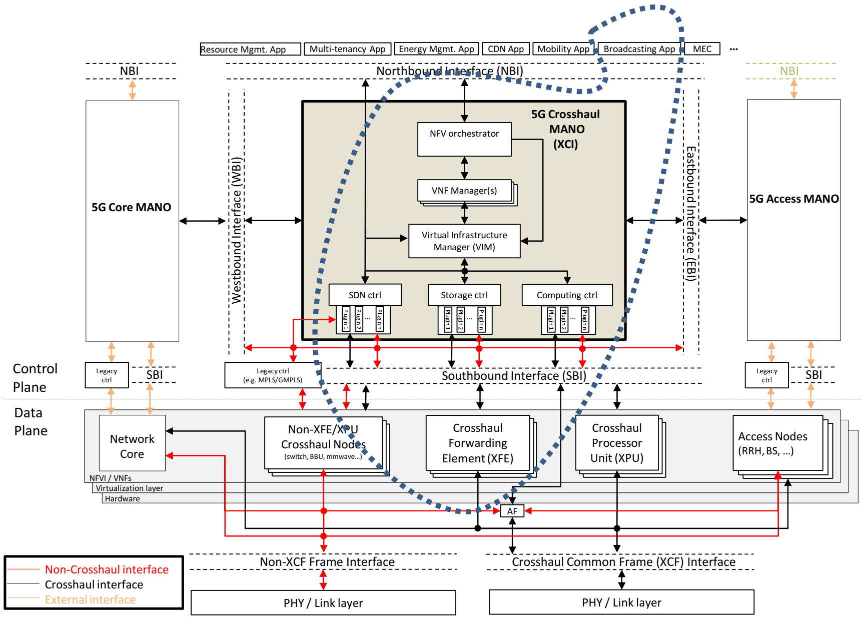
Title	*TV Broadcasting Application (CDNMA)*
Description of the Application	TV Broadcasting application aims to provide a solution for broadcasting built upon 5G-Crosshaul. The focus is on minimizing both the cost and the spectrum consumption of the next generation TV and broadcast-like content delivery that will be capable of addressing the future needs of Media/TV broadcasting & multicasting services utilizing the Crosshaul architecture.
Picture	
Business or technique drivers	Video is expected to contribute ~70% of all the mobile traffic by 2018. Content distribution is expected to be the dominant contributor to the mobile data traffic demand, therefore content media distribution is being more and more present in everyday life communications, anywhere, any time and in end-user multi-device environments and TV broadcasting and mobile broadband are undoubtedly essential parts of this.

	The trends of on-demand, mobile, and Ultra HD quality impose formidable challenges for TV and the delivery network of the future to be coped for 5G-Crosshaul.
Use cases and Scenarios	**Media use case: TV Broadcasting and multicasting** A TV broadcasting/multicasting service is offered starting from the content of a live-source which is processed till be finally transcoded to the objective format and bit rate (image resolution, scan format, video encoding standard and transport stream bit rate) and injected in the Crosshaul network. TVBA is a dedicated application that leads up to the TVB service definition and supervision over the 5G-Crosshaul network, it is in charge of the TVB service establishment by accomplishing the provided requirements (quality, resolution, bit rate/bandwidth, coverage...), the video injection and the control and management of the service (location, latency, packet-loss or bandwidth). The broadcast as a service application will take advantage of the Crosshaul network design, so it will be able to provide a faster service to the end user taking care of providing an acceptable QoE. The network is configured by selecting the forking nodes of the tree to optimize the content delivery and make sure a real-time delivery with the lowest possible delay is offered to the users. The control plane consists of the XCI, which provides the control and management functions to operate on the 5G-Crosshaul physical infrastructure; network nodes (XFEs) and processing units (XPUs). The XCI is composed of a SDN controller and a specific NFV architecture that will interact with the application plane through its NBI. The use of SDN (Opendaylight will be the SDN controller) effectively support in the data plane that processes besides network parameters and QoE monitoring which is needed for the reconfiguration. TVBA will interchange specific information about the different nodes and links of the network in order to control the network path of the broadcast tree.

<table>
<tr><td></td><td colspan="2">
</td></tr>
<tr><td>Technologies to be used or considered</td><td colspan="2">Video technologies for live content production as TV head-end
SDN controller
Algortihms for routing and building the broadcast tree</td></tr>
<tr><td rowspan="2">Information requirements
(Inputs)</td><td>What</td><td>Who may provide this</td></tr>
<tr><td>Global view of the network infrastructure:
➢ Nodes (XFEs, XPUs):
• Status (up, down)
• Connections
➢ Links:
• Status (up, down)
• Connection points
• Bandwidth, rate of data transfer (Mb/s)
• Load (%)
• Mean delay (ns)
• Mean packet loss rate (%)</td><td>XCI may provide information on network infrastructure (through NBI)
• TVBA or RMA with discovering mechanisms may provide the required information.</td></tr>
</table>

	User information: ➢ Location. User network entry point.	XCI may provide information on user (through NBI) MMA or another WP4 application with discovering mechanisms may provide the required information.
Outputs	**What**	**Who may need this**
	Content Delivery Rules: ➢ Routing rules ➢ Policies and constraints	XCI may enforce XFEs and involved servers follow the content delivery rules
App-specific functions	• TV service configuration through TV-Headend • Media workflow managing and monitoring • Broadcast tree routing algorithms • QoE metrics for the service	
Required (external) functions	• Joint BH/FH optimization (routing, scheduling, load balancing, etc.) and dynamic network configuration • Monitoring of Crosshaul network elements	

Function Architecture	
Implementation	Media technical means as head-end will be provided by Visiona. Content should be selected avoiding rights issues. Prototype of TV Broadcasting and Multicasting should be done starting from a simple Crosshaul topology. Different levels of topology complexity will be progressively added. First, a proof of concept will be developed in Madrid and later, it will be analysed if will be integrated and evaluated on the 5G Berlin test-bed.
Relation to other applications	RMA can provide improved algorithms for routing that can be analyzed within the TVB service. MMA is providing user information that can be used to improve the efficiency of the TVB service deployment.
Application mapping on WP1 system architecture (High level view)	This application aims to provide a solution for broadcasting built upon 5G-CrosshaulThe focus is on minimizing both the cost and the spectrum consumption of next generation TV and broadcast like content delivery. The broadcast as a service application will take advantage of the 5G-Crosshaul network design because the application will be network-aware, so it will be able to provide a faster service to the end user taking care of providing an acceptable QoE.

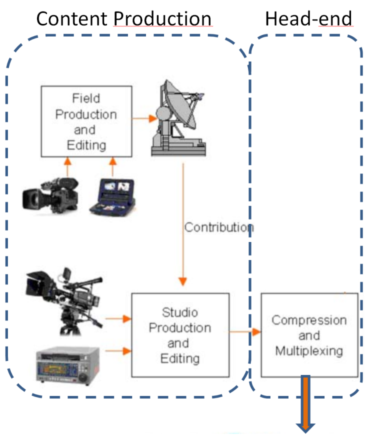


3.7.2 Design of TVBA Algorithms

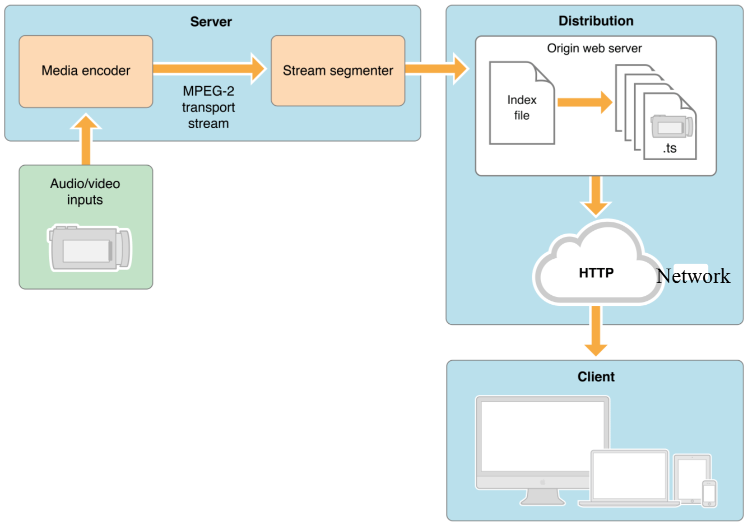
Definition of basic algorithms

The "*TV Broadcast application*" is a dedicated application composed of two algorithms that leads up to the TVB service definition and supervision over the 5G-Crosshaul network. The first algorithm is in charge of the TVB service establishment by accomplishing the provided requirements (quality, resolution, bit rate/bandwidth, coverage,…) and the video injection. The second one is oriented towards the control and management of the service during the lifetime.

- 1st Block. TV service definition and establishment
    - TV Head-end



    - Real-time adaptive streaming



Function: Live video service generation and injection.

- 2nd Block. Control & Management of the TVBA service

  Function: Control and management of the TV broadcasting supervising the quality of the service. It is built upon the monitoring information received from the network infrastructure and the fixed service requirements.

  Broadcast Algorithm:

  TVBA (Media Service parameters, Coverage)

  Begin

  Get Network Topology from XCI

  (List of all the switches, hosts and links from the topology manager)

  for each node

  Check suitability for deployment

  (The nodes should fulfil the criteria (geographical coverage, quality of service maintenance in terms of latency and bandwidth …) defined by the service

  end for

  if Complete path in Network Topology available **then**

  Optimize path (e.g. asking RMA to compute an

<table>
<tr><td></td><td>optimized path)
Send complete tree description to XCI
(Nodes, Links)
end if
else
Report the error: broadcast service deployment not possible
end if
While Broadcasting
Get Monitoring information from the network
(Broadcast tree, Content delivery constraints – delivery rules-)
If QoS alarm
Reconfigure Network
end if
Free resources
End</td></tr>
<tr><td>Requirements for XCI NBI</td><td>SDN Controller
The interface has to provide methods (API REST…) to get the network topology, to retrieve a list of all known nodes, hosts and links from the topology manager.
o Get nodes and all the links getting out/in from/to the node
• XFEs and port connections
o Get all link connections between nodes including their properties
• Links (connection points, Bandwidth, latency, delay, jitter)
o Provide QoS evaluation
The TVB application will receive specific monitoring information from the different nodes and links of the network. The monitored information will be:
Status (up, down)
Delay
Bandwidth</td></tr>
</table>

Requirements for NBI / SBI (Application)	• **NBI** It is an Over-the-Top (OTT) application. • **SBI** The TVB application will interact with some applications and functions to interchange valuable information to perform the service properly: ○ Mobility Management Application (MMA) To be discussed Handover and Coverage. ○ Resource Management Application (RMA) To be discussed Routing
Data Plane Requirements	The TVBA needs to be supported by the data plane providing minimum guaranteed requirements in terms of quality of the media service provided.

3.8 Mapping of 5G-Crosshaul applications and WP1 use cases

This section presents how the defined different applications are mapped to the five use cases defined in WP1, based on the main requirements and functions required. The 5G-Crosshaul use cases are described in the following table, based on D1.1 [1].

Table 2: 5G-Crosshaul use cases description.

Use Cases	Brief description
1. Vehicle mobility	Vehicle mobility is related to the 5G signals sent/received in vehicles during their motions. The most challenging situations are passengers using 5G services (video in particular) on a very high speed train (about 500 km/h) and messages among vehicles for emergency and security.
2. Media Distribution: CDN and TV broadcasting & Multicasting	This use case is related to the distribution over 5G networks of media contents, especially video traffic, and TV broadcasting which are expected to be the dominant contributors to the mobile data traffic demand.
3. Dense urban society	This use case takes into account the connectivity required at any place and at any time by humans in dense urban environments, considering both the traffic between humans and the cloud, and direct information exchange between humans or between humans and their environment.
4. Multi-tenancy	This use case contemplates a flexible sharing of backhaul/fronthaul physical resources by multiple network operators or service providers (i.e. multiple tenants). It is the key enabler to maximize the degree of utilization of Crosshaul deployments and minimize the overhead due to the costs of roll-out and maintenance.
5. Mobile edge computing	This use case is focused on the deployment of IT and cloud-computing capabilities within the Radio Access Network (RAN), in close proximity to mobile subscribers. Content, service and application providers can leverage on such distributed computing capabilities to serve the high-volume, latency-sensitive traffic on dense areas concentrating high number of users.

The following table presents the mapping of each use case described in Table 1 with the SDN/NFV applications defined in WP4.

Table 3: 5G-Crosshaul use cases and applications mapping

Use Cases	Main functions required	Required applications
1. Vehicle mobility	Mobility management functions FH/BH resource management functions Multi-tenancy functions	**MMA**: to solve the frequent HandOver (HO) problem challenged by high mobility and high data rate requirements of this use case. The MMA exploits the routing information, including train location, speed, direction, etc. to maintain the routing path and reduce the handover time, keeping a high level of successful handover without degrading user performance. **RMA**: to compute the optimum routing path on request between two provided nodes from MMA. **MTA**: to create and manage virtual networks of multiple virtual network operators (VNOs) in the vehicles, and also provide per-tenant information on QoS and resources utilization for each of them.
2. Media Distribution: CDN	Content distribution functions required for replicating the content FH/BH resource management functions in terms of routing Allows multiple CDN operators (tenants) for deployment of their network services	**CDNMA:** responsible for CDN infrastructure instantiation, control and management of the CDN service. **RMA**: to deal with the network resources to compute the optimal paths between the user and the CDN server assigned. **MTA**: to provide tenant identification and per-tenant monitoring information for each tenant. **MMA**: to provide the user network entry point to the CDNMA.

Use Cases	Main functions required	Required applications
2. Media Distribution: TV Broadcasting	Content distribution functions required for replicating the content FH/BH resource management functions in terms of routing Allows multiple TV service operators (tenants) for deployment of their TV services	**TVBA**: responsible for TV service requirements establishment, control and management of the video play-out. **RMA**: to deal with the network resources to compute the optimal paths for the broadcast tree. **MTA**: to provide tenant identification and per-tenant monitoring information for each tenant.
3. Dense urban society	FH/BH resource management functions Mobility management functions Fault management functions Energy and monitoring management functions Allows multiple virtual operators (tenants) for deployment of their virtual networks	**EMMA**: to monitor the power consumption of the system and provide information to be used for dynamic control of the network topology for energy saving. **RMA:** to deal with the network resources to optimize the optimal paths for FH/BH traffic and the RAN functional split, taking into account the newly deployed end points and property of dynamic crowd. The RMA shall also solve the problem of function and service placement over computing nodes. **MMA:** to handle the mobility of users in terms of optimizing handover, monitoring the user location and traffic offloading by placement of mobility anchors and breakout points. If the network entry point changes due to the dynamic crowd, the MMA will notify the MTA, RMA and EMMA for efficient network reaction. **MTA:** in charge of creation of virtual networks for one or multiple VNOs to share the FH/BH resources while meeting their individual SLAs, also providing per-tenant information on QoS and resources utilization for each.

Use Cases	Main functions required	Required applications
4. Multi-tenancy	Create tenants for deployment of virtual networks or network services FH/BH resources management functions Energy management and monitoring functions	**MTA:** in charge of creation of virtual networks for one or multiple VNOs or network services to share the FH/BH resources while meeting their individual SLAs, providing per-tenant information on QoS and resources utilization. **RMA**: to compute the optimum routing path on request of MTA to decide on the mapping between a virtual link and a physical path. **EMMA**: to provide to the MTA the monitoring services for each physical/virtual infrastructure or single physical/virtual elements, and provisioning of "energy-optimized" network paths or even "energy-optimized" virtual infrastructures.
5. Mobile edge computing	(Re)location of applications and virtual network functions on the distributed MEC servers due to mobility of the end users Energy management and monitoring functions FH/BH resource management functions	**RMA**: to deal with the network resources to compute the optimal paths to connect the VNFs in distributed MEC servers, as well as the location of the VNFs considering the computing resources. **MMA**: to compute the location and relocation of VNFs and services and the placement of MEC servers. **EMMA**: to monitor the power consumption of the MEC servers and provide information to be used for dynamic control of the VNFs for energy saving.

4 Interactions of 5G-Crosshaul applications

This chapter analyses the relationship between the different applications defined within WP4, and describes the workflows per application to show the interactions of each application to the XCI components and/or other WP4 applications.

4.1 Interaction between applications

The interactions among 5G-Crosshaul applications are exemplified in Figure 20, which clearly shows that the application layer is divided into two sublayers.

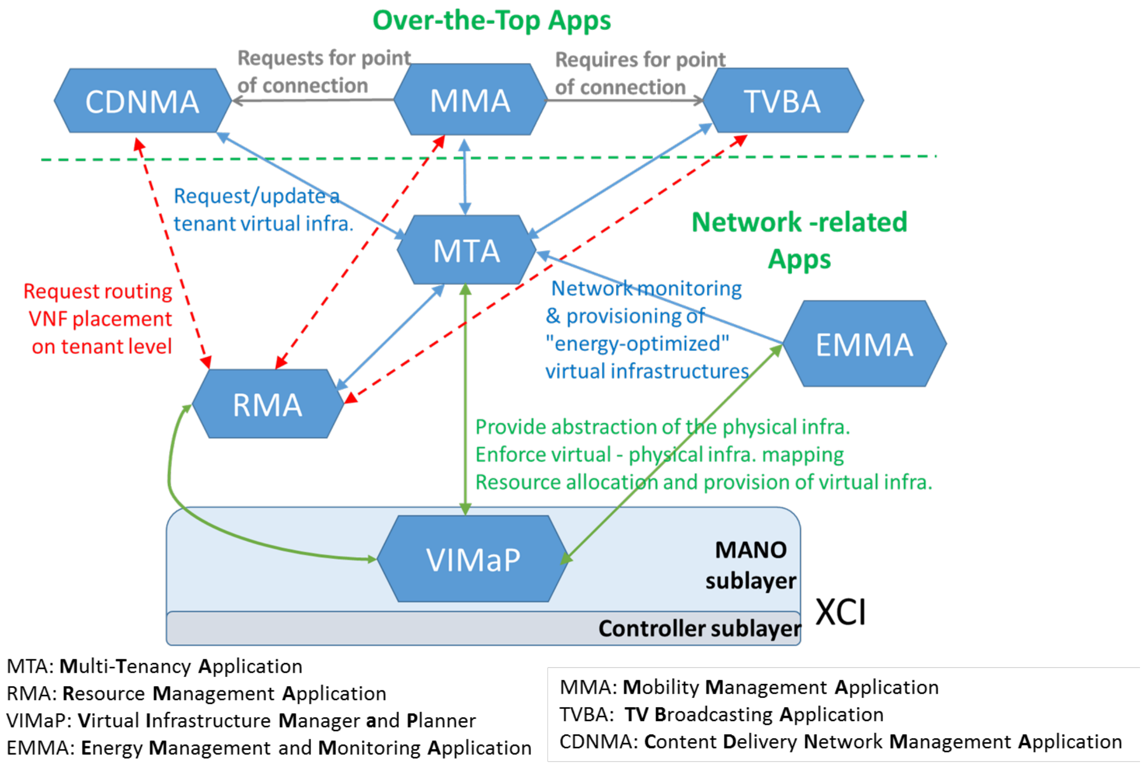


Figure 20: Interaction among 5G-Crosshaul applications

The higher sublayer includes OTT (Over-The-Top) applications, namely CDNMA, MMA and TVBA, which have a limited collaboration among them and interact with lower sublayer applications through their SBI. Interactions among OTT applications are limited to the MMA providing CDNMA and TVBA (if needed) information regarding the point of attachment, i.e. to give the information related to whether the user(s) moved from one point of connection (i.e. XFE) within the 5G-Crosshaul network to another. These interactions are shown in grey.

The lower sub-layer includes network-related applications: RMA, EMMA and MTA. The interactions among them are more complex than for the OTT apps, and they also interact with the XCI. In the figure, the XCI is functionally divided into two sublayers: a MANO sublayer, where the VIMaP application also resides, integrating the ETSI NFVO, VNFM and VIM, and a Controller sublayer including SDN controller and cloud controller. The VIMaP has the task of instantiating, deploying and provisioning resources (including networking, computing and storage resources) over the 5G-Crosshaul infrastructure, interacting with one or more underlying controllers. Thanks to its placement, the VIMaP can provide the view of the physical 5G-Crosshaul infrastructure, to other applications, enforce the physical mapping between the virtual infrastructure and the physical infrastructure and allocate the corresponding resources for all applications. In some cases, as with the EMMA, the interaction with the VIMAP (green line) is mediated by the NVFO, although such step is not shown in the figure for simplicity.

Most interactions involving OTT and Network-related apps go through either the RMA or the MTA. Both CDNMA and TVBA interact with the MTA for creating or updating a tenant, identifying the set of virtual nodes and virtual links and their constraints. In this case, the OTT operators, who request for a "tenant" to deploy their services on top of the 5G-Crosshaul infrastructure, can use the corresponding OTT apps (CDNMA, TVBA) to send their request on behalf of the OTT operators to the MTA for creating or updating a tenant in order to deploy their network services according to agreed Service Level Agreements (SLAs). In return, the OTT apps will receive a tenant ID from the MTA, which is used to identify a tenant inside the XCI. Indeed, the MTA can identify a heterogeneous set of virtual resources (i.e., a slice) and provide per-tenant monitoring information on resource utilization such as load, latencies and losses to each of the VNOs or tenants. The MTA is aided in this task by the RMA, which provides the mapping between virtual links and physical paths. A similar mapping interaction between MTA and RMA also occurs if the OTT applications require a change on the provision of one or more virtual links. In addition, the RMA can interact with the MTA and the OTT applications to receive the necessary updated information to perform efficient computation of paths within the 5G-Crosshaul network, as well as the placement of virtual network functions over computation resources (i.e. XPU).

The interaction between different applications is shown by solid blue lines in the figure. It should be pointed out, however, that the MTA does not instantiate virtual resources, but simply computes the optimized mapping of a virtual infrastructure onto the physical infrastructure, asking the XCI (and VIMaP) to do the actual instantiation (green line). It is also possible, for OTT applications, to directly interact with the RMA once the tenant has been created. In this case, the OTT application can request the RMA to compute routes across the virtual links created for that specific tenant. In this case the OTT applications only do routing on the tenant level, without requiring any changes on the established mapping between virtual links to physical paths. These interactions are shown in red dash lines in the figure.

Finally, the EMMA will expose services which may be consumed by the MTA. Such services are: (i) monitoring information for each physical infrastructure or single physical elements; (ii) monitoring information for each virtual infrastructure (not tenant!) or single virtual elements; (iii) provisioning of "energy-optimized" network paths and (iv) provisioning of "energy-optimized" virtual infrastructures (including network services with VNFs). In this interaction, shown by a blue line, the EMMA will only act as service provider.

4.2 Workflow of each application

4.2.1 Multi-Tenancy Application

The MTA offers two main services:

1) The deployment of Network Services (NS) with specific virtual network functions (VNFs) which are interconnected according to the defined VNF-FGs, as defined by the ETSI NFV architecture.
2) The deployment of Virtual Infrastructures (VI), which is composed of a coherent set of heterogeneous network, compute and storage infrastructure.

Figure 21 presents the workflow of deployment of network services for the OTT operators, with the following steps:

1. OTT operator sends a request to the MTA application for creating a tenant to deploy one or more their network services, either providing MTA the actual logical mapping (case a) or send SLA and ask MTA to compute the mapping (case b).
2. In case b) if MTA needs to decide the optimum placement of Virtual Machines (VMs) and the connections between them, then the MTA need to request the VIMaP application to provide an abstraction of the underlying infrastructure view.
3. The VIMaP returns the abstraction of underlying 5G-Crosshaul infrastructure.
4. The MTA provides the mapping of tenant to NS and VNF-Forwarding Graph (FG), as well as keeps track of all the mappings between tenants and real infrastructure by assigning a tenant ID (which is used to map a tenant to a NS or VNG-FG and used by the XCI to identify the tenants), and accept or reject tenancy requests depending on available resources. In case b), the MTA also computes the optimum placement of Virtual Machines (VMs) and the connections between them. After the tenant is created and the tenant ID is given, MTA then forwards the OTT operator requests to NFVO inside the XCI to initiate the network services.
5. The NFVO validates the requests and calls the VNF manger to instantiate the VNFs.
6. The VNF Manager validates the request and calls NFVO to perform the resource allocation.
7. The NFVO then requests the VIMaP to allocate the resources and instantiate VMs
8. The VIMaP validates the requests, whether there is an available resource to allocate.
9. If yes, then VIMaP allocates the computing and storage resources, otherwise reject.
10. For the network resources, the VIMaP calls SDN controller to configure network connectivity between VM endpoints.

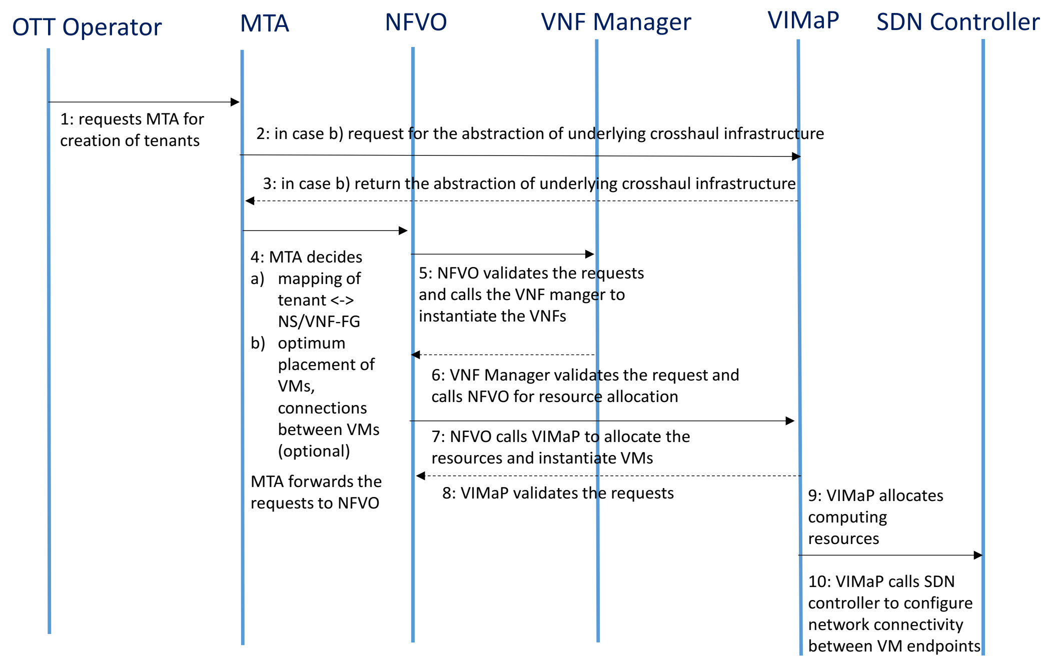


Figure 21: Workflow of deployment of a network service requested by OTT operator

Figure 22 presents the workflow of deployment of a virtual infrastructure for MVNO.

1. A tenant (VNO) requests the MTA to create a virtual infrastructure by allocating a network slice. A virtual infrastructure allocation request consists of:

2. Topology of the requested virtual infrastructure, e.g. a list of virtual nodes, links, computing nodes
3. Resource demand (bandwidth, computing and storage requirement)
4. The MTA processes the request from the VNO and it at first requests the VIMaP to provide an abstraction of underlying Crosshaul infrastructure.
5. The VIMaP returns the abstraction of underlying 5G-Crosshaul infrastructure.
6. The MTA computes the optimized resource allocation by solving typical Virtual Network Embedding (VNE) problem.
7. The MTA requests the RMA to calculate path for deciding the mapping a virtual link to a physical path
8. The RMA returns the physical path for the virtual link
9. The MTA decisions are then sent to the VIMaP, requesting for actual allocation of resources for the tenants and configuration of the mapping of virtual infrastructure to physical infrastructure. The VIMaP application is focused on allocation and direct instantiation of VMs to the tenants and configuration of their required computing and resources.
10. For the networking connection services, the VIMaP calls the SDN controller to allocate the networking resources. Finally, the SDN controller creates the virtual tenant network and allocates the requested networking resources.

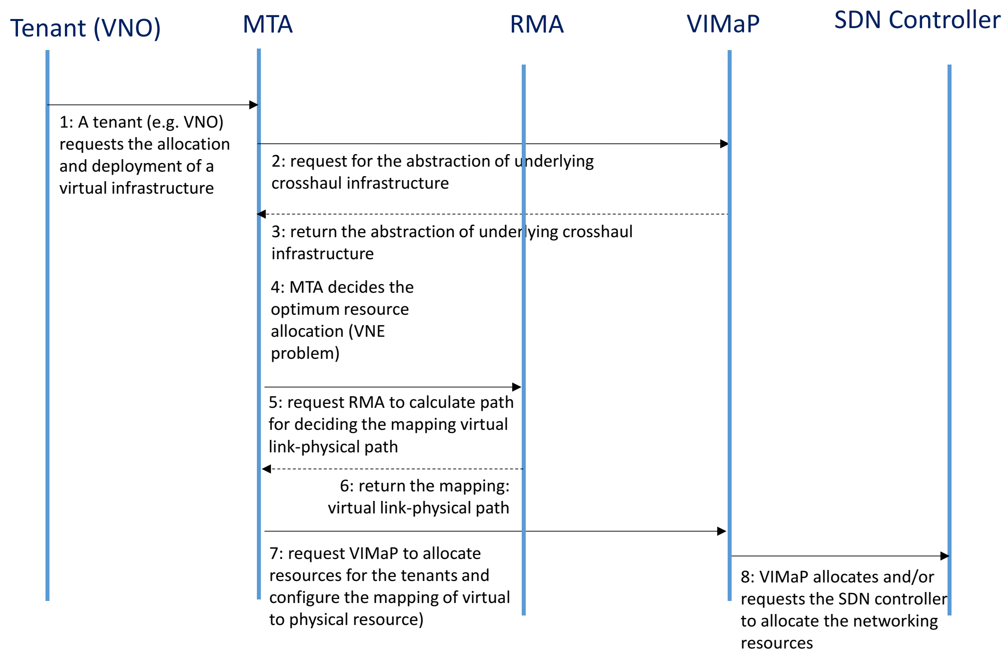


Figure 22: Workflow of deployment of a virtual infrastructure requested by VNO

4.2.2 Mobility Management Application

Figure 23 shows the entry process of a new user in the 5G-Crosshaul domain. It follows the procedure of the algorithm detailed in Section 3.2, firstly detecting the entry of the new user, getting the topology, deciding the best GW, notifying the CDNMA for the CDN Node placement and finally creating the path.

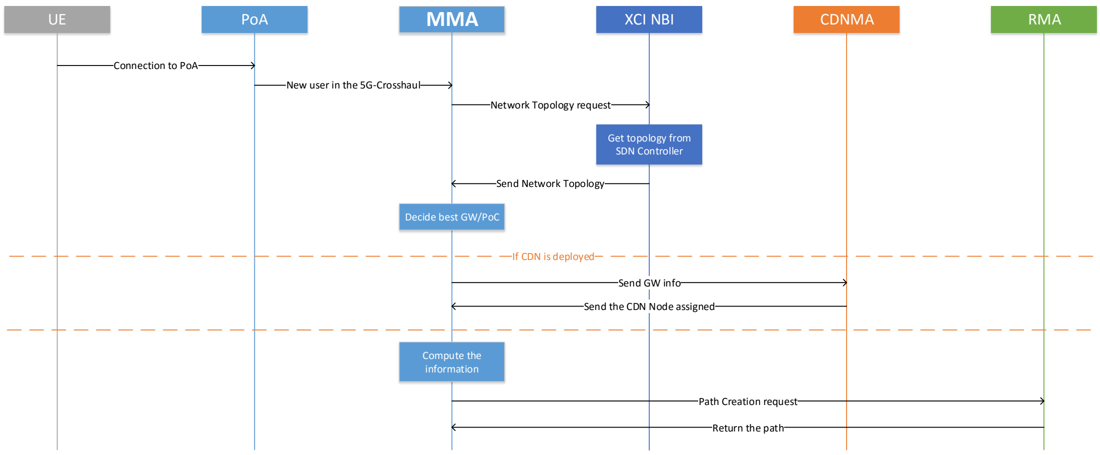


Figure 23: Workflow of MMA new user

The workflow depicted in Figure 24 details the handover process in the 5G-Crosshaul domain, notifying the CDNMA and changing (if it's necessary) the CDN Node and recomputing the info for the path creation.

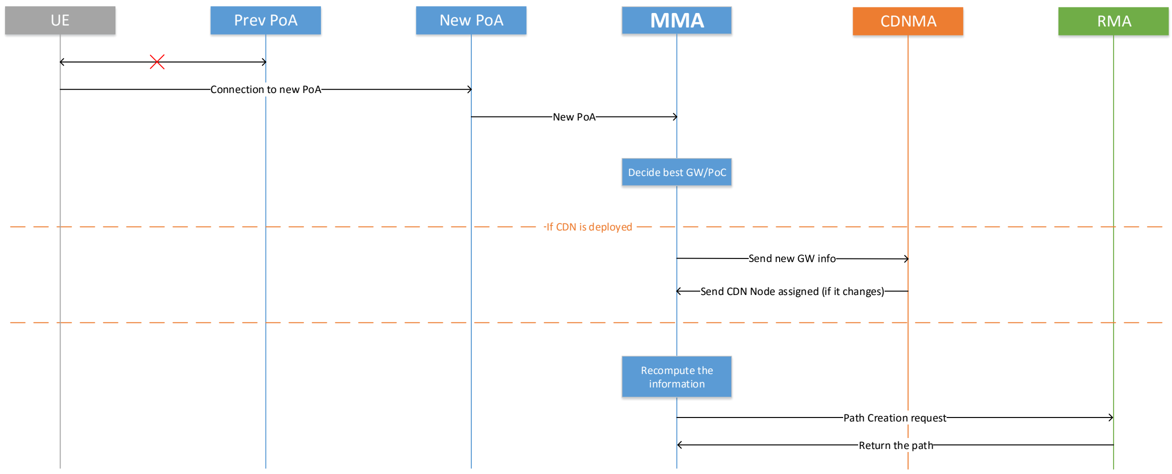


Figure 24: Workflow of MMA handover

The following figure shows the workflow of the high speed train use case, the MMA calculates the most appropriate proactive routing path for the handover process:

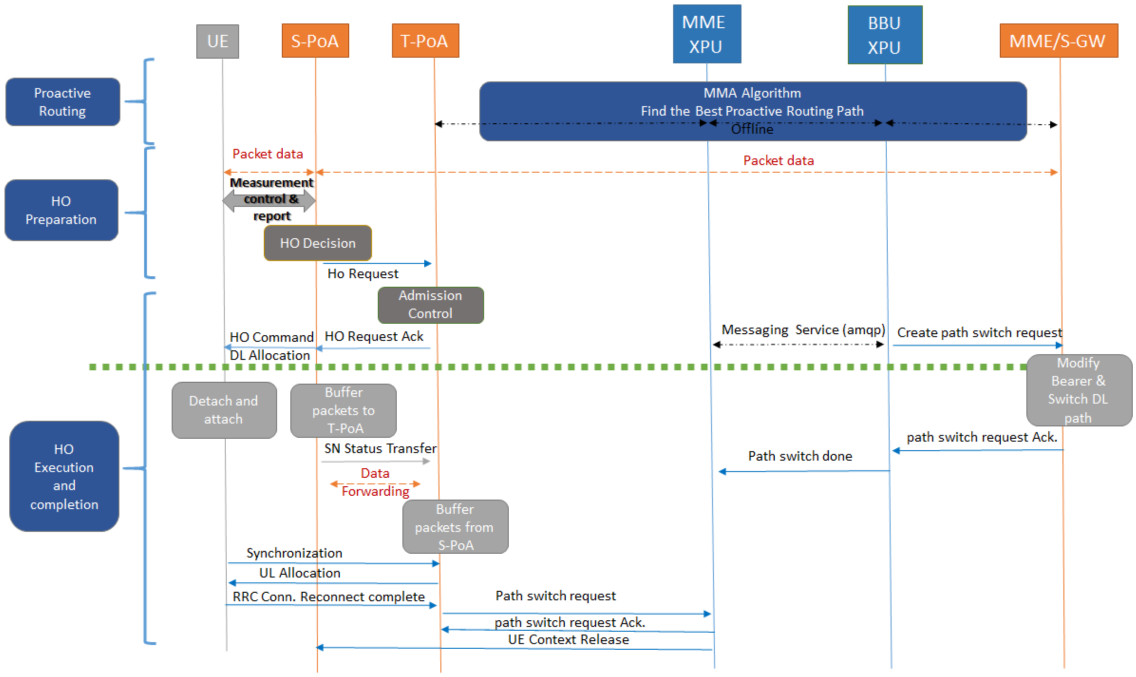


Figure 25: MMA workflow for high speed train use case

4.2.3 Resource Management Application

The RMA provides centralized and automated management of 5G-Crosshaul resources, to promptly provision transport services with an adequate quality while ensuring that resources are effectively utilized.

As described in Section 3.3, the RMA computes optimized solutions for dynamic resource allocation and network adaptation (e.g. to network failures), as well as for dynamic function placement. If required, the RMA scope can go beyond the network resources and also consider storage and computing resources.

The RMA can provide the outcome of the computation as a result of a query, in a server client-basis, as it happens for the Path Computation Element (PCE) architecture defined by IETF. Specifically, taking the terminology associated to the PCE as a reference, the RMA is currently perceived as "passive stateful". "stateful" since, for each active service, along with network state, the RMA will also store the state of all the computed paths or LSPs and their resources. Such information can be augmented with knowledge about the available and used computing and storage resources. This relevant information can be either located in a local database or requested to the XCI. "Passive" since the RMA should not instantiate and be the owner of the services installed in the 5G-Crosshaul. This latest aspect can be reconsidered at a later stage in the project, but for the moment the RMA is expected to provide an on-demand service that computes an optimal solution for the allocation of resources upon a request issued from the modules of the 5G-Crosshaul architecture or the applications developed in WP4.

On the service use-cases that have been defined in [1], Figure 26 shows the workflow for the case of the OTT service. In this case the OTT requests a service through the NBI by using a template specifying the requested resources to the NFVO. The NFVO will calculate the forwarding graph info and request the deployment of the virtual machines and the network resources to the VIM (the interactions described so far are not detailed since it is not in the scope of this section). The VIM can have an internal logic to compute the optimal allocation of network resources or can delegate this computation to the RMA. Indeed, the RMA can improve the 'default logic' of 5G-Crosshaul modules (in this case, the VIM), which can be used as a fallback when specific RMA algorithms are not available. As stated above, the computation performed by the RMA can be extended to storage and computing resources. If the RMA is queried, it performs the computation and provides a response, which is used by the VIM actually to instantiate the resources. Figure 26 does not show the following instantiation steps (out of the scope of this section).

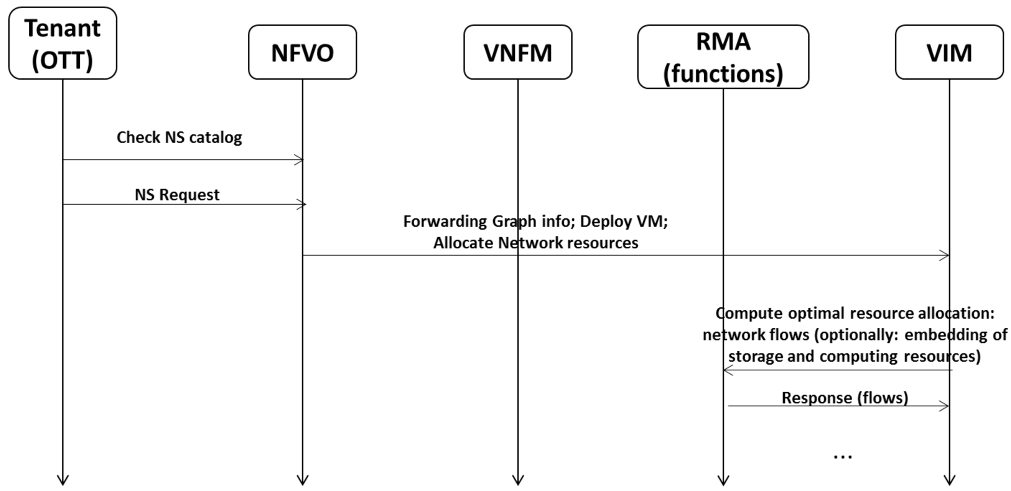


Figure 26: RMA workflow for the OTT service case

According to the service use-cases provided in [1], also the case with a Mobile Virtual Network Operator (MVNO) owning virtual resources only could be mapped in a similar logic to the one shown in Figure 26. In this case the MVNO will directly address the VIM, which in turn may request, as in the previous case, the computation of the optimal allocation of resources.

Finally, Figure 27 shows the workflow when considering the Network Operator service case reported in [1]. In the case in which the network operator needs to instantiate network services, it will issue a request to the SDN controller through its NBI. The SDN controller may have an internal logic for the path computation. Otherwise, it may request the computation of optimal paths to the RMA, which will provide a response including the relevant information for the allocation of the connections. The actual provisioning will be carried out by the SDN controller.

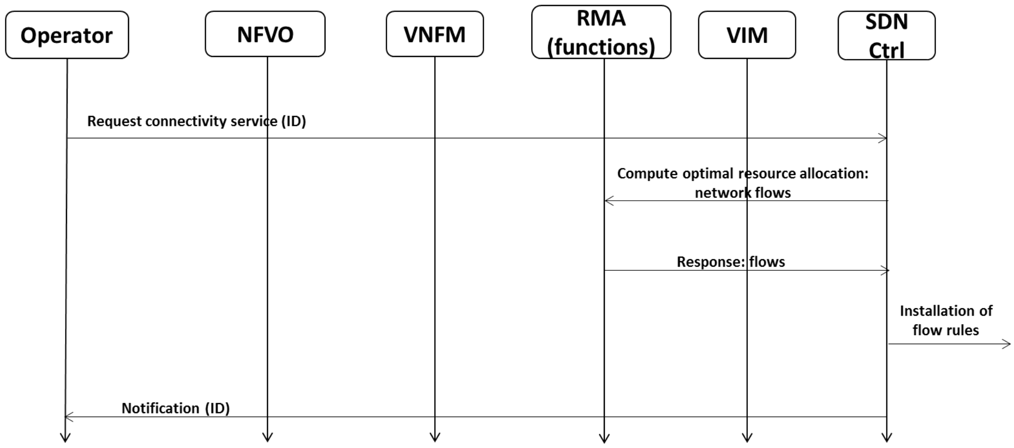


Figure 27: RMA workflow for the network operator service case

In order to perform the requested computation, the RMA will need to gather information about the status of network, computing and storage resources.

4.2.4 Energy Management and Monitoring Application

The EMMA is a network application, whose main goal is to monitor the system power consumption and the energy status of physical nodes, both network and computing, to minimize the energy expenditure while ensuring an acceptable quality of service.

As far as energy management is concerned, EMMA operates as follows:

- It identifies which physical network nodes are idle (i.e., nodes whose communication cards are all inactive), hence, can be switched off (BASIC FUNCTION).
- It computes alternate paths for traffic flows on the physical network topology, or it selects a path among the available ones for a new traffic flow, triggering path (re-)configuration with the objective of minimizing the energy consumption (NETWORK-RELATED FUNCTION)
- It computes the optimal placement of the requested virtual infrastructures over the available physical infrastructure, triggering their re-planning to maintain the energy consumption under configurable thresholds (COMPUTING-RELATED FUNCTION).

Note that the above energy-management functions can be conceived as being performed periodically, upon collection of updated information, or "on demand" (e.g., when an energy-efficient path for a new traffic flow must be found). Furthermore, the decisions made by EMMA will be based on the MNO energy policy (if specified), e.g., to set power consumption thresholds or preferred resources.

EMMA needs to interact with several entities, namely, the SDN controller, the NFVO and the VIMaP, either to collect information or to provide commands. Figure 15 depicts the main interactions and their content. The following interactions have been defined by assuming that the SDN controller includes: an entity for statistics collection (Net Stats), an entity for configuring the node operational state (Node Config), an entity that computes the physical network topology (including traffic path information), and an entity that provides paths on the physical network upon request. The latter exploits the information provided by the VIMaP.

Basic function:

1. EMMA collects energy statistics from the XCI (and sends them to the MNO).
2. EMMA may provide “nodes switch off” (or “node sleep”) commands to the SDN controller, to switch off / put to sleep physical network nodes.

Network-related function:

1. EMMA also collects information on physical topology, traffic paths and load over the links through the XCI.
2. EMMA may provide energy efficient paths for traffic flows as well as “nodes switch off” (or “node sleep”) commands to the SDN controller.

Computing-related function:

1. EMMA receives from the NVFO information on where VNFs have been placed.
2. EMMA may provide a new mapping of NVF on physical nodes to VIMaP, as well a “server sleep” command so as to put a server that has become idle to sleep.
3. As a consequence, VIMaP sends to the SDN controller a new network path to be used.
4. If any physical node has become idle, EMMA sends a “node switch off” (or “node sleep”) command to the SDN controller.

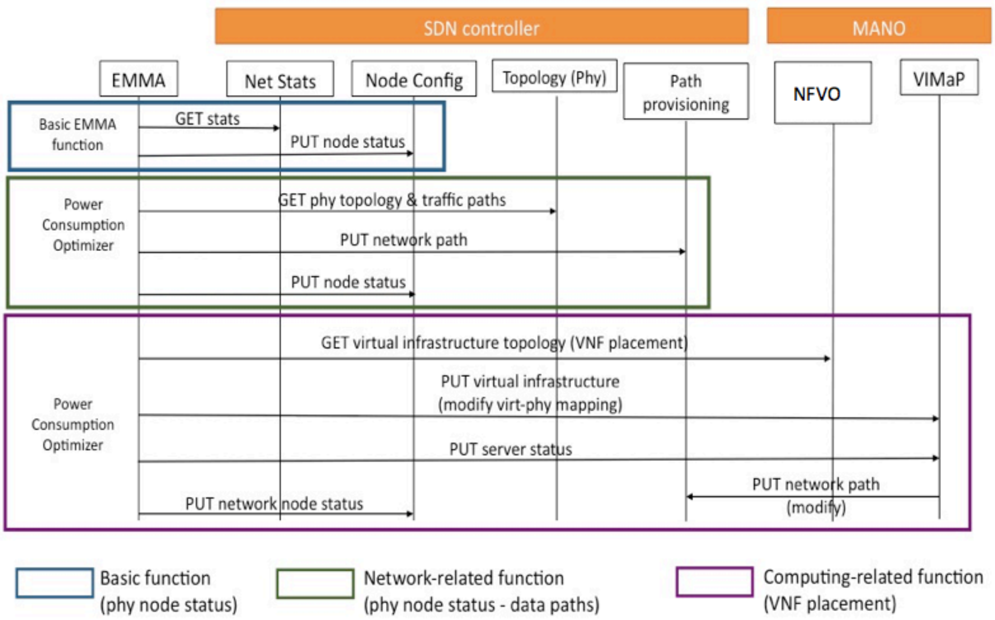


Figure 28: Workflow of EMMA

4.2.5 Content Delivery Network Management Application

A Content Delivery Network (CDN) is an Over-The-Top (OTT) service related to the distribution of media contents, especially video traffic, on a network infrastructure where a same set of resources supports multiple applications and services from different service providers. The deployment of the service is done through the XCI NBI. A CDN operator, "tenant", can deploy the CDN service over a XCI controlled physical or virtual infrastructure. For this, it uses the services and API offered by the XCI NBI, NFV MANO, and in particular the NFV-O (Orchestrator).

Based on the multitenancy definition from an operator point of view, a CDN infrastructure service considers essentially the "push mode" of resource awareness, meaning that, the CDN operator needs to know previously the network topology and location of compute nodes as provided by the infrastructure owner to decide where to deploy the CDN nodes.

The CDN controller, CDN Management application (CDNMA), manages the CDN infrastructure configuration and decides the content delivery rules. The cache nodes are deployed on virtualised server instances connected through one or more VNF-FGs in the 5G-Crosshaul network.

The CDNMA application will interact with other applications and the XCI NBI to provide the service required. Two workflows have to be taken into account. On one hand the workflow related to the CDN infrastructure instantiation, and on the other hand, the workflow regarding the control and management of the CDN service during the lifetime.

The following figures show the workflows for the simplest case, without regard multi-tenancy concept.

Figure 29 presents the workflow for the CDN infrastructure instantiation.

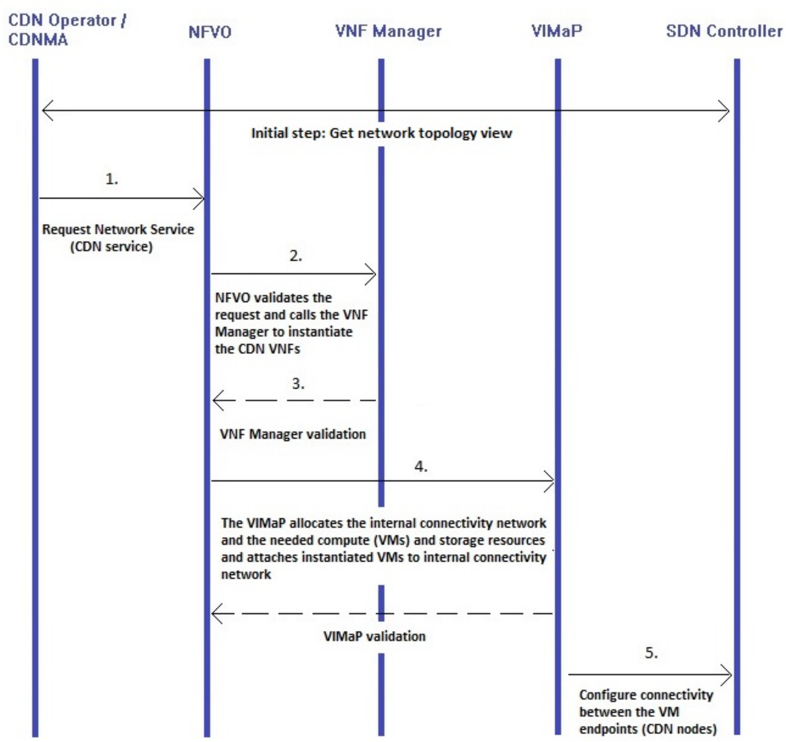


Figure 29: CDN instantiation workflow

1. CDN operator / CDN Management Application

- The CDN operator requests the CDN configuration (network service) to NVO, and provides the VNF templates (origin and replica server nodes), and the VNF-Forwarding Graphs (VNF-FGs) that will be needed to implement the service.
- The CDN nodes placement is decided by the CDN operator based on the network topology view from the SDN controller and location of compute nodes.

2. VNF Orchestrator (NFVO)

- The NFVO receives a request from the CDN operator to instantiate the CDN service. The NVFO validates the request and starts the end to end configuration process through the Service Function Chain (SFC) mechanism, according to the provided VFN-FG.
- The VNF-FG described by its descriptor file, is orchestrated by the NFVO.
- The NFVO validates the request and calls the VNF Manager to instantiate the CDN VNFs, with the instantiation data.
- Once the NFVO receives the VNF Manager validation, it requests the allocation of resources (compute, storage and network) needed for implementing the CDN infrastructure to the VIMaP.

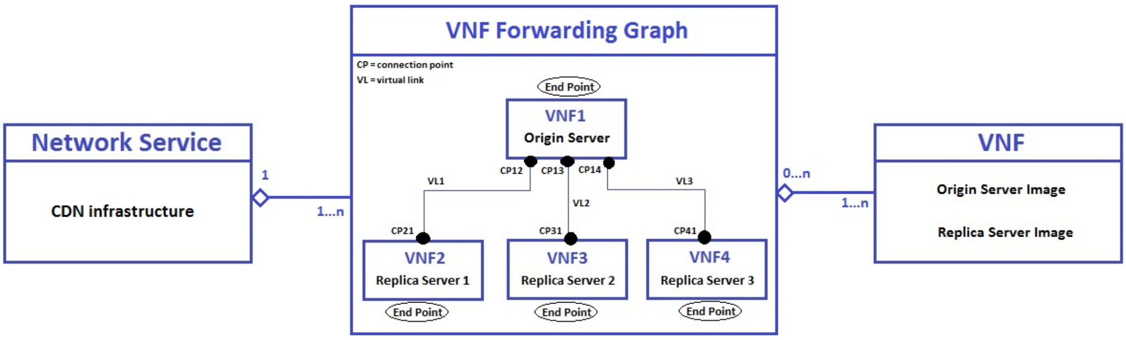


Figure 30: NFV entities to deploy and manage for the CDN service

3. VNF Manager

- The VNF Manager validates the request from the NFVO and processes it and calls the NFVO for resource allocation.
- The VNFs described by their descriptor file, are instantiated by the VNF Manager.
- The VNF Manager controls the lifecycle management of VNF instances.

4. VIMaP

- The VIMaP allocates the internal connectivity network and the needed compute (VMs) and storage resources and attaches instantiated VMs to internal connectivity network. The VIMaP follows the rules provided by the CDNMA to decide where VMs have to be instantiated.
- The VIMaP asks the SDN controller to provide and compute the paths between the CDN nodes according to the CDN graph (VNF-FG).

- The VIMaP utilizes southbound interfaces towards the NFVI to allocate resources. The allocation includes starting up services such as a VM.

5. SDN Controller

- Configures virtual connectivity between the VM endpoints (CDN nodes) at the implementation level.

Figure 31 presents the workflow for the CDN service management.

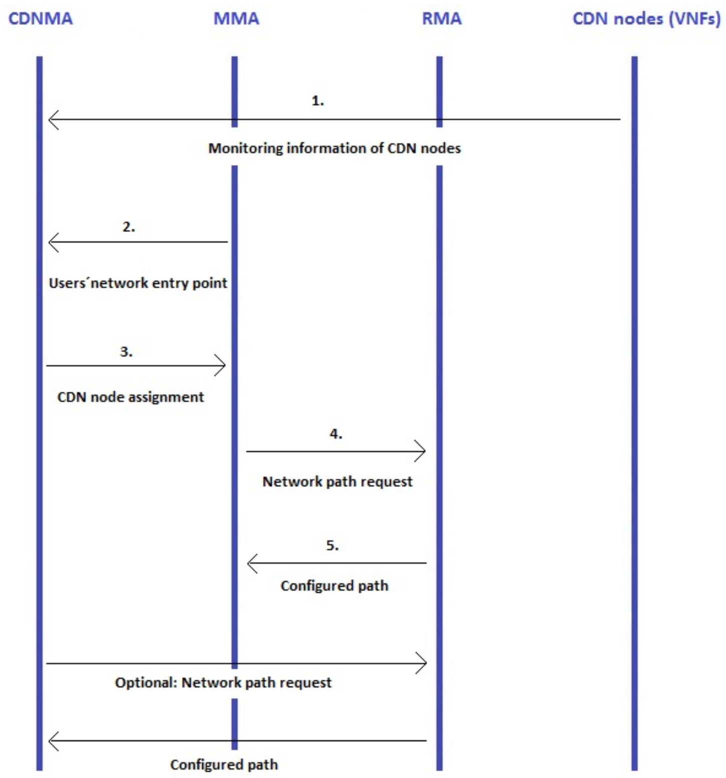


Figure 31: CDN service management workflow

1. CDN nodes (VNFs)

- The CDNMA application will receive specific monitoring information from the different CDN nodes (VNFs) deployed in the network. The specific monitoring information for each VNF will be included in the VNF descriptor.

2. MMA

- The MMA application will send the user network entry point to the CDNMA application and this one will decide the best CDN node to service the user request.

3. CDNMA

- The CDNMA application will decide the CDN node assignment for the users´ requests based on the information received from the MMA application, VNF monitoring and the policies defined by the CDN operator.

- It will send the assigned CDN node to the MMA application.
- Optionally, the CDNMA application could request the instantiation of network paths to the RMA without involving the MMA application.

4. RMA

- It will be involved in the network path instantiation based on the end points information received from the MMA application or the CDNMA application.

4.2.6 TV Broadcasting Application

The TVB application will be in charge of the service parametrization and the interaction with other applications and the XCI NBI to provide the service required.

To set the TV service parameters, which make it possible to adjust the system to the specific needs of the content provider and customer expectations over the 5G-Crosshaul network, TVBA will follow the workflow detailed below:

First, TVBA negotiates the parameters of the service. This means that the key information to deploy the service is sent to the XCI to establish the working conditions. TVBA provides the bandwidth needed (in the range of Mbps with meaningful differences between SD, HD or beyond HD resolutions), the initial coverage foreseen for the service (TV broadcasting is a point to multipoint service), and the “priority service token”, as TV Broadcast, as a premium service, should request some hierarchy over other coexisting services in the network.

The set of parameters of the service to analyze the viability of the service over the network are processed by the RMA. RMA will get the topology of the network, will review the switches and links to build the broadcast tree and will return the graph to the app.

The graph is managed by TBA through the SDN controller to establish the paths over the network.

New user engagement and user give up that modify the end points involved in the broadcast will be obtained by API provided from MMA and will drive to new network path instantiation based on the new information regarding the coverage.

Additionally, network monitoring functions and quality of service assessment (SDN compliant over the virtual network switches) will be conducted to get valuable info on the service deployment and to be ready to take a hard decision on the routing by building and reconfiguring the optimum broadcast tree for each moment.

In case the original service conditions can be held by the network and the broadcast tree can´t be built, the TVBA will rearrange the parameters of the service to be accomplished.

Finally, the TVBA is in charge of notifying the end of the TV service deployment. Thus, the application sends back the information regarding the tree to the SDN controller in order to free up the resources involved and closes the paths and turn down the priority assigned to the broadcast streams.

Figure 32 provides a summary of the TVBA workflow.

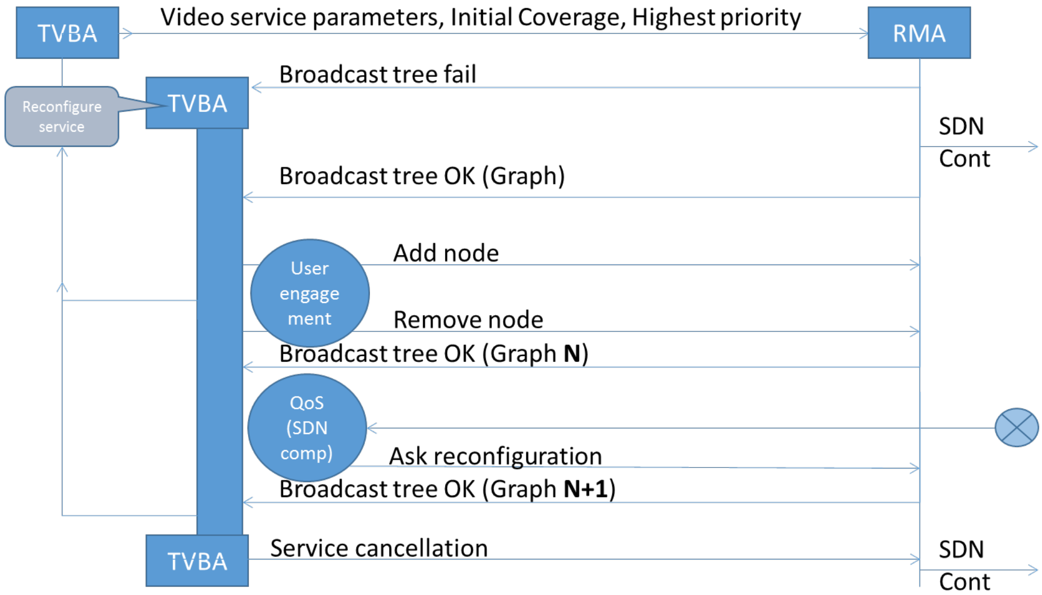


Figure 32: TVBA service management workflow

4.3 Application Layering

This section is devoted to the analysis on how the XCI behaves when there are multiple tenants providing services on top of a single Crosshaul infrastructure operator, with special focus on the functionality and placement of WP4 applications within the XCI framework. This section provides an explanation on the current view of this issue and may be subject to changes as the discussion evolves. Current understanding is based on the hierarchical recursion paradigm, where each new tenant is able to work on its private logical virtual view of the infrastructure, handling it with a complete new instance of the XCI, as if the tenant was working on top of a real physical infrastructure.

Figure 33 shows a schema of the current work hypothesis. At the bottom of the figure we can find the Crosshaul virtual infrastructure provider. This actor owns the Crosshaul infrastructure and works on top of real physical network entities (e.g. XFEs and legacy network nodes) and XPUs. In order to control this infrastructure, the virtual infrastructure provider instantiates its own XCI made of a NFV infrastructure that orchestrates SDN, storage and computing resources. As explained in [9] [10], the NFV infrastructure of the XCI is compliant to the standard ETSI NFV architecture.

Figure 33 focuses on the different applications provided by WP4 and where they should be logically placed so that multi-tenancy operation is possible.

The application functionality is logically located based on two assumptions:

Every tenant instantiates a complete XCI that works over a virtual infrastructure as if a real one was deployed.

Virtual Network Functions are local to each tenant and are not provided as a service. This implies that each tenant is in charge of allocating and managing its own NFVs.

This further implies a logical placement of the different elements (applications) orchestrating the multi-tenant scenario.

The recursion of the architecture is made possible by the coordinated work of three applications developed within WP4: Multi-tenancy (MTA), Virtual Infrastructure Manager Application and

Planner (VIMaP) and Resource Manager Application (RMA). MTA responsibilities lie on the orchestration of the different available resources according to the requests by each tenant. Based on that, the MTA will talk with the VIMaP/RMA to generate a virtual topology composed of networking, computation and storage resources complying with the tenant requirements. This virtual topology is exposed to each tenant based on an Intermediate Control Plane Interface (I-CPI). The I-CPI is expected to provide abstractions models so that each tenant is completely unaware of the fact of working on a virtual or a physical infrastructure, and moreover independently on vendor and protocol specific mechanisms.

This behaviour is recursively applied layer by layer. Figure 33 shows an example of a single Virtual Infrastructure Provider (VIP) and a recursive set of 3 tenants. The Virtual Infrastructure Provider is in charge of administering its physical network through the SBI and provides services to the other 3 tenants. It is important to understand that the VIP only sees two tenants (Tenant #1 and #2). Tenant #3 is hidden from the VIP through tenant #1, which will allocate part of its own resources to tenant #3.

The VIP will partition or slice its own resources to provide a virtual infrastructure consisting of network, computation and storage resources to tenant #1 and #2. This allocation will be based on each tenant request for a virtual network topology and resources, based on SLA agreement. Each tenant is free to use these resources for its own operation (tenant #2) or resell parts of it to another virtual operator (tenant #1). Each tenant will control its virtual network through the I-CPI.

In the case of tenant #2 and #3, the MTA application is drawn in a lighter color and in dashed lines. This represents the fact that this application is not required by these XCIs since they do not provide any service to another tenant.

In addition to the SBI and I-CPI,

Figure 33 shows also another interface: the Application Control Plane Interface (A-CPI) or Northbound. This interface is used to provide third party applications with mechanisms to control the functionality of the XCI. It is also expected, on the current working view of the XCI, that this interface is used by NFVs such as the Mobility Management Application (MMA), Content Delivery Network Management Application (CDNMA) and TV Broadcasting Application (TVBA). The reason for having these applications outside the XCI is because they are using the XCI and are not providing services to any other tenant. In addition, these applications are pertinent only to a specific tenant.

Example

Let us consider the MMA. This application requires as inputs the position (5GPoA to which the user is attached to) and the topology of the network, including the different processing points and points of connection with other operators. This information is local to the tenant and cannot be provided as a service by the underlying infrastructure provider.

Finally, it is worth noting that each virtual infrastructure is managed by the underlying provider, delegating control functions to each tenant according to the level of SLA signed between both operators. Considering this, the provider will take care of managing the real infrastructure and making sure that the requirements contracted by each tenant are met. Information related to failures and other events which occur within the crosshaul have to be monitored and such information will be forwarded to the tenant(s) depending also on the liaison between them, although the general assumption is that reconfiguration will occur at the provider level and only events that cannot be solved at the provider level will be scaled to each tenant. Monitoring services were discussed already in Section 4 but more in general, they are currently under evaluation and a more in-depth discussion on this will be provided in subsequent refinements of this IR. For example, in case of a physical link failure, the virtual infrastructure provider will reroute the flows going through this link so that the virtual topology of each tenant is not modified. In an event of general failure, leading to the split of the network or the degradation of the quality of service to a point where the contractual agreements cannot be guaranteed, then this issue will be scaled to the tenant layer.

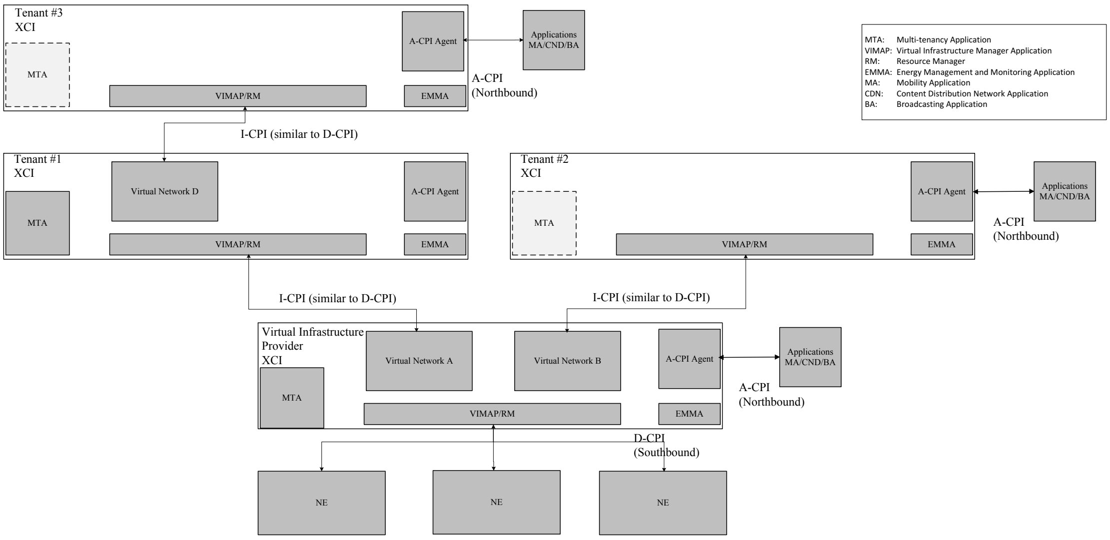


Figure 33: Recursion of XCI and layering of applications

5 XCI Northbound Interface

This section initially provides an overview of the interactions between the different 5G-Crosshaul applications and the Crosshaul Control Infrastructure (XCI). According to these, it provides a preliminary description of the XCI services that can be exploited by the application and the related requirements on northbound interface of the XCI.

5.1 Overview of the interactions between 5G-Crosshaul applications and XCI

The 5G-Crosshaul system architecture exploits SDN and NFV concepts decoupling between control and data plane and deploying Crosshaul network services as virtualized network services composed of Virtual Network Functions (VNFs) combined and interconnected together in dynamic chains instantiated on demand over the Crosshaul infrastructure. Moreover, as will be discussed in the next section, the applicability of the hierarchical and recursive models, introduced in the SDN architecture proposed by ONF [11], allows to deploy these virtual network services not only directly over the physical Crosshaul infrastructure, by also over the virtual infrastructure instances deployed for different tenants and sharing the same physical substrate.

In this context, the XCI provides mechanisms to configure the Crosshaul infrastructure, on both network and IT side (i.e. for both XPUs and XFEs), through unified APIs. These APIs must provide an abstract, protocol-independent and, where needed, technology-agnostic view of the underlying resources, at the associated virtualization level and with a view restricted to the virtual resources assigned to a given tenant. On the other hand, the XCI provides also a common framework to enable the deployment, management and orchestration of Crosshaul virtual network services composed of different VNFs, through APIs which are independent on the specific kind of Crosshaul VNF.

On top of the XCI, the applications implement the service-dependent logic and the network intelligence required to optimize the utilization of the available Crosshaul resources while guaranteeing the consistent coordination and orchestration of the Crosshaul services as well as their compliance with their QoS specification. Therefore, the XCI should be able, internally, to enforce the decisions taken from the applications and, externally, to expose a suitable level of programmability and granularity at its northbound APIs to allow the applications to specify their decisions and request the desired type of resource allocation and instantiation. On the other hand, the same XCI northbound interface should expose enough information about the capabilities, status, resource availability and performance of the underlying infrastructure as required by the applications to run their internal algorithms and take efficient decisions. Depending on the type of monitoring information, different interaction mechanisms should be supported. For example, polling of data related to resource availability is desirable for applications that needs to compute paths or resource allocations for on-demand instantiation of network services. On the other hand, applications that need to react to unexpected events (e.g. failures) or unpredictable conditions (e.g. excessive load on a network segment) must rely on subscribe/notify mechanisms for asynchronous communications.

SDOs like IETF and ETSI are working towards the standardization of APIs and information models which can be considered as valid starting points for the definition of the Crosshaul XCI north-bound interface. In particular, REST APIs and RESTCONF as reference protocol for the SDN controller north-bound interface are widely adopted in existing SDN controller frameworks. Several Working Groups (WGs) are defining YANG models for specific services. For example, YANG models for network topologies with different network technologies are under definition in the I2RS WG (Interface to the Routing System WG), while a topology with

Traffic Engineering (TE) parameters is available in an Internet Draft of the TEAS WG (Traffic Engineering Architecture and Signaling WG). Other YANG models are proposed to give an intent based representation of the network, useful to specify the desired virtual network infrastructures in multi-tenancy applications. Examples are the YANG model proposed by the Virtual Tenant Network application in OpenDaylight or the NEMO (NEtworking MOdelling) language proposed in the IRTF SDN Research Group [12].

In the NFV area, information models for VNFs and virtual network services descriptors (including their NFV graphs) are specified for both TOSCA and YANG/XML representation in the ETSI specification for the NFV Management and Orchestration [13]. In the next steps, these different models will be evaluated and proper ones will be selected for NBI as well as for interfaces between applications.

5.2 XCI services and Requirements on NBI

The following table provides a list of services which need to be provided by the Crosshaul Control Infrastructure, to be consumed by Crosshaul applications. The mapping of these services on specific functional components at the control layer will be addressed in WP3 [2]. For each service, the table provides a brief description and the list of consumers on the application side. The following tables (from Table 5 to Table 15) detail the type of interaction and exchanged information, as expected at the north-bound interface of each XCI service.

Table 4: List of XCI services.

XCI service	Service Description	Service consumers (apps)
Network topology service	The service provides information about the Crosshaul physical network topology, including nodes, ports and links, their characteristics, capabilities and resource availabilities.	Multi-Tenancy Application. (in case the Slice Orchestrator acts as decision entity). Resource Manager Application. Virtual Infrastructure Manager and Planner (VIMaP). Energy Management and Monitoring Application. CDN and TV Broadcasting Application.
IT infrastructure inventory service	The service provides a catalogue of the computing and storage resources available at the physical infrastructure, including information about resource capability and availability.	Multi-Tenancy Application. (in case the Slice Orchestrator acts as decision point). VIMaP. Energy Management and Monitoring Application. CDN and TV Broadcasting Application.

Network monitoring service	The service provides monitoring information collected by the physical network elements (the type of information exposed at the NBI depends on the monitoring capabilities of the network nodes).	Multi-Tenancy Application. Mobility Management Application. Resource Manager Application. Energy Management and Monitoring Application. CDN and TV Broadcasting Application.
Monitoring analytics service	The service provides information elaborated from the raw network monitoring data (e.g. estimation of SNR or SINR, interference level).	Resource Manager Application. Energy Management and Monitoring Application.
IT infrastructure monitoring service	The service provides monitoring information related to the physical XPUs.	Multi-Tenancy Application.
Provisioning of virtual infrastructures composed of VMs and virtual networks (IaaS model)	The service allows to perform CRUD operations for virtual infrastructures composed of VMs and virtual networks. The service integrates planning algorithms to map virtual resources to available physical resources and implements the “multi-tenant” concept. This service is supported at two levels: at the VIM for the request of virtual infrastructures with VMs and network resources, while at the SDN controller for the request is limited to the network scope only.	Multi-Tenancy Application.
Provisioning of network paths	The service allows to perform CRUD operations for network path configuration, i.e. to create, destroy, modify and retrieve network connections. The specification of the path hops as input parameter is optional. Proactive scheduling of network paths and support for primary and backup path	Mobility Management Application. Resource Manager Application. VIMaP. CDN and TV Broadcasting Application.

	are also required.	
Enforcement of traffic scheduling, shaping and QoS provisioning	The service allows to regulate traffic scheduling and shaping on network entities (potentially virtual network entities like OVS instances on servers). This feature needs to be supported by the underlying devices.	Multi-Tenancy Application. Resource Manager Application.
Creation and management of VNFs.	The service allows to instantiate, remove, modify, configure and, in general, operate specific VNFs.	Multi-Tenancy Application. Mobility Management Application. Energy Management and Monitoring Application. CDN and TV Broadcasting Application.
Creation and management of virtual network services.	The service allows to perform CRUD operations on virtual network services composed of multiple VNFs organized in Service Function Chains (SFCs).	Multi-Tenancy Application. Resource Manager Application (for dynamic function placement). Energy Management and Monitoring Application. CDN and TV Broadcasting Application.
Configuration of network devices.	The service enables a low level access to network device configuration for a limited set of commands (e.g. to regulate the power state/level of the devices).	Resource Manager Application Energy Management and Monitoring Application.

Table 5: Network Topology Service.

Network Topology Service			
Method	**Interaction**	**Input**	**Output**
Get <node(s), port(s), link(s), host(s)>	Polling	ID or filtering information	List of IDs and parameters of the requested entities
Notification <new entity, updated entity, removed entity>	Subscribe/notify	ID and/or parameters of the new, updated or deleted resource	--

Table 6: IT Infrastructure Inventory Service.

IT Infrastructure Inventory Service			
Method	**Interaction**	**Input**	**Output**
Get <host(s)>	Polling	ID or filtering information	List of capabilities of the given host (e.g. cpu, disk, memory)

Table 7: Network Monitoring Service.

Network Monitoring Service			
Method	**Interaction**	**Input**	**Output**
Get <nodeStatistics, portStatistics, flowStatistics, queueStatistics>	Polling	ID or filtering information	List of monitoring parameters for the given network entity (e.g. OF counters, delay, power consumption info, etc.)
Notification <target entity, threshold>	Subscribe/notify	Event to be notified and associated parameters that have generated the alert.	--

Table 8: Monitoring Analytics Service.

Monitoring Analytics Service			
Method	**Interaction**	**Input**	**Output**
Get <entityID>	Polling	ID or filtering information	Computed values.
Notification < target entity, threshold >	Subscribe/notify	ID and measure of the computed value (e.g. SNR value) that has generated the alert.	--

Table 9: IT Infrastructure Monitoring Service.

IT Infrastructure Monitoring Service			
Method	**Interaction**	**Input**	**Output**

Get <hypervisorDetail>	Polling	ID or filtering information	Monitored parameters (e.g. current workload, free disk, free ram, used memory, running VMs, used vcpus, etc)
Notification <target entity, threshold>	Subscribe/notify	ID and value of the parameter that has generated the alert.	--

Table 10: Provisioning of Virtual Infrastructures.

Provisioning of Virtual Infrastructures Service			
Method	**Interaction**	**Input**	**Output**
Get <stackID>	Polling	ID of an existing virtual infrastructure	List of the virtual resources allocated for the given virtual infrastructure (e.g. VMs, volumes, virtual networks, virtual topology, etc.)
Post/Put <stackDescription>	Configuration Command	Description of the desired virtual infrastructure in terms of virtual resources capabilities and constraints.	ID of the virtual infrastructure
Remove<stackID>	Configuration Command	ID of an existing virtual infrastructure.	--

Table 11: Provisioning of Network Paths.

Provisioning of Network Paths Service			
Method	**Interaction**	**Input**	**Output**
Get <connectionID>	Polling	ID of an existing network connection	Parameters of the given connection (e.g. status, end-points, bandwidth, network path, etc.)
Post/Put <connection-	Configuration Command	Description of the desired network connections (i.e. end-	ID of the virtual infrastructure

Description>		points, QoS parameters, [path], [scheduling], [recoveryOptions], …).	
Remove <connectionID >	Configuration Command	ID of an existing network connection.	--

Table 12: Enforcement of traffic scheduling, shaping and QoS provisioning.

Enforcement of traffic scheduling, shaping and QoS provisioning			
Method	**Interaction**	**Input**	**Output**
Get <queues, meters>	Polling	ID or filtering information	List of IDs and parameters of the configured queues/meters
Post/Put <queueDescription, meterDescription>	Configuration Command	Specification of the queues and meters to be configured (i.e. device and egress port ID, rate-limit and classifiers). It depends on the capability of the device.	ID of the queue/meter
Remove <queueID, meterID>	Configuration Command	ID of an existing queue or meter.	--

Table 13: Creation and management of VNFs.

Creation and Management of VNFs			
Method	**Interaction**	**Input**	**Output**
Query <VNF-ID>	Polling	ID or filtering information	Description and status of the instantiated VNF.
Query <VNF-packages>	Polling	ID or filtering information	VNF descriptor, sw images, manifest file, etc.
Instantiate <VNF descriptor>	Configuration Command	VNF descriptor (components, virtual deployment units, virtual links,	VNF-ID

		connection points, lifecycle events, auto-scale policies …)	
Scale / Modify <VNF-ID, VNF descriptor>	Configuration Command	ID of the VNF to be scaled/modified and desired specification	--
Terminate<VNF-ID>	Configuration Command	ID of the VNF to be terminated	--

Table 14: Creation and management of virtual network services.

Creation and Management of virtual network services			
Method	**Interaction**	**Input**	**Output**
Query <NS-ID>	Polling	ID or filtering information	Description and status of the instantiated virtual network service.
Query <NS-descriptor>	Polling	ID or filtering information	Available Virtual Network Service descriptors
Instantiate < NS-descriptor >	Configuration Command	Virtual Network Service descriptor (VNF descriptors, VNF Forwarding Graphs, Virtual Links, lifecycle events, …)	NS-ID
Scale / Update <NS-ID, NS descriptor>	Configuration Command	ID of the Virtual Network Service to be scaled/modified and desired specification	--
Terminate<NS-ID>	Configuration Command	ID of the Virtual Network Service to be terminated	--

Table 15: Configuration of network devices.

Configuration of network devices			
Method	**Interaction**	**Input**	**Output**
Post <node,	Configuration	It depends on the type	Status of the

command>	Command	of device and the type of command (e.g. setPowerState, addFlowEntry, …)	command.

6 Validation and evaluation

This chapter describes the validation and evaluation methodologies used in the 5G-Crosshaul applications. It details the different scenarios for the multiple applications, the project KPIs and the mapping between the test cases defined and the project objectives.

6.1 Methodology

This section combines together the information derived from the different developments planned within WP4, which belong to different test cases and partners.

Aligned with the WP3, a common evaluation and validation framework has been defined and adopted in WP4.

In detail, the validation methodology adopted and described below is based on the definition of a set of functional objectives that catches the final goals of each test performed on specific software prototypes or algorithms. Each test is structured with the aim of verifying a particular functionality or expected behaviour of the System Under Test (SUT) in a simplified environment which emulates the interaction with other architectural components or the data consumed by the component/algorithm itself. This approach allows identifying possible inappropriate behaviours that depend on the single component/algorithm in an isolated and controlled environment

Correlation between the global project objectives, expected results in terms of 5G KPIs, and individual applications achievements is provided to check WP4 related project objectives are properly addressed by the validations.

6.2 Scenarios

This section defines a set of scenarios for the validation and evaluation of different applications in line with the use cases defined in WP1.

6.2.1 Vehicle mobility – high speed train scenario

High speed train is a common public transportation for inter-city connection. Currently, the velocity of a high-speed train in commercial operation can achieve up to 300 km/h, and novel trains with more than 500 km/h have been tested and demonstrated in Japan. It can be foreseen that more and more people will use high-speed trains for traveling or business trips.

Passengers on high-speed trains will need mobile internet for entertainment (such as on-line gaming, HD video, social cloud services), or for business (such as accessing company information systems or having video conferences). Therefore, challenges in providing satisfying Quality of Experience (QoE) to a crowd of passengers (e.g., more than 500 people) on a high speed train with the speed up to or greater than 500 km/h need to be identified and addressed in a cost effective way.

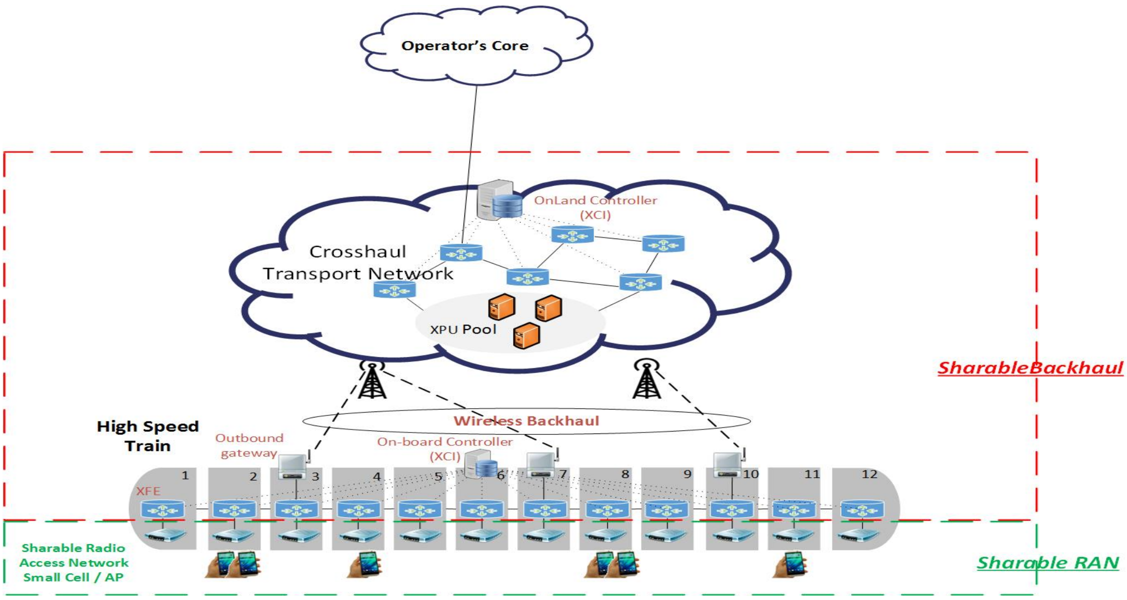


Figure 34: MMA Scenario for High speed train over 5G-Crosshaul transport network over 5G-Crosshaul transport network

Due to the obstacles like Doppler effect caused by high mobility and the penetration loss resulted from the metallic carriage, the direct access from on-board terminals to the base stations will likely be hampered. Therefore, as shown in Figure 34, one or several Points of Access (PoA), such as small cells or Wi-Fi APs, will be installed on a high speed train to provide broadband access to the passengers inside each car of the train. The mobile backhaul for the PoA is provided via an outbound gateway(s) which connect to the land base stations and the 5G-Crosshaul transport network.

6.2.2 Media distribution scenario: Content Delivery Network (CDN)

This scenario is focused on the media distribution, in particular, in Video on Demand (VoD). Video consumption is becoming the most demanding component of the traffic growth observed in current networks and this trend will even increase in the future 5G networks. 5G-Crosshaul provide efficient procedures and applications to deliver media content to end users saving transport resources and reducing transmission delays.

A CDN provider based on their CDN service requirements and the 5G-Crosshaul network would deploy a content delivery infrastructure by origin and replica servers connected between them which would be instantiated in different XPUs distributed along the 5G-Crosshaul network (Figure 35).

The coming requests in the network from the different users would be managed and served by the different controllers and applications defined in 5G-Crosshaul network. When a user requests a specific video, 5G-Crosshaul will be capable to provide the content from the more suitable server enhancing the quality of service.

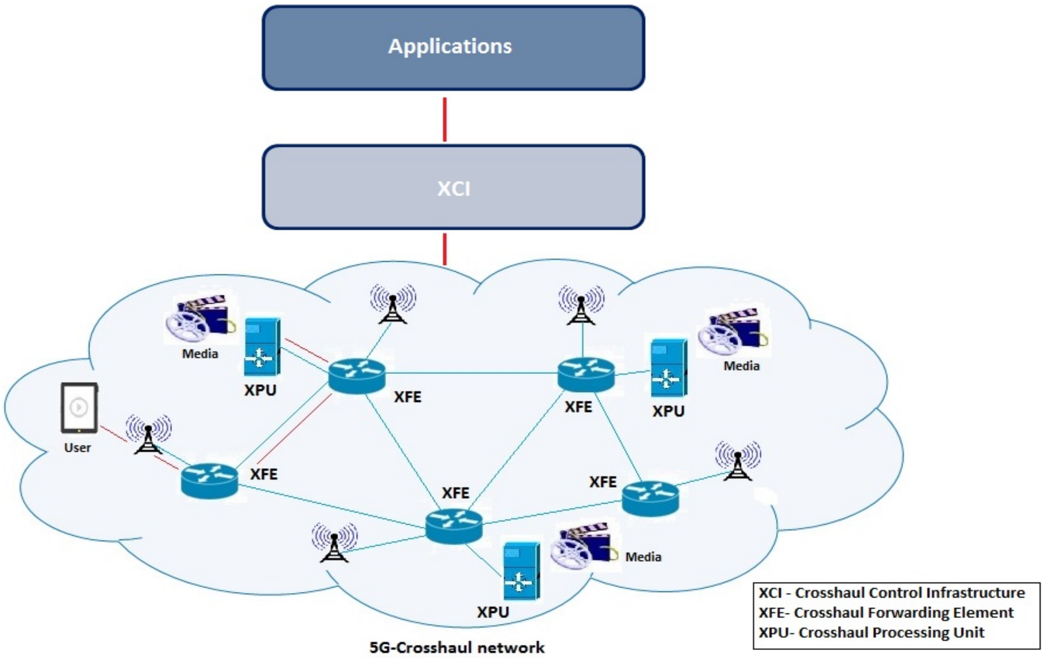


Figure 35: Media distribution scenario: Content Delivery Network (CDN)

6.2.3 Media distribution scenario: real time TV Broadcast

This scenario is focused on the live media distribution, in particular, in TV Broadcasting, a point to multipoint video delivery providing the real-time video service to a targeted area.

TV broadcasting and mobile broadband are essential parts of today's society. Content distribution is expected to contribute ~70% (the dominant contributor to the mobile data traffic demand) of all the mobile traffic in 2018.

A content provider may agree to the terms of the service (defining requirements for delay, video quality and expected QoS/QoE for the users) and the 5G-Crosshaul network would offer a content delivery infrastructure by determining, creating and reconfiguring the optimum path for the video transmission within the 5G-Crosshaul network (Figure 36).

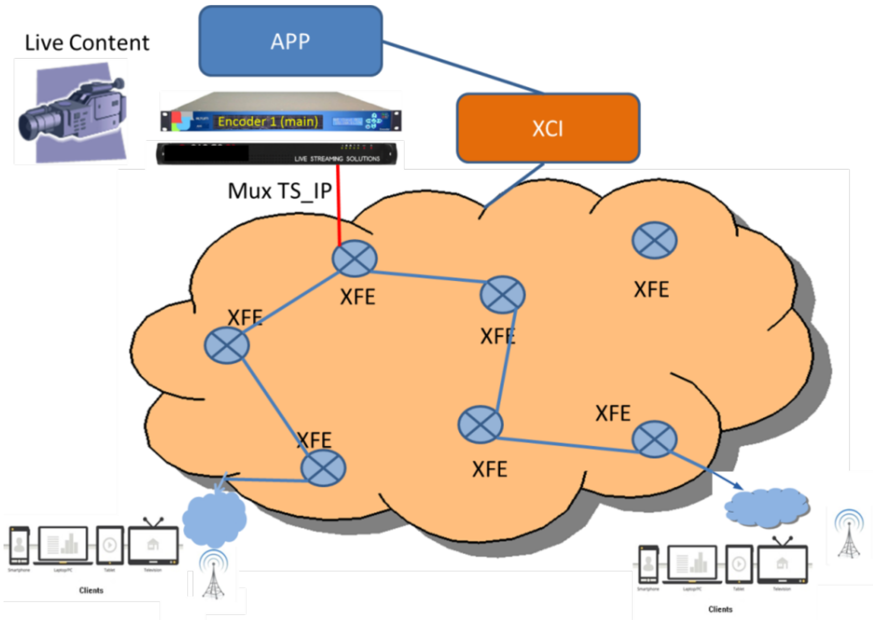


Figure 36: Media distribution scenario: real time TV Broadcast

6.2.4 Dense urban society scenario

In “dense urban information scenario”, network densification is necessary because of the high traffic volume created not only by smart phones, tablets and other electronic devices but also by augmented reality information, including sensors and wirelessly connected cameras, to name a few. Typical environments are indoor offices, shopping malls, university campuses, airports, train/bus stations, open squares, urban canyons, etc, where users tend to gather and move as “large and dynamic crowds” while they want to keep connectivity to the cloud. 5G-Crosshaul provides efficient deployment and management procedures for VNOs in such densely located access networks offering a flexible and intelligent transport between core network elements and the radio access side.

A typical 5G-Crosshaul network deployed by 5G-Crosshaul Physical Infrastructure Provider (X-PIP) is shown in Figure 37. By the increase of traffic demand, more small-cell BSs or/and RRHs are needed to be deployed in the dense urban information scenarios. 5G-Crosshaul offers flexible fronthaul/backhaul transport layer over cost effective wireless links or other wired technologies including Ethernet, copper and optical fiber. VNOs deploy their Virtual Networks (VNs) over the 5G-Crosshaul infrastructure via the Multi-Tenancy Application to increase the resource utilization with reduced CAPEX. The VNOs can operate their VNs effectively via Resource Management Application to dynamically control the network resources based on the mobility of dynamic crowds. If the dynamic crowds move, the Mobility Management Application will notify the RMA or/and MTA to adapt the network to fulfill users’ requirements. In the case of off-peak time, such as holidays, the Energy Management and Monitoring Application will be used to increase the energy saving by terminating transport links which are lightly or not used, and turning computing nodes in stand-by or even off. Therefore, 5G-Crosshaul is also effective to reduce OPEX in VNOs by using RMA and EMMA.

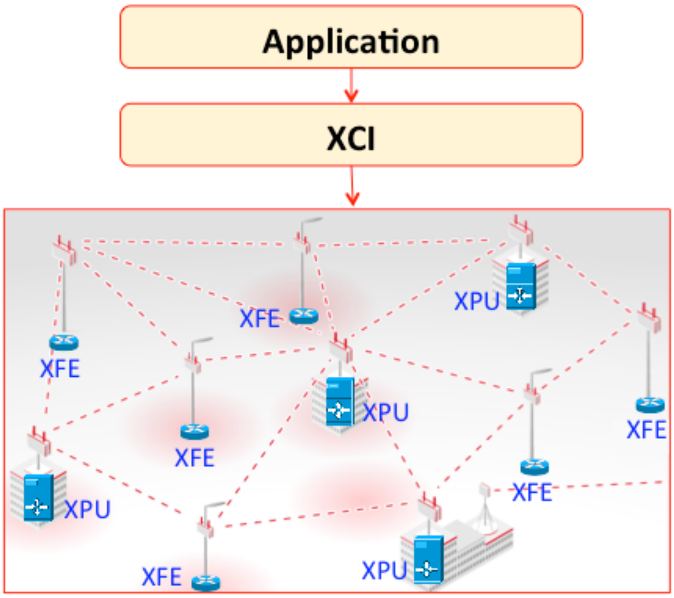


Figure 37: Dense urban information scenario

6.2.5 Multi-tenancy scenario

We consider a general multi-tenant scenario as shown in Figure 38, where the owner of Crosshaul (who owns the physical Crosshaul resources) is in charge of allocating virtual resources over its substrate network to multiple virtual networks, which are offered to different tenants. Each tenant owns a virtual infrastructure referred to as a slice, composed of a set of virtual nodes, e.g. virtual routers and switches, interconnected by virtual links over the substrate network and owning a subset or slice of the physical infrastructure including computing, storage and networking resources. The MTA is designed as a slice orchestrator (as the decision entity) to decide the allocation/modification/deallocation of resources for virtual infrastructures / slices which are isolated per tenant upon their requests. A Virtual Infrastructure is composed of a coherent set of heterogeneous network, compute and storage infrastructure resources. The MTA will allow both static and dynamic allocation of resources to the tenants. To enable the multi-tenancy service on network slicing, provisioning and allocation of resources in the system, the MTA requires the support from the control plane (XCI) to configure the mapping of virtual infrastructure onto the physical one and allocate the corresponding computing, networking and storage resources; and also requires the support from the data plane entities (e.g., XPFE) to identify the tenants by the tenant ID defined in the XCF frame.

Offering a virtual infrastructure implies that (part of) the Crosshaul infrastructure is used to create one or more tenant's virtual infrastructure, composed of:

- Nodes (a subset of the underlying infrastructure nodes, i.e. XFEs or XPUs)
- Links (bandwidth connections between nodes, identified by bandwidth and some quality parameters, e.g. min availability, max BER, max. latency/jitter, etc.)

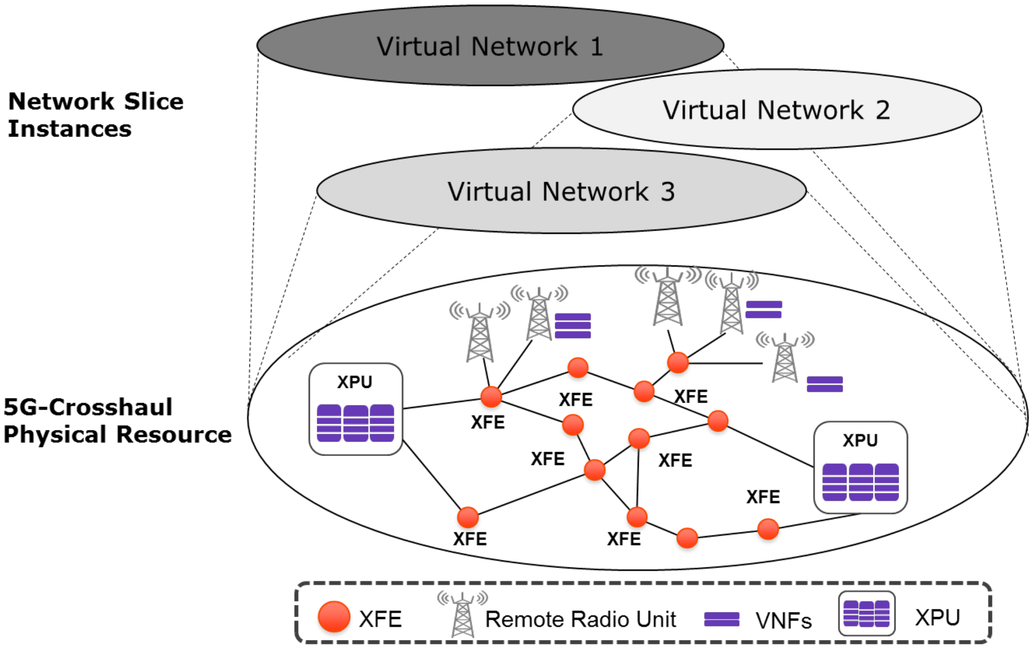


Figure 38: MTA scenario - creation of virtual infrastructure networks

A virtual infrastructure allocation request includes a set of virtual infrastructure nodes (virtual network nodes) and virtual infrastructure computing nodes (virtual hosts), a set of virtual infrastructure links, and requested SLA such as bandwidth managements for the connections between two end points. The VNOs can also ask for a slicing update when its demand and SLA requirements change.

To create a network slice, we need mechanisms for partitioning a network. In this scenario we consider the following partitioning approaches:

1. 5G-Crosshaul XPU has a containment relationship with e.g. Virtual Machines (VMs) or Containers depending on the type and use of a hypervisor. A virtual machine can, in turn, become an XPU for a given tenant slice, as part of the physical infrastructure.
2. 5G-Crosshaul XFE consists of some ports and interfaces; here the XFE can be partitioned by assigning a subset of the ports/ interfaces to a specific tenant or a group of tenants.
3. Links bandwidth can be partitioned, for example, by assigning VLANs to a specific tenant or a group of tenants.

Multi-tenancy sharing in high speed train scenario

In addition to the above general scenario, another multi-tenancy scenario that will be investigated in this project is multi-tenancy sharing in the high speed train scenario, as shown in Figure 39, where the mobile users taking a high speed train belonging to different Virtual Network Operators (VNOs) will move at a very high speed.

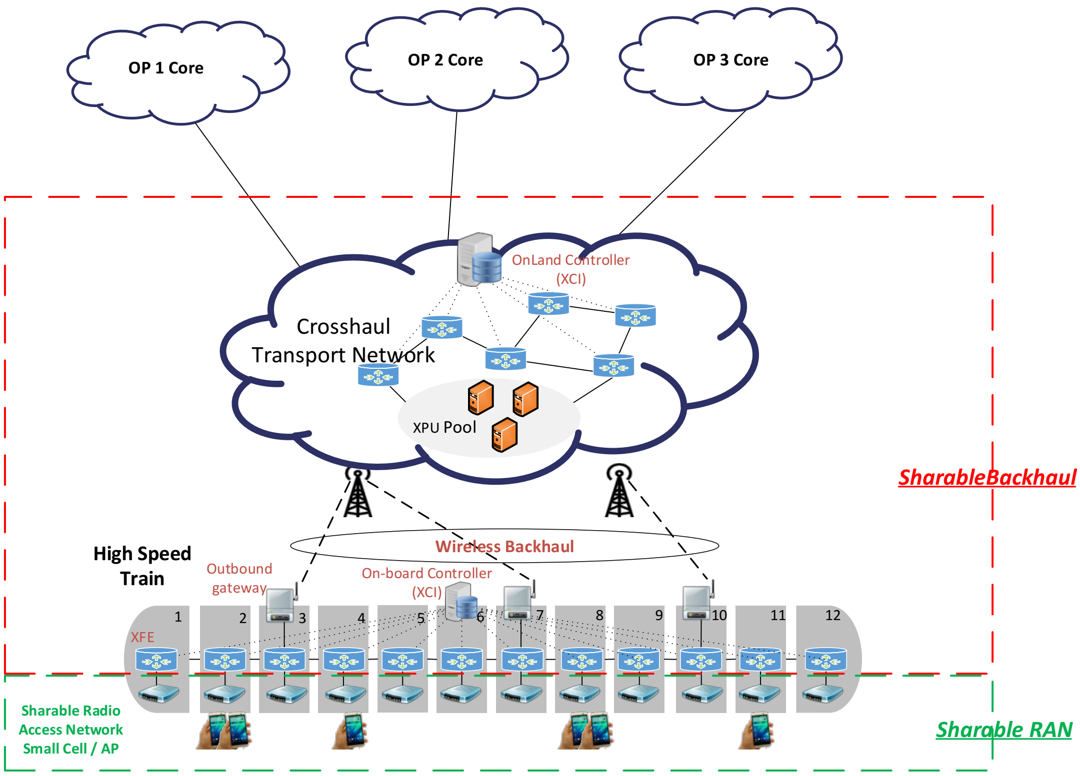


Figure 39: MTA Scenario for High speed train over 5G-Crosshaul transport network

In this scenario the on-board/on-land XCI can be used to manage the multi-tenancy sharing, it helps in partitioning the physical network into the virtual network with the help of VIMaP, which can be utilized by multiple operators to achieve multi-tenancy.

Inside a high speed train, a third party (i.e., VNP) negotiates with a PIP to deploy an on board multi-tenancy sharing environment such as deploying a BBU and multiple RRHs in each car of a high speed train to construct a shared fronthaul (FH) network or deploying a small cell in each car to create a sharing 5GPoA access network to provide multi-tenancy sharing in a high speed train. An on board XCI controller deployed by VNP will manage and allocate the available network resources on board to different VNOs, i.e., responsible for virtual network deployment for each tenant on board. Furthermore, all traffic from end users to the internet will only be transmitted through the outbound XFE to the 5G-Crosshaul backhaul network using 5G-Crosshaul Control Frame (XCF). The on-board controller maintains the virtual topology created by On-Land controller to the end devices and forwards the traffic to the most suitable outbound gateway.

For the backhaul network, the virtual network provider can initiate a request to the Land Controller, in response to the request of the On-Land controller which requests network slice from (VIMaP) and dynamically computes suitable path with in the provided slice and dynamically allocates bandwidth.

6.2.6 Mapping to Use Cases and Applications

Table 16: Applications mapping to the defined 5G-Crosshaul scenarios

Applications mapping to Scenarios	
Scenarios	Applications

Vehicle mobility – high speed train scenario	MMA will be used to solve the Handover (HO) problem in this scenario. MMA will exploit the deterministic trajectory of the train in advance. Consequently, the MMA will reduce the handover time and maintain a high level of successful handover without degrading the UE performance. RMA will compute the optimum routing path on request between two provided nodes from MMA. MTA will provide multi-tenant for MMA to be supported by different operators.
Media distribution scenario: Content Delivery Network (CDN)	CDNMA: responsible for CDN infrastructure instantiation, control and management of the CDN service. RMA: to deal with the network resources to compute the optimal paths between the user and the CDN server assigned. MTA: to provide tenant identification and per-tenant monitoring information for each tenant. MMA: to provide the user network entry point to the CDNMA.
Media distribution scenario: real time TV Broadcast	TVBA: responsible for TV service requirements establishment, control and management of the video play-out. RMA: to deal with the network resources to compute the optimal paths for the broadcast tree. MTA: to provide tenant identification and per-tenant monitoring information for each tenant.
Dense urban society scenario	MTA will be in charge of configuring per-tenant VNs. RMA will deal with the network resources to compute the optimal paths per-tenant taking into account the newly deployed end points and property of dynamic crowd. The RMA shall also solve the problem of function and service placement over computing nodes. EMMA would monitor the power consumption of the system and provide information to be used for dynamic control of the

	network topology for energy saving. MMA would monitor the user location in terms of entry point to the Crosshaul network. If the network entry point changes due to the dynamic crowd, the MMA will notify the MTA, RMA, and EMMA for efficient network reaction.
Multi-Tenancy scenario	MTA is in charge of creating tenants by deciding an optimum subset of nodes (node mapping) and links (link mapping) in the substrate network to create a virtual infrastructure for a tenant which satisfies its resource demand and SLAs, known as the virtual network embedding (VNE) problem. RMA will compute the optimum routing path on request of MTA to decide on the mapping between a virtual link to a physical path (which is composed of one or a set of physical links). EMMA may provide the MTA the monitoring services for each physical/virtual infrastructure or single physical/virtual elements, and provisioning of "energy-optimized" network paths or even "energy-optimized" virtual infrastructures (including network services with VNFs), depending on the use case requirements.

6.3 Project KPIs

This section shows the definition of the main KPIs followed by the project for the validation and evaluation of the scenarios managed by the designed applications. The KPIs serve an essential role as a key measure of how effectively the applications have performed in relation to identified, required and agreed-upon strategic objectives.

The main KPIs identified are represented in the following table:

Table 17: 5G-Crosshaul KPIs

KPI	Use case(s) of interest	Unit	Details
Bandwidth capacity	Vehicle Mobility (High Speed train), Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing	Mbps	Maximum reliable transmission rate that a path can provide
Bandwidth capacity density	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge	$Mbps/Km^2$	Aggregate data volume rate per geographical area unit

	Computing		
Guaranteed UE data rate	Vehicle Mobility (High Speed train)	Mbps	Guaranteed user data rate that the transport infrastructure can provide
UE density	Dense urban society, Vehicle Mobility (High Speed train)	$Terminal/Km^2$	Number of UEs supported per geographical area unit
Latency	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing	ms	Network latency is the time taken by data to get from one designated point (5G Point of Attachements…) to another (Core, MEC server…)
Jitter	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing	ms	Variation in latency as measured in the variability over time of the packet latency across a network.
Bit error rate	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing	%	Number of bit errors divided by the total number of transferred bits during a studied time interval
Packet Loss rate	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing	%	Percentage of packets lost and should have been forwarded by a network but were not with respect to packets sent
Time of recovery	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense	ms	Times it takes since a change or a failure occurs in the transport network and an application responsible reacts and re-configures the network

	Urban Society, Multi-tenancy, Mobile Edge Computing		accordingly.
Availability	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing	%	Probability that the services are not disrupted during the expected time.
Mobility support	Vehicle mobility	km/h	Maximum speed at which applications and XCI can handle user handovers successfully
Max number of simultaneous flows	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Mobile Edge Computing	n	Maximum number of simultaneous flows that the transport infrastructure can support
Density of simultaneous flows	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Mobile Edge Computing	n/ Km^2	Supported density of flows per unit of area
Energy efficiency/savings	Dense urban society	(%)	Percentage of energy savings in the transport infrastructure
Provisioning time	Vehicle Mobility, Media distribution (CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing	ms	Time required to instantiate a service or an adaptation action
Cost savings	Vehicle Mobility, Media distribution	%	Percentage of CAPEX/OPEX/TCO savings in

	(CDN & TV Broadcasting and Multicasting), Dense Urban Society, Multi-tenancy, Mobile Edge Computing		the transport segment

6.4 Mapping between test-cases, 5G-Crosshaul objectives and 5G-PPP KPIs

The 5G-Crosshaul project defines 8 objectives, specified in the Description of Action, which drive the execution of the activities across different work packages. Each objective is associated to a number of 5GPPP KPIs and their evaluation allows verifying if and how the project meets its defined objectives.

The test cases and the evaluation plans of each application are detailed in Appendix III: Evaluation plans and initial results per application.

Table 18 analyses the mapping between the WP4 application test cases and algorithms validation procedures defined in this IR to the related project objectives and 5GPPP KPIs, identifying the WP4 contributions to the validation of the whole project.

Table 18: Mapping between test-cases, objectives and KPIs

Objective	5GPPP KPI impact	Feature	Test Case / Algorithm
Obj.1: Design of the Crosshaul Control Infrastructure (XCI)	Energy efficiency improvement by at least a factor of 3.	Energy monitoring	NXW_EMMA_APP_01
		Automated configuration of devices powers state	NXW_EMMA_APP_03
Obj.6: Design scalable algorithms for efficient 5G Crosshaul resource orchestration	Scalable management framework: algorithms that can support 10 times increased node densities	Dynamic Mobility Management (MMA)	UC3M_MMA_001 UC3M_MMA_002 UC3M_MMA_003
		Resource Management (RMA) for Dense Networks	NEC_PCBSSPLIT_01 NEC_PCBSSPLIT_02 NEC_PCBSSPLIT_03
	Enable deployment of novel applications reducing the network management Operational	Multi-tenancy enables efficient sharing of Corsshaul infrastructure	NEC_MTA _01 NEC_MTA _02 ITRI_on-board MTA _01 ITRI_on-board MTA _02 ITRI_on-land MTA _01

	Expenditure (OPEX) by 10% in terms of provisioning.	Flexible functional placement (RMA)	NEC_PCBSSPLIT_01 NEC_PCBSSPLIT_02 NEC_PCBSSPLIT_03
	Increase of total 5G-Crosshaul network throughput by > 20% using resource optimization alone compared to current operators' practice.	Cost-efficient Crosshaul Resource Management focused on joint path computation, VNF Placement and BS functional split (Resource Management Application)	CREATE-NET_PCVNFP_01 CREATE-NET_PCVNFP_02 CREATE-NET_PCVNFP_03 NEC_PCBSSPLIT_01 NEC_PCBSSPLIT_02 NEC_PCBSSPLIT_03
Obj.7: Design essential 5G Crosshaul integrated (control & planning) applications	Reduce energy consumption in the 5G Crosshaul by 30% through energy management.	Energy-related policies Energy-efficient network path provisioning Energy-efficient Network Service Provisioning	NXW_EMMA_APP_02 NXW_EMMA_APP_04 NXW_EMMA_APP_05
	Reduction of 5G-Crosshaul infrastructure Capital Expenditure (CAPEX) by 20% due to automated planning.	Computation of optimal BS functional split and placement of virtual functions (Resource Management Application)	NEC_PCBSSPLIT_01 NEC_PCBSSPLIT_02 NEC_PCBSSPLIT_03
Obj.8: 5G Crosshaul key concept validation and proof of concept	Orchestration of 5G Crosshaul resources based on traffic load variation	Resource orchestration for CDN	ATOS_CDNMA_APP_01 ATOS_CDNMA_APP_02 ATOS_CDNMA_APP_03 ATOS_CDNMA_APP_04
		Resource orchestration for TV broadcasting services	VIS_TVBA_APP_01 VIS_TVBA_APP_02 VIS_TVBA_APP_03 VIS_TVBA_APP_04

			VIS_TVBA_APP_05
	Self-healing mechanisms for unexpected link failures	Resource Management Application on automated path computation	NEC_PCBSSPLIT_01 NEC_PCBSSPLIT_02 NEC_PCBSSPLIT_03

7 Conclusions

This document reported the progress on the design, implementation, interaction, validation and evaluation of the applications for the first year of 5G-Crosshaul project, i.e. July 2015 to June 2016.

This document introduces the 5G-Crosshaul applications, these applications are sitted on top of the XCI (although not restricted to) and through the NBI they receive as input suitable abstractions that are necessary to control Crosshaul resources

Each application implements its own algorithms to provide the service required. All algorithms are focused on the optimization and efficiency of the resources managed. Heterogeneous transport technologies, computing and storage resources are controlled to deliver the high quality services requested by the end users. MTA, MMA, RMA, VIMaP, EMMA, CDNMA and TVBA are the applications which were described in this deliverable.

These applications interact between them and with the XCI. The usage of REST APIs is adopted where possible, mostly making use of the HTTP protocol for message transport, YANG language for information model specification and Json encoding for resource description. This approach allows to easily modelling the communication between applications as a service consumer/provider interaction.

The validation and evaluation model used is based on functional testing that involves testing applications against the technical 5G-Crosshaul requirements. Functional testing has been defined using the functional specifications or by using the design specifications like use cases provided by the design team of each application.

8 Appendix I: NBI models of the Applications

8.1 Abstract modelling of applications NBI

This section provides the description of the abstract primitives which must be supported by the applications NBI, specifying the messages, their senders and receivers (i.e. the producer (SP) or the consumer (SC) of the service) as well as their parameters. The mapping of these primitives in specific protocol messages will be addressed in section 0.

8.1.1 Multi-Tenancy Application

The MTA provides two main interfaces: the first allows other applications to request the creation, querying and termination of a virtual infrastructure for a given tenant, while the second allows the same applications to operate existing virtual infrastructure, for example by configuring a forwarding rule on a virtual network node.

The MTA NBI primitives are the following:

Table 19: MTA NBI abstract primitives

Message	Direction	Parameters	Description
Create_VI_request	SC → SP	Virtual infrastructure description Tenant ID	Request to create a new virtual infrastructure.
Create_VI_response	SP → SC	Virtual infrastructure ID	
Delete_VI_request	SC → SP	Virtual infrastructure ID Tenant ID	Request to remove an existing virtual infrastructure.
Delete_VI_response	SP → SC	--	
Get_VI_request	SC → SP	Virtual infrastructure ID Tenant ID	Request for the details of a virtual infrastructure.
Get_VI_response	SP → SC	Virtual infrastructure description	
Get_VI_list_request	SC → SP	Tenant ID	Request for the list of all the virtual infrastructure associated to a tenant.
Get_VI_list_response	SP → SC	Virtual infrastructure ID []	
Get_VI_element_request	SC → SP	Tenant ID Virtual infrastructure ID VI element ID	Request for the details of a virtual element within a given virtual infrastructure.
Get_VI_element_response	SP → SC	VI element description	

Execute_action_request	SC → SP	Tenant ID Virtual infrastructure ID VI element ID VI action	Request to execute an action (e.g. a configuration command) on an existing virtual infrastructure.
Execute_action_response	SP → SC	Result code	

The main information elements are defined in the following tables.

Table 20: VI description attributes

Virtual Infrastructure description				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
VI ID	No	1	UUID	Unique VI ID
Tenant ID	No	1	UUID	Unique ID of the tenant owing the VI
Name	Yes	1	String	Human readable name for the VI.
VI element description	Yes	N	Element	Elements belonging to the VI.

Table 21: VI element attributes

VI element description				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
Element ID	No	1	UUID	Unique element ID
Type	No	1	Enum	Type of virtual element, e.g. virtual-bridge, virtual-router, virtual-link, virtual-port, virtual-machine
Name	Yes	1	String	Human readable name for the element.

Parent Element	No	1	UUID	ID of the parent entity the current element is associated to.
Element Attribute	Yes	N	<key, value>	A <key, value> map with the list of attributes associated to the given VI element.
Physical map	No	N	<key, value>	List of physical resources the virtual element is mapped to.

Table 22: VI action attributes

VI action				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
Action Type	No	1	Enum	Type of action to be executed. The allowed values depend on the type of element the action is executed on (e.g. start, stop, launch_script for virtual machines; drop_packet, forward_packet for virtual network nodes, etc.)
Action Parameter	Yes	N	<key, value>	A <key, value> map with the list of input parameters for the given action.

8.1.2 Mobility Management Application

The MMA provides features for optimizing traffic offloading for media distribution (CDN and TV Broadcasting) and mobility management. MMA will notify the change of the Network Entry Point to the CDN subscribed previously.

The MMA NBI primitives are the following:

Table 23: MMA NBI abstract primitives

Message	**Direction**	**Parameters**	**Description**
Add_Notification_Request	SC → SP	URL	Subscription request for notification reception
Add_Notification_Response	SP → SC	Result Code	
Delete_Notification_Request	SC → SP		Request for deleting the

Delete_Notification_Response	SP → SC	Result Code	subscription of notification reception
Edit_Notification_Request	SC → SP	URL	Request for editing the subscription (URL) of notification reception
Edit_Notification_Response	SP → SC	Result Code	
Get_CDN_Node_ Request	SP → SC	XPFE IP MAC	Notify to the CDN the presence of a new user for the CDN Node assignment.
Get_CDN_Node _Response	SC → SP	CDN NODE	

The information elements are defined in the following table:

Table 24: Attributes

MMA Description				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
URL	No	1	String	URL of notification
XPFE IP	No	1	String	IP of the PoC to the 5G-Crosshaul
MAC	No	1	String	MAC of the user
CDN Node	Yes	1	String	IP of the CDN Node assiged

8.1.3 Resource Management Application

The RMA NBI primitives are provided in Table 7. Two services are provided:

• PC (Path Computation) service: the request is for a path computation between source and destination nodes. The inclusion of some nodes can be requested (useful for an application that needs to connect nodes of a service chain) or excluded. The request can include bandwidth and latency requirements.

• PC-NFVP (Path Computation – NFV Placement) service: the request is for the placement of NFVs on XPUs and the computation of the path connecting them. Also in this case, nodes can be included or excluded, required bandwidth and latency can be expressed. The request must contain the ordered list of VNFs to be placed.

Given the aforementioned services, the RMA will provide the primitives to request them. The service will be synchronous, i.e., the RMA will provide a reply to the POST issued by the SC (e.g. an application or the network manager). This means that no primitive for getting the status of the computation is provided. This can be changed at a later stage in the case in which the consortium decides to move to an asynchronous service.

Table 25: RMA NBI abstract primitives

Message	**Direction**	**Parameters**	**Description**
Request_PC_service	SC → SP	Tenant ID	Request the PC

		Source node ID(s) Destination node ID(s) Include node ID(s) Exclude node ID(s) Bandwidth Latency	service
Request_PC-VNFP_service	SC → SP	Tenant ID Source node ID(s) Destination node ID(s) Include node ID(s) Exclude node ID(s) Bandwidth Latency VNFs	Request PC-VNFP service

The main information elements are defined in the following tables.

Table 26: PC service description attributes

PC services attributes				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
Tenant ID	No	1	UUID	Unique tenant ID
Source Node ID	No	1	UUID	Unique source node ID
Destination Node ID	No	1	UUID	Unique destination node ID
Include Node ID(s)	Yes	1:N	UUIDs list	List of ordered include unique node IDs
Exclude Node ID(s)	Yes	1:N	UUIDs list	List of ordered exclude unique node IDs
Bandwidth	Yes	1	Integer	Constraint on the bandwidth requirement for the path
Latency	Yes	1	Integer	Constraint on the latency requirement for the path

Table 27: PC-NFVP service description attributes

PC-VNFP services attributes

Attribute	Optional	Cardinality	Type	Description
Tenant ID	No	1	UUID	Unique tenant ID
Source Node ID	No	1	UUID	Unique source node ID
Destination Node ID	No	1	UUID	Unique destination node ID
Include Node ID(s)	Yes	1:N	UUIDs list	List of ordered include unique node IDs
Exclude Node ID(s)	Yes	1:N	UUIDs list	List of ordered exclude unique node IDs
Bandwidth	Yes	1	Integer	Constraint on the bandwidth requirement for the path
Latency	Yes	1	Integer	Constraint on the latency requirement for the path
VNFs	No	1:N	VNF list	List of ordered VNFs belonging to the forwarding graph

8.1.4 Virtual Infrastructure Manager and Planner

Let us note the special case of the VIMaP component, initially conceived as an application, but later redefined as an inherent part of the XCI, once the Crosshaul XCI integrated the ETSI NFV MANO as part of the main control architecture. Consequently, VIMaP is re-positioned as part of the XCI offering the services to applications.

The VIMAP APIs are offered to applications, detailed in D3.1 [2], Section 7. VIMaP allows CRUD operations for the management of VM, connectivity and flow management for their interconnection and high level constructs related to the offering of Virtualized Infrastructures (VI) and ETSI Network Services (NS).

In particular, the XCI NBI services that are (in part) implemented by VIMaP are:

- Topology and Inventory services (see [2], Section 7.1)
- Provisioning and Flow actions (see [2], Section 7.2)
- IT infrastructure and inventory (see [2], Section 7.3)

In some cases, these APIs may rely, in turn, on specific services offered by lower level components (e.g. SDN or cloud controller), thus acting as a single endpoint front-end.

Additionally, the VIMaP component offers specific APIs for the allocation of virtualized infrastructures for Network Services enabling Network Slicing (see [2] Section 7.5). Regarding the information model, the VIMaP objects (network slice and network_service_layout objects) are considered a virtual network from the point of view of the IT infrastructure and inventory service. A slice or a network service layout entails a set of calls defining the flow patterns between these components.

8.1.5 Energy Management and Monitoring Application

The EMMA application provides features for energy monitoring applied to both physical and virtual infrastructures, exposing consolidated energy-related measurements to external applications. This service is enabled through an interface which supports queries related to energy parameters collected and/or computed over entire infrastructures or single elements.

Moreover, the EMMA performs active procedures (triggering the proper architecture components) to establish or re-plan energy efficient network paths or virtual infrastructures, working on on-demand or automated basis. The external client invokes the on-demand services through an interface which allows building paths and VIs based on energy-aware metrics and objective functions. On the other hand, the automated re-planning procedures are not triggered explicit requests from external entities, but they are the result of internal processing and decisions taken considering energy policies, e.g. describing thresholds. In this case, the EMMA NBI must offer APIs to load and remove policies.

The EMMA NBI primitives are the following:

Table 28: EMMA NBI abstract primitives

Message	Direction	Parameters	Description
Get_Phy_Infrastructure_ Monitoring_Request	SC → SP	Monit parameter types	Query for energy parameters monitored on a phy infrastructure
Get_Phy_Infrastructure_ Monitoring_Response	SP → SC	Monit parameters type Monit parameters value	
Get_VI_ Monitoring_Request	SC → SP	VI ID Tenant ID Monit parameter types	Query for energy parameters monitored for the phy resources corresponding to the given VI
Get_VI_ Monitoring_Response	SP → SC	Monit parameters type Monit parameters value	
Get_Phy_Element_ Monitoring_Request	SC → SP	Phy element ID Monit parameter types	Query for energy parameters monitored on a given phy element (e.g. a server, a network node, etc.)
Get_Phy_Element_ Monitoring_Response	SP → SC	Monit parameters type Monit parameters value	
Get_Virtual_Element_ Monitoring_Request	SC → SP	VI ID Tenant ID Virtual element ID Monit parameter types	Query for energy parameters monitored for the phy resources corresponding to the given virtual element

Get_Virtual_Element_ Monitoring_Response	SP → SC	Monit parameters type Monit parameters value	
Create_Path_Request	SC → SP	Source Entity Destination Entity Objective Function Metric	Request to create a new path adopting criteria related to power consumption.
Create_Path_Response	SP → SC	Path ID	
Create_VI_request	SC → SP	Virtual infrastructure description Tenant ID Objective Function Metric	Request to create a new VI adopting criteria related to power consumption.
Create_VI_response	SP → SC	VI ID	
Add_energy_policy_ request	SC → SP	Policy target type Policy target ID Policy criteria Policy action	Request to create a new policy for performing automated actions on VIs o network paths (policy target) based on some energy-related criteria.
Add_energy_policy_ response	SP → SC	Result code Policy ID	
Remove_energy_policy_request	SC → SP	Policy ID	Request to remove an existing policy
Remove_energy_policy_response	SP → SC	Result code	
Change_policy_status_ request	SC → SP	Policy ID Policy Status	Request to activate or de-activate an existing policy.
Change_policy_status_ response	SP → SC	Result code	
Retrieve_policy_request	SC → SP	Policy ID	Request to retrieve the description and status of an existing policy
Retrieve_policy_ response	SP → SC	Policy ID Policy target type Policy target ID	

		Policy criteria Policy action Policy status	

The information elements are mainly simple type elements (UUIDs for identifiers, unit for result codes, enums for monitoring parameter types, objective functions, policy target types and policy status). The format of the "virtual infrastructure description" element is reported in the Appendix. The format of the other complex elements is reported below.

Table 29: Source/Destination entity attributes

Source Entity / Destination entity				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
Src/Dst node ID	No	1	UUID	Unique identifier of the network node.
Src/Dst port ID	No	1	UUID	Unique identifier of the port within the network node
Src/Dst resource type	Yes	1	Enum	Type of the input/output resource within the port (e.g. lambda, time-slot, VLAN, OF match …). It depends on the specific technology.
Src/Dst resource value	Yes	1	String	Input/output resource specification. The specific format depends on the type.

Table 30: Metric attributes

Metric				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
Metric Type	No	1	Enum	Type of metric to be considered (e.g. power consumption, the number of nodes to be switched on, etc.). In this context only energy-related parameters are considered.
Metric Value	No	1	String	The target value for the given metric.

Table 31: Policy criterion attributes

Policy Criterion				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**

Monitoring parameter type	No	1	Enum	Type of monitoring parameter to be considered when applying the policy.
Criterion	No	N	Enum	Type of comparison to be applied (i.e. ==; >=; <=)
Threshold	No	N	Enum	The threshold value for the given monitoring parameter.

Table 32: Policy action

Policy action				
Attribute	**Optional**	**Cardinality**	**Type**	**Description**
Action Type	No	1	Enum	Type of action to be executed (re-plan network path, re-plan VI, re-plan VM location, etc.).
Action Parameter	Yes	N	<key, value>	A <key, value> map with the list of input parameters for the given action.

Application NBI protocol specification

This section maps the abstract primitives defined in the previous section in REST API messages. Each Request/Response interaction identified before in terms of abstract primitives is modelled through an HTTP request/reply synchronous communication. Each entity is represented as a resource with a given set of attributes. The resource instance is mapped to the url of the HTTP messages, while its attributes are encoded in Json format and included in the body of the HTTP messages. Response codes are mapped to HTTP result codes.

8.1.6 Multi-Tenancy Application

The MTA REST APIs, with sample Json contents, are specified in the table below.

Table 33: MTA REST APIs

Abstract messages	**HTTP method, URI, json example**	**Response code**
Create_VI	POST /mta/tenant/{tenantID}/vi	202 – Accepted 400 – Bad Request 401 – Unauthorized 500 – Internal Server Error
Request example: {		

```
        "vi": {
                "vi-name": "Sample-VI",
                "vi-element": [{
                        "element-id": "vbr01",
                        "type": "virtual-bridge",
                        "name": "virtual-bridge-01"
                }, {
                        "element-id": "vif01",
                        "type": "virtual-interface",
                        "name": "virtual-interface-01",
                        "parent": "vbr01"
                }, {
                        "element-id": "vif02",
                        "type": "virtual-interface",
                        "name": "virtual-interface-02",
                        "parent": "vbr01"
                }]
        }
}
```


Response example:


```
{
        "vi": {
                "vi-id": "0001"
        }
}
```


Delete_VI	DELETE /mta/tenant/{tenant-id}/vi/{vi-id}	200 – OK 401 – Unauthorized 404 – Not found 500 – Internal Server Error
Get_VI	GET /mta/ tenant/{tenant-id}/vi/{vi-id}	200 – OK

		401 – Unauthorized 404 – Not found 500 – Internal Server Error

Response example:


```
{
    "vi": {
        "vi-id": "0001",
    "vi-name": "Sample-VI",
        "vi-element": [{
            "element-id": "vbr01",
            "type": "virtual-bridge",
            "name": "virtual-bridge-01",
     "physical-map": {
                "phy-node": "OF:00:00:00:00:00:00:00:E0"
            }
        }, {
            "element-id": "vif01",
            "type": "virtual-interface",
            "name": "virtual-interface-01",
            "parent": "vbr01",
            "physical-map": {
                "phy-node": "OF:00:00:00:00:00:00:00:E0",
                "phy-interface": "eth1"
            }
        }, {
            "element-id": "vif02",
            "type": "virtual-interface",
            "name": "virtual-interface-02",
            "parent": "vbr01",
            "physical-map": {
```

"phy-node": "OF:00:00:00:00:00:00:00:E0",
"phy-interface": "eth5"
}
}]
}
}

Get_VI_list	GET /mta/tenant/{tenant-id}/vi	200 – OK 401 – Unauthorized 404 – Not found 500 – Internal Server Error

Response example:

[{
"vi-id": "0001"
}, {
"vi-id": "0002"
}, {
"vi-id": "0003"
}]

Get_VI_element	GET /mta/tenant/{tenant-id}/vi/{vi-id}/virtual-bridge/{vbridge-id}	200 – OK 401 – Unauthorized 404 – Not found 500 – Internal Server Error

Response example:

{
"element-id": "vbr01",
"type": "virtual-bridge",
"name": "virtual-bridge-01",
"physical-map": {
"phy-node": "OF:00:00:00:00:00:00:00:E0"

} }		
Execute_action	POST /mta/tenant/{tenant-id}/vi/{vi-id}/virtual-bridge/{vbridge-id}/virtual-interface/{vif-id}/action	200 – OK 400 – Bad Request 401 – Unauthorized 404 – Not found 500 – Internal Server Error
Request example: { "action-type": "drop_packet", "action-parameter": [{ "flow-classifier": { "macethertype": "0x8000", "ipdstaddr": "10.0.0.3", "ipdstaddrprefix": "2", "ipsrcaddr": "10.0.0.2", "ipsrcaddrprefix": "2", "ipproto": "17" } }] }		

8.1.7 Mobility Management Application

The MMA REST APIs, with sample Json contents, are specified in the table below:

Table 34: MMA REST APIs

Abstract messages	HTTP method, URI, json example	Response code
Add_Notification	POST /mma/cdn/	200 – OK 400 – Bad Request 500 – Internal Server Error

Request example: { "url": "http://192.168.1.100:8080/test/" }		
Delete_Notification	DELETE /mma/cdn/	200 – OK 404 – Not found 500 – Internal Server Error
Edit_ Notification	PUT /mma/cdn/	200 – OK 400 – Bad Request 404 – Not found 500 – Internal Server Error
Request example: { "url": "http://192.168.1.200:8080/test/" }		
Get_CDN_Node	POST http://192.168.1.100:8080/test/	200 – OK 400 – Bad Request 401 – Unauthorized 404 – Not found 500 – Internal Server Error
Request example: { "xpfe_ip": "10.0.0.1" "mac": "11:22:33:44:55:66" } Response example: {		

```
        "cdn_node": "20.0.0.1"
}
```


8.1.8 Resource Management Application

The RMA REST APIs, with sample Json contents, are specified in the table below.

Table 35: RMA REST APIs

Abstract messages	HTTP method, URI, json example	Response code
Request_PC_service	POST /rma/request-pc-service	202 – Accepted 400 – Bad Request 401 – Unauthorized 500 – Internal Server Error


```
{
 "tenant-id": "0001",
 "source-node-ids": [
   {
    "node-id": "N0001"
   }
 ],
 "dest-node-ids": [
   {
    "node-id": "N0002"
   }
 ],
 "include-node-ids": [
   {
    "node-id": "N0003",
    "next-hop": "STRICT"
   },
   {
    "node-id": "N0004",
    "next-hop": "LOOSE"
```

```
    }
  ],
  "exclude-node-ids": [
    {
      "node-id": "N0005"
    },
    {
      "node-id": "N0006"
    }
  ],
  "bandwidth": 5000,
  "latency": 10
}
```


Response example:


```
{
  "path": {
    "ports": [
      {
        "node-id": "N0001",
        "port-id": "0001",
        "port-type": "ETH",
        "lambda-id": "",
        "order": 1
      },
      {
        "node-id": "N0001",
        "port-id": "0002",
        "port-type": "DWDM",
        "lambda-id": "1",
        "order": 2
      },
```

```
      {
        "node-id": "N0002",
        "port-id": "0001",
        "port-type": "DWDM",
        "lambda-id": "1",
        "order": 3
      },
      {
        "node-id": "N0002",
        "port-id": "0002",
        "port-type": "ETH",
        "lambda-id": "",
        "order": 4
      }
    ]
  }
}
```


Request_PC-NFVP_service	POST /rma/request-pc-nfvp-service	202 – Accepted 400 – Bad Request 401 – Unauthorized 500 – Internal Server Error


```
{
  "tenant-id": "0001",
  "source-node-ids": [
    {
      "node-id": "N0001"
    }
  ],
  "dest-node-ids": [
    {
```

```
    "node-id": "N0002"
  }
],
"include-node-ids": [
  {
    "node-id": "N0003",
    "next-hop": "STRICT"
  },
  {
    "node-id": "N0004",
    "next-hop": "LOOSE"
  }
],
"exclude-node-ids": [
  {
    "node-id": "N0005"
  },
  {
    "node-id": "N0006"
  }
],
"bandwidth": 10000,
"latency": 10,
"vnfs": [
  {
    "vnf-id": "VNF001",
    "vnf-type": "FIREWALL",
    "order": 1,
    "preferred-xpus": [
      {
        "xpu-id": "XPU001"
```

```
      },
      {
        "xpu-id": "XPU002"
      }
    ]
   },
   {
    "vnf-id": "VNF002",
    "vnf-type": "LOAD-BALANCER",
    "order": 2,
    "preferred-xpus": []
   }
 ]
}
```

Response example:

```
{
 "path": {
  "ports": [
    {
     "node-id": "N0001",
     "port-id": "0001",
     "port-type": "ETH",
     "lambda-id": "",
     "order": 1
    },
    {
     "node-id": "N0001",
     "port-id": "0002",
     "port-type": "DWDM",
     "lambda-id": "1",
     "order": 2
```

```
    },
    {
     "node-id": "N0002",
     "port-id": "0001",
     "port-type": "DWDM",
     "lambda-id": "1",
     "order": 3
    },
    {
     "node-id": "N0002",
     "port-id": "0002",
     "port-type": "ETH",
     "lambda-id": "",
     "order": 4
    }
  ]
 },
 "vnfp": [
  {
   "vnf-id": "VNF001",
   "xpu-id": "XPU001"
  },
  {
   "vnf-id": "VNF002",
   "xpu-id": "XPU002"
  }
 ]
}
```

8.1.9 Virtual Infrastructure Manager and Planner

As a basic component of the XCI, the VIMaP interface has been provided in [2].

8.1.10 Energy Management and Monitoring Application

The EMMA REST APIs, with sample Json contents, are specified in the table below.

Table 36: EMMA REST APIs

Abstract messages	HTTP method, URI, json example	Response code
Get_Phy_ Infrastructure_ Monitoring	GET /emma/phy-monit	200 – OK 401 – Unauthorized 500 – Internal Server Error

Response example:


```
{
	"phy-infrastructure": [{
		"network-node": {
			"node-id": "N0001",
			"power": 100,
			"port": [{
				"port-id": "0001",
				"port-type": "DWDM",
				"port-status": "ENABLED",
				"lambda": [{
					"lambda-id": "1",
					"power": 10
				}, {
					"lambda-id": "2",
					"power": 20
				}]
			}, {
				"port-id": "0002",
				"port-type": "DWDM",
				"port-status": "ENABLED",
				"lambda": [{
					"lambda-id": "1",
```

```
                        "power": 15
                    }, {
                        "lambda-id": "2",
                        "power": 30
                    }]
                }, {
                    "port-id": "0003",
                    "port-type": "DWDM",
                    "port-status": "DISABLED"
                }]
            }
        }, {
            "network-node": {
                "node-id": "N0002",
                "power": 50,
                "port": [{
                    "port-id": "0001",
                    "port-type": "DWDM",
                    "port-status": "ENABLED",
                    "lambda": [{
                        "lambda-id": "1",
                        "power": 10
                    }, {
                        "lambda-id": "2",
                        "power": 20
                    }]
                }, {
                    "port-id": "0002",
                    "port-type": "DWDM",
                    "port-status": "ENABLED",
                    "lambda": [{
```

```
                        "lambda-id": "1",
                        "power": 15
                    }, {
                        "lambda-id": "2",
                        "power": 30
                    }]
                }]
            }
        }]
}
```


Get_VI_Monitoring	GET /emma/vi-monit/{tenant-id}/vi/{vi-id}	200 – OK 400 – Bad Request 401 – Unauthorized 404 – Not found 500 – Internal Server Error

Response example:


```
{
        "vi": {
                "vi-id": "0001",
                "power": "100",
                "vi-element": [{
                        "element-id": "vbr01",
                        "type": "virtual-bridge",
                        "power": 50
                }, {
                        "element-id": "vif01",
                        "type": "virtual-interface",
                        "parent": "vbr01",
                        "power": 30
                }, {
```

```
                    "element-id": "vif02",
                    "type": "virtual-interface",
                    "name": "virtual-interface-02",
                    "parent": "vbr01",
                    "power": 10
                }]
        }
}
```


Get_Phy_Element_Monitoring	GET /emma/phy-monit/network-node/{node-id}	200 – OK 400 – Bad Request 401 – Unauthorized 404 – Not found 500 – Internal Server Error

Response example:


```
{
        "network-node": {
                "node-id": "N0002",
                "power": 50,
                "port": [{
                        "port-id": "0001",
                        "port-type": "DWDM",
                        "port-status": "ENABLED",
                        "lambda": [{
                                "lambda-id": "1",
                                "power": 10
                        }, {
                                "lambda-id": "2",
                                "power": 20
                        }]
                }, {
```

```
                        "port-id": "0002",
                        "port-type": "DWDM",
                        "port-status": "ENABLED",
                        "lambda": [{
                                "lambda-id": "1",
                                "power": 15
                        }, {
                                "lambda-id": "2",
                                "power": 30
                        }]
                }]
        }
}
```


Get_VI_Element_ Monitoring	GET /emma/vi-monit/{tenant-id}/vi/{vi-id}/virtual-bridge/{vbridge-id}	200 – OK 400 – Bad Request 401 – Unauthorized 404 – Not found 500 – Internal Server Error

Response example:


```
{
        "vi-element": {
                "element-id": "vbr01",
                "type": "virtual-bridge",
                "power": 50
        }
}
```


Create_path	POST /emma/net-path-mgt/net-path	202 – Accepted 400 – Bad Request 401 – Unauthorized 500 – Internal Server

		Error

Request example:

```
{
	"net-path": {
		"net-path-name": "Sample-Network-Path",
		"src-entity": {
			"node-id": "N_00001",
			"port-id": "P_00001",
			"resource-type": "VLAN-ID",
			"resource-value": 10
		},
		"dst-entity": {
			"node-id": "N_00004",
			"port-id": "P_00008"
		},
		"objective-function": "minimize-active-phy-node",
		"metric": [{
			"type": "max-power-per-node",
			"value": 30
		}, {
			"type": "max-power-per-port",
			"value": 10
		}, {
			"type": "min-guaranteed-bw",
			"value": 500
		}]
	}
}
```

Response example:

```
{
	"net-path": {
```

```
                "net-path-id": "0001"
        }
}
```


Create_VI	POST /emma/vi-mgt/tenant/{tenantID}/vi	202 – Accepted 400 – Bad Request 401 – Unauthorized 500 – Internal Server Error

Request example:


```
{
        "vi": {
                "vi-name": "Sample-VI",
                "vi-element": [{
                        "element-id": "vbr01",
                        "type": "virtual-bridge",
                        "name": "virtual-bridge-01"
                }, {
                        "element-id": "vif01",
                        "type": "virtual-interface",
                        "name": "virtual-interface-01",
                        "parent": "vbr01"
                }, {
                        "element-id": "vif02",
                        "type": "virtual-interface",
                        "name": "virtual-interface-02",
                        "parent": "vbr01"
                }],
                "objective-function": "minimize-active-phy-node",
                "metric": [{
                        "type": "max-power-per-node",
                        "value": 30
```

```
                }, {
                        "type": "max-total-power ",
                        "value": 100
                }]
        }
}
```


Response example:


```
{
        "vi": {
                "vi-id": "0001"
        }
}
```


Add_energy_policy	POST /emma/policy-mgt/tenant/{tenantID}/vi/{vi-id}/policy POST /emma/policy-mgt/net-path/{net-path-id}/policy	202 – Accepted 400 – Bad Request 401 – Unauthorized 404 – Not found 500 – Internal Server Error

Request example:


```
{
        "policy": {
                "policy-criteria": [{
                        "monitoring-parameter": "power",
                        "criterion": ">=",
                        "threshold": 120
                }],
                "policy-action": [{
                        "action-type": "re-plan net-path",
                        "action-parameter": [{
                                "metric": [{
                                        "type": "max-total-power",
```

"value": 100
}]
}]
}]
}
}

Response example:

{
"policy": {
"policy-id": "POL_0001"
}
}

Remove_energy_policy	DELETE /emma/policy-mgt/tenant/{tenantID}/vi/{vi-id}/policy/{policy-id} DELETE /emma/policy-mgt/net-path/{net-path-id}/policy/{policy-id}	200 – OK 401 – Unauthorized 404 – Not found 500 – Internal Server Error
Change_policy_status	PUT /emma/policy-mgt/tenant/{tenantID}/vi/{vi-id}/policy/{policy-id} /status/{policy-status} PUT /emma/policy-mgt/net-path/{net-path-id}/policy/{policy-id}/status/{policy-status}	200 – OK 401 – Unauthorized 404 – Not found 500 – Internal Server Error
Retrieve_policy	GET /emma/policy-mgt/tenant/{tenantID}/vi/{vi-id}/policy/{policy-id} GET /emma/policy-mgt/net-path/{net-path-id}/policy/{policy-id}	200 – OK 401 – Unauthorized 404 – Not found 500 – Internal Server Error

Response example:

{
"policy": {

```
                "policy-id": "POL_0001",
                "policy-target-type": "net-path",
                "policy-target-id": "0001",
                "policy-status": "DISABLED",
                "policy-criteria": [{
                        "monitoring-parameter": "power",
                        "criterion": ">=",
                        "threshold": 120
                }],
                "policy-action": [{
                        "action-type": "re-plan net-path",
                        "action-parameter": [{
                                "metric": [{
                                        "type": "max-total-power",
                                        "value": 100
                                }]
                        }]
                }]
        }
}
```

9 Appendix II: Implementation and development of 5G-Crosshaul applications

9.1 Multi-Tenancy Application

The implementation of the MTA is mainly focused on developing algorithms and simulative approach for validation /evaluation in this project, while a Proof of Concept (PoC) is not planned at this moment.

9.1.1 Functionalities

As described in D1.1, 5G-Crosshaul addresses two main Multi-tenancy services: i) the deployment of Network Services (NS) as defined by the ETSI NFV architecture and ii) the deployment of Virtual Infrastructure (VI), which is composed of virtual links, virtual nodes and virtual hosts (in other words, virtual hosts interconnected by network slices). **In WP4 we are focused on the deployment of Virtual Infrastructure service for the implementation of Multi-Tenancy Application (MTA),** given that the first service will be offered by the XCI (more specifically by the NFVO and VIM components).

In this work we focus on implementing two MTA functionalities: (i) slicing the infrastructure resources for different tenants given a physical infrastructure; and (ii) admission control to accept or reject tenant requests to ensure the SLA of the different tenants and avoid overloading the system.

- **Tenant resource (de)allocation decision function:** MTA decides the allocation/modification/deallocation of resources for virtual infrastructures, which are isolated per tenant upon their requests. This is done by solving the Virtual Network Embedding (VNE) problem. VNE takes care of the allocation of virtual resources both in nodes (XFEs, XPUs) and links. A dynamic mapping of virtual resources onto a physical hardware for multiple tenants should lead to maximize the utilization of the physical network, compute and storage infrastructure resources as well as maximize the energy efficiency by using least number of nodes or links to support all tenants.
- **Admission control function:** MTA should be able to accept or reject VNO requests that may lead to violating an SLA or a waste of hardware resources.

As a simplified information model, a virtual infrastructure allocation request encompasses:

- The virtual infrastructure nodes (i.e., virtual network nodes)
  - (typically abstracted into the basic functions such as switching or forwarding)
- The virtual infrastructure computing nodes (i.e., virtual hosts)
  - Virtual compute and storage nodes are allocated to tenants. From a physical point of view, the hardware resources for storage and CPU will be located in a subset of XPU nodes.
  - Storage and CPU is virtualized. It can correspond to 1:1 to physical resources or distributed across multiple physical disks.
- The virtual infrastructure links (also referred to as virtual connections).
  - Let us note that links of this virtual infrastructure do not necessarily correspond to a (physical) underlying infrastructure connection in a 1:1 mapping.
  - The virtual infrastructure link can be mapped to a set of physical links (also following different paths) provided the set of physical links will satisfy the QoS parameters required the virtual link. This can be done by applying state-of-the-art routing and/or traffic engineering algorithms. For example, virtual infrastructure nodes may be interconnected by multiple virtual links each characterized with

different level of availability. High availability may be supported within the physical layer e.g. with a 1+1 protection. With lower availability other resilience mechanisms can be considered (e.g. shared protection or restoration on the fly).

- Bandwidth management:
  - Each virtual infrastructure link is associated to an e.g., committed bandwidth (CIR, Committed Information Rate, reserved to it, even if it does not use) and an extended bandwidth (EIR, Extended Information Rate, shared with some other tenants).
  - It is the responsibility of the XCI to monitor and enforce allocated bandwidth use e.g. at the ingress of each virtual link.

9.1.2 High-level design

9.1.2.1 General Multi-Tenant Scenario

Figure 40 shows the high-level design of the MTA application for the general cases, including the design of the MTA and its interaction with the XCI via the XCI NBI. The main components are described in Table 37. As mentioned above, in this project we only focus on the deployment of virtual infrastructure service, and more specifically on developing the optimization algorithms for allocating virtual resource for the individual tenants in the Resource Slicing Optimizer. The implementation will be done in a simulation environment, while a Proof of Concept (PoC) is not planned at this moment.

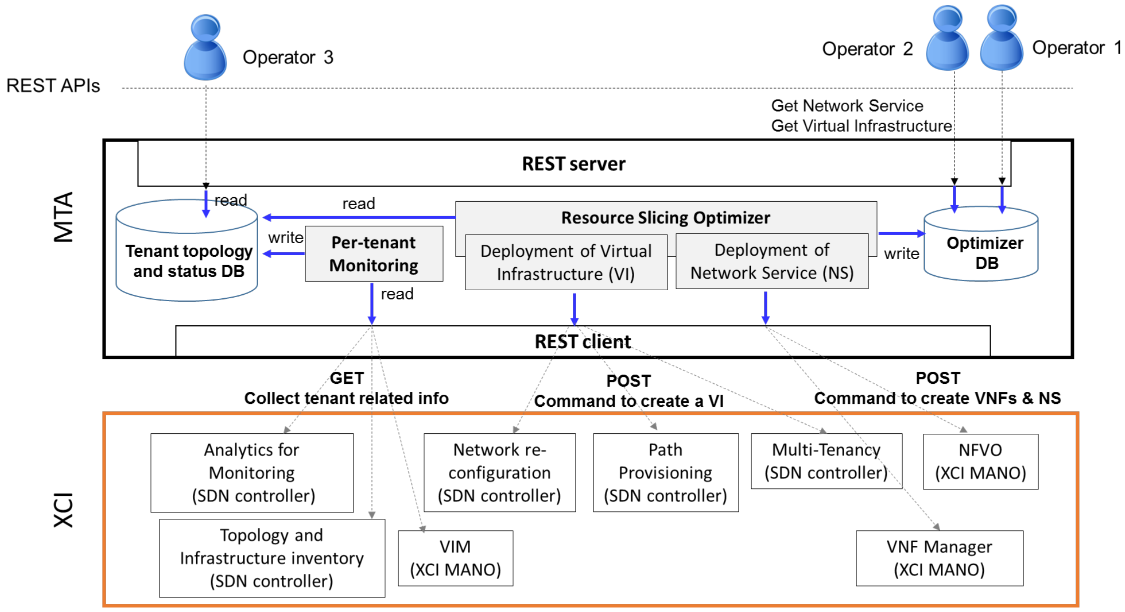


Figure 40: MTA high level design for general Multi-Tenant scenario

Table 37: MTA application components

Component	Description
REST Client	Entity in charge of interacting with XCI components via REST APIs.
Per-tenant Monitoring	Entity in charge of collecting tenant related information such as network topology, infrastructure inventory, network status, QoS and/or QoE performances, etc.

Optimizer DB	Database storing the decisions on the virtual infrastructure and topology owned by each tenant and their allocated resources including network storage and computing resources; the location of virtual functions and network services and their required computing resources computed by the *Resource Slicing Optimizer*.
Tenant topology and status DB	Database storing tenant information including topology, infrastructure inventory, network status, tenant service performances, etc.
Resource Slicing Optimizer	Entity in charge of creation of virtual infrastructure and network services. It computes the optimum allocation of virtual resources for the individual tenants by slicing the underlying physical resources. It also involves proper admission control to accept or reject tenant requests that may lead to violating an SLA or a waste of hardware resources. It interacts with the XCI components to command the configuration and request the provisioning of the computed virtual infrastructure and network services.
REST Server	Entity in charge of implementing the north bound REST API of the MTA application for retrieving per tenant infrastructure and monitoring data, and optimization actions on creation of a virtual infrastructure or a network service.

9.1.2.2 Simulation environment of High-speed train scenario

The MTA for High-Speed Train (HST) scenario consists of two parts: On-board MTA and On-land MTA. On-board MTA is applied inside the high-speed train to handle packets from different network operators and is composed of two modules: namely, *Packet Inspection module* and *Load Balancing module,* shown in
Figure 41.

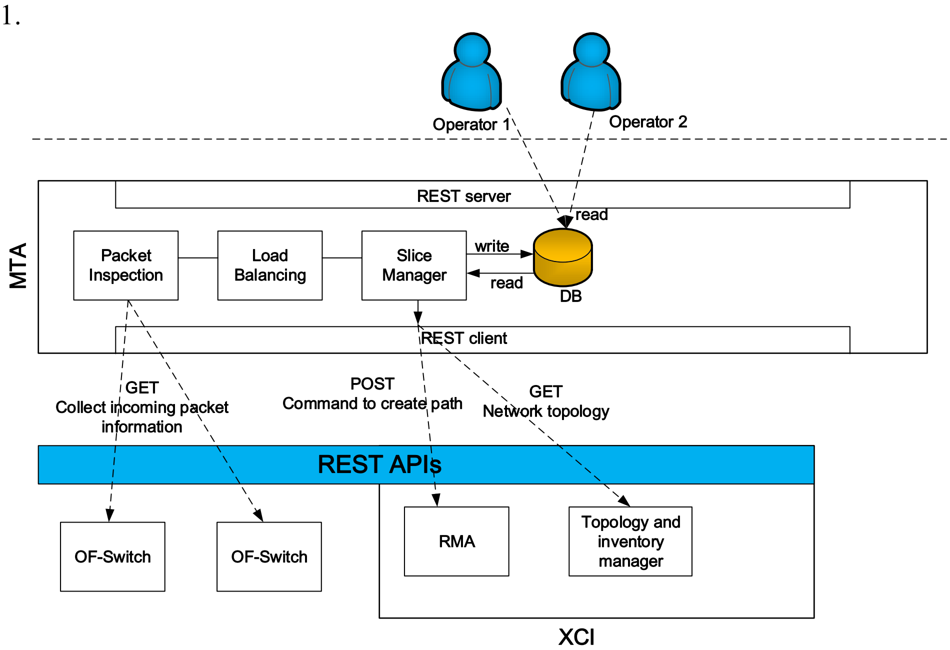


Figure 41: MTA high level design for high-speed train scenario

Then, On-land MTA is applied in a transport network shared by different network operators and is to act as slice manager to handle the slices for different network operators. Each slice is dedicated to a network operator when this operator requests to on-land MTA cooperating with VIMAP application to create/modify/delete slices in the shared transport network.

9.1.3 Implementation roadmap

General Scenario

Although the design of MTA in Figure 40 shows all important components, in this work the implementation is specifically focused on developing algorithms for the Resource Slicing Optimizer on the allocation of virtual resource to the individual tenants and admission control function. According to this scope, our roadmap for implementation is presented in Figure 42.

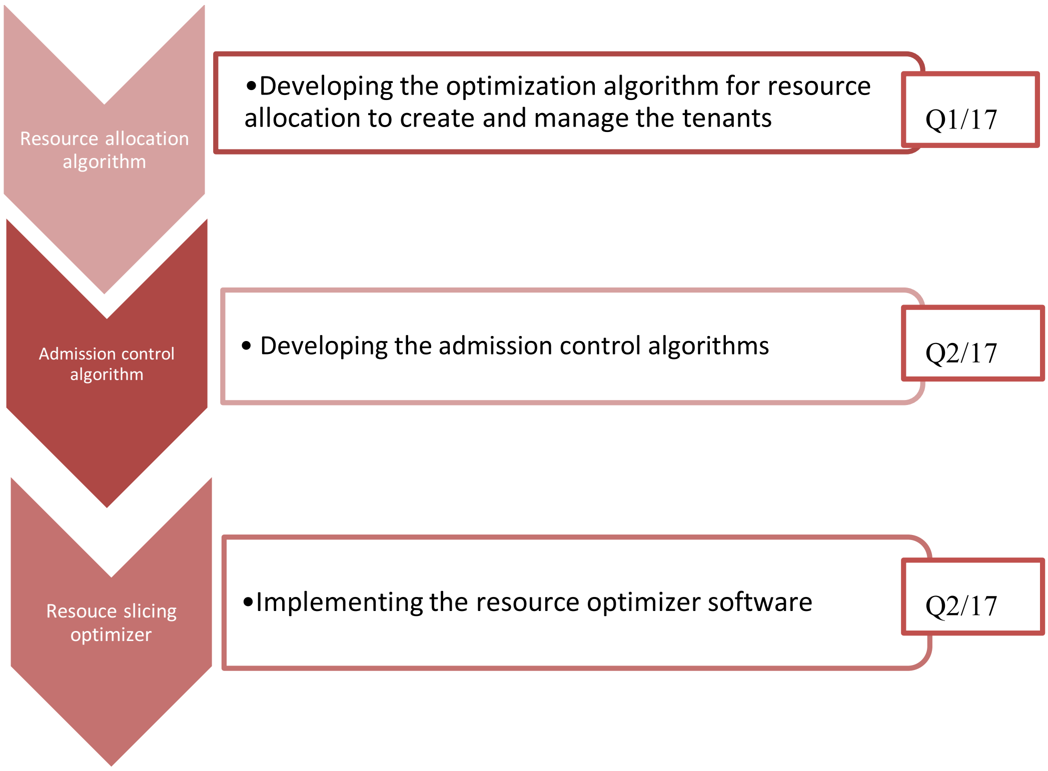


Figure 42: MTA roadmap for the general scenario

High-Speed Train scenario

The implementation roadmap of the MTA for High-Speed Train scenario is shown in Figure 43.

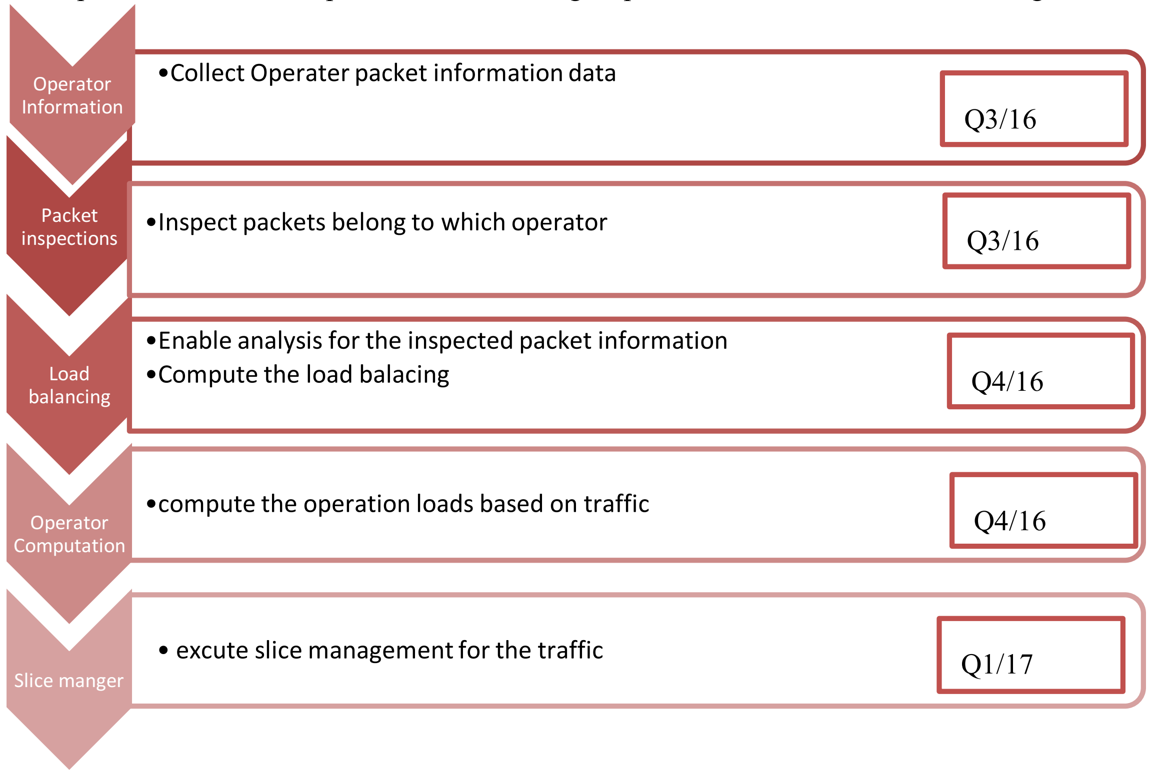


Figure 43: MTA roadmap for the High-Speed Train Scenario

9.2 Mobility Management Application

The MMA will implement the following functionalities:

1) The implementation will be formed of interconnected OpenFlow Ethernet switches with a WLAN interface working as PoA. When a user connects to the PoA it generates LLC packets, and the LLC packets will generate packet-in to the controller, this is how the controller is notified of the presence of a new user in the 5G-Crosshaul domain.
2) The forwarding will be based on a direct modification of flow tables at the 5G-Crosshaul Forwarding Element (XFE) using standard OpenFlow. The 5G-Crosshaul mobility will be based on a flat IPv6 network, on which traffic is forwarded to the nearer Point of Connection (PoC) to the Internet, to make it possible the MMA will handle the association of the Gateways (GW) to the users in addition to the Neighbor Discovery mechanisms. The assignment of the PoC to the GW will be made based on heuristic and proximity.
3) Depending on how the RMA executes the allocation of paths the mobility can be handled in different manners:
    a. Based on VLANs from PoCs to XPUs through the network, handling the user-VLAN mapping.
    b. Based on IP hop by hop.
    c. Based on MAC hop by hop.

- The GWs and the CDN nodes will be instantiated at the 5G-Crosshaul Processing Units (XPUs), the number of GW and CDN nodes will depend on the contractual agreements between tenants, but typically the GW and the CDN node will be instantiated in each XPU. The notification to the CDNMA will be made through the MMA REST API. In the case of a fixed user like in Figure 44 the GW is necessary because the PoC switch may not be SDN enabled, if we don't do that the IP stack may not work.

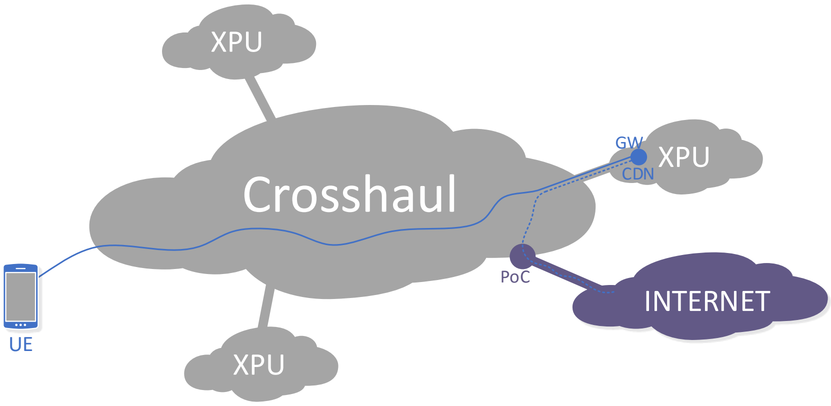


Figure 44: Fixed user

There may be the possibility of having the CDN Node in a different XPU of the PoC like in Figure 45.

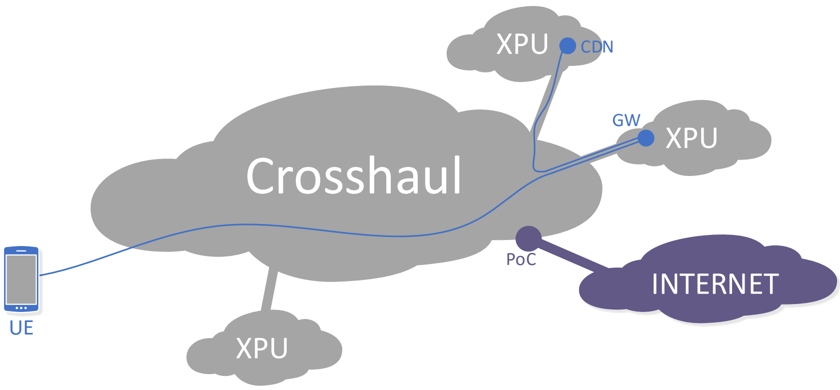


Figure 45: CDN node not located in the PoC XPU

The traffic of the user goes through the GW to the PoC/CDN node. When the user enters in a 5G-Crosshaul domain and attaches to a GW it is assigned a prefix, in a mobility scenario there will be several prefixes (one for each GW). The MMA can handle the old prefix in two different ways:

- All the traffic addressed to the previous prefix/GW will be tunneled to the new prefix/GW.

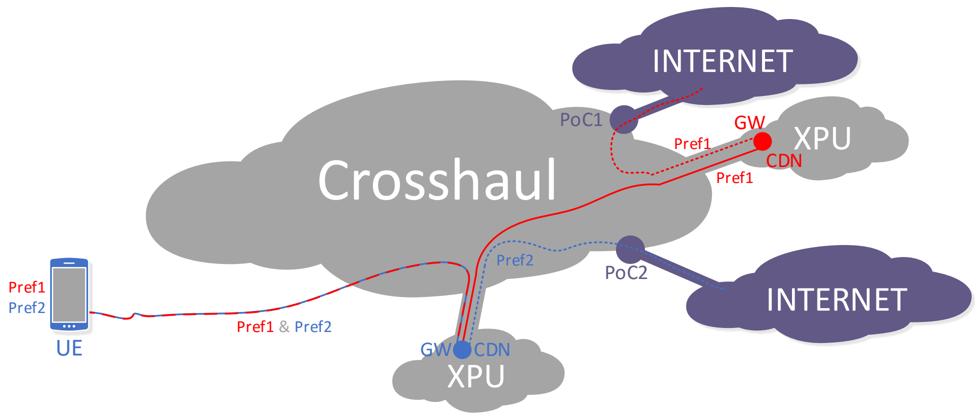


Figure 46: Mobile user scenario with GW tunnel

The path to the previous GW is maintained, so, the new and the previous GW work independently with their respective paths.

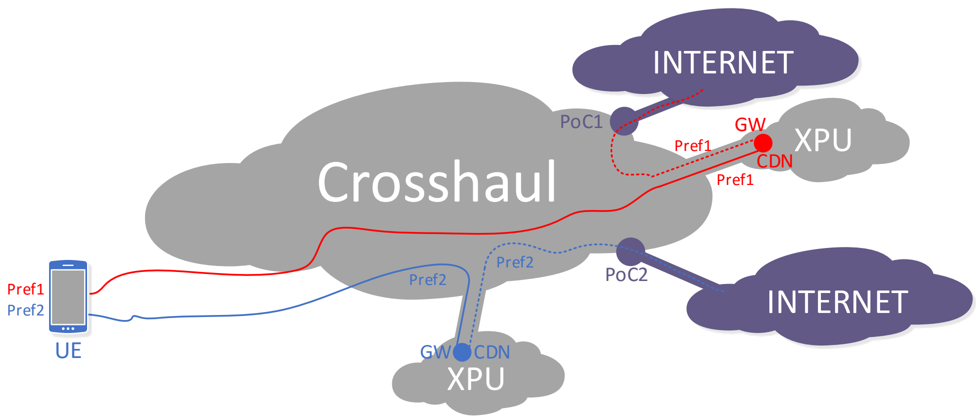


Figure 47: Mobility scenario with previous path maintenance

In both mobility scenarios if a change of CDN node is necessary the MMA will notify the CDNMA through the MMA REST API to instantiate a new CDN node nearer to the new GW.

9.2.1 High-level design

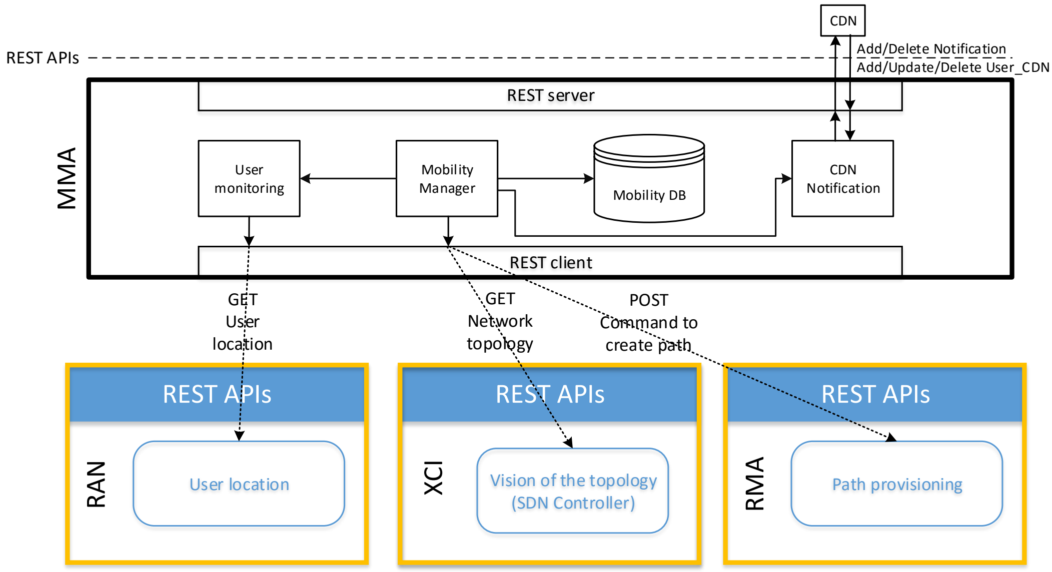


Figure 48: MMA high level design

9.2.2 Implementation roadmap

The implementation roadmap of the MMA is shown in Figure 25

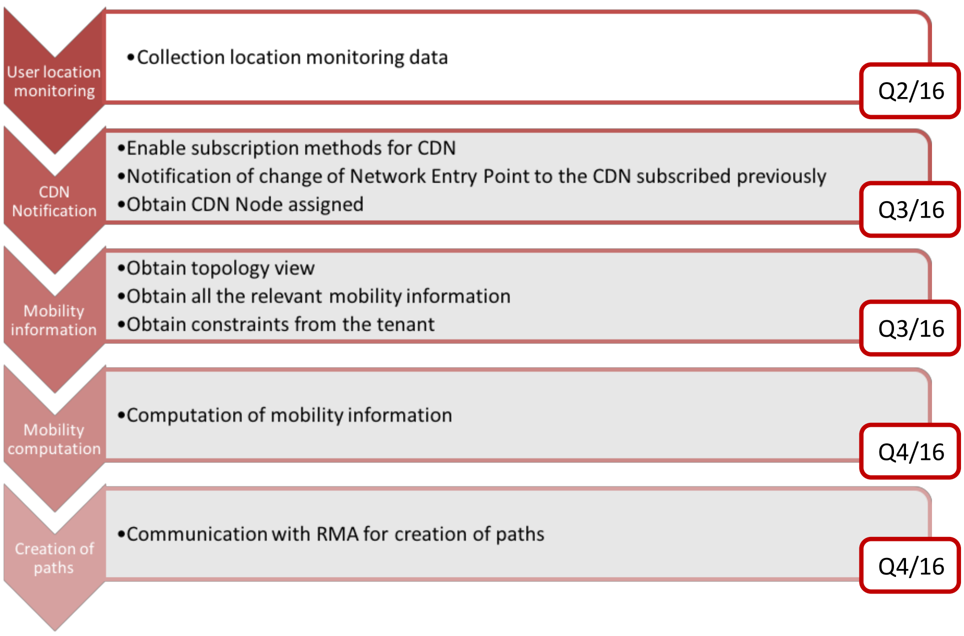


Figure 49: MMA roadmap

9.2.3 Current implementation

The current MMA implementation has the following functionalities:

4) User detection: the implementation will be formed of interconnected OpenFlow Ethernet switches with a WLAN interface working as PoA. When a user connects to the PoA it generates LLC packets, and the LLC packets will generate packet-in to the MMA controller (Ryu controller), this is how the controller is notified of the presence of a new user in the 5G-Crosshaul domain.
5) Get user information: from the LLC header received in the packet-in we can obtain the MAC address of the user and the IP address of the switch which is working as PoA. These information is stored in the controller.
6) IP address assignment: once the user is connected to the PoA it needs an IP address, it will be assigned with DHCP. The controller receives the DHCP discoveries and DHCP requests of the user as packet-in messages from the PoA and it responds these messages in packet-out messages. The IP address assigned to the MAC address is stored making the implementation persistent.
7) REST API: The REST API implemented in the MMA controller is focused on managing the CDN subscription for the notification of new users in the 5G-Crosshaul. The CDNMA can subscribe to the MMA sending a post with the notification URL, the notification can be edited and deleted. Once the CDNMA is subscribed, when a new user connects to the 5G-Crosshaul domain the MMA controller sends a POST to the URL with the user MAC address and the PoA IP address, the CDNMA will respond this post with the CDN node assigned.

9.3 Resource Management Application

9.3.1 Functionalities

In general, RMA is the application which is deputed to manage network resources in conjunction with storage and compute availability in the XPU nodes. Referring to the context of 5G-Crosshaul, the networked entities which are taken into consideration in this section are XPFEs. Therefore, RMA shall provide different specialized services which are necessary to manage the resources available the 5G-Crosshaul network at a given time and geographical location. RMA shall also manage the different transmission technologies which are part of the Crosshaul data plane. Indeed, different utilisation can be based on network load and type of available technologies (optical fibre, mm-Wave, copper, etc.) composing the intelligent transport network. To carry on with these tasks, RMA needs to acquire different type of information from the XCI interfaces in order to perform different services which can be requested to this application. Services are computed and the output then passed to the requesting entity.

The RMA functions will carry out the following tasks:

- Optimization of network path provisioning according to a cost function (e.g. maximise network resource utilization, minimize the cost, etc.). The RMA functions will consider the heterogeneous network resources deployed at the XFEs. Please consider that this requires the a-priori knowledge of the endpoints of the flow (e.g. the XPUs or the XFEs) by the application/client requesting the service. An example of application that may request this service is the MMA. In the case of an OTT application (e.g. TVBA, CDNMA), the input of the RMA function could be the position of the virtual machines hosting the VNFs that should be interconnected, in order to ensure that the computed path is constrained to such placement.

- Optimization of VNFs and network paths allocation according to a cost function, given the VNF forwarding graph provided by the application/client requesting the service. Examples of potential client applications include the TVBA and the CDNMA. Likely to the previous case, the RMA functions will perform the computation considering the multiple data plane technologies of the 5G-Crosshaul.

9.3.2 High-level design

The RMA functions can be implemented within the different modules of the 5G-Crosshaul architecture or in a dedicated application (i.e., the RMA). Figure 26 shows the high-level design in the latter case. In the former case, the RMA functions will gather the relevant information to perform the computation from the hosting module.

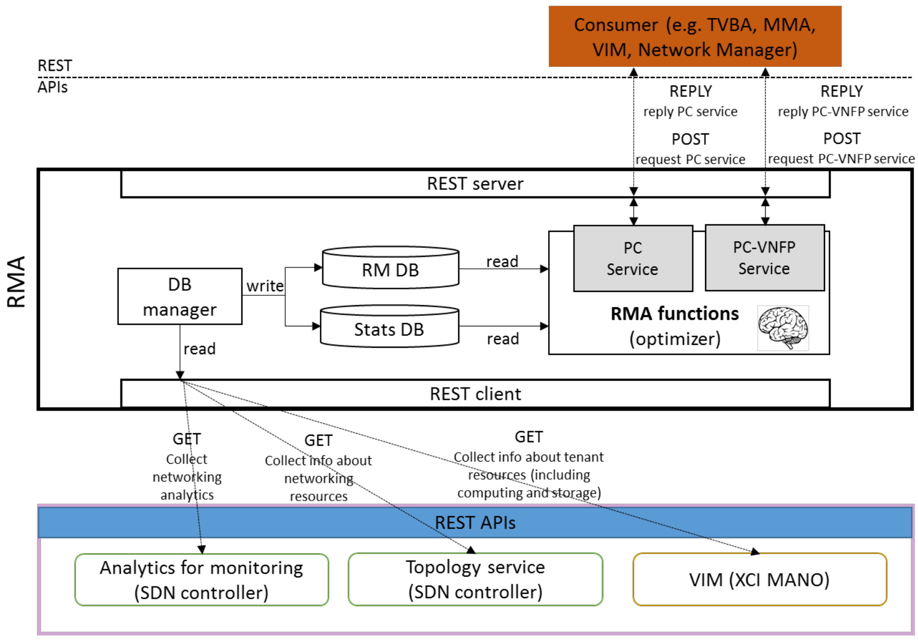


Figure 50: RMA application high-level design

The main internal components are described in the table below.

Table 38: RMA application components

Component	Description
REST Client	Entity in charge of interacting with the XCI components via REST APIs.
DB manager	Entity in charge collecting the relevant information that is needed for the resource management and optimization. Such information is collected from the various XCI modules (i.e., tenants' resources, including computation and storage, from VIM, network resources from SDN controller). The DB manager process this information and stores it in the databases.

RM (Resource Manager) DB	Database that stores the inventory of 5G-Crosshaul elements (XFEs, XPUs), the location of resources and their status, the list of existing services mapped to the resources they are using.
Stats DB	Optional database that stores the statistics related to the active services and the resources they are using.
RMA functions	The brain of the RMA application, i.e., it contains the algorithms that compute the optimum allocation of 5G-Crosshaul resources, leveraging the information stored in the Resource and Stats DBs. Currently, the RMA is conceived as a passive stateful element, which answers to query issued by other applications (e.g. MMA, CDNMA, TVBA) or XCI modules (e.g. VIM) via a REST interface. Two main services are foreseen at the moment: • Path Computation (PC) service: this service computes the allocation of network resources (given the endpoints of the network connection or the placement of the VNFs to interconnect); • Path Computation and Virtual Network Function Placement (PC-VNFP) service: this service computes the allocation of network, storage and computing resources (given the VNF forwarding graph)
REST server	Entity in charge of implementing the north bound REST API of the RMA application to allow other applications or XCI modules to request the RMA services.

9.3.3 Implementation roadmap

RMA algorithm for PC-VNFP service

Figure 51 below shows the roadmap for implementing the RMA algorithm responsible to deliver the PC-VNFP service.

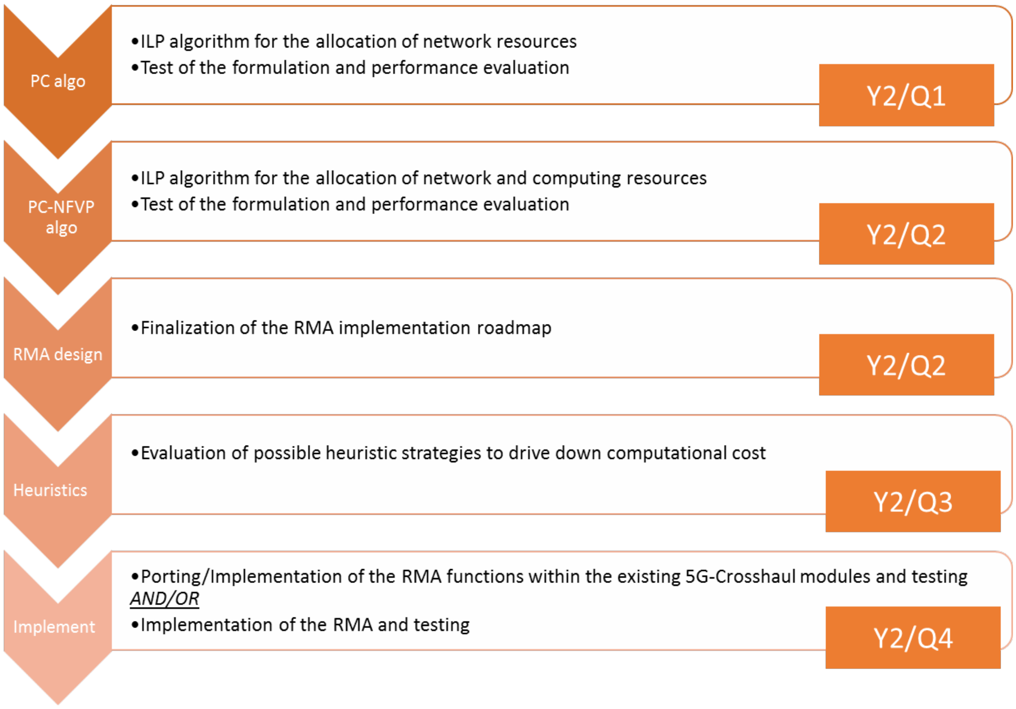


Figure 51: RMA implementation roadmap

9.4 Virtual Infrastructure Manager and Planner

The main objective of the implementation of a VIMaP prototype is the testing and validation of the component functional requirements, that it, it is able to allocate, dynamically and upon client request, a set of interconnected virtual machines (VMs) interconnected in an arbitrary and user-defined mesh. The VIMaP XCI component stays on top of a storage & computing controller and an SDN network controller, while providing a northbound API (NBI) to applications. The storage & computing controller is implemented in terms of a single OpenStack deployment, which controls several compute nodes (XPUs in terms of the 5G-Crosshaul infrastructure) distributed in different geographical locations within a single 5G-Crosshaul domain (NFVI administrative domain). The main consumed OpenStack services and components are Keystone (for identity management), Nova (for management of computing resources), Glance (for software images) and Neutron (for specific networking services).

An implementation of the VIMaP is ongoing, as part of the 5G-Crosshaul XCI, the details of the development of the component are provided in [2] for completeness and coherence. The main objectives of the planned tests to be carried out are the following:

- Functional validation and experimental assessment of the VIMaP NBI interface
- Functional validation of the interaction with the cloud controller (OpenStack)
- Functional validation of the interaction with the SDN controller
- Functional validation of the P-interface The P-interface will be tested to see if the P component (provided by a third party) is able to execute the placement function independently.

These main objectives involve the following features and functionality (only considering the VIMaP component)

- The ability to query, from the OpenStack cloud controller, the status of the compute nodes, and their capabilities

- The ability to upload images into the cloud controller, which will be launched when a VM instance is allocated.
- The ability to launch Virtual Machines with a set of associated resources
- The ability to query, from the SDN controller, the topology of the transport network, which needs to be augmented with the location and capabilities of the XPU
- The ability to query, from the SDN controller, the status of existing and/or configured flows, and their properties and attributes.
- The ability to provision, invoking the correct API from the SDN controller
- Finally, the ability to instantiate and terminate interconnected overlay VMs as defined by the VIMaP NBI.

As mentioned therein, the main tests that are considered are the following, and in line with the services that are exported by the VIMaP:

- Tests for topology detection
- Tests for XPU status and capability discovery
- Instantiation of Virtual Machines in remote locations
- Flow configurations across single- and multi-domain networks
- Functional assessment of the VIMaP NBI and, finally
- Enable external placement computation of VMs and flows via a P-component that can be implemented by a third party.

The VIMaP MANO component will be part of the targeted demonstrations in WP5. Below, a simplified version of the VIMaP roadmap, as detailed in [2]

Roadmap		
Features	Description	Timeline
VIMaP alpha release	Initial VIMaP Prototype Use of OpenStack keystone, nova, glance APIs programmatically. First integration with the SDN controller using the COP protocol, query topology, details about links and nodes and extended topology with location and capabilities of XPU within the testbed.	Q4 2016
VIMaP beta release	Ability to perform path computation and flow provisioning using the COP protocol and API, as exported by the SDN controller. Validation by means of dummy SDN controllers or simple networks. NBI REST API framework and basic messages.	Q1-2017
VIMaP release 1.0 core features	Able to request and release a VIMaP service using its REST based NBI.	Q2-2017

VIMaP release 1.0 enhanced features	Basic Graphical User Interface Proof-of-concept using the P-interface with a third party module.	Q3-2017
VIMaP final release	Final VIMaP prototype. Integration tests pass, Integration with ABNO SDN orchestrator over a multi-domain wireless-packet-optical network.	Q4-2017

9.5 Energy Management and Monitoring Application

9.5.1 Functionalities

The EMMA application is being developed as a Java application implemented from scratch and operating on top of the 5G-Crosshaul XCI services, invoked by REST APIs. It will implement the following main functionalities:

- Correlation of energy-related information for network and IT domains, as exposed by the XCI, to provide summarized energy consumption data to the administrator of the physical infrastructure. Based on this kind of information the EMMA will also adjust the status of the devices (e.g. putting nodes in sleeping mode).
- Optimization of network path provisioning regarding power consumption across the end-to-end path and possible re-planning of already established network paths for power consumption re-optimization (based on policies). As an example, the application will target a specific network technology (i.e. 5G-Crosshaul XPFEs).
- Optimization of VNFs and network paths allocation in network services, still minimizing the whole power consumption while guaranteeing the compliance with the service specification, e.g. regarding disjoint VNFs placement. The implementation will target a specific type of Network Service (i.e. the vEPC) to show a concrete use case.

As such, the application will expose towards the user (e.g. the administrator of the physical infrastructure) a set of REST API for operation and management actions.

9.5.2 High-level design

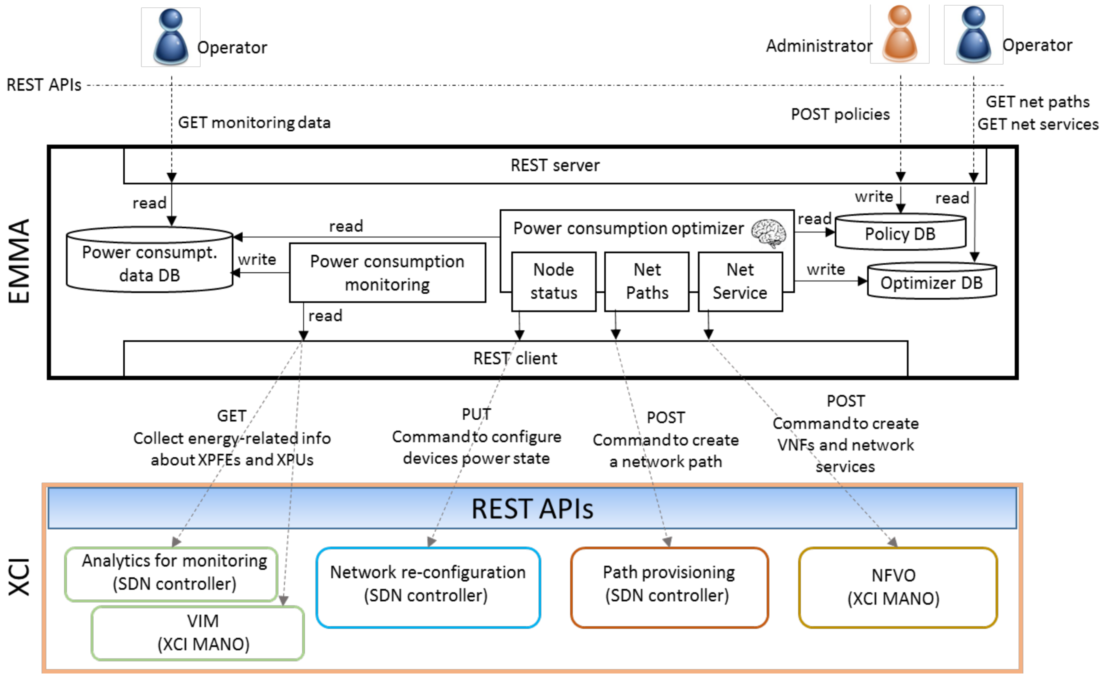


Figure 52: EMMA high-level software design

Figure 52 shows the high-level design of the EMMA application, where the main internal components are described in the table below.

Table 39: EMMA application components

Component	Description
REST Client	Entity in charge of interacting with the XCI components via REST APIs.
Power consumption monitoring	Entity in charge of coordinating the collection of power consumption information from XCI components (i.e. VIM for XPUs and SDN controller for XPFEs), processing and aggregating this information and storing and organizing the results in the internal *Power consumption data DB*.
Power consumption data DB	Internal no-SQL DB to store power consumption data, related to single XPUs or XPFEs, end-to-end network paths, VNFs and Network Services, or physical domains at the physical infrastructure level.
Power consumption optimizer	Entity in charge of computing the optimum allocation and status of physical resources, based on the information stored in the *Power consumption data DB* and the policies configured in the *Policy DB*. It acts at three different levels: • Regulation of power status of single nodes • Computation of energy efficient network paths • Computation of energy efficient VNF placement for Network

	Services. It also interacts with the XCI components to change the configuration of power status in selected infrastructure nodes (both network nodes and servers) and request the provisioning of the computed network path and Network Services.
Policy DB	SQL database storing the policies configured by the administrator.
Optimizer DB	SQL database storing the device status, network paths and Network Services computed by the *Power Consumption Optimizer*.
REST Server	Entity in charge of implementing the north bound REST API of the EMMA application for retrieving power consumption monitoring data and power optimization actions as well as for configuring energy management policies.

9.5.3 Implementation roadmap

We start by outlining the validation roadmap that was followed to validate the algorithms for the Power Consumption Optimizer in EMMA, remarking that it can be implemented in either a proactive or a reactive manner. In proactive manner, the system can monitor the network periodically and determine changes in flow allocation if convenient. For the evaluation of the PCO algorithms, we adhered to the following roadmap:

- we implemented the algorithms following a proactive approach;
- the algorithms were tested using the Mininet SDN network emulation environment as a forwarding plane and a controller whose functions are supported by the Open Network Operating System (ONOS);
- the PCO algorithms were then implemented as an application on top of ONOS.

Also, we remark that the performance of our algorithms was also compared to the optimum in some small-scale scenarios.

The PCO leverages the PacketProcessor interface of ONOS, in order to process packets, i.e., the processing function in the PacketProcessor interface of ONOS has been replicated within the PCO. This way the PCO can distinguish between traffic and control packets, as well as between unicast and multicast packets, thus further processing only unicast traffic packets. The routing algorithm is developed by leveraging ONOS built-in Reactive Forwarding application, which uses the shortest path to route incoming traffic. However, the topology considered while searching for the shortest path is not always the whole topology. The path selection method first tries to find a path through the links that are already carrying traffic so as to concentrate all traffic in the infrastructure which is just enough to route the existing traffic. If no path is found using the active links, we consider the whole network to find an efficient, feasible flow path.

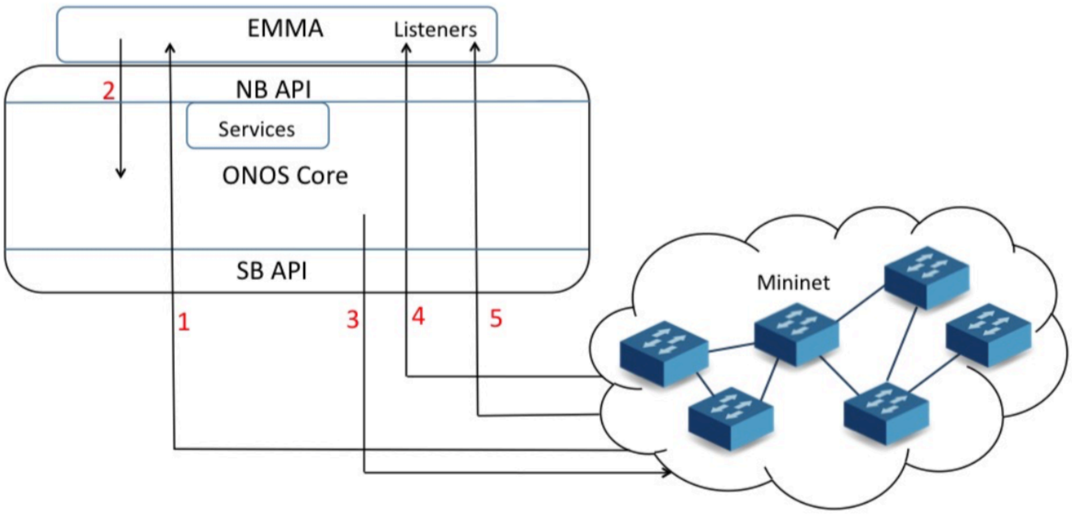


Figure 53: EMMA PCO implementation and emulation using ONOS and Mininet

Figure 53 highlights the main components and chain of events in the implemented application. The description of the set of actions taken by the PCO in EMMA and the set of events happening in the network are presented below. We assume that initially ONOS provides the PCO with the whole network topology. Specifically, the PCO gets the network topology via the TopologyService application of ONOS, to which the PCO has registered. The EMMA PCO can then use the network topology for path computation and flow provisioning.

1. Whenever a switch has to handle a packet for which it does not have a forwarding rule, it sends the packet directly to ONOS. The PCO intercepts the packet using the requestPacket method of the ONOS PacketService interface.
2. The PCO processes the packet and, if it is a unicast packet, it computes a flow path according to Algorithm 1. The PCO then creates a forwarding rule. In order to do so, it uses the application of ONOS, called TrafficSelector.builder(), so as to specify the pair of source and destination IP addresses to be matched when processing the packet at the switch. Then the PCO uses the TrafficTreatment.builder() application of ONOS so as to express a set of instructions, namely, forwarding the packet toward the intended output port and decrementing the packet TTL. Finally, the PCO uses the ONOS ForwardingObjective.Builder() application to create the fowarding rule based on the TrafficSelector and TrafficTreatment outputs.
3. The PCO uses the ONOS application FlowObjectiveService to send the rule and install it in the switch.
4. A FLOW ADDED event will be generated by the switch after the flow rule has been installed. Following the blueprint of the *FlowListener* interface of ONOS, we built a *FlowListener* interface within EMMA. Such an interface allows the PCO to detect a FLOW ADDED event, after which it starts accounting for the power consumption due to the newly added flow.
5. A FLOW REMOVED event will be generated by a switch whenever a flow finishes, or the application removes a flow from a switch when a better path is found. As for the FLOW ADDED event, the PCO can detect a flow removal through the *FlowListener* interface we developed, hence, to correctly compute the energy consumption of the active network. Note that when a FLOW REMOVED event corresponds to a flow

termination (i.e., it has not been triggered by a re-routing action), the PCO will check whether existing flows can be re-routed, according to Algorithm 2.

The packet size and the flow rate, which are needed for energy computation in EMMA, are obtained as the ratio of the number of bytes to the number of packets and of the number of packets to the flow duration, respectively. The number of packets, number of bytes and flow duration are variables are provided to the PCO by ONOS. Specifically, the PCO exploits the *getDevice* method of *DeviceService* in ONOS to get the list of available switches and the *getFlowEntries* method of the *FlowRuleService* of ONOS to get information (i.e., number of bytes, number of packets and flow duration) about each flow within a given switch.

The implementation roadmap of the EMMA application is shown in Figure 54. An initial prototype including monitoring functions only is expected for September 2016, while the complete prototype providing also energy management and optimization features is expected at the end of the second year of the project (i.e. June 2017).

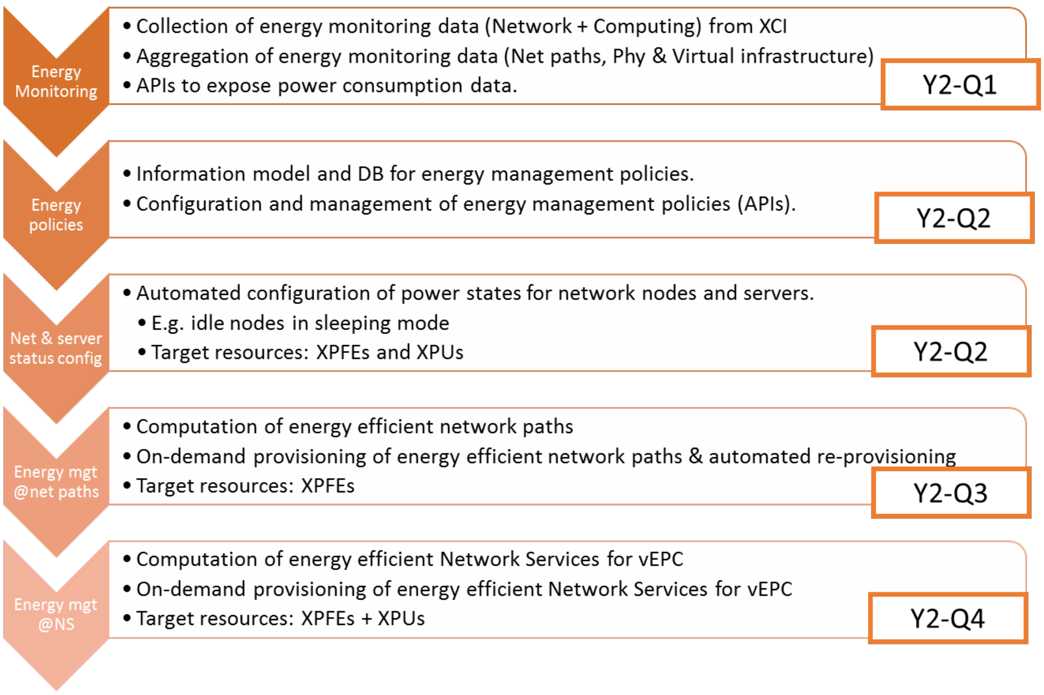


Figure 54: Roadmap for EMMA implementation

9.6 Content Delivery Network Management Application

9.6.1 Functionalities

The CDNMA will be a Web application programmed from scratch and operating on top of the 5G-Crosshaul XCI services. It will implement the following main functionalities:

- Request the CDN infrastructure instantiation to XCI NBI. Providing a complete descriptor file with detailed information about the CDN elements and how they have to be configured.
- Management and storage of specific monitoring information got from the CDN nodes and the network users.

- Management and update of the content delivery rules over the CDN nodes placed in the network, based on the monitoring information received and the logic defined by the CDN operator.

The application will show towards the operator (e.g. the CDN operator) a graphic interface running on a REST API for management and operation actions.

9.6.2 High-level design

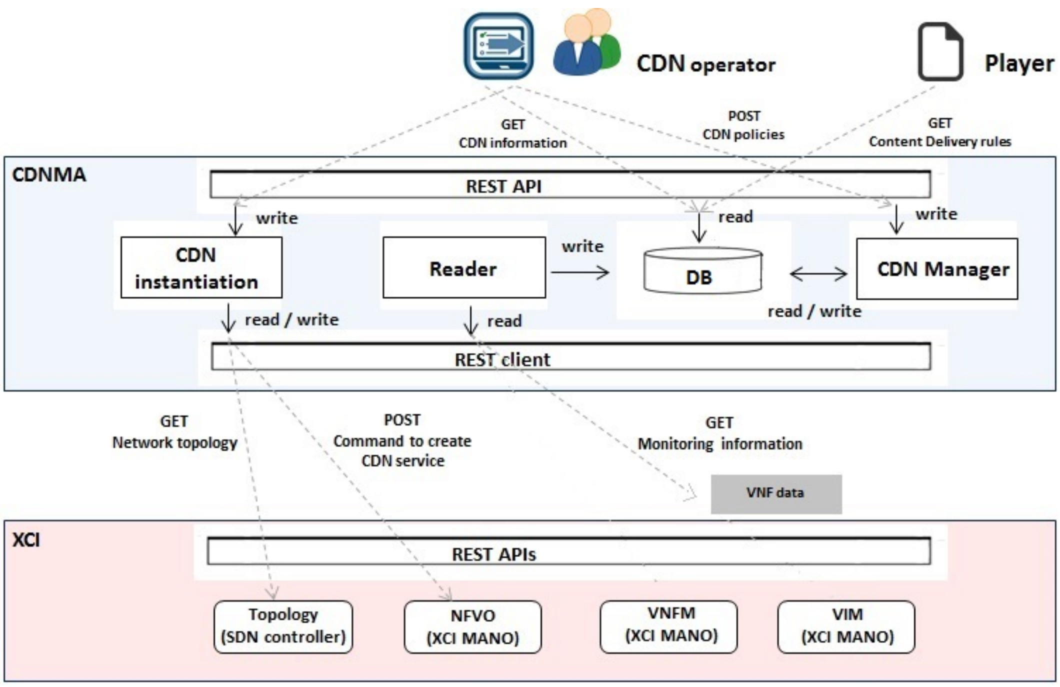


Figure 55: CDNMA high-level software design

The main internal components showed in Figure 55 are described in the table below.

Table 40: CDNMA application components

Component	**Description**
REST Client	Entity in charge of interacting with the XCI components via REST APIs.
CDN instantiation	Entity in charge of the CDN infrastructure instantiation. This entity will receive the CDN policies from the CDN operator regarding where to deploy the CDN nodes. It will also interact with XCI NBI to get the network topology and to request the creation of the CDN service through the REST APIs.
Reader	Entity in charge of getting the VNFs monitoring information.

DB	Internal no-SQL DB (MongoDB) to store the monitoring information and the content delivery rules.
CDN Manager	Entity in charge of deciding the content delivery rules, based on the CDN policies provided by the CDN operator and the monitoring information stored in DB.
Player	Video player that shows the video from the selected server.

9.6.3 Implementation roadmap

The main objective of the functional validation is to prove that the CDNMA application is able to instantiate upon client (CDN operator) request, a vCDN infrastructure based on the CDN operator criteria and the network infrastructure available. It will also be tested that it is able to control and manage the CDN service creating and implementing the content delivery rules. To that end, the main aspects that the tests aim are the following:

- Processing of client (CDN operator) request. The application will show towards the client a graphic interface running on a REST API for management and operation actions.
- Management of the network topology information got from the XCI NBI through a REST API.
- Request the vCDN infrastructure instantiation to XCI NBI. Providing a complete descriptor file with detailed information about the CDN elements and how they have to be configured.
- Management and storage of specific monitoring information got from the CDN nodes and the network users.
- Management and update of the content delivery rules over the CDN nodes placed in the network, based on the monitoring information received and the logic defined by the CDN operator.
- Performance of the web player (starting point). Video player that shows the video from the selected server.

The implementation roadmap of the CDNMA application is shown in Figure 56.

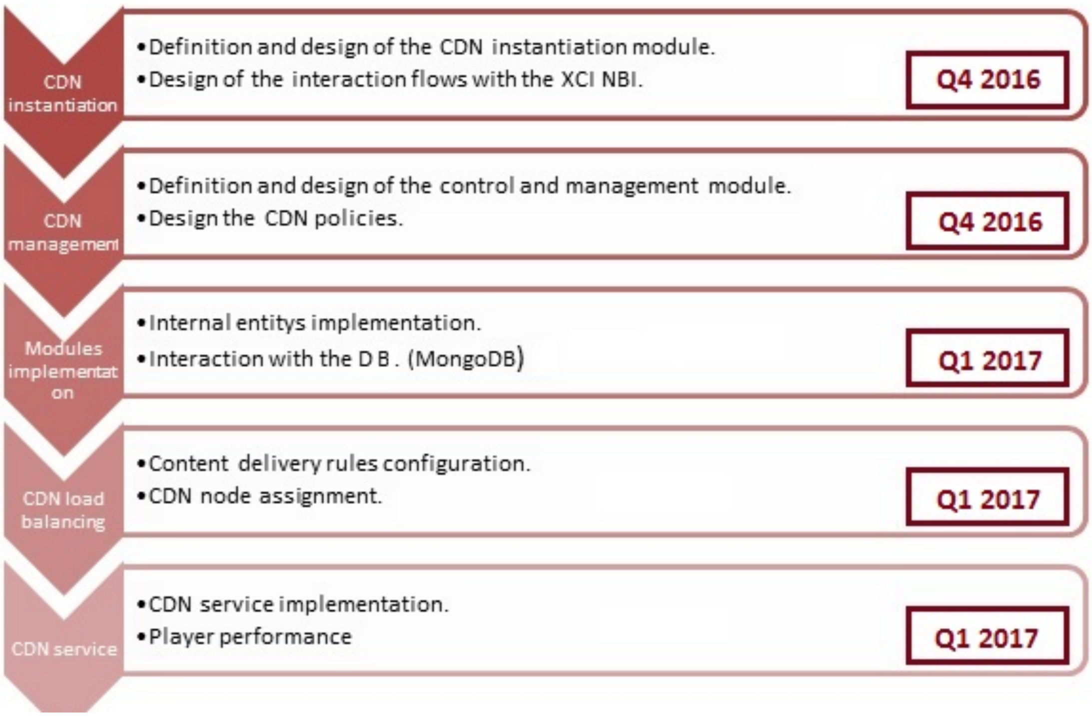


Figure 56: Roadmap for CDNMA implementation

9.7 TV Broadcasting Application

9.7.1 Functionalities

The TV Broadcast application (TVBA) belongs to the highest level of the application hierarchy of the 5G-Crosshaul running as an OTT service.

The TVBA will be an application programmed from scratch and operating on top of the 5G-Crosshaul XCI services. The application will show towards the operator a graphic interface running on a REST API for management, monitoring and operation actions.

It will implement the following main functionalities:

- Configure the video service from live streams or files and inject it into the network.
- Request the topology information to XCI NBI. Management and storage of specific monitoring information got from the SDN and the network users.
- Building the broadcast tree by means of algorithm calculation and delivery rules.
- Management and update of the content delivery rules over the switches placed in the network, based on the monitoring information received and the logic defined.
- QoS monitoring and content quality assessment.

9.7.2 High-level design

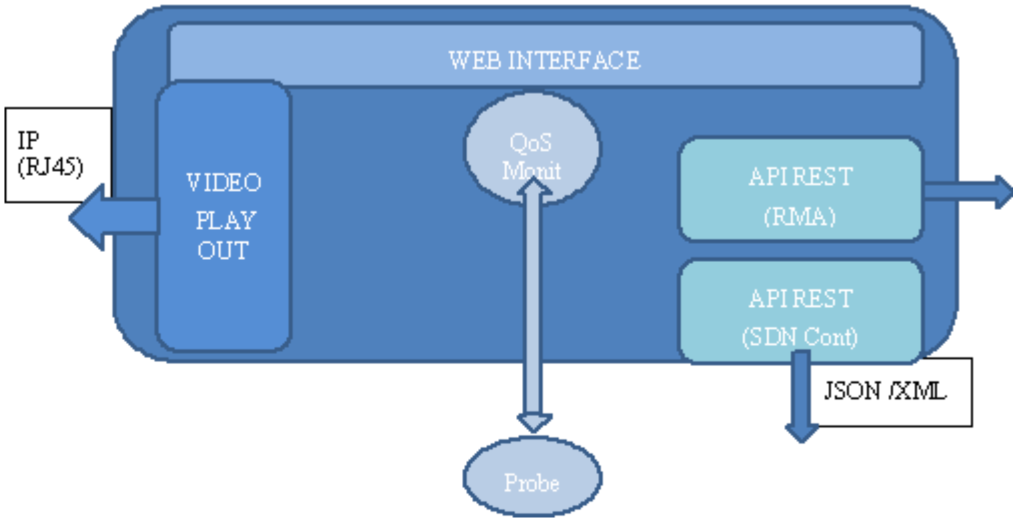


Figure 57: TVBA high-level software design

The main internal components showed in the Figure 57 are described in the table below.

Table 41: TVBA application components

Component	Description
REST Client	Entity in charge of interacting with the XCI components via REST APIs.
Video Play-out	It integrates the video service configuration, signalling and the stream play out.
QA Probe	Software in charge of video extraction from the nodes and the assessment of the quality.
Service Manager	Entity in charge of deciding the content delivery rules and building the broadcast tree through SDN controller.
User video device	End-user device for service consumption.

9.7.3 Implementation roadmap

The main objective of this testing and functional validation is to prove that the TV Broadcasting Application (TVBA) is able to instantiate a video service (up to High Definition resolution) and inject it to the Crosshaul network infrastructure available. It will also be tested that it is able to control and manage the service provisioning by monitoring the network status by means of creating and implementing the service delivery rules.

To that end, the main aspects that the planned set of tests aim are the following:

1) Video service configuration. Stream and file play-out. User control interface for management and operation.
2) Video service reconfiguration.
3) Deployment and functional testing of different software pieces of the server, virtual machines instantiation and network connectivity.
4) End-to-end service monitoring. Quality assessment/QoS estimation.

The implementation roadmap of the TVBA application is shown in Figure 58.

Figure 58: Roadmap for TVBA implementation

10 Appendix III: Evaluation plans and initial results per application

10.1 Multi-Tenancy Application

10.1.1 Function validation objectives

For validation and evaluation purpose, we will study a generic MTA scenario and a high speed train scenario for vehicle mobility use case in [1]

10.1.1.1 General MTA scenario

The main objective of the validation is to test and evaluate the ability of the MTA on slicing the infrastructure resources for different tenants given a physical Crosshaul infrastructure and to validate the admission control function which accepts or rejects tenant requests to ensure the SLA of all tenants and avoid overloading the system. The objective is to reduce the costs (i.e., CAPEX and OPEX) by sharing the infrastructure resources and maximize the cost-efficiency as well as the energy efficiency.

10.1.1.2 High Speed Train (HST) scenario

MTA for High Speed Train (HST) scenario basically consists of two parts; On-board MTA and On-land MTA. On-board MTA is applied inside high speed train to handle packets from different network operators and is composed of two modules namely *Packet Inspection module* and *Load Balancing module* shown in Figure 59. On-land MTA is applied in transport network shared by different network operators and is to play as slice manager to handle the slices for different network operators show in Figure 60. (Note that slice formulation and the path computation is provided by VIMAP and RMA applications but not discussed in MTA) Each slice is dedicated to a network operator when this operator requests to on-land MTA cooperating with VIMAP application (refer to IR 4.2) to create/modify/delete slices in the shared transport network. In the following description, it describes the functionalities provided by on-board MTA and on-land MTA. In the following description, two XCI controllers are used to interface with forwarding elements (i.e., XEFs in the shared transport network) and outbound gateways deployed in the high speed train scenario.

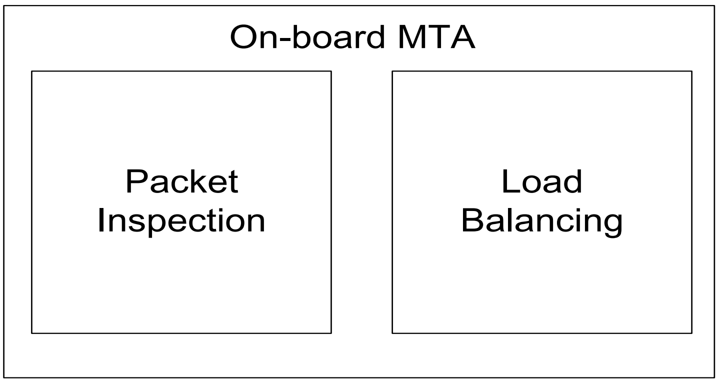


Figure 59: Component module in on-board MTA

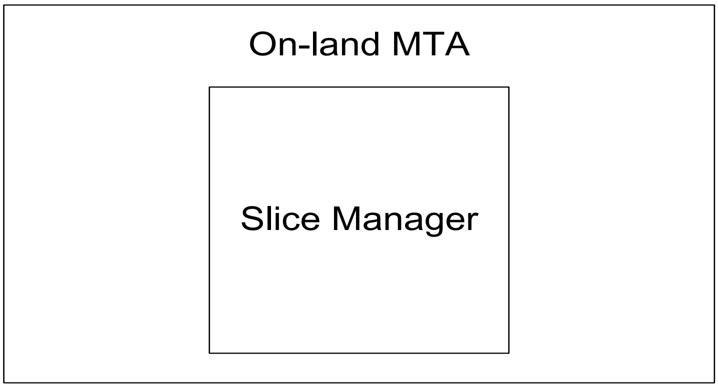


Figure 60: Component module in on-land MTA

On-board MTA and on-board controller

For on-board condition, there are two modules provided by on-board MTA, which are Packet Inspection module and Load Balancing module. These two modules work with on-board XCI controller to the server each packet transmitted from different network operators' UEs. The following is to describe the functionalities provided by Packet Inspection and Load Balancing.

- Packet Inspection Module

Packet Inspection of on-board MTA is to inspect packets forwarded from shared Point of Accesses (PoAs) such as shared small cell deployed in the high speed train to serve all UEs from different network operators.

When a packet forwarded from these shared PoAs (e.g., shared small cells) is received by a connecting OF switch which is used to connect all PoAs deployed on the high speed train, this packet will be forwarded to on-board XCI controller and extracted payload part of this received packet and submitted to Packet Inspection module to check the packet belonging to which network operator. Once this packet is determined the information of the network operator, on-board MTA commands on-board XCI to generate a special Crosshaul header for this packet and to consider the original packet as data payload to create a new packet for this packet shown in Figure 61. After finishing the process, connecting OF switch forwards the newly generated packet to Load Balancing module to determine which outbound gateways will be used to forwarded this generated packet.

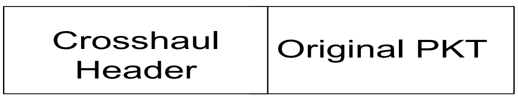


Figure 61: New generated packet (PKT)*

- Load Balancing Module

Load Balancing is to select an available/appropriate outbound gateway whose bandwidth is not used up currently. When Load Balancing module receives a packet from Packet Inspection module, it starts to check which outbound gateways can transmit this packet to wireless backhaul using Eq. (1).

$$B_i^{Outbound} - B_i^{Accumu.} \geq S^{PKT*}, i \in [1,2,\dots,m_1], \quad (1)$$

where $B_i^{Outbound}$ is bandwidth provided by outbound gateway i, $B_i^{Accumu.}$ is accumulative bandwidth of outbound gateway i, S^{PKT*} is size of the newly generated PKT*, m_1 is the number of outbound gateways on-board.

On-land MTA and on-land controller

On-land MTA is responsible for slice management by cooperating with VIMAP application. VIMAP application will help on-land MTA to assign a slice with an identifier (i.e., tenant ID) to a network operator. Each slice dedicated to a network operator can be composed necessary network resource and network functions to help the packet forwarded in transport network to correct core network of this network operator.

- Slice Manager Module

In slice manager module, on-land MTA can maintain the current slices assignment/distribution in the shared transport network and communicate with on-board MTA by exchange the corresponding slice information periodically. How to communicate with on-board MTA will be discussed in the future but not mention in this deliverable.

10.1.2 Roadmap

Table 42 shows the roadmap for validating the MTA algorithm for the general scenario and Table 43 for the High Speed Train (HST) scenario.

Table 42: Generic scenario

Roadmap		
Features	Description	Timeline
Tenant resource (de)allocation decision function	Decides the allocation/modification/deallocation of network, compute and storage resources for deployment of virtual infrastructures, which are isolated per tenant upon their requests, by solving the Virtual Network Embedding (VNE) problem.	Q1/2017
Admission control function	Accept or reject VNO requests that may lead to violating an SLA or a waste of physical resources.	Q2/2017

Table 43: High Speed Train (HST) scenario

Roadmap		
Features	Description	Timeline
Topology view	Get the network topology and inventory	Q3/2016
Packet Inspection	Inspect the content of received packet in on-board HST scenario	Q3/2016
Load balancing	Distribute traffic load of outbound gateways deployed on on-board HST scenario	Q4/2016
Slice Manager	Slice management and assignment for on-land condition	Q4/2016 to Q1/2017
Integrate on-board and on-land MTA	Integrate on-board and on-land MTA	Q1/2017 to Q2/2017

10.1.3 Validation environment

10.1.3.1 General MTA scenario

We consider a general MTA scenario described in section 6.2.5, where the owner of Crosshaul infrastructure decides the allocation of virtual resources over its substrate network to multiple tenants.

The scenarios for validation of the MTA algorithms depends on the physical topologies and the virtual infrastructure (tenant) requests.

Physical Crosshaul infrastructure

In order to get statistically meaningful network topologies, we will validate our MTA algorithms over a large set of physical Crosshaul infrastructure topologies. The first and second set of topologies will be based on two real backhaul networks of existing mobile operators in two European countries. In this case, we only consider Ethernet over different PHYs. We will define a set of link profiles and choose the one that match better the given capacities. We assume classic store-and-forward switching which induces a per-hop latency equal to propagation delay –distance(m)/200s for copper, distance(m)/300 s the rest–plus serialization delay (bits/rate).

The third set of topologies is purely random scenarios with a tree structure. To generate topologies with a tree structure, first we generate a set of XPUs connected in a ring. These will be located at the root of the trees. Second, hanging from each XPU, we create a pure random tree using independent Poisson processes to model the number of levels of the tree and nodes per level. The leaves of the tree correspond to RRHs. For each level of each tree, we randomly add a backup link with the upper layer of a neighboring tree, to add additional degrees of freedom for routing, following another Poisson process. Once we have the graph structure, we randomize the profile of all the links and their length. For each link and topology, we randomly pick one profile depending on its proximity to a XPU or an RRH. To this aim, we group links based on its proximity to a XPU and assign them the same link profile. Additionally, we assign high-capacity profiles to links closer to XPUs with more probability, to capture the fact that aggregation segments typically provide higher capacities.

The last set of topologies consists of arbitrary Waxman random graphs, based on the Erdos-Renyi random graph model, which is popular to evaluate realistic backhaul topologies.

Tenant requests with constant or stochastic demands

We will randomly create the tenant requests following the Poisson Process, and for each tenant we will decide its demand on the required network and cloud resources for a desired virtual topology graph, including the following information:

- Virtual node indexes.
- Virtual link information is provided as an adjacency matrix of size NxN where N is the number of virtual nodes. An entry equal to 1 indicates the existence of a direct bidirectional link, 0 if there is no direct link.
- Each virtual link request has associated a network demand:
  - Max, min, mean and variability of traffic load [Mb/s]
  - Estimation on max, min, mean, and variance of link delay [usec]
  - Max packet loss rate tolerated [%]
- Each virtual node has associated a computing demand:
  - Max, min, mean and variability of computing resources required [op/s]
  - Max, min, mean and variability of memory requirement [bytes]

We will explore different stochastic models for generating the demand of the tenants on the virtual links and computing demand of the virtual nodes.

10.1.3.2 High Speed Train (HST) scenario

Figure 62 shows the validation environment. In this figure, the proposal is to validate the condition of two network operators shared on-board and on-land network resources such as shared small cell, outbound gateways, and transport network. Two XCI controllers are used to interface with XFEs which are connected to two core networks of OP 1 and OP 2.
In this test case, when OF switch 1 receives a packet from shared SC 1, if it cannot process this received packet, it will send this packet to on-board controller. On-board XCI controller will extract the payload of this packet and send it to on-board MTA to check the content of the packet. The main functionalities of on-board MTA have two modules, one is Packet Inspection module and the other is Load Balancing module as mentioned in previous sections (i.e., Sec. 3.1.1).

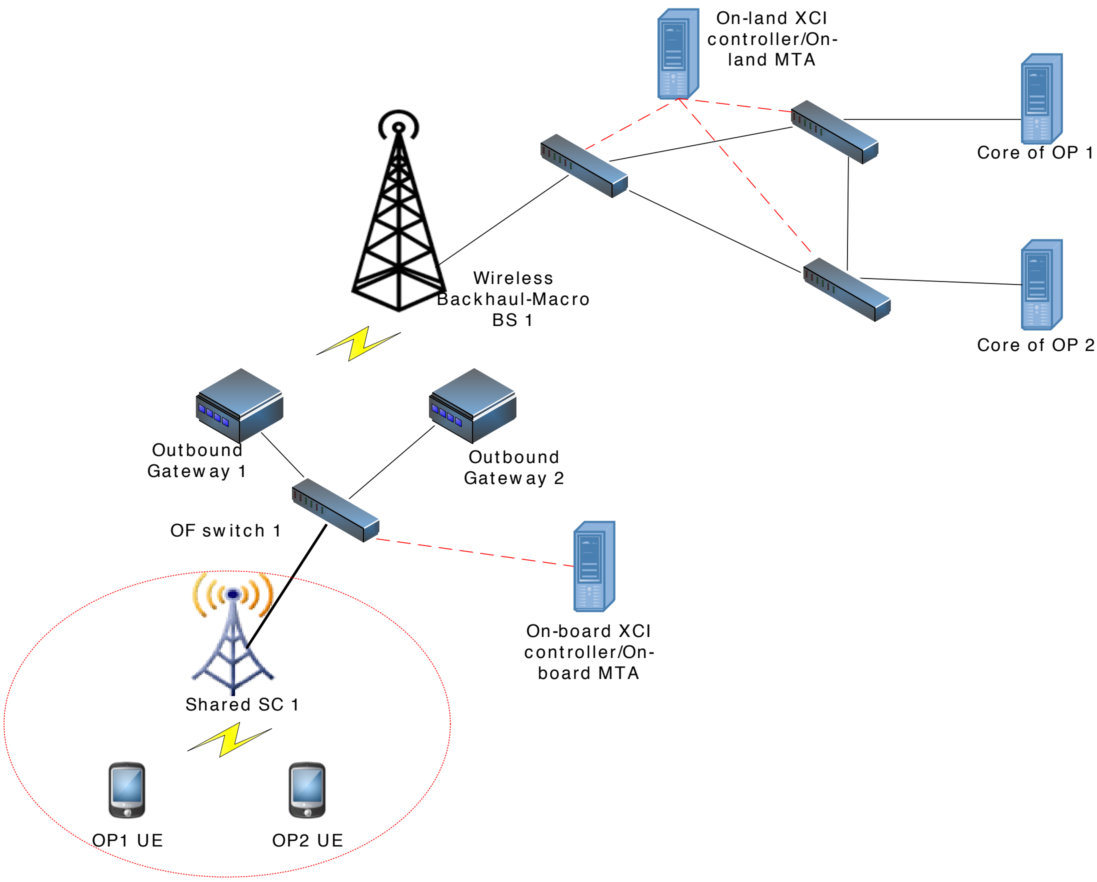


Figure 62: Validation environment of two network operators. The lower part of the figure shows on-board condition and the upper part of shows on-land condition.

When on-board MTA receives the payload of the packet, it executes Packet Inspection module to determine the information of network operators. For example, in a mobile network, the information of network operators cannot be recognized by packet header and needed to examine some part of the payload. In this manner, on-board MTA can know this packet belongs to which network operator who may want to share network resource. In this way, on-board MTA can manage and distribute on-board network resource efficiently according to the requirements of different network operators.

After the step of packet inspection, on-board MTA will know the packet exactly belongs to which network operator. Therefore, on-board MTA can command on-board XCI controller to tag a Crosshaul header which is related to slice information of shared transport network. As a result, in order to efficiently forward the packet to the desired core network of the corresponding network operator, this Crosshaul header encapsulated in front of the original packet will be processed by XFEs with preconfigured flow tables efficiently.
Load Balancing module basically is to efficiently make use of the bandwidth of outbound gateways deployed on-board. This module can prevent the overload of outbound gateways on-board. Also, on-board MTA can allocate the bandwidth of outbound gateways according to a policy which is agreed by all sharing network operators.

Simulation settings

In this validation environment, Mininet is used to generate on-land and on-board network topology. Since on-board condition usually relates to Vehicle mobility use case (refer to IR 1.1), it uses a host generated by Mininet to play as a shared small base station and ignores the radio condition between shared small cell and UEs from different networks since it is out of the scope of Crosshaul project.
It is noted that this host acting as a shared small cell on-board will generate the mobile packet. In a mobile packet, usually, it will contain the information of network operators and requires tunnelling mechanism between Control Plane (CP) and User Plane (UP) traffic and network entities in the core network. However, it will not be touched in Crosshaul. Therefore, to simplify the simulation, a packet is generated and determined to belong to a network operator randomly, i.e., belong to OP 1 or OP 2.
In addition, to simply the core network since a core network generally consists of special network entities such as authentication, authorization, accounting, and so forth, two hosts generated by Mininet play as core networks of OP 1 and OP 2.

10.1.4 Validation procedure

10.1.4.1 General MTA scenario

The basic MTA functional validation for the general multi-tenancy scenarios will be performed through the execution of the following testing procedures.

Table 44: MTA Test Card 1

Test Algorithm	NEC_MTA _01	Execution Status	Testing
Test Name	Tenant resource (de)allocation decision algorithm		
Objectives	Verify the tenant resource (de)allocation decision algorithm to solve the Virtual Network Embedding (VNE) problem on the node mapping and link mapping, and decide the dynamic allocation/modification/deallocation of network, compute and storage resources for deployment of a virtual infrastructure (i.e. a tenant) upon request. The objective is to reduce the costs and maximize the utilization of resources in cost-efficient manner, while at the same maximizing the energy efficiency.		
Related Use Cases	Multitenancy use case [1]		

Responsible	NEC
Related Test Cards	N/A (not applicable)

<table>
<tr><th colspan="2">SUT and topology</th></tr>
<tr><td>SUT</td><td>Tenant resource (de)allocation decision algorithm to determine the node mapping and link mapping, and decide allocation/modification/deallocation of network, compute and storage resources for deployment of a virtual infrastructure (a tenant).</td></tr>
<tr><td colspan="2">Test environment is explained in section 0.
Physical Crosshaul Topology
• Semi-random topologies based on a backhaul topology of an operator in two European countries.
• Random topologies with a tree structure
• Random topologies based on Waxman algorithm
Tenant requests with stochastic Tenant demands:
A virtual infrastructure allocation request consists of:
• a set of virtual infrastructure nodes (i.e., virtual network nodes)
• a set of virtual infrastructure computing nodes (i.e., virtual hosts)
• a set of virtual infrastructure links (also referred to as virtual connections)
Virtual link information is provided as an adjacency matrix of size NxN where N is the number of virtual nodes. An entry equal to 1 indicates the existence of a direct bidirectional link, 0 if there is no direct link. We will explore different stochastic models for generating the demand of the virtual links and on the computing demand at the virtual nodes.
• Each virtual link request has associated a network demand (constant/stochastic):
    ○ Max, min, mean and variability of traffic load [Mb/s]
    ○ Estimation on max, min, mean, and variance of link delay [usec]
    ○ Max packet loss rate tolerated [%]
• Each virtual node has associated a computing demand (constant/stochastic):
    ○ Max, min, mean and variability of computing resources required [op/s]
    ○ Max, min, mean and variability of memory requirement [bytes]</td></tr>
<tr><td>Additional Algorithms Information</td><td>• Abstraction of the underlying physical Crosshaul infrastructure (offered by the XCI)
• Routing algorithm to set the physical paths for a virtual link to connect two virtual nodes
• Algorithm for planning of a virtual infrastructure topology, specifying the demands on the required resources of the virtual nodes and virtual</td></tr>
</table>

	links in the tenant request

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Test of the tenant resource (de)allocation decision algorithm over a large set of semi-random and random topologies (explained in section 0), and different set of tenant requests with **constant demands** on the virtual links and on the computing resources of the virtual nodes. **Expected Results:** The tenant resource (de)allocation decision algorithm should reduce the costs in terms of CAPEX/OPEX and maximize the utilization of physical infrastructure resources with high cost-efficient and energy efficiency, while respecting networking constraints. Benchmarking algorithms based on state of the art approaches will achieve lower cost-efficient and energy efficiency, and higher cost, compared to the proposed tenant resource (de)allocation decision algorithm.	
2.	**Description:** Test of the tenant resource (de)allocation decision algorithm over a large set of semi-random and random topologies (explained in section 0), and different set of tenant requests with **stochastic demands** on the virtual links and on the computing resources of the virtual nodes. **Expected Results:** The tenant resource (de)allocation decision algorithm, which considers the stochastic demand of the tenants, should obtain multiplexing gain and thus improve the utilization of physical infrastructure resources. This leads to better cost-efficient and energy efficiency and lower cost, compared to the tests in the step 1 where only constant demands are given.	

Table 45: MTA Test Card 2

Test Algorithm	NEC_MTA _02	Execution Status	Testing
Test Name	Admission control algorithm		
Objectives	Verify the admission control algorithm to accept or reject a tenant request that may lead to violating SLA of existing tenants or a waste of physical resources, as well as to avoid overloading the system. The objective is to maximize the utilization of resources in cost-efficient manner, and also maximize the energy efficiency.		
Related Use Cases	Multitenancy use case [1]		

Responsible	NEC
Related Test Cards	N/A (not applicable)

<table>
<tr><th colspan="2">SUT and topology</th></tr>
<tr><td>SUT</td><td>Admission control algorithm to accept or reject a tenant request, based on the available physical resources and the required SLA of the tenants.</td></tr>
<tr><td colspan="2">Test environment is explained in section 0.
Physical Crosshaul Topology
• Semi-random topologies based on a backhaul topology of an operator in two European countries.
• Random topologies with a tree structure
• Random topologies based on Waxman algorithm
Tenant requests with stochastic Tenant demands:
A virtual infrastructure allocation request consists of:
• a set of virtual infrastructure nodes (i.e., virtual network nodes)
• a set of virtual infrastructure computing nodes (i.e., virtual hosts)
• a set of virtual infrastructure links (also referred to as virtual connections)
We will explore different stochastic models for generating the demand of the virtual links and on the computing demand at the virtual nodes.
• Each virtual link request has associated a network demand (constant/stochastic):
○ Max, min, mean and variability of traffic load [Mb/s]
○ Estimation on max, min, mean, and variance of link delay [usec]
○ Max packet loss rate tolerated [%]
• Each virtual node has associated a computing demand (constant/stochastic):
○ Max, min, mean and variability of computing resources required [op/s]
○ Max, min, mean and variability of memory requirement [bytes]</td></tr>
<tr><td>Additional Algorithms Information</td><td>- Abstraction of the underlying physical Crosshaul infrastructure and available resources (such information is provided by XCI)
- Stochastic modelling of the demands on the required resources of the virtual nodes and virtual links in the tenant request</td></tr>
</table>

<table>
<tr><th colspan="3">Test description</th></tr>
<tr><td>Step #</td><td>Step description and expected results</td><td>Status</td></tr>
</table>

1.	**Description:** Test of the admission control algorithm for accepting or rejecting a new tenant. **Expected Results:** The proposed admission control algorithm should be able to reject a new tenant request if the there is no more available resources or this tenant may lead to violating the SLA of existing tenants or a waste of physical resources. The proposed algorithm should achieve lower blocking probability (in other words allowing more tenants sharing infrastructure resources) while respecting networking constraints, and at the same time maximize the resource utilization and energy efficiency, compared to benchmarking algorithms based on state of the art approaches.	
2.	**Description:** Test of the admission control algorithm for accepting or rejecting a tenant update when the demand of this tenant changes over certain thresholds during the operation. **Expected Results:** The proposed admission control algorithm should be able to allow or deny a tenant changing its demand during operation. It will reject a tenant update, if there is no sufficient physical resource to meet this new demand request, or it may lead to violating the SLA of other existing tenants, or a waste of physical resources, or result in higher energy consumption. The proposed algorithm should achieve lower blocking probability (in other words allowing more tenants sharing infrastructure resources) while respecting networking constraints, and at the same time maximize the resource utilization and energy efficiency, compared to benchmarking algorithms based on state of the art approaches.	

10.1.4.2 High Speed Train (HST) scenario

The following defines the procedure of on-board MTA and on-land MTA.

Algorithm 1: on-board MTA

```
On-board and on-land MTA synchronize the information of slice assignment in shared transport network
If on-board XCI controller receives a PKT from shared PoAs deployed in HST
  forward PKT to on-board XCI controller and extract payload of PKT
  execute PacketInspection(payload of PKT)
End if

PacketInspection(payload of PKT):
  examine payload to get information of network operator
  request on-board XCI controller to generate corresponding header to be a new PKT*
  execute LoadBalancing()

LoadBalancing():
  check the available bandwidth of all outbound gateways
  if the available bandwidth of outbound gateways
    select an appropriate outbound gateway to forward PKT* to wireless backhaul
  else
    used up bandwidth of outbound gateways
    drop PKT*
  end if
```

Algorithm 2: on-land MTA

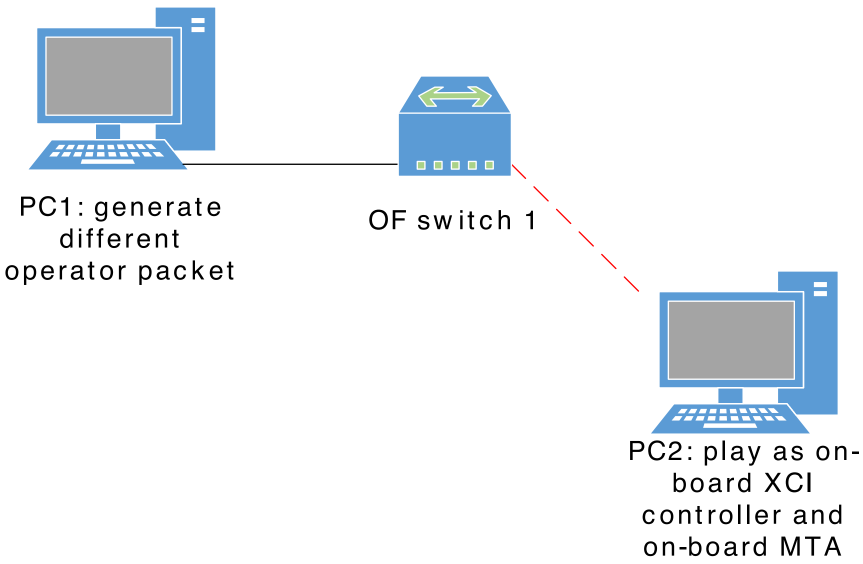
```
if a PKT is received from wireless  backhaul to shared transport network
  if XFEs by using pre-configured flow tables can handle this PKT
    process and forward this PKT correspondingly
  else
    forward to on-land XCI controller to be processed
end if

SliceManager():
  if receive a slice request from a network operator
    request VIMAP application to check whether a slice can be assigned to this network operator
      if available network resource can satisfy the slice request
        accept this request and generate a slice to this network operator
        update slice information
      else
        reject this request
      end if
  end if
```

The validation of the above algorithms is described in the following tables.

Table 46: High Speed Train Test Card 1

Test Algorithm	ITRI_on-board MTA _01	Execution Status	Testing
Test Name	Packet Inspection module		
Objectives	Inspect a packet to determine the information of network operator		
Related Use Cases	Multitenancy and Vehicle Mobility use cases		
Responsible	ITRI		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Packet inspection is developed to analyze the payload of a packet to determine the information of network operators
	
Additional Algorithms Information	• Extract payload of a PKT • Determine protocol structure used by network operators • Find the info. of a network operator

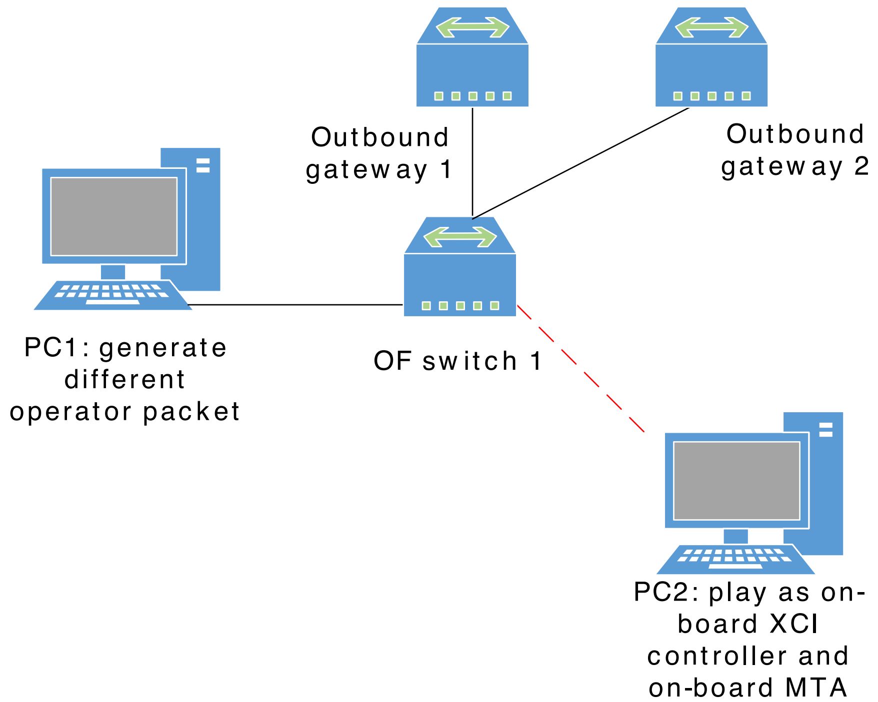
Test description		
Step #	Step description and expected results	Status
1.	**Description:** Generate network topology shown in the above. PC 1 plays as a packet generator to generate different operator packets. OF switch 1 is controlled by on-board XCI which has an NBI interface with on-board MTA. **Expected Results:** Generate a simple network topology.	
2.	**Description:** PC 1 generates a mobile packet and sends it to OF switch 1. OF switch 1 forwards it to on-board XCI controller. On-board XCI extracts the payload of this packet and sends it to Packet Inspection module of on-board MTA. **Expected Results:**	

	On-board XCI extracts the payload of this packet	
3.	**Description:** Packet Inspection determines the protocol structured used by network operators and then analyses the infomation of the network operator of this packet **Expected Results:** Packet Inspection module gets the information of the network operator of this packet	
4	**Description:** Packet Inspection of on-board MTA command on-board XCI controller to generate a new header for this packet **Expected Results:** A new header is generated and used as an outer header by processed by XFEs with pre-configured flow tables	

Table 47: High Speed Train Test Card 2

Test Algorithm	ITRI_on-board MTA _02	Execution Status	Testing
Test Name	Load Balancing module		
Objectives	Distribute traffic load on-board in a balancing manner for the outbound gateways on-board		
Related Use Cases	Multitenancy and Vehicle Mobility use cases		
Responsible	ITRI		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Load Balancing is developed to distribute traffic load on-board in a balancing manner to make use of the bandwidth of outbound gateways on-board.



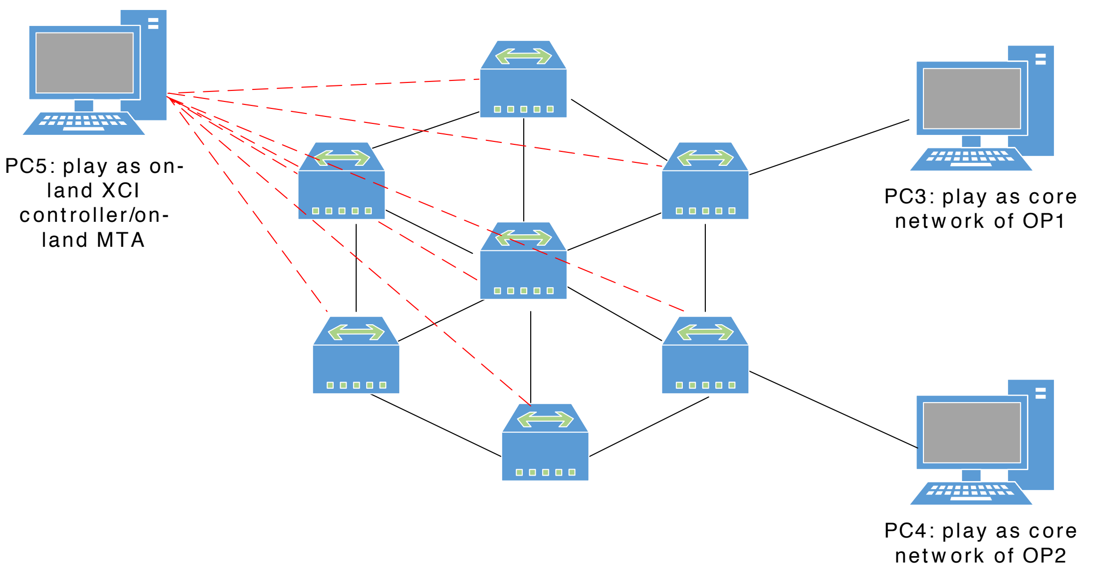
Additional Algorithms Information	• Before forwarding a packet through an outbound gateway on-board, it will check whether the available bandwidth of the outbound gateways can forward the current packet or not.

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Generate network topology shown in the above. PC 1 plays as a packet generator to generate different operator packets. OF switch 1 is controlled by on-board XCI which has an NBI interface with on-board MTA. Two outbound gateways are generated to forward the packet into shared transport network. **Expected Results:** Generate a simple network topology with two outbound gateways.	
2.	**Description:** PC 1 generates a mobile packet and sends it to OF switch 1. OF switch 1 forwards it to on-board XCI controller. On-board XCI extracts the payload of this packet and sends it to Packet	

	Inspection module of on-board MTA. After inspection, Load Balancing module will compute which outbound gateway is appropriate to forward the packet to shared transport network by using Eq. (1) **Expected Results:** Load Balancing module selects an appropriate outbound gateway to forward the packet	

Table 48: On-land MTA Test Card 1

Test Algorithm	ITRI_on-land MTA _01	Execution Status	Testing
Test Name	Slice Manager		
Objectives	Manage slices in shared transport network through VIMAP application		
Related Use Cases	Multitenancy and Vehicle Mobility use cases		
Responsible	ITRI		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Slice manager is to manage the slice conditions and update slice information with on-board MTA (if any updated infomation of slice in shared transport network is required.)
 	
Additional	• Slice management with the help of VIMAP application

Algorithms Information	

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Generate network topology shown in the above where PC5 acts as on-land XCI controller and MTA, PC3 and PC4 act as core networks of OP 1 and OP 2 **Expected Results:** Generate a simple shared transport network topology with two core networks of OP 1 and OP 2.	
2.	**Description:** Pre-configuring flow tables for each XFEs in shared transport network to handle the packets from OP 1 and OP 2. **Expected Results:** Each XFE used in shared transport network can process different operators' packets to corresponding core networks.	

10.2 Mobility Management Application

This section has the purpose of providing the evaluation and validation objectives of the Mobility Management Application (MMA), that will have two implementations within 5G-Crosshaul. On the one hand, the main goal of the MMA is to offer mobility management and optimize traffic offloading for media distribution. Therefore, MMA shall detect the new users in the 5G-Crosshaul domain, notify the CDNMA constantly (through a REST API), compute all the data for the path creation and also optimize the handover. And on the other hand, the Mobility Management Application is one of the OTT (Over-The-Top) applications of the 5G-Crosshaul. MMA aiming to reduce handover (HO) time in vehicle mobility use cases like high speed train scenarios.

10.2.1 Functional validation objectives

Offloading case

The MMA is in charge of the mobility management, hence, it requires a view of the network topology, the XCI SDN controller shall offer this information, particularly:

- The nodes (XFE and ports)
- The links (connection points and bandwidth)
- The host and node connections (XPU and ports)

MMA shall detect the new users in the 5G-Crosshaul domain and optimize the handover of all of the users. When a user enters in the 5G-Crosshaul domain the MMA obtain the information of the user, assigns an IP to the user, get the topology and assigns a GW/PoC based on heuristic and proximity, then the MMA notifies the CDNMA retrieving the CDN Node assigned and finally computes the information and creates the paths with the help of the RMA.

The validation objectives of the MMA are:

- **User detection:** detect the new users in the 5G-Crosshaul domain and get the necessary information of them (e.g., MAC address, access point).
- **IP assignment:** assign an IP to the new user (e.g., DHCP).
- **Topology view:** get the topology view from the SDN controller.
- **Assign GW/PoC:** assign the best GW/PoC based on heuristic and proximity.
- **REST API:** implement a REST API and a subscription method and notify the presence of a new user to CDNMA subscribed (or/and other applications) and retrieve the CDN Node assigned.
- **Path creation:** create the best path with the help of the RMA.
- **Handover:** perform and optimize the handover determining the target BSS and creating proactive paths.

High speed train use case

Crosshaul HandOver (HO) is a critical operation in mobility management, especially for high mobility scenarios because of more frequent HOs and higher sensitivity to link degradation. The MMA can use the services offered by the RMA to provide best paths between the different elements of the network, with the main goal of optimizing the routes or paths followed by mobile users' traffic towards the Internet or to a core service provided in a data centre as shown in Figure 23. Besides proactive route optimization, the centralized controller can control packet forwarding rules and mobility management (i.e., MMA application installed above the centralized controller) and give the intelligent computing for all kinds of network services. By implementing the concept of proxy in MMA, this can communicate with the RAN and core network on behalf of the core network and RAN, respectively. In this manner, handover signalling can be handled through those proxies to achieve simultaneous HO stages as shown in Figure 66, the HO time can be reduced without modifying existing RAN and CN. As depicted in Figure 64, MMA procedure can be summarized as follow:

- HO detection
  - Performed by forwarding specific packets to the controller for analysis.
  - Can be performed all the time or in a context-aware basis
- HO execution
  - RAN and CN proxies communicate with CN and RAN on behalf of the real ones, respectively.
  - Different HO stages, which are assumed to happen in order, can now happen simultaneously.
  - The application will handle the interaction among RAN, CN and the proxies to assure consistency.

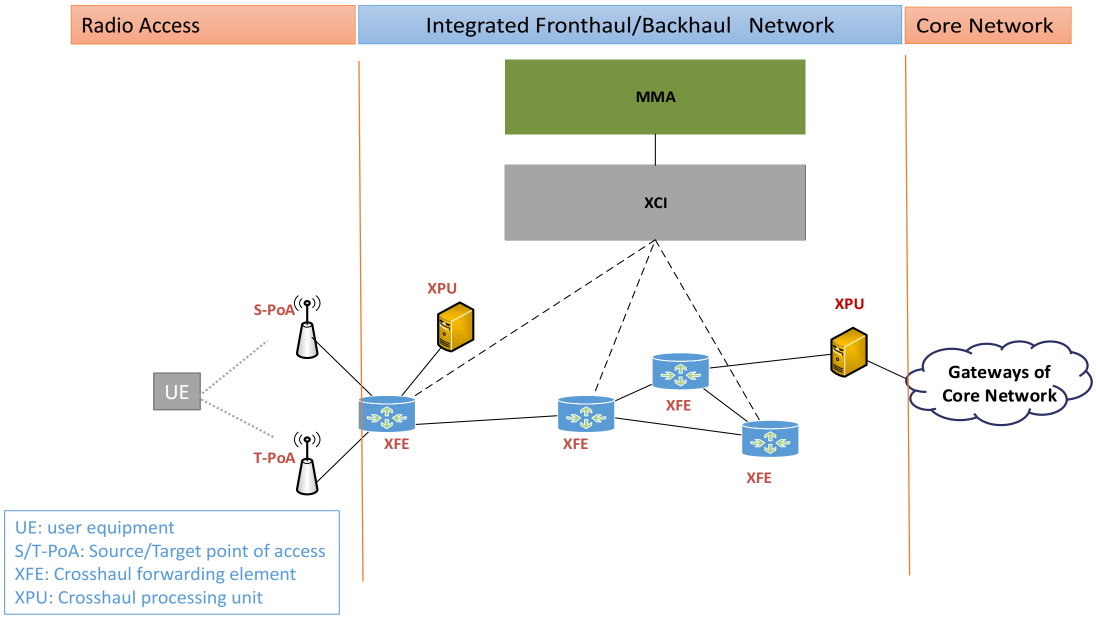


Figure 63: MMA interaction with 5G-Crosshaul for high speed train use case

The conventional HO procedure is shown in Figure 65 as reference, and the improved HO procedure by applying MMA is shown in Figure 66.

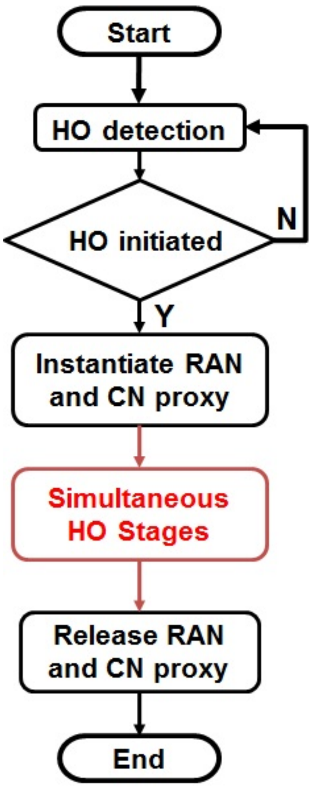


Figure 64: MMA Handover optimization procedure

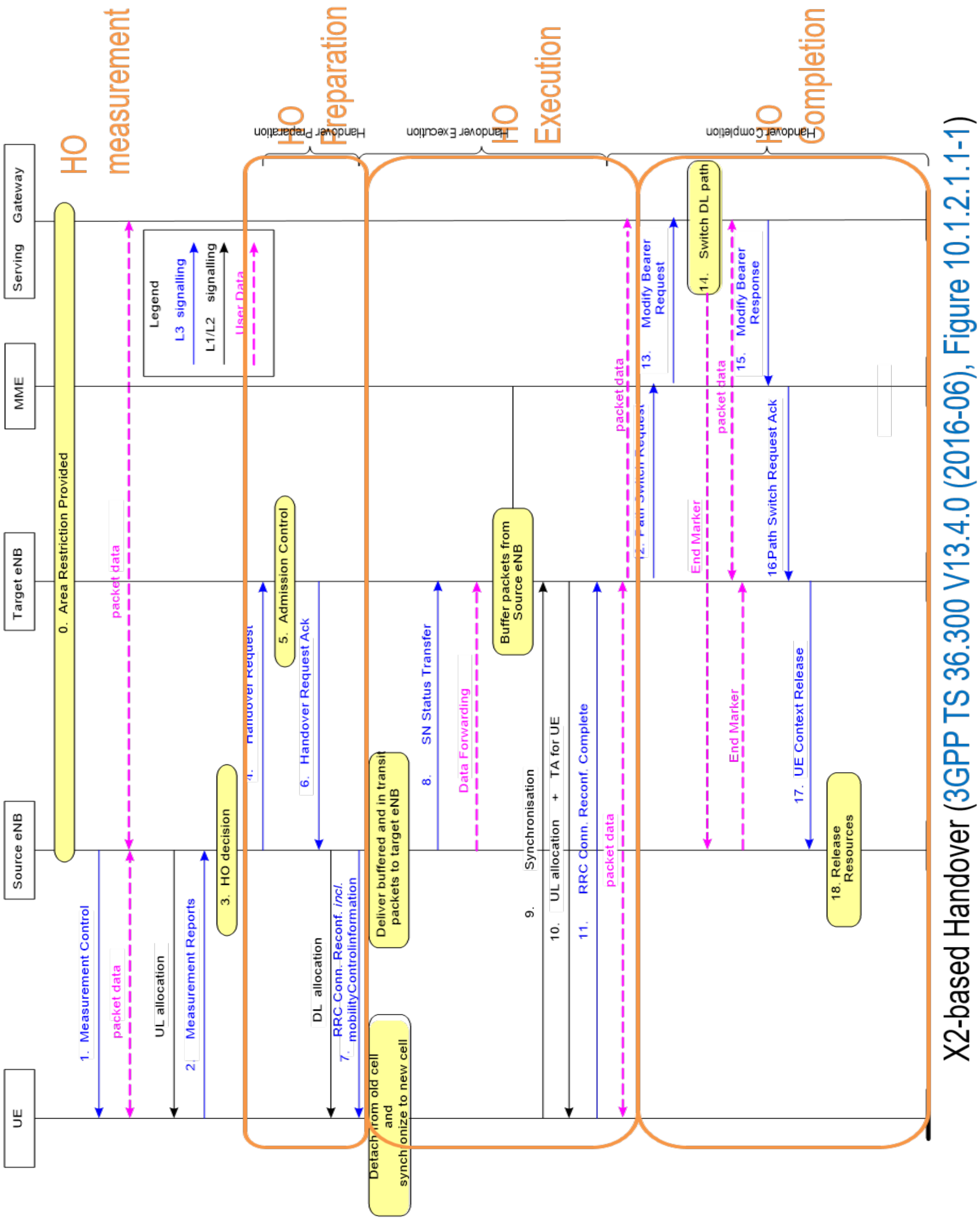


Figure 65: Conventional HO procedure (4G-LTE as example)

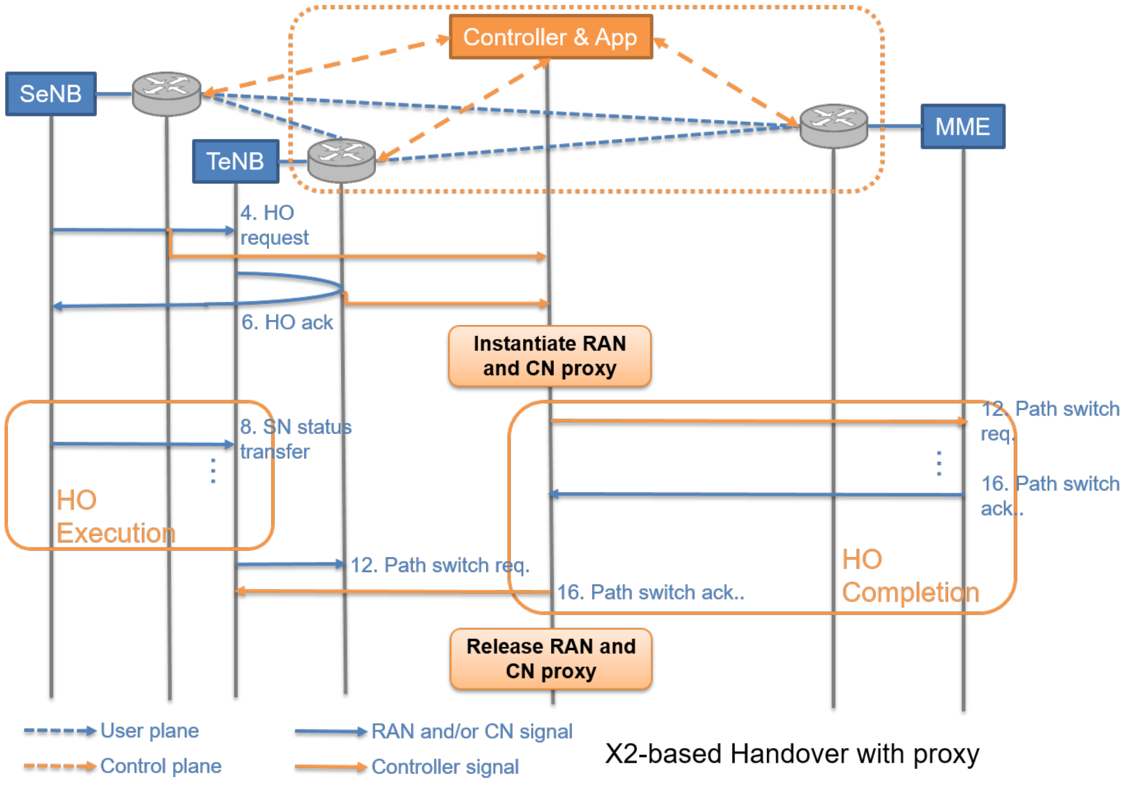


Figure 66: HO procedure with MMA

The validation objectives of the MMA for HST use case are:

- **Packet inspection:** detect the new HO requests in the 5G-Crosshaul domain and get the necessary information of them (e.g., MAC address, access point).
- **Parallel handover process:** perform handover completion processing while handover execution is still ongoing.
- **Complete handover testing:** perform simulated high-speed handover in Crosshaul environment.

10.2.2 Roadmap

Offloading case

Table 49 shows the roadmap for the MMA implementation following the functional validation objectives described above in 3.2.1:

Table 49: MMA Offloading case roadmap

Roadmap		
Features	Description	Timeline
User detection	detect the new users in the 5G-Crosshaul domain and get the necessary information	Q2/16
IP assignment	assign an IP to the new user	Q2/16
Topology view	get the topology view from the SDN controller	Q3/16

GW/PoC assignment	assign the best GW/PoC based on heuristic and proximity	Q4/16
REST API	implement a REST API and a subscription method	Q3/16
Path creation	create the best path with the help of the RMA	Q4/16
Handover	perform and optimize the handover	Q1/17

High speed train use case

Table 49 shows the roadmap for the MMA implementation following the functional validation objectives described above:

Table 50: MMA HST Roadmap

Roadmap		
Features	Description	Timeline
Topology view	get the topology view from the SDN controller	Q3/16
Packet inspection	detect the new users in the 5G-Crosshaul domain and get the necessary information	Q3/16
REST API	implement a REST API and a subscription method	Q3/16
Parallel handover processes	Perform Handover execution and completion at the same time.	Q3/16
Packet inspection and parallel process testing	Perform simulated high train speed handover in Crosshaul environment	Q4/16
Handover testing	optimize the handover	Q1/17

10.2.3 Validation environment

Offloading case

The current validation environment for testing the MMA is based on a physical access point to the 5G-Crosshaul domain represented by a Saxnet Meshnode III with four wireless interfaces and one wired interface, a physical machine acting as the controller connected through an out-of-band channel to the PoA and different mobile users.

The Saxnet Meshnode III is running a Debian 8.0 “Jessie” Linux distribution with OpenvSwitch and works as a datapath element and 5G-Crosshaul access point. The controller software utilized is Ryu and the controller machine uses Intel Core i7 and 16Gb RAM. The mobile users utilized are various smartphones (iOS, Android) and different desktop computers and laptops (Linux, Windows, OS X) .

This validation environment allows the testing of i) the user detection in the 5G-Crosshaul AP, ii) the IP assignment and iii) the REST API performance. The REST API is tested with a HTTP tool for GET/POST/PUT/DELETE requests and the CDN Node requests and responses are simulated locally with a script assigning random CDN Nodes.

High speed train use case

10.2.3.1 Description of validation environment

Figure 67 shows the validation environment. In this figure, the proposal is to validate the handover process. One XCI controller is used to interface with XFEs which are connected to the core networks.

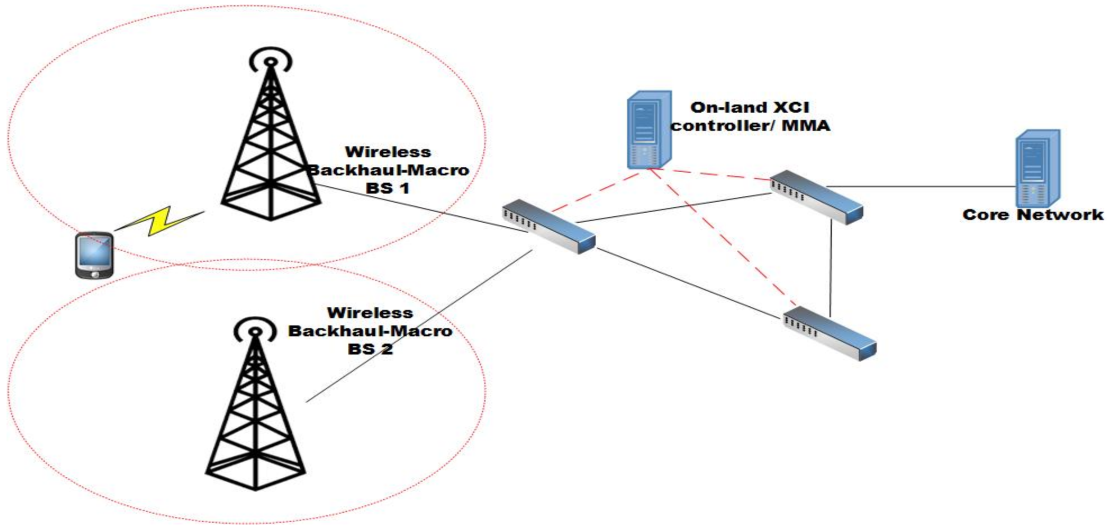


Figure 67: Validation environment of the handover process.

In this test case, when UE (which represents the high speed train) moves from BS1 coverage area to BS2 coverage area, BS1 will send a handover request. OF switch will receive the handover request among other controller messages shared between BS1 and BS2, if it cannot process this received packet, it will send this packet to XCI controller. The XCI controller will extract the payload of this packet and send it to on-board MMA to check the content of the packet. The main functionalities of MMA have two modules as shown in Figure 68, one is Packet Inspection module and the other is performing handover completion in parallel with handover execution as elaborated in the previous parts.

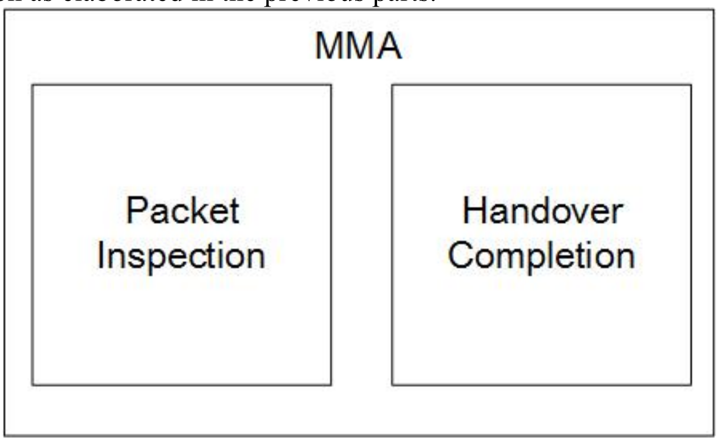


Figure 68: Component module in MMA

When MMA receives the payload of the packet, it executes Packet Inspection module to determine if the handover request exists. In this manner, MMA knows whether to perform handover completion. After the step of packet inspection, MMA will provide the proactive routing path between BS2 and core network. Therefore, MMA can command RMA to provide this path or it provides the calculation in MMA module itself.

10.2.3.2 Simulation settings

In this validation environment, Mininet is used to generate network topology. Since on-board condition usually relates to Vehicle mobility use case, it is used a host generated by Mininet to play as a shared small base station and ignores the radio condition between shared small cell and UEs from networks since it is out of the scope of Crosshaul project.
It is noted that the BS1 will generate a handover request the mobile packet. In a mobile packet, usually, it will contain the information of target BSl. Therefore, to simplify the simulation, a packet is generated and determined the handover process, i.e., from BS1 to BS2.

10.2.4 Validation procedure

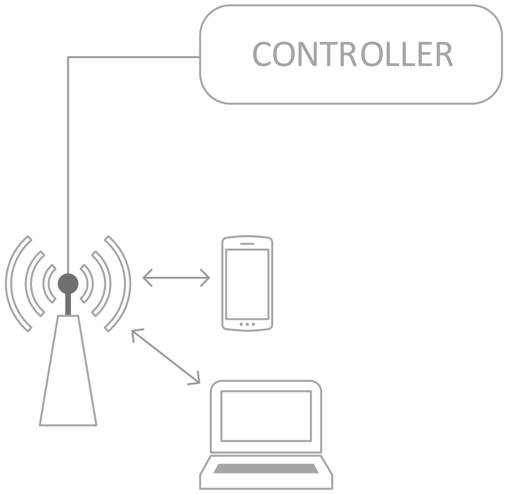
Offloading case

The following tables describe the validation procedures:

Table 51: MMA Offloading case test card 1

Test Card #	UC3M_MMA_001	Execution Status	Running
Test Name	User detection		
Objectives	Detect the new users in the 5G-Crosshaul domain and get the necessary information		
Related Use Cases			
Responsible	UC3M		
Related Test Cards	N/A		

SUT and topology	
SUT	MMA controller

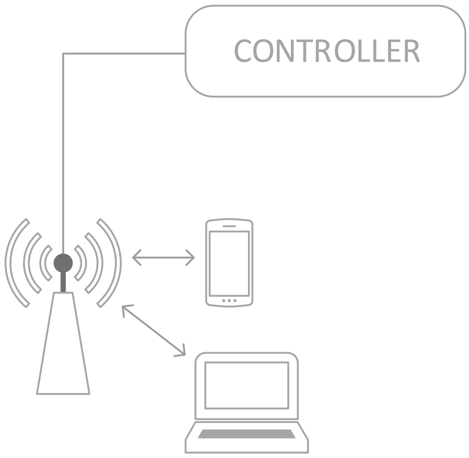
<table>
<tr><td>Test environment topology</td><td>
</td></tr>
<tr><td>Additional components/ algorithms</td><td>N/A</td></tr>
</table>

Test description		
Step #	Step description and expected results	Status
1.	**Description:** detect the connection of new users to the 5G-Crosshaul domain through the AP. **Expected Results:** detection of the presence of the new users	
2.	**Description:** once the user is connected we need to get the necessary information (MAC address of the user, PoA IP, PoA dpid) and store it. **Expected Results:** obtain the necessary information of the connected user and store it.	

Table 52: MMA Offloading case test card 2

Test Card #	UC3M_MMA_002	Execution Status	Running
Test Name	Assign IP address		
Objectives	Assign an IP address to the new users with DHCP		
Related Use Cases			
Responsible	UC3M		

Related Test Cards	UC3M_MMA_001

SUT and topology	
SUT	MMA controller
Test environment topology	
Additional components/ algorithms	N/A

Test description		
Step #	Step description and expected results	Status
1.	**Description:** once the UC3M_MMA_001 test is validated a new user is connected to the 5G-Crosshaul AP, the controller shall respond the DHCP messages and shall assign an IP address to the user and finally store all the information **Expected Results:** respond the DHCP messages, assign a new IP and store the information	
2.	**Description:** assign different IP addresses with DHCP to different users with no IP address conflict **Expected Results:** connect different users with no IP address conflict	

Table 53: MMA Offloading case test card 3

Test Card #	UC3M_MMA_003	Execution Status	Running

Test Name	REST API
Objectives	Test the correct operation of the REST API
Related Use Cases	
Responsible	UC3M
Related Test Cards	UC3M_MMA_001, UC3M_MMA_002

SUT and topology	
SUT	MMA controller
Test environment topology	N/A
Additional components/ algorithms	CDNMA (if there is no CDNMA, a dummy REST client and server for testing)

Test description		
Step #	Step description and expected results	Status
1.	**Description:** test the correct use of the REST API **Expected Results:** the API REST follows the model	
2.	**Description:** enable a notification subscription method through API REST **Expected Results:** the other applications (CDNMA) can subscribe for notification	
3.	**Description:** edit the subscription information (notification URL) **Expected Results:** notification information updated	
4.	**Description:** remove the subscription **Expected Results:** notification removed	

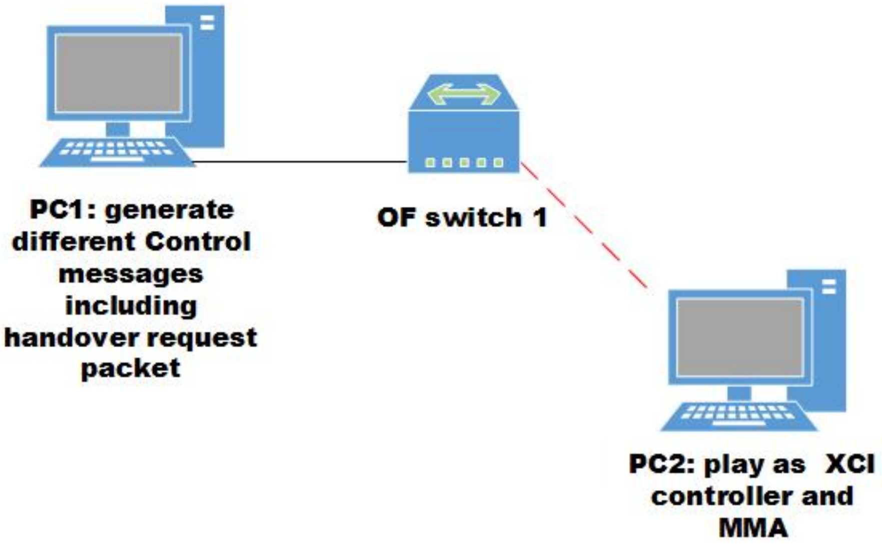
5.	**Description:** when a user connects to the 5G-Crosshaul domain (UC3M_MMA_001), the controller gets the information and assigns an IP address with DHCP (UC3M_MMA_002) the MMA shall notify to the CDNMA the PoA IP address and the CDNMA shall respond with the CDN Node assigned **Expected Results:** the CDNMA receives the notification and responses with the CDN Node, the MMA controller stores the information	
6.	**Description:** there are different user connected to the 5G-Crosshaul domain before the CDNMA subscription **Expected Results:** when the CDNMA is subscribed the MMA shall request to the CDNMA a CDN Node for each of the users connected	

High speed train use case

Table 54: MMA HST test card 1

Test Algorithm	ITRI_ MMA _01	Execution Status	Testing
Test Name	Packet Inspection module		
Objectives	Inspect a packet to determine the information of network operator		
Related Use Cases	Vehicle Mobility use cases		
Responsible	ITRI		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Packet inspection is developed to analyze the payload of a packet to determine the information of handover operation



Additional Algorithms Information	• Extract payload of a PKT • Determine protocol structure used by network operators • Find the info. of a handover request

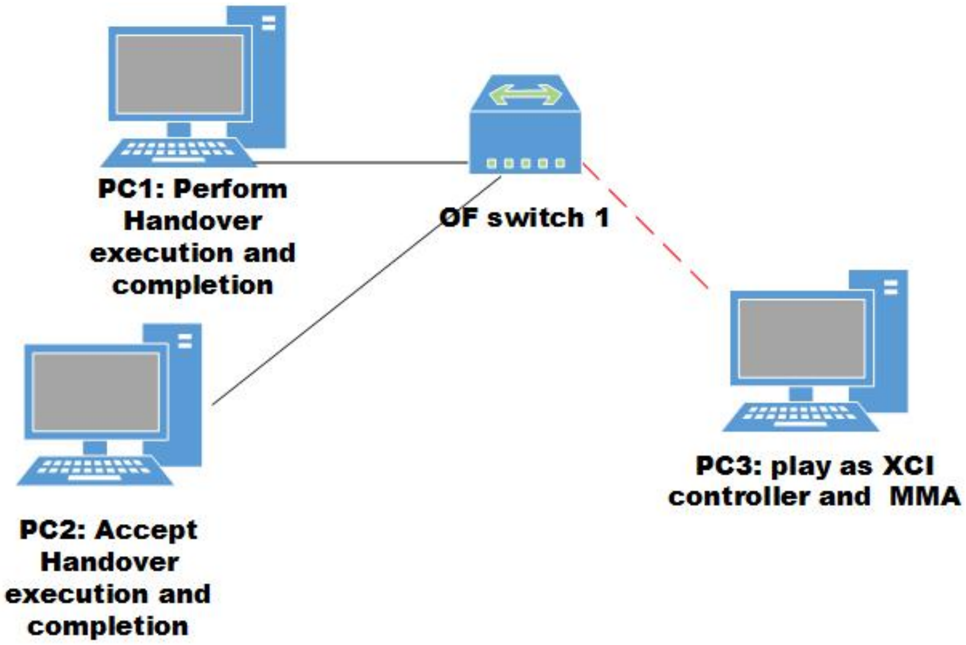
Test description		
Step #	Step description and expected results	Status
1.	**Description:** Generate network topology shown in the above. PC 1 acts as a packet generator to generate different operator packets including. OF switch 1 is controlled by on-board XCI which has an NBI interface with MMA. **Expected Results:** Generate a simple network topology.	
2.	**Description:** PC 1 generates a mobile packet and sends it to OF switch 1. OF switch 1 forwards it to on-board XCI controller. XCI extracts the payload of this packet and sends it to Packet Inspection module of MMA **Expected Results:** On-board XCI extracts the payload of this packet	
3.	**Description:** Packet Inspection determines the protocol structured used by	

	network operators and then analyses the info. of handover request of this packet **Expected Results:** Packet Inspection module gets the info. of handover operator of this packet	
4	**Description:** Packet Inspection MMA command XCI controller to generate a new header for this packet **Expected Results:** A new header is generated and used as an outer header by processed by XFEs with pre-configured flow tables	

Table 55: MMA HST test card 2

Test Algorithm	ITRI_MMA _02	Execution Status	Testing
Test Name	Parallel handover module		
Objectives	perform Handover execution and completion simultaneously		
Related Use Cases	Vehicle Mobility use cases		
Responsible	ITRI		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Parallel handover execution



Additional Algorithms Information	In this part, The MMA will help perform Handover execution and completion at the same time.

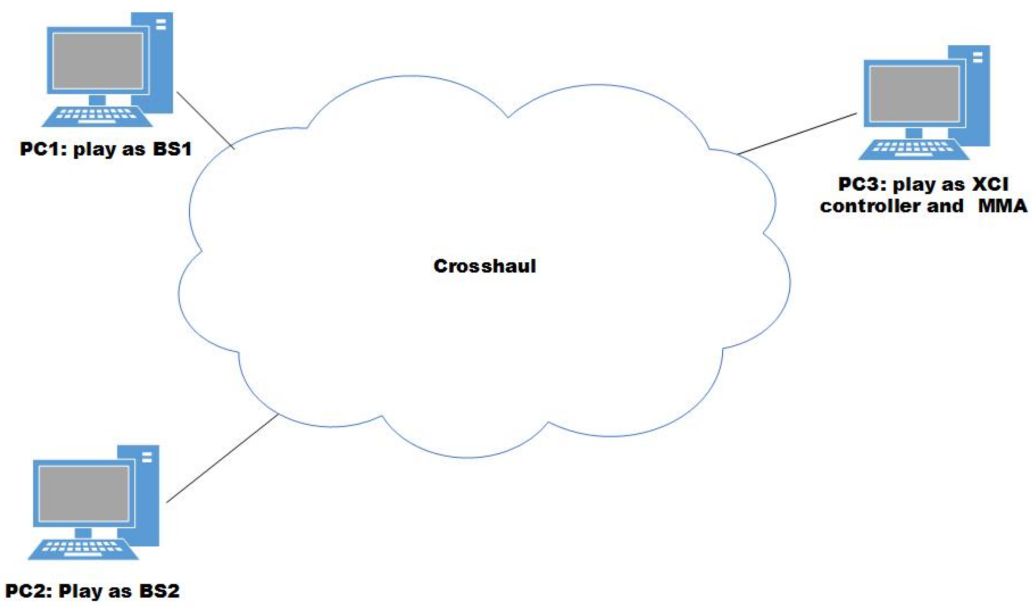
Test description		
Step #	Step description and expected results	Status
1.	**Description:** Generate network topology shown in the above. PC 1 acts as a packet generator to generate different handover execution and completion control packets. OF switch 1 is controlled by on-board XCI which has an NBI interface with MMA. Two outbound gateways are generated to forward the packet into shared transport network. **Expected Results:** Generate a handover process in the simple network topology and perform handover processes in parallel	
2.	**Description:** PC 1 generates a mobile packet and sends it to OF switch 1. OF switch 1 forwards it to on-board XCI controller. On-board XCI extracts the payload of this packet and sends it to Packet Inspection module of MMA.	

	After inspection, discover handover is existed and perform the handover algorithm. **Expected Results:** Perform parallel handover processes.	

Table 56: MMA HST test card 3

Test Algorithm	ITRI_MMA _03	Execution Status	Testing
Test Name	Handover testing in Crosshaul		
Objectives	Run both modules (Packet inspection and handover parallel process over complicated topology- close to the real testbed environment)		
Related Use Cases	Vehicle Mobility use cases		
Responsible	ITRI		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	MMA algorithm will be used to depict the handover reduction in emulated testbed environment



Additional Algorithms	Handover reduction in Crosshaul

Information	

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Generate network topology shown in the above where PC3 acts as XCI controller and MMA, PC1 and PC2 act as BS1 and BS2 **Expected Results:** Generate a simple shared transport network topology with two BSs.	
2.	**Description:** Pre-configuring flow tables for each XFEs in shared transport network to handle the handover request and ignite the parallel handover algorithm between BS1 and BS2. **Expected Results:** Each XFE used in shared transport network can process different handover packets to corresponding core networks.	

10.3 Resource Management Application

The purpose of this section is to provide a description of the validation objectives for the services provided by the Resource Management Application (RMA) to the different entities which form the 5G-Crosshaul XCI or to other Crosshaul applications. In general, RMA is the application which is deputed to manage network resources in conjunction with storage and compute availability in the XPU nodes. Referring to the context of 5G-Crosshaul, the networked entities which are taken into consideration in this section are XPFEs. Therefore, RMA shall provide different specialized services which are necessary to manage the resources available the 5G-Crosshaul network at any given time and geographical location. RMA shall also manage the different transmission technologies which are part of the Crosshaul data plane. Indeed, different utilization can be based on network load and type of available technologies (optical fibre, mmWave, copper, etc.) composing the intelligent transport network. To carry on with these tasks, RMA needs to acquire different type of information from the XCI interfaces in order to perform different services which can be requested to this application. Services are computed and the output then passed to the requesting entity.

10.3.1 Functional validation objectives

The RMA shall provide three types of services: Path Computation (PC) service, Path Computation-Virtual Network Functions Placement (PC-VNFP) service, and Path Computation-BS functional decomposition (PC-BS-split). The first refers to the service provided by this application in computing the optimal path for network flows based on specific inputs such as the identifier of a tenant network, the source and destination node addresses, as well as the bandwidth and latency demands of the flows. The PC service computes the optimal path trying to optimize some specific objective function $U(\cdot)$. In this endeavour RMA shall mainly deal with the network resources and technology types available within the 5G-Croshaul network. On the other hand, the PC-VNFP service stands for a more challenging task performed by this application: optimal path computation together with the placement of the VNFs on XPU nodes. Finally, PC-BS-split service is in charge of jointly maximizing the degree of BS centralization while routing flows between RRHs and XPUs. Maximizing the degree of BS centralization refers to offloading as much BS functionality into XPUs as possible, given networking constraints.

Besides the inputs already mentioned for the PC service, in this case the RMA necessitates also the list of VNFs to place in XPU nodes and the Forwarding Graph (FG) information which a flow requires in order to fulfil the specific service chain. RMA shall provide these services to any entity in the 5G-Crosshaul network which is requesting them, including any other application or module of the XCI. For the purpose of providing the services mentioned above, RMA ought to achieve the following objectives:

- **VNF placement:** Placement of services on XPU nodes in such a way that service requests are fulfilled and the objective function $U(\cdot)$ is optimized. To achieve objective function optimisation, each XPU node is equipped with a number of cores, with one core fully dedicated to one virtual machine (VM), and a VM hosting one VNF. In addition, it is also assumed that one core can serve up to a certain flow demand which, if exceeded, would require either VM scale-up or creation of a new VM for that VNF, within the same or in another XPU;
- **Flow allocation:** compute the optimal path for a flow in such a way that the FG is fulfilled, and the objective function $U(\cdot)$ is optimized.

Since 5G-Crosshaul includes the possibility of deploying VNFs on XPU nodes and connecting XPFEs elements through different transmission technologies, the optimization problem was formulated accordingly. In particular, the objective of RMA to perform PC and PC-VNFP services consists in the following minimization:

$$U = \min\left(C_{VNF} + C_f + C_d\right) \quad , \qquad (2)$$

where C_{VNF} is the cost associated to deploying a VNF over an XPU node; C_f denotes instead a fixed cost associated to a transmission technology and C_d is a dynamic cost associated to that technology. Costs are introduced here as a penalty that the system incurs in case of making specific decisions. Overall cost minimization clearly yields the optimal solution. All costs are unit-less and serve the purpose of describing the differences between selected transmission technologies.

Conceptually, the different cost components find the follow physical explanation:

- **VNF deployment cost:** Fixed cost paid when deploying VNFs on XPU nodes.
- **Fixed technology cost:** Cost associated to using a transmission technology. Particularly, a wireless transmission technology can reasonably exhibit lower fixed cost, thinking for instance to reflect simpler and quicker installation process. On the other hand, a fixed transmission technology would require larger initial effort for laying down the necessary infrastructure.
- **Dynamic technology cost:** Cost associated to the use of a specific transmission technology. This is the unit of cost paid by each flow for using such technology, and a flow with higher bandwidth demand shall incur in larger cost. Moreover, a fixed transmission technology shall exhibit in general a lower dynamic cost, which is an

expression of larger available bandwidth and higher reliability than a wireless transmission technology counterpart.

The objective of computing the optimal path for each flow seeks to minimize the overall cost U, when the cost of each flow is given by the summation of fixed plus dynamic costs.

The objective of minimizing the overall cost function is a possible way of approaching the problem of managing network resources in conjunction with compute resources. In this case the objective is that of using resources as efficiently as possible. In other words, the objective translates directly into the possibility of reusing those XPFEs and XPUs which have been already used by flows already allocated by the RMA algorithm. Only when the capacity of used links and nodes is exceeded, other technological edges and computing nodes will be used. This allows a less fragmented, and thus more efficient, use of resources. Resources which are not used can be conveniently put into a low energy state and reactivated when necessary (e.g. to cope with subsequent overloading conditions).

10.3.2 Roadmap

10.3.2.1 RMA algorithm for PC-VNFP service

Table 57 below shows the roadmap for implementing and validating the RMA algorithm responsible to deliver the PC-VNFP service.

Table 57: RMA algorithm for PC-VNFP roadmap

Roadmap		
Features	Description	Timeline
All Flow Allocation	RMA algorithm for simultaneous allocation of resources to a set F of flows, with each flow having different demand in a dual technology network, i.e., Ethernet and mmWave. Each flow has to fulfill a specific demand FG, or in other words that different services deployed on XPUs have to be executed in a specific order.	Oct. 2016
Flow-by-Flow Allocation	RMA algorithm for allocating resources to flow f, $\forall f \in F$, where F denotes the total set of flows to allocate. Flows have different demands and are allocated one by one updating residual network resources. Each flow has to fulfill a specific demand and FG, or in other words that different services deployed on XPUs have to be executed in a specific order.	Oct. 2016
Heuristic for Flow Allocation	Suboptimal solution to both All Flows Allocation, and Flow-by-Flow Allocation to run in polynomial time.	First Quarter 2017

10.3.2.2 RMA algorithm for PC-BS-split service

Table 58 below shows the roadmap for implementing and validating the RMA algorithm responsible to deliver the PC-BS-split service.

Table 58: RMA algorithm for PC-BS-split: roadmap

Roadmap

Features	Description	Timeline
Branch-and-bound backtracking algorithm	Nearly-optimal algorithm to maximize the amount of BS centralization while setting paths between RRHs and XPUs	Oct. 2016
Greedy algorithm	Polynomial algorithm to handle large-scale systems	Oct. 2016
Proof of concept	Experimental proof-of-concept comprised of software switches, NEC iPASOLINK E-band and microwave switches	First Quarter 2017

10.3.3 Validation environment

10.3.3.1 RMA algorithm for PC-VNFP service

Modelling mm-Wave links

Transmission at mm-Wave are known to exhibit quite a different behaviour if compared to radio transmissions that occur at lower frequency. Commonly they take place around 60 GHz frequency but also other options such as 28 GHz and 73 GHz have been considered as candidate. Communications at mm-Wave must face the higher attenuation caused by the absorption of the atmospheric oxygen, besides other propagation effects. Moreover, mm-Wave communications suffer from a very high attenuation which typically confines them to short-range (10 m or even shorter) communications. The typical way to overcome such severe attenuation consists in using highly directional antennas both in transmission and reception (i.e. directional communications). For these reasons, the capability of mm-Wave to diffract is very limited and consequently transmissions from source to destinations can be easily blocked, unlike lower frequencies where the radio environment is characterised by a rich scattering which gives rise to a very rich multi-path fading effect in overall. Furthermore, reflections of radio waves are exacerbated in mm-Wave communications, causing few reflected paths that can be almost as strong as the Line-of-Sight (LOS) component. Recent advances in mm-Wave technology and antenna technology have made possible to sue mm-Wave communications also to travel across longer propagation distances (few hundreds of meters) for point-to-point communications. One further clear advantage of using mm-Wave communications is the possibility of an extremely small form factor for antennas (comparable to the wave length of the transmission).

Several studies have been conducted in order to characterise the propagation and channel impulse response of mm-Wave communications, under different propagation environments. Notably some experimental, as well as analytical studies can be found in [14] [15] [16]. Relying on those research works, the characterisation of the propagation of mm-Wave transmissions can be done as shown in Table 59 below.

Table 59: Characterisation of propagation for mm-Wave transmissions occurring at different centre frequency [16]

Variable	Model	Model parameter Values	
		28 GHz	73 GHz

Omnidirectional path loss, PL	$\text{PL} = \beta + 10\,\alpha\, log_{10}(d) + \varepsilon$	$\beta = 72.0$, $\alpha = 2.92$	$\beta = 86.6$, $\alpha = 2.45$
Lognormal shadowing, ξ	$\xi \sim N(0, \sigma^2)$	$\sigma = 8.7\ dB$	$\sigma = 8.0\ dB$
Number of clusters, K	$K \sim \max\{Poisson(\lambda_c), 1\}$	$\lambda_c = 1.8$	$\lambda_c = 1.9$
Cluster power fraction	It follows a cluster arrival process with coefficients $\gamma_1,\ldots,\gamma_K$ for the distribution of power in the Kth cluster	$r_\tau = 2.8$, $\zeta = 4.0$	$r_\tau = 3.0$, $\zeta = 4.0$
BS cluster rms angular spread	σ is exponentially distributed $E(\sigma) = \lambda_c^{-1}$	Horiz $\lambda_c^{-1} = 10.5°$ Vert $\lambda_c^{-1} = 0°(*)$	Horiz $\lambda_c^{-1} = 10.5°$ Vert $\lambda_c^{-1} = 0°(*)$
UE rms angular spread	σ is exponentially distributed $E(\sigma) = \lambda_c^{-1}$	Horiz $\lambda_c^{-1} = 15.5°$ Vert $\lambda_c^{-1} = 6.0°(*)$	Horiz $\lambda_c^{-1} = 15.5°$ Vert $\lambda_c^{-1} = 3.5°(*)$

Table 59 shows the path-loss (PL) function of mm-Wave transmissions that occur at different frequency, and in specific at 28 GHz and 73 GHz. The characterisation is done assuming different path-loss exponents (β) and loss at one meter (α). The experimental work done in [16], allows also to use the 3GPP propagation model for the centre frequency around 2.5 GHz enriched with additional losses to model propagation properly, as representative for mm-Wave transmissions.

Relying instead of [19], the analytical work developed can be proficiently used to characterise the probability of success P_s as follows

$$p_s = 1 - \exp\left(-\lambda M_a\left(\frac{P_b G_{\max}}{\tau\sigma_N^2}\right)\right), \qquad (6)$$

where $M_a(t) = \frac{\Lambda_a((0,t])}{\lambda}$. Importantly the parameter $\Lambda_a((0,t])$ is written in the following way

$$\Lambda_a((0,t]) = \lambda\pi C\Bigg\{D^2\left[Q\left(\frac{\ln(D^{\alpha_l}/t) - m_l}{\sigma_l}\right) - Q\left(\frac{\ln(D^{\alpha_n}/t) - m_n}{\sigma_n}\right)\right] + t^{2/\alpha_l}\exp\left(2\frac{\sigma_l^2}{\alpha_l^2} + 2\frac{m_l}{\alpha_l}\right) \times Q\left(\frac{\sigma_l^2(2/\alpha_l) - \ln(D^{\alpha_l}/t) + m_l}{\sigma_l}\right) + t^{2/\alpha_n}\exp\left(2\frac{\sigma_n^2}{\alpha_n^2} + 2\frac{m_n}{\alpha_n}\right)\left[\frac{1}{C} - Q\left(\frac{\sigma_n^2(2/\alpha_n) - \ln(D^{\alpha_n}/t) + m_n}{\sigma_n}\right)\right]\Bigg\}$$

where Λ_a denotes the intensity measure for the corresponding Poisson Point Process (PPP) of the mm-Wave links that incur in a path-loss greater than the threshold t when the path-loss distribution is proven to follow an exponential distribution in [17]. In addition, λ denotes the density of mm-Wave links, D is the separating distance between transmitter-receiver pair, C the fractional LoS area in the model developed, Q() denotes the standard Gaussian Q-function, α_l, α_n are the path-loss exponents for LoS and NLoS conditions respectively, $m_j = -0.1\beta_j \ln 10$ and $\sigma_j = 0.1\xi_j$ are model parameters for $j=l$, n, and ξ_j is the corresponding path-loss standard deviation.

Results for both path-loss and probability of success are provided in Chapter 0.

Validation environment

The validation environment to test the RMA algorithm that performs the PC-VNFP service relies on the formulation of an equivalent Integer Linear Programming (ILP) problem. The solution to the ILP formulation is made in terms of the minimization of an objective function subject to (s.t.) several different constraints that stands for the PC-VNFP service provided by the RMA, amongst others. The advantage of solving the equivalent ILP formulation consists in determining the best possible solution, if it exists. The main disadvantage incurred is a potentially long execution time to solve the ILP. Simulations are currently in progress using Matlab simulator, and in particular using the Matlab optimization toolbox for solving Mixed ILP (MILP), with the latter a generalisation of ILP problems. Matlab simulation environment offers a great deal of flexibility in modelling different network topologies, modelling technology types and performing scalability tests.

The necessary inputs to compute the requested service are assumed to be made available through the NBI of the XCI (i.e. REST interfaces) or directly from the involved SDN controllers, and include:

- Topology and inventory for both switching and computing nodes,
- Network state changes and any node failure,
- Collection of network statistics and performance indicators.

In this validation environment, a random topology network is simulated in order to take into account a deployment of switching and computing nodes as generic as possible. Each XPFE element is assumed to embed a dual type of transmission technology to connect to another switching element. Particularly, gigabit Ethernet and mmWave can be selected in the optimization problem.

In this validation environment, each flow request consists of the tuple: (a) start node ID, (b) destination node, ID (c) bandwidth demand, (d) latency demand and (e) forwarding graph. The service chain of a flow indicates the requirement of a set of services in the specified order. The RMA algorithm determines the optimum path from source to destination node, which can fulfil the bandwidth and latency demands, as well as the constraint to traverse in order services inside the FG (e.g. flow $f \in F$ needs to go through the FG: $s_1 \rightarrow s_3 \rightarrow s_8$ before reaching the destination node). To accomplish this objective, the RMA algorithm can select any of the technological options available for transmission (i.e. either Ethernet or mmWave).

In this validation environment, flow generation is performed through a random process. For any given number of nodes and flows to allocate, flow ingress (source node) and flow egress (destination node) are selected at random, as well as the FG for each flow. Network topology remains unchanged as well as the source-destination pair and FG but simulations are repeated randomizing the order in which flows are allocated in the network. The randomization leads to different selection of computing nodes where to deploy services and switching elements to traverse. In other words, this yields different allocation strategy results.

Three types of tests performed in this validation environment include the simultaneous allocation of flows (i.e. All Flows Allocation), consecutive allocation of flows (Flow-by-Flow Allocation), and the testing of a suitable heuristic for allocating flows to run in polynomial time.

The objectives of the validation of the algorithms are to evaluate the effectiveness of the RMA algorithms in finding the efficient placement of network services and appropriate routing of flows that ensures minimum use of network resources and cost.

To this end we evaluate the following performance metrics obtained as a result of the experimental tests of the RMA algorithms:

- ILP execution time
- Percentage of links used out of the total available, in overall and per technology

The initial results of the performance evaluation of RMA algorithms look very promising.

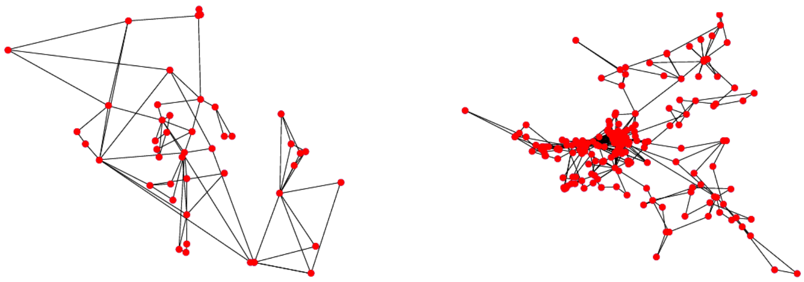
10.3.3.2 RMA algorithm for PC-BS-split service

The validation environment is twofold. First, in order to get statistically meaningful insights, we test our algorithms over a large set of topologies and extract three parameters for each simulation: degree of centralization, whether it is a feasible solution (no centralization is provided otherwise) and its elapsed time. The first and second sets of topologies are based on two backhaul networks of existing operators in Romania and Switzerland.

Table 60: Ethernet-based link profiles (proc. delay = 5 microseconds, packet size = 1500 Bytes)

Technology	Bandwidth (Gbps)	Prop. Delay (μs)	Distance (km)
mmWave (60-80 GHz)	0.9, 1.25, 1.5, 2, 3, 4, 8	1-20	0.3-6
μWave (6-60 GHz)	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.25, 1.5, 2	1-100	0.3-30
Copper (1000/10G/40GBASE-T)	1, 10, 40	0.05-0.5, 0.275, 0.15	0.001-0.1, 0.055, 0.03
SMF fiber @ 1310 nm (1000, 10G, 40, 100GBASE-EX, LR, LR-4	1, 10, 40, 1000	1-200, 50, 50, 50	0.2-40, 10, 10, 10
SMF fiber @ 1550 nm (1000, 10G, 40, 100GBASE-ZX, ER, ER-4)	1, 10, 40, 1000	1-350, 200, 200, 200	0.2-70, 40, 40, 40
TbE (*under development)	200, 400	1-50	0.2-10

We create semi-random scenarios based on the backhaul topology of two operators located in Romania and Switzerland. In this case, we know the distance between switching nodes, links connecting them and their capacities. Figure 69 illustrates their representative graphs. Based on this, we choose the link profiles from Table 60 that match better the given capacities. Note that, following NGFI, we only consider Ethernet over different PHYs. We assume classic store-and-forward switching which induces a per-hop latency equal to propagation delay – distance(m)/200s for copper, distance(m)/300 s the rest–plus serialization delay (bits/rate). Finally, we randomly pick nodes to be RRHs and XPUs.



(a) Romania topology. (b) Switzerland topology.

Figure 69: Snapshot of real topologies

The third set of topologies consists of purely random scenarios with a tree structure. To generate topologies with a tree structure, first we generate a set of XPUs connected in a ring. These will be located at the root of the trees. Second, hanging from each XPU, we create a pure random tree using independent Poisson processes to model the number of levels of the tree and nodes per level. The leafs of the tree correspond to RRHs. For each level of each tree, we randomly add a backup link with the upper layer of a neighbouring tree, to add additional degrees of freedom for routing, following another Poisson process. Once we have the graph structure, we randomize the profile of all the links and their length, using the same assumptions as in the previous scenarios (links in Table 60). For each link and topology, we randomly pick one profile depending on its proximity to an XPU or an RRH. To this aim, we group links based on its proximity to an XPU and assign them the same link profile. Additionally, we assign high-capacity profiles to links closer to XPUs with more probability, to capture the fact that aggregation segments typically provide higher capacities. Finally, RRH clustering is done as in the previous scenarios.

The last set of topologies consists of arbitrary Waxman random graphs, based on the Erdos-Renyi random graph model, which is popular to evaluate realistic backhaul topologies. The second environment where we will validate the PC-BS-split service corresponds to an experimental proof of concept comprised of a set of software (virtualized) switches, NEC iPASOLINK E-band and NEC iPASOLINK microwave switches. Using this environment, our algorithms should be capable of adapting the degree of BS centralization to changing wireless condition which will be emulated with attenuators in the wireless links.

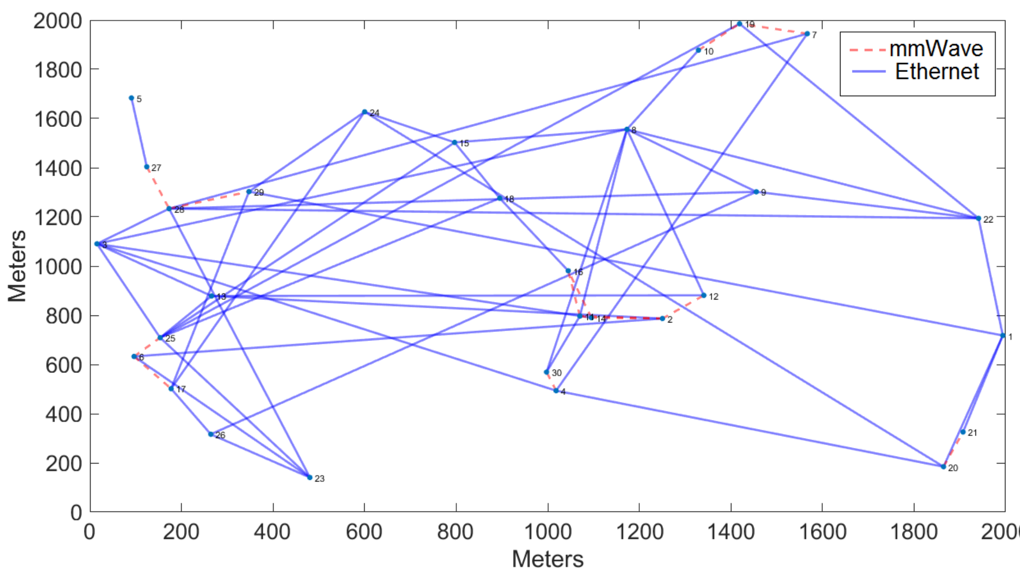
10.3.4 Validation procedure

10.3.4.1 RMA algorithm for PC-VNFP service

The algorithms which are under evaluation are described in in the following tables.

Test Algorithm	CREATE-NET_PCVNFP _01	Execution Status	Running
Test Name	Flow-by-Flow Allocation		
Objectives	Optimise use of network and compute resources through minimisation of the cost function for voice and video traffics		
Related Use Cases	Covers several use cases relevant to 5G-Crosshaul [1]		
Responsible	CREATE-NET		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	PC-VNFP Service for consecutive allocation of flows optimizing the use of network and compute resources.
Test environment topology: Random with uniform placement of nodes in the area.	



Additional Algorithms	- Demand of each flow - Ingress-Egress node pair for each flow

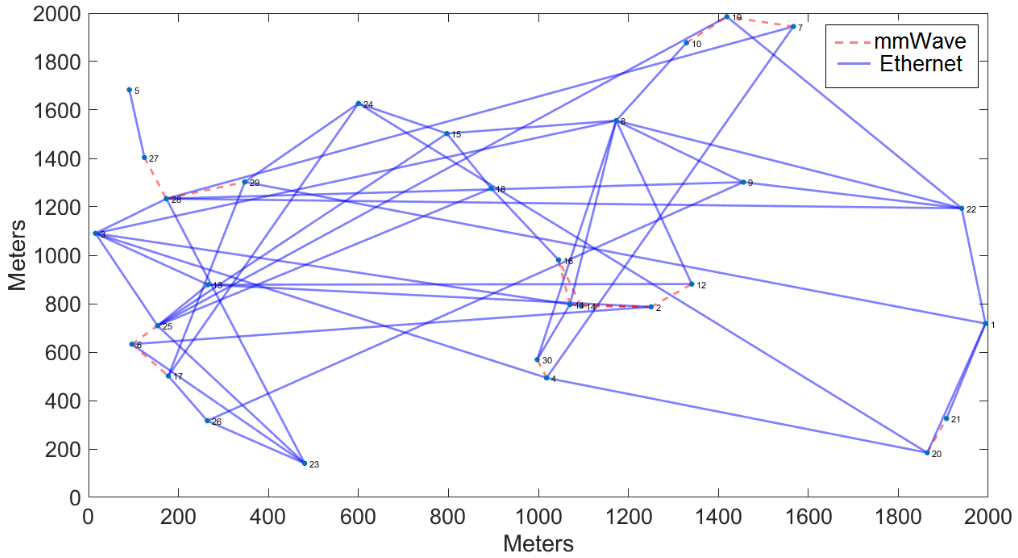
Information	- Forwarding Graph to be fulfilled by each flow

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Create a random topology network with dual technology option for connecting switching elements. Decide the fraction of switching and computing nodes out of the total available. **Expected Results:** Connected random network graph. Connectivity is achieved through mmWave and Ethernet links. A switching element can be connected with both technologies based on coverage distance constraint of mmWave transmitters. In particular, a mmWave link is added upon whether distance between switching elements falls within a certain fixed distance value. Network wide connectivity is achieved through the deployment of Ethernet links, since mmWave links alone cannot guarantee full connectivity condition.	
2.	**Description:** Assignment of the number of cores to computing nodes, and the maximum capacity demand created by incoming flows that each core can serve. Furthermore, one core is assigned to one service. **Expected Results:** Create the computing nodes	
3.	**Description:** Generation of any given number of flows with randomized ingress-egress node pair. The total number is divided in video and voice traffic flows. To each flow is assigned a <bandwidth, latency> demand and a FG to be fulfilled. **Expected Results:** Generation of the network load.	
4.	**Description:** Starting with the connectivity graph dummy switching nodes are added between those XPFEs which can connect to other XPFEs with either Ethernet or mmWave technologies. **Expected Results:** Create an equivalent expanded network graph in which a switching element is connected to another through a single technology option.	

	Technology link capacity is maintained constant even after introducing the dummy nodes.	
5.	**Description:** Consecutive allocation of the flows (Flow-by-Flow) in the network according to their demand and FG constraint. **Expected Results:** Minimization of used resources in the network for efficient utilisation.	
6.	**Description:** 1) Randomisation of the order of execution of flow allocations for each value of the number of nodes and flows 2) Varying the total number of flows to allocate for fixed number of nodes in the network 3) Varying the total number of nodes in the network for a fixed number of flows to allocate **Expected Results:** Obtain the utilisation of network and compute resources under different simulation conditions.	

Test Algorithm	CREATE-NET_PCVNFP _02	Execution Status	Running
Test Name	ALL Flow Allocation		
Objectives	Optimise use of network and compute resources through minimisation of the cost function for voice and video traffics		
Related Use Cases	Covers several use cases relevant to 5G-Crosshaul [1]		
Responsible	CREATE-NET		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	PC-VNFP Service for simultaneous allocation of flows optimizing the use of network and compute resources.
Test environment topology: Random with uniform placement of nodes in the area.	



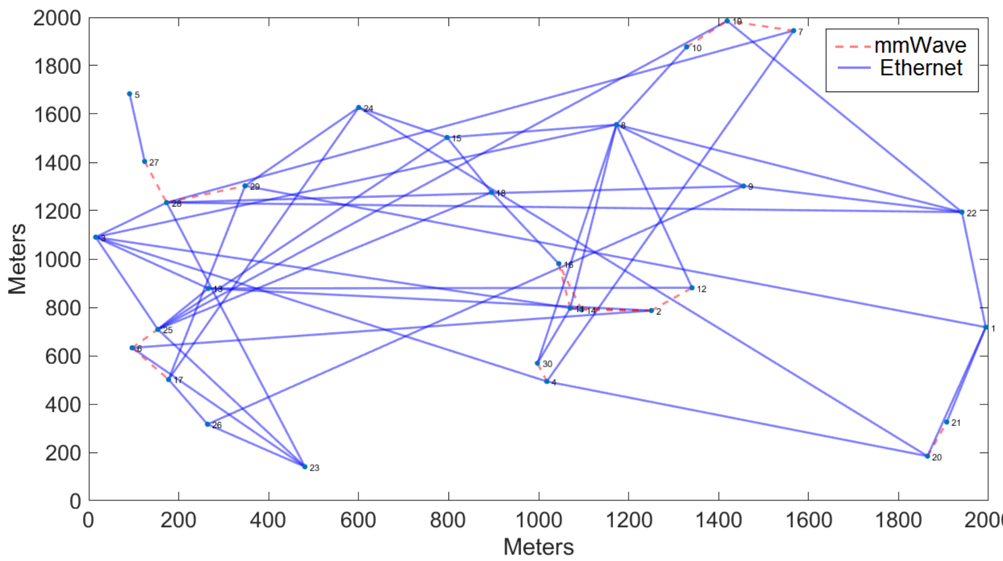
Additional Algorithms Information	- Demand of each flow - Ingress-Egress node pair for each flow - Forwarding Graph to be fulfilled by all flows

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Create a random topology network with dual technology option for connecting switching elements. Decide the fraction of switching and computing nodes out of the total available. **Expected Results:** Connected random network graph. Connectivity is achieved through mmWave and Ethernet links. A switching element can be connected with both technologies based on coverage distance constraint of mmWave transmitters. In particular, a mmWave link is added upon whether distance between switching elements falls within a certain fixed distance value. Network wide connectivity is achieved through the deployment of Ethernet links, since mmWave links alone cannot guarantee full connectivity condition.	
2.	**Description:** Assignment of the number of cores to computing nodes, and the maximum capacity demand created by incoming flows that each core can serve. Furthermore, one core is assigned to one service. **Expected Results:**	

	Create the computing nodes	
3.	**Description:** Generation of any given number of flows with randomized ingress-egress node pair. The total number is divided in video and voice flows. To each flow is assigned a <bandwidth, latency> demand and a FG to be fulfilled. **Expected Results:** Generation of the network load.	
4.	**Description:** Starting with the connectivity graph dummy switching nodes are added between those XPFEs which can connect to other XPFEs with either Ethernet or mmWave technologies. **Expected Results:** Create an equivalent expanded network graph in which a switching element is connected to another through a single technology option. Technology link capacity is maintained constant even after introducing the dummy nodes.	
5.	**Description:** Simultaneous allocation of the flows (All Flows) in the network according to their demand and FG constraint; randomisation of the order of execution of flow allocations. **Expected Results:** Minimization of used resources in the network for efficient utilisation.	
6.	**Description:** 4) Randomisation of the order of execution of flow allocations for each value of the number of nodes and flows 5) Varying the total number of flows to allocate for fixed number of nodes in the network 6) Varying the total number of nodes in the network for a fixed number of flows to allocate **Expected Results:** Obtain the utilisation of network and compute resources under different simulation conditions.	

Test Algorithm	CREATE-NET_PCVNFP _03	Execution Status	Planned
Test Name	Heuristic for Flow Allocation		

Objectives	Sub-optimal use of network and compute resources through minimisation of the cost function for voice and video traffics to run in polynomial time
Related Use Cases	Covers several use cases relevant to 5G-Crosshaul [1]
Responsible	CREATE-NET
Related Test Cards	N/A (not applicable)

SUT and topology	
SUT	Heuristic algorithm for the PC-VNFP Service to obtain sub-optimal allocation of flows in polynomial time optimizing the use of network and compute resources.
Test environment topology: Random with uniform placement of nodes in the area. 	
Additional Algorithms Information	• Demand of each flow • Ingress-Egress node pair for each flow • Forwarding Graph to be fulfilled by each flow

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Create a random topology network with dual technology option for	

	connecting switching elements. Decide the fraction of switching and computing nodes out of the total available. **Expected Results:** Connected random network graph. Connectivity is achieved through mmWave and Ethernet links. A switching element can be connected with both technologies based on coverage distance constraint of mmWave transmitters. In particular, a mmWave link is added upon whether distance between switching elements falls within a certain fixed distance value. Network wide connectivity is achieved through the deployment of Ethernet links, since mmWave links alone cannot guarantee full connectivity condition.	
2.	**Description:** Assignment of the number of cores to computing nodes, and the maximum capacity demand created by incoming flows that each core can serve. Furthermore, one core is assigned to one service. **Expected Results:** Create the computing nodes.	
3.	**Description:** Generation of any given number of flows with randomized ingress-egress node pair. The total number is divided in video and voice flows. To each flow is assigned a <bandwidth, latency> demand and a FG to be fulfilled. **Expected Results:** Generation of the network load.	
4.	**Description:** Starting with the connectivity graph dummy switching nodes are added between those XPFEs which can connect to other XPFEs with either Ethernet or mmWave technologies. **Expected Results:** Create an equivalent expanded network graph in which a switching element is connected to another through a single technology option. Technology link capacity is maintained constant even after introducing the dummy nodes.	
5.	**Description:** Heuristic algorithm based on *k*-shortest path computation fulfilling demands and forwarding graphs of the flows; randomisation of the order of execution of flow allocations. **Expected Results:** Sub-optimal solution executed in polynomial time to resource utilisation in the network.	

6.	**Description:** 7) Randomisation of the order of execution of flow allocations for each value of the number of nodes and flows 8) Varying the total number of flows to allocate for fixed number of nodes in the network 9) Varying the total number of nodes in the network for a fixed number of flows to allocate **Expected Results:** Obtain the utilisation of network and compute resources under different simulation conditions.	

10.3.4.2 RMA algorithm for PC-BS-split service

The algorithms which are under evaluation are described in the following tables.

Table 61: RMA algorithm for PC-BS-split service test card 1

Test Algorithm	NEC_PCBSSPLIT_01	Execution Status	Running
Test Name	Branch-and-bound backtracking algorithm (BBB)		
Objectives	Maximizing the degree of BS centralization using a nearly-optimal combinatorial algorithm based on branch-and-bound.		
Related Use Cases	Covers several use cases relevant to 5G-Crosshaul [3]		
Responsible	NEC		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Branch-and-bound backtracking algorithm (BBB)
Test environment topology (explained above): - Semi-random topologies based on a backhaul topology of an operator in Romania and Switzerland. - Random topologies with a tree structure - Random topologies based on Waxman algorithm	
Additional	1) Degree of centralization

Algorithms Information	2) Paths between RRHs and XPUs 3) Running time 4) Feasibility of the allocation

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Test of BBB over a large set of semi-random topologies based on Romanian and Swiss backhaul topology, as explained earlier. **Expected Results:** BBB should maximise the degree of centralization of BS functions while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to BBB.	
2.	**Description:** Test of BBB over a large set of pure random topologies based on tree structures, as explained earlier. **Expected Results:** BBB should maximise the degree of centralization of BS functions while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to BBB.	
3.	**Description:** Test of BBB over a large set of pure random topologies based on Waxman structures, as explained earlier. **Expected Results:** BBB should maximise the degree of centralization of BS functions while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to BBB.	

Table 62: RMA algorithm for PC-BS-split service test card 2

Test Algorithm	NEC_PCBSSPLIT_02	Execution Status	Running
Test Name	Greedy based algorithm with Greedy routing (GA-GR)		
Objectives	Maximising the degree of BS centralization using a greedy combinatorial algorithm based greedy routing.		

Related Use Cases	Covers several use cases relevant to 5G-Crosshaul
Responsible	NEC
Related Test Cards	N/A (not applicable)

SUT and topology	
SUT	Greedy based algorithm with Greedy routing (GA-GR)
Test environment topology (explained above): - Semi-random topologies based on a backhaul topology of an operator in Romania and Switzerland. - Random topologies with a tree structure - Random topologies based on Waxman algorithm	
Additional Algorithms Information	1) Degree of centralization 2) Paths between RRHs and XPUs 3) Running time 4) Feasibility of the allocation

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Test of GA-GR over a large set of semi-random topologies based on Romanian and Swiss backhaul topology, as explained earlier. **Expected Results:** GA-GR should provide a similar degree of centralization of BS functions to BBB while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to GA-GR.	
2.	**Description:** Test of GA-GR over a large set of pure random topologies based on tree structures, as explained earlier. **Expected Results:**	

	GA-GR should provide a similar degree of centralization of BS functions to BBB while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to GA-GR.	
3.	**Description:** Test of BBB over a large set of pure random topologies based on Waxman structures, as explained earlier. **Expected Results:** GA-GR should provide a similar degree of centralization of BS functions to BBB while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to GA-GR.	

Table 63: RMA algorithm for PC-BS-split service test card 3

Test Algorithm	NEC_PCBSSPLIT_03	Execution Status	Running
Test Name	Greedy based algorithm with Randomized Rounding routing (GA-RR)		
Objectives	Maximising the degree of BS centralization using a greedy combinatorial algorithm based randomized rounding routing.		
Related Use Cases	Covers several use cases relevant to 5G-Crosshaul		
Responsible	NEC		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Greedy based algorithm with Randomized Rounding routing (GA-RR)
Test environment topology (explained above): - Semi-random topologies based on a backhaul topology of an operator in Romania and Switzerland. - Random topologies with a tree structure - Random topologies based on Waxman algorithm	

Additional Algorithms Information	1) Degree of centralization 2) Paths between RRHs and XPUs 3) Running time 4) Feasibility of the allocation

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Test of GA-RR over a large set of semi-random topologies based on Romanian and Swiss backhaul topology, as explained earlier. **Expected Results:** GA-RR should provide a similar degree of centralization of BS functions to BBB while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to GA-RR.	
2.	**Description:** Test of GA-GR over a large set of pure random topologies based on tree structures, as explained earlier. **Expected Results:** GA-RR should provide a similar degree of centralization of BS functions to BBB while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to GA-RR.	
3.	**Description:** Test of BBB over a large set of pure random topologies based on Waxman structures, as explained earlier. **Expected Results:** GA-RR should provide a similar degree of centralization of BS functions to BBB while respecting networking constraints. Benchmarking algorithms based on state of the art routing approaches will fail to find a feasible allocation at a high rate or will render low degree of centralization compared to GA-RR.	

10.3.5 Validation Scenarios and Initial Results

10.3.5.1 RMA algorithm for PC-VNFP service

This section presents the validation scenario and initial results obtained from *RMA algorithm for PC-VNFP service.*

For the simulation, as anticipated, the fixed cost of Ethernet is chosen to be higher than that of mmWave, but this is exactly the opposite for the dynamic cost. For dynamic cost, this is quantified whereby the link success probability $p_s \in [0,1]$. For Ethernet it is assumed $p_s^{(eth)} = 1$, whereas $p_s^{(mmwave)} \leq 1$. The dynamic cost function is computed as $C_d = 1/p_s$ irrespective of the selected technology.

The probability of success for mmWave links is obtained combining the work done in [Table 1, Rappaport] and [Eq. (2), Andrews]. The work done in [Rappaport] provides useful experimental data which are plugged in to the analytical model in [Andrews]. The combination allows to model accurately the signal-to-noise-ratio (SNR), assuming the case of no interference between proximate mmWave links, a hypothesis which is clearly not totally realistic. Moreover, the experimental values proposed in [Table 1, Rappaport] allows modeling propagation at different central frequency, respectively 2.5 GHz, 28 GHz and 73 GHz. The first propagation model at 2.5 GHz relies on a 3GPP model with additional losses to mode mmWave propagation. For the sake of solving the ILP problem with which the RMA algorithm can provide the PC-VNFP service it is assumed that $p_s^{(mmwave)}(\|u-v\|)$, with $\|\cdot\|$ denoting the Euclidean norm.

For evaluating the effective utilization of network resources as a result of solving the ILP formulation when flows are allocated in consecutive fashion, the following ratio is computed as follows

$$\eta = \frac{number\ of\ used\ links}{total\ number\ of\ links} \quad . \qquad (2)$$

The ratio η allows to quantify network resource utilization and is computed separately for each technology option connecting XPFEs and for the total number of used links irrespective of the technology selected by the algorithm.

Validation scenario & Initial results *for the Flow-by-Flow RMA algorithm*

This sub-section presents the validation scenario for the Flow-by-Flow RMA algorithms (test case - CREATE-NET_PCVNFP _01). Numerical values used to obtain initial results are shown in Table 64.

Table 64: Numerical parameters used to simulate the RMA algorithm for PC-VNFP

Parameter	**Value**
Geographical area	[2000 × 2000] m
Coverage radius of a mmWave transmitter	≤ 200 m
Total number of simulations for given number of flows and nodes	50
Total number of nodes (XPU+XPFE)	10≤ n≤30

Percentage of XPUs	30%
Percentage of XPFEs	70%
Number of flows	$5 \le f \le 100$
Percentage of Video flows	70%
Percentage of Audio flows	30%
Latency constraint for video traffic	<100 ms
Latency constraint for voice traffic	<10 ms
Fixed cost [Eth, mmwave]	[100, 1]
Dynamic cost	$1/p_s$
Capacity [Eth, mmwave]	[10000, 2000] Mbit/s

The preliminary validation result of the Flow-by-Flow RMA algorithms (test case - CREATE-NET_PCVNFP _01) is reported here. From Figure 71, it can be noticed that the probability of success for mmWave links decreases as expected very rapidly with the distance separating transmitter and receiver and with the increase of the central frequency. To show subsequent ILP results, it is used the 3GPP propagation model at 2.5 GHz.

Figure 72 (a) shows the average ILP execution time for 30 nodes in the network varying the number of flows. Interestingly, the execution time scales linearly with the number of flows. Figure 72 (b) shows instead the ILP execution time varying the number of nodes, and 100 flows to allocate. In this case instead the execution time scales exponentially with the number of nodes. This effect is due to the fact that the ILP must compute a larger number of possible combinations of possible paths in order find the optimum solution for each flow.

Figure 73 shows the utilization of different link technologies over the overall available links in the network, while increasing the total number of flows with a number of nodes in the graph equal to 30. The figure shows also the overall utilisation of links in the network upon the same conditions. It can be noticed that even in correspondence of 100 flows to allocate, the overall utilisation of network resources remains below 50%. This result, allows to preliminary conclude that the objective of avoiding fragmentation in resources utilisation has been achieved, compatibly with the compound demand of flows.

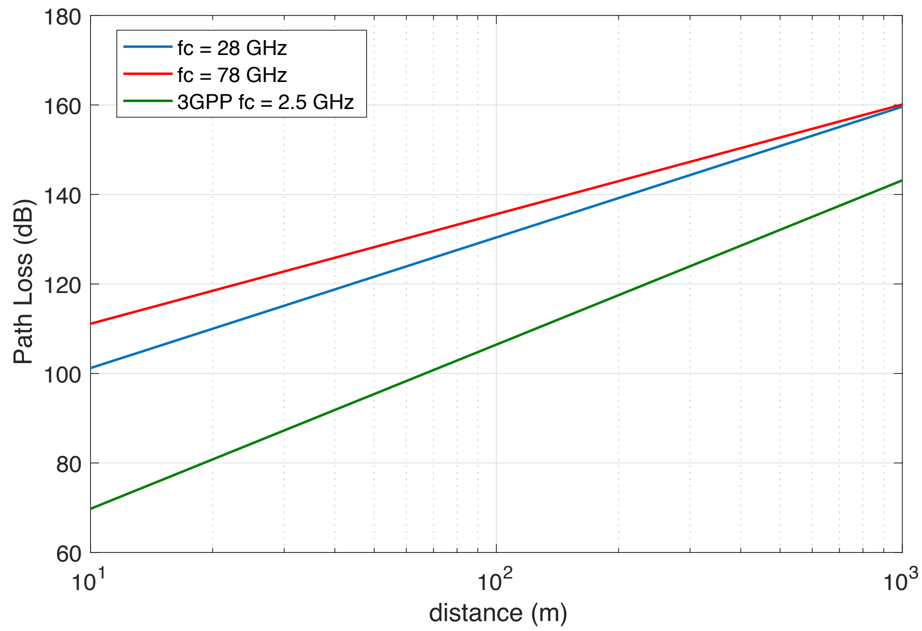


Figure 70: Path-loss of mm-Wave transmissions that occur at different frequency. The blue line denotes the propagation for the 28 GHz centre frequency; the red line for 73 GHz and the green line for the 3GPP equivalent model

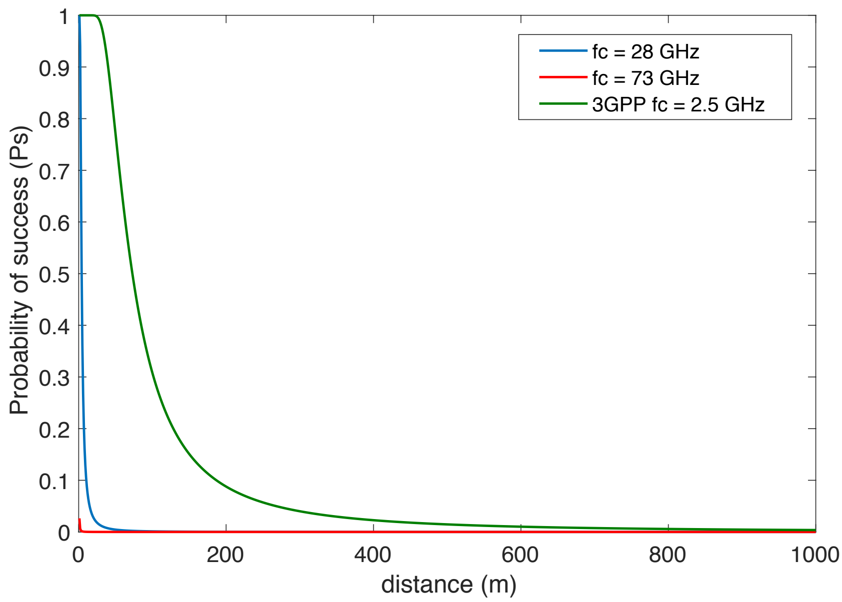
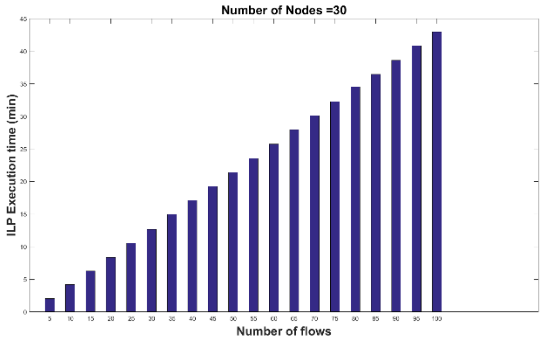
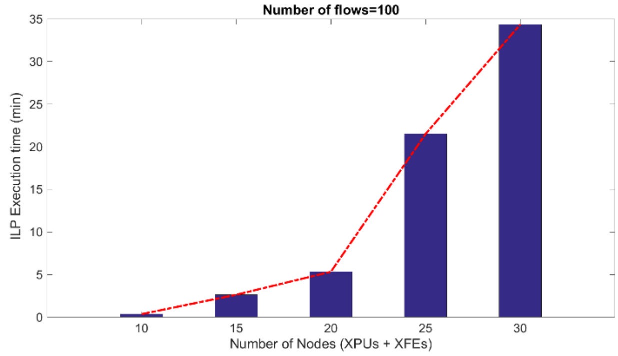


Figure 71: Probability of success for mmWave technology as a function of distance separation between transmitter and receiver. Curves are shown for different propagation models and different central frequency



(a)



(b)

Figure 72: ILP execution time for flow-by-flow allocation varying (a) the number of nodes and (b) the number of flows in the network

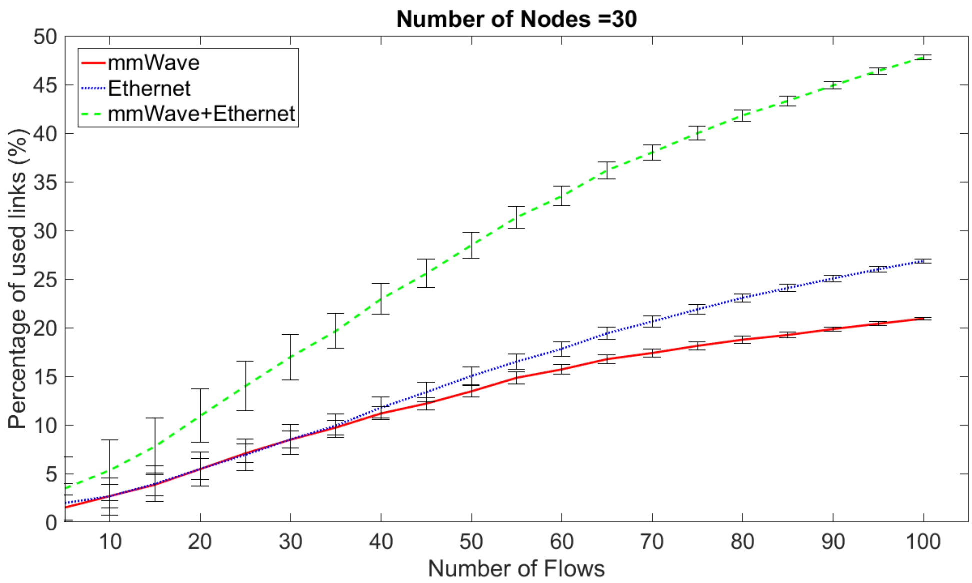


Figure 73: Percentage of the total number of links used in a network with 30 nodes (XPUs+XPFEs) deployed, while varying the number of flows to be allocated in the network on a Flow-by-Flow basis

Validation scenario ***for the ALL Flow Allocation RMA algorithm***

This sub-section presents the validation scenario for the ALL Flow Allocation RMA algorithms (test case - CREATE-NET_PCVNFP _02). Numerical values used to obtain initial results are almost identical as shown in Table 64, except in this case the number of flow, f used for the experiment is $5 \leq f \leq 65$. The preliminary validation result of the Flow-by-Flow RMA algorithms (test case - CREATE-NET_PCVNFP _02) is reported here.

Figure 74 shows the average ILP execution time for 25 nodes in the network varying the number of flows. The execution time increases with the increase of the number of nodes. This effect is due to the fact that the ILP must compute a larger number of possible combinations of possible paths in order find the optimum solution for each flow.

Figure 75 shows the utilization of different link technologies over the overall available links in the network, while increasing the total number of flows with a number of nodes in the graph equal to 15. The figure shows also the overall utilisation of links in the network upon the same conditions. It can be noticed that even in correspondence of 65 flows to allocate, the overall utilisation of network resources remains below 50%. This result, allows to preliminary conclude that the objective of avoiding fragmentation in resources utilisation has been achieved, compatibly with the compound demand of flows.

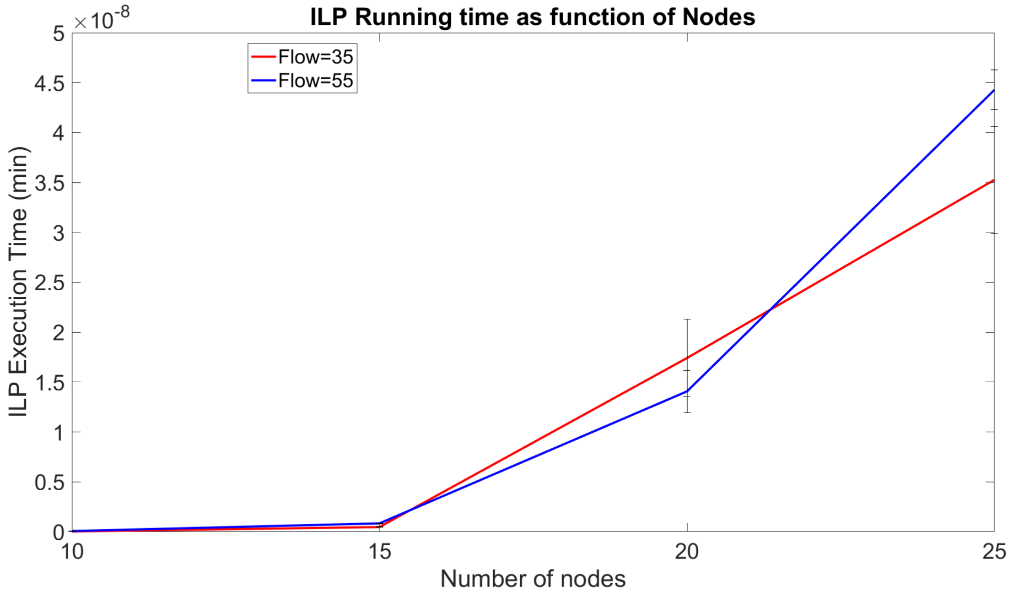


Figure 74: ILP execution time for All flow allocation varying the number of nodes

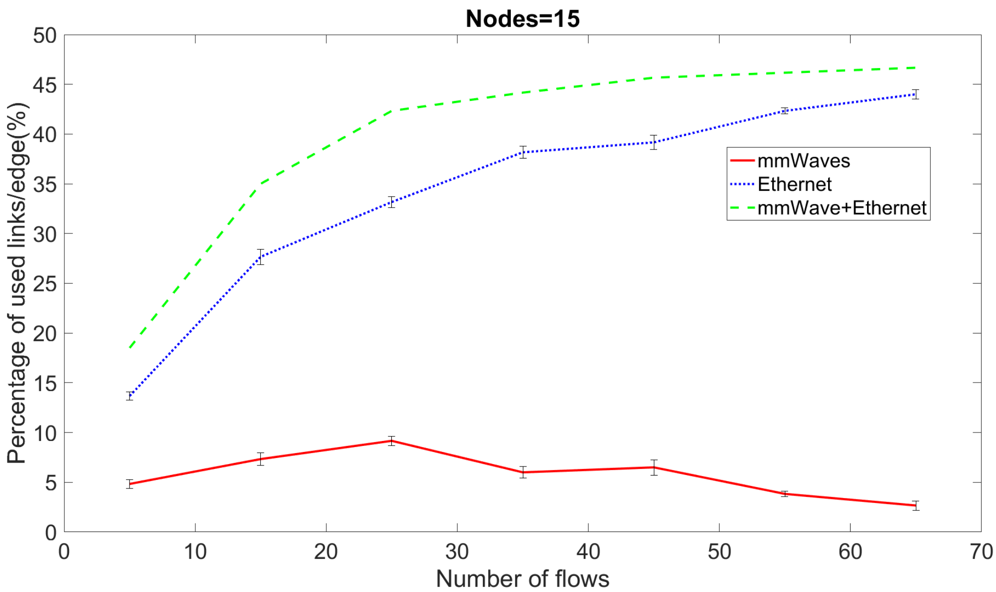


Figure 75: Percentage of the total number of links used in a network with 15 nodes (XPUs+XPFEs) deployed, while varying the number of flows to be allocated in the network on all Flow allocation basis

10.4 Energy Management and Monitoring Application

10.4.1 Functional validation objectives

The EMMA validation procedures are focused on the functional testing of the following main features:

- Monitoring of power consumption:
    - For the whole physical infrastructure, in an environment composed of XPUs and XPFEs based on Lagopus[1] software switches. Monitoring information are collected from the XCI SDN controller for the network elements and from the NFV MANO components (in particular the VIM) for the XPUs, using the XCI northbound interface;
    - For single instances of network connections or Network Services, through aggregation and elaboration of data related to physical devices, resource allocation and traffic/processing load;
    - For all the virtual instances assigned to single tenants, through aggregation of the previous data on a per-tenant basis.
- Configuration and management of energy-related policies.
- Automated configuration of the power states on network nodes (through the XCI SDN controller's NBI) and servers (through the XCI VIM's NBI).
- On-demand computation and provisioning of energy-efficient network paths, associated to a given tenant and with policy-driven characteristics (through the XCI SDN controller's NBI, operating over software switch based XPFEs).
- On demand computation and provisioning of energy efficient Network Services for vEPC, associated to a given tenant and with policy driven characteristics (through the XCI NFVO's NBI).

mmWave mesh case

In the case of mmWave mesh case, the following validation objectives are additionally considered since the mmWave mesh node plays a role of both XFE and (mmWave) access point (AP). Furthermore, two-tier network is assumed where the mmWave mesh network is overlaid on a legacy cellular network (macro cell) and both of them are in the same 5G-Crosshaul domain.

- Based on the traffic demands of users, initial mmWave nodes are activated to minimize energy consumption of the mmWave mesh network while satisfying traffic demands of users.
- Among the initially activated mmWave nodes, optimal network paths are calculated and provisioned to form mmWave mesh network with minimal energy consumption.
- If the traffic demands are not satisfied in a mmWave node, surrounding OFF state mmWave nodes are re-activated to a form multi-route multi-hop network until all traffic demands are satisfied.

10.4.2 Roadmap

Table 65: EMMA: roadmap

Roadmap		
Features	Description	Timeline

[1] https://lagopus.github.io/

Energy monitoring	• Collection and storage of power consumption for physical resources (XPUs and XPFEs) via XCI NBI • Computation of power consumption for single instances of network paths and Network Services and for all the service instances associated to a given tenant (based on topology, flows, traffic and VM load information retrieved from the XCI).	Sept 2016
Energy-related policies	• Configuration, storage and management of energy-related policies for tenants.	Dec 2016
Automated configuration of devices power state	• Automated configuration of the power state of network nodes (Lagopus-based XPFEs) and servers based on their current usage and load (through XCI NBI).	Dec 2016
Energy-efficient network path provisioning	• On-demand provisioning of energy-efficient network paths, taking into account the requesting tenant and its associated policies. Requests for actual provisioning of network connections will be delivered to the XCI SDN controller, via its NBI.	March 2017
Energy-efficient Network Service provisioning	• On-demand provisioning of energy-efficient Network Service instances for vEPC, taking into account the requesting tenant and its associated policies. Requests for actual provisioning of Network Service instances will be delivered to the XCI SDN controller, via its NBI.	June 2017

mmWave mesh case

Table 66: EMMA mmWave mesh case roadmapTable 66 shows the roadmap for the EMMA implementation in the case of mmWave mesh following the functional validation objectives described above in 3.4.1.1:

Table 66: EMMA mmWave mesh case roadmap

Roadmap		
Features	Description	Timeline
Initial mmWave node activation	Activate mmWave nodes to minimize energy consumption while satisfying traffic demands of users	Q3/16
Energy-efficient mmWave node re-activation	Re-activate mmWave nodes until all traffic demands are satisfied	Q3/16
Path creation	Create the best path with the help of the RMA	Q3/16
Proof-of-concept	Implement the EMMA with XCI NBI to be worked on a mmWave mesh demo that will	Q4/17

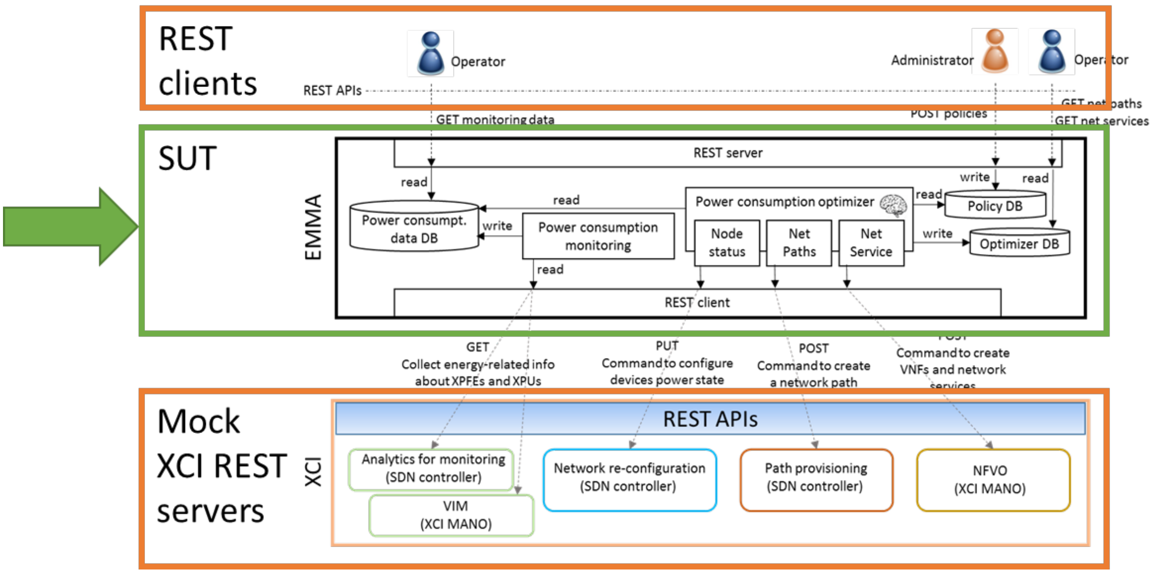
	be developed in WP5	

10.4.2.1 Validation environment

The validation environment is based on a VM where the EMMA application is deployed. EMMA constitutes the SUT; it is a modular Java application with components dedicated to specific features, like power consumption monitoring, policy management, power consumption optimization for network nodes, network paths and NFV Network Services, interaction with external components through REST APIs, Graphical User Interface (GUI). An SQL database is used to maintain and persist EMMA internal states and information (e.g. computed network paths or Network Services, policies), while a no-SQL database is used to store the time series associated to power consumption data. A detailed description of EMMA components is reported in Section 9.5; a picture summarizing the EMMA architecture and the software modules mentioned above is reported in Figure 52.

As shown in the picture, EMMA interacts with different components at the XCI, using the XCI's REST-based NBI. In particular EMMA acts as a REST client consuming the monitoring analytic, network re-configuration and network path provisioning services of the XCI SDN controller, the NS provisioning service of the XCI MANO NFVO and the XCI MANO VIM services for servers and VM monitoring and servers' status configuration (where supported). The EMMA-related XCI functionalities are developed in WP3, the plans for their evaluation are described in [2] and they are considered out of scope for WP4. For this reason, in this validation environment, all the XCI components will be emulated through some REST server mock components developed specifically for these tests and configured with a pre-defined set of data to feed the EMMA functionalities.

The interaction with client entities at the EMMA northbound interface is emulated through a REST client like curl2 or Postman3. REST clients and mock XCI REST servers will run in the same VM of the EMMA software. The whole environment is shown in the picture below.



[2] https://curl.haxx.se/
[3] https://www.getpostman.com/

Figure 76: EMMA validation environment – SUT and external mock components

mmWave mesh case

The validation environment for EMMA in the mmWave mesh case is drawn in Figure 77 where mmWave nodes are overlaid on a LTE macro cell. The mmWave node plays a role of both XFE (relay) and (mmWave) access with three or four sectors in both access and backhaul/fronthaul. The LTE macro BS plays a role of mmWave gateway as well in the cell to accommodate time-variant and spatially non-uniform traffic by forming a mmWave mesh network. A principle objective of the EMMA is to reduce energy consumption of mmWave mesh network by switching off mmWave nodes as much as possible in an area with small traffic demand. This validation environment corresponds to an experimental proof-of-concept in 5G-Berlin testbed where WiGig based mmWave nodes are used both for access and backhaul and software (virtual) switch is introduced between them to act as a XFE.

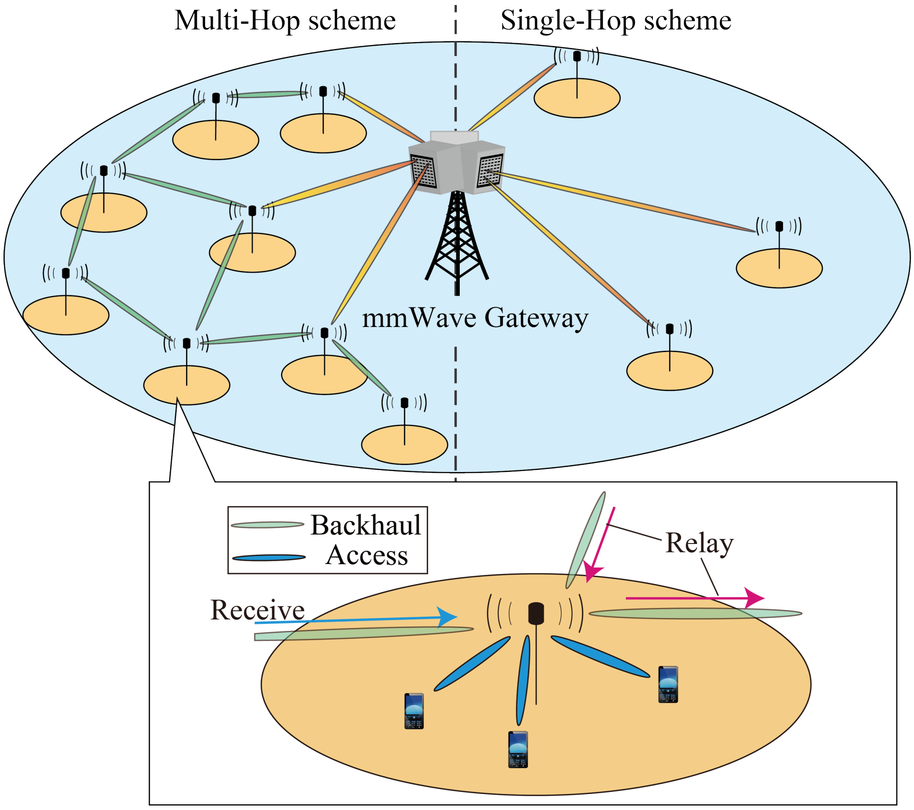


Figure 77: EMMA mmWave mesh case validation environment

10.4.3 Validation procedure

The EMMA functional validation will be performed through the execution of testing procedures in compliance with the common testing methodology adopted in the project. The main focus will be on the internal behavior of the EMMA software components, their interaction within the EMMA system and the interaction with the external components (REST clients at the EMMA northbound interface and mock REST servers at the interface with the XCI).

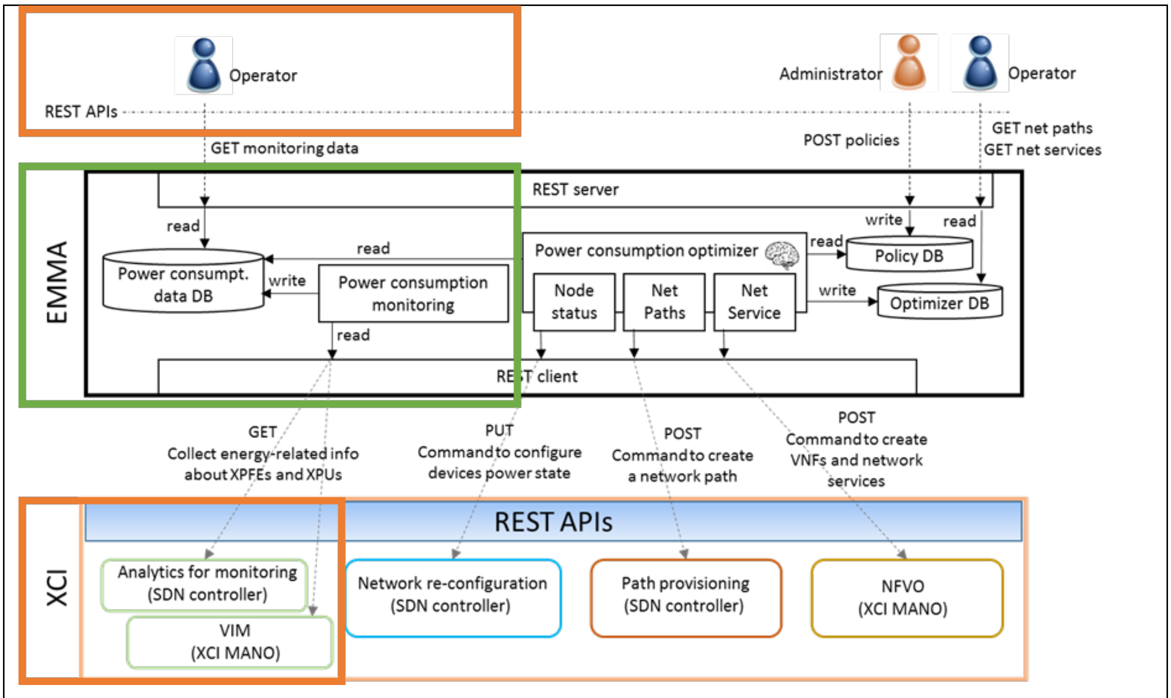
In terms of internal logic for computation of energy-efficient resource allocation, EMMA implements the algorithms described in section 4.1. Their evaluation is out of scope for the following test cards, since it is analyzed in detail in section 4.1.4 and 4.1.5, with reference to a simplified scenario where EMMA algorithms run directly on top of an ONOS SDN controller operating over an emulated, Mininet4-based network. This approach of splitting the evaluation of the EMMA algorithms on one hand and the internal software implementation of the application on the other hand allows to focus on the specific characteristics of each entity, adopting the most suitable methodology for the different objectives of the validation (i.e. quantitative KPIs evaluation for the former and functional validation of internal processes and compliance with external interfaces definition for the latter).

Table 67: Energy Monitoring test card 1

Test Card #	NXW_EMMA_APP_01	Execution Status	Planned
Test Name	Energy Monitoring		
Objectives	Verify the mechanisms to collect XPUs and XPFEs power consumption monitoring data at the XCI's NBI. Verify the mechanisms to aggregate power consumption data for: 1) Single instances of network connections 2) Single instances of Network Services 3) All the service instances associated to a given tenant Verify the mechanisms to store power consumption monitoring data in a consistent manner within the internal no-SQL database and expose them through the EMMA NBI.		
Related Use Cases	Dense urban information society		
Responsible	NXW		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Power consumption monitoring components of EMMA.
Test environment	

[4] http://mininet.org/



External components	- REST client to interact with the EMMA - Mock REST servers to emulate power consumption data provided by the XCI.

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Start the Mock XCI servers with a pre-configured set of power consumption data. Start the EMMA application. Using the Wireshark[5] tool verify the exchange of REST HTTP messages between EMMA and Mock XCI servers. **Expected Results:** Mock XCI servers and EMMA application up and running. EMMA periodically sends power consumption data requests to the XCI. Monitoring data are returned from the XCI as expected.	
2.	**Description:** Using the EMMA no-SQL DB CLI verify the consistency of the power consumption monitoring data stored in the internal DB.	

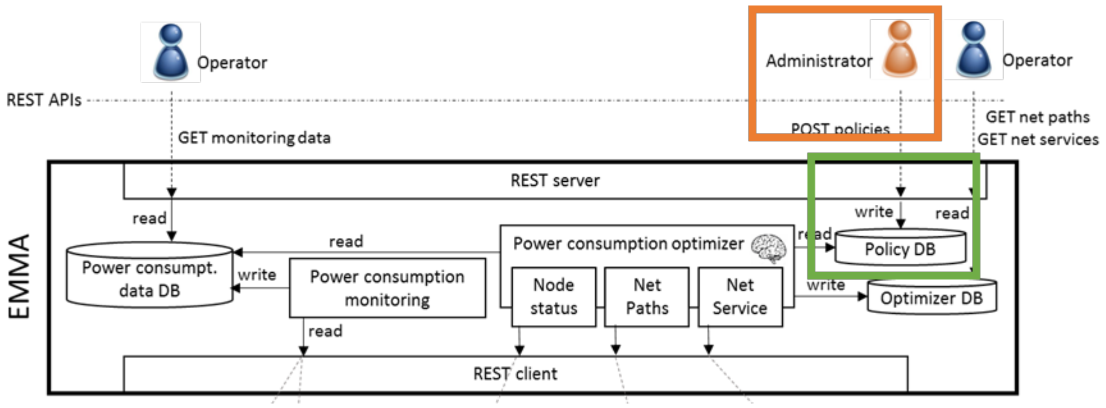
[5] https://www.wireshark.org/

	Expected Results: The internal DB includes power consumption data for physical infrastructure, network connections, Network Services and tenants organized in the expected time-series.	
3.	**Description:** Using the REST client retrieve power consumption information from the EMMA NBI for physical infrastructure, network connections, Network Services and tenants. **Expected Results:** Power consumption data are correctly retrieved.	

Table 68: Energy Monitoring test card 2

Test Card #	NXW_EMMA_APP_02	Execution Status	Planned
Test Name	Energy-related policies		
Objectives	Verify the mechanisms to configure, store and retrieve energy-related policies in EMMA.		
Related Use Cases	Dense urban information society		
Responsible	NXW		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Policy management components of EMMA.
Test environment	



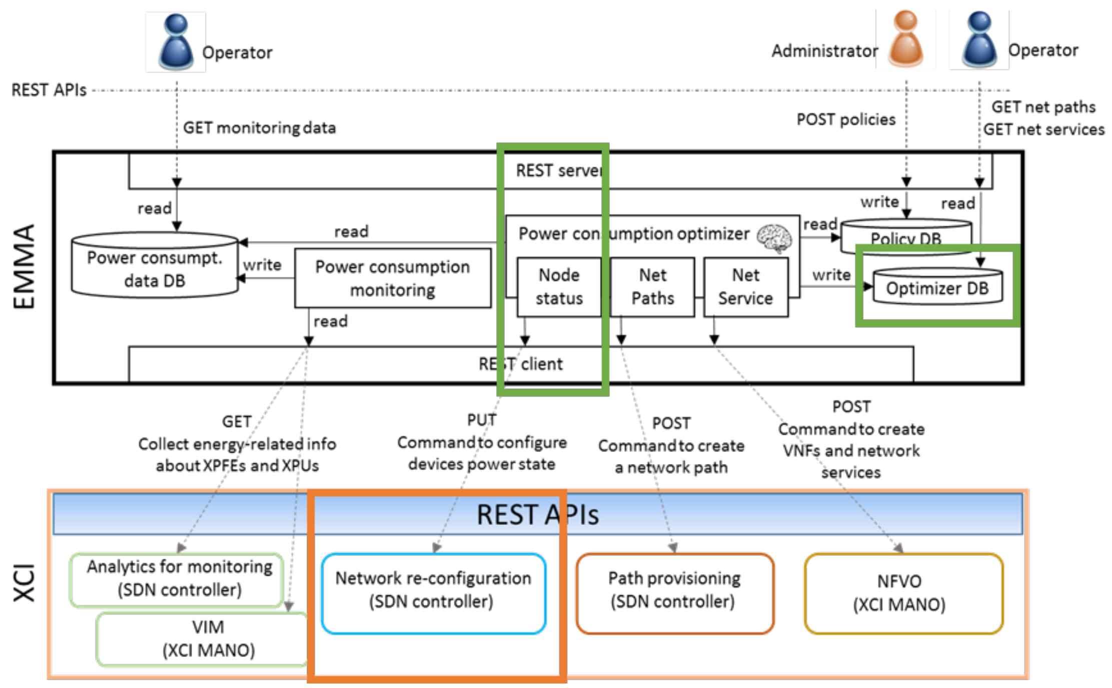
External components	REST client to interact with the EMMA

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Start the EMMA application. **Expected Results:** EMMA application up and running.	
2.	**Description:** Using the Wireshark tool, verify the exchange of REST HTTP messages between REST Client and EMMA. Using the REST client, add a set of energy-related policies with per-infrastructure provider scope and per-tenant scope. **Expected Results:** POST HTTP messages with policies in json format are correctly exchanged with between REST Client and EMMA. 201 CREATED HTTP messages are sent by EMMA as reply.	
3.	**Description:** Using the EMMA SQL DB CLI verify the presence of the configured policies in the internal DB. **Expected Results:** The internal DB includes the policies, properly organized per tenant or per infrastructure resource.	
4.	**Description:** Using the Wireshark tool verify the exchange of REST HTTP messages between REST Client and EMMA. Using the REST client retrieve the policies stored in the EMMA DB. **Expected Results:** All the policies are correctly retrieved following the defined classification.	

Table 69: Energy Monitoring test card 3

Test Card #	NXW_EMMA_APP_03	Execution Status	Planned

Test Name	Automated configuration of devices power states.
Objectives	Verify the EMMA mechanisms to detect inactive devices and put them in sleeping mode (through XCI NBI command). Verify the EMMA mechanisms to switch on devices (through XCI NBI command) when required by new services.
Related Use Cases	Dense urban information society
Responsible	NXW
Related Test Cards	N/A (not applicable)

SUT and topology	
SUT	EMMA power consumption optimizer for nodes power status
Test environment 	
External components	Mock REST servers to emulate the receiving and processing of power state reconfiguration commands at the XCI.

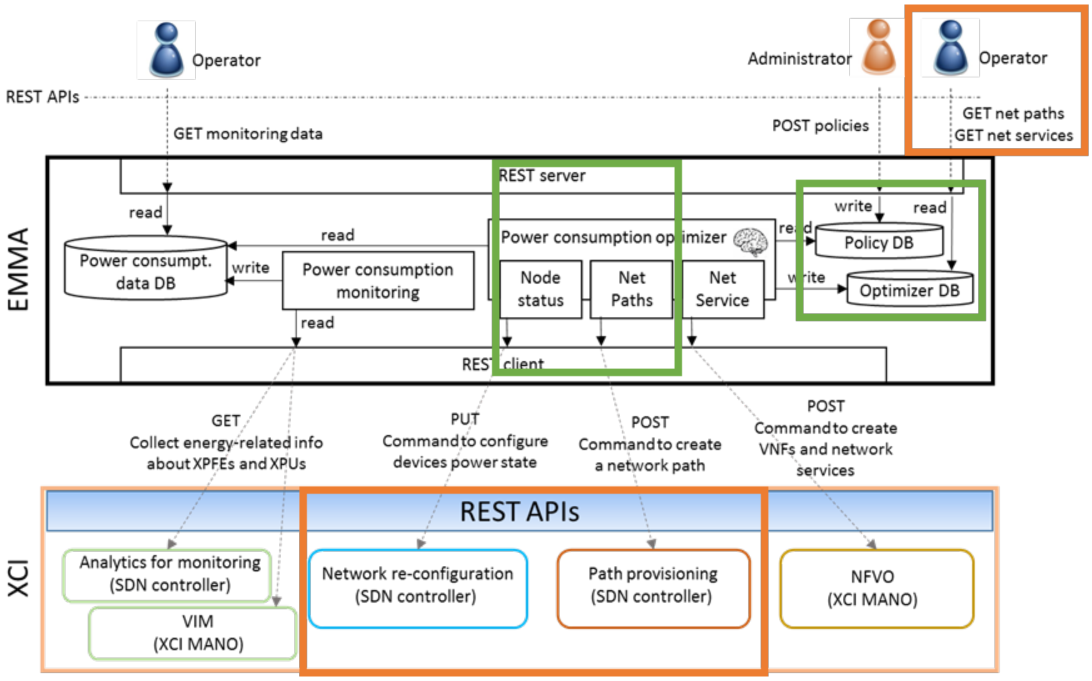
Test description		
Step #	Step description and expected results	Status
1.	**Description:** Start the Mock XCI servers with a pre-configured set of data for flows, network statistics and server load. These data must reflect a situation where XPFEs (XPUs) without any active flow (VM) are switched on and in active status. Start the EMMA application. Using the Wireshark tool, verify the exchange of REST HTTP messages between EMMA and Mock XCI servers. **Expected Results:** EMMA application up and running. EMMA periodically sends requests to the XCI to retrieve flows or VMs operating on network nodes or servers. These data are returned from the XCI as expected. EMMA sends a request to the XCI to change the power status (sleeping mode) of XPFEs (XPUs) without any active flow (VM). The REST message exchange between EMMA and XCI follows the expected format.	
2.	**Description:** Using the EMMA SQL DB CLI, verify the power status of the XPFEs and XPUs in the internal DB (active or sleeping mode). **Expected Results:** The power status is aligned with the configured one (sleeping mode for XPFEs without active traffic flows and active mode for XPFEs with active traffic flows).	
3.	**Description:** Using the EMMA internal CLI, notify the need to create a new network path which involves network nodes currently in sleeping mode. **Expected Results:** EMMA sends a requests to the XCI to change the power status (active mode) of the involved XPFEs.	
4.	**Description:** Using the EMMA SQL DB CLI, verify the power status of the XPFEs and XPUs in the internal DB. **Expected Results:** The power status is aligned with the configured one (sleeping mode for XPFEs without active traffic flows and active mode for XPFEs with active	

	traffic flows).	

Table 70: Energy Monitoring test card 4

Test Card #	NXW_EMMA_APP_04	Execution Status	Planned
Test Name	On-demand computation and provisioning of energy-efficient network paths.		
Objectives	Verify the EMMA mechanisms to instantiate on-demand an energy-efficient network path.		
Related Use Cases	Dense urban information society		
Responsible	NXW, POLITO		
Related Test Cards	NXW_EMMA_APP_03		
Additional Comments	The algorithms used for the computation of the network paths, as well as their evaluation, are described in section 4.1.		

SUT and topology	
SUT	EMMA power consumption optimizer for network paths
Test environment	



External components	• REST client to interact with EMMA and request a new network path. • Mock REST servers to emulate receiving and processing of power state configuration and network paths provisioning commands at the XCI.

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Start the Mock XCI servers. Start the EMMA application. The internal DB must be pre-configured with a given set of network paths and some network nodes in sleeping mode. Some of these inactive network nodes will be used in the additional network path to be created. **Expected Results:** Mock XCI servers and EMMA application up and running.	
2.	**Description:** Using the Wireshark tool, verify the exchange of REST HTTP messages between EMMA and Mock XCI servers and between REST client and EMMA. Using the REST client, send a request to create a new network connection. **Expected Results:** EMMA retrieves the energy-related policies associated to the requested network connection's tenant. EMMA retrieves the current network resource availability from the XCI SDN controller topology manager and the current power states of the XPFEs from the internal database. EMMA uses its internal algorithms to compute the energy-optimum path for the requested network connection, taking into account: 1) the connection parameters of the path, as retrieved from the policies (i.e. required bandwidth, protection mode, anti-affinity rules, etc.); 2) the current availability of the network resources and the current power status of the XPFEs. EMMA sends a requests to the XCI to change the power status (active mode) of the XPFEs traversed by the path, which are currently in sleep mode. When EMMA detects the XPFEs are back in active mode (information retrieved through GET requests to the XCI), it sends a request to the XCI SDN controller to create a new network connection, specifying the	

	expected network path. EMMA stores the information related to the new network connection and the new power status of XPFEs in its internal DB.	
3.	**Description:** Using the EMMA SQL DB CLI, verify the presence of the new network path and the power status of the traversed XPFEs in the internal DB. **Expected Results:** The power status is aligned with the configured one. The new network path is properly stored in EMMA internal DB.	
4.	**Description:** Using the REST client, send a request for to retrieve the network connections for the given tenant. **Expected Results:** The new network connection is properly included in the returned set.	
5.	**Description:** Using the REST client, send a request to delete the network connection previously created. **Expected Results:** EMMA sends a request to the XCI SDN controller to delete the previously created network connection and updates its internal DB. EMMA detects that some active XPFEs are no more in use (i.e. the ones activated specifically for the path previously created) and it sends a request to the XCI to change the power status (sleeping mode) of these XPFEs. EMMA updates the power status of these XPFEs in its internal DB.	
5.	**Description:** Using the EMMA SQL DB CLI, verify the status of the network path and the power status of the traversed XPFEs in the internal DB. **Expected Results:** The XPFEs power status is aligned with the configured one (sleeping mode for XPFEs without active traffic flows and active mode for XPFEs with active traffic flows). The network path is stored in EMMA internal DB with status CANCELLED.	

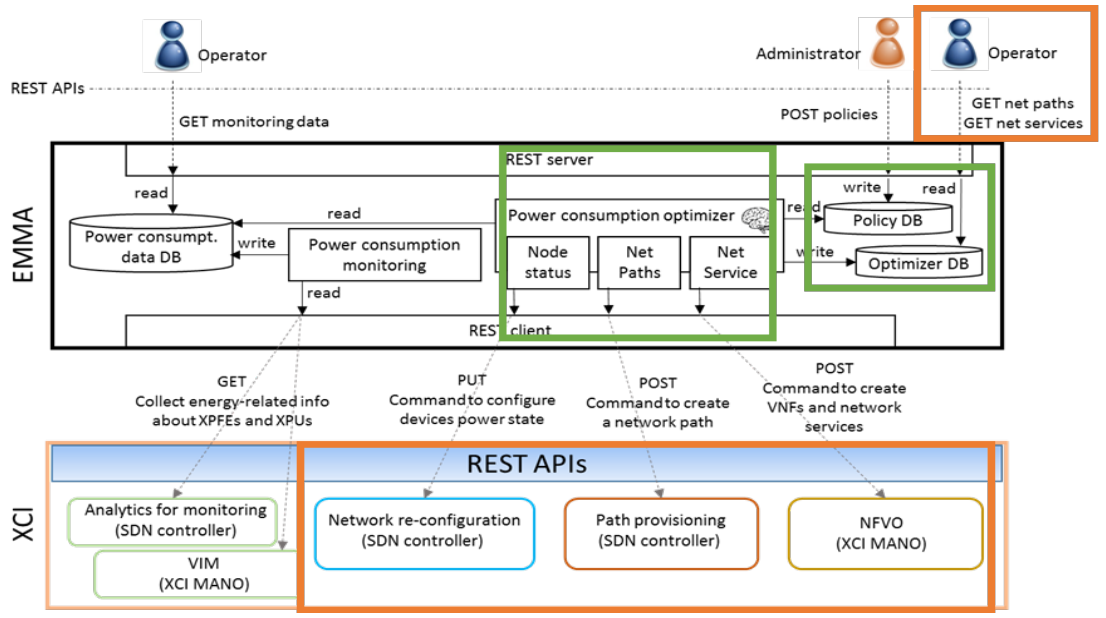
Table 71: Energy Monitoring test card 5

Test Card #	NXW_EMMA_APP_05	Execution Status	Planned

Test Name	On-demand computation and provisioning of energy-efficient Network Services.
Objectives	Verify the EMMA mechanisms to instantiate on-demand an energy-efficient Network Service.
Related Use Cases	Dense urban information society
Responsible	NXW, POLITO
Related Test Cards	NXW_EMMA_APP_03
Additional Comments	The algorithms used for the computation of the Network Services, as well as their evaluation, will be based on an evolution of the algorithms defined in section 4.1 and they are currently under specification.

SUT and topology	
SUT	EMMA power consumption optimizer for Network Services

Test environment



External components	• REST client to interact with EMMA and request a new Network Service. • Mock REST servers to emulate receiving and processing of power state configuration and net paths provisioning commands at the XCI.

<table>
<tr><th colspan="3">Test description</th></tr>
<tr><th>Step #</th><th>Step description and expected results</th><th>Status</th></tr>
<tr><td>1.</td><td>Description:
Start the Mock XCI servers.
Start the EMMA application. The internal DB must be pre-configured with a given set of Network Services and some network nodes and servers in sleeping mode. Some of these inactive resources will be used to allocate the virtual resources to provision the new Network Service.
Expected Results:
Mock XCI servers and EMMA application up and running.</td><td></td></tr>
<tr><td>2.</td><td>Description:
Using the Wireshark tool, verify the exchange of REST HTTP messages between EMMA and Mock XCI servers and between REST client and EMMA.
Using the REST client, send a request to create a new Network Service for a vEPC service instance.
Expected Results:
EMMA retrieves the energy-related policies associated to the requested Network Service’s tenant.
EMMA retrieves the current network and service resource availability from the XCI SDN controller topology manager and the XCI VIM, as well as the current power states of XPFEs and XPUs from the internal database.
EMMA uses its internal algorithms to compute the energy-optimum resource allocation for the vEPC instantiation, including VNFs placement and network paths for the requested Network Service, taking into account:
3) the service parameters of the requested NS, as retrieved from the NS description retrieved from the NFVO and from the configured policies (i.e. deployment flavour, required bandwidth, high-availability mode, anti-affinity rules, etc.);
4) the current availability of XPFEs and XPUs resources and their current power status.
EMMA sends a requests to the XCI to change the power status (active mode) of the XPFEs and XPUs, which will be used for the deployment of the NS and which are currently in sleep mode.
When EMMA detects the relevant XPFEs and XPUs are back in active mode (information retrieved through GET requests to the XCI), it sends a request to the XCI NFVO to create a new NS, specifying the expected VM location and network paths.
EMMA stores the information related to the new NS and the associated</td><td></td></tr>
</table>

	network connections, as well as the new power status of XPFEs and XPUs in its internal DB.	
3.	**Description:** Using the EMMA SQL DB CLI, verify the presence of the new Network Service and the new network path(s), as well as the power status of the traversed XPFEs in the internal DB. **Expected Results:** The power status is aligned with the configured one. The new Network Service and network path(s) are properly stored in EMMA internal DB.	
4.	**Description:** Using the REST client, send a request to retrieve the Network Services for the given tenant. **Expected Results:** The new Network Service is properly included in the returned set.	
5.	**Description:** Using the REST client, send a request to delete the Network Service previously created. **Expected Results:** EMMA sends a request to the XCI NFVO to delete the previously created Network Service and updates its internal DB. EMMA detects that some active XPFEs and XPUs are no more in use (i.e. the ones activated specifically for the NS previously created) and it sends a request to the XCI to change the power status (sleeping mode) of these XPFEs and XPUs. EMMA updates the power status of these XPFEs and XPUs in its internal DB.	
5.	**Description:** Using the EMMA SQL DB CLI, verify the status of the Network Service and the power status of the relevant XPFEs and XPUs in the internal DB. **Expected Results:** The XPFEs and XPUs power status is aligned with the configured one. The Network Service is stored in EMMA internal DB with status CANCELLED.	

mmWave mesh case

The following tables describe validation procedures for EMMA in the mmWave mesh case.

Table 72: mmWave mesh test card 1

Test Card #	HHI_EMMA_APP_01	Execution Status	Running
Test Name	Initial mmWave node activation & path creation		
Objectives	Verify the algorithm for initial mmWave node activation to minimize energy consumption of (mmWave) access. Verify the algorithm for initial path creation to minimize energy consumption of backhaul/fronthaul network by assuming ideal backhaul link scenario (backhaul/fronthaul capacity is much larger than the capacity of access).		
Related Use Cases	Dense urban information society		
Responsible	HHI		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Initial mmWave node activation & path creation
Validation environment explained in 3.4.3.1 with ideal backhaul link (backhaul/fronthaul capacity is much larger than the capacity of access): • Variable traffic distribution • Variable size of macro cell	
Additional Algorithms Information	• System satisfaction rate • Energy consumption

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Test of initial mmWave node activation by changing traffic distribution. **Expected Results:** The developed algorithm should activate initial mmWave nodes in accordance with the traffic distribution. The developed algorithm should minimize energy consumption while satisfying traffic demands.	

2.	**Description:** Test of initial path creation by changing traffic distribution and size of macro cell. Compare the results with that of single-hop topology. **Expected Results:** The developed algorithm should form mesh topology in accordance with the traffic distribution. The developed algorithm should minimize energy consumption while satisfying traffic demands. The developed algorithm should be scalable in terms of the size of macro cell.	

Table 73: mmWave mesh test card 2

Test Card #	HHI_EMMA_APP_02	Execution Status	Running
Test Name	Energy-efficient mmWave node re-activation & path creation		
Objectives	Verify the algorithm for energy-efficient mmWave re-activation & path creation to improve satisfaction of traffic demands.		
Related Use Cases	Dense urban information society		
Responsible	HHI		
Related Test Cards	mmWave mesh test case 1		

SUT and topology	
SUT	Energy-efficient mmWave node re-activation & path creation
Validation environment explained in 3.4.3.1: • Variable traffic load & distribution • Variable size of macro cell	
Additional Algorithms Information	• System satisfaction rate • Energy consumption

Test description		
Step #	Step description and expected results	Status
1.	**Description:**	

	Test of energy-efficient mmWave node re-activation & path creation by changing traffic load & distribution. Compare the results with other energy management algorithms. **Expected Results:** The developed algorithm should form mesh topology with multi-route multi-hop by following traffic load & distribution. The developed algorithm should minimize energy consumption while satisfying traffic demands. The improvement of energy consumption should be much larger than the other energy management algorithms.	

10.4.4 Initial Results of EMMA Power Consumption Optimizer

We evaluated the performance of the Power Consumption Optimizer (PCO) in EMMA against the optimal solution, as well as versus the simple case where no power saving strategy is adopted and the whole network is always active (hereinafter referred to as No Power Saving). The performance of the EMMA PCO and of the "No Power Saving" scheme is obtained by emulation in the system we implemented and that is described above. The solution of the optimization problem in Eq. (3) is instead obtained using the Gurobi solver, considering the same network as that emulated in our experiments with Mininet.

We stress that the algorithms validation scenarios are quite different from the implementation scenario, not being constrained by the availability of physical devices.

We derived the results assuming a default number of core and edge switches equal to 12 and 6, respectively; 10 hosts are connected to each edge switch. Links between any two core switches are set with probability 0.5 and the link capacity is set to 10 Mbytes/s. TCP traffic flows are generated using the Iperf tool, using 1500-byte packets. For each traffic flow, source and destination are selected at random among all possible hosts. Note that this is a worst-case assumption for the PCO, while it favours the No Power Saving strategy. The flow arrival rate is equal to 0.1 flows/s. The traffic flow duration is exponentially distributed with mean equal to 20 s. The complete list of default values that we adopted for the system parameters is reported in Table 74.

Table 74: Default settings for performance evaluation

Parameter	Value
Flow arrival rate	0.1 flows/s
Average flow duration	20 s
Number of core switches	12
Number of edge switches	half the no. of core switches
Link Capacity	10 MB/s
Hysteresis	10 s
P_{idle}	90 W [6]
$P(i, j, t)$	$0.644 \cdot \tau_{i,j}(t)$ nW [6]
Number of hosts per edge switch	10
Link prob. b/w switches	0.5
Packet size	1500 bytes
Experiment duration	500 s

In the following figures, we show the average power consumption per flow, as the flow generation rate and the number of network switches vary. The results have been obtained by averaging over 20 experiments. Note also that power consumption is computed based on traffic statistics and nodes operational states, consistently in all cases.

Figure 78 compares the performance of EMMA to the optimum as well as to that of the No Power Saving scheme, as the flow arrival rate varies and for the default number of core switches (namely, 12). Observe that EMMA matches the optimum very closely, for any value of traffic load (indicated by the flow arrival rate). The power saving it provides with respect to the case where all network switches are always on is very noticeable. Clearly, the power gain tends to shrink when many flows have to be allocated (high flow arrival rate), i.e., as an increasing number of switches and links have to be used.

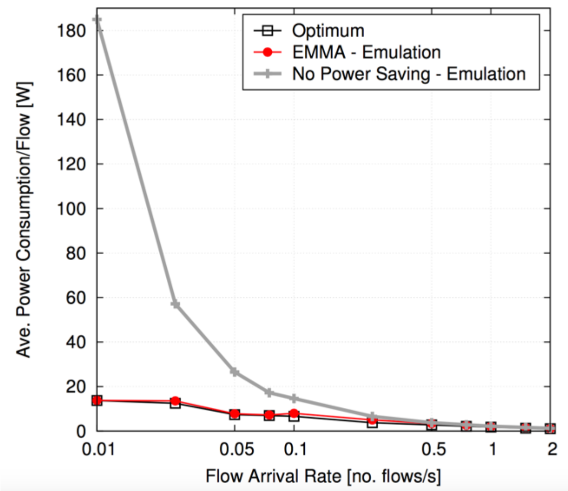


Figure 78: Comparing EMMA against the optimum and the No Power Saving scheme: Average power consumption per flow as a function of the flow arrival rate (no. core switches = 12).

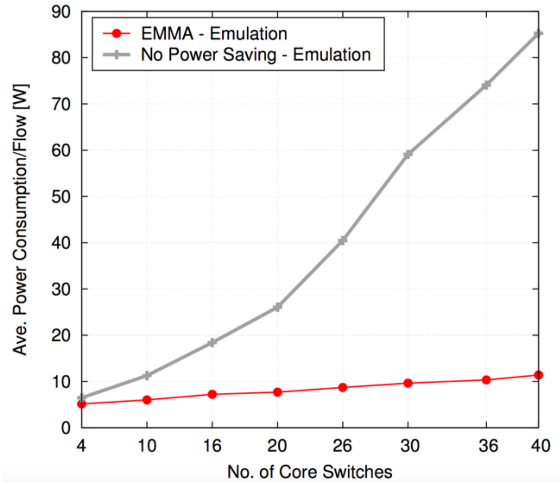


Figure 79: Average power consumption per flow vs. no. of core switches comparison between EMMA and No Power Saving (flow arrival rate = 0.1)

The behavior of EMMA compared to the No Power Saving scheme, as the network size varies, is presented in Figure 79. Here we do not show the optimum performance, as we could not solve the optimization problem for a number of core switches significantly larger than the default value. All results are therefore derived by emulation, under a flow arrival rate equal to 0.1. The plot confirms the excellent performance of EMMA: it reduces the power consumption per flow by a factor ranging from 2 (for 10 core switches) to 8 (for 40 core switches). As noted before, a smaller improvement is obtained only when the network size is small compared to the flow arrival rate (e.g., for 4-5 core switches).

Finally, Figure 80 depicts the gain that we can achieve with EMMA with respect to the No Power Saving strategy, as a function of the number of core switches and for a flow arrival rate equal to 0.05, 0.1, 0.5, 1. The gain is computed as the difference in power consumption between No Power Saving and EMMA, normalized to the power consumption of the former scheme. As expected, the gain that EMMA provides is higher for a lower value of flow arrival rate and a larger network size, since it is possible to aggregate more flows on the same links and there are idle switches that

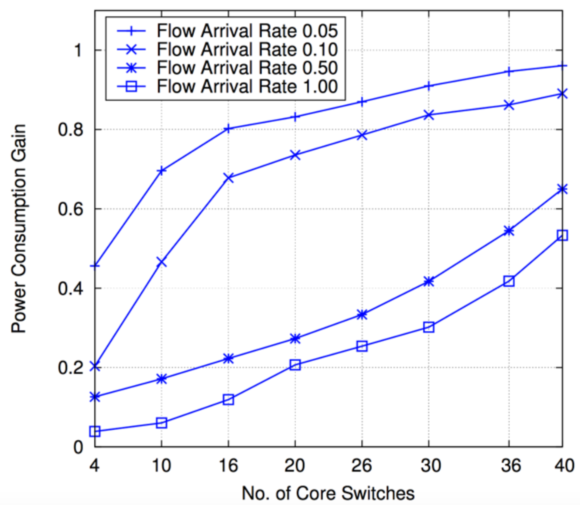


Figure 80: Gain in average power consumption per flow (derived by emulation) provided by EMMA with respect to No Power Saving, as the number of core switches and the flow arrival rate vary.

can be turned off. Interestingly, the gain we obtain is always quite high, with peak values that approximate 1.

10.5 Content Delivery Network Management Application

10.5.1 Functional validation objectives

The main objective of this testing and functional validation is to prove that the CDNMA application is able to instantiate upon client (CDN operator) request, a vCDN infrastructure based on the CDN operator criteria and the network infrastructure available. It will also be tested that it is able to control and manage the CDN service creating and implementing the content delivery rules. To that end, the main aspects that the tests aim are the following:

1) Processing of client (CDN operator) request. The application will show towards the client a graphic interface running on a REST API for management and operation actions.
2) Management of the network topology information obtained from the XCI NBI through a REST API.
3) Request the vCDN infrastructure instantiation to XCI NBI. Providing a complete descriptor file with detailed information about the CDN elements and how they have to be configured.
4) Management and storage of specific monitoring information obtained from the CDN nodes and the network users.
5) Management and update of the content delivery rules over the CDN nodes placed in the network, based on the monitoring information received and the logic defined by the CDN operator.
6) Performance of the web player (starting point). Video player that shows the video from the selected server.

10.5.2 Roadmap

Table 75: CDNMA roadmap

Roadmap		
Features	Description	Timeline
CDN instantiation	Provide a complete descriptor file for the vCDN infrastructure instantiation based on the CDN operator criteria and the network infrastructure available.	Q4/2016
Management of monitoring information	Get the specific monitoring information and store the information in the database.	Q4/2016 Q1/2017
CDN management	Create and enforce the content delivery rules.	Q1/2017
Video visualization	Web player that shows the video from the selected server based on the content delivery rules.	Q1/2017

10.5.3 Validation environment

The validation environment where the CDNMA application will be tested is located in a cloud environment and it is based on a Server HP DL380eG8 (8xLFF) composed of 2x Intel Hexa-Core Xeon E5-2420, 96 GB of RAM and 4 TB of hard disk. XenServer is used as open source virtualization platform. The application will run on a VM deployed for that purpose.

The environment will be composed of all necessary elements to check the main functionalities of the application. Given the fact that the application will interact with the XCI, and with some applications (i.e. MMA application), there will be a specific XCI MANO for the CDN implementation developed by ATOS and tested in the same environment that will prove the interaction between the XCI and the CDNMA application through the REST APIs created for this purpose. The application will also show towards the client a graphic interface running on a REST API for management and operation actions. Regarding the MMA application a mock-up will be developed to test the proper function between applications. It will also be considered an internal no-SQL DB (MongoDB) to store the monitoring information and the content delivery rules, and a web player to extract the content delivery rules and balance the load over the replica servers.

A detailed description of CDNMA components is reported Section 9.6; a picture summarizing the CDNMA architecture is reported in Figure 55.

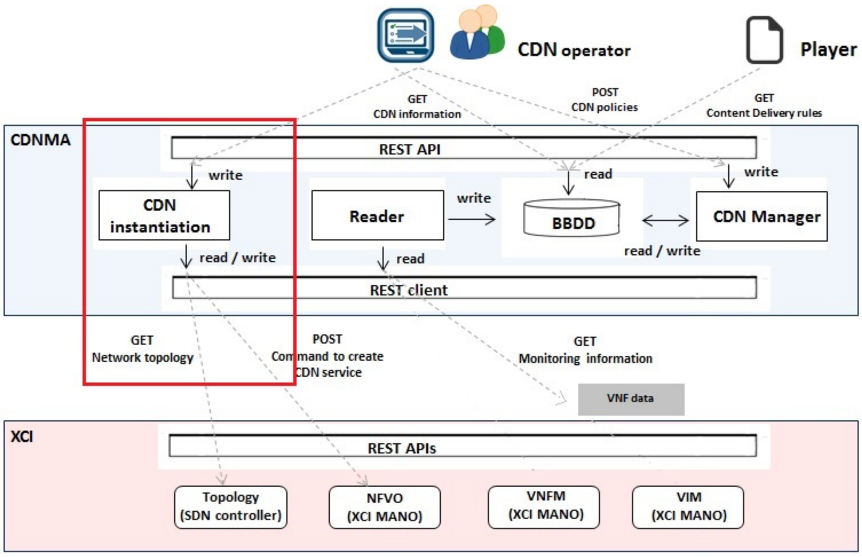
10.5.4 Validation procedure

The following test cards define the validation procedures that will be carried out for testing the main components involved in the CDNMA application.

Table 76: CDNMA test card 1

Test Card #	ATOS_CDNMA_APP_01	Execution Status	Planned
Test Name	CDN instantiation		
Objectives	Get the network infrastructure available Receive the CDN policies from the CDN operator regarding where to deploy the CDN nodes. Provide a complete descriptor file for the vCDN infrastructure instantiation		
Related Use Cases	UC2 – Media Distribution: vCDN		
Responsible	ATOS		
Related Test Cards	N/A (not applicable)		

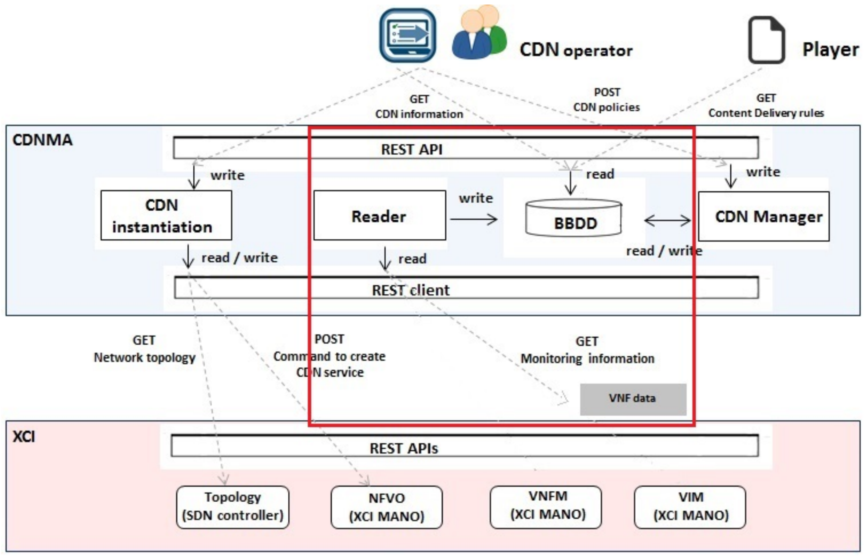
SUT and topology

SUT	CDN instantiation component for the vCDN deployment
Test environment topology	
Additional components/ algorithms	• REST client that provide the CDN policies • Mock-up of the XCI that emulates the network infrastructure available provided by the XCI

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Start the mock-up of the XCI with a predefined network infrastructure. Start the CDN instantiation component of the CDNMA application. Start the REST client with predefined CDN policies. **Expected Results:** Mock-up of the XCI and REST client are correctly configured and provide the data required. CDN instantiation component receive and get the necessary data.	
2.	**Description:** The CDN instantiation component processes the CDN policies and the network infrastructure information. **Expected Results:** The CDN instantiation component provides a complete descriptor file for the vCDN infrastructure instantiation.	

Table 77: CDNMA test card 2

Test Card #	ATOS_CDNMA_APP_02	Execution Status	Planned
Test Name	Management of monitoring information		
Objectives	Get the specific monitoring information from the VNFs Get specific information (location) about the users Store the information in the database		
Related Use Cases	UC2 – Media Distribution: vCDN		
Responsible	ATOS		
Related Test Cards	N/A (not applicable)		

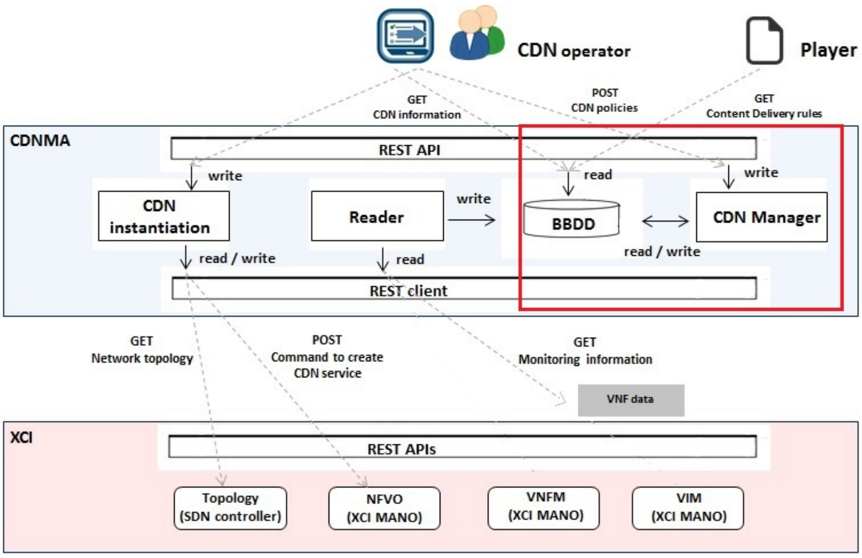
<table>
<tr><th colspan="2">SUT and topology</th></tr>
<tr><td>SUT</td><td>Reader component for the monitoring information</td></tr>
<tr><td>Test environment topology</td><td>
</td></tr>
<tr><td>Additional components/ algorithms</td><td>• Mock-up of the VNF data module that emulates the monitoring data provided by the VNFs
• Mock-up of the MMA application that emulates the user information provided by the MMA application</td></tr>
</table>

<table>
<tr><th colspan="3">Test description</th></tr>
<tr><td>Step #</td><td>Step description and expected results</td><td>Status</td></tr>
<tr><td>1.</td><td>Description:
Start the mock-up of the VNF data module with predefined data about the VNFs
Start the reader component of the CDNMA application.
Start the mock-up of the MMA application with predefined data about different users.
Expected Results:
Mock-up of the VNF data module and the mock-up of the MMA application provide the data required.
Reader component receive and get the proper data.</td><td></td></tr>
<tr><td>2.</td><td>Description:
Start the database.</td><td></td></tr>
</table>

	The reader component stores the monitoring information in the database. **Expected Results:** The monitoring information is properly stored.	

Table 78: CDNMA test card 3

Test Card #	ATOS_CDNMA_APP_03	Execution Status	Planned
Test Name	CDN Management		
Objectives	Receive the CDN distribution rules from the CDN operator. Get the monitoring information stored in the data base. Create the new content delivery rules and store it in the data base.		
Related Use Cases	UC2 – Media Distribution: vCDN		
Responsible	ATOS		
Related Test Cards	N/A (not applicable)		

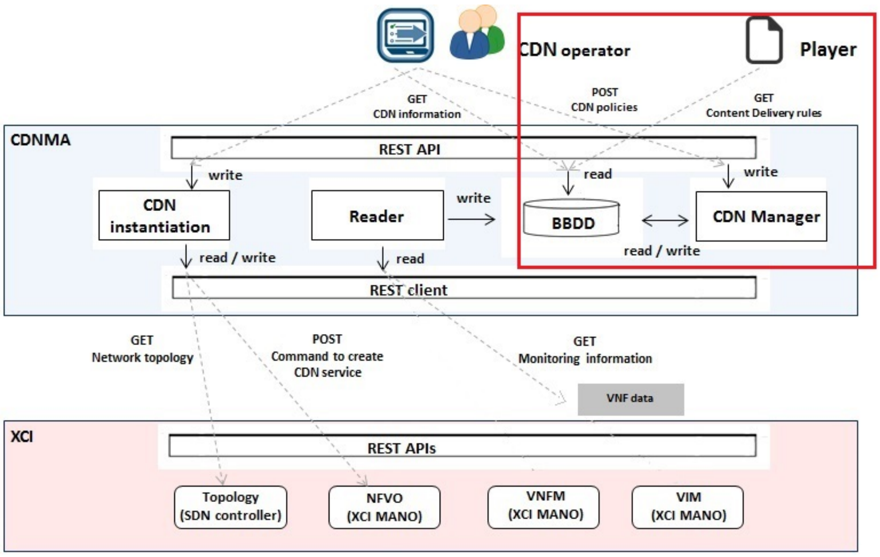
<table>
<tr><th colspan="2">SUT and topology</th></tr>
<tr><td>SUT</td><td>CDN manager component for the content delivery rules</td></tr>
<tr><td>Test environment topology</td><td>
</td></tr>
<tr><td>Additional components/ algorithms</td><td>• REST client that provide the CDN distribution rules</td></tr>
</table>

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Start the REST client with predefined CDN distribution rules. Start the CDN manager component of the CDNMA application. Start the database. **Expected Results:** REST client works properly and send the information required. The CDN manager component receives and gets the proper data.	
2.	**Description:** The CDN manager component processes the information and creates the new content delivery rules. The CDN manager stores the new content delivery rules in the data base.	

	Expected Results: The new content delivery rules are created and stored in the database and are available.	

Table 79: CDNMA test card 4

Test Card #	ATOS_CDNMA_APP_04	Execution Status	Planned
Test Name	Video visualization		
Objectives	Get the content delivery rules from the database Enforce the content delivery rules		
Related Use Cases	UC2 – Media Distribution: vCDN		
Responsible	ATOS		
Related Test Cards	N/A (not applicable)		

SUT and topology	
SUT	Web player that shows the video from the selected server based on the content delivery rules
Test environment topology	
Additional components/ algorithms	

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Start the video player. Start the database **Expected Results:** The video player works and extracts the content delivery rules from the data base.	
2.	**Description:** The video player redirect the user requests to the replica servers based on the content delivery rules. **Expected Results:** The user requests are served from the suitable replica servers.	

10.6 TV Broadcasting Application

10.6.1 Functional validation objectives

The main objective of this testing and functional validation is to prove that the TV

Broadcasting Application (TVBA) is able to instantiate a video service (up to High Definition resolution) and inject it to the Crosshaul network infrastructure available. It will also be tested that it is able to control and manage the service provisioning by monitoring the network status by means of creating and implementing the service delivery rules.

To that end, the main aspects that the planned set of tests aim are the following:

1) Video service configuration. Stream and file play-out. User control interface for management and operation.
2) Video service reconfiguration.
3) Deployment and functional testing of different software pieces of the server, virtual machines instantiation and network connectivity.
4) End-to-end service monitoring. Quality assessment/QoS estimation.

10.6.2 Roadmap

Table 80: TVBA roadmap

Roadmap		
Features	Description	Timeline
Video service configuration and injection	Configurable TV head-end to manage and operate video service.	Q3/2016
Management of network topology and monitoring information	• Management of the network topology information. • Management and monitoring of specific information from nodes and service deployment	Q4/2016 Q1/2017
Routing management	Create and enforce the content delivery rules for the broadcast tree deployment.	Q1/2017
QoS analysis	Video and quality metrics visualization for the service. KPIs monitoring.	Q2/2017

10.6.3 Validation Environment

The validation environment where the TV Broadcasting first test suite will be tested is located in a cloud environment and it is based on:

ID: svstation (desktop computer) *64 bits smbios-2.8 dmi-2.7 vsyscall32*

CPU Intel® Core™ i7-5930K CPU @3.50GHz

Xeon E7 v3/Xeon E5 v3/Core i7 DMI2

SCSI Disk 24TB

Ubuntu is the OS selected.

KVM is used as open source virtualization platform, the switching stack is provided by OpenvSwitch. The application will run on a VM deployed for that purpose.

The environment will be composed of all necessary elements to check the main functionalities of the application.

10.6.4 Validation procedure

The following test cards define the validation procedures that will be carried out for testing the main components involved in the TVB Application.

Setup

Before testing, we have to setup the environment. First we launch the physical servers with Ubuntu running on them, and then we install and run the virtual machines using the open source Kernel-based Virtual Machine KVM.

OpenDaylight will be used as SDN platform to manage the network.

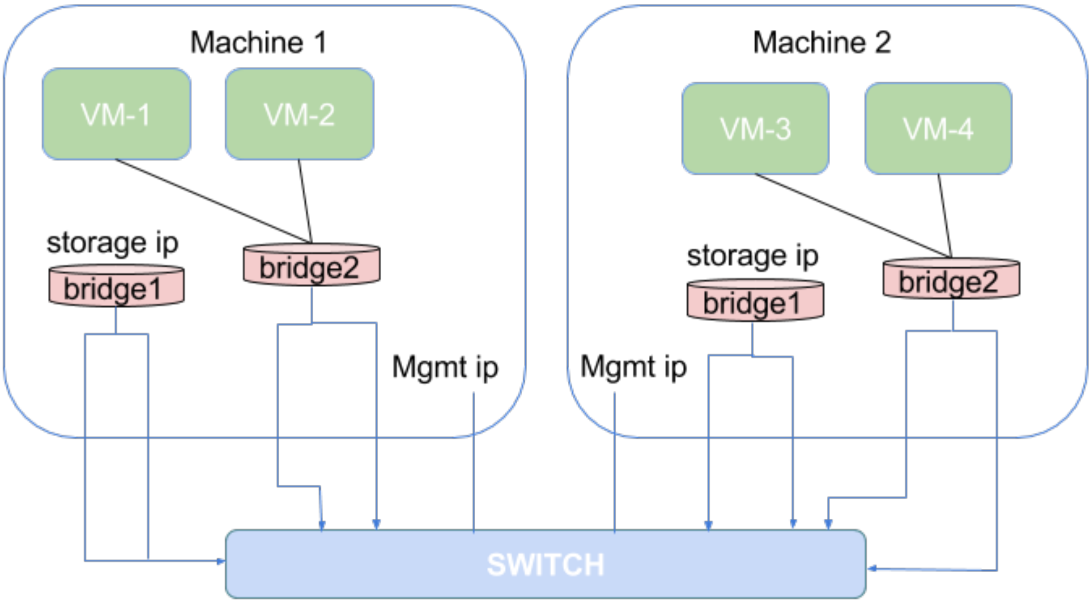


Figure 81: TVBA test suite

Table 81: TVBA test card 1

Test Card #	VIS_TVBA_APP_01	Execution Status	Executed
Test Name	Functional test of TV head-end		
Objectives	To test the functionality the TV head-end should provide as video source for the media distribution		
Related Use Cases	UC2 – Media Distribution [3]		
Responsible	Visiona		
Related Test Cards	VIS_TVBA_APP_02		

SUT and topology	
SUT	TV Head-end
Test environment topology	N/A
Additional components/ algorithms	This test will validate the funtionality of the TV head end independently (standalone test).

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Launching head-end application as a stand-alone application in the system. **Expected Results:** GUI providing the managing and monitoring environment for the service configuration	Passed
2.	**Description:** Video service configuration (file o live source, bit rate, and metadata) **Expected Results:** Perfectly conformed video service that can be monitored in the GUI.	Passed
3.	**Description:** Video stream generation and play out. **Expected Results:** Video stream physically deployed through the	Passed

	interface and conformed under the standard.	

Table 82: TVBA test card 2

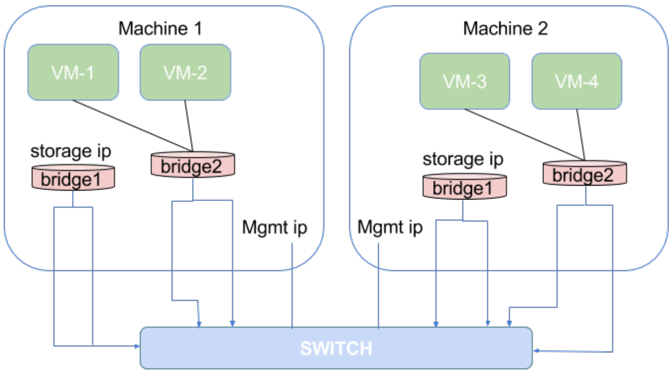
Test Card #	VIS_TVBA_APP_02	Execution Status	Executed
Test Name	Video service reconfiguration		
Objectives	To establish new parameters for the video service deployment (as future responsiveness to network status)		
Related Use Cases	UC2 – Media Distribution		
Responsible	Visiona		
Related Test Cards	VIS_TVBA_APP_01		

SUT and topology	
SUT	TV Head-end
Test environment topology	N/A
Additional components/ algorithms	This test will validate the funtionality of the TV head end independently (standalone test).

Test description		
Step #	Step description and expected results	Status
1.	**Description:** Selection and introduction of new video source and service reconfiguration **Expected Results:** Seamless service providing	Passed

Table 83: TVBA test card 3

Test Card #	VIS_TVBA_APP_03	Execution Status	Executed
Test Name	End machine connectivity		
Objectives	To assert that the communication between the machines is working		
Related Use Cases	UC2 – Media Distribution		
Responsible	Visiona		
Related Test Cards	VIS_TVBA_APP_04 & VIS_TVBA_APP_05		

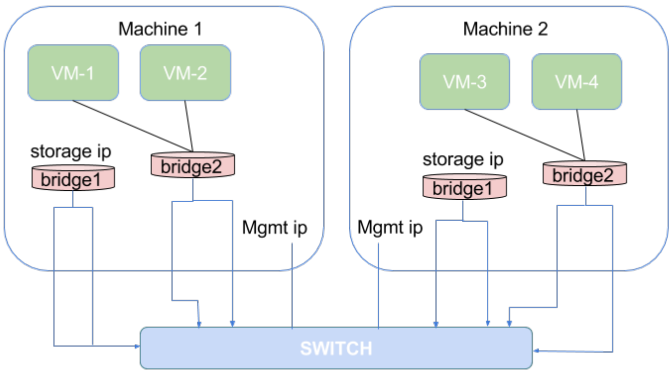
SUT and topology	
SUT	Virtual Machines
Test environment topology	
Additional components/ algorithms	• Kernel-based Virtual Machine • VRDP

Test description		
Step #	Step description and expected results	Status
1.	**Description:** With this test we want to check that the communication channel is open and working properly. We will use the ICMP (Internet Control Message Protocol) to send messages and get the response back from the servers in order assert that is able to receive and send data. 1) On the headless server, create a new virtual machine: 2) Make sure the settings for this VM are appropriate for the guest operating system that is installed. 3) Create a virtual hard disk for the VM 4) Add an IDE Controller to the new VM. 5) Set the VDI file created above as the first virtual hard disk of the new VM. 6) Attach the ISO file that contains the operating system installation that you want to install later to the virtual machine, so the machine can boot from it. 7) Enable VirtualBox remote desktop extension (the VRDP server): 8) Start the virtual machine are returned to the command line, then something went wrong. 9) On the client machine, fire up the RDP viewer and try to connect to the server. **Expected Results:** We will send and receive all the packages without any	Passed

	loss.	

Table 84: TVBA test card 4

Test Card #	VIS_TVBA_APP_04	Execution Status	Planned
Test Name	Topology		
Objectives	To assert the topology is well setup		
Related Use Cases	UC2 – Media Distribution		
Responsible	Visiona		
Related Test Cards	VIS_TVBA_APP_03 & VIS_TVBA_APP_05		

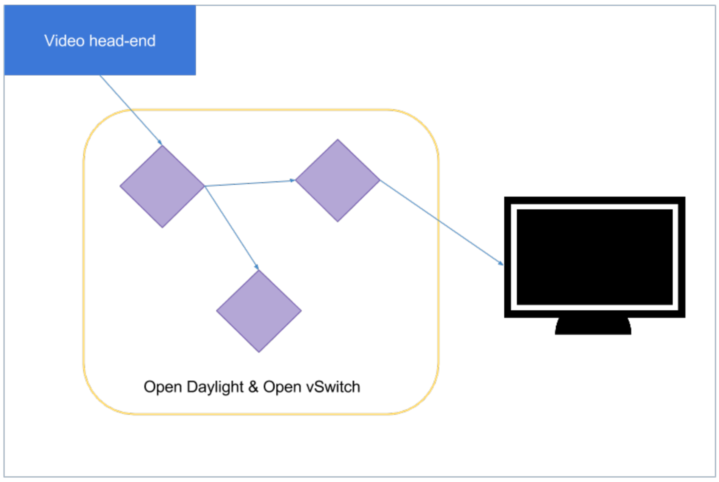
SUT and topology	
SUT	Virtual network
Test environment topology	
Additional components/ algorithms	N/A

Test description		
Step #	Step description and expected results	Status
1.	**Description:** With this test, we want to check that the network topology is established and the testbed suite is working properly. We will assert that the 3 networks nodes (as in 5Tonic), one working as an input and two	Planned

	working as potential outputs, are properly connected. The final deployment will take advantage of the use of the open source software ODL (OpenDaylight). **Expected Results:** To verify that the network is the one we designed with the same structure as in the test case.	

Table 85: TVBA test card 5

Test Card #	VIS_TVBA_APP_05	Execution Status	Planned
Test Name	End to end service visualization		
Objectives	To assert the video transmission and reception		
Related Use Cases	UC2 – Media Distribution		
Responsible	Visiona		
Related Test Cards	VIS_TVBA_APP_03 & VIS_TVBA_APP_04		

SUT and topology	
SUT	TV Service
Test environment topology	
Additional components/ algorithms	• Head-end • HD resolution TV

Test description		
Step #	Step description and expected results	Status
1.	**Description:** With this test, we want to check the end-to-end video transmission and reception from the head-end to the TV. One machine will send data through the network to the destination machine. This will take this video data and reproduce it properly. The video service will be analyzed by a small group that will provide a MOS result after a subjective test following BT.500 [2] and ITU-T P.910 [14]. **Expected Results:** Video service visualization and QoS analysis. The subjective test will confirm that the quality of the video service provided is good enough for the final user, and the perceived quality of the video service provided upon Crosshaul is, at least, as good as a DVB-based one.	Planned

11 Appendix IV: EMMA algorithm for mmWave meshed network

The environment for EMMA in the mmWave mesh case is drawn in Figure 82 where mmWave nodes are overlaid on a LTE macro cell [18]. The mmWave node plays a role of both XFE (relay) and (mmWave) access with three or four sectors (links) in both access and backhaul/fronthaul. The LTE macro BS plays a role of mmWave gateway as well in the cell to accommodate time-variant and spatially non-uniform traffic by forming a mmWave mesh network. A principle objective of the EMMA is to reduce energy consumption of mmWave mesh network by switching OFF mmWave nodes as much as possible in an area with small traffic demand. As it's hard to optimize ON/OFF status of mmWave nodes and backhaul/fronthaul paths all at once, the EMMA algorithm for mmWave mesh case is divided into three steps as summarized in Figure 83. In the first step (i), initial ON/OFF status of mmWave nodes are determined based on the traffic demands per mmWave node. In the next step (ii), initial paths of backhaul/fronthaul network are created to minimize power consumption. If isolated mmWave nodes exist even after step (ii), the final step (iii) re-activate remaining mmWave nodes with energy efficient manner to transfer the traffic for the isolated mmWave nodes. The details of algorithm description are given below where the mmWave gateway and mmWave nodes are called simply GW and AP henceforth.

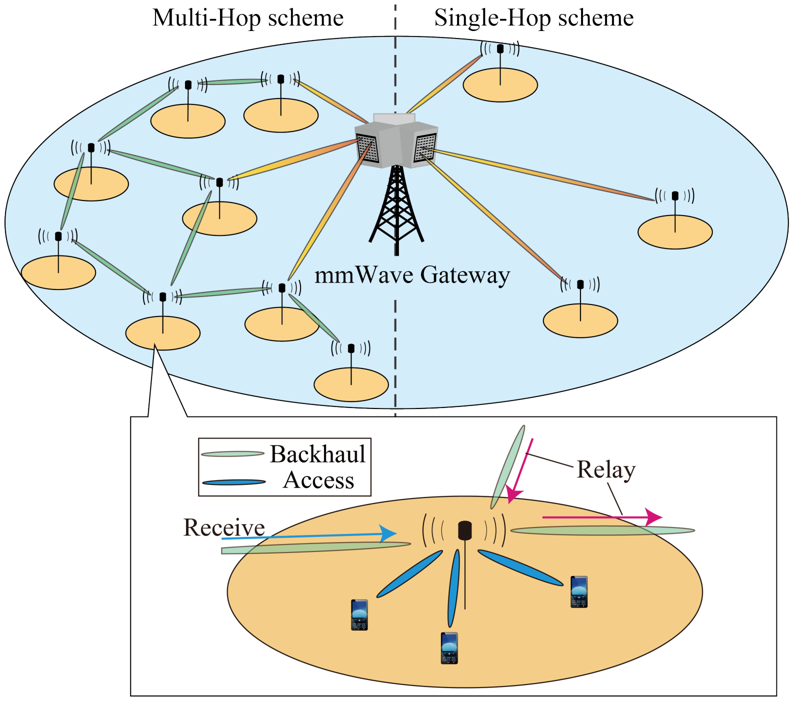


Figure 82: mmWave meshed network overlaid on a LTE macro cell.

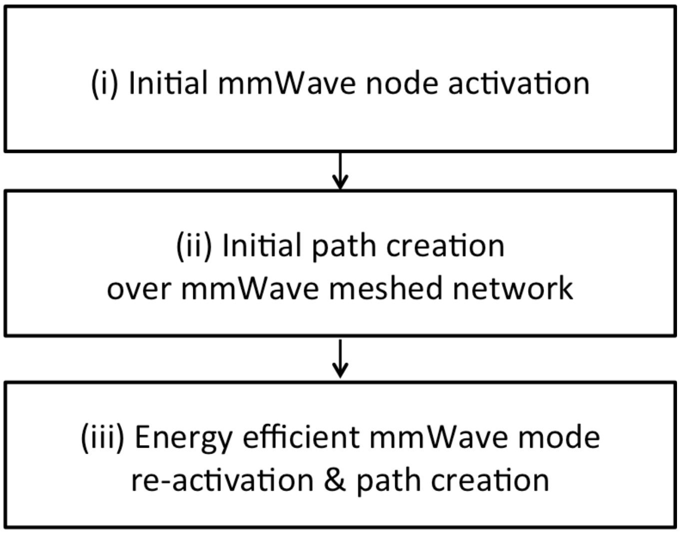


Figure 83: Flow chart of EMMA algorithm for mmWave mesh case.

(i) Initial mmWave node activation:
Step (i) determines initial ON/OFF status of each AP considering multi-tier connectivity of users with both LTE and mmWave access, and the goal of this applications is to reduce the total power consumption of the mmWave network as much as possible. In order to minimize the total power consumption, LTE should accommodate as many users as possible within its available bandwidth B_{LTE} and offloaded APs should be set OFF. As it is complicated to consider each user individually, an aggregated traffic demand T_i of users within the area of i-th AP is considered. If T_i is small and can be accommodated by LTE, LTE allocates some bandwidth b_i to the users as

$$b_i = T_i / \log_2(1 + \gamma_i),$$

to switch off AP_i, where γ_i is approximated SINR of signal from LTE macro BS to AP_i. If we define $i \in G_k$ as a state that T_i is accommodated by the k-th sector of LTE macro BS, the problem of initial mmWave node activation is described as follows.

$$for\ k = 1, 2, 3$$
$$\textbf{find}: group\ G_k\ s.t.$$
$$\textbf{minimize}: -|G_k|$$
$$\textbf{subject to}: \sum_{i \in G_k} b_i \le B_{\text{LTE}}$$

Where $|G_k|$ is the number of APs included in G_k. As a result, if T_i is accommodated by LTE, the corresponding users around AP_i will be accommodated by LTE, and AP_i can be set OFF to reduce power consumption. If AP_i is set ON, all the sectors of AP_i are activated regardless of the number of users in the area of AP_i.

(ii) Initial path creation over mmWave meshed network:
In step (ii), initial backhaul/fronthaul paths over mmWave meshed network are created for APs that were set ON in the step (i) from any sector of GW to satisfy users' traffic demand. If combinations of transmitters and receivers are given, graph-theoretical expression of APs and connectivity between them as nodes and edges enable us to find the shortest paths easily. Thus, the problem of initial path creation can be substituted for another problem to

find the optimal combination of the transmitter sectors in GW and receiver AP. Figure 84 shows a simple example to derive the combination of transmitter sectors and receiver AP. The filled cell is a hotspot with an aggregated traffic demand T which is larger than the capacity C_S of one sector in GW. In such a case, the optimized path should support backhaul/fronthaul route multiplexing from several sectors of GW to satisfy users' traffic demand. In the route multiplexing, one sector of GW supports x_1 and the other supports x_2 to fulfil the requirement as $x_1 + x_2 \geq T$. An reasonable way to determine the value of x_1 and x_2 to minimize power consumption of the network can be found by minimize TH defined as $TH = \sum Data \times Hop$, where Hop is the number of hops in the shortest path and $Data$ is x. By generalizing this example to a whole network with N_S sectors of GW and N_{AP} APs, the problem for initial path creation can be formulated as follows.

$$\begin{aligned}
&\textbf{find}: \boldsymbol{x}\ s.t.\\
&\textbf{minimize}: TH = \boldsymbol{f}^{\boldsymbol{T}}\boldsymbol{x}\\
&\textbf{subject to}:\\
&[A]\boldsymbol{t}_S = \boldsymbol{W}_S\boldsymbol{x} \leq C_S\mathbf{1}\\
&[B]\boldsymbol{t}_{AP} = \boldsymbol{W}_{AP}\boldsymbol{x} \geq \boldsymbol{a} \odot \boldsymbol{T}\\
&[C]\ \boldsymbol{x} \geq \mathbf{0}
\end{aligned}$$

Where $\boldsymbol{x} \in \mathbb{R}^{N_V}$ is the data amount to be transmitted from any sector of GW to any AP, $\boldsymbol{f} \in \mathbb{R}^{N_V}$ is the number of hops for $\boldsymbol{x}$, and $N_V = N_S \times N_{AP}$ is the total number of flows. $\boldsymbol{t}_S \in \mathbb{R}^{N_S}$ is the aggregated traffic load on the sectors of GW, $\boldsymbol{t}_{AP} \in \mathbb{R}^{N_{AP}}$ is the aggregated traffic supplied to the APs, $\boldsymbol{W}_S \in \mathbb{R}^{N_V \times N_S}$ is a mapping matrix between $\boldsymbol{t}_S$ and $\boldsymbol{x}$, $\boldsymbol{W}_{AP} \in \mathbb{R}^{N_{AP} \times N_S}$ is a mapping matrix between $\boldsymbol{t}_{AP}$ and $\boldsymbol{x}$, $\boldsymbol{a} \in \mathbb{R}^{N_{AP}}$ is the ON/OFF status of APs, and $\boldsymbol{T} \in \mathbb{R}^{N_{AP}}$ is the aggregated traffic of APs. The constraint [A] is on the capacity of each sector of GW, [B] ensures satisfaction of user's requests, and [C] assures the value of traffic not negative. By solving the above problem, the optimal paths can be created as $\boldsymbol{x}$ and corresponding shortest paths.

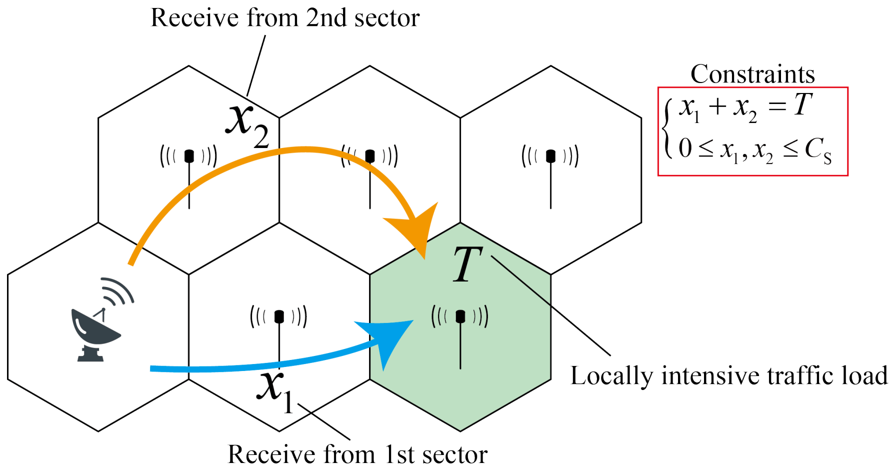


Figure 84: Concept of backhaul/fronthaul route multiplexing.

(iii) Energy efficient mmWave node re-activation & path creation:

If there are isolated APs that do not satisfy [B] in the step (ii), re-activation of APs for the purpose of relay is needed. The re-activated AP must communicate with the isolated AP at the maximum data rate, and must be nearest to the GW in terms of the number of hops among candidate APs for re-activation. From the viewpoint of power consumption, the number of re-activated APs should be small as much as possible. To achieve these requirements, the following procedure is employed for energy efficient AP re-activation and path creation.

1) Reset the capacity of sectors and links by considering the results from step (ii)
2) Tentatively re-activate all remaining APs and run step (ii) again
3) Pick up all combinations of the shortest paths and count the number of APs to be re-activated
4) Adopt a combination that minimizes the number of re-activated APs
5) Keep switching OFF the unnecessary APs

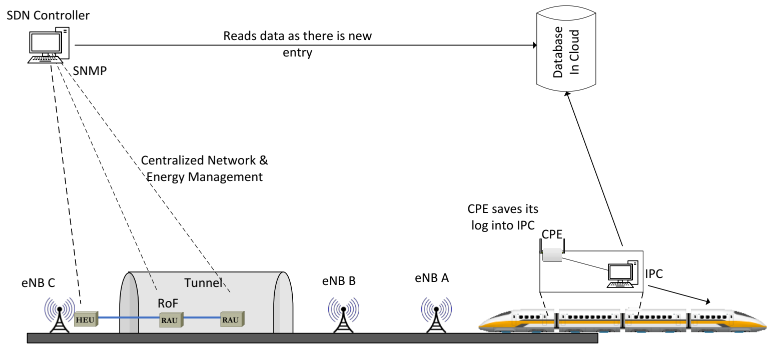
By this procedure, all isolated APs that are set ON can be connected with GW. It is noted that the number of activated backhaul/fronthaul links per AP depends on whether AP plays a role of relay for other APs or not. If AP does not have any relay links, the number of activated links will be one; otherwise it will be two or three.

12 Appendix V: Centralized Network and Energy Management (CNEM) for Radio-over-Fibre (RoF)

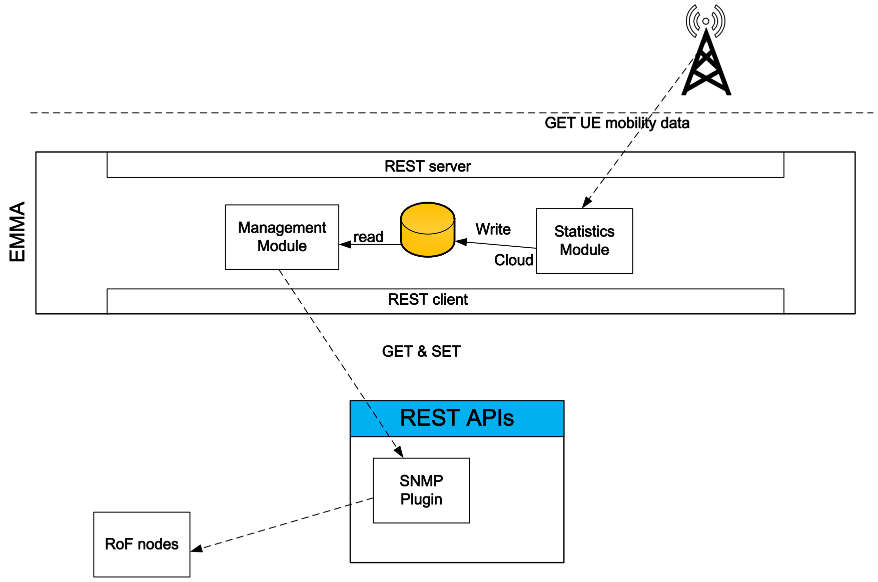
12.1 Description of EMMA for RoF

Energy Management for RoF is the special use case of EMMA application, which will be referred here and after as CNEM.

The CNEM application provides the energy management mechanism for power consumption optimization through turning off the idle RoF nodes.

Title	*Centralized Network and Energy Management for Radio-over-Fibre*
Description of the Application	CNEM is a network application with the main goal of managing the energy consumption of RoF nodes to optimize and minimize the energy footprint of the deployed distributed RoF nodes by leveraging the Crosshaul network without degrading the QoS of ground-to-train communications.
Picture	 *Figure 85: : Conceptual architecture of centralized network and energy management for RoF*
Business or technique drivers	The main driver is to reduce the energy consumption of deployed RoF nodes by turning them off when the trains are not in their proximity.
Use cases and Scenarios	In the case of RoF, the energy consumption can be reduced by dynamically managing the status of RoF Nodes (ON/OFF) and considering the mobility of the trains. Energy Management module leverages available context information, such as the mobility of the high-speed train and utilizes the real-time mobility information from the database indicating the relative location of the train along the deployed network. The SDN management application acts as an SNMP manager. It is able to monitor and control the status of the RoF system. The application reads the database updates in real-time and decides the status of the RoF system.

<table>
<tr><td rowspan="1">Technologies to be used or considered</td><td colspan="2">This application deals with the RoF nodes which basically constitute the Fronthaul. Thus it will be designed in a way to be agnostic to the Fronthaul transport technology and will be mapped to abstract energy parameters of Crosshaul components (i.e., RoF nodes)</td></tr>
<tr><td rowspan="3">Information requirements
(Inputs)</td><td>What</td><td>Who may provide this</td></tr>
<tr><td>Mobility information of the high-speed train.</td><td>Train gateways that attach Macro base stations deployed along the railway track</td></tr>
<tr><td>Information about the status of RoF nodes</td><td>XCI (SDN controller), through NBI</td></tr>
<tr><td rowspan="2">Outputs</td><td>What</td><td>Who may need this</td></tr>
<tr><td>Specification of ON-OFF status of RoF nodes</td><td>Application itself</td></tr>
<tr><td>App-specific functions</td><td colspan="2">• Statistics collection of high-speed train mobility from the macro base station.
• Computation and analysis of train relative location to the position of the RoF nodes.
• Classification of the direction of the high-speed train.
• Retrieving the status of the RoF nodes.
• Instructions for changing the status of the RoF nodes.</td></tr>
<tr><td>Required
(external) functions</td><td colspan="2">None</td></tr>
</table>

Functional Architecture	 *Figure 86: Energy management and monitoring application function description*
Implementation	An analytical approach supported by software emulation will be chosen as an initial methodology. A PoC of the implemented algorithms (OpenDaylight will be selected as an SDN controller, in addition to the real RoF node to carry out the PoC) The developed CNEM will be integrated into the high-speed train testbed which will be further evolved in WP5.
Relation to other applications	None

Algorithm

The initial algorithm for CNEM is described as pseudo code in the following (time index is dropped for simplicity):

Get current train location

Get train direction and speed estimation using current and previous train location

For each train T_i

For each RoF node R_j

S_j = current status of R_j ∈{ON, OFF}

If S_j == OFF

If (T_i is approaching R_j) & (T_i is within the coverage of R_j plus Guard_Distance_j)

S_j = ON

Else if S_j == ON

If (T_i is leaving R_j) & (T_i is beyond the coverage of R_j plus Guard_Distance_j)

$S\_j = OFF$

The algorithm for determining the optimum guard distance for maximum energy efficiency, subject to the requirement that the quality of ground-to-train communications is not degraded, is for further study. Besides, if precise location of train is not available, and other alternative information is used (such as physical cell ID in our case), the algorithm needs further modification. Some situations are not considered yet, such as outdated train location due to ground-to-train communications failure, and a more comprehensive algorithm will be provided in the next version.

13 Bibliography

[1] 5G-Crosshaul, "Deliverable 1.1, Initial specification of the system architecture accounting for feedback received from WP2/WP3/WP4".

[2] 5G-Crosshaul, "Deliverable 3.1, Baseline definition of XFE/XCI including specification of the southbound interfaces".

[3] C. Chen, "C-RAN, The Road Towards Green RAN," 2011.

[4] N. Chowdury, M. Rahman and R. Boutaba, "Virtual network Embedding with Coordinated Node and Link Mapping," in *IEEE INFOCOM 2009*, April, 2009.

[5] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. K. Ramakrishnan and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," *21st IEEE International Workshop on Local and Metropolitan Area Networks,* pp. 1-6, April 2015.

[6] S. Mehraghdam, M. Keller and H. Karl, "Specifying and placing chains of virtual network functions," *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference,* pp. 7-13, October 2014.

[7] A. Vishwanath, K. Hinton, R. Ayre and R. Tucker, "Modeling Energy Consumption in High-Capacity Routers and Switches," *IEEE Journal on Selected Areas in Communications,* vol. 32, no. 8, August 2014.

[8] P. Congdon, P. Mohapatra, M. Farrens and V. Akella, "Simultaneously Reducing Latency and Power Consumption in OpenFlow Switches," *IEEE Transactions on Networking,* vol. 22, no. 3, June 2014.

[9] 5G-Crosshaul, "Deliverable 2.1, Derailed analysis of the technologies to be integrated in the XFE based on previous internal reports from WP2 and WP3".

[10] G. Schaffrath and e. al., "Network Virtualization Architecture: Proposal and Initial Prototype," in *ACM VISA*, 2009.

[11] ONF, "SDN Architecture," Open Networking Foundation, June 2014.

[12] Y. Sia, S. Jiang, T. Zhou and S. Hares, "NEMO (NEtwork MOdeling) Language," 2015.

[13] ETSI, "Network Functions Virtualization (NFV); Management and Orchestration"," ETSI GS NFV-MAN 001, v1.1.1, Dec. 2014.

[14] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!," *IEEE access,* vol. 1, pp. 335-349, 2013.

[15] A. Ghosh, T. A. Thomas, M. C. Cudak, R. Ratasuk, P. Moorut, F. W. Vook, T. S. Rappaport, G. R. MacCartney, S. Sun and S. Nie, "Millimeter-wave enhanced local area systems: A high-data-rate approach for future wireless networks," vol. 32, no. 6, pp. 1152--1163, 2014.

[16] S. Rangan, T. S. Rappaport and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE,* vol. 102, no. 3, pp. 366--385, 2014.

[17] S. Singh, M. N. Kulkarni, A. Ghosh and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE Journal on Selected Areas in Communications,* vol. 33, no. 10, pp. 2196--2211, 2015.

[18] K. Sakaguchi, G. K. Tran, H. Shimodaira, S. Nanba, T. Sakurai, K. Takinami, I. Siaud, E. C. Strinati, A. Capone, I. Karls and others, "Millimeter-wave evolution for 5G cellular networks," *IEICE Transactions on Communications,* vol. 98, no. 3, pp. 388--40, 2015.

[19] http://www.etsi.org/deliver/etsi_gs/NFVINF/001_099/010/01.01.01_60/gs_NFV-INF010v010101p.pdf.

[20] Network Functions Virtualisation (NFV); Architectural Framework, ETSI GS NFV 002 v1.2.1 (2014-12) online http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf.

[21] Part of the OpenMANO project, an open source project that provides a practical implementation of the reference architecture for Management & Orchestration under standardization at ETSI’s NFV ISG https://github.com/nfvlabs/openmano.

[22] Open source software for creating private and public clouds http://www.openstack.org/.