Dynamically Reconfigurable Optical-Wireless Backhaul/Fronthaul with Cognitive Control Plane for Small Cells and Cloud-RANs

D3.2 Design and evaluation of scalable control plane, and of mobility aware capabilities and spatio-temporal demand prediction models

This project has received funding from the European Union's Framework Programme Horizon 2020 for research, technological development and demonstration

Advanced 5G Network Infrastructure for the Future Internet

Project Start Date: July 1st, 2015 Duration: 36 months

H2020-ICT-2014-2 671551 June 31st, 2016 – Final Version

Project co-funded by the European Commission

Under the H2020 programme

Dissemination Level: Public

Grant Agreement Number:	671551
Project Name:	Dynamically Reconfigurable Optical-Wireless Back-haul/Fronthaul with Cognitive Control Plane for Small Cells and Cloud-RANs
Project Acronym:	5G-XHaul
Document Number:	D3.2
Document Title:	Design and evaluation of scalable control plane, and of mobility aware capabilities and spatio-temporal demand prediction models
Version:	Final
Delivery Date:	July 1st, 2017 (**July 15th, 2017**)
Responsible:	I2CAT Foundation (**I2CAT**)
Editor(s):	Daniel Camps-Mur (**I2CAT**)
Authors:	Daniel Camps-Mur (**I2CAT**), Paris Flegkas (**UTH**), Koteswararao Kondepu (**UNIVBRIS-HPN**), Vladica Sark (**IHP**), Jesús Gutiérrez (**IHP**), Eduard García (**I2CAT-UPC**), Ilker Demirkol (**I2CAT-UPC**), Qing Wei (**HWDU**), Artur Hecker (**HWDU**), Emmanouil Pateromichelakis (**HWDU**), Panagiotis Spapis (**HWDU**), Xun Xiao (**HWDU**), George Limperopoulos (**COS**), Eleni Theodoropoulou (**COS**), Constantinos Filis (**COS**), Kostas Choumas (**UTH**), Dimitris Giatsios (**UTH**), Peter Legg (**BWT**), Jim Zou (**ADVA**), Stefan Zimmermann (**ADVA**)
Keywords:	Scalable Control plane, SDN, multitenancy, virtualisation, QoS, TE
Status:	Final
Dissemination Level	**Public** / Confidential
Project URL:	http://www.5g-xhaul-project.eu/

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

This document constitutes deliverable D3.2 from the 5G-XHaul project, and reports on the work carried out during the second year of the project within WP3.

This document builds upon the structure defined in deliverable D3.1, which defined the building blocks of the 5G-XHaul control plane, and extends it in several ways.

First, we develop the generic control plane architecture that was introduced in D3.1. In particular, we define the data-paths of various network elements, and the interfaces between controllers in the control-plane hierarchy. We also propose and evaluate various mechanisms that reduce signaling between network elements and controllers, and increase the reliability of the control plane.

Second, we delve into Traffic Engineering (TE) mechanisms for wireless based control plane areas. Efficient TE mechanisms are critical at the wireless edge where bandwidth is more constrained. Here, we propose specific mechanisms for millimetre wave (mmWave) and Sub-6 data-plane technologies.

Third, we specify in greater detail the south-bound interfaces used by the 5G-XHaul SDN controllers to control the various data-plane technologies considered in the project, including Time Shared Optical Network (TSON), Wavelength Division Multiplexing Passive Optical Network (WDM-PON) and mmWave.

Finally, we report on our investigations about the benefits of tightly integrating the Mobile and the Transport Networks. In particular, we describe mechanisms that allow the Radio Access Netwkork (RAN) to localise user when they move, and define various interfaces that allow the transport network to exploit information from the Mobile network.

1 Introduction

The main goal of the 5G-XHaul control plane is to provide an operator with the means to manage in an unified manner the heterogeneous 5G-XHaul data plane. The 5G-XHaul data-plane is composed of wireless connectivity between the Small Cells (SC) and the Macro Cells (MCs), a high capacity optical access technology based on WDM-PON, and a metro-optical segment based on Time Shared Optical Network (TSON). The interested reader is referred to deliverable D2.2 [2] for an in-depth description of the 5G-XHaul data-plane architecture.

Deliverable D3.1 [3] introduced a preliminary design of the 5G-XHaul control plane architecture, which builds upon the following concepts:

- **A hierarchical SDN control plane**, where the purpose of the hierarchy is two-fold. Firstly, establishing a hierarchy of Software Defined Networking (SDN) controllers allows to design and evolve control planes tailored to a given technology domain, for example wireless and optical, and secondly, it enables scalability in two ways. Firstly, the controller itself is in charge of a reduced number of devices and so its computational and IO requirements are relaxed. Secondly, limiting the size of a control plane area limits the number of Forwarding Information Base (FIB) entries maintained by the network elements. The latter feature is critical when one wants to introduce SDN to devices deployed at the edge of the newtork, e.g. SCs, where the cost pressure is very high.
- **An L2 overlay approach to network slicing**. Following the 5G architectural principles laid out in [8], the 5G network can be seen as a set of connected Virtual Network Functions (VNFs). Hence, the main role of the 5G-XHaul transport network is to provide a network abstraction interconnecting a set of tenant-specific VNFs. 5G-XHaul proposes a L2 abstraction where VNFs belong to a given Layer 2 Segment ID (L2SID), and a set of VNFs and L2SIDs compose a transport network slice. L2SIDs are implemented by maintaining per-tenant state (e.g. VNF MAC addresses) at the edge of the network, and mapping this state to pre-established transport tunnels using MACinMAC technology.
- **Per-technology Traffic Engineering (TE) and South-Bound interfaces**. Being the 5G-XHaul transport network composed of wireless and optical segments, specific TE mechanisms are required for each domain. In particular, in the wireless segment, where capacity is more limited, effective TE mechanisms are required that monitor and control transport tunnels to maintain the Service Level Agreements (SLAs) of a transport network slice.
- **A close interaction between the Mobile and Transport Networks.** The 4G Mobile Network can be seen as an overlay over an IP transport network. Thus, in 4G there are no interactions between the Mobile Network and the Transport Network, which is assumed to be overprovisioned. 5G-XHaul posits that this assumption will no longer hold in 5G, where the Backhaul (BH) and Fronthaul (FH) segments may become a bottleneck. Therefore, the 5G-XHaul control plane should be able to work together with the control plane of the 5G Mobile Network to optimize the end to end performance.

The document at hand further develops the previous concepts, while including exhaustive performance evaluations for the various mechanisms proposed. This deliverable is organized as follows.

Chapter 2 refines the 5G-XHaul control plane architecture. In particular, it introduces the notion of Local Controllers inside some of the network elements to favor latency and scalability. It refines the interfaces between the various controllers in the SDN hierarchy, and introduces the ABNO framework as a mechanism to implement the higher level 5G-XHaul SDN controllers. This chapter also presents and evaluates mechanisms that ease the scalability and reliability of an SDN based Transport Network, such as Dynamic Flow Rules (DFR) and the resilient control plane protocol.

Chapter 3 investigates SDN-based TE mechanisms. This chapter focuses on the wireless domain, since it is the most constrained in terms of capacity, and develops specific mechanisms for mmWave and Sub-6 wireless transport devices, since both technologies are considered in the 5G-XHaul data-plane described in deliverable D2.2 [2]. This chapter proposes mechanisms for path allocation, and for fast re-routing in case of link breaks.

Chapter 4 provides a detailed description of the SDN south-bound interfaces that are being developed to control the wireless and optical network elements considered in 5G-XHaul. In particular, the chapter describes the interface between a TSON edge node and the TSON controller, where emphasis is made on the mechanisms to specify the time slot allocation that provides sub-wavelength bandwidth granularity in TSON. Then, two NETCONF-based interfaces are defined, one to manage the WDM-PON, and another one to manage mmWave devices.

Chapter 5 delves deeper into the interactions between the Mobile and the Transport Network. First, the chapter reports on a measurement campaign carried out in Greece to measure several Mobile Network parameters. Despite surveying a 4G network the collected parameters have been instrumental in evaluating some of the mechanisms described in this chapter. Then, the chapter progressively introduces mechanisms that leverage the knowledge about the Mobile Network to optimize the Transport Network. First, the chapter describes localization techniques that would allow a Mobile Network to faithfully follow the movement of a user, a feature not available today. Then, the chapter presents two control mechanisms that exploit knowledge about user mobility, available in the Mobile Core Network, to provide per-user awareness in the transport, and to increase network performance.

Chapter 6 summarizes and concludes this document.

Finally, it is worth noting that some of the mechanisms described in this deliverable have been extended through several peer-reviewed publications, which are properly referenced throughout the document.

2 Refinement of 5G-XHaul control plane design: architecture, virtualization, and scalability aspects

In Deliverable 3.1 [3] we introduced a first blueprint of a control plane architecture, taking into account the requirements described in the same document. In this section, we elaborate further in our control plane design, providing more details on the data path structure of the transport network elements, the hierarchy and roles of the different controllers and the messages exchanged among the controllers.

This chapter is structured as follows. Section 2.1 provides a brief summary of the 5G-XHaul control plane architecture. Section 2.2 dives deeper on this architecture, providing detailed descriptions of the datapath structure in the various network elements, and the controller hierarchy. Section 3 proposes and evaluates Dynamic Flow Rules (DFR), a mechanism to reduce signalling between network elements and the controller in a transport network. Finally, Section 2.4 describes the Reliable Control Protocol (RCP), a mechanism to increase resiliency of the control plane connection between network elements and the controller.

2.1 Summary of the 5G-XHaul architecture

2.1.1 Design principles

The 5G-XHaul architecture is based on the following design principles:

- Full address space virtualization is offered through an overlay, implemented using encapsulation at the edge of the transport network. This means that different tenants can use overlapping L2 or L3 address spaces.
- Data plane scalability is achieved by isolating the forwarding tables of the transport network elements inside the 5G-XHaul infrastructure from any tenant related state (overlay). This is again achieved by encapsulating tenant frames at the edge of the network into transport specific tunnels.
- Scalability of the SDN control plane is achieved introducing the concept of areas. An area defines a set of transport network elements that are under the control of a logically centralized SDN controller. A control plane hierarchy is introduced whereby higher-level controllers coordinate the actions of area level controllers. At the lowest level of the hierarchy, the transport network elements feature local agents, which directly interact with the underlying datapath.
- Finally, the vision of converged heterogeneous technology domains, e.g. wireless and optical segments in the transport network, is enabled by: i) the previously introduced areas, which embody a single type of transport technology (e.g. wireless mesh, optical or Ethernet), and ii) a transport adaptation function (TAF) that maps the per tenant traffic at the edge nodes to the transport specific tunnels of a given area.

2.1.2 Dataplane Abstraction

In terms of the dataplane abstraction, the 5G-XHaul transport network can be seen as a collection of isolated virtual networks. Each of these virtual networks is controlled by a tenant, which brings its virtual entities, namely Virtual Network Functions (VNFs) and virtual Datapaths (vDPs). The latter are tenant-controlled datapaths, which can be instantiated for example using a software switch.

In order to support the previous principles, three types of transport nodes are defined in 5G-XHaul, which are depicted in Figure 2-1. At first, Edge Transport Nodes (ETNs), connect the tenant virtual entities (VNFs and vDPs) to the 5G-XHaul transport network, maintain the corresponding per-tenant state, and encapsulate tenant traffic into transport specific tunnels. Secondly, Inter-Area Transport Nodes (IATNs), support the necessary functions to connect different areas, which may be implemented using different transport technologies. Finally, regular Transport Nodes (TNs), support an area specific transport technology, and provide forwarding services between the ETNs and IATNs of that area.

All transport nodes embed one major function, the Forwarding Information Base (FIB). FIB is in charge of forwarding packets between VNFs/vDPs, which are either collocated in a single ETN or bound to different ETNs. In the second case, packets are inserted into pre-instantiated transport tunnels, implemented using encapsulation. Traffic from multiple slices can be combined into a single tunnel.

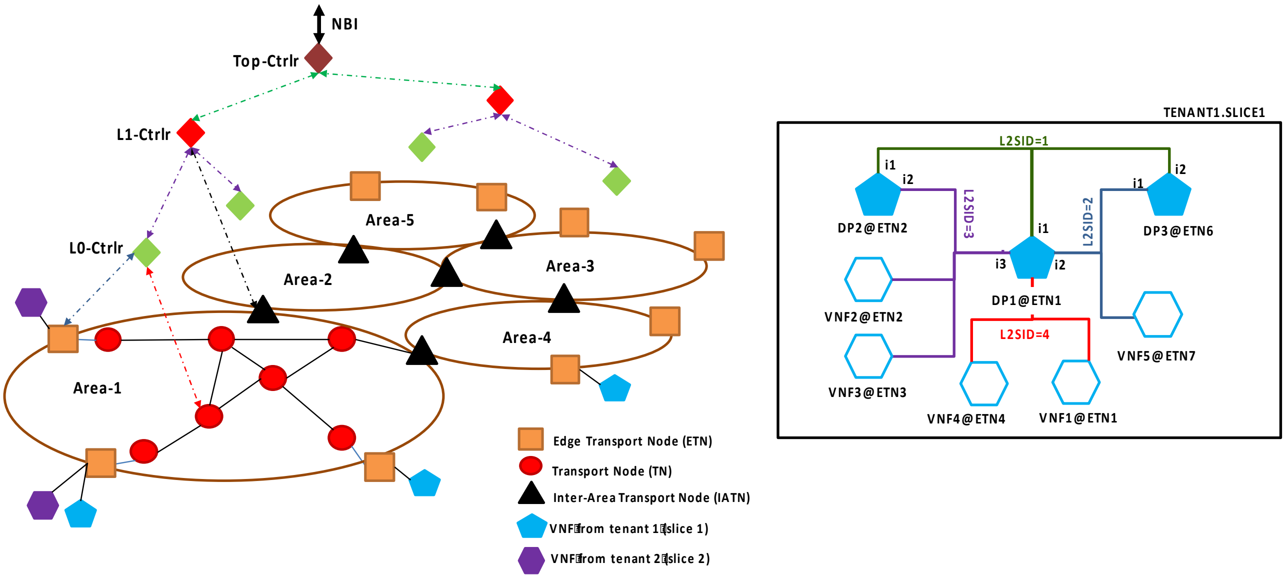


Figure 2-1: (a) 5G Transport Control Plane architecture, (b) Slice abstraction towards tenant.

The right part of Figure 2-1 depicts the abstraction that the 5G-XHaul control plane presents to each tenant, namely the transport network slice. Each tenant's goal is to connect through the 5G-XHaul network a set of tenant defined Virtual Network Functions (VNFs), deployed in distributed compute and storage facilities. Thus, the tenant is allowed to group VNFs on virtual layer two segments, known as Layer 2 Segment IDs (L2SIDs). Different L2SIDs can be connected through a vDP. The interested reader is referred to [2] for a detailed description of how a tenant slice maps on the physical architecture.

2.2 Refinement of 5G-XHaul control plane architecture

The control plane architecture of 5G-XHaul is based on a structured hierarchy of four types of controllers. At the lowest level we find the Local Agents (LAs), which are embedded in the transport network elements. For scalability reasons, these elements are organized into areas, as already mentioned. The Level-0 controllers, also referred to as Area Controllers (ACs), are responsible for control plane functionalities within a single area. Moving up the hierarchy, we find the Level-1 controllers, which are in charge of maintaining connectivity among a set of Level-0 areas. Finally, the Top-Controller is responsible for provisioning per-tenant slices and orchestrating connectivity across different 5G-XHaul areas. As we move up the hierarchy, the controllers operate with a higher level of abstraction. Below, we provide details on the functionalities involved in these types of controllers.

2.2.1 Two-Level Control Area Architecture

In the context of a single area, we propose a two-level SDN control architecture combining a longer-term, offline centralized control implemented at the Level-0 Area Controller with a more dynamic, on-line distributed handling of traffic fluctuations and events implemented at the Local Agents embedded in each transport node, as shown in Figure 2-2.

The AC is responsible for proactively setting up all tunnels and allocating the available resources in the area it controls, including alternatives used for load balancing and/or failure recovery for a given provisioning period. It has as input the network topology (including link capacities), the estimated traffic demand, and resource-oriented policy directives set by the Operator. The predicted traffic demand is the output of a process that translates and maps the requirements of the expected customer traffic, e.g. tenants that want to interconnect the VNFs located at specific ETNs, to the physical topology and aggregates them by "adding" their requirements such as the throughput into ingress-egress pairs and QoS Class supported demand, increasing in this way the scalability of the AC operation. The produced aggregated traffic forecast is based both on the SLAs as well as historical network monitoring data. In any case, it utilizes network-wide information, received from the network nodes through polling and/or asynchronous events. Based on this information, the AC can adapt to the network dynamics and update the transport tunnels in order to satisfy the tenant requirements.

The LAs are distributed controllers in every node (ETNs, IATNs and TNs) and are responsible for reacting to network events, performing dynamic allocation of resources to different QoS classes or slices and mapping or rerouting incoming traffic by VNFs to the tunnels provided by the AC. They are also responsible for issuing alarms to the AC in case there is no more available capacity to accommodate the incoming flows according to the configuration produced by the AC. In general, it operates on a shorter time-scale than the AC. In the following subsections, we elaborate further on the operations of the LAs for the three types of transport nodes.

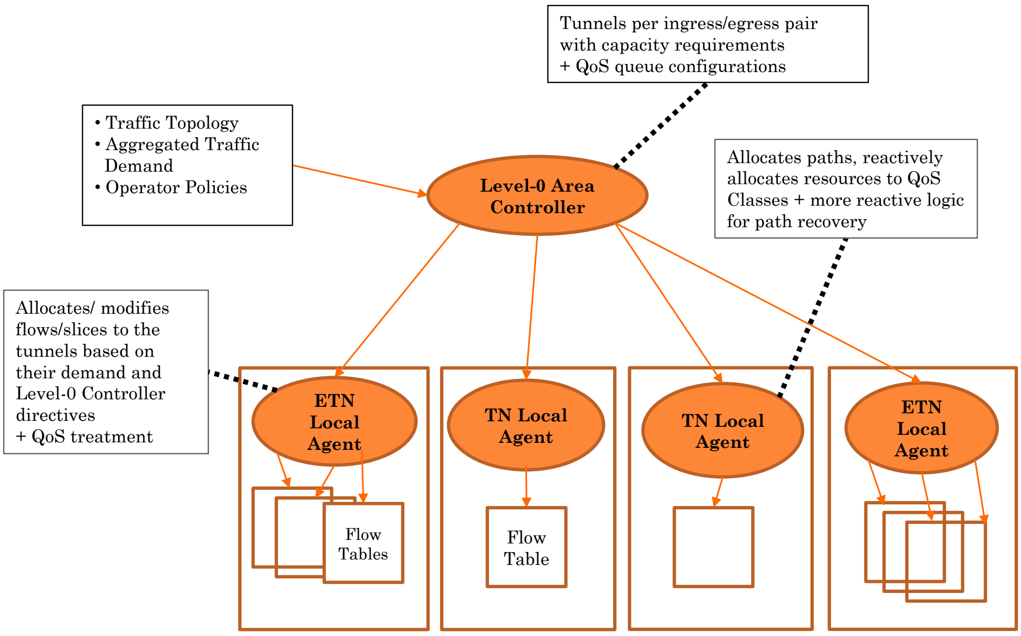

Tunnels per ingress/egress pair
with capacity requirements
+ QoS queue configurations
• Traffic Topology
• Aggregated Traffic Demand
• Operator Policies
Level-0 Area Controller
Allocates paths, reactively allocates resources to QoS Classes + more reactive logic for path recovery
Allocates/ modifies flows/slices to the tunnels based on their demand and Level-0 Controller directives
+ QoS treatment
ETN Local Agent
TN Local Agent
TN Local Agent
ETN Local Agent
Flow Tables
Flow Table

Figure 2-2: Two-level Control Area Architecture.

2.2.2 The ETN Local Agent and datapath structure

The main function of an ETN is to host virtual entities from several tenants, and to offer a datapath abstraction connecting them to the transport network. Apart from the ports where local virtual entities (VNFs, vDPs) are connected, it also features a port that connects it to all the other ETNs/IATNs in its area through transport tunnels; we will refer to it as the ETN's *external port*.

The Local Agent of each ETN maintains a set of four mappings:

i) *port* → *L2SID*

ii) *<L2SID, destination MAC>* → *<port, destination ETN>*

iii) *destination ETN* → *tunnel ID*

iv) *L2SID* → *transport slice ID*

Specifically for the second mapping, whenever the specified virtual entity belongs to another ETN, the port value is filled with the external port's number. So this mapping is a general directory of all the virtual entities this ETN should be aware of, both local and remote, where a virtual entity is uniquely defined by the <L2SID, MAC address> tuple.

The ETN's LA is responsible for updating these mappings. Note that the ETN needs to be aware of any addition, removal or migration of virtual entities attached to a virtual L2 segment where at least one of the ETN's own entities is connected, no matter where this takes place in the transport network. This is implemented with the hierarchical control plane of 5G-XHaul. The updating procedure is proactive. Whenever a tenant requests a modification in its virtual network (e.g. VNF migration to another ETN), the 5G-XHaul operator first makes sure that the rules for all potential flows from/to the affected virtual entities are installed before it allows the tenant to start sending flows through them.

In terms of the datapath structure, an ETN implements FIB and TAF, and its OpenFlow-controlled datapath features four flow tables (cf. Figure 2-3):

- Table 0: packet decapsulation
- Table 1: packet classification for layer 2 segmentation

- Table 2: packet forwarding (FIB)
- Table 3: L2SID tagging and packet encapsulation

The utilization of four flow tables, instead of using the more typical approach of a single flow table, enables the decrease of the total number of flow entries in each ETN. Note that this procedure assumes that all packets are Ethernet frames and Provider Backbone Bridging (PBB) is used for their encapsulation. The use of PBB for encapsulation has already been discussed in D3.1 [3]. In order to support multiple flow tables and PBB headers, OpenFlow version 1.3 or higher is required and should be supported by the ETN datapath.

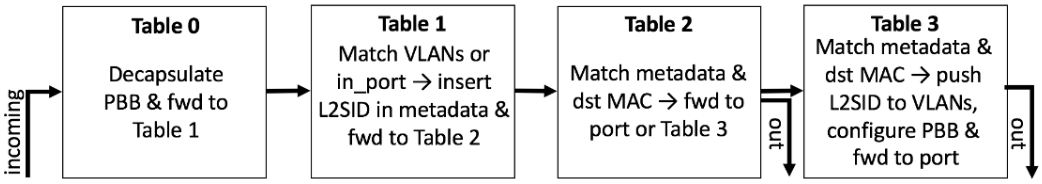

incoming
Table 0
Decapsulate PBB & fwd to Table 1
Table 1
Match VLANs or in_port → insert L2SID in metadata & fwd to Table 2
Table 2
Match metadata & dst MAC → fwd to port or Table 3
out
Table 3
Match metadata & dst MAC → push L2SID to VLANs, configure PBB & fwd to port
out

Figure 2-3: ETN as a datapath.

Packets arriving at the ETN datapath first enter *Table 0*, which decapsulates them if they have been received through the external port. Packets received from local virtual entities are not modified. In both cases, packets are moved to *Table 1* without a PBB header.

Table 1 marks the metadata of each packet it handles with the L2SID of the layer two segment that the source virtual entity belongs to, and moves the packet to *Table 2*. In the case that the packet has been received from a local virtual entity, the *port →L2SID* mapping is used to determine the L2SID. If the packet has been received from entities hosted in other ETNs, then their L2SID is defined by their virtual Local Area Network (VLAN) tags (one VLAN tag for maximum 4096 globally unique L2SIDs or two VLAN tags for even more). The VLAN tags are pushed before packet encapsulation in the source ETN.

Table 2 forwards the packet to the virtual entity that i) is identified by the packet's destination MAC address and ii) belongs to the segment identified by the L2SID of the packet's metadata. The forwarding process utilizes the mapping *<L2SID, destination MAC> →<port, destination ETN>*. If the port that is pointed to is the external port, then the packet is forwarded to *Table 3*. Otherwise, it is forwarded to the designated local port.

Table 3 first pushes the L2SID of the packet it receives (visible in its metadata) in its VLAN tags. Then it needs to push the PBB header. It puts its own address in the Backbone Source Address (B-SA) field. It puts the destination ETN address from the mapping *<L2SID, destination MAC> →<port, destination ETN>* in the Backbone Destination Address (B-DA) field. Then, it puts the Transport Slice ID from the mapping *L2SID→transport slice ID* in the Backbone Service Instance Identifier ID (I-SID). Recall that this ID is used for enforcing slice specific policies in the transport network, so we need it to be visible in the PBB header. The tunnel ID from the mapping *destination ETN →tunnel ID* is placed in the Backbone VLAN ID field (B-VID). Finally, the packet is forwarded to the external port of the ETN.

As an example, Figure 2-4 illustrates the encapsulation/decapsulation process of a packet going from VNF 3 to VNF 4. In the bottom row we can see the tenant view of the Ethernet frames exchanged, which is agnostic of transport-specific details. In the top row, we see the form in which the packet traverses the transport nodes, with its inner VLAN tag set with its L2SID, and encapsulated with the PBB header.

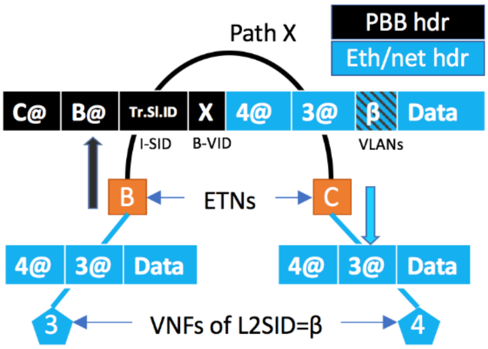


Figure 2-4: Packet encapsulation/decapsulation.

2.2.2.1 Load Balancing and Failover at the ETN

Mechanisms for load balancing across different tunnels towards a given destination ETN can be easily incorporated into the structure described above. For instance, the mapping “destination ETN → tunnel ID” can be extended to point to a list of tunnel IDs instead of just one. These tunnel IDs would then be included in the action buckets of an OpenFlow group of type “Select”, which can be configured to run an appropriate mechanism to select a bucket, such as round-robin or a bucket-weight-based mechanism. *Table 3* in the above structure would be modified so that it forwards the packets to this group.

In a similar way, failover mechanisms can be implemented by using groups of fast failover type, where different buckets insert different tunnel IDs. The liveness of each bucket in these groups would be controlled by an algorithm which monitors the health of the paths towards each ETN, and again *Table 3* would be modified to forward packets to these groups.

Both load balancing and failover can also be implemented at TN level, however, using the same tunnel ID. In chapter 3 of this deliverable, we will revisit such mechanisms in more detail.

2.2.2.2 Handling broadcast and multicast packets

A broadcast packet is a packet arriving at a local port of an ETN with destination MAC address FF:FF:FF:FF:FF:FF. From a tenant’s point of view, protocols based on L2 broadcast, such as Address Resolution Protocol (ARP), should work out of the box, hence it is mandatory to provide the required support. To handle this type of packets, the LA of the ETN has special entries in its mapping <L2SID, destination MAC> → <port, destination ETN>. Specifically, for each L2SID present in this ETN, there is an entry matching this L2SID and the broadcast MAC, pointing to a *list* of <port, destination ETN> tuples. This list contains all local ports connected to virtual entities of this L2SID, and all remote ETNs hosting such entities. The LA builds and updates this list dynamically, whenever it receives instructions from its area controller regarding addition, modification or removal of virtual entities belonging to this L2SID.

In terms of the required flow table entries, the LA pushes entries in *Table 2* where broadcast packets of each L2SID are matched and forwarded to respective OpenFlow groups of type “All”, whose buckets correspond to the specified local ports for each L2SID. In groups of type “All”, if a bucket directs a packet explicitly out the ingress port, then this packet clone is dropped. Hence, we do not need to have different matches for each ingress port. If there are virtual entities of this L2SID in remote ETNs, pipeline processing continues by forwarding the packet to *Table 3*. The latter features a flow entry for broadcast packets of each L2SID, forwarding these packets to groups of type “All”, whose buckets push the PBB header corresponding to each remote destination ETN hosting virtual entities of this L2SID before forwarding the packet to the external port. Broadcast packets that have arrived from the external port will not be forwarded to the external port (which would create loops), because of the property of the groups of type “All” described, to drop packet clones towards the ingress port.

As a side note, the procedure followed for broadcast packets can also be adopted for handling unknown unicast addresses. In theory such a case should never occur. However, in case such an event is encountered in practice, it might be desired to broadcast packets towards all currently known destinations of the specified L2SID, rather than drop them. In parallel, a pull request towards the area controller could report the event, so that proper instructions are received.

The procedure for multicast packets is very similar to the one described for broadcast packets. The LA, after being informed by its area controller about a multicast group and which virtual entities it entails, inserts an entry in the mapping <L2SID, destination MAC> → <port, destination ETN>, corresponding to the multicast MAC address, which again points to a list of <port, dst ETN> tuples. The port numbers are copied from the respective entries for the individual virtual entities. Given this entry, the process of programming the appropriate flow table entries is identical to the broadcast case discussed.

2.2.3 The IATN Local Agent and datapath structure

The IATN is responsible for moving packets across neighbouring areas. It is agnostic of L2SIDs and tenant-specific details; its role is to accurately forward packets according to their PBB header. The LA of each IATN maintains a set of two mappings:

i) *<destination ETN, in_tunnel ID> →<out_tunnel ID, out_port>*

ii) *<in_port, in_transport_slice ID, out_port> →out_transport_slice ID*

The first mapping is used to determine the tunnel ID that will be used for forwarding each packet, and the port to which it will be forwarded. The tunnel ID is part of the keys in this mapping, to allow for multiple end-to-end paths towards a given destination ETN. Also, note that in an IATN each port corresponds to a specific area. The second mapping contains the translations of transport slice identifiers between incident areas, since transport slice identifiers are area-specific.

The IATN LA needs to be aware of all potential ETN addresses in the network, as well as tunnel and slice IDs of different areas. This information cannot be provided by a L0-controller. Therefore, the IATN LA receives instructions directly from the corresponding L1-Controller, sitting on top of the controllers of the incident areas. In the case that the IATN is connecting areas belonging to different L1-areas, then it receives related instructions directly from the Top Controller. In the process of establishing an end-to-end path, the instructions from higher layer controllers to the IATN LA are sent after these controllers have received acknowledgements from the involved underlying area controllers about successful area tunnel setup, and have been thereby informed about the respective tunnel IDs and transport slice IDs.

The datapath of the IATN consists of two flow tables, whose operations are depicted in Figure 2-5. The first table matches the destination ETN and the tunnel ID fields of the received packet. Upon a match, an update of the tunnel ID and an output port setting are added to the action set, while the output port is also copied in the metadata. Then, the packet is moved to the second table, which matches the ingress port, the transport slice ID and the output port (from the metadata). Upon a match, it updates the transport slice ID with the possibly new value. The action set is then executed, and the packet leaves towards the port connecting to the output area with its tunnel and transport slice IDs updated.

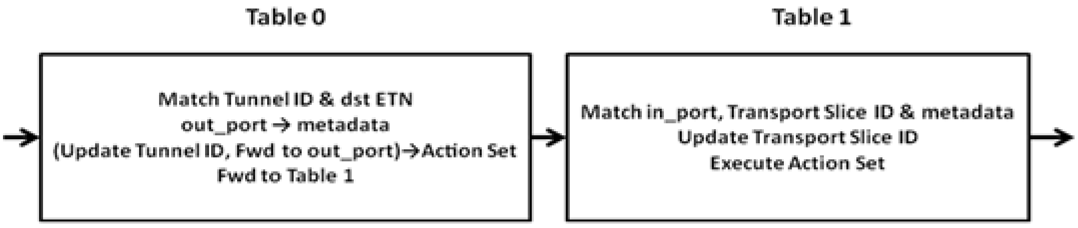


Table 0
Table 1
Match Tunnel ID & dst ETN
out_port → metadata
(Update Tunnel ID, Fwd to out_port)→Action Set
Fwd to Table 1
Match in_port, Transport Slice ID & metadata
Update Transport Slice ID
Execute Action Set

Figure 2-5: IATN as a datapath.

2.2.4 The TN Local Agent and datapath structure

Each TN Local Agent maintains a fundamental mapping between tunnel IDs and output ports, which enables the minimum functionality of routing packets towards the destination along a given path. Note that we generally consider unidirectional paths.

In addition, transport-slice IDs can be mapped to a specific QoS class in the datapath, so that queue-based policies such as rate-limiting, minimum rate guarantee or prioritization of certain flows can be implemented. In this way, the transport network is able to enforce slice or tenant specific policies.

In practice, some TNs might correspond to low cost datapaths not suited to host Local Agents. In this case, the area controller will communicate directly with the TN datapath through the OpenFlow protocol.

2.2.5 Area Controller structure

The (Level-0) Area Controller can be logically separated into two subcomponents, the ETN Controller subcomponent (ETNC) and the TN Controller subcomponent (TNC). The first is responsible for mappings between tenant traffic and the transport network through the orchestration of the ETNs in its area. The second is responsible for setting up transport tunnels in the area, monitoring their state, orchestrating QoS provision policies, as well as load balancing and failover. The TNC subcomponent maintains an up-to-date connectivity graph between the TNs in its area, the external interfaces of the ETNs in its area, and the IATN interfaces connecting the IATNs with this area. Role distribution allows the two subcomponents to be deployed in different locations, if required or preferred.

2.2.6 Higher Layers Control Plane Description

5G-XHaul control plane architecture, shown in Figure 2-6, introduces the concept of areas that define a set of transport network elements that are under the control of a logically centralized SDN controller. Coordination between area level (L0) controllers is introduced by a (recursive) control plane hierarchy that permits to establish, manage and maintain end to end connectivity across heterogeneous technology/vendor domains in a practically implementable way. Within 5G-XHaul, those segments include wireless and optical segments in the mobile backhaul, but this architecture may be extended to provide a homogeneous SDN-coordinated connectivity throughout the whole network.

To achieve the scalability, time sensitivity, and flexibility required by 5G control plane, independent areas are orchestrated by upper hierarchical layers or, depending on the situation, by the network management and orchestration functions (MANO). The role of MANO will be to trigger the establishment of the required connectivity at network creation process, or the automated, human-assisted or not, optimization and management of the connectivity in the network planning and optimization processes of the network. We have already referred to the envisioned integration with MANO in D2.1 [1], here we elaborate more on the concept.

5G-XHaul proposes the use of Application-Based Network Operations (ABNO) [10] to orchestrate multiple network domains. ABNO can be seen as a technology to instantiate the Level-1 and Top controllers that appear in the 5G-XHaul control plane architecture. The use of the Control Orchestration Protocol (COP) [11] is employed to orchestrate multiple domains.

Level-1 network controller, is in charge of coordinating Level-0 controllers connected to it, whereas Top controller is in charge of taking requests from application plane and translating them into high-level commands for the Level-1 network controllers.

Proposed SDN Interfaces

Overall architecture follows the Open Networking Foundation (ONF) architecture [9] using two main Controller plane interfaces (CPI): Application-controller plane interfaces (A-CPI), Data-controller plane interfaces (D-CPI).

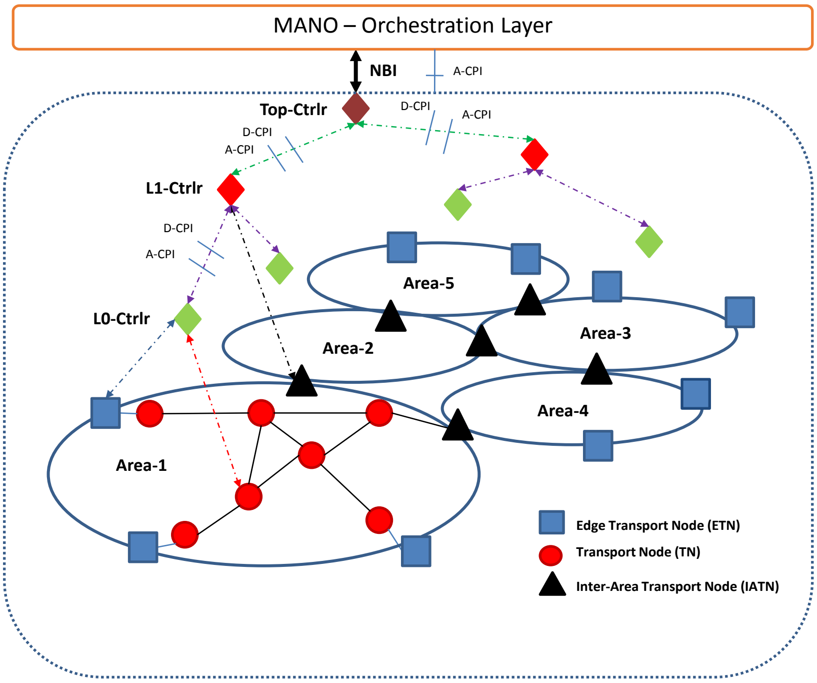


Figure 2-6: 5G-XHaul Control Plane Interfaces.

Both Top and Level-0 controllers satisfy client (higher layers working as SDN applications) requests, and request connectivity to the underlying resources.

The A-CPI interface represents the interaction by which a SDN-Application requests abstract resources as exposed by the SDN-Controller, whereas the D-CPI interface represents the interaction by which an SDN-Controller requests/consumes underlying network resources.

Top-controller

Top-controller is defined as a parent controller or a centralized "controller of controllers", which handles the automation of end-to-end connectivity provisioning, working at a higher, abstracted level and covering inter-domain aspects between different areas orchestrated by it. Top controller interfaces with the Layer-1 controllers to get topological information about the resources in each controller's domain. Each controller may have different interfaces (e.g. wireless, optical), which requires the orchestrator to have a method to support multiple technologies or interfaces.

When a client SDN-Application, such as the Management and Network Orchestration (MANO) layer in NFV deployments, or a legacy Network Management System (NMS), requests a service, the Top-controller is in charge of processing service request, making or delegating end-to-end path computation, and progressing the required control plane information and directives towards the lower layers of the control plane hierarchy. Once the result of the operation is received, Top-controller notifies the result of the operation to the appropriate application that made the request. Top-controller is also responsible of reporting to the application layer connectivity status updates, or any information that the application layer requires for service operation.

Top-controller communicates with the L1-controller through a Java Script Object Notation (JSON) (REST/API) interface.

Finally, as already mentioned, whenever an IATN stands in the border between different L1-domains, the respective IATN LA communicates directly with the Top-controller for the process of interconnecting paths of the different domains.

Layer 1- controller

L1-controller receives directives from the Top-controller, processes them, and forwards the necessary commands to the L0-controller. Additionally, it needs to update the network topology to the Top-controller through its North Bound Interface (NBI), so that there can be a centralised view of the resources. Finally, it is responsibility of the L1-controller to maintain connectivity with IATN LA, as defined in Section 2.2.3.

COP (Control Orchestration Protocol)

The Control Orchestration Protocol (COP) has been proposed as a research-oriented Transport API, technology and vendor independent, that permits to abstract technology specifics of a given transport domain. COP provides a multi-layer hierarchical control plane approach using YANG and RESTconf.

COP provides a solution for the interface between the Top-Controllers and L1 controllers, the interface between L1 and L0 controllers, as well as a detailed A-CPI for exposing resources to SDN applications (i.e., NMS, OSS, MANO) running on top of Top Controller, and it is well suited to integrate heterogeneous radio access networks (5G, mmWave, LTE/LTE-A, Wi-Fi, etc.) with transport optical networks, and the orchestration of transport resources for NFV data centre infrastructure interconnection.

COP is composed of four main base functions:

1. Topology service, which provides topological information about the network, including a common and homogeneous definition of the network topologies included in the Traffic Engineering Databases (TED) of the different control instances;
2. Call service, based on the concept of Call/Connection separation, and providing a common provisioning model which defines an end-to-end connectivity provisioning service.
3. Path computation service, providing an interface to request and return path objects which contain the information about the route between two endpoints;
4. Notification Framework, which provides an asynchronous mechanism for providing event information messages (event reports) for the COP.

The Topology Service provides abstracted topological information about the network. It must include a common and homogeneous definition of the network topologies included in the TED of the different control instances. The COP Topology provides the interface for the exchange of detailed network topology between SDN controllers. An abstract Topology object may consist of a set of nodes and edges, which form a tree structure. A Node must contain a list of ports/endpoints and their associated switching capabilities (e.g., packet/lambda switch capable). An Edge object is defined as the connection link between two Endpoints, which includes some traffic metrics and characteristics. Due to the need of conforming to a common model among different transport network technologies, the definition of the three main objects described (Node, Edge, Endpoint) must be extensible, able to include TE extensions to describe different switching capabilities (i.e., time-slots, packets, wavelengths, frequency slots).

The Call service is defined in the scope of COP as the provisioning interface of each domain SDN controller. The Call is defined as the API's basic data-container and it completely solves the provisioning of an End-to-End connectivity service. A Call object describes the type of service that is requested or served by it – e.g. mmW, Optical, DWDM, Multiprotocol Label Switching (MPLS). It also contains the endpoints among which the service is provided. The Call object also includes the list of the effective connections made into the dataplane, to support the service call. Moreover, a Call also includes a set of traffic parameters that need to be supported, such as Quality of Service (QoS), or allocated bandwidth. Finally, a Call shall include the Match parameter, which refers to the traffic description to which the Call refers. A Connection object is used for a single network domain scope. It should include the path object (i.e., the route) across the network topology the data traverses, which may be fully described or abstract depending on the orchestration/control schemes used. Each connection must be associated with a single control plane entity (e.g. a SDN controller) responsible for the configuration of the datapath. Finally, the Call also introduces the necessary TE parameters (e.g., bandwidth) that the service requests.

Both the Topology and Call services can be extended in order to not only provide abstracted information, but to provide low level technological dependent details, such as transmitted power, modulation formats, optical spectrum grid or CIR/PIR (Committed Information Rate/ Peak Information Rate).

The Path Computation service provides an interface to request and return Path objects, which contain the information about the route between two Endpoints. Path computation is highly related to the previous group of resources. In the service Call, the Connection object has been designed to contain information about the traversed Path. Furthermore, each component in the Path object is represented as an Endpoint with TE information associated to it.

Finally, the Notification Framework provides Notifications referring to the set of autonomous messages that provide information about events, such as alarms, performance monitoring (PM) threshold crossings, object creation/deletion, attribute value change (AVC), or state changes. Notification specifications are generally written around a model of a subscriber and a notification server. The term subscriber is used to name an entity that requests notification subscriptions and receives notification messages. The term notification server is used to designate an entity that recognizes events, turns them into notification messages, and transmits them to pertinent subscribers.

2.2.7 Integration of 5G-XHaul control plane architecture within an existing network

Figure 2-7 shows the relations between 5G-XHaul hierarchical control plane and external elements, such as the SDN Network Orchestrator and the MANO – Orchestration Layer.

SDN Network Orchestrator

For the sake of completeness, an external *SDN Network Orchestrator* has been included, which would be in charge of coordinating 5G-XHaul control plane with other external SDN network domains (e.g. fixed access, IP backbone, or optical transport). This functional separation needs to be understood as a means of taking into consideration the usual heterogeneity existent in operator networks.

It is not precluded that in practical commercial implementations the functionality of the Top controller is extended to be in charge also of SDN control of additional network segments.

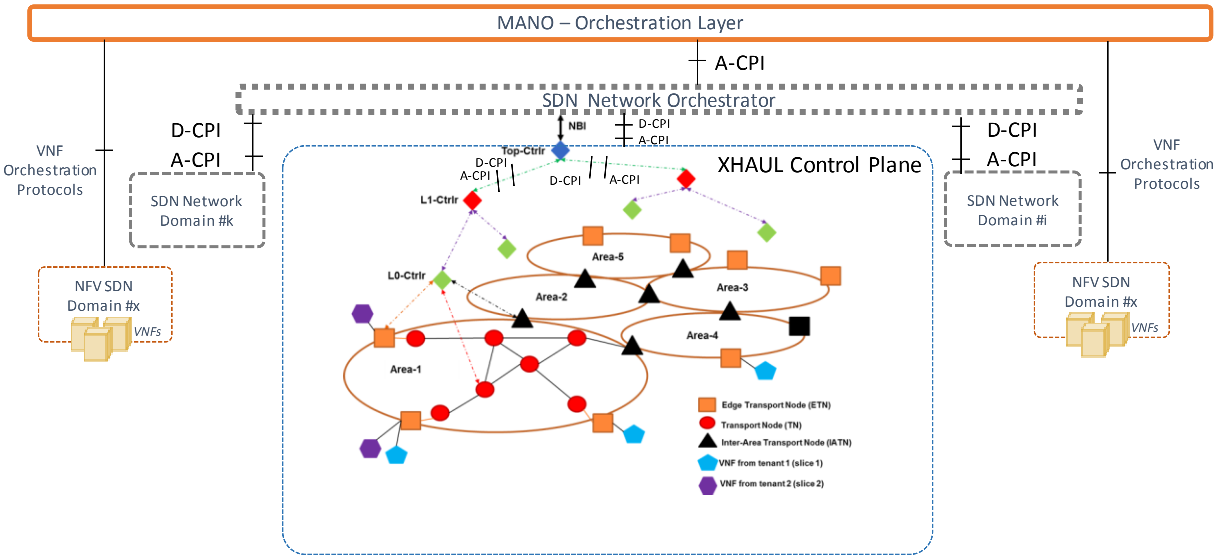


Figure 2-7: Relations between 5G-XHaul control plane and external elements.

MANO Orchestration Layer and NFV SDN Domains

Network virtualization requires a coordinated and consistent resource (compute nodes, storage and network) orchestration in order to automate the deployment and operation of the network services and their components while ensuring the carrier grade performance.

SDN is an essential part of NFV, and Virtualized Infrastructure Managers (VIM) integrate specific SDN controllers to assist with automatic connectivity deployment of: 1) the different components of VNFs and Network Services (NSs) within a Data Centre (DC), 2) the NFV domain to external networks, and 3) the different DCs.

In the network, SDN-based control allows to deploy a control plane that permits to abstract technology specific at each segment and the virtualized or physical nature of their resources for the different network segments and domains, providing southbound interfaces (SBIs) towards the network resources, and NBIs to OSSs and other SDN elements. Network control plane needs to attend request from heterogeneous NFV and legacy applications and services, and thus MANO orchestration layer represents one more client for the control plane (5G-XHaul Top Controller).

NFV and Transport SDN domains are proposed to be orchestrated through a higher control plane layer. While this approach still permits to guarantee the required dynamicity, flexibility and automation end to end connectivity through the transport infrastructure, the highly integrated nature of SDN controllers within VIMs in the NFV realms, makes it preferably that SDN is managed directly by NFV orchestration functions from an implementation perspective.

2.2.8 REST API exposed by Local Agents and L0-Controllers

In addition to the Control Orchestration Protocol, described above, we have implemented JSON/REST interfaces for the LAs of the transport network elements, and for the L0-Controllers. We are in the process of finalizing the relevant documentation using tools such as Swagger and Apiary, and shall release the relevant specifications publicly online.

The APIs of Local Agents provide a simplified abstracted way for L0-controllers (or higher level controllers in the case of IATN) to interact with the underlying datapaths. That is, the LAs hide all the SDN implementation related complexity (e.g. configuration of the flow tables in an ETN).

2.3 Dynamic flow rules

In this Section we introduce the notion of dynamic flow rules, which, in the context of 5G-XHaul, can be seen as a type of Local Agent embedded in the Transport Nodes (TNs). The ability of these sets of rules to remove burden from the L0-controllers and decrease the required configuration messages is well suited to the purpose of the two-level control area architecture described in Section 2.2.

2.3.1 Introduction and Concept of DFR

In SDN networks, after flow rules are statically installed at the SDN switches. If a flow rule needs to be modified, the controller needs to interact with the given switch to reconfigure it. That is the case, for instance, when a controller wants to modify flow rules in order to adapt to a mobility event. Control signaling, however, becomes signaling overhead. In a network with many mobile nodes (e.g. an LTE base station has 3-6 cells, each supporting 200 active mobiles, and a network could have 100,000+ base stations), the control signaling due to mobility events could generate congestion of the control channel and huge processing load at the controller.

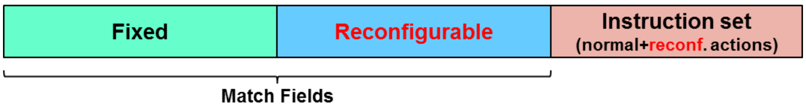


Figure 2-8: Dynamic flow rule.

Dynamic Flow Rules (DFR) is proposed to address the above control plane scalability issues in 5G-XHaul. DFR is a special type of flow rule based on the current OpenFlow (OF) specification [13]. Similar to OF defined flow rules, which we call Static Flow Rules (SFR), DFR consists of two main elements: match fields and instruction set. In contrast to fixed match fields in SFR, DFR includes also a new type of match fields, namely reconfigurable match fields (shown in Figure 2-8). The reconfigurable match fields have some initial value specified by the controller and can be changed by the TN itself by invoking the so-called matched reconfiguration action. The reconfiguration action is one new type of actions included in the instruction set of DFR together with the data plane actions as currently specified in OF. The reconfiguration action acts also on the forwarding tables of the SDN switch, and is able to modify the reconfigurable match fields, or modify/generate an associated flow rule. It is triggered in case that a change in the current value of the reconfigurable match fields is detected. The feature of rule association generates a structure of the flow rules, where the deletion/change of a flow rule can result in the deletion/change of the associated flow rules. With the above three new features, DFR is able to modify itself, and generate/modify associated flow rules dynamically, thus providing an embedded control architecture in the flow rules to distribute the control handlers from the logical centralized controller to the local SDN switches.

In the context of 5G-XHaul, this translates to a type of Local Agent in the transport network elements, which is embedded in the TNs.

2.3.2 Mechanism and use cases

The adaptation capability described above makes DFR especially useful in dynamic situations, where the traffic flow changes frequently and the prediction of the traffic is hard. One such example is the mobility management case.

Mobility management is a major function in a mobile network to ensure proper service delivery to moving subscribers. User Equipment (UE) can move freely from one Attachment Point (AP)/Base Station (BS) to another BS. Pre-installation of flow rules for all potential AP/BS costs valuable TCAM memory, while reconfiguration of flow rules from the controller for each mobility event increases the processing load at the controller and introduces reconfiguration latency, which may affect the service continuity of UE. DFR can be used in this case to automatically reconfigure the existing flow paths of UE after a mobility event, directly at the TN instead of going through the Area controller or even upper layer controllers.

For example in Figure 2-9(a), the controller installs in node S1 a UL DFR #A-B and an associated DL flow rule #B-A. When node S1 detects that the input port for the rule #A-B has changed, the reconfigure action in the DFR rule is executed to update the input port for #A-B from S2 to S3 and the output port of flow rule #B-A from S2 to S3. Afterwards, the update is reported to the controller so that it can reconfigure the network if necessary. The controller keeps the complete view of the flow rules in the network and is able to change the DFR rules when needed.

One other use case for DFR is enabling local bridging at the TN. This can be quite useful for cases like M2M communication. In the scenario depicted in Figure 2-9(b), m2 wants to communicate with m1, which has an established data path to the Packet Data Network (PDN). S1 is the anchor point for both m1 and m2. A DFR can be used at s1 to automatically set up a new flow path between m1 and m2 without the intervention of the controller when traffic arrives.

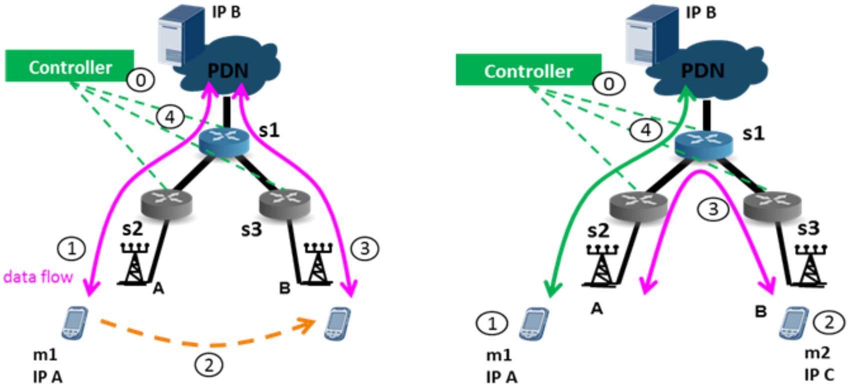


Figure 2-9: Use cases: a) Mobility Management (b) Traffic Bridging.

2.3.3 Implementations

Depending on the capabilities offered by the target TN, there are alternative DFR implementations. Controller could decompose a DFR into a set of flow rule installation messages that are recognized by the target TN. The implementation options include priority based approaches, pipe line based approaches and parallel matching approaches. Different implementation options require different number of rules installed at the SDN switch and result in different matching performance. More details can be found in [13].

2.3.4 Performance Evaluation

We modeled the DFR concept and ran a set of simulations to analyze its performance. Figure 2-10(a) and Figure 2-10(b) compare the queue length at the SDN controller, and the end to end packet latency in the mobility case assuming a tree topology and 10ms flow setup latency. We change the cluster size *c* (number of access points connected to one TN), number of mobile UEs *u* and number of hierarchical levels *l*. Each UE randomly generates flows with the flow length between 1 to 10 packets, and moves to another access point after transmitting a packet. Figure 2-10(a) shows that DFR effectively reduces the controller processing load by ca. 25%. The mean end-to-end packet delay is reduced by 30-40% for a cluster size between 2 and 4, as shown in Figure 2-10(b). Obviously, the signaling from/to the controller to get new configuration due to dynamic change in the network is also greatly reduced.

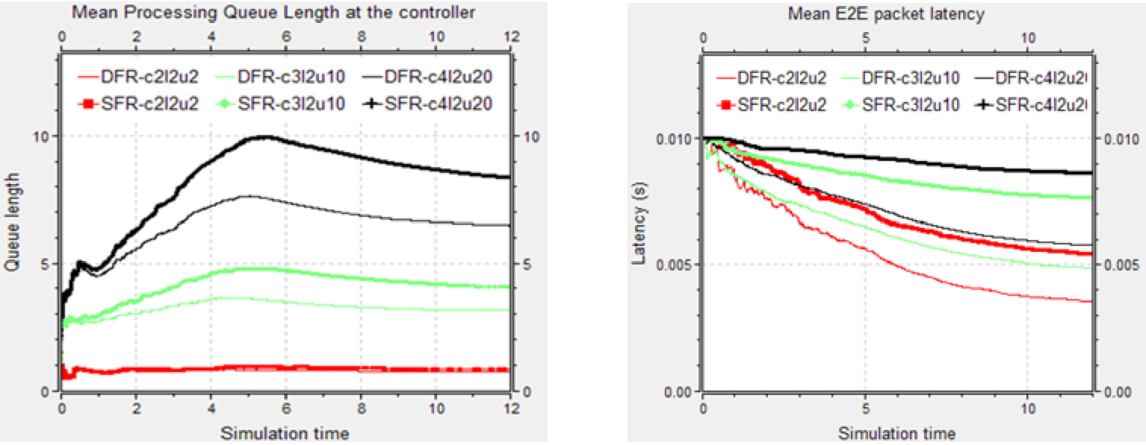


Figure 2-10: a) Controller Queue length b) Mean E2E packet latency.

The response latency of an SDN network to a dynamic event, i.e. the overall flow setup latency, T consists of 4 parts: the transport latency T_{Trans}, the queuing time of the Packet_In at the controller T_{Queue}, the processing latency at the controller $T_{ProcCtrl}$, and the flow installation time at switches T_{Conf}. T_{Trans} is the switch to controller round trip time of the SDN control plane signaling. $T_{ProcCtrl}$ is the time to process a Packet_In and to generate the Packet_Out and Flow_Modification messages. In case several SDN switches need to be configured for the same flow path, Flow_Modification messages will follow a certain order in order to avoid transient inconsistencies in the network [15]. Therefore, T_{Conf} is lower bounded by the configuration/processing time at an individual SDN switch T_{ProcSW} and upper bounded by ρ T_{ProcSW}, where $\rho > 1$ is the number of communication rounds between controller and switch to configure one flow path. Assuming 3 eNBs per access SDN switch, half of the sessions need mobility support, controller to switch distance up to 500 km, T_{ProcSW} =10ms [14], T_{Queue} + $T_{ProcCtrl}$ = ca 2ms, DFR can save the response latency by around 7 ms in the mobility case and 4.5 ms in the local traffic bridging case (shown in Figure 2-11). When configuring multiple switches for one flow path, SFR may double the response latency even with an optimized configuration sequence (i.e., $\rho = 3$). In contrast, DFR is not affected due to its distributed decision, and in turn achieves more than 70% of latency saving. This calculation assumes one hop optical connection between the controller and SDN switch. In case of non-ideal, out-of-band control channel (e.g. multihop, in-band signalling, congestion, etc.), further savings can be expected.

Depending on the different implementation options of DFR at the TN, 1 (Parallel match) or 2 (priority based) flow rules can represent multiple possible configurations at one SDN switch. This in turn greatly reduces the number of needed flow rules compared to preconfigured SFR, which can be estimated by $N_{Fe} = N_{FeU} + ß \times PortD$, where N_{FeU} is the number of upstream flow rules, ß is the aggregation factor, and PortD is the number of logical port downstream.

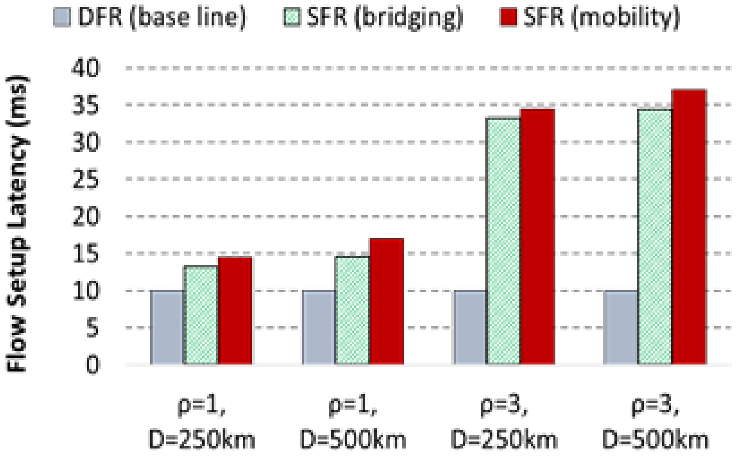


Figure 2-11: Comparison of flow setup latency.

2.4 Resilient Control Plane: Establishment and maintenance of control plane paths

2.4.1 Overview

The 5G-XHaul project advocates the introduction of SDN principles into the backhaul of mobile networks seeking to achieve a flexible backhaul solution to tackle the dynamic allocations and reallocation of numerous flows of different QoS.

Section 6.5.1 of 5G-XHaul deliverable D3.1 formally defines requirements on the control plane, among which R1 (scalability) and R2 (time sensitivity) are first in the list. This underlines the understanding of the project that any suggested control solution should work on different scales and still guarantee the expected timing constraints.

To fulfil those requirements, and, generally, to maximize the flexibility of the infrastructure without spoiling its controllability, we propose the concept of Resilient Control Plane (RCP). The main goal of RCP is to maintain a reliable signalling connection between the network elements and the controller in transport networks, where the control plane needs to be carried in-band.

2.4.2 Provided Services

RCP offers a zero-configuration, self-bootstrapping and self-maintaining communication service supporting different types of control plane communications, between controllers and transport network nodes. This supports resolution of both end-points, a controller to switch path setup and maintenance as well as piggy-backing messages of other protocols on top of the constructed RCP (most prominently of OpenFlow, but also of network management protocols like NETCONF, etc.).

2.4.3 Design Principles

RCP is designed as a multi-dimensional, dynamic interconnection of all network resources belonging to the same *control realm*, driven from and sustained at the resource layer (e.g. a 5G-XHaul control plane area). We deliberately delegate the construction and maintenance of the control interconnection to the *resource control agents* (RCAs) running at every resource element within the control domain, i.e. the local agent in the 5G-XHaul TNs. Acting locally has the advantage of being capable to react to local events in the fastest manner; moreover, with smart algorithms, this provides a good level of isolation from the decision layer: even if decisions are disastrous with respect to the immediate situation of a resource, they can be locally corrected, so that at least the logical binding to the control plane (CP) and, hence, the decision layer is always maintained.

As a simple example, if an L0-controller sends a flow rule to a controlled switch S_1 to block forwarding on a port currently used for control plane communications of some other switch S_2, in today's SDN that would lead to a full disconnect of S_2 from the controller. This is problematic, for instance, in OpenFlow-based networks with an in-band control plane. However, in RCP, the local RCA on S_2 will reconnect to a different suitable RCA (e.g. to the one on S_1). Once reconnected in such an indirect way (RCA on S_1 acts as a proxy for the RCA on S_2), S_2 will notify the responsible controller of the situation. The controller now can raise an alarm or stop the responsible control application or restore the previous state.

The main insight here is that the proposed view is holistic: as the flow rule is necessarily deployed on a controlled resource from the same control realm, both switches have a local RCA, the communications of which cannot be blocked by classical end-user control applications (since these are not forwarded packets, but resource-own control traffic). Using this, we obtain the following property: as long as S_i can block the forwarding of the traffic on S_j, the RCA of S_j will be able to logically reconnect to the controller over the RCA running on S_i. This is how the RCP handles the *self-inflicted failures*, which other approaches can only avoid by restraining the programmability (e.g. prohibiting the changes to some ports, carving out a static, legacy network for the logically out of band control communications, using a physical network that cannot be programmed from the same controller, etc.). A “ban” of an RCA from the control realm has to be implemented by blacklisting the RCA at the controller or other RCAs and not by interfering with the forwarding directly. Note that such bans are of no business for classical control applications interested in data plane forwarding modifications; thus, using RCP, a vendor is able to develop hardened, reliable solutions as part of its switch RCA and controller implementations.

In the examples above, one could argue that the problem is at the decision layer, i.e. in the control applications. However, requiring correctness there would result in a considerable complexity increase and unnecessary customization of network control applications, increasing the effort and impeding their portability. When writing network applications, in addition to the actual application logics, one would need to consider the resource control graph dependencies, the current topology, etc. On one hand, the control graph details are irrelevant to a casual developer and, ideally, should not play any role, so that network applications can be reused in other control realms or in changed infrastructure conditions. On the other hand, this requires instantaneous knowledge of the overall infrastructure situation, which is a very hard requirement for a distributed system. Therefore, requiring correct decisions from a decision layer is an unrealistic assumption and a bad design choice. It would strongly limit the admissible infrastructure dynamics, since non-local decision points cannot be informed in zero time and might make wrong decisions. Further, it would heavily shrink the scalability of the controller by imposing tight synchronization requirements. Most importantly, it would considerably increase the cyclomatic complexity of all network applications, which contradicts the ease of programmability.

2.4.4 Requirements on the infrastructure

There are two requirements on the infrastructure for the RCP to work:

1. Every controlled infrastructure resource, e.g. TN, must execute a resource control agent (RCA).
2. Every RCA must have a local, at least point to point, connectivity to at least one other RCA.

The execution of the control agent is just the representation of the controllability of the resource and can be compared to the presence of OpenFlow clients in OpenFlow switches. As for the connectivity requirement, it means that we do not consider lower layer channel/transport issues here; the channel can be a physical medium (layer 2 link) or a virtual channel, spanning uncontrollable resources (public paths, VPN, etc). This also answers the question, whether everything needs to be controlled.

If the condition 2 is not fulfilled, then the RCP essentially falls apart in two (smaller) RCPs. Interestingly, both remain controllable, as long as there is a controller connected to it.

2.4.5 Resource Elements

RCP is constituted of and spans all available resource elements within a control realm. A resource element of RCP is a network-capable part of a resource within the control realm. Typical resource elements of the RCP are switches and routers, both virtual and physical, yet the RCP also integrates the Controller communication facility, i.e. it handles controller’s communications in much the same ways as it handles switch communications on the control plane. Note that both controlled and controlling resources within the control realm - as defined by the respective owner/operator - become a resource element of the RCP. The separation of control and data is achieved virtually by the mechanisms in RCP. This is what we refer to as “holistic view”, as in sharp contrast to typical SDN work, we explicitly consider from the start the establishment and maintenance of the control plane communications.

2.4.6 Resource Control Agent

The RCA is a piece of dedicated software designed for and running on the resource element as implemented by the vendor or integrator of the resource. As a rough example, RCA can be compared to standard agents (e.g. OpenFlow or SNMP) in the network appliances. In 5G-XHaul the RCA is a type of Local Agent (LA) running in the TNs.

The RCA acts both as a local resource control element and as an RCP peer. For the latter, the RCA implements the R2R protocol suite, enabling it to exchange messages with other RCAs.

The Message Processor of the RCA is responsible for the internal treatment of the R2R protocol and the delivery of the contained messages: the incoming data is being processed and decoded, given to an internal security service for authentication and, potentially, deciphering, the PDUs are then answered directly (protocol exchange, e.g. ack, keep-alive, etc.) or the transported messages are dispatched to either the local resource control (i.e. this RCA is the destination of the message) or to an RCP peer (i.e. this RCA is proxying). The messages can also be passed through to the API, in case the RCA runs on a designated Controller with the control applications.

The RCP Peer establishes and maintains the RCP. It finds other RCAs from the same control realm (called *friends*), chooses *k-1* of them to connect the RCP (called *neighbours*), establishes and maintains internal RCP routing tables to route messages to other RCP destinations as requested.

The local resource control part mainly provides access to the controllable objects of the resource. In the 5G-XHaul context, this corresponds to the operations performed by the Local Agents of the transport network elements and the different controllers up the 5G-XHaul control plane hierarchy.

RCA internally links the local control and the RCA Peer. For instance, it uses local log files, errors and alarms to reconfigure the RCP in case of problems, enabling local and, thus, fastest detection.

RCA needs some initial configuration. Self-bootstrapping allows minimizing configuration to the bare minimum: every RCA only needs to have an ID and a security association (SA). The ID can be derived from the resource (e.g. the MAC address of the management port, etc.) and depends on the available SA: in practice, we expect it to be attached to a private/public key pair. The public key can be directly used as an ID as well. The SA must have a control realm name. This name can be explicitly given as a string, or it could result from the signatory information in the certificate (e.g. X.509) that might be used to confirm the public key to ID binding.

2.4.7 Resilient Control Plane Peer Operations

When RCA starts up, it periodically runs Friend Discovery (FD), Neighbour Selection (NS) and Routing. The output of FD is made available to NS. The output of NS is made available to Routing. Additionally to periodic execution, the RCA runs these modules in case of alarms.

Bootstrapping – Friend Discovery

When a resource element initially comes up, its RCA needs to find other RCAs from the same control realm. This is usually referred to as *bootstrapping*. The only input to this process is the initial configuration and, if available, the friend list from previous operations. RCA should use a mix of different approaches to find friends (UPnP, mDNS, SDP, DNS-SRV, special nodes, etc.) depending on infrastructure capabilities and the nature of the resource element. The overall active friend list is regularly updated and made available to the neighbour selection module.

Structural Adaptation through a Dynamic Neighbor Selection

Given a list of active friends, RCA decides which of these will be considered as control plane routing next hops. Assuming freedom of choice in the infrastructure, that decision is paramount, since the choice of the nodes defines the graph-theoretic structure of the RCP. That structure has a major impact on the resulting per-node state and possible communications in the control plane as well as on the resilience of the whole network. We engineered the RCP to fulfil QoS and resilience requirements in a pragmatic manner, in particular without overloading resources. Table 2-1 presents the structural properties and the resilience capacity of different possible structures. The average node degree represents the number of connections each node needs to maintain and thus indicates the local resource consumption and scalability. The network diameter represents the expected maximum number of control plane nodes between two nodes and is an indicator for the achievable worst case latency. The resilience regarding random faults and targeted attacks represent the capacity of the control plane to reroute traffic when, respectively random or specific nodes fail. The maintenance cost represents the complexity of the RCP modification, e.g. the number of messages needed to add or remove a node, and is an indicator for system scalability.

One realizes that a perfect structure does not exist. Higher degree structures exhibit a better resilience and good QoS posture but induce a high maintenance cost and scale badly. Scale-free networks have very good scaling properties but rely on high degree nodes that are easy targets for an intelligent attacker. Balanced solutions such as uniform degree networks or generic small worlds represent trade-offs: while their resilience is worse than that of random networks or full meshes, their QoS guarantees are lower than what can be achieved with full meshes or scale-free networks.

Since no structure exhibits all desired properties, our RCP is designed as *a structurally adaptive graph*. The main idea here is to conceive a runtime that can change its structure when needed. We refer to this process as *rewiring*: following a trigger, the RCA is able to find other neighbours in order to adopt the appropriate system-level shape, following the service degradation doctrine. Instead of guaranteeing all properties at all times, this idea advocates guaranteeing some properties depending on the context. For instance, in normal operational conditions, good QoS and low node footprint would appear as primary objectives. However, in case of repetitive faults and errors, the system can go into a degraded or *protected* mode, where resilience prevails on quality, and this up to a point where the communication becomes restricted to only essential commands.

Table 2-1: Node state, QoS and resilience properties of different structures.

	Average node degree (per-node state)	Network diameter (latency)	Resilience to random faults	Resilience to targeted attacks	Maintenance costs
Random graph	random uniform	*N* or disconnected	Very high	Very high	Bad
Full mesh	*N*	*1*	Very high	Very high	Very bad
Uniform degree network	*k* such that *k<N* or *k<<N*, e.g. *k~log N*	$f_k(N)$, e.g. *log N*, of *k = log N*	High, depends on *k*	High	Good, depends on *k*
Small world	*log N*	*log N*	Very high	Average	Good
Scale free network	*K* with $p(k) = k^{-a}$, highly skewed, on average *log N*	*log log N*	Very high	Low	Very good

Routing within the Unified Control Plane

Once all RCAs are interconnected, the RCAs can run separate, classic distributed routing protocols on the available neighbours so as to decide, how to reach each destination (e.g. distance vector or link state protocols resulting in the next hop neighbour for any destination). At least two alternatives to this are available, instead of such separate routing:

- Bind the routing to an infrastructural metric instead and only use neighbours, which are closer to the resource element of the RCA in the transport infrastructure.
- Bind the routing directly to the neighbour selection by passing the message from one RCA to another according to some common metric, e.g. distance from the intended recipient to IDs of known neighbours. This would allow constructing a distributed resolution service.

Finally, one could use the resolution service of the RCP and the control possibility to directly use the infrastructure to open channels to the final destination.

2.4.8 Evaluation Results

In this section, we show the evaluation results of different aspects of RCP operations. Every node (or resource) in the network executes an RCA as per explanation above, which follows the proposed R2R protocol.

Evaluation environment

To evaluate the R2R protocol, we implemented an RCA in Python under Linux and run it as an Open vSwitch (OVS) extension. Besides, the same RCA is used on the node running the controller (we use FloodLight, but it is meaningless for our experiments, since they concentrate on the RCP establishment); so, the controller platform is also considered just another resource.

We use a slightly altered version of Mininet, which permits us to run big topologies of such initially unconfigured *N-1* OVS + *1* controller nodes. Depending on the used topology, each node is connected by local links (or unconfigurable existing communication means of some kind) to *k-1* other nodes. No routing and no other special provisions are implemented (no routing protocol, no STP or other L2 auto-organization means; actually, the nodes have no initial IP configuration, no controller configuration, etc.).

On startup, our local RCA installs initial basic OpenFlow flow rules in the OVS to capture all R2R traffic; it also starts sending out R2R packets over its local OVS using PACKET_OUT. Again, note that by the nature of RCA/R2R, there is no controller yet at this stage. An RCA can only follow the procedure in Section 2.4.7 (a) to discover other RCAs in its neighbourhood and to actually construct and maintain the RCP, the routing, (b) to find the controller and (c) to make sure to have flow rules for the control plane traffic as a self-configured, constantly maintained result.

To obtain numeric results, we run networks of different topologies and of different sizes and we use data analysis to get stable results.

Convergence time

The first question we ask ourselves is the question about the RCP convergence. We notably measure the time it takes from the start-up of a completely unconfigured network until each and every OVS can directly communicate to the controller platform, i.e. using matching RCP/OpenFlow flow rules in the OVS. In this experiment, during and after the convergence, the network is not structurally altered (no new nodes are added, no nodes are removed, no new links are added/removed). In other words, the convergence time is the time from the startup and until all RCAs reach a stable state in a given network.

The results are presented in Figure 2-12. For networks (of different topologies) of respectively 25, 50, 75 and 100 nodes, RCAs reach a stable state after about resp. 20, 70, 130 and 150 seconds. After this, the network is fully converged, i.e. all keep-alive messages exchanged after that would be without effect, as long the topology remain unchanged.

We present this result for completeness, as an estimate of an initial bootstrapping procedure. Note however that full convergence is a very conservative estimate: usually, nodes can communicate to the controller much earlier, using another RCA to forward their messages to some other RCA, until these finally reach the controller. In contrast, the convergence time underlines the full convergence, with respect for the structural properties of the underlying network.

Recall that RCP constructed using R2R is not a single-rooted tree (like STP / RSTP) but can actually construct structurally different topologies, always supporting path diversity. Therefore, its scalability for forwarding of the traffic is much better; this however comes at a price of higher yet still realistically usable convergence times.

Note however that the RCP enables communication from the switches to the controller before the convergence is reached. We evaluate this in the next experiment.

Control Path Establishment time

In this experiment, we are interested in the control path establishment time. Note that the RCA network operations are not yet fully converged here, but RCAs are sometimes able to forward the control plane messages to the controller, such that the TCP control connection can be established from the OVS without having flow rules in the switch flow tables.

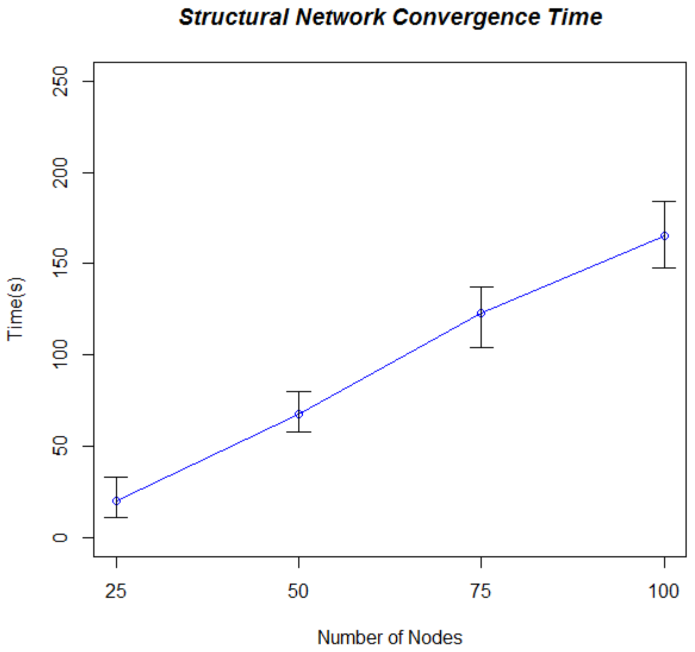


Figure 2-12: Convergence time (in seconds) for networks of different sizes.

In Figure 2-13, we notably present the ratio of the established control plane paths and show its evolution over time for the networks of different sizes: **25** (in blue), **50** (in red) and **100** (in grey) nodes. Besides, we compare the R2R protocol operations (RCA/RCP) to the only existing alternative today, notably to using OVS with Spanning Tree Protocol (STP) in an “in-band control channel” mode.

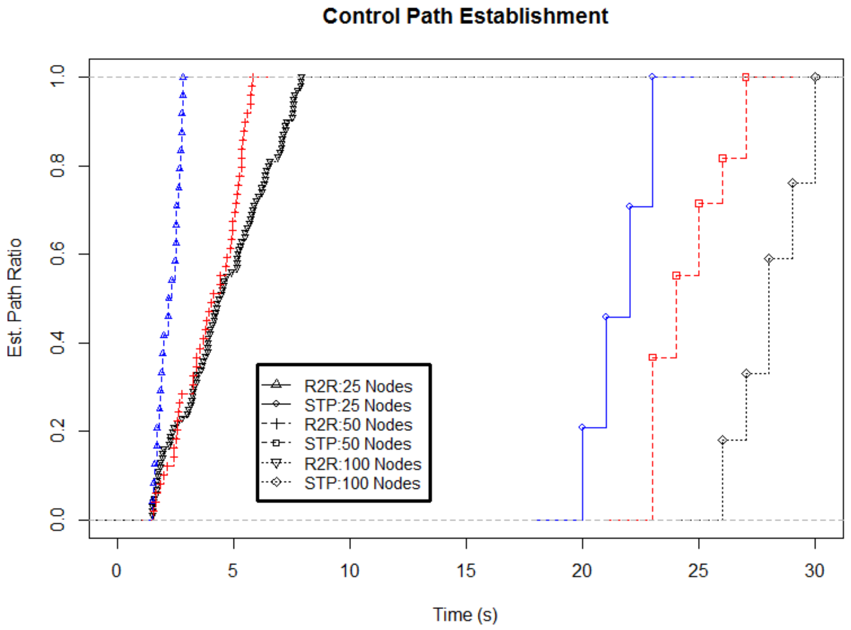


Figure 2-13: Control path ratio over time.

We can see that for every network size and for any ratio, R2R by far outperforms STP. For example, while R2R allows to establish all control paths after resp. 3 s for 25, 6 s for 50 and 8 s for 100 nodes, STP requires respectively 22, 26 and 29 s for the same sizes, which is factor 4-7 times faster.

3 Traffic Engineering mechanisms within the 5G-XHaul control plane areas

In this chapter we describe Traffic Engineering (TE) mechanisms that can be applied at a Level-0 area controller to control how traffic flows within a 5G-XHaul control plane area. Within the framework of the 5G-XHaul architecture, this chapter focuses on mechanisms that are related to the control of Transport Nodes (TNs). As discussed in the previous chapter, TNs are in charge of maintaining connectivity between Edge Transport Nodes (ETNs) through a set of pre-defined transport *tunnels* that connect pairs of ETN/IATNs within an area. Notice however that a particular transport tunnel can be allocated through different transport paths, as long as it maintains the same source and destination ETNs. The mechanisms presented in this chapter deal with the problem of finding the proper path allocation for these tunnels in order to maximize resource efficiency.

Specifically, the mechanisms that are described in this chapter address two main TE challenges:

- ***Challenge 1***: ***Efficient use of resources in control plane areas composed of wireless nodes***. Efforts have been focused on the wireless domain, because resources are much scarcer than in the optical domain. Three distinct wireless TE mechanisms dealing with mmWave and Sub6 technologies are proposed in this chapter.
- ***Challenge 2***: ***Distributed control plane agents for Fast Recovery***. In transport networks, the SDN control-plane is often relayed in-band along the data-plane traffic. In addition, the controller may be physically located far from the network elements. This fact may cause unacceptable delays if link failures need to be dealt with at the controller level. Therefore, 5G-XHaul proposes to offload fast failover functions to the TNs. Two different fast recovery mechanisms are proposed in this chapter.

This chapter is structured as follows. The first subsection describes the TE mechanisms for the wireless domain, and the second one the distributed fast recovery agents. The mechanisms presented in this section are included in various publications such as [13], [16], [17], and [18].

3.1 Challenge 1: Scheduling and path computation in the wireless domain

5G-XHaul considers two types of wireless transport technologies, mmWave based on IEEE 802.11ad radios, and Sub6 GHz, based on IEEE 802.11ac radios. The different nature of these technologies makes it necessary to consider tailored TE solutions for each type of technology. For example, Sub6 radios are much more subject to interference than mmWave radios, which use very directive beam patterns. Therefore, this section is organized in two sub-sections, section 3.1.1 describing mechanisms for TE in mmWave areas, and section 3.1.2 one focusing on Sub-6 areas. Control of hybrid areas comprising both Sub-6 GHz and mmWave networks is left as future work.

3.1.1 Traffic Engineering mechanisms for mmWave areas

As reported in deliverable D2.2 [2] and deliverable D3.1 [3], 5G-XHaul adopts a mmWave solution based on IEEE 802.11ad radios. These radios operate according to a TDMA schedule where a coordinator (PCP) assigns transmission slots to each of the mmWave links operating under its control. In 5G-XHaul, 802.11ad mmWave radios are linked together to form a mesh network, where paths are established by the Level-0 Area controller (AC). Therefore, TE engineering mechanisms for 802.11ad mmWave networks need to contemplate first the definition of paths across the mesh network, and then the allocation of the TDMA slots among the flows traversing a given node.

In this section we present and evaluate two mechanisms that solve the dual problem of path and slot allocation from a different perspective. The first scheme, described in section 3.1.1.1, describes a mechanism that operates at the flow level, assuming that knowledge of the traffic demand can be obtained at the flow level. The second scheme, described in section 3.1.1.2, addresses this problem from a *slicing* perspective, namely providing a differentiated level of service to the cells belonging to each slice through the mmWave mesh. Both schemes have applicability in a practical setting, depending on whether SLAs are defined at the slice level or at the level of individual flows.

3.1.1.1 A flow oriented mechanism for scheduling and path allocation

In our mathematical formulation, the complete network topology is characterized by a bipartite graph G=(V,E) where |V|=N, and |E|=M where V is the set of N TNs including source and destination nodes and the edges E are physical mmWave links between each pair of nodes. We denote $\mathscr{S}$ and $\mathscr{D}$ as the source and destination node set respectively and each flow per source-destination pair k as f_k^w where w denotes the flow number and the pair k is composed of the source node s_k and the destination node d_k.

We define the path selection and time-slot scheduling problem as an optimization formulation in which two main objectives are defined. First, we want to optimize the load balancing, that is, distribute traffic flows optimally across the network to minimize the utilization of the link with the maximum utilization. Second, we aim to minimize the total number of time slots needed. Due to the limitations on the values that the optimization variables can take and the linearity of the constraints, a complex integer programming optimization problem is represented as described in the following.

Path Selection Optimization

The path selection problem finds the optimal path assignment for each demand, following a minimization of the maximum utilized link. To do this we define an optimization goal of minimizing the maximum link utilization in the network. The solution will find the links to be used for the paths between each source and destination pair. We, hence, define the following decision variable, which represents the path-flow assignments.

$$x_{ij}^{kw} = \begin{cases} 1, & \text{if link } (i,j) \text{ transmits flow } w \text{ from pair } k \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The path selection problem formulation is then:

$$min\left(\max_{(i,j)\in E} \left\{ \frac{\sum_{k=1}^{||\mathscr{K}||} \sum_{w=1}^{W_k} f_k^w (x_{ij}^{kw} + x_{ji}^{kw})}{c_{ij}} \right\} \right) \quad (2)$$

s.t.

$$\sum_{k=1}^{||\mathscr{K}||} \sum_{w=1}^{W_k} f_k^w (x_{ij}^{kw} + x_{ji}^{kw}) \le c_{ij}, \forall (i,j) \in E \quad (3)$$

$$\sum_{j:N(i)} f_k^w x_{ij}^{kw} - \sum_{j:N(i)} f_k^w x_{ji}^{kw} = \begin{cases} f_k^w, i = s_k \in \mathscr{S} \\ -f_k^w, i = d_k \in \mathscr{D}, \\ 0, otherwise \end{cases} \quad (4)$$

$$\forall i \in V, \forall k \in \mathscr{K}, \forall w \in W_k$$

$$\sum_{j\in N(i)} x_{ij}^{kw} \le 1, \forall k \in \mathscr{K}, \forall i \in V, \forall w \in W_k \quad (5)$$

$$\sum_{(i,j)\in E} x_{ij}^{kw} \le MH_k^w, \forall k \in \mathscr{K}, \forall w \in W_k \quad (6)$$

$$x_{ij}^{kw} \in [0,1] \; \forall (i,j) \in E, \; \forall w \in W_k, \; \forall k \in \mathscr{K}$$

The first constraint (3) determines that for each link in the network, the summation of the flow loads that will be carried by the links represented by variable x_{ij} and x_{ji} between two ETNs must never be greater than the link capacity, c_{ij}=c_{ji}. Constraint (4) ensures that the minimum flow requirements are met and the Constraint (5) is defined in order to avoid having multiple paths per flow. Finally, Constraint (6) restricts the maximum number of hops that each flow will go through. This maximum number of hops MH_k^w are defined by each flow's Traffic Specification (TSPEC) requirements.

Time Slot Scheduling

The main optimization goal that will be developed in this section is aimed to minimize the number of time slots needed in order to send and receive all the traffic demand. Then, the solution that this optimization problem will give is the optimal assignment of resources available on each node, depending on outgoing and incoming traffic, which is defined by the variable u_{ij}^t as

$$u_{ij}^t = \begin{cases} 1, \text{ if link (i,j) is scheduled at time slot t} \\ 0, \text{ otherwise} \end{cases} \quad (7)$$

The path selection optimization solution defines a set of ordered links for each flow as a vector representing the hops traversed, which is represented as

$$l_k^w = [l_k^w(1), l_k^w(2), \ldots, l_k^w(H_k^w)] = [(s_k, j), (j, p) \ldots., (r, d_k)] \quad (8)$$

where each *h*-th element of the vector is a link that is part of the path between source-destination pair *k* for flow *w*, i.e., each hop is characterized by a pair *(i,j)*.

The optimization problem for the time slot scheduling problem is then given as follows.

$$min \sum_{i \in V} \sum_{t=1}^{T} \sum_{j:N(i)} \left(u_{ij}^t - \frac{u_{ij}^t u_{ji}^t}{2} \right) \quad (9)$$

s.t.

$$\sum_{k=1}^{||\mathscr{K}||} \sum_{w=1}^{||W_k||} f_k^w (x_{ij}^k + x_{ji}^k) \leq \sum_{t=1}^{T} \frac{c_{ij}}{T} (u_{ij}^t + u_{ji}^t), \forall (i,j) \in E \quad (10)$$

$$||H_k^w||((t_s + d_{tx} + t_{overh})) + \sum_{h=1}^{H_k^w} \left(\max_{t \in [0,\ldots,T]} (t u_{l_k^w(h+1)}^t) - \min_{t \in [0,\ldots,T]} (t u_{l_k^w(h)}^t) \right) d_t$$
$$\leq \beta_{f_k^w}, \forall k \in \mathscr{K}, \forall w \in W_k \quad (11)$$

$$\sum_{j \in N(i)} u_{ij}^t + u_{ji}^t \leq 1, \forall i \in V, \forall t \in T, \quad (12)$$

$$SINR_{ij} = \frac{\frac{P_{i,j}}{L_{ij}}}{No + \sum_{\substack{k \neq i \\ k \in N(j)}} P_{kj} u_{ij}^t} \geq \gamma_{ij}, \forall i,j \in E, \forall t \in T \quad (13)$$

$$u_{ij}^t \in [0,1] \; \forall (i,j) \in E, \; \forall T$$

The objective function (9) aims to minimize the total number of slots used in the network. The second term of the objective function is used to avoid counting one time slot twice if in the solution both uplink and downlink links are scheduled in the same time-slot. In the objective function *N(i)* is defined by the immediate neighbors of node *i* with whom node *i* can communicate either to send or receive traffic. Although the task of network discovery defines the set of possible neighbors for each node, for the sake of simplicity on our formulation it is assumed that each node *i* already knows its set of neighbors.

Constraint (10) ensures that the throughput requirements for each flow are satisfied per-link and that for each time slot, the scheduled flows cannot be greater than the maximum capacity of the link for one slot. In this constraint, we allow multiple flows to be scheduled on the same link providing that the per-slot capacity is not surpassed.

The per-flow timing constraint (11) is represented in the second constraint. Here H_k represents the number of hops that each flow from source and destination pair *k* will go through. Each element *h* represents an arc *(i,j)* of the path defined in the path selection problem. The delay introduced by the slot assignment is calculated by taking the difference between the time in which the flow is received and the time in which it is sent in the next hop. That way we can define how much time it takes to process each traffic on each node. The other terms include the switching delay, air transmission delay (d_{tx}) and overhead time from upper and MAC layers. The total overhead can be approximated as

$$t_{overh} = 2 * t_{SIFS} + t_{guard} + t_{PHY}, \qquad (14)$$

where we assume that the time needed to wait for a block ACK from receiver station is negligible. SIFS values as well as guard time and PHY header time for each frame sent by each station on each Service Period (SP) are given by the IEEE 802.11ad standard. The beam-forming delay and sector level sweeping are negligible compared to the other delay components, and hence, are not included in the formulation.

In (11), the constant d_t represents the air time for a time slot. The total delay cannot surpass the maximum latency threshold β_{fk} defined per flow type.

Constraint (12) exhibits the half-duplex limitations per node. For the sake of simplicity, on any given time slot *t*, each node is only able to receive or send data through one of its links. However due to the use of directive antennas in mmWave, in a TN two or more radio interfaces can send or receive traffic to/from other neighbor TNs.

Finally the SINR constraint is presented in (13), which defines that the time slot *t* can only be allocated to link *(i,j)* if the SINR value is higher than a predefined threshold. The possible interferer links of pair (i,j) are predefined given the network topology and state. The path loss calculation is done based on the alpha-beta-gamma (ABG) path loss model.

Joint Optimization

In the previous subsections, we described two sub-problems in which we solve path selection first and then find a solution for the optimal time-slot allocation. This approach finds an optimal solution for the path selection formulation and with these results, it allocates time-slots to each one of the links. An alternative approach is to define a joint optimization in which both objective functions are solved at the same time and the problem is treated as a multi-objective optimisation.

For this, we combine both objective functions into a single scalar objective function that includes both path selection and time slot allocation. We defined weights for each single-objective function.

The joint problem is then defined as

$$\begin{aligned} min\ \gamma_1 \sum_{i\in V}\sum_{t=1}^{T}\sum_{j:N(i)} \left(u_{ij}^t - \frac{u_{ij}^t u_{ji}^t}{2} \right) \\ +\gamma_2 \left(\max_{\forall(i,j)\in E} \left\{ \frac{\sum_{k=1}^{\|\mathscr{K}\|}\sum_{w=1}^{\|W_k\|} f_k^w (x_{ij}^{kw} + x_{ji}^{kw})}{c_{ij}} \right\} \right) \qquad (16) \end{aligned}$$

where

$$\gamma_1 + \gamma_2 = 1$$

For the joint optimisation formulation, the constraints (3)-(13) from the previous formulation are still valid and represent the set of constraints for the multi-objective optimization. The values of the weights on each objective function in the single scalar multi-objective function determine the Pareto Optimal Solutions from the multi-objective optimisation problem.

Shortest Path Heuristic Algorithm (SHPH)

A heuristic algorithm based on shortest path for the path selection and link scheduling problem is defined as follows. As the first step, the algorithm prioritizes each flow according to their latency and throughput requirements, this means that highly restrictive flows are going to be scheduled first in order to meet their requirements. For each one of these flows, a minimum hop count path is chosen (for which an optimization solver is required) and for each assigned link the remaining link capacity and a cost is determined, where the cost is the number of slots used at the link. This process is repeated until all high priority traffic has links allocated. For the remaining flows a least cost path to destination is calculated and the same process of link capacity and link cost calculation is carried out. After all traffic flows have scheduled links, minimization of used slots for every link is applied by allowing MAC aggregation whenever possible.

Evaluation Setup

In this section, we describe the performance evaluation procedure for the three approaches presented in this paper, namely Two-Stage Optimisation (TSO), where the two optimization problems are solved sequentially, the Joint Optimisation (JO), where the objective function is evaluated in the joint optimization problem presented, and Shortest Path Heuristic Algorithm (SHPH). The evaluated scenario is shown in Figure 3-1.

Following the commercial mmWave modules such as the ones from BWT, each TN is defined to be composed of 4 STAs (radio interfaces), whereas each STA is composed of a 90° steerable antenna element. The scenario defines 16 Transport Nodes (TNs), 2 source nodes and 3 destination nodes. Source nodes represent small cells, usually located on lampposts and the destination nodes represented by Macro cells, which in turn are connected e.g. through fiber to other sections of the transport network. TNs do not generate any flows but instead they receive and forward flows through any of their 4 STAs. The topology is inspired by a street layout, where the transport nodes and the small cells are deployed on the street level, generating a grid-like connectivity graph. Whereas, the macro cells have a wider coverage and higher degree of connectivity than the TNs and small cells. Following the terminology used in this project, the source nodes can correspond to Edge Transport Nodes (ETNs) and the destination nodes as Inter Area Transport Nodes (IATNs), where the technology domain changes (from wireless to e.g., WDM-PON).

For 5G traffic, the first type of traffic flows are pure backhaul traffic (BH) with peak data rates per source node of 10Gbps with latency requirement of 20 ms, which is defined by split C in D2.1 [1], and the other type of traffic flows are based on HARQ relaxed fronthaul traffic (FH) with average data rates per source node of 50 Gbps, which accounts for split B in D2.1 [1]. Each source node is defined to produce a total of 12 flows of both FH and BH traffic, which are evenly distributed to each one of the three destination nodes. Hence, the base load of each flow is assumed to be the aggregated total traffic values divided by 12 flows.

Parameter	Value
t_{SIFS}	$3\mu s$
d_{tx}	$3\mu s$
t_s	$10\mu s$
t_{guard}	$3\mu s$
t_{PHY}	$4.79\mu s$
d_t	$50\mu s$
PHY Layer	OFDM
Link Capacity 4G	$4.69Gbps$
MAC aggregation	A-MSDU
Distance between TNs (d_n)	$100m$
Frequency Band	$60GHz$
EIRP	$40dBm$
Antenna Gain	$8.5dBi$
Receiver Sensitivity (γ_{ij})	$-53dBm$
N_o	$-100dB$
θ	$24.4dB$
Γ	1.9
X_{ρ}^{ABG}	$4dB$
α	2.6

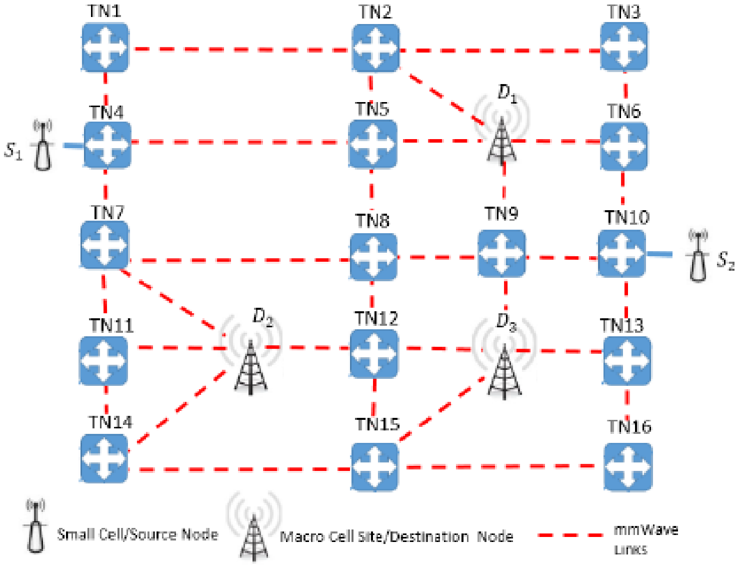


Figure 3-1: Parameters and topology of the evaluation setup.

Evaluation Results

In Figure 3-2 we present the evaluation of different load values by scaling the actual load of each flow in the range of 10%-100%. The values of the weights for the JO approach are 0.5 for both γ1 and γ2. As seen in the figures, the performance results obtained by the Joint Optimization approach shows that it provides a better overall solution. Given that the objective function for the first problem in the TSO approach is load balancing, it provides a better solution in terms of maximum link utilization metric, albeit a higher number of time slots required by it.

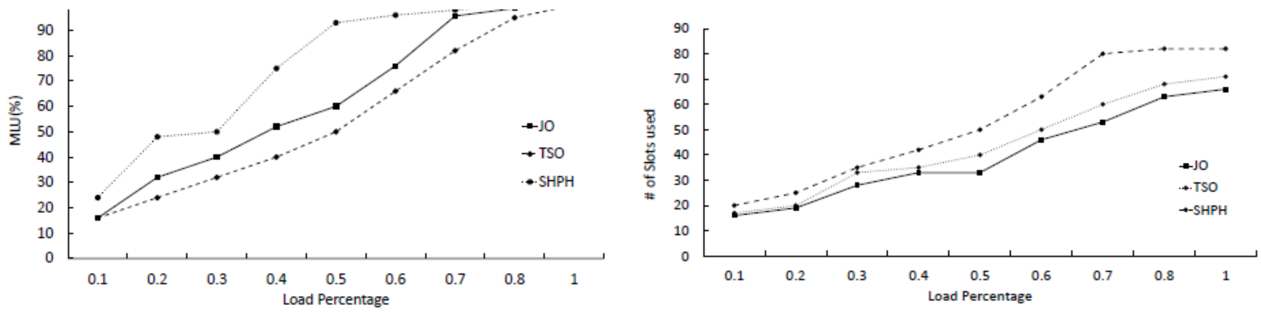


Figure 3-2: (%) Max Link Utilization and (%) of Slots used for JO, TSO and SHPH approaches.

Subsequently, for the resource utilization, the Joint Optimization attains a better performance overall compared to the alternative approaches, which shows the advantage of defining a multi-objective optimization. Thus, we can see that by finding first the optimal path selection for each flow on a load balancing scheme TSO approach does not fully exploit MAC aggregation, which has direct effect on resource utilization. When all flows have paths scheduled on the path selection, MAC aggregation acts on these predefined flows. This two-step optimization limits the search for optimal solutions to both sub-problems, which in turn reduces system performance overall. The heuristic SHPH method, on the other hand, yields the worst network performance results among the three methods, albeit achieving relatively comparable results to the second best method for each performance metric.

Time to solve or computation time on a generic PC with 4 core CPU is also evaluated. This represents the time that the solver needs to find an optimal solution for all demands. The time required to find a solution is the lowest for SHPH since it searches for much less feasible solutions given that allocation is done on a per flow priority basis. Its computation times are found to be around tens of milliseconds. On the contrary, since Joint Optimization does an exhaustive search on the different solutions taking into account both objective functions, its computation times are higher. Numerically, for the JO, the optimal result is found within 2s-6s, whereas for TSO the optimal result is found within hundreds of milliseconds. Nevertheless, the targeted problem can be applied as an offline or online (real-time) problem based on the network design. If a pre-provisioning of the forwarding decisions based on the traffic classes and the source-destination nodes is required, the achieved computational times for JO would be acceptable. Note that, through parallelization and/or with computationally much powerful servers, the computation times can be reduced significantly. Since such optimization method will be part of the control logic to be run at an SDN controller, a more powerful computational resource, and hence, much lower computation times are expected.

3.1.1.2 Slice aware mechanisms for scheduling and path allocation

In this section we describe a second TE mechanism for mmWave based control plane areas, which operates at the level of transport slices. Network slicing allows the support of logical autonomous networks on top of a common infrastructure offering a customized networking experience addressing distinct business demands. By enabling network slicing, 5G networks can assure the desired performance, but the limitation of radio spectrum brings new challenges in resource allocation and scheduling, especially for achieving the desired flexibility in resource sharing. In this context, this section analyzes the joint path selection and backhaul link scheduling problem in a transport network, assuming mmWave multi-hop backhaul (BH). For supporting efficiently multiple slices with diverse Key Performance Indicators (KPIs), a slice-tailored resource allocation, scheduling, and selection of BH links and redundant paths is proposed. Such resource allocation process is complemented with adaptive routing and other flexible related operations with particular focus on delay critical and throughput oriented slices.

In this study, we emphasize on two slice types, enhanced Mobile Broadband (eMBB) and Ultra Reliable Low Latency Communications (URLLC). URLLC slices require latency below 1ms and 99.999% reliability; on the other hand eMBB requires ultra high user throughput. In an Ultra Dense Network (UDN) with mm-Wave BH, the routing and scheduling policies should be adapted to meet the per slice requirements. This involves the resource allocation of BH links, the number of hops and the operation of transport nodes. Hence, the objective of this study is: Firstly, to dynamically identify BH links and paths to be scheduled per a given time window, taking into account the target global objective for the particular slice (in terms of maximizing BH capacity or minimizing latency or optimizing aggregate utility). Secondly, assuming different slices, to identify whether and how the incoming flows are stored in the queues and forwarded to the next hops (or destinations), based on link selections in the previous step and the per slice QoS requirements (delay, outage or data rate). To this end, context awareness is discussed as key enabler to minimize the signalling required for the dynamic flow scheduling.

System Model and Problem Description

In the system as illustrated in Figure 3-3, a cluster of small cells can be densely deployed to enhance capacity. The small cell can either relay traffic towards other small cells via mmWave BH or provide access to the end users. For the 5G-XHaul architecture, the small cell features two components, an ETN (function that connects to the 5G-XHaul network, e.g. mapping S1 or X2 interfaces to different transport tunnels), and a TN (with a forwarding entry for each tunnel towards the next TN). On top of the small cells we have one or more TNs which act as traffic aggregators. Our implementation assumption is that each traffic aggregator is co-located with Level-0 Area Controller which performs the scheduling functionality for this cluster of nodes. On top of the Level-0 ACs we have the Level-1 AC which orchestrates the functionalities and provides the slice requirements to the Level-0 ACs.

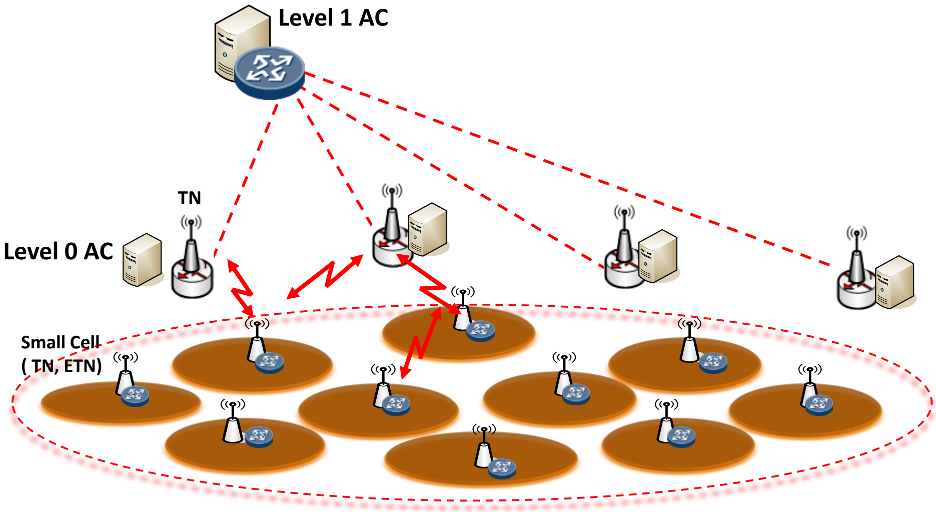


Figure 3-3: Physical deployment for two-level scheduling.

In this context, logically centralizing the control intelligence of the network in a local coordinator (e.g. Level-0 AC) will allow for fast scheduling and coordination to meet the tight KPIs. However, some challenges need to be considered especially when applied to highly dynamic environment, e.g. mmWave based BH/ access network [19]. This study extends [20], where the joint routing and scheduling problem was formulated as a capacitated vehicle routing problem, where a central depot allocates passengers to vehicles and assigns paths that reach their destinations with the minimum time [21]. In this study, we have one or more transport nodes responsible for a cluster of small cells, acting as depots, and we perform path selection from each of them to the same set of TNs using a form of BH link diversity to meet the per slice KPIs. The physical interpretation of a path resembles a tunnel, enclosing a group of nodes through which the traffic is forwarded by a variable number of hops. Each path can encapsulate multiple flows with different destinations and characteristics.

Let *G(V,E)* be the graph consisting of a set of *V* nodes (TN and underlying ETNs) and a set of *E* edges. An edge e ($e \in E$) e∈E is a connection between two nodes v1, v2 (v1,v2∈V). The presence of an edge *e* indicates that data can be exchanged between *v1* and *v2*. We assume *m* links in the network, where the links are considered un-directional. Here, we also introduce a set of ***S*** slices where slice ***s*** specific traffic originates from one or more sources (TNs) and terminates to one or more destination ETN(s) with the desired rate R_s and maximum delay, resiliency, etc. T_s (as defined by slice related KPIs and conveyed by the Level-1 AC to TNs). We define the potential demand of slice s over the link *e* as $l(e,s)$, to identify rate requirement of the slice that will be contributing to the load of that link over a time window, if this link is used to carry traffic of that slice. Each link *e* has a maximum capacity, which corresponds to the maximum rate that can be achieved over the link for each time instance, defined as $c_e, \forall e \in E$, and a desired data rate which corresponds to the summation of all the slice traffic traversing it, denoted as $\sum_{s \in S} l(e,s)$. Here, high directional antennas can be used to compensate for the high path attenuation. In this case, interference by other links is assumed to be negligible due to the high directivity of the antennas and the half duplex constraint (nodes only transmit or receive per time-instance). Using the definitions of the link capacity and the link load, a new parameter which captures the number of time-slots required for a link to satisfy its demand is shown:

$$f_e = \left\lceil \frac{\sum_{s\in S} l(e,s)}{c_e} \right\rceil \tag{3.1}$$

where $\lceil\cdot\rceil$ accounts for the ceiling function, i.e. the least integer that is greater than or equal to *x*. In other words, f_e which is interpreted as a cost function for each link, captures how many timeslots are required for the traffic to be carried on each link, so as to meet the data rate requirements. The worse the channel conditions, the higher the number of timeslots needed to reserve at one TN. Another key parameter, which is going to be used for the scheduling part formulation, is the set of all bi-partite sub-graphs of the graph *G(V,E)*, denoted as ***G'***. Each of these G'_i sub-graphs represents a combination of link activations (one set of the bi-partite graph is the transmitter nodes and the other set is the receiver nodes). Each of these sub-graphs is associated with a weighting factor $w_{G'i}$ which represents the fraction of time that this combination of active links endures.

The objective is how to select the best links and paths to be used from each source TN to each ETN in order to satisfy its data rate demand and delay constraints. The maximization of total BH throughput is equivalent to the minimization of the total number of timeslots, defining the ratio of the demand over the BH link capacity towards an ETN. In other words, the objective is to find paths the traffic should follow and links to be activated per slice and TN so as to maximize the system performance. The maximization of total BH throughput is equivalent to the minimization of the total number of timeslots, defining the ratio of the demand over the BH link capacity towards an ETN, which is equivalent to finding a path per traffic flow that minimizes the total cost under certain constraints.

$$\min \sum_{s\in S}\sum_{n\in N}\sum_{e\in E} f_e(s,n)\, x_e(s,n) \tag{3.2}$$

Subject to:

$$\sum_{e=\{n,j\}\in E} x_e(s,n) = k(s), \forall s \in S \tag{3.3}$$

$$\sum_{s\in S}\sum_{e=\{i,j\}\in E} x_e(s,n) \leq 1, \forall i \in V, \forall n \in N \tag{3.4}$$

$$\sum_{s\in S}\sum_{e=\{j,i\}\in E} x_e(s,n) = 1, \forall i \in V, \forall n \in N \tag{3.5}$$

$$\sum_{e=\{i,j\}\in E} f_e(s,n)\, x_e(s,n) \leq R_s\,, \forall s \in S, \forall n \in N \tag{3.6}$$

$$\sum_{n\in N} f_e(s,n) \leq T_s \sum_{i=1}^{|B|} w_{B_i} 1_{e,B_i}, \forall e \in E\,, \forall s \in S, \forall n \in N \tag{3.7}$$

$$\sum_{i=1}^{|B|} w_{B_i} = 1, w_{B_i} \geq 0, \forall B_i \in B \subseteq G' \tag{3.8}$$

The first 4 constraints (3.3) – (3.6) are the routing constraints, whereas constraints (3.7), (3.8) are the scheduling constraints. In the constraints, the variable $x_e(s,n)$ indicates if link *e* is used by slice *s* in node *N*. In (3.3), the number of links between each TN (denoted as node n) and all the TNs depend on the number of paths and is equal to the variable *k* (which relies on the slice requirements and can be pre-defined). The higher we set this value, the fewer hops are expected in total. In (3.4) and (3.5), the number of incoming links and outgoing links to/from each TN is set exactly or less than one. By this, all the TN must be able to receive traffic; however, it is optional to have outgoing traffic to other links. Constraint (3.6) is the maximum delay constraint that has to be considered when creating a path. This constraint might be variable depending on the traffic (i.e. lower threshold for URLLC slices and higher one for eMBB slices). Moreover, (3.7) shows that the cost of the link shall not exceed the pre-defined slice-based time window (T_s); finally (3.8) implies that the summation of the weights (i.e. the fraction of time each sub-graph is active) is set to one.

Two-level Approach

Our proposed framework decouples the initial problem in two sub-problems. At the first stage, we target to solve the path selection problem which has the form of an Integer Programming problem. By this, we identify links to activate and the number of slots to dedicate per set of links, such that the BH throughput is optimized. The next stage is the selection of the packet to be forwarded from the queues in a way to minimize the delay, taking into account the half duplex constraint, the multi-hop requirements and the queue buffers.

Path Selection Algorithm

The objective of this problem is to deliver to a set of ETNs with known traffic demands on minimum cost paths originating from each of the TNs. Based on the slice requirements, we may have different maximum allowable number of hops to meet certain latency KPIs (this can be defined by *k* factor). The algorithm follows a branch-and-bound scheme, where lower bounds are computed by solving a linear program (LP) relaxation of the problem. This relaxation is iteratively tightened by adding valid inequalities to the formulation according to the cutting plane approach. The exact method is known as a branch-and-cut algorithm and is thoroughly described in [22] for the case of the IP problem.

Scheduling Algorithms

After obtaining the paths and the number of timeslots that each link is going to be used for all destinations, the next phase is to find how to forward the packets from each source TN to all the ETNs, having a variable number of hops per path with the minimum delay and according to slice requirements. Here, depending on criticality of traffic, we may have to decide whether to use Queue-and Forward at the intermediate nodes or just forward traffic with higher priority. By using the latter, we will manage to meet the low latency KPI which can be critical for some slices (e.g. URLLC slice).

A. Scheduling part for Critical / URLLC Slices:

In the case of URLLC slices, TN scheduling should be tailored to achieve low latency, but at the same time assuring high reliability (since we might prefer non-ideal BH links for the forwarding of the traffic using minimum number of hops). This can be performed by the following policies:

- Forward traffic with minimum allowable number of hops, without queuing at each intermediate TN (TNs do not need to process the incoming slice traffic) to minimize latency.
- Perform joint transmission/reception from the source TN (traffic aggregator) to the destination ETNs (via redundant paths, assigned to this slice) at the same frequency band to increase reliability. Here the assumption is that the BH between source TNs is ideal (e.g. using fibers) to allow data exchange for joint transmission / reception.
- For traffic of the same slice, towards a different destination ETN, use distinct carrier frequencies or redundant paths. Local coordination between macro-cells will be required to avoid cross link/path interference.

B. Scheduling part for Non-Critical Slices (e.g. eMBB):

For the rest traffic, the packets are stored at each TN in separate queues per destination ETN. Based on different isolation we want to achieve among slices (full, partial, no isolation), we may have different carriers associated to different slices. So, we may have one queue per slice and destination. The target is to empty all the queues by the end of the given time window. Here, one constraint is related to half-duplexing, i.e. each node can either transmit or receive per time instance. Furthermore, the traffic that is forwarded via more than one hop should be stored in separate queues in the intermediate nodes. In each queue, First In, First Out (FIFO) policy is applied. For the solution of this problem, we showcase a throughput optimal algorithm which follows the back-pressure concept [23].

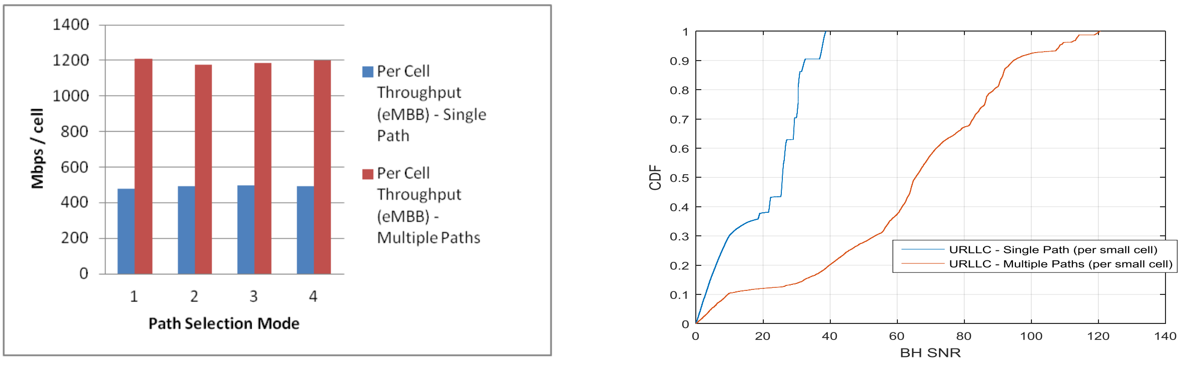
Evaluation

To evaluate our work, a 9 small cell deployment is considered as an exemplary TN cluster under four source ETNs which are placed around the cluster (as can be seen in Fig. 3-1). Monte Carlo simulations are performed in a wide area scenario, where four TNs are placed around the cluster. In particular, Table 3-1 provides a summary of the simulation parameters.

Table 3-1: Simulation Parameters.

Small Cells	9
ISD	20m
Users	Poisson arrivals per cell (λ=2.5)
Traffic	eMBB Slice: Random traffic demand per user (10-50Mbits) URLLC Slice: 1Mbits
Carrier	60GHz
Bandwidth	4 carriers of 100MHz (1 carrier to be used per TN)
Max. number of beams	4 per node (half duplex)
TTI size	0.1ms
BH channel model	For the capacity computation, the 60GHz path-loss models in LoS and nLoS are used by [24]
Processing Delay	L1 Processing ~0ms L2 Processing ~0.2ms
Snapshots	5000

The metrics used for the evaluations are: (i) the average BH link throughput (for eMBB) in case we have BH diversity by multiple paths (using different carriers), (ii) the Cumulative Distribution Function (CDF) of BH Signal-to-Noise-Ratio (SNR) (for URLLC case) with and without path diversity and (iii) the average delay from each of the source TNs to reach each destination ETN for both slice types. In the evaluation, users of the same slice were randomly dropped assuming to have similar requirements; thus not requiring differentiation on the routing and scheduling. The implementation of this scheme consists of two stages. The first stage is the extraction of results for the path selection problem for each of the *n* aggregation TNs. Here, we adjust the number of paths (based on the variable *k*), so as to find the optimal path selection in different cases. The higher the k, the more the short-hop paths that are extracted. For URLLC slices, k is set high (*k=7*) since there is the constraint to have minimum number of hops (from each TN to destination ETN). For eMBB, multiple sizes of *k* are considered for the extraction of paths; hence we tested scenarios with different path selection modes (path selection modes 1 to 4, are translated to *k*'s from 2 to 5). For the scheduling part, URLLC traffic is forwarded without queuing at the intermediate TNs and with high priority. This will slightly affect the delay of other types of traffic. For the rest traffic, the backpressure scheduling is used.



a)

b)

Figure 3-4: BH Throughput (left) and CDF of BH SINR (right) comparison.

Figure 3-4a, shows the comparison in BH throughput for the case we have single source TN (single depot in VRP problem as in [21]), versus the case that up to 4 source TNs (multiple depots) can provide extra capacity by utilizing more links for the ETNs. We perform Monte Carlo simulations per slice independently (eMBB and URLLC) and at each snapshot we try to find paths from all source TNs towards each ETN according to the slice demands (based on k values). The actual number of paths towards an ETN, depends on whether there are available links with acceptable channel quality (e.g. we might have NLoS) from each of the 4 source TNs towards the destination ETN. We can observe huge gain in terms of cell throughput for eMBB in Figure 3-4a when we have a form of BH diversity by multiple source TNs to the ETNs, instead of path originating from single source TN per ETN. Moreover, Figure 3-4b, shows the CDF of BH link SNR for URLLC (for which reliability is KPI) which can significantly improve when enabling multi-connectivity via redundant paths. BH throughput metric is not evaluated for this slice type, since the rate requirement is low.

3.1.2 Traffic Engineering mechanisms for Sub-6 areas

Sub-6 GHz wireless networks complement mmWave networks in the sense that offer lower capacity transport links, but these links are capable of operating on NLoS conditions. However, NLoS operation also translates into having radios that are more prone to interference. Therefore, the TE mechanisms introduced in the previous sections are not well suited for Sub-6 networks. In this Section we describe and evaluate a mechanism that is specifically tailored to minimize interference across a 5G-XHaul control plane area comprising Sub6 TNs. In particular, we propose a centralized mechanism to compute for each transport tunnel a main, or default path, and a backup path. Following the 5G-XHaul architecture, introduced in Section 2, transport tunnels are proactively installed by the controller in the TNs before traffic starts flowing.

The path allocation scheme described in this section, assumes that the Sub-6 TNs adopt the SDN architecture described in deliverable D4.11 [7], and which we briefly summarize next. In addition, for the sake of space this section contains a reduced description of the developed mechanism. The interested reader is referred to [17] for an in-depth explanation.

SDN architecture and forwarding model adopted in Sub6 TNs

In deliverable D4.11 [7] an SDN architecture for Sub-6 wireless devices is introduced where the different peer connections available through a 802.11ac radio are abstracted as virtual ethernet interfaces connected to an SDN agent. Is this abstraction that allows the Level-0 area controller to control the forwarding plane of the wireless nodes.

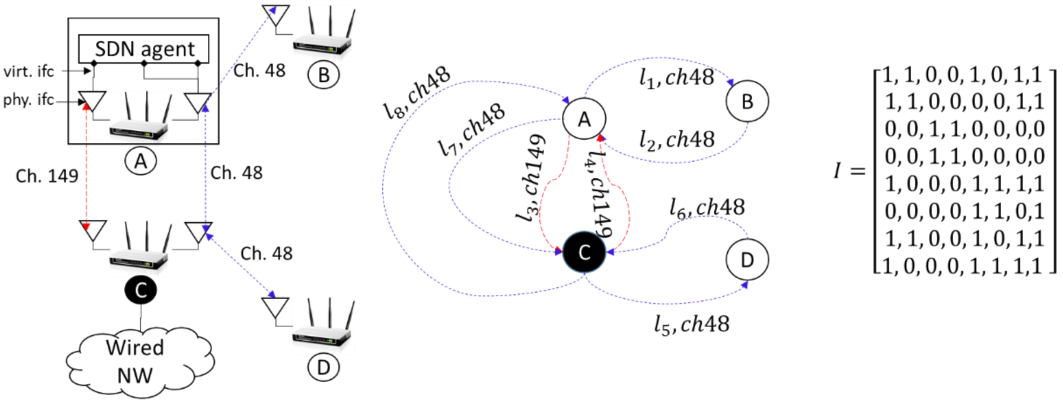

SDN agent
virt. ifc
phy. ifc
A
Ch. 149
Ch. 48
B
Ch. 48
C
Ch. 48
Wired
NW
D
$l_8, ch48$
$l_7, ch48$
$l_1, ch48$
A
B
$l_2, ch48$
$l_3, ch149$
$l_4, ch149$
$l_6, ch48$
C
D
$l_5, ch48$
$I = \begin{bmatrix} 1,1,0,0,1,0,1,1 \\ 1,1,0,0,0,0,1,1 \\ 0,0,1,1,0,0,0,0 \\ 0,0,1,1,0,0,0,0 \\ 1,0,0,0,1,1,1,1 \\ 0,0,0,0,1,1,0,1 \\ 1,1,0,0,1,0,1,1 \\ 1,0,0,0,1,1,1,1 \end{bmatrix}$

Figure 3-5: SDN architecture of wireless Sub-6 TNs.

Centralized computation of default and backup paths

Notice that although the multiple links are abstracted as virtual point to point (p2p) links, in reality different links are subject to interference. For example, in Figure 3-5, the links A-B and A-C and C-D through the "blue" channel interfere with each other. Hence, we describe in this section a centralized mechanism that allows a Level-0 Area Controller to allocate end to end paths across the control plane area, while considering the cross-link interference.

For the purpose of allocating paths for each transport tunnel, the Level-0 area controller (or the simply the controller) instantiates a logical network model depicted in the middle part of Figure 3-5. Automatic topology discovery is enabled using traditional OF mechanisms. Consequently, the controller maintains a set $\boldsymbol{N} = \{n_1, \dots, n_N\}$ of nodes in the network, and a set $\boldsymbol{L} = \{l_1, \dots, l_M\}$ of unidirectional links between nodes. Based on the discovered topology, the Level-0 area controller derives an interference matrix, $\boldsymbol{I} = \{i_{n,m} \in \{0,1\} \mid 1 < n, m \leq M\}$, depicted on the right hand side of Figure 3-5. For the case of Sub-6 TNs, $i_{n,m} = 1$ if the links l_n, l_m

share the same channel and the origin nodes of the two links are the same, or are one hop away in the network graph.

The main goal of the Level-0 area controller is to compute, for each tunnel, a default and a backup path connecting pairs of ETNs or ETN/IATNs within the control plane area. The controller uses a metric $\boldsymbol{\mu}_{\boldsymbol{P}x}$ to evaluate the effect on the overall network state of selecting path $\boldsymbol{P}_x$ as default path for a given tunnel, and a metric $\delta(\boldsymbol{P}_x, \boldsymbol{P}_y)$, which measures the similarity between two paths, in order to allocate backup paths for a given tunnel. We describe these metrics in detail next.

The vector $\boldsymbol{\mu} = \{u_1, \ldots, u_M \mid 0 < u_i < 1\}$ captures the network state by measuring the utilization level perceived by each link in the network. In addition $\boldsymbol{\mu}$ is periodically updated by the controller based on the OF port statistics received from the Sub-6 TNs, as described in D4.11 [14]. To understand how a potential path allocation for a new flow affects $\boldsymbol{\mu}$, we first define a path as a vector $\boldsymbol{P} = \{p_1, \ldots, p_M \mid p_i \in \{0,1\}\}$, where $p_j = 1$ if l_j is included in the path $\boldsymbol{P}$. Considering now that a new flow $\boldsymbol{f}$ of size ρ_f bps needs to be allocated in the network, it is possible to compute the level of utilization introduced by this flow when traversing link li as $u_{f,l_i} = {}^{\rho_f}/_{MTU_i}\,({}^{(MTU_i + MAC\_OH1)}/_{r_i} + MAC\_OH2)$, where MAC_OH1 and MAC_OH2 are respectively per-packet header and airtime related overheads, MTU_i is the MTU size of this link, and r_i is the average transmission rate (in bps) used in that link, which the controller obtains through OF statistics. Consequently, if $\boldsymbol{f}$ is allocated through path $\boldsymbol{P}_x$ the SODALITE controller estimates the resulting level of congestion in the network as:

$$\boldsymbol{\mu}_{\boldsymbol{P}x} = \boldsymbol{\mu} + \boldsymbol{I} * \boldsymbol{u}_{f,\boldsymbol{P}x}$$

Where $\boldsymbol{u}_{f,\boldsymbol{P}x} = \{u_1, \ldots, u_M\}$ contains the utilization level caused by flow $\boldsymbol{f}$ in the links traversed by path $\boldsymbol{P}_x$. Hence, the Level-0 area controller can use the metric $\boldsymbol{\mu}_{\boldsymbol{P}x}$ to compare the impact of allocating different default paths to a given transport tunnel.

Having defined a metric to compare the impact of main paths, we define now the metric $\delta(\boldsymbol{P}_x, \boldsymbol{P}_y)$ that will be used by the controller to evaluate the goodness of a potential backup path $\boldsymbol{P}_y$ given a main path $\boldsymbol{P}_x$. Intuitively, in order to increase reliability we should use backup paths to be as disjoint as possible from the default path. Notice that backup paths in 5G-XHaul are proactively set up to allow for a quick reaction to failure, as described in Section 3.2. We define $0 < \delta(\boldsymbol{P}_x, \boldsymbol{P}_y) < 1$ as a measure of the similarity between different paths in the following way:

$$\delta(\boldsymbol{P}_x, \boldsymbol{P}_y) = \lambda \frac{Intfc_{Px} \cap Intfc_{Py}}{Intfc_{Px}} + (1-\lambda)\frac{N_{Px} \cap N_{Py}}{N_{Px}}$$

Where $Intfc_{Pi}$ and N_{Pi} are respectively the set of physical radio interfaces and the set of nodes used by the links traversed by $\boldsymbol{P}_i$. Thus, if the same set of radio interfaces and nodes are used by $\boldsymbol{P}_x$ and $\boldsymbol{P}_y$, then $\delta(\boldsymbol{P}_x, \boldsymbol{P}_y) = 1$. On the other hand, if completely disjoint sets of radio interfaces and nodes are used by the two paths $\delta(\boldsymbol{P}_x, \boldsymbol{P}_y) = 0$. The parameter λ allows to trade-off the importance of being disjoint in terms of radio interfaces or nodes, which is related to the failure probability of each component. Based on the previously defined metrics, we are now in position to define different policies for path allocation that can be implemented by the 5G-XHaul Level-0 area controller. In [3] we have defined two different policies that trade-off network utilization and the reliability of the backup path. In this deliverable, for the sake of space, we will only describe one of the policies.

Sequential Policy: In this policy main and backup paths are allocated sequentially. Given a set of K candidate paths between two nodes, the controller first allocates the main path as the path that minimizes the maximum link utilization in the network. Once the main path is allocated, the controller searches the set of K paths and allocates as backup path the path that minimises $\delta(\boldsymbol{P}_x, \boldsymbol{P}_y)$. Formally:

$$\boldsymbol{P}_{main} = \underset{1 \le x \le K}{\operatorname{argmin}} \max \boldsymbol{\mu}_{\boldsymbol{P}x}$$

$$\boldsymbol{P}_{backup} = \underset{1 \le y \le K}{\operatorname{argmin}} \delta(\boldsymbol{P}_{main}, \boldsymbol{P}_y)$$

The sequential policy minimizes the network utilization incurred by the main path, at the cost of selecting sub-optimal backup paths. To mitigate this fact we introduce the Joint Policy.

Joint Policy: A joint allocation of the main and backup paths is considered through a multi-objective optimization problem controlled by a weight parameter $0 < \gamma < 1$. The paths are found as:

$$(\boldsymbol{P}_{main}, \boldsymbol{P}_{backup}) = \underset{x,y}{\operatorname{argmin}}(\gamma \max \boldsymbol{\mu}_{\boldsymbol{P}x} + (1-\gamma)\delta(\boldsymbol{P}_x, \boldsymbol{P}_y))$$

In order to find the optimum allocation for the sequential and the joint policies, the set of simple paths[6] between the source and destination nodes should be considered. However, the number of simple paths between two nodes grows exponentially with the size of the network, which renders this approach impractical. In addition, once the set of K paths is computed, the sequential policy has a complexity of $O(2K)$ and the joint policy of $O(K^2)$. It is therefore critical to use K as small as possible, but still be able to select good candidate paths. For this purpose we propose the following heuristic. Given a source destination pair, the Level-0 area controller computes a set of K-shortest paths using the WCETT metric [25]. The WCETT metric weights two factors, the sum of the Expected Transmission Time (ETT) of the links across the path, and a factor that penalizes using multiple links in the same channel. Hence, the WCETT metric should result in a set of candidate paths that are good in terms of minimizing network wide congestion, i.e. $\boldsymbol{\mu}_{\boldsymbol{P}x}$.

Performance Evaluation

In this section we evaluate the sequential and joint policies introduced in this Section. In our evaluation we consider network sizes between 6 and 14 nodes representing a Level-0 Sub6 control area. For each network size, we randomly generate 10 different grid-like topologies, representative of urban scenarios. The Sub-6 TNs are dual-radio, with each radio operating on a different channel, and the transmission rate for each link is randomly chosen between 200 Mbps and 36 Mbps, c.f. deliverable D4.11 [7]. All nodes in the network are collocated with an ETN and can thus be traffic source/sinks.

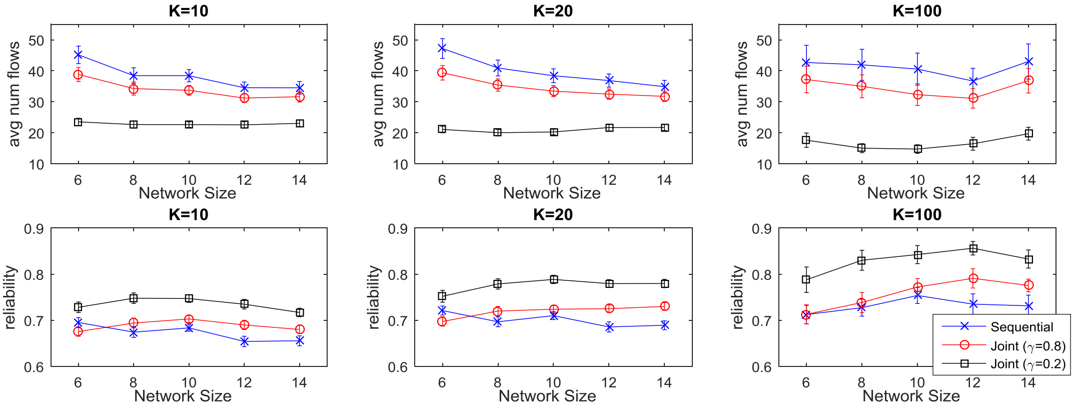


Figure 3-6: Evaluation of main and backup path allocation policies.

Figure 3-6 presents the results of the sequential (blue line) and joint policies, with $\gamma = 0.8$ (red line) and $\gamma = 0.2$ (black line), for the considered random network topologies. The performance of these policies is evaluated in terms of the number of admitted flows, which measures the amount of resources taken by the main path, and the reliability of the backup path. In the former case, flows are randomly generated with a size between 1 Mbps and 5 Mbps, then a path allocation decision is taken according to the policy under study, until the maximum network utilization reaches a configured threshold ($u_{THR} = 0.9$). The resulting number of admitted flows is reported in the upper row of Figure 3-6. To measure the reliability of the backup path, we sequentially remove each physical interface belonging to the main path, and increase a counter cnt if the backup path does not contain that interface. Reliability is then computed as ${}^{cnt}/_{N}$, where N is the number of physical interfaces in the main path. Reliability is depicted in the lower row of Figure 3-6. In addition, $K = \{10, 20, 100\}$ are considered in the study, to evaluate the effect of the K-shortest path heuristic on the performance of each policy.

Looking at Figure 3-6 we can see that for all values of K the policies behave as expected with the sequential policy providing the highest performance in terms of admitted flows, followed by the joint-0.8 and the joint-0.2 policies, whilst the trend is reversed when looking at the reliability of the backup path. The effect of K on the number of admitted flows is relatively small (only visible for larger topologies), which validates the ability of the WCETT K-shortest path heuristic to generate a set of candidate paths with a reduced network utilization. The performance of the reliability index improves more significantly, especially for the joint-0.2 policy, when $K = 100$ is considered. The reason is that with larger K, longer and potentially more disjoint backup paths are included within the set of candidate paths. However, longer backup paths may lead to a reduced performance until the SDN controller realizes that the main path has been disrupted and re-configures the network. We consider that $K = 20$ is a good compromise between computation complexity in the Level-0 area controller, and the performance reliability trade-off obtained under the different policies.

3.2 Challenge 2: Distributed control plane agent for fast recovery

As mentioned in the introduction of this chapter fast recovery mechanisms are essential to providing a carrier-grade service. However, strictly following the SDN architecture [9] where all the control plane mechanisms are centralized in the controller, poses a challenge with regard to fast recovery, especially when the controller is located far away from the TNs, as is the case of 5G-XHaul. To address this challenge 5G-XHaul has adopted the following generic scheme:

- A Level-0 area controller proactively programs a main, or default, path and a backup path in the TNs.
- A distributed fast recovery agent operates locally in each TN, detects if a given link has failed, and quickly switches the packets towards the backup path.
- After realizing that packets have been moved to the backup path, the Level-0 area controller may decide to re-configure the transport tunnels if necessary.

This section describes and evaluates two different fast recovery agents. The first one, is a generic agent that can be applied in any 5G-XHaul TN, regardless of the underlying physical technology, and is based on OpenFlow (OF) 1.3 group tables. The second mechanism is specific to Sub-6 wireless TNs based on 802.11 radios. The limited resources available in this type of radios justifies the design of a specific mechanism, which includes a custom cross-layer scheme to quickly detect link failures.

3.2.1.1 A generic fast recovery agent for 5G-XHaul control plane areas

As already mentioned, in the context of a 5G-XHaul control plane area, the latency caused by waiting for a remote Level-0 area controller (AC) to take action (i.e. reactive failover) might be unacceptable. Another concern is related to the possibility of temporary loss of connection with the AC. Thus, proactive approaches incorporating alternative paths directly at the data plane or, in the worst case, stored at the memory of local agents at each transport node, appear attractive.

Below we describe a proactive failover approach which works on an Ethernet substrate, and can therefore be used for any type of physical transport technology offering an Ethernet abstraction. Our approach builds on the notion of the tunnel ID, which is part of the PBB header, introduced in chapter 2, of packets traversing the TNs and determines the routing decisions. Note that OpenFlow version 1.3 or newer is required for supporting PBB. The tunnel ID in our design is not tied to a single node sequence, but also involves alternative routes towards the destination. We will be using the following terminology in this subsection, scoped to a given tunnel:

- *Path*: Any node sequence beginning and terminating at the source and destination ETN/IATN of the tunnel, respectively (and comprising only TNs in-between).
- *Primary path*: The default path used by the tunnel in the absence of link failures affecting it.
- *Backup subpath*: A node sequence beginning and terminating at nodes in the primary path (not necessarily the source/destination ETN/IATN), and featuring at least one TN not present in the primary path.
- *Backup path*: A path including a backup subpath in its node sequence.

Essentially, in the described scheme, whenever a link in the primary path fails, a backup path allows the flows using the tunnel to be rerouted through a backup subpath beginning at the node which detected the failed link or at a node preceding it in the primary path node sequence. This process is transparent to tenants utilizing these tunnels and even to the ETNs/IATNs communicating with them, and minimizes delays and dropped packets.

One thing to note is that, in principle, load balancing can take place for packets of a given tunnel ID, irrespective of potential link failures. This can be seen as a form of implicit failover. However, it does not deal with dropped packets until the load balancing mechanism completely avoids the malfunctioning path. Furthermore, centralized planning of load balancing solutions is based on different assumptions and guarantees with respect to planning failover mechanisms. In describing this scheme, we assume for simplicity of presentation that, at any given instant, traffic of a specific tunnel ID follows a unique path to the destination. However, the same approach is readily applicable to multi-path load-balancing tunnels.

In general, the AC plans primary and backup paths for its tunnels based on a combination of considerations, reflecting its view of the network status and expected tunnel utilization. Our scheme focuses on the functionality required at the individual TN level, so it does not make any assumptions about AC's particular policy. Given the AC's decision about the primary path and the subset of nodes to be used in backup paths for each tunnel, we have designed and implemented a simple OpenFlow-based fast failover scheme, with the following two properties: i) Guarantee of recovering from any single link failure, as long as the ETNs/IATNs in the area remain connected, ii) Prevention of loops irrespective of the number of failures that might take place.

To this end, we have identified two basic rules, which are sufficient conditions for a policy to meet the above requirements, while incurring a relatively small increase in the number of flow entries to be added in the transport nodes. To best illustrate these rules, we consider the example topology of Figure 3-7, scoped to a single tunnel which transfers packets from ETN A to ETN B. In this topology we assume that the primary path for the tunnel is the sequence A -> 1 -> 2 -> 3 -> 4 -> 5 -> B. The other TNs are only used for backup purposes.

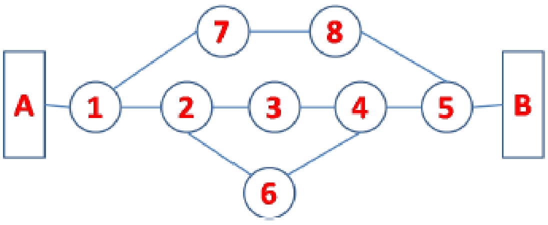


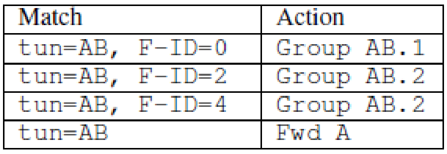
Figure 3-7: Example topology for presentation of fast failure recovery policy.

The first rule is to only provision backup subpaths for failures of links whose transmitting end is a node lying in the primary path of the tunnel, and not a TN in backup paths. For instance, if the link 2 -> 3 fails, the subpath 2 -> 6 -> 4 is used to circumvent the failure. If, in addition, the link 2 -> 6 is found not to be working, then the subpath 2 -> 1 -> 7 -> 8 -> 5 is used. However, no backup paths exist for this tunnel in case, for instance, links 6 -> 4 or 7 -> 8 fail. Note that only providing a single backup solution (2 -> 6 -> 4) would still satisfy the single link failure recovery guarantee. The programming of the second failover subpath is thus not strictly required; however, we have included such subpaths in our design to make it even more robust.

The second rule is that whenever a backup subpath used to circumvent a failure diverges from the primary path, it may only remerge with it at a TN closer to the destination ETN/IATN with respect to the transmitting end of the failed link. For instance, if link 4 -> 5 fails, TN 4 will not use subpaths 4 -> 6 -> 2 or 4 -> 3 -> 2 -> 6 -> 4, because after those diverge from the primary path they return to it at TNs 2 and 4 respectively, which are not closer to ETN B. Instead, the subpath 4 -> 3 -> 2 -> 1 -> 7 -> 8 -> 5 will be used.

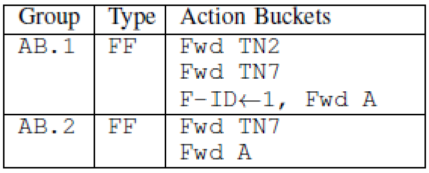
In order to support the full range of failover subpaths complying with the above rules, there is a small extension that needs to be made in the data plane. In some subpaths packets may move backwards along the primary path, such as in the subpath 4 -> 3 -> 2 -> 1 -> 7 -> 8 -> 5 in the example. In order for this to work without potential loops, the packet must be tagged with an identifier related to the node that detected the failure, which we call failure identifier or F-ID for short. The header field holding the F-ID will only be set to some nonzero value for backward moving packets, otherwise it will be zero. It is tunnel-specific, thus its maximum value is the number of hops in the primary path (e.g. only 3 bits are needed for 8-hop paths). It could be part of the B-VID, for instance. The F-ID will be matched by at most a single TN in the primary path. For instance, if link 4 -> 5 fails, the F-ID makes sure that the subpath 2 -> 6 -> 4 is not used to redirect these packets (something that would create a loop).

In order to better see how this policy is reflected in the flow tables of the TNs, we show in Figure 3-8 to Figure 3-11 the entries of the rule table and the group table for TN1 and TN3 in our example topology and tunnel AB. Note that OpenFlow groups of Fast Failover type are used, whenever we have a match. The action buckets in such groups are linked to port liveness statistics. Thus, if the port associated with an action bucket in the group is found not to be live, the next bucket is executed, without waiting for instructions from a remote controller. Also note that if a TN does not have a rule installed for a specific F-ID of a backward moving packet, then the default action is to forward the packet to the preceding TN without modifying its F-ID.



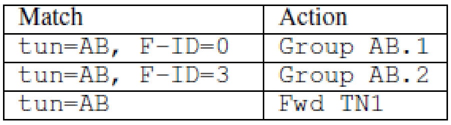
Match	Action
tun=AB, F-ID=0	Group AB.1
tun=AB, F-ID=2	Group AB.2
tun=AB, F-ID=4	Group AB.2
tun=AB	Fwd A

Figure 3-8: Rule table for TN1.



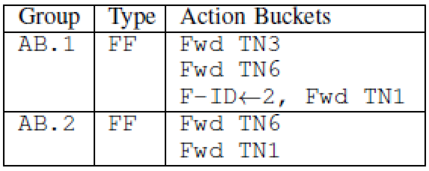
Group	Type	Action Buckets
AB.1	FF	Fwd TN2 Fwd TN7 F-ID←1, Fwd A
AB.2	FF	Fwd TN7 Fwd A

Figure 3-9: Group table for TN1.



Match	Action
tun=AB, F-ID=0	Group AB.1
tun=AB, F-ID=3	Group AB.2
tun=AB	Fwd TN1

Figure 3-10: Rule Table for TN2.



Group	Type	Action Buckets
AB.1	FF	Fwd TN3 Fwd TN6 F-ID←2, Fwd TN1
AB.2	FF	Fwd TN6 Fwd TN1

Figure 3-11: Group table for TN2.

There is a subtlety related to efficiency when packets are moving backwards. Ideally, whenever a TN receives a packet with a nonzero F-ID and it has a suitable subpath to reroute packets with this F-ID, we would also like this TN to reroute untagged packets of the same tunnel through this subpath, in order for them to avoid an unnecessary forth and back trip while the link failure persists. However, this cannot be simply implemented with the OpenFlow fast failover group table. It requires retaining state at TN level (not packet tags). In this case, the above procedure should be extended with the TN sending the first tagged packet it matches also towards its Local Agent (LA). The latter will update the flow entry for untagged packets, in order to match the one with tagged packets. This incurs an added delay for the untagged packets received in the meantime, which is however small compared to waiting for the reaction of a remote AC.

Assessment of additional TCAM requirement of proposed policy

The number of rules installed in the forwarding elements affects the FIB delay. Also, if this number grows excessively, low-end switching elements might not be able to store such amounts. We are thus interested in keeping the overhead of our failover scheme, in terms of additional TCAM memory requirement at the TNs, at moderate levels. Note that the maximum number of flow entries depends on the size of each area, the proportion between ETNs/IATNs and TNs, the area topology, and the AC's path planning policy.

As a first attempt to assess these numbers, we generated random "small world" network topologies of TNs and ETNs/IATNs, where each TN is connected in average with 4 other TNs, and varied both their total number and proportion. We considered both the simple case where a single unique path exists for connecting two ETNs/IATNs in an area, and the more demanding case where a number of redundant flow entries are installed according to the described scheme, for fast failover purposes. The assessment has been done using the R toolkit, and we have used shortest path routing as the AC's policy (i.e. we ignore potential constraints, for instance related to link capacity).

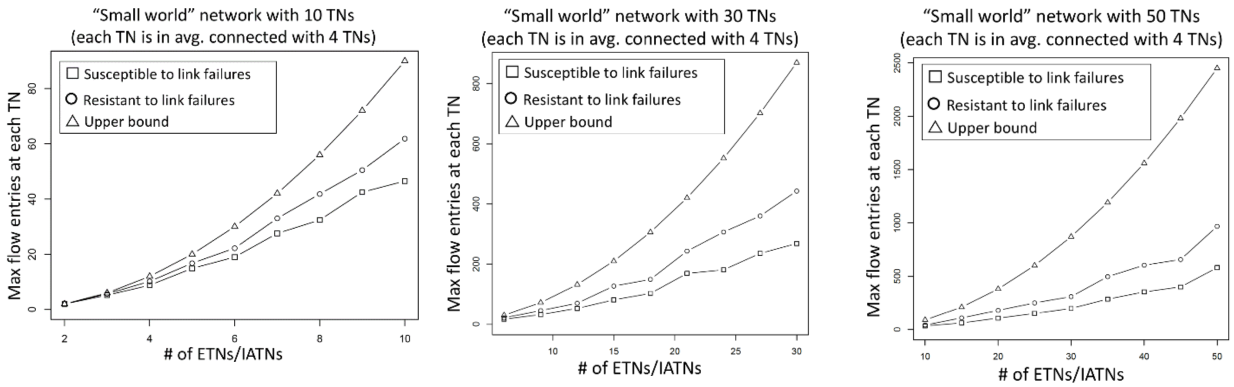


Figure 3-12: Number of flow entries with and without single-link failover.

Ignoring failover entries, if N is the number of ETNs/IATNs in the area, each TN can have at most N*(N-1) flow entries. This upper bound, however, is too pessimistic in practice. Our simulation results, shown in Figure 3-12, indicate that the real number required, assuming a "small world" topology, is much lower. Note that we generally consider one tunnel for each source-destination pair here, for simplicity. These numbers could be scaled by a number reflecting the requirement for different tunnels depending on QoS traffic classes for instance, but the scaling factor would be the same.

In the same figures we assess the increase in the number of flow entries due to the failover scheme. We can see that the application of the described fast failover scheme yields a relatively small increase in the requirement for additional flow table entries at the TNs, especially when the number of ETNs/IATNs in the area is fairly smaller than that of TNs. We thus see that single-link failures can be managed directly at the data plane with manageable overhead.

3.2.1.2 A fast recovery agent for Sub-6 wireless transport networks

In this Section we propose a local agent for fast recovery embedded in the Sub-6 TNs. We hereafter refer to this agent as the *Fast Local Link ReRoute* (FLRR) agent. The FLRR agent listens to link break notifications from the MAC layer of the Sub6 TNs, and reconfigures the SDN agent to quickly switch the affected flows to the pre-provisioned backup path. This section is organized in three subsections. First, we describe the operation of the FLRR agent, and second, we describe the low level mechanisms that report a link failure in the Sub-6 radios. An experiment illustrating the performance of the FLRR mechanism is reported in deliverable D5.1.

Operation of the Fast Local Link Reroute (FLRR) agent

After a default (main) and backup path have been determined, the Level-0 area controller assigns four different types of logical roles to the nodes along the paths of the transport tunnels: common nodes, switch nodes, intermediate nodes and Next-To-Last (NTL) nodes. The roles are assigned as:

- Nodes that originate links that are common to the default and backup paths are identified as *common nodes*, unless they are switch nodes, as detailed below.
- A node where the default and backup paths branch-off is defined to be a *switch node*.
- Nodes that originate links exclusively for the default or the backup path are identified as *intermediate nodes*, unless they are a NTL node, as detailed below.
- Whenever a node on the default path is the origin of a link that connects to a common or switch node, i.e., where the default and backup path merge, it is defined to be a *NTL node*.

A sample topology highlighting the role assigned to each node is depicted in the left part of Figure 3-13. Notice that an arbitrary topology could include further common, intermediate, switch or NTL nodes.

Once the roles have been assigned to the nodes on the default and backup paths, the controller proceeds with the installation of FLRR-specific rules in the SDN agent of each TN involved in the default and backup paths. There exist four different rules in FLRR: i) *forwarding rules*, present in all FLRR nodes, *regress rules*, present only in intermediate nodes, *crankback rules*, pushed locally by the FLRR agent as explained later, and *switch rules*, present only in switch nodes.

Each of the rules performs matching on two fields: the *path-ID* specific to the transport area (c.f. Section 5 in deliverable D3.1 [1]) and the input port via which the data packet entered the virtual switch. While the path-ID is the key identifier for different end-to-end data flows, the input port is necessary for FLRR to take autonomous recovery decisions (as explained later).

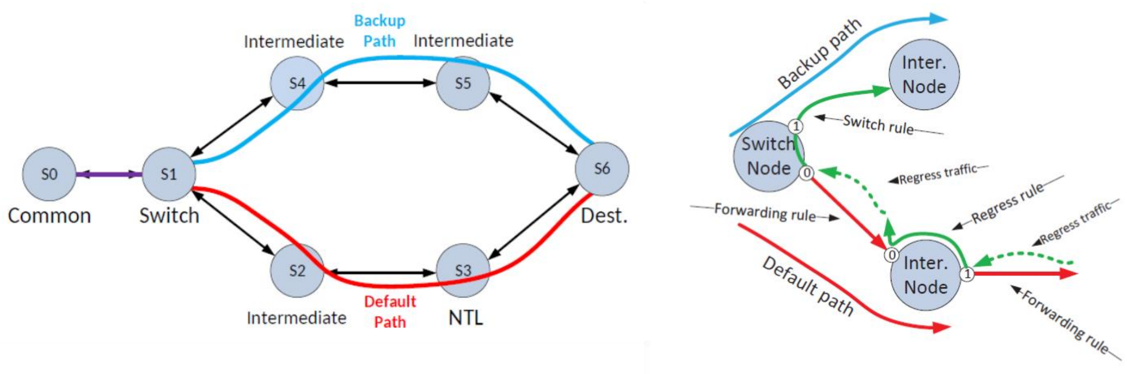


Figure 3-13: Node and rule types used by the FLRR agent.

Forwarding rules are used for the default and backup paths, pointing towards a path's destination, i.e., the gateway or entry nodes for up- and downstreams, respectively. Forwarding rules match on the input port through which packets enter the switch from the previous hop and they output the packets via the port that connects with the next hop on the path. Crankback rules are of temporary nature: once a link on the default path is considered broken, the forwarding rule of the node on which the link issue is detected is replaced with a crankback rule. The crankback rule sets the output port of a forwarding rule to be the same as the input port. This reverts the traffic direction and makes packets return to the previous hop.

On the way back, either a regress rule or a switch rule intercepts the regressing traffic, depending on whether the previous hop is an intermediate or a switch node, respectively.

A regress rule forwards regressing traffic, i.e. traffic that traverses an intermediate node in opposite direction, further back towards the previous node on the path. Further, once a regress rule starts matching packets, the existing forwarding rule for the same *path-ID* is replaced with a crankback rule, which avoids further traffic from being sent in direction of the broken link. Once regressing traffic hits a node with a switch rule, the traffic is redirected towards the backup path, over which the destination node can be reached. As soon as packets start matching the switch rule on a node, the output port of the original forwarding rule with the same *path-ID* is modified to point towards the backup path. This avoids any further packets from being sent over the broken default path, instead they are directly sent over the backup path.

The right part of Figure 3-13 illustrates a snapshot of an example topology where the Level-0 area controller just installed a set of FLRR rules. The continuous lines show the preinstalled rules. The dashed lines indicate how traffic would flow if a link break is detected on the default path, whereas the dotted lines indicate crankback rules that replace the original forwarding rules.

Quick detection of link breaks in Sub-6 TNs

The dynamics of link failures in wireless networks are more complex than in wired networks. For example, a link could temporarily fail due to congestion, or an energy conserving module in the Level-0 area controller could decide to switch off a TN.

Traditional schemes to detect a link break between adjacent nodes require the transmission of keep-alive messages, and consider a link broken after a configured number of consecutive keep-alive messages are lost. There is thus an inherent trade-off between link break detection time and signaling overhead. Maximizing the bandwidth available to user plane transmissions is especially critical in wireless backhaul networks, hence we propose a hybrid cross-layer scheme to detect link breaks, where existing MAC signaling is re-used for this purpose. The detailed description that follows is based on 802.11 radios, but similar principles could be applied to other radio technologies.

IEEE 802.11 radios periodically generate beacon frames to facilitate tasks such as node discovery. Sub6 TNs interpret the transmission of beacon frames from peer nodes as keep-alive messages for the purpose of maintaining link reliability. If a certain number of consecutive beacons is lost, or are received with a RSSI below a configured threshold, the Sub-6 radio notifies the FLRR agent that a peer node has vanished. Notice that link break detection time can be decreased by reducing the Beacon Interval (BI), which however increases signaling overhead. To resolve this tradeoff we also piggyback on another signal from the MAC to trigger a link break detection. If a packet transmitted from a node to its peer exceeds a configured number of retransmissions, that link is considered broken and the FLRR agent is notified.

The proposed hybrid scheme addresses the trade-off between link break detection time and signaling overhead in the following way. When there is no traffic flowing between two nodes, and therefore quick link break detection times are not critical, a link break is detected through a consecutive number of missed beacons, which may result in increased detection times if a large BI is used. On the other hand, when traffic is flowing between two nodes and link break detection time is critical to avoid packet loss, an excessive number of retransmissions is the signal used to quickly notify the FLRR agent about a link break.

As previously mentioned, the FLRR scheme described in this section has been used in deliverable D5.1 to illustrate a fast re-route of flows in a Sub6 control plane area deployed in the NITOS testbed.

4 DETAILED DEFINITION OF 5G-XHAUL SDN SOUTH-BOUND INTERFACES

In this chapter, we describe the south-bound interfaces for the optical and wireless network transport technologies considered in 5G-XHaul, specifically for TSON, WDM-PON, and millimetre wave TNs. A south-bound interface definition for Sub-6 TNs was already described in deliverable D4.11 [7]. In following subsections, we describe the SDN south-bound solutions for the considered technologies.

4.1 TSON SDN description

This section focuses on the Time-Shared Optical Network (TSON) SDN and OpenDayLight (ODL) controller integration, which allows allocating resources (one or more timeslots) to multi-granularity network services on one or more wavelengths. Figure 4.1 shows the high-level TSON software architecture that has been implemented and fully integrated in the ODL framework. The scope of ODL is to discover the physical topology, compute the sub-lambda light-paths between pairs of devices of the network graph, and allocate network resources, setup or remove cross-connections on these devices. The control plane (i.e. the ODL controller) is a set of applications that exchange data for data plane management, the Sub Lambda Allocation Engine (SLAE) and the Application layer. The TSON data plane is composed of different physical optical devices (e.g. FPGA NICs, optical fast switches) connected to the SDN ODL controller through a dedicated interface. Within the 5G-XHaul architecture TSON data-plane nodes correspond to TNs, while the TSON control plane would be implemented by a Level-0 Area Controller.

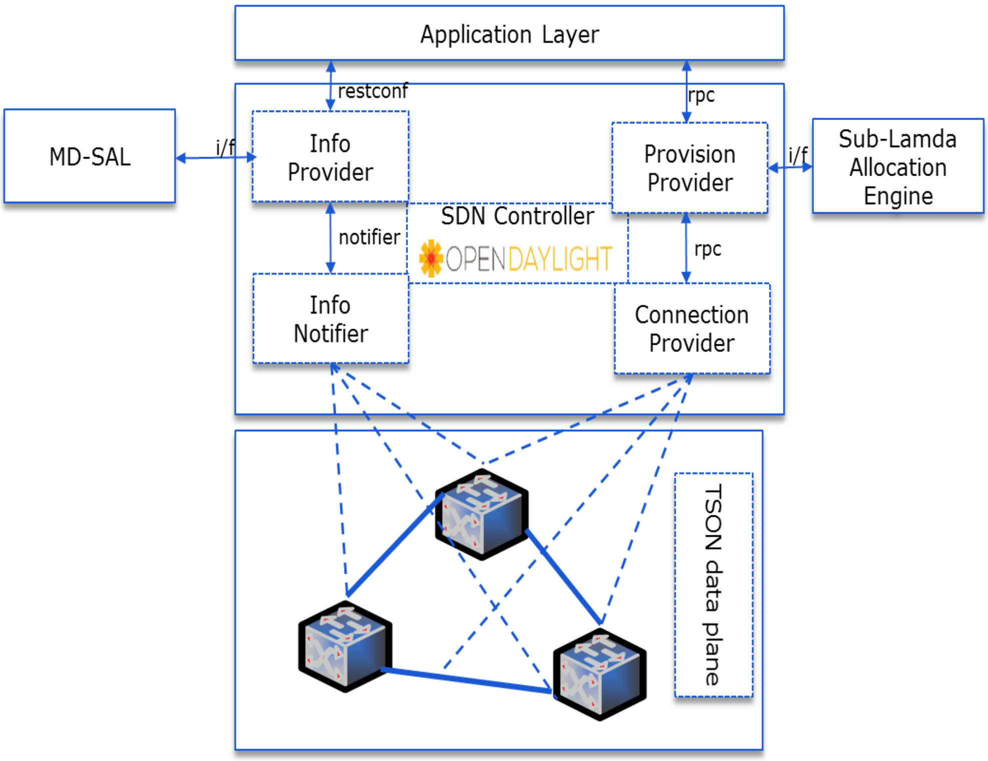


Figure 4-1: General architecture of TSON SDN.

The basic functions of TSON physical domains are implemented and matched into multiple internal modules within the Level-0 Area Controller to provision TSON devices on-demand. These modules are wrapped in ODL bundles that provide a set of services to configure and collect the information about the TSON physical domains. As described in deliverable D3.1 [3], the TSON ODL topology manager is composed of multiple internal modules such as: TSON connection service, TSON path computation service and TSON topology service. To be in-line with the common ODL approach, the interface of the services implementing the TSON basic functions is described through the YANG language. The detailed TSON internal modules and the YANG description of their interface has been reported in deliverable D3.1. In addition, the following subsections describe submodules depicted in the Figure 4-1 (i.e. Notifier, Provider), the ODL key functionalities and their roles.

4.1.1 TSON SDN ODL controller

The TSON SDN controller enables management of the TSON network. The TSON ODL controller consists of Info provider, provisioning, connection provider and Info notifier modules. The roles of these different modules are the following:

- **Info Notifier**: It provides information about the features in terms of resources of the different nodes available in the TSON domain. It establishes and maintains the connections with the TSON data plane and provides a description of the devices capabilities (e.g., equipment, ports and resources).
- **Info Provider**: Maintains information about the physical topology of underlying network. It also provides north-bound interface connection through *restconf* to retrieve the configuration parameters from the application layer.
- **Provisioning Provider**: It manages the Remote Procedure Call (RPC) messages coming from the application layer. According to RPC messages the optimal path is calculated by considering the data provided by SLAE. It sends the obtained result about the resources allocation calculation to the Connection Provider module to install the cross-connections.
- **Connection Provider**: It sends the provisioning commands to TSON data plane devices based on the RPC messages received from the Provisioning Provider module.
- **Sub Lambda Allocation Engine (SLAE):** It computes the optimal path of the TSON data plane.
- **Model Driven Service Abstraction Layer (MD-SAL)**: It stores information related to nodes, node connectors and links providing the discovered topology to the other modules of SDN controller.

4.1.2 High level view of TSON support and components in ODL

Figure 4-1 depicts the general architecture of TSON SDN overview. In Figure 4-2 we present the implemented ODL components to support TSON technology.

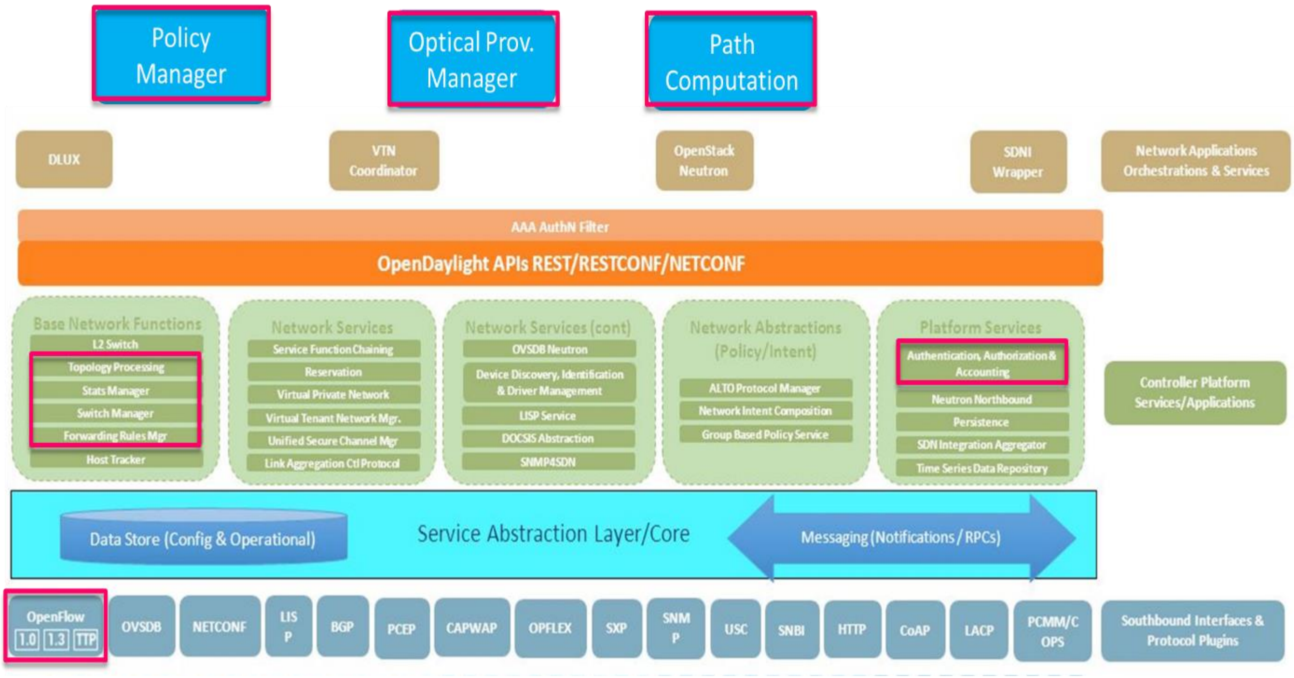

Policy Manager
Optical Prov. Manager
Path Computation
DLUX
VTN Coordinator
OpenStack Neutron
SDNI Wrapper
Network Applications Orchestrations & Services
AAA AuthN Filter
OpenDaylight APIs REST/RESTCONF/NETCONF
Base Network Functions
L2 Switch
Topology Processing
Stats Manager
Switch Manager
Forwarding Rules Mgr
Host Tracker
Network Services
Service Function Chaining
Reservation
Virtual Private Network
Virtual Tenant Network Mgr.
Unified Secure Channel Mgr
Link Aggregation Ctl Protocol
Network Services (cont)
OVSDB Neutron
Device Discovery, Identification & Driver Management
LISP Service
DOCSIS Abstraction
SNMP4SDN
Network Abstractions (Policy/Intent)
ALTO Protocol Manager
Network Intent Composition
Group Based Policy Service
Platform Services
Authentication, Authorization & Accounting
Neutron Northbound
Persistence
SDN Integration Aggregator
Time Series Data Repository
Controller Platform Services/Applications
Data Store (Config & Operational)
Service Abstraction Layer/Core
Messaging (Notifications / RPCs)
OpenFlow
1.0
1.3
TTP
OVSDB
NETCONF
LISP
BGP
PCEP
CAPWAP
OPFLEX
SXP
SNMP
USC
SNBI
HTTP
CoAP
LACP
PCMM/COPS
Southbound Interfaces & Protocol Plugins

Figure 4-2: High-level view of ODL components implemented to support TSON.

The description of the components included in Figure 4-2 is as follows:

- **OpenFlow java Library:** It manages the communication of low level interfaces between the controller and the devices operating with OpenFlow protocol by serializing and de-serializing the

OpenFlow messages. In our case, serialization means translating the high-level MD-SAL java objects to bytes.

- **OpenFlow plugin:** It provides a layer of abstraction between the OpenFlow java module and the MD-SAL components of ODL. In addition, it provides inventory information to the controller with statistics about the device flows and ports.
- **GUI:** It constitutes the graphical interface of the SDN controller. Also, it enables the display of the TSON topology.
- **Path Computation Manager (PCM):** It is the SDN controller module dedicated to the optical and multi-layer path computation. In this deployment, the actual computation is performed in the application layer to assign timeslot and wavelength allocation, while the PCM translates the result in ODL structures, making it available for the other controller components.
- **Optical Provisioning Manager (OPM):** It coordinates the provisioning of TSON connections, based on the network paths computed by the SLAE. It is responsible for the configuration of TSON look-up table. It has been extended to enable the configuration of the FPGA-based Optical NIC.
- **Topology Manager (TM):** It extends the ODL capabilities to maintain and update TSON topology based on information available in the TSON edge nodes. It leverages on the updates received as notifications from the ODL data store and it is triggered by the OpenFlow plugin. TM has been extended to classify FPGA-based optical NICs. The TSON topology is then exposed via REST API to enable the computation of TDM paths across the data plane.

4.1.3 OpenFlow Agent for FPGA NIC

To translate the information passed from the SDN controller to the FPGA-based NIC, we implemented the OpenFlow Agent shown in Figure 4-3. This C-based OpenFlow agent is acting between the SDN controller and FPGA-based NIC. The TSON FPGA-based NIC is currently implemented on NETFPGA SUME which has 4 10 Gbps SFP+ ports available. One of the ports is used to communicate with the SDN agent, the other three include one Ethernet interface (as a client service) and two TSON interfaces. The number of timeslots on which the NIC transmits data is controlled by an OpenFlow agent connected to the extended ODL controller [28]. The buffers inside the FPGA logic enable the assignment of timeslots to two output ports based on different matching fields such as: destination address, source address, and VLAN ID. Notice that these fields are consistent with the PBB transport encapsulation adopted by 5G-XHaul, and described in detail in Chapter 2. As shown in Figure 4-3, an OpenFlow agent constructs device configuration messages based on Ethernet framing. The FPGA-based NIC parses incoming packets (e.g., read from SFP) like an electronic switch, which implies that it can forward traffic according to their L2, L3 or even other layer features. This FPGA-based NIC not only schedules packets in TDM fashion but it also supports allocation of TDM slots on two different wavelengths. Here, from a resource allocation perspective, a flow table (instead of cross connection matrix) with flow entries including a match field and its associated actions (e.g. the timeslot within the TSON frame where packets will be forwarded) is maintained in the agent to indicate the existing flow forwarding rules and resource availability.

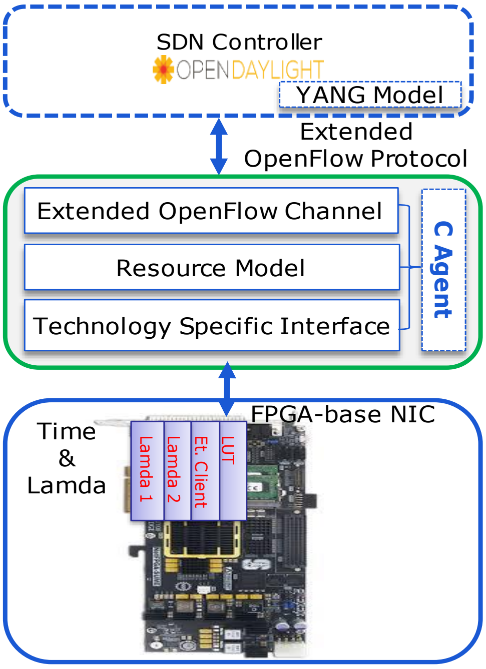


Figure 4-3: FPGA-based Optical Agent Design.

4.1.4 Results: TSON timeslot allocations through SDN

To achieve the required flexibility and programmability, the TSON implementation has several programmable parameters (reported in deliverable D3.1 [3]) such as: *Quality of Transmission (QoT)*, incurring an overhead from 0 to 39 KB, *size of timeslots*, taking values from 80 Bytes to 31.25 KB, *timeslot per frame count,* taking values from 4 to 100, and timeslot allocation on different wavelengths. For example, when a network operator and/or service provider requests dedicated network services, these are provisioned after evaluating requirements such as bit-rate, connectivity type, QoS and QoT. Thus, the corresponding parameter values and timeslot allocation at TSON nodes can be configured in order to fulfil these requirements. However, for evaluation purposes only, the extended Open Flow protocol also supports to configure timeslot allocation, while the other low level parameters remain fixed.

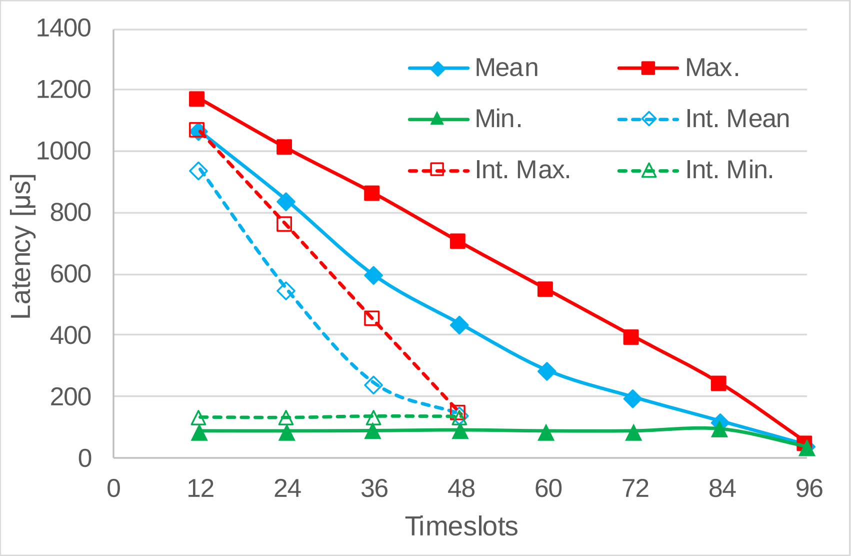


Figure 4-4: Latency measurements as function of allocated timeslots with Contiguous and Interleaved allocation schemes.

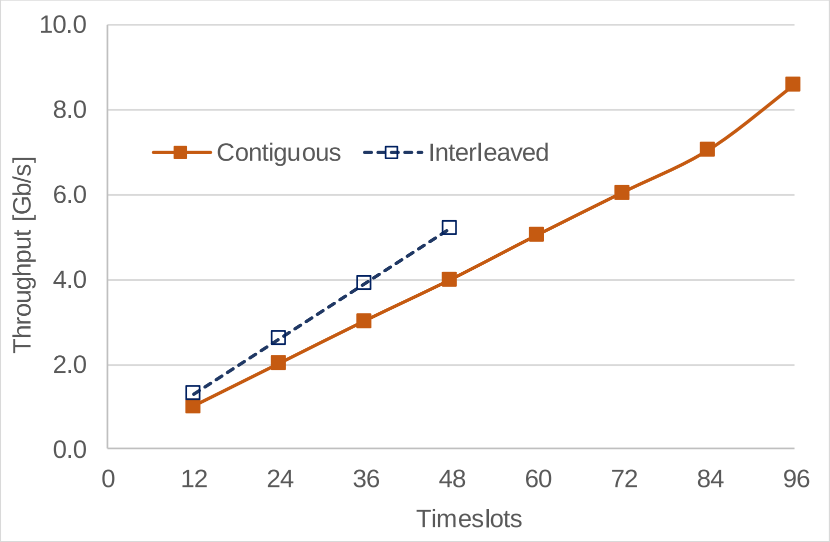


Figure 4-5: Throughput measurements as function of allocated timeslots with Contiguous and Interleaved schemes.

The performance of TSON network is evaluated in terms of throughput and latency for contiguous and distributed (i.e., interleaved) timeslot allocation when the number of slots allocated to a connection grows. A contiguous allocation is an allocation where all the time slots of a connection are packed together within the TSON frame. An interleaved allocation is an allocation where the slots assigned to the connection are spread within the TSON frame. Figure 4-4 shows the maximum, minimum and mean latency in the considered scenarios. We can see that interleaved allocation outperforms the contiguous one in maximum and mean latency, because interleaving reduces the maximum delay between data transmission slots. This is worth noting in order to avoid contiguous allocation and to optimise performance. Moreover, the maximum and mean latency measurements converge as timeslots increase, because the gap between transmission timeslots reduces. On the other hand, the interleaved minimum value is greater because always a transmission timeslot is followed by a non-transmission timeslot. Throughput measurements are presented in Figure 4-5, where it is shown that sustainable maximum data rates of up to 8.6 Gbps, 5.2 Gbps can be obtained in contiguous and interleaved timeslot allocation scenarios, respectively. However, it is worth noting again that interleaved allocation of timeslots performs better than contiguous one.

4.2 SDN interface for WDM-PON

As elaborated in deliverable D3.1 [3], the internal management and monitoring of the WDM-PON system developed in 5G-XHaul is realized by an embedded communication channel (i.e. Auxiliary Management Control Channel-AMCC) superimposed on the payload. The centralized local controller in the OLT sends commands to remote ONUs, and retrieves the physical parameters (defined in D3.1), as shown in Figure 4-6.

The WDM-PON system will be fully SDN-enabled for remote control and monitoring with ODL by hosting a NETCONF service in the Linux system of the OLT's internal controller board. The mentioned ODL would correspond to the Level-0 Area Controller which is in charge of managing WDM-PON.

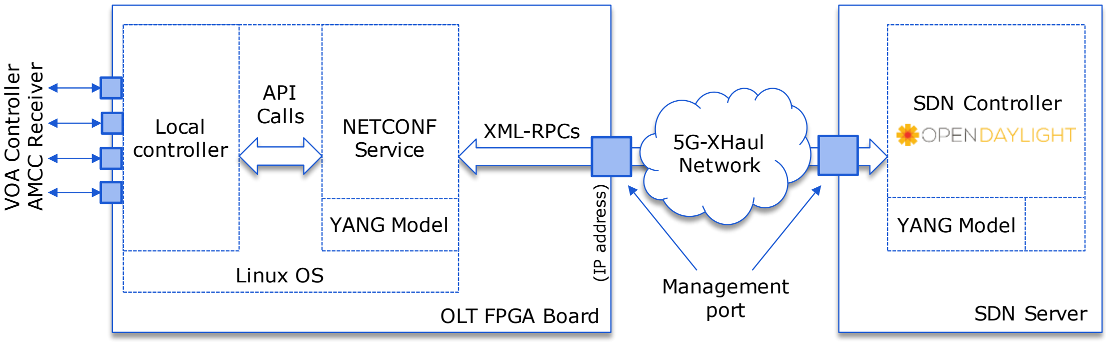


Figure 4-6: SDN South-bound interface of WDM-PON.

The NETCONF service internally communicates with the OLT's low-level controller software and the integrated FPGA to send commands to and fetch information from the embedded hardware components of the OLT, and therefore also to retrieve status information regarding the connected ONUs.

Although designed to be integrated as a pluggable module in the ADVA FSP3000R7 (ADVA F7) WDM platform, the OLT will provide an exclusive RJ45 management port for connection to the 5G-XHaul management network.

The NETCONF protocol is one of today's most used SDN management communication standards and can be seen as a next generation SNMP alternative, which in contrast to the latter uses XML messages exchanged through common stream-oriented transport protocols like HTTP(S) or SSH. It enables an SDN controller to perform RPCs in the OLT, fetch and update configuration data, and perform monitoring operations.

The capabilities of a NETCONF service interface are usually defined using YANG models, which describe the structure of the configuration and monitoring data and the names and parameters of the RPC functions. ADVA will create a suitable YANG model for the OLT that will be freely shared will all project partners, and which can be easily integrated into the NETCONF plugin of ODL.

We include in Table 4-1 a brief description of the OLT's remote control and monitoring features:

Table 4-1: Preliminary SDN features of WDM-PON.

Control items		
Command	**Refers to**	**Action**
Set input port - wavelength relationship	Input port / wavelength port	Set input switch to connect input port to fix-wavelength laser (provides input-port to ONU connection)
Turn off ONU	Specific wavelength ONU	Send command on specific wavelength to disable ONU TX
Turn on ONU	Specific wavelength ONU	Send command on specific wavelength to enable ONU TX
Monitoring items		
Parameter	**Refers to**	**Details**
RX power	Wavelength	Received power on specific wavelength to monitor operation
Wavelength deviation	Wavelength	Grid-deviation of wavelength received from ONU
ONU status	ONU	Status / health of ONU, identified by wavelength
Input-port - connection	Input port	Connected wavelength
ONU identifier	ONU	Identifier or geo-location of ONU on particular wavelength

A YANG definition for an excerpt of those features above could look like:

```
module adva-wdm-pon {
    namespace "urn:adva:params:xml:ns:yang:adva-wdm-pon";
    prefix pon;
    organization "ADVA Optical Networking";
    contact "http://advaoptical.com";
    description "NETCONF interface definition for OLT of ADVA WDM-PON";
    revision 2017-05-13 {
        description "Initial sample";
    }
    list onu {
        key 'wavelength';
        description "The list of connected ONUs identified by wavelength";

        leaf wavelength {
            type uint16;
            units "nm";
            description "The wavelength of the ONU";
        }
        leaf rx-power {
            type int16;
            units "dBm";
            description "The received power on the wavelength of the ONU";
        }
        leaf identifier {
            type string;
            description "The identifier of the ONU";
        }
    }
    rpc turn-on-onu {
        description "Turn on a specific connected ONU given by wavelength";
        input {
            leaf wavelength {
                type uint16;
                units "nm";
                description "The wavelength of the ONU";
            }
        }
    }
    rpc turn-off-onu {
        description "Turn off a specific connected ONU given by wavelength";
        input {
            leaf wavelength {
                type uint16;
                units "nm";
                description "The wavelength of the ONU";
            }
        }
    }
}
```

An operational NETCONF interface for WDM-PON will be reported in deliverable D3.3.

4.3 SDN control of a millimetre wave mesh nodes

In this section, we describe the SDN control of a TN comprising a network processor (NPU) and a number of mmWave modems using IEEE 802.11ad. Such nodes can be connected together wirelessly into a meshed architecture as shown in Figure 4-7. The SDN controller exploits a mixture of OpenFlow and Netconf/YANG mechanisms to control and manage the mesh nodes, as detailed in Table 4-2.

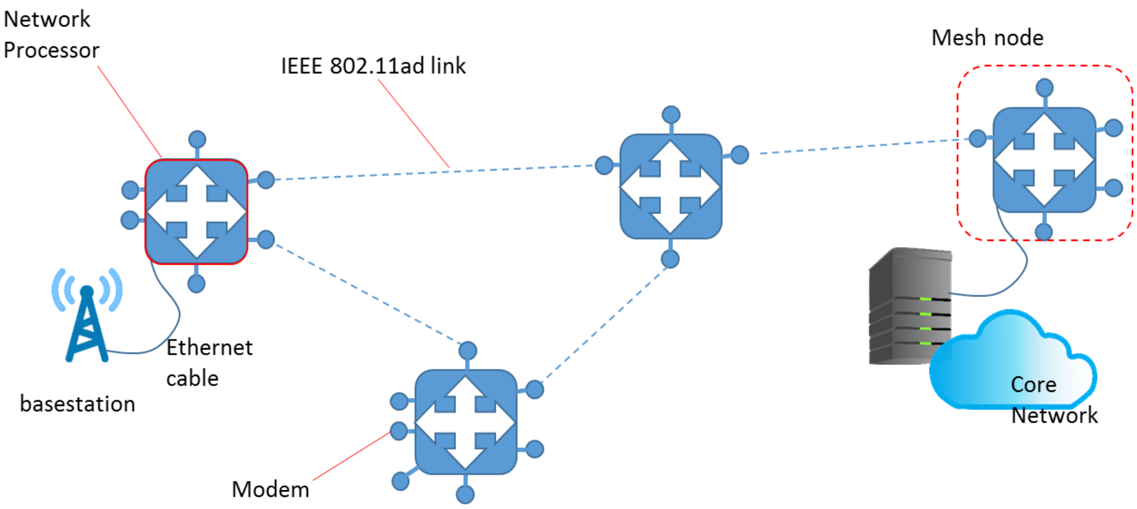


Figure 4-7: Backhaul from a small cell using meshed millimetre wave wireless backhaul.

Table 4-2: South-bound interface functionality for mesh node.

Function	South-bound mechanism
Capabilities	Openflow (alternatively Netconf/YANG)
Performance monitoring	Openflow (alternatively Netconf/YANG)
Wireless topology definition (PBSS)	Netconf/YANG
Forwarding/ switching	OpenFlow (with local failover re-routing)
NPU configuration and management	Netconf/YANG
Modem configuration and management	Netconf/YANG

4.3.1 Software architecture for control and management interfaces

An indicative software architecture supporting both OpenFlow and netconf/YANG interfaces is shown in Figure 4-8. The figure illustrates the four IEEE 802.11ad modems connected to an NPU. This NPU runs two SDN agents, an OpenFlow agent in charge of controlling forwarding within the mmWave mesh, and a NETCONF agent for management and configuration purposes. The NETCONF agent acts as a NETCONF server, and receives connection requests from the SDN controller. The management capabilities of the mmWave device are exposed through a YANG model that we describe next.

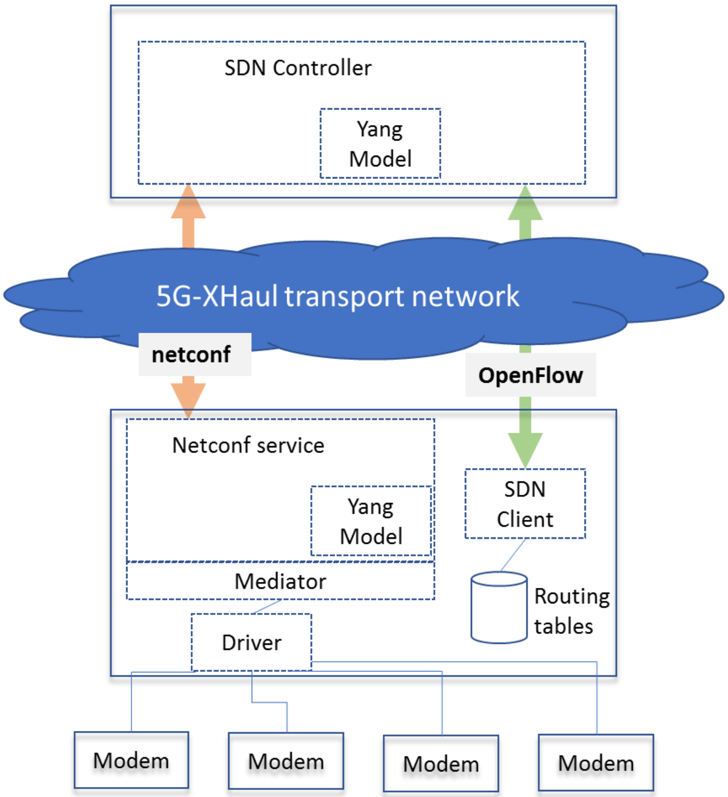


Figure 4-8: Software architecture for control and management of a mesh node.

4.3.2 Example of Netconf/YANG management interface

A YANG model of the management interface to a BWT Lightning module has been developed. The Lightning development module has been used for measurements' collection in WP4 [4][5] and comprises a single modem connected to a host CPU, and therefore represents a subset of the interface which would be required by a complete mesh node. The model is based on the Microwave Model developed by the Open Networking Foundation (ONF), [29], but is simplified to align with the functionality offered by the Lightning API to the driver software (see Table 4-3). It should be noted that the Lightning device would need to be extended with a mediator software layer to permit SDN control via NETCONF. A comparison between the model and the microwave model [29] is shown in Table 4-4.

Table 4-3: Driver API of BWT Lightning module.

Parameter	Description	Legal range	Default	Linux implementation	R/W
Adaptation	Enable or disable link adaptation	N (disabled) Y (enabled) C (clamped)	N	debugfs[1]	RW
Agc	Enables or disables Automatic Gain Control (AGC)	N (disabled) Y (enabled)	N	debugfs	RW
Amsdu	Set/get the maximum number of packets that can	1..32	1	debugfs	RW

[1] Debugfs may be replaced by NL80211_CMD API (Linux standard for WiFi control) with vendor specific extensions as appropriate.

	be aggregated before sending.				
bandwidth	Set/get the channel bandwidth	1 (Full band) 2 (Half band) 3 (Quarter band)	1	debugfs	RW
Beam	Set/get the antenna beam angle	-45 to +45 degrees	0	debugfs	RW
channel	Set/get the radio channel	2 (Fc = 60.48GHz) 3 (Fc = 62.64GHz)	2	iw <dev> set channel	RW
Ldpc_count[2]	Get ldpc iteration count	Number of occurences in the following bins 0,1,2,3,4-7,8-11,12-15,16-31, 31+	-	debugfs	R
Mcs	Set/get the modulation coding scheme	2-9	9	iw <dev> set bitrates	W
mcs_low	Set/get the MCS low water mark (for link adaptation)	2-9	2	iw <dev> set bitrates	W
mcs_high	Set/get the MCS high water mark (for link adaptation)	2-9	9	iw <dev> set bitrates	W
Rssi	Get the RSSI (taken as RCPI - 3dB)	RCPI values are specified in [30]	-		R
rf_gain	Set/get the receiver RF gain	-23.25 to -7.75dB	0x92	debugfs	RW
If/bb_gain	Set/get the receiver baseband gain	0 to 47dB	0x08	debugfs	RW
tx_power	Set/Get the Tx power	0,3,4 or 7dB tx backoff	0dB backoff	iw <dev> set txpower	W
SNR	Get the signal to noise ratio	Values are –13 dB to 50.75 dB in .25 dB steps.	-	debugfs	R
Black: configuration of the modem; Blue: initial config & subsequent monitoring as modem adapts it; Red: monitoring only					

[2] This is not currently fully implemented in the Lightning module

Table 4-4: Comparison of ONF microwave model and Lightning YANG model.

Microwave model feature	Supported in model of Lightning	Notes
Air interface capability	No	
Air interface configuration	Yes	
Air interface status	Yes	
Air interface current problems	No	Alarms of the modem are not currently exposed to the driver
Air interface current performance	Yes	
Air interface historical performance	No	There is no logging capability
Air interface diversity features	No	This feature set is intended for microwave links with multiple radios (for example, using space diversity)
Air interface hsb features	No	Hot standby wireless link – this is common with microwave links because the cost of maintenance of a faulty unit on a high pole is high
Co-channel group	No	This allows multiple air interfaces to be combined, for example for MIMO, cross-polarisation receivers
Container and structure classes	No	These are used to allow a flexible mapping of traffic flows onto the physical resources of the radio interface.

The UML model of the interface is shown in Figure 4-9. The attributes from Table 4-3 are mapped to the air interface configuration, status and performance containers. A YANG definition of this interface can be directly obtained from the UML diagram using tools such as the Javascript tool from the ONF Eagle Project [31].

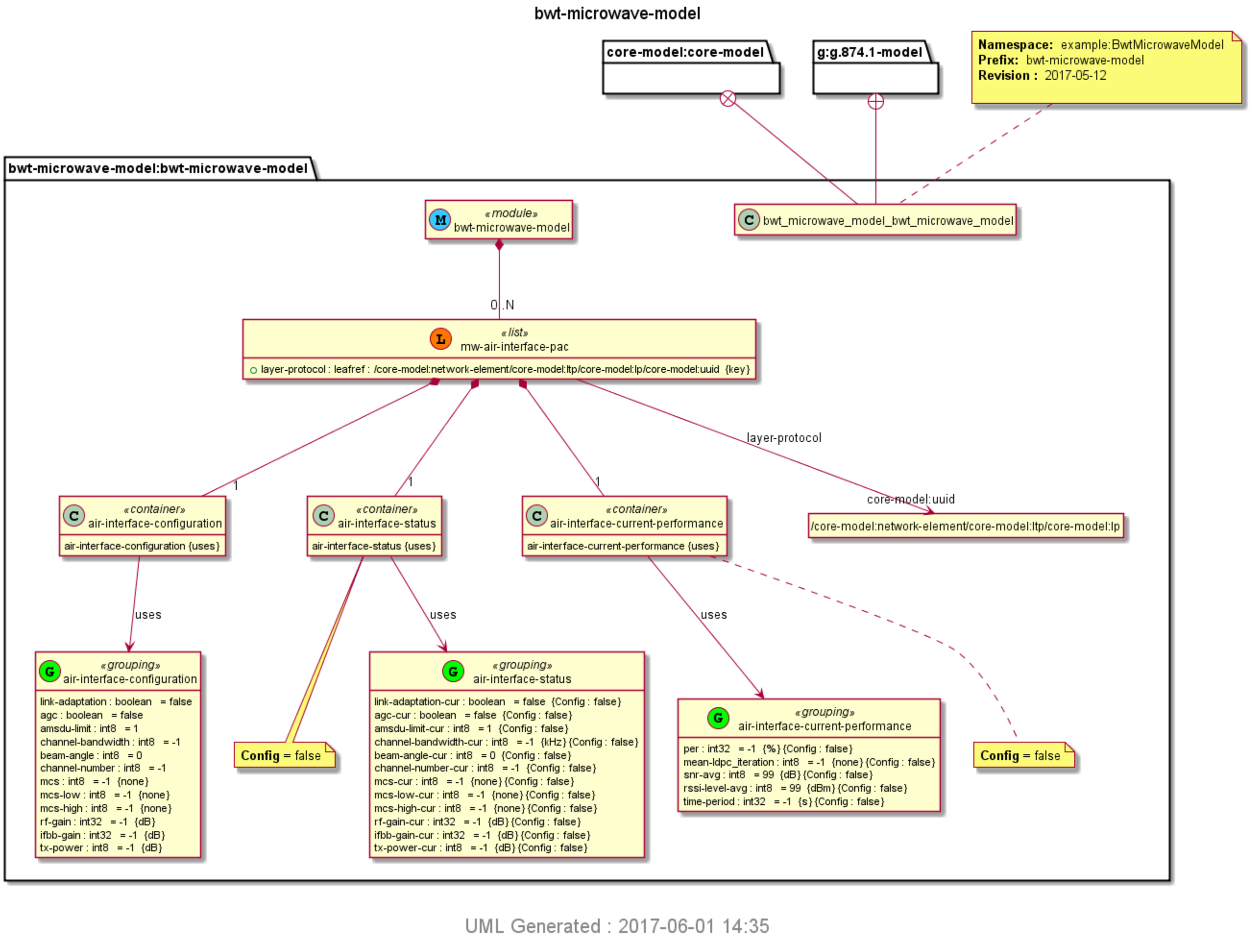

bwt-microwave-model
core-model:core-model
g:g.874.1-model
Namespace: example:BwtMicrowaveModel
Prefix: bwt-microwave-model
Revision : 2017-05-12
bwt-microwave-model:bwt-microwave-model
M «module» bwt-microwave-model
C bwt_microwave_model_bwt_microwave_model
0..N
L «list» mw-air-interface-pac
○ layer-protocol : leafref : /core-model:network-element/core-model:ltp/core-model:lp/core-model:uuid {key}
1
1
1
layer-protocol
core-model:uuid
/core-model:network-element/core-model:ltp/core-model:lp
C «container» air-interface-configuration
air-interface-configuration {uses}
C «container» air-interface-status
air-interface-status {uses}
C «container» air-interface-current-performance
air-interface-current-performance {uses}
uses
uses
uses
G «grouping» air-interface-configuration
link-adaptation : boolean = false
agc : boolean = false
amsdu-limit : int8 = 1
channel-bandwidth : int8 = -1
beam-angle : int8 = 0
channel-number : int8 = -1
mcs : int8 = -1 {none}
mcs-low : int8 = -1 {none}
mcs-high : int8 = -1 {none}
rf-gain : int32 = -1 {dB}
ifbb-gain : int32 = -1 {dB}
tx-power : int8 = -1 {dB}
Config = false
G «grouping» air-interface-status
link-adaptation-cur : boolean = false {Config : false}
agc-cur : boolean = false {Config : false}
amsdu-limit-cur : int8 = 1 {Config : false}
channel-bandwidth-cur : int8 = -1 {kHz} {Config : false}
beam-angle-cur : int8 = 0 {Config : false}
channel-number-cur : int8 = -1 {Config : false}
mcs-cur : int8 = -1 {none} {Config : false}
mcs-low-cur : int8 = -1 {none} {Config : false}
mcs-high-cur : int8 = -1 {none} {Config : false}
rf-gain-cur : int32 = -1 {dB} {Config : false}
ifbb-gain-cur : int32 = -1 {dB} {Config : false}
tx-power-cur : int8 = -1 {dB} {Config : false}
G «grouping» air-interface-current-performance
per : int32 = -1 {%} {Config : false}
mean-ldpc_iteration : int8 = -1 {none} {Config : false}
snr-avg : int8 = 99 {dB} {Config : false}
rssi-level-avg : int8 = 99 {dBm} {Config : false}
time-period : int32 = -1 {s} {Config : false}
Config = false
UML Generated : 2017-06-01 14:35

Figure 4-9: UML model of south-bound interface to a BWT Lightning device (using netconf/YANG).

5 Definition of 5G-XHaul RAN-Transport interface and initial evaluation of related control plane mechanisms

5G RANs are expected to become significantly more heterogeneous and dense, with small and large cells operating at both low and high frequencies, where mobility support, in this context, can get much more complex. Hence, 5G-XHaul proposes a tight integration between the Mobile and the Transport Networks, as one of the key 5G enablers to improve performance. In section 5 of deliverable D3.1 [3], we presented the different interfaces between the transport network and the mobile network proposed by 5G-XHaul. These are represented once more in Figure 5-1.

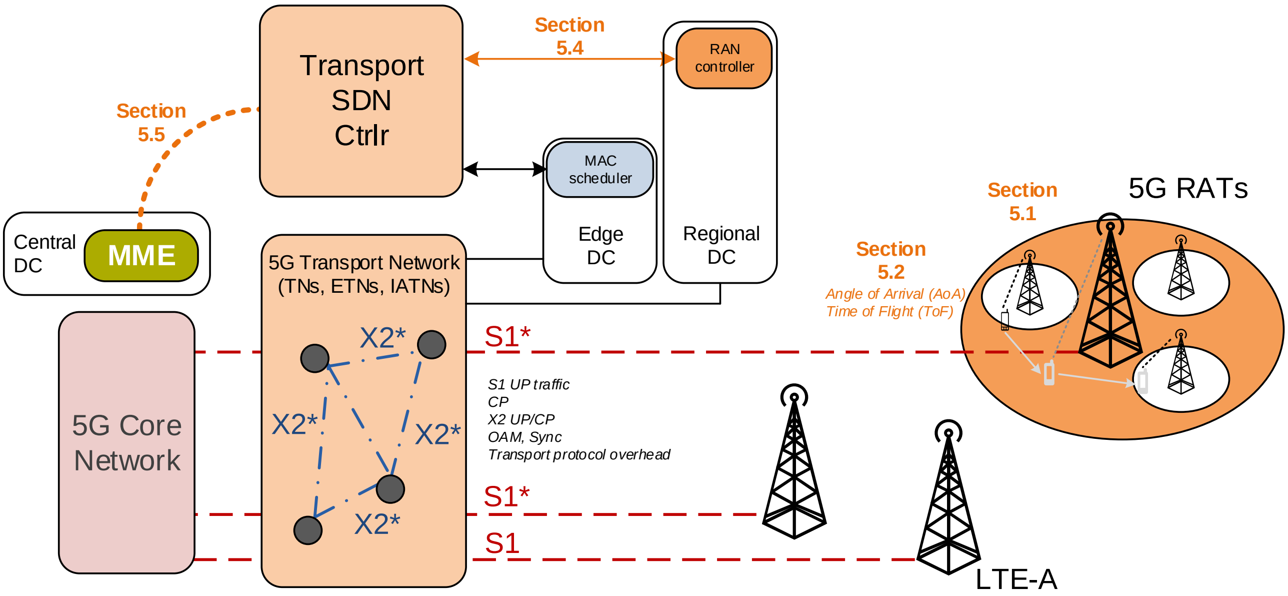


Figure 5-1: Envisioned interfaces between mobile network and transport network.

This chapter represents a more detailed description of the way 5G-XHaul tackles the problem of interfacing the Transport and the Mobile network. In particular, we provide two examples interfaces between the Mobility Management Entity (MME) and the transport network, as well as one example interface between the transport SDN controller and the RAN controller. In Figure 5-1 we highlight these interfaces and we point to the respective sections of this chapter where each interface is discussed.

Another key concept introduced in deliverable D3.1[3] is that the Transport network, i.e. BH and FH, should be optimized according to measurements taken by the RAN (right hand side of Figure 5-1). For this purpose, in this chapter we report a measurement campaign carried out by COSMOTE. The data obtained represent a valuable asset for WP3.

Finally, we envision that future 5G RANs should be able to track moving users, and that this information can also be used to optimize the Transport network. Therefore, this chapter investigates techniques to obtain location of mobile users, using Location-based PHY functionality embedded in RAN nodes.

This section is structured as follows: The measurement campaign, the description of the measurements and a brief introduction on the tools leveraged to obtain the data from the dense urban setting in Greece are described in Section 5.1. In Section 5.2 we describe and evaluate user localization techniques using Location-based PHY functionality. Section 5.3 briefly introduces the methods used to forecast short-term spatio-temporal traffic demand variations based on the measurements presented in section 5.1. An example RAN-Transport interface for the Wi-Fi based RANs is studied in Section 5.4. In Section 5.5 we present two interfaces between Level-0 Area Controller and Mobile Network.

5.1 5G-XHaul Measurement Campaign

In 5G-XHaul deliverable D3.1 [3], we emphasized the importance of the information exchange between the Mobile network and the Transport network.

In Section 7 of deliverable D3.1 [3] we identified a set of candidate parameters to be collected and processed by the 5G-XHaul transport controller. It is of the interest of 5G-XHaul to extract these parameters and to base our subsequent analysis on real data from a dense urban scenario at different scales using the abovementioned interfaces. Moreover, the development of spatio-temporal demand prediction models would require the utilization of the offered traffic and mobility-related information for a certain geographical area over time.

A dense urban area in the center of Athens, depicted in Figure 5-2, was selected for the measurement campaign to extract the different parameters. This area is characterized by high traffic load and high mobility conditions, and is served by 17 3G/HSPA/HSPA+ Node-BSs and 10 4G/LTE eNBs (see Table 7-4 (Annex 7.3) for more detail on the different BSs).

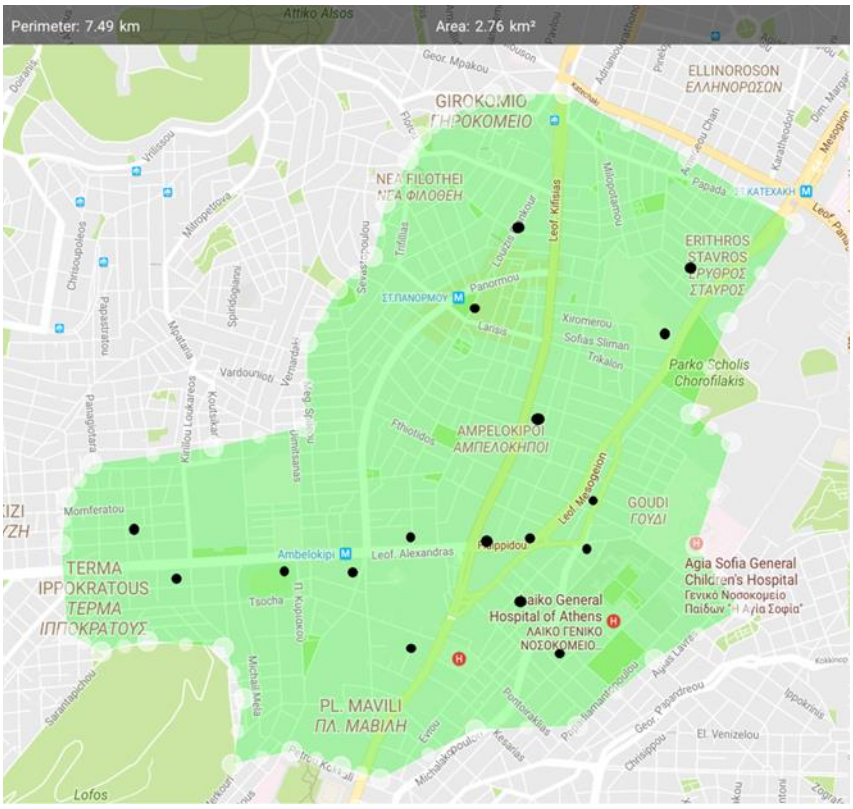


Figure 5-2: The geographical area under study in Athens city centre and BSs location (black circles).

A vehicle (mobile terminal) was used to collect the measurements. The measurements were collected via custom made Android applications (see Figure 5-3) that were co-developed by COSMOTE and UTH[3] in the context of the EU funded project FLEX (http://www.flex-project.eu/). More information about the tools' capabilities can be found in Annexes 7.1 and 7.2. These resulting measurements were then uploaded to a dedicated database, which has been made available to the 5G-XHaul partners for them to use it as input to the developed algorithms.

The retrieved information includes:

1. Traffic statistics at BS level for all Radio Access Technologies (RATs) supported and for all cells/sectors. For example, UL/DL Data traffic (MB), max/average UL/DL throughput per user (Mbps), traffic distribution per data rate, average/total number of UEs per eNB, number of simultaneous UEs.
2. Mobility-related events (e.g. handover, cell reselections), preferably combined with location-related information.
3. Network performance and QoE-related information (e.g. max UL/DL bitrate, latency) combined with location-related information.
4. Other (network) information, such as RAT supported per BS, number of cells/sectors per BS and per RAT, location info per BS, RRC Drop ratio, Drop Call Rate for voice, video and data sessions, RRC

[3] Indicative videos presenting the field measurements' collection process: https://drive.google.com/file/d/0B9U5IjOBumnBbmYtcHRDWC1nQWM/view?usp=sharing https://drive.google.com/file/d/0B9U5IjOBumnBa3VYclV3ZThRWFE/view?usp=sharing

Setup Failure Rate, E-RAB Failure Rate, Intra eNB Handovers Failure Rate, Inter eNB Handovers over X2/S1 interface, number of CS Fallback attempts in idle and connected mode, average CQI.

In the context of (1) and (4), COSMOTE made available statistics at BS level on a quarterly/hourly basis for one month – 2 x 14-days time-windows. For (2) and (3), COSMOTE provided field measurements at mobile terminal level for the same geographical area including signal strength, maximum download bitrate, maximum upload bitrate and latency; along with network-related info (*cellid*, *LAC*/*TAC*, *RAT*, *OPid*, *BSname*), timestamp, location info (latitude and longitude coordinates), etc. (see Table 5-1). The measurements collected by the mobile terminals are uploaded to a dedicated database (residing at a remote server) and can be depicted via a graphical user interface (WebGUI) shown in Figure 5-4 extracted into various formats (xml, csv, txt, etc.).

The methodology followed for the collection of the field measurements comprises the following steps:

1. Identification of the boundaries of the area covered by the 17 Base Stations.
2. Identification of routes for the drive tests.
3. Installation of the FLEXtools Applications to four mobile devices (one for each measurement type).
4. Conduction of drive tests in different days of the week, different hours of day (see Figure 5-4).
5. Depiction of the measurements through the WebGUIs for verification purposes.
6. Extraction of the stored measurements (in the database) to suitable file formats.
7. Uploading of the measurements to the BSCW server[4] for further use by the relevant partners.

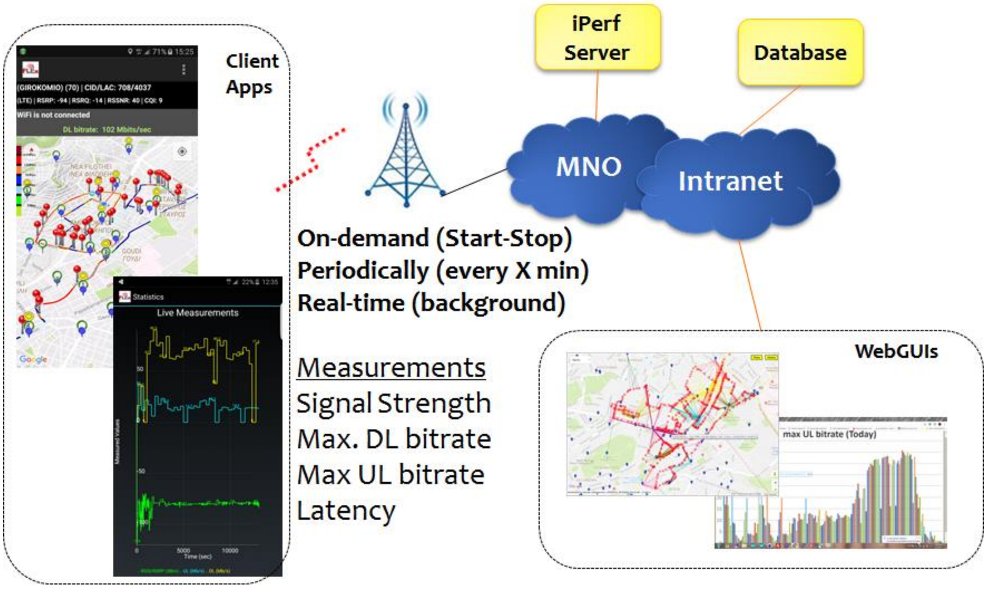


Figure 5-3: Measurements' collection tools (Android Apps, servers/databases and GUI).

Table 5-1: List of measurements collected.

DB info	Signal Strength	Max DL Bitrate	Max UL Bitrate	Latency
Timestamp	✓	✓	✓	✓
Lat, Lon	✓	✓	✓	✓
RSSI	✓	✓	✓	✓
RSRP	✓	✓	✓	✓
RSRQ	✓	✓	✓	✓
RSSNR	✓	✓	✓	✓
CQI	✓	✓	✓	✓
Cellid	✓	✓	✓	✓

[4] Important Note: The uploaded information in the BSCW shall be strictly utilized in the context of the 5G-XHaul project and since it contains sensitive network info it shall be treated as confidential info.

LAC/TAC	✓	✓	✓	✓
RAT	✓	✓	✓	✓
MCC/MNC	✓	✓	✓	✓
BSname	✓	✓	✓	✓
Max DL rate		✓		
Max UL rate			✓	
packetsize		✓	✓	
MT Brand/model	✓	✓	✓	✓

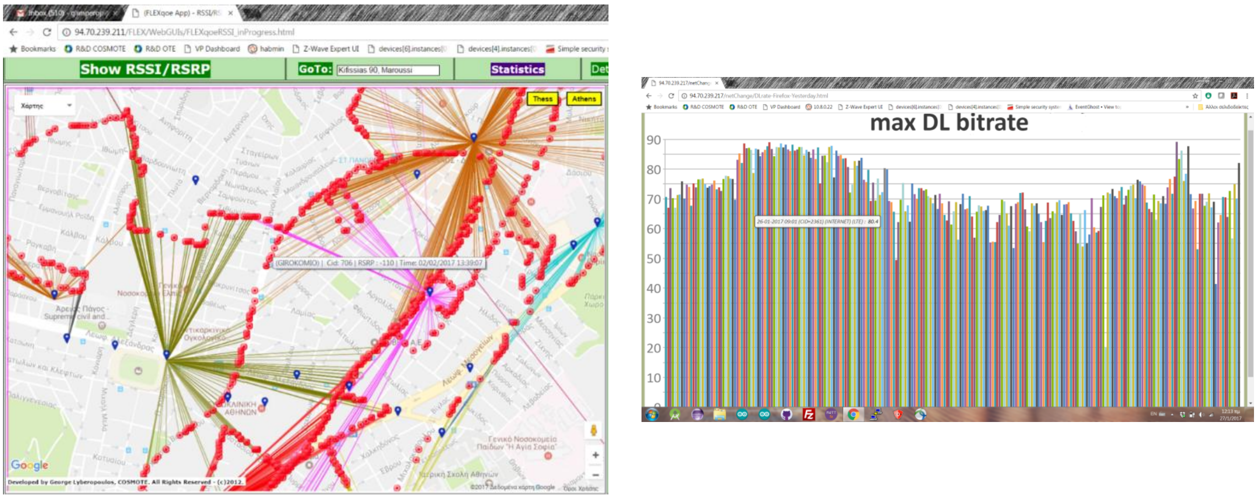


Figure 5-4: Measurements' depiction via the WebGUI.

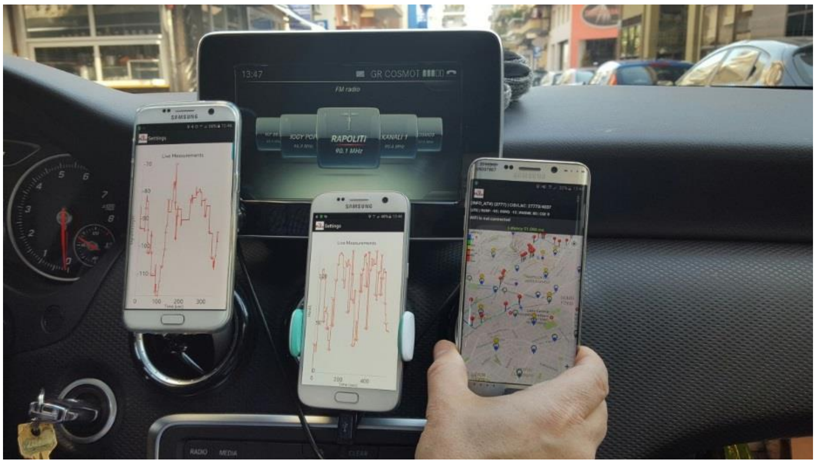


Figure 5-5: Field measurements (at terminal-level) during drive tests.

Based on the measurements obtained, the following network details could be identified:

- The coverage area / BS (on idle, while on session).
- Coverage issues (low RSSI/RSRP).
- Network performance issues (low DL/UL throughput, high latency).
- Mobility issues (high #cell reselections, high #handovers), along with location info.
- Network design Issues (unexpected coverage from a distant BS, unsuccessful handovers, etc.).

- User mobility “patterns” incl. (a) user speed, direction, etc. based on timestamp and location info), (b) average duration within cell (vs. daytime vs. direction), (c) candidate destination cell, etc.
- Offered traffic patterns based on mobility “patterns” and user activity (off-hook, active data session, etc.).

5.2 Methods for increasing the precision of users’ location in the RAN: trilateration

In 5G-XHaul deliverable D3.1 [3] we presented the evolution of the RAN architecture towards a cell-less one, where the network itself detects that a User Equipment (UE) is moving by scanning the signals transmitted by this UE. In dense urban scenarios, the number of UEs is high and they are in constant movement, thus if the position of these UEs must be estimated frequently, the associated signaling in the wireless medium can be significant. This motivates the design and implementation of location methods to estimate the movement of the users in the network, to assist handover or control plane decisions based on the context of the network at a certain point in time.

In 5G-XHaul we develop methods, which can be easily implemented in the RAN nodes, to detect or to track the movements of the UEs. This is depicted in the right hand side of Figure 5-1, and it represents a way of gathering real time information, aiming at a reduction of the signaling load. The position of the UEs can be discovered via Location-based PHY functionality embedded in RAN nodes.

Methods to estimate the Angle of Arrival (AoA) or the Time of Flight (ToF) deliver very accurate position and direction estimates. This approach enables precise spatio-temporal modelling, which requires a tighter integration between the mobile network and the 5G-XHaul transport network. These methods have been initially presented in deliverable D4.13 [6], where they have been described and assessed in static scenarios.

In this section these methods are extended and tailored to track the movement of UEs in the RAN, and can be implemented in both Sub-6 and mmWave nodes. Sub-6 technologies are expected to provide a coarse position and/or direction estimate of the UE, to feed (if required) a finer estimation of the position of the user by performing location estimation in mmWave. In the context of this deliverable we extend the methods presented in [6] to the RAN context, and we provide implementation details and initial results. These methods and evaluation results will be further extended for deliverable D3.3.

5.2.1 Estimation of the UE position using trilateration

In 5G-XHaul deliverable D4.13 we propose a method [33] that can be used for joint localisation of a large number of mobile stations (UEs), without increasing the number of needed transmissions [33][34]. This method can be easily implemented in small cells and can be used for location estimation of UEs. It is able to localise the nodes using trilateration, and requires a different number of anchor nodes, i.e. referent nodes with known positions which transmit frames are used for localisation purposes which could be considered Access Nodes (ANs) in the 5G-XHaul nomenclature – depending on whether 2D position estimation or 3D position estimation is desired. As an example, Figure 5-6 presents the general scenario where trilateration is performed using time-of-arrival (ToA) estimation at the mobile node (one is used for simplification).

The transmissions of the anchor nodes can be used for position estimation by unlimited number of UEs. Nevertheless, all of the mobile stations must receive the same broadcast (beacon) frames.

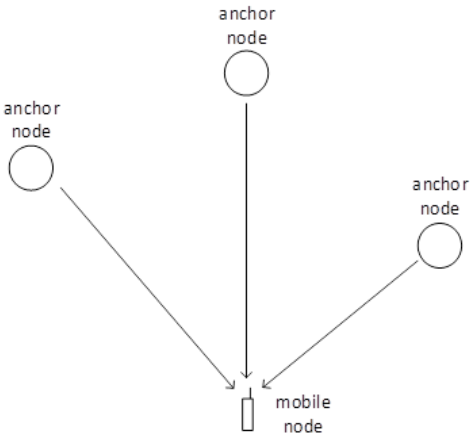


Figure 5-6: Estimation of the mobile node (UE) location using anchor nodes.

In this method, the mobile node (UE) can transmit its position to the network but, in some scenarios, it is also important for the network to be able to estimate the position of the mobile station. The extension of the method, also described in [33], involves bi-directional communication between anchor and UEs. The position of the UE is in this case estimated again employing trilateration by using Two Way Ranging (TWR) for estimating the distances between UE and the anchor nodes. The use of TWR relaxes the synchronisation requirements between the anchor nodes while the position of the mobile station can be still estimated precisely.

This method can in theory be implemented in both Sub-6 and mmWave nodes. Sub-6 nodes benefit the implementation of the method given the wide beam widths which allow multiple stations to receive the same transmitted signals. In mmWave bands, however, the beam widths are narrower, and this method is hardly applicable to multiple stations. The co-location of Sub-6 and mmWave nodes could favour the implementation of the method and, in turn, the achievable precision – assuming that the UE has multi-connectivity. According to [32], UEs are assumed to have dual RF interfaces: a legacy network link and a mmWave link, respectively.

5.2.2 Implementation of the localisation method in Sub-6 nodes

In this section we present the measurement campaign carried out to assess in a real scenario the feasibility of the positioning method presented in [33]. The setup involves two software defined radio (SDR) platforms (Ettus X310) – Sub-6 nodes – each of them consisting of two separate wireless transceivers and equipped with two omnidirectional antennas. Figure 5-7 shows a sketch of the measurement setup.

With a total of two SDRs, four independent transceivers are available. For the ToA approach, in a 2D space, a minimum of 4 nodes are required. If additional constraints are introduced for the area in which the mobile nodes can reside, the total number of anchor nodes can be reduced. Additional anchor nodes would increase the positioning precision.

We synchronize the anchor nodes using a common 10 MHz signal for clock synchronisation and 1 pulse-per-second (PPS) signal for timing synchronisation. The anchor nodes are controlled by a single host computer, PC1 in Figure 5-7. The host computer generates beacon frames which are transmitted from the radios periodically, each 10 ms from each of the four anchor nodes. The beacon frame itself consists of a preamble, a header and ranging sequence. The preamble is used for detection of an arriving frame. The header contains data about the anchor node, e.g. position of the anchor node, sequence number, etc. A CRC32 is inserted at the end of the header for error checking purposes. Appended to each beacon frame, a pseudo-noise (PN) sequence is included. This PN sequence is 1023 symbols long and is generated using the generating polynomial $x^{10}+x^3+x^2+1$. The beacon frame is modulated using BPSK modulation. The frame format is shown in Figure 5-8.

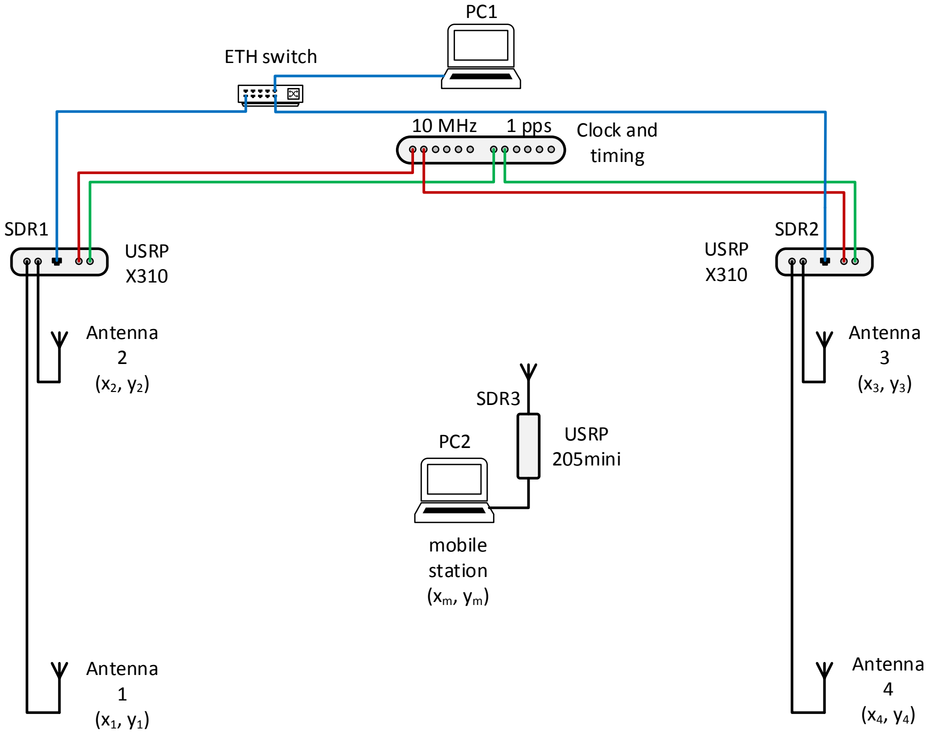


Figure 5-7: Sketch of the measurement setup.

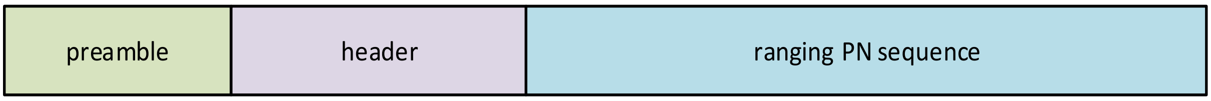


Figure 5-8: Beacon frame format.

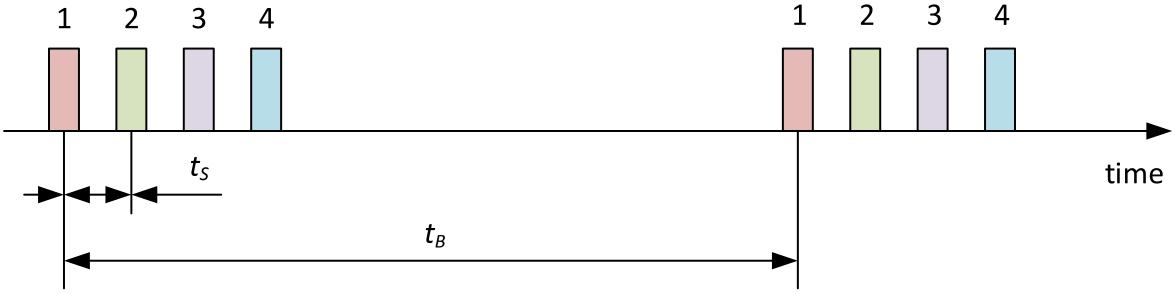


Figure 5-9: Scheduling of the transmissions between the anchor nodes.

The schedule for transmitting frames from the anchor nodes is given in Figure 5-9. The beacon frames are transmitted sequentially, not to interfere with each other, since all of the radios transmit on the same frequency. In a small cell scenario with multiple anchor nodes, different frequencies shall be used. However, this method itself is applicable in such scenario. Each of the anchor nodes (1, 2, 3 and 4) transmit sequentially, in time slots, each having a maximum duration of t_S = 400 us. The frames are 65 us long and, therefore, they do not occupy the complete slots. All transmissions are performed in the 5 GHz ISM band at a frequency of 5.75 GHz.

The mobile node, shown in the center of Figure 5-7, consists of a host computer and a SDR (Ettus B205mini) connected to the USB 3.0 port. This radio receives for a period of at least 11.6 ms to ensure that one group of beacon frames is available within the acquired samples. These samples are further processed to find the preambles and to extract the frames. The time of the reception of each extracted beacon frame also estimated. The data from the headers is extracted and CRC check is performed.

To obtain a precise ToA for each of the received beacon frames, we perform cross-correlation between the received beacon frames, containing a PN sequence, and a locally generated copy of the same PN sequence. Using the obtained cross-correlation peaks, the time of arrival of the received beacon frames can be estimated. Since precise estimation of the ToA is needed for precise distance estimation, the cross-correlation peaks are interpolated using quadratic interpolation.

Once the time of arrival of all the beacon frames from the anchor nodes is estimated, the position of the mobile node can be also estimated. To estimate the position of a mobile node using the obtained ToAs, a system of nonlinear equations is solved. Given that real measurements are performed, the obtained ToAs would not produce a single solution for the position of the mobile node. Therefore, a least squares approach for solving a system of nonlinear equations is usually used. In our tests we use numerical solution of the equations using the Levenberg-Marquardt method.

5.2.3 Scenario description and system setup

As shown in Figure 5-7, two SDRs, acting as anchor nodes with two transceivers each, were used for the tests. This means that a total of four transceivers, i.e. a total of four antennas, are used as anchor nodes. These nodes are positioned in a large office area having dimensions of 20 x 8 metres. The nodes' positions are shown in Figure 5-10. These nodes are connected to a single host computer. The red line represents the path the mobile node was moving during the tests.

The anchor nodes are equipped with omnidirectional antennas, having gain of 4.6 dBi in the 5 GHz band. The antennas are raised on a height of approx. 2 metres. The bandwidth is 25 MHz and the sample rate of the SDR is 50 MSps.

Two types of tests were performed: static and mobile. For both scenarios the mobile node is placed on a cart and its antenna is positioned at a height of approximately 2 metres, i.e. in the same plane with the anchor nodes. For the static tests, the mobile node was placed in a known position and samples of approximately 200 different position estimates in a time window of 2 seconds were acquired. The short time window is chosen to prevent from variations of the received signal caused by the changing environment, i.e. fading due to moving of people.

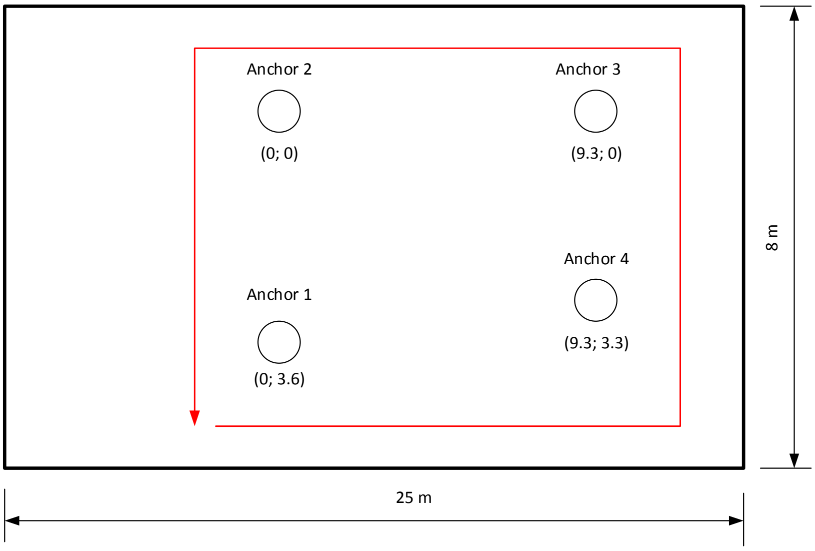


Figure 5-10: Area for testing the proposed localization method.

Table 5-2: Coordinate of the measurement positions for the static scenario.

Position number	X coordinate [m]	Y coordinate [m]
1	5.49	5.28
2	11.70	4.91
3	11.54	-0.88
4	-1.09	-1.30
5	-1.09	5.47
6	4.74	1.79
7	4.37	-0.99

The mobile scenario, on the other hand, is used to estimate the performance of the proposed localisation method in a dynamic scenario. In this scenario, the mobile node was moved in a rectangular trajectory, using the path shown with the red line. The obtained positions are saved and compared afterwards to the real path at which the UE was moved.

For the mobile scenario, the processing is performed in real-time and a total of 3 to 5 measurements per seconds are obtained. In the mobile case, additional filtering of the position is performed to remove the noise in the position estimates. Using a Kalman filter is a common choice, in this case we only perform moving average filtering of the obtained positions as well as limiting the mobile node maximal speed.

Before the developed test system is used for positioning, it must be first calibrated. Calibration is needed since different group delays are expected in both transmit and receive chains, as well as in the cables used for connecting the antennas to the transceivers.

5.2.4 Measurement results for the method and extension

We present now the results from the obtained measurements. For the static case, the measurements are obtained at 7 different physical positions. The positions' coordinates are listed in Table 5-2.

The position of the mobile node is estimated using the time of arrival of the beacon frames from the four available anchor nodes. Actually, the difference of the ToA of the beacon frames, at the receiver is used to estimate the positions. Depending on the position of the mobile node, with respect to the anchor nodes, precision and accuracy of the position estimate can vary. This is because the position of the anchor nodes is not optimal for every position of the mobile node. The time between the arrivals of the beacon frames from the different anchor nodes depends on the differences between the distances between the mobile node and the anchor nodes. Therefore, to quantify the positioning error it results better to quantify the time difference of arrival of the beacon frames at the mobile node. This difference of ToA can be also expressed as a difference in distance, and provides a better insight of the achievable precision when using the proposed method. Therefore, we use the difference of distances between the mobile node and the anchor nodes for estimating the precision of the developed system.

For each position, the difference of the time of arrival of the beacon frame from the first anchor node and each of the remaining three anchor nodes is estimated. To evaluate the precision of the estimation of the ToA difference, the cumulative distribution function (CDF) of these differences, around their mean value is calculated. Further, these CDFs are multiplied by the speed of light in order to obtain the values in metres instead in nanoseconds. These CDFs are plotted in Figure 5-11. For each position, three differences of the ToA are estimated (first and second, first and third and first and fourth anchor node). Since measurements for 7 positions were performed, a total of 7x3 = 21 CDFs are obtained. As can be seen in Figure 5-11, most of the ToA difference estimation errors, expressed in distance, would be below ±1 metres. Only in few cases the error exceedes up to ±2 metres, being these cases where the received power was significantly low.

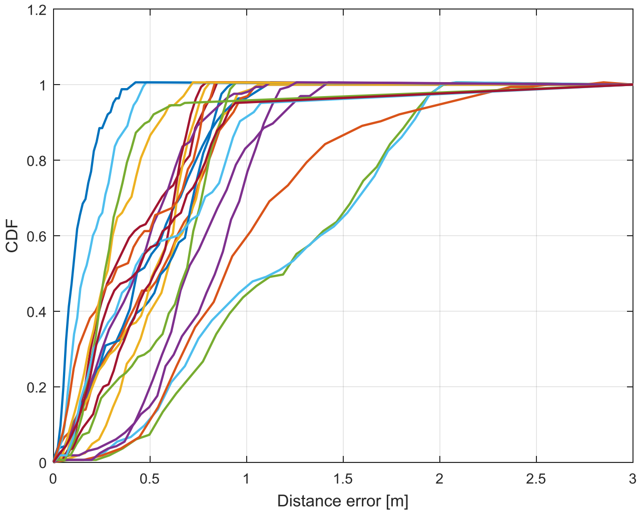


Figure 5-11: CDF of the difference of ToA of the received beacon frames between the first and second, third and fourth anchor node.

For the mobile case we compare the real path with the obtained i.e. estimated path. In Figure 5-12, a comparison of the ideal path, i.e. the path followed by the mobile node, and the estimated path is shown. This path was travelled by the mobile node for a total of three times in order to evaluate the repeatability of the results. As can be noticed, the estimated paths follow the ideal path. The positioning error in some cases is around 1 meter or slightly more. The positioning errors are mainly due to multipath in the indoor environment. Since these are the initial tests we have performed, the positioning error due to multipath is not addressed and, therefore, not compensated. It can be noticed in Figure 5-12, that there exists a larger positioning error in the corners of the ideal path. This is due to the use of moving average filter on the estimated positions.

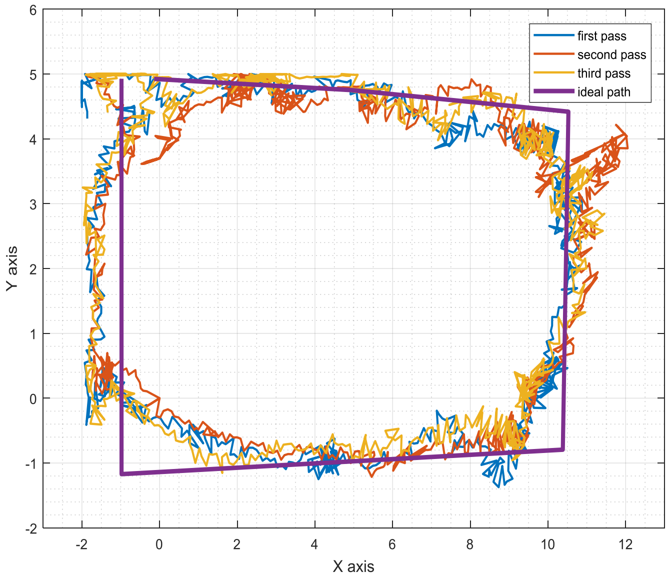


Figure 5-12: Real (ideal) path compared to the estimated paths. A total of three estimated paths are present.

5.3 Exploitation of RAN measurements in the transport: Demand prediction models

In 5G-XHaul, we foresee that the transport network can benefit from as much knowledge as we can obtain from the RAN. As stated in deliverable D3.1 [3], we want to provide the 5G-XHaul control plane with cognitive capabilities, namely the ability to measure the current network state and, based on past experiences, to forecast short-term spatio-temporal traffic demand variations. The development of predictive models for cellular network is key to adjust the network resources accordingly. The generation of spatio-temporal demand models though, requires the control plane to be able to predict the movements of the mobile users (UEs). This is possible when enough data is available from different base stations.

For this deliverable, in Section 5.1, COSMOTE has gathered datasets from their measurement campaign in the city of Athens. The measurements come in the form of a list of candidate parameters which could be collected and processed by the 5G-XHaul transport controller. The datasets include those collected on the UE's side via the applications included in Section 5.1, as well as those obtained on the cellular network operator's side. In this subsection we will mainly make use of the latter given the richer information they provide. The dataset spans two weeks at per hour granularity. The records in the dataset are indexed by a time stamp.

The goal of this section is to provide some insight on the characteristics of the traffic in a dense urban scenario, to analyze the different patterns present in the datasets, as well as to indicate how this information can be useful for the the controller to undertake the adequate measures and actions in response to the current RAN context. Understanding the temporal and spatial patterns of the traffic can be useful to estimate both short and long term changes in the network. These studies will be further extended in deliverable D3.3.

5.3.1 Temporal dynamics

We first start analysing the two-week period dataset with measurements from different base stations. It is first interesting to study the temporal dynamics of the aggregated traffic. The temporal dynamics of the data traffic are represented as a random process V, being V = <V_1, V_2, …, V_i> the traffic volume at time index i. In this case we analyse the data traffic at hourly time resolution. We can also aggregate consecutive traffic in V as a single element, which leads us to increase the dimensionality of the distribution. For our studies we will aggregate the traffic volume.

Figure 5-13 depicts the time series of the normalized aggregated traffic volume in MB for one of the two weeks (for the sake of representation). We appreciate from the figure the clear diurnal variations and the existence of a peak every day. The peak starts around noon and lasts up to early evening. This analysis of the temporal evolution reveals a repetitive daily binary-like behaviour represented by low traffic at night hours and a high traffic during day hours.

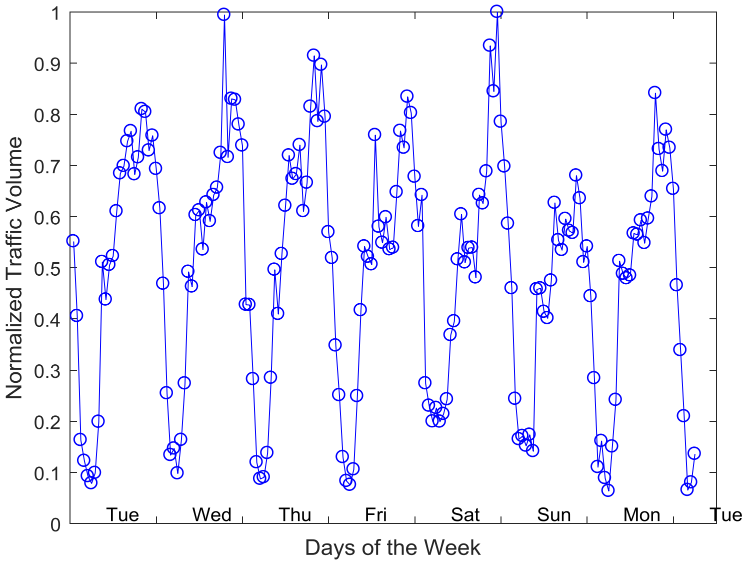


Figure 5-13: Time-series of the observed traffic (aggregate traffic) at per hour granularity (1-week period).

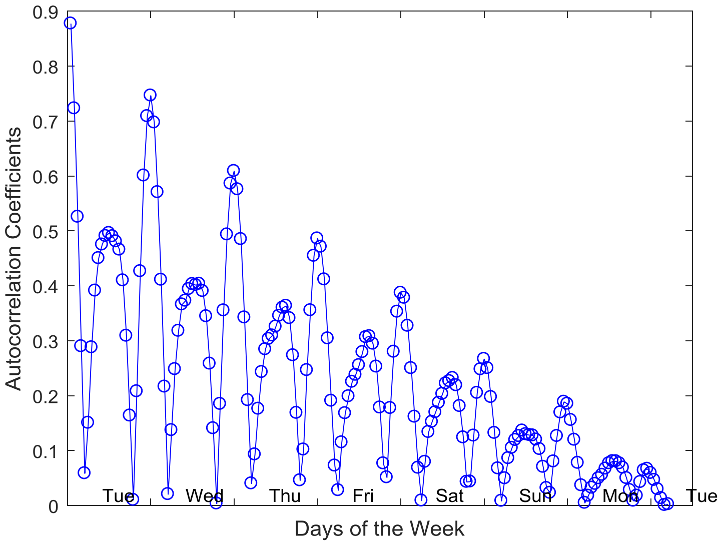


Figure 5-14: Autocorrelation function as analysis of temporal dependency (1-week period).

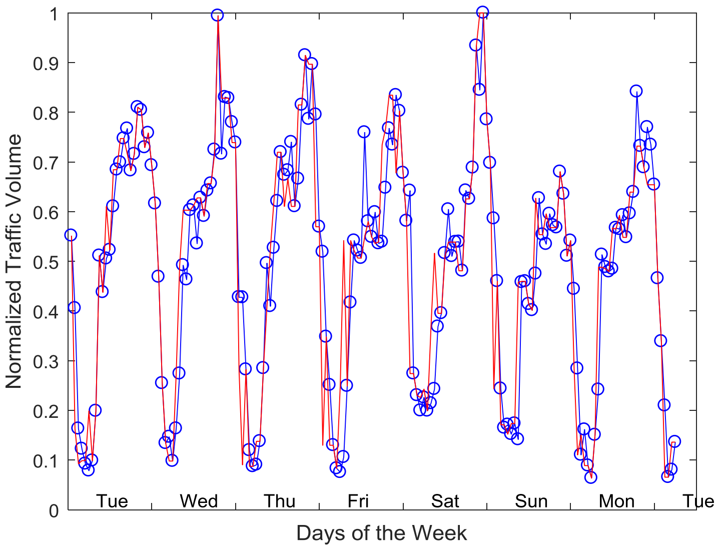


Figure 5-15: Forecast of the traffic based on Markov model (real trace – blue, prediction – red).

One of the relevant properties which can provide insight to analyse the statistical characteristics of the traffic is the autocorrelation. In Figure 5-14 we depict the autocorrelation function of the aggregated traffic. As can be seen, the dependency of traffic volume decays as the time lag decreases. The traffic volume at a given time, indicates a dependency of the previous states in multiples of 24 hours. This is noticeable as the correlation peaks are located in multiples of 24, i.e. 48, 72, etc.

5.3.2 Forecasting traffic

We presented in the previous subsection the temporal dynamics of the aggregated traffic, and we extracted the most relevant information. The results from this previous subsection show the dependency of the traffic on the previous 24 hours. Considering the temporal dynamics as a random process, we can state from these results that the order of this process is *n*=24. In this section, to forecast the traffic, we have made use of an *n*th order Markov chain with *q* possible values or states. For the case under study we consider the previous 24 lags. The objective is to predict the next state given the previous *n* states. The results in Figure 5-15 show that the model captures accurately the dynamics of the traffic volume with a mean squared error (MSE) of $6x10^{-3}$.

5.3.3 Mobility analysis

The goal in this section is to understand and model how the UE travels at different spatial and temporal scales, which can be done at both individual/aggregated levels. While studying the temporal evolution of the aggregate traffic can uncover important characteristics of the consumption patterns, it does not allow to understand how usages vary over different areas, a very useful analysis for networking studies. We have analysed the dataset corresponding to different base stations in order to identify patterns on the average number of users attached to them. Figure 5-16 depicts the evolution of the number of users with the time over a week-time period (at per hour granularity. As in the previous case (traffic volume), we can conclude that the behaviour is similar and can be easily predicted. We can here easily appreciate that, for the weekend period (Saturday and Sunday), the slope of the curve in the morning does not grow as steep as for that of the week days.

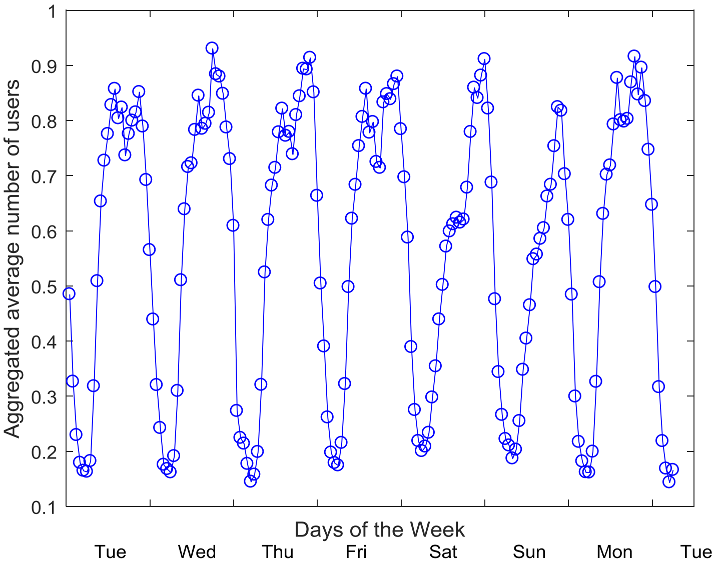
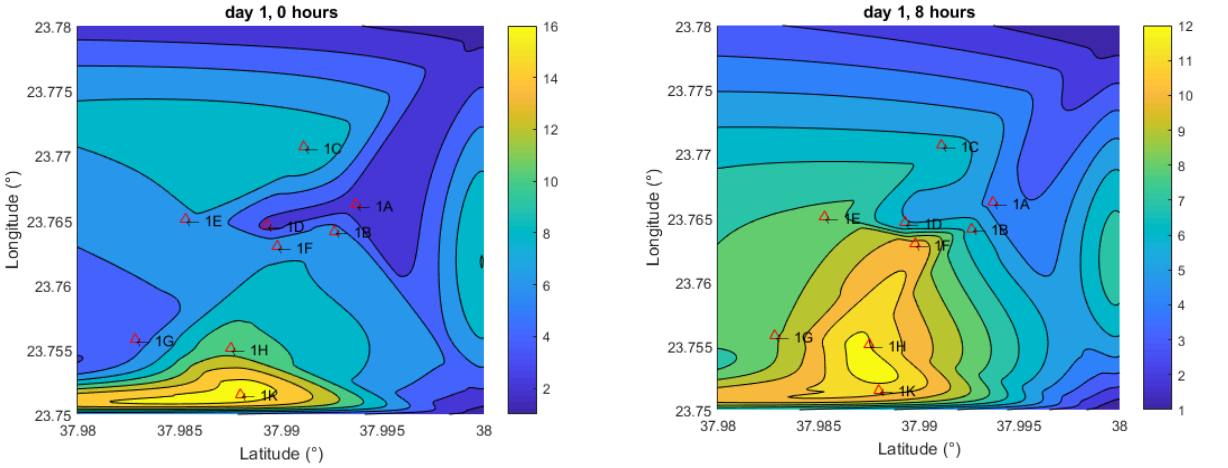


Figure 5-16: Time-series of the observed average number of users at per hour granularity (1-week period).

Figure 5-17 represents the evolution of the average number of users in time for a given area of Athens consisting of eight base stations. The time spans from Tuesday at midnight to the midnight of Wednesday, and it is evolution is captured in the caption of each of the figures. On the X and Y axis we have latitude and longitude respectively. We can observe that, for the same period of the day, the map shows clearly a similar trend both in activity and in density of users. If we give it a closer look to the density of the users, we realize that from 8:00 hours to 12:00 hours there exists a high density of users in the centre-low part of the graphs. This area is where many hospitals and health centres are located, which explains this larger density of users. The mobility models to be developed in D3.3 will analyse the behaviour of the network statistics with a larger measurement time, also including the logs obtained at UE level.



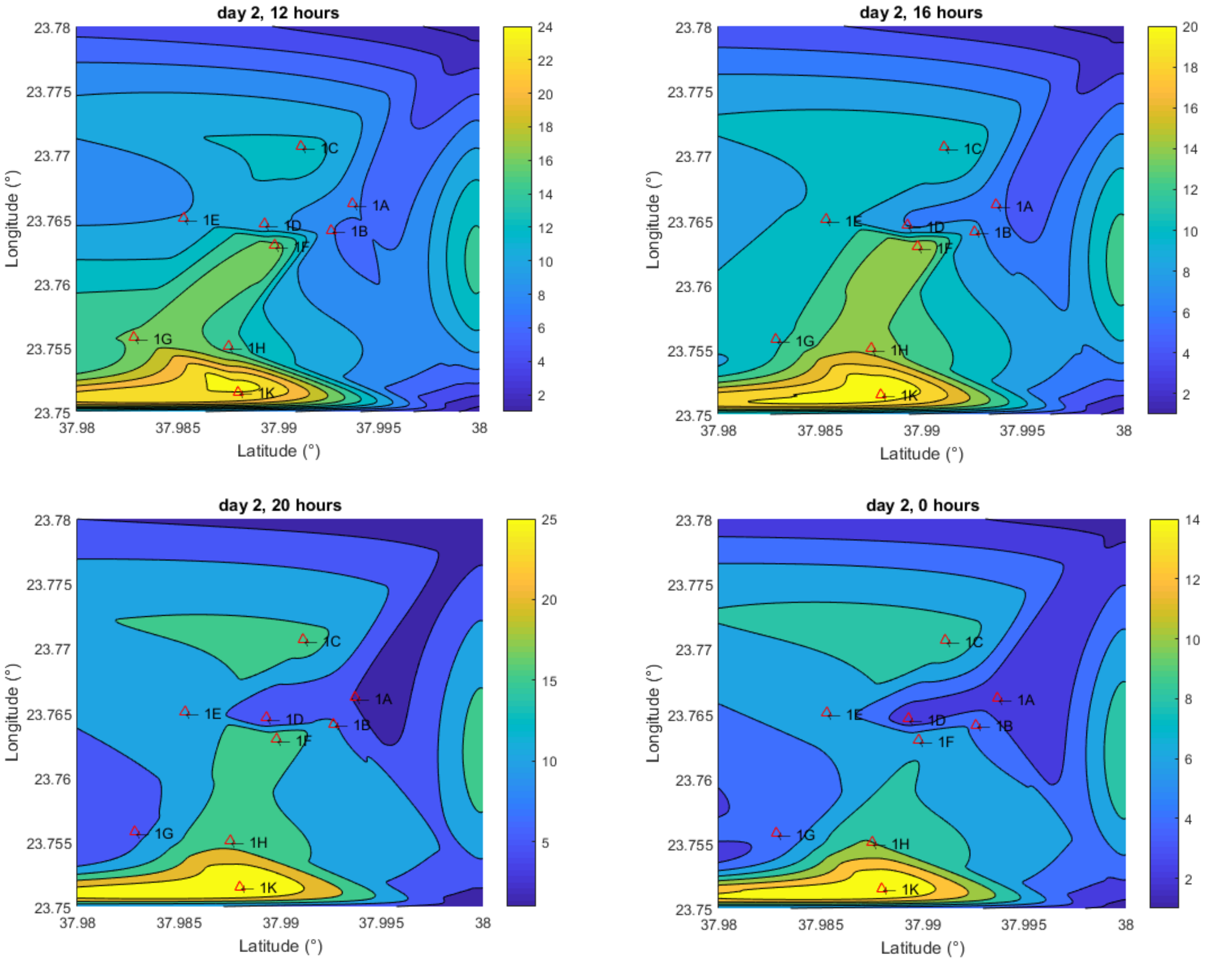


Figure 5-17: Colormap representing the evolution of the average number of users in a certain area of Athens.

5.4 An example RAN – Transport interface in the Sub-6 GHz domain

In [2] we justify the use of Sub-6 GHz RATs within the 5G-XHaul wireless backhaul (BH) architecture. In [7], we give details of the internal architecture of the Sub-6 GHz TNs envisioned in 5G-XHaul, and argue how that architecture addresses the challenges of integrating wireless interfaces into an SDN-based control plane. Recall that, for convenience, we focus our study of the Sub-6 GHz wireless BH on IEEE 802.11-based technologies (i.e. Wi-Fi) as its prominent position as NLoS RAT for data communications in small cells. Also anticipated in [7], is the fact that Sub-6 GHz BH TNs will probably compete for the same (scarce) radio resources used by the RAN. In the case of Wi-Fi-based Sub-6 GHz, the random medium access, intended to provide long-term fairness among all the contenders, will be detrimental to backhaul nodes in front of access nodes; all transmitters will get the same transmission opportunities although TNs may have to bear more traffic (aggregated flows from multiple access links). This RAN – transport competition would require a cooperative management involving both segments of the network. In this section, we propose an interface between the RAN control and the backhaul control and devise different mechanisms that try to take advantage of that interface. As a proof of concept, we implement the proposed architecture in a small testbed.

5.4.1 Sub-6 GHz joint RAN-backhaul architecture

As part of the 5G-XHaul architecture [2], the joint access/backhaul mesh for Sub-6 GHz builds on the wireless TNs defined in Figure 5-18, extended to host additional radio interfaces devoted to serve user devices as part of the access network. Figure 5-18 exemplifies the proposed joint control.

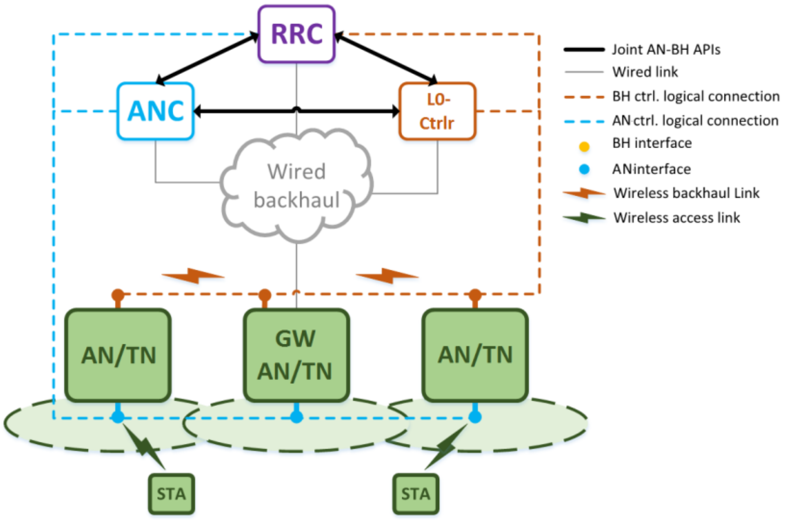


Figure 5-18: Joint Sub-6 GHz access/backhaul architecture.

The architecture proposed consists of hybrid BH TNs plus access nodes (AN) interconnected to form a wireless mesh, which provides multi-hop paths to/from terminal stations (STAs) to a wired core infrastructure. Note that, due to the nature of outdoor small cell deployments, wired connectivity is only reached through a limited set of nodes (namely, an IATN, cf. [2]), which work as gateways (GW). In Figure 5-18, we can see how unified “access/backhaul” network elements are composed of hardware boxes containing multiple Wi-Fi interfaces, each one being used either for the access or backhaul network. Access interfaces are controlled by a logically centralized access controller (ANC), and backhaul bound interfaces are controlled by a Level-0 Area Controller (L0-Ctrlr), as defined in [2]. In addition, a logical interface between the access and the backhaul controllers is required to enable coordination between them. A third logical entity, the radio resource controller (RRC) distributes resources between both segments.

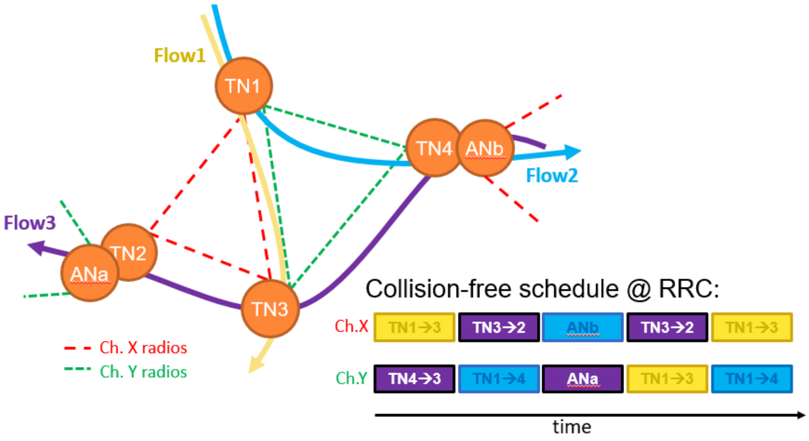
The backhaul control follows the design proposed in [7], where IEEE 802.11-based BH nodes’ PHY and MAC are extended to enable remote programmability so that the 5G-XHaul control plane is able to measure the state of the Sub-6 BH, while controlling forwarding rules and MAC transmission parameters. On the other hand, the RAN control is based on the BigAP approach [36], a centralized architecture for enterprise Wi-Fi networks providing support for seamless handover, for mobility management and load balancing. BigAP decides on the channel assignment to Wi-Fi APs on a long-term basis whereas the decision on which AP a particular STA is served is based on short-term information like channel-state information (mobility) and traffic conditions (load balancing).

In order to validate the architecture and illustrate the benefits of joint access/backhaul coordination, two use cases are implemented and demonstrated [37]: i) backhaul-aware access network control and ii) access/backhaul resource management.

Backhaul-aware access network control: the status of the wireless backhaul can determine the quality of service offered to the clients served through the access network. Therefore, the new access network control is able to provide association/handover decisions based on the status of the backhaul. On the other hand, network controlled association and handover entails creation/modification of optimal backhaul paths. To facilitate the optimization of the backhaul, the access controller informs the backhaul of association/handover decisions. All this is achieved by means of a REST API between access and backhaul controllers:

- ANC→L0-Ctrlr: POST new client station (STA) to inform upon a new association; the backhaul then computes the best path from the selected access node (AN) to the core gateway. The response to this call is a metric that corresponds to the backhaul capacity available through that AN; therefore, this call can be also used to decide the best candidate AN for a given STA, based on the status of the BH.
- ANC→L0-Ctrlr: PUT new station association information to inform of a network-driven handover (see previous bullet point).
- ANC→L0-Ctrlr: DELETE station when it leaves the access network; the backhaul controller removes the corresponding forwarding rules.
- L0-Ctrlr→ANC: POST a suggested handover in order for the backhaul controller to notify congestion. The call also suggests a possible handover that is expected to alleviate congestion in the backhaul.

Access/Backhaul resource management: the scarce radio resources available in the unlicensed bands used by Wi-Fi access and backhaul networks require an intelligent management. Note that the CSMA used in Wi-Fi will provide equal access probability to AN and TN interfaces sharing the channel, when the latter has to transport more traffic/flows (aggregation of multiple access links) and, therefore, deserves more resources. Thus, centralised scheduling of AN and TN interfaces using a hybrid TDMA/CSMA scheme allows to fairly distribute resources between both networks and to reduce collisions inherent to Wi-Fi. Indeed, the use of TDMA would enable the application of advanced network functions such as network slicing. hMAC [38] is a technique to provide a TDMA-like access in Wi-Fi interfaces. The interested reader is referred to [2] for an in-depth description and evaluation of hMAC. Time slots available at each radio link are thus managed by the RRC. The RRC then communicates the scheduling to the access nodes (ANx) and backhaul nodes (TNy) using an agent deployed at each device. Figure 5-19 provides an example of such hybrid TDMA scheduling, where four TN nodes and two AN nodes share two radio channels (Ch. X and Ch. Y) to serve three data flows. More details on the hybrid CSMA/TDMA operation can be found in [7].



Ch.X	TN1→3	TN3→2	ANb	TN3→2	TN1→3
Ch.Y	TN4→3	TN1→4	ANa	TN1→3	TN1→4

Figure 5-19: Example of scheduling of a TDMA-based backhaul and access network infrastructure.

5.4.2 Testbed description

In order to evaluate a proof-of-concept of the proposed architecture, a particular topology was designed, which is depicted in Figure 5-20. These are the different elements represented in the figure.

- STA: mobile terminals equipped with Wi-Fi (5xTP-Link WDR4300 with OpenWRT and 1xAndroid smartphone).
- TN: wireless backhaul nodes (Intel NUC D54250WYKH) with customized Linux (including BigAP and SDN agents) and equipped with two Atheros-based IEEE 802.11abgn interfaces operating in the 5GHz band.
- AN/TN: nodes operating both as access (AN) and BH TNs.
- TN/GW: Endpoints of wireless backhaul mesh connecting the wireless infrastructure to the wired core through an Ethernet connection.
- CORE network: access network controller, backhaul controller and traffic source server.

IP addressing is divided into three domains: i) access network, including end user stations (STAs) and AN interfaces; ii) backhaul network; and iii) core network. In the experiment, all STAs except for STA2 are static and act as traffic sinks (traffic used to load the network is generated from the core). For the sake of reproducibility, we limit all the Wi-Fi links to the most robust modulation (6 Mbps). Backhaul control is implemented in an ODL platform. In this implementation, the RRC provides a TDMA scheduling based on the number of flows going through each link, information that can be provided by the backhaul controller. A third server is used as TCP source, where downlink traffic is generated. Finally, we have to note that the deployment of a dynamic TDMA scheduling by the controller, requires precise synchronization of the nodes so that the time slots defined start at precisely the same instant in all the devices sharing the medium. In this work, synchronization is provided through a common Ethernet infrastructure. Note that a more comprehensive discussion on synchronization is given in [2].

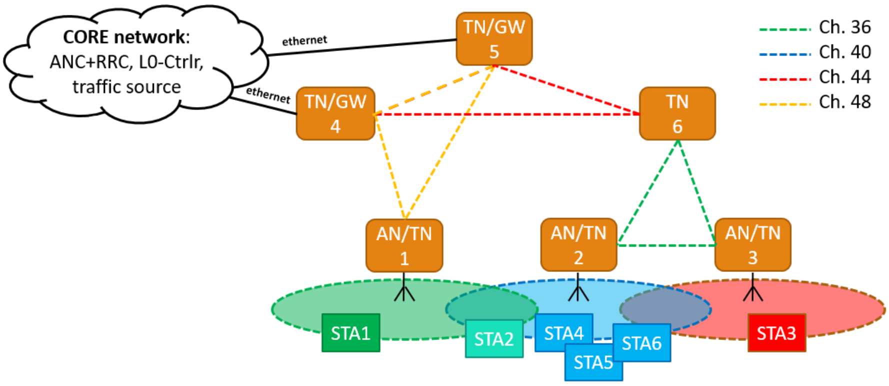


Figure 5-20: Topology of the joint access/backhaul architecture experiment.

Running the experiment

The scenario depicted in Figure 5-20 was set up in order to assess the potential of the joint operation of the access network controller (ANC) and the backhaul controller (L0-Ctrlr). In that scenario, we run different sets of experiments. The first one was intended to adjust the configuration of the TDMA parameters that the Sub-6 GHz TNs expose to the control plane (cf. [7]). For this particular implementation, we found that the slot duration should allow, at least, ten transmissions (up to 20 ms, to guarantee transmission of average-sized packets at lowest rates) for optimal performance. In a second set, we compare two different cases: *Legacy*, where there is no cooperation between access and backhaul control and medium access follows the legacy distributed CSMA/CA of IEEE 802.11; and *Joint Ctrl.*, implementing the proposed joint control of access and backhaul. Those two experiments follow a particular sequence, shaping a “story” that highlights the benefits of the proposed joint access-backhaul control. During the initialization phase of the experiment, the topology is built and the L0-Ctrlr starts receiving reports from the nodes (i.e. radio and packet statistics and list of neighbours as defined in [7]). The experiment then follows the particular sequence detailed below. Figure 5-21 shows the evolution of aggregated throughput and fairness throughout that sequence for the two cases:

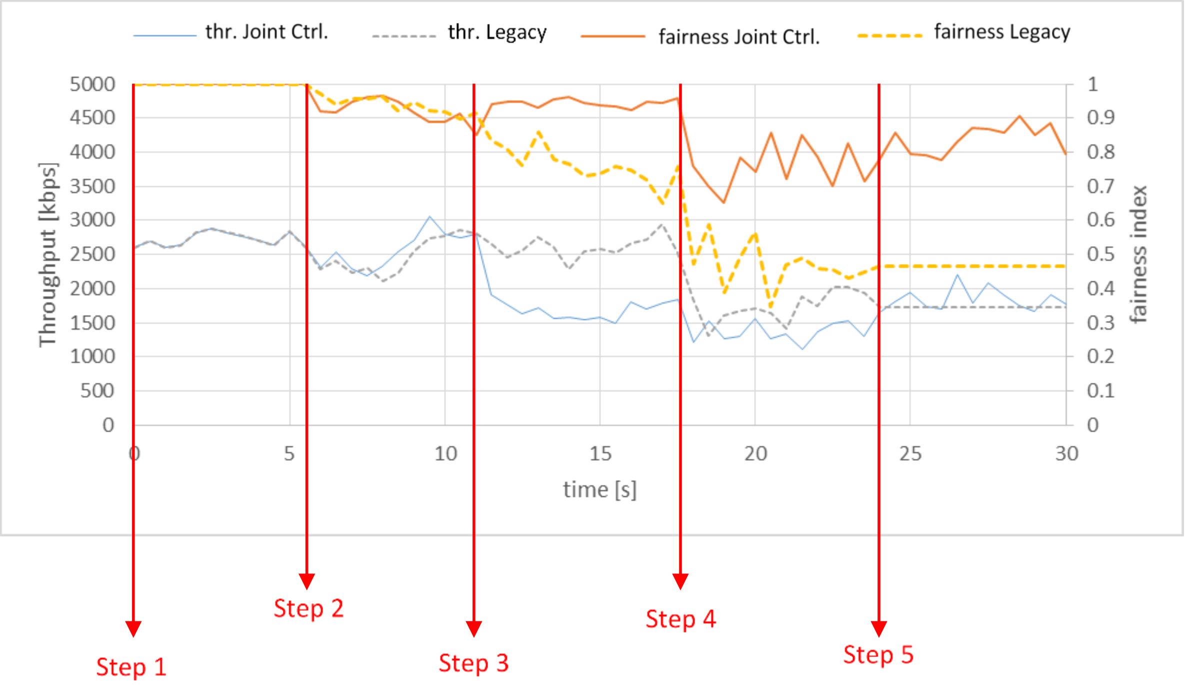


Figure 5-21: Throughput and fairness measurements for Joint Ctrl. and Legacy approaches.

1. STA1 is switched on (only reaches AN1). As a consequence:
    1. ANC handles association of STA1 with AP1 and notifies the backhaul controller using the ANC/L0-Ctrlr interface (see Figure 5-18).
    2. L0-Ctrlr computes the optimal path between core network and STA1 and sets new forwarding rules on the corresponding backhaul nodes (TN1, and TN4).
    3. **STA1 achieves around 2.7Mbps.**
2. STA2 switched on (can reach AN1 and AN2):
    1. ANC decides association of STA2 to AN2 and notifies the backhaul controller through ANC/L0-Ctrlr interface.
    2. L0-Ctrlr computes the optimal path between core network and STA2 and sets new forwarding rules on the corresponding backhaul nodes (TN2, TN6 and TN5).
    3. **STA1's throughput is reduced to 1.7Mbps and STA2 achieves 0.9Mbps**. Note that both AN1 and TN6 compete for the same channel 36. We also have to highlight here that the link TN5 → TN6 → TN2 has a capacity of only 1.5Mbps. The difference in the achieved throughput **decreases fairness**.
3. STA3 switched on (only reaches AN3):
    1. ANC handles association of STA3 with AP3 and notifies the backhaul controller using ANC/L0-Ctrlr interface.
    2. L0-Ctrlr computes the optimal path between core network and STA3 and sets new forwarding rules on the corresponding backhaul nodes (TN3, TN6 and TN5).
    3. Given that now TN6 carries two flows through channel 36 while AN1 carries only one, the *Joint Ctrl*. approach activates the TDMA to produce a fairer share of channel 36 (and channel 44, shared between TN5 and AN3).
    4. The ***Joint Ctrl*. approach achieves near-perfect fairness while, in the *Legacy* network this parameter degrades**. On the other hand, **throughput is notably reduced after applying TDMA** in the *Joint Ctrl*. approach.
4. STA4, STA5 and STA6 are switched on (only reach AN2)
    1. ANC handles association of those STAs with AN2 and notifies the backhaul controller using the ANC/L0-Ctrlr interface.
    2. L0-Ctrlr computes the optimal path between core network and STAs 4, 5 and 6 and sets new forwarding rules on the corresponding backhaul nodes (TN2, TN6 and TN5).
    3. According to the new balance of data flows, the *Joint Ctrl*. approach determines a new scheduling.
    4. Due to the increased interference, **in the *Legacy* case both throughput and fairness are notably reduced** while, on the other hand, ***Joint Ctrl*. is capable of keeping a reasonably high fairness without losing capacity** with respect to the previous step.
5. Backhaul-assisted handover
    1. L0-Ctrlr detects congestion (i.e. measured channel utilization exceeds a configurable threshold) and suggests the ANC to move STA2 from AN2 to AN1 using the ANC/L0-Ctrlr interface.
    2. In the *Joint Ctrl*. approach, the ANC proceeds with the handover of STA2 as suggested, then it informs the L0-Ctrlr through the common interface and, hence, L0-Ctrlr reconfigures the layer 2 path for STA2's flows accordingly (i.e. STA2 does not change its IP address). Note that this step does not occur in the *Legacy* case (In step 5, Figure 5-21 shows the average value measured during step 4 for the *Legacy* case); even though the ANC alone could manage association of STAs, AN2 is not seen as congested (channel 40 is clean and AN2 carries less than 700kbps) and would not trigger a re-association.
    3. The change in the balance of flows triggers a new TDMA slot allocation.
    4. The *Joint Ctrl*. approach is capable of reducing the congestion thus improving fairness and throughput.

5.5 Definition and evaluation of SDN Transport – MME interface for 4G RANs

In this Section we describe two example interfaces between a Transport SDN controller and the MME of the Mobile Network. We notice that the MME is a 4G entity, but we assume that a similar core network function will be available in 5G. The first interface described in section 5.5.1 is tailored to RANs where the user connections are identified explicitly through a *bearer*. The main idea of this interface is to extend the definition of bearers to the wireless BH, which would allow to define per-subscriber policies in the wireless BH. The second interface described in section 5.5.2 studies the benefits of exploiting long-term RAN information in order to optimize various network KPIs.

5.5.1 Interface between the MME and the 5G-XHaul Level 0 Area Controller

Figure 5-22 illustrates an architecture that interfaces a 5G-XHaul Level-0 controller, in charge of a Small Cell wireless backhaul, with a Mobile Network (MN). The proposed interconnection between 5G-XHaul and the MN is implemented through the MME, in case of a 4G MN, or an equivalent entity in case of a 5G MN. In Figure 5-22 we can see 4G/5G SCs connected to a 5G-XHaul TN that provides wireless backhaul connectivity. Following the 5G-XHaul architecture described in Section 2, TNs are grouped in control plane areas (c.f. Figure 2-1), and there is a logically centralized Level-0 Area Controller (AC) in charge of performing path allocation for each area. In particular, the AC programs TNs to allocate backhaul paths between SCs and gateways[5]; where gateways provide connectivity between a control area and the rest of the transport network, i.e. other 5G-XHaul control plane area. Notice that in practice SCs would be connected to the 5G-XHaul control plane area through an ETN. To simplify the explanation though, in this section we refer directly to the SCs, and assume that a SC has an embedded ETN.

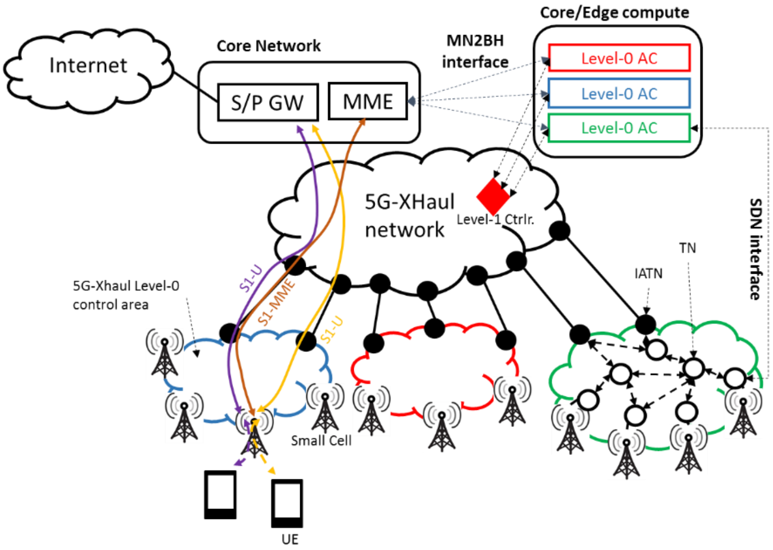


Figure 5-22. MN2BH interface between 5G-XHaul Level-0 ACs and MME

In this section we propose an instantiation of the 5G-XHaul architecture, where the Level-0 AC defines flows in the wireless backhaul to match the radio bearers used in the cellular network. For example, in LTE when a UE has data to transmit it sets up a uniquely identified bearer. Hence, a 5G-XHaul control plane area would use as tunnel ID in the wireless BH the same bearer identifier. This tunnel ID would be used to set up a path between the SC and a gateway (c.f. Figure 5-22). Notice that this is completely different from the approach currently used in BH networks, where the traffic from all users attached to a given base station is aggregated in a single BH path, typically identified with a VLAN tag. Following this architecture, an operator would be able to implement per-subscriber/application policies in the wireless backhaul.

Figure 5-22 depicts an interface between the MN, in particular the MME, and the Level-0 AC (*MN2BH interface*). This interface is needed to command the dynamic establishment of per-bearer transport paths in the control area. Next, we provide a detailed description of this interface.

[5] A gateway corresponds to an Inter-Area Transport Node (IATN) in the 5G-XHaul control plane architecture

5.5.1.1 Mobile Network to Backhaul Interface (MN2BH)

Figure 5-23 provides an example description of some procedures that could be executed over the *MN2BH* interface. In Figure 5-23, three phases are presented. In the first phase the UE changes from RRC IDLE to RRC CONNECTED state upon the arrival of an uplink packet. The eNB notifies the MME that the UE is in RRC CONNECTED state, after which an authentication procedure between the UE and the MME follows. Once the UE is authenticated, the MME notifies the 5G-XHaul Level-0 AC about the Tunnel End Point ID (TEID) used for the uplink part of the S1 bearer (*S1-SGW-TEID*), and about the eNB the UE is connected to. With this information the AC runs its path allocation engine and programs the newly computed path in the affected TNs. Then, the controller notifies the MME that the transport path is up, after which the MME can notify the eNB about the uplink GTP TEID, and the uplink path is established following the normal LTE procedure. After the uplink path is established (phase 2 in Figure 5-23), the eNB assigns a TEID for the downlink path (*S1-eNB-TEID*) and notifies the MME. Consequently, the MME notifies the AC, which proceeds to instantiate a downlink path in a similar manner as before. When selecting the downlink path the controller decides from which gateway the downlink traffic should be injected in the 5G-XHaul control plane area (c.f. Figure 5-22). Once the downlink transport path is available, the MME notifies the SGW and traffic begins to flow in the downlink direction.

In the third phase of Figure 5-23 an inactivity timeout in the eNB is triggered, and the eNB requests the MME to release the UE context in order to save resources. The MME authorizes the context release, and also requests the SGW to erase the downlink part of the S1 bearer *S1-eNB-TEID*). The eNB proceeds then to release the radio resources, and the UE transitions to RRC IDLE state. After this, the MME notifies the AC, and the downlink (*S1-eNB-TEID*) and uplink (*S1-SGW-TEID*) transport flows associated with that UE are released.

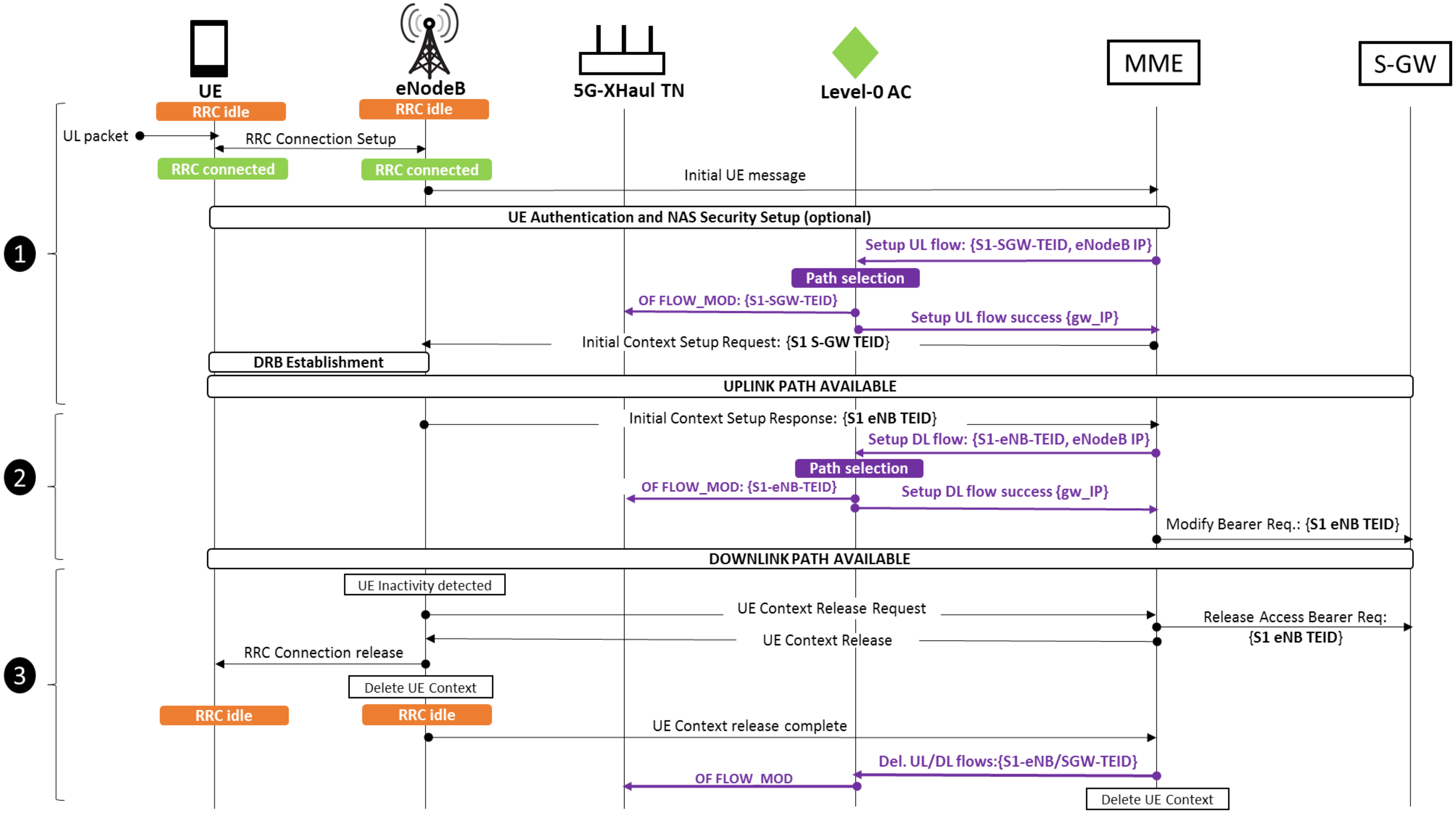

UE
eNodeB
5G-XHaul TN
Level-0 AC
MME
S-GW
RRC idle
RRC idle
UL packet
RRC Connection Setup
RRC connected
RRC connected
Initial UE message
UE Authentication and NAS Security Setup (optional)
1
Setup UL flow: {S1-SGW-TEID, eNodeB IP}
Path selection
OF FLOW_MOD: {S1-SGW-TEID}
Setup UL flow success {gw_IP}
Initial Context Setup Request: {S1 S-GW TEID}
DRB Establishment
UPLINK PATH AVAILABLE
Initial Context Setup Response: {S1 eNB TEID}
2
Setup DL flow: {S1-eNB-TEID, eNodeB IP}
Path selection
OF FLOW_MOD: {S1-eNB-TEID}
Setup DL flow success {gw_IP}
Modify Bearer Req.: {S1 eNB TEID}
DOWNLINK PATH AVAILABLE
UE Inactivity detected
UE Context Release Request
Release Access Bearer Req:
{S1 eNB TEID}
UE Context Release
RRC Connection release
3
Delete UE Context
RRC idle
RRC idle
UE Context release complete
Del. UL/DL flows:{S1-eNB/SGW-TEID}
OF FLOW_MOD
Delete UE Context

Figure 5-23. Example procedures in the MN2BH Interface.

5.5.1.2 Scalability Analysis

The proposed MN2BH interface introduces additional signaling, and requires the TNs to be able to hold a certain number of rules. Therefore, in this section we study the scalability properties of this interface, using as basis the network measurement campaign introduced in Section 5.1. In particular, we present a study that dimensions the number of rules that need to be maintained in each TN. The interested reader is referred to [17] for an extended study that also analyzes the amount of signaling required between the Level-0 AC and the TNs.

5.5.1.3 Dimensioning TN flow tables

An Active UE is defined as the one having one Signaling Radio Bearer (SRB) and at least one non-guaranteed bitrate Data Radio Bearer (DRB) successfully configured for it. Hence, the number of active UEs define the number of TN flows originating from or ending at a cell. To calculate the number of rules at a TN, we consider the FLRR scheme introduced in Section 3.2, and a worst case of 3 rules per flow (corresponding to the switch and common type nodes with a backup flow). Considering one UL and one DL flow for each active UE, and *n* UEs being backhauled through a TN, then *6n* rules need to be maintained at the TN.

To dimension the number of rules that will need to be maintained by a TN at peak traffic hours, we first find the busy hour from our LTE traces, to then generate an empirical CDF of the number of active UEs per cell during the busy hour. We then derive numerically the CDF of the number of rules that a TN would carry for a given number of cells as follows. First, we generate randomly the number of flows for each cell (by using the CDF of number of active UE), then sum these numbers and multiply it by 6 to find the number of rules for these random samples. After producing 100.000 tuples of random number of flows for all cells, we generate the corresponding CDF for the number of rules at a transport node, which is shown in Figure 5-24. Since we expect that the number of cells relayed by a given TN will be limited for small cell urban scenarios, we evaluate at most 20 aggregated cells.

As seen in Figure 5-24, the CDF generated from the traces for the number of rules per cell aligns with the 1 cell case, which validates the numerical method used for the evaluations. For the limited number of cells case, the transport nodes are hence expected to have less than 2*K* rules as seen in the figure. This number is lower than the capacity of commodity switches, which typically rely on Ternary Content Addressable Memory (TCAM) to implement the flow tables, such as between 2 - 24*K* rules [35]. These results show that the proposed MN2BH interface can be implemented on current LTE systems using commodity switches, without any capacity issues.

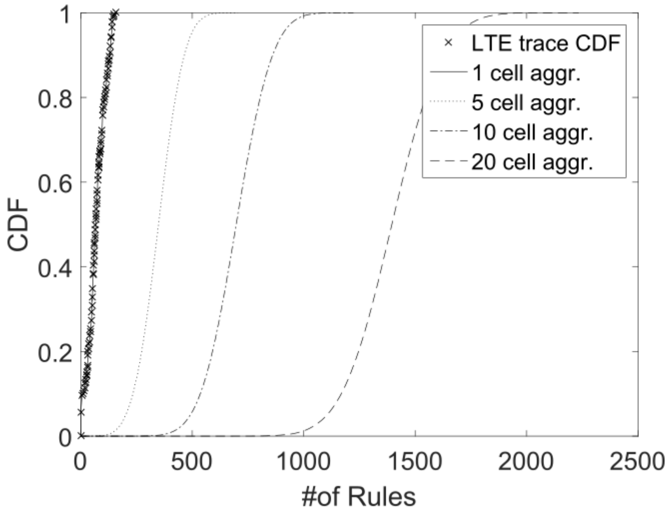


Figure 5-24. CDF of number of rules at a transport node carrying flows of a given number of cell.

5.5.2 TE/Data analysis based traffic prediction

According to the NGMN white paper [39], the end users will have access to a diverse set of services (ranging from high definition video and audio, web browsing, games, to keep alive messages etc.). In such demanding environments the use of context information is imperative for enabling the efficient use of the network resources [40]. Also, according to [40], the term context refers to “all the information that may be exchanged among the network entities, even heterogeneous ones so as to solve challenging networking problems such as management and control of the network resources". This generic definition of the context information implies that all information types may be used for the optimization of the network management and control such as radio measurements, mobility information/measurements (e.g., user speed, direction, etc.), etc.

Several Context Aware mechanisms take advantage of location information, mobility information, network statistics (access or backhaul), current service information [41][42], though they require a huge amount of signalling and increased processing capabilities.

On the other hand, the users tend to change their behaviours depending on the location, the time, and other characteristics as well, such as user equipment type, battery level of the user equipment, charging status of the user account (e.g., remaining credits, etc.), overall user income, user educational education level, etc. Such knowledge facilitates the network to predict the overall traffic requirements in certain locations for specific time periods and to proceed in the relevant actions in advance.

By using data mining mechanisms:

- we may identify the user behaviour parameters with greatest impact on traffic generation/service requests, and
- build user behavioral profiles,
- which can be used for predicting the overall throughput requirements in certain areas, or even for optimised mobility management, call admission control, etc.

The process, due to the nature of the data analytics consists of two phases, namely the offline and the online one. The former includes:

- the gathering of the user context and history and storing of this information,
- the application of feature selection filters for identifying the most relevant parameters,
- the creation/identification of user behavioural profiles in terms of accessed service, and mobility,

whereas the online includes

- the distribution of the profiles to the networking entities, and,
- the combination with online inputs (e.g., UE battery level, charging status, time, etc) to predict the actual user behavior.

Figure 5-25 provides an example of where the profiles can be used, as soon as they are defined/created. In this example the UE performs an attach request (steps 1 and 2). The MME retrieves the multidimensional profile list and stores those profiles that fit best to the dynamic characteristics (e.g., location, time, etc.). Additionally it provides the list of profiles to the eNB (steps 3, 4, 5, 6). The eNB stores the list of profiles and uses it when required by exploiting real time context information (e.g., battery, charging, etc.) (step 7). Then, based on the real time information from the UEs, the eNB informs the centralized (management) functions for the need of additional resources.

More generic estimations may take place without real time information. This however suggests a trade off since without the use of real time information the behaviour of the UEs may be inaccurate.

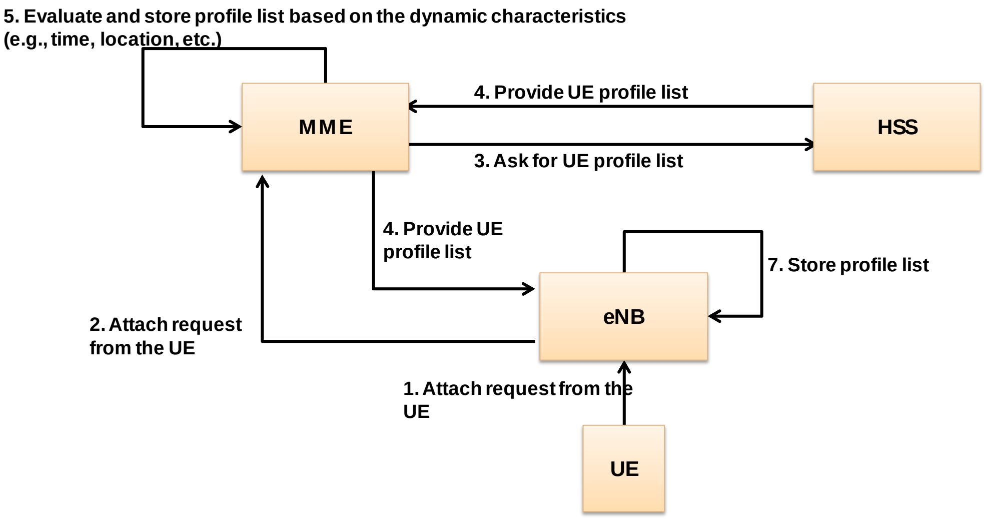

5. Evaluate and store profile list based on the dynamic characteristics (e.g., time, location, etc.)
MME
4. Provide UE profile list
HSS
3. Ask for UE profile list
4. Provide UE profile list
7. Store profile list
eNB
2. Attach request from the UE
1. Attach request from the UE
UE

Figure 5-25: Distribution of the Profiles.

Figure 5-26 shows the signalling cost of updating the RAN serving UEs in a shopping mall [43] providing updates regarding the UEs' position, accessed service, mobility, battery levels for a day of operation. Three schemes are considered, one where the RAN is informed about the UEs profiles/activity periodically every minute, one where the RAN is informed periodically every minute but with generic location information, and our proposed scheme where two generic UE profiles are defined[6]. Every time that a UE deviates from its assigned generic profile, the network is informed to act accordingly. In our evaluation we consider that UEs comply with their service profiles 50% and 90% of the time. As shown in the figure the reduction in signalling costs is significant. If we consider even higher concentrations of UEs even more significant gains can be achieved, highlighting the gains in the RAN.

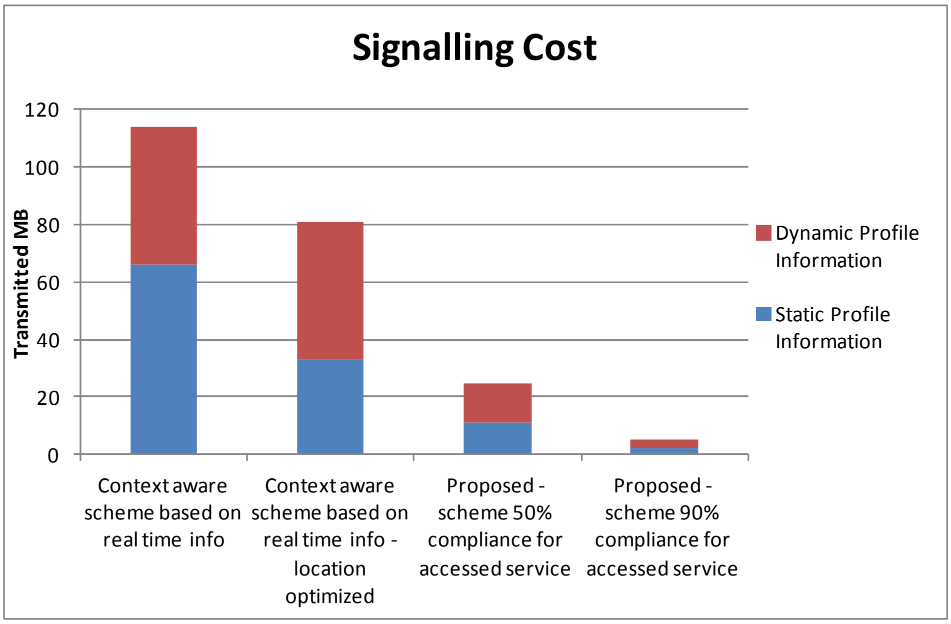


Figure 5-26: Context Information Exchange Cost.

The use of generic profiles for predicting the overall traffic demands based on previous user behaviours of course facilitates the accurate decision – as captured in Figure 5-27 (the simulation setup is a single floor Mall environment (2 macro eNBs, 3 home eNBs, and 1 GSM macro-cell). Here the macro/home eNBs are the

[6] Notice that appropriate profiles can be built offline using non-supervised learning techniques, such as clustering

access nodes mentioned before. With 10% of the users randomly changing their behaviour in terms of accessed service or mobility, or both the traffic engineering mechanism performs very accurately achieving 5.2% average error and less than 4% standard deviation with users who access VoIP, Video streaming, Web, and FTP services).

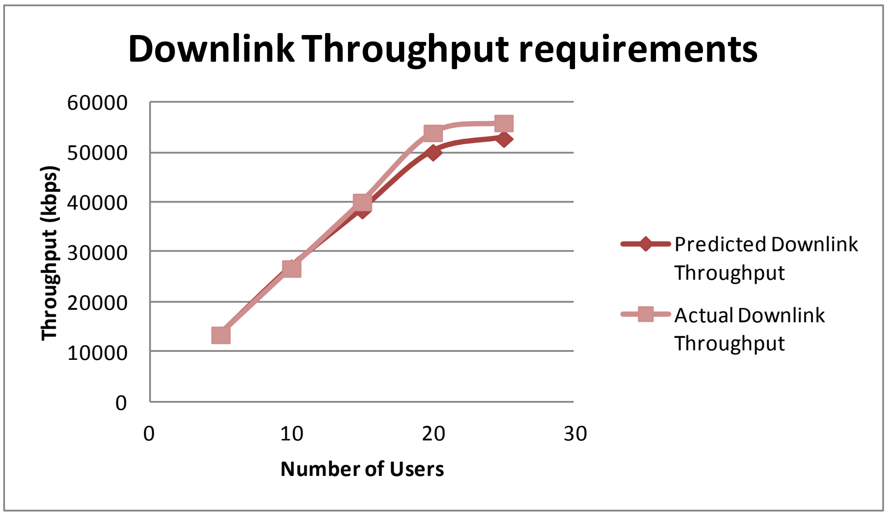


Figure 5-27: User downlink throughput requirements prediction using profiles.

6 Summary and conclusions

After laying out the principles of the 5G-XHaul control-plane in deliverable D3.1 [3], this deliverable has elaborated on the main architectural building blocks, while describing and evaluating a wide array of control-plane mechanisms that operate at various parts of this architecture.

In particular, this deliverable has described a refined definition of the 5G-XHaul control-plane architecture, specifying various interfaces, and proposing mechanisms that enhance the scalability of an SDN based control plane in transport networks. Then, we have presented and evaluated various Traffic Engineering (TE) mechanisms tailored to wireless transport areas, including mmWave and Sub6 technologies. The proposed TE mechanisms allow to allocate paths and support fast recovery. We have also extended the work started in deliverable D3.1, and we have further defined the south-bound interfaces between the 5G-XHaul network elements and their corresponding controllers. For example, we have presented a custom REST based interface to control TSON Edge nodes, and a NETCONF based interface to control WDM-PON and mmWave devices. Finally, we have delved deeper into mechanisms that enable a tighter intergration between the Mobile and the Transport Network. An exhaustive measurement campaign has been carried out to support these mechanisms. Inaddition, we have reported on techniques that allow to localise users as they move through the Mobile Network.

Looking towards the third and final year of the project, WP3 will focus its work, among others, on the following aspects: i) a detailed specification and demonstration of the end to end SDN control-plane hierachy providing connectivity across wireless and optical technology domains, and ii) the exploitation of RAN information, including user mobility and cell loads, to optimize the transport network.

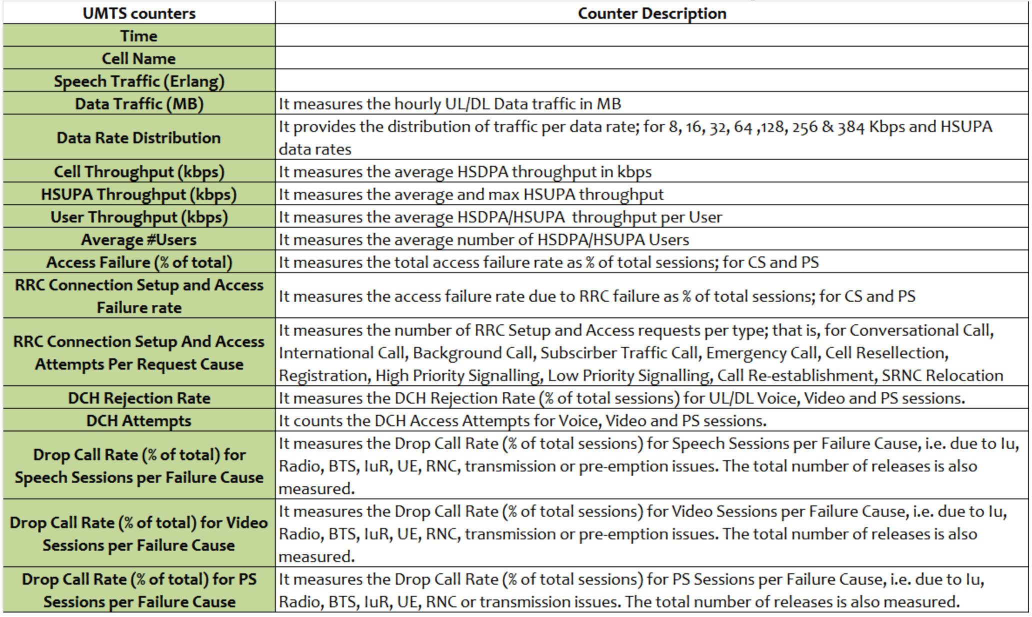
7 Annexes

7.1 Annex 1: Statistics at Base Station-Level

This Annex provides details regarding the real network statistics that were considered for the purposes of the 5G-XHaul project. More specifically:

Table 7-1 and Table 7-2 provide the UMTS and LTE counters respectively along with their description while Table 7-3 provides indicative network statistics on a per counter and cell basis.

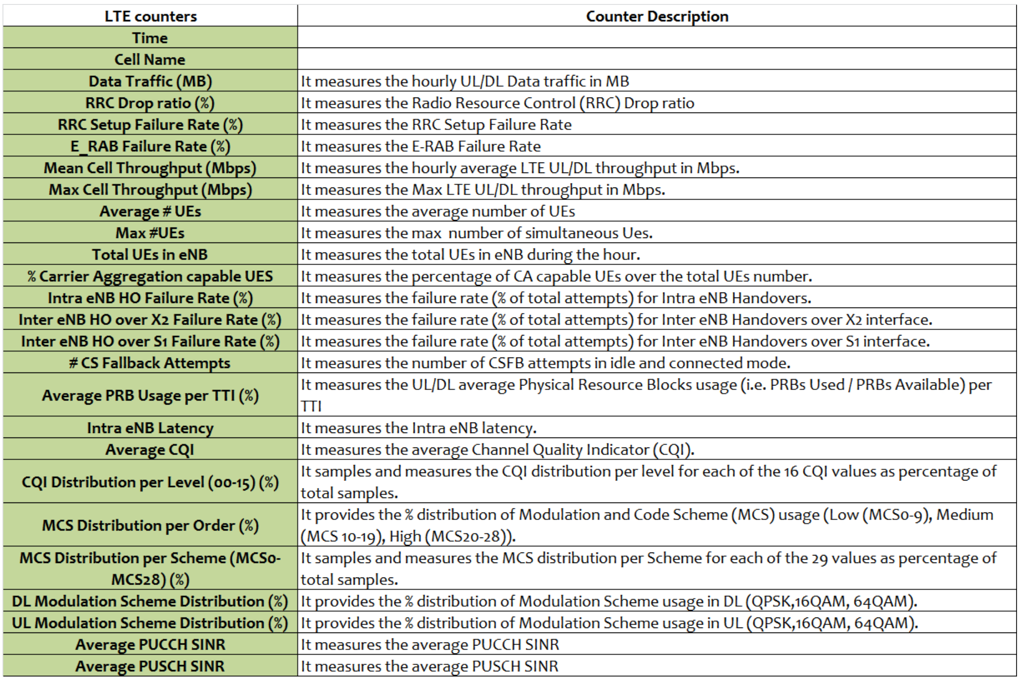
Table 7-1: UMTS Counters and their Description.



UMTS counters	Counter Description
Time	
Cell Name	
Speech Traffic (Erlang)	
Data Traffic (MB)	It measures the hourly UL/DL Data traffic in MB
Data Rate Distribution	It provides the distribution of traffic per data rate; for 8, 16, 32, 64 ,128, 256 & 384 Kbps and HSUPA data rates
Cell Throughput (kbps)	It measures the average HSDPA throughput in kbps
HSUPA Throughput (kbps)	It measures the average and max HSUPA throughput
User Throughput (kbps)	It measures the average HSDPA/HSUPA throughput per User
Average #Users	It measures the average number of HSDPA/HSUPA Users
Access Failure (% of total)	It measures the total access failure rate as % of total sessions; for CS and PS
RRC Connection Setup and Access Failure rate	It measures the access failure rate due to RRC failure as % of total sessions; for CS and PS
RRC Connection Setup And Access Attempts Per Request Cause	It measures the number of RRC Setup and Access requests per type; that is, for Conversational Call, International Call, Background Call, Subscirber Traffic Call, Emergency Call, Cell Resellection, Registration, High Priority Signalling, Low Priority Signalling, Call Re-establishment, SRNC Relocation
DCH Rejection Rate	It measures the DCH Rejection Rate (% of total sessions) for UL/DL Voice, Video and PS sessions.
DCH Attempts	It counts the DCH Access Attempts for Voice, Video and PS sessions.
Drop Call Rate (% of total) for Speech Sessions per Failure Cause	It measures the Drop Call Rate (% of total sessions) for Speech Sessions per Failure Cause, i.e. due to Iu, Radio, BTS, IuR, UE, RNC, transmission or pre-emption issues. The total number of releases is also measured.
Drop Call Rate (% of total) for Video Sessions per Failure Cause	It measures the Drop Call Rate (% of total sessions) for Video Sessions per Failure Cause, i.e. due to Iu, Radio, BTS, IuR, UE, RNC, transmission or pre-emption issues. The total number of releases is also measured.
Drop Call Rate (% of total) for PS Sessions per Failure Cause	It measures the Drop Call Rate (% of total sessions) for PS Sessions per Failure Cause, i.e. due to Iu, Radio, BTS, IuR, UE, RNC or transmission issues. The total number of releases is also measured.

Note: All counters are available on a per hour basis

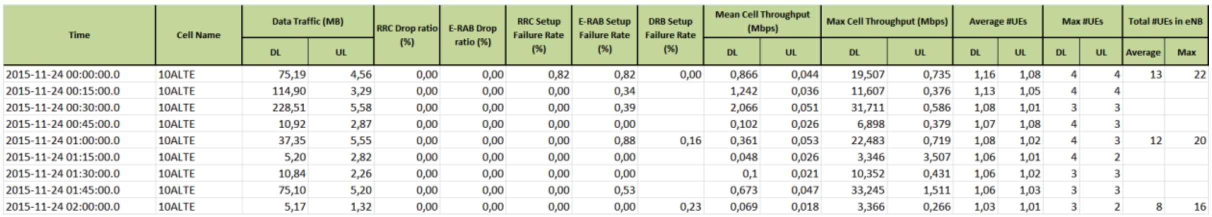
Table 7-2: LTE Counters and their Description.



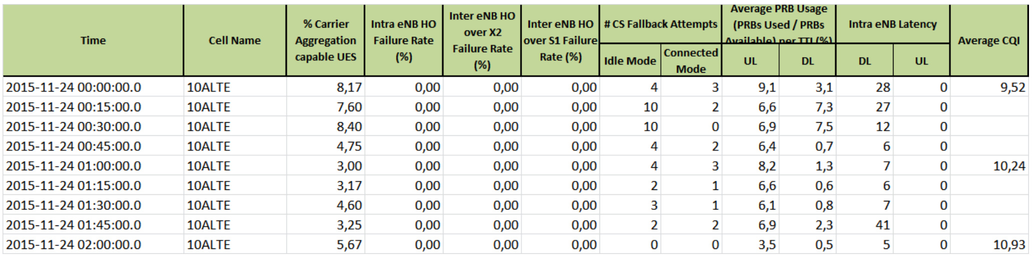
LTE counters	Counter Description
Time	
Cell Name	
Data Traffic (MB)	It measures the hourly UL/DL Data traffic in MB
RRC Drop ratio (%)	It measures the Radio Resource Control (RRC) Drop ratio
RRC Setup Failure Rate (%)	It measures the RRC Setup Failure Rate
E_RAB Failure Rate (%)	It measures the E-RAB Failure Rate
Mean Cell Throughput (Mbps)	It measures the hourly average LTE UL/DL throughput in Mbps.
Max Cell Throughput (Mbps)	It measures the Max LTE UL/DL throughput in Mbps.
Average # UEs	It measures the average number of UEs
Max #UEs	It measures the max number of simultaneous Ues.
Total UEs in eNB	It measures the total UEs in eNB during the hour.
% Carrier Aggregation capable UES	It measures the percentage of CA capable UEs over the total UEs number.
Intra eNB HO Failure Rate (%)	It measures the failure rate (% of total attempts) for Intra eNB Handovers.
Inter eNB HO over X2 Failure Rate (%)	It measures the failure rate (% of total attempts) for Inter eNB Handovers over X2 interface.
Inter eNB HO over S1 Failure Rate (%)	It measures the failure rate (% of total attempts) for Inter eNB Handovers over S1 interface.
# CS Fallback Attempts	It measures the number of CSFB attempts in idle and connected mode.
Average PRB Usage per TTI (%)	It measures the UL/DL average Physical Resource Blocks usage (i.e. PRBs Used / PRBs Available) per TTI
Intra eNB Latency	It measures the Intra eNB latency.
Average CQI	It measures the average Channel Quality Indicator (CQI).
CQI Distribution per Level (00-15) (%)	It samples and measures the CQI distribution per level for each of the 16 CQI values as percentage of total samples.
MCS Distribution per Order (%)	It provides the % distribution of Modulation and Code Scheme (MCS) usage (Low (MCS0-9), Medium (MCS 10-19), High (MCS20-28)).
MCS Distribution per Scheme (MCS0-MCS28) (%)	It samples and measures the MCS distribution per Scheme for each of the 29 values as percentage of total samples.
DL Modulation Scheme Distribution (%)	It provides the % distribution of Modulation Scheme usage in DL (QPSK,16QAM, 64QAM).
UL Modulation Scheme Distribution (%)	It provides the % distribution of Modulation Scheme usage in UL (QPSK,16QAM, 64QAM).
Average PUCCH SINR	It measures the average PUCCH SINR
Average PUSCH SINR	It measures the average PUSCH SINR

Note: Counters are available on a quarterly basis, except for Average/max UEs per eNB, CQI and average SINR PUCCH/PUSCH.

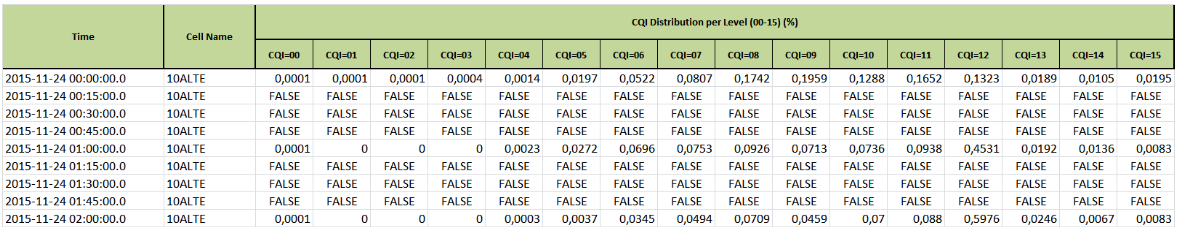
Table 7-3: Indicative Statistics on a per counter and cell basis (example = LTE)



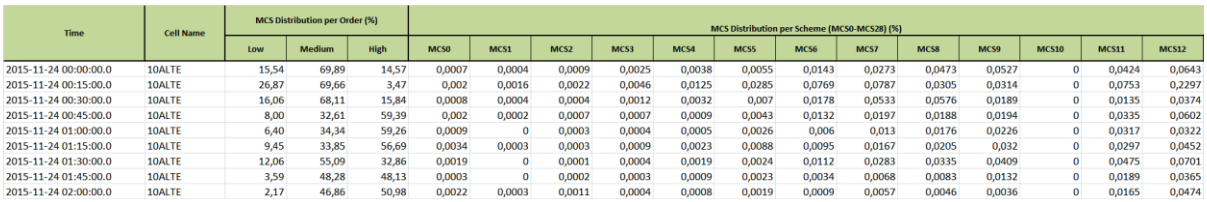
Time	Cell Name	Data Traffic (MB)		RRC Drop ratio (%)	E-RAB Drop ratio (%)	RRC Setup Failure Rate (%)	E-RAB Setup Failure Rate (%)	DRB Setup Failure Rate (%)	Mean Cell Throughput (Mbps)		Max Cell Throughput (Mbps)		Average #UEs		Max #UEs		Total #UEs in eNB	
		DL	UL						DL	UL	DL	UL	DL	UL	DL	UL	Average	Max
2015-11-24 00:00:00.0	10ALTE	75,19	4,56	0,00	0,00	0,82	0,82	0,00	0,866	0,044	19,507	0,735	1,16	1,08	4	4	13	22
2015-11-24 00:15:00.0	10ALTE	114,90	3,29	0,00	0,00	0,00	0,34		1,242	0,036	11,607	0,376	1,13	1,05	4	4		
2015-11-24 00:30:00.0	10ALTE	228,51	5,58	0,00	0,00	0,00	0,39		2,066	0,051	31,711	0,586	1,08	1,01	3	3		
2015-11-24 00:45:00.0	10ALTE	10,92	2,87	0,00	0,00	0,00	0,00		0,102	0,026	6,898	0,379	1,07	1,08	4	3		
2015-11-24 01:00:00.0	10ALTE	37,35	5,55	0,00	0,00	0,00	0,88	0,16	0,361	0,053	22,483	0,719	1,08	1,02	4	3	12	20
2015-11-24 01:15:00.0	10ALTE	5,20	2,82	0,00	0,00	0,00	0,00		0,048	0,026	3,346	3,507	1,06	1,01	4	2		
2015-11-24 01:30:00.0	10ALTE	10,84	2,26	0,00	0,00	0,00	0,00		0,1	0,021	10,352	0,431	1,06	1,02	3	3		
2015-11-24 01:45:00.0	10ALTE	75,10	5,20	0,00	0,00	0,00	0,53		0,673	0,047	33,245	1,511	1,06	1,03	3	3		
2015-11-24 02:00:00.0	10ALTE	5,17	1,32	0,00	0,00	0,00	0,00	0,23	0,069	0,018	3,366	0,266	1,03	1,01	3	2	8	16



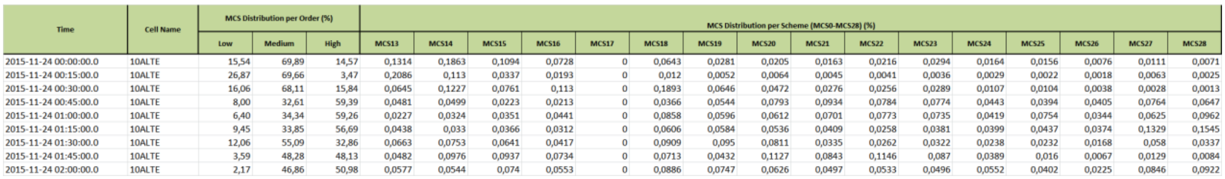
Time	Cell Name	% Carrier Aggregation capable UES	Intra eNB HO Failure Rate (%)	Inter eNB HO over X2 Failure Rate (%)	Inter eNB HO over S1 Failure Rate (%)	# CS Fallback Attempts		Average PRB Usage (PRBs Used / PRBs Available) per TTI (%)		Intra eNB Latency		Average CQI
						Idle Mode	Connected Mode	UL	DL	DL	UL	
2015-11-24 00:00:00.0	10ALTE	8,17	0,00	0,00	0,00	4	3	9,1	3,1	28	0	9,52
2015-11-24 00:15:00.0	10ALTE	7,60	0,00	0,00	0,00	10	2	6,6	7,3	27	0	
2015-11-24 00:30:00.0	10ALTE	8,40	0,00	0,00	0,00	10	0	6,9	7,5	12	0	
2015-11-24 00:45:00.0	10ALTE	4,75	0,00	0,00	0,00	4	2	6,4	0,7	6	0	
2015-11-24 01:00:00.0	10ALTE	3,00	0,00	0,00	0,00	4	3	8,2	1,3	7	0	10,24
2015-11-24 01:15:00.0	10ALTE	3,17	0,00	0,00	0,00	2	1	6,6	0,6	6	0	
2015-11-24 01:30:00.0	10ALTE	4,60	0,00	0,00	0,00	3	1	6,1	0,8	7	0	
2015-11-24 01:45:00.0	10ALTE	3,25	0,00	0,00	0,00	2	2	6,9	2,3	41	0	
2015-11-24 02:00:00.0	10ALTE	5,67	0,00	0,00	0,00	0	0	3,5	0,5	5	0	10,93



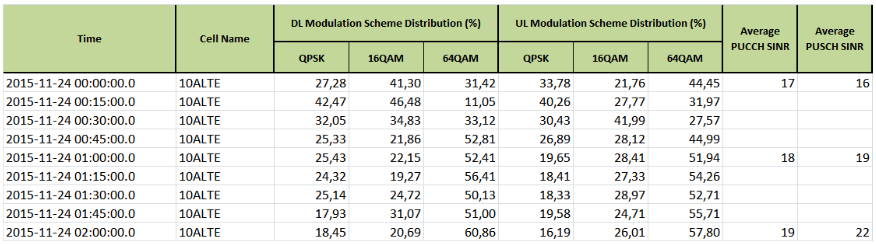
Time	Cell Name	CQI Distribution per Level (00-15) (%)															
		CQI=00	CQI=01	CQI=02	CQI=03	CQI=04	CQI=05	CQI=06	CQI=07	CQI=08	CQI=09	CQI=10	CQI=11	CQI=12	CQI=13	CQI=14	CQI=15
2015-11-24 00:00:00.0	10ALTE	0,0001	0,0001	0,0001	0,0004	0,0014	0,0197	0,0522	0,0807	0,1742	0,1959	0,1288	0,1652	0,1323	0,0189	0,0105	0,0195
2015-11-24 00:15:00.0	10ALTE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2015-11-24 00:30:00.0	10ALTE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2015-11-24 00:45:00.0	10ALTE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2015-11-24 01:00:00.0	10ALTE	0,0001	0	0	0	0,0023	0,0272	0,0696	0,0753	0,0926	0,0713	0,0736	0,0938	0,4531	0,0192	0,0136	0,0083
2015-11-24 01:15:00.0	10ALTE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2015-11-24 01:30:00.0	10ALTE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2015-11-24 01:45:00.0	10ALTE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
2015-11-24 02:00:00.0	10ALTE	0,0001	0	0	0	0,0003	0,0037	0,0345	0,0494	0,0709	0,0459	0,07	0,088	0,5976	0,0246	0,0067	0,0083



Time	Cell Name	MCS Distribution per Order (%)			MCS Distribution per Scheme (MCS0-MCS28) (%)												
		Low	Medium	High	MCS0	MCS1	MCS2	MCS3	MCS4	MCS5	MCS6	MCS7	MCS8	MCS9	MCS10	MCS11	MCS12
2015-11-24 00:00:00.0	10ALTE	15,54	69,89	14,57	0,0007	0,0004	0,0009	0,0025	0,0038	0,0055	0,0143	0,0273	0,0473	0,0527	0	0,0424	0,0643
2015-11-24 00:15:00.0	10ALTE	26,87	69,66	3,47	0,002	0,0016	0,0022	0,0046	0,0125	0,0285	0,0769	0,0787	0,0305	0,0314	0	0,0753	0,2297
2015-11-24 00:30:00.0	10ALTE	16,06	68,11	15,84	0,0008	0,0004	0,0004	0,0012	0,0032	0,007	0,0178	0,0533	0,0576	0,0189	0	0,0135	0,0374
2015-11-24 00:45:00.0	10ALTE	8,00	32,61	59,39	0,002	0,0002	0,0007	0,0007	0,0009	0,0043	0,0132	0,0197	0,0188	0,0194	0	0,0335	0,0602
2015-11-24 01:00:00.0	10ALTE	6,40	34,34	59,26	0,0009	0	0,0003	0,0004	0,0005	0,0026	0,006	0,013	0,0176	0,0226	0	0,0317	0,0322
2015-11-24 01:15:00.0	10ALTE	9,45	33,85	56,69	0,0034	0,0003	0,0003	0,0009	0,0023	0,0088	0,0095	0,0167	0,0205	0,032	0	0,0297	0,0452
2015-11-24 01:30:00.0	10ALTE	12,06	55,09	32,86	0,0019	0	0,0001	0,0004	0,0019	0,0024	0,0112	0,0283	0,0335	0,0409	0	0,0475	0,0701
2015-11-24 01:45:00.0	10ALTE	3,59	48,28	48,13	0,0003	0	0,0002	0,0003	0,0009	0,0023	0,0034	0,0068	0,0083	0,0132	0	0,0189	0,0365
2015-11-24 02:00:00.0	10ALTE	2,17	46,86	50,98	0,0022	0,0003	0,0011	0,0004	0,0008	0,0019	0,0009	0,0057	0,0046	0,0036	0	0,0165	0,0474



Time	Cell Name	MCS Distribution per Order (%)			MCS Distribution per Scheme (MCS0-MCS28) (%)															
		Low	Medium	High	MCS13	MCS14	MCS15	MCS16	MCS17	MCS18	MCS19	MCS20	MCS21	MCS22	MCS23	MCS24	MCS25	MCS26	MCS27	MCS28
2015-11-24 00:00:00.0	10ALTE	15,54	69,89	14,57	0,1314	0,1863	0,1094	0,0728	0	0,0643	0,0281	0,0205	0,0163	0,0216	0,0294	0,0164	0,0156	0,0076	0,0111	0,0071
2015-11-24 00:15:00.0	10ALTE	26,87	69,66	3,47	0,2086	0,113	0,0337	0,0193	0	0,012	0,0052	0,0064	0,0045	0,0041	0,0036	0,0029	0,0022	0,0018	0,0063	0,0025
2015-11-24 00:30:00.0	10ALTE	16,06	68,11	15,84	0,0645	0,1227	0,0761	0,113	0	0,1893	0,0646	0,0472	0,0276	0,0256	0,0289	0,0107	0,0104	0,0038	0,0028	0,0013
2015-11-24 00:45:00.0	10ALTE	8,00	32,61	59,39	0,0481	0,0499	0,0223	0,0213	0	0,0366	0,0544	0,0793	0,0934	0,0784	0,0774	0,0443	0,0394	0,0405	0,0764	0,0647
2015-11-24 01:00:00.0	10ALTE	6,40	34,34	59,26	0,0227	0,0324	0,0351	0,0441	0	0,0858	0,0596	0,0612	0,0701	0,0773	0,0735	0,0419	0,0754	0,0344	0,0625	0,0962
2015-11-24 01:15:00.0	10ALTE	9,45	33,85	56,69	0,0438	0,033	0,0366	0,0312	0	0,0606	0,0584	0,0536	0,0409	0,0258	0,0381	0,0399	0,0437	0,0374	0,1329	0,1545
2015-11-24 01:30:00.0	10ALTE	12,06	55,09	32,86	0,0663	0,0753	0,0641	0,0417	0	0,0909	0,095	0,0811	0,0335	0,0262	0,0322	0,0238	0,0232	0,0168	0,058	0,0337
2015-11-24 01:45:00.0	10ALTE	3,59	48,28	48,13	0,0482	0,0976	0,0937	0,0734	0	0,0713	0,0432	0,1127	0,0843	0,1146	0,087	0,0389	0,016	0,0067	0,0129	0,0084
2015-11-24 02:00:00.0	10ALTE	2,17	46,86	50,98	0,0577	0,0544	0,074	0,0553	0	0,0886	0,0747	0,0626	0,0497	0,0533	0,0496	0,0552	0,0402	0,0225	0,0846	0,0922



Time	Cell Name	DL Modulation Scheme Distribution (%)			UL Modulation Scheme Distribution (%)			Average PUCCH SINR	Average PUSCH SINR
		QPSK	16QAM	64QAM	QPSK	16QAM	64QAM		
2015-11-24 00:00:00.0	10ALTE	27,28	41,30	31,42	33,78	21,76	44,45	17	16
2015-11-24 00:15:00.0	10ALTE	42,47	46,48	11,05	40,26	27,77	31,97		
2015-11-24 00:30:00.0	10ALTE	32,05	34,83	33,12	30,43	41,99	27,57		
2015-11-24 00:45:00.0	10ALTE	25,33	21,86	52,81	26,89	28,12	44,99		
2015-11-24 01:00:00.0	10ALTE	25,43	22,15	52,41	19,65	28,41	51,94	18	19
2015-11-24 01:15:00.0	10ALTE	24,32	19,27	56,41	18,41	27,33	54,26		
2015-11-24 01:30:00.0	10ALTE	25,14	24,72	50,13	18,33	28,97	52,71		
2015-11-24 01:45:00.0	10ALTE	17,93	31,07	51,00	19,58	24,71	55,71		
2015-11-24 02:00:00.0	10ALTE	18,45	20,69	60,86	16,19	26,01	57,80	19	22

7.2 Annex 2: Network Performance and Mobility Monitoring Tools

In the context of FLEX EU-funded project (http://www.flex-project.eu/), COSMOTE and the University of Thessaly (UTH) co-developed an innovative multi-purpose platform ("FLEXtools") aiming at network performance/mobility and customer experience (QoE) monitoring, in a dynamic, future proof, user-friendly, comprehensive and cost-efficient way. The main goals of the "FLEXtools" are achieved through:

a. The depiction of access network related information (e.g., Cell-Id, RSSI/RSRP, LAC/TAC, BS/eNB location/info) at mobile terminal and at desktop
b. The collection, depiction and uploading to a dedicated server of performance-related measurements such as, maximum upload bitrate, maximum download bitrate, latency, signal strength, for post processing and evaluation

c. The identification of possible network issues (e.g., poor/no coverage, low throughput, high latency, unsuccessful handovers) via a graphical environment (WebGUI) even in real-time.

The FLEXtools platform is decomposed to client applications, server-side infrastructure and a graphical environment (WebGUIs), comprising a complete tool that can be utilized in multiple ways. More specifically the FLEXtools platform components are the following:

1. Three (3) client applications running on Android devices, in "on-demand" mode, "on-event" mode or "periodically. These include:

   a. The FLEX_QoE tool: It depicts 2G/3G/HSPA/HSPA+/4G/4G+ (and WiFi) network-related information (including BSs locations/capabilities/name, cell reselections locations/info, handover locations/info, etc.). On top of that, it collects, depicts and uploads –upon user demand-measurements related to quality that the end-user experiences (QoE), in terms of maximum upload bitrate, maximum download bitrate and latency, at the current user location (both indoors and outdoors) and while on the move. It uses Google Maps in order to present the gathered measurements to the user in real-time.
   b. The FLEX_netchanges tool: It is capable of measuring and uploading network performance related info (signal strength, latency, maximum download bitrate, maximum upload bitrate) periodically. The application runs as a background service.
   c. The FLEX_problems tool: It identifies/notifies, in real-time, on potential network issues/problems (e.g., areas exhibiting: huge number of cell reselections, poor coverage, gaps of coverage, high number of handover failures in a certain geographical area or a specific cell/sector).
2. Server-side infrastructure utilized to collect, store and process the related mobility/performance measurements.

3. A graphical environment (WebGUIs) with advanced filtering and presentation capabilities, enabling a complete network performance behavior presentation, as well as the identification of existing or potential network design/performance issues (coverage gaps, low RSSI/bitrate areas, unwanted cell reselections, handover failures, etc.).

Although the FLEXtools were primarily developed to cover the FLEX project needs, they have been designed and developed so as to address the needs of COSMOTE's (or other NatCOs) field engineers, network planners/optimizers, the Upper management and even the Customer Care. The FLEXtools constitute a future proof platform, supporting the gathering/processing of bulk network measurements at a national level, in a cost efficient way; involving OTE Group employees who could contribute by utilizing their own (Android) smartphones.

A brief overview of the FLEXtools' features/capabilities is given below.

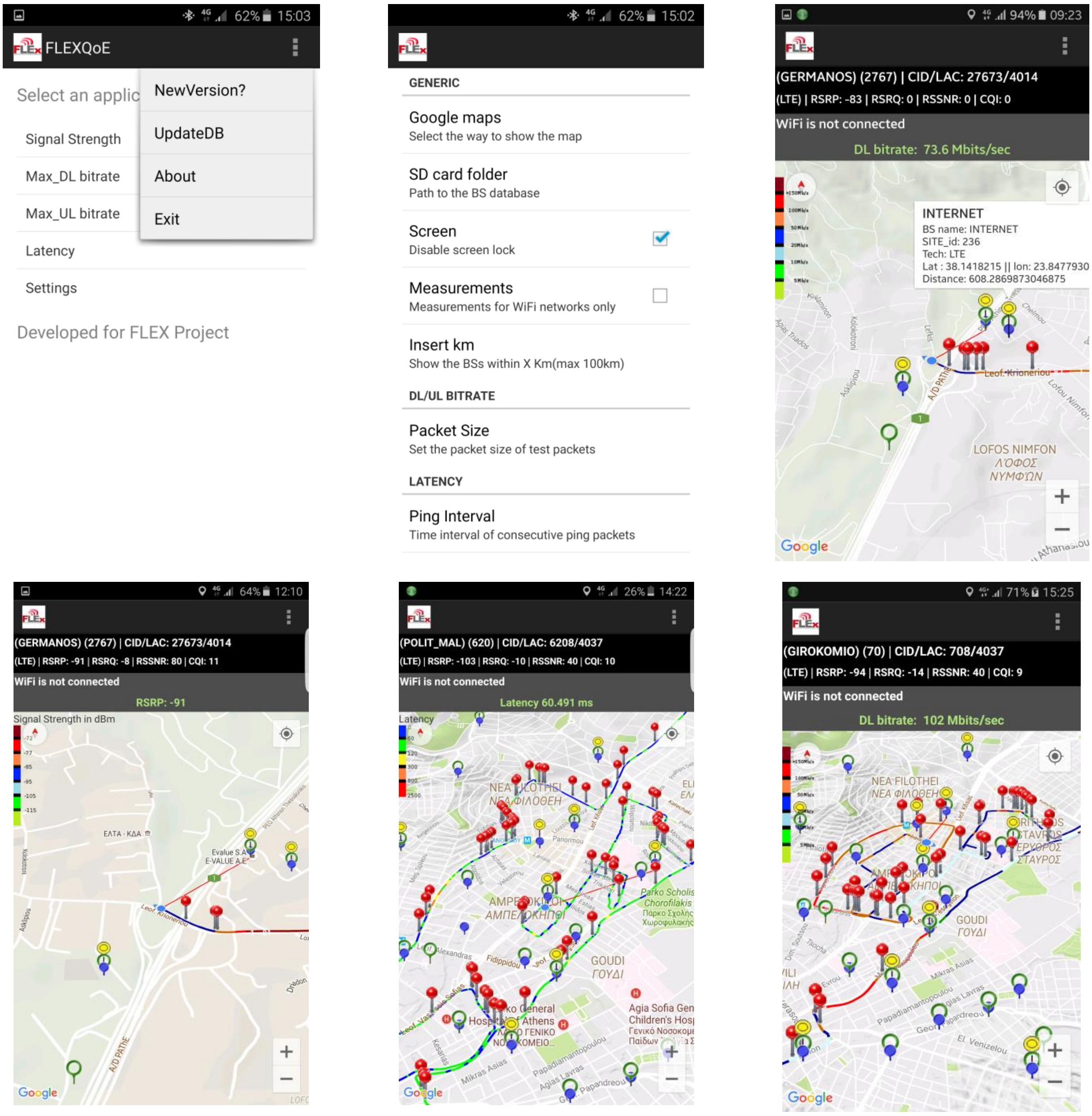
FLEX_QoE

The main features/capabilities of the FLEX_QoE tool are the following:

1. Presentation of 2G/3G/HSPA/HSPA+/4G/4G+ (and WiFi) network related information (incl. BSs locations/capabilities/name, cell reselections/locations info, handover locations/info, etc.) in real time, over Google Maps
2. Collection, presentation and uploading (to dedicated servers) of QoE related measurements in real time, such as signal strength (RSSI, RSRP, RSSNR, RSRQ, etc.), latency, maximum download bitrate and maximum upload bitrate (in both numeric and graphical formats), both indoors and outdoors for both cellular and WiFi technologies
3. Colored track/route (while on the move) depiction based on predefined measurement ranges (e.g. RSRP: from -85 to -100 dBm)
4. Line connecting the user's current location with the serving BS; for easy identification of the location/name of the serving BS
5. Secure transfer of information between the terminal and the server/databases
6. Depiction on terminal screen (over Google) maps of past measurements
7. Push notifications (logging started, measurements were uploaded successfully, etc.).
8. In-app application version update and BSs information update.
9. Depiction of collected measurements via a user-friendly web environment (WebGUI).

A video of the FLEX_QoE tool can be found here:
https://drive.google.com/file/d/0B9U5IjOBumnBbmYtcHRDWC1nQWM/view?usp=sharing,

while some indicative snapshots are depicted in Figure 7-1, Figure 7-2 and Figure 7-3.



GENERIC

Google maps
Select the way to show the map

SD card folder
Path to the BS database

Screen ☑
Disable screen lock

Measurements □
Measurements for WiFi networks only

Insert km
Show the BSs within X Km(max 100km)

DL/UL BITRATE

Packet Size
Set the packet size of test packets

LATENCY

Ping Interval
Time interval of consecutive ping packets

(GERMANOS) (2767) | CID/LAC: 27673/4014
(LTE) | RSRP: -83 | RSRQ: 0 | RSSNR: 0 | CQI: 0
WiFi is not connected
DL bitrate: 73.6 Mbits/sec

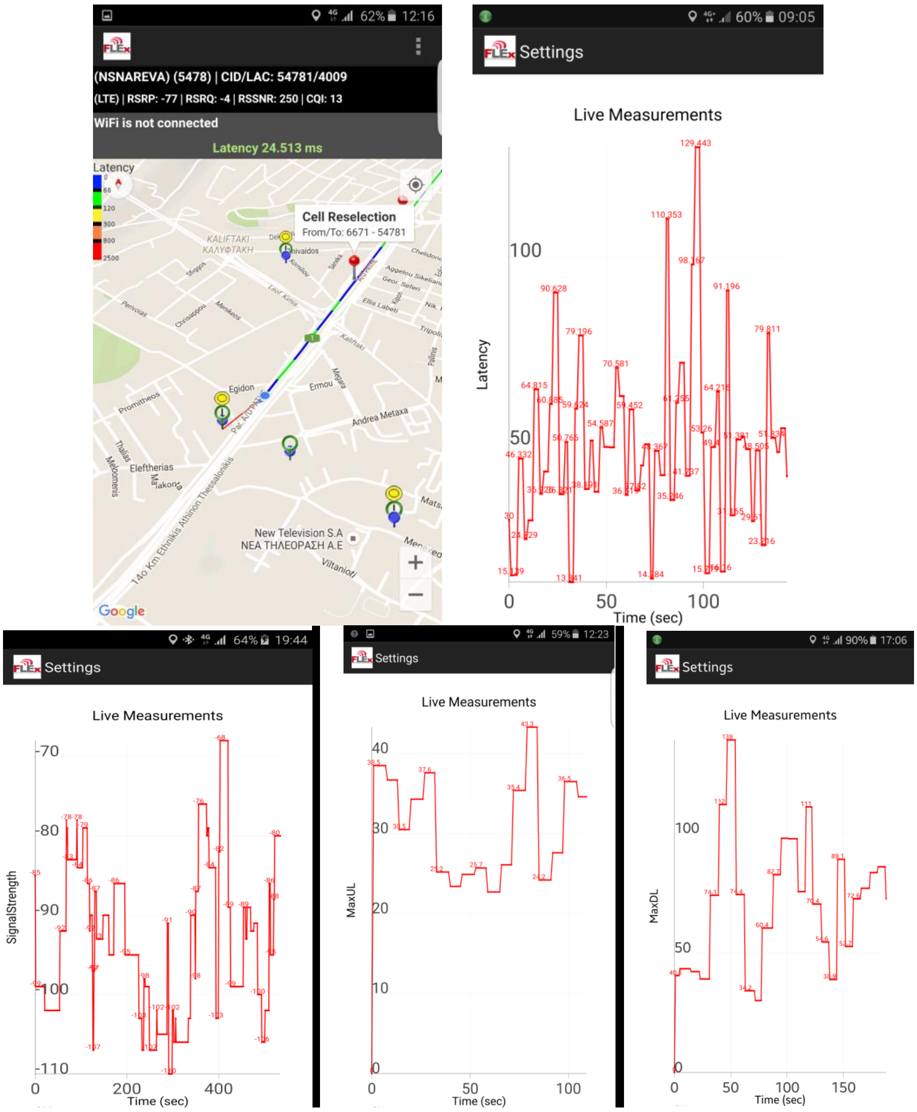
INTERNET
BS name: INTERNET
SITE_id: 236
Tech: LTE
Lat : 38.1418215 || lon: 23.8477930
Distance: 608.2869873046875

(GERMANOS) (2767) | CID/LAC: 27673/4014
(LTE) | RSRP: -91 | RSRQ: -8 | RSSNR: 80 | CQI: 11
WiFi is not connected
RSRP: -91

(POLIT_MAL) (620) | CID/LAC: 6208/4037
(LTE) | RSRP: -103 | RSRQ: -10 | RSSNR: 40 | CQI: 10
WiFi is not connected
Latency 60.491 ms

(GIROKOMIO) (70) | CID/LAC: 708/4037
(LTE) | RSRP: -94 | RSRQ: -14 | RSSNR: 40 | CQI: 9
WiFi is not connected
DL bitrate: 102 Mbits/sec

Figure 7-1: Snapshots of the FLEX_QoE tool.



(NSNAREVA) (5478) | CID/LAC: 54781/4009

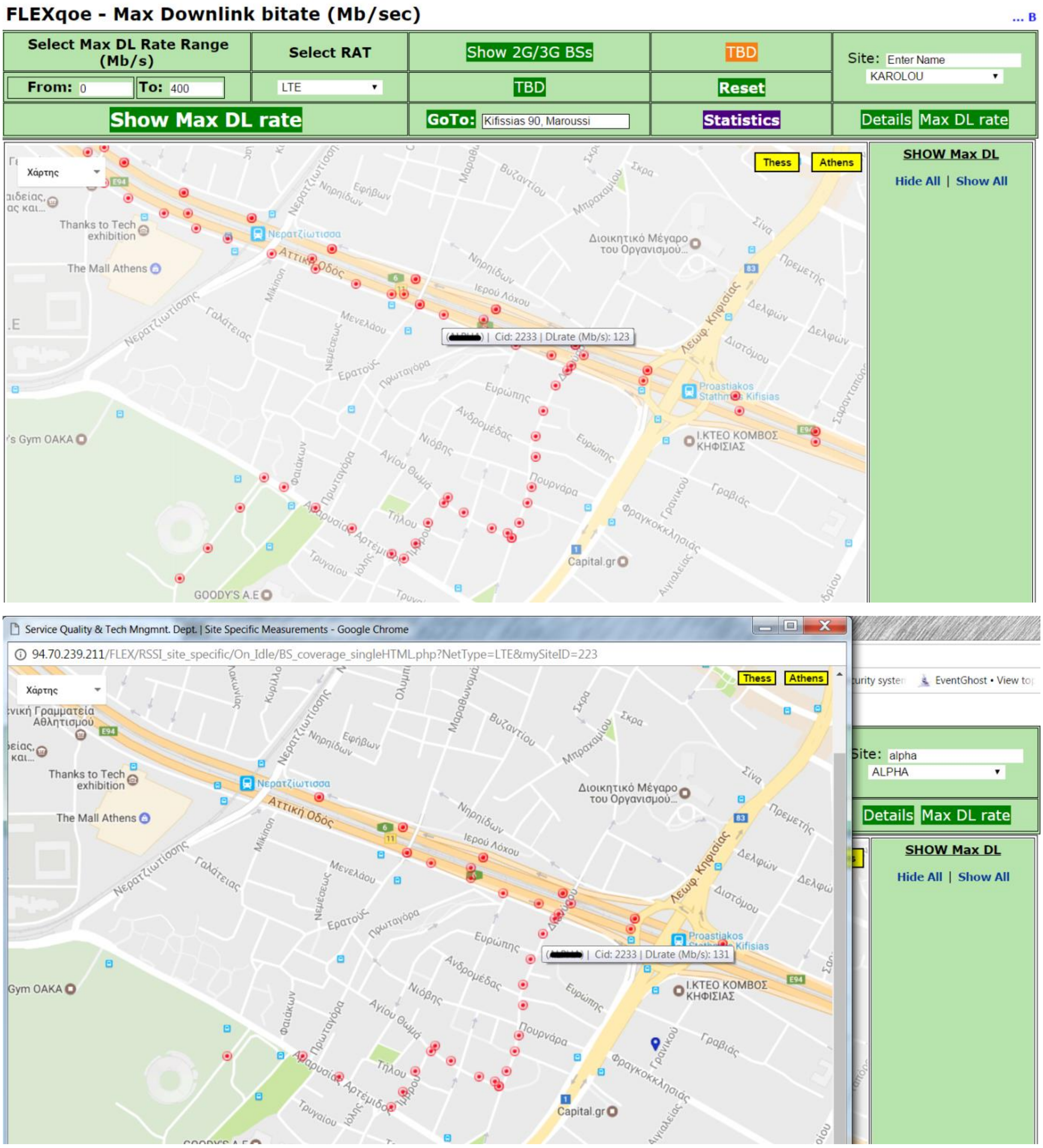
(LTE) | RSRP: -77 | RSRQ: -4 | RSSNR: 250 | CQI: 13

WiFi is not connected

Latency 24.513 ms

Cell Reselection
From/To: 6671 - 54781

Figure 7-2: Indicative snapshots of the FLEX QoE tool (@smartphone).



FLEXqoe - Max Downlink bitate (Mb/sec)

Select Max DL Rate Range (Mb/s)

From: 0 To: 400

Select RAT: LTE

Site: Enter Name KAROLOU

GoTo: Kifissias 90, Maroussi

[redacted] | Cid: 2233 | DLrate (Mb/s): 123

Service Quality & Tech Mngmnt. Dept. | Site Specific Measurements - Google Chrome

94.70.239.211/FLEX/RSSI_site_specific/On_Idle/BS_coverage_singleHTML.php?NetType=LTE&mySiteID=223

Site: alpha ALPHA

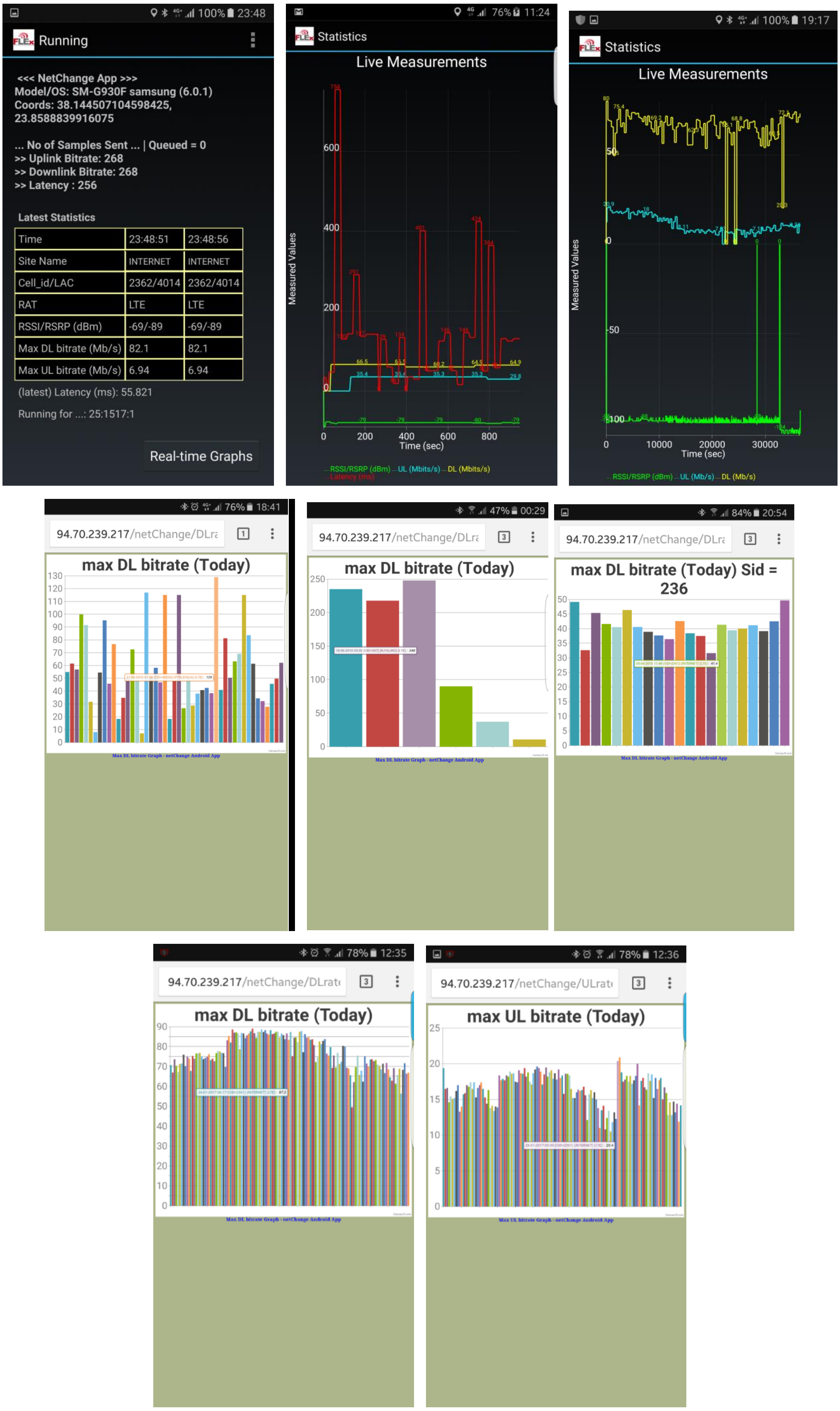
[redacted] | Cid: 2233 | DLrate (Mb/s): 131

Figure 7-3: Indicative snapshots of the WebGUIs (all measurements, Base Station specific ones).

FLEX_netchanges

The aim of this tool is to (automatically) collect, depict and upload network performance related info, such as signal strength (RSSI, RSRP, RSSNR, RSRQ, CQI), latency, maximum download bitrate and maximum upload bitrate periodically (e.g., every X minutes).

The FLEX_netChanges App serves as “real-time” network probes, offering the user/ Mobile Network Operator (MNO) immediate info on network performance e.g. in cases of Self-Organized Networks, network changes, etc. The data gathering can be visualized via a WebGUI. The application could be: (1) deployed by COSMOTE (or other NatCO), on own terminals distributed at specific locations -terminal operation could be remotely controlled and/or (2) offered as a commercial app (running in background- no mobile UI is required in this case).



<<< NetChange App >>>
Model/OS: SM-G930F samsung (6.0.1)
Coords: 38.144507104598425,
23.8588839916075

... No of Samples Sent ... | Queued = 0
>> Uplink Bitrate: 268
>> Downlink Bitrate: 268
>> Latency : 256

Latest Statistics

Time	23:48:51	23:48:56
Site Name	INTERNET	INTERNET
Cell_id/LAC	2362/4014	2362/4014
RAT	LTE	LTE
RSSI/RSRP (dBm)	-69/-89	-69/-89
Max DL bitrate (Mb/s)	82.1	82.1
Max UL bitrate (Mb/s)	6.94	6.94

(latest) Latency (ms): 55.821

Running for ...: 25:1517:1

100% 23:48
Running
Real-time Graphs
76% 11:24
Statistics
Live Measurements
Measured Values
Time (sec)
RSSI/RSRP (dBm) UL (Mbits/s) DL (Mbits/s)
Latency (ms)
100% 19:17
Statistics
Live Measurements
Measured Values
Time (sec)
RSSI/RSRP (dBm) UL (Mb/s) DL (Mb/s)
76% 18:41
94.70.239.217/netChange/DLra
max DL bitrate (Today)
Max DL bitrate Graph - netChange Android App
47% 00:29
94.70.239.217/netChange/DLra
max DL bitrate (Today)
Max DL bitrate Graph - netChange Android App
84% 20:54
94.70.239.217/netChange/DLra
max DL bitrate (Today) Sid = 236
Max DL bitrate Graph - netChange Android App
78% 12:35
94.70.239.217/netChange/DLrate
max DL bitrate (Today)
Max DL bitrate Graph - netChange Android App
78% 12:36
94.70.239.217/netChange/ULrate
max UL bitrate (Today)
Max UL bitrate Graph - netChange Android App

Figure 7-4: Indicative snapshots of the FLEX_netchanges tool.

Indicative snapshots of the FLEX netchanges tool are depicted in **Figure 7-4**, while a video of the FLEX_netchanges tool can be found under the link:

https://drive.google.com/file/d/0B9U5IjOBumnBa3VYclV3ZThRWFE/view?usp=sharing.

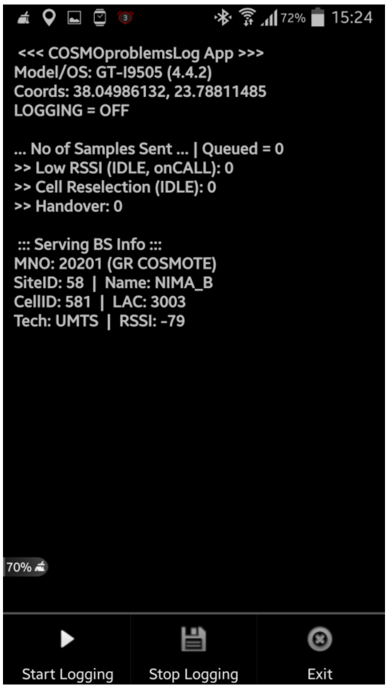
FLEX_Problems

The aim of the FLEX_problems tool is to notify the MNO, in real-time, on network issues/problems (e.g., areas exhibiting: huge number of cell reselections, poor coverage, no coverage, high number of handover failures). The client application runs on Android devices and could start either at power on or manually. The application could be: (1) utilized by OTE Group staff (mobile UI is required in this case) and/or (2) offered by COSMOTE as a commercial application (running in background- no mobile UI required). In either case, a graphical environment (WebGUI) shall be made available to the MNO so as to be informed on those network events.

More specifically, the basic features of the FLEX_problems client App are the following:

- It presents at terminal screen 2G/3G/4G network-related info (BS name, BS-id, RAT, cell-id, LAT/TAC, RSSI/RSRP, RSRQ, etc.)
- It "listens" to the environment (2G/3G/4G) continuously and the terminal status (offhook, busy)
- In case of an event (cell change on idle, handover, low-RSSI) it uploads the relevant measurements to a dedicated server in real-time.
- If the network is not available (handover failure, no coverage), it queues the "measurements" and uploads them (automatically) upon "network recovery"
- It presents at terminal screen info, in real-time, regarding the number of cell reselections, handovers, poor coverage location identified, along with the number of queued messages (if any).

Indicative snapshots of the FLEX problems tool are depicted in Figure 7-5.



```
 <<< COSMOproblemsLog App >>>
Model/OS: GT-I9505 (4.4.2)
Coords: 38.04986132, 23.78811485
LOGGING = OFF

... No of Samples Sent ... | Queued = 0
>> Low RSSI (IDLE, onCALL): 0
>> Cell Reselection (IDLE): 0
>> Handover: 0

 ::: Serving BS Info :::
MNO: 20201 (GR COSMOTE)
SiteID: 58  |  Name: NIMA_B
CellID: 581  |  LAC: 3003
Tech: UMTS  |  RSSI: -79
```

Figure 7-5: Indicative snapshots of the FLEX_problems tool.

WebGUI (under development)

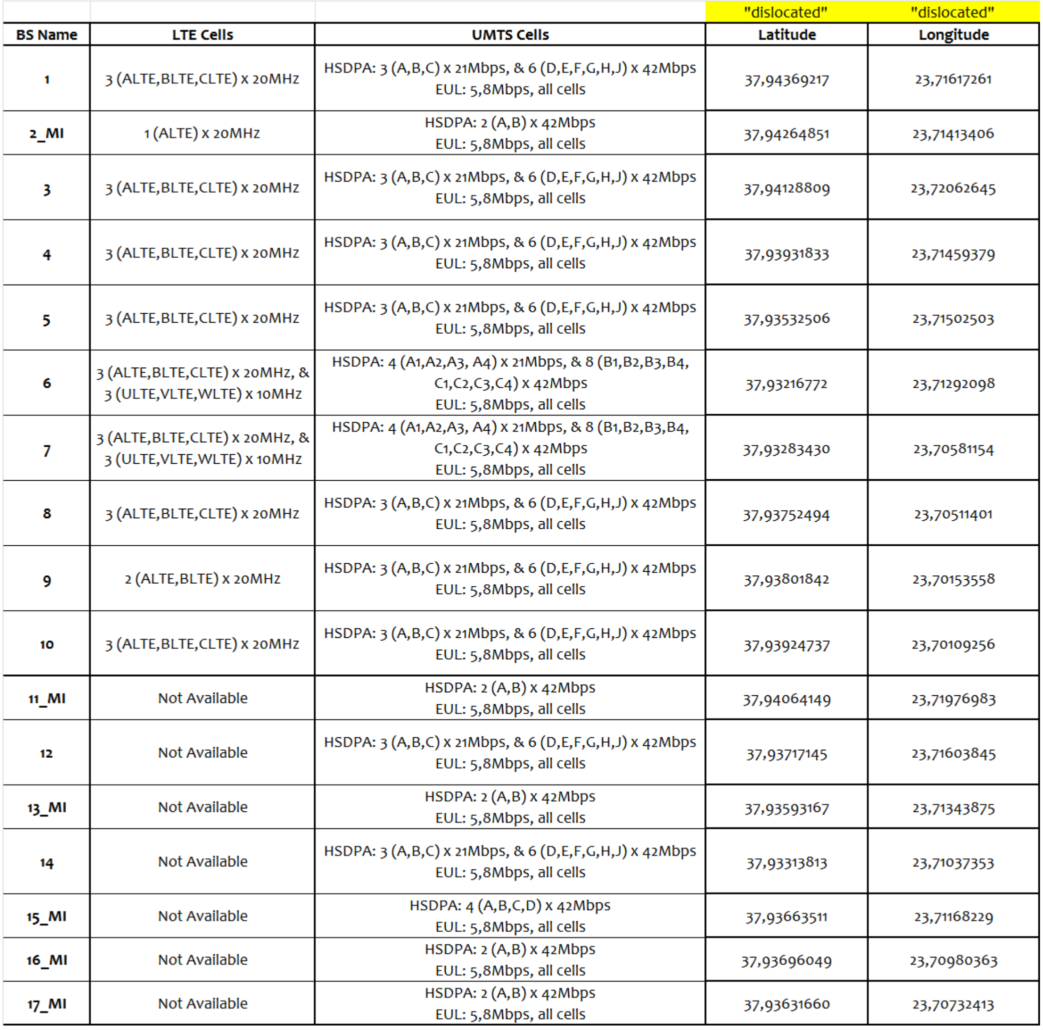
Through the Graphical Environment (see Figure 2), the user may select to depict:

- All Measurements (signal strength or latency or Max. upload and download bitrates)
- Measurements for a specific range(s) (e.g., -66 … -75 dBm)

- Measurements collected for a specific time period (from date … to date)
- Measurements collected for a specific IMEI/terminal/route
- The Serving BS at each measurement location (“linked-line” between measurement and Serving BS)
- Details on each measurement collected (e.g., serving BS, measured value, Cell-ID, LAC, date/time, coordinates)
- Cell Reselections, handovers, no 3G/2G coverage locations (+info)
- 2G/3G/4G BS/eNB locations & relevant info (Site name, SiteID, Coordinates, HSPA capability, etc.)
- 3G Site, Cell-id, LAC specific “coverage”
- Statistics e.g., #samples/range and #samples/technology, etc.

7.3 Annex 3: List of Base Stations covering the geographical area under study

Table 7-4: List of Base Stations covering the geographical area under study, along with their capabilities.



			"dislocated"	"dislocated"
BS Name	**LTE Cells**	**UMTS Cells**	**Latitude**	**Longitude**
1	3 (ALTE,BLTE,CLTE) x 20MHz	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,94369217	23,71617261
2_MI	1 (ALTE) x 20MHz	HSDPA: 2 (A,B) x 42Mbps EUL: 5,8Mbps, all cells	37,94264851	23,71413406
3	3 (ALTE,BLTE,CLTE) x 20MHz	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,94128809	23,72062645
4	3 (ALTE,BLTE,CLTE) x 20MHz	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,93931833	23,71459379
5	3 (ALTE,BLTE,CLTE) x 20MHz	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,93532506	23,71502503
6	3 (ALTE,BLTE,CLTE) x 20MHz, & 3 (ULTE,VLTE,WLTE) x 10MHz	HSDPA: 4 (A1,A2,A3, A4) x 21Mbps, & 8 (B1,B2,B3,B4, C1,C2,C3,C4) x 42Mbps EUL: 5,8Mbps, all cells	37,93216772	23,71292098
7	3 (ALTE,BLTE,CLTE) x 20MHz, & 3 (ULTE,VLTE,WLTE) x 10MHz	HSDPA: 4 (A1,A2,A3, A4) x 21Mbps, & 8 (B1,B2,B3,B4, C1,C2,C3,C4) x 42Mbps EUL: 5,8Mbps, all cells	37,93283430	23,70581154
8	3 (ALTE,BLTE,CLTE) x 20MHz	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,93752494	23,70511401
9	2 (ALTE,BLTE) x 20MHz	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,93801842	23,70153558
10	3 (ALTE,BLTE,CLTE) x 20MHz	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,93924737	23,70109256
11_MI	Not Available	HSDPA: 2 (A,B) x 42Mbps EUL: 5,8Mbps, all cells	37,94064149	23,71976983
12	Not Available	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,93717145	23,71603845
13_MI	Not Available	HSDPA: 2 (A,B) x 42Mbps EUL: 5,8Mbps, all cells	37,93593167	23,71343875
14	Not Available	HSDPA: 3 (A,B,C) x 21Mbps, & 6 (D,E,F,G,H,J) x 42Mbps EUL: 5,8Mbps, all cells	37,93313813	23,71037353
15_MI	Not Available	HSDPA: 4 (A,B,C,D) x 42Mbps EUL: 5,8Mbps, all cells	37,93663511	23,71168229
16_MI	Not Available	HSDPA: 2 (A,B) x 42Mbps EUL: 5,8Mbps, all cells	37,93696049	23,70980363
17_MI	Not Available	HSDPA: 2 (A,B) x 42Mbps EUL: 5,8Mbps, all cells	37,93631660	23,70732413

Notes:

(1) MI - means that the BS configuration corresponds to a micro BS.

(2) Statistics are provided per BS cell the naming of which follows the notation "BS_NAMECell_Name", e.g. 10A for UMTS cell A of BS 10, 2_MIALTE for the LTE cell ALTE of BS 2_MI etc.

(3) Latitude and Longitude information is dislocated from original ones, however the relative distances are kept, actually providing the network topology.

8 REFERENCES

[1] 5G-XHaul, D2.1: “Requirements Specification and KPIs Document”, 5G-XHaul Project, November 2015.
[2] 5G-XHaul, D2.2: “System Architecture Definition”, 5G-XHaul Project, June 2016.
[3] 5G-XHaul, D3.1: “Analysis of state of the art on scalable control plane design and techniques for user mobility awareness. Definition of 5G-XHaul control plane requirements”, July 2016.
[4] 5G-XHaul, D4.6 “Real time implementation of advanced MIMO algorithms in BWT DP1 platform”, September 2016.
[5] 5G-XHaul, D4.7: “Test results from field trial, real time algorithms in BWT DP1 platform”, December 2016.
[6] 5G-XHaul, D4.13: “Synchronization and Localization for Cooperative Communications”, April 2017.
[7] 5G-XHaul, D4.11: “Wireless backhauling using Sub-6 GHz systems”, December 2016.
[8] 5GPPP Architecture Working Group, View on 5G Architecture, June 2016. Online. Available here: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-For-public-consultation.pdf
[9] TR-521 SDN Architecture. Issue 1.1. ONF Foundation, February 2016
[10] A. Aguado, V. López, J. Marhuenda, Ó. González de Dios and J. P. Fernández-Palacios: ABNO: a feasible SDN approach for multi-vendor IP and optical networks , in Journal of Optical Communications and Networking, February 2015, Vol. 7, Iss. 2, pp. A356–A362.
[11] “Global transport SDN prototype demonstration,” OIF / ONF White Paper, 2014.
[12] “OpenFlow switch specification v1.5.1,” April 2015. [Online]. Available: https://www.opennetworking.org/sdn-resources/technical-library
[13] Q. Wei, D. Perez-Caparros, A. Hecker, "Dynamic Flow Rules in Software Defined Networks", European Workshop on Software Defined Networks, EWSDN 2016
[14] “ONOS Blackbird performance evaluation.” [Online]. Available: https://wiki.onosproject.org/display/ONOS/Master-Performance+and+Scale-out
[15] X. An, D. Perez-Caparros, and Q. Wei, “Consistent Route Update in Software-Defined Networks,” 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, 2014, pp. 84-89.
[16] D. Huerfano, I. Demirkol, P. Legg, "Joint Optimization of Path Selection and Link Scheduling for Millimeter Wave Transport Networks," ICC 2017, FlexNets Workshop, Paris, France, May 2017.
[17] A. Betzler, D.Camps-Mur, E.Garcia-Villegas, I. Demirkol, F. Quer and JJ Aleixendri, “SODALITE: SDN Wireless Backhauling for Dense Networks of 4G/5G Small Cells”, IEEE Transactions on Mobile Computing (under submission)
[18] D. Giatsios, K. Choumas, P. Flegkas, T. Korakis, D. Camps-Mur SDN Implementation of Slicing and Fast Failover in 5G Transport Networks Accepted at European Conference on Networks and Communications (EuCNC), Oulu, 2017.
[19] Y. Li, E. Pateromichelakis, N. Vucic, J. Luo, W. Xu and G. Caire, "Radio Resource Management Considerations for 5G Millimeter Wave Backhaul and Access Networks," in IEEE Communications Magazine, vol. 55, no. 6, pp. 86-92, 2017.
[20] E. Pateromichelakis, M. Shariat, A. Ul Quddus and R. Tafazolli, "Joint routing and scheduling in dense small cell networks using 60 GHz backhaul," 2015 IEEE International Conference on Communication Workshop (ICCW), London, 2015, pp. 2732-2737
[21] I. Gribkovskaia, G. Laporte and A. Shyshou. "The single vehicle routing problem with deliveries and selective pickups," Computers & Operations Research, vol. 35(9), pp. 2908-2924, 2008
[22] D. Naddef and G. Rinaldi, “Branch-and-cut algorithms for the capacitated VRP”, In book: The Vehicle Routing Problem, Society of Industrial and Applied Mathematics, 2001, pp.29-51
[23] L. Tassiulas and A. Ephremides, “Stability Properties of Constrained Queuing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks”, IEEE Tran. on Automatic Control, 37(12):1936–1948, Dec. 1992
[24] A. Maltsev, E. Perahia, R. Maslennikov, A. Lomayev, A. Khoryaev, and A. Sevastyanov, “Path Loss Model Development for TGad Channel Models,” IEEE 802.11-09/0553r1, May 2009
[25] Draves, Richard, Jitendra Padhye, and Brian Zill. ”Routing in multiradio, multi-hop wireless mesh networks.” Proceedings of the 10th annual international conference on Mobile computing and networking. ACM, 2004.
[26] lxc linux containers. http://lxc.sf.net.
[27] Gomez, C., Garcia, D., and Paradells, J. ”An OLSR parameter based study of the performance of real ad-hoc network environments”. In Wireless Conference 2005-Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European (pp. 1-6). VDE. April 2005.
[28] C. R. Jackson et al., "Demonstration of the Benefits of SDN technology for All-Optical Data Centre

Virtualisation," Proc. OFC (2017).

[29] Open Networking Foundation, “Microwave Information Model” TR-532 v1.0, Dec 2016.

[30] IEEE, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-2016”, 2016.

[31] Open Networking Foundation, “Wireless Transport SDN, Proof of Concept 2, Detailed Report”, June 2016

[32] mmMagic D4.1, “Preliminary radio interface concepts for mm-wave mobile communications”.

[33] Efficient Positioning Method Applicable in Dense Multi User Scenarios IEEE 802.11 WLAN WORKING GROUP SESSIONS (Interim), Warsaw, September 12 - 16, 2016, Poland.

[34] Precise Positioning Using Time of Arrival with Pseudo-Synchronized Anchor Nodes, AZ: DE 10 2016 123 794.9.

[35] Stephens, B., Cox, A., Felter, W., Dixon, C., and Carter, J. ”PAST: Scalable Ethernet for data centers”. In Proceedings of the 8th international conference on Emerging networking experiments and technologies (pp. 49-60). ACM. December 2012.

[36] A. Zubow, S. Zehl, and A. Wolisz, “BIG AP – Seamless Handover in High Performance Enterprise IEEE 802.11 Networks,” in Network Operations and Management Symposium (NOMS), 2016 IEEE, April 2016.

[37] E. Garcia-Villegas, D. Sesto-Castilla, S. Zehl, A. Zubow, A. Betzler, D. Camps-Mur, “SENSEFUL: an SDN-based Joint Access and Backhaul Coordination for Dense Wi-Fi Small Cells” in IWCMC’17.

[38] S. Zehl, A. Zubow, and A. Wolisz, “hMAC: Enabling Hybrid TDMA/CSMA on IEEE 802.11 Hardware,” Telecommunication Networks Group, Technische Universit¨at Berlin, Tech. Rep. TKN-16-004, 2016.

[39] NGMN 5G White Paper, February 2015.

[40] P. Makris, N. Skoutas, C. Skianis, A Survey on Context-Aware Mobile and Wireless Networking: On Networking and Computing Environments’ Integration, IEEE Communication Surveys and Tutorials, 2013.

[41] Dionysis Xenakis, Nikos Passas, Lazaros Merakos, Christos Verikoukis, “Mobility Management for Femtocells in LTE-Advanced: Key Aspects and Survey of Handover Decision Algorithms”, Communications Surveys & Tutorials, IEEE (Volume:16 , Issue: 1), 1st Quarter 2014.

[42] Younghyun Kim, HaneulKo, Sangheon Pack, Wonjun Lee, and Xuemin (Sherman) Shen, “Mobility-Aware Call Admission Control Algorithm With Handoff Queue in Mobile Hotspots”, IEEE Transactions on vehicular technology, Vol. 62, No. 8, October 2013.

[43] ICT-317669-METIS/D1.5 Updated scenarios, requirements and KPIs for 5G mobile and wireless system with recommendations for future investigations, 4.2015.

9 ACRONYMS

Acronym	Description
3GPP	Third Generation Partnership Project
5G	Fifth Generation Networks
5G-PPP	5G Infrastructure Public Private Partnership
ABNO	Application-Based Network Operations
AC	Area Controller
ADC	Analogue-to-Digital Converter
AoA	Angle of Arrival
API	Application Program Interface
ARQ	Automatic Repeat Request
AVC	Attribute Value Change
BER	Bit Error Rate
BB	Baseband
BBU	Baseband Unit
BH	Backhaul
BI	Beacon Interval
BS	Base Station
CAPEX	Capital Expenditures
CBR	Constant Bit Rate
CDF	Cumulative Distribution Function
CIR	Committed Information Rate
CN	Core Network
COP	Control Orchestration Protocol
CP	Cyclic Prefix
CPI	Controller Plane Interfaces
CPRI	Common Public Radio Interface
C-RAN	Cloud Radio Access Network (aka Cloud-RAN)
DAC	Digital-to-Analogue Converter
DC	Data Centre
DFR	Dynamic Flow Rules
DL	Downlink
DRB	Data Radio Bearer
e2e	end-to-end
eMBB	enhanced Mobile Broadband
eNB	Evolved Node B

EPC	Evolved Packet Core
ETN	Edge Transport Node
ETNC	ETN Controller subcomponent
ETT	Expected Transmission Time
FH	Fronthaul
FIFO	First In, First Out
FIB	Forwarding Information Base
FLRR	Fast Local Link Reroute
IATN	Inter-Area Transport Node
JO	Joint Optimisation
JSON	Java Script Object Notation
KPI	Key Performance Indicator
L2SID	Layer 2 Segment ID
LA	Local Agent
LoS	Line-of-Sight
LP	Linear Program
LTE	Long Term Evolution
MAC	Medium Access Control
MANO	Management and Network Orchestration
MCs	Macro Cells
MD-SAL	Model Driven Service Abstraction Layer
MIMO	Multiple-Input Multiple-Output
MME	Mobility Management Entity
mmWave	Millimetre Wave
MNO	Mobile Network Operator
MPLS	Multiprotocol Label Switching
MSE	Mean Squared Error
NBI	North-Bound Interface
NFV	Network Function Virtualisation
NGMN	Next Generation Mobile Networks
NLoS	Non-Line-of-Sight
NMS	Network Management System
NPU	Network Processor Unit
NS	Network Service
ODL	OpenDayLight
ONF	Open Networking Foundation
OPM	Optical Provisioning Manager

OS	Operating System
OTN	Optical Transport Network
P2P	Point-to-Point
PBB	Provider Backbone Bridge
PCM	Path Computation Manager
PIR	Peak Information Rate
PM	Performance Monitoring
PON	Passive Optical Network
QoE	Quality of Experience
QoS	Quality of Service
QoT	Quality of Transmission
RAN	Radio Access Network
RAT	Radio Access Technology
RCA	Resource Control Agent
RCP	Resilient Control Plane
RF	Radio Frequency
SDN	Software Defined Networking
SDR	Software Defined Radio
SINR	Signal to Interference plus Noise Ratio
SLA	Service Level Agreement
SLAE	Sub-Wavelength Lambda Allocation Engine
SLNR	Signal to Leakage plus Noise Ratio
SHPH	Shortest Path Heuristic Algorithm
SNR	Signal-to-Noise Ratio
SRB	Signaling Radio Bearer
STP	Spanning Tree Protocol
TAF	Transport Adaptation Function
TC	Transport Class
TCAM	Ternary Content-Addressable Memory
TDD	Time Division Duplex
TE	Traffic Engineering
TED	Traffic Engineering Database
TEID	Tunnel End Point ID
TM	Topology Manager
TNC	TN Controller subcomponent
ToA	Time of Arrival
ToF	Time of Flight

TSON	Time-Shared Optical Network
TSO	Two-Stage Optimization
TSPEC	Traffic Specification
UE	User Equipment
UL	Uplink
URLLC	Ultra Reliable Low Latency Communications
vDP	Virtual datapath
VIM	Virtualized Infrastructure Manager
VLAN	Virtual Local Area Network
VM	Virtual Machine
VN	Virtual Network
VNF	Virtual Network Function
VNO	Virtual Network Operator
VNP	Virtual Network Provider
WDM	Wavelength Division Multiplexing
WP	Work Package