European Commission | Horizon 2020 European Union funding for Research & Innovation

5G-XHaul

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

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

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 – Version 1.0

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:	D3.1
Document Title:	Analysis of state of the art on scalable control plane design and techniques for user mobility awareness. Definition of 5G-XHaul control plane requirements
Version:	1.0
Delivery Date:	July 1st, 2016
Responsible:	University of Thessaly (**UTH**)
Editor(s):	Paris Flegkas (**UTH**)
Authors:	Daniel Camps (**I2CAT**), Eduard García (**I2CAT-UPC**), Ilker Demirkol (**I2CAT-UPC**), Jesús Gutiérrez (**IHP**), Eckhard Grass (**IHP**), Anna Tzanakaki (**UNIVBRIS-HPN**), Sarvesh Bidkar (**UNIVBRIS-HPN**), Dimitra Simeonidou (**UNIVBRIS-HPN**), Wint Yi Poe (**HWDU**), Xun Xiao (**HWDU**), Qing Wei (**HWDU**), Artur Hecker (**HWDU**), Emmanouil Pateromichelakis (**HWDU**), Panagiotis Spapis (**HWDU**), George Limperopoulos (**COS**), Eleni Theodoropoulou (**COS**), Constantinos Filis (**COS**), Paris Flegkas (**UTH**), Dimitris Syrivelis (**UTH**), Kostas Choumas (**UTH**), Dimitris Giatsios (**UTH**), Thanasis Korakis (**UTH**), Peter Legg (**BWT**), Jim Zou (**ADVA**), Michael Eiselt (**ADVA**)
Keywords:	Scalable Control plane, SDN, multitenancy, virtualisation, QoS, TE
Status:	Draft
Dissemination Level	**Public** / Confidential
Project URL:	http://www.5g-xhaul-project.eu/

Table of Contents

List of Figures

List of Tables

1 EXECUTIVE SUMMARY

This deliverable provides an analysis of the state of the art on scalable control plane issues for 5G networks and defines the requirements of the 5G-XHaul control plane, presenting also an initial design. The proposed design enables the unified control through well-defined SDN southbound interfaces of a transport network infrastructure composed of heterogeneous technology domains in a scalable way, whilst enabling virtualization and slicing, support for QoS and user mobility awareness.

After providing a state of the art review of relevant 5G control architectures, the high-level requirements of the 5G-XHaul control system are defined and an initial view of the control plane architecture is presented together with the description of the functionality of the components. We then lay out the network virtualisation requirements to be fulfilled by the 5G-XHaul control plane and sketch an initial solution design for supporting multitenancy over the 5G-XHaul trasport network that will be further refined in subsequent WP3 deliverables.

Quality of Service (QoS) support in the data plane technologies considered in the project, namely the optical and wireless is also examined. Previous work on traffic engineering, QoS routing and end-to-end QoS across domains is described. The requirements to be fulfilled by the end to end QoS mechanisms defined in 5G-XHaul are also presented together with some initial design of QoS support in mmWave and in the control plane functionality. Moreover, we focus on the issues of SDN southbound interfaces for sub-6GHz technologies, such as LTE, WiFi and we further describe mmWave technologies. We also describe the southbound protocols for optical transport technologies (i.e. TSON and WDM-PON) and we define the requirements and an initial design proposal for the southbound protocol for candidate technologies in 5G-XHaul.

An analysis of the state of the art of the challenges of SDN when applied to large scale, highly dynamic networks is also presented. The requirements to be satisfied by the proposed 5G-XHaul solution are listed and an initial approach of the hierarchical and distributed control plane is described together with techniques such as rule caching, used to tackle issues of scalability and fast response to changing network conditions. Finally, we highlight the current and envisioned state of the art in RAN-transport interaction and user mobility awareness, focusing on the cases which makes this interaction necessary. We also put forward our requirements and views on this topic, drafting an initial design of our planned interaction with 5G RANs.

The work presented in this deliverable extends the high-level control plane architecture also described in D2.2 by providing detailed description of the functionality of the control-related components that will be developed and evaluated during the project. Their detailed implementation design and evaluation will be documented in D3.2 and D3.3 at the end of the second and third year of the project respectively.

1 INTRODUCTION

A major trend driving the design of future 5G networks is the desire to open the mobile networks to a plurality of industry verticals, instead of having a network focused only in delivering a mobile broadband service as it has been the case for the past generations. In this regard, the 5G-PPP initiative set up in Europe, has already recognized automotive, energy, health, media and manufacturing as the main verticals to be addressed by future 5G networks [1]. This trend has also been recognized by the NGMN, which has accordingly introduced the concept of network slicing [2].

Network slicing in 5G responds to the need of being able to address multiple vertical sectors in a cost efficient way using a single physical network infrastructure. The alternative, namely deploying purpose-built networks for each vertical is not economically sustainable. Thus, inspired by the advances witnessed in the past years in the area of cloud computing, network slicing leverages the concepts of virtualisation and softwarisation. Virtualisation is a technique that allows to provision abstract resources made of the slice or union of underlying physical resources; being the Virtual Machine (VM) abstraction the most common example of virtualization.

The 5G community is currently studying how resources at the various levels of the network – namely access, transport and core – can be virtualized. In the context of 5G, virtual resources are referred to as Virtual Network Functions (VNFs). Softwarization is related to the management of these virtual resources, which, in order to reduce operational costs, need to be managed in an automatic way. In addition, softwarization should provide the operator with the ability to easily compose and deploy new network services, which could be for example instantiated through different network slices.

Virtualisation and softwarisation will shape the architecture of future 5G networks, as recognized by the 5G-PPP Architecture Working Group in [3]. In particular, the virtualisation and softwarisation trends will also affect the design of the future 5G transport networks, namely the networks connecting the 5G access and core network (CN). In addition to the general requirement to support slicing, future 5G transport networks also have to address specific requirements such as the cost effective transport of the fronthaul (FH) and backhaul (BH) interfaces required to support centralized (C-RAN) and distributed RAN deployments. In addition, future 5G transport networks will have to enable novel functional splits, which can provide a better trade-off between the pooling and coordination gains offered by C-RAN architectures, and the light transport requirements imposed by distributed RAN architectures. Examples of these novel functional splits have been discussed in [4].

This deliverable provides an analysis of the state of the art (SotA) on scalable control plane issues for 5G networks and defines the requirements of the 5G-XHaul control plane, presenting also an initial design. The proposed design enables the unified control through well-defined SDN southbound interfaces of a transport network infrastructure composed of heterogeneous technology domains in a scalable way, whilst enabling virtualisation and slicing, support for QoS and user mobility awareness. The work presented in this deliverable extends the high-level control plane architecture also described in deliverable D2.2 by providing detailed description of the functionality of the control-related components that will be developed and evaluated during the project. Their detailed implementation design and evaluation will be documented in deliverables D3.2 and D3.3 at the end of the second and third year of the project respectively.

1.1 Organisation of the document

This deliverable is structured in eight main sections, plus references and list of acronyms.

Beyond this introduction, Section 2 presents the current state of the art on SDN architectures for 5G networks, including works from relevant scientific conferences and journals as well as ongoing work from European projects under the 5G-PPP umbrella. The section also lists the high-level requirements that the 5G-XHaul system should satsify and presents an initial view of the control plane architecture. The control plane components, their functionality and interactions are also described, thus providing a quite detailed view of the 5G-XHaul control plane system to be developed in the project.

Section 3 presents an analysis of the SotA on virtualisation and multi-tenancy techniques applied in data centres which provide the foundations of the proposed 5G-XHaul solution. We then lay out the network virtualisation requirements to be fulfilled by the 5G-XHaul control plane and sketch an initial solution design that will be further refined in subsequent WP3 deliverables.

Section 4 provides a review of the current work on QoS support for all data plane technologies considered in the project, namely optical and wireless. Moreover, QoS support in the control plane is also presented by describing the previous work on traffic engineering, QoS routing and end-to-end QoS across domains. The requirements to be fulfilled by the end-to-end QoS mechanisms defined in 5G-XHaul are also presented together with some initial design of QoS support in mmWave and in the control plane functionality.

Section 5 looks at the state of the art on available SDN control and southbound interfaces of the transport technologies used in mobile/wireless and optical networks. We focus initially on the issues for Sub-6 GHz technologies, such as LTE, WiFi and we further describe mmWave technologies. We also describe the southbound protocols for optical transport technologies (i.e. TSON and WDM-PON) and we conclude the section with the requirements and an initial design proposal for the southbound protocol for candidate technologies in 5G-XHaul.

Section 6 presents an analysis of the SotA of the challenges of SDN when applied to large scale, highly dynamic networks, such as a 5G-XHaul transport network across different domains (optical/wireless) and areas. The requirements to be satisfied by the proposed 5G-XHaul solution are listed and an initial approach of the hierarchical and distributed control plane is described together with techniques such as rule caching used to tackle issues of scalability and fast response to changing network conditions.

Section 7 highlights the current and envisioned SotA of the interaction between the mobile network and the transport network, as well as on user mobility awareness, focusing on the cases that make this interaction necessary. We also put forward our requirements and views on this topic, drafting an initial design of our planned interaction with 5G mobile networks.

Finally, Section 8 provides a summary and the main conclusions of this deliverable.

2 HIGH LEVEL 5G-XHAUL CONTROL PLANE ARCHITECTURE

In this section, the state of the art on SDN control plane architectures for 5G networks is provided, describing related work found in the literature as well as relevant SDN architectures adopted by EU 5G-PPP projects. Subsequently, the requirements and initial design of the 5G-XHaul architecture for the control plane are described.

2.1 State of the Art on SDN Control Plane Architectures for 5G

In [1], an SDN controller adjusts the bandwidth dynamically for each radio access point (RAP) to baseband unit (BBU). Hence, the SDN controller provides flexible management and router selection for all radio access network (RAN) to core network connections, where the SDN controller comprises two main parts: the unified control entity (UCE) and the unified data gateway (UDW). The role of UCE is to define control rules for the mobility management entity (MME), the service gateway control plane (SGW-C), and the packet data network gateway control plane (PGW-C). The UDW determines the rules for data forwarding of the service gateway data plane (SGW-D) and the packet data network gateway data plane (PGW-D). Meanwhile, the paper explores different network architectures and techniques which could be exploited in future 5G systems such as non-orthogonal multiple access (NOMA), massive multiple input and multiple output (MIMO), cooperative communications and network coding, full duplex (FD), device-to-device (D2D) communications, millimeter wave communications, automated network organization, cognitive radio (CR), and green communications.

The work described in [8], proposes a new architecture for leveraging SDN for 5G wireless networks, called SoftAir. The main goal in this paper is to exploit the benefits of using cloudification and virtualization to provide a scalable, flexible and more resilient network architecture. At the proposed architecture, the control plane provides the management and optimization tools for the data plane, which consists of software-defined base stations (SDBSs) in the RAN and software-defined switches (SD-switches) in the BH. The controller serves physical, MAC, and network functions on computers and remote data centers. The main contribution of the proposed SoftAir architecture can be categorised in five main properties. First, programmability, such that SDN nodes (SD-BSs and SD-switches) can be reprogrammed dynamically allocating different network resources. Second, cooperativeness, such that SDN nodes can be implemented and linked at data centers for joint control and optimization for improving the general network performance. Third, virtualizability, such that several virtual wireless networks can be implemented on a single SoftAir platform, while each network operates according to its own protocols and interacts with its allocated network resources without interfering with the other service providers. Fourth, openness, such that data plane elements (SD-BSs and SD-switches), have common data/control interface protocols, regardless of the different data forwarding technologies provided by different vendors, such as CPRI and OpenFlow, so that the data plane monitoring and management can be simplified. And finally, fifth, visibility, where controllers are able to have an overall view over the whole network collected from data plane elements.

[9] proposes a new multi-tiered cloud controller scheme and event processing mechanism for Software Defined Wireless Network (SDWN) architecture of the 5G network that results in a user-centric and service-oriented architecture. To provide the proper radio environment required for 5G wireless communication, SDN along with NFV are used to guarantee the isolation of heterogeneous radio access networks such as LTE, Wi-Fi and W-CDMA. The main contribution of this paper is to provide a ubiquitous cloud radio access for 5G wireless networks, enabling the coexistence of other heterogeneous wireless networks by designing the proposed multi-tiered cloud controller. Another significant contribution in this paper is the design of a layered cloud scheme for the multiple controllers with two parts: the Edge Controller (EC) and the Global Controller (GC). The main logic behind this design is to reduce the response latency and balance the network load for the cloud of the controllers. The EC processes events within a single RAN domain, and the GC takes events across various RAN domains into account. The proposed architecture has two main processing blocks: on-line transaction processing (OLTP) block and on-line analytical processing (OLAP) block. The OLTP block works on the low-level events and real-time statistics, such as events related to spectrum access. The OLAP block works on the high-level events and long-time statistics related to network status, such as load balancing mobile devices. The EC stores the statistics of all the access points belonging to one RAN into a monitoring server database. Second, according to the event type and the required QoE, the EC delivers the event to the OLTP and OLAP blocks, or forwards the event to the GC. The OLTP and OLAP blocks parse the events using specific algorithms. Third, the EC makes a decision based on the defined policies and

sends the control signal to the responsible component. Furthermore, NFV and virtualization allows to have several virtual operators sharing the same infrastructure and resources. Spectrum can be divided into slices, where each slice is dedicated to a virtual cloud of RANs based on the defined policy. So, clients have access to the Internet regardless of the type of radio access network or network operator being used.

The authors in [12] propose a new scheme to deliver data flows in an intelligent way for 4G networks, which could be extended to 5G networks. This paper proposes to build application-awareness in the transport network, simply by sharing the LTE QoS parameters with SDN controllers used in the transport network. With the LTE QoS parameters, an SDN controller can provide more accurate and consistent end-to-end QoS, which results in an application-aware network. Moreover, in LTE, QoS and QoE factors can be estimated using Deep Packet Inspection (DPI). Layer four through layer seven inspection is provided by DPI, which yields not only context-awareness but also extracting meta data attributes, and other data that is used to calculate networking performance statistics for each flow. Furthermore, the proposed scheme performs a dynamic end-to-end flow control from packet gateways in the 4G CN, e.g. P-GW. In particular, the P-GW sends a request to the SDN controller to obtain the network conditions on each path between the P-GW and each content delivery node. The information can be used at the user terminal to select the best available content server. A tight coupling between the mobile and the transport network is an architectural principle that will be explored in 5G-XHaul.

The work in [13] proposes a new scheme called SoftRAN, which proposes centralized control similar to the SDN architecture [185]. SoftRAN considers all base stations (BSs) in a geographical zone as one virtual aggregated BS, where individual base stations are just radio elements with some control logic. These radio elements are managed by a logically centralized element which makes control plane decisions for all the radio elements in that geographical zone. Hence, this logically centralized entity is called the controller of the huge base station. The controller maintains a whole view of the radio access network and provides a framework on which control algorithms can be implemented.

The paper [14] uses a programmable forwarding plane and an SDN controller in the mobile BH to reduce the delay of data forwarding in the network. Control functions of the MME (Mobility Management Entity) and the S-GW are implemented as northbound applications on this controller. An intercell handover scenario is described as an illustration of the benefits of this approach. In this scenario, without an SDN controller, a substantial amount of request/confirm signaling messages between eNodeB pairs and between eNodeBs and MMEs is required. In contrast, the SDN controller can just push new forwarding rules through OpenFlow messages, avoiding inter-eNodeB control message exchanges, while the selection of the target eNodeB can be based on a set of predefined policies, thus also suppressing eNodeB-MME control messages. It is estimated that using this architecture, the handover delay can be reduced to 2-3 times the round trip delay between the User Equipment (UE) and the SDN controller.

Finally, the CROWD project [22] proposed a two tier SDN architecture to control wireless and BH resources in very dense access networks. Regarding the physical infrastructure, CROWD considers LTE and Wi-Fi in the wireless access and OpenFlow switches in the BH network. The architecture clusters wireless and BH resources at the district level under the control of a CROWD Local Controller (CLC), and multiple CLC controllers are orchestrated by a CROWD Regional Controller (CRC). CLCs are in charge of configuring wireless and BH resources in short time scales, whereas CRCs optimize the network at longer time scales. CRCs also provide well defined APIs towards the applications and the network operators.

2.1.1 5G-PPP EU Projects

In this section we first review the architectural work put forward by the 5G-PPP association, and the architectures of some of the 5G-PPP projects that relate to 5G-XHaul. We conclude the section by positioning the architectural work carried out in 5G-XHaul in relation to other 5GPPP projects.

2.1.1.1 5G-PPP View on 5G Architecture

The 5G-PPP Architecture working group has published a White Paper [3] describing the 5G-PPP views on 5G architecture. Figure 1 highlights the main principles guiding the design of the 5G architecture identified by the 5G-PPP Architecture Working Group, namely: i) integration of heterogeneous technologies, ii) network slicing, in order to support multiple verticals with a single physical network, iii) support for end user and operational services, for example having a converged transport network for BH and FH traffic, iv) having native support for softwarisation, hence virtualising network functions, and effectively providing a “network of functions", and v) integrate communication and computation, which is enabled by a joint SDN and NFV architecture.

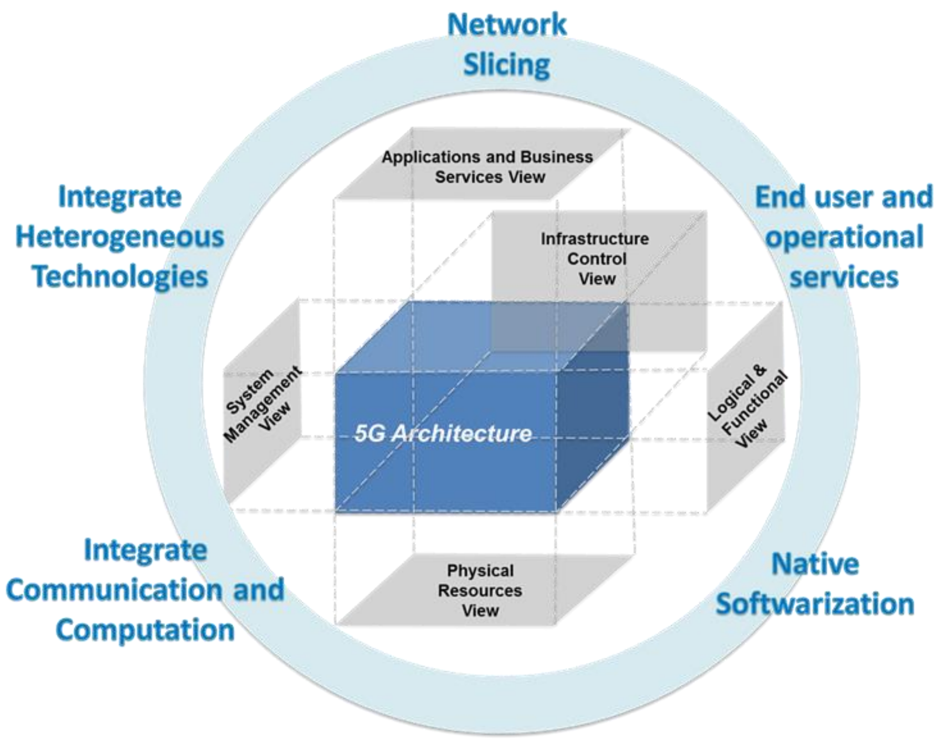

Figure 1. 5G Architecture views from [3].

Next, we will highlight how some 5G-PPP projects[1], and in particular 5G-XHaul, propose to address some of the previous architectural design principles.

2.1.1.2 SELFNET

SELFNET (https://selfnet-5g.eu/) is an EU project that aims at defining an architecture for providing Self-Organizing capabilities over 5G networks based on five differentiated layers with the following logical scope:

[1] The descriptions of the different 5G-PPP projects are derived from contributions that these projects have submitted to various 5G-PPP working groups.

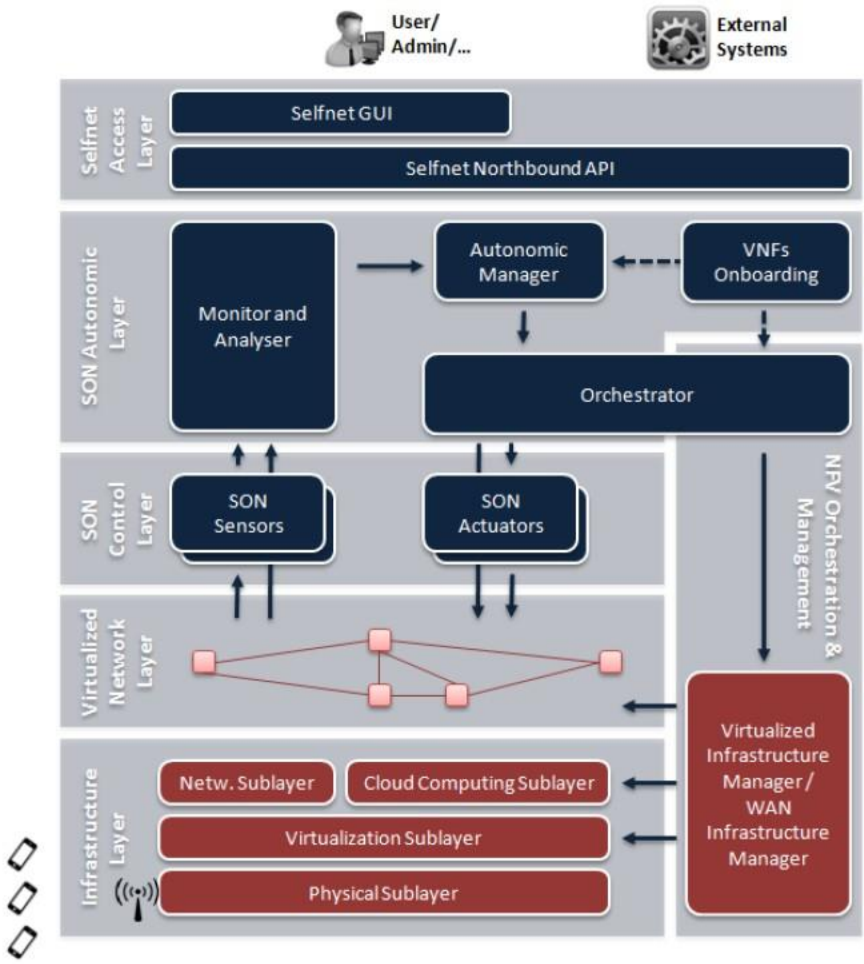

Figure 2: SELFNET overall architecture.

Infrastructure Layer: This layer provides the resources required for the instantiation of virtual functions (Compute, Network and Storage) and supports the mechanisms for that instantiation. All network functions managed autonomously by the SELFNET framework will be based on this infrastructure. To achieve its functionality, this layer encompasses different sublayers. The Physical sublayer provides physical connectivity, networking, and computation and storage capabilities over bare metal. The Virtualization sublayer provides virtualization capabilities to instantiate virtual infrastructures. The Cloud Computing sublayer provides multi-tenancy support to the infrastructure layer together with a centralized point for facilitating the management of the infrastructure.

Virtualized Network Layer: This layer represents the instantiation of the Virtual Networking Infrastructures created by the users of the infrastructure as part of their normal operational plan and those created by the SELFNET framework as part of the SON capabilities. In the context of SELFNET, all Network Functions will be virtual functions and they will be chained across the virtual network topology.

SON Control Layer: This layer contains the applications that will enable the collection of data from sensors deployed through the entire system (SON Sensors) and the applications that will be responsible for enforcing actions into the Network (SON Actuators) as part of the enabling mechanisms to provide network intelligence in 5G networks.

SON Autonomic Layer: This layer provides the mechanisms to provide network intelligence. The layer collects from the network pertinent information about the network behaviour, uses that information to diagnose the network condition, and decides what must be done to accomplish the system goals. It then guarantees the organized enforcement of the actions that are determined.

NFV Orchestration & Management Layer: It is worth emphasizing that the control of the chaining of the Network Function (NF) applications is envisioned as a management functionality to be able to control the topology of the Virtual Network layer depicted in the figure as Network Controller (SDN App) and included logically in the Virtual Infrastructure Manager (VIM) functionalities.

SON Access Layer: This layer encompasses the interface functions that are exposed by the framework. Despite the fact that internal components may have specific interfaces for the particular scope of their functions, these components contribute to a general SON API that exposes all aspects of the autonomic framework, which are "used" by external actors, like Business Support Systems or Operational Support Systems.

A GUI is also provided on top of the SON API where a network administrator can interact and configure SELFNET and also obtain the complete status of the network, acting as a command and control centre. This GUI will also enable the network administrator to stop, verify or manually enforce any of the actions that SELFNET is governing, allowing always network administrators to have control over their infrastructure.

2.1.2 5G-Crosshaul

In the context of 5G-Crosshaul (http://5g-crosshaul.eu/), resources encompass communication technologies, storage and compute and hence are of different nature. The 5G-Crosshaul system architecture is shown in Figure 3. The architecture highlights that Crosshaul specific switching and processing elements, respectively the *Crosshaul Forwarding Elements* (XFE) and the *Crosshaul Processing Unit* (XPU), are the key components of the network fabric at the user plane level. Control and signaling information are routed over the Crosshaul transport network. An *Adaptation Function* (AF) is used to integrate legacy, non-Crosshaul specific, network elements in the new transport network. 5G-Crosshaul resources are controlled by the XCI and are managed by the MANO layer. The MANO is compliant with the *European Telecommunications Standards Institute* (ETSI) reference architecture. In particular, a specific MANO is used for the 5G-Crosshaul transport network, whilst one or more distinct MANOs are in charge of the core and access networks.

The architecture allows decentralizing specific functions in the XPUs and disseminating intelligence throughout the whole transport system. When a *Cloud Radio Access Network* (C-RAN) is virtualized XPUs include some functions of the *Base Band Units* (BBUs). VNFs can be dynamically instantiated in the XPUs elements and can be moved at different locations under the management of the MANO. XFEs instead are multi-technology switches to meet different application constraints and to suit the different scenarios which have been envisaged for 5G.

Applications are used to manage and optimize the network. Resources are managed dynamically and planned without resorting to expensive pre-provisioning, while at the same time energy and mobility of users that may cause the sudden appearance of traffic peaks is part of a multi-objective optimization problem. Specific aspects of the network behavior are optimized by applications such as the *Resource Manager Application* (RMA), *Mobility Management Application* (MMA), *Energy Management and Monitoring Application* (EMMA), *Virtual Infrastructure Manager and Planner Application* (VIMaP), *Content Delivery Network Application* (CDNA) and *TV Broadcasting Application* (TVBA).

An important feature of the 5G-Crosshaul network is to support multi-tenancy. Tenants use and have allocated network, storage and compute resources within the transport system. An important distinction between the tenants depends on the level of control that each tenant has on the respective resources. Depending on the level of control, applications may access the controllers directly or access them via the NFV orchestrator.

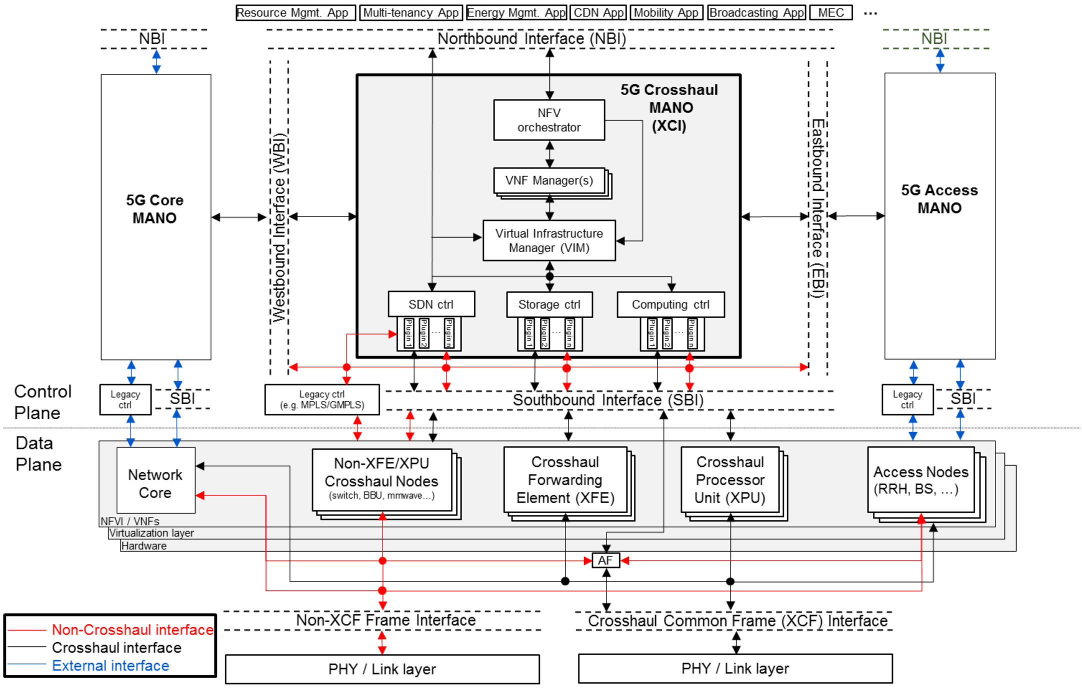

Figure 3: 5G-Crosshaul System architecture.

2.1.3 Superfluidity

The Superfluidity (http://superfluidity.eu/) architecture is the result of the combination of the emerging technologies involved in the project. Figure 4 depicts a high-level view of the overall architecture.

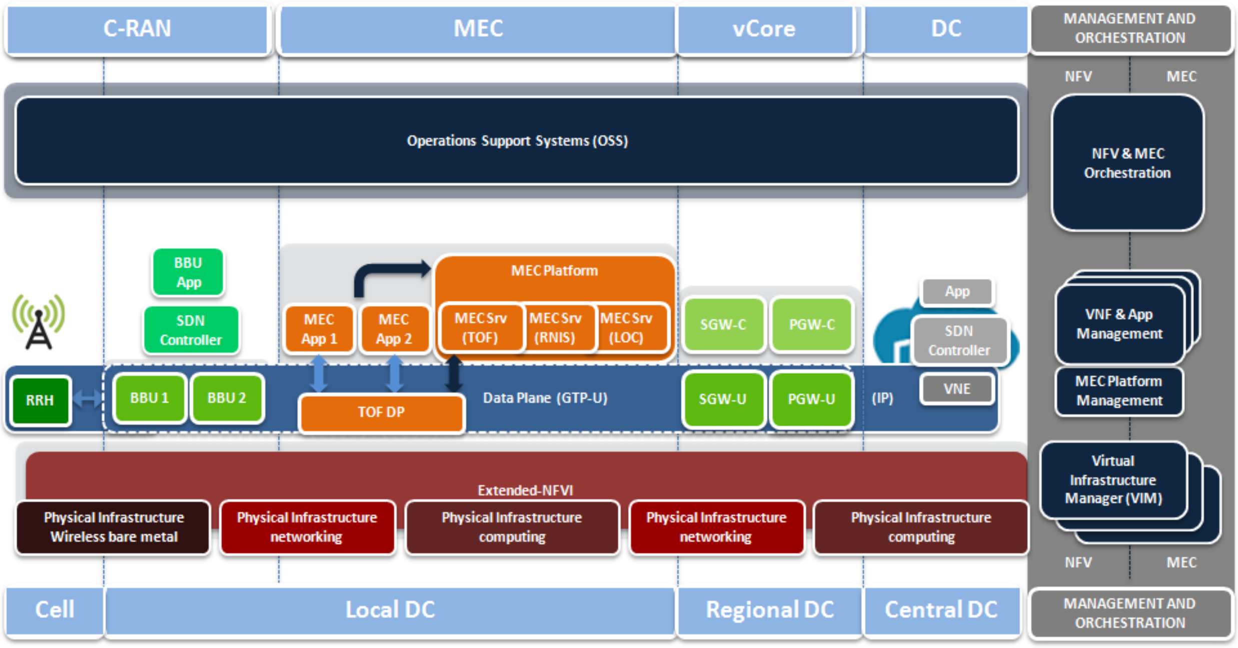

Figure 4: Overall Superfluidity Architecture.

At the top of the figure, the different components involved are depicted (C-RAN, MEC, vCore and DC), while at the bottom the different types of DCs are depicted (Cell-site, Local, Regional and Central). The big OSS blue box represents the traditional operational support systems, which deals with all the components in order to create services for end users.

An Extended-NFVI represents an evolution of the ETSI NFVI concept, considering the additional heterogeneity of its nature (including hardware, hypervisors, and other execution environments), and the federation of DCs at different geographies and different types. This extended-NFVI is common to all components, easing resource management and allowing an agile orchestration of services (superfluid).

Starting from the left, the C-RAN component is split into two blocks, corresponding to the RRH and the BBU parts, the first residing in the cell-site and the second in a local DC (not far away from the RRH). This shows the usual C-RAN view, where multiple approaches can be considered for the functional split among them. The BBU boxes can be implemented as NFVs and also follow the SDN concept, as depicted in the figure, by using an SDN Controller and building a BBU App on top, which control the whole BBU network element operations.

The MEC component appears next and is deployed in the same location (DC) as C-RAN. As C-RAN concentrates in the local DC a large number RANs, this is the right place to run MEC applications (*Note: Other options are also possible, e.g. closer to the core; however, it would result in higher latencies*). The MEC environment uses the same infrastructure as the C-RAN or other VNFs, making the solution very convenient and efficient.

The vCore component is the next component to appear, comprising the central nodes of a mobile network. This Core runs in the common Extended-NFVI, usually located in regional DCs, easing the agility and fluidity when the nodes need to be migrated and/or scaled. The Figure depicts the expected evolution of mobile core networks towards the SDN model, where control plane and data plane components are completely separated, and the former ones fully control the latter ones on data processing tasks. (*Note: The components shown are the ones resulting from splitting the 4G/LTE Core elements*).

The DC component corresponds to the traditional datacenter segment, where a large number of services are deployed. Those are located at central points and deal with a large amount of resources. Beyond traditional services, central DCs also implement NFV and SDN technologies, in order to control the network components inside the DC. For this reason, in this segment it is shown a generic VNE (Virtual Network Element), the SDN controller and a generic App element.

Finally, the right most vertical is the common management and orchestration layer, which is the brain of the whole system. This vertical is an NFV-like set of functions, which allows for an integrated view of everything: network, services, DCs and services, managing the entire system in a superfluid way, taking advantage of a common extended-NFVI and achieving an end user centric view of the ecosystem.

In the bottom of this vertical, a VIM-like set of components take care of the resource management on the different DCs, interacting among them to build a federated environment. In the middle, a set of VNF and App Managers handle the lifecycle details of particular VNFs and Applications. Also, a Platform Manager is used to manage service APIs and applications access to those APIs. Finally, orchestrators are able to orchestrate VNFs to create complex services and take the best decisions for Apps to be deployed and serve customers.

2.1.4 SESAME

The SESAME project (http://www.sesame-h2020-5g-ppp.eu/) proposes an approach, in which the integration of NFV and SDN (see the architecture in Figure 5) is necessary to implement a *Cloud Enabled Small Cell* (CESC) virtualized network. The CESC is a substantial evolution of the Small Cell (SC) concept, in which a multi-operator (i.e. multi-tenancy) enabled approach is coupled with an edge-based, virtualised execution environment. This is implemented by means of a micro-server facility to provide storage and compute capabilities. Multiple small cell network operators, physical and virtual, respectively SCNO and VSCNO, are able to use the shared infrastructure (that includes cloud based resources), each one using its own network 'slice'. The ensemble of SCs and micro-serves contribute to form the 'CESC Cluster.' At least one micro-server is enabled with backhaul connection to reach an operator's Evolved Packet Core (EPC).

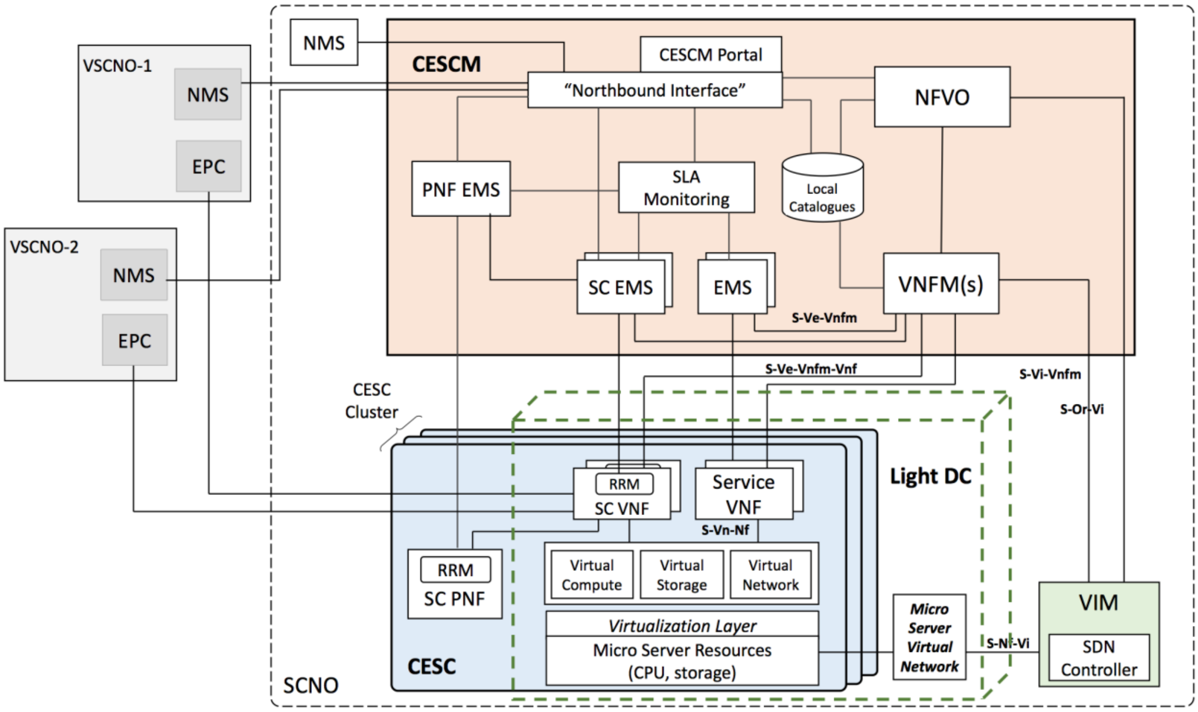

Figure 5: SESAME Architecture.

SDN and NFV allow achieving the adequate level of flexibility and performance isolation in the cloud infrastructure. Small cell network functions and Network Services (NS) are hosted in the Light Data Centre (Light DC), which is the cloud-enabled solution at the mobile network edge. The Light DC is the collection of a potentially large number of micro-servers, which are connected together through an internal distribution network. Therefore, the SC is split into physical and virtual network functions, respectively Small Cell Physical Network Function (SC PNF) and Small Cell Virtual Network Function (SC VNF), with the latter running in the edge cloud. Furthermore, the Light DC provides the NFVI where to deploy virtualised functions. This approach leverages the full potentials of the multi-tenancy environment to support 3GPP Multi-Operator Core Network (MOCN) requirements. On the other hand, the ETSI MANO framework allows placement of VNFs leveraging on the Light DC execution environment to deploy network intelligence and applications near the small cells.

The *CESC Manager* (CESCM) is the central service management and orchestration component in the architecture shown in Figure 5. It is responsible to integrate together the traditional 3GPP network elements management, and novel functional blocks. A single instance of the CESCM is able to operate over several CESC clusters at different PoPs, through a dedicated VIM per cluster. As shown by the figure, the NFV approach enables tenants to flexibly manage both PNFs and VNFs through the respective VNF EMS and PNF EMS. Virtualised network functions are not only restricted to SC VNF but also to others such asvirtual Deep Packet Inspection (vDPI) and vCache, referred to as *service* VNFs. The architecture shown above also makes use of Hardware Accelerators (HWA) for specific VNFs. The lifecycle of the virtualised function is managed by the VNFM.

The NFVO is one key component of the CESCM and provides functionalities for managing and orchestrating VNFs and NSs within the CESC cluster. Besides orchestration of the NFVI resources (i.e. storage, compute and networking), the NFVO is also responsible for managing the VNF Forwarding Graph (VNFFG) (two or more VNFs in the Light DC including PNFs as well) over the resources available in the Light DC. In case of the SESAME system, which is conceived to support multi-tenancy, management tasks are carried out automatically by the NFVO relying on global and local optimisation methods.

In the architecture shown in Figure 5 the VIM lays outside the CESCM and manages the NFVI that is created on top of the physical resources available within the Light DC. The VIM enables a unified view of the Light DC to users (i.e. SCNOs) and higher level tools. Moreover, the CESCM (through the NFVO) can communicate to the VIM (and the SDN controller attached to it) the NS to create and the specific VNF forwarding rules, including the requested connectivity (e.g. virtual paths creation) within the Light DC. The NFVO requests the VIM for available resources where to deploy the VNFs and it subsequently requests VMs creation.

In the system architecture proposed in Figure 5, SC VNFs need to be deployed over all CESCs (one per CESC per tenant) with the required network connectivity. Moreover, for a NS that includes service VNFs over HW accelerators (HWA), accurate deployment selection is crucial, considering simultaneously hardware requirements, radio coverage and Service Level Agreement (SLA) KPIs (e.g. latency). This calls for a more advanced placement process than in the radio agnostic procedure done by the VIM. The problem becomes even more complex with end-user's mobility (i.e. handover). Even for NS without service VNFs on HWA, radio handover forces the NFVO to affect the service placement and migration process. NFVO should react to such a situation whereby a hitless service migration. Although, the idea of service migration is not new, customization to the 5G network and radio requirements can be addressed with the system architecture in Figure 5.

Additional components of the architecture shown in Figure 5 are for example the CESCM *Portal*, which provides a web-based GUI for both SCNO and VSCNOs, thus giving access to management and orchestration functions without additional integration work. On the other hand, the *SLA Monitoring* component is used to collect network KPIs and alarms on a per tenant basis.

2.1.5 5G NORMA

The preliminary functional architecture degined by 5G NORMA (https://5gnorma.5g-ppp.eu/) shows the network logical functional blocks with their belonging to the different layers and their logical interconnections. This is a meta-architecture in that it manages and orchestrates end-to-end network slice instances, so only MANO functions are shown concretely, while control and data layer are both shown with abstract Virtual Network Functions (VNFs) and Physical Network Functions (PNFs) only.

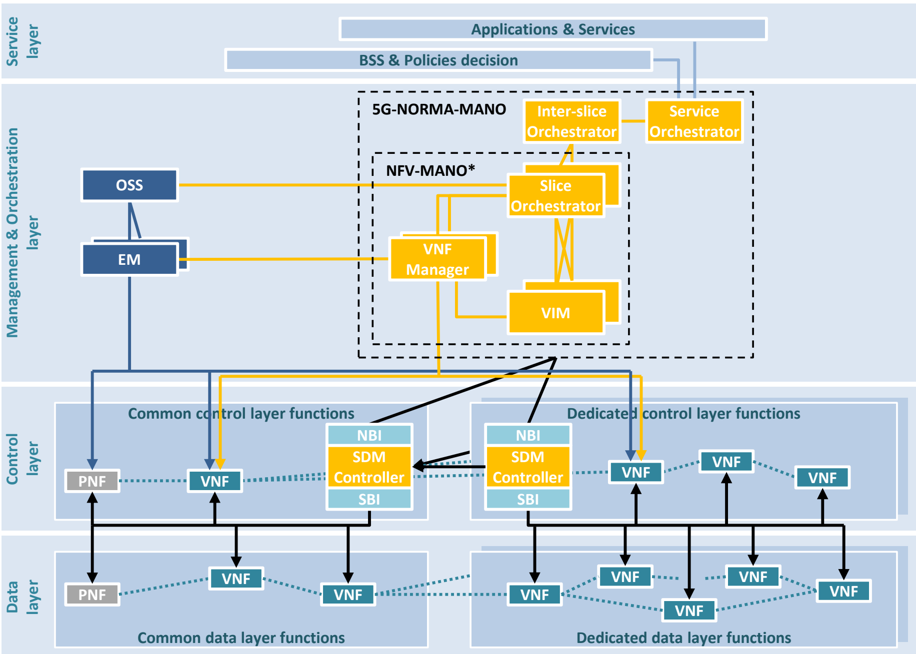

Figure 6: Preliminary 5G NORMA functional reference architecture.

The functional architecture reuses most of the elements of the NFV architectural framework, with some additions to cope with the support of network slicing and allow of the control of mobile network functionalities in a flexible way. These requirements result into the incorporation of three new functional elements.

The **Inter-slice Orchestrator** is owned and operated by the service provider. It has a comprehensive view of the subset's resource requirements and overall resources of according infrastructure providers. It handles the dynamic provisioning of the slices and *manages the resource sharing (physical and virtualized resources) among slices*, i.e., scaling up a slice may need another one to scale down. It executes policies decision to solve conflicting requirements between slices for sharing (virtual) resources and links, e.g. rules based on different slices' priority policies. Based on its coordination decision it triggers the Slice Orchestrator for creating, updating or releasing the slice. It provides the input parameters and rules to the Slice Orchestrator for virtual resource orchestration. In case where some slices, possibly belonging to distinct tenants, share some VNF and/or PNFs, i.e., the common control and data layer functions, it coordinates the allocation of such resources among slices and tenants during their life cycle. A tenant who wants to optimize the resources among all the slices it owns may want to operate an Inter-slice Orchestrator on its own besides the one operated by his service provider. If and how this may be supported will be the outcome of the upcoming design iterations.

The **Slice Orchestrator** is owned and operated by the service provider that operates the slice for the tenant which is not precluding that tenant and service provider may be the same. There is one instance per slice. It includes all functionality of the ETSI NFV Orchestrator, namely it optimally *(re)allocates NFs* in its slice (cf. deployment view) and performs *lifecycle management of its slice*, i.e., it binds together all VNFs' life-cycle management via their respective VNF Managers.

The **SDM Controller** is a key function of the 5G NORMA architecture. It is assumed to have an SDM controller instance per network slice. It controls all of the network slice's dedicated PNFs and VNFs, indicated by black edges from its southbound interface (SBI) to all NFs (same rationale for overlaid edges and arrow heads as for the EM to NF interface, i.e. neither multicast nor inter-NF communication). The SDM Controller allows for the reconfiguration in the order of tens of milliseconds, to dynamically influence and optimize the performance of its network slice within the given amount of resources assigned to its network slice, i.e. at the time of the last (re)orchestration. On the other hand, the (re-)configuration done by EMs only occurs after (re-)instantiation and can be considered to take place at a different time scale (rater seldom, with extents in the order of several seconds).

Following the SDN spirit, the SDM Controller also exposes a Northbound Interface (NBI) towards the 5G NORMA-MANO functions, whose scope is two-fold. The 5G NORMA-MANO to SDM Controller direction is used to define all the QoE / QoS constraints that have to be fulfilled for a given traffic identifier, that may range to a single flow to an entire network slice. The granularity of this API (that goes beyond the simple NF re-configuration) will be determined during the project, but some examples of its envisioned operation are provided next. For instance, the UL/DL scheduler can be dynamically configured by the SDM Controller to provide the needed QoE-related KPIs to HD Video Users flows, while maintaining resources for Best Effort user flows. The network capacity may be another KPI that the SDM Controller must fulfill, taking decisions about NF reconfiguration and routing.

For efficiency reasons or due to the characteristics of a NF, as well as to transparently support multi-service for a single user terminal, it may be necessary to share some PNFs and VNFs among multiple network slices. For these **common network functions**, a separate SDM Controller is introduced. The SDM Controllers of all network slices that share the common NFs connect to the SDM Controller responsible for common NFs via their eastbound/westbound interfaces. This SDM Controller coordinates access of all the SDM Controllers of all the network slices that use its common NFs as part of their end-to-end network slices. It resolves potential conflicting requests.

Every instance of a NF instance, dedicated as well as common, including the SDM Controller, is owned and operated by exactly one service provider. While dedicated NF instances are used by exactly one network slice, a single instance of a common NF is used by multiple network slices, possibly owned by different service providers. A single end-to-end slice may use common NFs of different common NF owners, i.e., a single slice may span a single chain of dedicated NFs (those of the slice owner) and multiple chains of common NFs, each owned by a certain service provider, including itself.

2.1.6 Relation between 5G-XHaul control architecture and other 5G-PPP architectures

5G-XHaul proposes an overarching layered architecture that is well aligned with the architectural principles laid out by the 5G-PPP association. This overarching layered architecture is depicted in Figure 7 and has been introduced in deliverable D2.2 [6].

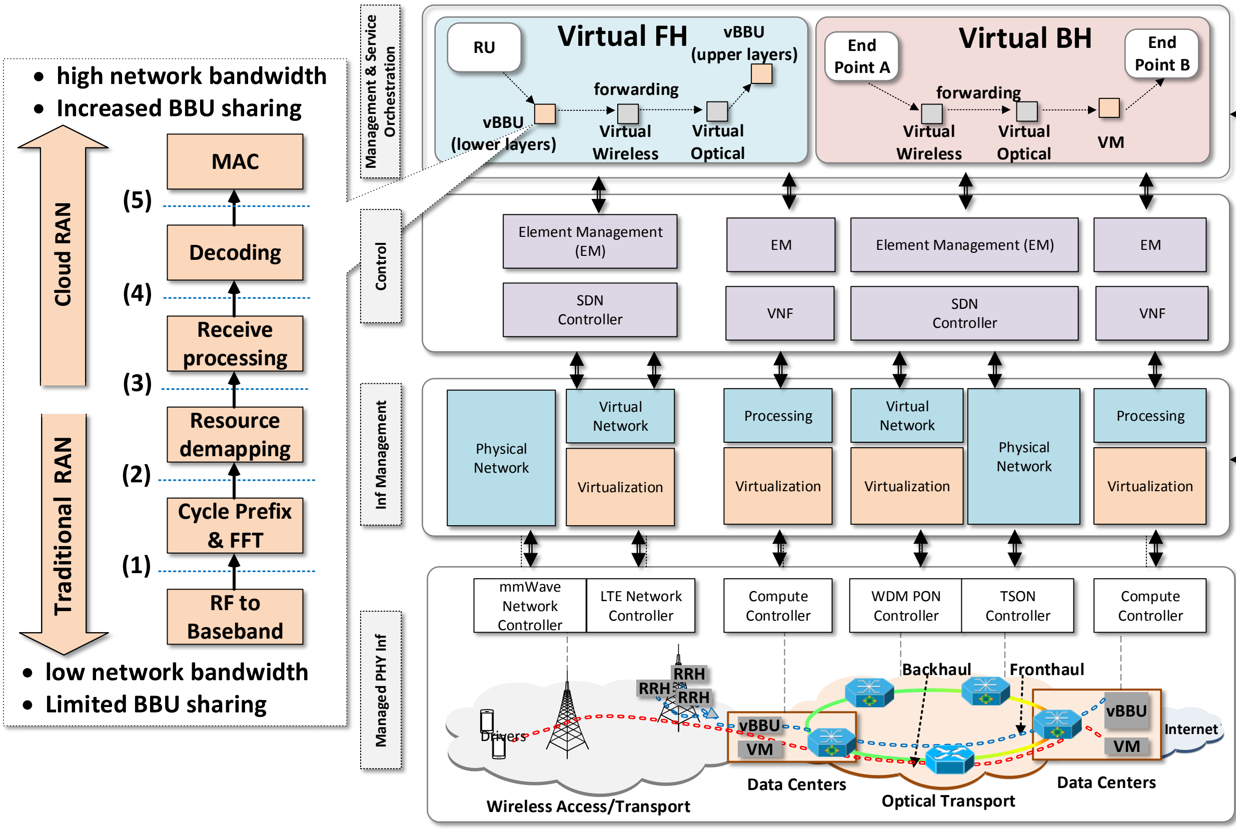

Figure 7. 5G-XHaul overarching layered architecture from [6].

The layered architecture described in Figure 7 embodies the principles of heterogeneous functional splits, and of virtualization, where network functions are implemented as Virtual Network Functions (VNFs), and is thus aligned with the 5G-PPP architectural principles. At the bottom of the layered architecture we have the managed physical infrastructure containing both network and compute resources. Above the managed physical infrastructure we have the infrastructure management layer that virtualizes both network and compute resources, and offers these virtual resources to an external control layer, controlled by the 5G-XHaul tenants. Finally, a management and orchestration layer is in charge of creating service function chains orchestrating the per-tenant virtual resources. The 5G-XHaul layered architecture is well aligned with the 5GPPP views on architecture and with complementary efforts carried out in other 5G-PPP projects. For example, the SELFNET layered architecture shares a similar structure with the 5G-XHaul architecture, although SELFNET puts more emphasis on autonomic aspects. Integration of network and compute resources, and the fact of using separate controllers for each technology domain are also highlighted by the 5G-Crosshaul architecture. The concept of a mobile architecture composed of VNFs that can be flexibly allocated through distributed compute resources is one of the core components of the 5G-NORMA architecture. 5G-XHaul shares with Superfluidity the view of distributed compute and storage resources instantiated at cell sites, local, regional and central data centres.

However, whereas Figure 7 presents an overall architecture describing the 5G-XHaul overarching view, the development focus of 5G-XHaul is on the lower two layers of the layered architecture, namely the managed physical infrastructure layer and the infrastructure management layer. In particular, 5G-XHaul assumes a multi-technology transport network, comprising wireless, optical and Ethernet technologies, which are stitched together through a common SDN based control plane. The main goal of this heterogeneous transport network is to connect in a scalable way distributed compute and storage resources, and to provide virtualization capabilities allowing tenants to control their transport slices. These heterogeneous transport technologies are bound together by a common control plane, for which a set of requirements and an initial design are introduced in the next section.

2.2 5G-XHAUL Control System Architecture Requirements and initial Design

2.2.1 Requirements on the 5G-XHaul Control Architecture

Table 1 provides an overview of the requirements to be fulfilled by the 5G-XHaul control plane architecture.

Table 1: Control Architecture Requirements.

Number	Summary	Description
R1	Multi-tenancy	Tenants should be allowed to define 5G-XHaul virtual slices of the transport infrastructure and control the resources allocated to them through a well defined API.
R2	Data Plane scalability	Data plane should be able to provide efficiently forwarding services to traffic belonging to different tenants over heterogeneous transport technologies.
R3	Control Plane scalability	The 5G-XHaul control plane should be able to control a transport network of the scale of metro networks in terms of network elements integrating both wireless and optical transport technologies. The control plane should be able to cope with a large number of tenants enforcing their policies on the slices they define.
R4	QoS Support	The 5G-XHaul system should be able to provide differentiated QoS transport classes to traffic belonging to different tenants based on the services they want to deploy on their networks e.g. fronthaul, backhaul, etc.
R5	Efficient Traffic Engineering	The 5G-XHaul control system should enable traffic engineering techniques to be applied to efficiently allocate the available resources between tenant traffic and services. These techniques should include multi-path routing, resilience, point-to-point, point-to-multipoint and multicast path setup, and energy efficiency (switching down network elements in times of low load).
R6	SDN–enabled south bound interfaces to transport devices	Different data plane devices should offer appropriate SDN south bound interfaces to provide to the control plane a rich amount of monitoring and configuration parameters for enforcing policies.
R7	Mobile Network-transport interaction	The 5G-XHaul system should be able to utilise information from the mobile network with respect to mobility patterns to efficiently engineer the resources of the transport network.
R8	North-bound interface (NBI)	The 5G-XHaul control plane should offer well defined NBIs that allow a higher level framework, such as a MANO orchestrator, to control the resources offered by the 5G-XHaul system.

2.2.2 Initial 5G-XHaul Control System Architecture

In this section, an initial view of the control plane architecture is presented together with the description of the component functionality and their interactions. This overview of the 5G-XHaul control plane architecture has also been presented in Section 5 of D2.2 [7] as part of the whole 5G-XHaul system architecture.

Figure 8 illustrates the initial 5G-XHaul control plane architecture that is based on the following design principles:

i. Full address space virtualisation 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 in their slices.

ii. 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.

iii. Scalability of the SDN control plane is achieved introducing the concept of *areas*. An area defines 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 are used to coordinate the actions of area level controllers.

iv. Finally, the vision of a converged multi-technology (wireless, optical and packet switch) 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.

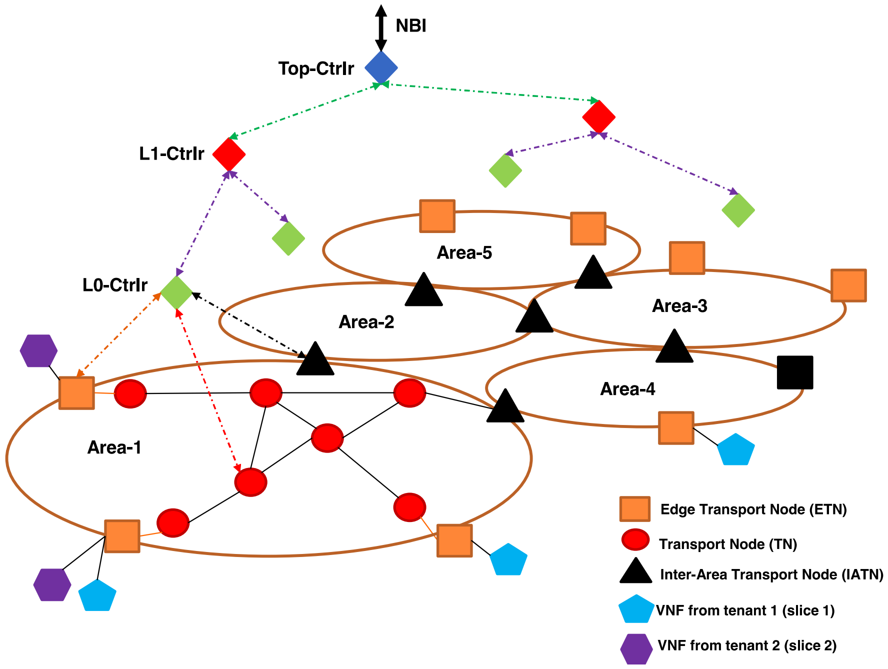

Figure 8: 5G-XHaul Control Plane Architecture.

In order to support the previous principles, three types of transport nodes are defined in 5G-XHaul that are depicted in Figure 8. First, *Edge Transport Nodes* (ETNs), connect the tenant VNFs to the 5G-XHaul transport network, maintain the corresponding per-tenant state, and encapsulate tenant traffic into transport specific tunnels. Second, *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.

The interested reader is referred to Section 5.2 in [6] for an example of how the control plane architecture described in Figure 5 can be instantiated over the physical network architecture defined in 5G-XHaul.

2.2.2.1 Edge Transport Nodes (ETNs)

ETNs maintain per-slice state providing 5G-XHaul tenants the required abstraction to operate on their slices. To address this requirement, 5G-XHaul will introduce the notions of *tenant ID* (T-Id) and *slice ID* (S-Id), which are globally unique identifiers for the end-to-end slices instantiated by a tenant, as later discussed in section 3; e.g. *opA.slice1*, where *opA* is the tenant ID for operator A and *slice1* is the slice ID of a slice that operator A wants to deploy for a specific service. There is thus a 1:N relationship between tenant and slice IDs. While the above identifiers need to be globally unique in the control plane, in the data plane different local transport slice IDs may be used in each 5G-XHaul area, namely a *Transport Slice ID* (TrSlice-Id) will be used to represent in the data plane a given *T-Id.S-Id*, where *TrSlice-Id* may be different in each 5G-XHaul area (c.f. Figure 8). Notice that having a notion of slice ID at the data plane is useful in order to easily deploy policies at the tenant or slice level, e.g. deploying an ACL that binds all the traffic for a tenant to a given QoS class, or that drops all the traffic of a malfunctioning slice. Thus, a function is embedded in the ETNs that performs the mapping between the global T-Id.S-Id of the control plane and the local Transport Slice ID in the data plane. In particular, an ETN embeds three major functions that are discussed next, namely: i) per-tenant Logical Datapaths, ii) a Forwarding Information Base (FIB), and iii) a TAF. These components are illustrated in Figure 9.

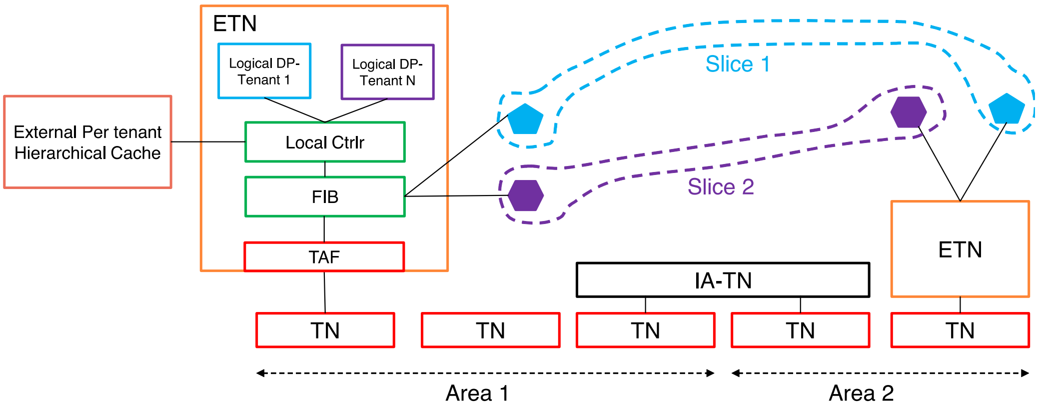

Figure 9: Detailed overview of an ETN.

On the control plane, an ETN instantiates a logical datapath for each tenant having VNFs connected to that ETN. Logical datapaths receive high level control policies from the tenants own control plane, and push those policies to a local SDN controller in the ETN. The local controller in the ETN obtains the rules from each logical datapath, adds appropriate context, and pushes the rules to the ETN FIB. Introducing multi-tenancy support at this last controller layer, which runs on the actual forwarding element, leaves intact the underlying performance-centric multi-stage FIB architecture design that performs the actual packet forwarding. Hence, a critical aspect in the design of the ETN is the datapath delay introduced by the ETN FIB. In order to minimize this delay, tenant-specific rules are often evicted, resulting in a small number of simultaneous rules hosted in the FIB. Consequently, a hierarchical structure of rule caches is used to scale up to a large number of rules coming from different tenants. Finally, the FIB matches tenant-specific rules and inserts packets into transport-specific tunnels, which are pre-instantiated and it is expected that traffic from multiple slices can be combined into a single transport tunnel. An initial design of the mentioned hierarchical cache system is provided in Section 6.

An ETN uses encapsulation to isolate transport network elements from per-tenant related state. A Transport network Adaptation Function (TAF) is included in the ETN that pushes the corresponding transport header before injecting the packets into the transport network, whereby the transport header signals three major pieces of information: i) the transport tunnel (path) in the area, the ii) local Transport Slice ID, and iii) the QoS allocated to that tunnel. Each ETN features a TAF corresponding to the transport technology used in the 5G-XHaul area where the ETN is located. At the moment, 5G-XHaul has proposed an initial design for an Ethernet TAF that is described in Section 3. However, TAFs for additional transport technologies may be considered in upcoming deliverables.

2.2.3 Transport Nodes (TNs)/Inter-area TNs (IATNs)

Transport Nodes (TNs) connect ETNs and IATNs within a given 5G-XHaul area (c.f. Figure 8). The concept of a TN is technology agnostic, thus a TN could be represented by a mmWave wireless node at the street level, by an Ethernet switch at the acces or metro segment, or by an active optical node at the metro network (e.g. TSON). Regardless of the actual technology, in 5G-XHaul a TN offers a dataplane abstraction where forwarding, along with some other primitives like bandwidth provisioning or reliability, can be programmed by a logically centralized control plane.

Since the set of ETNs and IATNs available in a given area are fairly static, 5G-XHaul assumes that transport tunnels between ETNs/IATNs in an area are pre-provisioned. A pre-provisioned transport tunnel means that the ETNs and IATNs in that area have an interface representing such tunnel, and that the required TNs have the corresponding entries in their FIBs. Notice however that pre-provisioned transport tunnels do not need to be static, as tunnels can be reconfigured by the control plane if the network state changes. For example, the control plane may switch down a set of TNs for energy saving, while relocating all the affected transport tunnels to other TNs; the ETNs would be agnostic to this relocation. In addition, pre-provisioned tunnels may be point to point tunnels, or multicast trees connecting a set of ETNs/IATNs in a given area. In the case of a multicast tree, the transport technology in the 5G-XHaul area needs to have support to replicate the data packets along the interfaces participating in each multicast group. In 5G-XHaul multicast group membership is managed by the logically centralized control plane. In the case of a 5G Mobile Network a multicast tree can be useful for example to connect base stations implementing a cooperative transmission scheme. Finally, transport tunnels are associated to a set of transport classes. In particular, 5G-XHaul has proposed a set of four transport classes, described in [3], dimensioned to transport fronthaul traffic, backhaul traffic, as well as traffic resulting from other functional splits. Thus, multiple transport tunnels to a given ETN/IATN may be pre-provisioned representing the different transport classes.

Inter-Area Transport Nodes (IATN) provide connectivity between neighboring 5G-XHaul areas (c.f. Figure 8). As illustrated in Figure 9, an IATN can be understood as an inter-connection function sitting above one TN for each area being connected by the IATN. The different areas can use the same or different transport technologies. The IATN interconnection function contains a control plane function and a data plane function that are described next.

In the control plane, an IATN needs to discover the areas that it has access to, and convey the corresponding area identifier to the 5G-XHaul control plane. In addition, a unique identifier is required for an IATN that also needs to be conveyed to the control plane. This information is required by the control plane to be able to allocate paths at the area level.

In the data plane, an IATN needs to implement the forwarding principle used in each of its connected areas. Thus, an IATN includes the corresponding TAF for each area it connects. IATNs maintain their own FIB function that maps tunnels from one area to tunnels of another area. In case, an IATN interconnects areas belonging to the same technology, technology specific optimizations are possible to accelerate the datapath that are currently being investigated.

2.2.4 Hierarchical Controller Design

In 5G-XHaul the control plane is composed of a hierarchy of controllers as illustrated in Figure 8. The top level controller, hereafter referred to as the *Top* controller is responsible for provisioning per tenant slices, and orchestrating the required connectivity across different 5G-XHaul areas and domains (e.g., optical transport domain, wireless transport domain). At the lowest level of the hierarchy we find the *Level*-0 controller that is responsible for the provisioning and maintenance of transport tunnels between ETNs and IATNs of a given 5G-XHaul area; a Level-0 controller operates at the level of individual network elements. A set of Level-0 controllers are logically organized under a Level-1 controller, which is in charge of maintaining connectivity between the corresponding Level-0 areas, and operates with a higher level of abstraction, namely maintains state at the area level instead of maintaining state for each network element as Level-0 controllers do. Notice that the proposed architecture is recursive in the sense that a Level-i controller can be defined to coordinate a set of Level-(i−1) controllers. Hereafter, and without loss of generality, we assume a three level hierarchy consisting of Level-0, Level-1 and Top controllers.

Dimensioning the number of network elements under a Level-0 controller, or the number of Level-0 controllers under a Level-1 controller depends on many factors and is a current area of research in 5G-XHaul. A Level-0 controller is assumed to be in charge of an area instantiating a single type of transport technology, i.e. a mmWave area, an Ethernet area, or an active optical area. Thus, the number of elements under a Level-0 controller will be very dependent on the particular technology. For example, for scalability reasons a large number of mmWave transport nodes deployed at the street level can be partitioned into a plurality of areas and Level-0 controllers, whereas the optical or Ethernet switches composing the metro network could be controlled respectively by a different Level-0 controller. Notice that having technology specific controllers allows to develop solutions tailored to the control plane of each technology. Upper level controllers, i..e Level-1 and Top controllers, do not need to be technology specific since they operate at a higher abstraction level (the area level). In practice for scalability reasons controllers at each level will be deployed in clusters of synchronized controllers.

In Section 6 an intial design of the hierarchical control plane is provided, including an intial set of functions allocated at each control layer.

3 MULTI-TENANCY OVER THE 5G-XHAUL TRANSPORT NETWORK

In this section we provide an overview of the state of art in multinenancy and virtualization in networking. We start in Section 3.1 by surveying general virtualization primitives to then focus on network virtualisation techniques used in data center environment, which provide the foundations for the design carried out in 5G-XHaul. Based on the state of the art analysis, in Section 3.2 we derive the requirements for the multi-tenancy mechanisms supported by 5G-XHaul, and provide an initial design.

3.1 State of the Art on virtualisation/multitenancy

Multi-tenancy represents an extremely challenging networking environment. Nowadays, more and more people and companies demand computational and networking resources from professionally managed infrastructures, without investing the significant capital and operational expense required for running their own infrastructure. These entities, hereafter referred to as 'tenants', utilize a subset of the resources provided by these infrastructures, however, in a transparent way that do not confuse them with side effects coming from the fact that 'virtualization' is used. Virtualization is a high technology trend that enables the provisioning of abstract resources, which are slices or unions of physical resources. The most well-known example is the Virtual Machine (VM) that depends on abstractions of the hardware resources of physical host machine(s) e.g. a VM can exploit a storage memory given as an abstract hard disk, that could be either half of an existing physical hard disk or the union of two physical ones.

As described in D2.1 [1], the goal of 5G-XHaul is essentially to build a multi-tenant converged fronthaul(FH)/backhaul(BH) and wireless/optical transport network, where a 5G-XHaul operator leases connectivity services to 5G-XHaul tenants, for example a 5G mobile operator[2]. However, the 5G-XHaul vision goes beyond the traditional leasing of transport connectivity services between fixed locations, and embraces a softwarised vision of the network where tenants connect virtualised network functions (VNFs) and workloads across geographically distributed locations. In 5G-XHaul a tenant should be able to fully interact with its own virtual network in two ways: the tenant's endpoints (e.g. hosts and VMs) send packets, and the tenants control plane (e.g. routing protocols, manual configuration) configures network elements (e.g. switches and routers). Tenants should be able to define a set of virtual network elements (or datapaths) that they can configure (through their decoupled SDN control planes) as they would physical network elements, as well as to send packets to this network using the same switching, routing, and filtering service as they would have in their own physical network.

In this section we review the relevant state of the art related to network virtualisation and multi-tenancy. First, we highlight why the virtualization primitives adopted by traditional transport network technologies fall short to address the 5G-XHaul requirements. Then, we review the work carried out in virtualisation for multi-tenant data centres, which will be foundational for the network virtualisation solution adopted in 5G-XHaul. Finally, we lay out the network virtualisation requirements to be fulfilled by the 5G-XHaul control plane and sketch an initial solution design that will be polished in subsequent WP3 deliverables.

3.1.1 Traditional network virtualization primitives

Network technologies operating at different layers have independently developed mechanisms to virtualize network resources. The result is that scattered across different technologies and layers we have available today a set of virtualisation primitives. For example, a common virtualisation technology is the use of Virtual LANs (VLANs) [23] to virtualise L2 broadcast domains. VLANs are often used in operator networks and in data centres, to isolate tenants for security and scalability reasons. Network Address Translation [24] is commonly used in home networks as well as operator networks to share a single public IP address among many end hosts; NAT is thus a form or virtualizing a single IP address resource. Virtual Private Networks (VPNs) [25], virtualize a direct L2 or L3 connection among distributed customer sites over a shared IP/MPLS infrastructure operated by an ISP. A very common implementation is to use MPLS VPNs, whereby Provider Edge (PE) routers instantiate a virtual routing and forwarding table (VRF) for each customer VPN. A VRF contains the PE routers connecting the different customer sites, as well as the MPLS Label Switch Path (LSP) connecting to the destination PE. MPLS VPNs allow customers to use overlapping IP address spaces.

[2] The interested reader is referred to [1] for a detailed description of the different actors interacting with the 5G-XHaul system.

A more recent approach to virtualization is FlowVisor [26] that allows to virtualize an entire network element, namely an Openflow switch. In particular, FlowVisor enables the same OpenFlow-based forwarding plane to be shared among multiple logical networks, each with distinct forwarding logic. It is implemented as an OpenFlow proxy that intercepts messages between the OpenFlow switches and controllers, making sure that each controller is able to observe and control the slices of its corresponding switches. In [26], authors propose a two-layer hierarchy where the top layer consists of the tenant controllers configuring the corresponding network slices, while the bottom layer is a group of tree-like connected FlowVisor instances (usually only one instance) that is responsible for the network virtualization and the provision of each network slice to the corresponding tenant.

The previous virtualization primitives are often used to enable multi-tenancy services. An example are the E-Line, E-Tree and E-LAN connectivity services defined by the Metro Ethernet Forum (MEF) [27]. An E-Line service consists of a point to point connection between two customer devices connecting over the shared infrastructure. An E-Tree establishes a tree type of connection between a root device and a set of leafs, and an E-LAN enables an “any to any” connectivity model at layer two. MEF services are identified at the customer demarcation points using VLAN tags or physical ports.

Although the previous virtualization primitives provide some form of resource sharing and isolation, and can enable multi-tenancy, they fall sort to address all the 5G-XHaul multi-tenancy requirements identified in deliverable D2.1 [1]. The reason is that the presented virtualization primitives are tightly coupled to the physical infrastructure, which greatly complicates operations in an environment where tenant workloads and VNFs are supposed to be able to relocate between attachments points in a dynamic fashion. This can be understood through the following example. When defining a MEF service an operator needs to configure the ports of the physical switch connecting to the customer equipment with the appropriate VLAN. If the tenant decides to change its point of attachment, due to e.g. a VM migration, then the operator providing the MEF service needs to get involved and reconfigure the VLANs in the new switch. Hence, a solution where tenant workloads can freely move through the physical infrastructure without requiring changes on the physical network infrastructure is required to fulfil 5G requirements. Next, we survey the solutions that have appeared in the data centre space that need to fulfil similar requirements.

3.1.2 Data centre virtualization

Flexible instantiation and mobility of VMs are requirements that have recently appeared in the cloud networking space, and that have been implemented in large centralized data centres. Following the 5G trend, 5G-XHaul attempts to enable a cloud-like operation but over a geographically distributed area connected through a transport network. Therefore, in this section we review the existent work on network virtualisation for multi-tenant data centres, which will lay the foundation for the initial 5G-XHaul network virtualisation design presented in Section 3.2.

The considered approaches on data centre network virtualisation are compared in terms of: i) the network abstraction exposed to tenants, ii) the mechanisms employed to achieve virtualization, and the degree of achieved virtualisation, iii) the control plane mechanisms that allow tenant VMs to discover each other, and iv) the assumptions that each mechanism makes on the underlying physical infrastructure

3.1.2.1 Tenant abstraction

A common tenant abstraction in a multi-tenant data centre is that of a layer two (L2) switch. This means that a tenant perceives all its VMs, as if directly connected to a layer two switch, thus belonging to the same broadcast domain. This abstraction is convenient because all tenant L2 and L3 protocols will work out of the box, and hence it enables an easy migration of tenant workloads to the cloud. An example of network virtualisation solution offering a L2 switch abstraction can be found in [28] or [30]. The L2 abstraction can be easily generalized to a combined L2/L3 abstraction, whereby tenant VMs are grouped into subsets of VMs connected at L2, and then L2 domains can be routed at L3. This abstraction maps nicely to the structure of web applications, where load balancers, web servers and database servers are often segregated in different routed L2 domains. An example of this tenant abstraction can be found in [29] where the virtualisation layer already offers tenants the possibility of instantiating a virtual router connecting the different L2 domains. This is the same approached followed by some commercial products like VMWare’s NVP [15]. A complementary proposal on tenant abstraction can be found in [31], where it is argued that for efficiency reasons a tenant abstraction should not only describe connectivity patterns between VMs, but also bandwidth requirements. The reason is that if the virtualisation platform is made aware of the bandwidth requirements between tenant

VMs, then it can more efficiently embed virtualisation requests (or slices) coming from multiple tenants over the physical infrastructure. In this regard in [31] a virtual oversubscribed cluster abstraction is proposed where subsets of VMs are grouped in L2 domains operating at a bandwidth, B, and different L2 domains are connected to each other at L3 with an oversubscription factor O. The authors demonstrate how the proposed abstraction increases the number of simultaneous tenants served, as compared to an L2 abstraction, where full bandwidth any to any communication patterns between VMs need to be accommodated.

All the previous abstractions have in common that the control plane of the tenant's virtual network is a pre-defined one (L2 or L3), and is already offered by the virtualisation platform. A solution to enable tenants to execute their own custom control plane, as posed by the 5G-XHaul architecture described in deliverable D2.2 [6], is to consider the tenant control functions as VNFs, and have the virtualisation substrate forward traffic between these VNFs. For example, a tenant's VNF could be a virtual switch controlled by the tenant using a protocol like openflow. An alternative, is to virtualize the individual elements of the physical infrastructure and offer tenants a direct control interface for their virtual network. This approach is described in [32], where a network of)OpenFlow switches is virtualized allowing each tenant to define an arbitrary virtual network topology. Elements of the virtual topology are also openflow switches, and can hence be controlled by the OpenFlow controller hosted at the tenant's premises. The technique followed in [32] to implement such tenant abstraction is to intercept the LLDP messages generated by the tenant controllers, and return them to the tenant controller as if coming from the corresponding virtual switch.

Finally, another interesting abstraction is the one proposed in Network Virtualization Platform (NVP) [36], In NVP a tenant can define multiple layer two domains where subsets of their VMs are attached, and this layer two domains can be connected with tenant specific datapaths, such as a virtual router, which could also implement access control policies to define security groups, isolate a VM, etc.

3.1.2.2 Means to achieve virtualization

In the data centre space virtualization consists of allowing each tenant to connect VMs using a custom address space, where the tenant VMs are instantiated on different physical servers, possibly sitting on different racks. Thus, solutions can be classified according to whether they allow tenants to use overlapping L2 or L3 address spaces (full address virtualisation), or not. In a nutshell, all proposed virtualisation solutions use some form of encapsulation that decouples the tenant address space from the address space used by the physical infrastructure, and maintain some form of mapping between the addresses of the tenant VMs and the transport addresses of the servers where these VMs are instantiated. In this section we review the concrete virtualisation mechanisms used by some representative solutions in the state of the art.

VL2 was an early data centre virtualisation approach proposed in [28]. VL2 does not provide full address virtualization. Instead, the full IP address space is partitioned into Locator Addresses (LA), used by the physical infrastructure switches, and Application Addresses (AA), used by the tenant VMs. A VL2 agent is defined that operates inside the hypervisor of physical servers, and intercepts the packets generated by the tenant VMs. For each packet the VL2 agent appends an outer IP header containing the LA address of the physical server where the destination tenant VM is located. Notice that VL2 allows for VM mobility as long as the VL2 agents are updated with the corresponding mapping between AA and LA addresses. The mechanisms used to update this mapping are discussed in Section 3.1.2.3.

Unlike VL2, Netlord [29] is a system that provides full L2 and L3 address virtualisation, meaning that tenants can use overlapping L2 and L3 address spaces. In Netlord an interface inside a VM is uniquely identified by a 3-tuple composed of: a tenant Identifier (tenant_ID), a MAC Address Space ID (MACASID), and the interface MAC address. Thus, only this 3-tuple needs to be unique in the system. A Netlord Agent (NLA) in the origin server hypervisor intercepts outgoing packets from a VM, and encapsulates them with a custom IP and Ethernet headers that are used both to route the packet through the physical infrastructure and to convey the information that the receiving NLA needs to deliver the packet to the correct target VM. The encapsulation mechanism used in Netlord allows to reuse commodity switches in the underlying physical infrastructure. However, Netlord functionalities are required not only at the Netlord agent in the hypervisors, but also at the edge, i.e. Top of Rack (ToR), switches. In Netlord an outer Ethernet header containing the MAC addresses of the ingress and egress ToR switches is added to the transmitted tenant frames. In addition an outer IP header is also added with the following encoding. The NLA agent encodes in the source IP address field the MAC address space identifier (MACASID) for this packet, and in the destination IP address field the NLA encodes both the port of the egress switch connecting to the server that hosts the target VM, and the tenant ID. Hence, upon receiving a packet, the egress ToR switch drops the Ethernet header, looks up the destination IP address, and resolves the outgoing port for the packet. Upon receiving a packet, the destination NLA is able to recover the 3-tuple identifying the target interface by: i) obtaining the tenant ID from the destination IP address, ii) the MACASID from the source IP address, and iii) the destination MAC from the Ethernet destination address at the inner Ethernet header.

VXLAN is a standardized encapsulation mechanism [30] to enable virtualisation of layer two segments. Similar to the MAC address space ID (MACASID) in Netlord, VXLAN defines a VXLAN Network ID (VNI) representing a L2 broadcast domain. Different VNIs can connect to each other through a L3 router. Thus, VNIs are the equivalent in the virtualized domains of VLANs in the physical infrastructure. The main difference is that the VNI is 24 bits long, compared to the 12 bits available in VLAN ID, hence allowing to multiplex a much larger number of tenants over the shared infrastructure. In terms of encapsulation, VXLAN defines a custom header transported over UDP. VXLAN encapsulation is performed by a software agent running in the servers' hypervisor, called Virtual Tunnel End Point (VTEP). VTEPs maintains a binding between all the tenant MAC addresses of the VNIs connected to that VTEP, and the transport IP address of the VTEP where the VM containing a particular MAC address is connected. Upon receiving a packet, the target VTEP looks up the VNI in the VXLAN header, pops the IP, UDP and VXLAN headers, and delivers the payload to the target VM. VXLAN defines transport multicast groups for each VNI to efficiently transmit per-VNI layer two broadcast frames over the physical infrastructure. Notice that unlike Netlord, VXLAN only requires the presence of the VTEP in the servers' hypervisor, and does not require any special treatment from the physical network devices. However, unlike Netlord, VXLAN does not explicitly include the tenant ID in the encapsulation header, which hinders the deployment of tenant specific policies.

Finally, SEC2 [35] uses network virtualization that is supported through Forwarding Elements (FEs) and a Central Controller (CC). FEs are essentially Ethernet switches with the ability to be controlled from a remote CC that stores address mapping and policy databases. FEs perform address mapping, policy checking and enforcement, and packet forwarding. The network architecture has three levels: one core domain, multiple edge domains identified by an edge domain id (eid), and several customer networks connecting to an edge domain. To isolate different customers within each edge domain, SEC2 uses VLAN with the scope limited within the same edge domain, thus, eliminating the limitation of the number of customers that can be supported due to VLAN ID size. In order to communicate across edge domains, SEC2 uses MAC in MAC encapsulation [16]. SEC2 provides full L3 address virtualisation.

3.1.2.3 Control plane: Discovering tenant addresses

In the description of the virtualisation mechanisms provided in Section 3.1.2.2 it has not been discussed how the binding between the tenant addresses and the corresponding transport addresses is obtained, or refreshed if VMs moved. This is the main job of the control plane, which is discussed in this section.

In VL2, the VL2 agent residing in the server hypervisor maintains a mapping between Locator Addresses (LA) and Application Addresses (AA). For this purpose VL2 contains a Directory Service (DS) providing eventual consistency between the LA to AA address mappings. The DS in VL2 is implemented through a read optimized cluster of DS servers, for load balancing purposes, which receive requests from VL2 agents when these do not know a particular AA to LA mapping. The goal of the DS cluster is to respond to queries from VL2 agents in less than 10 milliseconds. On the other hand, VL2 agents discover the AA addresses of the VMs connecting to them and report new bindings to a cluster of Replicated State Machine (RSM) servers. This RSM servers are optimized for writing, and the RSM cluster is synchronized using Paxos [17] so that the servers in the cluster eventually converge to a common shared state. The read optimized DS servers refresh their stored bindings from the RSM servers every thirty seconds. Once a binding is looked up from the DS, it is cached in the requesting VL2 agent to speed up subsequent look ups. Notice that the described system provides eventual consistency, in the sense that it is possible for a DS server or VL2 agent to contain a stale cache entry. In this case the target VL2 agent will generate and error message and the corresponding cache entry will be invalidated. The underlying assumption in the design of the VL2 Directory Service is that VM mobility is infrequent, and that most of the bindings will only need to be set up at provisioning time.

To maintain the bindings between 3-tuples and the MAC address of the corresponding egress ToR switch, Netlord uses a custom protocol called NL-ARP, which combines the functions of ARP and DHCP. Like VL2, Netlord assumes that instantiation of new VMs or VM mobility are rare events. When the Netlord Agent (NLA) in the servers' hypervisor detects a new binding, it broadcasts a NLA-HERE message to all other NLAs in the network. A special tenant ID is reserved to exchange information among all NLAs. Equivalently, the NL-ARP protocol also supports NLA-WHERE and NLA-NOTHERE messages to cope with stale entries. In order to minimize NLA-ARP load, the entries learned through NL-ARP are never expired, and instead are corrected through the previous messages. In [29] it is shown that in a data centre environment NL-ARP results in negligible overhead. However, it is uncertain how this protocol would behave over a transport network like the ones considered in 5G-XHaul.

VXLAN [30] features two mechanisms to construct the bindings between the MAC addresses of a VNI and the transport IP address of their corresponding VTEP: i) address learning in the data plane, and ii) address learning in the control plane using MP-BGP [18]. In the data plane, VXLAN can learn address bindings by means of flooding, as it is done by regular Ethernet bridges. Upon receiving a frame for which there is no binding in the VTEP forwarding table, the VTEP broadcasts the frame to all VTEPs active in that VNI using the pre-setup multicast group. Thus, when the VTEP where the target MAC address is connected receives the broadcast frame, it replies with a unicast frame to the origin VTEP, which learns and caches the address binding; an example of such broadcast/unicast frame exchange is an ARP Request/response exchange. Notice that even though VXLAN recommends to map the layer two broadcasts into per-VNI multicast groups, this approach may introduce significant amount of flooding in the physical infrastructure reducing the bandwidth available to other connectivity services. Hence, some VXLAN implementations contemplate an alternative control plane to construct address bindings based on MP-BGP. Compared to flooding, which is reactive, using MP-BGP address bindings are constructed in a proactive manner. VTEPs monitor the MAC addresses of the tenant VMs connected to them, and as soon as a new binding is learnt, a VTEP communicates it to the other VTEPs in that VNI. This approach is similar to the NLA-HERE messages defined in Netlord. However, using MP-BGP comes with the advantage that the scalability mechanisms defined in BGP can be immediately reused. For example, instead of propagating the bindings directly to all VTEPs in a VNI, in BGP a VTEP can deliver the new binding to a Route Reflector (RR) node, which then distributes the update to the affected VTEPs. In addition, to increase scalability a cluster of RR nodes can organized in a hierarchical way [19]. The role of MP-BGP is thus to distribute address bindings to the corresponding VTEPs, but a solution is needed to form unique addresses instead of the plain tenant addresses, which could overlap between tenants. For this purpose, in [20] a new Network Layer Reachability Info (NLRI) element for MP-BGP is being defined that adds a Route Distinguisher (RD) to disambiguate VXLAN tenant addresses. The proposed RD may be composed of the VNI, if this is guaranteed to be unique, or may contain additional information in scenarios where VNIs cannot be guaranteed to be unique, such as a data centre interconnect scenarios. Alternative to MP-BGP, some virtualization solutions are also using XMPP as a mechanism to distribute tenant VM to transport node address bindings on a per VPN basis [37].

3.1.2.4 Data plane: Underlying Topology and forwarding

All the virtualisation solutions discussed so far share an agent at the edge of the network that encapsulates tenant frames into transport tunnels. In this section we discuss what assumptions each solution makes on the underlying physical infrastructure.

VL2 [28] is very tight to the topology of the physical network infrastructure because it assumes that the network is wired in a Fat-Tree topology. A Fat-Tree [21] is a form of Clos network, i.e. non-blocking, which guarantees that there is no oversubscription, meaning that all servers connecting to the network can always communicate to any other server in the network at the full rate of their network interface, regardless of where the other server is located[3]. A characteristic of the Fat-Tree topologies is the abundant number of equal cost multi paths to a given destination, required to avoid blocking. Hence, VL2 makes use of multi-path to balance flows and maximize utilisation. In particular VL2 uses Valiant Load Balancing (VLB), whereby to route to a given destination a random intermediate switch is selected, which then uses ECMP routing to reach the final destination.

Netlord shares with VL2 the intent to proactively spread load across multiple paths. For this purpose, Netlord makes use of SPAIN [33], a technique to enable efficient use of multiple paths in network of layer two switches. SPAIN switches run a routing algorithm to all possible destinations, namely ToR switches in Netlord, and then use VLAN tags to encode a given path between two end points. Thus, Netlord agents (NLAs) contain a SPAIN agent that inserts a VLAN to the outer Ethernet header in order to signal the path to be followed by the packet; this can be understood as a form of source routing. In particular, NLAs hash the

[3] Of course, this is assuming that there is no congestion, i.e. multiple servers willing to talk to the same destination simultaneously.

5-tuple of tenant packets in order to obtain the corresponding VLAN for a given destination. In this way, different flows are forwarded through different paths, but packets within one flow always follow the same path to avoid re-ordering. SPAIN also proactively monitors the health of different paths by periodically injecting custom traffic into the transport tunnels.

Finally, VXLAN [30] is the simplest mechanism regarding the assumptions made on the underlying physical network, as it only assumes that the underlying network can forward IP frames. Making no assumption on the underlying network maximises the deployment scenarios for VXLAN, but can also jeopardize performance if the underlying network is not overprovisioned.

3.2 Requirements and Initial Design

Based on the SotA analysis presented in the previous section we describe in Section 3.2.1 a list of requirements to be fulfilled by the virtualisation solution designed in 5G-XHaul, and provide in Section 3.2.2 and initial design, which will be further revised and evaluated in upcoming deliverables.

3.2.1 Requirements on 5G-XHaul virtualisation

Table 2 provides an overview of the requirements to be fulfilled by the 5G-XHaul virtualisation mechanisms.

Table 2: 5G-XHaul virtualisation requirements.

Number	Summary	Description
R1	Tenant workloads	Tenants should be allowed to connect to a 5G-XHaul virtual slice standard workloads (VMs or VNFs) that assume a TCP/IP stack. Hence, protocols like ARP, and other protocols based on L2 broadcasts, should work out of the box.
R2	Tenant abstraction	When defining a 5G-XHaul slice, tenants should be allowed to connect directly their VNFs assuming a layer two switch abstraction. In addition, tenants should also be able to define custom datapaths connecting the tenant's VNFs. Custom datapaths will be controlled by the tenant through a northbound interface.
R3	Address space virtualisation	5G-XHaul tenants will be able to use overlapping L2/L3 address spaces. The physical infrastructure will impose no restrictions on the address spaces used by the tenants.
R4	Management plane	5G-XHaul will offer a management plane to allow the 5G-XHaul infrastructure owner to set up new slices, monitor SLA compliance, and detect failures.
R5	Efficiency	The 5G-XHaul control will have mechanisms to minimize flooding over the shared infrastructure. The 5G-XHaul virtualization mechanism will benefit from a tight coupling with the physical infrastructure.
R6	Per-tenant policies	The 5G-XHaul infrastructure owner should be able to deploy per-tenant policies, which will affect all the slices instantiated by the tenant.
R7	Scalability	The 5G-XHaul virtualization solution should be able to scale to practical metro networks, and to instantiate a large enough number of slices to address the use cases described in deliverable D2.1 [1].
R8	Transport agnostic	The 5G-XHaul virtualisation should be able to operate under different underlying transport technologies

3.2.2 Initial Design

In this section we describe an initial design of the 5G-XHaul virtualization mechanisms that will enable multi-tenancy and transport network slicing. It is to be noted though that this initial proposal is subject to be modified in subsequent deliverables. In addition, the current proposal does not cover all the aspects of a full virtualization solution, but will instead focus on the area of the abstraction presented to tenants, and the encapsulation mechanisms required to enable virtualization.

This section builds on the initial control plane design described in Section 2.2, where the different types of transport nodes participating in 5G-XHaul have been introduced: i) Edge Transport Nodes (ETNs), ii) Inter-Area Transport Nodes (IATNs), and iii) Transport Nodes (TNs). This section also builds on the internal components of an ETN illustrated in Figure 9.

Figure 10 illustrates the abstraction offered by the 5G-XHaul network to a 5G-XHaul tenant. Hereafter we refer to this abstraction as the 5G-XHaul *slice*. A slice provides a tenant the means to describe its connectivity requirements, and provides tenants a virtual representation of their own transport network that the tenants are allowed to control through the 5G-XHaul north-bound interfaces. In addition, a 5G-XHaul slice has a spatial component as tenants have to declare in which ETNs, locate in different geographic locations, they want to use in their slice. Notice though that the 5G-XHaul slice only considers the network aspects, while it is assumed that compute requirements in each ETN would be provisioned through another system.

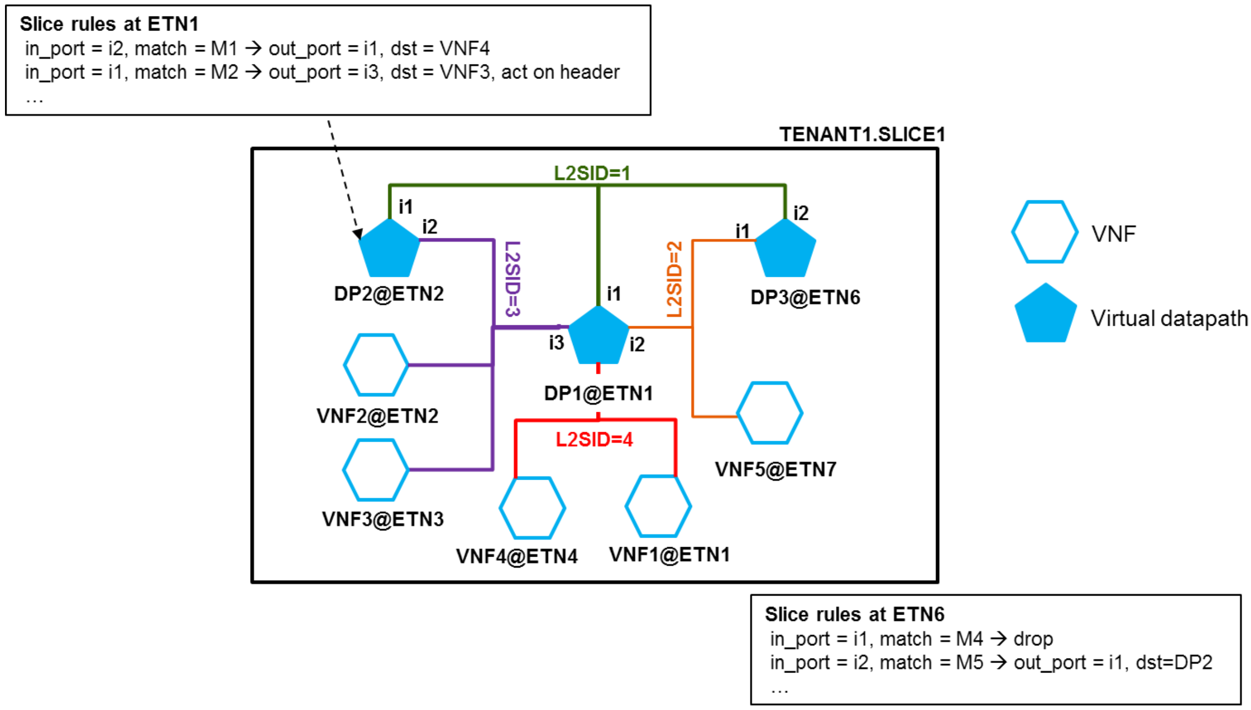

Figure 10: Tenant abstraction proposed in 5G-XHaul.

A 5G-XHaul slice is composed of the following components:

- **Virtual layer two segments**. Emulate a layer two segment, i.e. a broadcast domain, over a set of tenant virtual entities (VNFs or vDPs).
- **Virtual Network Functions (VNFs)**. Identified within a slice by a MAC address (VNF_@) scoped to a single layer two segment. VNFs are physically bound to an ETN. Notice that VNFs per-se are considered outside of the scope of the 5G-XHaul slice. From the perspective of a 5G-XHaul slice a VNF is treated as a network end point.
- **Virtual datapath (vDP)**. Datapath containing custom network control logic defined by the tenant. Unlike a VNF that is only attached to a single layer two segment, a virtual datapath may have several interfaces, each one connected to a different virtual layer two segment. Each interface of a virtual datapath is identified by a MAC address (vDP_@). Figure 10 provides examples of the rules that a tenant can define to control the behaviour of a vDP. These rules provide a <match, action> abstraction, where the defined matches may include the logical ports of a vDP, the addresses of the VNFs/vDPs, and packet modification actions. The definition of these rules, and the north-bound interface used to provision these rules are left for future work.

- **Layer two segment identifier (L2SID)**. This is the fundamental identifier in a 5G-XHaul slice. It is similar to the MACASID in Netlord or the VNI in VXLAN. L2SIDs are unique system wide, meaning that L2SIDs cannot be reused within or across slices. Therefore, the size of the LS2ID limits the maximum number of slices supported in the system. Following up on the requirements devised in D2.1 [1] the L2SID field should have a size of at least 24 bits, allowing for 16.7M layer two segments. As a matter of example a 24 bit L2SID could be defined by: i) stacking two VLAN tags in the data plane, ii) by using a VXLAN header (although not necessarily over UDP+IP), iii) by injecting a fake UDP header and encoding the L2SID in the source and destination ports (16 bits each), or iv) by injecting a fake IP header and encoding the L2SID in the source or destination IP addresses (32 bits each). We leave as future work the definition of the actual encoding of this field.

Figure 10 provides a logical view of a transport slice, which must be instantiated over the physical infrastructure. Figure 11 illustrates one such possible instantiation. It can be seen how, as explained in Section 2.2, the 5G-XHaul infrastructure is partitioned in areas composed of ETNs, IATNs and TNs. Thus, the VNFs and vDPs defined in a slice are always instantiated in the ETNs which may be located in the same or different 5G-XHaul areas. The rational is to limit the virtualization layer to the edge of the network, while keeping the forwarding elements in the transport network, TNs and IATNs, as simple as possible. Notice that layer two segments in a slice can be defined regardless of the actual ETN where a VNF/vDP is attached to. Indeed the 5G-XHaul control should support that VNFs/vDPs move to a different ETN.

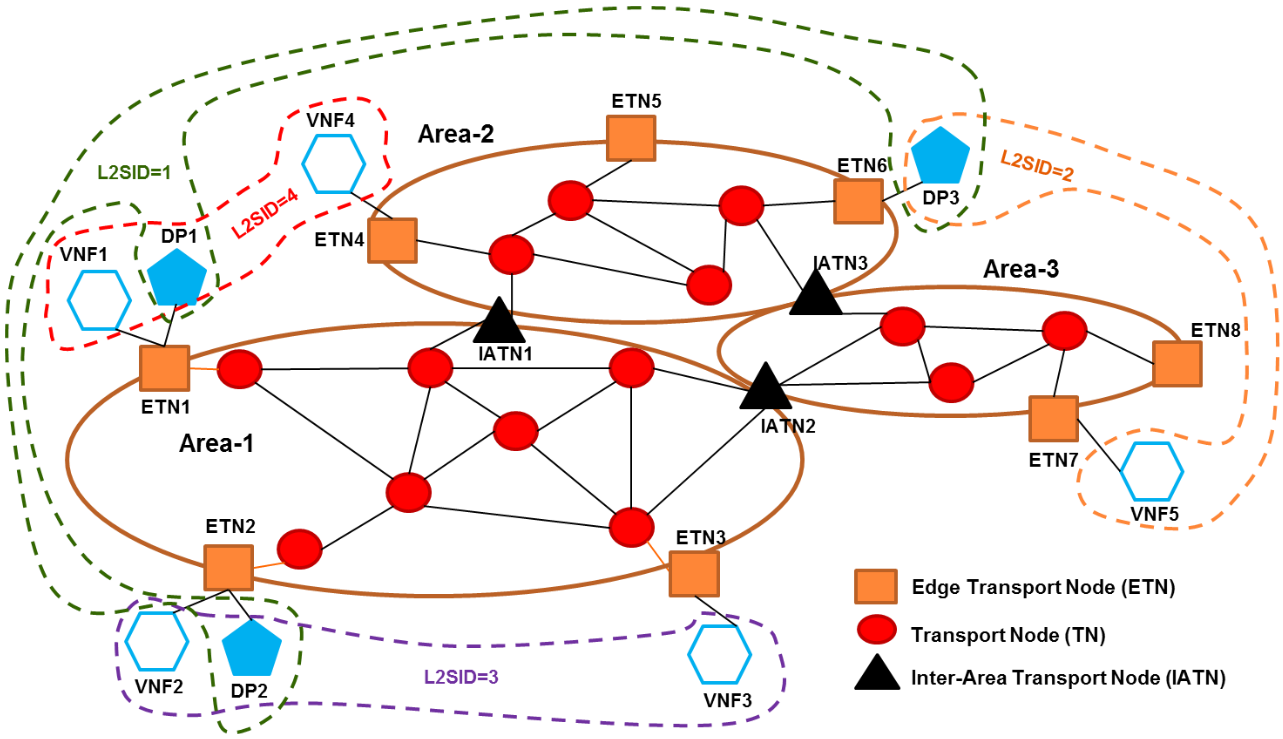

Figure 11: Instantiation of a 5G-XHaul slice over the physical infrastructure. All virtual elements are instantiated over ETNs, which will be connected through the infrastructure.

The main task of the virtualisation solution defined in 5G-XHaul is to "wire" the forwarding information base (FIB) in the participating ETNs (c.f. Figure 9), in order to enable the connectivity defined by the tenants in a slice. For this purpose, the fundamental bindings that need to be maintained by an ETN are the following:

- On the TX side: ***<L2SID, VNF_@/vDP_@> → destination ETN**_@*. Where "destination ETN_@" is the transport address of the ETN where this particular VNF/vDP is located.
- On the RX side: ***<L2SID, VNF_@/vDP_@> → ETN port.*** Where "ETN port" is the internal interface in the receiving ETN connecting to the VNF/vDP having the address equal to VNF_@/vDP_@ for the layer two segment with ID equal to L2SID.

A control plane function is assumed to deliver these bindings. However, hereafter we assume that these bindings are already populated in the ETNs, and leave as future work the definition and evaluation of this control plane function.

Another binding can be derived from the fact that the L2SID is unique in the system, namely L2SID can be linked to the tenant instantiating that layer two segment. This is important, because having an explicit tenant ID inside the packet is useful to provision per-tenant policies in the ETNs or even inside the transport network (TNs and IATNs).

Figure 12 provides a conceptual overview of the steps followed by a packet inside the ETN datapath[4]. Notice though that Figure 12 only deals with the data plane components of the ETN, namely the Forwarding Information Base (FIB) and TAF defined in Section 2.2.

On the lower left corner of Figure 12 we can see that an ETN is pre-provisioned with a set of interfaces, sitting between the FIB and the TAF blocks (c.f. Figure 9), representing transport tunnels to all the ETNs and IATNs in the same 5G-XHaul area of this ETN, or representing a multicast tree grouping some of the aforementioned ETNs/IATNs in the 5G-XHaul area. For example, the datapath of ETN1 in Figure 12 would contain pre-instantiated tunnels to ETN2, ETN3, IATN1, IATN3 and possibly multicast trees grouping some of the previous nodes. On the lower right corner of Figure 12 we can see how VNFs, which may belong to different slices, connect to the ETN FIB. For the purpose of this description a VNF can be understood as a Virtual Machine[5] executing a tenant specific logic connected to the 5G-XHaul slice.

Once the ETN FIB receives a packet from a VNF, the first action performed by the FIB is to insert in the packet header the corresponding L2SID; where there is a binding between the VNF port and the L2SID. It is assumed that this binding is provisioned when a VNF is added to a slice by a management system. A VNF generates Ethernet packets, having as Ethernet source address (ETH.SA) the address of the VNF in the slice where it operates, i.e. VNF_@, and as Ethernet destination address (ETH.DA), the address of a VNF or vDP in the same slice residing in the same layer two segment. Thus, a second stage in the ETNs FIB is to look up the tuple <L2SID, ETH.DA> and derive the transport address of the ETN where the VNF/vDP with address ETH.DA for the layer two segment L2SID is attached. The result of this look up can be twofold: i) if ETH.DA is attached to another ETN, the look up returns the ID of the internal tunnel that provides connectivity to the destination ETN, and ii) if ETH.DA is attached to the same ETN, for example if ETH.DA contains the address of a vDP instantiated in this ETN, the look up returns the internal interface or the follow up stage where the packet needs to be delivered. In the case that ETH.DA corresponds to vDP instantiated in this ETN, the packet goes through the corresponding virtual datapath that may modify the packet headers, including the L2SID field. For example, a virtual datapath instantiating an IP router would modify the Ethernet and LS2ID fields. After executing the virtual datapath a look up is performed based on the <LS2ID, ETH.DA> fields of the resulting packet to determine the transport address of the final ETN where the VNF/vDP with address ETH.DA in this layer two segment is attached, and the tunnel ID required to reach the destination ETN.

There is an important point to consider in the datapath design described in Figure 12. As previously explained an ETN has pre-provisioned tunnels *only* to the ETNs and IATNs inside the same 5G-XHaul area. However, the destination ETN for a given packet may be located in a different 5G-XHaul area. In this case, the packet should be routed to an IATN in the 5G-XHaul area that will later on forward the packet to the next area in the direction of the destination ETN. Notice that restricting the transport tunnels to only the ETNs and IATNs in a given area is beneficial because reduces the FIB pressure on the TNs/IATNs, which only need to maintain state to reach the other ETNs/IATNs inside the same area. However, it poses a problem when the destination ETN is in a different area, because both the transport address of the final destination ETN and the one of the destination IATN inside the area should be included in the transport header pushed by the TAF; otherwise the IATN receiving the packet would not be able to continue forwarding the packet until the final destination ETN. Whether transporting both addresses in the transport header is possible will depend on the actual transport technology used in the underlay. Next, we discuss an implementation assuming an Ethernet transport underlay based on a MAC in MAC (Provider Backbone Bridging) [16] TAF.

[4] This is a conceptual description, hence actual implementation may vary

[5] Or equivalent, e.g. container or unikernel

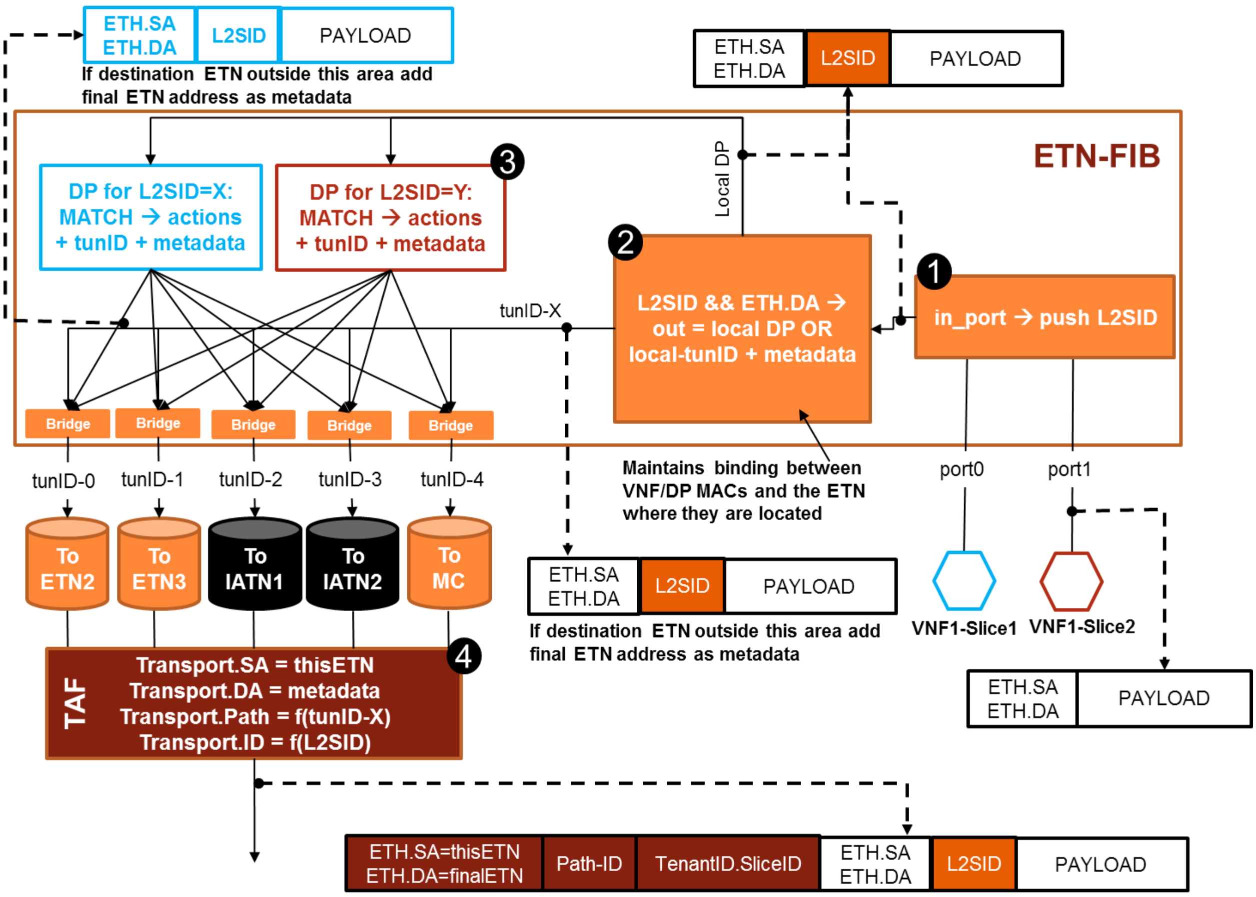

Figure 12: Functional view of an ETN transmission pipeline.

Figure 13 illustrates the header added by an Ethernet TAF. Essentially, the Ethernet TAF adds an outer Ethernet header consisting of an Ethernet source address, set to the address of the originating ETN, an Ethernet destination address, set to the address of the final destination ETN (not the intermediate IATN), a Backbone VLAN field (B-VID), whose value will be discussed later, and a 24 bit I-SID field.

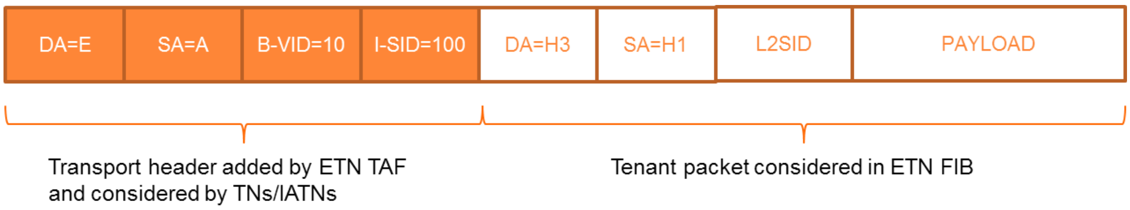

Figure 13: Proposed encapsulation in the Ethernet TAF.

The previous encoding leaves unsolved the problem of how to signal the destination IATN in the same 5G-XHaul area, in case that the destination ETN is located in a different area. For this purpose, the following provisional solution is considered. The B-VID field contains a 12 bit VLAN-ID field that can be used to encode up to 4096 different paths per each destination address within the same 5G-XHaul area. Notice that having multiple paths to a given destination is useful for load balancing purposes, or to implement different forwarding policies. For example, and ETN may decide to forward packets from different tenants through different paths. In any case, for the scenarios considered in 5G-XHaul, 4096 paths for each ETN/IATN within an area are considered excessive in practice. Therefore, part of the VLAN space in the B-VID field devoted to signal multiple paths for a given destination, can be segregated and used instead to signal a particular IATN as destination inside a given 5G-XHaul area. For example in the case of Area 1 in Figure 11 two IATNs are present. Hence, one could allocate 6 bits of the VLAN-ID in the B-VID field to signal paths to one of the IATNs, resulting in 32 possible paths to each IATN, and 64 possible paths to each ETN in that area.

Another important aspect to highlight in the design of the Ethernet TAF is that the I-SID field in the transport header depicted in Figure 13 can be used to encode a tenant identifier, which only needs to be unique inside a 5G-XHaul area. Encoding a tenant identifier in the transport header is useful because per-tenant policies can be applied both at the ETN and IATN level. For example the I-SID field could be hashed in the TAF to decide the particular path to a given destination to be followed, or the I-SID field can be used to hash different flows to different QoS classes, for example if Stochastic Fair Queuing (SFQ), or an equivalent mechanism, is applied in TNs and IATNs as a simple means to provide isolation between tenants.To conclude the description of this preliminary 5G-XHaul virtualisation solution, Figure 14 describes the receive datapath that allows the destination ETN to deliver the packet to the appropriate VNF. First, upon receiving a packet, the TAF checks if the destination address in the transport header coincides with the address of this ETN, and in that case pops the transport header and forwards the packet to a follow up stage. This second stage looks up the <L2SID, ETH.DA> pair, where ETH.DA is the one in the inner Ethernet header, and obtains the internal port of the locally attached VNF the packet needs to be delivered to, or the vDP in case this packet is addressed to a virtual datapath instantiated in this ETN. In the former case, the packet is delivered to the VNF after popping the L2SID field, and in the latter case the packet is processed according to the tenant's defined virtual datapath and transmitted back to the TX pipeline illustrated in Figure 12.

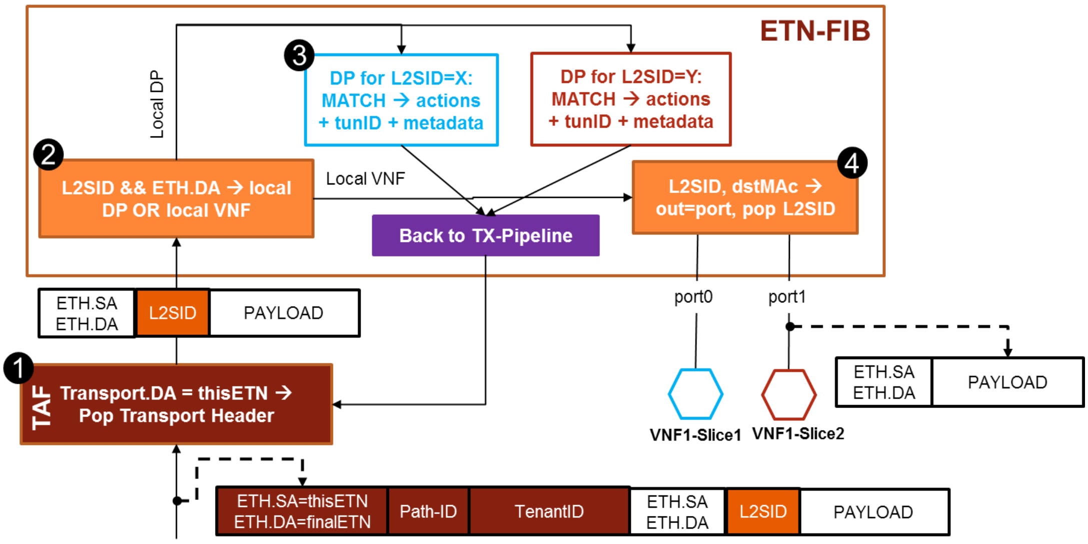

Figure 14: Functional view of the ETN reception pipeline.

4 QOS SUPPORT IN 5G-XHAUL

In this section we introduce the QoS mechanisms and algorithms that form the basis of the QoS capabilities to be supported in 5G-XHaul. This is an essential component of a carrier grade system, such as 5G-XHaul, which will enable a 5G-XHaul provider to enforce and guarantee SLAs to its tenants.

We structure this section in the following way. First, we survey in Section 4.1 technology specific QoS mechanisms included in each of the technologies considered in 5G-XHaul, namely mmWave and Sub6 in the wireless domain, TSON and WDM-PON in the optical domain, and Ethernet in the packet switch domain. Thus, Section 4.1 describes mechanisms included in the medium access layer of each technology devoted to support different types of QoS. Section 4.2 takes a closer look at the wireless domain, and surveys Traffic Engineering (TE) and QoS routing mechanisms proposed in the literature, which sets the scene for some of the developments that will be carried out in 5G-XHaul in upcoming deliverables. Section 4.3 focuses on inter-technology aspects, and specifically on how to provide consistent QoS across heterogeneous technology domains, which is one the main challenges faced by the design of the control plane in 5G-XHaul. Finally, section 4.4 extracts a set of requirements from the presented state of the art discussion, and introduces an initial overview of the 5G-XHaul QoS specific mechanisms that will be further elaborated on upcoming WP3 deliverables.

4.1 State of the Art on technology specific QoS aspects

We describe in this section specific mechanisms included in each technology considered in 5G-XHaul, which allow a network element to support different types of QoS, or QoS classes. We structure the discussion into QoS mechanisms for mmWave in Section 4.1.1, Sub-6 in Section 4.1.2, WDM-PON and TSON in section 4.1.3, and finally Ethernet in Section 4.1.4.

4.1.1 QoS support in mmWave

A meshed backhaul can be made by linking together a number of mm-Wave (802.11ad) local area networks. In 802.11ad a number of wireless stations (STAs) are grouped into a personal BSS (Basic Service Set) or PBSS. Within a PBSS, one STA is identified as the PBSS Control Point (PCP) and is responsible for managing the scheduling of transmissions between the STA of the PBSS. IEEE 802.11ad defines specific QoS-related mechanisms including contention-based access (contending station identities and time windows are defined by the PCP) and the possibility to follow a scheduled TDMA (Scheduled channel time allocation) and polling-based access (dynamic channel time allocation), which allow a more precise QoS provision than the typical contention-based access of the 802.11 family [38].

All STAs within a PBSS operate on the same wireless carrier frequency. Backhaul traffic is passed to/from the small cell from/to the core-network by exploiting a cascade of PBSS wireless links, interconnected by the switch elements of the mesh node. This is a multi-hop network, where the performance of each hop for a given data flow depends on a number of factors such as the load of competing flows on the PBSS, the physical data rate of the link (dependent on path loss and interference) and its error characteristics. Since line-of-sight operation is assumed, the mesh may require mesh nodes only placed to relay traffic without an associated small cell to backhaul.

Furthermore, meshes built using the IEEE 802.11ad (WiGig) technology are able to exploit the established QoS mechanisms of 802.11 based on traffic categories (TC) and traffic streams (TS). Individual MAC service data units (MSDUs) are identified by a traffic identifier (TID) value. There are 16 possible TID values, eight identify TCs, and the other identify parameterised TSs. The TID is assigned to an MSDU in the layers above the MAC.

TCs indicate the user priority (UP) of the MSDU. The 8 values (0 to 7) map directly to the 802.1D (Ethernet) priority tags. Each TS has an associated traffic specification (TSPEC) which details characteristics and QoS expectations of the flow.

In conclusion, the 802.11ad QoS mechanisms can provide rich information to allow differentiation between different fronthaul/backhaul flows (for example, flows of different traffic classes).

4.1.2 QoS support in Sub-6GHz

Nowadays IEEE 802.11-based technologies share the definition of a QoS provisioning framework established in the IEEE 802.11e amendment [39]. Notwithstanding, QoS related issues are permanently under continuous scrutiny [40], with the result of new amendments being released by the IEEE P802.11 working group.

IEEE 802.11e introduced new access mechanisms for single-hop radio links (client station ←→ access point), namely: Enhanced Distributed Channel Access (EDCA) and Hybrid Coordination Function Controlled Channel Access (HCCA).

HCCA is a centralized approach whereby the access point (AP) schedules transmissions through polling (the scheduling algorithm is implementation dependent). This coordinated scheduling enabled by HCCA allows a precise configuration of the QoS. However, support of HCCA is not mandatory and there are very few (if any) commercial implementations available today. Furthermore, HCCA cannot be applied to multi-hop networks, which define the configuration envisioned by 5G-XHaul for wireless backhauling.

According to the distributed scheme, EDCA, frames for transmission are classified into four different access categories (AC) according to 802.1D user priorities:

- **VO**: for voice (UP values 6 and 7).
- **VI**: for video (UP values 5 and 4).
- **BE**: for best effort (UP values 3 and 0).
- **BK**: for background traffic (UP values 2 and 1).

Later, IEEE 802.11aa [41] introduced intra-AC traffic differentiation which defines two new access categories in order to extend the granularity of EDCA with regards to voice and video traffic (A_VO for UP 7 and A_VI for UP 4). Besides, with IEEE 802.11aa Stream Classification Service, data streams can be arbitrarily mapped into access categories (i.e. classification not based on 802.1D).

Each access category is represented by a different transmission queue. With EDCA, the standard Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) access is honoured, where each access category plays with the same rules (i.e. random contention) but applies different access parameters to obtain a differentiated performance. Hence, EDCA cannot guarantee QoS levels (e.g. minimum throughput or bounded delay) but it is suitable for its application over multi-hop wireless paths with some caveats [42]. For multi-hop mesh networks, the IEEE 802.11s amendment [43] introduces yet another (optional) channel access scheme, called Mesh Coordination Function Controlled Channel Access (MCCA), whereby stations can reserve time intervals for periodic data transmission in a distributed manner. Yet, MCCA still relies on contention and, although notably reduced, it also shows the same endemic drawbacks of the CSMA (e.g. hidden node problem [44]).

In order to provide guaranteed QoS over a multi-hop wireless backhaul, the alternative to the traditional random contention of the IEEE 802.11 family consists in adopting a deterministic access such as TDMA. As a first step, it can be achieved in a distributed way, as proposed in [45], where the MAC combines 802.11's DCF random access and regular scheduling of packet transmissions in order to repeat the sequence of channel accesses which resulted successful. A similar approach is proposed in [46], which adds a more adaptable TDMA frame structure, depending on varying traffic conditions. However, it is accepted that a more precise control is achieved through a centralized radio resource management. In [47], authors demonstrate the benefits of a centralized TDMA scheduling over a set of IEEE 802.11-based access points. Although their implementation is focused on the RAN, the same idea could also be applied to the BH radio links. The drawback of this approach is the requirement of a very precise time synchronisation of the transport nodes and the central controller; recent works state that μs-level synchronisation can be achieved even over-the-air [48].

4.1.3 QoS support in TSON and WDM-PON

TSON is proposed as a metro network technology for efficient transport of packet traffic over WDM networks in the core. TSON supports TDM operation for Ethernet frames and consists of frames with certain number of time slices. In each frame the number of time slices and the duration of the time slice is configurable. With the flexibility of configuring time-slice duration as well as the frame size there is a wide range of steps in which TSON can support guaranteed data rate and latency. The details of the TSON operation and configuration using the southbound protocol are explained in Section 5.

The granularity of the traffic bandwidth per time slice which can be configured on Griffin [101] goes from 6.8 Mb/s up to 8.8 Gbps. Working with the time slice duration, the minimum latency that can be achieved having single time slice goes from 13.1 µs to 118.2 µs. Using different combinations of time-slice duration and TSON frame size, various levels of QoS can be achieved and hence TSON can act as a transport platform to map QoS between different technology domains (packet-to-optical or wireless-to-optical etc.). Details about the QoS capabilities of TSON technology can also be found in the relevant publications [100][101][102].

Being WDM-PON a physical layer (layer 1) technology, i.e. higher layers are not terminated in the WDM-PON domain, QoS information on higher layers is not accessible, and WDM-PON can be considered transparent regarding QoS. However, the WDM-PON based fronthaul considered in 5G-XHaul supports QoS monitoring by providing input on the health of the physical layer transport to the control plane, such as the operation temperature, transmitted/received optical power, as well as alarms on the outage or malfunction of the optical modules.

4.1.4 QoS support in Ethernet for joint backhaul and fronthaul

The FH network has stringent delay and jitter requirements; in 5G the BH network will also have critical delay requirement for some new emerging services such as tactile internet, while traditional Ethernet provides only best effort service in which congestion will be a big problem. If there is congestion, delay, jitter and packet loss will occur. A possible solution to resolve this problem is to pre-allocate a forwarding path for a fronthaul and/or backhaul flows, and reserve bandwidth along the path. Pre-allocating resources will avoid the problem of delay and packet loss due to congestion. MPLS(-TP) is ideal for this purpose, MPLS labels can be used for path definition, and bandwidth can be reserved along the path. The path definition and resource reservation can be achieved by a centralized SDN controller or by signalling such as RSVP-TE. Similar work for Ethernet is ongoing, namely 802.1Qca [53].

Conventional Ethernet/MPLS/IP forwarding do not take delay and jitter into account when looking for a path for a packet or flow. In 5G-XHaul, the SDN controller should know the network's delay and jitter value, or similar parameters, such as if [49] pre-emption is supported. The network elements should be able to report this capabilities/performance to the controller through a southbound protocol (c.f. Section 5). The SDN controller should also take delay and jitter capability into account when calculating a path.

In a conventional Ethernet architecture, QoS support is realized via scheduling algorithms, congestion control as well as traffic policing and shaping. However, requirements such as delay and jitter, reliability, etc, are not well addressed. In this regard, the IEEE TSN task group is developing new mechanisms and protocols for better QoS performance in Ethernet. The TSN work includes: 802.1Qbu[49], 802.1Qbv[50], 802.1Qcc[51], 802.1CB[52], and other planned work.

4.1.4.1 Ethernet Scheduling Algorithms

Scheduling algorithms include Priority Queue (PQ), Weighted Round Robin (WRR), Weighted Fair Queue (WFQ), (HQoS), etc.

In the PQ mechanism, packets are classified into queues with different priorities, and the queue with highest priority is served first. In a worst case, queues with low priority may starve, which may not be acceptable. Normally, the priority of a packet is set in the 3-bit ToS field in Ethernet header, which can support up to 8 priority levels.

WRR serves queues in a round-robin order, and empty queues are skipped. A weight value is used to allocate bandwidth to different queues. For example, if there are three queues: A, B and C, and their weights are 40%, 40% and 20% respectively; then in a cycle queue A and B will be served twice and queue C will be served once. WRR does not take packet size into account, which means the bandwidth allocation is not so accurate; flows with large packet size will actually have more bandwidth. WFQ [54] [55] is similar to WRR, but takes the packet length into account. WFQ supports fair bandwidth distribution for variable packet length.

Hierarchical Quality of Service (HQoS) suggested by [56] is a QoS technology that employs multiple levels of QoS scheduling. The flow classification can be based on C-VLAN, P-VLAN, CoS and Option 82, etc. PQ/WFQ can be used at each scheduling level. Figure 15 shows an example of multi-level scheduling from [56] (Figure 24).

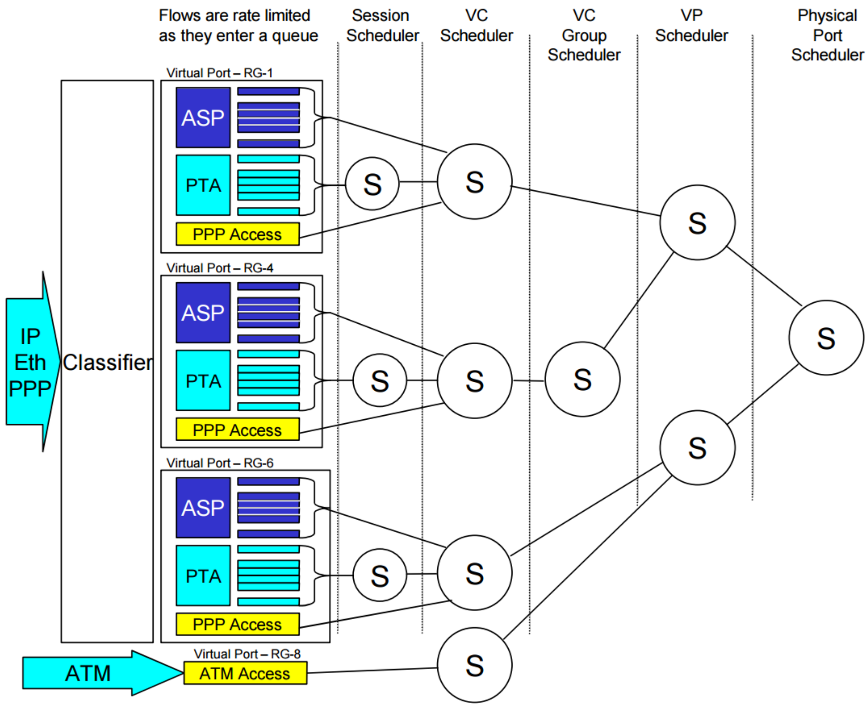

Figure 15: HQoS Architecture in [56] (Figure 24).

4.1.4.2 Ethernet Congestion Control

When the total amount of output traffic exceeds a physical threshold (e.g. a physical link), or a logical threshold (such as bandwidth for a tenant which is identified by a VLAN value, or a service identified by a priority value), congestion control methods must be considered. Congestion control is implemented in Ethernet via Tail Drop, Random Early Detection or PAUSE mechanisms.

In a network element, when there is congestion, packets in the tail of a queue are dropped by default. The problem of this method though, is that for TCP based communications, tail drop may introduce fluctuations in the network links, which lead to underutilized resources.

When the average queue buffer lengths are more than a threshold (e.g. 50%), Random Early Detection (RED) [57] begins to delete packets in the queue; as the buffer length increase, more packets are deleted. Weighted Random Early Detection (WRED) is an extension to RED. If there are multiple virtual queues in a port, WRED can be used, and there is a weight value attached to each virtual queue. At a certain threshold value, queues for low priority traffic can delete more packets; and the weight values are configurable. The advantage of RED/WRED is that, it starts dropping packets before the queue is full, hence avoiding significant fluctuations in network speed.

Ethernet PAUSE frame, as defined in Annex 31B of [58], can be used to inform the sender to stop sending data for a specified period of time. The downside of the PAUSE operation is that it may cause head-of-line (HOL) blocking, delay all the traffic in a link, rather than only the flows causing a problem. A solution is to implement multiple virtual output queues, e.g. output queues based on CoS or VLAN, and perform flow control on a virtual queue basis.

4.1.4.3 Ethernet Traffic Policing and Shaping

Traffic policing guarantees the committed data rate, and traffic shaping ensures that the traffic is sent at a smooth data rate while reducing traffic bursts.

Committed Access Rate (CAR) provides the network operator with the means to define bandwidth rate limits on an ingress port, egress port, or a virtual/sub port. CAR is implemented using tokens. For example when a packet is forwarded, certain amount tokens are removed; if there is no sufficient tokens for a packet, the packet is dropped.

In single-rate traffic policing, two token buckets are used. The capacity of one token bucket is the committed burst size (CBS). The capacity of the other token bucket is the peak burst size (PBS). Tokens are filled into the CBS bucket and the PBS bucket at the rate of CIR (Committed Information Rate). When the two buckets are full, extra tokens are dropped.

When the network traffic is complex, double rate traffic policing can be used, whereby tokens are filled into the CBS bucket at the rate of CIR and into the PBS bucket at the rate of PIR (Peak Information Rate).

Traffic shaping is similar to traffic policing, but with a Generic Traffic Shaping (GTS) queue. If a packet does not have sufficient tokens for transmission, then it can wait in the GTS queue for more tokens, rather than getting dropped immediately. Therefore, Traffic Shaping may increase the packet delay as compared to Traffic Policing.

4.1.4.4 Advanced QoS mechanisms in Ethernet

802.1Qbu [49] defines a mechanism for time-critical frames to suspend the transmission of a non-time-critical frame even if part of the frame is already on the wire, and send one or more time-critical frames before resuming the transmission of the non-time-critical frame. This pre-emption can be performed over a non-time-critical frame multiple times. 802.1Qbu can reduce the transmission delay and jitter significantly, which is critical to support fronthaul.

802.1Qbv [50] develops a scheduling mechanism for the transmission of frames based on timing derived from IEEE Std 802.1AS, in order to get a deterministic transmission delay.

The Stream Reservation Protocol (SRP) has been accepted by the professional, industrial, consumer, and automotive markets. 802.1Qcc [51] aims to add some enhancements to SRP which include support for more streams, support to layer 3 streams, deterministic stream reservation convergence, etc.

Finally, 802.1CB [52] defines solution for duplicate frame creation and elimination, which enhances reliability. This is useful in cases that there is packet drop or corrupt packet but retransmission does not work well.

4.2 State of the Art on Traffic Engineering (TE) and QoS routing in wireless transport networks

Given its specific weight in 5G-XHaul, in this section we survey TE and QoS routing for wireless transport networks. We start by surveying TE mechanisms in general in Section 4.2.1, while putting special emphasis on the proposals adopting the SDN architecture, which will be the basis for the 5G-XHaul control plane. Then, in Section 4.2.2 we survey previous work on QoS routing, both for the Sub-6 and mmWave wireless technologies considered in 5G-XHaul.

4.2.1 Traffic Engineering (TE) in wireless networks

Traffic engineering relates to extraction of traffic requirements and their respective placement [59]. Thus the goal of traffic engineering is to optimize load distributions and to minimize bandwidth consumption [60]. Traditionally, traffic engineering had been modelled with Poisson arrival models. The Poisson traffic model has worked well for voice traffic; however is not applicable to IP traffic which has different requirements captured by QoS requirements (e.g., end-to-end delay, jitter, or packet loss probability), in addition to the resiliency of the system to failures. The afore description mainly relates to the core network but the current trend in wireless networks, with the huge variety of accessed services in relation with the high network flexibility described in the 5G use cases makes traffic engineering important for them as well [61].

The key traffic engineering aspect in wireless networks studied in the past is the identification of the proper path/radio for certain services. However, the limited flexibility and service variety made the system modelling easier. Threshold based solutions for xMBB services manage to properly identify and distribute traffic, with the drawback of defining the service classes in an arbitrary way [62]. On top of the traffic characterization, multipath schemes distribute flows among paths thus ensuring resiliency one the one hand, at the cost of increasing complexity [62]. Proper traffic characterization may have even more benefits if multicast services are identified [63].

The previous schemes mainly target human traffic with limited number of service types and without considering emerging services. Thus, new proposals targeting these services are proposed in the literature [64][65]. These schemes target the definition of service classes specifically for these new services using e.g., data mining schemes thus avoiding the arbitrary selection of the service classes and using a more objective way to extract the traffic differentiation thresholds. However, these schemes target only certain service types and do not use a holistic framework that could be extended to other services as well. Additionally, these schemes, are not flexible enough to identify every potential new service class and focus on specific service types (e.g., in [65] advanced meter reading, distributed energy resources handling for e.g. wind and solar parks, and demand and supply side management are considered).

Apart from the flexibility required by 5G services, traffic engineering solutions also have to consider dynamic deployment or operational conditions. For example, backhaul limitations as well as changing backhaul link characteristics have to be considered when placing the traffic in the respective links and when identifying optimum paths (in case of multihop) [66]. Thus, traffic engineering schemes have to consider the varying nature of links on the one hand and the traffic characteristics on the other so as to place them in the respective links [67].

One potential way for handling the above challenges is the analysis of past actions of the UEs in various networking environments and the extraction of behavioural patterns. This, since it is based on the past behaviour of the users will enable more accurate predictions and more proper extraction of service classes – these classes could consist of combinations of services (such as video streaming and VoIP). The combination of the previous mechanisms with additional context information (such as location, date, weekday, time period, etc.) could enable more accurate decisions for both for access and backhaul. On the other hand the cost for gathering past user behavioural information and the extraction of the user behavioural profiles is a demanding process that has to be carefully optimized to avoid introducing further bottlenecks in the network operation.

4.2.1.1 State of the Art for TE in SDN-based Wireless Systems

Although traffic engineering (TE) has been studied extensively under ATM, IP/MPLS networks in the past, the unique features of SDN require SDN-specific TE methods [68]. Moreover, there have been very few studies on QoS/TE for SDN-based wireless networks, which also require specific solutions due to the lossy behaviour of wireless links, temporary congestions due to fading, and in-network interference between the links.

The three main components of TE for such networks were a) Admission Control, which decides whether to accept or reject a new connection evaluating whether its QoS requirements can be satisfied, b) Bandwidth Enforcement, through well-known traffic shaping methods such as Leaky Bucket, and c) Traffic classification. In IP, route optimization was the focus of TE solutions, mostly focusing on load balancing or shortest path based methods.

In [70], a hybrid architecture comprising a distributed OLSR daemon to configure the in band control network and a centralized SDN controller to configure the data plane is proposed, while demonstrating an Internet gateway balancing policy. In [69] an architecture for wireless backhauling, where the SDN controller operates on an abstracted view of the mesh network is proposed and is only used to configure end-to-end flows. Compared to these studies, [88] is the first paper to propose forwarding policies that consider both network and available computing resources enabled by specific OpenFlow wireless extensions.

In [88], we have devised an SDN-based solution for wireless mesh networks, proposing two SDN forwarding policies for multi-radio mesh nodes that can mitigate precarious network states (e.g. link congestion). The proposed policies consider both network state and available computing resources in the network elements, and have been evaluated in a testbed of constrained devices (Raspberry Pi). Through experiments, we showed that the proposed policies can mitigate external interference, achieve flow balancing, and are lightweight enough to work on constrained devices.

4.2.2 State of the Art in QoS routing for wireless transport systems

In wireless backhaul networks, one of the key challenges is the QoS provisioning in a multi-radio multi-channel backhaul network [85]. The nodes in the wireless mesh backhaul are going to support different applications (video, voice) with different QoS requirements. Therefore the provisioning of QoS considering different applications is a major challenge that should be further investigated. To this end, intelligent mechanisms are required to achieve efficient channel utilization, load balancing and network capacity.

In multi-hop backhaul, the way that packets can be routed across the network should be further investigated to ensure that the path selection not only considers links of high bandwidth, but also takes into account link stability to avoid frequent route fluctuations and channel diversity to minimize interference. Moreover, routing algorithms can provide balanced load at backhaul nodes to prevent bottlenecks.

To this end, routing and scheduling can be seen as two key operations that can provide high resource utilization, load balancing and capacity in dense small cell networks. In the state-of-the art literature, multiple routing and scheduling schemes were proposed and grouped [86] for different implementation options (e.g. whether routing and scheduling is separated or jointly performed in distributed or centralized manner).

The following sub-sections give an extensive SotA of QoS routing and scheduling algorithms assuming different backhaul technologies (e.g. Sub-6 GHz and mmWave), which are the key wireless technologies considered in 5G-XHaul.

4.2.2.1 Sub-6 GHz QoS routing algorithms

In the literature, wireless backhaul systems are mainly investigated under the Wireless Mesh Networks (WMN) topic [79]. WMNs define a wireless network, where the end-to-end communication is done through multi-hop communication of the WMN nodes. WMNs are self-organized and self-configured, with WMN nodes automatically establishing and maintaining mesh connectivity among themselves. Although there are different communication technology options for WMNs, IEEE 802.11 is the substrate that is studied most commonly due to the wide-spread availability of IEEE 802.11 devices and the related WMN standard of IEEE 802.11s, which includes (non-QoS) algorithms for multi-hop communication.

In this subsection, we present the state of the art on Sub-6 GHz routing algorithms, pointing out the design principles of such algorithms and the related requirements incurred by 5G-XHaul project. First, we provide the design principles of such algorithms:

- WMN nodes can be multi-channel, for which the joint channel assignment and routing algorithm can provide the most efficient solution.
- WMN nodes can have multiple radios, allowing simultaneous communication at different channels.
- A routing algorithm might define a multi-path routing, where the communication between the source and destination is done using multiple routes. This option can improve reliability, if the paths are chosen accordingly.
- The routing metric used to choose between alternative routes is crucial for end-to-end performance.

Most studies on QoS-aware wireless multi-hop routing focus on mobile ad hoc networks (MANET). However, since these studies consider mainly unlicensed bands, and hence uncontrolled interference, they mostly aim to improve certain performance metrics such as delay instead of providing QoS guarantees. In [71], the authors introduced a distributed joint channel assignment and routing protocol in multi-channel multi-radio ad hoc networks, where at each hop a local optimization is performed by selecting the least interfered channel according to a channel interference index. To find the least interfered path for network load balancing on a global scale, a length-constrained widest-path routing is done. The proposed approach is shown to improve system goodputs and end-to-end delays. In [72], a fuzzy logic solution is proposed for improving traffic regulation and the control of congestion to support both real-time multimedia (audio/video) services and non-real-time traffic services for MANETs. In [73], the lack of realistic lower layer models in previous works has been highlighted and a QoS-aware routing protocol (and admission control mechanism) considering mobility, shadowing, and varying Signal to Interference plus Noise Ratio (SINR) has been proposed. It is shown that proactively maintaining backup routes for active sessions, adapting transmission rates, and routing around temporarily low-SINR links can improve the reliability of services.

In [74], the authors proposed a QoS-aware routing protocol for multi-radio MANET to support real-time multimedia communication. In this proposal, channel usage info is embedded in control messages of OLSRv2, so that each node can obtain a view of topology and bandwidth information of the whole network. Based on the obtained information, a source node determines a logical path with the maximum available bandwidth to satisfy application QoS requirements. In this way, congested links are avoided and the load on the network can be distributed.

Among the few studies on QoS routing in WMNs, in [89] the importance of accurate bandwidth estimation is highlighted for QoS routing protocols for which, a method for available bandwidth estimation, based on dual carrier sensing and packet probing is proposed. In addition, a hop-by-hop QoS routing protocol, enabling alternate route identification when shortest paths are congested is derived. Through simulations, the improvement achieved by such approach in the overall system throughput is shown. In [75], an on-demand bandwidth-constrained QoS routing protocol for multi-radio multi-rate multi-channel WMNs with the IEEE 802.11 DCF MAC protocol is proposed. A distributed threshold-triggered bandwidth estimation scheme is used, which predicts the residual bandwidth of a path with the consideration of inter-flow and intra-flow interference. Through simulations, it is shown that the proposed approach can discover paths that meet the end-to-end bandwidth requirements of flows (admission control) and protect existing flows from QoS violations.

In [76], the problem of QoS provisioning in terms of end-to-end bandwidth allocation in WMNs is studied. It is shown that the problem of finding a feasible path is NP-complete under the considered interference models. A k-shortest path based algorithmic framework is used to find the optimal solutions through formulating them as optimization models. An on-line dynamic routing is also proposed and is shown to have a comparable performance to the optimal QoS routing algorithm. Moreover, based on their results, authors claim that, contrary to wireline networks, minimizing resource consumption should be preferred over load distribution even in lightly loaded WMNs.

In [77], the problem of identifying the maximum available bandwidth path in WMNs is studied, and it is proven that the proposed hop-by-hop routing protocol could satisfy the consistency and loop-freeness requirements. The consistency property guarantees that each node makes a proper packet forwarding decision, so that a data packet does traverse over the intended path. Through simulations, it is shown that the proposed available path bandwidth metric outperforms existing path metrics in identifying high-throughput paths.

In [78], a joint QoS-aware routing protocol and channel assignment is proposed for IEEE 802.11 multi-radio systems. The method is built on OLSR and uses a distributed bandwidth estimation, where each node estimates the available bandwidth from channel busy states. The bandwidth information is piggybacked to HELLO and topology control (TC) messages for dissemination of topology and available bandwidth information. Simulation-based evaluations show significant performance improvement compared to single radio OLSR, multi-radio OLSR and OLSR with differentiated services (DiffServ) in terms of network aggregate throughput, end-to-end packet delivery ratio, delay and delay jitter.

As a result, the QoS routing studies for MANETs consider the varying topologies and link conditions along with the lack of a centralized unit. On the other hand, the QoS routing studies on WMNs consider a static topology, yet, an environment with uncontrolled interference (unlicensed bands), while lacking a centralized unit. Hence, the proposed solutions mainly are distributed sub-optimal methods with no QoS guarantees due to the potential out-of-network interference. In 5G-XHaul, we will focus on licensed bands for Sub-6 GHz systems that can provide QoS guarantees and will develop centralized routing solutions, based on SDN principles, that satisfies the data rate and delay constraints for each flow.

4.2.2.2 mmWave QoS routing algorithms

In the literature, there are several works that focus on the use of millimeter wave communications in WPAN and outdoor mesh networks. Most of these studies turn the scheduling and routing into an optimization problem in which the objective function is generally to maximize throughput. As will be detailed in this subsection, common constraints considered in these formulations are the delay and traffic demands of individual flows.

In [80] the concurrent transmission scheduling problem was introduced. Since millimeter wave communications provide highly directional antennas and great amount of bandwidth, there is a chance to exploit spatial time division multiple access to allow both interfering and non-interfering links to transmit simultaneously in the same time slot. Based on the SINR at each receiver, the corresponding flow throughputs were introduced in order to prioritize certain flows that need to be allocated above others. The optimal scheduling problem was formulated in which there are transmission requests of data from the nodes to the Controller and the controller is in charge of maximizing total throughput by finding the maximum number of flows scheduled, subject to minimum flow throughout requirements. A heuristic algorithm was proposed based on a slot by slot decision in which the idea is to try to schedule as many flows as possible in the network. To do this a hybrid multiple access of CSMA/CA and TDMA is defined, in which there is a super frame that consists of three phases: A beacon period for network synchronization and control messages, a contention access period used to transmit requests to the controller, and finally a channel time allocation period (CTAP) for data transmissions. The CTAP period contains timeslots that are allocated to certain flows depending on the optimization results, so the controller makes scheduling decisions based on the maximization of network throughput.

Other works focused on scheduling schemes that take into account interference suppression and beam searching mechanisms in order to again, maximize network throughput. In [81] another proposed scheduling algorithm was developed to avoid interference by using optimization algorithms based on SINR (Signal to Interference plus Noise Ratio) and SLNR (Signal to Leakage plus Noise Ratio) measurements and simple priority factor scheduling. A scenario was defined in which a picostation schedules beams to each user equipment (UE) on a given time slot. In the SINR based scheduling, the scheduler selects the beam with highest SINR in each iteration and computes the interference from other selected beams to the same user. In the same way, the SNLR based scheduling selects the highest SNLR at each step and computes the interference caused by this particular beam to other users. In the conventional priority factor scheduling (PF), each UE is scheduled to transmit depending on a priority factor that relates the instantaneous data rate to an average data rate of user *i*. Simulations showed that SNLR, SINR and PF scheduling schemes function in a better way that conventional Round Robin scheduling in which all beams associated to each UE are divided into groups and each group is assigned a time slot.

Other works like [82] add to the optimization problem of concurrent transmissions the idea of beam-searching, and the fact that an exhaustive search for highly directive beam alignment between receiver and transmitter introduces alignment overhead, which puts restrictions on the time needed to obtain a scheduling decision. Specifically, this work defines the search taking into account sector-level and beam-level beamwidths which need to be correctly dimensioned and optimizes throughput, taking special care to avoid increasing alignment overhead too much (i.e not so narrow beams). So, a joint beamwidth selection and transmission scheduling optimization is proposed in which the objective goal is to maximize system throughput needs to be resolved by the controller, restricting also time for beamwidth alignment.

In [83] a more practical scenario is considered, where the routing and scheduling using millimeter wave backhaul is addressed in which the objective is to select backhaul links and paths to maximize throughput and minimize delay for users in a network. This approach takes as main objective to design a dynamic link scheduling algorithm to maximize BH capacity for a given time window. In [83] a Central Unit (CU) serves as a controller and traffic aggregator for dense small cell networks, access points (s-AP) have been provided with access links for backhaul based on 60GHz. It is assumed that the channel knowledge is given in the CU. The optimization problem is the minimization of the total number of time slots, defining the ratio of the demand over the BH link capacity towards an s-AP. The objective is to find the paths that traffic should follow and links to be activated to maximize system performance. The novelty of this solution is to propose a two stage problem using LP relaxation: a path selection algorithm and a packet scheduling problem. After defining the time slot that each link is going to use, a scheduling algorithm decides how to forward packets throughout the network with a minimum number of hops. Packets are sent to their destinations and intermediate nodes store these packets in queues and in turn forward them to adjacent nodes. Simulation results were shown with interesting discoveries. Regarding the path selection algorithm findings were that in long-distance links Non-Line-of-Sight (NLoS) impacts performance and in the scenario of low number of paths between CU and s-APs, the short distance LOS links could increase throughput performance. Regarding the scheduling algorithms it was shown that the average time delay decreases with the number of paths created to each of the nodes. There is in fact a trade-off between maximizing throughput of the network and coping with maximum delay when more paths are created, because when more paths between CU and s-AP are defined, the time needed by the CU to send all traffic to all nodes increases.

In [84], the authors discussed the problem of minimum time length link scheduling in 60 GHz ad hoc wireless networks using directional antennas with directional beamforming, under both traffic demand and signal to interference and noise ratio constraints. They discussed both single hop and multi-hop scenarios. For the multi-hop scenario, it is assumed that some nodes do not have direct link to each other, e.g., if their distance is longer than the transmission range or if the LoS path between them is blocked. To reach these nodes, intermediate nodes are needed to relay the traffic. For this scenario, a complex problem formulation was developed, incorporating both route selection and flow conservation constraints.

Finally, a related problem is that of topology design for mmWave networks. Whilst the geographical location of wireless backhaul nodes may be known, often they are co-located with small cells, and an open design question is which nodes should connect to which. This dictates the toplogy of the network. Previous work has considered the merits of tree, ring and mesh topologies [87]. It concluded that meshes offered higher capacity and resilience. With a mmWave 802.11ad backhaul the topology design is the same as the design of the membership of individual PBSSs.

4.3 State of the Art on end-to-end and multi-technology QoS

In this section we review end to end QoS schemes that are relevant to the design of the 5G-XHaul control plane. Notice though that the 5G-XHaul infrastructure is comprised of multiple technology domains, but it is administered by a single entity. It is important to keep this in mind in the context of this section, because existent work on inter-domain or cross-domain often deals with interconnecting domains administered by different entities, e.g. different operators. The same distinction applies when referring to end-to-end QoS. In the context of 5G-XHaul the end-to-end context refers to the multiple technology domains embodied in a 5G-XHaul deployment. Nevertheless, existent work in inter-domain QoS still provides valuable insight into the design of the 5G-XHaul QoS aspects, and thus it is reviewed in this section.

The problem of guaranteeing QoS across multiple technology domains, can be broken up into two sub-problems: i) defining QoS classes that are consistent across domains, whereby a QoS class specifies a per-hop behavior in each forwarding element, and ii) computing end to end paths traversing different domains, which can be subject to QoS (bandwidth, delay) or reliability (disjoint forwarding elements, fast reroute) constraints. In addition, the end-to-end or inter-domain QoS problem has been addressed at the IP layer or at layers below IP. Even though 5G-XHaul proposes a forwarding plane that operates below the IP layer, in this section we also review inter-domain solutions for IP networks as they also offer valuable insights in the design of end-to-end QoS mechanisms for 5G-XHaul.

A building block for inter-domain QoS at the IP level is the use of DSCP markings [90]. DSCP codes are marked in the IP header (IPv4 or IPv6) and signal a QoS class for the forwarded packet. An example of supported classes can be found in [90]. Hence, inter-domain QoS is possible if each domain treats DSCP codes in a consistent matter, i.e. DSCP codes translate into a particular forwarding behavior (e.g. policy and scheduling) in the network elements of a domain. Below the IP layer other approaches exist, like the traffic classes (TCs) defined in 802.1Q for Ethernet, or the priority bits in the MPLS header [91]. In the case of 802.1Q eight traffic classes are defined, which offer a consistent framework across all IEEE 802 technologies. In practice upon entering an Ethernet domain the DSCP codes in the IP header are translated to an equivalent TC in the 802.1Q header; an exemplary mapping between DSCP and 802.1Q TCs is described in [92]. QoS mapping is also relevant in the wireless access domain. In the case of wireless IEEE 802 technologies, such as Wi-Fi, generated packets are already marked with the appropriate TC, which can then be mapped to an 802.1Q TC in a subsequent Ethernet segment. In the case of cellular networks alternative QoS signaling mechanisms have been defined. For example, LTE QoS is instantiated through the concept of a *bearer* that defines a particular treatment of the packets of a flow in the radio and transport network segments. In particular, in LTE, radio bearers are defined in the wireless segments (between UE and eNodeB), which are then mapped to a given DSCP code in the transport network, which is assumed to be an IP network. The eNodeB is the entity in the cellular network in charge of performing this mapping. When a packet is received from the radio, it is encapsulated into an IP tunnel between the eNode and a gateway in the core network, and the appropriate DSCP code is signaled in the outer IP header of this tunnel. In turn if the transport network provides an Ethernet interface, the DSCP code in the outer IP header is mapped to an appropriate TC in the Ethernet header [93], which allows transport equipment to provide the appropriate treatment to the packet.

The other problem that arises when defining end-to-end QoS mechanisms is the computation of paths crossing different administrative and/or technology domains. At the IP layer BGP is the path vector protocol used for inter-domain routing in the internet. BGP does not provide QoS mechanisms, hence some research works have proposed BGP extensions in this direction that allow to choose a route according to the advertised QoS metrics [94]. Below the IP layer BGP has also been extended to interconnect different optical domains using Optical BGP (OBGP) [95]. OBGP is a distributed approach allowing different domains to exchange availability of optical wavelengths. An alternative approach is the Path Computation Engine (PCE) architecture propose in the IETF [96]. The PCE architecture advocates for having a separate entity outside of the forwarding path in charge of computing paths in a domain, which is in line with the SDN architecture. Forwarding elements can then request the PCE server for routing paths, instead of computing the paths themselves. This architecture comes with certain advantages, like the fact that more powerful CPUs can be embedded in the PCE server in order to run complex path computation algorithms. In a multi-domain scenario, each domain is controlled by a different PCE, and PCEs from different domains coordinate with each other in order to define end to end paths. There are two main models for inter-PCE coordination, namely peer to peer or hierarchical. In [97] a peer to peer model for PCE interconnection is proposed where the originating PCE sequentially requests routes towards the destination domain to its peer PCEs, to finally select the best subsequent path. The hierarchical PCE model has been widely studied in the literature. For example a multi-layer hierarchical PCE control plane for routing in ASON networks has been proposed by the ITU [98], where parent PCEs aggregate wavelength availability information from child PCEs. In [99] a hierarchical PCE approach for end to end path computation in inter-domain GMPLS networks is proposed, where A PCE controlling a domain abstracts this domain as a single node towards its parent PCE. Thus, PCEs at the higher layers compute inter-domain routes with a simplified topology. Aggregating per domain topology for inter-domain path computation comes with the advantage of hiding topology details, and reducing the complexity to compute end to end paths. However, topology aggregation may result in sub-optimal end to end paths. In [100] two inter-domain PCE coordination schemes are compared that use different levels of topology aggregation. First, Backward Recursive PCE-Based Computation (BRPC) is considered where each domain is represented by a set of entry and exit Border Nodes (BNs). In BRPC the domain hosting the final destination node, referred to as D_n, starts computing the shortest paths within its domain between all its entry BNs and the destination node. Then, domain D_n delivers to the PCE in charge of the domain D_{n-1} a reduced topology consisting of the entry BNs in D_n and the final destination with appropriately weighted links. Hence, domain D_{n-1} solves the same shortest path problem between its entry BNs and the final destination and delivers the resulting simplified topology to domain D_{n-2}. This procedure is repeated until the domain hosting the source node is reached. BRPC is compared to a per-domain PCE approach where the origin domain, D_1, starts computing its shortest path towards the next domain D_2, which then solves its shortest path towards D_3 constrained by the entry node decided by D_1. This procedure is repeated until the domain hosting the destination node is reached. Notice that the per-domain approach is simpler but may result in suboptimal end to end paths.

4.4 5G-XHaul QoS Requirements and Initial Design

In this subsection, the fundamental requirements of QoS-based networks are analyzed within the scope of 5G backhaul/fronthaul networks, and the specific requirements that should be addressed within 5G-XHaul are derived (Section 4.5.1). Initial approaches that are being evaluated in 5G-XHaul to address specific requirements are provided in Section 4.5.2. These solutions will be further elaborated on the upcoming WP3 deliverables.

4.4.1 5G-XHaul QoS Requirements

According to the discussion presented in this section, Table 3 summarizes the requirements to be fulfilled by the QoS mechanisms defined in 5G-XHaul.

Table 3: 5G-XHaul QoS requirements.

Requirement	Technology domain	Summary	Description
R1	All domains	Traffic description	In 5G-Xhaul, the traffic description is crucial for proper allocation of network resources to support the required QoS. The description should include type of traffic (CBR, VBR, etc.) and the corresponding traffic parameters, such as Peak rate, Burst size, etc. Traffic specification (TSPEC) of 802.11ad detailed in Section 4.1.1 and the transport classes defined in D2.1 can be a baseline.
R2	All domains	QoS policy specification and contract	The QoS parameters to be included in the QoS contract (Service Level Agreement, SLA), and hence to be considered in 5G-XHaul, should be defined. These could relate to QoS guarantees, QoS monitoring, etc. The QoS mechanisms defined in 5G-XHaul should allow to de-

			fine QoS policies between different tenants at the transport level, and also should accommodate the different transport classes defined in D2.1 [24].
R3	All domains	QoS signalling	5G-XHaul should provide a solution for the signalling of QoS configuration updates within the network.
R4	All domains	QoS marking / Packet classification	5G-XHaul should provide a consistent QoS marking scheme that can be recognised across the different technology domains considered.
R5		QoS-aware Xhaul mechanisms	5G-XHaul will develop mechanisms that would enable the transport network to ensure that the QoS requirements are satisfied. A subset of related mechanisms is provided below.
R5.1	mmWave, sub6	Backhaul/Fronthaul Routing and Scheduling	5G-XHaul should provide scalable scheduling, 1) to allow efficient utilization of backhaul links by dynamically (de-)activating / steering beams towards different small cells (based on traffic and channel variations), 2) to establish and re-configure multi-hop paths to accommodate different traffic classes (based on per class requirements, load / energy consumption statistics)
R5.2	Ethernet, TSON	Priority Queuing and scheduling	5G-XHaul should support Priority Queuing on a port in order to give some flows highest priority; and scheduling algorithm such as WFQ, etc., in order to fair scheduling in other case
R5.3	Ethernet	Resource reservation	Use VLAN as a way of path definition and perform correspondent path calculation. Support the add /remove of VLAN tag and bandwidth reservation based on VLAN/Port
R5.4	All domains	Scalability	5G-XHaul should provide mechanisms to allow path computation across technology domains. A hierarchical PCE approach will be considered, where topology aggregation can be used to reduce complexity but should not result in sub-optimal paths. The considered inter-domain path computation should also allow to compute back up paths for fast recovery
R6	All domains	Admission control	Given a new flow or slice request, the admission control scheme should decide to accept or reject that new request. The acceptance should maintain the QoS of the already established connections.
R7	All domains	Traffic policing	The 5G-XHaul network must police entering traffic to detect any violation of the QoS contracts. Policing is performed on each packet before entering the network. Example solutions are described in Section 4.1.4.3.
R8	All domains	Traffic Shaping	If sources send traffic without shaping, packets may be detected as nonconforming and be

			discarded. Traffic shaping controls the shape of the traffic before sending it to the network. Example solutions are described in Section 4.1.4.3.
R9	All domains	Congestion Control and Buffer Management	A network can take either proactive or reactive measures to control congestion. Some example solutions are described in Section 4.1.4.2.

4.4.2 Initial design of 5G-XHaul QoS mechanisms

Based on the requirements listed in the previous section, in this section we introduce an initial design for the 5G-XHaul TE and QoS mechanisms on the mmWave and Ethernet domains. The presented mechanisms will be further developed in upcoming WP3 deliverables.

4.4.2.1 5G-XHaul routing and scheduling in mmWave transport networks

In 5G-XHaul, we devise a QoS-aware forwarding algorithm deployed in a mmWave SDN controller that will take into consideration the data rate and delay constraints of the traffic classes defined in D2.1. The considered objectives for this forwarding algorithm are:

- Maximize Throughput: Without any given flow demands, this objective can be used to initialize the network.
- Load Balancing (Min-Max Link Utilization): To distribute the load to the network, so the new demands will result in less route reconfigurations.
- Minimum Number of Resources Required: To minimize the airtime needed to carry a given demand set. An extension is Minimum Number of Nodes Required, to be able to turn off nodes.
- Minimum Number of Scheduling/Routing Table Updates for Link/Node Failures.

The devised QoS-aware forwarding algorithm will be formulated first as a mathematical optimization problem. Then, the use of an IP Solver or a sub-optimal heuristic method will be considered.

Although centralized QoS-aware forwarding algorithms can be implemented in SDN-enabled networks, one challenge to be tackled in 5G-XHaul is the maximum 50 ms – 250 ms of allowed service interruption (hence, failure recovery) requirement defined by NGMN. In an SDN architecture, the controller decides the forwarding rules based on the last known network state. Hence, controller-initiated path restoration depends on: Statistics collection interval, delay in statistics data communication, processing delay of the alternative routes, and delay in route update dissemination.

For such a quick failure recovery, in addition to centralized failure detection/recovery methods to be applied by the SDN controller, localized detection and recovery actions should be derived and implemented. Such local actions can also be necessary if the (wireless) connection to the controller is lost or the controller goes down. Since actions are taken locally, inconsistencies may occur that can lead to data path loops or dead ends. This could be avoided by only installing sets of backup paths that, when merging all possible edges, do not result in loops. For dead ends, the alternative path can be set to null to trigger back signalling or crankback routing in case of failures.

4.4.2.2 5G-XHaul QoS mechanisms for mmWave wireless transport

For the mmWave technology, 5G-XHaul considers both an existing technology, namely IEEE802.11ad based meshing, and a future mmWave technology available in the 5G time-frame. When using IEEE802.11ad a number of aspects are defined in the standard, particularly the physical layer and the MAC. However, the algorithms that exploit the MAC toolbox (e.g. scheduling of frame transmissions) are not standardised, and the networking of 802.11ad WLANs together to form the mesh is also subject to implementation. This section describes initial design thinking for an 802.11ad solution and for future technologies.

Examples of operation of the 802.11ad MAC is described are Deliverable D2.2 [6]. The scheduler, running in the PCP (coordinator) of each PBSS, is able to manage time allocations to meet the QoS needs of the different transport flows and transport classes. The following fields of the TSPEC for 802.11ad are particularly relevant to supporting QoS sensitive backhaul/fronthaul traffic:

- Traffic type: *can indicate Periodic traffic.*
- User Priority: *if prioritization is needed.*
- Minimum service interval: *minimum time between two successive service periods.*
- Maximum service interval: *maximum time between two successive service period.*
- Minimum Data Rate.
- Mean Data rate.
- Peak data rate.
- Burst size: *maximum burst size that can arrive at the peak data rate.*
- Delay bound: *maximum time in microseconds to transport an MSDU.*
- Reliability: *physical layer error-rate.*

Figure 16 shows a section of a BH network, with the transfer of packets from a small cell to a point-of-presence linked to the core network (CN). Packets are delivered over two hops from the highlighted small cell to the CN ingress point. With a suitable switch implementation, for example, using an ASIC, the switching delay can be very small (< 1μs), and we also neglect the time to transfer packets from the switch to the MAC (and vice versa). Therefore, the latency is dictated by the delay incurred over the wireless links.

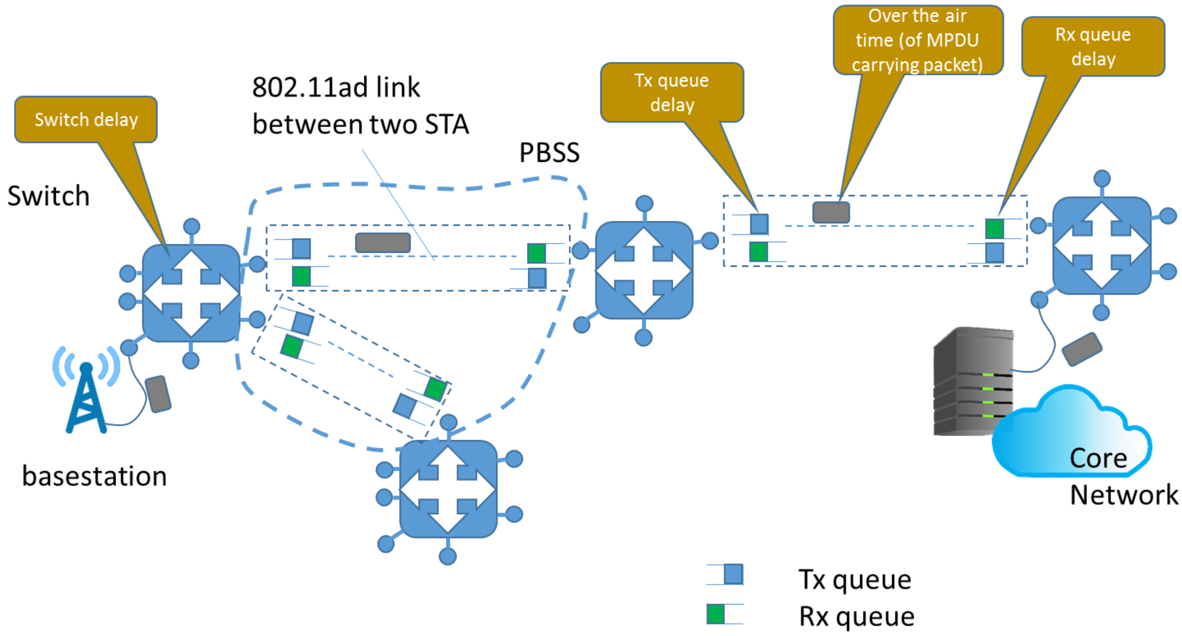

Figure 16: Example of an 802.11ad-based backhaul network with delay elements pin-pointed.

For the 802.11ad MAC a typical air interface frame transmission, from one STA to another, is shown in **Figure 17: An example of an air interface frame for 802.11ad.** The frame carries a MAC payload formed by aggregating MAC PDUs (MPDU) together into an AMPDU. After a guard interval (the guard interval is only present when switching between service periods), a PHY preamble is sent which includes training and channel estimation fields and a header that indicates the length of transmission, the MCS, Packet Type and other parameters. Following the payload blocks, there is a short inter-frame spacing (SIFS) followed by a transmission of an acknowledgement message, the block ACK, sent from the AMPDU receiver to the transmitter. This acknowledges the success/failure of each MPDU within the AMPDU.

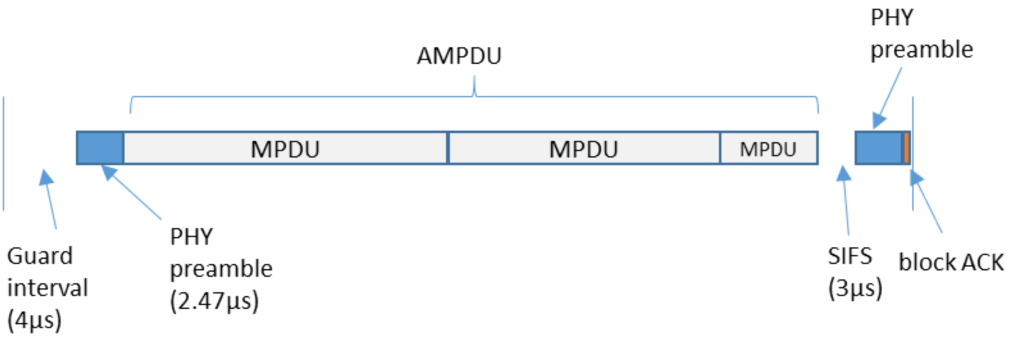

Figure 17: An example of an air interface frame for 802.11ad.

The overhead per air interface frame to transmit an AMPDU is approximately 4+2.47+3+2.47=11.94 μs (here we ignore the transmission time of the block ACK).

With respect to the scheduling of the air interface frames, a number of methods are supported by 802.11ad. These fit into a framing structure with a period called the beacon interval. The schedule is determined by the PCP. Allocations can either be service periods (SP) or contention-based access periods (CBAP). With an SP, the transmitting and receiving STA are specified. With CBAP, a pair of STAs are defined, and they contend for the air time. An example beacon interval is shown in Figure 18. The beacon header interval carries control signalling including one or more of beacon transmission, beamforming training, and announcement management frames.

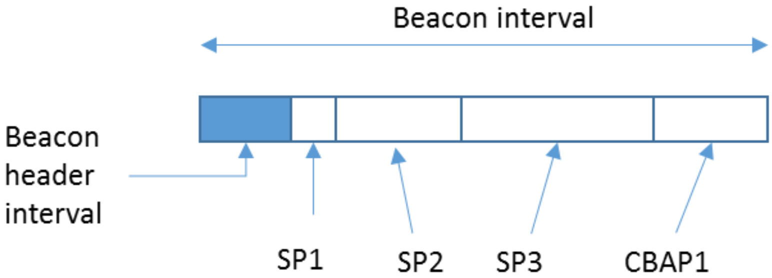

Figure 18: An example of a beacon interval.

SPs can be scheduled in three different ways (Figure 19):

i. Pseudo-static allocation – a periodically reoccurring SP for a given transmitter /receiver.

ii. Using announcement management frames at the start of a beacon interval.

Using dynamic scheduling, where a grant is issued before each SP.Using (i), the PCP can pre-allocate service periods for each (unidirectional) link of the PBSS to send data. This is essentially a circuit-switched air interface allocation. The advantages are that latency before access to the air for a given link is known a priori, and if the service periods are short the latency is also short. Furthermore, the overhead from control signalling is minimised since there is no need to poll STAs to determine air time requirements and grants are infrequent (only needed to refresh the pseudo-static allocation). On the other hand this method wastes air interface time if a link in the PBSS is unable to fully exploit its allocation, whilst at the same time data is queued up for other links.

Methods (ii) and (iii), in particular, allow flexible statistical multiplexing and packet scheduling. Whilst unnecessary idle times on the air can be avoided, the overhead from polling and grants must be considered.

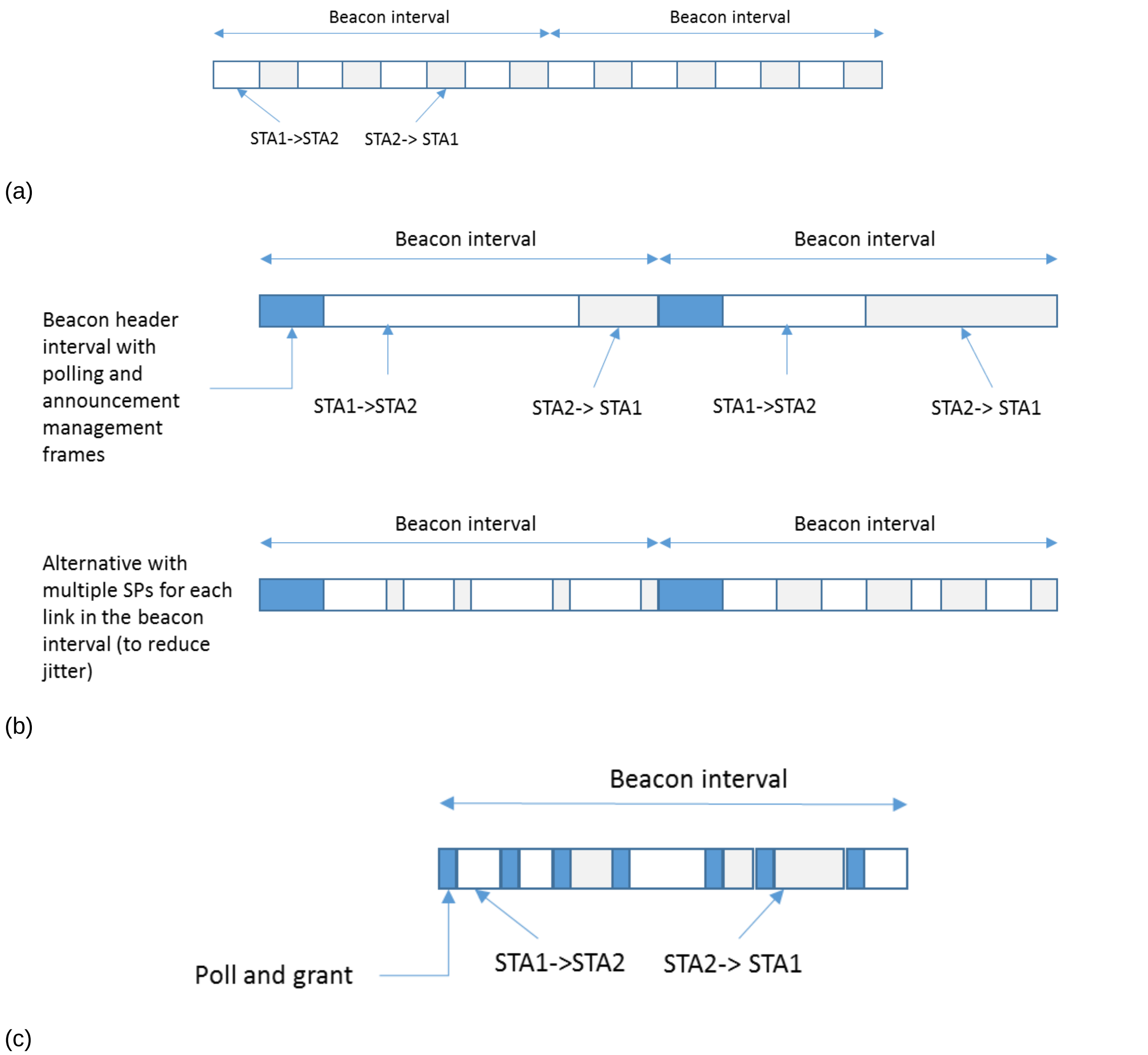

Figure 19: Different scheduling methods for an 802.11ad PBSS with two STA using service periods, (a) pseudo-static allocation, (b) announcement management frames, (c) dynamic scheduling. Note: control signalling not associated with scheduling is not shown – this would be common to all methods.

Allowing a throughput reduction permits lower latency because:

- The PCP is able to use air time freed up for frequent poll/grant signaling (method (iii)). Alternatively, able to accommodate idle air time of pseudo-static scheduling (method (i))
- The PCP can employ shorter and less efficient service periods so that the scheduler can react quickly and give air time to highest priority traffic. Additionally MPDUs can be shorter, and since data is not released from an MPDU until it has all been received (the error detection is per MPDU) this reduces latency of the payload packets.

The mean latency is linearly dependent on the scheduling period, as explained in Figure 20. At low scheduling periods the air time for transmission of MPDUs as a fraction of the beacon interval drops significantly and this results in the fall in the maximum sustainable load. Since link loading will tend to increase as data is backhauled from the small cell towards the ingress point to the core network, low latencies can be achieved for the initial hops (say < 50 µs if we assume negligible switching delay and negligible scheduling advance) with an increase for later hops (say 200 µs with the same assumptions). Note, this is a best-case if we assume the last hop links are running close to their nominal throughput (e.g. 4 Gbps for a MCS12 link) since the traffic is unidirectional, not very bursty and no errors are modelled. Also we have ignored other control signalling (beaconing and beamforming), and assumed a MCS12 link – with slower links the time overhead for poll/grant transmission grows because the code rate for these also drops, and the over the air time also increases. Note that this constraints the support of fronthaul traffic over mmWave technologies to topologies with a limited number of hops.

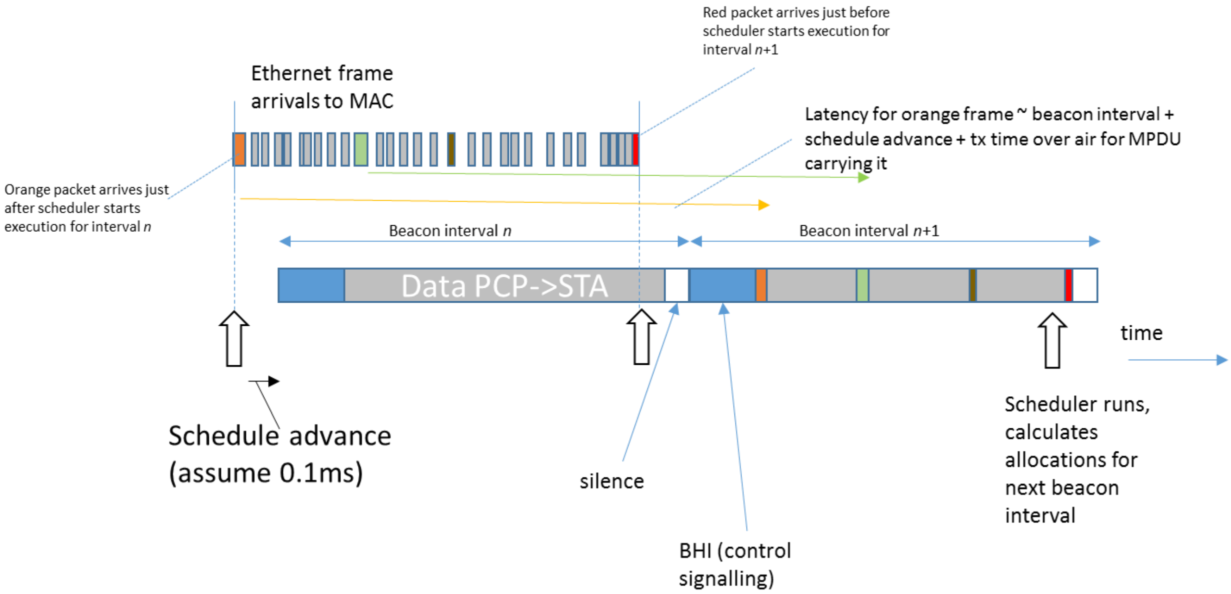

Figure 20. Explanation of linear dependency of 11ad latency on scheduling interval.

Future work will consider the support of more than one transport class over an 802.11ad mesh, with algorithms offering the differentiated QoS being developed and evaluated. Initial attention will be paid to the MAC, but extensions to the networking aspect (forwarding behavior of mesh nodes) is possible. The network topology will be assumed to be defined by an external planning tool and will not be addressed. Failover handling is also of interest, with the recovery time being dependent on the link status monitoring within the 802.11ad technology, and the forwarding correction using distributed and/or centralized intelligence.

For the case of two-level scheduling, which is discussed in Section 4.2, routing and scheduling should be supported as new functionalities at the SDN controller for the case of mmWave multi-hop mesh network. A physical deployment for such scenario could be the case of small cells which can also relay traffic to other small cells via flexible in-band backhauling. To this end, local schedulers (e.g. 802.11ad nodes) coordinate multiple small cells by providing local scheduling of data flows given different service requirements. Less granular scheduling is performed in global schedulers that coordinate clusters of 802.11ad schedulers. An exemplary physical deployment is shown in Figure 21.

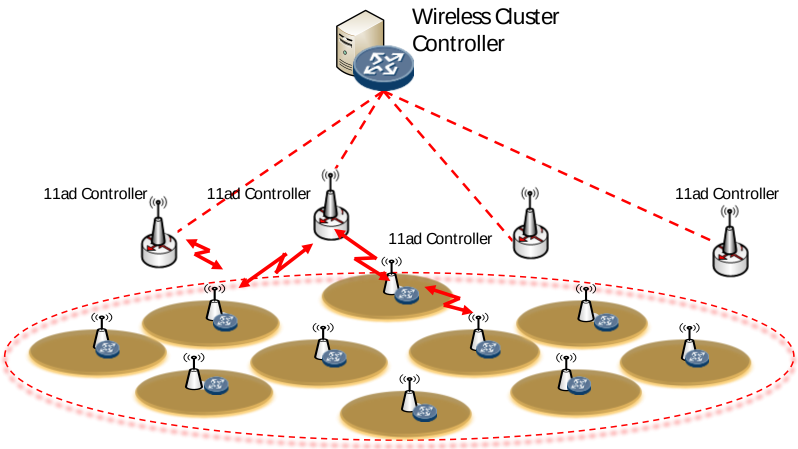

Figure 21: Physical deployment for two-level scheduling.

In the proposed two-level scheduling, the routing part, which performs the selection of the multi-hop paths to forward traffic with different QoS, is based on the average backhaul channel conditions and load/energy consumption long term statistics. In a more dynamic manner, flow scheduling is also supported to schedule the forwarding of data to next hops/users, in a way that the service-oriented KPIs can be met.

The algorithms to be used are going to be further studied in D3.2; however we provide here some initial considerations for the architecture and the inputs/outputs for the scheduler's design. Below, a high level message sequence chart is provided that shows what information is required and extracted by the schedulers. As can be seen, the Local Scheduler selects the per-flow time schedules and fast switch on/off decisions (e.g. for load balancing). In coarser time scale, the Cluster Scheduler selects which links/paths to be activated per traffic and destination.

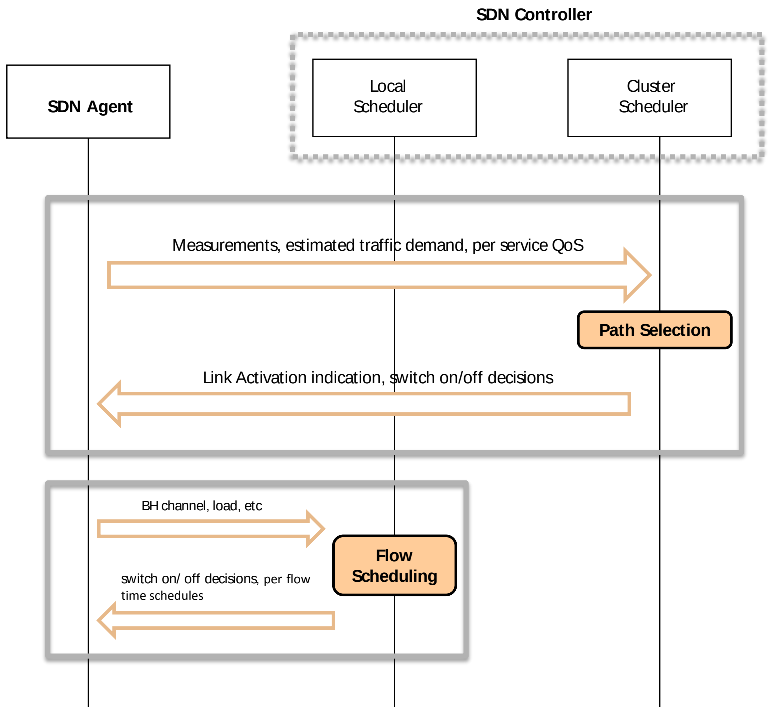

Figure 22: Two-level mmWave Scheduler's Inputs/Outputs.

4.4.2.3 Initial design considerations on 5G-XHaul Ethernet QoS mechanisms

Ethernet can be used for both fronthaul and backhaul. Traditionally Ethernet is widely used for best effort service. If Ethernet is used for fronthaul, in order to meet the critical delay and jitter requirements, a path must be defined and bandwidth reserved along the path for a fronthaul flow, so as to avoid congestion. In case of congestion, delay, jitter and packet loss will be a common problem.

This can be achieved by configuring VLAN. In an Ethernet switch, a VLAN will be configured between one uplink port and one downlink port. One VLAN Tag configured in multiple adjacent switches define a data path. Bandwidth should also be reserved for this VLAN on the port.

The 5G-XHaul level-0 SDN controller can then be used to calculate the path, configure VLAN tags, and make the bandwidth reservation along the path. In addition to resource reservation, priority queues can be used as an alternative and complementary to support different QoS.

5 SDN SOUTHBOUND INTERFACES OF 5G-XHAUL TRANSPORT TECHNOLOGIES

As the communications industry tries to keep up with the exponentially growing demand for mobile video traffic, advances in networking technologies need to take place in order for contemporary and future networks to accommodate this demand. These advances bring in significant complexity and the need for greater coordination among network elements, especially in Radio Access Networks (RANs), and fronthaul and backhaul transport networks, for efficient operation. In this context, 5G-XHaul proposes a unified control plane for a multi-technology transport network. Such unified control plane builds on the principles of SDN architecture, where a centralized control plane interacts with different technologies through technology specific southbound interfaces (c.f. Section 2.2.2). In this section we review southbound interfaces for the candidate wireless, optical and packet technologies considered in 5G-XHaul. In addition, we derive requirements for the these interfaces and provide an initial design for some of the southbound protocols considered in the 5G-XHaul.

This chapter is structured in the following way. In Section 5.1 we initially focus on the Sub-6 GHz technologies for wireless transport, and then describe southbound interfaces for mmWave technologies. In section 5.2 we describe the southbound protocols for fixed network transport technologies, specifically for Time Shared Optical Networks (TSON) and WDM-PON. In section 5.3 we describe southbound protocols to control Ethernet switches. We conclude in section 5.4 by deriving requirements for southbound interfaces and proposing an initial design for the technologies considered in 5G-XHaul.

5.1 State of the Art on Southbound Interfaces for Wireless Technologies

In this section we review SDN southbound interfaces for wireless Sub-6 GHz technologies on Section 5.1.1, and for mmWave technologies on Section 5.1.2. We survey both research works looking at more fundamental aspects of architecture and interface design, and at concrete protocol and interface definitions.

5.1.1 State of the Art on SDN southbound interfaces for sub-6GHz wireless technologies

Although 5G-XHaul focuses on wireless technologies for the transport network, in this section we present some of the state of the art approaches for SDN programmability of the wireless RAN, since the same techniques could be used in the transport. In particular, we focus on wireless technologies in the sub-6GHz bands. These include both approaches driven by the limitations of each technology, and built for a specific technology, and approaches that are more agnostic and applicable to several protocols for wireless communications.

5.1.1.1 LTE RAN SDN approaches

LTE's distributed control plane is suboptimal in the cases of dense deployments, with multiple mobile clients and limited licensed spectrum. In such cases, radio resource allocation, handover managements, interference management and load balancing become very difficult to manage and coordinate across the different LTE entities (base stations, core networks). To this aim, an approach for SoftRAN has been proposed [13]. SoftRAN defines a centralized control plane, by abstracting all the base station in a geographical area, as virtual base stations, consisting of a controller and radio elements.

In SoftRAN, the base stations are abstracted, and considered as just radio elements with a minimal control logic. The centralized controller is in charge of instrumenting their operation inside the dense network. The radio resources are abstracted as a 3D resource grid, based on the base station index, time and frequency. From the network operator's perspective, the base stations in a single geographical area are conceived as a single "big" base station, with the 3D resource grid.

The operation of the centralized controller is designated as follows: the radio elements inside an area, periodically send the local network state from their area and perspective. Given these updates, the local controller updates the "RAN Information Base", consisting of interference maps, flow records and network operator preferences, in case of prioritizing certain flows over others.

Nevertheless, the implementation of such a scheme in a real-network topology, using contemporary equipment seems to be infeasible, as the UEs will continuously sense the physically deployed base stations in cases of handovers, and will need to carry out the traditional handshaking procedures. A thorough approach would imply the restructuring of the LTE stack for both UE and base stations. Nevertheless, the application of such a scheme, through a centralized control plane would lead to smoother handovers, with less ping-pong effects, reduce dropped connections, as well as smoothen the transmission powers used by multiple base stations in dense setups. Moreover, a requirements analysis has been conducted regarding the backhaul bandwidth capacity needed for supporting the SoftRAN architecture. For a given backhaul capacity of 500Mbps, up to 20 base stations can be supported, with 50 clients per cell, and 10 flows per client.

Similar to the previous approach, OpenRAN [106] proposes an SDN network inspired by Cloud-RAN. It consists of wireless spectrum resource pools, a cloud computing resource pool and an SDN controller. The wireless spectrum pool is covering multiple heterogeneous wireless network. According to the dynamic network requirements, the OpenRAN SDN controller establishes the virtual base stations in the wireless spectrum resource pools, and the corresponding virtual baseband unit in the cloud, by automatically installing the PHY and MAC protocols. Other mobile network SDN approaches in the RAN include SoftCOM [107], an approach proposed by Huawei, which is a Cloud based architecture, leveraging SDN and NFV, to transform future networks and operations.

5.1.1.2 Wi-Fi SDN approaches

More specifically to WiFi, the Odin framework [108] has been proposed. Odin introduces programmability in Enterprise WLANs which need to support a wide range of services and functionalities, such as authentication, authorization, accounting and load balancing. Odin is implemented as follows: a master application is running on top of an OpenFlow controller and has a global view of flows in the network, clients connected to the network, and the infrastructure that comprises the network. Odin agents run on the WiFi APs. Together, the master and agents implement a Wi-Fi split-MAC (see CAPWAP[109]). A TCP connection is used between the agent and the master (established at boot time). It serves as Odin's control channel, and used by the master to invoke commands on the agents and collect statistics from them. Applications running on the Odin framework implement network services using interfaces exposed by the master.

OpenRF [112] is another solution applied to WiFi devices solely. It consists of a cross-layer architecture for managing MIMO signal processing. OpenRF enables access points on the same channel to cancel their interference at each other's clients, while beamforming their signal to their own clients. OpenRF is self-configuring, so that network administrators need not understand MIMO or physical layer techniques. The Intel WiFi driver is used to support OpenRF on off-the-shelf Intel cards. OpenRF is able to manage the complex interaction of cross-layer design with a real network stack, TCP, bursty traffic, and real applications.

5.1.1.3 Technology agnostic wireless RAN SDN approaches

Similar to SoftRAN, OpenRadio [105] has been proposed but with applications to both LTE and WIFi. OpenRadio is an approach for a programmable wireless network dataplane. OpenRadio is abstracting the PHY and MAC layers through a programmatic interface, so that an operator is able to describe a protocol by defining rules to match subsets of traffic streams, and actions in order to process them. Example of such actions can be encoding/decoding data, scheduling traffic on the channel, etc. OpenRadio's declarative interface is built on a principled refactoring of decision and processing paths in wireless protocols. Specifically, wireless protocols are finely partitioned into two separate parts: processing blocks which specify how a protocol transforms analog waveforms into bits, and decision logic which specifies when different processing transformations are used. For example, the 54 Mbps OFDM decoding sequence of 802.11g is a processing specification, while interpreting the signal field of the 802.11 PHY layer control header to determine that the payload will be at 54 Mbps is a decision specification. The processing plane encapsulates the series of signal processing algorithms used in a PHY processing chain where each module corresponds to a single algorithm. For example, the processing plane will have modules for FFT, 64-QAM mapping and slicing, convolutional encoding, Viterbi decoding, etc. These modules correspond to blocks in a signal processing chain. The decision plane encapsulates all the decision logic functionality.

OpenRadio has been designed in order to utilize multi-core DSP architectures. Nevertheless, the majority of the computation is spent in the processing plane of the compatible protocols (LTE/WiFi/3G), while the decision overhead is negligible. Heavy decision problems can however incur inter-core communication overheads, thus implying that the decision/processing plane separation is not the best approach.

5.1.2 Protocols to implement Southbound Interfaces for Sub-6 Technologies

A number of technologies are utilized per each method for the low level handling of the wireless networking technologies, each one speaking to a different API, or accessing variable complexity parameters. Some of the demonstrated southbound interfaces are the following:

- SNMP: Simple Network Management Protocol (SNMP) is an Internet-standard protocol for collecting and organizing information about managed devices on IP networks and for modifying that information to change device behaviour. Through standardized interfaces and objects, configuration parameters of the wireless device can be accessed (as long as the device is SNMP compliant). Approaches leveraging the SNMP protocol for low level SDN support include services that are deployed in testbed environments (e.g. [113],[114],[115]), as well as the obsolete version of OpenRoads (OpenFlow Wireless) [116].

- Interaction with the wireless driver: This solution is applicable only to devices that support an open source driver (mainly WiFi devices). Such solutions are adopted by the OpenRadio, OpenRF and Odin frameworks. The southbound interface usually consists of low level commands for communicating with the wireless device, or via the proc filesystem for the interaction of the kernel space with the user space. Other solutions include the provisioning of APIs directly inside the wireless driver, such as the one adopted by [117], where very low parameters e.g. the Interframe space for WiFi can be configured or monitored.

- Technology Specific APIs: This solution is more universal, and relies upon the existing interfaces that commercial off-the-shelf equipment provision. For example, in [118] authors have developed a service for accessing and configuring parameters in different equipment for the LTE network. Depending on the components, either JSON APIs or secure shell commands are used for configuring or reading parameters from the equipment. The authors bundle their service with Open-vSwitch, for enabling also OpenFlow configurations beyond the LTE network.

5.1.3 State of the Art on mmWave southbound interfaces

A mmWave meshed backhaul solution has been developed by Interdigital within the 5G-Crosshaul project [118]. The mesh is controlled by a central controller running on OpenDaylight, whilst the mesh nodes implement an OpenVSwitch with an OpenFlow southbound interface, Figure 23. Mesh control includes self-organization of the multi-hop topology, low-latency routing, QoS support, and failover management for high reliability, Figure 24. Based on the information available it appears though that the proposed SDN mesh solution uses standard OpenFlow for forwarding, which is agnostic to wireless conditions. Therefore, in the proposed solution the SDN controller cannot deploy traffic engineering algorithms able to cope with the variability of the wireless medium.

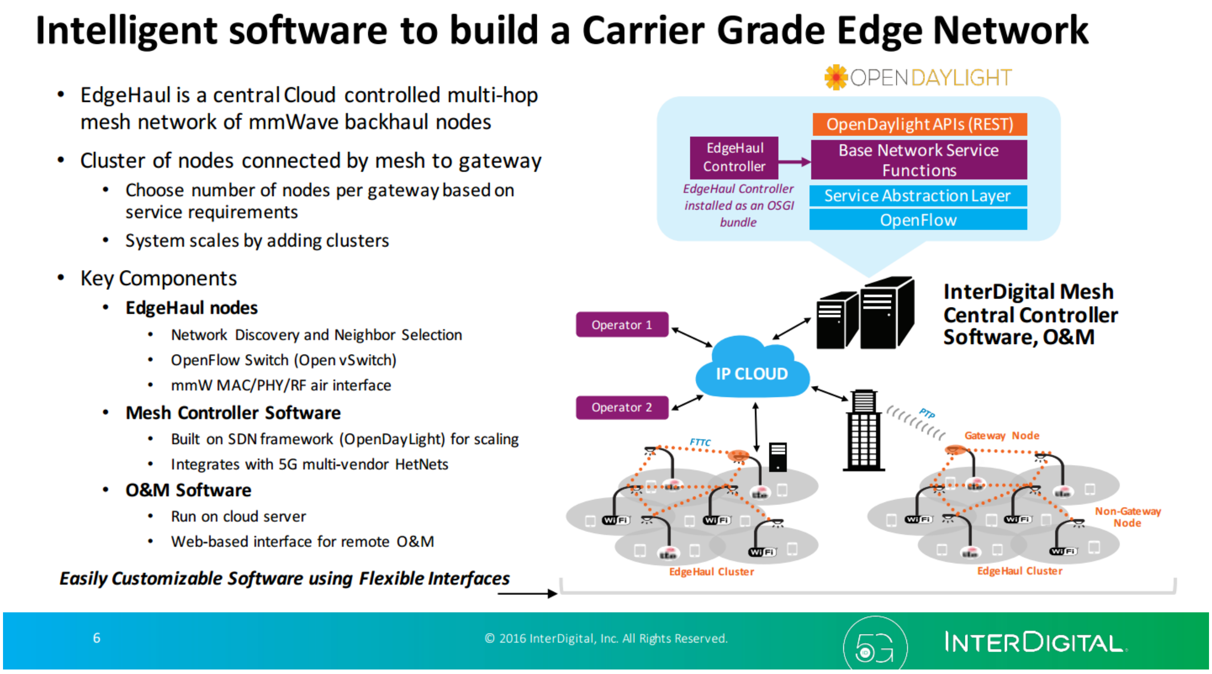

Figure 23: Edgehaul network design (taken from [120]).

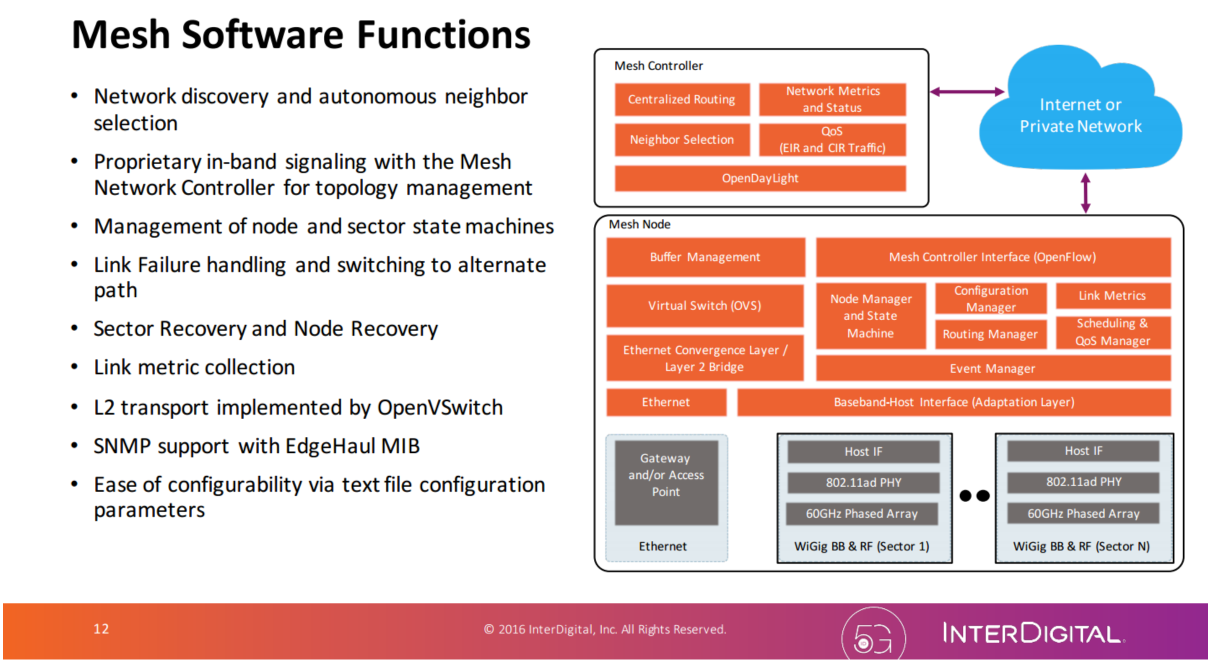

Figure 24: Edgehaul mesh software functions (taken from [120]).

Another paper in [109] considers a hybrid control architecture for a wireless meshed backhaul infrastructure. Load balancing and routing tables may be determined by a central SDN controller, or in case of failed or delayed communication to the controller a distributed algorithm operates. Since SDN control signalling, for example, OpenFlow messaging, is carried in-band over the wireless links of the mesh, the risk of message loss or delay is considerable. The paper demonstrates performance advantages for the hybrid architecture.

5.1.3.1 Management, and OAM operations to be supported by a mmWave SDN southbound protocol

In this section we discuss the API between the MAC entity of a mmWave mesh node and a SDN controller. It is assumed that the mesh is made from 802.11ad [110] radios connected by switches/routers, as in [119] and as described in Deliverable D2.2 [6]. This API could be the southbound interface or it could be an internal interface between an SDN agent running in the mesh node and the MAC. Control of the routing of data through the mesh is similar to the SDN control of the data plane in a conventional routed or switched packet network, and will not be discussed further here. Failover recovery is envisaged to occur above the MAC, either in the network element of the mesh node alone, or in combination with the SDN Controller.

Besides the mesh routing/forwarding, another key aspect is the execution of the desired mesh topology. Whilst the mesh node locations are known, the grouping of 802.11ad STAs into different personal BSSs (PBSS, Personal Basic Service Set) is required to form an 802.11ad mesh. Within a PBSS, one STA is identified as the PBSS Control Point (PCP) and is responsible for managing the scheduling of transmissions between the STAs of the PBSS. All STA within a PBSS operate on the same wireless carrier frequency. The topology may be centrally calculated, and then the assignment of STAs to PBSS (and any frequency allocation there) can be pushed down from the SDN controller(s) and agents to the MACs. We describe now through an example how the selection of the node acting as PCP in an 802.11ad based wireless mesh network can affect the topology of the network. Figure 25 provides an example physical topology, assuming that each 802.11ad node contains two back to back antennas, and highlights a set of potential PBSSs that can be created.

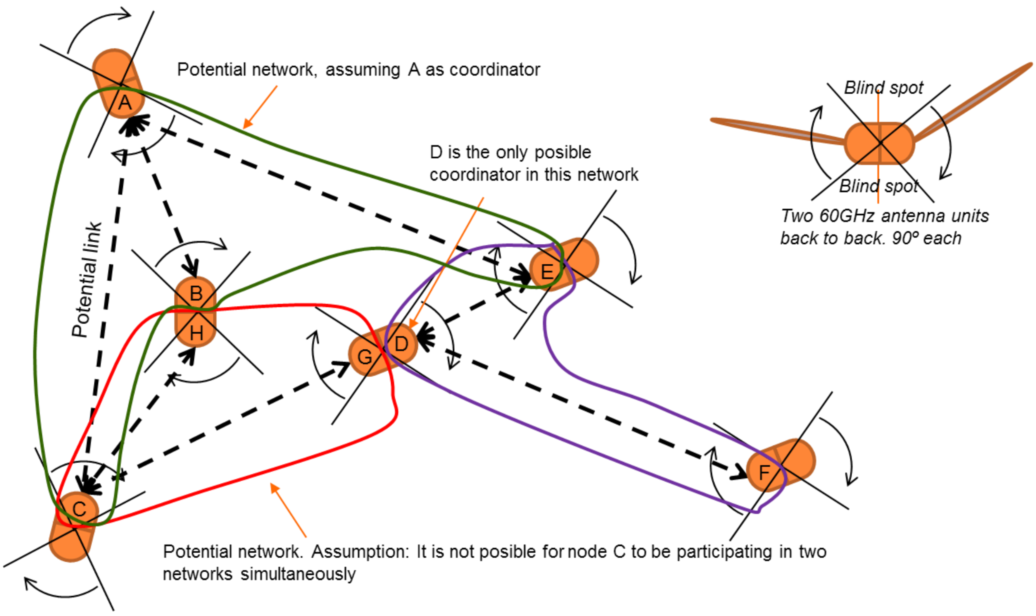

Figure 25: Example 802.11ad topology and potential PBSS configuration.

Note that in Figure 25 node C can either be part of the green PBSS or of the red PBSS. Likewise, node E can be part of either the green PBSS or the purple PBSS. Figure 26 depicts the resulting physical topologies depending on the PBSS where nodes C and E participate.

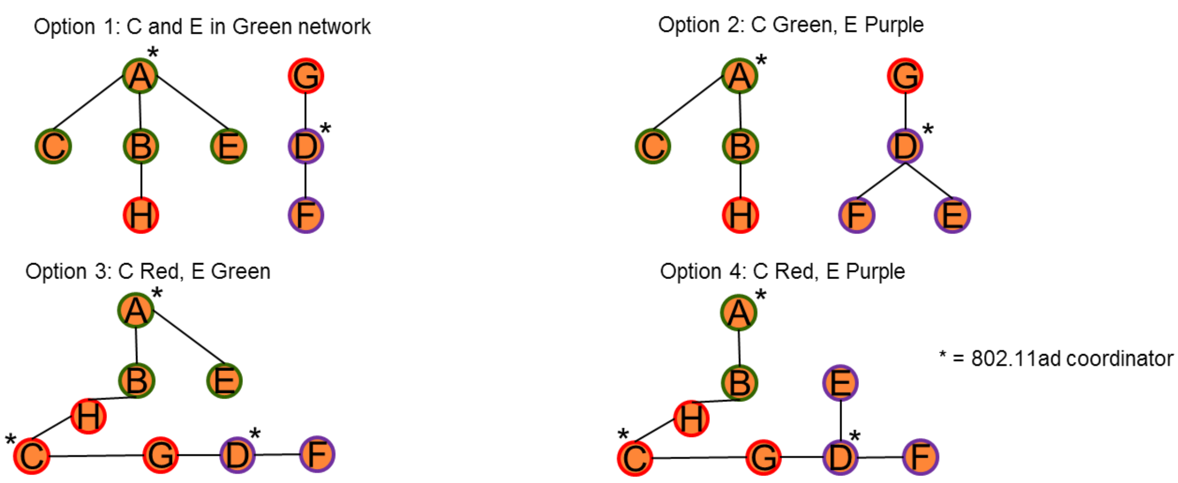

Figure 26: Physical topologies resulting from different PBSS configurations.

There may be the need to redesign one or more PBSS during the live operation of the mesh, although not all use-cases require it. If a link fails the failover algorithm may exploit an alternative path, without needing to change any PBSS configuration (although in one PBSS a STA to STA link will no longer exist if it has failed). If a mesh node is added then in some circumstances the new STAs can join existing PBSSs without external control input. However, if for energy reasons the SDN controller decides to shut down some transport nodes the impact may be big enough to justify a PBSS reconfiguration. 5G-XHaul will study how to design south-bound interfaces that allow to remotely configure an 802.11ad PBSS.

Another type of management actions to be supported by a mmWave 802.11ad device is for example an switch off command, which could be useful for failover testing (STA fails rather than a particular link) or energy efficiency algorithms. 5G-XHaul will study what south-bound interfaces are best suited to carry this type of management actions.

Scheduling operation

Considering the scheduling operation within the PCP of a PBSS, a conventional approach is for each MAC to expose, to the switch, a number of QoS queues identified by different Traffic Identifiers (TID) for each destination MAC address in the PBSS. The PCP then shares the air time between these transmit queues according to a hard-coded queue discipline that respects the QoS parameters associated with each TID.

Protection for SDN control signalling is needed because it is sent in-band over the wireless mesh. The use of a unique TID for the signalling permits the assignment of a higher priority for scheduling, more robust channel coding, higher retransmission limits and dedicated receiver reordering queues.

In principle, the SDN controller can influence the scheduler by either controlling the mapping of packets to TIDs, or by modifying the associated QoS parameters. For example, if a flow is mapped to a traffic stream (TS) then the values of the traffic specification (TSPEC) such as Mean Data Rate and Minimum Data Rate can be set. More direct control could also be envisaged, for example, specification of the air time allocations (on average) for each TID of each link. Knowing the MCS per link, which should be quasi-static for line-of-sight links the mean throughput as a result of the allocations would be known. An interface to allow an SDN controller interact with a PCP scheduler will be studied in 5G-XHaul.

Monitoring operation

A rich set of mmWave specific parameters can be read from the MACs and used by the SDN controller, for example:7

- MSDU/MPDU/bytes sent/rx per link.
- MCS per link per direction.
- Failed transmissions (MPDU errors) per link.
- Radio measurements (e.g. SNR, RSSI, per link).

- Average pkts queued (per queue, so per link per TID).
- Channel number (implies frequency).

Using these measurements, the SDN controller would be able to compute the effective bandwidth available to each (TX,RX) pair and to set TSPEC values appropriately (see above). 5G-XHaul will design southbound protocols that allow an 802.11ad MAC to convey the previous parameters to an SDN controller.

5.2 Southbound Interfaces for Optical Technologies

Optical transport networks are an integral part of the overall 5G network architecture and hence bringing them under the same management/control framework of SDN is essential for the end-to-end service provisioning and optimal resource allocation. In this context, there has been some work already done for the long-haul optical transport networks in [121][122] for elastic optical networks with fixed-grid / flexible-grid programmable optical components. The general approach in [121] has been to extend the OpenFlow protocol specification to include features and capabilities of the optical devices and then expose these through an OpenFlow Agent that speaks a technology specific southbound protocol (e.g. SNMP, TL1 etc.) and Extended Openflow protocol as northbound. This approach enables the common OpenFlow abstraction across the packet as well as optical transport network and facilitates using existing open-source SDN controller software to manage a converged network. The southbound protocol from the controller point of view in this case is the extended OpenFlow protocol. There are some other approaches for enabling software-defined control of the optical devices through a hybrid OpenFlow+GMPLS approach using appropriate protocol for certain features of the devices or properties that are important across the network. Recently, with OpenFlow specification 1.4, there is some support to specify properties of optical devices and as vendors start implementing these specifications, the OpenFlow support will be natively available for optical transport equipment as well. In case of 5G-XHaul, we are proposing to use the optical solutions at the edge of the network to connect RAN side equipment to the core network and hence we focus particularly on Time-Shared Optical Network (TSON) technology and WDM-PON technology. In following sub-sections, we describe the state-of-the-art SDN southbound solutions for these technologies.

5.2.1 Southbound interface in Time Shared Optical Network (TSON)

State of the Art

TSON is a novel TDM over WDM technology with flexible timeslots and frame lengths to support different sub-lambda service granularities and guarantees. Since it was proposed by the HPN group at University of Bristol, state of the art work on the TSON control plane is what is described in the following sections.

Introduction

The basic functions for the operation of TSON domains are implemented in internal modules, within the Software Defined Networking (SDN) controller, that cooperate together for the on-demand provisioning of connectivity between TSON nodes, usually TSON edge nodes. These modules are implemented in OpenDaylight bundles that provide a set of services, which act as consumers of the TSON driver, making use of the methods exposed by its interface to configure and collect information about the TSON domain. Following the common OpenDaylight approach, the interface of the services implementing the TSON basic functions is described through the YANG language. It is exposed as an internal interface within the SDN controller, but it may also be exported as an external REST interface that makes use of the HTTP protocol.

TSON Connection Service

The Connection Service provides the functionalities to create, manipulate and destroy intra-domain TSON connectivity and is one of the services that expose a REST interface towards external entities. The main remote procedure calls included in its YANG model are related to the creation, modification and deletion of network paths, with the specification of attributes like end-points, tributary traffic, bandwidth, timing constraints in case of advance reservations, and resilience options. Notifications are used to notify events related to the established network connections, for example a failure or a modification triggered by internal procedures (e.g. for the automated recovery).

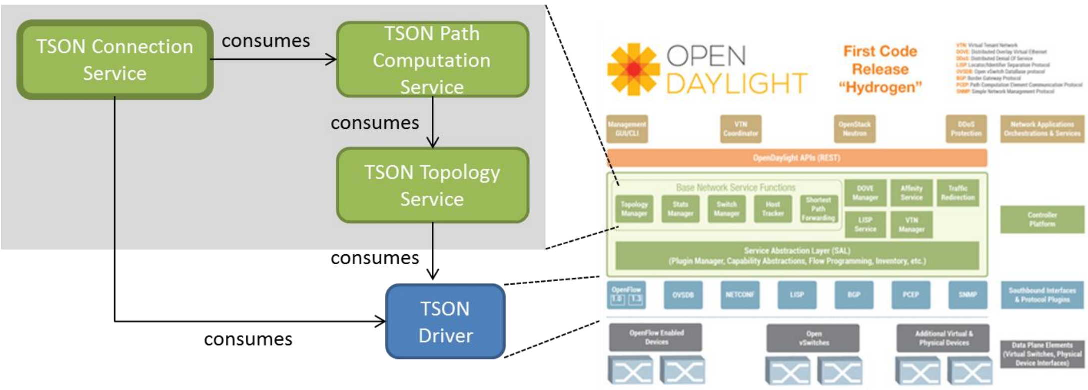

Figure 27: TSON Connection Service in CONTENT SDN controller.

Internally, the Connection Service is implemented as a consumer of other OpenDaylight services, as shown in Figure 27. In particular, it relies on the TSON path computation service to retrieve the list of the TSON nodes that must be configured to support a given traffic flow, together with the details of the ports and the resources (wavelengths and time-slots for TSON ports) to be cross-connected. Moreover, it interacts with the TSON driver to request the actual configuration to be performed on the virtual nodes at the data plane.

The Connection Service is a stateful service and it is responsible for the whole lifecycle of each TSON intra-domain connection, from creation to tear-down. For example, during the setup phase it is responsible to implement any rollback function in case of cross-connection failures in one of the nodes in the path. Moreover, when a path is established the Connection Service is responsible to organize the recovery of the service in case of data plane failures notified from the TSON driver. The specific protection or restoration mechanisms to be applied depend on the parameters requested for each connection.

TSON Path Computation Service

The TSON Path Computation Service is responsible to elaborate the paths associated to a given TSON connection, taking into account the characteristics of the connection (e.g. bandwidth, metrics or objective functions to be applied, nodes to be excluded), the current availability of the TSON resources in the virtual infrastructure and the specific constraints of the technology (i.e. the continuity of wavelengths and time-slots along the path).

The TSON Path Computation Service is a consumer of the TSON Topology Service, an extended version of the basic topology service available on the SDN controller that includes the details of the TSON resources. The interaction between the TSON Path Computation Service and the Topology Service is used for the initial synchronization of the topology available on the data plane, as well as for asynchronous notifications related to resource failures occurred at runtime. These procedures allow the Path Computation Service to maintain a continuously updated view of the topology status.

On the other hand, the Path Computation Service acts as the only responsible for path elaboration. It is invoked by the Connectivity Service (which is a consumer of the Path Computation Service) whenever a new path needs to be created, in order to obtain the list of nodes and resources to be configured, and it receives a confirmation when the path is correctly established. Moreover, the Path Computation Service is also notified by the Connectivity Service whenever an existing path is modified or removed on the Virtual Infrastructure. Consequently the Path Computation Service is able to maintain a synchronized view of the availability of the data plane resources and the details of the established flows, and it is consequently able to operate in a stateful manner.

It should be noted that while the Path Computation Service is implemented as an internal service of the SDN controller that maintains the control on the lifecycle of the computed path, the actual computation can be delegated to external applications implementing algorithms specialized for the given technology. In this case, it must be guaranteed that these algorithms are always running on an updated version of the topology view. The algorithms can be implemented through an external module, called SLAE (Sub-wavelength Lambda Allocation Engine), which returns the nodes and the ports to be cross-connected to create the requested path, together with the wavelength(s) and the time-slot(s) to be used, in compliance with the wavelength and time-slot continuity. The algorithms implemented in the SLAE have the objective to maximize the utilization and the efficiency of the wavelengths and time-slots available in the data plane.

These capabilities of TSON nodes are categorised into two main branches of (1) configuration, and (2) operation, which are described in XML format and are accessed by REST services.

In the configuration mode the TSON TDM features of the system can be determined, where parameters such as time slice length, overhead and frame duration are exposed. In the operation mode on the other hand the TSON light paths are established and updated where time slices are allocated to different flows at the ingress node, which are transparently switched in the TSON core.

TSON XML data model

The TSON XML model with sample information is printed in the following section. The XML message is sent from the controller to the TSON agent through REST interfaces. It carries both the configuration and the operation information required at the TSON node.


```
<Request>
<Request_type>Commit_All</Request_type>
<Request_id>10</Request_id>
<Request_Mode>Burst</Request_Mode>
<Request_source>0</Request_source>
<Request_destination>1</Request_destination>
<Request_Kchar>7</Request_Kchar>
<Request_Burstsize>8</Request_Burstsize>
<Request_Burstnumber>9</Request_Burstnumber>

<Request_TimeSliceCNTRL0>10 129 12 13 14</Request_TimeSliceCNTRL0>
<Request_TimeSliceCNTRL1>10 11 12 13 14</Request_TimeSliceCNTRL1>

<Request_Lambda0Buffer0ts>11111</Request_Lambda0Buffer0ts>
<Request_Lambda0Buffer1ts>11111</Request_Lambda0Buffer1ts>
<Request_Lambda0Buffer2ts>01101</Request_Lambda0Buffer2ts>
<Request_Lambda0Buffer3ts>01101</Request_Lambda0Buffer3ts>
<Request_Lambda1Buffer0ts>01101</Request_Lambda1Buffer0ts>
<Request_Lambda1Buffer1ts>01101</Request_Lambda1Buffer1ts>
<Request_Lambda1Buffer2ts>01101</Request_Lambda1Buffer2ts>
<Request_Lambda1Buffer3ts>01101</Request_Lambda1Buffer3ts>
<Request_Header0>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</Request_Header0>
<Request_Header1>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</Request_Header1>
```

```
<Request_Header2>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</Request_Header2>
<Request_Header3>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</Request_Header3>

<Request_Header0_mask>0001111011111101</Request_Header0_mask>
<Request_Header1_mask>0001111011111101</Request_Header1_mask>
<Request_Header2_mask>0001111011111101</Request_Header2_mask>
<Request_Header3_mask>0001111011111101</Request_Header3_mask>
</Request>
```

The TSON node as the physical FPGA node agent, use the REST API as the vertical interface to communicate with other platforms. The REST API exposes a number of configuration and operational features, which are defined using XML of TSON node as programmable features to the external applications. The requests made through the controller are identified by the function requests (Request_type), the sequential number of command (Request_id) for logging purposes, and the mode of operation.

```
<Request_type>Commit_All</Request_type>
<Request_id>10</Request_id>
<Request_Mode>Burst</Request_Mode>
```

The TDM features of the TSON technology such as time slice size, overhead size, and frame size are directly controllable. The TSON nodes, which are located in the same network, need to have the same configurations as the TSON operates in a synchronised manner and thus having identical TDM attributes is essential. The controller will adopt some features which guarantee that the virtual nodes in the same domain have common TDM features. The lines below demonstrate the XML fields which are used for TDM configuration on the nodes. (Request_Kchar) refers to the overhead to be considered for each time slice, (Request_Burstsize) relates to the duration of time slices bursts which comprised overhead and data, and (Request_Burstnumber) which describes the frame size, as each frame is based on the repetition of a number of time slices.

```
<Request_Kchar>7</Request_Kchar>
<Request_Burstsize>8</Request_Burstsize>
<Request_Burstnumber>9</Request_Burstnumber>
```

Apart from the TDM attributes, operational information such as time slice allocation, header matching, switching information is sent to the nodes to provision light paths for different traffic flows. The configuration information as demonstrated below comprise of the source and destination of the light paths (Request_source), (Request_destination), the TX/RX information which are identified by (Request_Lambda1Buffer3ts), the switching information where the signals out of TSON are directed optically (Request_TimeSliceCNTRL0), and the Ethernet header (Request_Header0) and the masking bit maps (Request_Header0_mask) for identifying different flows at the ingress.

```
<Request_source>0</Request_source>
<Request_destination>1</Request_destination>
<Request_TimeSliceCNTRL0>10 129 12 13 14</Request_TimeSliceCNTRL0>
<Request_Lambda1Buffer3ts>01101</Request_Lambda1Buffer3ts>
```

```
<Request_Header0>0 1 2 3 4 5 6 7 8 9 10</Request_Header0>
<Request_Header0_mask>0001111011111101</Request_Header0_mask
```

TSON basic REST API

After deploying the service, a URI is used to communicate with the TSON interface. The URI for testing purposes is set to (http://localhost:8080/AbstractionInterface/Agent0webresources), where the local host is used for calling services locally on the test machine. The path to call the FPGA (either physical or virtual) is ("Call_FPGA") which is added to the URI.

After calling the service for testing using the http monitor the following information are received which demonstrate the correct HTTP communication with code 200 after committing the new information including both configuration and operation data.

```
Request: POST http://localhost:8080/AbstractionInterface/Agent0webresources/Call_FPGA?
timestamp=1414502761877
Status: 200 (OK)
Time-Stamp: Tue, 28 Oct 2014 13:26:02 GMT

Sent:
Commit_All 10 Burst 0 1 7 8 9 10 129 12 13 14 10 11 12 13 14 11111 11111 01101 01101
01101 01101 01101 01101 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0001111011111101 0001111011111101 0001111011111101 0001111011111101

Received:
Commit_All 10 Burst 0 1 7 8 9 10 129 12 13 14 10 11 12 13 14 11111 11111 01101 01101
01101 01101 01101 01101 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0001111011111101 0001111011111101 0001111011111101 0001111011111101
```

5.2.2 Southbound interface in WDM-PON

The WDM-PON based fronthaul not only needs to respect the strict parameters imposed by specifications such as CPRI or OBSAI, but also to provide flexibility in order to support load-balancing and switching capabilities [123]. For this purpose, WDM-PON provides configurable, transparent, low-latency point-to-point connections between ports at the central OLT and individual distributed ONUs. Depending on the connection requirements, power consumption can be reduced for unused WDM-PON ONU terminals. In order to take advantage of these features and to optimize the data path provisioning on the network, similar to other technologies, communication to a centralized management and control plane will be required. Internal management of the WDM-PON domain is provided via an auxiliary management and control channel (AMCC) enabling communication between the ONUs and the OLT. The purpose of this AMCC is the physical layer control and monitoring of the ONUs. For instance, the wavelength setting and control is performed via this AMCC. The controller in the OLT can also switch on or off a particular ONU when it is not in use, in order to save power consumption.

The centralized management and control plane controls the WDM-PON domain as a whole. This also includes decoupling the control plane from the data plane and centralizing it into an SDN based intelligent network controller, having a complete view of the whole network. The control plane can monitor and change the forwarding behaviour of the WDM-PON data plane in real time through vendor-independent (and standardized) software interfaces.

In general, the electrical or optical cross-connect and optical active components at the OLT can be deployed with the intent for SDN integration, enabling different control and monitoring functions for flexible management. Particularly, to achieve the dynamic allocation of traffic load, the central SDN controller can reconfigure the data routing path in the electrical/optical switch. Moreover, the status of activated ONUs can be read from the OLT, including the channel number, wavelength deviation, received power level, etc.

The following items are available at the SDN interface of the WDM-PON domain, located at the OLT:

Table 4: 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

5.3 Southbound Interfaces for Ethernet Technology

Ethernet switching or packet processing based on Ethernet headers has been part of the OpenFlow specification from version 1.0 [124]. There are multiple improvements/enhancements to the OpenFlow specification over the years to enable Ethernet packet parsing/processing based on various parts of the header [125]. There is some work undertaken by operators to use SDN for controlling the MEF based Carrier Ethernet equipment as Carrier Ethernet is becoming a preferable choice of packet transport in the telecommunications industry. Even with standards compliant Ethernet switching equipment, there are certain features of the device or a particular vendor's way of implementing the standards. In this case, we require extensions to the OpenFlow southbound protocol to expose these features and be able to control them to take advantage of the proprietary implementation for performance or cost reasons. In 5G-XHaul we will assume that SDN enabled Ethernet switches are at least compliant to OpenFlow 1.3 [125].

5.4 Requirements and Initial Design for the SDN southbound for 5G-XHaul technologies

In this subsection, we describe the requirements of the southbound protocol implementation for the SDN control of 5G-XHaul technologies. These requirements are captured in the following table followed by the discussion of how southbound protocol capabilities of each technology currently satisfy these requirements. This will be a very initial discussion and based on these we will propose an initial design for developing a common framework for southbound protocol implementation for all 5G-XHaul candidate technologies.

5.4.1 5G-XHaul southbound interface requirements

Table 5: 5G-XHaul southbound interface requirements.

Requirement	Summary	Description
R0	Capabilities Information	The southbound protocol implementation should be able to retrieve device capabilities in terms of amount of resources, switching/forwarding and monitoring the state.
R1	Network State Monitoring	The southbound protocol implementation for each technology should support collecting device state information at the periodicity required by each technology.
R2	Topology Discovery	The southbound protocol implementation should be able to initiate topology discovery, read topology information (or peer-node information) from the devices.
R3	Forwarding/Switching Configuration	The southbound protocol implementation should be able to configure the switching configuration as well as data processing capabilities of the device.
R4	Resource Allocation	The southbound protocol implementation should be able to support allocation of required resources to various data-processing units (or flows/connections) at certain granularities.
R5	Managements Actions	The selected southbound protocol should allow the control to trigger certain management actions on the network elements, like for example switching it down for energy saving purposes.

5.4.2 Initial Design for the 5G-XHaul southbound protocol implementation

In this sub-section, we propose an initial design of the SDN southbound interface for various candidate transport technologies used in 5G-XHaul. There is already some work done on the southbound protocols for these technologies and initially we will use the existing southbound interfaces to control the network elements. Even the existing open-source SDN controllers support multitude of southbound protocols including SNMP, TL1, CAPWAP etc. to support a plethora of networking devices. The important part of bringing devices under the control of SDN controller is to be able to create right abstractions for the devices' data model and expose their properties and status through the controller for network applications to use. The OpenFlow protocol as southbound is an option and could be explored as an initial design approach. The approach towards controlling devices using SDN is shown in the Figure 28 below.

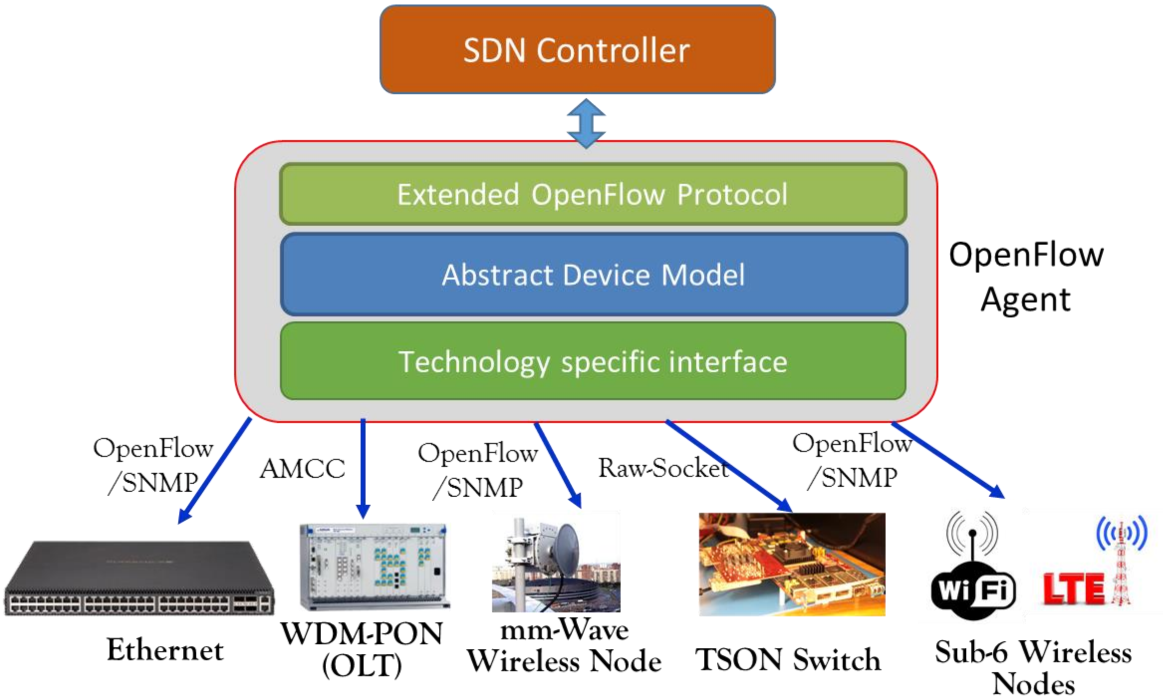

Figure 28: OpenFlow SDN southbound approach for 5G-XHaul technologies.

In the proposed approach, we need to build OpenFlow agents for each candidate technology and device as there are device specific control interfaces for different vendors and technologies. The openflow specification natively supports packet processing capabilities of devices and recently support some optical device capabilities as well. However, there is no support for representing and controlling wireless capabilities of the devices in OpenFlow yet. In this context, we need to develop OpenFlow agents for wireless technologies. Based on the initial assessment of the capabilities of southbound protocol support, most of the devices (Ethernet, mm-Wave and Sub-6) support some sort of Openflow interface or SNMP interface. However, for WDM PON and TSON, we need to develop OpenFlow agents to support the proposed approach.

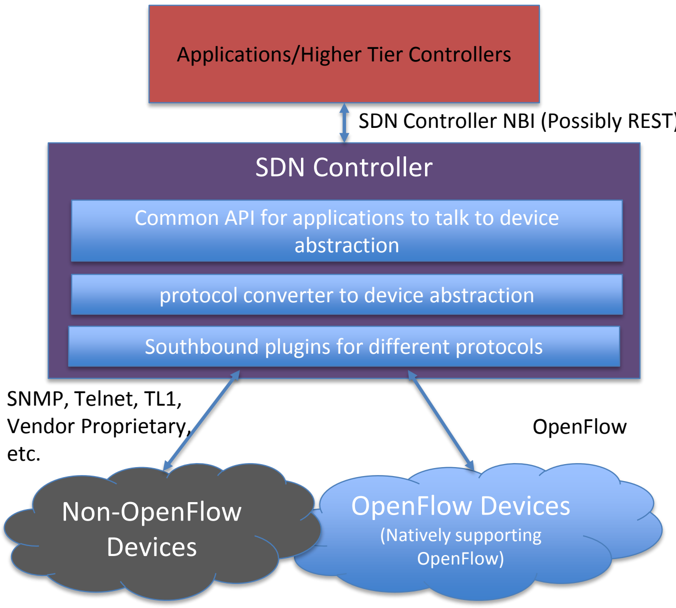

Figure 29: Protocol specific SDN southbound approach for 5G-XHaul technologies.

The other approach that we can take in 5G-XHaul is (see Figure 29) to implement southbound protocol plugins for various technologies depending on the suitability of protocols to expose and control various technology specific parameters. The existing SDN controllers support a framework for southbound drivers for specific technologies and devices and then converting into network state abstractions maintained by the controller. The applications or higher tier controllers can then use the API exposed by the SDN controller.

Next, we introduce an initial southbound interface design for each of the technologies considered in 5G-XHaul.

5.4.2.1 Sub-6 wireless transport technologies

The transport wireless Sub-6 technologies considered in 5G-XHaul are based on IEEE 802.11 radios. The interested reader is referred to Deliverable D2.2 [6] for a detailed description of the considered technologies. In this section we describe the initial design for the southbound interfaces that will be developed in 5G-XHaul to control the Sub-6 wireless nodes.
Two types of south-bound protocols will be considered: i) an OpenFlow based south-bound plugin with wireless extensions, which will be used to control forwarding within the Sub-6 wireless mesh, and ii) a REST/NETCONF based southbound plugin to manage the state of the IEEE 802.11 radios.

The southbound based on OpenFlow extensions will support requirements R1 (Network state monitoring), based on extensions to OpenFlow port statistics, R2 (Topology Discovery), based on LLDP, and R3 (Forwarding/Switching), based on abstracting the point to multi-point nature of the wireless medium with multiple point-to-point links. The management protocol based on REST/NETCONF will support requirements R4 (resource allocation), by allowing for example to configure the wireless channel used for each radio or the time slots allocated to each link, and R5 (management actions), by for example allowing to shut down a device or instantiate new virtual interfaces.

5.4.2.2 mm-Wave Technology

The transport wireless mmWave technologies considered in 5G-XHaul are based on IEEE 802.11ad radios. The southbound interface will be similar to that used with Sub-6 radios, so that elements of the design can be re-used. R0 (Capabilities) and R1 (Network state monitoring) will also be realised using OpenFlow extensions. With respect to R2 (Topology Discovery), the main functionality is the ability to push down a PBSS configuration for an 802.11ad STA, namely the PBSS identity, channel assignment and whether the STA is the PCP. If this is not used the STA will determine the PBSS configuration autonomously but this may be sub-optimal. The control protocol for R2 and R3 (Forwarding/Switching) is still to be decided. The management protocol for R4 (resource-allocation) and R5 (management actions) is currently open.

5.4.2.3 Time Shared Optical Networks

The TSON technology currently supports the approach of a protocol specific southbound interface. The implementation of southbound protocol is through a TSON agent which provides REST API to the SDN controller. The controller can use this REST API to send configuration data in the XML format as described in section 5.2.1 of this document. Therefore from the SDN controller point of view, there is only one southbound interface based on REST APIs that is currently sufficient for all requirements mentioned in the requirements table. The XML data model sent over REST will be extended as part of the development of southbound interfaces to support different monitoring data and certain management actions as the TSON solution is further developed with its data-plane capabilities.

5.4.2.4 WDM-PON

The WDM-PON technology considered in 5G-XHaul currently supports the control of ONUs through the OLT using the AMCC protocol (both upstream and downstream). The initial design to implement an SDN interface for the WDM-PON system will involve the development of a NETCONF agent running on an embedded Linux over the soft-processor within the FPGA board of the OLT. We will define the YANG models to represent the data model of the WDM-PON OLT node and define RPCs to set and retrieve data through the controller. All the requirements will be fulfilled using the NETCONF protocol where the SDN controller will implement the NETCONF client and the OLT will implement the NETCONF agent which will further translate NETCONF RPC calls to AMCC receiver/transmitter APIs through the local controller on the OLT.

5.4.2.5 Ethernet Transport

Ethernet transport technology currently uses OpenFlow protocol for all southbound communications between the SDN controller and the switch. As the initial design, we will use the OpenFlow southbound interface for all requirements. The management actions (R5) could be supported over OVSDB or using the CLI of the switch depending on the support available for specific device.

5.4.3 Requirements mapping to initial design summary table

Table 6: Mapping of requirements to initial design

	R0 (Capabilities)	R1 (Monitoring)	R2 (Topology)	R3 (Forwarding / Switching)	R4 (Resource allocation)	R5 (Management Actions)
Sub-6 (LTE, WiFI)	OpenFlow Extensions	OpenFlow Extensions	LLDP	Protocol specific approach	NETCONF / RESTCONF	Device and driver Specific commands
mmWave	OpenFlow Extensions	OpenFlow Extensions	Autonomous / TBD	TBD	TBD	TBD
TSON	Protocol specific approach (REST API)	Protocol specific approach (REST API)	Protocol specific approach (REST API)	Protocol specific approach (REST API)	Protocol specific approach (REST API)	Protocol specific approach (REST API)
WDM PON	NETCONF	NETCONF	NETCONF	Does not apply	NETCONF	NETCONF
Ethernet	OpenFlow	OpenFlow	LLDP packets analysed by controller	OpenFlow	OpenFlow	OVSDB, CLI

6 SCALABLE CONTROL PLANE DESIGN

The advantages of logically centralizing the control intelligence of the network in a controller, as proposed by SDN, are well-known. However, it also presents some challenges when applied to large scale, highly dynamic networks, such as a 5G-XHaul transport network across different domains (optical/wireless) and areas. The burden of determining, installing and reconfiguring data paths due to network status changes (e.g. topology, and link capability) and varying service requirements (e.g. due to instantiation of new slice from a tenant, UE mobility, etc.) may exceed the computation capabilities of a central controller. Meanwhile, the response time of the network changes greatly affects user experience. Therefore, the placement of the controller needs to be considered to ensure that the latency of the control channel can meet the latency budget in different use cases. In summary, the performance limitations of the current SDN architecture can be classified in three major areas:

- Scalability. The performance of the SDN controller is closely related to the number of Openflow datapaths to be managed in parallel;
- Latency. The response time to changes in the network topology depends on the transport latency between the controller and SDN switches and the amount of simultaneous events to be processed;
- Resiliency. Since the controller is the brain of the SDN network, resilience of the controller itself and the control channels from the switches to the controller greatly affects the correct operation of the complete SDN network.

In the literature, there are many proposals to overcome the limitations related to the three areas mentioned above. Some works propose a distributed/hierarchical control architecture. Agent-based solutions can be seen as another type of distributed controller, where a local agent is installed in each SDN switch and takes over some of the tasks from the applications on top of the central controller. For instance, [135] delegates some of the offloading functionalities to local agents in order to reduce the redundant signaling and the processing overhead at the controller. The agent based approaches require some additional application layer processing capabilities at the SDN switch, which might not always be feasible. A parallel solution to the distributed controller architecture itself is the flow rule system. For example, hierarchical flow rule cache, and dynamic and adaptive flow rules.

This chapter is organized as follows. In section 6.1 we review the state of the art on SDN controller design and placement strategies. In section 6.2 we introduce previous works dealing with the design of SDN rule caching systems, to enhance the scalability of SDN control planes. In section 6.3, we survey techniques that have been proposed to improve the reliability of in-band SDN control channels, which is a very relevant problem when applying SDN to transport networks. In section 6.4, we survey existent approaches to design the northbound interface of an SDN controller. Finally, in section 6.5 we derive requirements to be fulfilled by the SDN controller in 5G-XHaul, and present an initial design that will be further elaborated in subsequent WP3 deliverables.

6.1 State of the Art on SDN Controller Design and Placement Strategies

SDN is considered as a key enabler to realize re-configurability in future networks, both the Internet as well as the 5G network. Any reasonably sized network cannot operate on a single controller, since the controller becomes a single point of failure for the network. Any network that requires scalability and redundancy will need distributed controllers. In the literature, there are many works for distributed controller designs [126][127][128][129][134]. In general, SDN controllers can be classified into two categories: *replica* and *hierarchical* controllers. In a replica SDN controller design (illustrated in Figure 30), multiple controllers can be viewed as a logically centralized controller, however, the controllers are physically distributed across multiple locations. In this case, all controllers maintain a global view of the network to manage the same global functionalities for the underlying devices. Having a global view at each controller, the network can be controlled in an efficient way to achieve a better performance. On the other hand, such a design requires a full synchronization of states among controllers and thus, inter-controller interaction must be taken into account in controller placement strategies.

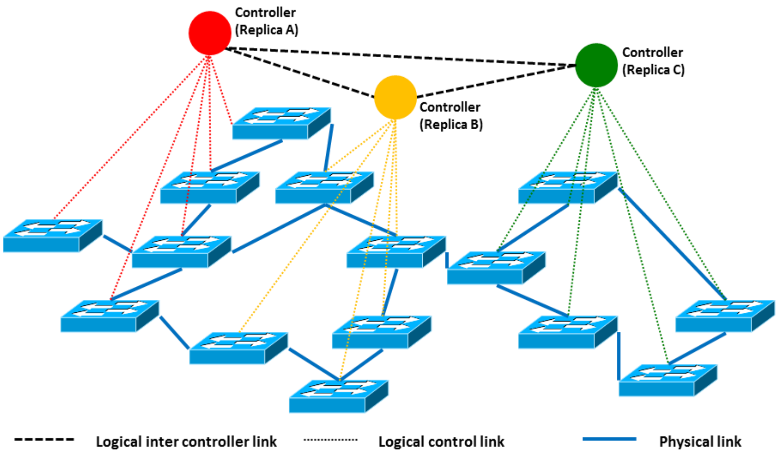

Figure 30: Controller Designs - Replica Controllers.

HyperFlow [127] is the first distributed control plane for OpenFlow and it is an example of a replica controller. HyperFlow is implemented as an application for NOX controllers [161], where NOX is again the first framework for OpenFlow controllers. The HyperFlow application is in charge of synchronizing NOX controllers' network-wide views, redirecting OpenFlow commands targeted to a non-directly-controlled switch to its respective controller, and redirecting replies from switches to the request-originator controllers. HyperFlow keeps the state of the controllers in a distributed data store and simultaneously enables local caching in the controllers. In general, when there are multiple controllers, they are able to make decisions for many flows by just consulting their local cache, or requiring state retrieval from remote controllers and resulting in extra delay in the control plane service time. HyperFlow avoids this extra delay by proactively pushing all necessary information to all controllers, enabling them to locally serve all flows. More specifically, all the controllers have a consistent network-wide view and run as if they were controlling the whole network. They all run the exact same controller software and set of applications. Each switch is connected to the best controller in its proximity. Upon controller failure, affected switches are reconfigured to connect to an active nearby controller. Each controller directly manages the switches connected to it and indirectly programs or queries the rest (through communication with other controllers and propagation of selected locally generated controller events). To propagate controller events to others, HyperFlow uses a publish/subscribe messaging system. It implements this system by using WheelFS [164], which is a distributed file system designed to offer flexible wide-area storage for distributed applications.

Onix [128] is another distributed system which runs on a cluster of one or more physical servers, each of which may run multiple Onix instances. As the control platform, Onix is responsible for giving the control logic programmatic access to the network (both reading and writing network state). In order to scale to very large networks (millions of ports) and to provide the requisite resilience for production deployments, an Onix instance is also responsible for disseminating network state to other instances within the cluster. OptimalFlow [171] is another SDN controller that monitors a single SDN domain, and redesigns the network according to the solutions delivered by an integer linear programming (ILP) optimization problem. The developed ILP problem encapsulates a shortest path routing objective. OptimalFlow exposes two communication interfaces to enable a hierarchical control plane. Its northbound interface reduces a complete switch infrastructure to an emulated (software) SDN switch, which exposes the domain's edge ports to the upper tiers and can be monitored and controlled through the OpenFlow protocol. OptimalFlow's southbound interface connects to an OpenFlow controller, to monitor and control a network of emulated or real SDN switches.

DISCO (Distributed SDN COntrol plane) [165] is one more approach for multi-domain SDN networks. It is implemented on top of the Floodlight [162] OpenFlow controller and the AMQP [163] protocol, which provides routing, messaging with orientation and prioritized querying. It relies on a per domain organization, where each DISCO controller is in charge of an SDN domain, and provides a unique lightweight and highly manageable control channel used by agents that can be dynamically plugged into the different domain controllers. The agents share aggregated network-wide information between the domains and hence provide end-to-end network services. Contrary to other distributed SDN control planes, DISCO discriminates heterogeneous inter-domain links such as high-capacity MPLS tunnels and SATCOM interconnections with poor bandwidth and latency. More specifically, a DISCO controller is composed of two parts: an intra-domain and an inter-domain part. The intra-domain part gathers the main functionalities of an SDN controller and it is responsible for monitoring a domain and managing the flow prioritization. The inter-domain part is based on agents that provide the communication among the multiple controllers by building channels between neighbouring controllers and sharing information with the link state and the host presence. DISCO, as a Floodlight implementation, consists of several modules that are responsible for various functionalities, such as the information storage collected from all the controllers, the gathering of information related to the flow throughput, latency and packet loss rate, the management of the network SLAs (Service-Level Agreements) between the tenants and the service providers, etc.

HyperFlow, Onix and DISCO try to distribute the control plane, however, they maintain some centralized logic as all replica controllers do. Although they use a distributed file system or a distributed hash table or a pre-computation of all possible combinations respectively, these approaches impose a strong requirement: a consistent network-wide view in all the controllers. On the contrary, there are also hierarchical SDN controllers (their design is illustrated in Figure 31). Each hierarchical controller is responsible to manage a set of devices from the corresponding layer. Thua, each controller may have different level of control and functionalities to manage the underlying devices. The lower layer controllers send local views to the upper layer without maintaining the global view of the network. Each controller requires only aggregation of views between the layers and there is no state synchronisation among controllers.

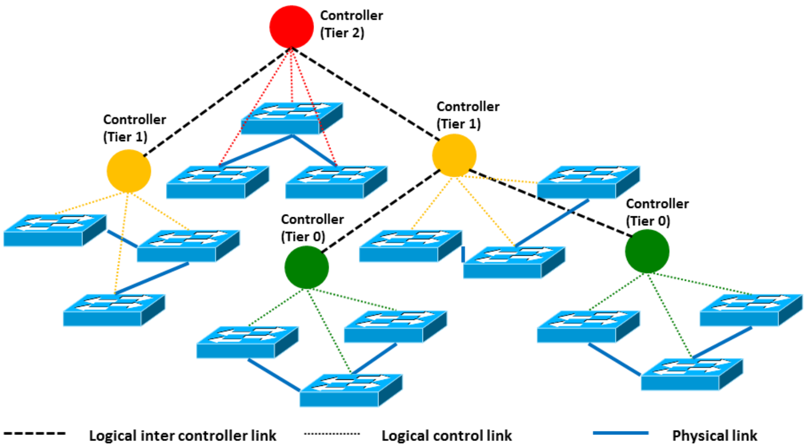

Figure 31: Controller Designs - Hierarchical Controllers.

Kandoo [167] proposes a hierarchical distribution of the controllers based on two layers of controllers: (i) the bottom layer, a group of controllers with no interconnection, and no knowledge of the network-wide state, and (ii) the top layer, a logically centralized controller that maintains the network-wide state. Each one of the bottom layer controllers is responsible for each of the individual SDN domains, while the top layer root controller leads all the bottom layer controller. The bottom layer controllers handle the frequent events of the local applications, while simultaneously they cooperate with the root controller for handling the rare events of the non-local applications. It is important to note that the data flow in Kandoo is not always bottom-up. A local application can explicitly request data from an application deployed on the root controller by emitting an event, and applications on the root controllers can send data by replying to that event. For instance, a topology service running on the root controller is able to send topology information to local applications by replying to events of a specific type. In Kandoo, the network state should be pushed proactively and then the pushed configuration should be adaptively refined afterwards. Further, Logical xBar introduces a recursive building block to construct abstracted hierarchical SDN networks [170].

In Orion [134], a hybrid hierarchical control plane is presented for large-scale networks. The proposed hybrid hierarchical control plane architecture is relative to the flat control plane architecture (such as HyperFlow and Onix) and the centralized hierarchical control plane architecture (such as Kandoo and Logical xBar). Orion can reduce the computational complexity growth of SDN control plane from superlinear to linear by constructing abstracted hierarchical network views. Thus, the computational complexity of the control plane is reduced. Meanwhile, an abstracted hierarchical routing method is utilized to reduce the path stretch problem based on the architecture of Orion. M2cloud [169] is another framework that employs two-level controllers with decoupled functions. It classifies the controllers to local controllers and the global controller.

The local controllers are connected to global controller in order to exchange local events and share network view, as Kandoo is doing also. Similarly, the Network Virtualization Platform (NVP) [36] offers a hierarchical and scalable control plane. It has been proposed for datacenters, adopting a two-layer hierarchy with multiple interacting controllers. The top layer consists of the so-called logical controllers that compute the universal flows, which are an intermediate representation similar to OpenFlow but abstract out all network edge specific details such as ingress, egress or tunnel port numbers. On the other hand, the bottom layer consists of the so-called physical controllers that translate the location-independent portions of universal flows with node and location-specific state. The resulting physical flows, which are now valid OpenFlow instructions, are then pushed down to the edge nodes, but more details regarding the rules format at each layer are given later.

ALICANTE [166] is a previous FP7 project that proposed a new Internet architecture. According to this architecture, the network is sliced to logically isolated planes based on light virtualization. The proposed architecture is conceptually similar to SDN, although not following full SDN specifications. The ALICANTE controlled network consists of data plane slices named Content Aware Networks (CANs), which are controlled by the CAN Managers. CAN Manager and the Intra-domain Network Manager play together the role of an SDN controller for a network domain. Actually, the whole network is a multi-domain logical network governed by several “SDN controllers” – which cooperate for resource management and routing. The degree of centralization is configurable in ALICANTE by defining the placement of CAN Managers and the sets of routers to be controlled.

6.1.1 State of the Art on SDN Controllers placement

One of the many issues that need to be resolved to achieve an SDN enabled 5G network is the controller placement. In particular, someone e.g. the administrator needs to solve the problem: given a network topology comprising switches (e.g., SDN enabled BSs) and compute/storage resources, where should the controllers (e.g. wireless cluster controllers) run? If there is a fixed number of controllers to be placed, what is the best way to place them?

In recent years there has been promising work on the controller placement problem [130][131][132][133]. The initial work of SDN controller placement in [130] introduces the problem of SDN controller placement as a problem of deciding how many and where SDN controllers can be put in a given topology. Their focus throughout the paper remains on latency reduction where the controllers are placed in a position that optimizes the average or the worst case latency. They introduce simple placement algorithms which try all possible locations and run in linear manner; yet they do not yield optimal results. In [131], the authors focus on trying to satisfy multiple objectives in a controller placement problem such as, minimizing controller to switch latencies and controller to controller latencies, while providing load balancing. They establish a pareto-optimal frontier where the performance of one of those metrics can only be increased with the loss in another metric. The system designers need to decide on the optimal trade-off. In [132], the authors focus on assigning switches to statically deployed controllers in an efficient manner, in particular, the assignment maximizes the number of disjoint paths between the controller to the switches to increase reliability. [133] proposes a MIP formulation of the controller placement problem. The authors presume distinct subsets within the network graph, notably switches, number of available controllers and potential controller positions respectively, along with other parameters. Then the proposed model can minimize the cost function which is sum of planning costs, switch to controller costs and controller to controller costs (e.g. for states synchronization) under different constraints (such as controller capacity limit, CP bandwidth per switch).

Yet, none of this work considers a generic solution which can be utilized for different types of controller designs, i.e., both replica and hierarchical controllers. What is more important is that virtualization is a key enabler in SDN networks. In a truly virtualized SDN environment, the controllers will be virtual entities connected to each other via virtual control links and requiring additional support functions, such as databases, to form a control plane. Placement of virtualized controller is still a new topic to be investigated.

6.2 State of the Art on Dynamic/Adaptive Flow Rules and Hierarchical Rule Caching Systems

Dynamic and adaptive flow rules is a solution that addresses the challenges of the distributed SDN controllers, especially in a dynamic environment. By adding the adaptation capability to the flow rules, the amount of required controller – switch communication can be reduced. This in turns reduces the processing load and processing time at the controller. The reaction time of the network is greatly reduced and thus the user experience is improved. Different to the architecture based approach, dynamic and adaptive flow rules do not depend on the local control plane intelligence at the switch (e.g., agent based approach) and do not require additional inter-controller signaling (e.g., states synchronization among distributed controllers). In addition, adopting dynamic and adaptive flow rules is an orthogonal scalability solution to the architecture based solutions, therefore can be used together with distributed/hierarchical controller architecture and complement each other.

There are many works in the literature which try to introduce more flexibility into the SDN network by enriching the flow rules. In [173], Ethane is presented that is an early appeared SDN network architecture with a single controller installing only reactively flows, based on the first packet of each TCP/UDP flow. In this paper, authors investigate what is the maximum size of a network that could be served by a single controller and the results indicate the necessity for distributed and hierarchically connected controllers.

The hierarchical rule caching architecture enables the support of high speed but usually small size memory caches, that are located close to the network device for quick response, as well as the operation of lower speed caches that, however, have more space for rule caching and are less expensive. In an ideal world, all possible rules that represent all the possible combinations that are available in a given flow space would be proactively installed in each network device. Since this is not possible, the intermediate caches presented before come to the rescue. Evidently, all rules cannot fit in the 'lowest' (or 'closest' to the device) cache, and this is why the 'higher' caches are also required. In the hardware OpenFlow switches, there is a high speed Ternary Content Addressable Memory (TCAM) that is considered to be the lowest rule cache. The switch stores the OpenFlow rules to this memory and once a new packet comes, it performs a parallel lookup to quickly identify the highest-priority match for this packet. While TCAM enables fast lookups with flexible wildcard rule patterns, the cost and power requirements limit the number of rules the switches can support. The hardware switch does a fast lookup in the TCAM, and if there is a 'cache miss' (meaning that there is no rule in TCAM matching the packet), then a *pull (request-response)* process is initiated from TCAM to the upper layer cache to find such a rule. If, again, there is a cache miss, then the pull request is forwarded to the controller. There is also the proactive *push (request)* process that is initiated by the higher caches and moves the rules to the lower caches proactively. For example, MicroTE [35] is a system that tries to predict network traffic and adapts to the traffic fluctuations by dynamically and proactively updating the rules in the switches.

In [172], authors present CacheFlow that is a transparent caching layer that logically sits between a SDN controller and a collection of OpenFlow switches. It makes a collection of hardware and software switches, which act like a single switch with an infinite rule capacity, and the CacheManager module, which receives OpenFlow commands from the controller and uses the OpenFlow protocol to distribute rules to the underlying switches. The switches form a cache hierarchy, where packets that experience a cache miss in one layer are forwarded in the data plane to a switch at the next layer for handling. For example, a simple configuration could be one hardware switch connected directly to a shared cache consisting of multiple software switches. On the one hand, the hardware switch provides high port density, high throughput, and a modestly-sized TCAM. On the other hand, the software switches provide high rule capacity at reasonable throughput to handle “cache misses” in the hardware switch. Of course, the cache hierarchy could be wider (with multiple switches at each layer) or taller (with multiple layers).

The challenging issues in this layered rule caching architecture are to estimate the number of rule caching layers that should be exploited, or the available memory space at each layer (as the memory size expands, the delay of the response increases), or the algorithm that defines which rules are placed at each cache and how they are stored. In [174], authors partition the field space into logical structures called buckets, which are cached along with all the associated rules. Through CAB, the rule dependency problem is resolved with small storage overhead, while the flow setup requests are reduced by an order of magnitude, saving control bandwidth by a half and significantly reducing average flow setup time. In [175], authors aggregate again flow entries by spanning several transport connections. In [176], authors present Minimum Weighted Flow Provisioning (MWFP) that deals with the NP-hard problem of estimating how many rules should be stored in the lowest cache of TCAM and how many should be retrieved reactively by the remote controller, in order to minimize the sum of remote processing cost and TCAM occupation cost. They propose an offline algorithm by adopting a greedy strategy if the network traffic is given in advance, as well as two online algorithms with guaranteed competitive ratios.

In DIFANE [177], a compromised architecture is presented that leverages a set of authority switches serving as a middle layer between the controller in control plane and switches in data plane. The endpoints rules are pre-computed and cached in authority switches. Once the first packet of a new flow arrives the switch, the desired rules are reactively installed from authority switches rather than the controller. In this way, the flow setup time can be significantly reduced. However, the caching of the pre-computed rules in the authority switches consumes large TCAM space.

In [178], a new caching structure is proposed that improves network performance with a limited rule space in the SDN-enabled mobile access network. A two-layer rule space is exploited in each SDN device, which is managed by the SDN controller. A cache prefetching mechanism is also used with the consideration of user mobility. More specifically, a memory manager is inserted as a module in the existing programmable SDN devices, and a cache manager as a network application is deployed on the centralized controller. All rules are stored in SDN devices before the cache manager updates the rules. In the cache manager, three modules are used to support the cache mechanism: the rule recorder, the topology monitor, and the rule updater. The rule recorder stores all rules and the characters of their corresponding flows, while the rule updater sends updated information to the SDN enabled switches. All topology information is stored in the topology monitor to support the decision of rule updating. The rule updater receives the cache miss information and checks if this information causes cache updating. This updater checks the rule recorder and decides which rules need to be replaced by the rules for the new flows. How to make this decision is a problem dependent on the predicted flow traffic and the cache access history.

The P4 language specification [136][137] defines the concept of action profile that enables several entries in the forwarding table to share the same action set. Action profiles can also be dynamically bound to a match entry by using an action selector. The P4 action selector chooses a particular action profile entry for each packet either pseudo-randomly or deriving the decision from header fields and/or meta data. While this option brings dynamicity to the forwarding tables of the SDN switches, the set of actions from which the action selector can choose is still operating only on the data plane (e.g. modify packet header, forward to next forwarding table, etc.). [139] implements a Finite State Machine (FSM) at the SDN switch. The switch performs different actions (e.g., reconfigure the flow tables) based on the local state at the switch.

The problem of layered rule caching is even more challenging when the hardware switches are replaced by the widely used Open vSwitch (OvS) [138] software instances. The actual software structures that get looked up in these virtual switches feature an aggressive eviction scheme (i.e. forwarding rules that remain unused for a user-defined number of milliseconds get evicted). The user space cache that supports the kernel datapath resides in a user program called OvS-vswitchd deamon - which is designed to run on the same machine. Therefore, at the kernel level, the actual datapath lookup tables are always configured reactively. Once a packet appears and the datapath lookup table has an entry, then the packet goes through the 'fast-path', which means that is quickly forwarded without searching in the user level cache. Otherwise, if there is no entry then the packet goes through the 'slow-path' and the user level cache is transparently searched. In case the user space cache does not have a relevant entry an appropriate request is generated to the controller. In order to reduce the miss penalty, the controller can proactively push flow rules to the userlevel cache, but he/she *cannot interfere with actual datapath cache which is always reactively configured*. Evidently, the user space cache can be a larger data structure, but given that the datapath lookup is rather small, the misses can be relatively frequent, so the user space cache size should not be excessively large to the point where a lookup is rather slow. Moreover, OvS provides the action learn that allows modifying an existing flow table based on the content of the flow currently being processed at the switch.

6.3 State of the Art on Reliable SDN control channels

In a programmable network, a reliable control channel is a key pillar that enables an in-band controller to operate the network in a centralized way. The reason is that network updates from the controller to remote switches rely on the control plane forwarding. This is extremely important, especially in a high dynamic environment where the network topology changes frequently. One example would be mmWave backhauling network in 5G-XHaul, where the small cell base stations are connected via mmWave links in multi-hops to the base station with wired connections (e.g., optical fiber). The network topology can be changed due to several reasons:

- Switch on/off a small cell
- Change of the 802.11ad coordinator
- Change of the membership of 802.11ad service group, i.e., Personal Basic Service Set (PBSS).

Assuming an in band control channel by mmWave transport, the key challenge here is to maintain a reliable control channel in highly dynamic network topologies.

Centralized management of the control plane connectivity includes monitoring its reliability and re-establishing the control plane connectivity when several links are unavailable. According to our study, both of them are computationally complex problems. The first one is #p-complete [126] and the second one is NP-complete [127]. Therefore, it is necessary to design self-organizing solutions to guarantee the reliability of the control plane.

When it comes to self-organizing, it is natural to consider existing well-studied distributed routing protocols. Various routing protocols were proposed to meet different traditional network systems ranging from local area network (LAN), Internet, data-center network, to mobile ad-hoc network (MANET) and so on. For instance, OSPF [128] can be used for LAN, BGP [129] for multi-domain networks, and DSDV [130], AODV [131], DSR [132] for MANETs. However, those protocols are known to come with large flooding overheads, complex routing policy, not to scale to large network sizes, and not to be agile to network dynamics.

Firstly, these protocols were proposed originally for traditional networks where once the routing tables are established, the forwarding policy is static thus there is no need on frequent control traffic forwarding. While in a dynamic environment, flow tables need to be dynamically updated by a (remote) controller, and control traffic forwarding is frequent. As a result, using an existing protocols directly to organize the control plane does not satisfy the requirements in 5G-XHaul scenarios.

Secondly, given different types of network infrastructures, a routing protocol needs to be chosen accordingly. For example, routing protocols for wireless sensor networks focus on power consumption while routing protocols for MANETs consider more the effects of dynamics to the network connectivity. It is impossible to decide on the appropriate routing protocol to self-organize the SDN control plane because the control plane requirements depend on the actual scenario and are unknown beforehand.

One may argue that the particular type of network could be known a priori, and thus various standardized routing protocols could be pre-installed in the network appliances. Firstly, this is equivalent to run two networks in parallel where one of them will be a traditional network. Secondly, this is wasteful and costly to implement only for the control plane communication.

Moreover, one may expect the control plane to support more advanced functions rather than connectivity only, e.g. distributed controllers, thus an upper layer peer-to-peer protocol such as distributed hash table (DHT) may be needed. P2P DHT protocol/application (e.g., CAN [133], Pastry [133], Kademila [134] and Chord [135] can be deployed over the lower layer routing protocols, however, it is clear that any P2P DHT protocol needs networking support from the lower layer, which has already had the above issues. A multi-layer solution makes the control plane even heavier.

Issues in traditional multi-layer solutions have been realized and lightweight cross-layer protocols have appeared particularly from MANET research. One classical work is the virtual ring routing (VRR) protocol proposed in [137]. Its idea is to structure all nodes in a wireless ad-hoc network as an ID-based ring overlay topology. For each node, the protocol establishes routes to their virtual neighbors in a logical ID space. One key feature here is that organizing nodes as a ring facilitates the support of DHT functionality like a Chord system. Later, similar protocols were proposed such as in [138][139][140][141], which used different approaches to establish routes among virtual neighbors or more extra nodes. Similar examples can be also found in [142][143][144][145][146].

Directly using VRR-like protocols as the control plane solution can only fulfill our goals partially. First of all, those protocols are not self-adapting because they merely consider the connectivity of the network nodes. For an SDN control plane, as it transports control traffic that has a higher priority than data traffic, its performance is more critical. The required self-organizing protocol needs to autonomously see the conditions of the network and adapt the overlay topology of the control plane balancing between resilience and efficiency.

Second, VRR-like protocols are not aware of the underlying network topology. Actually, a P2P system has no such need because it originally aims for distributed file sharing among equal peers. In their settings, there is no the controller entity location issue. In an SDN, however, the underlying topology directly determines the potential location(s) instantiating the controller(s). Therefore, a good control plane is aware of the topology of the underlying infrastructure and if an SDN operator requires the SDN to self-instantiate its controller(s), the control plane should be able to provide such candidate locations.

6.4 State of the Art on SDN North bound interfaces

A North-Bound Interface (NBI) is an API that interfaces an SDN platform with external applications making use of it. An analogy in the realm of Operative Systems is the POSIX interface for Unix systems, which allowed applications to be easily ported between different Unix-like systems. Similarly, the NBI for SDN platforms should allow network control applications to be ported between different SDN controllers. Therefore, the NBI is a critical component for ecosystem creation, and different approaches have been proposed to date both in the industry and academia, without having reached consensus yet.

Initial proposals for an NBI have been put forward by the communities contributing to open source SDN controller platforms such as OpenDayLight (ODL) [179] and ONOS [129]. A common approach is to generate a NBI that exposes network or configuration state through a REST API. The Representational State Transfer (REST) is an architectural framework for distributed systems, whereby data elements are exposed as resources over which a set of operations can be performed, namely GET, PUT, POST and DELETE operations. These operations are executed through request/response interactions between a client and a server. The REST paradigm is very much extended in the Web, and is supported by multiple tools in many programming languages, which is key to foster the quick adoption of SDN controllers in the market.

REST is however less suited to handle event based interactions, where for example an application needs to be notified when certain state in the network has changed. An approach to deal with this limitation is to use a NBI tightly bound with the controller platform itself, such as the OSGI/Java interfaces used in ODL. Using internal OSGI/Java interfaces enables event based communication between the applications and the SDN controller, but forces the applications to be tightly bound to the controller platform, i.e. developed using the same programming language, which hinders market adoption. Therefore, other approaches to NBI design have recently appeared based on message oriented middleware (MOM). A MOM decouples the SDN controller from the applications by means of message queues and publish/subscribe capabilities. In particular, message queues can support both request/response interactions triggered by the applications, and a publish/subscribe framework for event-based interactions. Several MOM protocols exist that can be applied to SDN controllers, such as AMQP [180], STOMP [181], MQTT [182], and XMPP [183]. An initiative to develop a MOM based NBI for ODL is currently discussed in the Messaging4Transport project [184].

A recent trend in the design of NBIs is the use of “Intent based interfaces”. This design approach is pushed by the NBI working group of the ONF [185], and its goal is to design an NBI interface that can be easily ported to different SDN controller platforms. Intent based interfaces should be declarative allowing the control applications to focus on what action or state they desire to be deployed in the network, instead of how this action or state is deployed in the network. Thus, an intent can be described as composed by an *Operation,* an *Object* and a *Condition*. For example a control application could express an intent in the following way: “I want a 10GB connection between Port 1 and Port 2, but I can accept to take 5GB to 8GB”. The reader is referred to [186] for a list of use cases on Intent based interfaces. Intent interfaces are currently implemented through REST APIs. Information on the intent framework supported in ODL can be found in [187], and on the intent framework supported in ONOS in [188].

Another approach to the design of NBIs considered within the IETF community is the use of an ALTO server to complement the SDN controller. Application-Layer Traffic Optimization (ALTO) [189] is an IETF protocol used to deliver applications information about the network. In particular, an ALTO server collects topology data from a network and aggregates it to build a set of network maps considering aspects such as provider policy, privacy, confidentiality, etc. Applications can access this information from an ALTO server encoded in JSON through a REST interface. Hence, in an environment where ALTO is integrated with an SDN controller, the SDN controller can become a source of network information for the ALTO server, and the control applications can query the ALTO server for information about the network. Thus, ALTO can be seen as part of a potential NBI.

Another body of work on NBIs for SDN controllers has focused on how to integrate an SDN controller to higher level orchestration entities, for example with OpenStack [190] in the case of data centre deployments. OpenStack defines a network related plug-in called Neutron [191], which can be seen as a control application that communicates with an underlying SDN controller in charge of deploying the network services requested by Neutron. As a matter of example, ODL offers the Virtual Tenant Network (VTN) service [192] that can be hooked to Neutron to instantiate virtual networks connecting the Virtual Machines (VMs) instantiated by OpenStack. VTN is an ODL service that provides virtual network provisioning, flexible traffic control, and automatic detouring on link failures. Consequently, a network in OpenStack is mapped to a virtual bridge (vBridge) in VTN, and a port in OpenStack is mapped to vInterface in VTN. Having a NBI interface that allows a higher level SDN/NFV orchestrator to control an SDN platform is very relevant in the case of 5G-XHaul, where the transport network is going to be a component orchestrated by a higher level 5G entity.

In the academic community several network operating systems have been proposed each of them offering a particular NBI to interfaces running on top of it. A relevant example is Onix [193] that offers to applications the abstraction of a Network Information Base (NIB) as a data store containing all network state and configuration capabilities. Applications can read and write to the NIB using a REST interface, where Onix provides a framework to resolve potential conflicts. Finally, also within the academic community a second kind of NBIs have been proposed that are defined by the abstractions offered by SDN programming languages. Examples of this type of NBIs can be found in Frenetic [194], Pyretic or NetKAT [195]. The interested reader is referred to [196] for a survey on this type of solutions.

6.5 5G-XHaul hierarchical control plane Requirements and Initial Design

Based on the works introduced in the previous sections, we introduce in this section the requirements to be fulfilled by the 5G-XHaul control plane, and present an initial design of the 5G-XHaul controllers and rule caching system that will be further elaborated on subsequent WP3 deliverables.

6.5.1 5G-XHaul requirements on control plane

The 5G-XHaul SDN controller should satisfy the requirements mentioned below and summarized in Table 7, that will be further elaborated in subsequent deliverables.

Table 7: 5G-XHaul SDN controller requirements

Requirement	Summary	Description
R1	Scalability	The distributed architecture of the 5G-XHaul controller will offer scalability concerning the network size.
R2	Time sensitivity	The rule caching architecture of the 5G-XHaul controller should enable quick response to reactive rule forwarding to the datapaths.
R3	NBI allows loosely coupled applications	The 5G-XHaul NBI will allow the development of applications loosely coupled to the 5G-XHaul platform. In this sense REST or message oriented NBIs will be considered preferentially
R4	Exposed state	The 5G-XHaul NBI will expose long and short term network state to applications, as well as providing a framework for applications to communicate desired policies in a declarative manner.
R5	Integration with NFV orchestrator	The 5G-XHaul NBI will allow integration of the 5G-XHaul controller to a higher level NFV orchestrator

6.5.2 Initial design on 5G-XHaul hierarchical SDN controller and rule caching system

The 5G-Xhaul controller should be distributed and follow a layered approach, where each layer is responsible for different operations (c.f. Section 2.2.2). The major functionality carried out at each controller level can be illustrated through an example. Consider a tenant defining a slice according to the abstraction introduced in section 3.2.2. As we have already explained in Section3, the tenant indicates the ETN where each VNF/PNF included in the slice is connected, where the selected ETN may be located in different 5G-XHaul areas. Thus, once defined, the slice is submitted to the 5G-XHaul Top controller through a north bound interface (NBI). The responsibility of the 5G-XHaul control plane is then to wire the transport tunnels connecting the ETNs involved in the tenant's slice. The first task of the Top controller is to look up the Level-1 controllers in charge of the ETNs included in the layer two segments defined in the received slice. Once the Level-1 controllers are identified, the Top controller runs a path allocation algorithm to establish a path between all the ETNs participating in the same layer two segment. The path determined by the Top controller is expressed as a set of Level-1 controller areas, whereby a Level-1 controller area is composed of all the 5G-XHaul areas, where the corresponding Level-0 controller is controlled by the Level-1 controller. For each determined path, the Top controller requests the involved Level-1 controllers to allocate a transport connection between ETNs under their control, or between an ETN and a neighbouring Level-1 area. In order to determine these paths, the Level-1 controllers run a path allocation algorithm that returns the set of 5G-XHaul areas composing the path, along with the corresponding Level-0 controllers in charge of each area. Consequently, for each path, the Level-1 controller submits a request to the corresponding Level-0 controller, which runs a path allocation algorithm to identify the transport tunnels and paths connecting each involved ETN and IATN within the 5G-XHaul area under its control. Once the process completes the tenant slice is fully connected and communication may begin. Table 8 illustrates the high level functionality included at each controller level.

Table 8: High level functionality at the different controller levels

Level	Function	Description
Level-0	Path Allocation	Allocate tunnel path inside the 5G-XHaul area
	Topology Management	Maintain topology within 5G-XHaul area
	QoS & OAM	Maintain per tunnel statistics and OAM metrics
	ETN end point discovery	Discover per tenant end points connected to an ETN in the area
	IATN Discovery	Discover IATNs in the 5G-XHaul area, and the areas they connect
	NBI to Level-1	North Bound Interface to Level 1 controller
Level-1	Inter-Area Path Allocation	Allocate paths between ETNs at area level
	Inter-Area Topology Management	Maintain connectivity graph between 5G-XHaul areas
	Area level QoS & OAM	Maintain area level QoS and OAM metrics
	NBI to Top controller	North Bound Interface to Top controller
Top	Inter-Level 1 Path Allocation	End to End path allocation at Level-1 controller level
	NBI to service	North Bound Interface to service / VIM / orchestrator
	Tenant and Slice management	Generation and assignment of unique slice IDs and Tenant IDs

Moreover, the 5G-XHaul controller should have the responsibility of cache monitoring and orchestration support. The caches should be inquired for their health and statistics to determine the overall effectiveness of cache operation. The cache manager will also decide and enforce rule updates (i.e. changing the behaviour of existing rules) and will decide which cache should host which rules. It will also need to handle coherency issues. With the aforementioned support the 5G-XHaul controller should be able to tune the caching system, at runtime, to realize an elastic control plane.

An indicative cache node anatomy is depicted in Figure 32. The idea is to build one type of software cache node and let the cache manager decide its role and position in the hierarchy, at runtime. The node should feature a Push/Pull API to support rule transactions, a rule object store where rules in the form of objects will be stored and cache hierarchy information store where the role of the cache will be realized. The ability to retarget the same node at different level of the hierarchy at runtime, lies at the generic APIs that allow for a clean separation of concerns, the configurable cache hierarchy info, and most importantly, the flexible Rule format. In the example below, we outline the basics of a rule object. It should have a RuleID that will uniquely identify and enable its update, Matching Info and Forwarding Actions which can be quite generic and appropriate for specific rule generation and, finally, rule handling functions that can be invoked to make rules more specific. This design will allow the cache manager, to dynamically reconfigure cache hierarchy, switch nodes off if not needed or engage more and flexibly realize optimal behaviour.

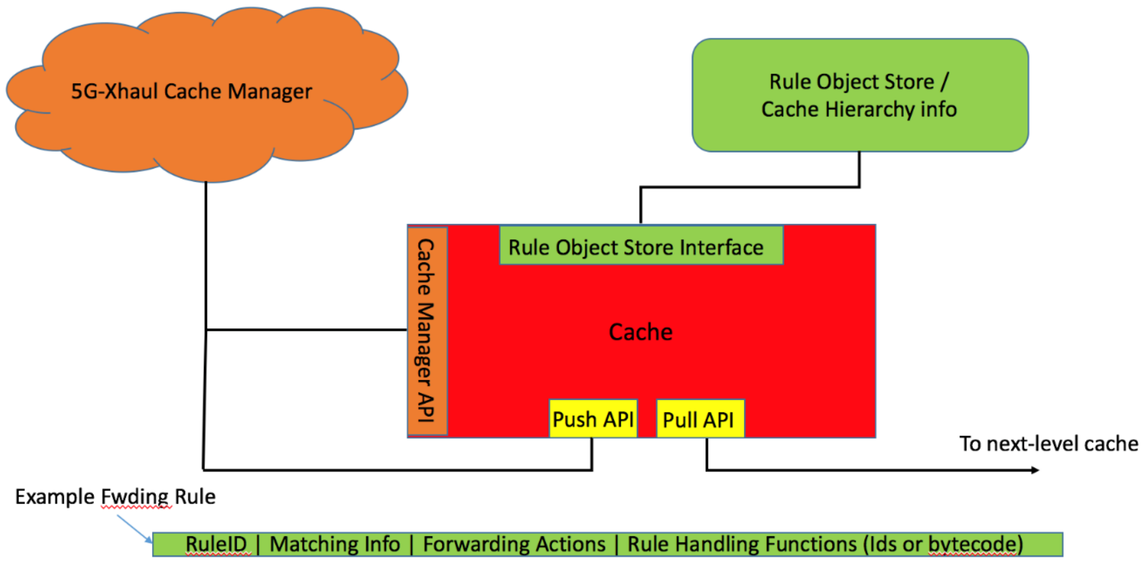

Figure 32: Initial Cache Node Design.

Finally, in the 5G-XHaul controller architecture, Dynamic and Adaptive flow rules can be used both by the local ETN controller (local Ctrlr) as a template to adapt to different tenants and in External Per-tenant Hierarchical Cache as a flow rule abstraction (as shown in Figure 33). This allows for certain degree of local adaptability. For instance, multiple tenants can require the same transport service (e.g., use the same transport class between two ETNs). The same flow rule template can be used where the matching field of tenant ID can be adaptive to the actual use case. In the hierarchical rule cache system, the tenant rules stored at each hierarchical external cache level are used by a group of switches at the same level (e.g., several ETNs in the same area). Making the rules as general as possible could reduce the number of rules stored at each external cache. The general rules can adapt to the situation at the target ETN where it needs to be installed. Here the adaptation is in the abstraction level and is independent to the actual transport technology. For instance, different ETNs might have different virtual port pointing to the same logical DP NFV of the tenant.

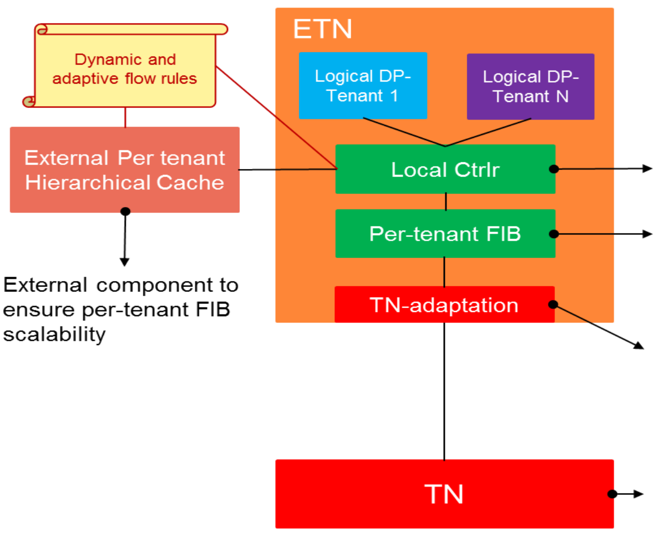

Figure 33: Usage of Dynamic and Adaptive Flow rules in 5G-XHaul control plane.

7 USER MOBILITY AND RAN-TRANSPORT INTERACTION

In essence, for 5G, the transport network will convey traffic to and from different RAN domains, using different 5G and legacy RATs. The upcoming 5G networks are expected to support the evolution of current services, and new services, which will imply new requirements on the RAN. This would consequently bring new requirements on the transport network which, in short term, will not be able to cope with the demand for beyond-2020. Therefore, a new joint mobile and transport networks perspective has to be brought up for the evaluation of transport technologies, supporting the belief that no single solution can solve the holistic 5G transport problem.

Typically, the transport network maps RAN traffic to a relatively small number of quality of service (QoS) classes used by transport network operators. In the 5G context, the vast deployment of small cells into existing mobile networks can improve throughput and users' QoS. In such a dense wireless deployment with mobile nodes and limited spectrum, it however becomes a difficult task to allocate radio resources, implement handovers, manage interference, balance load between cells, etc.

There are use cases for network flexibility that require coordination between the control and management of the mobile network and transport network resources, namely mobile-transport network interaction. These use cases are particularly interesting as they determine the control plane architecture where relevant information needs to be shared between controllers of different domains. These use cases are presented in Section 7.3.

This section is structured as follows. In Section 7.1, we survey the state of the art in Mobile-transport interaction and mobility awareness. In Section 7.2, we introduce specific Mobile-transport interaction requirements derived from the candidate technologies considered by other 5G-PPP projects in the design of 5G RANs. In Section 7.3, we provide an accurate analysis of use cases that require a joint Mobile-transport interaction. We conclude this chapter in Section 7.4, where we put forward our requirements and views on this topic, and draft an initial design of our planned interaction with 5G mobile network.

7.1 State of the Art: Mobile Network - Transport Interaction, User mobility awareness

In the current 5G vision, there exist many research paths which require certain level of coordination between the transport network and the mobile network. We first cover a general view of this interaction, to then move to one of the specific use cases tackled within 5G, which is the user mobility awareness and the way the transport and mobile networks must interact upon this.

7.1.1 Mobile Network -Transport Interaction

The main goal from the coordination between the mobile and transport domains is to improve the Quality of Experience (QoE). In current networks, sources of degradation of the QoE are often an indication that network resources are overutilized, i.e. the network could not be dimensioned for the worst-case traffic scenario. Overprovisioning is the approach currently leveraged by mobile operators, offering low complexity at the expense of high cost and inefficient use of resources. The other approach, to be considered nowadays, is the allocation of transport resources on demand to cope with the specific transport needs, which may vary over time. In this context, novel networking functionalities are playing a role currently, such as NFV and SDN.

SDN benefits and builds on coordination and information exchange between RAN and the transport. Caching is another feature that capitalises on context-awareness and instantaneous network information (e.g., system interference, QoS requirements, and transport capacity) to achieve gains in alleviating the transport loads during peak hours, consequently, coordination between RAN and transport is crucial. This coordination also plays a key role in terms of the energy efficiency in the transport, tackling with wired and wireless transport technologies and architectures as part of the energy consumption model.

There are several underlying sources for dynamicity in the network that might impact the transport network [197]:

- Adding/removing network resources/elements: connectivity, compute and storage resources;
- Service deployment: deployment of new services, such as new end-user services or supporting services to service providers;
- UE dynamicity: dynamic traffic patterns from user movement/migration, variations in user activity;

- Service dynamicity: dynamic service usage patterns with wide range of service requirements and impact of dynamic system behaviour;
- Failures and service windows: re-routing traffic and minimizing impact;
- Weather conditions: changing weather conditions impacting performance in the transport network (microwave, free-space optics).

7.1.2 Joint transport and backhaul design - Awareness

A joint radio access and backhaul design was introduced by the BuNGee project [198], which raised the benefits of such joint operation to optimise the performance and efficiency. The iJOIN project identified joint BH/RAN design as an enabler for the next generation of networks, bringing up the terminology of RAN as a service (RANaaS). This joint design tackled RAN awareness on one side, and joint RAN/BH functional design [200] on the other side. iJOIN defined a flexible RAN architecture, i.e. neither fully distributed nor fully centralised [199].

As examples of BH/RAN awareness are BH aware resource allocation (e.g., [201], [202]) and cell association (e.g., [203]). In [204] the authors propose adjusting the radio coverage of a cell in view of the BH availability and capacity, exploiting reinforcement learning to adjust the cell range extension offset. The RAN therefore leverages from BH information to redistribute users in a way that maximises user QoE, and that adapts to temporal BH constraints. A centralised optimisation mechanism to also adjust the cell range extension offset was presented in [205], with the goal of minimising the mean network packet delay. In [206] the authors tackle BH latency and resilience via a BH-aware user association, with the goal of improving QoS while performing load balancing.

From the SDN perspective, the resources in different network segments, e.g. mobile and transport networks, are managed independently by separate logically centralized controllers. Multiplexing of time and frequency resources between the transport and the mobile network links attracts attention, where the two links are operating on the same frequency band in a dynamic TDD, or when they operate in the same spectrum, requiring strategies and algorithms for controlling transmit power, channel allocation and data rate.

Works tackling problems regarding radio resource management (RRM) focus on two classes of solutions: resource virtualization and resource abstraction: Regarding resource virtualization the research works in this direction [207] have attempted to determine which functions should be centralized and virtualized on the cloud and deployment model, such as fully centralized or partially centralized; The resources, e.g. base stations in a geographical area, are abstracted as a big base station or a big cell that consists of a RAN controller and radio elements. The RAN controller will dynamically schedule and allocate radio resources to each radio element. The works in this direction, e.g.[208], focus on methods of allocating radio resources from a radio resource pool in the controller.

The optimal level of transport resources abstraction is being discussed within standardization bodies, as in [209]. In [210], the authors propose a proof of concept of an SDN-based control plane for joint orchestration of radio and transport resources. Different domains of radio, transport, and cloud are studied in [211] together by an orchestration layer, where different types of resources from several individual control domains provide a joint view towards applications or higher-level orchestration. Finally, in [212], authors offer a RRM perspective to the 5G BH problem, discussing the potentials of BH-aware resource allocation in a multi-RAT environments, and proposing the usage of a unfied wireless BH bandwidth allocation in a small cell case study employing in-band backhauling and massive MIMO.

7.1.3 Support for mobility: mechanisms and user mobility awareness

The envisioned 5G heterogeneous networks and their deployment entail a number of challenges in terms of capacity, coverage, Mobility management (MM) and mobility load balancing across multiple network tiers [226]. In particular, MM is essential to ensure a continuous connectivity to mobile user equipment (UE) while maintaining satisfactory quality of service (QoS). The 5G network environment then poses the following challenges on MM:

- Increased number of potential handovers, and thus core network (CN) signalling.
- Handover failure and ping-pong issues given the small cell size.

Given the above challenges, new management procedures and network architectures that account for these mobility challenges are required.

Predicting users' mobility in wireless networks has received a great deal of attention recently, strongly motivated by a wide range of applications which increase year by year. Traffic in the different RAN areas is expected to fluctuate with the movement of users and dynamic service patterns. This dynamicity could be exploited for several reasons, such as improved utilization of centralized RAN resources through pooling, for a leaner dimensioning and utilization of transport resources by provisioning connectivity only to areas where it is needed, and for saving energy both in the RAN and transport by powering off cells and transport interfaces that are not in use.

UE mobility is currently a key aspect in LTE networks. Efficient MM in the RAN is required for a seamless service experience for users on the move. MM is currently implemented in the Mobility Management Entity (MME), who is in charge of collecting a series of functionalities ranging from tracking areas - where to page the UE - to UE context management. Thanks to SDN and NFV it is possible to re-design part of the system, allowing a distribution of the functions for improving both resilience and scalability in the network. As the RATs support multi-connectivity (between different nodes or inter-RAT between RAT and legacy systems as LTE-A), the MM should support the required coordination between the nodes even when the transport network is capacity limited, or introduces some latency.

An L2 mobility approach similar to the LTE radio handovers management with late path switch was implemented in [227]. A L2 tunnel between source and target APs is established for packet forwarding. This tunnel could be based e.g. on Ethernet MAC-on-MAC encapsulation. In the meanwhile, the target AP issues a more permanent path switch request to the SDN controller, that sends the reconfiguration command to the various switches along the path.

On horizontal handovers, controllers can explore geolocation to reduce path changes [229], [230]. They can also improve vertical handovers, speeding up control signaling and providing load balance across different technologies [231]. Even at times without UE mobility, a centralized controller could balance the traffic among EPC nodes, or find new routes in response to failures [232], [233]. Although these aspects are identified as open issues by several authors, they are left only as motivation, and few practical solutions are presented.

In [226] the authors claim to decentralise part of the MME functionalities down into the access network, either in the base stations operating as L2 switches, or at the L3 routers interconnecting several base stations. These decentralised features would be operated in nearby data centers.

Mobility prediction is a technique to identify future targeted base station in advance, to reduce handover latency, and finally to enhance handover performance in wireless networks. Since 5G development is in the early stages, there is a unique opportunity to develop and integrate mobile radio based positioning technology in 5G from the beginning. With an initial integration of appropriate positioning technology the impact on communication can be minimized and synergies between communication and positioning can be exploited.

The most common technique of mobility prediction is based on user's mobility history. Based on this technique, a network requires to track and record movement information about the user. The existing algorithms estimate the next location of a specific user by inspecting the data available about her past mobility, i.e., her past trajectory, and exploit the inherent repeated patterns in the data. These patterns correspond to a (regular) behavior of the user, e.g. commuting from home to work or visiting favourite restaurants, and need to be extracted from the data to provide accurate predictions. To this aim, one has to observe the behavior of the user over long periods of time. As presented in [214], the network searches the user's route and predicts a set of potential handovers based on the user's history and current location. However, this technique is only applicable for regular user or if the network has the user's mobility history. The authors in [215] have considered spatio temporal technique to predict the next location of the user. The technique considers the user's location and time as well. Besides the frequently visited places, the total time that the user has spent in the places is also computed. Many other mobility prediction methods and algorithms based on these techniques have been devised over the last decade, see e.g. [216][217][218][219].

Another possibility is the use of Markov Chain techniques to predict user's movement [220]. Standard Markov Chains are memoryless, as the prediction of the next state only depends on current state. However, in [221], the authors have extended the limitation of standard Markov Chains as the prediction depends on n-previous location.

The distance between the UE and the access node can be estimated from the received signal strength (RSS) of an uplink pilot signal with known transmit power by converting the resulting propagation loss into a distance with the help of a path loss model [8]. Alternatively, such a distance can be estimated with the time of arrival (ToA) method, which measures the propagation time of a signal from the UE to the AN [222], typically requiring very accurate clock synchronization.

Yet another technique for mobility prediction is based on user movement direction [223]. Here the prediction is based on the user's direction and signal strength. The user's direction can be calculated based on an angle of direction between mobile node and desirable access point. If the angle is less than an angle threshold, the mobile node is moving towards the access point and handover can be triggered. This proposed technique shows a decrease of latency in L2. The authors in [224] propose a prediction based on user's moving direction where the handover request is triggered based on conventional received signal strength (RSS). Predicted cell is defined by calculating a variance of relative distance from a mobile user to each of candidate target cells. Moreover, the position of users can be calculated via Angle of Arrival (AoA) estimation [226].

7.2 Long-term transport network requirements

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. It is also expected that the use of technological solutions like beamforming, network MIMO and FD-MIMO will make the concept of cell edge much more diffuse, equalizing the SINR distribution inside the cell. Also, the use of high frequency bands (combined with beamforming) makes the reliability of the radio connection between user and network more dependent on the potential presence of obstacles rather than on interference or coverage levels.

In this sense, there are several developments expected to happen that may affect to the transport infrastructure:

- The possible evolution of the RAN architecture towards a cell-less one, where the user plane for the UE is not dependent on the cell it is connected to[6], but on a specific, non-variable identifier which is valid through the whole network or part of it (e.g., the one that is controlled by the same physical or virtual network entity). In this context, it is not the UE that detects that it has moved to a different part of the network by scanning the signals that the cells broadcast, but the network that detects that UE is moving by scanning the signals transmitted by the UE. From the UE viewpoint, there are no handovers within the area, but seamless mobility between different radio points. The support of this architecture implies that the transport network should be able to map the UE's traffic flow templates (that include the IP address and port) to different radio points in a dynamic way. The upgrade rate can be very high (e.g., on a per packet basis), so this may be a very stringent requirement for the controller of the transport network connectivity plane. The cell-less architecture should require also to support the capacity to reconfigure the anchor point where flows from the UE that come from different radio points are combined. The transport infrastructure should also be able to support the fast transmission of the control plane critical information, e.g., messages indicating the change of the radio point to transmit to or receive from the UE. This may require the support of frame preemption capabilities in the transport infrastructure. This transition to a cell-less architecture is illustrated in Figure 34.

[6] Scheduling, scrambling, ciphering and other user plane functionalities are linked to the temporal identifier that is assigned by the cell the UE is connected to.

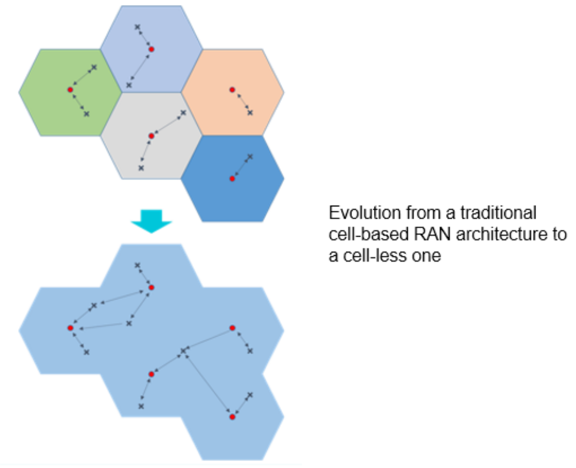

Figure 34: Cell-Less Architecture.

- The need to support multicast or other multipoint cooperative mechanisms in order to overcome blockage issues when using high frequency bands with beamforming. In addition to the blockage that may happen due to large obstacles, like mountains or buildings, high frequency links may be obstructed by small obstacles, like vehicle, people or the body of the user. For the support of these features, the transport infrastructure should be able to create multicast trees in a dynamic way, so information may be transmitted the UE from different radio points simultaneously.
- Support of multipath communications, in order to take advantage of the availability of different RATs in a seamless way. It is expected that multipath anchoring will extend from the TCP (multipath TCP) and PDCP (inter-site Carrier Aggregation, LTE LWA) layers currently supported to other layers, mainly to the MAC layer, so all the protocol stack among different RAN air interfaces will be shared except for the PHY layer, as represented in Figure 35. The transport infrastructure in this context should be able to support the exchange of information related to the scheduling process, that will include not only scheduling decisions but also measurements required to make them. For the support of these long term requirements, tight synchronization of the transport nodes will be necessary, both in frequency and phase.

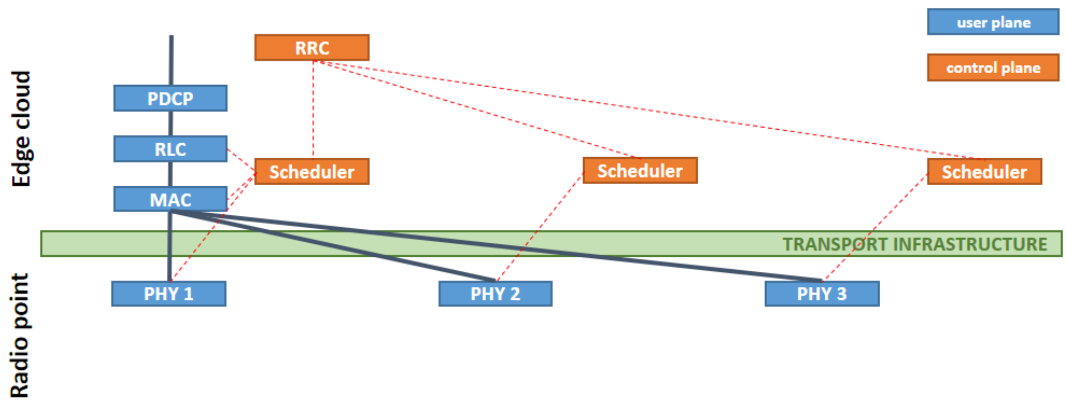

Figure 35: Envisioned 5G multi-connectivity protocol stack.

7.3 Use cases for information exchange between the Mobile and the Transport Networks

Unlike in 4G, where it is assumed that the transport network can be overprovisioned, in 5G the mobile network design needs to consider the performance of the transport network. In particular, the characteristics of the transport network will dictate the optimal allocation of RAN signal processing functions between a Remote Unit (RU) and Centralized Unit (CU) [235], which is critical to implement cooperative and interference mitigation techniques that increase spectral efficiency. Thus, parameters like the transport network available rate and latency are key in the selection of the RAN configuration to be used.

We next present a set of use cases that motivate the need for a mobile network and transport information exchange:

- *Proactive congestion avoidance*: Lack of coordination between the mobile network and transport may result in the RAN triggering a handover to a target base station that is then connected to a congested link in the transport network. In order to avoid these situations, the RAN must be aware of the transport network congestion levels when triggering handovers between cells.
- *Load balancing*: Information about the RAN enables the transport network to more effectively balance the traffic load between base stations and mobile network functions across different paths. This results in a better utilization of resources within the transport, as well as in an overall improvement of QoE.
- *Fairness*: Currently, transport networks do not offer the same granularity in QoS profiles as the RAN provides. Transport networks are therefore unable to distinguish among different types of traffic, which may result in unfairness, or policy violations, upon congestion. A RAN-transport information exchange would for example allow to appropriately re-classify RAN traffic into transport QoS classes in order to comply with the policies defined by the Mobile Network operator at all times.
- *Self-backhauling*: Early 5G deployments require means for incremental deployment as initially the density of 5G base stations with dedicated backhauling would be limited. A useful technique which can be beneficial in future systems is self-backhauling. The support of wireless self-backhauling is a technique studied by some 5G RAN proposals. If such capabilities are available, RAN and transport should coordinate to decide when it is best to make use of self-backhauling.
- *Energy Saving*: If operating in isolation, the mobile network and the transport may take conflicting decisions when trying to minimize energy consumption by independently switching off RAN and transport nodes. Energy efficiency is a clear example where RAN-transport coordination is required for a global system optimization.
- *Mobility Management*: To support the required coordination between the nodes to obtain a seamless service experience clustering of base stations can be used. In this case, the network consists of a set of clusters (e.g. mmWave Access Points, APs) that are able to deal with the network dynamics. These APs can be configured by the network as the UE moves, being one of them connected to the core and, depending on the network context, to decide which of them connects to the core.
- *Cell-less RAN architecture*: Some 5G RAT proposals operating at very high frequencies, where blockage and path loss due to NLoS are very significant, are studying the possibility of not having a mobile device exclusively attached to a single base station, but rather be able to receive/transmit data from/to different base stations according to channel measurements performed by the mobile device. Changing points of attachment at such short time scales requires a very tight coordination between RAN and transport in order to quickly reconfigure the downlink and uplink paths.
- *BH network resource sharing*: resource sharing at the backhaul level can help network operators to recover from network failure or link congestion. Works in this category are targeted at solutions based on SDN and slicing mechanisms to create multiple virtual BH networks and allow one operator to share a portion of its own resources with another operator. In other words, these solutions helps re-direct mobile traffic from the communication links of one network operator to another for the purpose of load sharing in heavy traffic or link failure conditions.

7.4 5G-XHaul Requirements and Initial Design on mobile – transport interaction and mobility awareness

7.4.1 5G-XHaul requirements on mobile-transport interactions

The 5G-XHaul transport solution should address not only requirements associated with the support of mobility procedures as they are currently defined, but also those that may result in the future from the evolution of the 5G networks.

We present in Table 9 the initial set of requirements that must be supported by the 5G-XHaul transport solution in terms of transport and RAN interaction and mobility awareness.

Table 9: 5G-XHaul Mobile-Transport interaction and mobility-related requirements.

Requirement	Summary	Description
R1	Backhaul/RAN awareness	Information about the status of the RAN should be available for the transport network control, and vice versa. This interaction should happen at different time scales depending on the characteristics of the information exchanged.
R2	QoS support in the transport network	The transport network should be able to take into account the QoS policies defined for the RAN/Core in order to configure the transport services provided.
R3	Support of self-backhauling	The transport network should be able to use the RAN spectrum resources for backhaul purposes in case this is the best option available from a tech-nical and economic viewpoints.
R4	Support to dynamic multicast services	Dynamic multicast services should be supported by the transport network in order to provide resilience for the mmWave RAN.
R5	Support of fast RAN-transport network interaction	In order to support seamless mobility associated to a cell-less RAN architecture, the transport network should be able to re-route the user data plane to different base stations in a per transport block ba-sis.
R6	Mobility related information gathering	Access from the Transport Controller to information gathered by other network elements, including probes, which may be used to configure the transport network.

7.4.2 5G-XHaul initial design on mobile-transport interactions and mobility awareness

The goal of the 5G-XHaul is to design/implement a transport network infrastructure capable of enabling 5G Mobile Network operators (MNOs) to efficiently engineer the transport network resources by utilising RAN-related information.Therefore, an important service to be provided by the 5G-XHaul transport solution is the ability (of MNOs) to interact with the mobile network in a more tightly coupled way than what is possible in 4G, in order to improve QoE. This idea would represent a holistic view of the network when talking about scalable control plane design. Figure 36 depicts the 5G-XHaul transport solution as a bridge between the 5G Core Network and the RAN, representing some of the interfaces envisioned within the 5G-PPP community [236] [237].

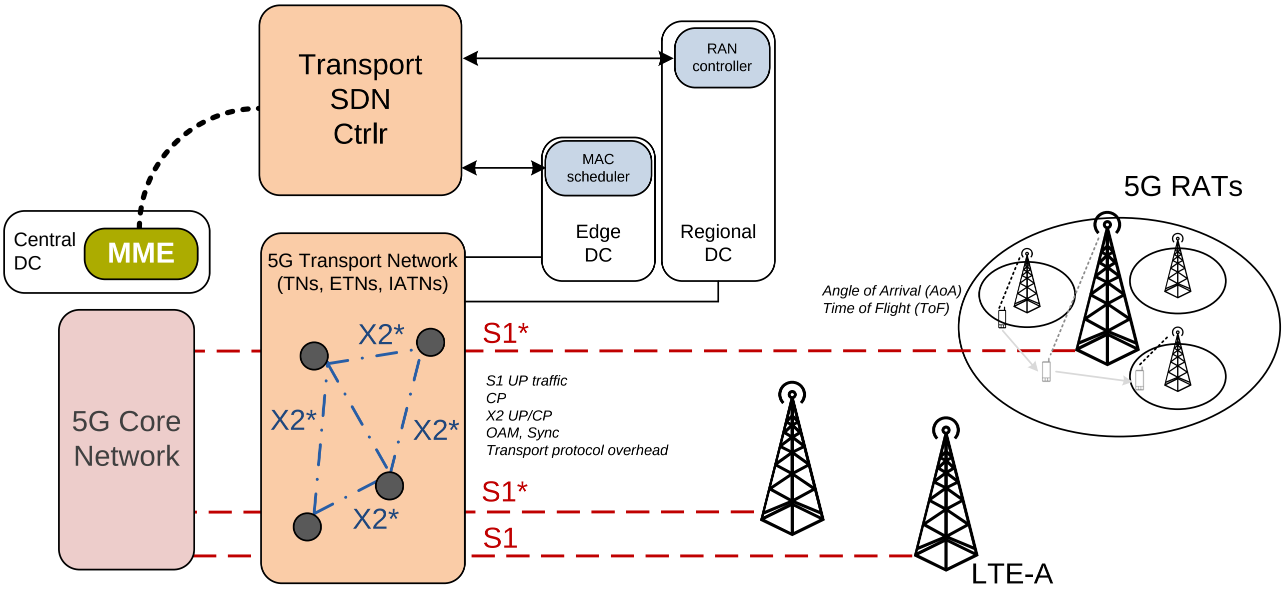

Figure 36: Envisioned interfaces between mobile network and transport.

A core function of the 5G-XHaul transport controller (Figure 36), is the ability to interface with either 4G or 5G RANs in order to estimate spatio-temporal demand variations in the RAN and allocate the required transport resources accordingly. For this purpose, the 5G-XHaul will consider several interfaces that could facilitate the exchange of information (at different timescales) between the transport and RAN networks, such as:

- The interface between the upper tiers of the 5G-XHaul control plane and the MME – or its 5G equivalent – in order to monitor the varying traffic demands at macro level (e.g. by means of anonymised traces, such as the number of users per tracking area, or estimated weekly patterns), assisting thus the planning-related transport network decisions.
- The interface between the RAN controller and the layer-0 controller (c.f. section 2.2.2) -at shorter timescales- that could be used to optimise RAN transport coordination for handover or load balancing purposes.
- The interface between the MAC scheduler[7] and the 5G-XHaul ETN (c.f. section 2.2.2) allowing the MAC scheduler to quickly change the point of attachment of a flow, which is a critical feature in cell-less architectures e.g., in mmWave RANs [234].

The design principles for the identified interfaces depend on the timescale of the available side information i.e., long-term information obtained from observations over long periods of time, such as statistics related to users' requests and average communication times with BSs and other UTs, or short-term information related to instant changes e.g., instantaneous channel state information and real-time location information. It is obvious that the collection of long-term information will lead to a low overhead traffic, while the usage of short-term information can provide better performance/results but requires more frequent updates.

In the following, we provide an initial design for the above-mentioned interfaces and we discuss candidate techniques that will be developed in the project for mobility awareness.

7.4.2.1 Long Term (LT) Mobile-Transport Interface

Long term/Macro mobility measurements from the RAN refer to those that could be used in the transport network to optimize the allocation of the resources (e.g. in the mmWave mesh, switch off ONU transceivers).

[7] In 4G, the MAC scheduler is a function inside the RAN protocol stack in charge of scheduling packets from different radio bearers on the available radio resources. An equivalent meaning is assumed for 5G.

Here, both the Level-0 and Level-1 controllers (c.f. section 2.2.2) play a role at different levels (intra-area and inter-area, respectively). Figure 37 presents the exchange of information between the 5G-XHaul transport controllers and the MME and RAN controllers:

- MME (or 5G equivalent) exchanges data with the Level1-Ctrlr (c.f. section 2.2.2). The data to be shared could be the number of UEs per tracking area. Historical forecasts or predictions on users' mobility can be calculated to anticipate the movement of the users. The timescale for exchanging information among the RAN and the transport would be days, weeks, etc.
- RAN controller exchanges information with the Level0-Ctrlr (c.f. section 2.2.2). The SDN–RAN controller is responsible for controlling and managing the radio resources for the radio access elements.The BSs have to be linked to an ETN, which is controlled by the Level0-Ctrlr. The information exchanged can range from the estimated movement of the UEs, to mobility predictions, to load balancing, etc. The timescale for exchanging such information between the RAN and the transport would be in minutes. An overview of the information (parameters) that could be exchanged over this interface is given in Section 7.4.2.3.

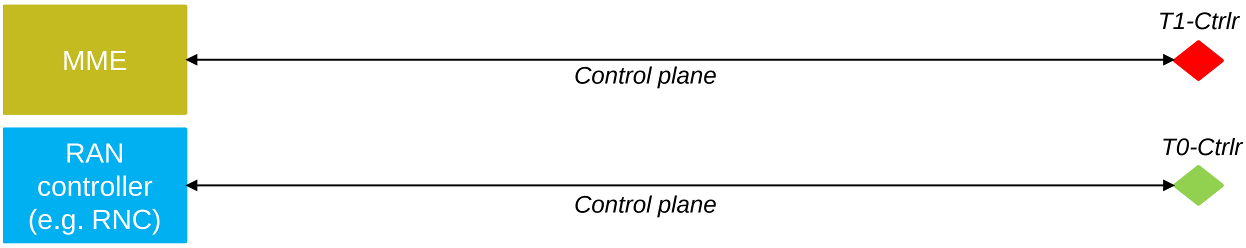

Figure 37: Long Term (LT) RAN-Transport interface.

This LT information incurs a low overhead to then obtain less fine grained information, which may also expire and not reflect the current status after a period of time. This would lead to inaccurate measures.

7.4.2.2 Short Term (ST) Mobile-Transport Interface

In this approach (see Figure 38), the interactions between the RAN and the transport network refer to cell-less architectures, for instance, in mmWave RAN, where the MAC scheduler may decide based on UE measurements to transmit data from a different BS.

The centralized MAC scheduler selects the target BS at resource block level depending on the UE measurements and sends a command to the transport controller so that the target BS is updated accordingly. The MAC scheduler is a function attached to an ETN and locally communicates with its corresponding ETN to update the target endpoint. The timescale for exchanging information among the RAN and the transport is in milliseconds.

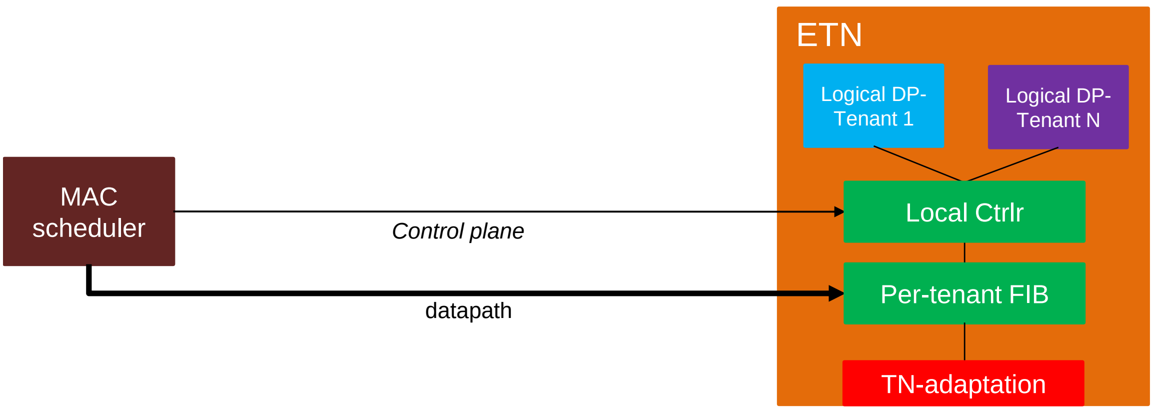

Figure 38: ST Mobile-Transport interface.

7.4.2.3 Mobile network parameters to be mapped to the initial design

Based on the description of the interfaces introduced in the previous sections, a list of candidate parameters that could be collected and processed by the 5G-XHaul (i.e., the transport controller) to undertake the adequate measures and actions required within the transport network. These parameters could include:

1. Connectivity-related info (e.g. list of attachment points, active attachment point).
2. Access network-related info (e.g., cell-id, LAC/TAC, RAT, RSSI/RSRP for the mobile network and SSID, BSSID, RSSI, etc. for the WiFi network).
3. QoS/QoE-related info (e.g., latency, maximum upload and download bitrates that can be achieved at the current location).
4. Terminal-related info (manufacturer, phone model, OS release, CPU/memory/battery info, etc.).
5. Location and timestamp info.

Our aim is to have a reduced/minimum set of parameters given the increase of signalling overhead over the network when sharing all this information. This set of parameters could be sent by the UE periodically or at specific events, e.g., low signal, successful/unsuccessful handover, cell reselection, etc. The number of parameters to be exchanged is not fixed and is dependent on the algorithm that runs at the controller side, whose efficiency impacts the timescale of the exchange of information via the interfaces. This algorithm and its operation, in turn, will be determined by the services to be offered, which might define which parameters to be exchanged. The following parameters could be exchanged via each of the interfaces introduced in 7.4.2.1 and 7.4.2.2:

- **MME – Level1-Ctrlr**
  - Latency (min/max/Average)
  - Latency (%packets lost)
  - Latency (Packet size)
  - (max) download bitrate (measured value)
  - download bitrate (packet size)
  - download bitrate (%packets lost)
  - (max) upload bitrate (measured value)
  - upload bitrate (packet size)
  - upload bitrate (%packets lost)

- **RAN controller – Level0-Ctrlr**
  - The cell-id/LAC each mobile is “connected” to ->
    - (a) #Active Users / cell and
    - (b) #Potential Active Users / Cell (based on “status”)
  - RAT: e.g. (GPRS, EDGE, UMTS, HSPA/+, 4G, 5G)
  - PSC/PCI
  - RSSI
  - RSRP
  - RSRQ
  - RSSNR –SINR
  - CQI
  - Ev/No
  - Timing Advance
  - Neighboring Cells Info
  - List of Bands / Frequency Used

- **MAC scheduler – ETN**
  - QoS data from the Policy and Charging Rules Function (PCRF): minimum guaranteed bandwidth, maximum allowed bandwidth, packet loss rates, relative priority of users, etc.
  - Messages from the UEs regarding the radio channel quality, the strength or weakness of the signal, etc.
  - Measurements from the radio receiver regarding radio channel quality, noise and interference, etc.

- Buffer status from the upper layers about how much data is queued up waiting for transmission.

7.4.2.4 Techniques for user mobility awareness

One of the key elements of the 5G-XHaul control plane is its 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. Given the reduced cell size in future very dense 5G networks, understanding traffic demand variations is considered to be essential for MNOs to optimize the overall network performance. The generation of spatio-temporal demand models though requires the control plane to be able to predict the movements of the mobile users. In 5G-XHaul several alternative designs will be leveraged, each will then require different degrees of integration between the RAN and the transport network.

Apart from the already mentioned interface with the MME (see Figure 36), we may also extract information from the RAN to track the movement of users. This can be carried out via Location-based PHY functionality embedded in RAN nodes like, for example, the Angle of Arrival (AoA) or the Time of Flight (ToF) techniques, which may deliver very accurate position and direction estimates. This last approach enables more precise spatio-temporal modelling with the downside of requiring a tighter integration between the mobile network and the 5G-XHaul transport network. The 5G-XHaul cognitive control plane will thus allow for robust spatio-temporal traffic models able to operate with different granularities of RAN mobility information, while being flexible enough to be integrated with different 5G RANs.

8 SUMMARY AND CONCLUSIONS

This deliverable provides an analysis of the state of the art on scalable control plane issues for 5G networks and defines the requirements of the 5G-XHaul control plane, presenting also an initial design. The proposed design enables the unified control through well-defined SDN southbound interfaces of a transport network infrastructure composed of heterogeneous technology domains in a scalable way, whilst enabling virtualization and slicing, support for QoS and user mobility awareness.

After providing a state of the art review of relevant 5G control architectures, the high-level requirements of the 5G-XHaul control system are defined and an initial view of the control plane architecture is presented together with the description of the functionality of the components. We then lay out the network virtualisation requirements to be fulfilled by the 5G-XHaul control plane and sketch an initial solution design for supporting multitenancy over the 5G-XHaul trasport network that will be further refined in subsequent WP3 deliverables.

Quality of Service (QoS) support in the data plane technologies considered in the project, namely the optical and wireless is also examined. Previous work on traffic engineering, QoS routing and end-to-end QoS across domains is described. The requirements to be fulfilled by the end to end QoS mechanisms defined in 5G-XHaul are also presented together with some initial design of QoS support in mmWave and in the control plane functionality. Moreover, we focus on the issues of SDN southbound intfraces for sub-6GHz technologies, such as LTE, WiFi and we further describe mmWave technologies. We also describe the southbound protocols for optical transport technologies (i.e. TSON and WDM-PON) and we define the requirements and an initial design proposal for the southbound protocol for candidate technologies in 5G-XHaul.

An analysis of the state of the art of the challenges of SDN when applied to large scale, highly dynamic networks is also presented. The requirements to be satisfied by the proposed 5G-XHaul solution are listed and an initial approach of the hierarchical and distributed control plane is described together with techniques such as rule caching, used to tackle issues of scalability and fast response to changing network conditions. Finally, we highlight the current and envisioned state of the art in RAN-transport interaction and user mobility awareness, focusing on the cases which makes this interaction necessary. We also put forward our requirements and views on this topic, drafting an initial design of our planned interaction with 5G RANs.

The work presented in this deliverable extends the high-level control plane architecture also described in D2.2 by providing detailed description of the functionality of the control-related components that will be developed and evaluated during the project. Apart from providing a detailed review of the related work in the areas related with the control plane functionality of the 5G-XHaul system, the requirements and initial design of the proposed solutions have also been presented. This design will be refined in the second year of the project and components of the control plane will be developed and evaluated.

The hierachical SDN control plane, the virtualisation support and multi-tenant operation of the transport network as well as the various SDN southbound interfaces of the data plane technologies considered in the project will be developed, integrated and deployed in the two project testbeds for experimentation and demonstration that will be performed in WP5. At the same time, the proposed algorithms related to QoS-enabled resource management for sub6 and mm-Wave and traffic engineering will also be developed and evaluated through simulations.

The detailed implementation design of the 5G-XHaul control system components and the evaluation results will be documented in D3.2 and D3.3 at the end of the second and third year of the project respectively.

9 REFERENCES

[1] 5GPPP, 5G empowering vertical industries. Online. Available here: https://5g-ppp.eu/wp-content/uploads/2016/02/BROCHURE 5PPP BAT2 PL.pdf

[2] NGMN Alliance, NGMN 5G White Paper, White paper, Feb. 2015.

[3] 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

[4] China Mobile NGFI White paper: China Mobile, ALU, Nokia, ZTE, Broadcom, Intel, White Paper of NGFI (Next Generation Fronthaul Interface) Version 1.0, October 4, 2015

[5] Deliverable D2.1 Requirements Specification and KPIs, 5G-XHaul Project, March 2016.

[6] Deliverable D2.2 System Architecture Definition, 5G-XHaul Project, June 2016.

[7] Zheng Ma, ZhengQuan Zhang, ZhiGuo Ding, PingZhi Fan, HengChao Li, ”Key techniques for 5G wireless communications: network architecture, physical layer, and MAC layer perspectives”, Science China Information Sciences, Springer , April 2015, Volume 58, Issue 4, pp 1-20.

[8] Ian F. Akyildiz, Pu Wang, Shih-Chun Lin, ”SoftAir: A software defined networking architecture for 5G wireless systems”, Computer Netwworkins, Elsevier , 2015.

[9] Guolin Sun, Feng Liu, Junyu Lai, and Guisong Liu, ”Software Defined Wireless Network Architecture for the Next Generation Mobile Communication: Proposal and Initial Prototype”, Journal of Communications , December 2014.

[10] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, ”OpenAirInterface: A Flexible Platform for 5G Research”, ACM SIGCOMM Computer Communication Review, v.44 n.5, October 2014.

[11] H. Hung Cho, C. F Lai, TK Shih, H. C Chao, ”Integration of SDR and SDN for 5G”, IEEE Access, vol. 2, pp. 1196-1204, 2014.

[12] Hyunwoo Nam, Doru Calin, Henning Schulzrinne,”Intelligent Content Delivery over Wireless via SDN”, IEEE WCNC , 2015.

[13] Aditya Gudipati , Daniel Perry , Li Erran Li , Sachin Katti, ”SoftRAN: software defined radio access network”, Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, August 2013.

[14] Rafat Jahan, ”Unlocking the True Potential of 5G: Techniques for Latency Reduction”, White Paper, TCS' Telecom Business Unit, 2015.

[15] Teemu Koponen, Keith Amidon, Peter Balland, Martín Casado, Anupam Chanda, Bryan Fulton, Igor Ganichev, Jesse Gross, Natasha Gude, Paul Ingram, Ethan Jackson, Andrew Lambeth, Romain Lenglet, Shih-Hao Li, Amar Padmanabhan, Justin Pettit, Ben Pfaff, Rajiv Ramanathan, Scott Shenker, Alan Shieh, Jeremy Stribling, Pankaj Thakkar, Dan Wendlandt, Alexander Yip, and Ronghua Zhang, 'Network virtualization in multi-tenant datacenters', in Proceedings of Networked Systems Design and Implementation (NSDI), 2014.

[16] Allan, David, Nigel Bragg, Alan McGuire, and Andy Reid. "Ethernet as carrier transport infrastructure." IEEE Communications Magazine 44, no. 2 (2006): 95-101.

[17] Chandra, Tushar D., Robert Griesemer, and Joshua Redstone. "Paxos made live: an engineering perspective." Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing. ACM, 2007.

[18] T. Bates, et.al., RFC4760-Multiprotocol Extensions for BGP-4, January 2007

[19] Juniper, Understanding BGP Route Reflectors for Simplifying Scaling of BGP Networks. Online. Available: http://www.juniper.net/documentation/en_US/junos14.2/topics/concept/routing-protocol-bgp-security-route-reflector-understanding.html

[20] A. Sajassi, A Network Virtualization Overlay Solution using EVPN, draft-ietf-bess-evpn-overlay-04, June 10, 2016

[21] Al-Fares, M., Loukissas, A., & Vahdat, A. (2008). A scalable, commodity data center network architecture. ACM SIGCOMM Computer Communication Review, 38(4), 63-74.

[22] Ali-Ahmad, Hassan, Claudio Cicconetti, Antonio La Oliva, Martin Draxler, Rajesh Gupta, Vincenzo Mancuso, Laurent Roullet, and Vincenzo Sciancalepore. "CROWD: an SDN approach for DenseNets." In Software Defined Networks (EWSDN), 2013 Second European Workshop on, pp. 25-31. IEEE, 2013.

[23] Draft Standard, P. (1997). 802.1 Q/D10, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks, Copyright by the Institute of Electrical and Electronics Engineers.

[24] Tsirtsis G. Network address translation-protocol translation (NAT-PT). Network. 2000 Feb.

[25] Rosen, E. C., & Rekhter, Y. (2006). BGP/MPLS IP virtual private networks (VPNs).

[26] Sherwood R, Gibb G, Yap KK, Appenzeller G, Casado M, McKeown N, Parulkar G. Flowvisor: A network virtualization layer. OpenFlow Switch Consortium, Tech. Rep. 2009 Oct 14:1-3.

[27] MEF, Ethernet Services Definitions - Phase 2, April 2008, available Online: https://www.mef.net/PDF_Documents/technical-specifications/MEF6-1.pdf

[28] Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P and Sengupta, S. (2009, August). VL2: a scalable and flexible data center network. In ACM SIGCOMM computer communication review (Vol. 39, No. 4, pp. 51-62). ACM.

[29] Mudigonda, J., Yalagandula, P., Mogul, J., Stiekes, B., & Pouffary, Y. (2011, August). NetLord: a scalable multi-tenant network architecture for virtualized datacenters. In ACM SIGCOMM Computer Communication Review (Vol. 41, No. 4, pp. 62-73). ACM.

[30] Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks, RFC 7348

[31] Ballani, H., Costa, P., Karagiannis, T., & Rowstron, A. (2011, August). Towards predictable datacenter networks. In ACM SIGCOMM Computer Communication Review (Vol. 41, No. 4, pp. 242-253). ACM.
[32] Al-Shabibi, A., De Leenheer, M., Gerola, M., Koshibe, A., Parulkar, G., Salvadori, E., & Snow, B. (2014, August). OpenVirteX: Make your virtual SDNs programmable. In Proceedings of the third workshop on Hot topics in software defined networking (pp. 25-30). ACM.
[33] Mudigonda, Jayaram, Praveen Yalagandula, Mohammad Al-Fares, and Jeffrey C. Mogul. "SPAIN: COTS Data-Center Ethernet for Multipathing over Arbitrary Topologies." In NSDI, vol. 10, pp. 18-33. 2010.
[34] Benson, T., Anand, A., Akella, A., & Zhang, M. (2011, December). MicroTE: Fine grained traffic engineering for data centers. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies* (p. 8). ACM.
[35] F. Hao, T. Lakshman, S. Mukherjee, and H. Song, 'Secure Cloud Computing with a Virtualized Network Infrastructure', in Proceedings of USENIX HotCloud, June 2010.
[36] Teemu Koponen, Keith Amidon, Peter Balland, Martín Casado, Anupam Chanda, Bryan Fulton, Igor Ganichev, Jesse Gross, Natasha Gude, Paul Ingram, Ethan Jackson, Andrew Lambeth, Romain Lenglet, Shih-Hao Li, Amar Padmanabhan, Justin Pettit, Ben Pfaff, Rajiv Ramanathan, Scott Shenker, Alan Shieh, Jeremy Stribling, Pankaj Thakkar, Dan Wendlandt, Alexander Yip, and Ronghua Zhang, 'Network virtualization in multi-tenant datacenters', in Proceedings of Networked Systems Design and Implementation (NSDI), 2014.
[37] P. Marques, et.al., BGP-signaled end-system IP/VPNs, draft-ietf-l3vpn-end-system-05, October 8, 2015. Available at: https://tools.ietf.org/html/draft-ietf-l3vpn-end-system-05?_utm_source=1-2-2
[38] C. Hemanth and T.G. Venkatesh, Performance Analysis of Service Periods (SP) of the IEEE 802.11ad Hybrid MAC Protocol, IEEE Transactions on Mobile Computing, 15(5) 2016
[39] IEEE 802.11e, IEEE Standard for Information technology--Local and metropolitan area networks--Specific requirements--Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements," in IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11-1999), Nov. 2005.
[40] Kosek-Szott, Katarzyna, et al. "What's new for QoS in IEEE 802.11?." Network, IEEE 27.6 (2013): 95-104.
[41] IEEE 802.11aa, IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: MAC Enhancements for Robust Audio Video Streaming," in IEEE Std 802.11aa-2012 (Amendment to IEEE Std 802.11-2012 as amended by IEEE Std 802.11ae-2012) , May 2012
[42] Lyakhov, Andrey, et al. "Starvation effect study in IEEE 802.11 mesh networks." Mobile Adhoc and Sensor Systems, 2009. MASS'09. IEEE 6th International Conference on. IEEE, 2009.
[43] IEEE 802.11s, IEEE Standard for Information Technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 10: Mesh Networking," in IEEE Std 802.11s-2011, Sept. 2011
[44] Krasilov, Artem, Andrey Lyakhov, and Alexander Safonov. "Interference, even with MCCA channel access method in IEEE 802.11 s mesh networks." Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on. IEEE, 2011.
[45] Tinnirello, Ilenia, and Pierluigi Gallo. "Supporting a pseudo-TDMA access scheme in mesh wireless networks." Wireless Access Flexibility. Springer Berlin Heidelberg, 2013. 80-92.
[46] Khan, Yahya, et al. "Self-Organizing TDMA MAC Protocol for Effective Capacity Improvement in IEEE 802.11 WLANs." 2015 IEEE Globecom Workshops (GC Wkshps). IEEE, 2015.
[47] Yang, Zhice, et al. "Enabling TDMA for today's wireless LANs." Computer Communications (INFOCOM), 2015 IEEE Conference on. IEEE, 2015.
[48] Mahmood, Arif, Reinhard Exel, and Thilo Sauter. "Delay and jitter characterization for software-based clock synchronization over WLAN using PTP." Industrial Informatics, IEEE Transactions on 10.2 (2014): 1198-1206.
[49] IEEE, 802.1Qbu - Frame Preemption, work in progress, 2016 http://www.ieee802.org/1/pages/802.1bu.html
[50] IEEE, 802.1Qbv - Enhancements for Scheduled Traffic, work in progress, 2016 http://www.ieee802.org/1/pages/802.1bv.html
[51] IEEE, 802.1Qcc - Stream Reservation Protocol (SRP) Enhancements and Performance Improvements, work in progress, 2016 http://www.ieee802.org/1/pages/802.1cc.html
[52] IEEE, 802.1CB - Frame Replication and Elimination for Reliability, work in progress, 2016 http://www.ieee802.org/1/pages/802.1cb.html
[53] IEEE, 802.1Qca - Path Control and Reservation, work in progress, 2015 http://www.ieee802.org/1/pages/802.1ca.html
[54] A. Demers, S. Keshav, S. Shenker, "Analysis and simulation of a fair simulation of a fair queuing algorithm" Proceedings of ACM SIGCOMM, 1989.
[55] Lixia Zhang, "Virtual clock: a new traffic control algorithm for packet-switching networks," Proc. Of ACM SIGCOMM, 1990.
[56] Broadband Forum, Technical Report DSL Forum TR-059, DSL Evolution - Architecture Requirements for the Support of QoS-Enabled IP Services, 2003, https://www.broadband-forum.org/technical/download/TR-059.pdf

[57] Floyd, S., and Jacobson, V., Random Early Detection gateways for Congestion Avoidance V.1 N.4, August 1993, p. 397-413. Abstract.
[58] IEEE, 802.3-IEEE Standard for Ethernet, 2012, http://standards.ieee.org/getieee802/download/802.3-2012.zip
[59] Y. Lee et al., "Traffic Engineering in Next-Generation Optical Networks," IEEE Commun. Surveys & Tutorials, vol. 6, no. 3, 2004, pp. 16–33.
[60] N. Wang, K. H. Ho, G. Pavlou and M. Howarth, "An overview of routing optimization for internet traffic engineering," in IEEE Communications Surveys & Tutorials, vol. 10, no. 1, pp. 36-56, First Quarter 2008.
[61] 5G-XHaul - D2.1 Requirements Specification and KPIs Document, Nov. 2015
[62] T. Gyires, "Traffic Engineering in Fourth Generation Wireless Networks," Sixth International Conference on Networking, 2007. ICN '07., Martinique, 2007, pp. 3-3.
[63] D. Parniewicz, M. Stasiak and P. Zwierzykowski, "Traffic Engineering for Multicast Connections in Multiservice Cellular Networks," in IEEE Transactions on Industrial Informatics, vol. 9, no. 1, pp. 262-270, Feb. 2013.
[64] M. Laner, P. Svoboda and M. Rupp, "Detecting M2M traffic in mobile cellular networks," IWSSIP 2014 Proceedings, Dubrovnik, 2014, pp. 159-162.
[65] C. Müller, M. Putzke and C. Wietfeld, "Traffic engineering analysis of Smart Grid services in cellular networks," Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on, Tainan, 2012, pp. 252-257.
[66] O. Salami, A. Bagula and H. A. Chan, "Traffic engineering framework for inter-working multi-hop wireless networks," 2011 IEEE GLOBECOM Workshops (GC Wkshps), Houston, TX, 2011, pp. 1133-1138.
[67] J. Lessmann, "Resource optimization in realistic mobile backhaul networks," 2015 IEEE International Conference on Communications (ICC), London, 2015, pp. 3861-3866.
[68] Ian F. Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, Wu Chou, A roadmap for traffic engineering in SDN-OpenFlow networks, Computer Networks, Volume 71, 4 October 2014, Pages 1-30, ISSN 1389-1286.
[69] Seppnen, Kari, Jorma Kilpi, and Tapio Suihkom, Integrating WMN based mobile backhaul with SDN control, Mobile Networks and Applications 20.1 (2015): 32-39.
[70] Detti, Andrea, et al. Wireless mesh software defined networks (wmSDN), IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE, 2013.
[71] H. Chiu, K. Yeung, K.-S. Lui, J-CAR: an efficient joint channel assignment and routing protocol for IEEE 802.11-based multi-channel multi-interface mobile ad hoc networks, IEEE Transactions on Wireless Communications, 8 (4) (2009), pp. 1706–1715
[72] L. Khoukhi, S. Cherkaoui, Intelligent QoS management for multimedia services support in wireless mobile ad hoc networks, Computer Networks, 54 (10) (2010), pp. 1692–1706
[73] L. Hanzo-II, R. Tafazolli, QoS-Aware routing and admission control in shadow-fading environments for multi-rate MANETs, IEEE Transactions on Mobile Computing, 10 (5) (2011), pp. 622–637
[74] S. Kajioka, N. Wakamiya, H. Satoh, K. Monden, M. Hayashi, S. Matsui, et al., A QoS-aware routing routing mechanism for multi-channel multi-interface ad-hoc networks, Ad Hoc Networks, 9 (5) (2011), pp. 911–927
[75] T. Liu, W. Liao, Interference-aware QoS routing for multi-rate multi-radio multi-channel IEEE 802.11 wireless mesh networks, IEEE Transactions on Wireless Communications, 8 (1) (2009), pp. 166–175
[76] B. Bakhshi, S. Khorsandi, Complexity and design of QoS routing algorithms in wireless mesh networks, Computer Communications, 34 (14) (2011), pp. 1722–1737
[77] R. Hou, K.-S. Lui, F. Baker, J. Li, Hop-by-hop routing in wireless mesh networks with bandwidth guarantees, IEEE Transactions on Mobile Computing, 11 (2) (2012), pp. 264–277
[78] Yuhuai Peng, Yao Yu, Lei Guo, Dingde Jiang, Qiming Gai, An efficient joint channel assignment and QoS routing protocol for IEEE 802.11 multi-radio multi-channel wireless mesh networks, Journal of Network and Computer Applications, Volume 36, Issue 2, March 2013, Pages 843-857, ISSN 1084-8045, http://dx.doi.org/10.1016/j.jnca.2012.11.003.
[79] F. Akyildiz, X. Wand, W. Wang, Wireless mesh networks: a survey, Computer Networks, 47 (4) (2005), pp. 445–487.
[80] Jian Q., Lin X. Cai , Xuemin S. and Jon W. Mark, STDMA-based Scheduling Algorithm for Concurrent Transmissions in Directional Millimeter Wave Networks, Communications (ICC), 2012 IEEE International Conference on, Ottawa, ON, 2012, pp. 5221-5225.
[81] Hong Li, Lei Huang and Yi Wang, Scheduling Schemes for Interference Suppression in Millimeter-Wave Cellular Network, 2015 IEEE 26th International Symposium on Personal, Indoor and Mobile Radio Communications, pg. 2244-2248, 2015.
[82] H. Shokri-Ghadikolaei, L. Gkatzikis and C. Fischione, Beam-searching and transmission scheduling in millimeter wave communications, Communications (ICC), 2015 IEEE International Conference on, London, 2015, pp. 1292-1297
[83] E. Pateromichelakis, M. Shariat, A. ul Quddus and R. Tafazolli, Joint routing and scheduling in dense small cell networks using 60 GHz backhaul, ommunication Workshop (ICCW), 2015 IEEE International Conference on, London, 2015, pp. 2732-2737.297
[84] Z. He, S. Mao and T. T. S. Rappaport, "On Link Scheduling Under Blockage and Interference in 60-GHz Ad Hoc Networks," in IEEE Access, vol. 3, no. , pp. 1437-1449, 2015.

[85] N. Nandiraju, D. Nandiraju, L. Santhanam, H. Bing, W. Junfang and D.P. Agrawal,, "Wireless Mesh Networks: Current Challenges and Future Directions of Web-In-The-Sky," Wireless Communications, IEEE , vol.14, no.4, pp.79-89, August 2007.
[86] H. Jianhua, Y. Kun, K. Guild and C. Hsiao-Hwa, "Application of IEEE 802.16 Mesh Ne works as the Backhaul of Multihop Cellular Networks," IEEE Communications Magazine, vol.45, no.9, pp.82-90, September 2007
[87] Fang-Chun Kuo, Frank A. Zdarsky, Johannes Lessmann and Stefan Schmid, Cost-Efficient Wireless Mobile Backhaul Topologies: An Analytical Study, Globecom 2010
[88] August Betzler, Ferran Quer, Daniel Camps-Mur, Ilker Demirkol, Eduard Garcia-Villegas, "On the Benefits of Wireless SDN in Networks of Constrained Edge Devices", (to appear in) EuCNC 2016, Athens, Greece, June 2016
[89] Ergin, Mesut Ali, et al. "Available bandwidth estimation and admission control for QoS routing in wireless mesh networks." Computer Communications 31.7 (2008): 1301-1317.
[90] Babiarz, J., Chan, K., & Baker, F. (2006). Configuration guidelines for DiffServ service classes (No. RFC 4594).
[91] Minei, Ina. "MPLS DiffServ-aware traffic engineering." Juniper Networks (2004).
[92] V. Fineberg, "A practical architecture for implementing end-to-end QoS in an IP network," in IEEE Communications Magazine, vol. 40, no. 1, pp. 122-130, Jan 2002.
[93] H. Ekstrom, "QoS control in the 3GPP evolved packet system," in IEEE Communications Magazine, vol. 47, no. 2, pp. 76-83, February 2009.
[94] Prior, R., & Sargento, S. (2006, June). Towards inter-domain QoS control. InComputers and Communications, 2006. ISCC'06. Proceedings. 11th IEEE Symposium on (pp. 731-739). IEEE.
[95] Francisco, Mark Joseph, Lambros Pezoulas, Changcheng Huang, and I. Larnbadaris. "End-to-end signaling and routing for optical IP networks." In Communications, 2002. ICC 2002. IEEE International Conference on, vol. 5, pp. 2870-2875. IEEE, 2002.
[96] A. Farrel, et.al., A Path Computation Element (PCE)-Based Architecture, RFC4655, August 2006.
[97] L. Gommans et al., "Applications drive secure lightpath creation across heterogeneous domains," in IEEE Communications Magazine, vol. 44, no. 3, pp. 100-106, March 2006.
[98] G.7715/Y.1706, "Architecture and Requirements for Routing in ASON", ITU-T Recommendation Draft.
[99] Matsuura, Hiroshi, Tatsuro Murakami, and Kazumasa Takami. "An interdomain path computation server for GMPLS networks." IEICE transactions on communications 88, no. 8 (2005): 3329-3342.
[100] Dasgupta, Sukrit, Jaudelice C. De Oliveira, and Jean-Philippe Vasseur. "Path-computation-element-based architecture for interdomain MPLS/GMPLS traffic engineering: overview and performance." Network, IEEE 21.4 (2007): 38-45.
[101] B. Rahimzadeh Rofoee, G. Zervas, Y. Yan and D. Simeonidou, "Griffin: Programmable Optical DataCenter With SDN Enabled Function Planning and Virtualisation," in Journal of Lightwave Technology, vol. 33, no. 24, pp. 5164-5177, Dec.15, 15 2015.
[102] B. R. Rofoee et al., "First demonstration of service-differentiated converged optical sub-wavelength and LTE/WiFi networks over GEAN," Optical Fiber Communications Conference and Exhibition (OFC), 2015, Los Angeles, CA, 2015, pp. 1-3.
[103] B. R. Rofoee et al., "Hardware virtualized flexible network for wireless data-center optical interconnects [invited]," in IEEE/OSA Journal of Optical Communications and Networking, vol. 7, no. 3, pp. A526-A536, March 2015.
[104] Aditya Gudipati, Daniel Perry, Li Erran Li, and Sachin Katti. 2013. SoftRAN: software defined radio access network. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking (HotSDN '13). ACM, New York, NY, USA, 25-30.
[105] Manu Bansal, Jeffrey Mehlman, Sachin Katti, and Philip Levis. 2012. OpenRadio: a programmable wireless dataplane. In Proceedings of the first workshop on Hot topics in software defined networks (HotSDN '12). ACM, New York, NY, USA, 109-114.
[106] Mao Yang, Yong Li, Depeng Jin, Li Su, Shaowu Ma, and Lieguang Zeng. 2013. OpenRAN: a software-defined ran architecture via virtualization. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM '13). ACM, New York, NY, USA, 549-550.
[107] Huawei SoftCOM. Reshaping the Future of Network Architecture, 2013
[108] Lalith Suresh, Julius Schulz-Zander, Ruben Merz, Anja Feldmann, and Teresa Vazao. 2012. Towards programmable enterprise WLANS with Odin. In Proceedings of the first workshop on Hot topics in software defined networks (HotSDN '12). ACM, New York, NY, USA, 115-120.
[109] CAPWAP protocol binding for IEEE 802.11.http://www.ietf.org/rfc/rfc5416.txt, March 2012.
[110] IEEE 802.11ad standard draft D0.1, [Available]: www.ieee802.org/11/Reports/tgad update.htm.
[111] Swarun Kumar, Diego Cifuentes, Shyamnath Gollakota, and Dina Katabi. 2013. Bringing cross-layer MIMO to today's wireless LANs. In Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM (SIGCOMM '13). ACM, New York, NY, USA, 387-398.
[112] K. Choumas, N. Makris, T. Korakis, L. Tassiulas and M. Ott, "Testbed Innovations for Experimenting with Wired and Wireless Software Defined Networks", proceedings of CNERT 2015 workshop of IEEE ICDCS 2015, Columbus, Ohio, USA, 29 June - 2 July 2015.
[113] Gautam Bhanage, Ivan Seskar, Dipankar Raychaudhuri, "A Service Oriented Experimentation Framework for Virtualized WiMAX Systems", Volume 90 of the series Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering pp 152-161

[114] Gautam Bhanage, Ivan Seskar, Rajesh Mahindra, and Dipankar Raychaudhuri. 2010. Virtual basestation: architecture for an open shared WiMAX framework. In Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures (VISA '10). ACM, New York, NY, USA, 1-8.
[115] Kok-Kiong Yap, Masayoshi Kobayashi, Rob Sherwood, Te-Yuan Huang, Michael Chan, Nikhil Handigol, and Nick McKeown. 2010. OpenRoads: empowering research in mobile networks. SIGCOMM Comput. Commun. Rev. 40, 1 (January 2010), 125-126.
[116] Fortuna, C., Ruckebusch, P., Van Praet, C., Moerman, I., Kaminski, N., DaSilva, L., Tinirello, I., Bianchi, G., Gringoli, F., Zubow, A., Chwalisz, M., Wolisz, A., Leblon, R., Seskar, I., Choi, S., de Rezende, J. F. (2015). Wireless Software and Hardware platforms for Flexible and Unified radio and network controL. EuCNC Workshop on 5G Testbeds and Hands-on Experimental Research.
[117] N. Makris, Ch. Zarafetas, S. Kechagias, T. Korakis, I. Seskar and L. Tassiulas, "Enabling Open Access to LTE network components; the NITOS testbed paradigm, in the proceedings of IEEE Soft5G, London, 17 April 2015.
[118] http://www.interdigital.com/presentations/edgehaul-millimeter-wave-small-cell-backhaul-system
[119] http://www.interdigital.com/presentations/mwc16-edgehaul
[120] "A service-based model for the hybrid software defined wireless mesh backhaul of small cells", J. Núñez-Martínez et al., 11th International Conference on Network and Service Management (CNSM), 2015
[121] Mayur Channegowda, Reza Nejabati, and Dimitra Simeonidou, "Software-Defined Optical Networks Technology and Infrastructure: Enabling Software-Defined Optical Network Operations [Invited]," J. Opt. Commun. Netw. 5, A274-A282 (2013)
[122] S. Gringeri, N. Bitar and T. J. Xia, "Extending software defined network principles to include optical transport," in IEEE Communications Magazine, vol. 51, no. 3, pp. 32-40, March 2013.
[123] "C-RAN The Road Towards Green RAN", China Mobile Research Institute: White Paper, 2011
[124] http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf
[125] https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.0.pdf
[126] Yalin Liu and Artur Hecker and Riccardo Guerzoni and Zoran Despotovic and Sergio Beker, "On optimal hierarchical SDN", IEEE ICC, 2015.
[127] Tootoonchian, Amin and Ganjali, Yashar, "HyperFlow: A Distributed Control Plane for OpenFlow", Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, 2010.
[128] Koponen, Teemu and Casado, Martin and Gude, Natasha and Stribling, Jeremy and Poutievski, Leon and Zhu, Min and Ramanathan, Rajiv and Iwata, Yuichiro and Inoue, Hiroaki and Hama, Takayuki and Shenker, Scott, "Onix: A Distributed Control Platform for Large-scale Production Networks", Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, OSDI, 2010.
[129] Berde, Pankaj and Gerola, Matteo and Hart, Jonathan and Higuchi, Yuta and Kobayashi, Masayoshi and Koide, Toshio and Lantz, Bob and O'Connor, Brian and Radoslavov, Pavlin and Snow, William and Parulkar, Guru," ONOS: Towards an Open, Distributed SDN OS", Proceedings of the Third Workshop on Hot Topics in Software Defined Networking (HotSDN), 2014.
[130] Heller, Brandon and Sherwood, Rob and McKeown, Nick, "The controller placement problem", Proceedings of the first workshop on Hot topics in software defined networks (HotSDN), 2012.
[131] Hock, David and Hartmann, Matthias and Gebert, Steffen and Jarschel, Michael and Zinner, Thomas and Tran-Gia, Phuoc, "Pareto-optimal resilient controller placement in SDN-based core networks", Teletraffic Congress (ITC), 2013.
[132] Muller, Lucas F and Oliveira, Rodrigo R and Luizelli, Marcelo C and Gaspary, Luciano P and Barcellos, Marinho P, "Survivor: an enhanced controller placement strategy for improving SDN survivability", IEEE GLOBECOM, 2014.
[133] Sallahi, Afrim and St-Hilaire, Marc, "Optimal Model for the Controller Placement Problem in Software Defined Networks", IEEE Communications Letters, 2015
[134] Y. Fu, J. Bi, Z. Chen, K. Gao, B. Zhang, G. Chen, and J. Wu, "A Hybrid Hierarchical Control Plane for Flow-Based Large-Scale Software-Defined Networks," Network and Service Management, IEEE Transactions on, vol. 12, no. 2, pp. 117–131, June 2015.
[135] A. Y. Ding, J. Crowcroft, and S. Tarkoma, "Poster: SoftOffload: A Programmable Approach Toward Collaborative Mobile Traffic Offloading," in Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, ser. MobiSys '14. New York, NY, USA: ACM, 2014, pp. 368–368.
[136] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," SIGCOMM Comput. Commun. Rev., vol. 44, no. 3, pp. 87–95, Jul. 2014.
[137] "The P4 specification version 1.0.2," March 2015. [Online]. Available: http://p4.org/spec
[138] "Open vSwitch manual." [Online]. Available: http://openvswitch.org/support/dist-docs/ovs-ofctl.8.txt
[139] G. Bianchi, M. Bonola, A. Capone, and C. Cascone, "OpenState: Programming Platform-independent Stateful OpenFlow Applications Inside the Switch," ACM SIGCOMM
[140] T. Husfeldt and N. Taslaman, "The exponential time complexity of computing the probability that a graph is connected," in Parameterized and Exact Computation, Springer, 2010, pp. 192-203.
[141] P. Kolman and O. Pangr{\'a}c, "On the complexity of paths avoiding forbidden pairs," Discrete Applied Mathematics, vol. 157, no. 13, pp. 2871-2876, 2009.

[142] J. Moy, "Open Shortest Path First Routing Protocol (Version 2)," 1994.
[143] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," 1995.
[144] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in ACM SIGCOMM computer communication review, 1994.
[145] C. Perkins, E. Belding-Royer and S. Das, "Ad hoc on-demand distance vector (AODV) routing," 2003.
[146] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in Mobile computing, Springer, 1996, pp. 153-181.
[147] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, A scalable content-addressable network, vol. 31, ACM, 2001.
[148] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in Middleware 2001, 2001.
[149] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in Peer-to-Peer Systems, Springer, 2002, pp. 53-65.
[150] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," ACM SIGCOMM Computer Communication Review, vol. 31, no. 4, pp. 149-160, 2001.
[151] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea and A. Rowstron, "Virtual ring routing: network routing inspired by DHTs," ACM SIGCOMM Computer Communication Review, vol. 36, no. 4, pp. 351-362, 2006.
[152] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," Selected Areas in Communications, IEEE Journal on, vol. 17, no. 8, pp. 1369-1379, 1999.
[153] C. Cramer, T. Fuhrmann and others, Self-stabilizing ring networks on connected graphs, Universitat Karlsruhe, Fakultat fur Informatik, 2005.
[154] T. Fuhrmann, P. Di, K. Kutzner and C. Cramer, Pushing chord into the underlay: Scalable routing for hybrid manets, Universitat Karlsruhe, Fakultat fur Informatik, 2006.
[155] S. Jain, Y. Chen and Z.-L. Zhang, "Viro: A scalable, robust and namespace independent virtual id routing for future networks," in INFOCOM, 2011 Proceedings IEEE, 2011.
[156] J. Carlson, "PPP Transparent Interconnection of Lots of Links (TRILL) Protocol Control Protocol," 2011.
[157] R. Jacob, A. Richa, C. Scheideler, S. Schmid and H. Taubig, "A distributed polylogarithmic time algorithm for self-stabilizing skip graphs," in Proceedings of the 28th ACM symposium on Principles of distributed computing, 2009.
[158] R. Jacob, S. Ritscher, C. Scheideler and S. Schmid, "A self-stabilizing and local delaunay graph construction," in Algorithms and Computation, Springer, 2009, pp. 771-780.
[159] S. Kniesburges, A. Koutsopoulos and C. Scheideler, "Re-chord: a self-stabilizing chord overlay network," in Proceedings of the twenty-third annual ACM symposium on Parallelism in algorithms and architectures, 2011.
[160] M. Onus, A. W. Richa and C. Scheideler, "Linearization: Locally Self-Stabilizing Sorting in Graphs.," in ALENEX, 2007.
[161] Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff, Martín Casado, Nick McKeown and Scott Shenker, 'NOX: towards an operating system for networks', in Proceedings of ACM SIGCOMM, 2008.
[162] L. V. Morales, A. F. Murillo and S. J. Rueda, 'Extending the Floodlight Controller', in Proceedings of Network Computing and Applications (NCA), 2015.
[163] https://www.amqp.org
[164] Jeremy Stribling, Yair Sovran, Irene Zhang, Xavid Pretzer, Jinyang Li, M. Frans Kaashoek, and Robert Morris, 'Flexible, wide-area storage for distributed systems with WheelFS', in Proceedings of Networked Systems Design and Implementation (NSDI), 2009.
[165] S. Martin, L. Chiarello and G. Leduc, 'DISco: A distributed information store for network challenges and their outcome', in Proceedings of IEEE Network Operations and Management Symposium, 2012.
[166] Eugen Borcoci, Mihai Constantinescu and Marius Vochin, 'Multi-controller Scalability in Multi-domain Content Aware Networks Management', in Proceedings of ICNS, 2014.
[167] Soheil Hassas Yeganeh and Yashar Ganjali, 'Kandoo: a framework for efficient and scalable offloading of control applications', in Proceedings of HotSDN, 2012.
[168] Teemu Koponen, Keith Amidon, Peter Balland, Martín Casado, Anupam Chanda, Bryan Fulton, Igor Ganichev, Jesse Gross, Natasha Gude, Paul Ingram, Ethan Jackson, Andrew Lambeth, Romain Lenglet, Shih-Hao Li, Amar Padmanabhan, Justin Pettit, Ben Pfaff, Rajiv Ramanathan, Scott Shenker, Alan Shieh, Jeremy Stribling, Pankaj Thakkar, Dan Wendlandt, Alexander Yip, and Ronghua Zhang, 'Network virtualization in multi-tenant datacenters', in Proceedings of Networked Systems Design and Implementation (NSDI), 2014
[169] Zhongjin Liu, Yong Li, Li Su, Depeng Jin, and Lieguang Zeng, 'M2cloud: software defined multi-site data center network control framework for multi-tenant', in Proceedings of ACM SIGCOMM, 2013.
[170] J. McCauley A. Panda, M. Casado, T. Koponen, and S. Shenker, 'Extending SDN to Large-Scale Networks', in Open Network Summit, 2013.
[171] B. Genge and P. Haller, 'A Hierarchical Control Plane for Software-Defined Networks-based Industrial Control Systems', in Proceedings of IEEE/IFIP Networking, 2016.
[172] Naga Katta, Omid Alipourfard, Jennifer Rexford and David Walker, 'Rule-Caching Algorithms for Software-Defined Networks', in Proceedings of HotSDN, 2014.
[173] Martin Casado, Michael J. Freedman, Justin Pettit, Jianying Luo, Nick McKeown and Scott Shenker, 'Ethane: Taking Control of the Enterprise', in Proceedings of SIGCOMM, 2007.

[174] Bo Yan, Yang Xu, Hongya Xing, Kang Xi and H. Jonathan Chao, 'CAB: A Reactive Wildcard Rule Caching System for Software-Defined Networks', in Proceedings of HotSDN, 2014.
[175] Nick Shelly, Ethan J. Jackson, Teemu Koponen, Nick McKeown and Jarno Rajahalme, 'Flow caching for high entropy packets', in Proceedings of HotSDN, 2014.
[176] H. Huang, S. Guo, P. Li, W. Liang and A. Y. Zomaya, 'Cost Minimization for Rule Caching in Software Defined Networking', in IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 4, pp. 1007-1016, April 2016.
[177] Minlan Yu, Jennifer Rexford, Michael J. Freedman, and Jia Wang, 'Scalable Flow-Based Networking with DIFANE', in Proceedings of SIGCOMM, 2010.
[178] M. Dong, H. Li, K. Ota and J. Xiao, 'Rule Caching in SDN-Enabled Mobile Access Networks', in IEEE Network, vol. 29, no. 4, pp. 40-45, July-August 2015
[179] “OpenDaylight”, June 2016. [Online]. Available: https://www.opendaylight.org
[180] Vinoski, Steve. "Advanced message queuing protocol." IEEE Internet Computing 6 (2006): 87-89.
[181] Streaming Text Oriented Messaging Protocol, available at http://stomp.github.io/
[182] Hunkeler, Urs, Hong Linh Truong, and Andy Stanford-Clark. "MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks." Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on. IEEE, 2008.
[183] Saint-Andre, Peter, Kevin Smith, and Remko Tronçon. XMPP: the definitive guide. " O'Reilly Media, Inc.", 2009.
[184] Pradeeban Kathiravelu, Messaging4Transport – Message Oriented Middleware Bindings for MD-SAL, available at: https://wiki.opendaylight.org/images/9/90/Messaging4Transport.pdf
[185] ONF, North Bound Interface Working Group (NBI-WG) Charter, v1.1, June, 2013. Available online: https://www.opennetworking.org/images/stories/downloads/working-groups/charter-nbi.pdf
[186] ODL Network Intent Composition: Use Cases, available online: https://wiki.opendaylight.org/view/Network_Intent_Composition:Use_Cases
[187] D. Bainbridge, What’s the Intent. Available online: http://www.slideshare.net/opendaylight/whats-the-intent
[188] ONOS Intent framework. Available online: https://wiki.onosproject.org/display/ONOS/Intent+Framework
[189] R. Alimi, et.al., RFC7285: Application-Layer Traffic Optimization (ALTO) Protocol, September 2014.
[190] Sefraoui, Omar, Mohammed Aissaoui, and Mohsine Eleuldj. "OpenStack: toward an open-source solution for cloud computing." International Journal of Computer Applications 55.3 (2012).
[191] Openstack Neutron. Available online: https://wiki.openstack.org/wiki/Neutron
[192] NEC, OpenDayLight Virtual Tenant Network (VTN). Available online: https://wiki.opendaylight.org/images/0/0e/NEC_VTN_Model_0606.pdf
[193] Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T. and Shenker, S., 2010, October. Onix: A Distributed Control Platform for Large-scale Production Networks. In OSDI (Vol. 10, pp. 1-6).
[194] Foster, Nate, Rob Harrison, Michael J. Freedman, Christopher Monsanto, Jennifer Rexford, Alec Story, and David Walker. "Frenetic: A network programming language." In ACM SIGPLAN Notices, vol. 46, no. 9, pp. 279-291. ACM, 2011. [PRTC] Pyretic
[195] Anderson, Carolyn Jane, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. "NetKAT: Semantic foundations for networks." ACM SIGPLAN Notices 49, no. 1 (2014): 113-126.
[196] Kreutz, Diego, Fernando MV Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103, no. 1 (2015): 14-76.
[197] Radio access and transport network interaction – a concept for improving QoE and resource utilization, Ericcson Review, 2015.
[198] P. Blasco and M. Dohler, “The ETSI BRAN 1 Gbps/km2 cost-efficient cognitive communications architecture,'' in Proc. 3rd Int.Workshop Cognit. Inf. Process. (CIP), May 2012, pp. 1_6.
[199] C. J. Bernardos, A. De Domenico, J. Ortin, P. Rost, and D. Wubben, ``Challenges of designing jointly the backhaul and radio access network in a cloud-based mobile network,'' in Proc. Future Netw. Mobile Summit (FutureNetworkSummit), Jul. 2013, pp. 1_10.
[200] E. Pateromichelakis, A. Maeder, A. De Domenico, R. Fritzsche, P. de Kerret, and J. Bartelt, ``Joint RAN/backhaul optimization in centralized 5G RAN,'' in Proc. Eur. Conf. Netw. Commun. (EuCNC), Jun. 2015, pp. 386_390.
[201] K. M. SaidulHuq, S. Mumtaz, J. Bachmatiuk, J. Rodriguez, X. Wang, and R. L. Aguiar, ``Green HetNet CoMP: Energy ef_ciency analysis and optimization,'' IEEE Trans. Veh. Technol., vol. 64, no. 10, pp. 4670_4683, Oct. 2015.
[202] K. M. S. Huq, S. Mumtaz, J. Rodriguez, and C.Verikoukis, “Investigation on energy ef_ciency in HetNet CoMP architecture,'' in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2014, pp. 1112-1117.
[203] A. Abboud, E. Bastug, K. Hamidouche, and M. Debbah. (2015). Distributed Caching in 5G Networks: An Alternating Direction Method of Multipliers Approach. [Online]. Available: http://laneas.com/
[204] M. Jaber, M. Imran, R. Tafazolli, and A. Tukmanov, ``An adaptive backhaul-aware cell range extension approach,'' in Proc. IEEE Int. Conf. Commun. Workshops (ICC), London, U.K., Jun. 2015, pp. 74_79.
[205] G. Zhang, T. Q. S. Quek, A. Huang, M. Kountouris, and H. Shan, “Backhaul-aware base station association in two-tier heterogeneous cellular networks,'' in Proc. IEEE 16th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC), Jun./Jul. 2015, pp. 390_394.
[206] H. Beyranvand, W. Lim, M. Maier, C. Verikoukis, and J. Salehi, “Backhaul-aware user association in FiWi

enhanced LTE-A heterogeneous networks,'' IEEE Trans. Wireless Commun., vol. 14, no. 6, pp. 2992_3003, Jun. 2015.
[207] ICT-671680 METIS-II, Deliverable 5.1, "Draft Synchronous Control Functions and Resource Abstraction Considerations", May 2016.
[208] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, "LTE Mobile Network Virtualization," Mobile Networks and Applications, vol. 16, no. 4, pp. 424-432, Aug. 2011.
[209] D. Cecarelli and Y. Lee. (2015, Jun.). Framework for abstraction and control of transport networks. Internet Eng. Task Force, Freemont, CA, USA. IETF draft. [Online].Available: https://tools.ietf.org/html/draft-ceccarelliteas- actn-framework-00
[210] Rostami, A., Wang, K., Ghebretensae, Z., Öhlen, P., Skubic, B.: First experimental demonstration of orchestration of optical transport, RAN and cloud based on SDN. In: Proceedings of OFC (2015).
[211] P. Öhlén et al., "Data plane and control architectures for 5G transport networks," Optical Communication (ECOC), 2015 European Conference on, Valencia, 2015, pp. 1-3.
[212] N. Wang, E. Hossain, and V. K. Bhargava, ``Backhauling 5G small cells: A radio resource management perspective,'' IEEE Wireless Commun., vol. 22, no. 5, pp. 41_49, Oct. 2015.
[213] A. F. Cattoni et al., "Ethernet-based mobility architecture for 5G," Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on, Luxembourg, 2014, pp. 449-454.
[214] H. Ge and Z. Lu, "A History-Based Handover Prediction for LTE Systems," in proceeding of 4th International Conference on Ubi-Media Computing (U-Media), 2009, pp. 1–4.
[215] J. Wang and B. Prabhala, "Periodicity based Next Place Prediction," Nokia Mobile Data Challenge 2012 Workshop. p. Dedicated task, vol. 2, no. 2, 2012.
[216] Y. Chon, H. Shin, E. Talipov, and H. Cha, "Evaluating mobility models for temporal prediction with high-granularity mobility data," in Pervasive Computing and Communications (PerCom). IEEE, 2012.
[217] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating next-cell predictors with extensive wi-fi mobility data," Mobile Computing, IEEE Transactions on, vol. 5, no. 12, pp. 1633–1649, 2006.
[218] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "Nextplace: a spatio-temporal prediction framework for pervasive systems," in Pervasive Computing. Springer, 2011, pp. 152–169.
[219] P. Jacquet, W. Szpankowski, and I. Apostol, "A universal predictor based on pattern matching," Information Theory, IEEE Transactions on, vol. 48, no. 6, pp. 1462–1472, 2002.
[220] A. Ulvan, M. Ulvan, and R. Bestak, "The Enhancement of Handover Strategy by Mobility Prediction in Broadband Wireless Access," in Proceedings of Networking and Electronic Commerce Research Conference (NAEC 2009), 2009.
[221] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Next Place Prediction using Mobility Markov Chains," in Proceedings of the First Workshop on Measurement, Privacy, and Mobility, 2012, pp. 1–6.
[222] M. Win, A. Conti, S. Mazuelas, Y. Shen, W. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," IEEE Commun. Mag., vol. 49, no. 5, pp. 56–62, May 2011.
[223] A. Sadiq and K. Bakar, "Mobility and Signal Strength-Aware Handover Decision in Mobile IPv6 based Wireless LAN," in Proceedings of the International MultiConference of Engineers and Computer Scientists, 2011, vol. I.
[224] T. Kim and J. Kim, "Handover Optimization with User Mobility Prediction for Femtocell-based Wireless Networks," International Journal of Engineering and Technology, vol. 5, no. 2, pp. 1829–1837, 2013.
[225] D. Inserra, A. M. Tonello and N. Moret, "Positioning Based on 2-D Angle of Arrival Estimation," Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd, Yokohama, 2011, pp. 1-5.
[226] 3GPP, Evolved Universal Terrestrial Radio Access (E-UTRA); Mobility Enhancements in Heterogeneous Networks. Technical Report 3GPP TR 36.839 V11.1.0 (Oct. 2012).
[227] Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2.
[228] R Wang et al. ,"Mobility-Aware Caching for Content-Centric Wireless Networks: Modeling and Methodology", to appear in IEEE Communications Magazine.
[229] X. Jin et al., "SoftCell: scalable and flexible cellular core network architecture," in Proc. ACM CoN-EXT'13, Dec. 2013, pp. 163–174.
[230] P. Gurusanthosh et al., "SDMA: A semi-distributed mobility anchoring in LTE networks," in Proc. IEEE MoWNeT'13, Sep. 2013, pp. 133–139.
[231] J. Kempf et al., "Moving the mobile evolved packet core to the cloud," in Proc. IEEE WiMob'12, Oct. 2012, pp. 784–791.
[232] G. Hampel et al., "Applying software-defined networking to the telecom domain," in Proc. IEEE INFOCOM'13, Apr. 2013, pp. 133–138.
[233] S. Said et al., "New control plane in 3GPP LTE/EPC architecture for on-demand connectivity ser-vice," in Proc. CloudNet'13, Nov. 2013.
[234] R. J. Weiler, M. Peter, W. Keusgen, E. C. Strinati, A. De Domenico, I. Filippini, A. Capone, I. Siaud, A.M. Ulmer-Moll, A. Maltsev and T. Haustein, "Enabling 5G backhaul and access with millimeter-waves", In EuCNC 2014, Bologna, Italy, June 2014 (pp. 1-5).
[235] 5G-XHaul Project, Deliverable D2.1 "Requirements Specification and KPIs Document", submitted on March 1st, 2016.
[236] METIS II White Paper on 5G RAN Architecture and Functional Design.

[237] mmMAGIC D3.1. Initial concepts on 5G architecture and integration, 2016.
[238] Ren-Huang Liou; Yi-Bing Lin; Shang-Chih Tsai, "An Investigation on LTE Mobility Management," IEEE Transactions on Mobile Computing, vol.12, no.1, pp.166,176, Jan. 2013.

10 ACRONYMS

Acronym	Description
3GPP	Third Generation Partnership Project
5G	Fifth Generation Networks
5G-PPP	5G Infrastructure Public Private Partnership
ADC	Analogue-to-Digital Converter
API	Application Program Interface
ARQ	Automatic Repeat Request
BER	Bit Error Rate
BB	Baseband
BBU	Baseband Unit
BH	Backhaul
BS	Base Station
CAPEX	Capital Expenditures
CBR	Constant Bit Rate
CN	Core Network
CP	Cyclic Prefix
CPRI	Common Public Radio Interface
C-RAN	Cloud Radio Access Network (aka Cloud-RAN)
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DAC	Digital-to-Analogue Converter
DC	Data Centre
DL	Downlink
e2e	end-to-end
eNB	Evolved Node B
EPC	Evolved Packet Core
FCS	Frame Check Sequence
FEC	Forward Error Correction
FH	Fronthaul
ICN	Information-Centric Networking
ILP	Integer Linear Programming
ISP	Internet Service Provider
IT	Information Technology
ITS	Intelligent Transport Services
ITU	International Telecommunication Union

KPI	Key Performance Indicator
LLC	Logical Link Control
LLR	Log-Likelihood Ratios
LoS	Line-of-Sight
LTE	Long Term Evolution
LUT	Lookup Table
MAC	Medium Access Control
MANET	mobile ad hoc networks
MBB	Mobile Broadband
MCS	Modulation and Coding Scheme
MIMO	Multiple-Input Multiple-Output
MME	Mobility Management Entity
mmWave	Millimetre Wave
MNO	Mobile Network Operator
MPLS	Multiprotocol Label Switching
MTC	Machine-Type-Communications
MVNO	Mobile Virtual Network Operator
MWFP	Minimum Weighted Flow Provisioning
NBI	North-Bound Interface
NFV	Network Function Virtualisation
NGFI	Next Generation Fronthaul Interface
NGMN	Next Generation Mobile Networks
NLoS	Non-Line-of-Sight
NOMA	Non-Orthogonal Multiple Access
OBSAI	Open Base Station Architecture Initiative
OLAP	on-line analytical processing
OLT	Optical Line Terminal
OLTP	on-line transaction processing
ONU	Optical Network Unit
ORI	Open Radio Interface
OS	Operating System
OTN	Optical Transport Network
P2P	Point-to-Point
PBB	Provider Backbone Bridge
PGW	Packet Data Network Gateway
PGW-C	packet data network gateway control plane
PGW-D	packet data network gateway data plane

PDCP	Packet Data Convergence Protocol
PON	Passive Optical Network
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
RB	Resource Block
RF	Radio Frequency
RLC	Radio Link Control
RRH	Remote Radio Head
RU	Remote Unit
SDN	Software Defined Networking
SDO	Standards Developing Organisations
S-GW	Service Gateway
SGW-C	Service Gateway Control plane
SINR	Signal to Interference plus Noise Ratio
SLA	Service Level Agreement
SLAE	Sub-Wavelength Lambda Allocation Engine
SLNR	Signal to Leakage plus Noise Ratio
TAF	Transport Adaptation Function
TC	Transport Class
TCAM	Ternary Content-addressable
TDD	Time Division Duplex
TSON	Time-Shared Optical Network
UC	Use Case
UDW	Unified Data Gateway
UE	User Equipment
UL	Uplink
VBR	Variable Bit Rate
VLB	Valiant Load Balancing
VM	Virtual Machine
VN	Virtual Network
VNO	Virtual Network Operator
VNP	Virtual Network Provider
WDM	Wavelength Division Multiplexing
WP	Work Package