SEMANTIC

end-to-end Slicing and data-drivEn autoMAtion of Next generation cellular neTworks with mobIle edge Clouds

Marie Skłodowska-Curie Actions (MSCA)
Innovative Training Networks (ITN)
H2020-MSCA-ITN-2019
861165 - SEMANTIC

WP3 – Optimizations for integrated access/X-haul and end-to-end slicing

D3.2: Design and optimizations for integrated access/X-haul, e2e slicing and traffic steering

Contractual Date of Delivery:	31/12/2022
Actual Date of Delivery:	31/12/2022
Responsible Beneficiary:	EUR
Contributing Beneficiaries:	CTTC, EUR, IQU
Security:	Public
Nature:	Report
Version:	V2.2

Document Information

Version Date: 20/12/2022
Total Number of Pages: 39

Authors

Name	Organization	Email
Swastika Roy	CTTC	sroy@cttc.es
Suvidha Sudhakar Mhatre	Iquadrat Informatica S.L.	s.mhatre@iquadrat.com
Pavlos Doanis	EURECOM	pavlos.doanis@eurecom.fr

Document History

Revision	Date	Modification	Contact Person
V1.0	22/11/2022	Merging contributions of ESRs	Pavlos Doanis (EUR)
V1.4	24/11/2022	Editor Review	Thrasyvoulos Spyropoulos (EUR)
V2.2	20/12/2022	Revisions by ESRs	Pavlos Doanis (EUR)

Table of Contents

List of Acronyms and Abbreviations

Acronym	Description
3GPP	Third Generation Partnership Project
ETSI	European Telecommunications Standards Institute
5G	5th generation mobile network
B5G	Beyond 5G
6G	6th generation mobile network
AI	Artificial Intelligence
XAI	Explainable AI
BBU	Base Band Unit
vBBU	Virtual BBU
CN	Core Network
MEC	Multi Access Edge Computing
RAN	Radio Access Network
C-RAN	Cloud RAN
CPU	Central Processing Unit
CU	Centralized Unit
DU	Distributed unit
e2e	End-to-End
eMBB	Enhanced Mobile Broadband
mMTC	Massive Machine Type Communication
uRLLC	Ultra-Reliable Low Latency Communication
muRLLC	Massive uRLLC
KPI	Key Performance Indicator
QoS	Quality of Service
NS	Network Slicing
NFV	Network Function Virtualization
SDN	Software-Defined Networking
vSDNC	Virtual SDN Controller
SLA	Service Level Agreement
VNF	Virtual Network Function
VL	Virtual Link
IAB	Integrated Access and Backhaul
CPU	Central Processing Unit
CU	Central Unit

DU	Distributed Unit
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
QL	Q-Learning
DQN	Deep Q-Network
NN	Neural Network
DNN	Deep NN
CNN	Convolutional NN
MANO	Management and Orchestration
ML	Machine Learning
FL	Federated Learning
E2E	End-to-end
ZSM	Zero Touch Network and Service Management
NSI	Network slice instance
NSSI	Network slice subnet instance
SON	Self-Organizing Network
TRP	Transmission / Reception Point
MS	Monitoring System
AE	Analytic Engine
CL	Closed Loop
OTT	Over The Top service
MILP	Mixed Integer Linear Programming
PRB	Physical Resource Block
AN	Access Network
OAI	Open Air Interface
MNO	Mobile Network Operator
RIC	RAN Intelligence Controller
O-RAN	Open RAN
UE	User Equipment
ERAB	Enhanced radio Access Bearer
vDFE	Virtual Digital Front End
O-RU	Open RAN Radio Unit
O-DU	Open RAN Distributed Unit
RBG	Resource Block Group

List of Figures

List of Tables

1 Executive summary

This deliverable incorporates some initial proposed optimization frameworks and algorithms towards optimal network slicing, contributed by the SEMANTIC ESRs, in the framework of WP3 (Optimizations for integrated access/X-haul and end-to-end slicing). The proposed solutions consider an end-to-end slicing perspective as well as novel technologies, like integrated access/X-haul and traffic steering. Each Chapter of this report focuses on different aspects of network slicing and corresponds to the work of a different ESR. It includes a state-of-the-art Section (optionally), and then introduces the system model utilized, the problem formulation, a proposed solution, and possibly some preliminary results. In Chapter 3, the slice resource allocation problem at the RAN-Edge domain is examined, and a zero-touch Federated Learning solution combined with Explainable AI techniques is proposed. Chapter 4 tackles the dynamic slice embedding problem, introducing a generic framework suitable for multi-domain networks and end-to-end slice KPIs, while a multi-agent deep Reinforcement Learning approach is proposed. Finally, Chapter 5 provides a framework for slice resource allocation and management in the RAN-Edge domain, proposing distributed deep Reinforcement Learning solutions at different timescales to optimize the resource utilization and distribution.

2 Introduction

The standardization and deployment of the fifth generation (5G) of mobile networks is ongoing during the last few years. Enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low latency communications (uRLLC) are three major communication scenarios. Concurrently, both academia and industry have focused on research for beyond 5G (B5G) and towards 6G networks. These networks are not limited to the three major use cases of 5G [1], they are envisioned to support more vertical application scenarios, having unique features and specific capabilities (e.g., latency, peak date rate, etc.). The parallel provisioning of such heterogeneous vertical services urges a highly flexible, adaptive, and intelligent network architecture, directly contradicting the “one-size-fits-all” network design paradigm.

6G networks are expected to support KPI requirements that are an order of magnitude higher compared to 5G. On the report of a recent white paper [2], 6G KPIs require 1 Tbps peak data rate, around 20-100 Gbps user experienced data rate, 0.1 ms end-to-end latency, 10 million devices/km2, and near 100% coverage, which in turn demand the adoption of several new technologies [1]. Moreover, there will be new unique features such as a space-air-ground integrated network (SAGIN), which will ensure global coverage and on-demand services [3], diversified services with stringent QoS requirements, and ubiquitous intelligence penetrating every corner of the network (spanning from end-users, the network edge, to the remote cloud).

One of the key enablers for supporting various different services with stringent Quality of Service (QoS) requirements is Network Slicing (NS). This technology allows for the flexible management of services, by creating customized virtual networks ("slices") on top of the physical network infrastructure [4], [5], and it is based on network function virtualization (NFV) and Software Defined Networking (SDN). More specifically, NFV allows the creation of virtual resources and network functions for flexible resource management, while SDN provides a global view of network infrastructure and simplifies centralized network management for network optimization. A network slice instance is a complete logical network that includes a set of network functions and the supporting network resources, and it can meet specific network characteristics required by a service instance [6], as mentioned in 3GPP TR 28.801 [7].

Optimal slicing aims to: (i) ensure slice performance isolation (achieve at least a minimum QoS defined by the Service Level Agreement (SLA)); (ii) ensure efficient utilization of the limited network resources. This is extremely important, since the better utilization of the available network resources, the more slices can be hosted with high reliability, and the higher the revenue for the network operator. Or, from a different point of view, the less resources required to host a given number of slices with high reliability, the less the operating expenses and capital expenditures. [8], [9].

Different “slicing problems” have been considered in recent related literature [10], however there are two main versions of the problem that we will examine in this deliverable. The first is the allocation of a physical node’s resources among the hosted slices [11], [12]. The other is the problem of slice embedding, where slices are represented as graphs of Virtual Network Functions (VNFs) and Virtual Links (VLs), that need to be mapped among physical nodes and links respectively, while satisfying each slice's resource demands [13], [14]. In Chapters 3 and 5 we will address the former, by allocating CPU resources to CU (Central Unit) VNFs and radio resources to end users in the RAN-Edge domain respectively, while in Chapter 4 we will address the latter. A survey of the related literature focusing on the slice embedding problem has been already provided in the previous deliverable (D3.1), while some additional related work regarding the resource allocation problem will be presented in the corresponding Chapters.

Despite interesting initial attempts to tackle these problems, a number of challenges arising from the vision of B5G/6G slicing remain. Beyond 5G networks will involve slices whose VNFs will be spread across multiple technological (and administrative) domains; this immensely increases the optimization complexity due to the combinatorial nature of placing multiple (correlated) VNFs, for multiple slices, among multiple computation nodes. It also requires suitable modelling of end-to-end KPIs. Moreover, the majority of the parameters that affect the performance of each network component (and thus hosted VNFs) are often unknown a priori, dynamically changing, and sometimes even non-stationary, rendering traditional static and centralized optimization methods (whether discrete, continuous, or stochastic) problematic, if not altogether inapplicable. This is why multi-agent Deep Reinforcement Learning (DRL) is employed in Chapters 4 and 5 to tackle these problems.

6G networks are expected to intelligently support a massive number of simultaneous and heterogeneous slices. Consequently, the challenges of scalability and sustainability will affect the deployment of artificial intelligence (AI)-driven zero-touch management and orchestration (MANO) of end-to-end (E2E) slices. In this respect, ETSI has standardized the zero-touch network and service management (ZSM) framework, where a reference architecture and AI-based closed-loop management automation have been proposed [15]. However, the traditional centralized approach for monitoring, analyzing, and controlling the underlying raw data will be problematic, because it suffers from significant overhead, delay, and a single point of failure. On the other hand, decentralized approaches ensure scalability, low data exchange and, therefore, more security. In this view, distributed artificial intelligence (AI) approaches, particularly Federated Learning (FL) techniques can play a vital role in monitoring scattered data across the network while reducing the computational costs and enabling fast local analysis and decision. Nonetheless, both the convergence delay and computation cost often limit FL capability under non-IID real network data. These aspects are going to be considered in the FL approach proposed in Chapter 3.

Furthermore, in real deployment, both the operator and the slice tenant need to understand the behavior of the FL model, in order to trust AI's decisions. Thus, Explainable Artificial Intelligence (XAI) empowered Federated Learning (FL) is getting a lot of attention due to the end-user trust and secured operation. This approach, which can build an advanced AI-based trust model, ensure hassle-free processes, and improve security to the 6G heterogeneous networks, will be also examined in Chapter 3.

3 Transparent Zero-touch Resource Allocation solution for 6G NS

3.1 State of the Art

3.1.1 ZSM & ML in 6G

As stated in the Introduction, to automatically orchestrate and manage network slices, or more specifically network resources, across different domains, along with ensuring the end-user's QoE and QoS, a comprehensive scope of both Zero-touch Network and Service Management (ZSM) and Network Slicing (NS) techniques are currently being studied [16], [17]. Machine Learning (ML) is one of the key enabling technologies that many ZSM frameworks are adopting to bring intelligent decision-making to the network management system and improve the overall network performance. Specially designed and trained ML models can effectively manage and control physical and virtual network resources to mitigate slice Service Level Agreements (SLA) penalties or runtime costs. However, it is very challenging to design such purpose-specific ML.

The work of [18] proposes a cognitive management architecture with ML techniques following the Monitor, Analyze, Plan and Execute over a shared Knowledge (MAPE-K) control loop for real-time accurate bandwidth prediction for mobile users. On the other hand, [19] demonstrates and proposes an ML model to empower self-organizing networks (SONs) for traffic management after clustering and forecasting cellular traffic. But all these above-mentioned works leave a vital gap in systematizing, organizing, and automating all the necessary steps and actions for building efficient ML models in the ZSM framework, although they have an important contribution towards ML-based ZSM in 5G. In this aspect, the authors in [18] proposed a novel unified methodology approach to Cognitive Network & Slice Management in virtualized multi-tenant 5G networks with the application of ML, and this methodology can handle various runtime costs such as unnecessary slice resource overprovisioning and the lack of desirable overprovisioning. Additionally, various works [16], [20], [21], [22] have studied VNF placement, service monitoring, NF/VNF profiling etc., by considering NFV MANO systems to achieve the ZSM goals. Both ZSM and NS will play an essential role in fulfilling the requirements of, not only 5G, but also B5G and 6G. Still, now, there is not enough work considering this fact. We have a plan to work on this site.

3.1.2 Federated learning in ZSM

A fast growing area of machine Learning (ML) that has recently attracted a lot of attention from the research community is Federated learning (FL). The reason is that traditional ML schemes are cloud-centric and require the data to be sent and processed in the central server, which becomes impractical nowadays if the amounts of data traffic from heterogeneous services are massive since it causes high transmission delay and hinders user privacy also. Such schemes are also not suitable for the network slicing case because network slices are isolated, making it challenging to collect data and build centralized machine learning models that identify the performances of network slices. Consequently, it is crucial time to adopt a decentralized learning approach to handle efficiently distributed network slices.

However, to achieve the vision of Zero Touch Management (ZSM) of network slices in 5G, FL can consider analyzing the network slice performances and auto-build the decision mechanisms to react accordingly as mentioned in [23]. The work of [24] aims to design centralized and federated deep learning techniques for predictive horizontal and vertical autoscaling with QoS- prioritized and cost-prioritized objectives in multi-domain networks. But FL performs poorly due to the non i.i.d. data samples, which need to be considered for the next-generation network.

It is expected that 5G and 6G networks will accommodate an enormous number of data traffic by 2030. Thus, an automatic data processing framework needs to be developed, to allow edge learning. Furthermore, FL can meet 6G requirements [25] such as massive ultra-reliable low latency communications (mURLLC), scalable architecture, human-centric services etc. FL can also handle the resource allocation, signal detection, and user behavior prediction problems in the upcoming networks [25]. Hence, to meet and solve the challenges of developing 6G networks with their unique features and associate requirements, FL with ZSM and NS can be studied.

3.1.3 Explainable AI in 6G

Another open challenge with AI is the lack of transparency, interpretability, and trust compared to the other simple and self-explaining traditional models. The reason for this, the AI model is a black box that is not human-understandable. So, from the real-time deployment point of view, it is a barrier to further development. Even if any AI or deep learning model can specify complex model patterns or perform better, such models are not preferable for any critical decision-making services [26].

Besides, the critical characteristic of 6G is that it is "human-centric" [27] rather than "machine-centric." It signifies that all the corresponding "smart things" in the 6G network will function intelligently for humans as a colossus but as a smart black box. In this respect, Explainable AI (XAI) provides methods and techniques to properly explain the AI system and its decisions, which helps gain the trust of the human in the loop.

XAI is the key, especially for 6G stakeholders, such as service providers, end users, and permitted auditors. Many research works exist on XAI and B5G/6G separately but compared to that. Compared to that, only some research works consider and explore XAI's potential for implementing AI-enabled human-centric B5G/6G networks. Also, full automation of ZSM based framework depends on the interpretability and transparency of the AI/ML models to enhance trust and transparency for people to use 6G networks [28]. Some research works of XAI [29], [30], [31] demonstrate the importance of explainability for the management and orchestration of various services in the beyond 5G networks. Evaluating the performance of XAI models, the paper [32] introduces some basic metrics for continuing the research on XAI for any field.

3.2 Explainable FL for Trustworthy Slice Resource Allocation

In this Section, we will present an Explainable Federated learning approach to the FL optimization task for resource allocation for 6G network slicing with non-IID datasets at the RAN-Edge domain.

The main contributions of this paperwork are:

- To deal with the FL resource provisioning task at the local analytic engines (AEs), we formulate the corresponding SLA-constrained optimization problem under the proxy-Lagrangian framework and solve it via a non-zero sum two-player game strategy.
- To ensure trustability under massive slicing, we mainly focus on explainable AI (XAI) for transparent zero-touch service management (ZSM) of 6G network slices.

3.2.1 Proposed Network Model

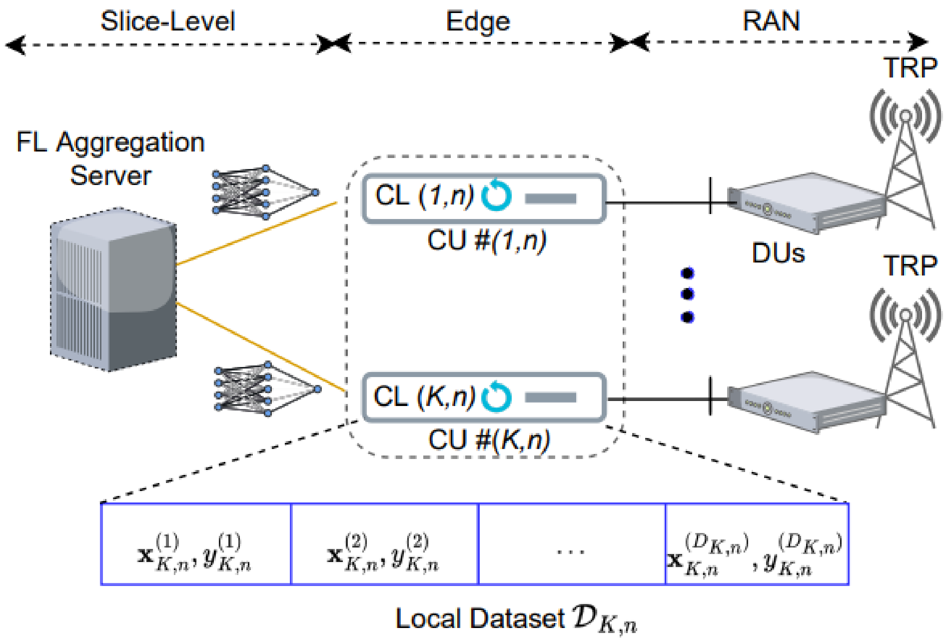

Figure 3-1: Decentralized closed loops (CLs) architecture

We propose an explainability-aware federated learning for slice-level resource allocation in 6G and characterize the performance thereof in terms of both SLA and XAI metrics. The adopted network architecture is depicted in Figure 3-1 which shows a 6G Edge-RAN where distributed analytic engines (AEs) are placed along with the corresponding monitoring systems (MSs). Its topology follows a per-slice central unit (CU)/distributed unit (DU) functional split, wherein each transmission/reception point (TRP) is co-located with its DUs. Basically, each CU consists of a monitoring system (MS) and an AI-enabled slice resource allocation function called analytic engine (AE) in a closed loop way.

Each CU k $(k = 1, \dots\dots, K)$ runs as a virtual network function (VNF) on top of a commodity hardware, and performs slice level RAN key performance indicators data collection to build its local datasets for slice n $(n = 1, \dots\dots, n)$ i.e., $D_{k,n} = \{x^{(i)}_{k,n}, y^{(i)}_{k,n}\}_{i=1}^{D_{k,n}}$, where $x^{(i)}_{k,n}$ stands for the input features vector while $y^{(i)}_{k,n}$ represents the corresponding output.

Table 1: Dataset Features and Output

Feature	**Description**
OTT Traffics per TRP	Apple, Facebook, Facebook Messages, Facebook Video, Instagram, Netflix
CQI	Channel quality indicator reflecting the average quality of the radio link of the TRP.
MIMO Full-Rank	Usage MIMO full-rank spatial multiplexing in (%)
Output	**Description**
CPU Load	CPU resource consumption (%)

3.2.2 Network Configuration

We consider below three primary slices to analyse the proposed Explainable FL policy, defined as follows:

- eMBB: Netflix, Youtube and Facebook Video,
- Social Media: Facebook, WhatsApp and Instagram,
- Browsing: Apple, HTTP and QUIC

Here, the traffic associated with each mentioned slice is the sum of the underlying OTTs' traffics that collects from the hourly traffics of the slices for five days, and the overall summary of those datasets are presented in Table 1.

3.2.3 SLA Violation-Aware Federated Resource allocation

Here, we will give brief overview of FL optimization for resource allocation framework which is going to be modified in time according to our needs.

As we know, an SLA is established between slice n tenant and the infrastructure provider so that any assigned resource to the tenant should not exceed a range $[\alpha_n, \beta_n]$ with a probability higher than an agreed threshold γ_n .This corresponds to solving a statistically constrained local resource allocation problem with both empirical cumulative density function (ECDF) and complementary ECDF (ECCDF) constraints at FL round t $(t = 0, \dots \dots \dots \dots, T-1)$, i.e.

$$\min_{W_{k,n}^{(i)}} \frac{1}{D_{k,n}} \sum_{i=1}^{D_{k,n}} l\left(y_{k,n}^{(i)}, \hat{y}_{k,n}^{(i)}\left(W_{k,n}^{(i)}, x_{k,n}\right)\right),$$

$$\text{s.t. } F_{x_{k,n}} \sim D_{k,n}(\alpha_n) = \frac{1}{D_{k,n}} \mathbb{1}\left(\hat{y}_{k,n}^{(i)} < \alpha_n\right) \leq y_n,$$

$$\hat{F}_{x_{k,n}} \sim D_{k,n}(\beta_n) = \frac{1}{D_{k,n}} \mathbb{1}\left(\hat{y}_{k,n}^{(i)} > \beta_n\right) \leq y_n$$

which is solved by invoking the so-called *proxy Lagrangian* framework [33]. This consists of first on considering two Lagrangians as follows:

$$\mathcal{L}_{W_{k,n}^{(i)}} = \frac{1}{D_{k,n}} \sum_{i=1}^{D_{k,n}} l\left(y_{k,n}^{(i)}, \hat{y}_{k,n}^{(i)}\left(W_{k,n}^{(i)}, x_{k,n}\right)\right) + \lambda_1 \psi_1 \left(W_{k,n}^{(t)}\right) + \lambda_2 \psi_2 \left(W_{k,n}^{(t)}\right),$$

$$\mathcal{L}_{\lambda} = \lambda_1 \varphi_1 \left(W_{k,n}^{(t)}\right) + \lambda_2 \varphi_2 \left(W_{k,n}^{(t)}\right),$$

where $\varphi_{1,2}$ and $\psi_{1,2}$ represent the original constraints and their smooth surrogates, respectively. Specifically, the indicator terms are replaced with Logistic functions. This optimization task turns out to be a non-zero-sum two-player game in which the $W_{k,n}^{(t)}$-player aims at minimizing $\mathcal{L}_{W_{k,n}^{(i)}}$, while the λ-player wishes to maximize $\mathcal{L}_{\lambda}$ [Lemma 8] [33]. While optimizing the first Lagrangian w.r.t. $W_{k,n}$ requires differentiating the constraint functions $\psi_1 (W_{k,n}^{(t)})$ and $\psi_2 (W_{k,n}^{(t)})$, to differentiate the second Lagrangian w.r.t. λ we only need to evaluate $\phi_1 \left(W_{k,n}^{(t)}\right)$ and $\phi_2 \left(W_{k,n}^{(t)}\right)$. Hence, a surrogate is only necessary for the $W_{k,n}$-player; the λ-player can continue using the original constraint functions. Via Lagrange multipliers, the λ-player chooses how much to weigh the proxy constraint functions but does so in such a way as to satisfy the original constraints and ends up reaching a nearly optimal nearly feasible solution [33].

3.3 Preliminary Works

In this section, we provide our preliminary work and associated results. We consider our previous work as a baseline for accomplishing our target research work.

Based on our proposed SLA-aware FL optimization for the resource allocation approach mentioned above, we have designed a novel SLA-driven stochastic FL policy for selecting a subset of AEs participating in the FL task at each round. It enhances the convergence time while maintaining the exact computation cost no matter how many AEs increase over the network and also ensure scalability under massive slicing.

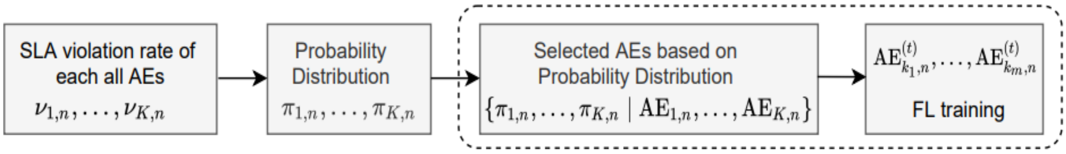

Figure 3-2: policy for AE selection Proposed

3.3.1 SLA-Driven Stochastic Federated Learning Policy

We aim to select only a subset of active AEs in each FL round to optimize the federated learning computation time and the underlying resource consumption. In this regard, we propose an SLA-driven stochastic AE selection policy. Upon the completion of the training at round t, each $AE(k,n)$ evaluates the generalization of its FL model using a typical test dataset $\widetilde{\mathcal{D}}$ of size $\widetilde{D}$---that is common to all monitoring systems of slice n--- and calculates the so-called SLA violation rate as,

$$v_{k,n} = \frac{1}{\widetilde{D_n}} \sum_{i=1}^{\widetilde{D_n}} \mathbb{1}\left[\left(\tilde{y}_{k,n}^{(i)} < \alpha_n\right) \bigcup \left(\tilde{y}_{k,n}^{(i)} > \beta_n\right)\right]$$

Next, at each FL round t, as presented in Figure 3-2, all the participating AEs send their SLA violation rates to the server which generates a probability distribution using $softmin$ function as,

$$\pi_{k,n} = \frac{\exp\{-v_{k,n}\}}{\sum_{\rho=1}^{K} \exp\{-v_{\rho,n}\}} ,$$

Wherein AEs with low SLA violation are given a high probability of FL participation to drive the model convergence, but also AEs with high SLA violation might take part in the FL training with a low probability to guarantee the generalization that could stem from their datasets. Based on the probability distribution, only a subset of $m < K$ AEs is drawn at each FL round to

$$AE_{k1,n}^{(t)}, \ldots\ldots\ldots\ldots\ldots, AE_{km,n}^{(t)} \sim \{\pi_{1,n}, \ldots\ldots\ldots, \pi_{K,n} \mid AE_{1,n} \ldots\ldots\ldots., AE_{K,n}\}$$

Thus, the AEs would have stochastically participated in the FL task while avoiding the concurrent training at each round. And the model averaging at round t is performed as,

$$W_n^{(t+1)} = \sum_{k \in \{k_1, \ldots, k_m\}} \frac{D_{k,n}}{D_n} W_{k,n}^{(t)}$$

Where D_n is the sum of datasets sizes over all slice $n's$ AEs.

3.3.2 Parameter Settings and Baseline

We consider three primary slices to analyze the proposed FL policy, mentioned in detail at the section 2.4.2. We use vectors α, β for the resource bounds, and β for the thresholds corresponding to the different slices for a particular resource. The parameters settings are mentioned in Table 2.

Table 2: Settings

Parameter	Description	Value
N	#Slices	3
K	#AE	100
m	#Selected AEs	50
$D_{k,n}$	#Local dataset size	1000 samples
T	#Max FL rounds	30
L	#Local epochs	160
R_λ	#Lagrange multiplier radius	Constrained: 10^{-5}
η_λ	#Learning rate	0.02

3.3.3 Simulation Results

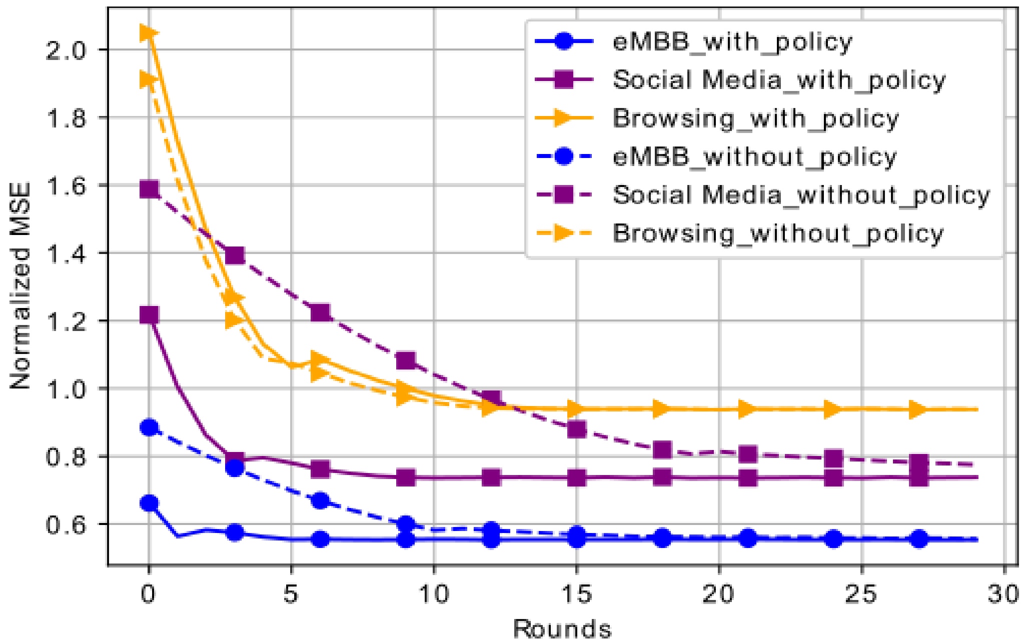

Figure 3-3: FL training MSE loss vs. number of FL rounds with and without proposed policy for $m = 50$ and $K = 100$. SLA bound, with α = [0, 0, 0], β = [4, 7, 10] and γ=[0.01, 0.01, 0.01] in constrained case

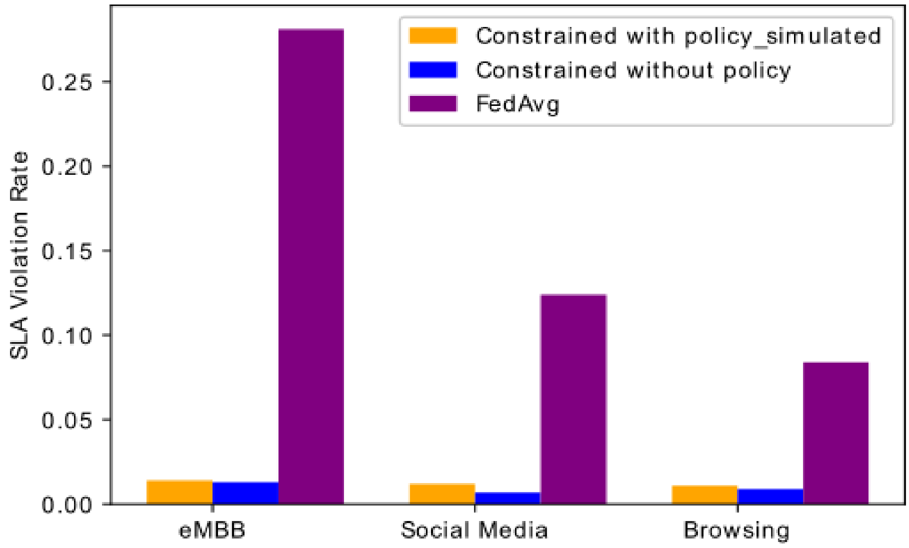

Figure 3-4: SLA violation rate with α = [0, 0, 0], β = [4, 7 , 10] % and γ = [0.01,0.01,0.01]

In the simulated scenario, resources at CU-level are dynamically allocated to slices according to their traffic patterns and radio conditions (average CQI, MIMO full-rank usage).

Observing Figure 3-3, we can conclude that for all slices, policy-based FL converges faster than without policy, since the AEs with lower violation rate have more chances to participate in the training. Noted that even if, we increase the total number of AEs as the number of slices and CUs increases in the physical network, which ensures scalability. Supporting results will be added in our future deliverables.

Finally, to investigate the trade-off, the CPU SLA violation rates of the slices are shown in Figure 3-4, where it is observed that the policy-driven constrained FL presents significantly lower violations compared to FedAvg while preserving the performance of constrained FL without policy (i.e., around 1 %).

Note that we choose FedAvg as our baseline, where resources are optimized in the unconstrained scenario.

3.4 Conclusion

Sixth-generation (6G)-enabled massive network slicing is a strong enabler for the expected pervasive digitalization of the vertical market. In such a context, artificial intelligence (AI)-driven zero-touch network automation should present a high degree of scalability and sustainability, especially when deployed in live production networks wherein the collected monitoring datasets at different points are non-independent and identically distributed (non-IID). In this contribution, we present a service-level agreement (SLA)-driven stochastic policy to guarantee a scalable and fast operation of constrained federated learning (FL)-based analytic engines that perform statistical slice-level resource provisioning at RAN-Edge in a non-IID setup. The simulated scenario demonstrates the superiority of the solution in reducing SLA violation, convergence time and computation cost compared to different FL baselines, showcasing thereby a higher scalability.

However, to achieve our actual objectives for developing transparent FL resource allocation for 6G network slicing, we will consider our preliminary work as a primary framework and work based on this to fulfil our goals.

3.5 Future Research Direction

Here, we present a 6G RAN-edge network architecture, as well as preliminarily works and results, based on which we will proceed to implement our proposed idea. Our main aim is to include the XAI approach in our current implemented framework. Moreover, to gain more trust and reliability in our proposed solution, we may do a comparative analysis of existing XAI and show some related results. This approach is helpful and trustable for decision-making cases of any kind of critical service in the telecommunication field. Furthermore, it will be beneficial for any telecom operator or service provider to broaden their deployment and services, which is the future goal of the 6G network.

4 Dynamic end-to-end slice embedding in beyond 5G networks

In this Chapter we will focus on the problem of dynamic slice embedding, assuming that resource allocation per node is performed by a given scheduling algorithm (e.g., proportionally fair sharing), whose impact is captured by our model. In Section 4.1, we provide a system model that supports a multi-domain setup and diverse end-to-end SLAs. The proposed model is generic, so it can potentially incorporate flexible functional split and Integrated Access and Backhaul, by considering suitable physical layer VNFs at the Radio Access Network (RAN). In Section 4.2, we first discuss how such problems can be optimally solved, in theory, with tabular Reinforcement Learning (RL) algorithms, and more specifically with Q-Learning (QL) [34], even under, a priori, unknown demand dynamics for each slice. While such methods are inapplicable in realistic problem sizes, due to the high state and action complexity, they are useful in providing a baseline for approximate ones, in small enough setups, as well in grounding more advanced algorithms with good theoretical properties [35]. Building on top of QL, we demonstrate how the Deep-Q-Network (DQN) algorithm [36] can be applied in our problem to reduce the prohibitive state complexity, by approximating the Q-function with a simple Deep Neural Network (DNN). Then, to tackle also the high action space complexity faced by centralized RL algorithms in the examined problem, we propose a multi-agent scheme based on multiple DQN agents.

4.1 System model

In this section, we present our system model. The main components are the physical network and the virtual networks on top of it. We also discuss the modelling of end-to-end KPIs, the control we have over the system and the cost associated with a control decision. Since there is plenty of notation to keep track of, we will use the example shown in Figure 4-1 to explain the various quantities involved, throughout the problem setup.

4.1.1 Physical Network and slices

As is common in related literature, we represent both the underlying physical network (PN), and the VNF chains (“slices”) to be deployed on top of it, by graphs (e.g. [13], [14]).

Physical Network:

It consists of physical nodes (servers or routers) interconnected by physical links, and is represented by a weighted undirected graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is the set of nodes and $\boldsymbol{E}$ is the set of links. It possibly comprises multiple (technological or administrative) domains.

Each physical node and link is characterized by its capacity:

- Node capacity $\boldsymbol{b_v}$: It is the capacity of (physical) node $\boldsymbol{v}$ to host VNFs of that domain (it could also be 0 for some nodes, e.g. for routers). This capacity could be resource blocks, CPU cores, containers, etc., depending on the domain.
- Link capacity $\boldsymbol{b_e}$: It is the capacity of edge (or path) $\boldsymbol{e}$ between two PN nodes (e.g., bandwidth).

In the example of Figure 4-1, there are 3 domains, Cloud RAN (CRAN), Multi-access Edge Computing (MEC), and Core Network (CN), and each of them comprises a number of nodes where VNFs could be executed (CRAN: 3 servers, MEC: 2 servers, CN: 3 servers) and associated capacities $\boldsymbol{b_v}$. The servers are connected through physical links (of capacity $\boldsymbol{b_e}$), or paths comprising multiple links and nodes of the network.

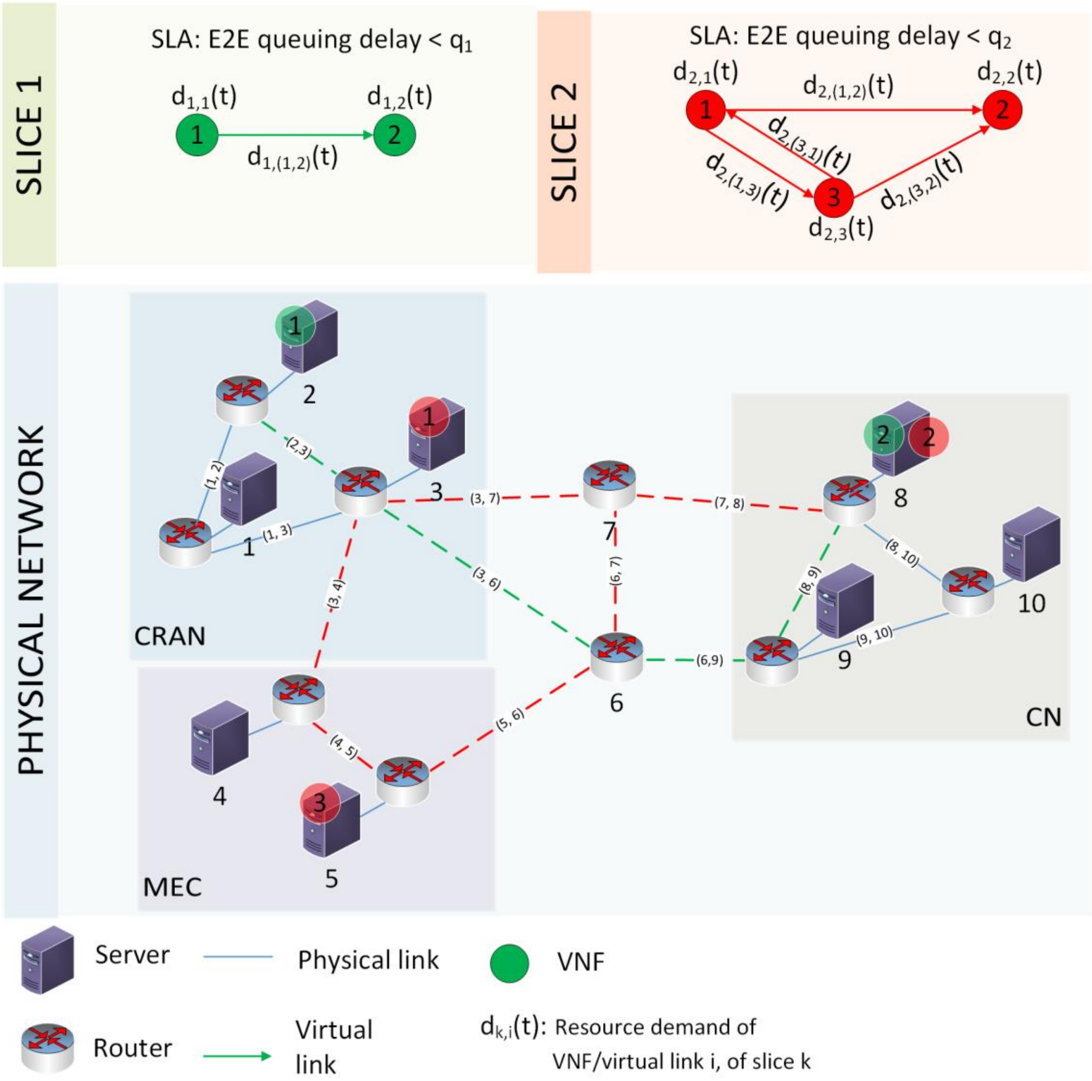

Figure 4-1: Graphical illustration of the system model. The embedding of 2 slices onto the network is depicted.

Network Slices:

We assume that a set of slices K (K in total) must be hosted on top of the PN. We slightly generalize the common view of a slice, in related 5G literature, and depict a slice k as follows. Each Slice ("VNF chain") is represented by a directed graph $\boldsymbol{H_k} = (\boldsymbol{N_k}, \boldsymbol{L_k})$ of VNFs (set $\boldsymbol{N_k}$) that model the various processing tasks required by the flows of this slice, and directed (virtual) links (set $\boldsymbol{L_k}$) indicating the order of how these tasks are applied.

A simple example is Slice1 of Figure 4-1, which consists of 2 VNFs and the flows must first traverse VNF1 to receive some initial processing, and then VNF2 for the remaining processing required in order to provide the corresponding service to the end-users. However, our model can be fairly generic, allowing for both loops (e.g., flows passing by the same VNF multiple times), as well as probabilistic routing of flows (e.g., to capture the scenarios where not all flows of a slice require all VNFs in the same order). An example of such a chain is Slice2 of Figure 4-1, where a percentage of flows from VNF1 proceed to VNF2 directly, while the rest must pass through VNF3, possibly going back to VNF1 as well.

Each VNF and Virtual Link (VL) requires some amount of resources from the Physical Network, and these resource demands are dynamically changing according to the user generated traffic:

- VNF demands $d_{k,n}$: Each VNF n of slice k is associated with a resource demand $d_{k,n}(t)$ that is a function of time t (time is slotted), and will be imposed on the PN node where the VNF is executed.
- VL demands $d_{k,l}$: Similarly, each (virtual) link l of slice k is associated with a resource demand $d_{k,l}(t)$. Note that this load will be added to all physical links along the PN path between the execution nodes hosting the two VNFs connected by virtual link l.

Remark: These demands are often unknown, stochastic, non-stationary (correlation between VNFs of the same slice is also common); the main reason why we require a learning-based optimization algorithm to tackle this problem.}

The resource demands of all slices (VNFs and VLs) hosted by the network at time t are organized in a vector denoted by D_t . For example the demand vector in Figure 4-1 is:

$$D_t = (d_{1,1}, d_{1,2}, d_{1,(1,2)}, d_{2,1}, d_{2,2}, d_{2,3}, d_{2,(1,2)}, d_{2,(1,3)}, d_{2,(3,1)}, d_{2,(3,2)})$$

Notice that D_t is combinatorial, so the number of possible values $|D_t|$ it can take increases exponentially with the number of hosted slices.

Service Level Agreements q_k: Each slice k comes with some slice-specific requirement q_k, which defines a maximum (or minimum) value for an end-to-end KPI metric. For example, in Figure 4-1, the KPI metric is the end-to-end queuing delay.

4.1.2 End-to-end KPI modelling

In order to calculate the SLA violation cost we need to model different end-to-end KPI metrics that assess the performance of a slice k as a function of the configuration and the demand. We give two examples of end-to-end KPIs:

- Queuing delay: Assuming an M/G/1/Processor Sharing (PS) type of scheduler (an M/G/1/PS processor with classes has been shown to be a good approximation for many proportionally fair wireless schedulers [37]), we can calculate the mean delay experienced by a VNF/VL on the host node/link, using a standard closed form formula for M/G/1PS queues [38]. This is a function of the host's capacity and the aggregate resource demands of the hosted VNFs/VLs. Then, in the case of a simple chain slice, the end-to-end queuing delay is the sum of the delays on the traversed nodes and links. As an example, in Figure 4-1, the delay experienced by Slice 1 is the sum of the delays in node 2, link (2,3), link (3,6), link (6,9), link (8,9), and node 8. In the case of a more complex slice, a Jackson network type of analysis could be applied to calculate the delay [38]. Note that this modelling captures the resource allocator scheduler impact.

- Underprovision: A penalty is paid when the aggregate demand of the VNFs/VLs embedded on a physical node/link exceeds its nominal capacity [12]. It can be used as a proxy for slice performance on a physical node/link when details about queuing delay are not available.

Note that within the proposed framework any other end-to-end KPI can be incorporated (e.g. to model bandwidth).

4.1.3 Control decisions and associated cost

Control decisions:

The control we have over the system is the embedding of slices on top of the Physical Network (placement of VNFs to physical nodes and VLs to physical links). For the moment we assume that the routing path between any 2 servers of the network is given by a routing algorithm, so the direct control we have is only the placement of VNFs to servers (we call this the configuration of the system). However, we can indirectly control the routing and avoid link congestion by placing VNFs to suitable nodes.

Configuration $\boldsymbol{C_t}$: The configuration of the system at time t is a vector C_t that represents the mapping of VNFs to nodes for all slices. Each element $\boldsymbol{c_{k,v}} \in \boldsymbol{C_t}$, indicates the host node of VNF v of slice k. As an example, in Figure 4-1, the first VNF of Slice1 is hosted by node 2, so $\boldsymbol{c_{1,1} = 2}$, while the configuration is $\boldsymbol{C_t} = (\boldsymbol{c_{1,1}}, \boldsymbol{c_{1,2}}, \boldsymbol{c_{2,1}}, \boldsymbol{c_{2,2}}, \boldsymbol{c_{2,3}})$.

Notice that, similarly to the demand vector, the configuration vector is also combinatorial. This is one of the major challenges in the slice embedding problem, leading to high optimization complexity for realistically sized scenarios.

Operational Costs of Physical Network Infrastructure:

Given the (usually unknown) demands D_t and the configuration C_t at time t, we assume that the system suffers an instantaneous cost related to both the network performance (i.e. direct cost to the operator) and the slice performance (e.g., indirect cost related to SLA violations). We choose to consider the following cost quantities in this work (other components can be straightforwardly added to the framework):

- *Type 1 cost: Node utilization (g_1).* It is equal to the number of active servers (those that host at least one VNF) and accounts for energy consumption expenses. The idle servers can be set to sleep mode and save energy, while minimizing this cost also facilitates the admission of new slices by maximizing the free space of resources [39].
- *Type 2 cost: SLA violation (g_2).* When the maximum value q_k defined by the SLA of a slice is exceeded, a penalty is paid to the slice tenant. This penalty may take any form that is appropriate to model the impact of violating the corresponding KPI (e.g., linear, quadratic, etc.). So, the SLA violation cost is equal to the sum of these penalties over all the hosted slices.
- *Type 3 cost: Reconfiguration (g_3).* The cost for migrating VNFs from their host servers to other servers in the PN. It relates to the overhead generated for reassigning all VNFs and the delays incurred by this action, which may lead to penalties for SLA violations [11]. The simplest way to define it is to consider this cost equal to the number of migrating VNFs (all migrations have equal cost).

The total cost g of a control decision is simply the weighted sum of the three different types of costs described above:

$$g = w_1 g_1 + w_2 g_2 + w_3 g_3$$

, where the weights w_1, w_2, w_3 are scalar values to be defined by the network operator.

4.2 Reinforcement Learning algorithms

Our goal in the dynamic slice embedding problem is to decide the configuration C_t at every time t, (i) towards optimizing the total system cost (consisting of the various cost components presented in the previous Section), while (ii) not knowing a priori how demands D_t evolve over time. This is an

online learning and control problem, for which Reinforcement Learning (RL) schemes are a natural candidate. Below, we first define the main components of any RL scheme, namely its state and action space, and the rewards (often referred to as "costs'" in minimization problems [35]). Then, after discussing why this problem is very difficult to solve, we present some of the RL algorithms that can be utilized to maximize these rewards (or minimize the costs) over a discounted infinite time horizon.

4.2.1 RL problem formulation

State (S_t):

The state of the system at any time-slot t consists of (i) the slices' configuration $\boldsymbol{C_t}$ on top of the physical network, and (ii) the currently observed resource demand D_t. Consequently:

$$S_t = (C_t, D_t)$$

RL agent action (A_t):

The action A_t that our RL agent needs to take (the control decison) is a good (re-)configuration C_{t+1} (without knowing the future demand D_{t+1}.

Reward (r_t):

Given some observed state S_t, an action A_t taken by the RL agent, and the next state S_{t+1} revealed by the environment, the reward obtained at time $t+1$ is equal to the negative of the cost g defined in the previous section:

$$r_{t+1} = -(w_1 \cdot g_1(A_t) + w_2 \cdot g_2(S_{t+1}) + w_3 \cdot g_3(S_t, A_{t,}))$$

The minus sign is introduced because the RL agents typically try to maximize the expected accumulated rewards over a discounted infinite horizon.

4.2.2 The curse of dimensionality in the slice embedding problem

In the Section where we described the system model, we highlighted the combinatorial nature of the demand and configuration vectors. Consequently, the state and action spaces are also combinatorial and grow very fast with the size of the system. Even for relatively small scenarios the number of states and actions can be billions!

Toy example:

As an example, consider the (oversimplified) case of a single domain network, which hosts 10 slices, and each slice consists of only one VNF (assume no VLs); we also have 5 servers, while the resource demand of each VNF can take only one out of 2 distinct values (high or low level). Even in this toy scenario, the number of possible states is:

$$|S| = 5^{10} \cdot 2^{10} = 10^{10}$$

, where the first term is due to the 5 different servers where each of the 10 VNFs can be placed, and the second term is due to the 2 levels of resource demand that each of the 10 VNFs may impose on the host. Similarly, the number of possible actions is:

$$|A| = 5^{10}$$

It is obvious that the dynamic slice embedding problem suffers from the notorious curse of dimensionality, meaning that the state and action spaces grow very fast with the size of the system, making it very hard to solve for realistically sized scenarios. Consequently, we need an algorithm that can tackle the high dimensionality problem. We will start from one of the most standard RL algorithms (QL), highlight the implications due to the size of state and action spaces, and build on top of that to eventually come up with a proposed solution.

4.2.3 Q-Learning (QL)

The QL agent is equipped with a table containing the Q values of all possible state-action pairs. The $Q(S, A)$ value indicates how good it is to take an action A, while being in a state S. Consequently, a QL agent can behave optimally as soon as it knows all the correct Q values. These values are randomly initialized and the agent converges to the correct ones by interacting with the system during the training phase, where it updates the value of the visited state-action pair at each time-slot according to:

$$(1-\alpha)Q(S_t, A_t) + \alpha \cdot r_{t+1} + \gamma \cdot \max_{A \in \mathcal{A}}(Q(S_{t+1}, :))$$

In Figure 4-2 there is a schematic representation of the interaction between the agent and the system.

Such schemes provably converge to the global optimal solution of the problem, without any a priori knowledge of the resource demands D_t [35]. The important downside of such "tabular" algorithms is that every possible state-action pair must be encountered on training enough times each, in order to ensure convergence to a good estimate of the respective Q value. Considering the exploding number of states and actions in our problem, this leads to a very slow convergence. For the same reason, the size of the Q-table may lead to prohibitively large memory requirements. Finally, due to the exploding number of actions, the argmax operation over all actions calculated by the agent at each time-step is very expensive. Consequently, QL can be applied only in extremely small toy scenarios to give the optimal policy, but it is not a scalable algorithm that can be generally applied for dynamic slice embedding.

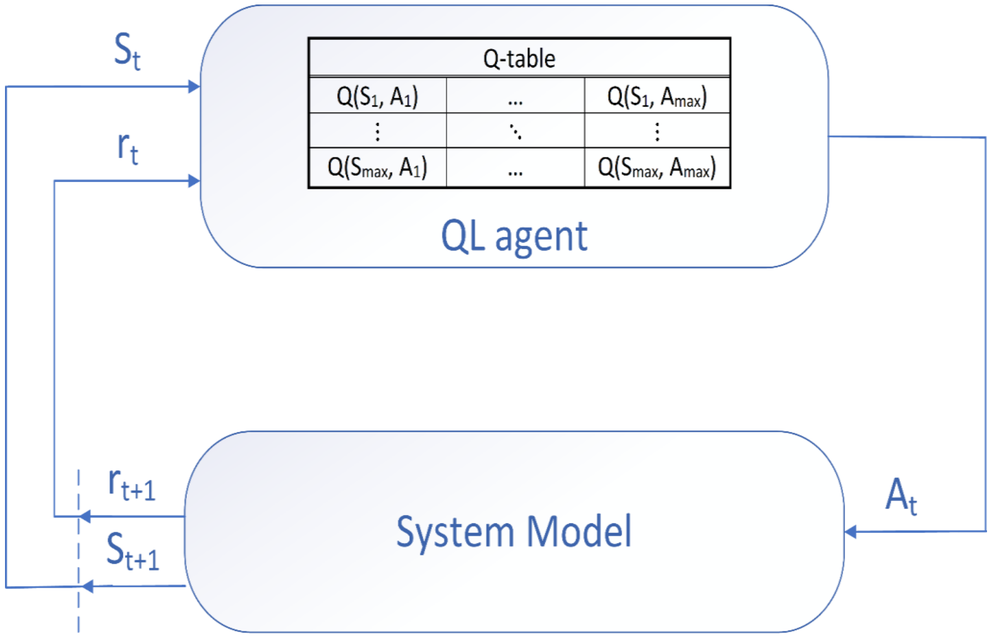

Figure 4-2. Schematic representation of the interaction between the Q-learning agent and the system

4.2.4 Deep Q-Network (DQN)

A first step towards tackling the scalability issues of QL is to approximate the Q function by replacing the Q-table with a parametrized function (like a Deep Neural Network (DNN)). A well- known and successful approach is the DQN introduced in [36]. DQN is also equipped with some additional components, like the target network and experience replay memory, which are very important for the algorithm's stability during training and the sample efficiency (fewer samples are needed for training). As depicted in Figure 4-3, the DNN takes as an input the state of the system S_t, and outputs the Q-values of all possible configurations. The benefit of using a DNN is that the agent needs to learn fewer parameters compared to the Q-table (faster convergence, lower memory requirements), and that the update of the DNN parameters (with stochastic gradient descent) at each timeslot affects the Q-values of multiple state-action pairs (faster convergence). The drawback is that the DNN has as many output neurons as the number of possible actions/configurations (combinatorial). Consequently, the number of parameters will still eventually explode, and the argmax operation at each timestep is still very expensive. However, with DQN we have managed to tackle the problem of the large state space, and now only the action space size limits the scalability of the algorithm

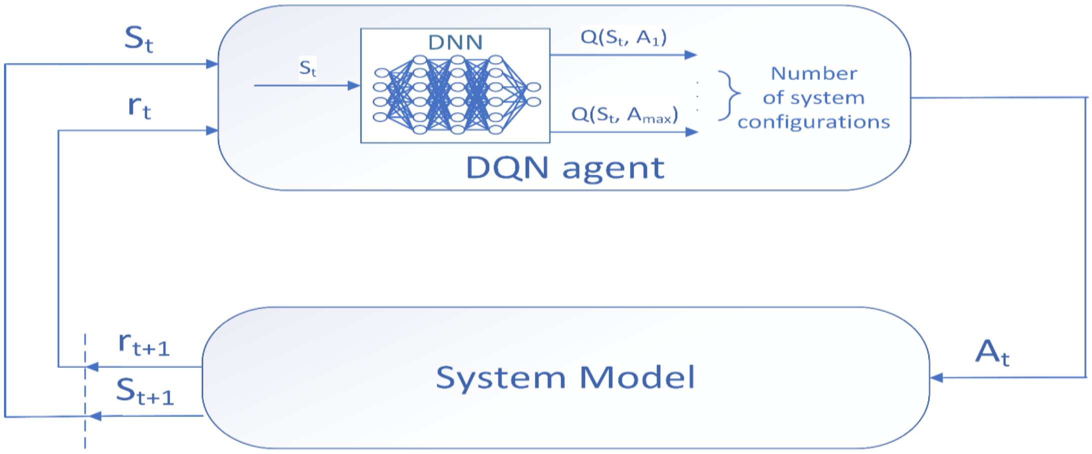

Figure 4-3. Schematic representation of the interaction between the DQN agent and the system model

4.2.5 Multi-agent DQN

A natural approach for problems where the action space can be decomposed, is a multi-agent RL scheme. In our case, A naturally decomposes into action subspaces per VNF (or it could be also per slice). We therefore consider that there is one independent DQN agent per VNF, equipped with its own DNN, experience replay memory, and target network, that decides only for the placement of the specific VNF. As depicted in Figure 4-4, all agents observe the global state S_t and each agent decides for the server where the associated VNF must migrate next. After each agent has taken an action; the global action A_t is formed, a reward and a new state are observed. These are broadcast to all agents' buffers, and then each agent makes a gradient update of its own DNN parameters. The number of output neurons for each agent's DNN is now equal to the number of possible host servers where the associated VNF can be placed (which is not combinatorial, it increases linearly with the number of servers). Consequently, each agent has fewer parameters to learn (faster convergence, lower memory requirements), and we have bypassed the computationally heavy argmax operation over combinatorial actions. Also, note that since all the agents view the same global state, they can operate in parallel at each time-step. So, this multi-agent scheme with independent DQN agents is a scalable solution, suitable for dynamic slice embedding (reconfiguration).

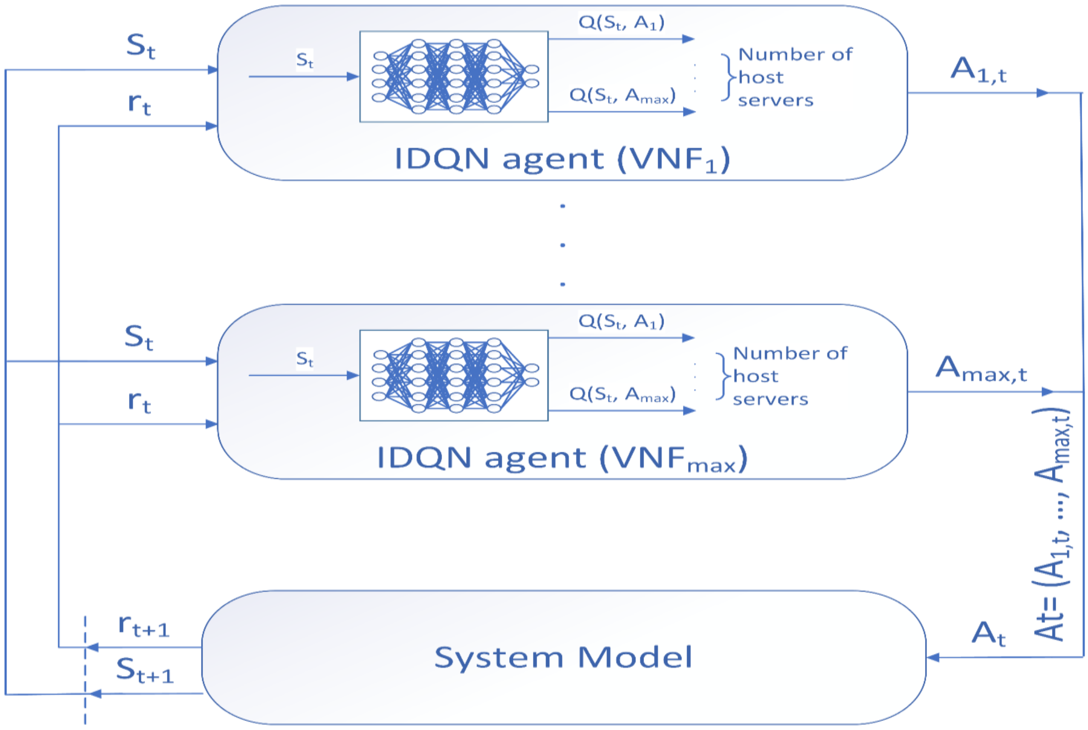

Figure 4-4. Schematic representation of the interaction between the multiple DQN agents and the system model

4.3 Preliminary results

Here we provide some preliminary results indicating the potential of the DQN based approaches. We consider a very small toy scenario in order to be able to apply the tabular Q-learning algorithm, which gives the optimal policy, and use it as a benchmark to assess the quality of policies given by IDQN and DQN. Consequently, the setup is a single domain network, with 2 servers and 4 slices, where each slice consists of a single VNF with a 2-level high-low resource demand. Moreover, the demands are Markovian.

The simulation results can be summarized by Figure 4-5. The plot depicts the average cost as a function of the time-slot, where the cost is averaged over 10 different runs with different initializations of the random seeds (but in the same dataset). Also, high frequency components have been filtered out. All algorithms start with randomly initialized parameters (of the Q-table or the DNN), so in the beginning they demonstrate a higher cost. As the training progresses, the algorithms start learning the system's dynamics and optimizing their policies, so the cost decreases. Towards the end of the training (right side of the plot) all algorithms end-up to policies with similar costs (similar quality). However, QL needs around 70000 timeslots to converge while DQN and IDQN have converged from the first few thousand timeslots. Consequently, from this toy scenario the takeaway is that DQN based algorithms can find good quality solutions and much faster than tabular QL. However, the scenario's action space is still very small to observe any difference between IDQN and DQN.

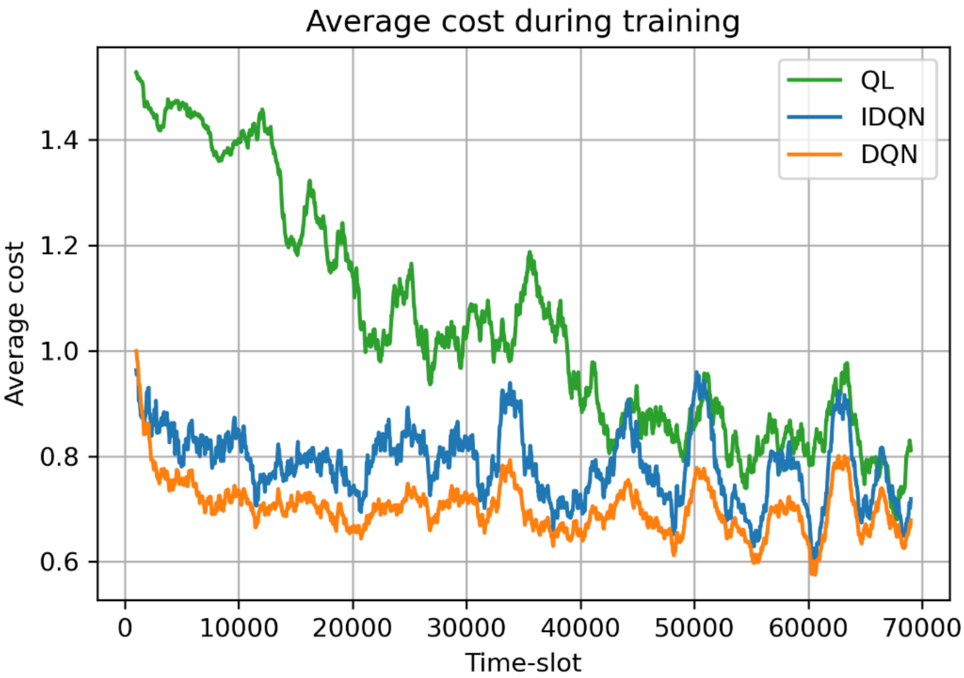

Figure 4-5. Convergence plot

4.4 Conclusion and future work

In this chapter we proposed a flexible model for the dynamic slice embedding problem, suitable for multi-domain setups and diverse end-to-end SLAs. The proposed framework is generic and could possibly accommodate different KPIs, as well as concepts like Functional Split and Integrated Access and Backhaul in the RAN domain. Within this framework we examined the application of Reinforcement Learning algorithms to solve the problem at hand, highlighted the challenges encountered by vanilla tabular RL algorithms, and proposed a multi-agent DQN scheme which provides a more scalable solution.

Future work includes the introduction of a theoretically grounded framework for (multi-agent_ DRL algorithms in the context of network virtualization, and evaluation of performance mainly through simulations with existing real traffic datasets. Part of the evaluation might also take place on real platforms.

5 DRL based slice resource allocation and management for beyond 5G and 6G

5.1 State of the Art

5.1.1 Network slicing

Authors in [40] overview on Machine Learning (ML) and AI based slicing approaches including deep Q learning for a Mixed Integer Linear Programming (MILP) problem for baseband unit (BBU) capacity allocation and Physical Resource Block (PRB) management. The research work discussed, aims to minimize the waste of resources as well as guarantee the service requested by user. It indicates interesting approach for rapid parameterization of slice, its management to filter and analyses the traffic parameters. It compares the results with static slicing baseline model. The work provides intelligent resource management that reduces the ability to reuse a network service for different tenants. Migration of Network Functions (NFs) between access network (AN) and core network (CN) for service support to provide low latency and reduction of congestion between them. The proposed method has been tested on fronthaul simulation based on Open Air Interface (OAI) and FlexRAN Controller which uses USRP B210 boards as radio units and Huawei E3372 as UEs. The algorithms have been tested for 10 MHz FDD uplink and downlink transmission of 500–1200 bytes packet size between 2 user equipment for UDP and ICMP protocol. Some of the main takeaways show that dynamic slicing can provide better resource management optimization, enhance the Quality of Service (QoS) and improve stable core network for 5G. Slice isolation, diversification, deployment and, advance management can lead to exploit slice properties and better virtualized network implementation to achieve the required key performance indicators (KPIs). Whereas, real time dynamic slicing approaches enhance user equipment (UE) acquisition, slice configurations and system runtime optimization, though the intelligent management lack to ensure the multilevel control loop at different time scale that could have been explored and utilized towards further improvements which will be discussed in proposed work.

Network function virtualization (NFV) and software defined network (SDN) based network architecture to support centralized and distributed control for user and control planes has been studied in [41], [42] and [43]. It distinguishes between three distinctive and interworking RAN slicing options suitable for various deployment scenarios. The architecture discussed supports centralized coordination within one or multiple slices. The research is mostly focused on virtualization of network functions as well as definition of RAN slices for 5G networks under RAN sharing and software defined RAN which can be interoperated to understand its working for O-RAN networks. It helps to understand mobile virtual network operator infrastructure which shares virtual network functions (VNFs) between different mobile network operators (MNOs) and infrastructure providers. For the design of RAN Intelligence Controller (RIC) architecture in case of centralized and virtualized resource management, the perspective of slicing functional blocks and its interfaces plays a vital role [44]. The research work in [44] also provides insights about cloud and radio monitoring architecture as well as its approaches. This two-tier architecture with small cells utilizes edge/cloud computing for offloading of several latency-oriented tasks. It also defined main data center at the core part of network for management and orchestration of mobile radio signalling. Advanced approaches with better slice management, orchestration and resource handling can be achieved specifically with the help of AI based techniques implemented as a part of O-RAN modules or a 3rd part application. It is in reference to discussed work as indicated in our proposed system model and problem formulation.

5.1.2 AI/ML for wireless communication

AI based slice management framework for supervised solutions can be achieved by data training. It works well with reinforcement learning (RL) approaches where different forms of interaction with the system that has to be controlled are possible [45]. AI for admission control of slices has trade-off between resource sharing and key performance indicator (KPI) fulfilment needs to be tackled by the use of feed forward neural network. It can also be exploited in a way to allocate and change the allocation for intra as well as inter slice radio resources by keeping in check with KPIs as we proceed in our proposed solution. Upon arrival of a new slice request, the system takes the action (i.e., accepting or rejecting the request) that maximizes long-term revenue; each neural network (NN) is in charge of forecasting the revenue associated with one action. AI for resource orchestration may come across the tradeoff between under provisioning and over dimensioning which can be tackled by convolutional NN (CNN) architecture for time series prediction with a dedicated loss function as it allows exploiting inherent spatial correlations in the traffic generated at different geographical locations. The trade off and its effect can be then utilized as one of the decision-making factors towards long term solution. AI for Slice Scheduling at Radio Access suffers through a key challenge of network slice design of a radio access network (RAN) virtualization mechanism that jointly provides isolation between network slices and adapts the allocation of pooled physical resources to the needs of each virtual RAN. A combination of unsupervised learning (deep auto-encoder) and deep reinforcement learning is a promising solution. Deep deterministic policy gradient (DDPG) algorithm, implemented by actor-critic NN structures, can deal with large and/or continuous action spaces, which are common in resource control problems [46]. The data driven framework can be effectively designed to allocate capacity to individual slices by adopting an original multi timescale forecasting model [11],

- It uses deep Learning architectures and a traditional optimization algorithm
- It anticipates resource assignments that minimize the comprehensive management costs induced by resource overprovisioning, instantiation and reconfiguration, as well as by denied traffic demands
- Isolation of resources across slices inherently increases network capacity requirements, and a dynamic, preemptive and efficient allocation of resources to slices

It can further be improved with adaptation of RL techniques to learn the environment and intelligent agent better in terms of resource utilization, traffic smoothening and load balancing. These approaches are exploited and considered as main basis for our system model. The use of AI can enable zero-touch networks, i.e., fully self-operating communication infrastructures where forecasting holds a fundamental role. To resolve issues such as,

i. unnecessary resource overprovisioning,
ii. non-serviced demands
iii. resource instantiation
iv. resource reconfiguration

Capacity forecasting tries to accommodate the demand and to limit overprovisioning, by reconfiguring resources at every re-orchestration opportunity and it minimizes costs (i) and (ii). Whereas, a long-timescale orchestrator operates over extended intervals that span multiple re-orchestration opportunities. It allocates a dedicated capacity to each slice and also reserves an additional shared capacity accessible by any slice. Each capacity remains constant across the interval reducing (iii). Only the shared capacity is then reallocated at every re-orchestration opportunity by a short-timescale orchestrator, while the configuration of the dedicated capacity is preserved throughout the extended interval, thus reducing cost (iv). Both long- and short- timescale orchestrators help build better distribution of resource among the slice resources and is inspirational to go with multi timescale approach for resource handling [11].

Deep learning models trained offline can be defined to perform end to end dynamic and reliable resource slicing under dataset dependent generalized non-convex service level agreement (SLA) constraints [43]. Authors in [43] discuss DNNs to model and estimate the required resources at each virtual network function (VNF) such as physical resource blocks at a transmission/reception point (TRP), radio resource connected (RRC) user's licenses at a virtual baseband processing unit (vBBU), enhanced radio access bearers (ERAB) and signalling connections at a virtual digital front end (vDFE) and virtual SDN controller (vSDNC). The DNN model shows two SLA based approaches, violation-based SLA and resource bound based SLA. The contribution includes slice scheduler that allows existence of slice with bandwidth based and resource-based reservation simultaneously. [1] discusses a deep neural network architecture, which is trained via a dedicated loss function and returns a cost-aware capacity forecast. This forecast can be directly used by operators to take short- and long-term reallocation decisions. A deep learning architecture is utilized by authors in [45], which exploits space- and time-independent correlations typical of mobile traffic, and computes outputs at a data-center level to jointly solve the problem of capacity in network slicing. It leverages a customized loss function that targets capacity forecast letting the operator tune the balance between over-provisioning and demand violations. Furthermore, [47] provides long-term forecasts over configurable prediction horizons, operating on a per-service basis in accordance with network slicing requirements. DeepCog aims at forecasting the (constant) capacity that should be allocated over a long-term horizon, so as to minimize the monetary cost incurred by the operator.

The overall studied literature shows the gap in exploiting the multi time scale approach towards resource configuration for inter and intra slice management. It can further be explored using DRL methods and managing trade-offs to smooth out the underlying traffic for each slice in aim to increase the slice performance while keeping in check with quality of service. This specific gap is tackled in the system model and research work and we explore different aspects of real and non-real time scales for radio resource allocation. In addition to this the usual problem faced in DRL for constructing limited number of state and action space in terms to get reward can be tackled using various additional architectures or by defining constraints on action space based on use case and vertical applications.

5.2 DRL based slice resource allocation and management

The main objective is to develop a robust framework for RAN slicing resource allocation as well as management for enhanced mobile broadband and ultra-low latency services at the edge of the network, using cutting-edge technologies, such as distributed RL and federated learning (FL). The proposed framework will be autonomous and able to dynamically group the VNFs to form an end-to-end slice specific to end user request and achieve defined QoS. The slicing will be focused on reducing service latency and increase the quality of service while ensuring optimal resource allocation by the use of intelligent agent placement at the edge of the network. The solution will follow the guidelines of the standardization, such as the 3rd Generation Partnership Project (3GPP), Open radio access network forum (O-RAN) and ETSI, as they are working towards defining a unified 5G architecture for network operators to implement as 5G and beyond networks. Specific contribution includes robust modelling and optimization framework for dynamic RAN slicing in 5G and beyond networks for diverse vertical application using reinforcement and deep learning techniques with multiple intelligent agents for each slice.

5.2.1 System model

The architecture in Figure 5-1 shows the O-RAN and 5G based network to serve the users requesting variety of vertical applications. It has four main blocks as, local edge, remote edge, orchestrator and core cloud. The local edge includes O-RU (O-RAN radio units), which are access points or front-end transceivers located at specified distance as per indoor or outdoor scenario with size of a WIFI router. This includes part of L1 as per 3GPP based split options. For the architecture we consider split 7.2 which indicates O-RU with low PHY. Whereas, O-DU has High PHY, L2 and other higher layers along with near real time RAN intelligence controller (near RT RIC) seating at this edge referred as remote edge in the system model. The O-DU is connected with O-RU using open fronthaul interface such as CPRI, eCPRI or other type of operator defined interface (OFI). O-RU will be assigned with specified bandwidth, power and association decision provided by algorithm and it will be serving underlying end users. The radio link established between O-RU and end users requires robust radio resource allocation and management to ensure the defined quality of service. The DRL based algorithm will be seating at the remote edge to distribute intra-slice resources in most efficient and intelligent ways. The orchestrator also seating at the edge will have higher layers and functionalities of slice management modules proposed by 3GPP. It includes network slice management functions, network slice subnet management functions, network slice selection function, non-real time part of RIC, other subnet controllers and slice managers. Slice orchestrator will also be connected to virtual infrastructure management entities and core cloud respectively to provide end to end connection and internet services.

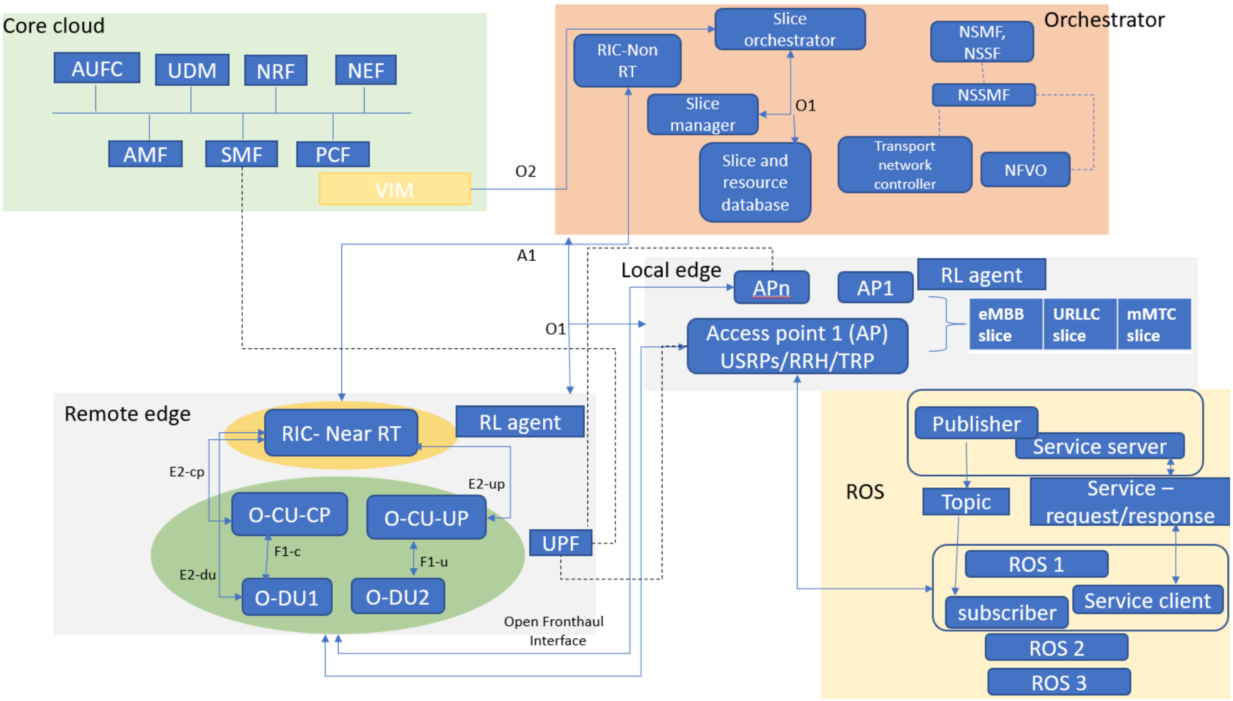

Figure 5-1: 5G Architecture based on ORAN and 3GPP for proposed system model

The network is multi-service, multi-tenant and multi-vendor and adapts intelligent network approaches to utilize the available resources in most optimal way. Few of the important aspects about such intra and inter slice radio resource as well as other subnet resource allocations are dynamic resource handling, isolation and keeping service level agreements in check which include maintaining quality of service. The different vertical applications can be categorized into 5G use cases. Each of these use case corresponds to specific set of data rate, delay, and other QoS requirements. Hence each of such use case will be categorized or fulfilled by different network slices.

The proposed system model is shown in Figure 5-2. The DRL algorithm will be sitting at the remote edge. This edge has O-DU, near RT RIC, databases/storage and intelligent agent for each slice. In this specific proposal, we consider intelligent agents for enhanced mobile broadband (eMBB) and ultra-low reliable latency (URLLC) slice respectively appearing as separate xAPPS at near RT RIC. The modules and architecture are supported and enabled by O-RAN and MEC capabilities. The high-level slicing is executed by orchestrator part and reported to remote edge whereas, the PRB allocation within resource block group distribution is executed at the remote edge. So, each slice has available resource block groups (RBGs) assigned to them to serve underlying nodes. The resource allocation will be done based on state and action to achieve maximum reward and value. The algorithm uses value-based approach for Markov's decision process. The aim here is to achieve maximum quality of service within optimal distribution of resources within each of the slice. Further in this proposal, we will consider exchange of information between these intelligent agents to priorities allocation of RBGs at first place which involve reporting to non-RT RIC and use DRL to execute interslice resource allocation.

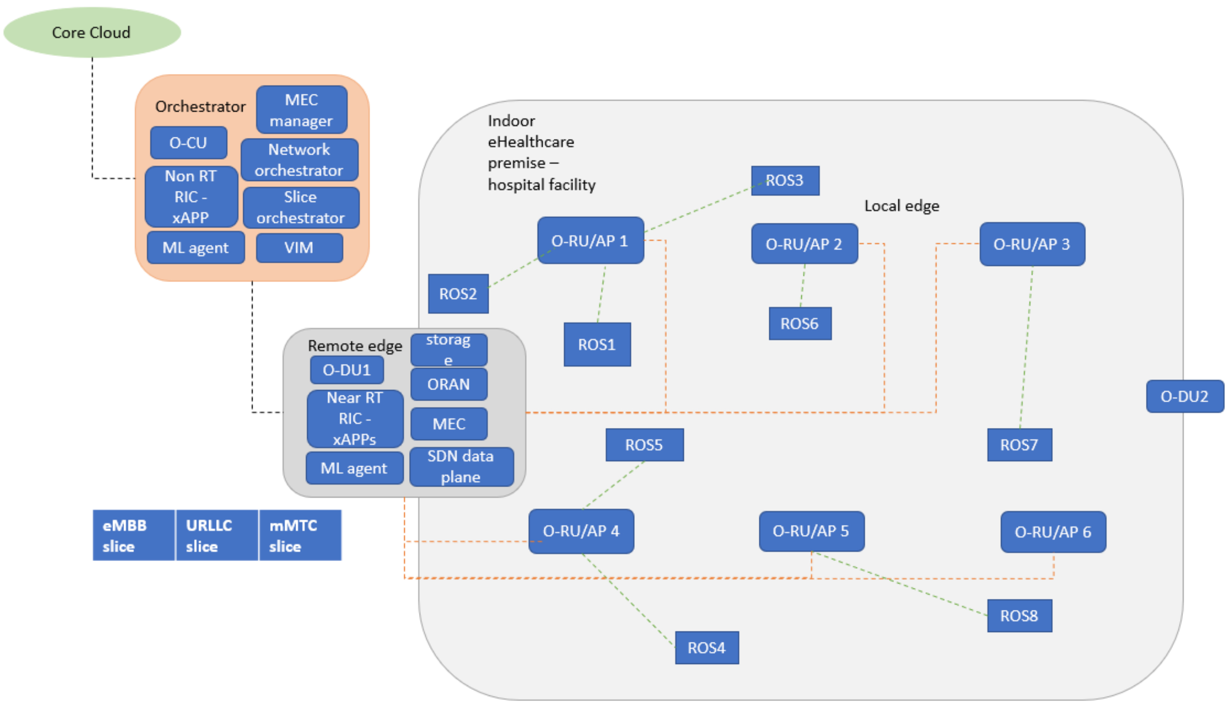

Figure 5-2: Proposed system model

5.2.2 Problem formulation

The RAN network subnet slice instance (NSSI) slice resource management is executed at two different control loop levels, real time and near real time as defined by O-RAN specifications. For initial real time dynamic resource management and allocation, the default RAN RRM configuration is given by service management and orchestration (SMO) via O1. Hence each RAN NSSI has pre-allocated part of bandwidth to serve the associated users. Here the performance of slice is optimized based on requested QoS configurations to achieve service level simulator (SLS). A time frame consists of a number of subframes. The total frame duration is divided into a set of time slots $\tau = \{1,2,3,\ldots,T\}$ each with 1 ms and consisting of one physical resource block of 180 kHz. The entire system model comprises of $UE = \{1,2,3,\ldots,K\}$ set of users k∈ $\{K_{ul}, K_{dl}\}$,$O-RUs = \{1,2,3,\ldots,M\}$

set of O-RUs, and the spectrum which can be shared among all users in UL and DL and it is divided in $PRB = \{1,2,3, \dots, PRB\}$ sub-bands or sub-channels. We assume that AWGN at all users are independent circular symmetric complex random variance with zero mean and variance of σ^2. The channel model is considered as a frequency selective flat fading with the pedestrian mobility of the users in the system model. The CSI can be extracted from the database of the near RT RIC as reported by E2 node that lies at the edge server. Therefore, we assume that perfect CSI is available for the resource allocation and management at the edge server. Each user is associated with O-RUs on either UL or DL. The data rate of DL can be expressed as,

$$R_k^D = BW_k \, log_2\left(1 + \frac{b_{k,m} P_m^D \left|h_{k,m}^D\right|^2}{I_{k,m}^D + \sigma^2}\right) \quad (1)$$

For eMBB intelligent agent, the objective is to maximize the system throughput by jointly optimizing the uplink and downlink (UL and DL) scheduling, PRB allocation and O-RU selection for eMBB RAN NSSI while checking whether each user is achieving the minimum QoS requirements while not exceeding the assigned BW for respective RAN NSSI. It also checks the fronthaul capacity constraints Therefore, our problem can be mathematically formulated as,

$$max \sum_{k \in K} \sum_{m \in M} \left[a_{k,m,} R_k^U + \, b_{k,m} R_k^D\right] \quad (2)$$

Here, the fronthaul capacity $N_{fronthaul}$ is defined as the maximum number bits transmitted over fronthaul link.

KPI	Threshold value
DL datarate, R^{Dmin} UL datarate, R^{Umin}	20Gbps 10Gbps
DL datarate per user, R_k^{Dmin} UL datarate per user, R_k^{Umin}	100Mbps 50Mbps
Latency, d_k^{max}	20ms

Table 4.1: KPIs and its vales

The delay in the system model, both for uplink and downlink transmission, is calculated based on the transmission time delay d^{tx}, queuing delay d^{que}, and retransmission delay d^{rtx}, by simply taking the sum of all three parameters. The queuing delay and HARQ retransmission delay is assumed a constant value available at near RT RIC reported by E2 nodes form RRC protocol. Whereas, transmission delay is calculated based on packet size for respective user L_k and link capacity $C_{m,k}$ between associated O-RU and user. For URLLC intelligent agent, the objective is to maximize the data rate of serving users by jointly taking optimal decision for association and required number of PRBs to serve requested URLLC users within maximum delay budget and hence can also be indicated by equation (2).

Whereas, for Non-RT RIC intelligent agent, the objective is to maximize the probability of QoS achievement, P_{slice} and utilization, U^{slice} for a given traffic load by using weighted combination of ρ and β respectively based on the information provided by NSSI intelligent agents and hence can be represented as,

$$\max_{\Omega_{slice}} \mathrm{F}(\,P_{slice}, U^{slice}) \tag{3}$$

Subjected to,

$$N_t^e + N_t^u \leq N_{total,t} \qquad t\forall TTI$$

$$where, \qquad N_{total,t} = \left(BW_{total} - 2BW_{guard}\right)/BW_{PRB}$$

Here, $N_t^e, N_t^u, N_{total,t}$ are number of PRBs available at eMBB, URLLC slice and total number of PRBs available respectively. Total number of PRBs available are calculated based on total bandwidth, BW_{total}, guard band BW_{guard} and bandwidth for a single PRB, BW_{PRB}.

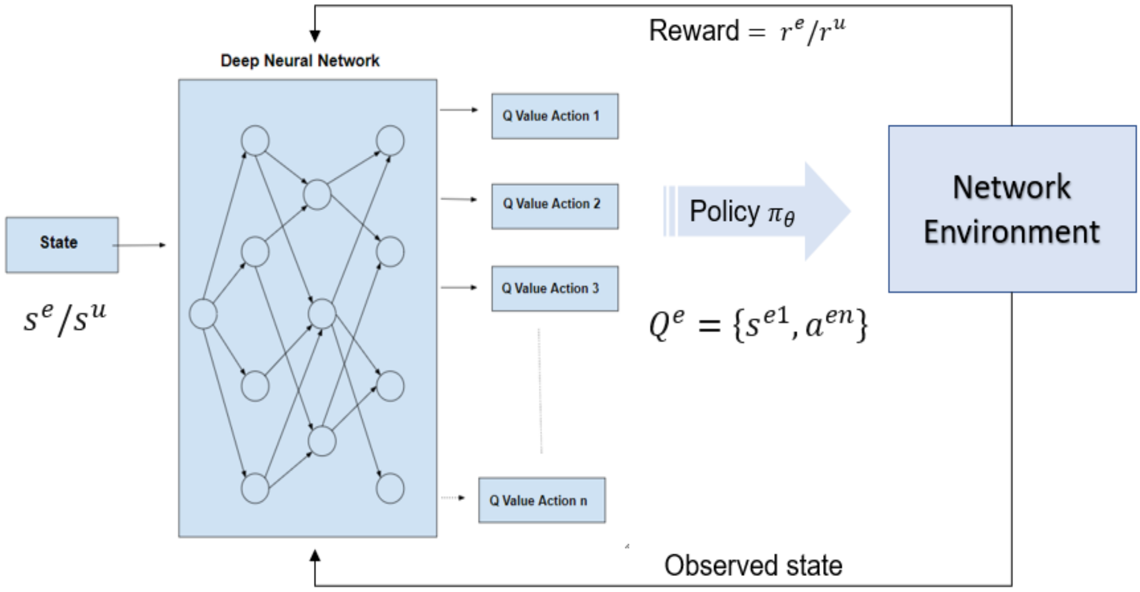

Figure 5-3: Deep Q learning framework

The Markov's decision process (MDP) is defined with tuple of $\{S_e, A_e, R_e\ , \gamma_e\}$ corresponding to state, action, reward and discount factor of NSSI intelligent agent and the Q value, target Q value and loss function is calculated as,

$$TD_s = {r_e}^{i+1} + \gamma \max_{a_e \epsilon A_e} Q^{i+1}(s_e, a_e) \tag{4}$$

We try to minimize the loss function based on TD error at each s steps,

$$L_s = \left[TD_s - \left(Q^i(s_e, a_e)\right)^2\right] \tag{5}$$

$$Q^i(s_e, a_e) = Q^i(s_e, a_e) + \alpha \times L_s$$

i.e., $$Q^i(s_e, a_e) = Q^i(s_e, a_e) + \alpha \left[\underbrace{{r_e}^{i+1} + \gamma \max_{a_e \epsilon A_e} Q^{i+1}(s_e, a_e)}_{\text{Target Q}} - \underbrace{\left(Q^i(s_e, a_e)\right)^2}_{\text{predicted Q}}\right] \tag{6}$$

5.3 Conclusion

This chapter discuss RAN edge domain slice resource allocation and management problem and proposes a solution using deep RL technique for different slices at different timescale to smooth out the utilization of slice resources based on the service demands. The future immediate research plan includes executing the local simulator set up for the proposed system model and problem formulation, to obtain the convergence and achievable matrices as per abovementioned objectives. Further extend the proposed system model and framework to tackle delay latencies up to 5ms in multi-user multi-service multi-tenant scenario as well as improve network reliability up to 99.99% within delay budgets by introducing novel traffic steering techniques.

6 Conclusions and Future Work

This deliverable provided the main frameworks that will be utilized in WP3 towards optimizing end-to-end slicing in beyond 5G and 6G mobile networks. The system models proposed in Chapters 3, 4, and 5, can support functional split at the RAN domain and thus can accommodate the IAB concept. Moreover, traffic steering can be indirectly enforced by suitable VNF placement, according to the generic framework introduced in Section 4.1, and alleviate from link congestion. The proposed algorithms include mainly XAI based Federated Learning, Distributed RL, and multi-agent Deep RL schemes, which can provide scalable solutions and tackle the difficulties introduced by the combinatorial nature of the slicing problems. Some of the proposed models and solutions are not finalized yet, so they will be more clearly defined in the next deliverable, which will also include the corresponding validation results. Hence, future work includes the further refinement of the proposed solutions and their validation either through simulations or by applying them to a real platform.

7 References

[1] X. You et al., "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts," Sci. China Inf. Sci. 64, 110301 (2021)..

[2] N. Rajatheva et al., "White paper on broadband connectivity in 6G," *arXiv preprint arXiv:2004.14247,* 2020.

[3] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Communications Magazine,* vol. 55, 2017.

[4] X. Shen et al., "AI-Assisted Network-Slicing Based Next-Generation Wireless Networks," *IEEE Open Journal of Vehicular Technology,* vol. 1, pp. 45-66, 2020.

[5] M. Z. Chowdhury, M. Shahjalal, S. Ahmed and Y. M. Jang, "6G Wireless Communication Systems: Applications, Requirements, Technologies, Challenges, and Research Directions," *IEEE Open Journal of the Communications Society,* vol. 1, pp. 957-975, 2020.

[6] D. Naboulsi, M. Fiore, S. Ribot and R. Stanica, "Large-Scale Mobile Traffic Analysis: A Survey," *IEEE Communications Surveys & Tutorials,* vol. 18, pp. 124-161, 2016.

[7] 3GPP, "Study on Management and Orchestration of Network Slicing," 3GPP, TR 28.801, 2018.

[8] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 2429-2453, 20018.

[9] X. Foukas, G. Patounas, A. Elmokashfi and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," *IEEE Communications Magazine,* vol. 55, no. 5, pp. 94-100, 2017.

[10] R. Su et al., "Resource allocation for network slicing in 5g telecommunication networks: A survey of principles and models," *IEEE Network,* vol. 33, no. 6, pp. 172-179, 2019.

[11] D. Bega, M. Gramaglia, M. Fiore, A. Banchs and X. Costa-Perez, "AZTEC: Anticipatory Capacity Allocation for Zero-Touch Network Slicing," 2020.

[12] D. Bega, M. Gramaglia, M. Fiore, A. Banchs and X. Costa-Perez, "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019.

[13] S. Vassilaras et al., "The Algorithmic Aspects of Network Slicing," *IEEE Communications Magazine,* pp. 112-119, 2017.

[14] F. Schardong, I. Nunes, A. Schaeffer-Filho, "Nfv resource allocation: a systematic review and taxonomy of vnf forwarding graph embedding," *Computer Networks,* 2021.

[15] ETSI GS ZSM, "Zero-touch network and Service Management (ZSM); Reference Architecture," ETSI GS ZSM 002, 2019.

[16] V. Sciancalepore, F. Z. Yousaf and X. Costa-Perez, "z-TORCH: An Automated NFV Orchestration and Monitoring Solution," *IEEE Transactions on Network and Service Management,* vol. 15, pp. 1292-1306, 2018.

[17] R. Wen et al., "On Robustness of Network Slicing for Next-Generation Mobile Networks," *IEEE Transactions on Communications,* vol. 67, pp. 430-444, 2019.

[18] I. Yahia, J. Bendriss, A. Samba, and P. Dooze, "CogNitive 5G networks: Comprehensive operator use cases with machine learning for management operations," in *20th Conference on Innovations in Clouds, Internet and Networks, ICIN 2017*, Paris, France,, 2017.

[19] L. Le, D. Sinh, B. P. Lin, and L. Tung, "Applying Big Data, Machine Learning, and SDN/NFV to 5G Traffic Clustering, Forecasting, and Management," in *4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018*, Montreal, QC, Canada, 2018.

[20] S. Moazzeni et al, "A Novel Autonomous Profiling Method for the Next-Generation NFV Orchestrators," in *IEEE Transactions on Network and Service Management*, 2021.

[21] M. Bunyakitanon, X. Vasilakos, R. Nejabati and D. Simeonidou, "End-to-End Performance-Based Autonomous VNF Placement With Adopted Reinforcement Learning," *IEEE Transactions on Cognitive Communications and Networking,* pp. 534-547.

[22] B. Monchai, A. P. da Silva, X. Vasilakos, R. Nejabati, and D. Simeonidou, "Auto-3P: An autonomous VNF performance prediction & placement framework based on machine learning," *Computer Networks 181,* vol. 107433, 2020.

[23] B. Brik, and A. Ksentini, "On Predicting Service-oriented Network Slices Performances in 5G: A Federated Learning Approach," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, 2020.

[24] T. Subramanya and R. Riggio, "Centralized and federated learning for predictive VNF autoscaling in multi-domain 5G networks and beyond.," *IEEE Transactions on Network and Service Management,* vol. 18, pp. 63-78, 2021.

[25] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research," *IEEE Network,* vol. 34, pp. 134-142, 2020.

[26] G. Camps-Valls, M. Reichstein, X. Zhu and D. Tuia, "ADVANCING DEEP LEARNING FOR EARTH SCIENCES: FROM HYBRID MODELING TO INTERPRETABILITY," in *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020.

[27] C. Benzaid and T. Taleb, "AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions," *IEEE Network,* vol. 34, pp. 186-194, 2020.

[28] S. Wang, M. Qureshi, L. Miralles-Pechuan, T. Huynh-The, T. R. Gadekallu, and M. Liyanage, "Explainable AI for B5G/6G: Technical Aspects, Use Cases, and Research Challenges," *ArXiv,* 2021.

[29] A. Terra, R. Inam, S. Baskaran, P. Batista, I. Burdick and E. Fersman, "Explainability Methods for Identifying Root-Cause of SLA Violation Prediction in 5G Network," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020.

[30] Y. Wu, G. Lin and J. Ge, "Knowledge-Powered Explainable Artificial Intelligence for Network Automation toward 6G," *{IEEE Network,* vol. 36, pp. 16-23, 2022.

[31] C. Li, W. Guo, S. C. Sun, S. Al-Rubaye and A. Tsourdos, "Trustworthy Deep Learning in 6G-Enabled Mass Autonomy: From Concept to Quality-of-Trust Key Performance Indicators," *IEEE Vehicular Technology Magazine,* vol. 15, pp. 112-121, 2020.

[32] R. R. Hoffman, S. T. Mueller, G. Klein and J. Litman, "Metrics for Explainable AI: Challenges and Prospects," *ArXiv,* vol. abs/1812.04608, 2018.

[33] A. Cotter, H. Jiang, and K. Sridharan, "Two-Player Games for Efficient Non-Convex Constrained Optimization," *coRR,* vol. abs/1804.06500, 2018.

[34] C.J.C.H Watkins, and P. Dayan, "Q-learning," *Machoine Learning,* vol. 8, p. 279–292, 1992.

[35] D. Bertsekas, Reinforcement Learning and Optimal Control, Athena Scientific, 2019.

[36] V. Mnih, K. Kavukcuoglu, D. Silver, et al., "Human-level control through deep reinforcement learning," *Nature,* vol. 518, pp. 529-533, 2015.

[37] T. Bonald, and A. Prouti`ere, "Wireless downlink data channels: User performance and cell dimensioning," in *MobiCom*, 2003.

[38] M. Harchol-Balter, Performance Modeling and Design of Computer Systems: Queueing Theory in Action, USA: Cambridge University Press, 2013.

[39] M. Shojafar, N. Cordeschi and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Transactions on Cloud Computing,* 2019.

[40] M. Maule, P. -V. Mekikis, K. Ramantas, J. Vardakas and C. Verikoukis, “Real-time Dynamic Network Slicing for the 5G Radio Access Network”, IEEE Global Communications Conference (GLOBECOM), DOI:10.1109/GLOBECOM38437.2019.9013965, February 2020..

[41] H. Bakker, M. Doll et al., "“RAN architecture components – final report”," 5G NORMA deliverable D4.2, June 2017.

[42] H. Chergui and C. Verikoukis, "“Big Data for 5G Intelligent Network Slicing Management”," in *IEEE Network, DOI:10.1109/MNET.011.1900437, July 2020.*.

[43] H. Chergui and C. Verikoukis, ""Offline SLA-Constrained Deep Learning for 5G Networks Reliable and Dynamic End-to-End Slicing”," *IEEE Journal on Selected Areas in Communications,,* pp. DOI: 10.1109/JSAC.2019.2959186,, 2019..

[44] B. Han et al., "Admission and Congestion Control for 5G Network Slicing," in *IEEE Conference on Standards for Communications and Networking (CSCN)*, Paris, 2018.

[45] D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs and X. Costa-Perez, "Network Slicing Meets Artificial Intelligence: An AI-Based Framework for Slice Management," *IEEE Communications Magazine,* June 2020.

[46] L. S. Russo, L. Righi, D. Pubill, "LTE as a Service: leveraging NFV for realising dynamic 5G network slicing," in *IEEE Global Communications Conference (GLOBECOM),DOI: 10.1109/GLOBECOM38437.2019.9014330,*, 2019.

[47] J. Mei, X. Wang and K. Zheng, "An intelligent self-sustained RAN slicing framework for diverse service provisioning in 5G-beyond and 6G networks," *Intelligent and Converged Networks,* vol. 1, no. 3, pp. 281-294, Dec 2020.