Grant Agreement No.: 814956
Research and Innovation Action
Call Topic: ICT-22-2018 EU-China 5G Collaboration

5G HarmoniseD Research and TrIals for serVice Evolution between EU and China

D5.2: Final report of 5G technology and service innovations

Version: v2.0

Deliverable type	R (Document, report)
Dissemination level	PU (Public)
Due date	31/05/2021
Submission date	18/10/2021
Lead editor	Sławomir Kukliński, Lechosław Tomaszewski (Orange)
Authors	Sławomir Kukliński, Lechosław Tomaszewski, Robert Kołakowski (Orange); Xianfu Chen, Tao Chen (VTT); Nathan Gomes, Shabnam Noor, Phillippos Assimakopoulos, Huiling Zhu, Ignas Laurinavicius (UKent), Akis Kourtis (Orion), Na Yi (UoS)
Reviewers	Jolanta Fabisiak (Orange); Nikolaos Tsampieris (ERTICO); Cédric Crettaz (MI)
Work package, Task	WP5 (T5.1, T5.2, T5.3)
Keywords	5G, Slicing, Radio access, Transport network, Virtualisation

Abstract

5G-DRIVE dedicates Work Package 5 (WP5) to advancements in "5G Technology and Service Innovations". Its main purpose is to ensure that the 5G test-bed implementations continue to rigorously evolve along the lines of real-world use cases as well as the 5G PPP vision. The work focuses on four distinct aspects, each one forming the core of a dedicated task: T5.1 – Radio Access and Transport Network; T5.2 – Network Virtualization and Slicing; T5.3 – 5G New Services, and T5.4 – Security and Privacy. This document includes the final achievements of T5.1, T5.2, and T5.3.

Disclaimer

[1] *http://creativecommons.org/licenses/by-nc-nd/3.0/deed.en_US*

Executive summary

5G-DRIVE, an innovative 34-month project, was focused on harmonising research and trials between the EU and China in the area of service evolution for enhanced Mobile Broadband (eMBB) and Vehicle-to-Everything (V2X). Whereas a significant part of the project was focused on trial and experimentation, this work package was devoted to research activities. The main research achievements of Tasks 5.1-5.3 of Work package 5 are presented in this deliverable. WP5 activities in the area of V2X security and privacy are presented in deliverable D5.3.

The technical topics described in this document deal with the performance evaluation of the existing solutions (especially in RAN and network virtualisation). Some mechanisms that can be exploited in future releases of the 5G network or beyond have also been proposed. The research topics presented in this document include:

- The approaches related to the improvements of the massive Multiple Input Multiple Output (mMIMO) and virtual resource management under the new distributed architecture of RAN, which consists of the Remote Units (RU), Distributed Units (DU) and Central Units (CU). To that end, we have looked into RAN performance optimisation by adopting Artificial Intelligence (AI). Furthermore, the transport network optimisations and the use of Software-Defined Networking/Network Function Virtualization (SDN/NFV) optimisations and analogue fronthaul are discussed.
- In the context of network slicing, we have focused on analysing the performance of virtual networks and MANO orchestration, on issues related to network slicing enabled RAN, and on integrating MEC with O-RAN and network slicing.
- Network virtualisation raises performance issues. We have evaluated the performance of in-software implemented network functions (i.e. VNFs) and assessed their performance.

The key results described in the deliverable are the following:

- In the RAN area, there is still room for MIMO improvement as well as advanced scheduling mechanisms that can be tailored for specific applications. We have used a customized ML technique with good results in order to avoid the radio overtraining problem. The use of correlation-based terminal offloading to MEC has provided reduced energy consumption. Evaluation of URLLC downlink transmission modes for efficient MEC task offloading has shown the benefits of multi-user message aggregation and STC. The usage of ML in RAN applications raises scalability uses. To that end, we have proposed distributed learning framework that effectively solves the issue.
- The RAN transport is a cost-sensitive issue as the number of 6G base stations as well as per user throughput is expected to be very high. Fronthaul architectures with analogue transport and digital signal processing at the end stations are promising as they can achieve high spectral efficiencies, increased flexibility and reduced latency. Such a DSP-assisted fronthaul has thus been proposed as an alternative to digital (packetised) fronthaul. However, the fronthaul has to be as scalable and flexible as possible for many reasons. It can be used primarily in 5G and beyond applications requiring support for multiplexing of signals with different numerologies, bandwidths, and massive MIMO. Also, so that it can operate within a network slicing/orchestration regime, perhaps in combination with a digital fronthaul/midhaul. Furthermore, such a fronthaul can be used to extend WDM systems (such as the one proposed by ORAN) by increasing their resource allocation resolution. To this end, digital techniques for frequency domain multiplexing/de-multiplexing large numbers of channels, one operating on the pre-Inverse Fast Fourier Transform (IFFT) “frequency-domain” samples while the other does so on the post-IFFT “time-domain” samples, can be used. The frequency-domain samples technique is very flexible and offers both lower overall complexity and better performance in terms of EVM. The time-domain approach is also flexible, but it requires significantly higher complexity and suffers from very narrow channel spacings, impairing the potential for achieving very high spectral efficiencies. However, under specific conditions, namely, when transporting non-power of 2 numbers of channels and/or when employing larger channel

spacings, the time-domain samples approach can lead to significantly reduced sampling rates and may thus be preferable. Both techniques can be used in DSP-assisted front hauling for 5G (and beyond) mobile networks offering flexibility that is not achievable by traditional SCM methods. In contrast, combinations of the two techniques can be envisaged for an even more flexible system. Indeed, a thorough investigation of such a combined approach has been carried out. It has been shown that appropriately combining the multiplexing techniques can balance sampling rate and complexity requirements, leading to hardware simplification while maintaining improved performance. Moreover, a radio-over-fibre fronthaul with intensity modulation in the downlink and phase modulation with interferometric detection in the uplink for simplified and power-efficient remote units has been proposed and demonstrated. An experimental investigation and verification of theoretical and simulation performance models have been conducted, demonstrating the ability of such an architecture to transport single-channel and multi-channel 5G-type radio waveforms that are have been digitally processed at DU and RU.

- Network slicing is a revolutionary technology that enables the customisation of the network according to service needs. Unfortunately, the technology deployment is much slower than expected. At the moment of writing this deliverable, there was none commercially deployed slicing-enabled 5G network. In the framework of this work package, some laboratory tests for evaluating the lifecycle performance of the OSM platform (a MANO compliant orchestrator) have been performed. The platform has shown excellent performance and stability, showing that even multiple instances of EPC (MAGMA template) can be efficiently orchestrated in a batch mode. The Katana slice manager has been used for the energy-efficient orchestration of slices. As RAN slicing is still an open research area, we have made an overview of approaches to RAN slicing. This high-level description has been provided in the context of the extension of the O-RAN platform - an industry RAN that so far has no support for network slicing. To that end, we have also proposed a new architecture of O-RAN that supports network slicing by adding components supporting network slicing to the near-RT RIC controller. We have also analysed a potential integration of network slicing enabled O-RAN with MEC and SON solutions. We have found that these platforms have both complementary and overlapping functionalities. Integrating them may bring multiple benefits. Having analysed the current MANO-based orchestration approaches, including the 3GPP one, we have found that the solution can impose many issues in a large scale deployment. One of the identified problems of MANO is the lack of separations of concerns. Thinking about beyond 5G networks, we have proposed a new concept, 6G-LEGO, an ecosystem enabling the creation of self-managed slices that can be easily stitched together. In this approach, the orchestrator functionality is simplified and reduced to slice agnostic resource orchestration only.
- As the network function virtualisation raises performance issues, we have designed and implemented the virtual Deep Packet Inspection function (vDPI), and we have made a performance analysis of the component. The analysis has shown the excellent performance of the function showing that we could minimise the impact of software processing on DPI efficiency.

A significant part of the presented activities was tightly linked with WP3 and WP4 scope of work and has already been reported in deliverables D3.2, D3.3, and D4.4. Moreover, the initial WP5 achievements were already presented in D5.1. The research on some of the technical topics was conducted in cooperation with China partners, and it mainly concerns the O-RAN issues.

Table of contents

List of figures

List of tables

Abbreviations

3GPP	The Third Generation Partnership Project
4G	The Fourth Generation of Mobile Communications
5G	The Fifth Generation of Mobile Communications
5GC	5G Core network
5GS	5G System
5GPPP	Fifth Generation (5G) Public Private Partnership
AC	Admission Control
ADC	Analogue to Digital Converter
AI	Artificial Intelligence
AMF	Access and Mobility Function
AoA	Angle of Arrival
AoD	Angle of Departure
API	Application Programming Interface
AWG	Arbitrary Waveform Generator
BER	Bit-Error Rate
BG	Border Gateway
BIP	Binary Integer Programming
BOF	Birds of a Feather
BPSK	Binary Phase Shift Keying
BS	Base Station
BSS	Business Support System
CH, ch	Channel
CM	Chain Manager
CN	Core Network
CP	Control Plane
CP-OFDM	Cyclic Prefix Orthogonal Frequency Division Multiplexing
CPRI	Common Public Radio Interface
CPU	Central Processing Unit
CSC	Core Slice Components
CU	Central Unit
CVaR	Conditional Value at Risk
CWL	Continuous Wave Laser
DL	Direct Learning
DNN	Deep Neural Network
DO	Domain Orchestrator
DPDK	Data Plane Development Kit
DRF	Dominant Resource Fair
DSP	Digital Signal Processor
DU	Distributed Unit
E2E	End-to-End
EM	Element Manager
eMBB	Enhanced Mobile Broadband
EMS	Element Manager System
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
EU	European Union
EVM	Error Vector Magnitude
FCAPS	Fault, Configuration, Administration, Performance and Security
FFT	Fast-Fourier Transform
F-OFDM	Filtered Orthogonal Frequency Division Multiplexing

GB	Grant-Based
GF	Grant-Free
GHz	Giga Hertz
GPU	Graphics Processing Unit
GS	Group Specification
GSM	Global System for Mobile Communications
GSMA	GSM Association
HARQ	Hybrid Automatic Repeat Request
HLS	Higher Layer Split
HTTP, http	HyperText Transfer Protocol
ICIC	Inter-Cell Interference Coordination
ICT	Information and Communication Technology
ID, id	Identifier
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet Engineering Task Force
IF	Intermediate Frequency
IFA	International Financial Architecture
IFFT	Inverse Fast-Fourier Transform
IMT	International Mobile Telecommunications
IoT	Internet of Things
IoV	Internet of Vehicles
IP	Internet Protocol
ISG	Industry Specifications Group
ISM	In-Slice Management
IT	Information Technology
ITS	Intelligent Transportation System
ITU	International Telecommunications Union
ITU-R	International Telecommunications Union – Radiocommunication Sector
ITU-T	International Telecommunications Union – Telecommunication Standardization Sector
KPI	Key Performance Indicator
KQI	Key Quality Indicator
KVM	Kernel-based Virtual Machine
LAN	Local Area Network
LLS	Lower Layer Split
LMMSE	Linear Minimum Mean-Square Error
LO	Local Oscillator
LR	Lattice-Reduction
LTE	Long Term Evolution
MaaS	Management as-a-Service
MAC	Medium Access Control
MANO	Management and Orchestration
MBB	Mobile Broadband
MC	Mission-Critical
MCPTT	Mission-Critical Push-To-Talk
MDAS	Management Data Analytics Service
MDO	Multi-Domain Orchestrator
MEC	Multi-access Edge Computing
MEP	MEC Platform
MF	Matched Filter
MIMO	Multiple Input Multiple Output

ML	Machine Learning
MMF	Mobility Management Function
mMIMO	Massive Multiple Input Multiple Output
MMSE	Minimum Mean-Square Error
mMTC	Massive Machine Type Communications
mmW	millimetre Wave
MSps	Millions of Samples per second
MTC	Machine Type Communications
MU	Mobile User
MU-MIMO	Multiuser Multiple-Input Multiple-Output
MZM	Mach-Zehnder Modulator
NBI	North Bound Interface
NCU	Network Capability Unit
NF	Network Function
NFV	Network Function Virtualisation
NFVI	Network Function Virtualisation Infrastructure
NFVO	Network Function Virtualisation Orchestrator
NG	Next Generation
NGMN	Next Generation Mobile Networks
NMS	Network Management System
NR	New Radio
NS	Network Slicing
NSD	Network Service Descriptor
NSI	Network Slice Instances
NSSAAF	Network Slice-Specific Authentication and Authorization Function
nVR	Non-Visual Region
NZ	Nyquist Zone
OAI	Open Air interface
OAM	Orchestration and Management
OFDM	Orthogonal Frequency Division Multiplexing
OSM	Open Source MANO
OSS	Operations Support System
OTA	Over-The-Air
PCS	Protocol Coding Sublayer
PDCP	Packet Data Convergence Protocol
PHY	Physical Layer
PLMN	Public Land Mobile Network
PoP	Point of Presence
PRB	Physical Radio Block
QAM	Quadrature Amplitude Modulation
QoE	Quality of Experience
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RAN	Radio Access Network
RAT	Radio Access Technology
RF	Radio Frequency
RIC	RAN Intelligent Controller
RID	Requester ID
RLC	Radio Link layer Control
RoF	Radio over Fibre
ROS	Resource Orchestration Support

RRC	Radio Resource Control
RRM	Radio Resource Management
RRU	Remote Radio Unit
RSRP	Reference Signal Received Power
RSSI	Received Signal Strength Indicator
S-NSSAI	Single Network Selection Assistance Information
SS-RSRP	Synchronization Signal Reference Signal Received Power
SS-SINR	Synchronization Signal Signal to Interference plus Noise Ratio
SAF	Slice Authentication Function
SBA	Service-Based Architecture
SCC	Slice Chain Configurator
SCM	Subcarrier Multiplexing
SDN	Software Defined Networking
SDR	Software Defined Radio
SEF	Slice Exposure Function
SG	Study Group
SGD	Stochastic Gradient Descent
SIC	Soft Interference Cancellation
SIM	Subscriber Identity Module
SIMO	Single-Input, Multiple-Output
SINR	Signal to Interference plus Noise Ratio
SLA	Service Level Agreement
SM	Slice Manager
SMF	Single Mode Fibre
SNR	Signal-to-Noise Ratio
SNS	Sub-Network Slices
SO	Slice Orchestrator
SON	Self-Organizing Network
SOS	Slice Operations Support
SSB	Single Side-Band
SSC	Sub-network Slice Configurator
STT	Slice Termination Time
SWA	Software Assurance
TCP	Transmission Control Protocol
ToC	Table of Contents
TP	Tenants Portal
TR	Technical Report
TS	Technical Specification
TSN	Time-Sensitive Networking
TTA	Telecommunications Technology Association
TTC	Telecommunications Technology Committee
TTD	True Time Delay
UDP	User Datagram Protocol
UE	User Equipment
UI	User Interface
UP	User Plane
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communications
V2X	Vehicle-to-Everything
VCA	Vehicle Certification Agency
vDPI	VNF Deep Packet Inspection
VIM	Virtualized Infrastructure Manager

VL	Virtual Link
VLAN	Virtual Local Area Network
VM	Virtual Machine
VNF	Virtual Network Function
VNFC	Virtual Network Function Component
VNFM	Virtual Network Function Manager
VPI	Virtual Photonics Incorporated
VR	Visual Region
VSDNC	Vehicular-SDN Controller
WAVE	Wireless Access in Vehicular Environments
WiFi, Wi-Fi	Wireless Fidelity
WG	Working Group

1 Introduction

1.1 Context of the deliverable

5G-DRIVE was an innovative, 34-month long project focused on harmonizing research and trials between EU and China in service evolution for enhanced Mobile Broadband (eMBB) Vehicle-to-Everything (V2X) communications. The 5G-DRIVE project was implemented by the consortium comprising 17 partners from EU academia, industry and commercial areas. The project's objectives were structured into three main areas: technical, regulatory, and business objectives. The 5G-DRIVE has dedicated an entire Work Package 5 (WP5) to improvements in "5G Technology and Service Innovations". The activities of this work package were focused on different research topics that were split into the following tasks:

- **Task 5.1: Radio Access and Transport Network.** The task activities concerned RAN improvements, Distributed Massive MIMO and RAN transport issues in beyond 5G mobile communications.
- **Task 5.2: Network Virtualization and Slicing** was focused on different aspects of network virtualization and slicing (performance analysis of the existing concepts, evaluation of algorithms and definition of beyond 5G network slicing concepts).
- **Task 5.3: 5G New Services** addresses novelties required at the service level to flexibly provision, replace, and migrate network functions.
- **Task 5.4: Security and Privacy aspects of 5G and the Internet of Vehicles** that tackles security and privacy challenges within the complex 5G ecosystem.

The initial outcome of the WP5, covering all the mentioned tasks, is included in Deliverable D5.1. This deliverable is a final deliverable of WP5 and concerns only activities of Tasks 5.1-5.3. The activities of Task 5.4 are reported in a companion Deliverable D5.3. It has to be noticed that some activities of WP5 concerning cooperation with Work Package 3 (WP3) and Work Package 4 (WP4) are included in deliverables of those Work Packages.

1.2 Scope and organization of the deliverable

The main purpose of WP5 is to develop and document key 5G improvements to support real operational scenarios in terms of expected functionalities, as well as scalability and performance characteristics. There are many technical areas, in which the 5G network can be improved. In 5G-DRIVE, we have selected specific topics from different network areas that are described in this deliverable. The technical areas were used to shape the structure of the document that is following:

- **Section 1** (current section) serves as an overall introduction to the document and discusses the scope of WP5.
- **Section 2** discusses Radio Access Technologies, including:
  - beam squint exploitation in millimetre-wave multi-carrier systems,
  - 3D beamforming evaluation,
  - orthogonal-SGD based learning approach for MIMO detection over URLLC,
  - correlation-based dynamic task offloading for user energy-efficiency maximization,
  - distributed learning-based edge traffic offloading in 5G networks and beyond,
  - distributed learning framework for the resource orchestration in computation offloading.

- **Section 3** focuses on RAN transport. It is devoted to:
  - phase-modulated RoF (Radio over Fibre) for efficient 5G fronthaul uplink,
  - flexible and efficient fronthaul by incorporating a combined multiplexing technique,
  - description and comparison of DSP-assisted analogue 5G fronthaul approaches.
- **Section 4** focuses on network virtualization and slicing, including aspects of MANO operations. This section discusses:
  - state of the art of RAN slicing,
  - a proposal of implementation of network slicing in O-RAN,
  - a proposal of O-RAN, network slicing, SON and MEC integration,
  - the AI-driven RAN resource orchestration for multi-tenant RAN slicing,
  - network slicing implementation using slice templates, based on the GSMA Generic Slice Template concept,
  - implementation of Deep Packet Inspection function as VNF.
- **Section 5** summarizes the outcomes of WP5 that includes standardization efforts, publications exchange of knowledge between WP5 with other work packages of 5G-DRIVE, as well as results of cooperation in the research area with Chinese partners.

The conceptual map of topics addressed within the document against the background of the 5G architecture is presented in Figure 1.

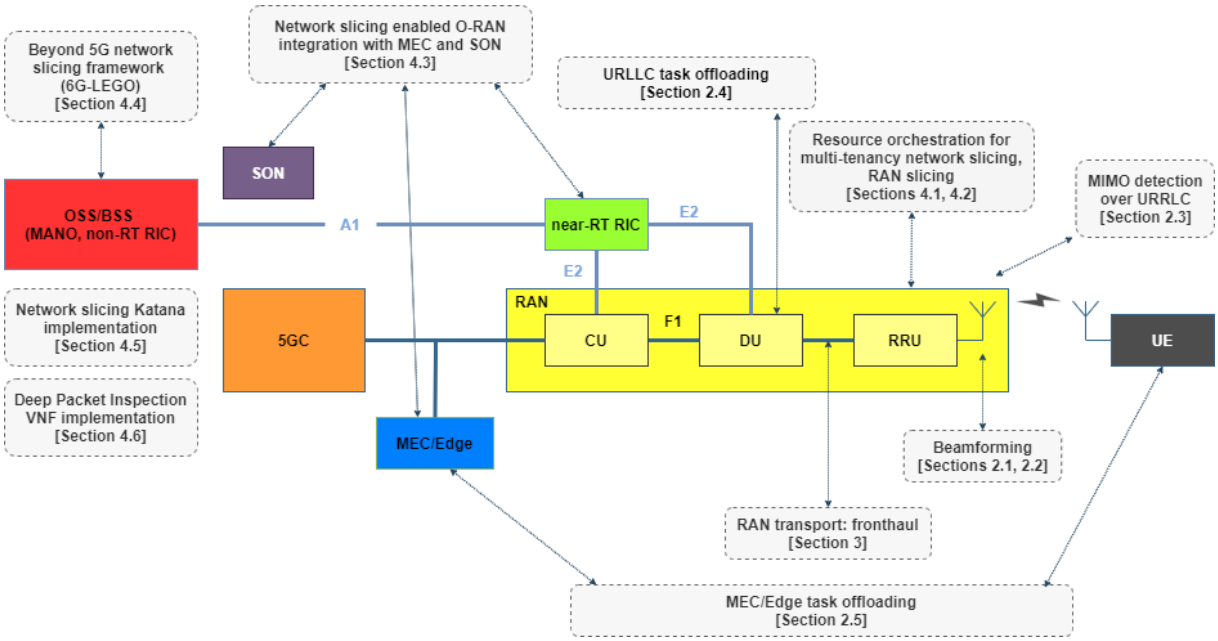


Figure 1: Map of WP5 activities.

2 Radio access

2.1 Beam squint exploitation in millimetre-wave multi-carrier systems

To support millimetre-wave (mmW) communications successfully, a large number of antennas (in the order of hundreds or thousands) need to be implemented to mitigate significant propagation and scattering losses. Phased array is one of the popular choices due to its low complexity. However, phased arrays are only a good implementation for narrowband systems, as phase shifters can be configured at only the carrier frequency. That is, the configuration of an approximation assuming phase shifter values remains stationary for all transmission frequencies, as there are practical limitations, such as the cost of hardware. Then only approximated performance could be obtained at frequencies other than the carrier frequency, which does not work well for wideband implementation. If the Angle of Arrival (AoA) or Angle of Departure (AoD) of a signal is not on the broadside, i.e. not 0 degrees, the beam direction is frequency-dependent. Any beam that is transmitting or receiving over a frequency that is not the carrier frequency gets steered away (squinted) as a function of frequency. In a wide bandwidth, this results in much smaller gains at edge frequencies, which is known as beam squint.

Currently, there are only few methods in solving the beam squint issue. True Time Delays (TTD) [1] is a hardware solution, suggesting the implementation of TTD circuit elements that would make phase-shifting frequency-independent even over wider frequency bands. However, this solution is undesirable in massive multiple input multiple output (mMIMO), collocated or distributed [2], as when the number of phase shifters and antennas is very large, the power consumption, implementation cost and circuit complexity scale make it impractical. Another method [3] was proposed to increase the density of the codebook to combat the beam squint effects, which is more practical. However, having very large codebooks can lead to long beamforming times, introducing latency. Other studies were performed on hybrid beamforming systems [4], [5] and, as such, introduced extra flexibility into the system through a digital precoder. The latency and complexity of the system could be very high due to the adaptation of a digital precoder, especially in a system with multiple users. Therefore, it is important to study the beam squint problem and look for more efficient ways of solving it to reduce the latency introduced by the codebook-based solution and the implementation complexity that is associated with the hardware solution. Therefore, this work aims to minimize the implementation cost of beam squint mitigation by studying analogue beamforming, and a subcarrier to beam allocation scheme is proposed to improve the throughput.

2.1.1 System model

Consider an orthogonal frequency division multiplex (OFDM) mMIMO system with N_t transmit antennas and M subcarriers labelled m = 1, 2,⋯ , M. The N_t antennas are deployed in the shape of a Uniform Linear Array, as shown in Figure 2. The number of radio frequency (RF) chains determines how many users' data streams can be scheduled at a single time instance. All antennas are isotropic and of the same shape. Denote λ_c as the wavelength of the carrier frequency, f_c. The distance d between two adjacent antenna elements is set to be half wavelength, i.e. $d = \lambda_c/2$. AoD or AoA is denoted as ϑ, relative to the broadside. The angle increases clockwise, and the analytical limit is $\vartheta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. We will define a virtual angle ψ to keep expressions shorter, which is given by:

$$\psi = \sin\theta \tag{2-1}$$

Denote β_n as a phase shift value of the nth antenna element. The phase shifters are implemented in a beamforming vector, which is denoted as:

$$\boldsymbol{w} = [e^{j\beta_1}, e^{j\beta_2}, , e^{j\beta_{N_t}}]^H \tag{2-2}$$

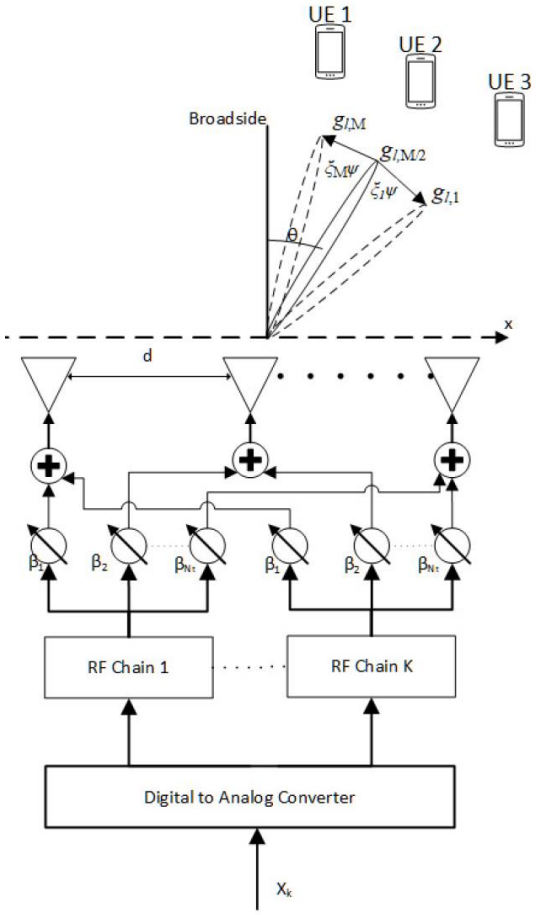


Figure 2: Beam squint effect demonstrated in a multiuser scenario.

With the carrier frequency f_c, given a target angle θ_F, the goal of a beamforming system is to maximise the array gain towards angle θ_F, which can be converted to ψ_F following (1). The goal can be achieved by setting up phase shifters to ensure all of the N_t antenna elements have the same time delay for receive or transmit signals at the carrier frequency f_c, which is a solution for optimal implementation cost of the system in a multi-carrier environment. Following [6], to focus on an angle ψ_F, the phase shifter configuration is given by:

$$\beta_n(\psi_F) = 2\pi \cdot \lambda^{-1} \cdot d \cdot (n-1) \cdot \psi_F \tag{2-3}$$

Corresponding to the beamforming vector ***w*** for any frequency *f*, the ULA response vector at any angle ϑ is derived as:

$$a(\theta) = [1, e^{j2\pi\lambda^{-1}1d\psi}, e^{j2\pi\lambda^{-1}2d\psi}, \dots, e^{j2\pi\lambda^{-1}(N-1d)\psi}], \psi \in [-1,1] \tag{2-4}$$

Then, based on the array response vector $a(\vartheta)$ at frequency *f*, and pre-designed beamforming vector, *w*, the array gain normalized by the square root of the number of antenna elements N_t is given by:

$$g(w,\theta) = \frac{1}{\sqrt{N_t}} w^H a(\theta) = \frac{1}{\sqrt{N_t}} \sum_{n=1}^{N_t} e^{j[2\pi\lambda^{-1}(n-1)d\psi - \beta_n]} \tag{2-5}$$

where (·) denotes the Hermitian transpose.

2.1.2 Beam squint model

As stated before, phase shifters are designed for the carrier frequency, meaning that they are fixed, regardless of operating frequency, within bandwidth *B*. This introduces beam squint. Beam squint can be effectively defined as the ratio ξ of operational frequency *f* to the carrier frequency f_c:

$$\xi = \frac{f}{f_c} \tag{2-6}$$

where $f \in [f_c - B/2, f_c + B/2]$. Then, the fractional bandwidth is defined as:

$$b=B/f_c \tag{2-7}$$

and $\xi \in [1 - b/2, 1 + b/2]$. Since ξ is dependent on b, any reduction in b will reduce the range of ξ available. For example, in a wide bandwidth of 2.5 GHz with a centre frequency 73 GHz (considered as the carrier frequency), ξ varies from 0.9829 to 1.0166. Figure 3 demonstrates that the gain at edge frequencies is much smaller than at the centre frequency for a large number of antennas N_t = 256. Any frequency corresponding to gains that are 3 dB or smaller than the maximum gain is considered unusable for transmission to the user. In [6], the array gain derived in (2-5) is modified for a subcarrier with a ratio ξ, AoA/AoD ψ, and beam focus angle ψ_F. This is expressed as $g(\xi\psi - \psi_F)$, where:

$$g(x) = \frac{\sin\left(\frac{N_t \pi x}{2}\right)}{\sqrt{N_t}\sin\left(\frac{\pi x}{2}\right)} e^{j\frac{(N_t-1)\pi x}{2}} \tag{2-8}$$

A subcarrier with ξ at AoA/AoD ψ has a gain equivalent to a gain for the carrier frequency f_c at AoA/AoD $\psi' = \xi\psi$. The maximum array gain is achieved by:

$$g_{max} = \max_{\psi\in[-1,1]} g(\psi - \psi_F) = \sqrt{N_t} \tag{2-9}$$

It can be seen that at frequency f, the maximum array gain is achieved at angle $\psi = \psi_F/\xi$, which makes the beam steer away from the focus angle ψ_F when $\xi \neq 1$, i.e. $f \neq f_c$. The beam squint effect is illustrated in Figure 3. As the bandwidth changes, so does the range of ξ. When the bandwidth is narrow, $\xi \approx 1$ or $|\Delta\xi| \approx 0$, from Figure 3, this means that the beamforming gain $g \approx \sqrt{N_t}$ for the entire bandwidth. As the bandwidth becomes relatively large e.g. crosses $|\Delta\xi|$ =0.008 the performance is optimal for a small range of frequencies around the centre, but at the edges, when ξ= 0.9829 or 1.0166, the gain is approaching the gain of the first sidelobe of the same beam. In Figure 2, these edge frequencies correspond to the dashed beams.

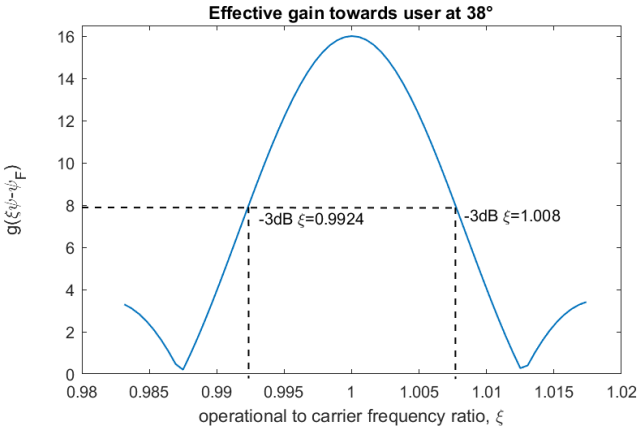


Figure 3: Effective gain across entire normalized bandwidth in a single carrier system, $N_t = 256$.

2.1.3 Problem analysis – OFDM system channel capacity

Here, the receiver has only one antenna, and only transmitter beam squint is considered. The analysis is performed on multiple user scenarios, and a single user implementation with multiple RF chains is studied as a special case. Equal power is allocated across all subcarriers. This is common under the condition of no channel state information. mmW bands with narrow beams ensure a sparse channel [7]. As a result, line of sight (LoS) is assured.

Within bandwidth *B*, before beamforming, the antenna receiving a signal receives power $P = P_t/K$, where P_t is the total power of the system and *K* is the number of scheduled users. The noise power of AWGN split across M subcarriers is $\sigma^2 = \frac{B}{M}N_0$. The channel capacity for a multiple user OFDM system with beam squint at AoD ψ and beam focus angle ψ_F is thus:

$$R_{BS} = \sum_{k=1}^{K} \frac{B}{M} \sum_{m=1}^{M} \log\left(1 + \frac{P_t \left| g_k(\xi_m \psi_k - \psi_{F,k}) \right|^2}{K \frac{B}{M} N_0}\right) \quad (2\text{-}10)$$

where $\xi_m = 1 + \frac{(2m-M+1)b}{2M}$. In the beam squint environment, variables $g_k(\cdot)$ and $\psi_{F,k}$ show that the beamforming gain and focus angle, respectively, for a specific user. This specification is important, as, depending on the user location, the number of beams available for use, e.g. user *k*, varies when as beam squint is stronger further away from the broadside. For comparison, the channel capacity for a multi-user system with no beam squint denoted as R_{NBS} is derived. Based on the single user capacity obtained in [6] at AoD ϑ, and beam focus angle ψ_F, the multi-user system capacity can be simply *K* times of that achieved in a single-user scenario, as without beam squint, i.e. ξ= 1 for all users, and $g_1 = \cdots = g_k = \cdots = g_K = g_{max}$. Therefore, the capacity of *K* users is given by:

$$R_{NBS} = \sum_{k=1}^{K} \frac{B}{M} \log\left(1 + \frac{P_t |g_{max}|^2}{K \frac{B}{M} N_0}\right) \quad (2\text{-}11)$$

Figure 4 shows a comparison of the maximum achievable throughput for the scenarios with beam squint (R_{BS}) and without beam squint (R_{NBS}) as a function of fractional bandwidth defined in (2-7), when the number of users in the system is one and three, respectively. As fractional bandwidth *b* increases, the range of ξ extends, which leads to strengthening the beam squint effect. As such, while the gain is constant for the R_{NBS} scenario, the throughput does not change, however as beam squint is introduced for the R_{BS} scenario, the capacity degrades significantly as the bandwidth increases. This shows that beam squint effects for wide communication bandwidths are a significant issue, especially when there are multiple users.

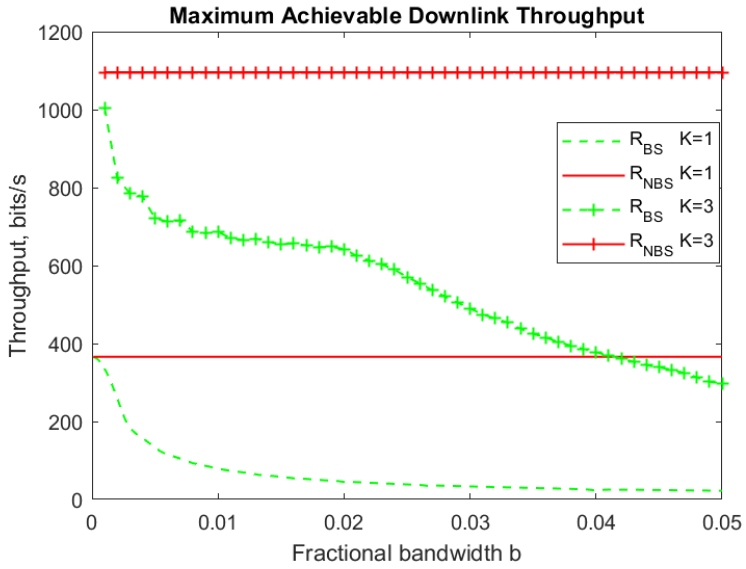


Figure 4: Achievable downlink throughput versus fractional bandwidth. Number of antennas N_t = 256.

2.1.4 Subcarrier-to-beam allocation

When serial data containing *K* users' data enter the transmission system, conventionally, these data will be distributed to each respective user and their exclusively assigned beams. Due to beam squint effects, the data transmitted will suffer heavy losses on a high number of subcarriers assigned to each user. Hence in the work, a subcarrier-to-beam allocation (SBA) scheme is proposed, which acquires information about beam squint and uses this information to maximize the gain $g_k(x)$ towards any given user. The scheme is proposed based on the concept of interleaving. The input data, as the input of every single beam, will be split across $L_k < N_t$ beams, and this may result in some data of a user, e.g. user *k*, being mixed in with other users' data on the beams allocated to them. A beam may not be exclusively allocated to one user. Given a user, due to beam squint effects, the edge subcarriers would have the beams to point away from the user and result in degraded gains. As the beam pattern is fixed, for the purpose of interleaving, given a user, SBA algorithm selects *L* significant beams, according to

the information of the number of beams present within an angle range given by $\theta_{max}(f) \in [\theta_F - \Theta_s, \theta_F + \Theta_s]$, where:

$$\Theta_s = |\theta_{max}(f) - \theta_F| = \left| \sin^{-1}\left(\frac{\sin(\theta_F)}{1+\frac{f}{f_c}} \right) - \theta_F \right| \quad (2\text{-}12)$$

and $f \in [-\frac{B}{2}, \frac{B}{2}]$ is the operating frequency relative to the carrier frequency.

Then, interleaving will be carried out among the L_k selected beams for the data of user k. Therefore, the number of selected beams, L_k, would affect the system performance. In the SBA, a gain difference factor, $e_{l,m}(\theta)$, is defined for each beam at AoD θ to find suitable subcarriers for transmission across all chosen beams, which is given by $e_{l,m}(\theta) = \frac{g_{l,m}(\theta) - g_{max}}{g_{max}}$, where $g_{l,m}(\theta)$ is the gain of beam l on subcarrier m at a given angle θ. Each subcarrier that has gain difference factor higher than a pre-defined threshold e_{th} is stored in a subcarrier gain matrix $\mathbf{S} = \{S[l,m] = g_{l,m}(\theta)\}_{L_k \times M}$. After it is generated, $S[l,m]$ is examined, and the beam with the highest gain on subcarrier m, denoted by $l_{k,m}$ is allocated to user k. Its gain is recorded by $G[m]_k = S[l_{k,m}, m]$. The method of allocating subcarriers to beams is summarised in Algorithm 1.

Due to the nesting of loops, the complexity of the algorithm for a full user range is $O(KL_kM)$. None of these variables scales quickly. The number of usable beams for beam squint compensation increases slowly with N_t, number of users that would be available to benefit from the algorithm depends on their spacing and varies between $[1, N_{RF}]$. M can take a multitude of discrete values that can be constant depending on the system and it allows for further simplification of the complexity to $O(KL_k)$.

Using information that is returned from Algorithm 1, it is now possible to rearrange input data to ensure it reaches the required beam to benefit from the possible gains. A second algorithm is proposed to meet this goal. If an ordered arbitrary serial data set $X \in \mathbb{R}$ containing K users' data is to be mapped to beams, conventionally, one user's data would only be exclusively allocated to one beam. In the SBA scheme, using Algorithms 1 and 2 will override the selection, as one user may get allocated more than one beam. Then, following Algorithm 1, which has successfully returned a subcarrier to beam allocation set for each user separately, Algorithm 2 is developed for data interleaving, which creates a data allocation indication matrix, or the output stream $\mathbf{D} \in \mathbb{R}^{L \times M \times K}$. The element $d_{l,m,k}$ of **D** takes the value of one if the beam L is selected for user k to transmit data on subcarrier m. Then, this data stream is a modified input serial data set of each user, with all the information remapped following the rule created by Algorithm 1. An input x_m from X_k is selected and mapped to the $d_{l,m}$ slot. The final output of the algorithm is interleaving all matrices $\mathbf{D}_k$ to form the final allocation matrix **D,** which has inputs X remapped in an order of ascending subcarrier index across multiple beams.

Algorithm 1 Subcarrier to Beam Allocation

Initialise subcarrier index $m = 1$ and user index k
Locate number of beams L_k within Θ_s from main beam.
for all significant beams $l = 1, 2, \cdots, L_k$ **do**
  Calculate phase shifter values β
  Apply $AF[\theta]$ to each phase shifter
  Calculate the gain $g_l[\theta]$
  Normalise gain to number of antenna elements $\frac{g_l[\theta]}{N_t}$
  Find maximum gain $g_{max} = \max g_{l,m}[\theta]$
  for all $g_{l,m}[\theta]$ $m = 1, 2, \cdots, M$ **do**
    if $e_{l,m} = \frac{|g_{l,m}[\theta] - g_{max}|}{g_{max}} \cdot 100 \leq e_{th}$ **then**
      Generate a usable subcarrier gain matrix $\mathbf{S} = \{S[l,m] = g_{l,m}[\theta], l = 1, \cdots, L_k, m = 1, \cdots, M\}$
    end if
  end for
end for
Choose highest gain subcarriers for beam allocation $G_k[m] = \max_l(S[l,m]), m = 1, 2, \cdots, M, l = 1, 2, \cdots, L_k$
Locate beam indexes corresponding to the gains $G_k[m]$, $A_k = \{l_{k,m} = \arg\max_l(S[l,m]), m = 1, 2, \cdots, M, l = 1, 2, \cdots, L_k\}$
Return A_k, G_k

Algorithm 2 Data Interleaving

Initialise data set $X = \bigcup X_k, k = 1, 2, \cdots, K$ and data allocation matrix $\mathbf{D} = \mathbf{0}^{L \times M}$
Load A_k from Algorithm 1
Insert data into $\mathbf{D}_k = \{d_{l,m} = x_m : x_m \in X_k, l_m = \mathbf{A}_{k,m}, m = 1, 2, \cdots, M\}$
Interleave data $\mathbf{D} = \sum_{k=1}^{K} \mathbf{D}_k$
Return $\mathbf{D}$

2.1.5 Simulation results and performance evaluation

A. Bit Error Rate (BER)

When Binary Phase Shift Keying (BPSK) is adopted in the system, Figure 5 shows the BER performance of the system at frequency fc = 73 GHz and with a bandwidth BW = 2.5 GHz in an AWGN LOS channel. In this figure, BER_{BS} represents the results when no method is used to mitigate the effect of the beam squint in the system. BER_{BS} represents the performance of the proposed scheme when beam squint exists in the system. BER_{BS} represents the performance of the system without beam squint. From Figure 5, an improvement is shown when number of antennas $N_t > 64$. The performance under 128 antennas shows a slightly better, while the performance has improved significantly when there are 256 antennas, with an overall gain of 12 dB and only 1.5 dB loss over the system without beam squint.

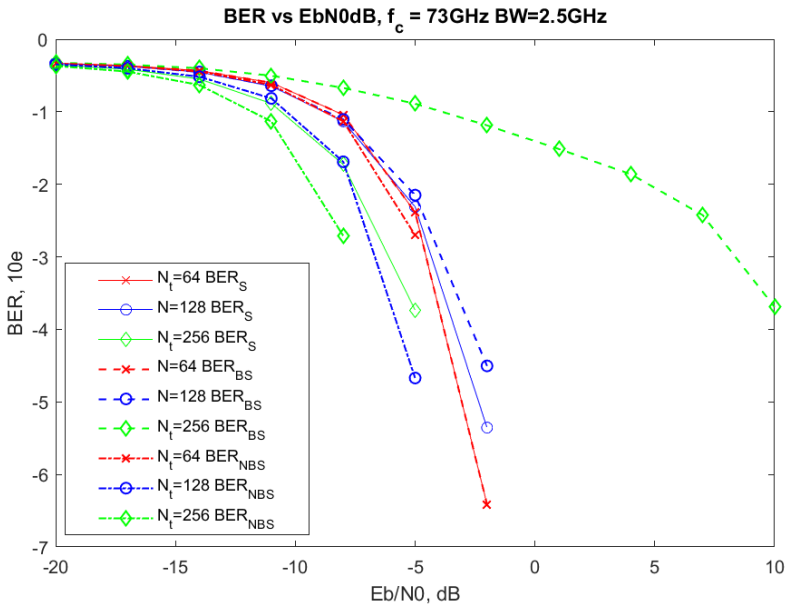


Figure 5: BER versus SNR.

B. Capacity

In this subsection, the impact of increasing the number of antennas the system capacity, when the number of users *K* takes the value of one and three. Among *K* UEs, one is selected as a reference UE. In the simulations, user location is randomly generated in the range of [0,π/2] for the reference UE, while the other UEs had their positions set to be relative to the reference UE's location. To study the scenario, in which performance is significantly affected by beam squint, the correlation between UEs' locations ensures that the three adjacent beams would be signalling towards the three users with maximum gains. Figure 6 shows, with a different number of users in the system, how the number of antennas affects the averaged throughput over different UE locations, as the performance over the entire direction range is similar to the proposed algorithm. When Algorithms 1 and 2 are applied, the capacity increases significantly as the number of antennas increases, as shown by the result for RS single user and RS multi-user in Figure 6.

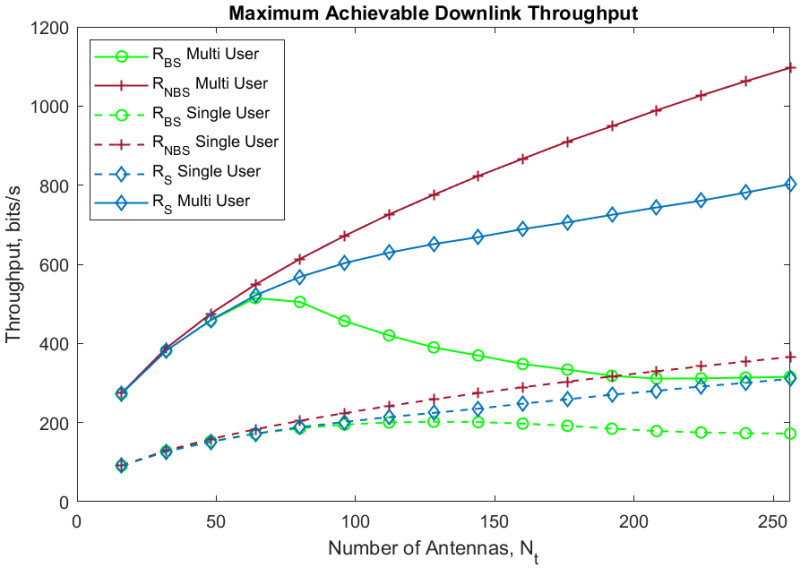


Figure 6: Achievable throughput versus the number of antenna elements.

2.2 3D Beamforming

3D beamforming technique is attractive in massive MIMO wireless communications owning the beneficial capability of controlling the strength of radiation field energy in different directions in the spatial domain. Performances of 3D beamforming in wireless communications obviously rely on the properties of the beam pattern. Therefore, it is of importance to investigate in depth the properties and applications of 3D beamforming, analysing system performance of massive MIMO communications, and evaluating the actual performance through field trials.

2.2.1 Application scenario

Due to ultra-high data rate transmissions, the coverage area of a 5G base station (BS), named next-generation nodeB (gNB), should be much smaller than that of the 4G BS. In particular, in dense areas, where there are a number of high-rise buildings, the gNB must provide services through 3D beamforming. Figure 7 shows one application scenario, where there are high-rise buildings served by 5G massive MIMO, which provides 3D beams not only horizontally, but also vertically. Each vertical layer of the formed beams covers a number of floors in a high-rise building. Thus, to cover the whole high-rise building, multiple layers of beams are required.

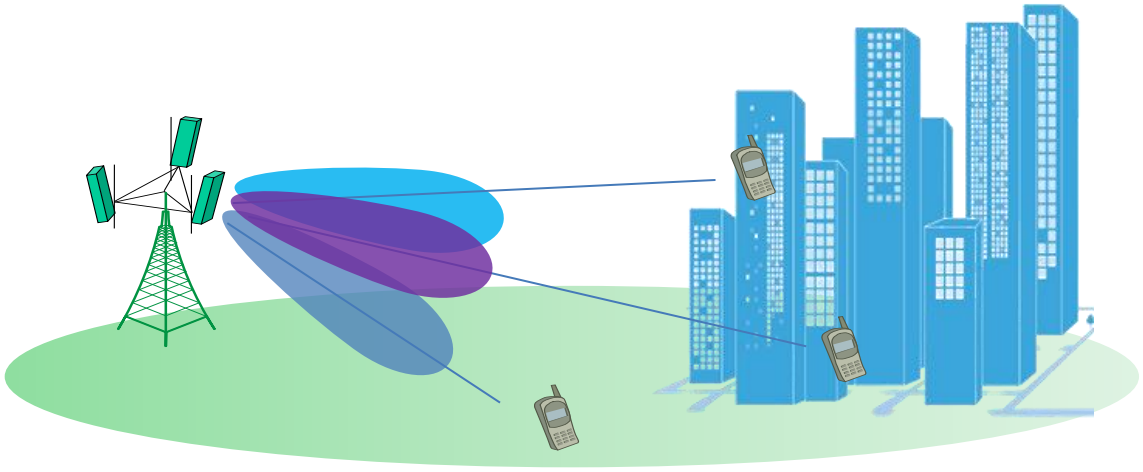


Figure 7: 3D beamforming application scenario

In order to study the 3D beamforming, linear horizontal and vertical arrays will be described in the following sections.

2.2.2 Linear arrays

First, we present the horizontal linear array. For each linear horizontal array, the spacing between two antenna elements is $d_h = 0.5\lambda$, where λ is the RF signal wavelength. *N* and θ represent the number of horizontal antenna elements and an angle of departure (AoD) at the horizontal plane. Assuming that each antenna element is weighted by a weighting factor, the normalized array factor of any horizontal beam *n*, where $n = 1, 2, \cdots, N$, is given by [8]:

$$A_n(\theta) = \frac{\sin(\frac{d_h}{\lambda}\pi N \sin\theta - \beta_n)}{\mathrm{N}\sin(\frac{d_h}{\lambda}\pi \sin\theta - \frac{1}{N}\beta_n)} \quad (2\text{-}13)$$

where β_n is given by:

$$\beta_n = \left(-\frac{N+1}{2} + n\right)\pi \tag{2-14}$$

Second, we present the linear vertical array. In practice, the vertical angle of coverage should be smaller than the horizontal angle of coverage. Therefore, fewer layers of vertical beams may be considered. Assuming there are *L* vertical antenna elements with the spatial separation 0.83λ between two adjacent elements. The L antenna elements are divided into *M* sub-arrays (equivalent units), with each sub-array consisting of L/M antenna elements. Note that each sub-array is weighted with one phase angle (one RF chain). Assuming that each sub-array has three antenna elements (i.e. $L/M = 3$, the spatial separation between two equivalent units is $d_v = 0.83\lambda \cdot L/M = 0.83\lambda \cdot 3 = 2.49\lambda$. Assuming that ϕ represents an AoD at the vertical, the normalized array factor of any vertical beam *m*, $m = 1, 2, \cdots, M$, is given by:

$$B_m(\phi) = \frac{\sin(\frac{d_v}{\lambda}\pi M \sin\phi - \beta_m)}{\mathrm{M}\sin(\frac{d_v}{\lambda}\pi \sin\phi - \frac{1}{M}\beta_m)} W_L(\phi) \tag{2-15}$$

where β_m is given by (2-14) with substitution of *N* and *n* with *M* and *m*, respectively, $W_L(\phi)$ is a window function depending on the number, *L*, of each equivalent element and given by [9]:

$$W_L(\phi) = \left|\frac{\sin(0.83\pi L \sin\phi)}{0.83\pi L \sin\phi}\right| \tag{2-16}$$

When *L=1*, $W_L(\phi) = 1$ for vertical AoD, which is a constant window function. Figure 12 shows the beam patterns for *M*=4 and *L*=12. It can be seen that for four equivalent antenna units, the four vertical beams may cover 24-degree angles.

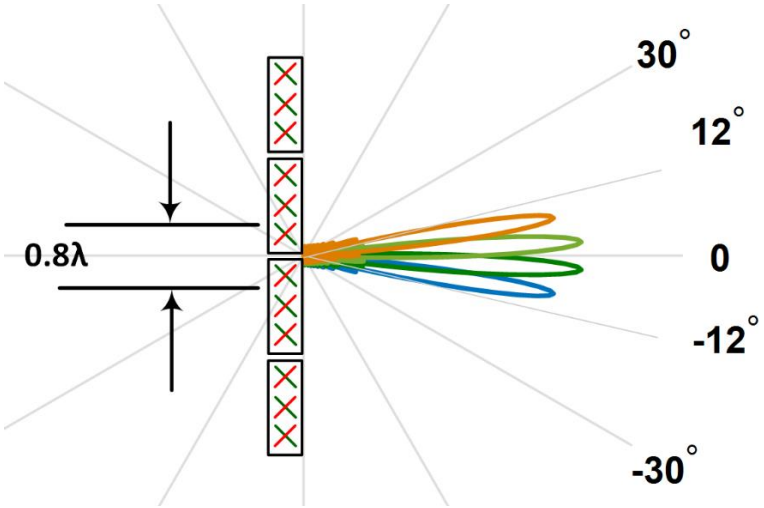


Figure 8: Equivalent vertical antenna sub-arrays, four sub-arrays (4 RF chains) with three antenna elements in each sub-array.

2.2.3 Rectangular antenna arrays

The conceptual block diagram of the rectangular antenna arrays used in WP3 is shown in Figure 13. The rectangular antenna array structure has 32 antenna elements (4 rows and 8 columns). Since each sub-array has 3 vertical antenna elements, the rectangular array has 96 antenna elements in total. Each sub-array is connected to one dedicated RF chain, and there are up to 32 RF chains.

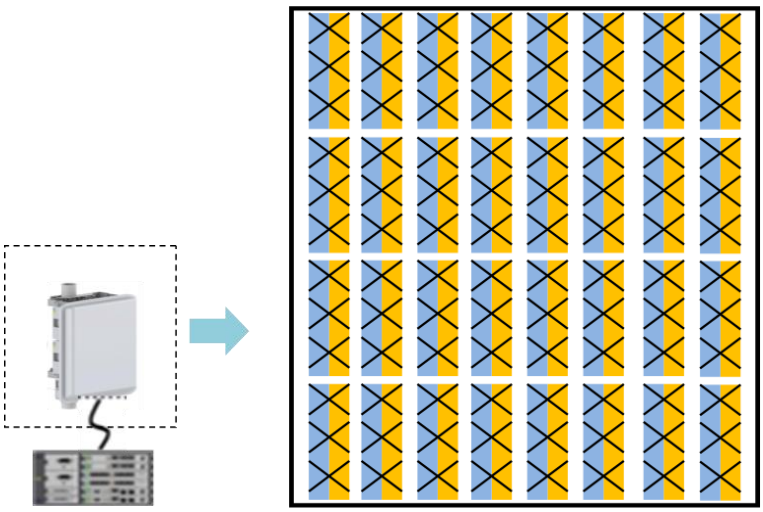
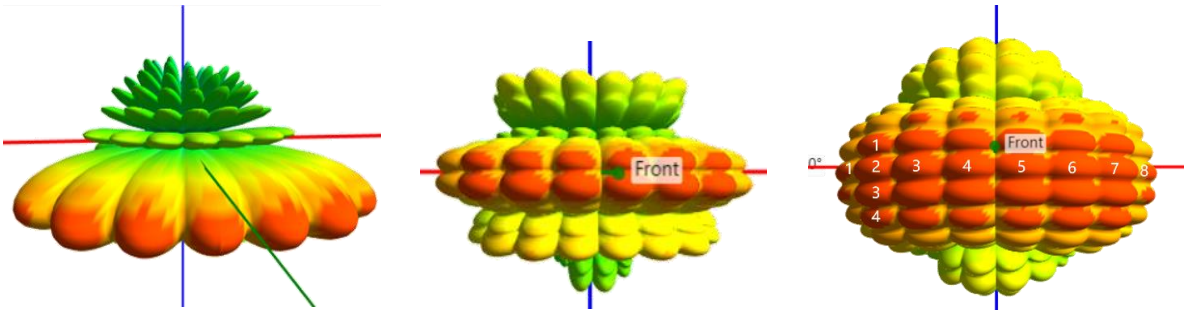


Figure 9: Rectangular antenna configuration (4×8 sub-array configuration)

When L/M=3, the space between two adjacent sub-arrays is 2.4λ. By considering downlink digital beamforming similarly to the Butler method, 32 sub-arrays may form 32 beams with four rows (vertical) and eight columns (horizontal). The normalized array factor of any 3D beam is given by:

$$C_{nm} = A_n(\theta) \cdot \mathrm{B}_m(\phi) \tag{2-17}$$

Figure 10 illustrates the beam pattern generated according to (2-17). In the vertical plane, all the beams are within 24 degrees, while in the horizontal plane, all the beams are within 120 degrees (or, typically, one sector). The vertical range of 24 degrees may cover the most realistic scenarios for building coverage. If each beam supports one user, 32 users may be supported by this complete MIMO architecture. Further, if antenna polarization technology is used [9], each beam may support two users through the two polarizations, and the number of users may be doubled so that 64 users may be supported by only using the spatial domain.



(a) One vertical layer (b) Two vertical layers (c) Four vertical layers

Figure 10: Complete beam patterns, (a) one layer of vertical beams (1×8 sub-array configuration), (b): two layers of vertical beams (2×8 sub-array) configuration, (c) 4 layers of vertical beams (4×8 sub-array configuration)

2.3 Orthogonal-SGD based learning approach for MIMO detection over URLLC

Detection of multiple-input multiple-output (MIMO) signals through machine learning (ML) has demonstrated remarkable advantages in terms of their strong parallel-processing ability, good performance-complexity trade-off, as well as self-optimization with respect to the dynamics of wireless channels [10]. More remarkably, data-driven ML approaches are model-independent, i.e. they learn

to detect signals without the need of an explicit model of the signal propagation, which have been reported in our previous work in D5.1 [11] and other publications in [12], [13], [14]. This is particularly useful for receivers to reconstruct signals from random nonlinear distortions, which are often very hard to handle with hand-engineered approaches. Meanwhile, ML-assisted wireless receivers can also be model-driven, which can take advantage of the model knowledge to mitigate the curse of dimensionality problem inherent in the deep learning procedure. Moreover, ML and hand-engineered approaches can work together to form a synergy when conducting signal detection.

Despite already numerous contributions in this domain, there are very few results that have been reported so far, concerning the wireless channel over-training problem. More specifically, current ML-assisted receivers are trained mainly for a specific channel model, such as the MIMO Rayleigh-fading channel. However, a receiver that is well trained for one channel model is often too sub-optimum or even unsuitable for other channel models. This is also known as the training set over-fitting problem in the general artificial intelligence domain. In the literature, there are a couple of ways to handle the over-training problem. One approach is called continual learning, which aims to inject new knowledge without forgetting previously learned knowledge. Consequently, machines will always adapt themselves to be better optimized for the latest training samples (i.e. new channel models in telecommunications). The other approach is called multi-task learning, which aims to improve all training tasks simultaneously by combining their common features. These approaches have already achieved promising results in traditional ML applications, such as natural language processing or image/video recognition; however, it is still not clear whether these approaches can be cost-effective to handle wireless channels that are random, continuous, and infinite in their states.

In this subsection, we introduce our initial results of a novel algorithm to tackle the wireless channel over-training problem when machines learn to detect communication signals in MIMO fading channels. The basic idea lies in the discovery and exploitation of the orthogonality of training samples between the current training epoch and past training epochs. More specifically, the O-SGD algorithm does not update the neural network simply based upon training samples of the current epoch. Instead, it first discovers the correlation between current training samples and historical training data and then updates the neural network with those uncorrelated components. The network updating occurs only in those identified null subspaces. By such means, the neural network can understand and memorize uncorrelated components between different training tasks (e.g. channel models). This idea is evaluated for the artificial neural network (ANN)-assisted MIMO detection with various channel models. It is shown, through computer simulations, that O-SGD is very robust to channel model variations as well as SNR variations.

ML signal detection faces great challenges when the MIMO channel matrix is randomly time-varying; as in this case, the set of possibly received signals becomes infinite. More seriously, the randomness of the MIMO channel will result in the channel ambiguity, i.e. the receiver's observation y might correspond to various combinations of the channel matrix H and the transmitted signal block x even in the noiseless case. In this case, ML is not able to conduct signal classification since the bijection between y and x does no longer hold. Theoretically, the channel ambiguity can be resolved by feeding the machine with the full channel knowledge, i.e. the input to the ANN-assisted MIMO receiver consists of the received signal block y as well as the channel matrix H or more precisely, its vector-equivalent form h, which is often called the data-driven approach. However, the dimension of the H-defined training input grows much faster than the y-defined training input, and this could result in inefficient learning at the ANN training stage. In this regard, the model-driven approach demonstrates remarkable advantages by replacing the received signal block y with its matched filter (MF) equalized version $H^H y$ and the channel matrix H with the corresponding version $H^H H$; cf. the block diagram of the ANN-assisted MIMO detection in Figure 11. By such means, the growth rate for both inputs is largely scaled down.

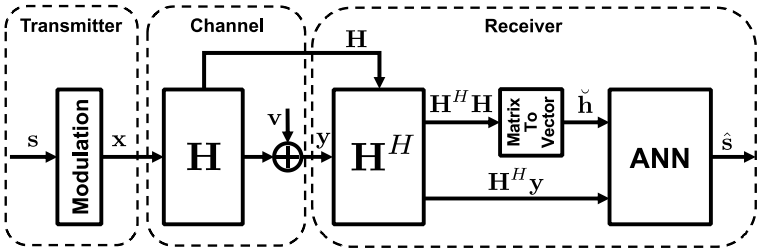


Figure 11: Block diagram of the ANN-assisted MIMO detection.

The basic idea of the proposed algorithm lies in the discovery and exploitation of the orthogonality of the training samples between the current training epoch and previous training epochs. Specifically, the algorithm does not update the neural network simply based upon the current training input. Instead, it discovers the correlation between the current training samples and the historical training data. By such means, ANN can understand and memorize uncorrelated components between different training data set.

The proposed algorithm is demonstrated through simulation-based experiments here. Our computer simulations are structured into three experiments. **Experiment 1** aims to demonstrate our hypothesis on the existence of the channel over-training problem in ANN-assisted MIMO signal detection. **Experiment 2** and **Experiment 3** evaluate the performance of the conventional SGD algorithm and the proposed O-SGD algorithm by training multiple tasks sequentially and simultaneously in the time domain, respectively. The size of the MIMO system is 4-by-8, and QPSK modulation is considered at the transmitter side. The key metric utilized for performance comparison is the average BER over sufficient Monte-Carlo trails of multiple block fading channels. Moreover, the signal-to-noise ratios (SNR) is defined as the average received information bit-energy to noise ratio per receive antenna (i.e. E_b/N_0).

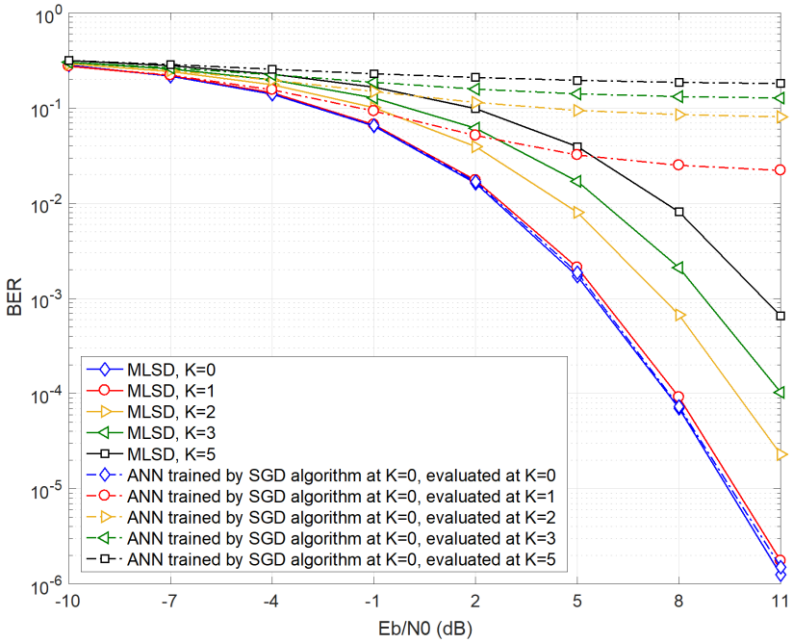


Figure 12: BER as a function of Eb/N0 for ANN-assisted MIMO signal detection. ANN is trained for the Rayleigh fading channel (i.e. K = 0) and evaluated under various channel models.

Experiment 1: In this experiment, an ANN-assisted MIMO receiver optimized for the Rayleigh fading channel (i.e. K = 0) is evaluated under multiple other channel models. The aim of this experiment is to demonstrate the existence of a channel over-training problem in ANN-assisted MIMO signal detection. Moreover, the training is operated at E_b/N_0 = 8 dB with a mini-batch size of 500, as the above configurations are found to provide the best performance.

Figure 12 shows the average BER performance of the ANN-assisted MIMO receiver trained by the SGD algorithm. The baseline for performance comparison is the optimum maximum-likelihood sequence detection (MLSD). It is shown that the ANN-assisted MIMO receiver achieves near-optimum performance under the trained channel model (i.e. K = 0); the performance gap to the MLSD is almost negligible. However, the detection performance significantly decreased when other channel models are considered (i.e. K = 1, 2, 3, 5). The performance gap between the ANN-assisted MIMO receiver and MLSD is more than 10 dB at a high SNR regime. The above phenomena coincide with our hypothesis that the channel over-training problem exists in the ANN-assisted MIMO signal detection.

Experiment 2: In this experiment, ANN-assisted MIMO receiver is firstly trained under the Rayleigh fading channel (i.e. K = 0) by using either SGD or O-SGD algorithm. After training converge, a new training task (i.e. Rician fading channel with k = 1) is operated on the previously trained ANN. The detection performance is evaluated under both channel models. The training E_b/N_0 is set at 8 dB, and the size of the mini-batch is 500.

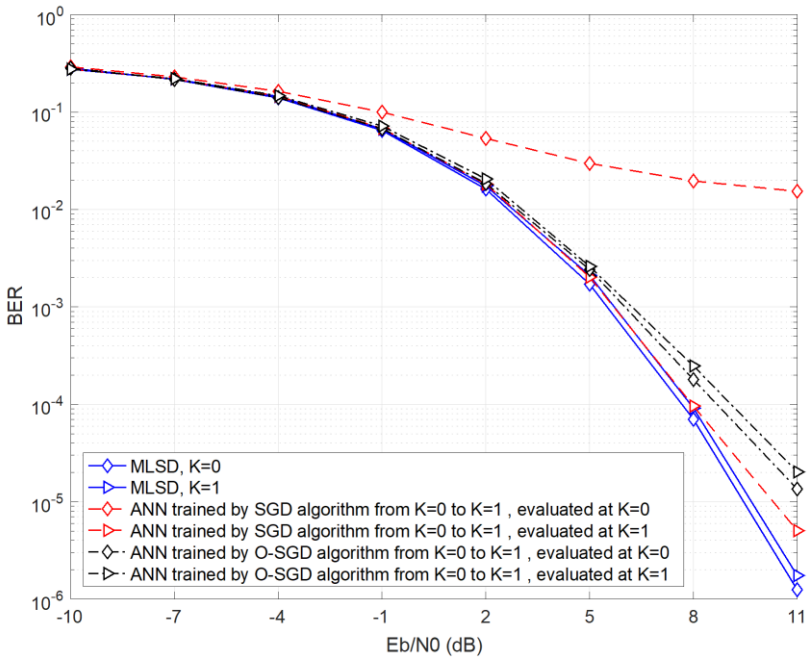


Figure 13: BER as a function of Eb/N0 for ANN-assisted MIMO signal detection. ANN is firstly trained under the Rayleigh fading channel (i.e. K = 0) and then trained for the Rician fading channel (i.e. K = 1) and evaluated under both channel models.

Figure 13 shows the average BER performance of the ANN-assisted MIMO receiver trained by either SGD or O-SGD algorithm. The baseline for performance comparison is the optimum MLSD. It is shown that the SGD algorithm fails to remember the previously learned knowledge, as the BER performance for the first task (i.e. K = 0) is around 9 dB away from MLSD at BER of 10^{-2}. Conversely, it achieves a near-optimum performance for the second channel model (i.e. K = 1). The gap between MLSD and SGD is less than 1 dB at BER of 10^{-5}. On the other hand, the proposed O-SGD algorithm shows promising learning capabilities for both tasks. The performance gaps to the optimum MLSD are 1.2 dB and 1.4 dB at BER of 10^{-4} for K = 0 and K = 1, respectively. It is also observed that the first task slightly outperforms the second task by around 0.5 dB at BER of 10^{-4}.

Experiment 3: In this experiment, ANN-assisted MIMO receiver is trained under a mix of multiple Rician fading channel models with K randomly varies in the range of [0,5] by using either SGD or O-SGD algorithm. Besides, the training settings remain unchanged, as we introduced in the previous experiments.

Figure 14 shows the average BER performance of the ANN-assisted MIMO receiver trained by either SGD or O-SGD algorithm. The baseline for performance comparison is the optimum MLSD. It is shown that the proposed O-SGD algorithm achieves promising detection performance under all the selected

channel models (i.e. K = 0, 1, 2, 3, 5). The gap between MLSD and O-SGD is less than 1 dB. By contraries, SGD fails to conduct a good signal detection, specifically at a high SNR regime. The gap to the optimum MLSD is around 2.5 dB at BER of 10^{-4} for K =0 and K = 1, and more than 5 dB for the other three channel models. This phenomenon indicates that the proposed O-SGD algorithm is able to achieve promising performance when multiple tasks are learned simultaneously.

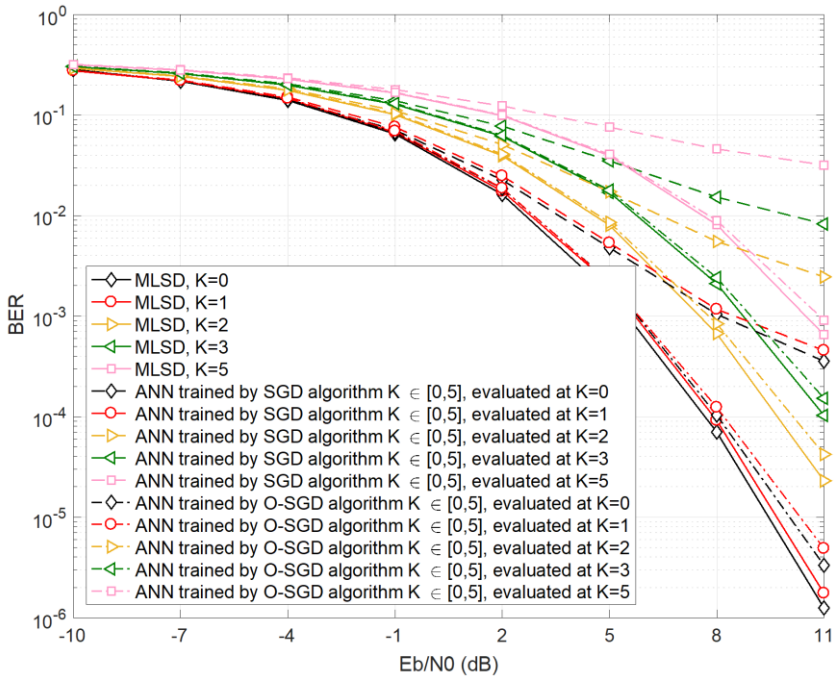


Figure 14: BER as a function of Eb/N0 for ANN-assisted MIMO signal detection. ANN is trained under a mix of multiple Rician fading channel models in the range of K ∈ [0,5] and evaluated under a number of selected channel models.

It has been shown in our work that O-SGD can discover the orthogonality between the current training epoch and previous training epochs and update the neural network by exploring the uncorrelated components among different training tasks. Simulation results have shown that the proposed O-SGD algorithm significantly outperforms the conventional SGD algorithm under multiple channel models.

The O-SGD algorithm has recently been employed in the deep end-to-end learning for uplink multiuser-MIMO joint transmitter and non-coherent receiver design (JTRD) in fading channels [15]. As depicted in Figure 15, a deep neural network (DNN) is utilised to detect multiuser-MIMO signals in a non-coherent manner, i.e. no knowledge about **H** is available at the receiver side. To associate such a detection, users (transmitters) must have their codebooks jointly designed and optimised. End-to-end learning is therefore invoked. Our research reveals that: 1) joint codebook design can significantly enhance the multiuser detectability; 2) the gain of joint codebook design is related to the time-domain degrees of freedom that are available for the codebook optimisation; 3) end-to-end deep learning could find us an efficient way to find the (near-)optimum random codebook, which can be hardly found in conventional engineering methods; 4) the use of O-SGD can offer more accurate training result than the conventional SGD algorithm. In addition, it is found that the conventional Xavier initialization method is not suitable for the end-to-end multiuser-MIMO network learning, as it can cause the user biased-convergence problem at the training stage. Therefore, we proposed a novel DNN initialization method, particularly for the multiuser end-to-end learning, with which the biased-convergence problem gets perfectly resolved.

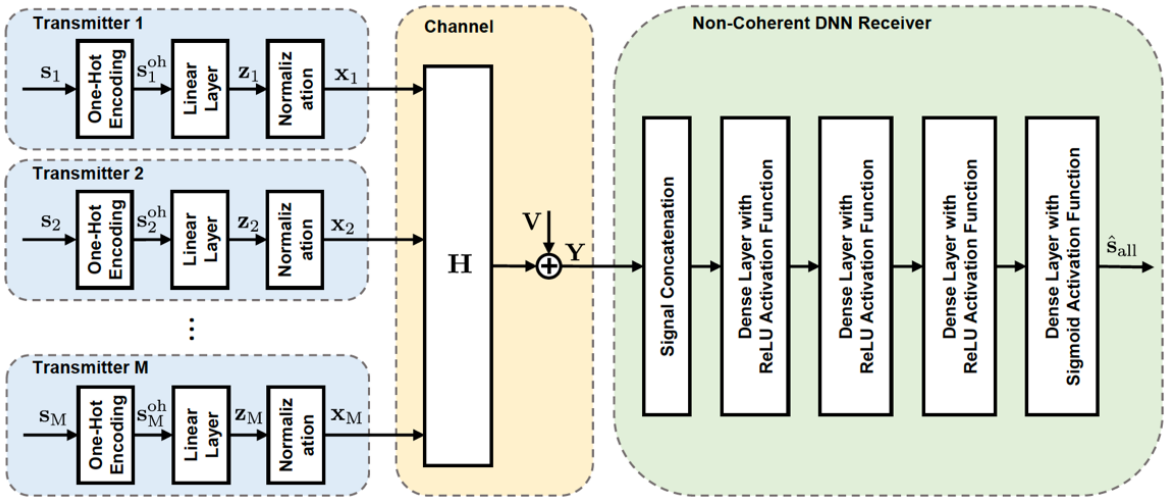


Figure 15: Block diagram of the proposed JTRD-Net approach.

To showcase the significance of our contribution, we plot the average BER performance as a function of the average SNR in Figure 15. Simulation results were obtained by taking the average of sufficient Monte Carlos trials in correlated MIMO Rayleigh-fading channels. The baseline for performance comparison includes conventional minimum mean-square error (MMSE) or maximum-likelihood detection (MLD) algorithms with the MMSE channel estimation. It can be observed that the proposed approach (i.e. JTRD) significantly outperforms those coherent detection algorithms by up to 5 dB in SNR. This is not so surprising as: 1) the coherent detection algorithms suffer performance degradation from the channel estimation error, and 2) the coherent detection algorithms perform optimisation only at the receiver side and thus do not take advantage of the transmit-side degrees of freedom. Moreover, the deep-learning-based optimisation could learn to exploit the channel correlation behaviour at the training stage. This is the extra gain deep learning could really enjoy. Finally, it is worthwhile to highlight that the showcase is about the optimisation of short block-length (e.g. two information bits per burst). This means that the proposed approach is ideally suitable for the URLLC use cases.

2.4 URLLC for task offloading

In this section, our contribution to MEC task offloading is presented. First, the transmission modes of downlink have been investigated, and then a novel task offloading model has been proposed.

2.4.1 On URLLC, downlink transmission modes for MEC task offloading

The use of multi-access edge computing (MEC) for mobile computing-task offloading has found many, and still increasing applications such as connected automated vehicles [16], industrial automation [17], and virtual/augmented reality [18]. The basic principle is to allow Mobile Computing Devices (MCDs) to offload their computation-intensive tasks to the MEC through cellular radio uplink, with the aim of trading off the communication overhead for mobile computing-power saving as well as the overall computing-latency reduction. Current research activities in this domain mainly focus on the uplink procedure, which includes MEC selection, uplink resource allocation, as well as multiuser and multi-task management. Their primary objective is to minimize the overall computing latency as well as MCD's energy consumption based upon the hypothesis of ultra-reliable and extremely low-latency downlink communication for the feedback of computing outcomes; for instance, most of the published results were based on theoretically zero-latency and 100% reliability for the downlink when conducting their optimizations [19]. This hypothesis is driving the extreme physical-layer design for the downlink, which has to consider various factors such as very limited time-domain degrees of freedoms (DoF), short messages, as well as the random time of message arrival. While the uplink research is still going

on, it is also the right time to work towards the hypothesis of ultra-reliable low-latency communications (URLLC) downlink.

The primary objective of URLLC downlink design is to find latency-constrained transmission protocols that can achieve a very low outage probability. Earlier contributions in the URLLC domain have suggested the use of multiuser message aggregation and joint encoding. It is anticipated that the message-aggregated encoding technique could bring significant improvements to the reliability by leveraging the coding gain and channel frequency diversity gain at the price of decoding complexity at MCDs, medium access control (MAC) complexities, as well as security and privacy concern. Nevertheless, it remains unclear how the coding gain and diversity gain behave under various constraints of the temporal DoF; and how the lower MAC protocol should be designed to support the extreme use of available DoF in the time, frequency and spatial domains.

In this subsection, we strive to initially answer the above questions through numerical study. Our investigation is based on orthogonal frequency-division multiplexing (OFDM) systems, appreciating their wide applications and recognized advantages in wireless communications. Specifically, this work relies on the Polyanskiy-Poor-Verdu formula on the finite-length coded channel capacity-bound [20], which is extended, in our work, from the quasi-static fading channel to space-time coded OFDM systems with the channel frequency-selectivity. The research focus is on the downlink protocol, with the uplink procedure being translated into the time or bandwidth uplink budget. The numerical results reveal that the message-aggregated encoding technique contributes considerable reliability enhancement to the downlink. However, our theoretic analysis also shows that due to the channel dispersion of frequency selective channels, the coding gain contributes few enhancements to the reliability. With respect to the uplink-downlink duplexing mode, time-division duplexing (TDD) outperforms frequency-division duplexing (FDD) in terms of the channel frequency diversity gain. On the other hand, FDD takes advantage of having more temporal DoF in the downlink, which can be translated into spatial diversity gain through the employment of space-time coding (STC).

Consider an access point (AP) connected with a set of MCDs, with each having a single receive-antenna, through the OFDM air interface with the total signal bandwidth $\boldsymbol{B_s}$. MCDs are uploading their computing tasks to the MEC through AP. The time and bandwidth consumption of the uplink communication are τ_{ul} and B_{ul}, respectively. Denote τ_{max} to be the overall latency constraint for each task offloading, and $\boldsymbol{\tau_{mec}}$ to be the time consumption at the MEC. AP has to complete the downlink procedure within the time constraint: $\boldsymbol{\tau_{dl} \leq \tau_{max} - \tau_{ul} - \tau_{mec},}$. The bandwidth that can be utilized for the downlink is: $\boldsymbol{B_{dl} < B_s}$. The setup of τ_{dl} and $\boldsymbol{B_{dl}}$ is different when the MEC offloading procedure operates in different duplexing modes. Specifically, they are given by

$$\boldsymbol{TDD: \tau_{dl} = \tau_{max} - \tau_{ul} - \tau_{mec} - \tau_{wait},}$$

$$\boldsymbol{B_{dl} = B_s}$$

$$\boldsymbol{FDD: \tau_{dl} = \tau_{max} - \tau_{ul} - \tau_{mec},}$$

$$\boldsymbol{B_{dl} = B_s - B_{ul}}$$

where $\boldsymbol{\tau_{wait}}$ denotes the delay in TDD, during which the downlink waits for the uplink to complete their transmissions (cf. Figure 16 for an illustrative explanation). Note that we use common notations $\boldsymbol{\tau_{dl}}$ and $\boldsymbol{B_{dl}}$ for TDD and FDD mainly for the notation simplicity; and later on, they will serve as variables in the Polyanskiy-Poor-Verdu formula.

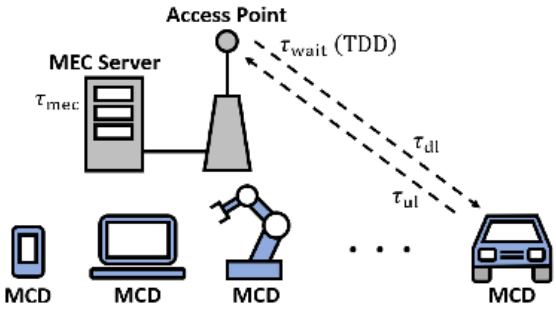


Figure 16: System model and latency component of the MEC task offloading.

Computing outcomes of the MEC are represented in the form of short messages, say K bit/message. For every downlink transmission, the MEC will generate $N(>0)$ short messages, where N is a random integer with its probability distribution determined by the MEC capacity and upper-layer protocols. When AP encodes N messages together and sends them to all MCDs in one go, the downlink data-rate is given by:

$$R = \frac{(N)(K)}{(\boldsymbol{\tau_{dl}})(B)}$$

where $\boldsymbol{B}$ is the signal bandwidth allocated for the $\boldsymbol{N}$ messages ($\boldsymbol{B \geq B_{dl}}$). Defining $\bar{\boldsymbol{R}}(\boldsymbol{\tau_{dl}}, \boldsymbol{B}, \boldsymbol{N})$ to be the instantaneous maximum achievable-rate for the downlink, the outage probability is measured by

$$\boldsymbol{p_{out} = PROB\ (\bar{R}(\tau_{dl}, B, N) < R)}$$

Here, each AP-to-MCD channel is assumed to be i.i.d., and thus they have the identical outage probability in fading channels. Moreover, it is perhaps worth noting that pout is the outage probability of every single AP-to-MCD link. In the case of extreme URLLC, AP normally has no temporal DoF to offer retransmission, and thus a single-link outage probability is more meaningful than a system outage probability.

Initial numerical evaluations are carried out here to evaluate the performance of both FDD and TDD. Due to the strict latency requirement, ΔB is considered to be 30 kHz. Moreover, $\boldsymbol{T_{cp}}$ is considered as one-eighth of the OFDM symbol duration. $\boldsymbol{B_s}$ is considered as 99.84 MHz, i.e. 3328 subcarriers. For TDD, each MCD is allocated 104 subcarriers. For FDD, the downlink bandwidth $\boldsymbol{B_{dl}}$ is 15.36 MHz, i.e. 512 subcarriers, while each MCD is allocated 16 subcarriers. For FDD, $\boldsymbol{\tau_{dl}}$ = 1 ms, while for TDD, $\boldsymbol{\tau_{dl} + \tau_{wait}}$ = 1 ms. Consider a pessimistic case in TDD, where the downlink needs to wait for one uplink transmission time interval (TTI) to start the transmission. The uplink TTI is considered to be 7 OFDM symbols as in the long-term evolution (LTE) system, so $\boldsymbol{\tau_{wait}}$ ≈ 0.25 ms. The length of one downlink message $\boldsymbol{K}$ is 32 bytes, i.e. 256 bits, as in the URLLC requirement. The evaluations are structured into two experiments.

Experiment 1: The objective of this experiment is to investigate the reliability improvement of adopting multiuser aggregation. Fig. 3 shows the operating SNR (the SNR to achieve 10^{-5} outage probability) of both FDD and TDD. It is shown that multiuser message aggregation can bring significant reliability improvement. When $\boldsymbol{N}$ = 8, the operating SNR of FDD and TDD has a reduction of 20.9 dB and 13.5 dB, respectively, it is also shown that as $\boldsymbol{N}$ increases, the decrement of operating SNR decreases gradually. This is because the frequency diversity order $\boldsymbol{M}$ is determined by the channel tap delay and is much smaller than the number of subcarriers $\boldsymbol{M}$. As $\boldsymbol{M}$ increases linearly to $\boldsymbol{N}$, the frequency diversity gradually approaches its limit, and the reliability improvement increment per MCD decreases gradually. In this case, TDD's frequency diversity order is closer to the limit, and its frequency diversity gain should be less significant than FDD. This phenomenon can also be observed in Figure 17.

Moreover, TDD outperforms FDD in terms of reliability. Although their performance gap keeps decreasing as N increases, TDD's operating SNR is still 13 dB lower than FDD when N = 8. This is

because TDD has significantly more frequency diversities than FDD. When N = 6, the operating SNR of TDD is already lower than 15 dB. While for FDD, the operating SNR is 26 dB when N = 8.

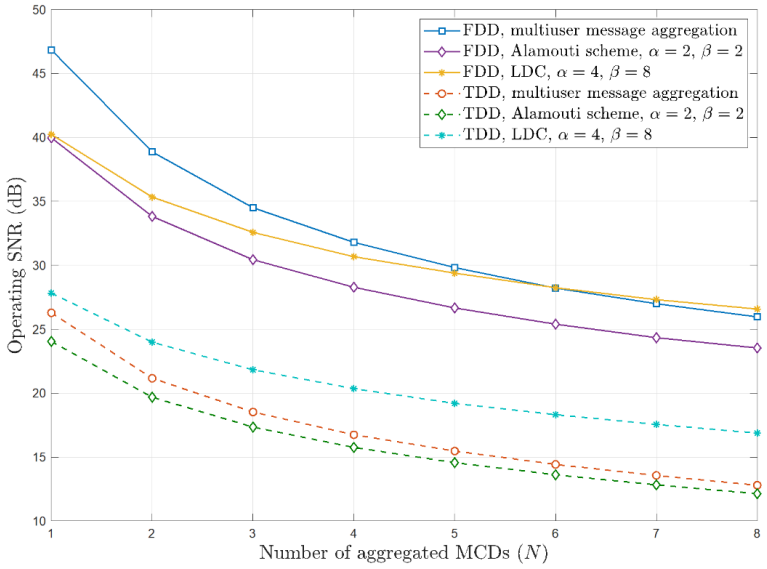


Figure 17: The operating SNR (the SNR to achieve 10^{-5} outage probability of FDD and TDD when adopting multiuser message aggregation and STC).

Experiment 2: The objective of this experiment is to investigate the behaviour of spatial diversity gain when adopting STC. Two STC schemes are considered here: Alamouti scheme where the transmitter diversity $\alpha = 2$ and the rate-loss factor $\beta = 2$. Due to temporal DoF payment, as well as the LDC scheme where $\alpha = 4$ and $\beta = 8$. It is shown in Figure 17 that when adopting the Alamouti scheme, the operating SNR is improved for around 4 dB and 1 dB for FDD and TDD, respectively. These improvements are due to the extra frequency diversity order introduced by spatial diversity. Moreover, the reliability improvement for TDD is smaller than FDD. This is because TDD has less temporal DoF, which leads to more rate loss. Such a result also reveals that adopting the Alamouti scheme cannot make significant reliability improvement for TDD. On the other hand, the performance of adopting LDC is showing obvious degradation compared to adopting the Alamouti scheme. This is because the high temporal DoF payment ($\beta = 8$) leads to severe rate-loss. For TDD, adopting LDC even leads to 4 dB performance degradation compared to not adopting STC. Such a result shows that high temporal DoF payment cannot improve the reliability of URLLC transmissions. Moreover, when adopting STC, TDD still outperforms FDD in terms of reliability.

This subsection has presented the fundamental behaviour of coding gain and diversity gains under URLLC performance requirements, as well as a comparison between duplexing modes for MEC downlink transmissions. For this purpose, the Polyanskiy-Poor-Verdú formula on the finite blocklength coded channel capacity bound has been extended from the quasi- static fading channel to the frequency selective channel. Through numerical analysis, it was found that multiuser message aggregation can significantly improve reliability. However, theoretical analysis reveals that these improvements are contributed by frequency diversity gain alone, without coding gain. When exchanging the temporal DoF for spatial diversity through adopting STC, the reliability improvement depends on the temporal DoF payment. For low temporal DoF payment, the downlink reliability can be improved. But for high temporal DoF payment, adopting STC could lead to a negative effect on the reliability. Moreover, in all numerical examples, TDD significantly outperforms FDD (around 10 dB or above), taking advantage of extra frequency diversity. Further details of this piece of work can be found in our publication [15].

2.4.2 Correlation-based dynamic task offloading for user energy-efficiency maximization

Task offloading to Multi-access Edge Computing (MEC) has emerged as a key technology to alleviate

the computation workloads of mobile devices and decrease service latency for computation-intensive applications. Device battery consumption is one of the limiting factors that need to be considered during task offloading. In this work, multi-task offloading strategies have been investigated to improve device energy efficiency. Correlations among tasks in the time domain as well as task domain are proposed to be employed to reduce the number of tasks to be transmitted to MEC. Furthermore, a binary decision tree-based algorithm is investigated to jointly optimize the mobile device clock frequency, transmission power, structure and number of tasks to be transmitted. MATLAB based simulation is employed to demonstrate the performance of our proposed algorithm. It is observed that the proposed dynamic multi-task offloading strategies can reduce the total energy consumption at the device along various transmit power versus noise power point compared with the conventional one.

With the continuous development of mobile communication technology and the rapid development of mobile Internet, mobile terminals represented by smart phones, tablet computers, laptops, and smart assistants are now widely used. But the mobile terminal receives limiting factors such as volume, weight, performance, power, etc. Its working ability is still in a serious and tedious state, which cannot meet the increasing demand of people. Although the mobile terminal has made great progress in hardware technology (for example, the continuous replacement of CPU/GPU, the continuous improvement of the chip manufacturing process from 28 nm to 14 nm to the current 7 nm, 5 nm etc.), it is still far from what people need. Moreover, with the emergence of new concepts such as autonomous driving, telemedicine, and Industry 4.0, which need ultra-reliability, low latency, ordinary equipment is unfortunately unable to support the requirements. Meanwhile, with the emergence of machine learning, artificial intelligence and other emerging technologies, the rapid development of image recognition, speech recognition and other applications, virtual reality and augmented reality game applications are emerging endlessly. The operation of these applications requires a large number of computing resources and storage resources, and they are all computationally intensive applications at a time. Due to the limitation of mobile terminals or some other devices, when computationally intensive applications are running on smart terminals, the endurance of the terminal and the performance of the application are very problematic. How to solve this problem of resource limitation and energy consumption has become a huge challenge today.

Most of the literature suggests changing the task allocation method, transmission power, clock frequency to optimize the task offloading algorithm. Some work proposed task splitting, which is a way to change the task structure to reduce the latency and improve the local device energy efficiency. However, they only split the task but did not consider the redundancy of sources. In our work, we propose to split the task into the smallest executable task, named as a unit, and then select the unit by using the correlation between them. Furthermore, both time and task domain correlation are considered in our work to select the necessary tasks to be processed. If any unit is found repetitive in either the time domain or task domain, it will be filtered out rather than offloaded to MEC. To further evaluate our ideas, MATLAB based simulations were designed. Considering a system contains four users and a single MEC. Each user has a different number of tasks, and different type of task has different size. Each user has its dedicated orthogonal channel to communicate to MEC. The channel bandwidth is set to 20 MHz, and the channel conforms to the Rayleigh distribution. The SNR of the user terminal is set to 10 dB, 20 dB, 30 dB, 40 dB and 50 dB in our simulations. The maximum computing rate in user's device is two × 10^9 cycles per second, and the maximum computing rate in MEC server is 20 × 10^9 cycles per second, the size of each task is between 1-3 M, and the number of processing revolutions they require is between 1500-4500 cycles and $k = 10^{-11}$. There are two latency requirements, which are 50 ms and 100 ms.

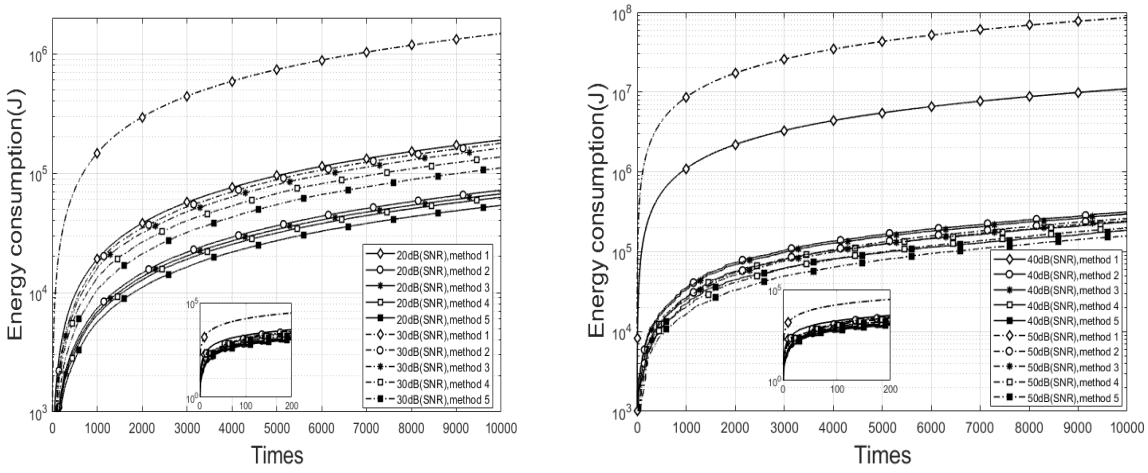


Figure 18: Energy consumption of proposed algorithms SNR = 20/30 dB (left), (b) SNR = 40/50 dB (right).

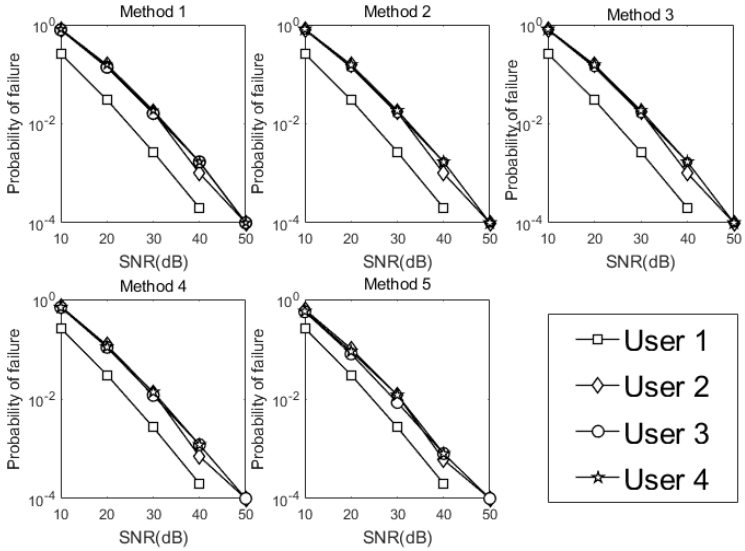


Figure 19: Probability of task processing failure under different methods and SNR.

In the experiment, two proposed approaches simulated with the other three baselines. Method 1 (Baseline 1): the most primitive task offloading algorithm, without considering any transmission power and other optimizations. Method 2 (Baseline 2): Further optimize the processing frequency and transmission power based on Method 1 [21]. Method 3 (Baseline 3): Based on Method 1 and Method 2, consider the impact of split tasks on task offloading [22]. Method 4: Consider the relevance of the task on the time domain to reduce the amount of data. Method 5: Base on Method 3, consider the correlation in the time domain and in different tasks to reduce the amount of data. Figure 14 (a) and (b) show the energy consumption of the five different algorithms on the local device when the SNR on the user terminal is 20 dB, 30 dB, 40 dB and 50 dB, respectively. In the same SNR, our proposed algorithms have better performs Fig. 5 shows the probability of task failure under different methods and SNR. From Figure 18 and Figure 19, with the same SNR, our proposed algorithms have better performs in energy-saving and reliability, and as the increase of SNR, with the same method, the energy cost may increase too, this is because we record that the energy consumption of the failed tasks that can be known in the stage of decision making is zero. The larger the SNR, the more tasks that can be processed, so it will also cause more energy consumption. Further details of this piece of work can be found in our publication [23].

2.5 Intelligent computation task offloading in beyond 5G networks

Beyond 5G networks (B5G) are expected to enhance the 5G capabilities towards the support of seamless wireless connectivity with reliability and latency guarantees. In the meantime, a multitude of mobile applications is emerging and gaining popularity, leading to a surge in computation demand. However, the mobile terminals (MTs) are in general resource-constrained by the limited physical size. To mitigate the burden from computation-intensive tasks with stringent URLLC requirements, multi-access edge computing (MEC) is becoming one key technology by provisioning computing resources at the edge in close proximity to MTs [24]. The trend of merging wireless communications and MEC motivates the focus of computation offloading in beyond 5G networks, as illustrated in Figure 20.

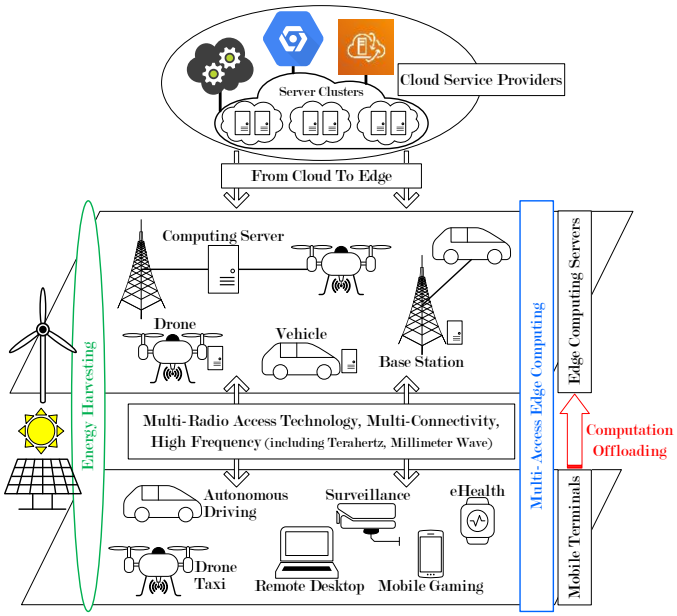


Figure 20: In beyond 5G networks, the computation performance for MTs can be potentially improved by offloading tasks to the edge computing servers for remote execution.

The integration of MEC into beyond 5G networks holds great potentials for improving the computation performance for MTs [25]. In such MEC systems, an MT decides whether the computation should be processed locally or offloaded to the edge computing servers for remote execution. The design of computation offloading policies remains daunting. Basically, the performance of computation offloading is bounded by two factors, namely, wireless communication and remote execution [26]. The wireless communication is performed by establishing a link between an MT and an edge computing server, controlling the input data size of the computation tasks to be offloaded, and selecting the frequency as well as the transmit power. For the remote execution, computation resource allocation and task scheduling at the edge computing servers are of essence for computing efficiency.

To date, there have been extensive studies on computation offloading in MEC systems. In [27], You *et al.* formulated convex optimization and mixed-integer problems for the optimal resource allocation in a multiuser MEC system based on, respectively, time-division multiple access and orthogonal frequency-division multiple access. A common drawback of finite-time optimization is that the computation offloading parameters are considered to be irrelevant under different MEC system states, and hence the long-term performance cannot be sustained. An infinite-time single-agent Markov decision process (MDP) was applied to investigate the dynamic computation offloading for a MEC system with wireless energy harvesting-enabled MTs in [28], where the Lyapunov optimization technique constructs an approximately optimal solution. To attain the optimal computation offloading policy, Xu *et al.* put forward an online reinforcement learning (RL) algorithm [29]. However, the centralized decision-making limits the scalability of most existing RL-based algorithms due to the huge decision space and the overwhelming information collection from a MEC system.

When there are multiple decision-makers in a MEC system, the degree of cooperation plays a vital role in the design of computation offloading policies. In the framework of a multi-agent MDP, an MT (i.e. a decision-maker) is regarded as an agent [30]. At each time point of observation, the MEC system is in a state, and with the computation offloading policy, each agent takes action, which is decision-making in terms of a vector of the computation offloading parameters. The MEC system responds by moving to a new state according to the probability distribution and sending feedback (i.e. the reward or cost signal) to each agent. Owning to the sharing nature of communication and computation resources, the actions from different agents are coupled. With complete cooperation among the agents, the overhead incurred from inter-agent communications is overwhelming and prohibits the autonomous local actions at each agent. Without any cooperation, the actions by each agent based on only the local information can be sub-optimal. Therefore, the objective of this study is to develop a distributed learning framework for optimized computation offloading beyond 5G networks, leveraging RL and supervised learning.

2.5.1 Challenges of computation offloading in beyond 5G networks

To facilitate efficient computation offloading in beyond 5G networks, the following inherent technical challenges need to be carefully addressed.

- **Offloading decision-makings**

In addition to being locally processed by the MT in the MEC systems, a computation task can be fully or even partially offloaded to and executed by the edge computing servers. Computation offloading is indeed a complex decision-making process, which accounts for the mobilities, the communication qualities, the computing capabilities and the resource availabilities. The challenge arises from how to manage the computation offloading process. Like the majority of studies, the offloading decision-making is formulated as a finite-time optimization problem given the communication as well as the computation resources. The objective is either to minimize the energy consumption at each MT while keeping the latency acceptable for a specific computation task or to find a trade-off between the two. However, the literature usually assumes a static MEC system environment. In beyond 5G networks, the large-scale deployments make the environment much more complicated, and it becomes more complex to re-formulate the offloading decision-making problem in accordance with the spatial and temporal dynamics. Moreover, repeatedly collecting the global network information is costly.

- **Resource allocation**

After offloading decision-making being made, the proper communication and computation resources have to be allocated accordingly, which is influenced by the edge computing sever selections, the task types and the latency requirements. Specifically, if the task is separable, the latency requirement violation resulted from resource deficit can be avoided through parallel processing the computation at multiple edge computing servers., which consist of the edge computing servers and the MTs with abundant computation and storage resources. On the contrary, the latency requirement is restricted by the resource abundance of the selected edge computing server and the MT. It is noteworthy that the edge computing server selection is highly dependent on the mobilities, because of which handovers occur. To ensure the reliability requirement of task offloading, an MT initiates the handovers when it roams out of the coverage of the associated edge computing server or the experienced communication quality deteriorates. In the beyond 5G networks, an enormous number of devices with heterogeneous communication and computing capabilities are deployed. To deal with the heterogeneity and the numerous computation tasks with distinct reliability and latency requirements, resource allocation asks for a flexible and distributed framework.

- **Distributed learning framework**

In line with the preceding discussions, this sub-section develops a distributed learning framework for the computation offloading problem beyond 5G networks, taking into consideration the relevance and the coupling among the decisions made by multiple MTs across the time. Different from the assumption of a static environment, a multi-agent MDP is adopted to capture the dynamics that

originates from the correlated uncertainties in a MEC system. The uncertainties range from the variations in wireless communications between the MTs and the edge computing servers (e.g. the time-varying channel gains and the constantly changing network topologies) to the randomness in task computations (e.g. the sporadic task arrivals and the unpredictable energy sources) [31].

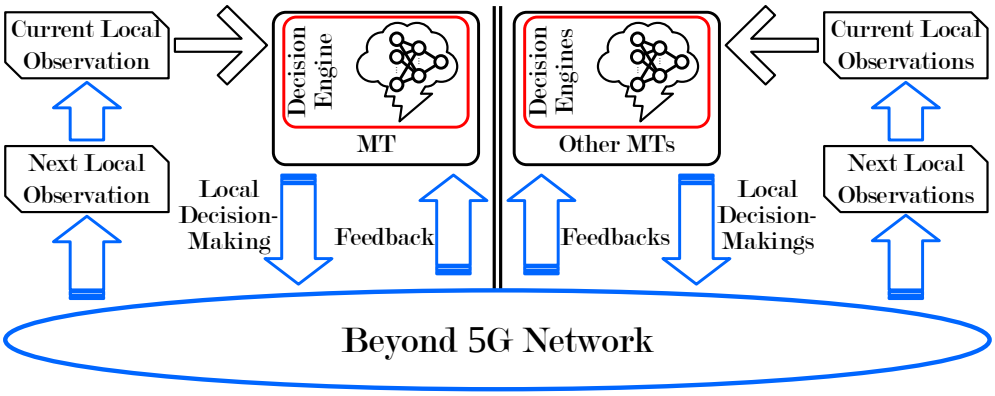


Figure 21: Proposed distributed learning framework. Each MT in the MEC system learns the optimal decision-making policy for computation offloading by making use of only the local observation information.

2.5.2 Proposed framework

From the definition of a multi-agent MDP, it is found that the obstacles to solving an optimal policy for an agent mainly lie in the state information sharing, the policy profile from other agents identifying the feedback and the knowledge of the global state transition probability. RL techniques can be explored to learn the optimal policy without a priori knowledge of the global state transition probability. However, on the one hand, the feasibility of full state information sharing among the MTs is not obvious due to the large-scale deployment of a beyond 5G network. On the other hand, the formulation of feedback received at each MT from the MEC system can be generally based on a combination of the consumed energy and the experienced latency, which is determined by the joint decision-makings from all MTs. Under this context, the multi-agent MDP falls into a stochastic game.

As depicted in Figure 21, a distributed learning framework is proposed, whereby each MT learns the optimal policy and does not rely on the state information sharing with other MTs in the MEC system. In the proposed framework, the local observation made by an MT at each current time point is composed of the current local state and the currently accessible contextual information. In particular, the local state encapsulates the local awareness of the wireless communication variations and the task computation randomness. The contextual information, which is the critical dimensionality of the local observation, allows an MT to behave in a distributed and autonomous manner. At each time point, the local decision-making of each MT refers to the selection of a number of computation offloading parameters, while the emitted feedback from the MEC system is a measure of the energy consumption and the latency of task computation, as previously noted. An MT is, therefore, enabled to independently learn an optimal computation offloading policy from the interactions with the MEC system, transforming the multi-agent MDP into a single-agent MDP.

A novel categorization of contextual information is introduced below.

1. Conjecture – The offloaded computation tasks are executed by the edge computing servers. It is feasible for an MT that the cumulative distribution information of the historical decision-makings by other MTs in the MEC system can be proactively retrieved from the central network controller (e.g. a base station (BS) controller). Accordingly, each MT is capable of forming and updating conjectures on the behaviours of other MTs during the independent learning processes [32].
2. Abstraction – It is easy to see that received feedback of an MT from the MEC system is related to specific joint decision-making in a specific global state. That is, the classification of the feedback values into a finite number of intervals is equivalent to the abstraction of the local states of other MTs with bounded regrets [33].

3. Prediction – To exploit the side information in computation offloading from the past (e.g. the experienced latency and the energy consumption during wireless communications), a recurrent neural network architecture can be incorporated into the decision engine of each MT for an accurate local prediction of the global state at each time point [34].

Yet, in most practical MEC systems, each MT still faces a potential obstacle from the explosion of local observation and decision-making spaces. A sufficient description of the uncertainties behind the local observations requires measuring various variables. The recent advances in neural networks inspire us to empower the decision engine of an MT with a deep neural network, which has been proven to be a universal function approximator to represent the higher dimensional space of local observations [35]. For an MT, the number of decision-makings grows exponentially as the number of computation offloading parameters increases. Depending on the structure of the computation offloading problem, a linear decomposition approach can be employed to break the per-MT single-agent MDP into a series of simpler single-agent MDPs, each decision-making space of which is much smaller [33].

2.5.3 Resource orchestration in computation offloading: a case study

In computation offloading, one interesting question is how to orchestrate the radio resources between traditional communication and the computation services. The resource orchestration is particularly challenging when taking into account the dynamics in the beyond 5G networks. As a case study, this sub-section applies the proposed distributed learning framework for resource orchestration in computation offloading. It is assumed that a mobile network operator (MNO) implements a beyond 5G network, where the radio access network (RAN) is connected to a resource-rich edge computing server via fibre links. The RAN sharing allows multiple wireless service providers (WSPs) to serve their respective MTs [36]. All MTs move across the discrete-time in the service region covered by the RAN according to a Markov mobility model. At each time point, the data packets and the number of computation tasks arriving at each MT follow a Poisson arrival process and a Markov chain, respectively. The data packet arrivals get queued until transmissions, while the arrived tasks at any current time point must be computed until the next time point. The duration of an interval (in seconds) between any two consecutive time points is set to be a constant.

At each time point, the WSPs compete with each other for the limited frequency bands from the MNO on behalf of the MTs. If being granted a frequency band after the band competition, an MT proceeds to decide the number of data packets scheduled for transmissions and the number of computation tasks to be offloaded to the edge computing servers. When the total number of data packet arrivals exceeds the queue size limit, overflows happen, resulting in packet drops. The remaining computation tasks arrived at the MT are processed by the local central processing unit (CPU). Given the MT mobilities, the time and energy consumptions during inter-BS handovers are fixed and cannot be optimized. The time consumption for sending back the computation outcomes from the edge computing services to the MTs is further neglected since the outcomes are typically much smaller than the input data size of a computation task. Thus, it is assumed that the latency for task computation is kept being constant. At each MT, the energy consumed by data transmissions and local CPU for task processing can then be calculated using the information of frequency bandwidth, channel gain to the associated BS, the total size of scheduled packets as well as offloaded tasks, required CPU cycles to accomplish one input bit of the computation task and local CPU-cycle frequency. The queueing delay and the overflows are the two important metrics to quantify the quality of traditional communication, for which the queue length and the number of packet drops are chosen.

2.5.4 Problem formulation and solution

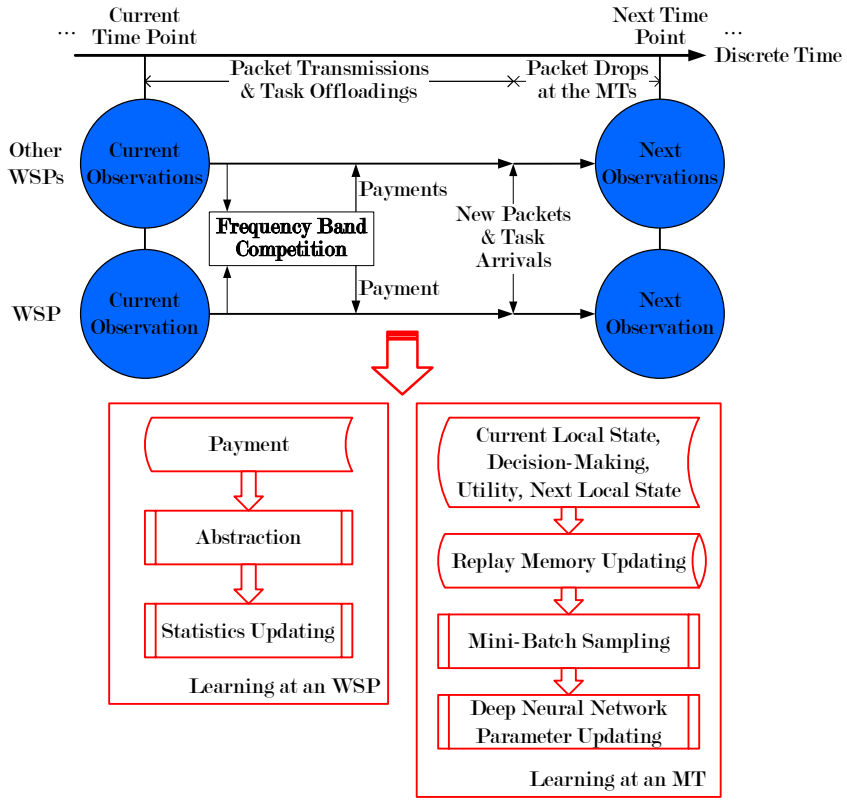


Figure 22: Flowchart of the online distributed deep RL algorithm.

The target of each WSP is to design a control policy, which optimizes the decision-making of frequency band competition, data packet scheduling and computation task offloading for its subscribed MTs under the global network states across the discrete-time, such that the long-term discounted payoff is maximized. When all WSP play the optimal control policies, the optimized long-term discounted payoff of the WSP is defined as a function of the global network states. At each time point, the immediate payoff received by the WSP is comprised of two parts: 1) the sum of weighted immediate utilities over the subscribed MTs; and 2) the payment to the MNO for frequency band utilization. The immediate utility of an MT is a reward signal of realized transmit energy consumption, local CPU energy consumption, queue length and packet drop after joint decision-making from all WSPs being performed under the global network state. The multi-agent MDP formulation of the resource orchestration problem is readily seen. The coupling in decision-makings and the sharing of limited frequency bands among the WSPs makes the solving of an optimal control policy extremely hard.

Based on the proposed distributed learning framework and the prior work as in [37], an online distributed multi-agent deep RL algorithm is developed, which decouples the decision-makings and enables the WSPs to independently learn the optimal control policy. More specifically, the local states of the MTs subscribed to other WSPs are replaced with the abstraction from the payment value classifications. The original multi-agent MDP then turns to be a single-agent MDP for each WSP, where the abstract long-term discounted payoff is a function of the local observations while the local observation at each time point includes the local state information of the subscribed MTs and the abstraction. To reduce the dimensionality of decision-making by the WSP, the abstract long-term discounted payoff is linearly decomposed into a series of per-MT long-term discounted utility functions and the long-term discounted payment. The linear decomposition brings another advantage of letting the MTs make local decisions of data packet scheduling and computation task offloading, while the WSP is responsible for frequency band competition only. The long-term discounted payment can be easily learned by exploring the conventional RL algorithms. To approach the per-MT long-term discounted utility functions, a deep Q-learning algorithm is developed to address the massive local state space and the unknown statistical network dynamics of an MT. Figure 22 shows a flowchart of

the developed online distributed deep RL algorithm for communication and computation resource orchestration employed by each WSP together with the subscribed MTs.

2.5.5 Performance evaluation

The developed distributed deep RL algorithm is evaluated using numerical experiments based on TensorFlow. The experiments simulate a RAN with 4 BSs distributed over a 2-kilometre by 2-kilometre square service region. In the beyond 5G network, 3 WSPs with each serving 6 MTs are assumed. Each MT uses a deep neural network with two hidden layers, and each layer contains 16 neurons. A total of 11 frequency bands of equal bandwidth 500 kHz is shared among the WSPs. The data sizes of a packet and a computation task are set to be 3000 bits and 5000 bits, respectively. The numbers of task arrivals are random integers less than six across the time points, while the duration between two consecutive time points is 0.01 seconds. Running 1 bit of the task requires 737.5 CPU cycles. CPU of each MT operates at the frequency of 2 GHz, while the maximum transmit power is 3 Watts. In the payoff function of each WSP, the weight of each MT is chosen as 1. For the sake of comparisons, the following two benchmark algorithms are considered, namely:

1. Channel-Aware – The MNO allocates the frequency bands to the MTs according to the channel gain information submitted by the WSPs. The priority of each MT is to transmit as many data packets as possible.
2. Queue-Aware – The WSPs compete for the frequency bands to minimize the queue lengths and the packets drops of the MTs.

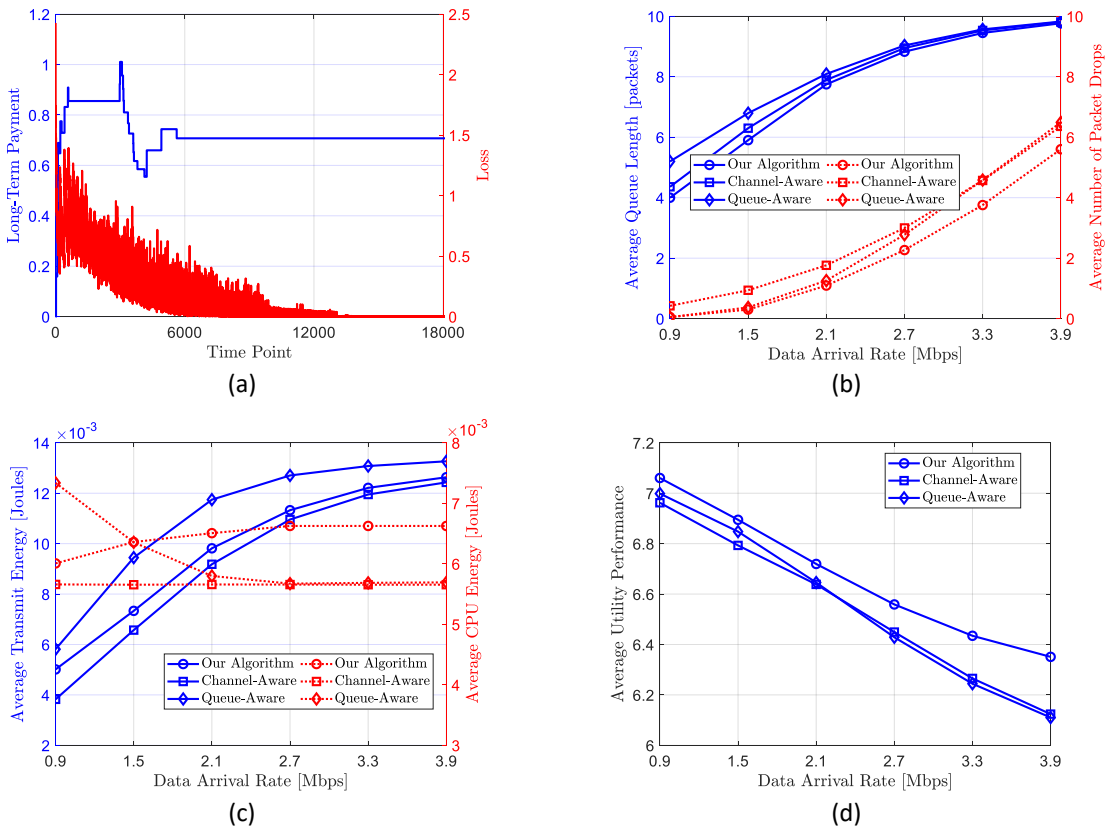


Figure 23: Convergence behaviour of the developed online deep RL algorithm for resource orchestration.

The experimental results are showcased in Figure 23, and Figure 23a examines the convergence in variations of the long-term payment for a WSP and the loss for an MT from updating, respectively, the abstraction statistics and the deep neural network parameters during the online learning process, where the data arrival rate of traditional communication for the MTs is 1.8 Mbit/s. The curves tell that

the developed online deep RL algorithm converges within around 13000-time points. It displays the average performance for the algorithm and the two benchmark algorithms across the discrete-time in Figure 23b, Figure 23c and Figure 23d by changing the data arrival rates. As the communication traffic load increases, it can be observed from Figure 23b that both the average queue length and the average number of packet drops increase, though Figure 23c shows that the MTs consume more transmit energy on average for data transmissions. Obviously, the increase in communication traffic load deteriorates the average utility performance for all three algorithms, as in Figure 23d. When employing the Queue-Aware algorithm, the increase in communication traffic load implies a higher probability for an MT to be allocated a frequency band for transmitting data packets and offloading computation tasks compared to the Channel-Aware algorithm. The average local CPU energy consumption hence decreases for the Queue-Aware algorithm. On the contrary, to reduce the packet drops by more data packet transmissions, the algorithm leaves more computation tasks to be locally processed, leading to an increase in average local CPU energy consumption. Overall, the algorithm achieves better average utility performance than the Channel-Aware and the Queue-Aware algorithms.

2.5.6 Conclusions and future directions

This study concentrates on the investigation of computation offloading in the beyond 5G networks. The dynamic uncertainties and the sharing of communication, as well as computation resources, necessitate the adoption of a multi-agent MDP to formulate the computation offloading problem when encountering multiple decision-makers. To solve the multi-agent MDP problems, a distributed learning framework is proposed, under which an MT is able to behave in a totally autonomous way with the awareness of the contextual information. As a case study of the communication and computation resource orchestration, it demonstrates that the developed online deep RL algorithm under the distributed learning framework achieves a significant performance improvement in terms of average utility for each MT.

One important issue that prevents the practical implementations of the proposed distributed learning framework is the time cost in training the developed algorithms. As can be seen from the case study, the convergence of an online learning algorithm is expensive. The following research directions deserve further investigation.

1. Off-Policy Learning Algorithm Developments: The distributed RL algorithms require online interactions with the network elements, which is the main reason slowing down the training process. For an MT, the learning algorithm training aims at finding an optimal probability matching between a state and the actions. To minimize the training cost, one idea is to develop off-policy learning algorithms that utilize the pre-collected online interaction data for offline training.
2. Transfer Learning for Training Acceleration: The training of learning algorithms can be potentially enhanced if the historical learning experience can be leveraged. When the communication and computation requests from the network exhibit significantly temporal and spatial correlations, transferring the parameters of a converged learning algorithm among the MTs dramatically speeds up the training process, allowing faster decision-making.
3. Heterogeneity among MTs: For the large-scale beyond 5G network deployments, there exist heterogeneities among MTs in the communication as well as computation capabilities and the contextual information availability. These heterogeneities have severe impacts on the stability of the training process if the MTs simultaneously learn the computation offloading policies. The development of distributed learning algorithms for such heterogeneous networks is demanding.

3 RAN transport

3.1 Phase-modulated RoF for efficient 5G fronthaul uplink

The use of Phase Modulation with Interferometric Detection (PM-ID) has been demonstrated in [38] for the uplink of a DSP-assisted analogue fronthaul link, employing a comprehensive model exhibiting a good match with experimentally measured performance for both single and multi-channel transmission. The use of phase modulation in the uplink results in improvement of the energy efficiency of the RU, as no electrical bias is required for the Phase Modulator (PM). Moreover, there is no need for a laser source to be present at the RU, with the possibility of the optical wavelength for the uplink being centrally controlled and distributed from the DU.

Figure 24 shows the proposed architecture. The downlink comprises a traditional external IM-DD link, with the incorporation of a remotely delivered optical carrier through an optical multiplexer. The optical carrier is de-multiplexed at the RU, where it is phase modulated via a PM. The modulated optical signal is transported through an optical link and received by the DU, where a Mach-Zehnder Interferometer (MZI) converts interferometric phase to amplitude before balanced photo-detection (only unbalanced photo-detection has been used in the experimental testbed due to component unavailability).

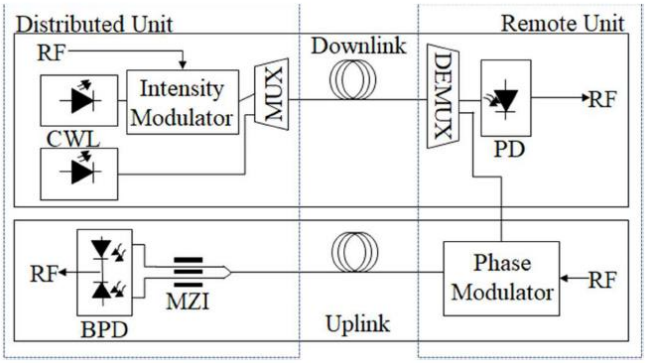


Figure 24: Proposed 5G RoF fronthaul (downlink and uplink).

3.1.1 Single-channel transmission via a phase-modulated RoF link

In 5G New Radio (NR), traditional Cyclic Prefix-Orthogonal Frequency Division Multiplexing (CP-OFDM) will continue to be employed as the physical layer transmission scheme for both the uplink and downlink, with Discrete Fourier Transform-OFDM (DFT-OFDM) considered for uplink modulation in some scenarios [39]. Several filtered variants of OFDM (F-OFDM) have been proposed for improving performance across a wide range of system metrics at the expense of increased complexity [40]. In this work, a comparison between experimentally measured and simulated EVM and Dynamic Range (DR) for single-channel transmission for both CP-OFDM and F-OFDM, with 64-QAM subcarrier modulation, has been demonstrated. The baseband signal creation takes place in MATLAB and includes the generation of frequency-domain QAM samples, pilot insertion for tracking changes in the channel frequency response, Inverse-Fast Fourier Transform (IFFT), Cyclic Prefix insertion and shaping filter (in the case of F-OFDM). The time-domain In-phase and Quadrature (I/Q) sampled signal is then downloaded into an Arbitrary Waveform Generator (AWG), which performs digital-to-analogue conversion and RF up-conversion. The photo-detected signal is amplified and captured with a Tektronix 72340DX oscilloscope and processed offline in MATLAB with time-correction, filtering (in the case of F-OFDM), FFT, frequency-domain equalization of the channel frequency response and demodulation, followed by EVM estimation.

The measured EVM versus input RF power for an MZI Full-Scale Range (FSR) of 6 GHz is compared to simulation results for both CP-OFDM and F-OFDM waveforms in Figure 25. For CP-OFDM, the

measured input power range (DR) is 23 dB, with respect to the 3GPP EVM specification of 8% in the case of 64-QAM [41], while for F-OFDM, the measured DR is 22.5 dB. The back-to-back (i.e. without an optical link) EVM for CP-OFDM at an RF input power of -10 dBm is approximately 2.5%.

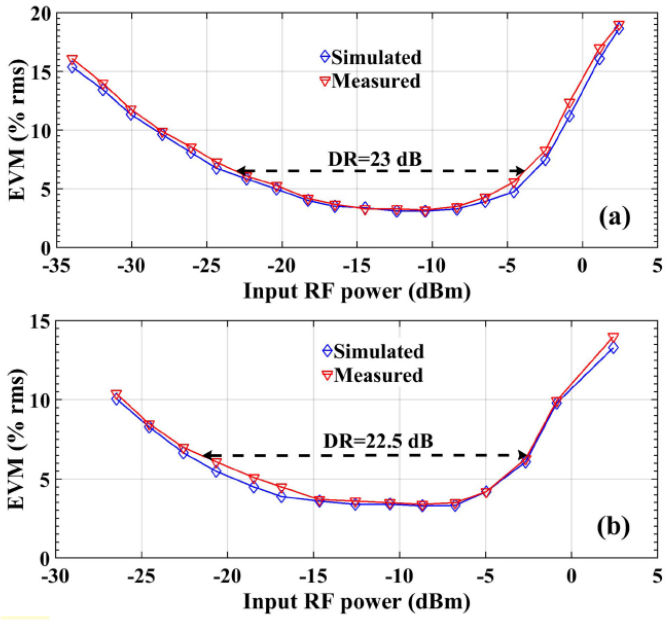


Figure 25: Measured and simulated EVM versus input RF power for (a) CP-OFDM and (b) F-OFDM signal. RF frequency is 2 GHz, and FSR is 6 GHz.

The comparison between measured and simulated EVM versus input RF power for an MZI-FSR of 10 GHz is shown in Figure 26. Similar EVM behaviour is shown, with the measured DR being 20 dB, compared to a DR of about 23 dB for an FSR of 6 GHz. The lower DR is a result of lower link gain at the RF frequency of 2 GHz for an FSR of 10 GHz (where the gain peaks at 5 GHz).

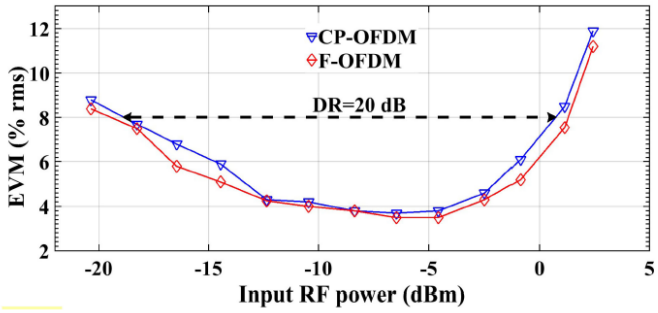


Figure 26: Measured EVM versus input RF power at 10 GHz FSR at an RF frequency of 2 GHz.

3.1.2 Multiple-channel transmission via a phase-modulated RoF link

The set-up for the experimental and simulation-based measurements of multi-channel transmission over the phase-modulated RoF link is shown in Figure 27. Similar to the single-channel experiments, the multiplex creation is carried out in MATLAB using a frequency-domain samples approach with NZ mapping, with the resulting time-domain digitized signal downloaded to AWG [42], which performs Single-Sideband (SSB) up-conversion of the multiplex to RF. For the simulation measurements, AWG is represented by an interpolation block followed by the SSB up-converter. At the receiving end, the received signal is captured by the oscilloscope so that it can be processed offline in MATLAB. The channels are de-multiplexed using a digital filter bank, the Cyclic Prefix for each channel is removed, and per-channel FFT and frequency-domain equalization are performed. Finally, following demodulation, the EVM performance of each channel is estimated.

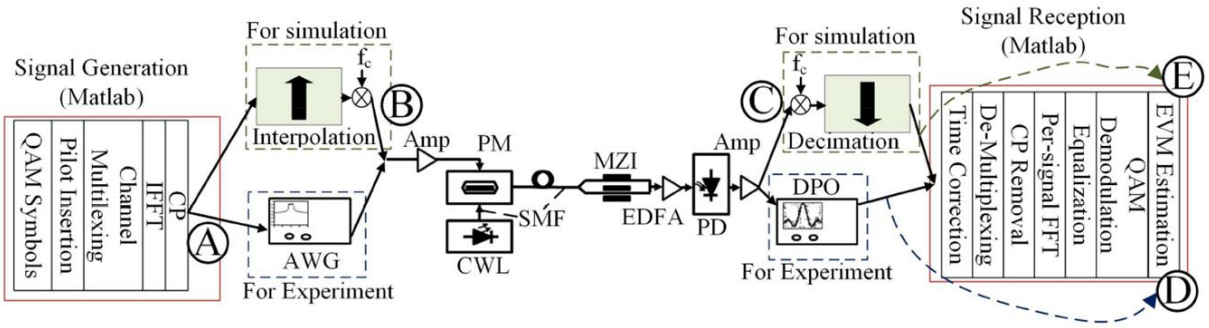


Figure 27: Measurement and simulation set-up for multi-channel transmission. A: Spectrum view of input multiplexes; B: Spectrum view of input to optical link (simulation); C: Spectrum view of output from the optical link (simulation).

Figure 28 to Figure 30 show the input and output to/from optical link spectra (points B and C) and the respective EVM (as a % of the RMS constellation value) per channel results (points D and E). Note that the best fit (dotted) traces that represent average trend behaviour are used or all EVM results to aid the visualization of how the EVM performance changes with frequency.

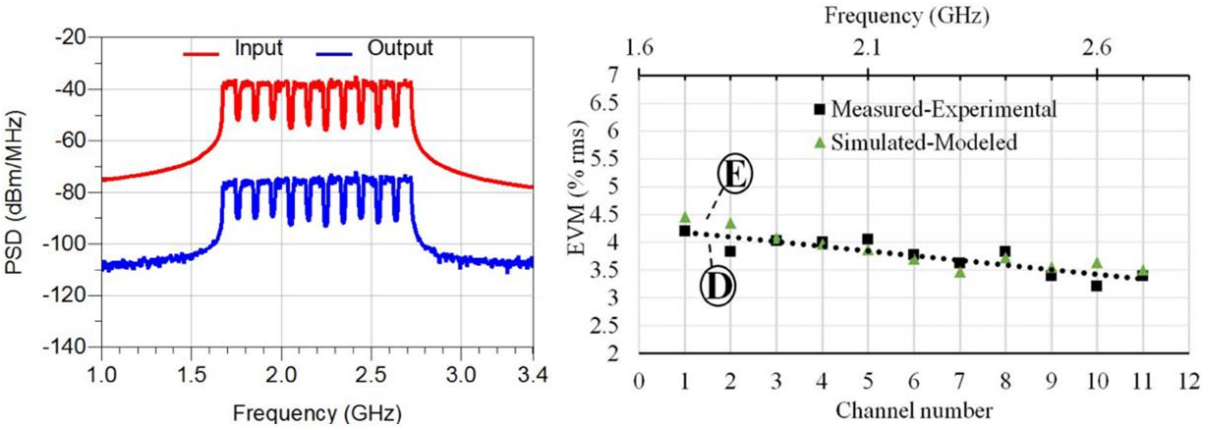


Figure 28: (Left) Spectrum view of input (point B in Figure 27) and output (point C in Figure 27) from the optical link for the 11-channel multiplex with FSR = 6 GHz and fc = 1.6 GHz (simulation). (Right) Measured-experimental and simulated-modelled EVM performance (Right) Measured-experimental and simulated-modelled EVM performance (points D and E respectively in Figure 27).

Comparisons between measured and simulated-modelled EVM results at an FSR of 6 GHz, with a multiplex consisting of 11 channels up-converted to an RF frequency of 1.6 GHz, is shown in Figure 28. Each channel has a subcarrier spacing of 60 kHz and a bandwidth of 72 MHz, resulting in an aggregate bandwidth (including pilot subcarriers) of 792 MHz (without frequency guard bands in-between channels) or more than 1 GHz (with frequency guard bands). The resulting aggregate data rate is approximately 4.3 Gbit/s. The specific number of channels and the bandwidth occupied by the multiplex have been chosen based on the bandwidth capabilities of AWG. There is a good match between the measured and simulated-modelled EVM results across all channels, and the EVM values are well within 3GPP specifications for 64-QAM. As expected, due to an increased RF gain around the half-FSR point at 3 GHz, the EVM traces show a reduction in EVM (corresponding to an increase in SNR) for channels closer to the FSR gain peak.

Figure 29 shows a simulation-based performance prediction at an FSR of 10 GHz with a larger aggregate bandwidth, 16-channel multiplex up-converted to an RF frequency of 3.1 GHz. Each channel has a subcarrier spacing of 120 kHz and a bandwidth of 144 MHz resulting in an aggregate bandwidth of 2.3 GHz and a user data rate in excess of 12.4 Gbit/s. Again, the EVM performance is well within the 3GPP limits for 64-QAM modulation and shows the expected dip close to the FSR gain peak at 5 GHz.

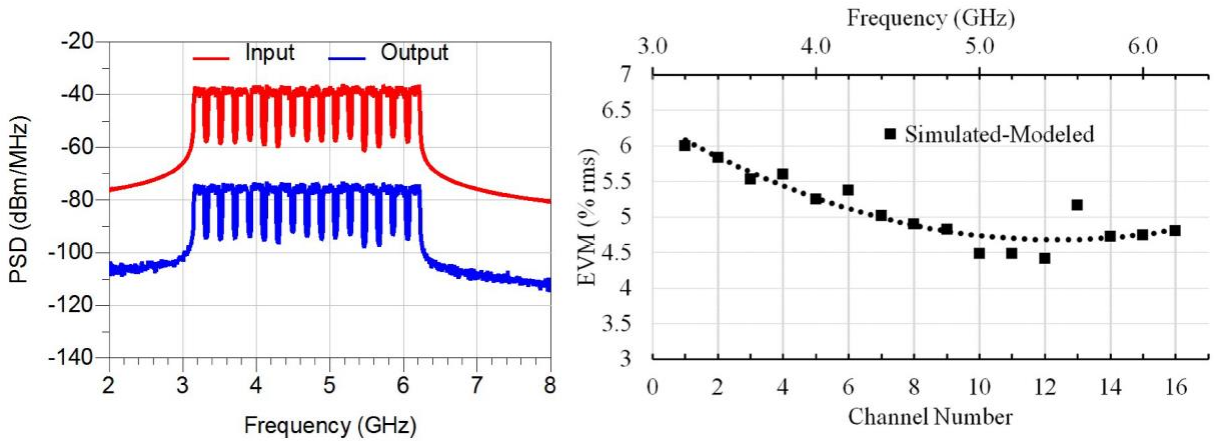


Figure 29: (Left) Spectrum view of input (point B in Figure 27) and output (point C in Figure 27) for the 16-channel multiplex with higher performance optical link, 20 km fibre span and with FSR = 10 GHz, fc = 3.5 GHz (simulation). (Right) Simulated-modelled EVM performance (point E in Figure 27).

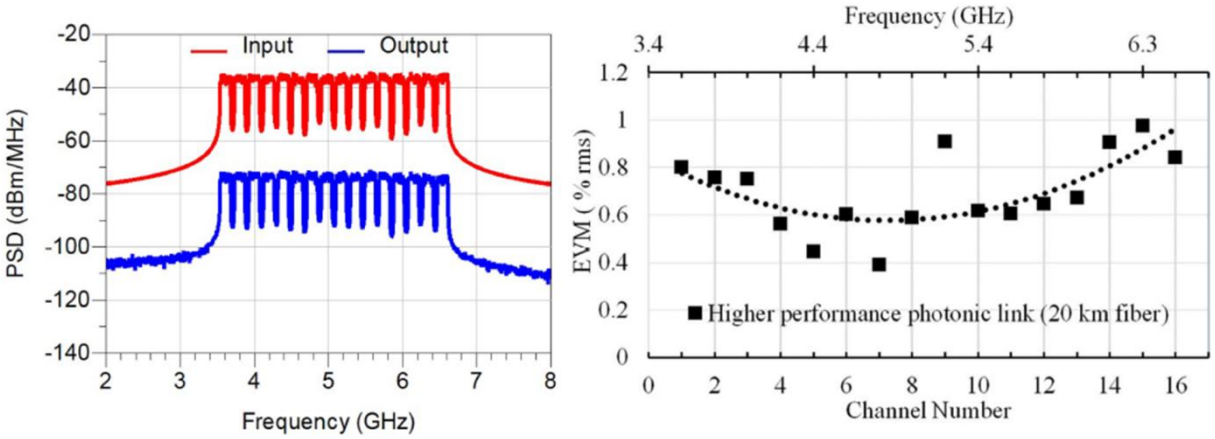


Figure 30: (Left) Spectrum view of input (point B in Figure 27) and output (point C in Figure 27) from the optical link for the 16-channel multiplex with FSR = 10 GHz and fc = 3.1 GHz (simulation). (Right) Simulated-modelled EVM performance (point E in Figure 27).

Figure 30 shows a simulation-based performance prediction using the same 16-channel multiplex and an FSR of 10 GHz. However, the multiplex has been up-converted to an RF frequency of 3.5 GHz, the optical link employs higher performance photonic components, and the optical fibre span has been increased to 20 km. The EVM performance is very good across all channels as a result of the higher RF gain of the link (due to increased input optical power from the Continuous Wave Laser, CWL). The multiplex is now centred approximately at the half-FSR gain peak point at 5 GHz, resulting in a dip in the average EVM trend (dotted line) approximately at the half-FSR point. This corresponds to Channel 8, with channels farther away from this point exhibiting progressively worse performance (as expected). Thus, in general, there is a clear potential for optimization in performance via appropriate RF frequency placement of multiplexes based on FSR employed by PM, while larger FSR values are advantageous for multiplexes with larger aggregate bandwidths.

3.2 DSP-assisted 5G and beyond fronthaul

A concept diagram of a next-generation mobile network is shown in Figure 31. The Core Network (CN) is connected through the backhaul to a Central Unit (CU), where some higher-layer protocols are executed. The F1 interface between the CU and the Distributed Unit (DU), connecting the Packet Data Convergence Protocol (PDCP) and Radio Link Control (RLC) layers of the protocol stack, has already been standardized by the 3GPP [39] and [43]. This transport network section connecting the CU to the DU is often termed the midhaul. The remaining RAN protocol stack processing is split between the DU and the Radio Unit (RU), with the transport network for this F2 interface being termed the fronthaul.

The 3GPP has not reached a consensus yet on the split point in the RAN functions in this part of the network, though possible options within the baseband Physical (PHY) layer have been proposed [39], [43], [44], [45], [46], [47], [48].

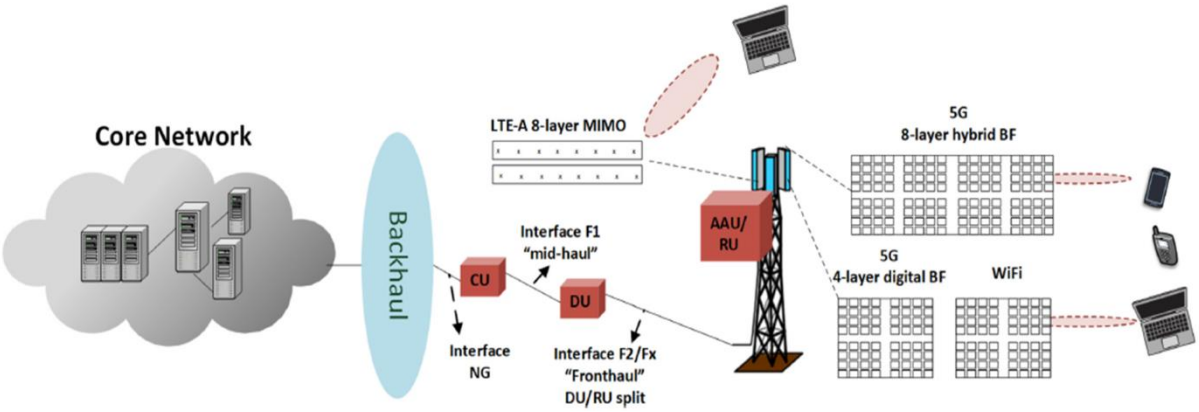


Figure 31: High-level functional description of the end-to-end 5G (and beyond) network, with the focus on the edge of the network (midhaul and fronthaul.

An important feature of 5G and beyond systems will be the increasing use of Multiple-Input Multiple-Output (MIMO)/massive MIMO (mMIMO) antenna systems, which has been shown in the form of an Active Antenna Unit (AAU) in Figure 31. Due to the complexity of handling and transporting signals by each antenna element within AAU [43] and [47], current arrays are partitioned into subarrays (or layers), with the requirement to transport a pre-coded signal for each subarray only. As a result, hybrid beamforming is used with some analogue phase/amplitude control, creating different possible beam directions for each subarray. For such an AAU, a fronthaul needs to transport different streams for the layers, as well as some control signals for amplitude and phase weights. With the new DU-RU digital fronthaul splits, 5G channel bandwidths up to 400 MHz and eight layers (with both bandwidth and number of layers expected to increase in the future), bit-rate requirements are expected to be between 16 Gbit/s and hundreds of gigabits per second, depending on the chosen split point within the RAN PHY layer [47]. Moreover, due to increased fronthaul latency and latency variations (packet jitter), the split point between the DU and RU has to be moved higher within the PHY layer. This means that functional splits that can provide the highest data rate reductions can also lead to higher latency constraints, impeding the use of distributed MIMO techniques.

As discussed in deliverable D5.1, the idea of analogue transport for the final part of the 5G fronthaul has gained popularity in recent times due to its high spectral efficiency. Unlike traditional analogue Subcarrier Multiplexing (SCM) techniques in fibre optic communications that lack flexibility and reconfigurability due to their reliance on microwave components, the use of digital processing in the DU and RU enables adaptation and scalability to variable signal bandwidths and multiplex sizes. It also enables the possibility of interworking with Software-Defined Radio (SDR) for orchestration and slicing of fronthaul resources. This type of DSP-assisted analogue fronthaul link employing a classical Mach-Zehnder Modulator (MZM)-based RoF topology (external Intensity Modulation-Direct Detection, IM-DD) in the downlink [42] has been presented in deliverable D5.1.

This section has been divided into two main parts:

In Section 3.2.1, comparisons between different DSP-assisted channel aggregation techniques in terms of computational complexity, processing latencies etc., are presented, along with performance comparisons in the form of Error Vector Magnitude (EVM) estimates for back-to-back, simulated and experimental optical links.

Section 3.2.2 discusses and demonstrates a flexible and efficient DSP-assisted mobile fronthaul that employs a combined multiplexing technique that enables a compromise for sampling rate requirements while maintaining low complexity and good performance.

3.2.1 Comparison of DSP-assisted techniques for the 5G and beyond fronthaul

In analogue RoF, radio signals can be transported in the form of a multiplex created via SCM techniques that use the different carrier frequencies of the radio signals or through translation to different intermediate frequencies (necessary for MIMO, when the radio signals are at the same frequency). Traditionally, systems employing microwave/RF up/down converters, amplifiers, filters, splitters/combiners, etc., have been used by neutral host providers with propriety equipment under bespoke, scenario-dependent setup conditions [49], [50], [51], [52]. This leads to a lack of deployment flexibility, adaptability and scalability, which are key attributes for the 5G and beyond RAN as it has to be able to meet differing requirements across a wide range of use cases such as Augmented Reality/Virtual Reality, gaming, and immersive applications for Industry 4.0 [43] and [53]. Most of these requirements arise from the need to accommodate variable 5G bandwidths, latencies, and numerologies[2]. Moreover, a system may include not only different long-term evolution (LTE) and 5G numerologies but also those employed by wireless local area networks (WLANs) in a Heterogeneous Network (HetNet) deployment. As a result, the type of DSP-assisted SCM/Frequency-Domain Multiplexing (FDM) approach for an analogue fronthaul that has been discussed so far is inherently flexible and is able to scale to different bandwidths, numerologies, and modulation formats.

In [42], the focus was on the simplification of receiver-side (RU) processing and on the proposed mapping technique, allowing the use of arbitrarily low sampling rates and analogue bandwidth at the receiving end by the introduction of limited analogue processing. However, a detailed analysis and performance comparison between different DSP-assisted FDM approaches is currently missing from the available literature. This type of analysis is very important due to the challenges facing digital transport fronthaul links with current and future mobile network generations.

In [54], two main techniques are analysed: one operating on time-domain samples and the other operating on frequency-domain samples. The latter is based on the multiplexing technique presented in [42], but this is the only similarity between the DSP-assisted approach in [42] and the techniques presented in [54]. The focal points of this work are the processing elements of the two FDM techniques and their effect on overall system complexity and performance. The computational complexity, sampling rates, processing latencies, and analogue performance in terms of EVM are considered to fully understand the relative advantages and disadvantages of each approach.

A conceptual view of the proposed DSP-assisted SCM/FDM architecture is shown in Figure 32. At the DU, digital samples of each stream are mapped onto their channels, which are then multiplexed digitally to form a composite FDM signal. After digital-to-analogue conversion, the signal is modulated on to an optical carrier for analogue RoF transport. At the receiving end, there may be some analogue processing, analogue-to-digital conversion and digital processing to recover the digitized samples of the transported streams.

The two types of processing techniques for channel multiplexing are shown in Figure 33. The frequency-domain samples technique is shown first, which multiplexes channels using frequency-domain samples and employs a single-IFFT operation to convert the frequency-domain multiplex into a time-domain waveform. Here, the term “single-IFFT operation” means that only a single IFFT is used to convert the multiplex into the time domain. The time-domain samples technique is shown next, which employs digital up-converters (DUCs) and combines a number of IFFT outputs (with each IFFT outputting a single time-domain channel) into a composite multiplex. Note that both techniques create a frequency-domain multiplex. Moreover, both techniques can easily adapt to changes in the RF/mmW frequencies that the transported channels need to occupy at the RU, and both are inherently flexible, i.e. the channels composing a multiplex can have different bandwidths and/or employ different modulation schemes. If the generation of mmW signals at the RU is required, an SCM/intermediate

[2] Numerologies are the different specified numbers and spacings of the subcarriers in the OFDM signal waveforms employed.

frequency (IF) RoF (SCM/IF-RoF) scheme can be employed with remote local oscillator delivery through optical heterodyning (as was demonstrated in [42]) or with electrical up-conversion at the RU [55]. In either case, dispersion-related effects are minimal.

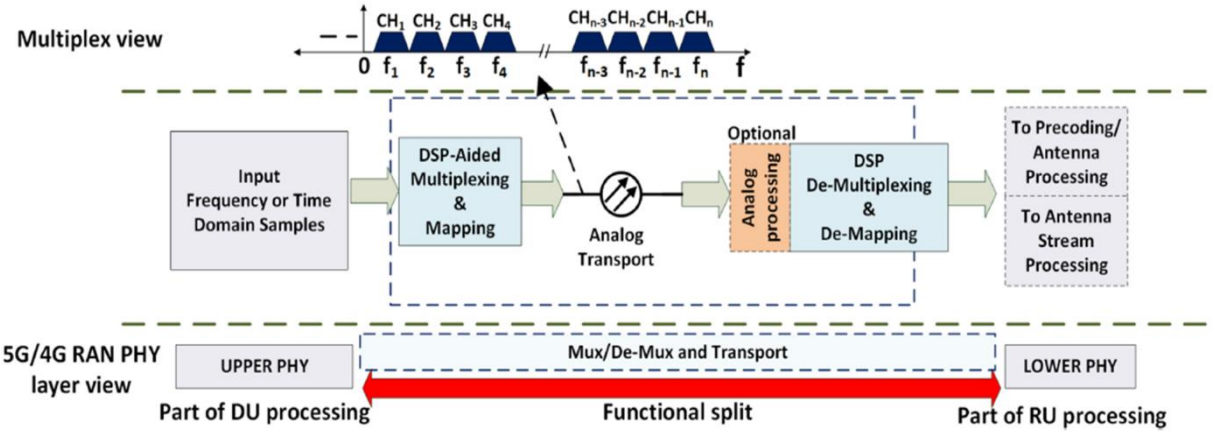


Figure 32: Conceptual view of the proposed DSP-assisted SCM architecture and the DU and RU processes. The DSP-assisted multiplexing part is further elaborated in Figure 33, and the de-multiplexing part in Figure 34.

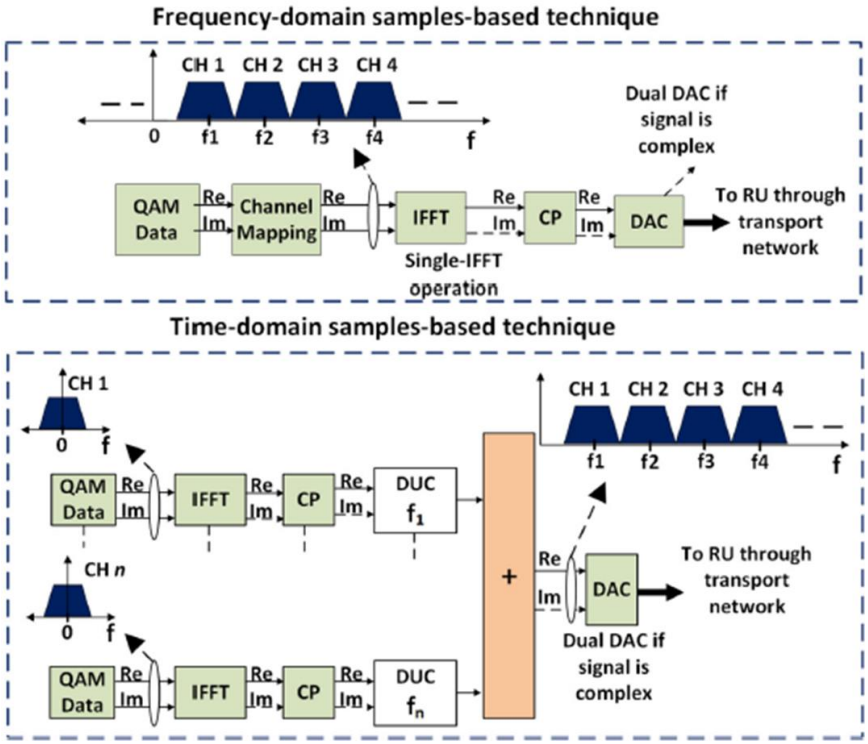


Figure 33: Functional depiction of the two multiplexing techniques. Note that one numerology is shown here for simplicity, but in both cases, the processes within each technique can be scaled to generate channels comprising different numerologies.

There are two more operational regimes that are interesting but have not been demonstrated in this work. The multiplexes can comprise different numerologies by appropriate control of sampling rates and can comprise a mixture of SSB and Dual Side-Band (DSB), with conjugate symmetry-derived channels (for the frequency-domain samples technique, some of this flexibility was demonstrated in [42]). A combination of the two techniques can also be envisaged, whereby each single IFFT process is used to multiplex several channels, but the outputs of each single IFFT are combined to form a "super-multiplex" through their own DUCs. This could be used to aggregate groupings of channels based on mobile operators, RAN technologies, or other relevant mobile network associations, such as network slices. Such a combined approach is discussed and demonstrated in Section 3.2.2.

Following analogue transport over the fronthaul, a number of de-multiplexing approaches are possible at the RU. For a fully digital approach, shown in Figure 34 (a), assuming the RU has sufficient sampling rate capabilities, the received signal multiplexes are converted the analogue directly to the digital domain through an ADC (note that some wideband filtering is usually employed prior to ADC). Each channel is then directly down-converted to baseband through a Digital Downconverter (DDC).

In the case of sampling rate limitations, the digital-domain processes can be preceded by a Track-and-Hold Amplifier (THA) and minimum analogue-domain filtering, as shown in Figure 34 (b) and described in [42]. Here, we consider only the basic digital techniques though more simplified, receiver structures are possible employing band-pass sampling of channels mapped into NZs such that channels can be "obtained" at predefined frequency locations (i.e. at predefined intermediate frequencies), as described in [42].

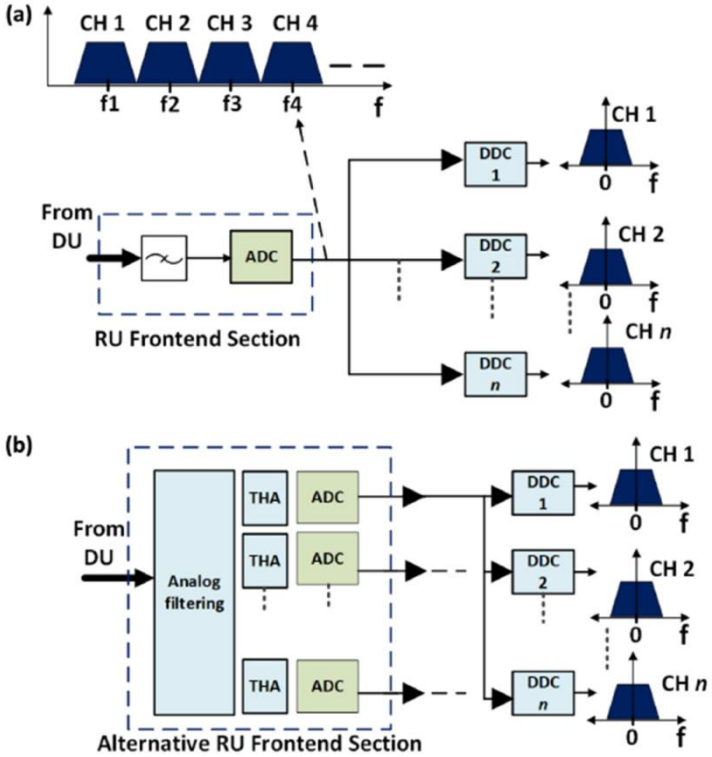


Figure 34: Functional depiction of different de-multiplexing approaches. (a) Fully digital and (b) with minimal analogue processing using a THA.

The time-domain samples technique complexity is principally determined by the DUC/DDC. The filtering/interpolation section of the DUC/DDC is a multi-stage implementation, consisting of a half-band filter, a Cascaded Integrator-Comb (CIC) compensator filter and a CIC interpolator. This is shown in Figure 35. The only difference between the DUC and DDC is in the ordering of the filtering stages. Note that this design represents a typical interpolation section in digital DUCs, although variations of this implementation are found (especially in the second stage filter, which is sometimes implemented as another half-band filter) [56] and [57]. For the work presented here, all three filters are linear-phase Finite Impulse Response (FIR) implemented in a computationally efficient poly-phase structure [58]. Complexity results assume a minimum-order approach: the passband ripple and stopband attenuation are chosen, and the order of each filter (and, therefore, its complexity) is a consequence of these choices. Furthermore, only integer interpolation/decimation factors are assumed. Finally, a digital quadrature mixer, fed by a numerically controlled oscillator, is used to up-convert or down-convert the signal (in DUC or DDC, respectively). The filter structure uses the multi-rate algorithm available in MATLAB [59].

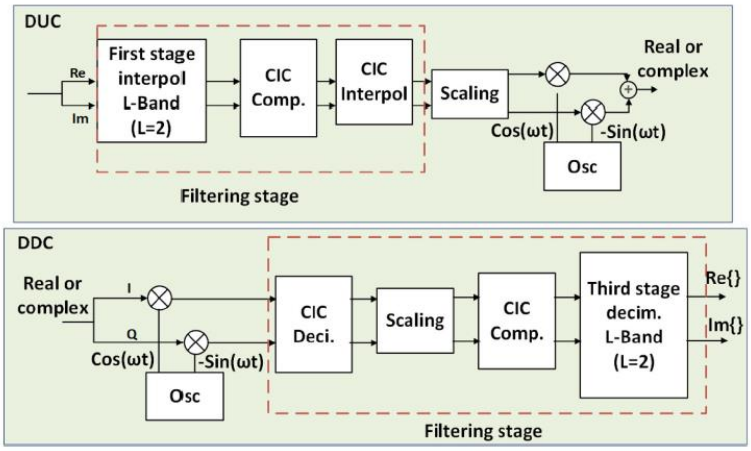


Figure 35: DUC (top) and DDC (bottom) processing stage.

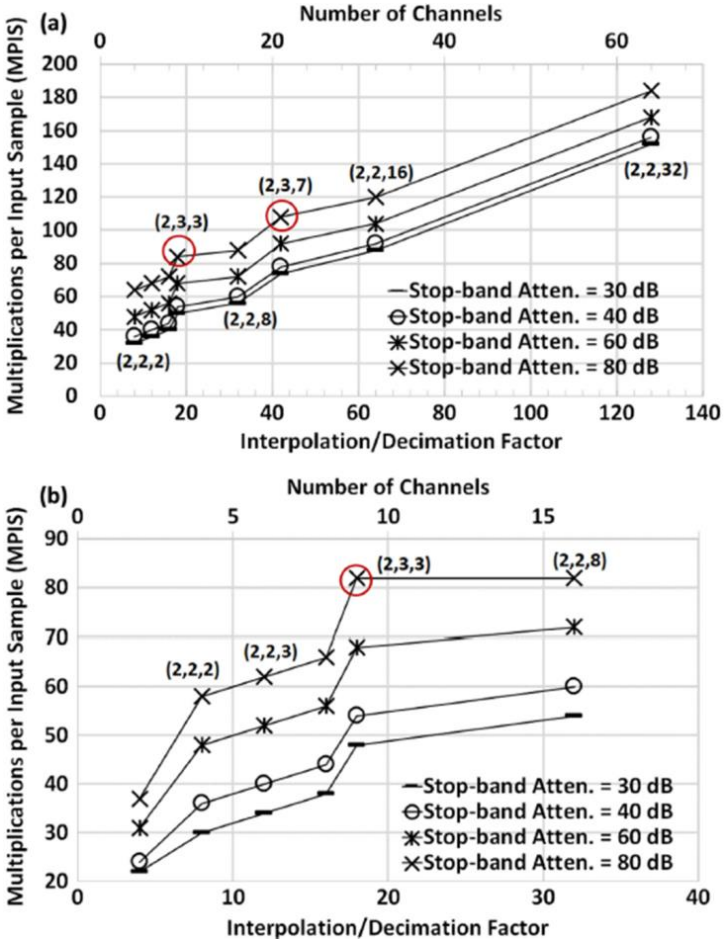


Figure 36: Computational complexity of time-domain samples approach, given as the number of Multiplications Per Input Sample (MPIS), of a single DUC/DDC stage for different oversampling factors and stopband attenuation factors, for a channel bandwidth (BW) of (a) 100 MHz and an IFFT size of 1024 and (b) 400 MHz and an IFFT size of 4096. Interpolation factors for each filtering stage are shown as annotations.

Figure 36 shows the MPIS of the time-domain samples approach for the filtering stages of the DUC (or DDC) for different oversampling/ interpolation factors and stopband attenuations, assuming a per-channel bandwidth of 100 MHz and 400 MHz. Note that for these results, the interpolation factor is varied in accordance with the number of channels to be multiplexed, but the given MPIS values are for a single DUC (or DDC). Thus, the total MPIS value would require scaling of the values shown in the figure by the number of DUC/DDCs (which will be equal to the number of channels in the multiplex).

Complexity scales approximately linearly with oversampling factor. In general, the overall complexity is minimized by assigning most of the interpolation/decimation to the CIC interpolator/decimator stages rather than the compensation stage. However, some choices of overall multiplication/division require assigning a higher value of interpolation/decimation to the intermediate compensation stage leading to deviations from the linear trend. The complexity also scales approximately linearly with stopband attenuation but is not generally affected by channel bandwidth.

The MPIS for the frequency-domain samples approach for the two different channel bandwidths are shown in Figure 37. The MPIS scales logarithmically with the number of channels, while larger bandwidths lead to increased complexity, as they require larger IFFT sizes. The assumed per-channel IFFT lengths are 1024 for 100 MHz bandwidth channels and 4096 for 400 MHz bandwidth channels.

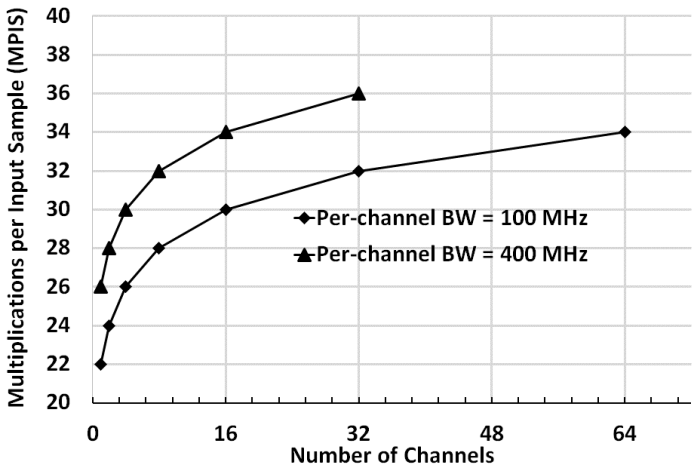


Figure 37: Computational complexity given as the number of computations per sample (MPIS) for different numbers of channels.

Figure 38 shows required sampling rates for both times- and frequency-domain samples techniques, normalized to the per-channel sampling rate. Two cases are shown: in the first, the channel spacing is smaller than the bandwidth of the guard bands provided by null subcarriers in the IFFT of the channel; in the second case, the channel spacing is larger than this. In the cases studied, the guard bands were approximately 25% of the channel bandwidth. While the time-domain samples technique has sampling rates that gracefully scale with the number of multiplexed channels, the frequency-domain samples technique requires significant adjustments to accommodate non-power-of-2 numbers of channels. Furthermore, for both techniques, some parameter adjustment needs to be made for larger channel spacing: either the DUC oversampling rate must be increased (time-domain samples technique), or the IFFT length must be increased to the next power-of-2 (frequency-domain samples technique). These step changes in sampling rate requirements for the frequency-domain samples technique put it at a clear disadvantage for larger (larger than 25% of the channel bandwidth in this case) channel spacings and for multiplexes comprising non-power-of-2 numbers of channels. Note that there is an implied assumption of powers of-2 in the IFFT for the OFDM signals (as typically used in 3GPP standards). The possibility of using efficient non-powers-of-2 digital Fourier transform processes is left for future investigation.

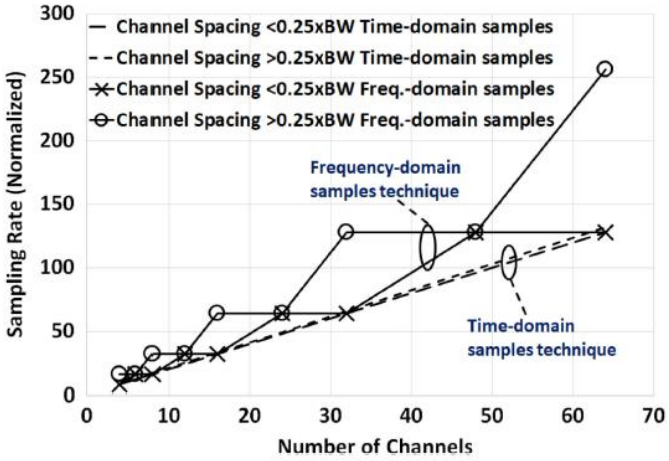


Figure 38: DU sampling rates normalized by the per-channel sampling rate for the frequency-domain and time-domain samples techniques.

A MATLAB-VPI (Virtual Photonics Inc.) co-simulation environment was used in our assessment and is depicted in Figure 39. VPI controls the simulation and calls the MATLAB transmitter processing functions, the outputs of which are passed as waveform samples to the VPI optical link model. At the receiver side, received sample streams from the VPI modelled link output are passed to MATLAB. Finally, performance estimates, such as EVM, of the de-multiplexed channels are performed in "run-time" in the MATLAB receiver code. The modelled example of optical link comprises an MZM fed by a Continuous-Wave Laser (CWL). The optical signal at the output of the MZM is amplified by an Erbium-Doped Fibre Amplifier, transmitted over a short-length Single-Mode Fibre (SMF) patch cord, and received by a high-speed PIN-photodiode (PIN-PD). The co-simulation environment and the matching of the optical link model to an experimental setup have been described in [42], albeit in this work, it is used with increased input optical powers from CWL (10 dBm). The setup is used purely as an example to demonstrate the implementation of the multiplexing techniques with a noisy RoF link, which has been previously characterized and should not be considered in itself a proposed fronthaul link.

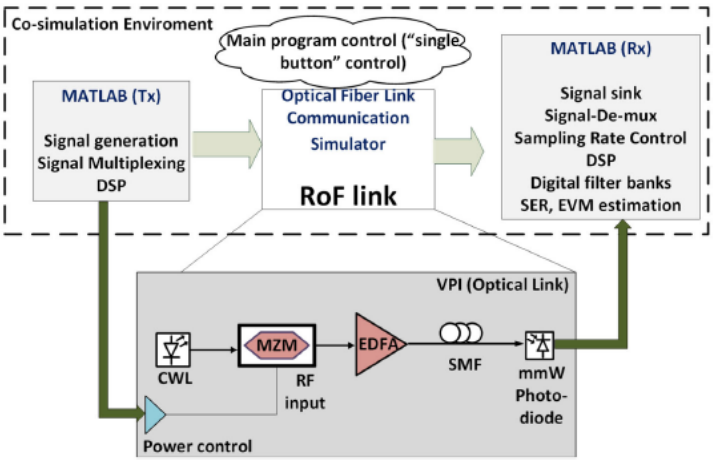


Figure 39: Co-simulation environment and the modelled optical link.

As shown in Figure 40, EVM is used to characterize performance, and results are reported for both a baseline case (no optical link) and for a case with the modelled optical link. In both cases, the multiplex comprises eight channels, while an oversampling factor of 32 (of the per-channel sampling rate) is used. In general, the oversampling factor does not have a huge effect on EVM performance. The per-channel sampling rate is 122.88 MSps for the 100 MHz channels, which comprise 832 data and 192 null subcarriers. The sampling rate used at the receiver for the multiplex is approximately 3.93 GS/s. A subcarrier modulation scheme of 16-QAM has been used. Note that an arbitrarily lower sampling rate could be used with the receiver structure shown in Figure 34 (b). The baseline case is used to establish

EVM constraints that arise purely from the multiplexing/de-multiplexing process of each technique. The case with the optical link is used to show how much the multiplexing techniques might affect overall performance, given the existence of noise and nonlinearities in real systems. Generally, for the baseline case, the EVM performance improves significantly up to stopband attenuations of 40 dB, beyond which there is a lesser improvement, even a slight degradation in some cases. The effect of channel spacing is significant due to reduced inter-channel interference with larger spacings. However, this effect is much less pronounced for the frequency-domain samples technique: it is less sensitive to channel spacing as a result of the generation process for the multiplex, which will always create channels that are orthogonal through its single IFFT operation. For this reason, the frequency-domain samples technique results in better EVM performance, but the difference in performance between the two techniques diminishes as the stopband attenuation is increased. With the optical link included, similar performance trends are observed, although the difference between the techniques is somewhat obscured by the noise floor.

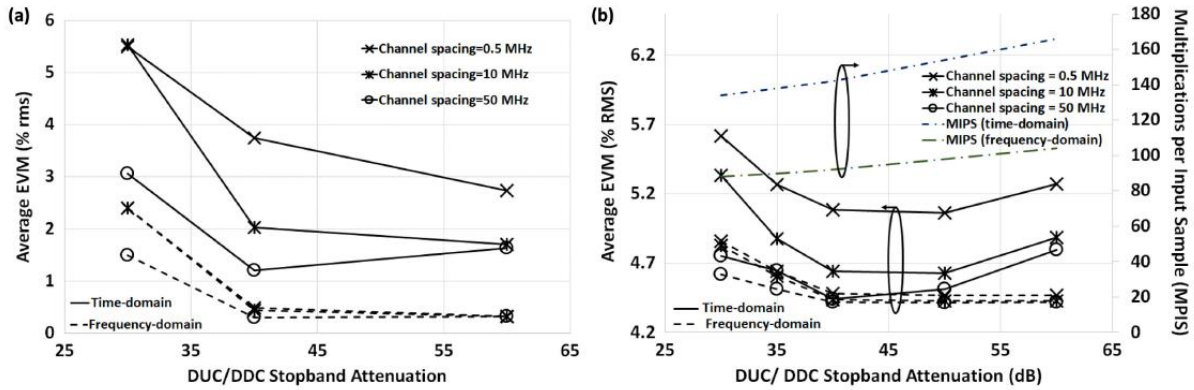


Figure 40: Average EVM (% RMS) results for time-domain samples and frequency-domain samples techniques for a multiplex comprising eight channels and an oversampling factor of 32 (of the per-channel bandwidth). (a) Baseline case (no optical link) for 100 MHz bandwidth channels, (b) following transmission over the modelled RoF link and MPIS results for 100 MHz bandwidth channels. Note that for the frequency-domain samples technique, the stopband attenuation is for the DDC at the receiver, while the MPIS results are for a single DUC/DDC.

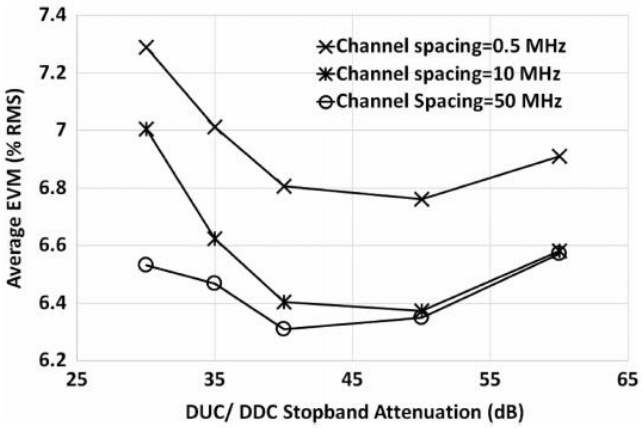


Figure 41: Average EVM (% RMS) results for time-domain samples technique and a multiplex comprising sixteen 100 MHz channels and an oversampling factor of 32 (of the per-channel sampling rate).

As a general confirmation of the performance trends observed in these results, an additional simulation was carried out for sixteen 100 MHz channels using the time-domain samples technique, the results of which are shown in Figure 41. In this case, the sampling rate used at the receiver is approximately 3.93 GS/s. The trends remain for larger multiplexes; that is, the time-domain samples technique suffers from very narrow channel spacings, with improvement in performance occurring with larger stopband attenuations and/or larger channel spacings. Note that for the 50 MHz channel

gap results, the sampling rate is somewhat higher to accommodate the larger channel spacing (larger than 25% of the channel bandwidth).

Figure 42 shows experimentally measured average EVM results for the two techniques, with a multiplex comprising eight 100 MHz channels, following transmission over a short span RoF link. In this case, the sampling rate used at the receiver is approximately 2 GS/s. The measurement setup is essential, as depicted in Figure 39, but the optical link model (and its constituent components) is replaced by the real-world equivalents. The input multiplex is generated in MATLAB, as before, and is downloaded into an AWG. Following transmission over the RoF link, the received multiplex is captured by an Oscilloscope for offline processing in MATLAB. More detailed information on the experimental setup can be found in [42], where the same setup was employed.

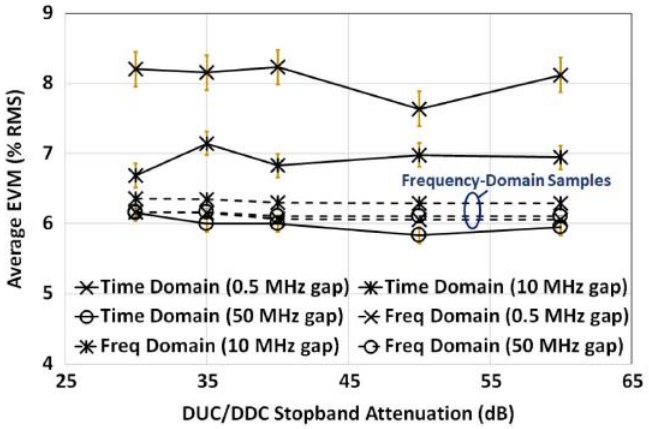


Figure 42: Average EVM (% RMS) for different channel spacings/gaps and DUC/DDC stopband attenuations for the experimental results for the two multiplexing techniques and a multiplex comprising eight 100 MHz channels and an oversampling factor of 16 (of the per-channel sampling rate).

3.2.2 Flexible and efficient fronthaul by incorporating a combined multiplexing technique

The results presented in Section 3.2.1 indicate that a combined technique that operates in both time-domain and frequency-domain samples is promising. The two techniques discussed in Section 3.2.1 and this combined technique are conceptually depicted in Figure 43 [60].

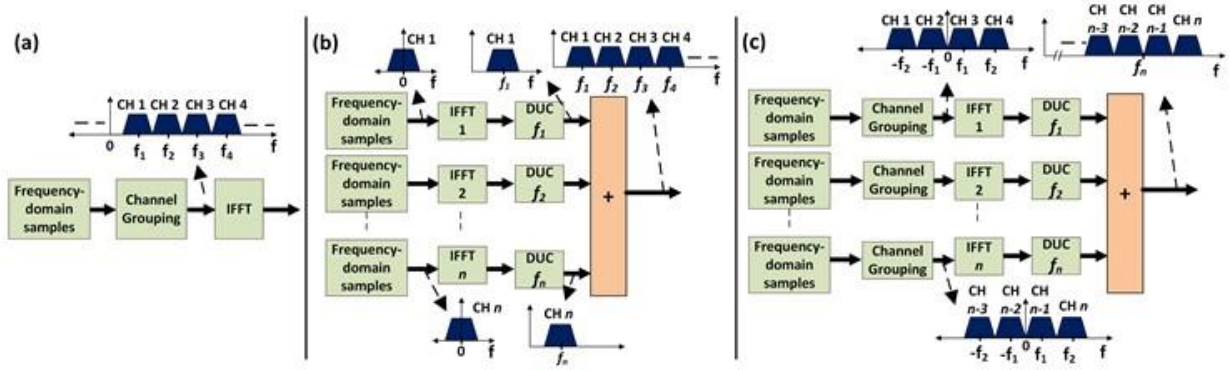


Figure 43: Conceptual depiction of the different multiplexing techniques. (a) Frequency-Domain Samples technique. (b) Time-Domain Samples technique. (c) Combined technique. The f1, f2,..., fn, correspond to the DUC centre frequencies. DUC, Digital UpConverter; IF, Inverse Fast-Fourier Transform; CH, Channel.

The frequency-domain samples technique employs a “single-IFFT”, which leads to the possibility of very large IFFT sizes, however, and increased sampling rates as discussed in Section 3.2.1. Despite the large IFFT size, it was shown that the technique using frequency-domain samples possesses less computational complexity and generally leads to improved performance compared to that using the multiplexing of time-domain samples. However, as higher sampling rates are often required with this technique, a combination of multiplexing techniques is a promising alternative. This combined

technique is shown in Figure 43 (c) and enables realizable sampling rates and flexible multiplex creation and de-aggregation [60]. Here, whole groups of channels, grouped using the frequency-domain samples technique of Figure 43 (a), are upconverted through a DUC to an appropriate frequency location within the multiplex. Note that pre-IFFT, we do not employ conjugate symmetry, and therefore the channels depicted in the positive side of the spectrum are independent of those on the negative side of the spectrum.

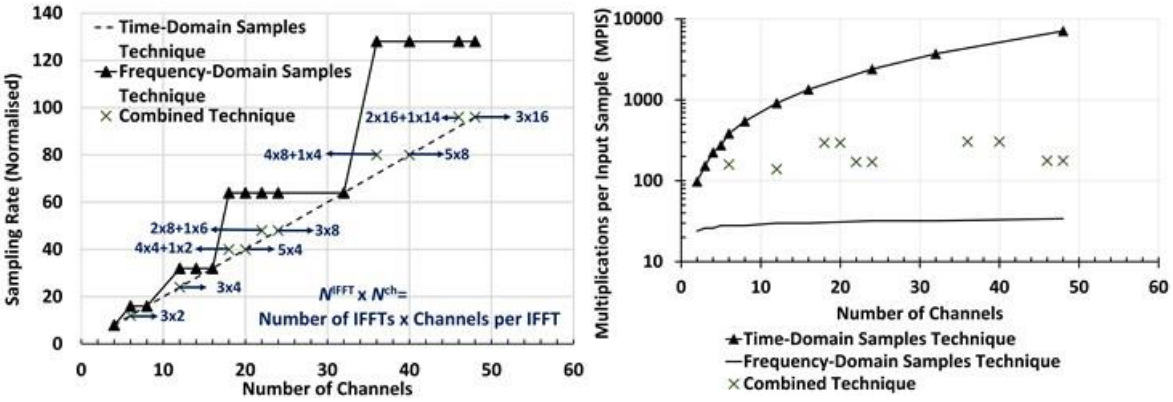


Figure 44: (a) Sampling rates (normalized to per channel sampling rate) versus the number of channels in the final multiplex. (b) Computational complexity is given as a number of Multiplications Per Input Sample (MPIS) versus the number of channels in the final multiplex.

Figure 44 (a) shows the sampling rate requirement for the three techniques, with annotations for the combined technique configurations. For example, if 12 channels are to be multiplexed, the time domain samples technique simply uses digital up-converters to up-sample from each channel, leading to a 24× (that of the individual channels) sampling rate. However, when frequency domain samples are used, the samples have to be arranged at the input of an IFFT with 32× the number of samples of each channel, and a 32× sampling rate is required. This is where a combined approach can be helpful. If the 12 channels are arranged in three groups of 4, each group can be multiplexed first using their frequency domain samples, and the time-domain samples from each of these IFFTs used to create the final multiplex. The resultant highest sampling rate remains at 24× that of the individual channels. Note that a number of combining options are possible, for example, arranging six groups of 2 channels in the previous example. However, more IFFTs require more DUCs, increasing the overall complexity. Therefore, the manner, in which the techniques are combined, is important. As a further example, a multiplex of 36 channels can be divided into four groups of 8 and 1 group of 4, with an overall sampling rate of 80× that of each channel. Note that the individual groupings are chosen such that they can be multiplexed using efficient power-of-2 IFFTs. Using the frequency-domain samples technique in this case for the whole multiplex would result in a sampling rate of 128× that of each channel, while a 72× sampling rate would be needed for the time-domain samples technique.

While enabling sampling rate requirement reductions compared to the technique using only frequency-domain samples, the combined technique also leads to some compromise in complexity. This complexity, measured as the number of multiplications per input sample (MPIS), is shown in Figure 44 (b) for all three techniques. The employed DUC interpolation factor is equal to twice the number of multiplexed channels (as indicated in the x-axis of the figure) or to the number of per-IFFT channel groupings for the combined technique. The channel parameters are based on 5G 3GPP specifications: Each channel has a bandwidth of approximately 100 MHz and a subcarrier spacing of 120 kHz, while the individual channel sampling rate is 122.88 MHz. For the time-domain samples technique, an IFFT length of 1024 is employed. The complexity results are shown in Figure 44 (b) for the combined techniques correspond to the groupings shown as annotations in Figure 44 (a). For the combined technique, this corresponds to groupings of channels into as few IFFT processes as possible (thus having more channels per IFFT) and using the smallest number of DUCs. As the multiplex size increases, the complexity of the combined technique remains relatively low (compared to the time-domain technique) by progressively grouping more and more channels into each single-IFFT process.

Figure 45 shows EVM results for the three multiplexing techniques, for narrow channel gaps of 0.5 MHz ((a), (b)) and wider channel gaps of 15 MHz ((c), (d)). Figure 45 (a) and (c) correspond to the baseline cases (i.e. back-to-back, no link) while (b) and (d) correspond to the cases with the optical link included. All channels have a subcarrier spacing of 120 kHz, while for the time-domain samples technique, an IFFT length of 512 is employed. For the combined technique, a channel grouping of 3×4 is employed (that is, 3 IFFTs with four channels each). EVM is given as the % root-mean-square (RMS) value, averaged across all subcarriers and all channels within the multiplex, following the transmission of 10 frames, while in all cases, 16-QAM modulation has been employed.

In the baseline case, the time-domain technique exhibits worse EVM performance (as was also demonstrated in Section 3.2), as channels within the multiplex are not orthogonal (not formed in the same IFFT process). As the stopband attenuation increases, the EVM performance improves for both techniques, and both show improved performance with larger channel gaps (though more evident for the time-domain samples approach). The EVM performance of the combined technique is between that of the two extreme techniques but generally closer to that of the frequency-domain samples technique. This is expected due to the grouping of a number of channels into channel groups that are orthogonal. For the measured experimental performance, the trends observed in the baseline case are not always clear due to the link-introduced noise floor. Still, the performance of the combined technique is comparable to that of the frequency-domain samples technique in all cases; there is negligible relative EVM degradation.

Thus, the combined technique, by appropriately combining the time-domain and frequency-domain samples techniques, can balance sampling rate and complexity requirements, leading to hardware simplification while maintaining improved performance.

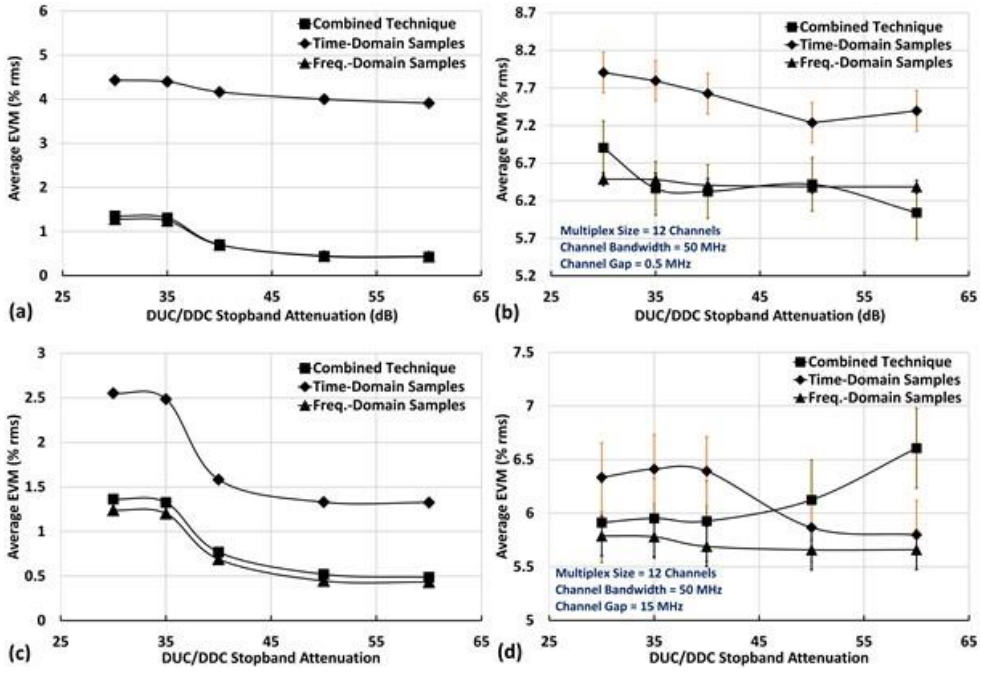


Figure 45: EVM for different DUC/DDC stopband attenuations for multiplex comprising twelve 50 MHz channels. (a) Back-to-back case for channel gap of 0.5 MHz and (b) after transmission through optical link. (c) Back-to-back for channel gap of 15 MHz, and (d) after transmission through the optical link. Note that for the frequency-domain samples technique, there is no DUC employed at the transmitter (only a DDC at the receiver).

4 Network virtualization and slicing

4.1 RAN slicing issues

4.1.1 Overview

The deployment of 5G mobile networks is ongoing. So far, in most countries, so-called non-standalone 5G networks are installed. This variant of 5G can be seen as an evolution of the 4G network by the integration of 4G RAN with 5G RAN nodes and modifying 4G Core Network (EPC). The main gain of this approach is high-quality data transmission and further leverage of systems parameters such as maximum bandwidth and spectral efficiency, as well as minimized latency or transmission error rates. However, the development of the network, as well as provided services, were solely oriented to the mobile consumers' experience. The progressing technological improvements in electronics have led to the popularization of the Internet of Things (IoT) concept – interconnected large scale machine networks acquiring, processing and transmitting diverse types of data. Together with the IoT concept, several other ideas utilizing massive numbers of devices are being implemented, such as Smart Cities or augmented factorial automation, which both are a part of an idea stipulated as the fourth industrial revolution (Industry 4.0).

To satisfy both mobile consumers' as well as business clients' requirements, 5G Stand-Alone (SA) has introduced a new approach to communication in comparison to previous generations of mobile networks. Instead of creating a single, multi-service network that can cope with multiple and diverse requirements, it has been proposed to create a kind of virtual networks tailored to specific service types. Such an approach is possible due to the incorporation of the network virtualization concept (ETSI MANO-based) into 5G standards. Following the NGMN approach, 5G SA allows for the dynamic creation of separate logical networks, known as network slices.

Three basic network slice classes have been defined [61]: Ultra Reliable Low Latency (URLLC), Enhanced Mobile Broadband (eMBB) and Massive Machine Type Communication (mMTC), which respectively address the requirements for low latency, fast transmission speeds and communication with a large number of devices. 3GPP has extended the list by the addition of V2X communications as the fourth separate service class [62] recently.

However, the network slicing separates not only the user plane operations but also the control plane ones. The control plane operations can be customized for the need of the service (type of mobility, etc.). Network slicing concerns both the 5G Core network (5GC) functions (AMF, UPF, etc.) and the Radio Access Network (RAN). The 5GC has specialized functions that support network slicing (NSSF, etc.).

As long as the need for implementation of RAN slicing is widely understood, no consensus on its granularity or slicing implementation has yet been established. With several technical, management, and architectural requirements accompanied by imprecise 3GPP standardization, fully functional and universal RAN slicing implementation seems to be a distant perspective. The existence of multiple slices may lead to scalability problems and a need for efficient resource sharing. Moreover, the so-called slice isolation property should provide a lack of impact of the congestion of one slice on data-plane QoS of other slices.

The goal of the paper is to discuss the 5G RAN slicing options in the context of the requirements of different slice service types (URLLC, eMBB, mMTC) and the limitations of the existing approaches.

4.1.2 RAN slicing status

Current standardization efforts undertaken by the 3GPP have led so far to the establishment of basic principles concerning RAN slicing [63]. According to 3GPP, to fully support network slicing in RAN, several features have to be incorporated. Firstly, gNB has to support traffic handling for each slice by using different PDU sessions, with each PDU session associated with a slice. Such an approach

facilitates the network to create slices by providing different L1/L2 configurations and scheduling. The 3GPP has no plans to provide the details on how to implement slice RAN slicing – this is left to vendors. Typically, the RAN slicing is implemented as a Physical Resource Blocks scheduler mechanism.

4.1.2.1 QoS of different types of network slices

The aforementioned slice types differ significantly in terms of QoS. Table 2 presents the main features differentiating each of the defined slice type [64] (the values skipped in the table are service type dependent and can vary contingent upon the network's performance). In the case of typical slices' requirements from the network are heterogeneous. Generally, eMBB slices require large bandwidth and can deliver services on a satisfactory level with a delay in the range of dozens of milliseconds. As exemplary services concern high bitrate data streaming, MTUs can be relatively long.

Slice Type	Bandwidth	Latency	Reliability	Connection density
eMBB	20 Gbit/s (DL) 10 Gbit/s (UL)	10 ms	-	-
URLLC	-	< 1 ms	99.9999%	-
mMTC	-	< 10 s	-	1 M/km^2

Table 1: 5G slice types with associated peak requirements.

In regard to the URLLC slices, the user plane latency should not exceed 1 ms (500 µs in RAN), and the reliability factor for transmission of a packet of the size of 32 bytes should not be lower than 99.9999%. It is expected that the traffic streams are used for control of devices, and their bitrate is below 1 Mbit/s.

The mMTC slice type has neither large bandwidth nor low latency requirement. However, it demands from the network an ability to support a high number of randomly transferred in the UL small packets. Since mMTC slice is generally designed to support large sensor networks, the required bitrate is typical of the size of kbit/s (Mbit/s in terms of factory automation), and MTU's will be typically small. The latency heavily depends on the type of service, but in general, it should not exceed 10 s. mMTC slices used for support of medical devices, requirements may vary in terms of reliability and latency resembling more URLLC's demands than the mMTC's ones [65].

For deployment of network slicing, some slice-specific mechanisms have to be implemented:

- RAN awareness of slices – the ability of NG-RAN to provide differentiated traffic handling depending on the slice type. It includes mechanisms like grant-free transmission and mini-slots that can be used for URLLC traffic (both mechanisms will be discussed in details later on):
- Support of selection of the RAN part of the network slice (sub-network slice) based on the Single Network Slice Selection Assistance Information (S-NSSAI) provided by UE or 5GC,
- Support for provisioning diverse QoS by a slice,
- Enablers for policy enforcement between slices in terms of resource management,
- Selection of Core Network entities based on NSSAI,
- Ensuring resource isolation between slices,
- Access control policies,
- Maintenance of awareness of slice availability,
- Support for UE associated with multiple network slices simultaneously in the form of maintaining only one signalling connection,
- The granularity of slice awareness by indicating S-NSSAI corresponding to appropriate PDU Session,
- Slice isolation meaning a lack of mutual impact between coexisting slices (a congested slice should not impact the traffic in other slices,

- Validation of the UE rights to access a network slice.

Substantial contribution to the RAN slicing concept is likely to appear within Release 16 in TR 38.820 “Study on the enhancement of Radio Access Network (RAN) slicing for NR”.

4.1.2.2 QoS of RAN slicing from UE’s perspective

RAN slicing, in terms of UE, concerns the technical and procedural requirements that UE has to fulfil in order to be able to attach to one or more slices provisioned by the NG-RAN. According to [63], there is no strict limit on the number of slices, to which UE can be attached simultaneously; however, eight is a number treated as sufficient for the time being.

Currently, access to slices is realized by performing registration (initial access) or by the PDU-session establishment (additional slice attachment) procedure [66]. Following successful CN resources allocation for the transmission *RRCConnectionReconfiguration* procedure is performed, where UE is supposed to reconfigure Signalling Radio Bearer (mutual bearer for each attached slice) and establish Dedicated Radio Bearer (DRB) for the session.

4.1.3 RAN slicing approaches

With no direct 3GPP’s outline concerning the implementation of slicing in RAN, several concepts are being considered in terms of both technical realizations as well as slicing granularity with the aim to provide mechanisms for effective resource allocation and usage. This section is devoted to possible approaches to RAN slicing, presenting their advantages, drawbacks and limitations in terms of their utilization in a fully functional network. Several options of network slicing with different ways of sharing of radio components have been presented in Figure 46. The main distinction regards slice isolation and sharing efficiency [67].

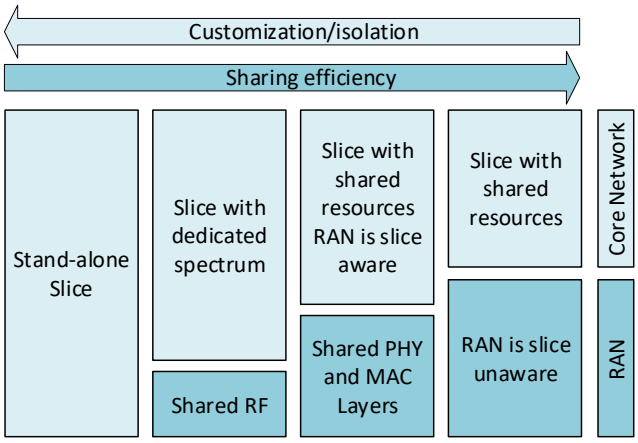


Figure 46: RAN slicing options.

4.1.3.1 Slicing from the RAN protocol stack point of view

From the protocol point of view, RAN slicing can be viewed as a combination of allocated radio resources in the physical layer together with a specific configuration of upper protocol layers (Figure 47).

Allocation of radio resources, as well as scheduling in MAC, will be described in further parts of the article. Before discussing the network slicing issues, first RAN split options and NR features, as well as mechanisms that can be exploited for the purpose of network slicing, will be outlined.

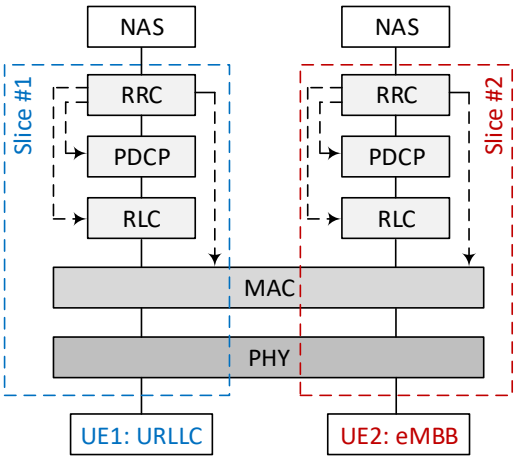


Figure 47: Slicing from the protocol point of view.

4.1.3.2 5G NR split options

The RAN split options have been proposed as a means of reducing the data rate requirements over the fronthaul section of the RAN and the overall RAN cost. Traditional transport methods that rely on the transmission of sampled and quantized radio samples (most notably using the Common Public Radio Interface (CPRI) specification [68]) cannot scale to the data rate demands imposed by the larger bandwidths and multi-antenna transmission schemes (such as mMIMO) employed in 5G. A detailed summary of split options discussed within 3GPP is presented in [46], while the split options are depicted in Figure 48. More details about the Data Link Layer split (also termed High Layer Split, HLS) are shown in Figure 49, while Figure 50 depicts some of the intra-PHY split option (also termed Low Layer Split, LLS), with Option 8 corresponding to the traditional regime of sampled radio waveform transport (e.g. CPRI). Note that the fronthaul section is generally assumed to be part of a centralized or cloud RAN (C-RAN) deployment and is distinct from the small-cell concept (the small cell forum has, in turn, defined split options for use in small cell deployments [69]). 3GPP has reached an agreement on HLS (defined over interface F1-U/C), which is essentially already standardized in the form of Dual Connectivity (DC) Option 3C. However, no agreement on the LLS has been reached to date (expected to be defined over the F2 interface). At the same time, the CPRI consortium has released an updated specification incorporating packet-based transmission (IP/Ethernet) and different functional split options [44].

Each split option comes with its own advantages and disadvantages, so choosing one option over another is fundamentally an application-dependent process. Data rate reduction, latency constraints, centralization gains, joint processing constraints, ease of migration (from traditional Option 8) are factors that can inform the choice of the split point. Generally, it is assumed that the split point choice should strive to achieve a balance between data rate reduction, centralization gains and the ability to process signals jointly. Thus option 7.2 can be seen as the preferable option due to decentralized precoding. With data rates scaling linearly with the number of layers (instead of the number of antennas), significant data rate reductions can be achieved mainly in mMIMO applications. Option 6 is another interesting split point offering the most substantial reduction in data rate from the available LLS options (as it directly transports bits and not modulation symbols) and has been already demonstrated for 4G systems [48], [70], [71].

All LLS options have significant latency constraints, as they reside within the HARQ real-time loop. These constraints become even tighter with the use of mini-slots in 5G. The HLS splits, on the other hand, have significantly reduced (by orders of magnitude) latency constraints but cannot be used for joint processing of signals.

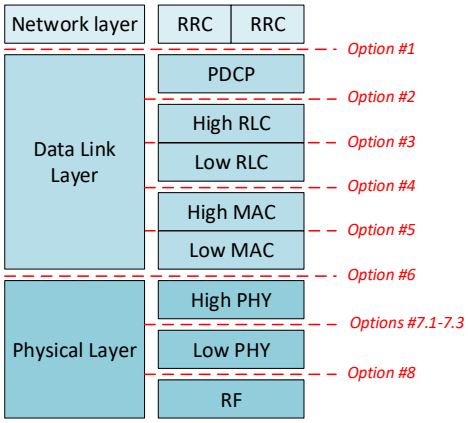


Figure 48: gNB split options proposed by 3GPP [46].

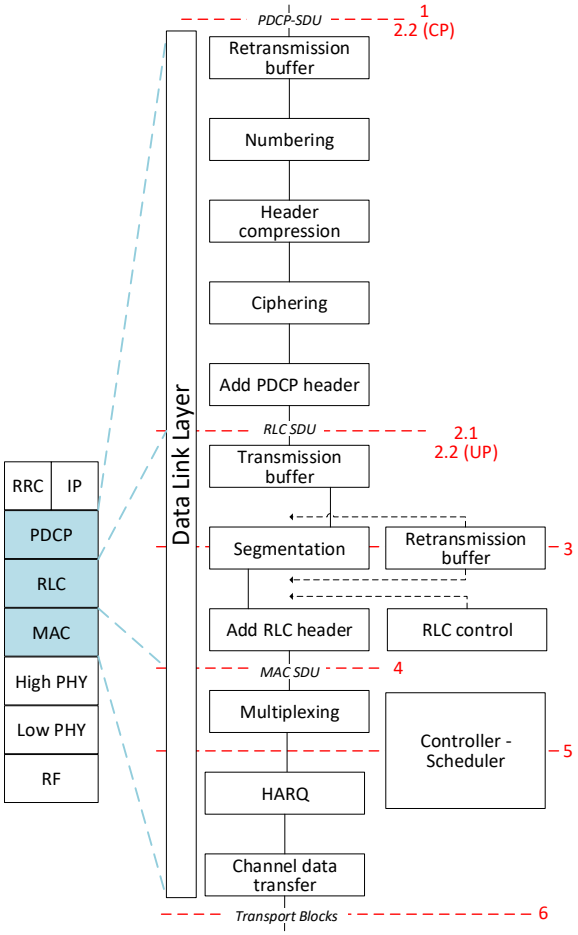


Figure 49: Functional split options in Data Link Layer.

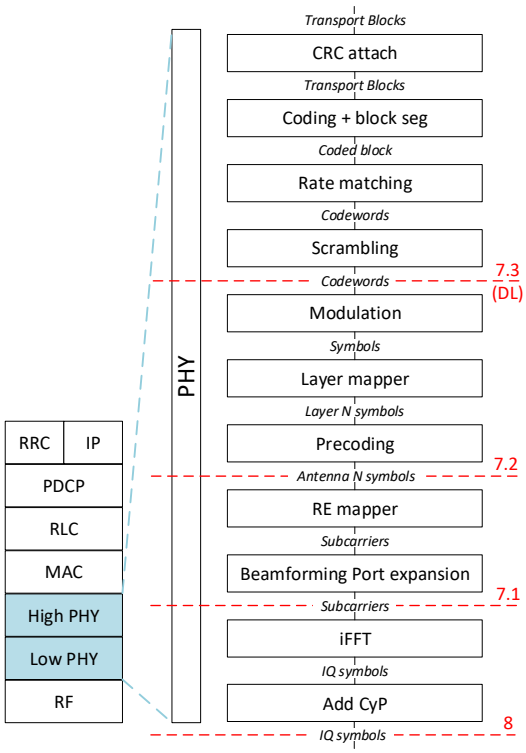


Figure 50: Functional split options in PHY layer.

A flexible RAN split, making significant use of virtualization techniques, has also been proposed [72]. For RAN slicing, the idea of a flexible split based on a slice is an interesting one. The transport network can optimize the deployment of RAN functions based on slice requirements. eMBB slices, for example, will most likely necessitate the use of multi-antenna transmission techniques meaning that a pre-/post-antenna processing split point (in the DL/UL) would be beneficial. For URLLC slices, a variable DL/UL split point may be beneficial, allowing reduced data rates in the DL and joint processing in UL to improve reliability. mMTC type slices can employ more centralized processing, maximizing centralization gains. Variable split points can also be used to offer local connectivity to edge computing nodes on an "as and when" needed basis. Such an approach can be beneficial for URLLC slices.

Note that the association of a slice with a split point has to consider the QoS requirements of the transport network (usually quantified through a number of KPIs). For packet-based transport, a number of Ethernet schedulers are available and can be rated according to their slice isolation capability [48]. These are schedulers operating at Layer 2 (of the transport network), and most are defined as part of the 802.1 Time-Sensitive Networking (TSN) specifications [73], although not all fall strictly under the TSN definition. Techniques operating at higher networking layers are specified by the IETF Deterministic Networking working group [74]. A summary of TSN approaches, specifically focused on URLLC, is presented in [75].

Soft and softer isolation approaches would include strict priority and pre-emptive schedulers (as defined, for example, by IEEE 802.1CM (specifically for use in an Ethernet fronthaul) [76]. Time-aware shapers (based on 802.1QBV [77]) can offer hard isolation by incorporating transmission windows for high and low priority traffic, in what is essentially a TDM approach, eliminating latency variation. However, the extensive use of buffering/guard periods may introduce latency issues. These techniques can be used in some combination to offer diversified QoS within a slice. For example, time-aware scheduling can be used in conjunction with pre-emption to reduce the need for large guard periods (potentially at the cost of an increase in delay variation).

Another scheduler that can offer hard isolation is a gap-filling aggregator whereby quiet periods in-between the transmission of high priority traffic are employed for the transmission of low priority traffic [48]. Multiple such low-priority-high-priority sessions can exist simultaneously for different slices while at the same time allowing diversification of QoS within each slice by incorporated class-of-service differentiation among different high-priority flows and among different low-priority flows.

An alternative transport mechanism is FlexE (often considered a Layer 1 transport scheme), where again a TDM-type approach is used but applied directly within the physical layer [78]. FlexE has the ability to provide hard isolation of slices. Scheduling of traffic is carried out with the use of calendars that are incorporated within the Protocol Coding Sublayer (PCS) of the Ethernet PHY. To make more efficient use of the available transport resources, some dynamic adaptation of the calendar-based scheduling is required, perhaps through SDN and appropriately defined APIs [79]. Note that in FlexE, the scheduling decisions are removed from the MAC layer and placed within PHY.

However, dual-scheduling can be applied whereby intra-slice scheduling is carried out at the MAC layer while inter-slice scheduling is handled by the PHY scheduling.

4.1.3.3 5G NR numerology

5G NR introduced several alterations to the physical layer in comparison to 4G. In comparison to LTE Physical Resource Blocks (PRBs), the following changes have been introduced:

- elastic OFDM symbol length (non-fixed subcarrier spacing) – essential in eMBB and mMTC communication: short symbols for eMBB (large subcarrier spacing), and long symbol duration for mMTC (small subcarrier spacing allowing multiple devices),
- configurable Transmission Time Interval (TTI) – the shorter TTI, the lower latency (vital in URLLC cases).

There are five possible RB's configurations allowed by the 3GPP, called numerologies, presented in Table 2.

Numerology (μ)	$\Delta f = 2^{\mu} \cdot 15$ [kHz]	Cyclic prefix	OFDM sym. per slot	T_{slot} [ms]
0	15	Normal	14	1
1	30	Normal	14	0.5
2	60	Normal/Extended	14/12	0.25
3	120	Normal	14	0.125
4	240	Normal	14	0.0625

Table 2: 5G NR numerologies.

Release 15 does not allow using multiple numerologies in the same radio channel. Typically, wider channels use higher numerology what is justified by the computational complexity of the FFT/IFFT transforms.

4.1.3.4 Scheduler-based slicing

The most popular network slicing implementation at RAN is, at present, based on the scheduler. The 5G RAN scheduler is working on Physical Resource Blocks (PRB), i.e. time-frequency blocks, which size and duration follow the 5G numerology (frequency raster and transmit time interval). In the case of slicing, the scheduler should work on both a slice and user level. The slice level scheduling should provide isolation between slices (i.e. lack of impact of congestion in one slice on another slice).

Scheduler-based resource allocation is not a new issue. There is a number of papers focused on the fairness of resource allocation for 4G and 5G RAN. The scheduling based on Proportional Fairness (PF) is so far the most popular approach to resource sharing [80] and [81]. The approach is trying to maximize both the total throughput of the network and the minimal level of service provided to users. PF assigns scheduling priorities to each data flow that is inversely proportional to its anticipated resource consumption.

The Dominant Resource Fair (DRF) algorithm addresses the problem of fair allocation of hybrid resources to users with heterogeneous demands [82]. It is seen as the max-min fairness approach extended to multiple resource types. The allocation of resources to a user is determined by the user's dominant share, i.e. the maximum share of any resource that the user has been allocated. DRF seeks to maximize the minimum dominant share across all users. In [83], the Bottleneck Max Fairness (BMF) approach has been proposed. In this concept, every transaction receives an allocation of some bottleneck resource that is maximal for that type of resource.

Undoubtedly, scheduling should involve the PHY layer properties, significantly increasing the scheduler's complexity. It is worth a mention that the scheduler should preserve the isolation between the slices.

The eMBB slice is typically used for video streaming. Its traffic is characterized by big MTUs, allowed long delays and high bitrate. The scheduler-based slicing is the most common implementation of RAN slicing; however, as presented in the paper, several other mechanisms can also be used for that purpose.

The mMTC traffic is characterized by intermittent communication from many sources. Due to the energy efficiency aspects of the devices, in most cases defining a sleep schedule might be needed. Therefore, mMTC scheduling comes with time constraints but not as critical as in the URLLC case – mMTC involves more UL traffic than DL traffic. Since the DL traffic is relatively rare, a simple scheduler like RR or PF may be used. For the UL traffic, we may distinguish between periodic update (i.e. MTC are activated during a predefined time interval) and event-driven MTC traffic pattern. For the periodic update, we propose using pre-fixed scheduling (e.g. Semi Persistent Scheduling – SPS). In this case, the Slice Orchestrator (SO) should indicate when the MTC will be triggered by the application to send reports.

URLLC slice type requires low delay; therefore, the 3GPP has proposed mechanisms enabling low latency scheduling, which includes punctured, pre-emptive scheduling that utilizes mini-slots in the downlink (DL) and grant-free access in the uplink (UL). The DL mechanism enables the puncturing of the eMBB traffic by inserting mini-slots (2, 4 or 7 blocks) [84] in an eMBB transmission that overwrite initial data that was to be transmitted within a slot. Rearrangement of radio resources is to be carried out immediately after URLLC transmission request in case of no other free resources to be assigned for URLLC traffic. Lost parts of eMBB transmission can be recovered by using low-density parity-check (LDPC) encoding. In case of error correction failure, the lost data is re-transmitted within the next transmission cycle as in the standard Hybrid Automatic Repeat Request (HARQ) process [85]. Punctured pre-emptive scheduling, despite providing low delay for URLLC traffic, can result in increased latency of eMBB traffic and therefore should not be used as a basic but rather as an optional

mechanism in critical situations. An example of DL scheduling with the mini-slots inserted into eMBB traffic is presented in Figure 51.

The 3GPP has proposed two scheduling schemes for URLLC, instant scheduling and reservation-based scheduling. The immediate scheduling approach prioritizes URLLC packets by interrupting other ongoing transmission and scheduling URLLC packets in their place by the use of the previously described mini-slot mechanism. Consequently, this approach can drastically degrade other services. The reservation-based scheduling to avoid negative impact on other services, use a reservation-based frame for URLLC. This approach, however, results in overheads in the control signalling and may cause wasting allocated slots in case of no URLLC transmission. Two types of reservation can be implemented: dynamic or semi-static. In spite of the advantages that pre-emptive punctured scheduling can bring, the optimal eMBB and URLLC scheduling decisions cannot be decoupled. To satisfy the requirements of both types of traffic, it is essential to use joint optimization [86].

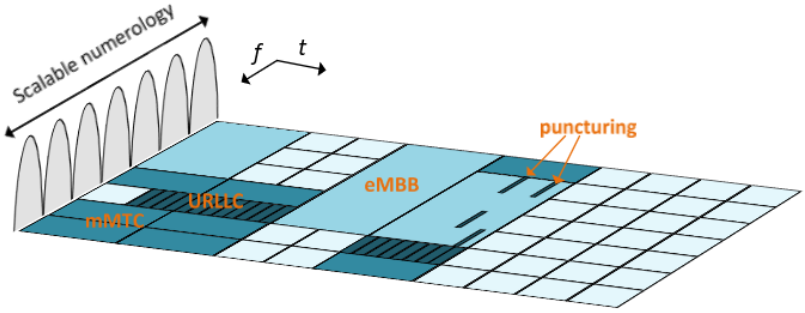


Figure 51: 5G scheduler-based slicing – an example with mini-slot usage.

Generally, the access to the wireless UL channels is controlled by the grant-based scheduling (GB) mechanism that uses a four-way handshake procedure to ensure exclusive rights of the user to the assigned channel. The process enables avoiding potential collisions at the cost of large latency and signalling overhead, undesirable in URLLC services. Therefore, to allow the immediate transmission of small data, grant free (GF) access mechanism is proposed as a solution [87], where handshake procedure is skipped, and data together with required control information is sent in the initial transmission. Such an approach enables latency reduction at the cost of potential collisions within a channel. Two variants of GF transmissions have been introduced: pre-located over dedicated resources (suitable for traffic with fixed pattern) and shared among multiple users through contention (more efficient and flexible in terms of resource utilization). To diminish the impact of eventual collisions, several enhancements for GF access are being understudied. The most popular include sending multiple replicas of the data called K-repetition trans-mission or proactive repetition, which is based on re-sending data until an acknowledgment is received [88].

Different approaches to the 5G PRB scheduler are presented in many scientific papers. In [86], a joint eMBB and URLLC scheduler was proposed, which can meet the requirements of the URLLC traffic (low latency) while maximizing the utility of the eMBB traffic. In [89], the radio link has been logically divided into high bit rate slices and low delay slices – in this case, the goal was to minimize the delay. In [90], a proportional fair resource allocation algorithm that allocates resources to incoming URLLC traffic while ensuring the reliability of both eMBB and URLLC has been described. A novel approach to punctured pre-emptive scheduling has also been proposed in [90], where a Conditional Value at Risk (CVaR) concept has been introduced. CVaR is a risk-sensitive-based resource allocation for the URLLC traffic that minimizes the risk of the eMBB transmission by protecting the eMBB users with the low data rate. The approach has been tested by the simulations and proved to allocate resources efficiently while maintaining a satisfactory reliability level of both eMBB and URLLC. In [91], such an approach was presented and further extended in [92] to operate using resources from multiple base stations (BSs). The mentioned studies refer to radio resource allocation to each slice based on channel quality conditions, which satisfies bandwidth, but not necessarily latency demands. The approach, in which latency is considered, was presented in [93], where a proactive RAN slicing scheme to support haptic communications was described. The proposed solution computes the number of radio resources

allocated to each slice periodically and then uses a dynamic queuing scheme for resource assignment to nodes based on individual latency requirements. Latency demands, however, are not considered during slice creation. Another way is creating slices in mixed traffic scenarios based on bit rate demands. Such an approach was presented in [94], where slicing operation considers a combination of resource-oriented (e.g. resource occupation) and rate-oriented parameters (e.g. aggregate bit rate) to establish and limit the number and characteristics of the resources allocated to each slice. A similar approach has been described in [95], where authors compute the necessary amount of resources per slice based on the aggregate Guaranteed Bit Rate (GBR) demands of the services. Serving both elastic and inelastic traffic efficiently might be problematic. In [96], an interesting solution has been proposed to enhance handling data flow in both situations. The proposition enables achieving certain latency levels for inelastic traffic without guarantee of meeting latency requirements due to their exclusion from the slice creation process. Once slices are defined, radio resource allocation is conducted based on partitioning schemes.

Several approaches of radio resource partitioning have been proposed in the literature; however, the most common one is based on treating the partitioning process as an optimization problem. For instance, a partitioning scheme in [91] and [92] is defined as a general integer programming problem or, as in [92], a Binary Integer Programming (BIP) problem. Solutions enabling maximization of the overall resource usage have been presented in [93] or [97], further enhanced by reinforcement learning. Another approach oriented for solving an optimization problem has been proposed in [90]. Based on solving a risk-sensitive partitioning optimization problem, it enables satisfying the reliability requirements of eMBB and URLLC services. A large group of solutions use game theory to achieve more efficient partitioning. In [98], the scheme that uses bankruptcy theory for the allocation of resources to slices improved by using cooperative sharing has been presented. Another solution utilizing game theory has been described in [99], which is able to reach a Nash equilibrium under certain conditions. Other interesting solutions include the dynamic partitioning process [100] or slice-aware admission control [95]. The first approach introduces the concept of a Slice Broker responsible for resource management and monitoring per each service slice. The second concept is based on the Markov stochastic model and enables sharing resources with diverse guaranteed bit rate services in multi-tenant scenarios.

In [101], a RAN solution that includes two functional splits is described. The first one is the MAC-PHY split and the second one is the PDCP-RLC split. The change of the split can be done during network runtime is presented. The authors consider only CU and DU RAN devices (no RU). In the PDCP-RLC split, the PDCP function is implemented in the CU, whereas the RLC, MAC, PHY, and RF functions are located in the DU. One, the PDCP function is compute-intensive ciphering; therefore, it is better to centralize such operation; moreover, the fronthaul traffic of such split is very similar to the user traffic, as only a PDCP header is added to each IP packet. In the MAC-PHY split, the PDCP, RLC, and MAC functions are implemented in the CU, whereas the PHY and RF functions are located in the DU. The DU is, therefore, less complicated. Furthermore, some coordination techniques such as coordinated scheduling are better supported with the coordinated link adaptation combined with MAC. The fronthaul traffic, in this case, is much more intensive, and the latency requirements are increased in comparison to the PDCP-RLC case. To avoid time-consuming orchestration (dynamic function deployment), the authors of the concept have proposed a replication-based approach, in which the MAC and RLC functions are simultaneously deployed in both DU and CU. Only one instance of the deployed functional blocks is deployed at a time. The roles can change on demand. The main drawback of the approach is that MAC and RLC processes are running simultaneously on both units, even when they are not used. As there is a set of variables and data structures that are created, modified, and used by both functions at run-time, the switching from PDCP-RLC to MAC-PHY case (or vice versa) also requires context-switching, i.e. some variables (state) have to be also transferred to the destination before completing the migration. To make the interruption short, the migration of the context can be done in a proactive way. The authors using so-called soft migration have obtained almost 0% packet loss rate. The concept of duplicated (or multiple) RAN functional components can be used for different network slices due to their difference in volume and delay requirements. We have left the topic for further study.

4.1.3.5 Other approaches to RAN slicing

There some works that consider Radio Admission Control (RAC) as a technique that can be used for RAN slicing. These include solutions such as multi-tenant admission control [102] or the NVS concept using tenant-specific admission control [103]. A promising solution has also been presented in [104], where a joint admission control and network slicing approach has been described, where spectrum allocation, admission control and spatial multiplexing is determined by a heuristic algorithm.

4.1.3.6 Impact of RAN split on RAN slicing

The RAN split has been typically evaluated in the context of efficient RAN transport. The preferred option is 7.2, but for small nodes, Option 6 is also considered. A flexible RAN split research is also ongoing. It is, however, important to evaluate to split impact on the capabilities of using mini-slots.

In [105], a joint RAN slicing and RAN functional split are presented. An exemplary split for different slice types is presented in Figure 52.

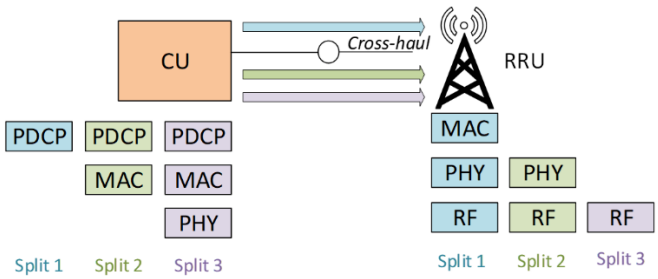


Figure 52: Functional splits for different slice types.

The optimization goal is joint routing (from user to Centralized Unit) and functional split optimization from the slicing point of view. This paper aims to exploit the high density of RU by jointly analysing routing in the RAN and user association. The optimization has concerned with data transport efficiency and the computational resources of RRU and CU units. In fact, the concept has been applied to 4G RAN (or C-RAN); therefore, no DU has been considered. The authors used the Mixed-integer Non-Linear Programming (MINLP) approach converted into a linear problem (i.e. MIP), by proper problem reformulation. The obtained results show that despite the higher cost, the proposed approach provides more overall throughput.

An interesting approach has also been proposed in [97], where an adaptive functional split concept enabling the dynamic realization of PDCP-RLC or MAC-PHY split has been presented. The core of the concept is RLC and MAC functionalities duplication in DU and CU, as presented in Figure 53.

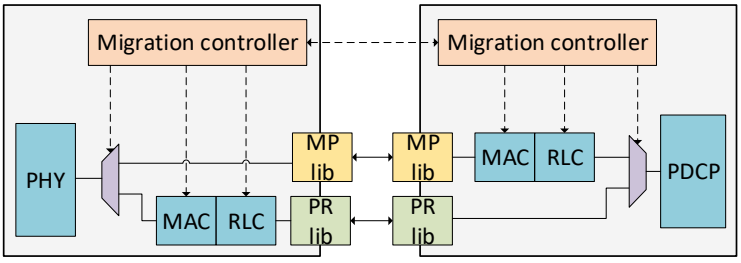


Figure 53: Functional architecture of the adaptive functional split (duplicated MAC and RLC components) [101].

Simultaneous double deployment of RLC and MAC entities enables a fast switch of the split as migration (supervised by the Migration Controller) involves the only activation of the pair of inactive RLC-MAC together with the deactivation of currently active entities. The replication-based approach helps in solving latency issues that appear in a standard virtualization-based approach (deployment on a VM and migration of a VM).

4.1.4 Radio Resource Management and RAN slicing

Radio Resource Management (RRM) functionalities are essential for supporting an effective split of resources among the slices. Generally, to fulfil a given traffic demand, deployment and exploitation of RAN resources can be achieved by using the following basic RRM functions:

- Spectrum planning – assignment of spectrum resources to cells and their arrangement into carriers,
- Inter-Cell Interference Coordination (ICIC) – mitigation of the inter-cell interference by the establishment of limitations,
- Packet Scheduling (PS),
- Admission Control (AC).

RAN slicing can be realized at different levels of RRM.

RAN slicing at the spectrum planning level can be implemented by the allocation of part of the spectrum (i.e. the number of carriers) to slices according to slice requirements. Such an approach ensures excellent traffic isolation between slices; moreover, the tenant can implement their own RRM policies (i.e. ICIC, PS and AC). The problem of this approach is the inefficient usage of radio resources in case of dynamic demands.

In the ICIC-based RAN, carriers of each cell are shared by multiple slices. The approach provides higher granularity (RB's level instead of carriers) and faster resource allocation reconfiguration in comparison with the spectrum planning-based approach.

However, as pointed out in [106], the most important RRM functions needed to achieve flexible RAN slicing are PS and AC, which have been widely described in section 4.1.3. Despite providing necessary functions for legacy networks, RRM mechanisms need substantial enhancements enabling, i.e. SLA awareness for slices or inter-working of RRM functions in a fully sliced network – legacy RRM functionalities are typically focused on cell-centric performance instead of UE-centric.

A crucial need for RRM enhancements has also been pointed out by O-RAN Alliance in [107]. Apart from mechanisms needed to ensure slice SLA and its isolation, an additional challenge is posed by V2X and UAV communication due to heterogeneity of requirements, handovers' frequency or need of gathering and provisioning supplementary service-related data such as 3D radio coverage or UE's location. The O-RAN addresses mentioned issues by architectural enhancements within the O-RAN framework in the form of, i.e. RAN Intelligent Controller (RIC). The O-RAN framework has been further described in section 4.3.

4.1.5 RAN slicing challenges and open issues

Despite numerous approaches to RAN slicing within presented solutions, several challenges and open issues remain unaddressed.

The main issues regard PRB scheduler's configuration and optimization to enable service provisioning, ensuring predefined KPIs for each deployed slice. The first issue is related to the scheduler's behaviour under heavy traffic with a large number of deployed slices. Scheduling implementation should ensure stable performance in spite of a number of slices. In terms of PRB scheduling, another challenge is the implementation of efficient scheduling algorithms, which will enable fair resource allocation to diversified slices, ensure slice SLA and simultaneously remain relatively simple. In the case of the complexity of the scheduling algorithm, there is a high risk of performance instability, which might result in an unacceptable delay. Taking into consideration the requirements for latency-critical applications (URLLC slices) as well as additional scheduler mechanisms such as puncturing and mini slot insertion, it is essential for the scheduling algorithm to operate with near to 0 ms latency.

Another open issue is the aspect of dynamic instantiation of virtual RAN components. Several problems arise that mostly regard fast and efficient deployment of virtual RAN functions in mobility-management related scenarios such as handovers. It is essential to provide service continuity and low

latency, especially in the case of URLLC transmissions (i.e. UAV or V2X scenarios), due to the typically critical impact on the safety of possible network failures to perform accordingly to SLA.

4.1.6 RAN slicing implementations

In [108], a RAN slicing implementation using a two-level scheduling process has been presented. The architecture is based on the legacy LTE concepts, incorporating logical channels and their mapping to Evolved Packet System (EPS) bearers. The main difference is related to the abstraction of Physical Resource Block (PRB) and their virtualization (virtual RBs). To manage the physical resources, an additional layer called Resource Mapper (RM) was added, which aim is to interface between shared PRB and the Slice Resource Manager (SRM). SRM is responsible for scheduling resources for UE's belonging to its slice, while the RM layer role is to accommodate verbs to PRBs according to the number of resources allowed to each slice. The proposed two-level scheduling process has been implemented by partitioning MAC operations into two levels. The first, handled by SRM, ensures intra-slice traffic scheduling, while the second, managed by RM, is responsible for PRB assignment to UE's based on the mapping provided by each active SRM, amount of resources allowed for each slice, slice priority and the actual channel state (e.g. PRB availability). Communication between the eNodeB controller and RM is provided by an additional block called the Agent. The proposed two-level scheduling is preferable over joint scheduling due to lower complexity and no requirement for multidimensional scheduling. The proposed approach has been deployed and tested using Open Air Interface (OAI) in emulation mode, FlexRAN protocol used to configure eNodeB and the implementation of two-level scheduling.

Another interesting solution has been described in [109], where ORION, a novel RAN slicing system enabling dynamic on-the-fly virtualization of base stations, has been presented. It also provides flexible customization of slices in accordance with service demands allowing its usage in an end-to-end network slicing. The concept of a system is based on one central physical resources management entity called Base Station Hypervisor. It is responsible for managing RAN slices within physical BS's, ensuring their full isolation as well as facilitating efficient sharing of physical resources. RAN slices are realized through the creation of virtual base stations over the Hypervisor – specifically, each virtual base station is a composition of a virtual control plane that manages the user plane revealed to it by the Hypervisor. The virtual control plane also acts as a local RAN-level slice controller, responsible for resource allocation management to UEs attached to a slice.

4.2 Stochastic resource orchestration for multi-tenancy network slicing

Network slicing is a proposing technology to support diverse services from mobile users (MUs) over a common physical network infrastructure. In this study, we consider radio access network (RAN)-only slicing, where the physical RAN is tailored to accommodate both computation and communication functionalities. Multiple service providers (SPs, i.e. multiple tenants) compete in bidding for a limited number of channels across the scheduling slots, aiming to provide their subscribed MUs with the opportunities to access the RAN slices. An eavesdropper overhears data transmissions from the MUs. We model the interactions among the non-cooperative SPs as a stochastic game, in which the objective of an SP is to optimize its own expected long-term payoff performance. To approximate the Nash equilibrium solutions, we first construct an abstract stochastic game using the channel auction outcomes. Then we linearly decompose the per-SP Markov decision process to simplify the decision makings and derive a deep reinforcement learning-based scheme to approach the optimal abstract control policies. TensorFlow-based experiments verify that the proposed scheme outperforms the three baselines and yields the best performance in average utility per MU per scheduling slot.

4.2.1 System model

As shown in Figure 54, we focus on a system model with RAN only slicing, where an eavesdropper intentionally overhears the data transmissions of the MUs. The time horizon is divided into discrete scheduling slots, each of which is indexed by an integer $k \in \mathbb{N}+$ and is assumed to be of equal duration

δ (in seconds). RAN consists of a set B of physical BSs covering a service area, which can be represented by a set L of small locations, with each being characterized by uniform signal propagation conditions [110]. We use Lb to denote the serving area of a BS b ∈ B. For any two BSs b and b' ∈ B (b' ≠ b), we assume that Lb∩Lb'= Ø. We denote the geographical distribution of BSs by a topological graph TG = ⟨B, E⟩, where E = {eb,b': b ≠ b', b,b' ∈ B} with eb,b'= 1 if BSs b and b' are neighbours and otherwise eb,b'= 0. Suppose that ISPs provide both MEC and traditional mobile services to MUs while each MU can subscribe to only one SP. Let N_i be the set of MUs of a $SP_i \in I = \{1,\cdots,I\}$.

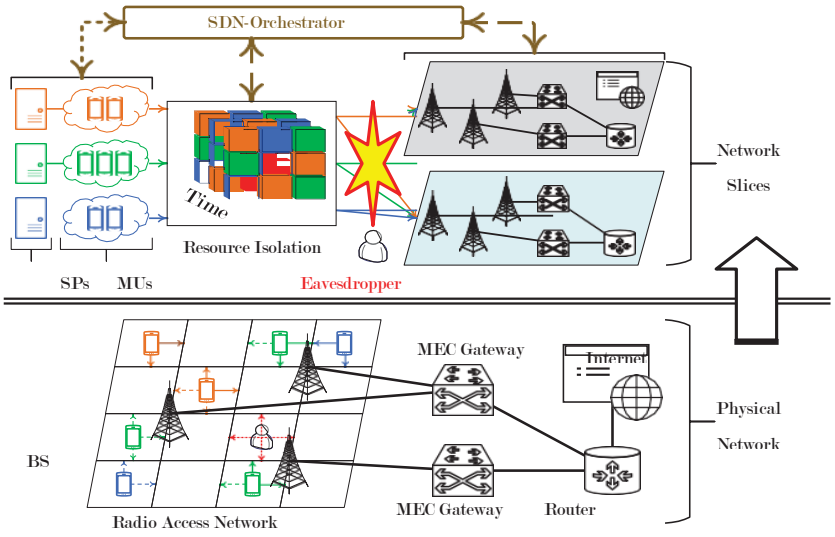


Figure 54: RAN slicing architecture.

Across the scheduling slots, the MUs and the eavesdropper move within L following the Markov mobility model [111]. Denote by $N^k_{b,i}$ the set of MUs of $SP_i \in I$ moving into the area of a BS b ∈ B during a slot k. We assume that a MU at a location can only be associated with the BS that covers the location. In the network, all MUs share a set J = {1, ⋯, J} of orthogonal channels with the same bandwidth η (in Hz). The SPs compete for the limited channel access opportunities for their MUs. Specifically, at the beginning of a scheduling slot k, each SP_i submits an auction bid $\beta^k_i = (v^k_i, C^k_i)$, where v^k_i is the valuation over $C^k_i = (C^k_{b,i}: b \in B)$ with $C^k_{b,i}$ is the number of requested channels in the service area of a BS b. After receiving $\beta^k = (\beta^k_i: i \in I)$, the SDN-orchestrator performs channel allocation and calculates payment τ^k_i for each SP_i. Let $\rho^k_n = (\rho^k_{n,j}: j \in J)$ be the channel allocation of a MU $n \in N = \cup_{i\in I} N_i$, where $\rho^k_{n,j} = 1$ if channel j is allocated to MU n ∈ N during slot k and $\rho^k_{n,j} = 0$, otherwise. We also apply the following constraints for centralized channel allocation at the SDN-orchestrator during a slot:

$$\left(\sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}^k_{b,i}}\rho^k_{n,j}\right)\cdot\left(\sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}^k_{b',i}}\rho^k_{n,j}\right)=0,$$

$$\text{if } e_{b,b'}=1, \forall e_{b,b'}\in\mathcal{E}, \forall j\in\mathcal{J}; \quad (1)$$

$$\sum_{i\in\mathcal{I}}\sum_{n\in\mathcal{N}^k_{b,i}}\rho^k_{n,j}\le 1, \forall b\in\mathcal{B}, \forall j\in\mathcal{J}; \quad (2)$$

$$\sum_{j\in\mathcal{J}}\rho^k_{n,j}\le 1, \forall b\in\mathcal{B}, \forall i\in\mathcal{I}, \forall n\in\mathcal{N}_{b,i}, \quad (3)$$

which ensure that one channel cannot be allocated to MUs associated with two adjacent BSs to avoid interference during data transmissions, while in the service area of a BS, one MU can be assigned at most one channel, and one channel can be assigned to at most one MU. Denote $\varphi^k = (\varphi^k_i: i \in I)$ as the winner vector at the beginning of a scheduling slot k, where $\varphi^k_i = 1$ if SP_i wins the channel auction and $\varphi^k_i = 0$ indicates that no channel will be allocated to the MUs of SP_i during the slot. The SDN-orchestrator determines φ^k via the Vickrey-Clarke-Groves (VCG) pricing mechanism [112], namely,

$$\begin{aligned} \boldsymbol{\phi}^k &= \arg\max_{\boldsymbol{\phi}} \sum_{i\in\mathcal{I}} \phi_i \cdot \nu_i^k \\ \text{s.t.} \quad & \text{constraints (1), (2) and (3);} \\ & \sum_{n\in\mathcal{N}_{b,i}^k} \varphi_n^k = \phi_i \cdot C_{b,i}^k, \forall b \in \mathcal{B}, \forall i \in \mathcal{I}, \end{aligned}$$

where $\phi^k_n = \sum_{j\in J} \rho^k_{n,j}$ and $\varphi = (\varphi_i: i \in I)$ with $\varphi_i \in \{0, 1\}$.

The payment τ^k_i for each SP_i can be calculated to be $\tau^k_i = \max_{\varphi - i} \sum_{i' \in I\setminus\{i\}} \varphi_{i'} \cdot v^k_{i'} - \max_{\varphi} \sum_{i' \in I\setminus\{i\}} \varphi_{i'} \cdot v^k_{i'}$, where $-i$ denotes all the competitors of SP_i.

Let $L^k_{n,(u)}$ and $L^k_{(e)} \in L$ be the geographical locations of a MU $n \in N$ and the eavesdropper during a scheduling slot k, respectively. As in [110], we assume that the average channel gains $H^k_{n,(u)} = h_{(u)}(L^k_{n,(u)})$ and $H^k_{n,(e)} = h_{(eu)}(L^k_{n,(u)}, L^k_{(e)})$ of links between MU n and the associated BS as well as the eavesdropper are determined by the respective distances. At the beginning of each scheduling slot k, MU n independently generates a random number $A^k_{n,(t)} \in A=\{0, 1, \cdots, A^{(max)}_{(t)}\}$ of computation tasks according to a Markov process [113]. We represent a computation task by $(\mu_{(t)}, \vartheta)$, where $\mu_{(t)}$ and ϑ are, respectively, the input data size (in bits) and the number of CPU cycles required to accomplish one input bit of the computation task. For a computation task, two decisions are available: 1) to be processed locally at the MU; or 2) to be offloaded to the MEC server in the computation slice for execution. The computation offloading decision for MU n at a slot k specifies the number $R^k_{n,(t)}$ of tasks to be transmitted to the MEC server. Then the remaining $A^k_{n,(t)} - \phi^k_n \cdot R^k_{n,(t)}$ tasks are to be processed locally. Meanwhile, a data queue at a MU buffers the packets from the traditional mobile service. Let W^k_n and $A^k_{n,(p)}$ be the queue length and the random new packet arrivals for MU n at the beginning of a slot k. We assume that the data packets are of a constant size $\mu_{(p)}$(bits) and the packet arrival process is independent among the MUs and identical and independently distributed across time. Let $R^k_{n,(p)}$ be the number of packets that are scheduled for transmission from MU n at scheduling slot k. The queue evolution of MU n can be written as the form below:

$$W_n^{k+1} = \min\left\{W_n^k - \varphi_n^k \cdot R^k_{n,(\mathrm{p})} + A^k_{n,(\mathrm{p})}, W^{(\max)}\right\}$$

where $W^{(max)}$ is the queue length limit.

To ensure security, the energy (in Joules) consumed by a MU $n \in N$ for transmitting $\phi^k_n \cdot R^k_{n,(t)}$ computation tasks and $\phi^k_n \cdot R^k_{n,(p)}$ data packets with a secrecy-rate [114] during a slot k can be calculated as:

$$P^k_{n,(\mathrm{tr})} = \begin{cases} \dfrac{\delta\cdot\eta\cdot\sigma^2\cdot\left(2^{\frac{\varphi^k_n\cdot\left(\mu_{(\mathrm{t})}\cdot R^k_{n,(\mathrm{t})}+\mu_{(\mathrm{p})}\cdot R^k_{n,(\mathrm{p})}\right)}{\eta\cdot\delta}}-1\right)}{H^k_{n,(\mathrm{u})}-H^k_{(\mathrm{e})}\cdot 2^{\frac{\varphi^k_n\cdot\left(\mu_{(\mathrm{t})}\cdot R^k_{n,(\mathrm{t})}+\mu_{(\mathrm{p})}\cdot R^k_{n,(\mathrm{p})}\right)}{\eta\cdot\delta}}}, & \text{if } H^k_{n,(\mathrm{u})} > H^k_{(\mathrm{e})} \\ 0, & \text{otherwise,} \end{cases}$$

where σ^2 is the background noise power spectral density. Let $\Omega^{(max)}$ be the maximum transmit power for all MUs, namely, $P_{n,(tr)} \leq \Omega^{(max)} \cdot \delta$, $\forall n$ and $\forall k$. For the remaining $A^k_{n,(t)} - \phi^k_n \cdot R^k_{n,(t)}$ computation tasks that are to be locally processed, the CPU energy consumption is:

$$P^k_{n,(\mathrm{CPU})} = \varsigma \cdot \mu_{(\mathrm{t})} \cdot \vartheta \cdot \varrho^2 \cdot \left(A^k_{n,(\mathrm{t})} - \varphi^k_n \cdot R^k_{n,(\mathrm{t})}\right)$$

where ς is the effective switched capacitance [115], and ϱ is the CPU-cycle frequency of the MU-end devices.

4.2.2 Stochastic game formulation

At a scheduling slot k, the local state of a MU n ∈ N is described as $\chi^k_n = (L^k_{n,(u)}, L^k_{(e)}, A^k_{n,(t)}, W^k_n) \in X = L^2 \times A \times W$, where the SDN-orchestrator broadcasts the information of $L^k_{(e)}$ to all MUs. Then $\chi^k = (\chi^k_n : n \in N) \in X^{|N|}$ characterizes the global network state, where |N| means the cardinality of the set N. Define by $\pi_i = (\pi_{i,(c)}, \pi_{i,(t)}, \pi_{i,(p)})$ a control policy of a $SP_i \in I$, where $\pi_{i,(c)}, \pi_{i,(t)} = (\pi_{n,(t)} : n \in N_i)$ and $\pi_{i,(p)} = (\pi_{n,(p)} : n \in N_i)$ are the channel auction, the computation offloading and the packet scheduling policies, respectively. The joint control policy of all SPs is given by $\pi = (\pi_i : i \in I)$. With the observation of χ^k at the beginning of each scheduling slot k, SP_i announces the auction bid β^k_i to the SDN-orchestrator and decides the $R^k_{i,(t)}$ computation tasks as well as $R^k_{i,(p)}$ packets to be transmitted following π_i. That is, $\pi_i(\chi^k) = (\pi_{i,(c)}(\chi^k), \pi_{i,(t)}(\chi^k_i), \pi_{i,(p)}(\chi^k_i)) = (\beta^k_i, R^k_{i,(t)}, R^k_{i,(p)})$, where $R^k_{i,(t)} = (R^k_{n,(t)} : n \in N_i)$ and $R^k_{i,(p)} = (R^k_{n,(p)} : n \in N_i)$. Accordingly, SP_i realizes an instantaneous payoff:

$$F_i\left(\boldsymbol{\chi}^k, \varphi^k_i, \mathbf{R}^k_{i,(\mathrm{t})}, \mathbf{R}^k_{i,(\mathrm{p})}\right) = \sum_{n \in \mathcal{N}_i} \alpha_n \cdot U_n\left(\chi^k_n, \varphi^k_n, R^k_{n,(\mathrm{t})}, R^k_{n,(\mathrm{p})}\right) - \tau^k_i$$

Where $\phi^k_i = (\phi^k_n : n \in N_i)$ and $\alpha \in R_+$ is the unit price to charge a MU n for achieving utility:

$$U_n\left(\chi^k_n, \varphi^k_n, R^k_{n,(\mathrm{t})}, R^k_{n,(\mathrm{p})}\right) = U^{(1)}_n\left(W^{k+1}_n\right) + U^{(2)}_n\left(D^k_n\right) + \ell_n \cdot \left(U^{(3)}_n\left(P^k_{n,(\mathrm{CPU})}\right) + U^{(4)}_n\left(P^k_{n,(\mathrm{tr})}\right)\right)$$

Taking expectation with respect to the sequence of per-slot instantaneous payoffs, the expected long-term payoff of a $SP_i \in I$ for a given initial global network state $\chi^1 = \chi \stackrel{\text{def}}{=} \chi_n = (L_{n,(u)}, L_{(e)}, A_{n,(t)}, W_n) : n \in N$ can be expressed as in:

$$V_i(\chi, \pi) = (1-\gamma) \cdot \mathsf{E}_{\pi}\left[\sum_{k=1}^{\infty} (\gamma)^{k-1} \cdot F_i\left(\chi^k, \varphi_i\left(\pi_{(\mathrm{c})}\left(\chi^k\right)\right), \pi_{i,(\mathrm{t})}\left(\chi^k_i\right), \pi_{i,(\mathrm{p})}\left(\chi^k_i\right)\right) | \chi^1 = \chi\right] \quad (4)$$

where $\gamma \in [0,1)$ is a discount factor. $V_i(\chi, \pi)$ is also termed as the state-value function of SP_i. The aim of each SP_i is to devise a best-response control policy π^*_i such that $\pi^*_i = \mathrm{argmax}_{\pi i}, V_i(X, \pi_i, \pi_{-i}), \forall \chi \in X^{|N|}$. Due to the limited number of channels and the stochastic nature in networking environment, we formulate the interactions among multiple non-cooperative SPs over the scheduling slots as a stochastic game, SG, in which I SPs are the players and there are a set $X^{|N|}$ of global network states and a collection of control policies $\{\pi_i : \forall i \in I\}$. A Nash equilibrium (NE), which is a tuple of control policies $\langle \pi^*_i : i \in I \rangle$, describes the rational behaviours of the SPs in an SG. For the I-player SG with expected infinite-horizon discounted payoffs, there always exists a NE in stationary control policies [116]. Define $V_i(\chi) = V_i(\chi, \pi^*_i, \pi^*_{-i})$ as the optimal state-value function, $\forall i \in I$ and $\forall \chi \in X^{|N|}$.

4.2.3 Deep reinforcement learning

From (4), it can be observed that the expected long-term payoff of an $SP_i \in I$ depend on the information of not only the global network state across the slots but also the joint control policy π. In other words, the decision makings of non-cooperative SPs are coupled in the SG, which makes it challenging to find the NE. In this section, we elaborate on how SPs play the SG only with limited local information.

To capture the coupling of decision makings among the SPs, we abstract SG as AG [33], in which an $SP_i \in I$ behave based on its own local network dynamics and abstractions of states at other competing SPs. Let $S_i = \{1, \cdots, S_i\}$ be an abstraction of the state space X_{-i}, where $S_i \in N_+$ and $S_i \ll |X_{-i}|$. It can be observed that the behavioural couplings in SG exist in the channel auction, and the payments of SP_i depend on X_{-i}. This allows SP_i to construct S_i by classifying the value region $[0, \Gamma_i]$ of payments into S_i intervals, i.e. $[0, \Gamma_{i,1}], (\Gamma_{i,1}, \Gamma_{i,2}], (\Gamma_{i,2}, \Gamma_{i,3}], \ldots, (\Gamma_{i,Si-1}, \Gamma_{i,Si}]$, where $\Gamma_{i,Si} = \Gamma_i$ is the maximum payment and we let $\Gamma_{i,1} = 0$ for a special case, in which SP_i wins the channel auction with no payment. With this regard, SP_i abstracts $(\chi_i, \chi_{-i}) \in X^{|N|}$ as $\tilde{\chi}_i = (\chi_i, s_i) \in = \tilde{X}_i \times S_i$ if the payment in previous scheduling slot belongs to $(\Gamma_{i,si-1}, \Gamma_{i,si}]$.

Let $\tilde{\pi}_i = (\tilde{\pi}_{i,(c)}, \pi_{i,(t)}, \pi_{i,(p)})$ be the abstract control policy in AG played by a $SP_i \in I$ over $\tilde{\chi}_i$, where $\tilde{\pi}_{i,(c)}$ is the abstract channel auction policy. Likewise, the abstract state-value function for SP_i under $\tilde{\pi} = (\tilde{\pi}_i : i \in I)$ can then be defined as in:

$$\tilde{V}_i(\tilde{\chi}_i, \tilde{\pi}) = (1-\gamma) \cdot \mathsf{E}_{\tilde{\pi}}\left[\sum_{k=1}^{\infty} (\gamma)^{k-1} \cdot \tilde{F}_i\left(\tilde{\chi}_i^k, \varphi_i\left(\tilde{\pi}_{(c)}\left(\tilde{\chi}^k\right)\right), \pi_{i,(t)}\left(\chi_i^k\right), \pi_{i,(p)}\left(\chi_i^k\right)\right) | \tilde{\chi}_i^1 = \tilde{\chi}_i\right]$$

$\forall \tilde{\chi}_i \in \tilde{\chi}_i$, where $\tilde{\chi}^k = (\tilde{\chi}^k_i = (\chi, s^k_i) : i \in I)$ with s^k_i being the abstract state at slot k and $F_i(\tilde{\chi}^k_i, \phi_i(\tilde{\pi}_{(c)}(\tilde{\chi}^k))$, $\pi_{i,(t)}$, $\pi_{i,(p)}(\chi^k_i))$ is the immediate payoff with $\tilde{\chi}^k = (\tilde{\chi}^k_i : i \in I)$ and $\tilde{\pi}(c) = (\tilde{\pi}_{i,(c)} : i \in I)$. In AG, a SP solves a single- agent Markov decision process (MDP). Suppose all SPs play $\tilde{\pi}^*$ in AG. Denote $\tilde{V}_i(\tilde{\chi}_i) = \tilde{V}_i(\tilde{\chi}_i, \tilde{\pi}^*)$.

There remain two challenges involved in solving the optimal abstract state-value functions for each $SP_i \in I$ using dynamic programming methods: 1) *a priori* knowledge of the abstract network state transition probability is not feasible; and 2) the size of the decision making space $\{\tilde{\pi}_i(\tilde{\chi}_i) : \tilde{\chi}_i \in \tilde{\chi}_i\}$ grows exponentially as $|N_i|$ increases. On the other hand, the channel auction decisions and the computation offloading as well as packet scheduling decisions are made in sequence and are independent across an SP and its subscribed MUs. We are hence motivated to decompose the per-SP MDP in AG into $|N_i|+1$ independent MDPs.

We can easily find that at a current scheduling slot, β_i of an $SP_i \in I$ needs $(s_i, P(s'|s,\iota-1))$ and $(U_n(\chi_n), z_n, L_n)$ from each subscribed MU $n \in N_i$, where $s' \in S_i$ and $\iota \in \{1, 2\}$. We propose that SP_i maintains over the slots a three-dimensional table Y^k_i of size $S_i \cdot S_i \cdot 2$. Each entry $y^k_{s,s',\iota}$ in Y^k_i represents the number of transitions from $s^{k-1}_i = s$ to $s^k_i = s'$ when $\varphi^{k-1}_i = \iota - 1$ up to slot k. Y^k_i is updated using the channel auction outcomes. Then, we estimate the abstract network state transition probability at a slot k as:

$$\mathbb{P}\left(s_i^k = s' | s_i^{k-1} = s, \phi_i^{k-1} = \iota - 1\right) = \frac{y^k_{s,s',\iota}}{\sum_{s'' \in \mathcal{S}_i} y^k_{s'',s',\iota}}$$

based on which $U_i(s_i)$, $\forall s_i \in S_i$ is learned via:

$$\mathbb{U}_i^{k+1}(s_i) = \begin{cases} \left(1-\zeta^k\right) \cdot \mathbb{U}_i^k(s_i) + \zeta^k \cdot \left((1-\gamma) \cdot \tau_i^k + \gamma \cdot \sum_{s_i^{k+1} \in \mathcal{S}_i} \mathbb{P}\left(s_i^{k+1} | s_i, \phi_i^k\right) \cdot \mathbb{U}_i^k\left(s_i^{k+1}\right)\right), & \text{if } s_i = s_i^k \\ \mathbb{U}_i^k(s_i), & \text{otherwise} \end{cases}$$

with $\zeta^k \in [0,1)$ being the learning rate.

Without *a priori* statistics of MU mobility and computation task as well as packet arrivals, Q-learning finds $U_n(\chi_n)$ for each MU $n \in N$ by defining the right-hand-side of:

$$\mathbb{U}_n(\chi_n) = \max_{R_{n,(t)}, R_{n,(p)}} \left\{(1-\gamma) \cdot U_n\left(\chi_n, \varphi_n\left(\tilde{\pi}^*_{(c)}(\tilde{\chi})\right), R_{n,(t)}, R_{n,(p)}\right) + \gamma \cdot \sum_{\chi'_n \in \mathcal{X}} \mathbb{P}\left(\chi'_n | \chi_n, \varphi_n\left(\tilde{\pi}^*_{(c)}(\tilde{\chi})\right), R_{n,(t)}, R_{n,(p)}\right) \cdot \mathbb{U}_n(\chi'_n)\right\}$$

as the optimal state action-value function Q_n: X×{0, 1} × A × W → R. In turn, we arrive at:

$$\mathbb{U}_n(\chi_n) = \max_{\varphi_n, R_{n,(t)}, R_{n,(p)}} Q_n\left(\chi_n, \varphi_n, R_{n,(t)}, R_{n,(p)}\right)$$

where an action (ϕ_n, $R_{n,(t)}$, $R_{n,(p)}$) under a current local state χ_n consists of the channel allocation, computation offloading and packet scheduling decisions.

The success of a deep neural network in modelling the Q-function inspires us to adopt a deep reinforcement learning (DRL) method [117]. We can then approximate the Q-function by a double deep Q-network (DQN) [118].

Mathematically, $Q_n(\chi_n, \phi_n, R_{n,(t)}, R_{n,(p)}) \approx Q_n(\chi_n, \phi_n, R_{n,(t)}, R_{n,(p)}; \theta_n)$, $\forall n \in N$, where we include in θ_n the set of parameters that are associated with the DQN of a MU n. During the DRL process, each MU $n \in N_i$ of a $SP_i \in I$ is assumed to be equipped with a finite replay memory to store the latest M historical experiences. To perform experience replay [119], MU n randomly samples a mini-batch $O^k_n \subseteq M^k_n$ to train the DQN parameters.

4.2.4 Numerical results

This section conducts numerical experiments based on TensorFlow to quantify the performance of the derived DRL-based scheme for multi-tenant cross-slice resource orchestration with secrecy preserving in a software-defined RAN. We set up an experimental network with 4 BSs being placed at an equal distance 1 km apart in the centre of a 2×2 km^2 square service area. The entire area is divided into 1600 locations, with each of 50×50 m^2. The average channel gains for a MU n ∈ N at the location $L^k_{n,(u)} \in L_b$ covered by a BS b ∈ B during a slot k are given by $h_{(u)}(L^k_{n,(u)}) = H_0 \cdot (\xi_0/\xi^k_{b,n})^4$ and $h_{(e)}(L^k_{n,(u)}, L^k_{(e)}) = H_0 \cdot (\xi_0/\xi^k_{n,(e)})^4$, where $H_0 = -40$ dB is the path-loss constant, $\xi_0 = 2$ m is the reference distance, while $\xi^k_{b,n}$ and $\xi^k_{n,(e)}$ are the distances between MU n and BS b as well as the eavesdropper.

The mobilities of all MUs, as well as the eavesdropper and the computation task arrivals of all MU, are independently and randomly generated. The packet arrivals follow a Poisson arrival process with average rate λ (in packets/slot). We design for each MU a DQN with two hidden layers with each consisting of 16 neurons. Other parameter values used in the experiments are listed in Table 3.

For comparison purpose, three baseline schemes are developed and simulated, namely:

1) Channel-aware control policy (Baseline 1) – At the beginning of each slot k, the need of getting one channel at a MU ∈ N is evaluated by $(H^k_{n,(u)} - H^k_{(e)})$,
2) Queue-aware control policy (Baseline 2) – Each MU calculates the preference between having one channel or not using a predefined threshold of the queue length,
3) Random control policy (Baseline 3) – This policy randomly generates the value of obtaining one channel for each MU at the beginning of each slot.

Parameter	Value
Set of SPs $\mathcal{I}$	$\{1, 2, 3\}$
Set of BSs $\mathcal{B}$	$\{1, 2, 3, 4\}$
Number of MUs $\lvert\mathcal{N}_i\rvert$	$6, \forall i \in \mathcal{I}$
Channel bandwidth η	500 KHz
Noise power spectral density σ^2	-174 dBm/Hz
Scheduling slot duration δ	10^{-2} second
Discount factor γ	0.9
Utility price α_n	$1, \forall n \in \mathcal{N}$
Packet size $\mu_{(\mathrm{p})}$	3000 bits
Maximum transmit power $\Omega^{(\max)}$	3 Watts
Weight of energy consumption ℓ_n	$3, \forall n \in \mathcal{N}$
Maximum queue length $W^{(\max)}$	10 packets
Maximum task arrivals $A^{(\max)}_{(\mathrm{t})}$	5 tasks
Input data size $\mu_{(\mathrm{t})}$	5000 bits
CPU cycles per bit ϑ	737.5
CPU-cycle frequency ϱ	2 GHz
Effective switched capacitance ς	$2.5 \cdot 10^{-28}$
Exploration probability ϵ	0.001
Replay memory size M	5000
Mini-batch size $\lvert\mathcal{O}^k_n\rvert$	$200, \forall n \in \mathcal{N}, \forall k$
Activation function	Tanh
Optimizer	Adam

Table 3: Parameters used in simulations.

With the three baselines, after the centralized channel allocation by the SDN-orchestrator at the beginning of each slot, a MU proceeds to offload a random number of computation tasks and schedule a maximum feasible number of data packets if being assigned a channel.

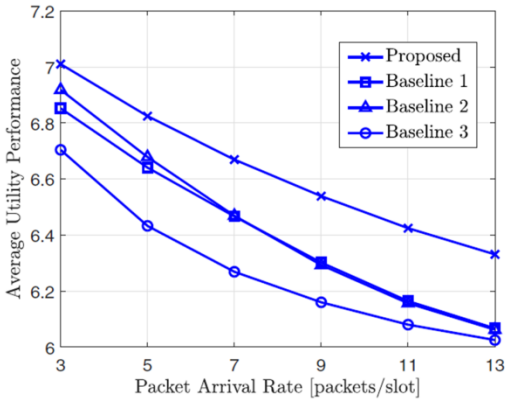


Figure 55: Average utility performance per MU across the learning procedure versus average packet arrival rates.

We first demonstrate the average utility performance per MU per scheduling slot achieved from the proposed DRL-based scheme and the three baselines under different average packet arrival rates. In this experiment, we assume that J = 11 channels are shared among the MUs for access to the computation and communication slices. The results are depicted in Figure 55, from which we can observe that the proposed scheme achieves a significant performance gain. However, the average utility performance decreases as the average number of random data packet arrivals increases. The reason behind is that in order to ensure secrecy, more data packet arrivals lead to larger queue length, more packet drops and higher energy consumption across the MUs.

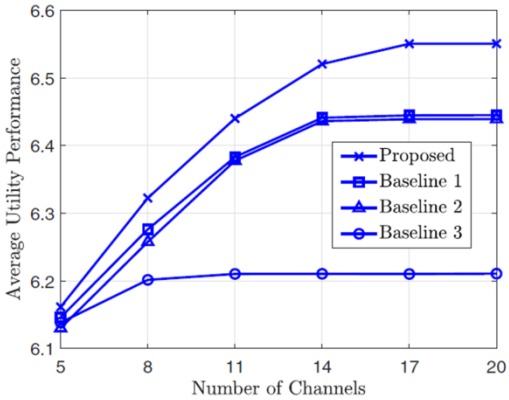


Figure 56: Average utility performance per MU across the learning procedure versus numbers of channels.

Then in Figure 56, we exhibit the average utility performance versus the number of channels, where the average packet arrival rate is fixed to be λ = 8. More channels available in the system provide more opportunities for the MUs to transmit the data of computation tasks to be offloaded and scheduled packets. Hence better average utility performance can be expected by the MUs. When there are sufficient channels in the network, the data transmissions of all MUs with secrecy preserving can be fully satisfied. Both experiments show that the proposed scheme outperforms the three baselines.

4.3 O-RAN extensions

The O-RAN approach (developed by the O-RAN Alliance [120]) is an open-source platform for building management and control of 5G RAN (NR) with generic IT hardware and standardized interfaces. The concept lies in adding some functional elements to the architecture while conforming to the 3GPP standards and also to propose extensions required by O-RAN functionalities. The activity of the O-RAN Alliance is twofold. The first area of activity of the consortium is related to O-RAN specifications. The second one deals with the open-source implementation of the concept by the O-RAN Software Community [121]. Access to O-RAN specifications is free, but it requires prior admission. Unfortunately, the O-RAN Alliance requires the specification to be kept confidential, and it cannot be

cited. The interested reader can, however, contribute to the project. The below description is based on publicly available information – presentations of O-RAN Alliance members, non-confidential documents available at the O-RAN website and O-RAN Software Community wiki.

The list of O-RAN uses cases include cross-layer traffic steering, QoE optimization, V2X proactive handover support (position prediction using navigation data), flight path-based dynamic UAV resource allocation, RAN energy-saving and energy-aware IoT operations, cross-layer RAN optimization, MIMO beam-forming optimization, and automation of RAN operations. The key idea of the concept is to adapt the Radio Resource Management (RRM) operations (admission control, mobility management, radio link management, advanced SON functions, etc.) according to applications' needs [122]. Due to the collection of monitoring data concerning UEs and network, the concept should enable the prediction of QoE, mobility pattern, cell traffic, network quality and users' distribution.

Despite not implementing RAN slicing in the current version of O-RAN, the platform seems to be the most promising solution in terms of the realization of the RAN slicing concept in the future. The reference architecture of the platform is presented in Figure 57. The concept uses two RAN Intelligent Controllers (RICs), non-Real-Time RIC and near-Real-Time RIC. The main functionality of the non-RT RIC is service and policy management, RAN analytics and model training for the near-RT RAN functionalities, essential for RIC near-RT runtime execution.

The near-RT RIC interacts with the RAN management system (i.e. OSS/BSS), which sometimes is referred in the documents to a **non-Real Time RIC (non-RT RIC).** The main functionality of the non-RT RIC is service and policy management, RAN analytics and model training for the near-RT RAN functionalities, essential for near-RT RIC run-time execution. The ONAP platform [123] is seen as OSS/BSS. ONAP functionalities important for O-RAN include orchestration of applications and the ability of control loop-based operations. These operations of non-RT RIC, in contrast to near-RT RIC, are much slower and typically used for semi-static, intent-based management operations for non-real-time management (reaction time >> 1 s) via the A1 interface, e.g. inventory/policy/configuration management. In some drafts, the non-RT RIC is responsible for SON operations. The O-RAN framework enables RAN components implementation in cloud environments. In such a case, O1 supports typical FCAPS (Fault, Configuration, Accounting, Performance, Security) and Service Management and Orchestration, and additional interface, i.e. O2 is specified for support of virtual resource management and other cloud-related management functions. Another interface between the management system and near-RT RIC, named O1, is used for the orchestration of xApps.

The **near-Real Time RAN Intelligent Controller (near-RT RIC)** interacts with RAN nodes (CU, DU) using O-RAN specific E2 interface. The interface is used for feedback loop-based RAN nodes control. The approach requires the collection of fine-grained monitoring data and implementation of decision engines responsible for taking required actions. The near-RT RIC can handle multiple RAN nodes, but no interface between near-RT RICs has been defined yet. The near-Real-Time RIC platform can be used by various applications. The application may deal with various aspects of RAN management and RRM: radio connections, mobility, interference, QoS, etc. Other functions may include AI-leveraged QoS management, connectivity management or seamless handover control.

This RIC provides RRM functionalities with embedded intelligence. It enhances original RRM functionalities such as per-UE controlled load-balancing, interference detection and mitigation, etc. It is assumed that in O-RAN, RRM functions can be fully installed in CU to enable proper RAN functionality in case of lack of RIC, but when near-RT RIC is present, its RRM mechanisms should be used by the CU.

The used gNB split is 7-2x (work on other splits is in progress). The reaction time of near-RT RIC is specified to be in the range of 10 ms – 100 ms. It allows for the deployment of near-RT RIC applications (called xApps) that may use two near-RT RIC databases, one consisting of information about UEs (UE-NIB) and another one consisting of information about RAN nodes (R-NIB). The near-RT RIC has a component for the mitigation of conflicts caused by xApps requests. It also contains a library of functions supporting AI-driven operations (model repository, inference), and each xApp may subscribe to the relevant parameters.

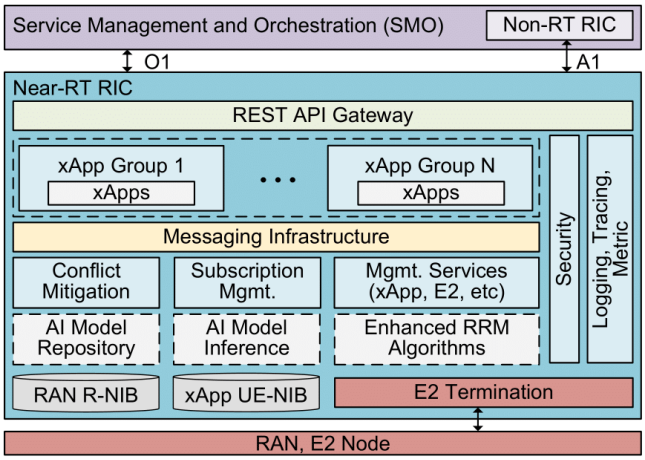


Figure 57: O-RAN reference architecture [124].

The work on O-RAN specification is still in progress, and not all details have been published yet; moreover, it seems that still exist some gaps in the concepts that have to be solved yet:

- Arbitrage of APPs reconfiguration orders. Multiple applications (called APPs) can send different reconfiguration goals at the same time. Therefore, an arbiter is needed to select, which one has higher priority than another one, or a new configuration should be enforced, which is a compromised way fulfils the goals of both (all) applications. There is also a need to find a balance between fulfilling the egoistic goals of an application (a typical reconfiguration goal) and the RAN operator goal, which has to satisfy all O-RAN customers according to their service classes, The O-RAN near-RT RIC has a component responsible for the coordination of multiple requests, but there is still lack of details.
- O-RAN stability problem. This problem is related to the observation of the stability of the feedback loop control that is implemented using near-RT RIC. Multiple control loops having different goals, delayed system response, or failures may lead to unstable and even chaotic behaviour of the nodes controlled by RIC. Moreover, delayed system response may lead to unstable behaviour of nodes controlled by the near-RT RIC. In some situations, several iterations are needed to achieve the goal. This may increase the system response time, and as a result, a ping-pong effect can be observed. This problem is not addressed by O-RAN.
- Cooperation of multiple near-RT RIC. So far, no details about such cooperation are provided. It can be realized in a hierarchical way using a “slow path” via the non-RT RIC or via the proposed peer-to-peer Y2 interface. The work on the Y2 interface is in progress (early stage). Due to the programmable nature of the near-RT RIC, each application should be aware of another instance of near-RT RIC and provide handling of Y2 at the application level. The preferred solution in order to handle users’ mobility between RICs is the stateless approach. If the stateless approach is not implemented, the state context has to be exchanged between RICs (in a similar way to MEC Application Mobility:
- R-NIB structure. So far, the R-NIB structure is not defined in detail. It should provide cell as well as UEs related measurements. In the case of tailoring of per-service oriented operations, its structure should include information, to which service (e.g. V2X or UAV) a specific UE belongs.
- 5G SON. In the LTE network, a Self-organizing Network (SON) concept that is used for self-configuration, self-healing and self-optimization has been defined. It seems that a similar set of functions can be identified for the purpose of O-RAN. So far, such functionality has not been identified. Moreover, the work on the 5G SON is in the early stage in 3GPP. The SON concept that provides 4G RAN management automation can and should be implemented in O-RAN. Such possibility has been identified but not defined in detail.
- So far, there are no restrictions to access R-NIB and UE-NIB databases by xApps. It raises isolation and security concerns. A creator of xApp is able to implement the gathering of confidential

information (associated with network operator or other xApps) about cells or UEs status (load, positions, etc.).

- It is necessary to define, which UE is used with which xApp. Such an assignment has to be done at the UE level. It is not specified how to make it.
- So far, O-RAN services do not interact with 5GC CP, so end-to-end O-RAN – 5GC services cannot be created.
- Despite the nonexistence of important mechanisms, the O-RAN concept seems to be the most promising one in terms of service-aware automated RAN and end-to-end operations. The O-RAN Alliance has started to work on O-RAN and NS integration, but the work is at the early stage yet.

In the following subsections, we will describe a proposal of O-RAN supporting RAN slicing and O-RAN, MEC, SON and network slicing integration.

4.3.1 Network slicing enabled O-RAN

The basic O-RAN reference architecture can and should be modified to include RAN slicing support. In line with the discussion presented in this paper, network slicing requires:

- modification of MAC in order to support scheduling of different traffic types (eMBB, URLLC, etc.), including mini-slot and grant free access mechanisms for support non-scheduled URLLC transmissions;
- customization of RRM to obtain per slice behaviour as a complementary mechanism to the scheduler that shares the radio link between different applications according to their SLA;
- using of R-NIB and slice information in order to proactively provide appropriate radio coverage and radio link quality on a per slice type level;
- partitioning of the Application Layer of the near-RT RIC in order to separate slice operation spaces and their privileges and restrictions. This includes access to only part of R-NIB that includes the information about UEs that are attached to the specific slice. In general, there will be operations performed by the O-RAN operator, operations that are typical for each slice type (i.e. common for multiple instances of such slice and operations specific for each slice instance.

The proposed near-RT RIC modifications are shown in Figure 58.

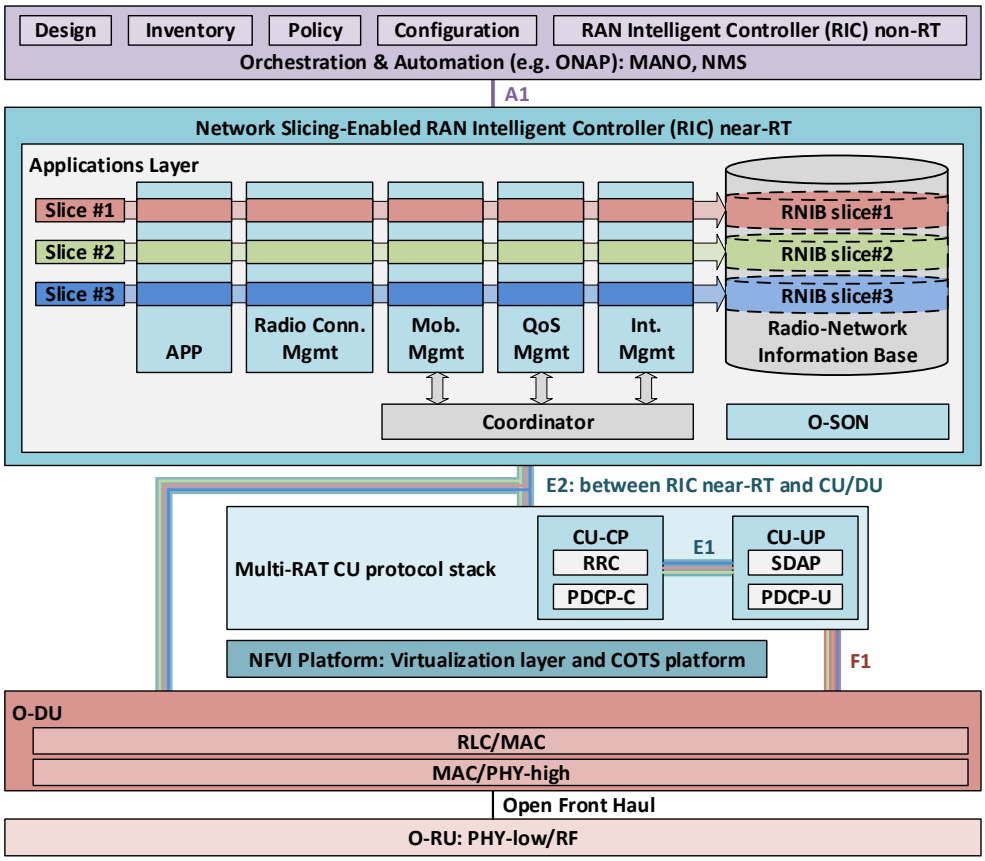


Figure 58: O-RAN with RAN slicing support.

In the proposed concept, the near-RT RIC functions are sliced in the way, in which all its functional elements are partitioned, and each slice has its own full constellation of these partitions composing the "virtual RIC" dedicated to the slice. The components are piggy-backed to the main component that realizes the function, for example, Main Mobility Management Application. The Main Function of each category has the following functionalities:

- basic category behaviour for the users that are attached to none slice;
- analyse the KPIs of its category for the whole area served by the near-RT RIC;
- implements policies provided by non-RT RIC for the category; collect orders from the piggy-backed functional components and executes their orders according to their priorities and system state;
- executes commands obtained from non-RT RIC, that includes manual control;
- it observes and records the commands sent to the underlying units (CU/DU) in order to evaluate system stability.

The main role of the piggy-backed per-slice functions is to customize the behaviour of its category via the interaction with the application (UAV, V2X) in order to provide manual control. In the case of R-NIB, each slice has its own view of R-NIB information relevant only to this slice (i.e. information only about UE that are attached to this slice only), which is isolated from the information that is relevant for other slices. This own view is exposed to the RIC applications of this slice as well as constraints related to the reconfiguration of the system capability.

The near-RT also consists of the Coordinator component. The role of the component is to evaluate the mutual impact of the categories (mobility, QoS) and using sophisticated algorithms (including AI-based ones) is trying to provide cross-category optimization. It also, using the situation-aware approach, is trying to predict congestion or KPIs degradation and takes countermeasures.

The SON component role is a reduced set of functionalities of the 4G SON. It is responsible mostly for the self-configuration of CUs and DUs and fault handling. The performance near-real-time

management that was part of 4G SON is now handled by other components of the near-RT RIC.

Virtual RICs expose their A1 interfaces to OSSes of their slices, and the master RIC exposes its A1 interface to the OSS of the RAN infrastructure owner. The virtual E2 interfaces of virtual RICs are located between the virtual RIC applications and their hosting counter-partners of the master RIC. The common E2 interface between the master RIC and lower layers contains the mediated and arbitrated communication related to all slices (Figure 58). O-DU implements scheduler-based RRM slicing mechanisms. The CU protocol stack may be per-slice differentiated.

- The main benefits come with the possibility of programmable RRM that benefits from RNIB and allows independent handling of each slice. To implement network slicing in O-RAN, the following modifications should be made:
- The scheduler has to be modified to support different types of network slices and multiple instances of slices;
- RRM should be aware of the existence of multiple slices, or multiple RRMs components have to be implemented. There should be a common mechanism and per-slice mechanisms dynamically deployed (orchestrated);
- The virtual per-slice near real-time RIC has to support each slice (or slice type operations separately;
- A different split for different slice types (combined with different transport options) can be used, but so far, O-RAN supports 7.2x only (work on splits 6 and 8 is in progress);
- There should also be some other mechanism deployed on a per-slice level that provides proactive way radio coverage for V2X, UAV or other services. Such use case-oriented mechanisms may be implemented as 3rd party application within the virtual RIC;
- The programmability of virtual RIC should be dynamic, i.e. 3rd party applications should be dynamically implementable. This way, the internal mechanisms can be changed during the RAN slice lifetime. Such programmability may be achieved through the implementation of a message bus for internal communication of all components – internal and hosted – of RIC.

4.3.2 O-RAN, network slicing, SON and MEC integration approach

The O-RAN Alliance [120] is working on a programmable solution that automates NR operations. The RAN automation in the 4G network has been provided by the Self-Organizing Network (SON) concept, which implements the feedback loop-based real-time management [125]. Unfortunately, the 5G-SON development is at an early stage. Yet another noteworthy concept is Multi-access Edge Computing (MEC), which primary role is shortening UP data paths to minimize the communication latency and optimize data traffic distribution by dynamic deployment of applications closer to the edge. MEC also facilitates the exploitation and provision of information related to RAN and User Equipment (UE) via APIs. Whereas MEC is already well-defined for 4G networks, its integration with the 5G network, especially with NS, is still in progress [126]. Our analysis of O-RAN, MEC, SON and NS approaches has shown that these technologies are partly both complementary and overlapping. Moreover, some mechanisms developed within one solution can be efficiently reused by others. These observations motivate to present an integrated concept of O-RAN, MEC, SON and NS. The focus has been laid on O-RAN and providing extensions to its existing architecture, which so far is not complete and does not support SON, MEC or NS.

- SON provides 4G and 5G RAN management automation [125], which includes self-configuration of newly deployed base stations, performance optimization and fault management. Self-optimization mechanisms concern coverage, capacity, handover, QoS, energy consumption and interference control. Self-healing includes automatic detection and mitigation of failures. SON is based on feedback loops; therefore, for its implementation, it is necessary to monitor RAN and reconfigure it in near real-time on that basis. The list of SON functions has been completely defined for LTE. However, for 5G RAN (NR), the work is still in progress – some of the 5G-SON services and related

procedures have already been specified. There is no detailed architecture of SON provided by 3GPP. In general, it is assumed that SON is a part of the management system (OAM); however, its implementation allows for the distribution of SON functions. It is assumed that OAM provides SON with relevant measurements, information about alerts and allows it to reconfigure network nodes or functions [125]. In 4G, the NM-Centralized SON is implemented as a part of the Network Management system (i.e. OSS/BSS), whilst in EM-Centralized SON, the SON algorithms are executed at the Element Management level. According to [127], 5G SON algorithms can operate on different levels of the network: (i) in the Cross-Domain Layer, (ii) in the Domain Layer and (iii) at the Network Function Layer. Accordingly, four types of SON are distinguished: Cross Domain-Centralized SON (C-SON) and Domain-Centralized SON that both execute in the management system, the Distributed SON (D-SON) located in the Network Function layer and Hybrid SON as a combination of the aforementioned. SON can use the Management Data Analytics Service (MDAS) [127], [128]. It is expected that SON will also operate in 5GC and address the NS (resource allocation optimization, collecting slice relevant data, solving inter-slice issues, etc.), but the work is still in progress. Despite the standardization efforts, the deployed SON solutions are vendor-specific and not interoperable. One of the issues with SON is the lack of detailed implementation architecture and interfaces. Neither SON monitoring database nor ways of SON functions' conflicts resolutions have been defined. So far, the SON concept does not use the NFV paradigm or orchestration of SON functions and is only mentioned in O-RAN documents.

- MEC by ETSI is dedicated to the standardization of an open environment for the integration of various applications across multi-vendor computing platforms tightly integrated with the multi-technology RAN. The synergy of IT and telco worlds at the edge of the communication network gives numerous benefits related to re-shaping the overall use case-related architecture, i.e. locating the applications near the customer, receiving contextual information from RAN as well as optimizing the traffic distribution, resources utilization and network performance. Within the MEC architecture [129], two major parts can be distinguished: MEC system-level comprised of OSS, applications/infrastructure orchestration entities and application life cycle management API proxy, and MEC host-level consisting of MEC Platform (MEP) that hosts MEC applications and exposes API to them, MEC Platform Manager responsible for the management of platform itself as well as applications life cycle, Virtualization Infrastructure and its Manager and finally the underlying network (e.g. local, external or 3GPP network). The fundamental mechanisms of MEC are (i) seamless inter-platform application mobility, platform services APIs for e.g. users' location and radio conditions contexts exposure, (ii) underlying data network traffic steering for selective applications-related data redirection or (iii) application implementation and orchestration. The architectural framework allows for MEC implementation with or without NFV. The concept was developed for the 4G network. Hence, it is not well integrated with 5G and NS yet. MEC platform APIs expose RAN data to MEC applications; however, in contrast to O-RAN, MEC is unable to influence the RAN configuration.

The presented analysis of O-RAN, SON, MEC and NS show overlaps and complementarities between them. Their proper integration with a new decomposition can bring essential benefits in terms of the removal of redundant functional blocks and providing overall synergy. However, a new functional decomposition is required to reduce the overall complexity and enable cross-layer operations. It is also noteworthy that non-integrated implementation of the analysed system may lead to conflicting decisions able to degrade system performance. For instance, the MEC platform may adapt the MEC application to NR conditions (without impact on the NR behaviour). At the same time, the O-RAN may try to adapt NR to the application needs. The analysis of the technologies presented in Section 4.3 has led us to the following conclusions regarding the benefits of their mutual integration:

- SON and O-RAN integration: The benefits of integration lie in the usage of the same servers (hosts or edge data centre), monitoring databases and the ability of cross-operations – the SON decisions can be re-used by xApps. The SON functions can be implemented as semi-permanent xApps, which interact with other apps exposing their information and services. They implement the operator's, not services' goals.

- MEC and O-RAN integration: MEC hosts can be combined with the near-RT RIC hosts, and the MEC-based CP services can be O-RAN services. MEC databases (about UE locations, cell performance, RNIS) can be integrated with O-RAN databases, and MEC Application Orchestrator (MEAO) can be used for xApps orchestration, as it provides application mobility. This mechanism should be reused in a multi-near-RT RIC environment, solving an essential problem of the inter-near-RT RICs cooperation. Each near-RT RIC/MEC host or edge data centre should contain the MEP where MEAO could orchestrate MEC UP functions.
- Network Slicing and O-RAN integration: The NS is the missing feature of O-RAN. It impacts O-RAN in several ways. First, the slice xApps have to be defined in NR slice templates (a new feature currently absent in 3GPP NR slicing). Second, each slice needs a separate partition of the database containing information about the NR nodes and attached UEs. These partitions should also keep information about slice-level KPIs/KQIs. Third, the UE attachment to slices, in line with 3GPP specifications, can be performed by NSSF and UDM as defined in [62]. The NS approach solves the problem of individual UE handling by near-RT RIC and security issues. The E2 interface has to be modified to support NS by the PRB scheduler. The end-to-slice template should define the interactions between NR xApps and the 5GC sub-network slice counterparts, solving this way the problem of lack of cooperation between O-RAN (xApps) and 5GC. The CU/DU nodes should be modified to support NS.

A new architecture integrating the mentioned technologies is presented in Figure 59. As already stated, we assume immersive integration of several technologies, but for the clarity of the description, the terminology introduced by these technologies will be used. We propose to use the same host (or edge cloud) for the virtualized implementation of integrated and modified near-RT RIC, SON, MEC and NS functions called “Integrated near-RT RIC” (I-near-RT RIC). We assume the communication within the I-near-RT RIC via a message bus. Hence, no interactions between the components of RIC need to be described. The most shared components of the architecture are the two databases: (i) R-NIB with information about the NR nodes and (ii) UE-NIB with information about all UEs in the area served by the I-near-RT RIC (both databases’ names follow the O-RAN terminology). The information stored in both databases is used by the Management Data Analytics Service (MDAS) component that provides data analytics and predictions as defined in [128], also at the NR slice level. It is assumed that the MEP API interfaces to both databases are provided (i.e. RNIS, LS).

The R-NIB, UE-NIB and MDAS information is used by the O-SON functions – “xSApps” (SON-dedicated xApps) installed in all I-near-RT RICs (a distributed SON approach) using the orchestrator. Their goal is NR management automation, as defined by 3GPP. The SON functions can be dynamically deployed/updated. They interact with non-RT RIC for policy-based management. The non-RT RIC is responsible for communication between SON components of I-near-RT RICs (if needed). The xSApps can also expose their APIs to xApps for sharing the NR nodes’ status (e.g. failure) or management policy preferences. O-SON is also responsible for resource allocation to slices. All xApps and xSApps reconfiguration requests go through the Coordinator/Stability Observer component. Its role is resolving requests conflicts based on priorities but also observing the system stability by monitoring variance of predefined KPIs, and restoring the last stable configuration, if necessary. Moreover, it identifies troublesome xApps/xSApps and alerts the non-RT RIC, which can decide to stop them. The Coordinator component is already defined in near-RT RIC, but its combination with the Stability Observer is missing.

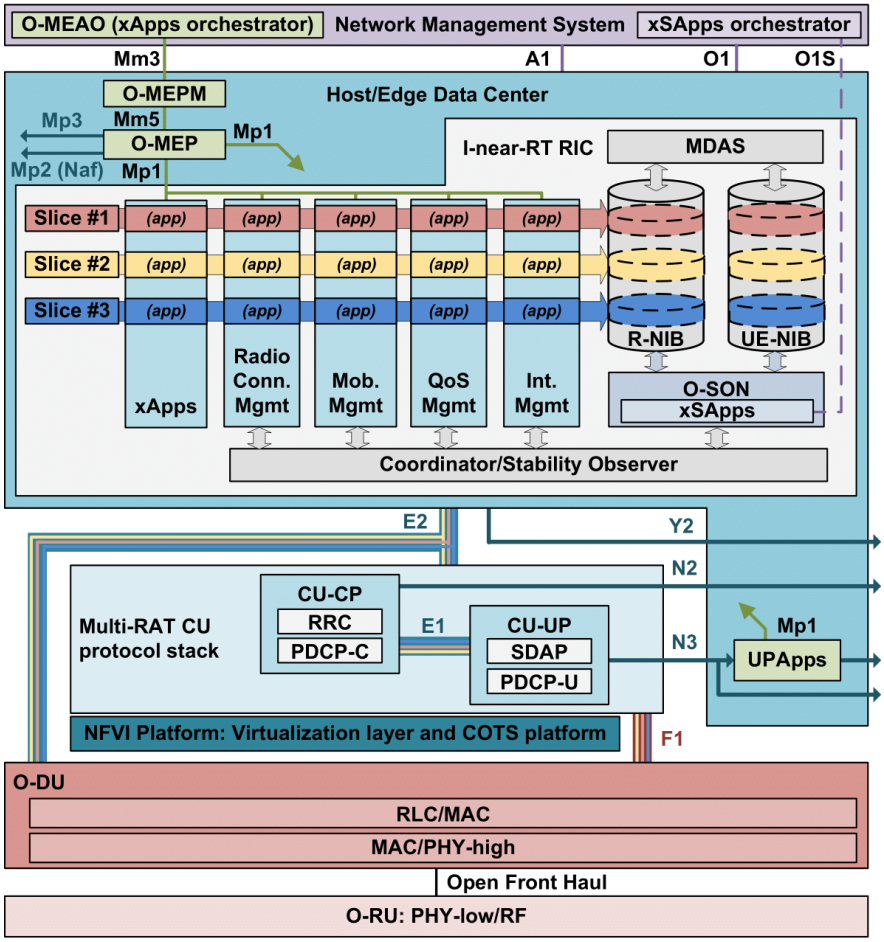


Figure 59: The integrated O-RAN, SON, MEC platform showing internal components of the I-near-RT RIC.

The implementation of NS alters the near-RT RIC architecture and O-RAN interfaces functionality but also solves some of the O-RAN issues. First, using slice-allocated xApps enables advanced operation of NR on per-slice level, as defined in O-RAN specifications – the NR is no more only a set of pipes of differentiated QoS (e.g. eMBB, URLLC). It also provides differentiation of i.a. mobility-related operations. Second, the use of NS provides the interaction between 5GC (sub-)network slice and NR (sub-)network slice using native 5GC NS-related and already defined mechanisms. Using them, both sub-network slices can be stitched together, while user authentication and slice selection can proceed according to 3GPP [62]. An important novelty is the existence of VNFs in NR on a slice level. This changes the way, in which end-to-end 5G slices should be orchestrated. The concept of integration of NR sub-network slices with 5GC sub-network slice in the case of two I-near-RT RICs is shown in Figure 60. In the figure, the only shown components of 5GC and Network Management System (NMS) are these essential for NS runtime operations. In the proposed concept, the I-near-RT RIC functions are sliced in a way, in which all its functional elements are partitioned, and each slice has its own full constellation of these partitions, forming the “virtual RIC” dedicated to the slice. Each xApp deployed in I-near-RT RIC belongs to a slice, and the subscription rules (access to databases), customized MDAS functions and policies related to resource allocations have to be enforced for the slice. The xApps are piggy-backed to the main component that realizes the function, e.g. Main Mobility Management Application. NS requires modification of MAC to support scheduling of different traffic types (eMBB, URLLC, etc.), including minislot and grant-free access mechanisms for support of non-scheduled URLLC transmissions, customization of RRM to obtain per-slice behaviour as a complementary mechanism to the scheduler that shares the radio link between different applications according to their SLA, using of R-NIB and slice information to provide proactively appropriate radio coverage and radio link quality per slice type. The E2 interface can be used for NS information exchange with other nodes (CU/DU). Other interfaces, i.e. between the components of 5GC, RAN or MEC, are compliant with their original definitions presented within ETSI or 3GPP normative documents. The detailed specification of the

implementation of RRM mechanisms for NS is out of the scope of this paper.

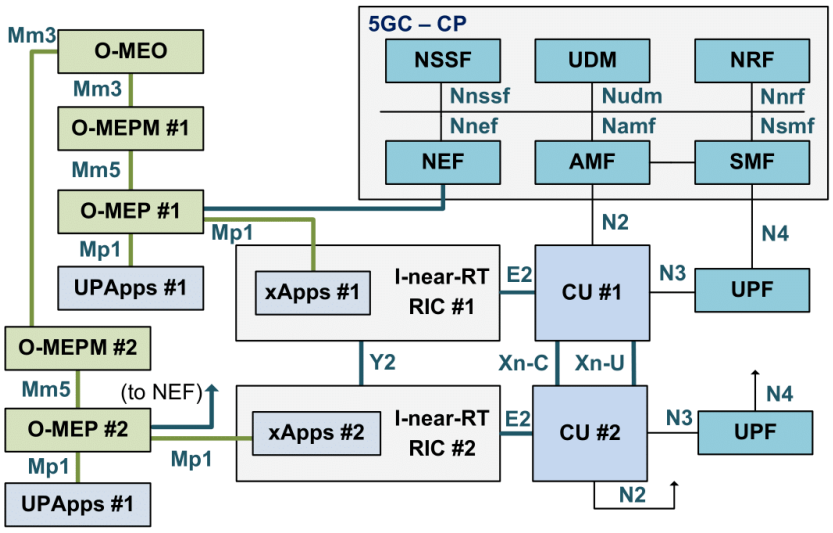


Figure 60: Overview of integrated O-RAN, 5GC-CP and MEC for two I-near-RT RIC domains.

3GPP has already specified the NMS role in NS [130]. The operations of NMS have to be altered to cope with the NR components virtualization. So far, the virtualized implementation of I-near-RT RIC has only been assumed, but CU and DU can be virtualized as well. The use of SON decentralizes the management operations and simplifies the NMS. The I-near-RT RIC implements MEC services in a different way that is described in 3GPP specifications. However, it is worth recalling that the work on the integration of NFV-based MEC for 5G networks is still ongoing. In our concept, the CP MEC applications are directly implemented just as xApps. MEP is integrated with 5GC CP through the Mp2 interface (as a specific Application Function). Hence, it is seen as Naf by the 5G CP. MEC is decomposed, and the core part is called "O-MEP". The MEP APIs are now provided by the R-NIB and UE-NIB databases.

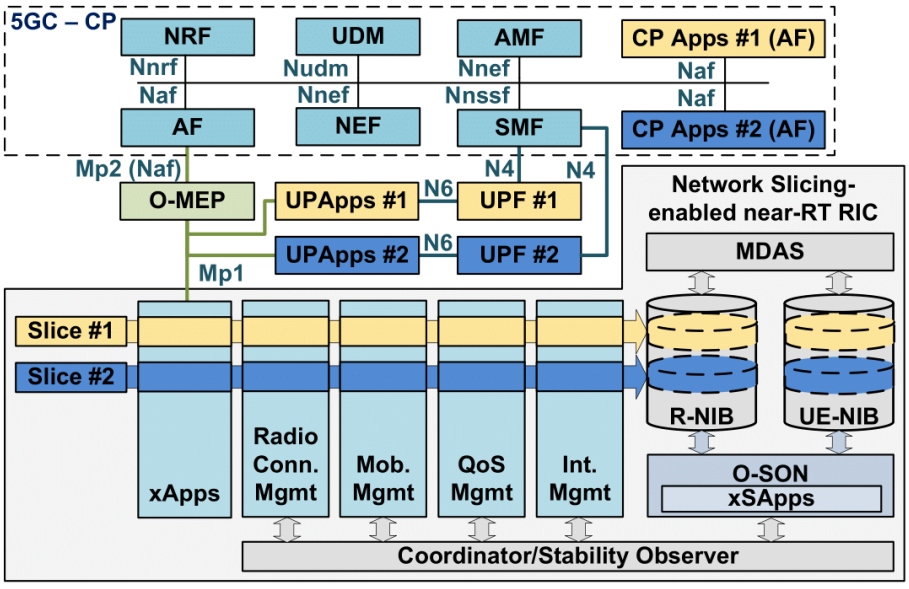


Figure 61: Overview of two end-to-end slices deployed within one I-near-RT RIC domain.

Moreover, MEC xApps may use data analytics services offered by MDAS. The VNFM functionality provided by MEAO is not needed, as it can be provided by the non-RT RIC. The application mobility mechanism of MEC is essential in our concept for slice-level peer-to-peer communication between I-near-RT RICs. It enables providing UE mobility support by migration of slice-level xApps to another I-near-RT RIC or UE context transfer. It is provided by the modified MEAO called (O-MEO). The MEC

concept brings a significant disruption to the O-RAN architecture regarding UP functions orchestration. L4-L7 operations can also be programmed, providing the well-known benefits of MEC related to traffic redirection and application-level processing by the edge-based application server. In contrast to O-RAN xApps, UP applications (UPApps) can handle the UP traffic. However, in some cases, a tandem of xApp and UPApp is required to implement MEC-like services, as access to I-near-RT RIC databases and mechanisms as well as the simultaneous UP processing of user data is needed. The interface between both components does not need to be defined, as it is application-specific.

In Figure 61, we have shown a case of two slices with CP (xApp) and UP RAN (UPApp) applications. The figure also shows the integration of the slices with 5GC. Due to the incorporation of modified MEC within the architecture, the communication between applications belonging to the same end-to-end slice (xApps, CPApps, UPApps) can be established using Mp1 and Mp2 (Naf) interfaces.

4.3.3 Conclusions

In this section, we have described O-RAN, SON, MEC and network slicing technologies, emphasizing synergies between them but also identifying the overlapping components of their architectures. Based on the analysis presented in Section 4.3.1, the integration concept of these technologies, which heart is the I-near-RT RIC, has been described. We have shown that the O-RAN-centric approach is beneficial, and such integration solves some of the issues not well-addressed by O-RAN yet. As we have shown, due to the integration, some components of the contributory technologies can be removed or reused. Naturally, the presented concept is a very high level one, as it concerns the integration of very complex and not yet fully specified systems. However, we deeply believe that it will blaze the trail of technological integration as its potential benefits are indisputable. The work on this paper has suffered due to the confidentiality of the O-RAN specification. We hope that the O-RAN Alliance policy will be changed in the nearest future, thus attracting more scientists interested in contributing to the evolution of the O-RAN approach.

4.4 6G-LEGO – a network slicing framework for beyond 5G networks

At present, most network slicing concepts, including the 3GPP one, use a centralized management and orchestration approach defined by ETSI NFV (Network Functions Virtualization) MANO (Management and Network Orchestration) framework [131]. In ETSI NFV, a network slice is simply considered as an NFV Network Service (NS), among others. The MANO framework is responsible for analysing the abstracted description of a slice, for providing optimal placement of slice virtual functions within the infrastructure, and for the dynamic allocation of resources to slices during their run-time.

The ETSI NFV framework enables the dynamic deployment of network functions as software-only entities (termed as virtualized network functions, VNFs), abstracted from the underlying hardware, and is intended for telco-oriented implementation of softwarised communication networks. The ETSI NFV MANO [132] is responsible for the lifecycle management and dynamic resource allocation to Network Services (i.e. network slices in ETSI terminology) that are implemented as a set of interconnected VNFs. The NFV framework is VNF functionality agnostic and provides no specific support for network slicing except the “priority” parameter of Network Service Deployment Flavour for resource shortage conflict resolution [133]. The Operation/ Business Support System (OSS/BSS) that is placed atop of the MANO stack plays a key role in the whole ETSI NFV picture. It is responsible for run-time slice management. However, the functionality of OSS/BSS had not been defined by ETSI till NFV Release 3. The mentioned release [131] has introduced slice-related management functions of the OSS/BSS (at the levels of communication service, network and sub-network slice) identically to the 3GPP vision [128]. Moreover, OSS/BSS handles user subscriptions, performs policy-based management of slices and services, provides Key Performance Indicators (KPIs) monitoring for Service Level Agreement (SLA) fulfilment, collects accounting data, etc.

The concept of network slicing, however, requires more mechanisms than those currently defined by NGMN or within NFV MANO. The gaps include mechanisms for slice description, slice selection and

matching, interactions between the slice provider and slice tenants, to mention a few. Some of them can be also specific to the networking technology, e.g. Radio Access Network (RAN). Several of the mentioned issues have been solved by 3GPP; others are handled by the ITU-T Study Group 13, which has already published some recommendations [134], [135], [136].

The proposed by 3GPP 5G network slicing approach [62] follows the NGMN and ETSI NFV concepts. It allows the user to be attached to up to 8 slices simultaneously [63]. The User Plane Function (UPF) chain may be considered as a "dedicated user plane". The control plane, based on Service-Base Architecture (SBA), supports a flexible extension of the control plane but also has components that support network slicing-related operations. The 5G System exposes the control plane services to external systems via the Network Exposure Function (NEF) [62], [137]. The Network Slice Selection Function [62], [138] plays an important role in support of network slicing by assisting with the selection of the Network Slice Instances that will serve a particular UE (User Equipment). Another key function related to network slicing is the Network Slice-Specific Authentication and Authorization Function (NSSAAF), which aids the Access and Mobility Management Function (AMF) in the verification of UE's rights to attach to a specific slice. The procedure is executed during the UE registration and is performed by means of the Extensible Authentication Protocol [62], [139], [140]. Network slicing support is also provided by the Session Management Function (SMF), which selects the appropriate UPF based on the Single–Network Slice Selection Assistance Information (S-NSSAI). The 5G System signalling procedures, e.g. admission control, handover, session management, are slice aware. Enhancements for RAN slicing are scarcely under study [141]. The 3GPP approach to 5G System orchestration and management [130], [142], [143] supports network slicing (at the level of network function, sub-network slice, network slice and communication service management), referring to ETSI NFV mechanisms, with the exception of slice selection and authentication mechanisms. The 3GPP management system architecture is complementary to the ETSI NFV MANO stack [128]. It allows the slice operator (tenant) to obtain selected management data and to subscribe to slice management operations [130].

The 3GPP approach has many limitations, e.g. it provides no separation of the management plane of slices, and it is significantly complex to implement. Its inclusion has an impact on multiple components and protocols of the 5G network architecture, while the centralization of slice management and orchestration functions raises significant scalability concerns. It is worth noting that this concept has never been commercially deployed yet.

The 6G-LEGO concept presented in the section aims at solving some of the above-mentioned limitations of 5G network slicing. It provides a clear separation of management planes of deployed network slice instances (NSIs). To that end, each NSI has its own management as well as "in-slice" mechanisms of users' authentication. Moreover, it allows for the creation of sub-network slices per domain and their concatenation in order to obtain an end-to-end slice or to add services to slices. To efficiently achieve that goal, 6G-LEGO sub-network (i.e. single-domain) slices are self-contained and self-described. Due to the proposed separation of concerns, the interactions between functional blocks of the architecture are minimized. The architectural approach of the 5G!Pagoda project [144] serves as the basis for our work. This architecture allows for hierarchical orchestration, the tenants can directly manage their slices, and the management plane is a part of each slice [145]. Independent slicing of each plane (i.e. user, control and management planes) and recursive vertical stitching of them are allowed. To that aim, the concept of a common slice, which consists of functions that can be used by any other slice, is introduced. Moreover, slice-level operations such as vertical or horizontal slice stitching, slice selection, matching or exposing an abstracted view of a slice are allowed.

4.4.1 6G-LEGO concept description

The different implementation options, as well as the shortcomings of the current 3GPP approach to network slicing, as described in Section 4.4, have been driving the development of the 6G-LEGO framework. The concept can be implemented in any type of networking solutions beyond 5G, and we see it as a candidate for 6G networks. Due to the fact that this concept considers slices as "bricks", we

have called it 6G-LEGO. In order to do that efficiently, the term “slice” has been slightly redefined. In 6G-LEGO, a slice is a set of interconnected logical entities that are grouped for a specific purpose and are implemented using isolated resources. Such an extended meaning of slices enables the creation of not only the communication network but also the management or service platform in the form of a slice. Moreover, we are able to operate at the slice level by adding services to slices, etc. In contrast to the existing approaches, the framework defines the slice template as an object that is ready to be deployed with minimal external support.

The 6G-LEGO framework introduces the following novel features:

- We see the use of a common OSS/BSS for all slices as an obsolete approach inherited from hardware-based networking solutions that is raising scalability issues, provides weak isolation of slice management operations and requires OSS/BSS modifications. The added OSS/BSS functions should support the run-time management of each deployed slice instance. Instead, we propose that each slice has its management plane implemented as a part of the template in a similar way as other planes of the slice. The local, in-slice OSS/BSS cooperates with a central (global) OSS/BSS to achieve the overall management goals. The in-slice OSS/BSS provides a management interface to the slice tenant that can be used for slice run-time management and orchestration of additional slice functions. The local OSS/BSS is not generic, but it can be customized for a specific slice type (template). As it will be typically implemented using VNFs, its functionalities can be dynamically updated, providing that way management plane programmability. The in-slice OSS/BSS may interact with the MANO orchestrator in order to proactively allocate resources (based on QoS/QoE objectives) or to update the network slice template. The proposed feature increases the isolation between the slices and improves the overall system scalability as it reduces the interaction between the slice and OSS/BSS. The role of the external (central) OSS/BSS system is significantly reduced since it is now used mostly for analysis of slice KPIs and accounting (a common practice of operators in current networks). The In-Slice Management (ISM) [146] concept simplifies the integration of a slice and makes the slice orchestrator slice-agnostic. The slice template includes nearly all functionalities needed to run the slice. In the case of multi-domain slice management, a hierarchical approach is proposed, in which the necessary inter-domain management functions are implemented as a set of VNFs that interact with OSS/BSSes of all single-domain (sub-network) slices (i.e. ISMs) that compose the end-to-end-slice.
- Due to the implementation of the in-slice OSS/BSS, the central OSS/BSS is slice agnostic, and the orchestrator is focused mostly on resource-oriented operations. The orchestrator for the purpose of a specific slice can be driven by its in-slice OSS/BSS (ISM). That allows for the implementation of a proactive method (described earlier in the paper) of resource allocation. Such an approach improves the scalability of the orchestrator.
- The proposed definition of slices allows their concatenation in a horizontal and vertical way. The vertical slice concatenation corresponds to the attachment of services and management functions to slices of the same domain. The horizontal concatenation of sub-network slices can be used if the end-to-end slices are composed of multiple, single-domain slices. A sub-network slice is typically created in each technological or administrative domain. For concatenation of sub-network slices, the framework provides supporting functions, called Slice Operations Support (SOS), that are part of each slice and are programmed during the end-to-end slice template preparation process.
- We propose a hierarchical end-to-end orchestration of slices where the inter-domain orchestrator is playing a key role. This orchestrator is not a MANO orchestrator, but it is an entity that interacts with domain-level orchestrators as slaves. Before slice deployment, it interacts with the Sub-network Slice Configurator, whose main role is the preparation of the end-to-end slice template before its deployment. The preparation means the modification of some parts of the sub-network slice templates, especially the in-slice OSS/BSS and the SOS parts, according to the need for end-to-end slicing. For example, a selected subnetwork slice

may include mechanisms responsible for users' authentication, whereas these mechanisms are removed for other slices. Furthermore, some mechanisms that enable the stitching of neighbouring slices can be added to the slice template by the inter-domain orchestrator.

The proposed structure of the 6G-LEGO slice together with slice-related operations is described in the next subsections.

4.4.1.1 The structure of 6G-LEGO slice templates and their usage

The slice template comprises a set of functions that implements autonomic operations of a slice or sub-network slice (cf. Figure 62) to minimize the interactions with other components of the framework. To that aim, the slice template includes embedded functions responsible for the management and orchestration support of the slice (in-slice OSS/BSS, i.e. ISM) [146]. The benefits of this approach are described later in this section.

The 6G-LEGO slice template also includes functions responsible for slice operations – support for slice discovery, users' authentication, mobility, and proper traffic redirection for horizontal (for inter-domain operations) and vertical stitching of sub-network slices (e.g. for adding services to the deployed slices). In order to adapt the slice operations to the needs of the inter-slice communications, the functions responsible for slice operations have to be modified before the slice deployment by the orchestrator operator. These functions of a slice are grouped in the Slice Operations Support (SOS) [145].

Each 6G-LEGO slice template is composed of: i) Core Slice Components, ii) the slice management part, called Slice Manager (SM); iii) Resource Orchestration Support (ROS); and iv) the Slice Operations Support (SOS) part. The role of each part of the sub-network slice is described below.

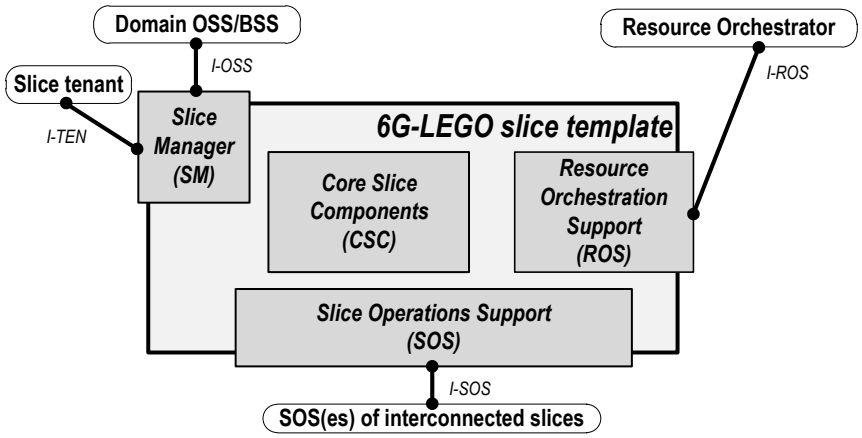


Figure 62: Generic 6G-LEGO slice template structure.

4.4.1.1.1 Core Slice Components

The Core Slice Components (CSC) part of the slice represents the main functionality of a slice as requested by the slice tenant. This functionality is identical to the implementation in the non-sliced network. Following the concept defined in [145], the per-plane slicing of the CSC functions is allowed.

Slice Manager

The Slice Manager (SM) is an implementation of the ISM paradigm. It provides slice-specific run-time management, and it exposes an interface for the purpose of slice management to the slice tenant (I-TEN). This tenant-oriented interface provides the functionalities for KPIs monitoring, users management, policy-based slice management, slice security monitoring, etc. If a tenant is not interested in slice management, the interface is connected to the OSS/BSS of the orchestration domain and the slice management is provided by the domain operator. SM is not Core Slice Components (CSC) agnostic, and therefore, it should be defined together with the CSC and included in the slice template.

The SM functionalities can be modified during run-time using the slice update mechanism. That allows for the dynamic deployment of the management functionalities. In contrary to the 3GPP approach, 6G-LEGO hence makes the slice management programmable. The mechanism can be used, for example, for the dynamic addition of security functions to a slice. As outlined in [146], the preferred approach to SM implementation is the automated/autonomic management paradigm since the management of many slice instances cannot be done efficiently in a manual way. Using an AI-driven approach, the I-TEN can be defined as an intent-based interface providing a relatively easy management interface to the slice tenant, which in most cases is not a professional network operator. One of the benefits of the ISM paradigm is the minimization of involvement of the global OSS/BSS in slice management, which contributes to management scalability and reliability improvement. It also minimizes the OSS/BSS modifications that are generally required in order to efficiently support the management of each type of slice.

Resource Orchestration Support

The Resource Orchestration Support (ROS) part of a slice is a component that supports resource orchestration. It cooperates with SM in order to proactively allocate resources to a slice (using slice usage predictions) before the resource congestion occurs and enables QoE-driven resource allocation. By analysing the slice traffic, ROS can also request the relocation of a specific virtual function or the deployment of a new one. It may play the role of a dedicated VNFM of MANO and OSS/BSS that interacts with NFVO for slice update. After the sub-network slice deployment, ROS has the knowledge about the slice topology, the location of virtual functions and the allocation of resources to the sub-network slice. The initial information is obtained by ROS from the orchestrator. ROS plays an important role during the sub-network initialization phase – it obtains the sub-network slice configuration details from the orchestrator and sends the data to SM, which configures all virtual functions of the sub-network slice. ROS must be included in the sub-network slice template.

The Slice Operation Support

The Slice Operation Support (SOS) part can be seen as a set of slice-level control plane entities supporting slice stitching, authentication and selection. The SOS entities are the following:

- The Border Gateway(s) (BG) supports the stitching of sub-network slices by providing proper adaptation and configuration of inter-slice protocols for the transport of the user and control planes. BGs can expose an abstracted slice view to other slices.
- The Slice Exposure Function (SEF) is used in order to describe the slice properties and configuration. In the case of a combination of several sub-network slices, SEF has to be appropriately updated. The role of SEF is similar to the Network Exposure Function (NEF) of 5G as defined by 3GPP [137].
- The Slice Authentication Function (SAF) is used for authentication of users and for mutual authentication of stitched sub-network slices. The Slice Users Database (SUD) stores the information about the users that are attached to a slice. SEF and SAF take part in the process of slice selection. In the case of horizontal stitching of several sub-network slices, only one of them implements the functionality.
- The Mobility Management Function (MMF) is used for handling mobility of users attached to the end-to-end slice as well as the “mobility” of slice entities, i.e. their relocation, to improve slice KPIs. The latter is realized via the ROS functionality. MMF also reflects the decisions made by the Multi-

Domain Orchestrator (MDO) regarding optimal slice placement by preparing the slice and its users for relocation.

The SOS functionality should be configured just before slice deployment according to slice- or tenant-specific needs. As a result of slice stitching, some functions of the SOS of some sub-network slices can be removed, e.g. in a chained sub-network slice, only one of them may have a SAF and SEF. More about the usage of SOS for slice stitching is presented in the next subsection.

Slice Chain Configurator

The Slice Chain Configurator (SCC) entity is used in case a slice is a member of any chain of sub-network slices. It keeps the identity of the chain and the identity and role of the sub-network slice of the chain.

Chain Manager

In each chain, there is a sub-network slice that has an entity called the Chain Manager (CM). It plays the manager role of the chain of sub-network slices by interacting with SMes of all sub-network slices that are members of the chain. Its role is explained in details further on.

4.4.1.1.2 Operations on 6G-LEGO sub-network slices

Sub-network slice stitching is at the heart of the 6G-LEGO concept. In this section, we provide some details on how the stitching of slices contributes to the flexibility of the presented framework and how the functions of SOS support slice stitching.

Horizontal stitching of sub-network slices

The creation of the end-to-end slice over multiple technological or administrative domains can be problematic. In the case of different administrative domains being under the control of different operators, the issue is typically the reluctance of domain operators to share the details regarding their infrastructure that is needed to deploy slices. In the case of different technological domains, the issue is the necessity to use domain-specific orchestrators. These problems can be solved by defining a set of single-domain templates, which can be implemented by each operator upon request and later on stitched. Another motivation for sub-networks stitching is the split of the same technology domain into smaller orchestration domains for the sake of scalability. All the mentioned reasons enforce the motivation for the creation of end-to-end slices as a combination of domain-level sub-network slices. We call this operation the horizontal stitching of sub-network slices. The operation is similar to the interconnection of multiple networking domains. In general, the use of horizontal slice stitching results in a less efficient implementation than the use of a single inter-domain orchestrator, yet it provides the above-mentioned advantages.

Vertical stitching of sub-network slices

There are multiple reasons to provide the so-called vertical stitching of slices, which lies on “piggybacking” of one slice to another already-deployed slice. There are two variants of vertical stitching:

- Recursive vertical stitching of slices that relies on the use of existing APIs (i.e. the Slice Exposure Function, SEF) of a slice can be used to enrich its functionality by the addition of a sub-network slice in the same domain. Later on, the functions of both combined sub-network slices can be used by another sub-network slice, and this operation can be recursive. The described approach is of premium importance for incrementally composing advanced services.
- The disjoint vertical stitching of slices relies on piggybacking of many mutually isolated slices to the same, shared subnetwork slice. This case can be used for multiple short-time lived slices in order to reduce their footprint (some functions are common to all slices of the same type, fewer functions have to be instantiated for each slice instance) and to shorten their deployment time. Moreover, such an approach can be enforced due to the limitation in the slicing of some technologies (legacy networks, RANs).

Yet another approach of the concept is the Management as a Service (MaaS), i.e. the creation of a slice that has SM functions but is able to manage several slices of the same type (SM is not slice type agnostic). Such a set of common functions sometimes referred to as a Common Slice (in opposite to Dedicated Slice [144], [147]) can also be used for the implementation of inter-slice operations. For example, functions like the authentication of users or slice selection can be common to all slices or to a group of slices. Moreover, a common implementation of mobility management is beneficial when users have the capability of simultaneous attachment to multiple slices. In the presented concept, we have followed the Common Slice idea, i.e. all common functions are grouped together that is justified by the ease of management and the elegance of the overall approach. It should be noted, however, that common functions have a negative impact on the isolation properties of slices.

An example of stitching of several sub-network slices is shown in Figure 63. In the presented case, the Border Gateways (BGs) of each SOS are used for the interconnection between pairs of Sub-Network Slices (SNS). BGs provide the abstractions and interconnection of the different technological slices.

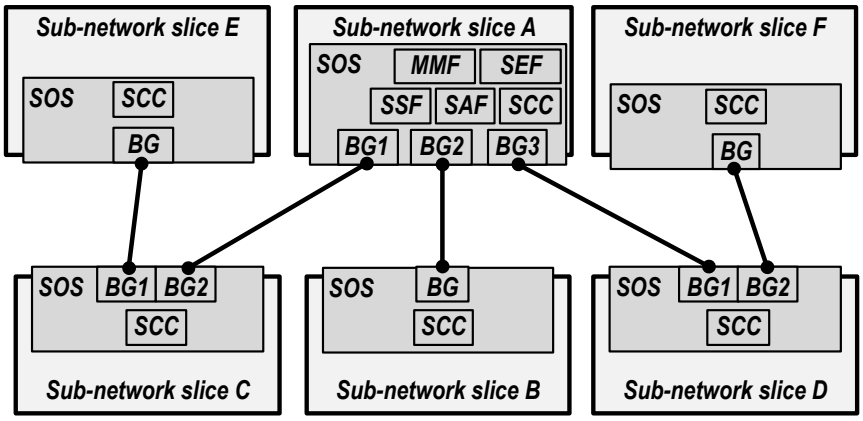


Figure 63: Stitching of several sub-network slices to compose an end-to-end chain (example).

The goal of the abstraction is to minimize the dependence on the technology, which has been used for the creation of each sub-network slice. The mentioned figure can be interpreted in several ways:

- if SNS B, C and D are SNSes of different Radio Access Technologies (RAT), e.g. WiFi, 4G, 5G, they provide in parallel connectivity to the 6G Core (SNS A) that has been configured during the deployment phase to authenticate the users and to also provide some other functions.
- SNSes C (RAN), A (Core), D (cloud) can be considered as horizontally-stitched SNSes, whereas SNSes A and B, E and C, and F and D can be respectively seen as vertically stitched.

It has to be noted that the framework allows different access technologies to be used for the same slice and that several 5G/6G core networks can be implemented using the proposed framework. This is a strong differentiator with the 5G network slicing, in which a single 5G Core (NSSF, AMF, etc.) has to support all other slices. Due to the proposed feature, the 6G-LEGO framework allows for RAN aggregation and RAN sharing.

In Figure 64, we illustrate the usage of a common sub-network slice. In the presented case, the slice is responsible for the management (via SM), orchestration support (via ROS) and authentication of users (via SOS) of all sub-network slices and its users that are vertically attached to the slice. In this way, the footprint of the dynamically deployed sub-network slices is smaller, and their deployment is faster.

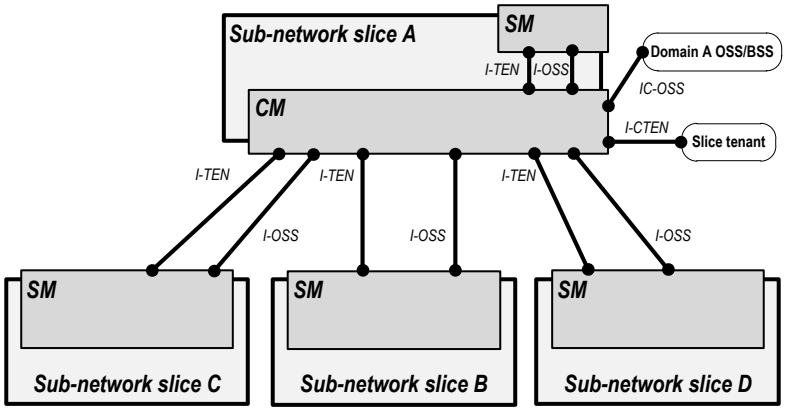


Figure 64: An example of usage of the common sub-network slice.

In chaining sub-network slices, one of the chained subnetwork slices has to play the role of the Chain Manager (CM). CM keeps the information about the chained SNSes and defines their roles. This operation is supported by the SCC component of SOS.

A *conditio sine qua non* of the described concept is the creation of sub-network slice templates with the customized SOS part of each of them before the sub-network slice deployment.

In the proposed solution, the slices and their functions are mutually isolated. For the initial attachment of a user to a slice, it is necessary to provide a mechanism that informs the users about the deployed slices and their properties. For that purpose, we propose the creation of an end-to-end slice (a chain of sub-network slices) that interacts with MDO in order to gather information about the deployed slices and their features. Such a slice can be seen as a "default slice" and is used for the initial interaction with the 6G-LEGO framework.

4.4.1.1.3 Management of the end-to-end slices

In a classical ETSI NFV MANO case, also applied to 5G network slicing, the management of all deployed Network Services (i.e. network slices) is performed by an OSS/BSS, external to the slices that interact with the NFVO and Element Managers (EMs) of VNFs of slices. As it has been described, the 6G-LEGO framework uses embedded sub-network slice management and the Slice Manager (SM) that is a part of a sub-network slice. SM exposes two management interfaces – the main management interface to slice tenant (I-TEN) and another one to the domain level OSS/BSS (I-OSS), mostly for SLA and accounting purposes.

In the case of chaining of several sub-network slices, there is a need, however, to provide the overall management of the chain. According to the 6G-LEGO philosophy, the interactions between sub-network slices should be minimized and as agnostic as possible. It is worth noting that in current networks, there exist multiple, domain-level management systems, but the interactions between them are non-existent or minimal (typically for KPIs exchange). In our proposal, the Chain Manager (CM) plays the role of a "higher-level" Slice Manager (inter-domain) that is managing the end-to-end slice via the interaction with all SMes of the sub-network slices that compose the chain.

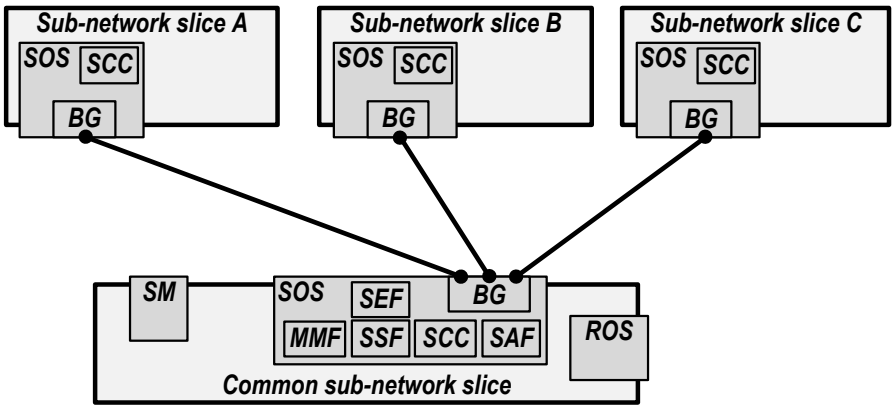


Figure 65: Management of several sub-network slices that compose an end-to-end chain (example).

CM is also involved in the end-to-end KPI calculations, the policy-based management of a slice, etc. For predefined chains, CM can be defined a priori by the template provider; however, the best solution would be a dynamic creation of the component. Unfortunately, the automatic creation of a CM for any chain of sub-network slices is a challenging task. CM should be deployed in the same sub-network slice, in which the Chain Master Controller is implemented. Typical interactions between SMs and CM are shown in Figure 65. It has to be emphasized that during the slice run-time, the most involved entities in the slice management are the Slice Managers (SMs), and the involvement of CM is limited; therefore, even manual management in such a case is applicable. CM interacts with OSS/BSS that manages the sub-network slice, which has an embedded CM for the purpose of reporting of the chain (end-to-end slice) KPIs and accounting.

4.4.1.1.4 Orchestration of 6G-LEGO network slices

In this section, the 6G-LEGO orchestration is outlined. The main idea of 6G-LEGO lies in the use of domain-level slices or sub-network slices that are created by the domain level orchestrators in order to obtain end-to end-slices. The proposed approach is not tied to a specific orchestration technology. However, we will refer to the ETSI NFV MANO framework as a reference solution. The virtualized infrastructure is a classical infrastructure of virtualized resources (computing, storage and connectivity) of an administrative domain that can be allocated to multiple slices. In some cases, the infrastructure can include hardware entities (Physical Network Functions according to the ETSI NFV naming).

In 6G-LEGO, there is a functional split between the slice-dependent management and the resource-oriented orchestration (slice agnostic). The management part of the sub-network (single domain slice), as well as resource orchestration support, are both a part of the sub-network slice template (ROS component), which is in line with the 6G-LEGO philosophy. The overall orchestration architecture is presented in Figure 66.

For the lifecycle orchestration of slices, the generic MANO approach is still applicable with minor changes. First of all, the orchestration of multiple domains is made in a hierarchical way using the Multi-Domain Orchestrator (MDO) on top of Domain Orchestrators (DOs) that are involved in the lifecycle and run-time orchestration of network slices. In the architecture, we also introduce the Tenants Portal (TP) and the Sub-network Slice Configurator (SSC), which roles will be described in details further on. First, we describe the usage of the framework for the life cycle management of slices and later their role in the run-time orchestration of network slices.

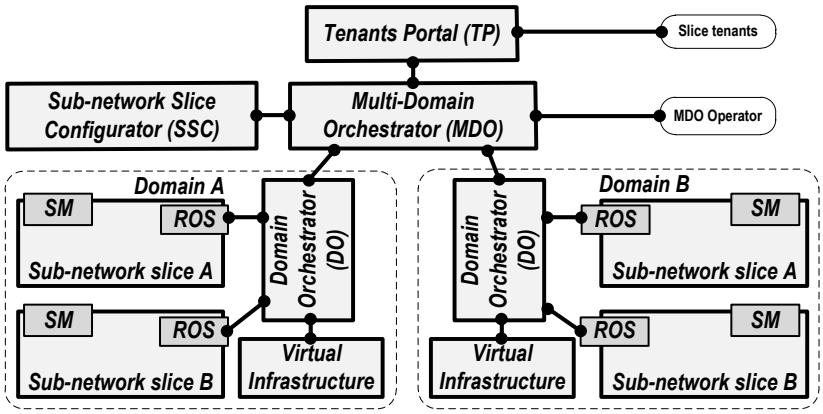


Figure 66: Orchestration architecture of 6G-LEGO (only orchestration-related interfaces are shown).

Slice lifecycle orchestration

The orchestration component roles in the slice lifecycle orchestration are as follows:

- **Tenants Portal**. The portal is a single point of contact for tenants and is mainly used for the purpose of slice template selection and slice life cycle management (deployment, update, termination). For the slice selection procedure, the Tenants Portal has a list of all slice templates (sub-network templates or templates of chains) that can be deployed. Each sub-network slice or a chain has its descriptors and a list of configurable parameters that are available to the Tenants Portal. Such a list is provided to the Tenants Portal by MDO. During the slice request dialogue, some slice deployment options can be negotiated. The Tenants Portal can also be used for the slice KPIs monitoring by slice tenants if it is not directly provided by the embedded slice mechanisms, i.e. SM. For the purpose of KPI monitoring, the Tenants Portal has a database that stores the KPIs of each slice. The KPIs are defined in the slice template and are calculated by the management components of a slice (SM). The Tenants Portal may provide visualization of slice KPIs. A proposal for such slice-agnostic KPIs can be found in [148]. The portal also keeps all the accounting data concerning slice tenants. For that purpose, the requested and obtained slice KPIs are compared for SLA validation. On the one hand, the Tenants Portal interacts with tenants, and on the other hand, it interacts with the Multi-Domain Orchestrator (MDO) and indirectly with the Sub-Network Slice Configurator (SSC).
- **Multi-Domain Orchestrator (MDO).** MDO performs multiple roles. This is the entity that is contacted for the deployment of all slices; it also provides the Tenants Portal with information about KPIs of all already-deployed slices. It includes the database of all templates that can be used for slice deployment within the framework (even those, which are used by other orchestrators of the framework). When a new slice request is sent to the Multi-Domain Orchestrator, it analyses the feasibility and options of its deployment. All the operations can be qualified as a slice pre-deployment phase.
- **Sub-network Slice Configurator (SSC).** The role of SSC is to modify the sub-network slice template optionally according to the requirements of the chained sub-network slices. The SOS modifications correspond to the addition or removal from each sub-network SOS components that are, respectively, necessary or not needed for a specific sub-network slice chain. ROS of each sub-network slice obtains SSC information about the initial configuration of each sub-network slice that has to be deployed during the sub-network slice initialization. The information about chain-specific configuration for known templates is stored in SSC.
- **Domain Orchestrators (DO).** DOs obtain the sub-network slice templates modified by SSC (the modification typically concerns the SOS part of the slice template) from MDO and are responsible for the deployment of each sub-network slice that compose an end-to-end slice. DOs are domain-specific orchestrators of resources combined with the domain OSS/BSS. The domain OSS/BSS part of a DO is focused on all the operations of its domain. It collects statistics related to resource usage, provides the inventory of running slices and their dependencies that are the implication of vertical

and horizontal stitching of sub-network slices, which are deployed within this domain. The OSS/BSS part provides FCAPS functionalities for its domain, i.e. it may impact the domain orchestrator to implement a template in a specific way. It also provides an interface to the domain operator and to DMO. The orchestration part of DO is responsible for requesting from the Infrastructure resource allocation to a slice (or sub-network slice) and the optimal placement of template virtual functions. However, the initial configuration of a deployed slice is performed by SM of the slice. Therefore, the domain orchestrators (DOs) are slice agnostic. DO may have to collect all the information related to deployed slice performance and faults. After the deployment of each sub-network slice, its ROSes forward to respective SMes the slice configuration parameters that SMes have to use for configuring each component of the sub-network slice. During the deployment, each DO provides ROS with information about slice deployment configuration that includes the information about the deployment of each virtual function, its placement, resource allocation as well as the sub-network slice graph.

After the successful completion of the mentioned operations, the sub-network slice SM or, in the case of a chain, CM reports to the Tenants Portal that the slice is ready to be used, completing that way the slice deployment phase.

The slice termination phase details are omitted in the description as the operations of this phase are pretty simple. The only aspect specific to the 6G-LEGO framework operations is related to the transmission of all slice-related statistics via DMO to the Tenants Portal for the purpose of accounting. It is noteworthy that the slice termination will also terminate SM of the terminated slice.

Slice run-time orchestration

During the run-time phase, each SM of the sub-network slice carries out the performance analysis and, on that basis, it is trying to perform reconfigurations in order to maintain the slice KPIs at the required level. SM, in cooperation with ROS (which is aware of the sub-network slice graph, resource allocation and consumption), may propose to change the resource allocation or to relocate a certain virtual function from one data centre to another one for the sake of data transport efficiency. All slice template modifications or virtual function placement requests are sent via ROS to DO. After the change of the virtual function placement, SM makes necessary configuration changes, and ROS updates its slice deployment-related database. The mentioned SM-ROS cooperation allows for the application-driven (sometimes QoE-based) resource allocation that is hard to achieve in the present MANO architecture, and it provides an efficient extension to the classical management that assumes a fixed allocation of resources.

4.4.2 6G-LEGO framework implementation remarks

In the previous section, we have described the 6G-LEGO framework. In this section, we provide an analysis of its potential implementation. The proposed approach differs significantly from the 3GPP's 5GC-centric one. We have proposed a different decomposition of functions with a much higher distribution level. Due to the embedding of many functions in the slice template, these functions do not need to be implemented as a part of the framework. Some of the 5GC functions can be reused by 6G-LEGO, even as a part of the slice template. Nonetheless, particular functions that do not exist in 5GS have to be developed and implemented from scratch. The set of these functions include the Tenants Portal, SSC and MDO.

There is no need to specify the slice internal interfaces as they are slice-specific. There is, however, a need to specify all the interfaces that are used for the interaction of a slice with external entities.

The interfaces that need to be specified within a slice template include:

- SM-DO interface. This interface is mostly used for providing information about slice health and performance (KPIs).
- SM-Tenant interface. This interface is a customized web interface. Therefore, its specification is not a critical one.

- ROS-DO interface. ROS provides interfaces to DO that deals with orchestration. ROS includes the VNFM functionality of MANO and interacts with NFVO using OSS/BSS-NFVO-like interfaces. Details of the modification of the interfaces have yet to be specified.
- BG interfaces. The SOS functions interact with the entities external to the slice. BG has to provide an interface for data exchange in a slice chain. Typically, some standardized IP protocols will be used for that purpose.

The SEF may use the modified and extended concepts of 3GPP NEF. The SAF functions can be taken from 3GPP, and it plays a similar role to the NSSF of 5GC. The slice Chain Managers are slice specific and do not have to be specified.

4.4.3 6G-LEGO framework components and their interfaces

Tenants Portal

The Tenants Portal (TP) provides the interaction between tenants and MDO. It plays the role of the business portal used for triggering the operations related to the lifecycle management of the slices. The interface between the portal and MDO has to be defined but not standardized. It is used for passing to the TP the information about the possibility of slice deployment, and according to Tenants Portal requests, MDO provides the lifecycle management of multi-domain slices. The GST/NEST templates proposed by GSMA (with extensions) can be used for negotiations with tenants.

Sub-Network Slice Configurator

The Sub-Network Slice Configurator (SSC) converts the requests obtained from MDO, and it thus compiles sub-network slice templates with customized components of the SOS. The SSC-MDO interface should be standardized.

Multi-Domain Orchestrator

MDO interacts with SSC, DOs, TP and SMes of slices. Its main function is the deployment, termination and high-level monitoring of the status of the deployed slices. This is not a MANO orchestrator. The interaction of MDO with SSC and TP has been already described. The MDO-DO interface is used for the deployment of a sub-network slice template in a specific domain. This interface should provide the lifecycle operations abstractions to deal with different types of DOs.

Domain Orchestrator

The Domain Orchestrator (DO) is composed of two parts. The first one is the domain-level OSS/BSS that is master for all orchestration operations. It performs all FCAPS-oriented operations. The second part of DO is the resource orchestrator (MANO-like). The SM-DO interfaces used for triggering orchestration operations by SM can follow the NFV specifications. For specific technological domains, different than MANO orchestrators should be used. MDO should be aware of their specificity and capabilities.

Virtual Infrastructure

The Virtual Infrastructure is not specific to the 6G-LEGO framework. The ETSI MANO approach or other domain-specific approaches (e.g. for RAN) can be used.

4.4.4 Concluding remarks

The 6G-LEGO model minimizes the external operations related to each slice and uses some programmable abstractions to simplify slice stitching and compose the end-to-end slices as a set of “bricks”. To achieve that goal, we looked at a cloud model considering a slice as a distributed application, which internals are not known to the cloud (infrastructure) or the orchestrator operator. Thus, we have proposed to include a programmable management plane of each slice in the slice template (e.g. SM). Moreover, the slice template can be customized before its deployment by adding necessary functions, such as support for slice selection and authentication. Using its orchestration-

related function (ROS), the slice itself is able to provide proactive resource allocation and slice update (adding or removing slice functions). Additionally, we have proposed the horizontal and vertical concatenation of sub-network slices. Such operations are possible due to the programmable interfaces of slices and the in-slice management concept that we introduced. The vertical and horizontal slice stitching allows for deploying multi-domain slices or adding virtualized service platforms to slices. 6G-LEGO is agnostic to radio access technologies (RATs). In fact, our approach enables the deployment of multiple, independent instances of a complete mobile network. The functionality of the orchestrator in 6G-LEGO has been simplified – it is slice agnostic, mostly used in the slice deployment phase, and it is focused on resources only. As compared to 5GS, the number of interfaces of 6G-LEGO that have to be standardized is reduced since there is no need to specify intra- and inter-slice interfaces.

The presented concept still remains at a high level and thus requires significant work before it can be materialized. We list hereafter the most critical issues that have to be solved before obtaining a deployable 6G-LEGO framework. More work on the MaaS approach, i.e. common management of several or all slices based on the same template, is needed. MaaS (as an option) can efficiently address the drawback of the proposed solution, i.e. the increased slice footprint caused by adding the management plane to the slice. The slice template creation and preparation entity also plays a significant role in 6G-LEGO. Such a complex and intelligent entity has to be developed from scratch. The 6G-LEGO orchestrator can be seen as a redefined MANO orchestrator with reduced functionality. This redefinition has yet to be specified in details, and the exiting solution appropriately modified. The SOS entities responsible for slice selection, authentication and mobility have to use standardized interfaces to interact with the end-user's equipment. MDO requires a more detailed specification of its internal functions. The slice stitching operation needs further investigation regarding slice exposure and handling of the dependencies between vertically stitched slices. An interesting challenge is the automatic generation of the Chain Management components that can provide end-to-end slice management through the interaction with Slice Managers. All these issues are open and left for future work. In this regard, the paper is providing some research directions for the efficient implementation of slices in 6G networks.

4.5 Network slicing implementation using network slice templates

4.5.1 Description of the solution under test

The slicing mechanism has been integrated into the joint project testbed, and some initial measurements have been collected. The measurements were performed using VNFs from both WP3 and WP5, using the Katana Slice Manager. It is a novel network slicing solution [149] aimed towards 5G deployments, with the support of end-to-end slicing for multiple layers of the infrastructure. The design and implementation of Katana are based on the 3GPP Technical Report TR 28.801 "Telecommunication management; Study on management and orchestration of network slicing for next-generation network" [150]. Following the concepts of the 3GPP specification, the following definitions have been implemented:

- Network Slice Instance (NSI): NSI includes all functionalities and resources necessary to support a certain set of end-to-end communication services.
- Network Slice Template (NST): NST describes the Network Slice to be created.
- Components of an NSI: The NSI is comprised of Virtual or Physical Network Functions (NFs). These NFs can be dedicated to NSI or shared among multiple NSIs. If NFs are interconnected, Slice Manager contains information relevant to connections between these NFs, such as topology of connections, network graph, link requirements, etc.
- Network Slice Subnet Instance (NSSI): NSI may be composed of NSSIs. The NSSIs may represent different domains of the physical infrastructure, such as NFVI, Transport Network, RAN, etc. NSSI may include other NSSI(s). For example, a 5G Core network can constitute NSSI.

The Slice Manager is a centralized software component that manages all management and orchestration entities of the assigned infrastructure. In addition, it provides an interface for creating, modifying, monitoring, and deleting slices. Through the North Bound Interface (NBI), the Slice Manager interacts with a coordination layer or directly with the network operator. It receives the Network Slice Template (NST), which describes the particular slice details corresponding to the service requirements. NST includes details such as the list of NFV components (Network Services) that need to be instantiated, WAN configuration, QoS, monitoring level, Life-Cycle stages, etc. NST follows the GSMA Generic Slice Template [152].

Slice Manager maps these details to specific actions for provisioning network slices. It also provides an API for managing and monitoring all slice instances. Through the South-Bound Interface (SBI), it interfaces to the components of the Management and Orchestration Layer (MANO), namely the NFV Orchestrator (NFVO), the Element Management System (EMS) and the WAN Infrastructure Manager (WIM), in order to control every device on the user plane. The sequence of a slice creation is depicted in detail in Figure 67.

An example of the workflow for the creation of a network slice is the following:

- Experimenter requests through the portal the creation of a new slice using Slice Manager's NBI, selecting a particular slice profile, in order to deploy his Communication Service (i.e. comprising of a number of NS plus, not mandatory, a 5G NC system).
- NST is created and parsed by the Slice Manager.
- Slice Manager runs the Placement process to determine where to instantiate each Network Service and creates the Network Graph for the slice.
- Following the placement decisions, the Slice Manager communicates with VIM, WIM and EMS in order to provide resources (Sub-Network Slices).
- VIM creates a new tenant for the slice.
- WIM creates virtual links or/and flows on SDN switches with specific resource-QoS requirements, as declared in NST, in order to activate appropriate traffic steering.
- EMS provides the required resources and configurations (e.g. associate traffic or user ids to APNs, spectrum frequency allocation, bandwidth, etc.).
- Slice Manager communicates with the NFVO in order to make the deployment and instantiation of the Network Services included in the Communication Service.
- Slice Manager returns a slice id to the operator for further management and monitoring purposes.

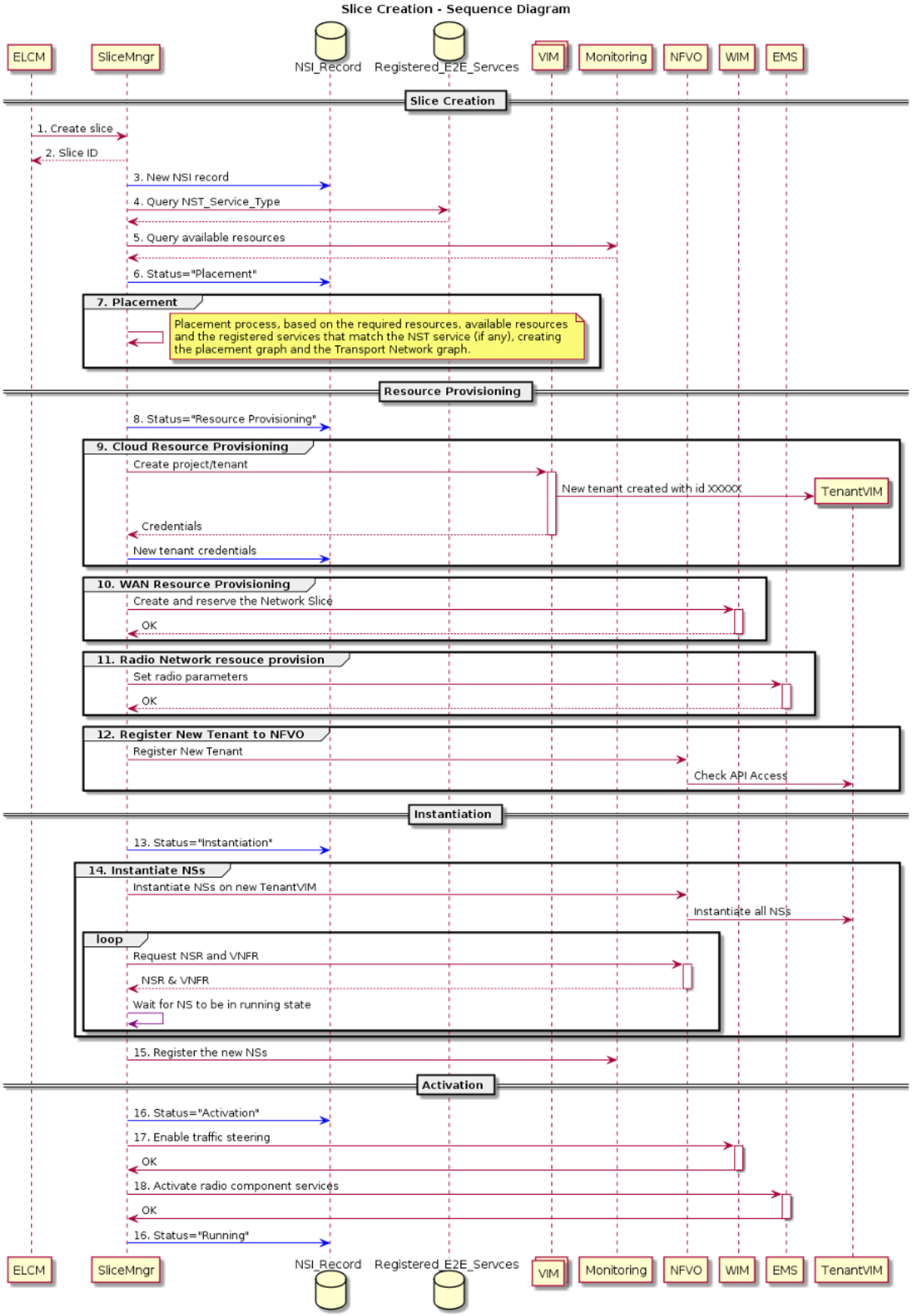


Figure 67: Sequence diagram of slice creation.

Based on the onboarded NST, Slice Manager has to make the mapping between the available user plane resources and the described slice requirements. At this stage, Slice Manager calculates and decides on the placement of each Network Service and its components, based on numerous

parameters, such as slice requirements, available resources, geographic location, duration of the slice, network policies, etc.

The proposed Slice Manager architecture is illustrated in Figure 3, and its implementation is open source. Slice Manager is based on a highly modular microservices architecture. Each microservice is running on a Docker container. The key advantages of this architectural approach are i) simplicity in building and maintaining applications, ii) flexibility, and iii) scalability, while the containerized approach makes the applications independent of the underlying system.

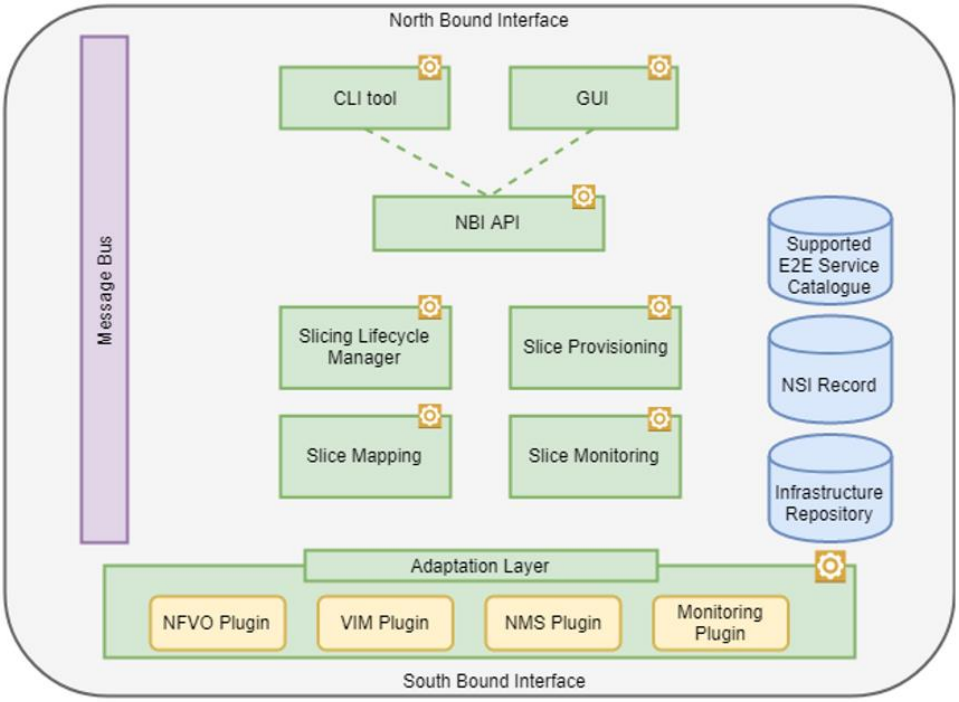


Figure 68: Slice Manager Architecture.

As seen in Figure 68, Slice Manager is split into multiple components-specific functions. Below we briefly describe the core components:

- **North Bound Interface API.** The North Bound Interface API module implements RESTful APIs that can be consumed by a component of upper layers, e.g. the Experiment Lifecycle Manager, a user/experimenter or the Slice Manager administrator. This module provides intelligence of the manager, implementing the Create, Read, Update and Delete functions. The role of this component is twofold. On the one hand, it receives requests from the NBI API module and takes any necessary actions for the activation, modification or deactivation of a network slice. On the other hand, it receives messages from the Slice Monitoring module regarding the status change of a deployed slice. It interacts with the other components of the Slice Manager in order to trigger the process that needs to start, depending on the received messages.
- **Slice Mapping.** This module hosts a very important process that runs during the slice creation phase, the placement process. This process is responsible for optimally selecting the infrastructure resources to be used for a new slice, based on the slice requirements, as they are described in the NST and the available resources of the infrastructure layer.
- **Slice Provisioning.** The Slice Provisioning module receives requests from the Slicing Lifecycle Manager service in order to set up, configure or delete the Wide Area Network paths, the isolated NFVI tenant spaces and all the required Network Services, to configure the radio component parameters and register the newly created slice to the Monitoring system. It does so by using the Virtual Infrastructure Manager (VIM), NFV Orchestrator (NFVO), Network Management System (NMS) and Monitoring Plugins of the Adaptation Layer.

- **Slice Monitoring.** The Slice Monitoring module is responsible for monitoring the health and the status of every deployed slice. It uses the Adaptation Layer plugins to send status check messages to the MANO components below the Slice Manager and reports any slice status change to the Slicing Lifecycle Manager service.
- **Adaptation Layer.** The Adaptation Layer module provides a level of abstraction regarding the underlying domain technology, making it feasible for the Slice Manager to operate over any MANO layer component without any modifications to its core functionality, as long as the proper plugin has been loaded. This module is comprised of VIM, NFVO, NMS and Monitoring plugins, one for each of the MANO layer components supported by the Slice Manager. The responsibility for each plugin is to receive request messages from the Slicing Lifecycle Manager and translate them to the proper API call of the supported component. After that, it is in charge of properly handling the responses from the underlying components to the API calls.

4.5.2 5G slicing latency and energy consumption evaluation

In this section, a set of experimental tests conducted for an assessment of the differentiation in latency performance exhibited by certain slice configurations has been presented. Additionally, a set of tests were also carried out in order to evaluate the energy efficiency of the selected slicing scheme and show the trade-off between latency with slicing and energy consumption. The assessment has been performed on the 5G network infrastructure based on Amarisoft 5GC and NR implementations configured in the Standalone Mode (Option 2). The slice manager was appropriately extended in order to communicate and modify various parameters at the RAN and Core configuration of the 5G system and enforce certain slicing policies through a dedicated custom build EMS. The slicing experiments were performed using the cache and the vDPI VNFs, developed in WPs 3 and 5. As both virtualized services were similar in terms of slicing in the deployment domain, the tests were focused on the RAN slicing domain and what performance different types of slicing can achieve, mainly in the latency domain.

The slicing is enforced and measured on the platform in two steps. The first step includes the manipulation of the RAN layer numerology (i.e. configuration of waveform parameters), in order to minimize the transmission windows, by fine-tuning the Scheduling Request Period (srPeriod) and slot Period parameters of the gNB. The approach aims to minimize the round-trip time (RTT) in the RAN domain by modifying the Time Division Duplex (TDD) slot to Downlink-Uplink (DL-UL) pattern and reduce the scheduling request period. Furthermore, the srPeriod is a special physical layer message for UE to request UL Grant in order to transmit in Physical Uplink Shared Channel (PUSCH). Regarding slot Period, the main idea is based on TDD, where the UL-DL channels transmit simultaneously, and by tailoring the time periods of the lots, the performance can be tailored to our needs.

The second step involves slicing enforcement at the 5G Core domain by allocating different PLMNs or APNs to different services per group in order to differentiate traffic and be able to enforce the necessary policies.

There is currently no support at RAN for multi-slicing, although 3GPP standards provide the relevant provisions at the RAN level using different types of slices (i.e. URLLC/eMBB/mMTC) [153]. Since there are no devices that can utilize more than one slice at the same time (limitation due to APN dependency and SIM card structure), parallel slicing can be achieved via allocating an eMBB slice with specific bandwidth at the backhaul and Core network and perform the URLLC slice enforcement at the RAN level by modifying the srPeriod and slot Period parameters at the gNB. This paradigm of concurrent slicing is critical to the operation and robustness of emergency communications for First Responders, as it maps different service requirements to slice instances of variant bandwidth and latency.

Most of the currently available slicing mechanisms depend heavily on NFV/SDN capabilities plus network virtualization and QoS/traffic prioritization. Katana, for the needs of the proposed platform, enforces policies both on the Core and the RAN domain of the infrastructure. Based on this implementation, set experimental tests were performed, which measured the latency for various

packets sizes of a 5G system after having allocated an eMBB slice and then enforcing different URLLC slices over it.

The first set of results are shown in Figure 69, where an eMBB slice of 100 Mbit/s was allocated at the backhaul, and a set of URLLC slices were enforced and measured in terms of latency, for srPeriod values of {1,10,40} and slot Period values of {2.5,5}. As it can be deducted, the RAN modifications can greatly affect the latency of the system for all packet sizes, and in some cases, an improvement of up to 57% in the case of srPeriod = 40, slot Period = 5 and srPeriod = 1, slot Period = 2.5 for a packet size of 128 bytes.

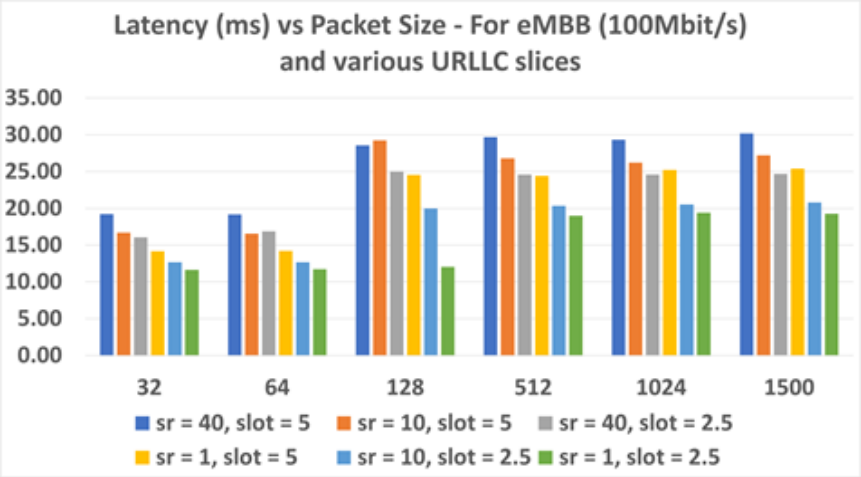


Figure 69: Latency results for a 100 (Mbit/s) eMBB slice and 6 different URLLC slices.

Furthermore, in the second set of experimental tests, an eMBB slice of 200 Mbit/s was allocated, and the same set of RAN URLLC slice parameters were enforced. The results, as depicted in Figure 70, indicate a similar behaviour to the eMBB slice results of 100 Mbit/s, with the case of 128-byte packets to demonstrate the largest improvement again.

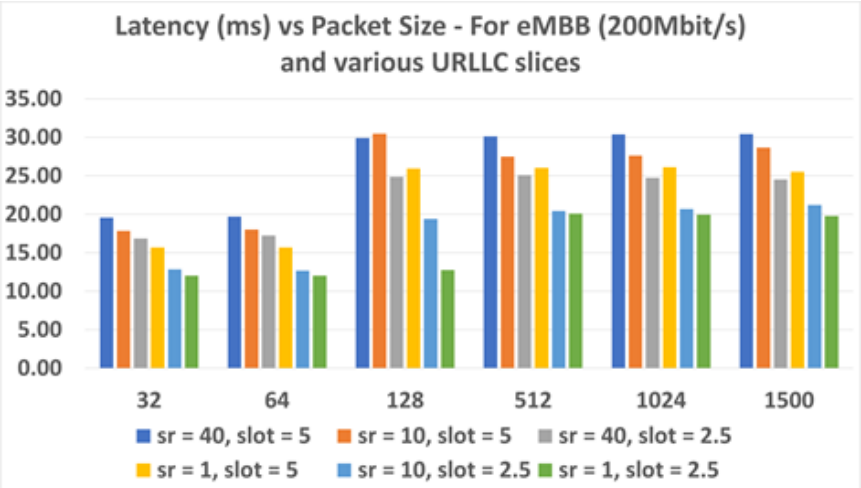


Figure 70: Latency results for a 200 (Mbit/s) eMBB slice and 6 different URLLC slices.

It is evident from both set of results that the case of the URLLC slice of srPeriod = 1 and slot Period = 2.5 display the best performance in terms of latency for a URLLC slice, with an average minimum latency of ~12 ms. These values can be indicated as the best achievable latency performance measured in the proposed 5G platform for end-to-end measurements.

In the next set of experiments, the average energy consumption for various slices was measured. This set depicts the overall energy efficiency of 5G slicing and how latency and various bitrates affect the overall energy consumption of UE. In Figure 71, a set of experimental results is presented in terms of energy consumption (mA) regarding different packet sizes and the minimum and maximum latency

configurations, as they are deducted from the previous experiments, i.e. minimum (sr = 1, slot = 2.5) and maximum (sr = 40, slot = 5).

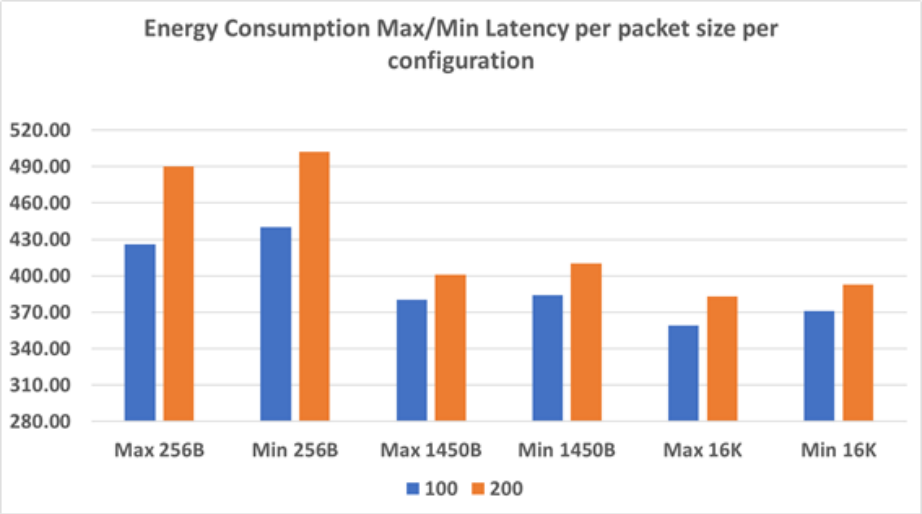


Figure 71: Energy consumption results for various packet sizes for the minimum and maximum latency eMBB slices.

This set of experiments was also performed over the different eMBB slices (100, 200 Mbit/s) in order to fully investigate the effect of different packet sizes on different data paths. As it can be deducted, the main factor contributing to higher energy consumption is the small packet size, which is expected, as the device performs significantly more transmissions to attain the same level of bitrate to attain the slice bitrate requirement. Furthermore, the most important conclusion of these experiments is that lower latency requires more energy, as can be clearly seen from Figure 6. The configuration for minimum latency (sr = 1, slot = 2.5) consumes more energy in all different packet size cases. Therefore, the trade-off between energy consumption and latency exists in the 5G domain. However, for the investigated eMBB and URLLC slices, the additional energy consumption ranges between 1.5% to 3% in order to achieve ~10 ms lower latency. Consequently, it should be noted that 5G low latency communications are energy efficient, as with minimal additional energy consumption, they can significantly reduce communication latency, a critical factor for emergency communications. Another minor conclusion drawn is that higher bitrate also consumes more energy in all cases, which is a logical induction as the channel transmits more data, thus more energy.

4.6 Deep Packet Inspection VNF implementation

Deep Packet Inspection[3] (DPI) is the practice of filtering and examining IP packets across Layers 2 through 7. Although Stateful Packet Inspection (SPI, often employed by firewalls) is more restricted, DPI may extend to headers and protocol structure, thus allowing for the implementation of advanced cybersecurity or load-balancing measures. DPI can be an effective detection tool for a multitude of cyberattacks such as Denial of Service (DoS) or provide insight on network traffic statistics for optimized rate control. In the context of WP5, DPI is utilized to monitor traffic per the protocol and does not inspect any communication payloads. DPI VNF (vDPI) is constructed for this purpose. The vDPI

[3] Deep Packet Inspection (DPI) is a type of data processing that inspects in detail the data being sent over a computer network, and usually takes action by blocking, re-routing, or logging it accordingly. While deep packet inspection can be used for innocuous reasons such as making sure that data is in the correct format or checking for malicious code, it can also be used for more nefarious motives such as eavesdropping and censorship. There are multiple headers for IP packets; network equipment only needs to use the first of these (the IP header) for normal operation, but use of the second header (such as TCP or UDP) is normally considered to be shallow packet inspection (usually called as Stateful Packet Inspection) despite this definition.

service used in WP5 is composed of one VNF, with multiple VNF components connected with virtual links.

4.6.1 vDPI architecture

The vDPI VNF comprises several VNF components (VNFCs), as illustrated in Figure 72:

- **VNFC-1 (Forwarding and Classification):** This VNFC handles routing and packet forwarding. It accepts incoming network traffic and consults the flow table for classification information for each incoming flow. Traffic is forwarded by using default policies until it is properly classified and alternate policies are enforced. It is often unnecessary to mirror packet flow in its entirety in order to achieve proper identification. Since a smaller number of packets may be utilized, the expected response delay can, therefore, be close to negligible, especially since the vDPI does not require to be deployed on the path of traffic. In a case where the Inspection, Forwarding and Classification VNFCs are not deployed on the same compute node, traffic mirroring may introduce additional overhead. A classified packet can be redirected, marked/tagged, blocked, rate-limited, and reported to a reporting agent or monitoring/logging system within the network.
- **VNFC-2 (Inspection):** The traffic inspection VNFC implements the filtering and packet matching algorithms and is the necessary basis to support additional forwarding and classification capabilities. It is a key component for the successful implementation of the vDPI and the most computationally intensive. The component includes a flow table and an inspection engine. The flow table utilizes hashing algorithms for fast indexing of flows, while the inspection engine serves as the basis for traffic classification.
- **VNFC-3 (Internal Metrics Repository):** The internal metrics repository acts as local storage. It can export vDPI metrics to a *Grafana* interface (Figure 73).
- **VNFC-4 (Node Exporter):** This VNF component gathers resource utilization data from the VNF and streams it to a *Prometheus* server for further monitoring.

The vDPI lifecycle is managed by the NFVO and can receive configurations. The orchestrator is in charge of starting, stopping, pausing, scaling and configuring vDPI. Thus, the Forwarding and Classification component acts as a managing/controlling VNFC and is assigned a floating IP for management. Internal communication is implemented via virtual links. Policies are relayed from the orchestrator and translated within the managing VNFC.

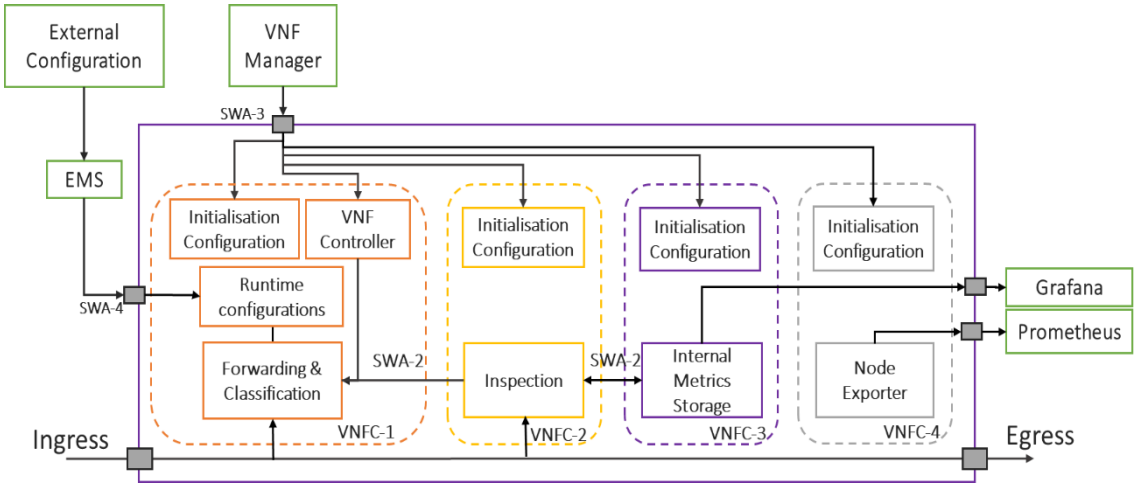


Figure 72: Architecture of vDPI VNF.

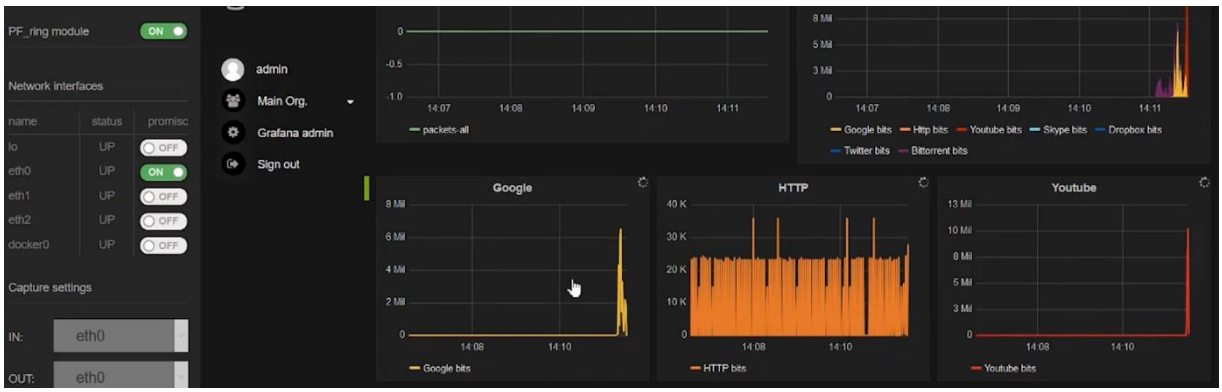


Figure 73: vDPI Grafana-based user interface showing monitoring statistics per the protocol and per domain.

4.6.2 Implementation and Specifications

The implementation of the vDPI components is based on a variety of technologies allowing performing traffic inspection as well as packet capturing. The following technologies are currently envisioned to be used in the implementation of this vNSF:

- **nDPI** [154]: is an open-source alternative to the OpenDPI [155] library, maintained by NTOP. Its goal is to extend the original library and add new protocols that are otherwise available only on the paid version of OpenDPI. Furthermore, nDPI is modified to be more suitable for traffic monitoring applications by optimizing the DPI engine. One of its major advantages is that nDPI can support application-layer detection of protocols, regardless of the port being used.
- **PF_RING** [156]: is a set of library drivers and kernel modules, which enable high-throughput packet capture and sampling. The PF_RING kernel module library polls packets through the *Linux* NAPI. Packets are copied from the kernel to the PF_RING buffer for analysis with the nDPI library.
- **DPDK (Data Plane Development Kit)** [157]: comprises a set of libraries that support efficient implementations of network functions through access to the system's network interface card (NIC). DPDK offers to network function developers a set of tools to build high-speed user plane applications. DPDK operates in polling mode for packet processing instead of the default interrupt mode. The polling mode operation adopts the busy-wait technique, continuously checking for state changes in the network interface and libraries for packet manipulation across different cores.

The PF_RING implementation selected for the vDPI has the capacity of maintaining uninterrupted connectivity with the OpenStack network. DPDK has the capacity to bypass the Linux kernel, leading to high-performance packet capture but is less robust and fault-tolerant than PF_RING. The following table gathers the technical specifications for the vDPI VNF (cf. Table 4).

General information	**VNF name**	Virtual Deep Packet Inspection (vDPI)
	Prepared by	ORION
	Release No. & Date	V0.4 – M22 (development freezes)
	The due date for the final version	M25 (final integration)
UC2 Requirements	1. **Deployment**: The VNF is on-boarded, enabled and instantiated by OSM. 2. **Lifecycle Management**: The VNF receives lifecycle management messages through the SWA-3 interface from the VNF Manager. 3. **Functionality**: The VNF parses network traffic and provides metrics. 4. **Display**: The VNF exports network monitoring data to be visualized in a Grafana UI. 5. **Resource monitoring**: The VNF exports resource utilization data.	
Base Image	The base image is based on Ubuntu 16.04 and the KVM hypervisor.	
Inputs	Ingress traffic, Lifecycle management messages, External configurations	
Outputs	Egress traffic, Network monitoring data, Resource monitoring data	

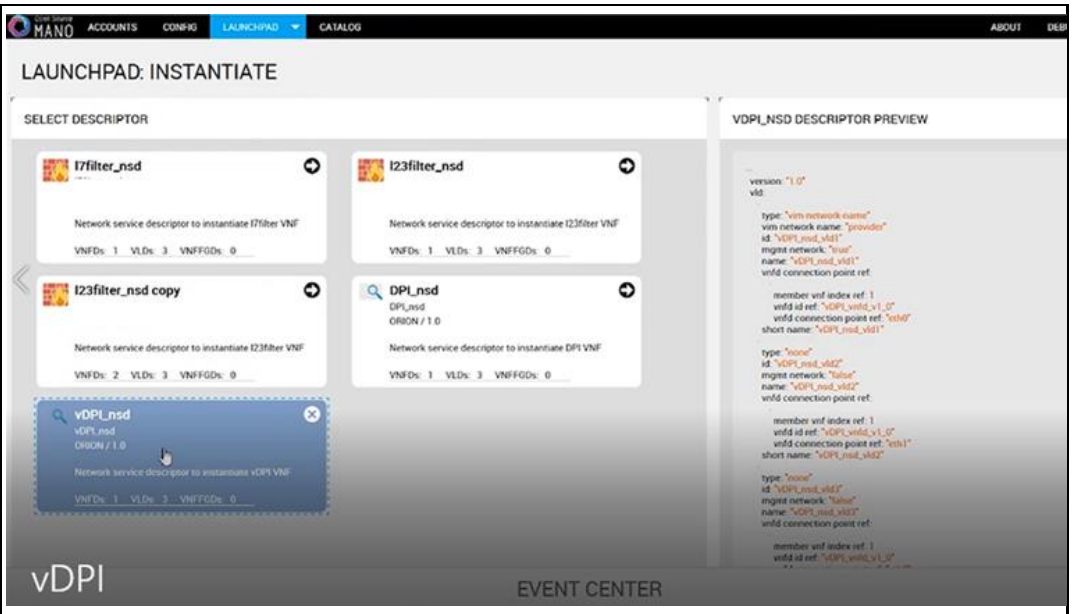
Acceptance Criteria	The VNF is able to provide its functionality (classification/monitoring) without significant contribution to E2E latency.
Testing & Quality control	To test if the VNF fulfils the requirements, additional infrastructure is required, such as an instance of OSM and an OpenStack NFVI-PoP, a traffic generator (e.g. moongen) and/or network traffic captures with labelled data.
Additional information	The VNF package contains the VNF image, charm configuration and related descriptors.
Resource Requirements	Two virtual processors deployed in 40 GB storage/4 GB memory flavour are recommended.

Table 4: vDPI technical specifications.

4.6.3 VNF: Integration and Testing Results

This group of tests has been performed for the vDPI network service in order to evaluate its performance. The results collected were summarized in the tables below.

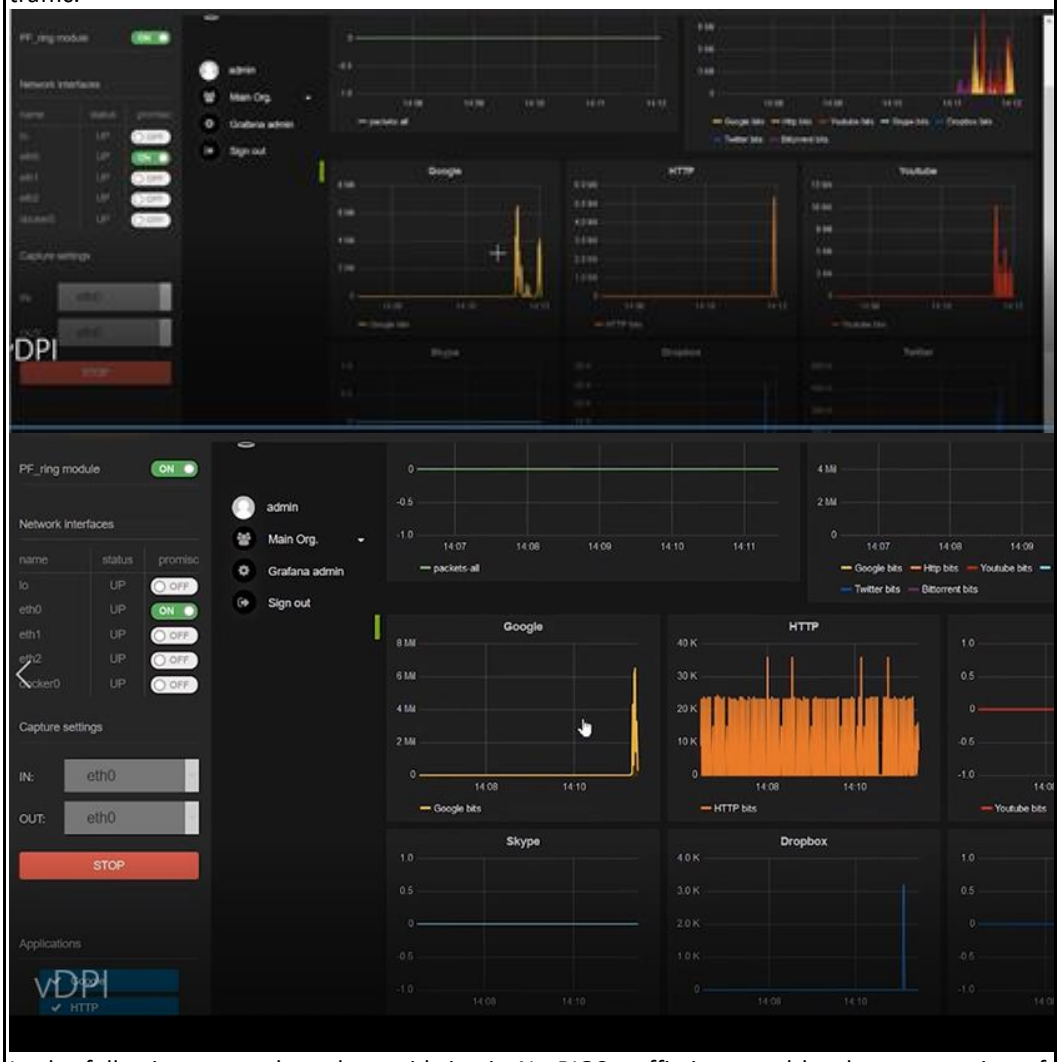
Test Case ID	**WP5-ORION-01**			
Description	ORION vDPI deployment and lifecycle management			
Executed by	ORION		Date	M28
Purpose	This test aims to verify the correct deployment and configuration of the ORION vDPI in an OpenStack-based NFVI, using the OSM orchestrator.			
Components involved	OSM orchestrator, OpenStack, ORION vDPI, ORION vDPI VNF/NS descriptors.			
Tools	OSM, OpenStack.			
Metrics	Deployment time: 2 minutes			
Pre-test conditions	VNF package & VNF/NS descriptors for the OSM version used in the test (Rel. 5). Existing resources in OpenStack (2 vCPUs, 8 GB RAM, 20 GB storage).			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	The operator initiates the action to deploy the vDPI service at the edge.	OK
	2	Check	The system translates the request to an OSM compliant descriptor and instantiates the NSD, which then is realized at the OpenStack.	OK
Evidence	The following screenshots verify the following system responses: • The VNF image is uploaded to OpenStack • VNF/NS descriptors are on-boarded in OSM • A new vDPI network service (NS) is successfully instantiated through OSM through OSM • The running vDPI service is successfully terminated through OSM			

	
Verdict	Success: The VNF was deployed successfully in OpenStack NFVI and was able to receive and apply lifecycle messages via OSM.
Comments	N/A

Test Case ID	**WP5-ORION-02**			
Description	ORION vDPI functional testing.			
Executed by	ORION		Date	M28
Purpose	This test aims to verify the correct function of the ORION vDPI.			
Components involved	OSM orchestrator, OpenStack, ORION vDPI, ORION vDPI VNF/NS descriptors.			
Tools	ORION network capture with multiple protocols to be detected by the vDPI, a VM to start the TCPREPLAY[4].			
Metrics	The application protocols detected by the vDPI, triggered by the pre-captured PCAP file.			
Pre-test conditions	A running vDPI instance in the ORION OpenStack NFVI and an edge enabler is running in order to facilitate the local breakout for the traffic LTE traffic to reach the edge vDPI.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	Configure vDPI to listen for traffic on the appropriate interface	OK
	2	Stimulus	A browser with access to the vDPI front-end	OK
	3	Stimulus	Replay ORION traffic capture	OK
	4	Result	The defined protocols in the replayed traffic are monitored by the vDPI and visualized in the front-end	OK
Evidence	Start of traffic replay: vDPI front-end interface:			

[4] See: https://tcpreplay.appneta.com/

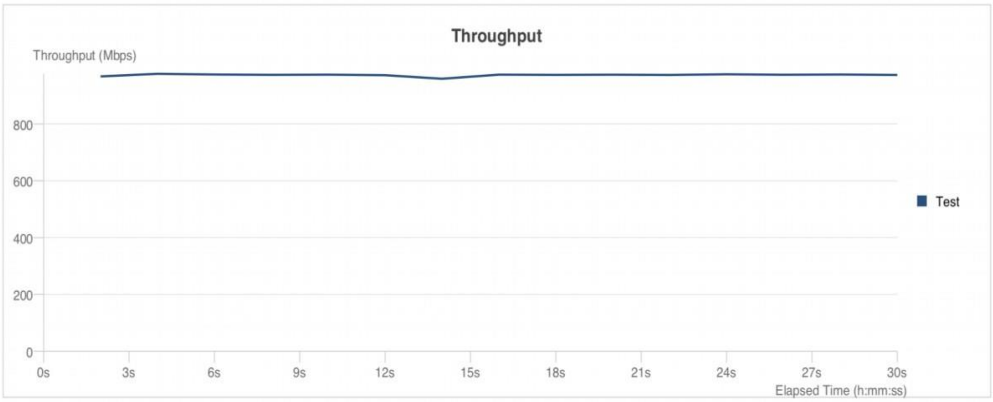
In the following screenshots, the vDPI detects normal HTTP, *Google*, *YouTube* and *Dropbox* traffic.



In the following screenshot, the rapid rise in NetBIOS traffic is caused by the propagation of WannaCry malware infection[5]. The vDPI can then be chained with a firewall VNF so cyberattack traffic can be blocked to prevent additional infection in the network.

[5] For further details also see, *i.a.*: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack

Verdict	Success: The VNF successfully recognized multiple types of traffic and application protocols. The VNF detected attack traffic from a malware infection that was propagating through the network.
Comments	N/A

Test Case ID	**WP5-ORION-03**			
Description	ORION vDPI performance & monitoring.			
Executed by	ORION	Date		M28
Purpose	This test aims to test the performance of the vDPI as the traffic increases and monitor resource consumption.			
Components involved	OSM orchestrator, OpenStack, ORION vDPI, ORION vDPI VNF/NS descriptors.			
Tools	ORION network capture with multiple protocols to be detected by the vDPI, a VM to start the TCPREPLAY.			
Metrics	As traffic increases: • % CPU utilization • % RAM utilization			
Pre-test conditions	A running vDPI instance in the ORION OpenStack NFVI with DPDK enabled, a running instance of Prometheus.			
Test Sequence	Step	Type	Description	Result
	1	Stimulus	1 Gbit/s of application traffic was transmitted to the receiving interface	
	2	Stimulus	The vDPI reports the received and processed traffic	
	3	Stimulus	CPU and RAM are also reported in the monitoring dashboard.	
	4	Result	The results show a direct relationship between the traffic received and the CPU, RAM utilization	OK
Evidence				
Verdict	Success: The vDPI reported recognized traffic of about 930 Mbit/s, and with average CPU utilization of 15% and a peak of 25%—the RAM utilization 350 MB. The application mix that was used for the tests simulated ten users using different applications from *Facebook* to *Netflix*.			
Comments	The purpose of this test is to measure the impact of high-volume application traffic in the vDPI. Consequently, the tests concluded that complicated application traffic profiles instigate high CPU usage in comparison to RAM usage, which is merely affected.			

5 Summary

In this deliverable, the main research achievements of Tasks 5.1-5.3 of Work package 5 have been presented. They dealt will selected technical topics that concerned performance evaluation of the existing solutions (especially in the area of RAN and network virtualization) but also some mechanisms that can be exploited in future releases of the 5G network or beyond have been proposed. It is worth noting that the initial WP5 achievements were already presented in D5.1.

In the RAN area, there is still room for MIMO improvement as well as advanced scheduling mechanisms that can be customized for specific applications. The RAN transport is a cost-sensitive issue as the number of 6G base stations as well as per user throughput is expected to be very high.

Fronthaul architectures with analogue transport and digital signal processing at the end stations are promising as they have the potential to achieve high spectral efficiencies, increased flexibility and reduced latency. Such a DSP-assisted fronthaul has thus been proposed as an alternative to digital (packetized) fronthaul. However, the fronthaul has to be as scalable and flexible as possible for a number of reasons. Primarily, so that it can be used in 5G and beyond applications that will require support for multiplexing of signals with different numerologies, different bandwidths and massive MIMO. Also, so that it can operate within a network slicing/orchestration regime, perhaps in combination with a digital fronthaul/midhaul. Furthermore, such a fronthaul can be used to extend WDM systems (such as the one proposed by ORAN) by increasing their resource allocation resolution. To this end, and continuing on from work presented in D5.1, digital techniques for frequency domain multiplexing/de-multiplexing large numbers of channels are contrasted: one operating on the pre-Inverse Fast Fourier Transform (IFFT) “frequency-domain” samples while the other does so on the post-IFFT “time-domain” samples. The frequency-domain samples technique is very flexible and offers both lower overall complexity and better performance in terms of EVM. The time-domain approach is also flexible, but it requires significantly higher complexity and suffers from very narrow channel spacings, impairing the potential for achieving very high spectral efficiencies. However, under specific conditions, namely, when transporting non-power of 2 numbers of channels and/or when employing larger channel spacings, the time-domain samples approach can lead to significantly reduced sampling rates and may thus be preferable. Both techniques can be used in DSP-assisted front hauling for 5G (and beyond) mobile networks offering flexibility that is not achievable by traditional SCM methods, while combinations of the two techniques can be envisaged for a system that is even more flexible. Indeed, a thorough investigation of such a combined approach has been carried out. It has been shown that appropriately combining the multiplexing techniques can balance sampling rate and complexity requirements, leading to hardware simplification while maintaining improved performance. Furthermore, the DSP techniques presented here are agnostic to the underlying optical technology used to transport the signals (whether these are radio channels or “bits”). Indeed, a radio-over-fibre fronthaul with intensity modulation in the downlink and phase modulation with interferometric detection in the uplink for simplified and power-efficient remote units has been proposed and demonstrated. An experimental investigation and verification of theoretical and simulation performance models have been conducted, demonstrating the ability of such an architecture to transport single-channel and multi-channel 5G-type radio waveforms that are have been digitally processed at DU and RU.

Network slicing is a revolutionary technology that enables customization of the network according to service needs. Unfortunately, the technology deployment is much slower than expected, which negatively impacted the work package activities. Till the end of the project, it was impossible to establish a low-cost testbed that could support network slicing experimentations. Moreover, project trials with network slicing have been postponed. At the moment of writing of this deliverable, there was none commercially deployed slicing-enabled 5G network. The lack of the Standalone 5G testbed also had a negative impact on the originally proposed work on context-aware services. In the framework of this work package, some laboratory tests for the evaluation of the lifecycle performance of the OSM platform (a MANO compliant orchestrator) have been performed. The platform has shown excellent performance and stability, showing that even multiple instances of EPC (MAGMA template)

can be efficiently orchestrated in a batch mode. The Katana slice manager has been used for energy-efficient orchestration of slices.

As RAN slicing is still an open research area, we have made an overview of approaches to RAN slicing. This high-level description has been provided in the context of the extension of the O-RAN platform, an industry RAN focused component that so far has no support for network slicing. To that end, we have also proposed a new architecture of O-RAN that supports network slicing by adding components supporting network slicing to the near-RT RIC controller. We have also analysed a potential integration of network slicing enabled O-RAN with MEC and SON solutions, as we have found that these platforms have both complementary and overlapping functionalities, and integration of them may bring multiple benefits. Having analysed the present MANO-based orchestration approaches, also applied by 3GPP, we have found that in large scale deployment, the solution can impose many issues that are listed in the document. One of the identified problems of MANO is the lack of separations of concerns. Thinking about beyond 5G networks, we have proposed a new concept, 6G-LEGO, the ecosystem enabling the creation of self-managed slices that can be easily stitched together. Moreover, in this approach, the orchestrator functionality is simplified and reduced to slice agnostic resource orchestration only.

As the network function virtualization raises performance issues, we have designed and implemented the virtual Deep Packet Inspection function (vDPI), and we have made a performance analysis of the component. The analysis has shown the excellent performance of the function showing that we were able to minimize the impact of software processing on DPI efficiency.

A significant part of the presented activities is tightly linked with WP3 and WP4 scope of work and has already been reported in the deliverables D3.2, D3.3, and D4.4. The research on some of the technical topics was carried in cooperation with partners from China. It especially concerns O-RAN issues.

The 5G network is still evolving, and the new 3GPP Releases are developed continuously. The work performed within this work package shows that the solution still can be improved. Some of the topics described in this deliverable go beyond 5G and can be seen as candidates for 6G networks, but there is no doubt that the work on many of the topics mentioned in this document has to be continued.

6 References

[1] S. K. Garakoui, E. A. M. Klumperink, B. Nauta, F. E. van Vliet, "Phased-array antenna beam squinting related to frequency dependency of delay circuits", in 2011 8th European Radar Conference, Oct. 2011, pp. 416–419.

[2] H. Zhu, "Performance comparison between distributed antenna and microcellular systems", IEEE Journal on Selected Areas in Communications, vol. 29, no. 6, Jun. 2011, pp. 1151–1163.

[3] M. Cai, J. N. Laneman, B. Hochwald, "Beamforming codebook compensation for beam squint with channel capacity constraint", in 2017 IEEE International Symposium on Information Theory (ISIT), Jun. 2017, pp. 76–80.

[4] G. Li, H. Zhao, H. Hui, "Beam squint compensation for hybrid precoding in millimetre-wave communication systems", Electronics Letters, vol. 54, no. 14, 2018, pp. 905–907.

[5] J. P. Gonzalez-Coma, W. Utschick, L. Castedo, "Hybrid lisa for wideband multiuser millimeter-wave A6 communication systems under beam squint", IEEE Transactions on Wireless Communications, vol. 18, no. 2, Feb. 2019, pp. 1277–1288.

[6] M. Cai, K. Gao, D. Nie, B. Hochwald, J. N. Laneman, H. Huang, K. Liu, "Effect of wideband beam squint on codebook design in phased-array wireless systems", IEEE Global Communications Conference (GLOBECOM) 2016, Dec. 2016, pp. 1–6.

[7] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems", IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 3, Apr. 2016, pp. 436–453.

[8] J. Wang, H. Zhu, L. Dai, N. J. Gomes, J. Wang, "Low-complexity beam allocation for switched-beam based multiuser massive MIMO systems", IEEE Transactions on Wireless Communications, vol. 15, no. 12, Dec. 2016, pp. 8236-8248.

[9] J. Kuang and W Deng, "5G Massive MIMO technology and effect to networking and optimization", China Mobile Communications, vol 43., Aug. 2019, pp. 202–207.

[10] S. Xue, A. Li, J. Wang, N. Yi, Y. Ma, R. Tafazolli, T. Dodgson, "To learn or not to learn: Deep learning assisted wireless modem design", ZTE magazine, 2019.

[11] 5G-DRIVE project, "D5.1: First year report of 5G technology and service innovations", [Online]. Available: https://5g-drive.eu/download/3272/. Accessed: 24/05/2021.

[12] S. Xue, Y. Ma, N. Yi, R. Tafazolli, "Unsupervised deep learning for MU-SIMO joint transmitter and noncoherent receiver design", IEEE Wireless Commun. Lett., vol. 8, no. 1, Feb. 2019, pp. 177–180.

[13] T. J. O'Shea, T. Erpek, T. C. Clancy, "Deep learning-based MIMO communications", CoRR, vol. abs/1707.07980, 2017.

[14] S. Xue, Y. Ma, A. Li, N. Yi, R. Tafazolli, "On unsupervised deep learning solutions for coherent MU-SIMO detection in fading channels", in 2019 IEEE Int. Conf. Commun., May 2019, pp. 1–6.

[15] J. Wang, Y. Ma, N. Yi and R. Tafazolli, "On URLLC Downlink Transmission Modes for MEC Task Offloading", in Proc. VTC-SPRING 2020, Antwerp, 2020, pp. 1–6.

[16] B. Gu, Z. Zhou, "Task offloading in vehicular mobile edge computing: A matching-theoretic framework", IEEE Vehicular Technology Magazine, vol. 14, no. 3, Sep. 2019, pp. 100–106.

[17] A. Tsokalo, H. Wu, G. T. Nguyen, H. Salah, F. H. P. Fitzek, "Mobile edge cloud for robot control services in industry automation", in 2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC), Jan. 2019, pp. 1–2.

[18] X. Yang, Z. Chen, K. Li, Y. Sun, H. Zheng, "Optimal task scheduling in communication-constrained mobile edge computing systems for wireless virtual reality", in 2017 23rd Asia-Pacific Conference on Communications (APCC), Dec. 2017, pp. 1–6.

[19] B. Yang, X. Cao, J. Bassey, X. Li, T. Kroecker, L. Qian, "Computation offloading in multi-access edge computing networks: A multitask learning approach", in 2019 IEEE International Conference on Communications (ICC), May 2019, pp. 1–6.

[20] Y. Polyanskiy, H. V. Poor, S. Verdu, "Channel coding rate in the finite blocklength regime", IEEE Trans. Inf. Theory, vol. 56, no. 5, May 2010, pp. 2307–2359.

[21] S. Guo *et al.*, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing", in Proc. IEEE INFOCOM 2016 – The 35th Annual IEEE International Conference on Computer Communications, 2016, pp. 1–9.
[22] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications", IEEE Access, vol. 6, Feb. 2018, pp. 12825–12837.
[23] S. Zhang, N. Yi, Y. Ma "Correlation-based device energy-efficient dynamic multi-task offloading for mobile edge computing", in Proc. VTC-SPRING 2021, pp. 1–6.
[24] N. Abbas, Y. Zhang, A. Taherkordi, T. Skeie, "Mobile edge computing: A survey", IEEE Internet Things J., vol. 5, no. 1, Feb. 2018, pp. 450–465.
[25] K. Zhang, Y. Zhu, S. Maharjan, Y. Zhang, "Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of things", IEEE Netw., vol. 33, no. 5, Sep.–Oct. 2019, pp. 12–19.
[26] M. S. Elbamby *et al.*, "Wireless edge computing with latency and reliability guarantees", Proc. IEEE, vol. 107, no. 8, Aug. 2019, pp. 1717–1737.
[27] C. You, K. Huang, H. Chae, B. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading", IEEE Trans. Wireless Commun., vol. 16, no. 3, Mar. 2017, pp. 1397–1411.
[28] Y. Mao, J. Zhang, K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices", IEEE J. Sel. Areas Commun., vol. 34, no. 12, Dec. 2016, pp. 3590–3605.
[29] J. Xu, L. Chen, S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing", IEEE Trans. Cogn. Commun. Netw., vol. 3, no. 3, Jul. 2017, pp. 361–373.
[30] R. S. Sutton, A. G. Barto, "Reinforcement Learning: An Introduction", Cambridge, MA: MIT Press, 1998.
[31] Y. Sun, M. Peng, Y. Zhou, Y. Huang, S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues", IEEE Commun. Surveys Tuts., vol. 21, no. 4, Q4 2019, pp. 3072–3108.
[32] X. Chen, Z. Zhao, H. Zhang, "Stochastic power adaptation with multiagent reinforcement learning for cognitive wireless mesh networks", IEEE Trans. Mobile Comput., vol. 12, no. 11, Nov. 2013, pp. 2155–2166.
[33] X. Chen, Z. Han, H. Zhang, G. Xue, Y. Xiao, M. Bennis, "Wireless resource scheduling in virtualized radio access networks using stochastic learning", IEEE Trans. Mobile Comput., vol. 17, no. 4, Apr. 2018, pp. 961–974.
[34] X. Chen, C. Wu, T. Chen, H. Zhang, Z. Liu, Y. Zhang, M. Bennis, "Age of information-aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective", IEEE Trans. Wireless Commun., vol. 19, no. 4, Apr. 2020, pp. 2268–2281.
[35] K. Zhang, Y. Zhu, S. Leng, Y. He, S. Maharjan, Y. Zhang, "Deep learning empowered task offloading for mobile edge computing in urban informatics", IEEE Internet Things J., vol. 6, no. 5, Oct. 2019, pp. 7635–7647.
[36] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, M. Zorzi, "Toward 6G networks: Use cases and technologies", IEEE Commun. Mag., vol. 58, no. 3, Mar. 2020, pp. 55–61.
[37] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach", IEEE J. Sel. Areas Commun., vol. 37, no. 10, Oct. 2019, pp. 2377–2392.
[38] E. Moutaly, P. Assimakopoulos, S. Noor, S. Faci, A. Billabert, N. J. Gomes, M. L. Diakite, C. Browning, C. Algani, "Phase Modulated Radio-Over-Fiber for Efficient 5G Fronthaul Uplink", in Journal of Lightwave Technology, vol. 37, no. 23, Dec. 2019, pp. 5821–5832.
[39] 3GPP, "Release description; Release 15", 3GPP TR 21.915, v15.0.0, Oct. 2019.
[40] Y. Liu *et al.*, "Waveform design for 5G networks: Analysis and comparison", IEEE Access, vol. 5, 2017, pp. 19282–19292.
[41] 3GPP, "NR; Base Station (BS) radio transmission and reception", 3GPP TS 38.104, v17.1.0, Apr. 2021.

[42] S. Noor, P. Assimakopoulos, N. J. Gomes, “A flexible subcarrier multiplexing system with analog transport and digital processing for 5G (and beyond) fronthaul”, J. Lightwave Technol., vol. 37, 2019, pp. 3689–3700.

[43] P. Sehier, P. Chanclou, N. Benzaoui, D. Chen, K. Kettunen, M. Lemke, Y. Pointurier, P. Dom, “Transport evolution for the RAN of the future [Invited]”, J. Opt. Commun. Netw., vol. 11, 2019, pp. B97–B108.

[44] Ericsson AB, Huawei Technologies Co. Ltd, NEC Corporation, Nokia, “Common Public Radio Interface: eCPRI Interface Specification”, eCPRI Specification V2.0, 2019, [Online]. Available: https://www.gigalight.com/downloads/standards/ecpri-specification.pdf. Accessed: 24/05/2021.

[45] N. J. Gomes, P. Sehier, H. Thomas, P. Chanclou, B. Li, D. Munch, P. Assimakopoulos, S. Dixit, V. Jungnickel, “Boosting 5G through Ethernet: how evolved fronthaul can take next-generation mobile to the next level”, IEEE Veh. Technol. Mag., vol. 13, 2018, pp. 74–84.

[46] 3GPP, “Study on new radio access technology: radio access architecture and interfaces”, 3GPP TR 38.801 v14.0.0, Apr. 2017.

[47] N. J. Gomes, P. Assimakopoulos, “Optical fronthaul options for meeting 5G requirements”, in 20th International Conference on Transparent Optical Networks (ICTON), Bucharest, Romania, 2018, pp. 1–4.

[48] P. Assimakopoulos, J. Zou, K. Habel, J.-P. Elbers, V. Jungnickel, N. J. Gomes, “A converged evolved Ethernet fronthaul for the 5G era”, IEEE J. Sel. Areas Commun. 36, 2018, pp. 2528–2537.

[49] ITU-T, “Radio-over-fiber (RoF) technologies and their applications”, ITU-T Recommendation G.Sup55, 2015, [Online]. Available: https://www.itu.int/rec/T-REC-G.Sup55/en. Accessed: 24/05/2021.

[50] ITU-T, “Radio over fibre systems”, ITU-T Recommendation G.9803, 2018, [Online]. Available: https://www.itu.int/rec/T-REC-G.9803/en. Accessed: 24/05/2021.

[51] S. A. Khwandah, J. P. Cosmas, I. A. Glover, P. I. Lazaridis, N. R. Prasad, Z. D. Zaharis, “Direct and external intensity modulation in OFDM RoF links”, IEEE Photon. J., vol. 7, 2015, pp. 1–10.

[52] N. J. Gomes, P. Assimakopoulos, M. K. Al-Hares, U. Habib, S. Noor, “The new flexible mobile fronthaul: digital or analog, or both?”, in 18th International Conference on Transparent Optical Networks (ICTON), Trento, Italy, 2016, pp. 1–4.

[53] C. Westphal, “Challenges in networking to support augmented reality and virtual reality (Invited)”, in IEEE Conference on Computing, Networking and Communications (ICNC) (2017).

[54] S. Noor, P. Assimakopoulos, M. Wang, H. A. Abdulsada, N. Genay, L. A. Neto, P. Chanclou, N. J. Gomes, “Comparison of digital signal processing approaches for subcarrier multiplexed 5G and beyond analog fronthaul”, J. Opt. Commun. Netw. 12, 2020, pp. 62–71.

[55] M. Sung, S.-H. Cho, J. Kim, J. K. Lee, J. H. Lee, H. S. Chung, “Demonstration of IFoF-based mobile fronthaul in 5G prototype with 28-GHz millimeter wave”, J. Lightwave Technol., vol. 36, 2018, pp. 601–609.

[56] T. Tsou, “Ettus research future directions: 4th OAI Workshop”, 2017, [Online]. Available: https://www.openairinterface.org/docs/workshop/4_OAI_Workshop_20171107/Talks/TSOU_oai-paris-ettus.pdf. Accessed: 24/05/2021.

[57] National Instruments, “Building an Affordable 8×8 MIMO Testbed with NI USRP”, 2014.

[58] N. J. Fliege, Multirate Digital Signal Processing: Wiley, 1994, p. 229.

[59] Mathworks, “dsp.CICInterpolator”, [Online]. Available: https://uk.mathworks.com/help/dsp/ref/dsp.cicinterpolator-system-object.html. Accessed: 24/05/2021.

[60] P. Assimakopoulos, S. Noor, N. J. Gomes, “Flexible and Efficient DSP-assisted Subcarrier Multiplexing for an Analog Mobile Fronthaul”, IEEE Photonics Technology Letter (PTL), vol. 33, no. 5, Mar. 2021.

[61] ITU-R, “IMT Vision — Framework and overall objectives of the future development of IMT for 2020 and beyond”, ITU-R Recommendation M.2083-0, Sep. 2015.

[62] 3GPP, “System Architecture for the 5G System”, 3GPP TS 23.501, v17.0.0, Mar. 2021.

[63] 3GPP, “NR; Overall description; Stage-2”, 3GPP TS 38.300, v16.5.0, Mar. 2021.

[64] 3GPP, “Service requirements for the 5G system”, 3GPP TS 22.261, v18.2.0, Apr. 2021.

[65] 3GPP, “Study on Scenarios and Requirements for Next Generation Access Technologies”, 3GPP TR 38.913, v16.0.0, Jul. 2020.

[66] 3GPP, “Procedures for the 5G System (5GS)”, 3GPP TS 23.502, v17.0.0, Mar. 2021.

[67] S. E. Elayoubi, S. B. Jemaa, Z. Altman, A. Galindo-Serrano, “5G RAN Slicing for Verticals: Enablers and Challenges”, IEEE Commun. Mag., vol. 57, no. 1, Jan. 2019, pp. 28–34.

[68] CPRI, “Common Public Radio Interface (CPRI); Interface Specification”, CPRI Specification V7.0, Oct. 2015 [Online]. Available: http://www.cpri.info/spec.html. Accessed: 24/05/2021.

[69] Small Cell Forum, FAPI and nFAPI specifications, Rel. 9.0, May 2017.

[70] G. S. Birring, P. Assimakopoulos, N. J. Gomes, “An Ethernet-based fronthaul implementation with MAC/PHY split LTE processing”, in Proc. Global Commun. Conf., Singapore, Dec. 2017, pp. 1–6.

[71] K. Miyamoto, S. Kuwano, T. Shimizu, J. Terada, A. Otaka, “Performance evaluation of Ethernet-based mobile fronthaul and wireless CoMP in split-PHY processing”, J. Opt. Commun. Netw., vol. 9, no. 1, 2017 pp. A46–A54.

[72] iCIRRUS project, [Online]. Available: https://www.icirrus-5gnet.eu/. Accessed: 24/05/2021.

[73] IEEE, Time-Sensitive Networking (TSN) Task Group, [Online]. Available: https://1.ieee802.org/tsn/. Accessed: 24/05/2021.

[74] IETF, Deterministic Networking (DetNet) Working Group, [Online]. Available: https://datatracker.ietf.org/wg/detnet/documents/. Accessed: 24/05/2021.

[75] A. Nasrallah *et al.*, “Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards and Related 5G ULL Research”, IEEE Comms Surveys & Tutorials, vol. 21, no. 1, 2019, pp. 88–145.

[76] IEEE, Time-Sensitive Networking for Fronthaul, IEEE Standard P802.1CM, 2018, [Online]. Available: http://www.ieee802.org/1/pages/802.1cm.html. Accessed: 24/05/2021.

[77] IEEE, Enhancements for Scheduled Traffic, IEEE Standard 802.1Qbv, 2016, [Online]. Available: http://www.ieee802.org/1/pages/802.1bv.html. Accessed: 24/05/2021.

[78] OIF, “Flex Ethernet Implementation Agreement”, IA OIF-FLEXE-01.0, Mar. 2016.

[79] K. Katsalis, L. Gatzikis, K. Samdanis, “Towards Slicing for Transport Networks: The Case of Flex-Ethernet in 5G”, in Proc. IEEE Conf. on standards for Commun. and Networking (CSCN), Paris, France, Oct. 2018, pp. 1–7.

[80] D. Avidor, S. Mukherjee, J. Ling, C. Papadias, “On some properties of the proportional fair scheduling policy”, in 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No. 04TH8754), Barcelona, Spain, 2004, pp. 853–858, doi: 10.1109/PIMRC.2004.1373820.

[81] T. Girici, C. Zhu, J. R. Agre, A. Ephremides, “Proportional fair scheduling algorithm in OFDMA-based wireless systems with QoS constraints”, J. Commun. Netw., vol. 12, no. 1, Feb. 2010, pp. 30–42.

[82] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, I. Stoica, “Dominant Resource Fairness: Fair Allocation of Multiple Resource Types”, p. 14.

[83] T. Bonald, J. Roberts, “Scheduling network traffic”, SIGMETRICS Perform. Eval. Rev., vol. 34, no. 4, Mar. 2007, p. 29.

[84] Qualcomm, “Future of 5G: Building a unified, more capable 5G air interface for the next decade and beyond”, Feb. 2020.

[85] K. I. Pedersen, G. Pocovi, J. Steiner, S. R. Khosravirad, “Punctured Scheduling for Critical Low Latency Data on a Shared Channel with Mobile Broadband”, in 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), Toronto, ON, 2017, pp. 1–6.

[86] A. Anand, G. de Veciana, S. Shakkottai, “Joint Scheduling of URLLC and eMBB Traffic in 5G Wireless Networks”, arXiv:1712.05344 [cs], Aug. 2018.

[87] N. H. Mahmood, R. Abreu, R. Bohnke, M. Schubert, G. Berardinelli, T. H. Jacobsen, “Uplink Grant-Free Access Solutions for URLLC services in 5G New Radio”, in 2019 16th International Symposium on Wireless Communication Systems (ISWCS), Oulu, Finland, 2019, pp. 607–612.

[88] M. C. Lucas-Estañ, J. Gozalvez, M. Sepulcre, “On the Capacity of 5G NR Grant-Free Scheduling with Shared Radio Resources to Support Ultra-Reliable and Low-Latency Communications”, Sensors, vol. 19, no. 16, Aug. 2019, p. 3575.

[89] T. Dang, M. Peng, “Delay-Aware Radio Resource Allocation Optimization for Network Slicing in Fog Radio Access Networks”, in 2018 IEEE International Conference on Communications Workshops (ICC Workshops), Kansas City, MO, USA, 2018, pp. 1–6.

[90] M. Alsenwi, N. H. Tran, M. Bennis, A. Kumar Bairagi, C. S. Hong, “eMBB-URLLC Resource Slicing: A Risk-Sensitive Approach”, IEEE Commun. Lett., vol. 23, no. 4, Apr. 2019, pp. 740–743.

[91] R. Kokku, R. Mahindra, H. Zhang, S. Rangarajan, “NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks”, IEEE/ACM Trans. Networking, vol. 20, no. 5, Oct. 2012, pp. 1333–1346.

[92] R. Kokku, R. Mahindra, H. Zhang, S. Rangarajan, “CellSlice: Cellular wireless resource slicing for active RAN sharing”, in 2013 Fifth International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 2013, pp. 1–10.

[93] A. Aijaz, “Hap-SliceR: A Radio Resource Slicing Framework for 5G Networks With Haptic Communications”, IEEE Systems Journal, vol. 12, no. 3, Sep. 2018, pp. 2285–2296.

[94] R. Ferrus, O. Sallent, J. Perez-Romero, R. Agusti, “On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration”, IEEE Commun. Mag., vol. 56, no. 5, May 2018, pp. 184–192.

[95] Vila, O. Sallent, A. Umbert, J. Perez-Romero, “An Analytical Model for Multi-Tenant Radio Access Networks Supporting Guaranteed Bit Rate Services”, IEEE Access, vol. 7, 2019, pp. 57651–57662.

[96] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, X. Costa-Perez, “A Machine Learning Approach to 5G Infrastructure Market Optimization”, IEEE Trans. on Mobile Comput., vol. 19, no. 3, Mar. 2020, pp. 498–512.

[97] J. He, W. Song, “AppRAN: Application-oriented radio access network sharing in mobile networks”, in 2015 IEEE International Conference on Communications (ICC), London, 2015, pp. 3788–3794.

[98] Y. Jia, H. Tian, S. Fan, P. Zhao, K. Zhao, “Bankruptcy game-based resource allocation algorithm for 5G Cloud-RAN slicing”, in 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, 2018, pp. 1–6.

[99] P. Caballero, A. Banchs, G. De Veciana, X. Costa-Perez, “Network Slicing Games: Enabling Customization in Multi-Tenant Mobile Networks”, IEEE/ACM Trans. Networking, vol. 27, no. 2, Apr. 2019, pp. 662–675.

[100] V. Sciancalepore, F. Cirillo, X. Costa-Perez, “Slice as a Service (SlaaS) Optimal IoT Slice Resources Orchestration”, in GLOBECOM 2017 – 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1–7.

[101] M. Alba, J. H. G. Velasquez, W. Kellerer, “An adaptive functional split in 5G networks”, in IEEE INFOCOM 2019 – IEEE Conference on Computer Communications Workshops (INFOCOM

WKSHPS), Paris, France, 2019, pp. 410–416.

[102] T. Guo, R. Arnott, "Active LTE RAN Sharing with Partial Resource Reservation", in 2013 IEEE 78th Vehicular Technology Conference (VTC Fall), Las Vegas, NV, USA, 2013, pp. 1–5.

[103] J. Perez-Romero, O. Sallent, R. Ferrus, R. Agusti, "Admission control for multi-tenant Radio Access Networks", in 2017 IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, 2017, pp. 1073–1078.

[104] H. M. Soliman, A. Leon-Garcia, "QoS-Aware Frequency-Space Network Slicing and Admission Control for Virtual Wireless Networks", in 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA, 2016, pp. 1–6.

[105] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, C. Verikoukis, "Sliced-RAN: Joint Slicing and Functional Split in Future 5G Radio Access Networks", in ICC 2019 – 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1–6.

[106] B. Khodapanah, A. Awada, I. Viering, J. Francis, M. Simsek, G. P. Fettweis, "Radio Resource Management in context of Network Slicing: What is Missing in Existing Mechanisms?", in 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1–7.

[107] O-RAN Alliance, "O-RAN Use Cases and Deployment Scenarios", White Paper, Feb. 2020.

[108] A. Ksentini, N. Nikaein, "Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction", in IEEE Communications Magazine, vol. 55, no. 6, June 2017, pp. 102–108.

[109] X. Foukas, M. K. Marina, K. Kontovasilis, "Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture", in Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking – MobiCom '17, Snowbird, Utah, USA, 2017, pp. 127–140.

[110] X. Chen *et al.*, "Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks", IEEE J. Sel. Areas Commun., vol. 33, no. 4, Apr. 2015, pp. 627–640.

[111] J. Nicholson, B. D. Noble, "BreadCrumbs: Forecasting mobile connectivity", in Proc. ACM MobiCom, San Francisco, CA, Sep. 2008.

[112] Z. Ji, K. J. R. Liu, "Dynamic spectrum sharing: A game theoretical overview", IEEE Commun. Mag., vol. 45, no. 5, May 2007, pp. 88–94.

[113] X. He *et al.*, "Privacy-aware offloading in mobile-edge computing", in Proc. IEEE GLOBECOM, Singapore, Dec. 2017.

[114] Y. Wu *et al.*, "Secrecy-based energy-efficient data offloading via dual connectivity over unlicensed spectrums", IEEE J. Sel. Areas Commun., vol. 34, no. 12, Dec. 2016, pp. 3252–3270.

[115] T. D. Burd, R. W. Brodersen, "Processor design for portable systems", J. VLSI Signal Process. Syst., vol. 13, no. 2–3, Aug. 1996, pp. 203–221.

[116] M. Fink, "Equilibrium in a stochastic n-person game", J. Sci. Hiroshima Univ. Ser. A-I, vol. 28, 1964, pp. 89–93.

[117] V. Mnih *et al.*, "Human-level control through deep reinforcement learning", Nature, vol. 518, no. 7540, Feb. 2015, pp. 529–533.

[118] H. van Hasselt, A. Guez, D. Silver, "Deep reinforcement learning with double Q-learning", in Proc. AAAI, Phoenix, AZ, Feb. 2016.

[119] L.-J. Lin, "Reinforcement learning for robots using neural networks", Carnegie Mellon University, 1992.

[120] ORAN Alliance, [Online]. https://www.o-ran.org/. Accessed: 24/05/2021.

[121] O-RAN Alliance, O-RAN Software Community (SC), [Online]. https://www.o-ran-sc.org/. Accessed: 24/05/2021.

[122] O-RAN Alliance, “O-RAN Use Cases and Deployment Scenarios”, White Paper, Feb. 2020.

[123] Open Network Automation Platform, “ONAP Platform Architecture”, [Online]. https://www.onap.org/architecture. Accessed: 24/05/2021.

[124] O-RAN Alliance, “O-RAN: Towards an Open and Smart RAN”, White Paper, Oct. 2018.

[125] 3GPP, “Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements”, 3GPP TS 32.500, v16.0.0, Jul. 2020.

[126] L. Tomaszewski, S. Kukliński, R. Kołakowski, “A new approach to 5G and MEC integration”, in: I. Maglogiannis, L. Iliadis, E. Pimenidis (eds.), “Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops”, IFIP Advances in Information and Communication Technology, vol. 585, Springer, Cham, 2020, pp. 15–24.

[127] 3GPP, “Management and orchestration; Self-Organizing Networks (SON) for 5G network”, 3GPP TS 28.313, v17.0.0, Dec. 2020.

[128] 3GPP, “Management and orchestration; Architecture framework”, 3GPP TS 28.533, v16.7.0, Apr. 2021.

[129] ETSI MEC ISG, “Multi-access Edge Computing (MEC); Framework and Reference Architecture”, ETSI GS MEC 003, v2.1.1, Jan. 2019.

[130] 3GPP, “Management and orchestration; Concepts, use cases and requirements”, 3GPP TS 28.530, v17.1.0, Apr. 2021.

[131] ETSI NFV ISG, “Network Functions Virtualization (NFV); Evolution and Ecosystem; Report on Network Slicing Support with ETSI NFV Architecture Framework”, ETSI GS NFV-EVE 012, v3.1.1, Dec. 2017.

[132] ETSI NFV ISG, “Network Functions Virtualisation (NFV); Management and Orchestration”, ETSI GS NFV-MAN 001, v1.1.1, Dec. 2014.

[133] ETSI NFV ISG, “Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Network Service Templates Specification”, ETSI GS NFV-IFA 014, v3.4.1, Jun. 2020.

[134] Y.3110 IMT-2020 network management and orchestration requirements (09/17), ITU-T Recommendation Y.3110, Sep. 2017.

[135] Y.3111 IMT-2020 network management and orchestration framework (09/17), ITU-T Recommendation Y.3111, Sep. 2017.

[136] Y.3112 Framework for the support of multiple network slicing (05/18), ITU-T Recommendation Y.3112, May 2018.

[137] 3GPP, “5G System; Network Exposure Function Northbound APIs; Stage 3”, 3GPP TS 29.522, v17.0.0, Dec. 2020.

[138] 3GPP, “5G System; Network Slice Selection Services; Stage 3”, 3GPP TS 29.531, v17.0.0, Mar. 2021.

[139] 3GPP, “Security architecture and procedures for 5G System”, 3GPP TS 33.501, v17.0.0, Mar. 2021.

[140] 3GPP, “5G System; Network Slice-Specific Authentication and Authorization (NSSAA) services; Stage 3”, 3GPP TS 29.526, v17.0.0, Mar. 2021.

[141] 3GPP, “Study on enhancement of Radio Access Network (RAN) slicing for NR”, 3GPP TR 38.832, v1.0.0, Mar. 2021.

[142] 3GPP, “Management and orchestration; Provisioning”, 3GPP TS 28.531, v16.9.0, Apr. 2021.

[143] 3GPP, “Management and orchestration; Generic management services”, 3GPP TS 28.532, v16.7.0, Apr. 2021.

[144] 5G!Pagoda, “D3.1 – Slice Components Design – ver. 1.0”, 5G!Pagoda Project, Aug. 2017.

[145] S. Kukliński *et al.*, “A reference architecture for network slicing”, in 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, 2018, pp. 217–221.

[146] S. Kukliński, L. Tomaszewski, “DASMO: A scalable approach to network slices management and orchestration”, in NOMS, Taipei, 2018, pp. 1–6, doi: 10.1109/NOMS.2018.8406279.

[147] B. Sayadi *et al.*, “SDN for 5G Mobile Networks: NORMA Perspective”, in CROWNCOM, Grenoble, France, 2016, pp. 741–753.

[148] S. Kukliński, L. Tomaszewski, “Key Performance Indicators for 5G network slicing”, in NetSoft, Paris, France, 2019, pp. 464–471.

[149] M.-A. Kourtis *et al.*, “5G Network Slicing Enabling Edge Services”, 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE, Nov. 2020, pp. 155–160.

[150] 3GPP, “Telecommunication management; Study on management and orchestration of network slicing for next generation network”, 3GPP TR 28.801 v15.1.0, Jan. 2018.

[151] 3GPP, “Telecommunication management; Management concept, architecture and requirements for mobile networks that include virtualized network functions,", 3GPP TS 28.500, v16.0.0, July 2020.

[152] GSMA, “NG.116 Generic Network Slice Template v2.0”, [Online]. https://www.gsma.com/newsroom/wp-content/uploads/NG.116-v2.0.pdf. Accessed: 24/05/2021.

[153] R. Stoica, G. T. F. d. Abreu, “Massively Concurrent NOMA: A Frame-Theoretic Design for Non-Orthogonal Multiple Access”, 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018, pp. 461–466.

[154] L. Deri, M. Martinelli, T. Bujlow, A. Cardigliano, “nDPI: Open-source high-speed deep packet inspection”, 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, 2014, pp. 617–622.

[155] OpenDPI, [Online]. Available: https://github.com/thomasbhatia/OpenDPI. Accessed: 24/05/2021.

[156] PF_RING ZC, [Online]. Available: https://www.ntop.org/products/packet-capture/pf_ring/pf_ring-zc-zero-copy/. Accessed: 24/05/2021.

[157] M. Kourtis *et al.*, “Enhancing VNF performance by exploiting SR-IOV and DPDK packet processing acceleration”, 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, 2015, pp. 74–78.

[158] 5G-DRIVE project, “D3.2: Joint specification for eMBB trial”, confidential – internal report.