Federal University of Pará
Technology Institute
Electrical Engineering Graduate Program

Network Slice Admission Using Reinforcement Learning and Information-Centric Networking for Mobile Networks

Pedro dos Santos Batista

TD: 17/2019

UFPA / ITEC / PPGEE
Guamá University Campus
Belém – Pará – Brazil
2019

Federal University of Pará

Technology Institute

Electrical Engineering Graduate Program

Pedro dos Santos Batista

Network Slice Admission Using Reinforcement Learning and Information-Centric Networking for Mobile Networks

UFPA / ITEC / PPGEE

Guamá University Campus

Belém – Pará – Brazil

2019

Federal University of Pará

Technology Institute

Electrical Engineering Graduate Program

Pedro dos Santos Batista

Network Slice Admission Using Reinforcement Learning and Information-Centric Networking for Mobile Networks

Ph.D. thesis submitted to the Examining Board of the Electrical Engineering Graduate Program from the Federal University of Pará to award the Ph.D. Degree in Electrical Engineering in the telecommunications area.

Advisor: Aldebaro Barreto da Rocha Klautau Júnior

UFPA / ITEC / PPGEE

Guamá University Campus
Belém – Pará – Brazil

2019

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

B333n Batista, Pedro
Network Slice Admission Using Reinforcement Learning and Information-Centric Networking for Mobile Networks / Pedro Batista. — 2019.
xvii, 78 f. : il. color.

Orientador(a): Prof. Dr. Aldebaro Klautau
Tese (Doutorado) - Programa de Pós-Graduação em Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal do Pará, Belém, 2019.

1. network slicing. 2. information-centric networking. 3. mobile networks. I. Título.

CDD 384

Federal University of Pará
Technology Institute
Electrical Engineering Graduate Program

Network Slice Admission Using Reinforcement Learning and Information-Centric Networking for Mobile Networks

Author: Pedro dos Santos Batista

PH.D. THESIS SUBMITTED TO THE EXAMINING BOARD APPROVED BY THE COMMITTEE OF THE ELECTRICAL ENGINEERING GRADUATE PROGRAM FROM THE FEDERAL UNIVERSITY OF PARÁ AND JUDGED APPROPRIATE TO AWARD THE PH.D. DEGREE IN ELECTRICAL ENGINEERING IN THE TELECOMMUNICATIONS AREA.

Approved on: ____/____/________

Examining board:

____

Advisor – Internal – UFPA
Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior

____

Member – Internal – UFPA
Prof. Dr. Eduardo Coelho Cerqueira

____

Member – Internal – UFPA
Prof. Dr. Jasmine Priscyla Leite de Araujo

____

Member – External – UFPA
Prof. Dr. Antonio Jorge Gomes Abelém

____

Member – External – UFG
Prof. Dr. Kleber Vieira Cardoso

Homologation:

____

UFPA–ITEC–PPGEE coordinator
Prof. Dr. Maria Emília de Lima Tostes

Agradecimentos

Sou imensamente afortunado graças as pessoas que aparecem na minha vida, tenho um ímã que, na maior parte do tempo, atrai as melhores pessoas que estão pelo mundo para o meu convívio.

Agradeço primeiramente à minha família, que esteve sempre ao meu lado. Em especial à minha mãe Rosilda e meu pai Alfonço, que me mostraram o caminho até aqui. Mesmo que eu nem sempre tenha seguido por onde me guiaram, sempre me deram apoio incondicional em todos os momentos da minha vida.

Aos meus irmãos, Luiz e Paula, que apesar das brigas conseguem aturar meu gênio forte, e por mais que pareça estarmos longe, sempre os sinto perto. E claro, aos meus tios, tias e primos que me recebem com carinho e alegria, mesmo eu passando mais tempo longe de Concórdia do que eu deveria e gostaria.

Minha noiva, Carolinne Melo, que investiu tanto nesse trabalho quanto eu. Me sustentar no estresse gerado no desenvolvimento dessa tese foi uma batalha vencida com maestria. Por sinal, ainda hei de descobrir um desafio ou ambição que eu tenha e ela não me ajude a conquistar com eterna alegria e satisfação. Agradeço também à sua família que sempre nos proporciona alegria.

Não poderia deixar de agradecer meu orientador, outra vez o meu ímã atuou e colocou em meu caminho um dos melhores professores da UFPA, que é também meu orientador, conselheiro, e amigo Aldebaro Klautau. O agradeço pelos inúmeros conselhos e por sua inesgotável energia investida no LASSE, fazendo do laboratório um excelente ambiente de trabalho, que reúne e constrói mentes excepcionais focadas em melhorar a vida de todos nós. Admiro a forma com que ele executa sua profissão, gerando impacto imensurável na nossa sociedade. Agradeço também aos professores do PPGEE, que proporcionaram a alicerce necessário para desenvolver este trabalho.

Entre idas e vindas, foi no LASSE, que pra mim ainda se confunde com LaPS, que encontrei minha segunda família. E a Kelly é a responsável pelo funcionamento desse lugar. Poucos lugares vamos encontrar pessoas tão competentes e amigáveis juntas. Todo o pessoal do TROBE, Bruno, Camila, Carol, Ilan, Jully, Leonardo, Marcus, Müller, Silvia, Yuichi e Yuki compõem a melhor geração produzida pelo LASSE, com a qual tive o prazer de compartilhar grande parte da minha vida, sendo em vários trabalhos ou inúmeras praias e cervejas.

Durante meu estágio tive o prazer de conhecer e receber ajuda de várias pessoas durante processo de desenvolvimento deste trabalho como Peter Ölén, Neiva Linder, Kim

Laraqui, Shah N. Khan, Mateus Santos e Mikael Prytz. Além de ter reencontrado e encontrado diversos amigos que tornaram a vida à distância mais prazerosa, Lian, Eduardo, Thiago, Carlos, Denise, Iliézer e Amanda.

Gostaria também de agradecer a comunidade *open source* pelo ótimo trabalho, disponibilizando recursos para que projetos como este sejam possíveis.

Sou grato também as agências governamentais financiadoras de pesquisa pelo investimento feito em mim. Assim como empresas colaboradoras do LASSE.

À banca examinadora, sou grato pelo tempo investido na avaliação deste trabalho. Ajudando a aprimorá-lo, e melhor contribuir com a comunidade acadêmica.

Pedro dos Santos Batista

Abstract

The evolution of the current most popular mobile network (4G), the so-called 5G, is targeting an increased traffic load at a lower cost. Thus, optimization of the delivery network plays an essential role at 5G; another aspect of the evolution is that 5G has the ambition to be highly customized, e.g., reliable enough to be used in industrial automation and cheap enough to be used for mobile broadband services. In this context, this thesis assesses two aspects of 5G: the first is to use information-centric networking (ICN) to improve the efficiency of multimedia delivery in mobile broadband services; and the second is the application of a reinforcement learning strategy as an enabler for the highly configurable network, which could pose a challenge to be understood and configured manually. ICN aims at circumventing several issues of current internet protocol, among them, achieving a more efficient multimedia distribution. Given the significant growth rate of video transmission over mobile networks, it is sensible to consider how mobile networks can leverage ICN. There is a substantial body of work considering ICN for fixed networks and also for the core of mobile networks. Less attention has been dedicated to ICN on the radio access network (RAN) or ICN-RAN, which has currently a user plane based on many connection-oriented protocols. To fully benefit from ICN, mobile networks must enable it on the RAN, not only on the core. This work details an ICN deployment on the RAN of the fourth and fifth generation of mobile networks and also presents a testbed that enables proofs of concept of this ICN-RAN using 4G. The results indicate, for example, that evolving ICN features can be tested with currently available tools, but the lack of hardware accelerators and optimized code limit the bit rate that can be achieved in real-time processing. In the context of network customization, the most prominent enablers are the so-called network slices. Slices can be understood as a part of the network that is customized to deliver certain services. The service requirements are imposed by the tenant, which acquire slices from an infrastructure provider. The 5G infrastructure provider must optimize the infrastructure resource utilization, usually admitting as many slices as possible. However, infrastructure resources are finite and admitting all the slices could increase the probability of service level agreement violation. This thesis investigates the application of reinforcement learning agents that learn how to increase the infrastructure provider revenue by intelligently admitting network slices that bring the most revenue to the system. We present a neural networks-driven agent for network slice admission that learns the characteristics of the slices deployed by the tenants from their resource requirements profile and balances the benefits of slice admission against orchestration and resource management costs.

Keywords: network slicing; information-centric networking; mobile networks.

Resumo

A evolução das redes móveis mais populares atualmente (4G), as 5G, tem como um dos objetivos suportar aumento de tráfego e ao mesmo tempo diminuir o custo. Assim otimização na entrega de conteúdo é importante para essa nova rede; um outro aspecto é que 5G tem a ambição de ser uma rede altamente adaptável, isto é, ela deve ser confiável o suficiente para ser utilizada em automação industrial e ao mesmo tempo barata o suficiente para ser usada em serviços de banda larga. Nesse contexto, esta tese estuda dois aspectos do 5G, o primeiro é o emprego de redes orientadas a conteúdo (ICN) para melhorar a eficiência de entrega de conteúdo multimídia em serviços de banda larga móvel; o segundo é o desenvolvimento de um agente que utiliza aprendizado por reforço como um facilitador para as novas redes altamente configuráveis, as quais podem se tornar um desafio para serem entendidas e configuradas manualmente. O ICN tem como objetivo circunver vários problemas do atual protocolo de internet, dentre eles, uma entrega de conteúdo mais eficiente. Dado significativa taxa de crescimento de transmissão de vídeos em redes móveis, é sensível avaliar como as redes 4G/5G podem se beneficiar de ICN. Existem muitos trabalhos que avaliam o emprego de ICN em redes fixas e para o núcleo das redes móveis. Menos atenção tem sido dedicada ao emprego de ICN nas redes de acesso a rádio (RAN) ou ICN-RAN. Este trabalho descreve o emprego de ICN na RAN de 4G/5G, e também apresenta uma bancada de testes que permite o desenvolvimento de provas de conceitos usando ICN-RAN em 4G. Os resultados indicam, por exemplo, que a avaliação de diversas funcionalidades de ICN podem ser realizadas, mas que a falta de aceleradores de *hardware* e código otimizado limitam a taxa de bit que pode ser alcançada em tempo real. No contexto de adaptação da rede, a tecnologia mais promissora é o fatiamento da rede. Fatia de rede pode ser entendida como parte da rede que é personalizada para determinados serviços. Os requisitos de cada serviço são impostos pelo inquilino, o qual adquire fatias do provedor de infraestrutura. O provedor de infraestrutura 5G tem que otimizar a utilização de seus recursos, costumeiramente essa utilização é aumentada ao admitir fatias, porém, os recursos na infraestrutura são finitos e admitir todas as fatias pode aumentar o risco de violação de acordos de prestação de serviços, o que implica em multas que podem diminuir o lucro. Nesta tese, é investigado o uso de um agente treinado por aprendizado por reforço que aprende como aumentar o lucro do provedor de infraestrutura. Tal agente, baseado em redes neurais, aprende as consequências da admissão de fatias na rede baseado no inquilino e no seu perfil de utilização de recursos, aprendendo assim, a balancear os benéficos da admissão em contraste com os custos de orquestração e gerenciamento de recursos.

Palavras-chave: fatiamento de redes; redes orientadas a conteúdo; redes móveis.

List of Figures

List of Tables

List of abbreviations and acronyms

3GPP	3rd Generation Partnership Project
4G	Fourth generation mobile networks
5G	Fifth generation mobile networks
A-CRC	Aggregate content routing controller
ACPT	Accept all policy
ANN	Artificial neural network
APN	Access point name
BL	Baseline agent
BM-SC	Broadcast multicast service center
C-RAN	Centralized radio access network
C-RNTI	Cell radio network temporary identifier
CCN	Content-centric networking
CDN	Content delivery network
CO	Central office
COTS	Commercial off-the-shelf
CRC	Content routing controller
CS	Content store
D2D	Device to device
DNS	Domain name server
E-MBMS	Evolved multimedia broadcast multicast service
E-UTRAN	Evolved universal terrestrial radio access network
EBI	Evolved packet system bearer identity
ECOMP	Enhanced Control, Orchestration, Management & Policy

eNodeB	Evolved node B
EPC	Evolved packet core
EPS	Evolved packet system
ETSI	European Telecommunications Standards Institute
F-TEID	Fully qualified tunnel endpoint identity
FI	Future internet
FIB	Forwarding information base
G-CRC	Global content routing controller
GBR	Guaranteed bit rate
GPP	General purpose processor
GPRS	General packet radio service
GTP	General packet radio service tunneling protocol
GTP-C	Control plane general packet radio service tunneling protocol
GTP-U	User plane general packet radio service tunneling protocol
GUTI	Globally unique temporary identifier
HSS	Home subscriber server
HTML5	Hypertext markup language version 5
HTTP	Hypertext transfer protocol
HTTPS	Hypertext transfer protocol secure
ICN	Information centric networking
ICN-RAN	Information-centric networking on the radio access network
ICNRG	Information-Centric Networking Research Group
IMSI	International mobile subscriber identity
InP	Infrastructure provider
IP	Internet protocol
IPsec	Internet protocol security

IRTF	Internet Research Task Force
L-CRC	Local content routing controller
LTE	Long-term evolution
MAC	Medium access control
MANO	Network Function Virtualization Management and Orchestration Framework
MBMS-GW	Evolved multimedia broadcast multicast service gateway
MCE	Multi-cell/multicast coordination entity
MDP	Markov decision process
MEC	Mobile edge computing
ML	Machine learning
MME	Mobility management entity
MPTCP	Multipath transmission control protocol
NAS	Non-access stratum
NDN	Named data networking
NFD	Named data networking forwarding daemon
NFV	Network function virtualization
OAI	OpenAirInterface
ONAP	Open Network Automation Platform
OPEN-O	Open Orchestrator Project
OSM	Open Source Network Function Virtualization Management and Orchestration Framework
OSPF	Open shortest path first
PC	Personal computer
PDCP	Package data convergence protocol
PDN	Packet data network
PDN-GW	Packet data network gateway

PIT	Pending interest table
PoC	Proof of concept
QoS	Quality of service
RAN	Radio access network
RDC	Regional data center
RL	Reinforcement learning
RLC	Radio link control
RND	Random policy
RRC	Radio resource control
RRU	Remote radio unit
RTT	Round-trip time
S-GW	Serving gateway
SAE	System architecture evolution
SDN	Software-defined networking
SDR	Software-defined radio
SLA	Service level agreement
SPP	Special purpose processor
SRB	Signaling radio bearer
TCP	Transmission control protocol
TEID	Tunnel endpoint identity
UDP	User datagram protocol
UE	User equipment
UFPA	Federal University of Pará
UMTS	Universal mobile telecommunications service
URL	Uniform resource locator
USB	Universal serial bus

USIM	Universal subscriber identity module
USRP	Universal software radio peripheral
UTF	Unicode transformation format
VNF	Virtual network function
VoIP	Voice over IP
vPP	Virtual packet processor

Contents

1 Introduction

Mobile networks have been growing every day. During 2017 it grew 71% globally, by 2022 it is expected to represent 20% of all internet protocol (IP) traffic [1], and by 2024 account for 95% of all internet subscriptions [2]. The network requires innovation to keep up with this growth, particularly important is improved multimedia delivery [3]. Video traffic represents 60% of current mobile traffic, and it is predicted to continue increasing [2]. Motivated by information-centric networking's (ICN's) advanced features for video and popular content delivery; and the importance of media aware network optimization, the first part of this thesis discusses the adoption of ICN in the modern radio access network (RAN) or ICN-RAN.

Another aspect of the mobile network evolution is the support for private networks or services offers for new verticals. The industry predicts a high number of network deployment for different segments, for example, manufacturing and public safety [2]. However, this new verticals should be accommodated in the same network used by mobile broadband users, so that the infrastructure provider can decrease costs. These circumstances impose a higher degree of network configuration, given that the requirements for manufacturing are different from the ones of mobile broadband, e.g., reliability. Network slicing is the technology that enables such adaptability. Therefore, the second part of this thesis proposes a method to control the admission of network slices in a mobile network. Such method helps to guarantee that the network can fulfill the requirements of the new slice introduced in the system.

1.1 Information-Centric Networking and Mobile Networks

Amongst the strategies which have been being studied to develop the future internet (FI) two stand out: the *evolutionary* and the *clean-slate*. The first has as strategy evolve current networks to fulfill the demands of the future. The second, instead, promote solutions that do not maintain compatibility with the current networks, so that it can be created with fewer restrictions.

One of the clean-slate projects, the Named Data Networking (NDN) [4], which is an ICN architecture, stands out because it focuses on popular content delivery. For example, in current networks, to deliver the same content, the flows are usually replicated [5], with NDN this does not happen because the flows are natively aggregated.

ICN is positioned to be a relevant part of the FI architecture as indicated by the corresponding large number of research efforts and applications (see, e. g. [6] and references therein). The high demand for multimedia content is one of the drivers of ICN. Despite its appeal, ICN has to overcome and coexist with many well-established mobile networking technologies and business models built on legacy IP networks. At the current maturity stage of this technology, proofs of concepts and testbeds of ICN for mobile networks are essential.

Assessing the applicability of ICN in mobile networks, this thesis formalizes a specific ICN-RAN deployment and presents a complete testbed to attest the deployment feasibility. For this, we use fourth-generation mobile networks, or Long-Term Evolution (LTE) [7], and named data networking [4]. The testbed further allows experimentation with protocols and commercial-off-the-shelf (COTS) user equipments (UEs). This thesis is particularly aligned with previous contributions advocating that comprehensive testbeds are essential for complementing theoretical and simulation-based ICN contributions [8]. There is significant previous work on ICN for mobile networks, and the following discussion briefly anticipates how the current work is related to previous ones.

While most ICN research has been carried in the context of *fixed* networks [9], [10], there are several publications addressing ICN for mobile networks [11]–[22]. These previous papers can be organized into two broad categories: ICN *deployments* and *services*. The works in the former category focus on how the ICN protocol is better incorporated in mobile networks [11]–[15], while the ones in the second assume that ICN is already enabled and target improved services through, e.g., efficient algorithms for caching [16]–[18], forwarding strategies [19]–[21] and producer and consumer mobility [22].

In the *deployments* category, 5G Americas [11], Suthar and Stolic [12], and Ravindran *et al.* [15] present general considerations about the adoption of different ICN deployments in mobile networks. One conclusion of Suthar and Stolic [12] is that there is not much benefit in adopting ICN in the control plane of mobile networks. Ravindran *et al.* [15], in its turn, names some ICN deployments and proposes an integration of ICN into future mobile networks using network function virtualization (NFV). 5G Americas [11] gives insights on the benefits of ICN-RAN and its association with mobile edge computing (MEC). Kim [14] investigates the maintenance of routing information for the mobile producer. To the best of the author's knowledge, the only implementation of ICN into mobile networks was done in Gomes and Braun [13], using an emulated environment. The experiment was in the communication between UE and LTE base station and concluded that, from the computational cost point of view, it was feasible. Instead of an emulated environment, this thesis presents a first experiment integrating ICN in a full stack prototyped mobile network.

While not as related to the current work as the ones in the *deployments* category, it is useful to briefly describe some representative works in the *services* category. Mobility pattern is used in Zhang *et al.* [17] as information for caching in high mobility scenario, such as millimeters waves in fifth generation mobile networks (5G). In a similar scenario, Gomes *et al.* [18] caches the content in the network edge and migrates it as the user moves. Ventrella *et al.* [22] proposes a publish-subscriber ICN model in an internet of things 5G mobile network scenario. Jin *et al.* [16] presents a further review of cache studies in mobile and ad hoc networks. Nishiyama *et al.* [20] proposes a local forwarding for communications of short duration in mobile networks, and a routing method for communications of longer duration, based on the open shortest path first (OSPF) routing protocol. Gomes and Braun [21] makes use of forwarding for load balancing between LTE and Wi-Fi access networks. Carofiglio *et al.* [19] addresses the use of a forwarding strategy that also makes a load balancing but considering the number of pending *interests* in each interface, along with congestion control for outgoing *interests*.

Together with the testbed and proof of concepts (PoCs), a contribution of this work is a thorough description of the ICN-RAN deployments. The testbed presented in this work adopts an ICN-RAN deployment that has the advantage of allowing cache at the edge (LTE base station) and also supports legacy UE. Given the new technology, the standardization activities are still ongoing [23]; and previous works in the literature do not address implementation aspects that are emphasized in this thesis.

1.2 Network Slicing

Communication networks have been continuously evolving towards an ever-increasing complexity in both integrating new technologies and supporting new verticals. The former requires a cross-domain and cross-technology network deployment and optimization. While the latter imposes heterogeneous requirements on the network operators and the infrastructure that must support them [24].

5G networks, the most recent evolution of mobile communication networks, are anticipated to be a platform not only for integrating new and revolutionary technologies, such as software-defined networking (SDN) and NFV, but also support the requirements of new verticals through network slicing and multi-tenancy [25]. However, with such a wide range of technologies being integrated and support for different verticals developed, 5G networks have become extremely complex for management and control using the traditional network practices. One consequence of the 5G complexity is a large number of configurable parameters that exist in the network's cloud, radio access, and control and management domains. Doing a large number of possible configurations manually is bound to trigger suboptimal configurations. This may lead not only to service disruption and failures, but also adversely affect the network infrastructure's revenue generation capacity.

In this context, it is well recognized that network automation expecting minimal human intervention is required to handle this high network complexity [24]. In existing networks, automation is generally an add-on feature that is mostly driven by pre-defined sets of rules for specific contexts of use cases, such as load balancing, mobility management, and interference management. In 5G networks, though, network automation driven by machine learning and artificial intelligence is anticipated to be a core feature that will drive most of the network management and control functions autonomously.

An important feature of 5G networks to support multi-tenancy is the network slicing concept. Network slicing enables the network operator or infrastructure provider (InP) to facilitate different service providers (or tenants) in the network by dynamically assigning resources to tenants [26]. The tenants, in turn, offer revenue for the resources allocated to their deployed services in their dedicated slice.

The network slicing concept has been considered in different scales, abstraction levels, and network segments in the context of multi-tenant 5G networks [26]–[29]. Nevertheless, regardless of how network slices are defined, they are eventually mapped onto a shared network infrastructure. That infrastructure has to be managed by the InP to optimize both resource utilization and revenue generated from the deployed slices. The management of those abstract infrastructure resources is a theme investigated by this thesis. We present a reinforcement learning-based agent that aims to optimize the revenue generated from deploying different slices in the network while ensuring that the deployed services can elastically scale their resource consumption footprint when needed.

1.3 Contributions

In the first part of this thesis, we show that information-centric networks can be integrated into the radio access network of LTE to provide improved and efficient delivery of multimedia in future mobile networks. We identify and formalize the ICN-RAN network architectures discussed in the literature and compare them in terms of functionalities, providing an overview discussion of their potential strengths and weakness. We also propose a deployment option using a specific functionality split that allows the usage of legacy user equipment in ICN-enabled LTE RAN. Moreover, we give insights on the challenges of integrating ICN and RAN by implementing a testbed in which we prototype a live LTE network with ICN. This prototype allowed the evaluation, from a functionality point-of-view, of the architecture.

The other main contribution was the application of reinforcement learning to enable the management of slices in future mobile networks. Our admission control proposal considers, from the InP point-of-view, the tenant-specific slice behavior, which allowed better resource management and increased revenue. These benefits emerge because the InP leverages the tenant-specific statistics to reduce the chance that tenants suffer service level agreement breaches.

Those two topics were described as academic articles, accepted, and published. Therefore, parts of this thesis are copyright protected. The articles included in this thesis are shown in the list that follows.

- P. Batista, I. Araújo, N. Linder, K. Laraqui, and A. Klautau, "Testbed for ICN media distribution over LTE radio access networks", *Computer Networks*, vol. 150, pp. 70–80, Feb. 26, 2019, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2018.12.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S138912861831380X (visited on 07/05/2019).
- P. Batista, S. N. Khan, P. Öhlén, and A. Klautau, "Tenant-Aware Slice Admission Control Using Neural Networks-Based Policy Agent", in *Cognitive Radio-Oriented Wireless Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Poznan, Poland: Springer International Publishing, 2019, pp. 17–30, ISBN: 978-3-030-25748-4. DOI: 10.1007/978-3-030-25748-4_2.

While developing the concepts for this thesis, the main author was involved in an ecosystem and had the chance to apply the primary knowledge of this thesis to different studies, e.g., the use of machine learning for applications other than slice admission and the development of mobile networks for humanitarian engineering. Those studies produced other publications, and the list that follows summarizes them (two of them are accepted to be published and are marked accordingly).

- A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO Data for Machine Learning: Application to Beam-Selection Using Deep Learning", in *2018 Information Theory and Applications Workshop (ITA)*, Feb. 2018, pp. 1–9. DOI: 10.1109/ITA.2018.8503086.
- C. Natalino, M. R. Raza, P. Öhlén, P. Batista, M. Santos, L. Wosinska, and P. Monti, "Machine-Learning-Based Routing of QoS-Constrained Connectivity Services in Optical Transport Networks", in *Advanced Photonics 2018 (BGPP, IPR, NP, NOMA, Sensors, Networks, SPPCom, SOF) (2018), Paper NeW3F.5*, Optical Society of America, Jul. 2, 2018, NeW3F.5. DOI: 10.1364/NETWORKS.2018.NeW3F.5. [Online]. Available: https://www.osapublishing.org/abstract.cfm?uri=Networks-2018-NeW3F.5 (visited on 07/12/2019).
- P. Batista, L. Brito, E. Oliveira, M. Neto, and A. Klautau, "Pilot Projects for GSM Community Networks in Amazon", presented at the Workshop of ICT for

Development (WTICp/D), Belém, Brazil: Brazilian Computer Society, 2017. [Online]. Available: https://sbrc2017.ufpa.br/wp-content/uploads/2017/08/proceedingsWTICpD2017v2.pdf (visited on 07/12/2019).

- B. Vilas Boas, M. Dias, P. Batista, A. Oliveira, and A. Klautau, "CELCOM Project: Engineering Practice via Community Networks in Amazon", *International Journal of Engineering Education (IJEE)*, to appear, Special Issue on Open Source & Collaborative PBL in Engineering Education 2019, accepted, to appear.

- C. Batista, R. Cunha, P. Batista, A. Klautau, and N. Neto, "Utterance Copy in Formant-based Speech Synthesizers Using LSTM Neural Networks", presented at the Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil: Brazilian Computer Society, 2019, accepted, to appear.

- M. Dias, J. Paulo, V. Soares, P. Batista, and A. Klautau, "Techno-economic study to connect Amazon using community cellular networks", presented at the Workshop of ICT for Development (WTICp/D), Belém, Brazil: Brazilian Computer Society, 2017. [Online]. Available: https://sbrc2017.ufpa.br/wp-content/uploads/2017/08/proceedingsWTICpD2017v2.pdf (visited on 07/12/2019).

- C. Nahum, J. Soares, P. Batista, and A. Klautau, "Emulation of 4G/5G Network Using OpenAirInterface", presented at the XXXV Brazilian Communications and Signal Processing Symposium, São Pedro, Brazil, 2017, ISBN: 978-85-66836-18-9. [Online]. Available: http://www.sbrt.org.br/sbrt2017/anais/anais_sbrt_2017.pdf.

1.4 Summary of the Remaining Content

This chapter briefly introduced the objectives of this work, highlighted the goals achieved, and presented some related work. We proceed with a summary of the subjects discussed in the next chapters.

Chapter 2 introduces the technologies used for this work, giving particular attention to topics of interest. It includes the description of fourth generation mobile networks and exemplifies the functionalities of its elements by describing the UE attach procedure. The other main technology is reinforcement learning, which is also introduced. This technique is exemplified by its application in an simplified slice admission problem.

Chapter 3 discusses content delivery in the current most popular mobile network, the 4G. It also shows how this thesis is positioned among related work and the different options for ICN adoption in mobile networks. Furthermore, it illustrates what problems would be facilitated, as well as the difficulties for ICN-RAN adoption. This chapter also includes the description of testbed implemented in this thesis, specifying the details of implementation and used equipment, followed by a discussion of the insights revealed by it.

Chapter 4 provides the related work on network slicing management and platforms supporting network slice deployment in the scope of virtualized 5G networks. We describe a general management loop that might be used to model different network operations. We apply this concept for slice management, and then isolate the slice admission problem. We present our reinforcement learning agent to tackle this problem together with the system model as well as the simulation scenarios and results.

Chapter 5 concludes with a summary and discussion of impacts of this work and possible further development.

2 Theoretical Foundations

With the goal of introducing the concepts used for the development of this thesis, this chapter starts describing the two technologies that were integrated in the development of the first part of the thesis: ICN and evolved packet system (EPS). The former is described focusing on NDN, an ICN implementation. The later is discussed in its two parts: LTE and evolved packet core (EPC), including the description of the UE attachment procedure to illustrate the functionalities of EPS components. The second part of this thesis deals with a reinforcement learning (RL) agent. Consequently, this chapter also introduces the primary concepts for this learning strategy. These concepts are introduced in the context of a simplified version of the slice admission problem studied in Section 4.

2.1 Information-Centric Networking

ICN replaces the current connection-oriented approach for networking, which relies on IP, to focus on the content and its distribution. For example, ICN simplifies caching and therefore can be an essential enabler to improve video distribution over networks [39].

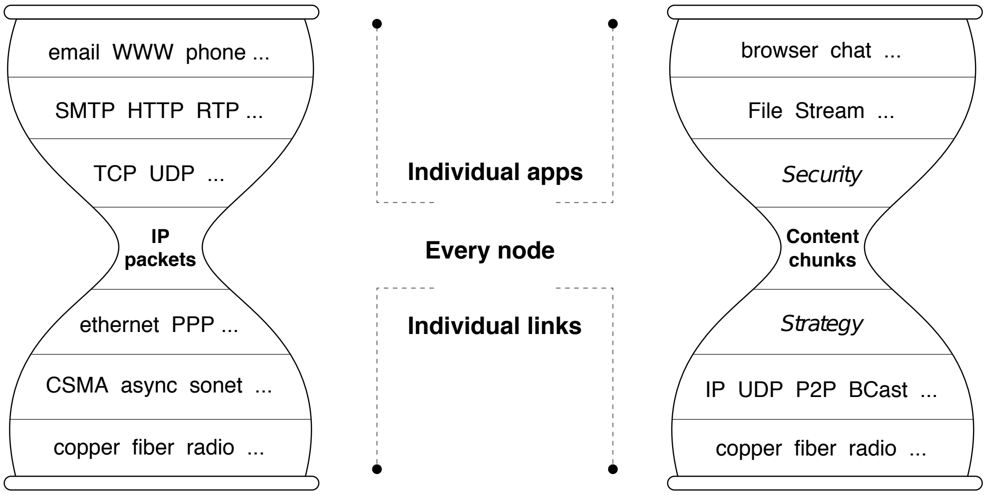
Even though ICN is inspired by IP, they do not maintain compatibility between each other, allowing ICN to be clean-slate. A comparison of these architectures is shown in Figure 1, we can see that the two architectures share the transport layer. Furthermore, the IP can even be used for ICN transport. Another characteristic worth mentioning is that ICN has security embedded in the network, thus, allowing the application to be developed independently, without worrying about security, trusting in the network. With IP, security is implemented as part of the application, protecting the connection instead of the content, which adds complexity to development.

On the ICN paradigm, the content is requested from the network without the need of maintaining a connection with the data producer. On IP, instead, first the content producer has to be identified, a connection established, and then the content can be requested. This data focus is an important design principle of ICN, which simplifies some problems of the current network, e.g., consumer mobility.

Among the proposed ICN architectures and corresponding implementations, NDN[1] and content-centric networking (CCN) [40] are the most prominent. They influence the on-going ICN standardization process carried out by the Internet Research Task Force (IRTF) Information-Centric Networking Research Group (ICNRG) [41]. For this work, we chose to use NDN because it has an open and active community, which also organizes the biggest

[1] https://named-data.net/

Figure 1 – Hourglass architecture comparing IP and ICN. The figure contrasts the fundamental unity of the two architectures, for the IP, an end-to-end connection and, for NDN, the content.



Source: Zhang *et al.* [4].

ICN conference[2], making the access to project information and help easier.

In both NDN and CCN, packets contain a name that indicates a proper content in the network. The *interest* is used to request this content, and the *data* to carry the content itself.

The NDN architecture allows the application to define the naming scheme. The NDN standard requires only that names are unique and hierarchically structured. The NDN standardizes how the bytes that form the names are translated to legible characters. The strategy is inspired by the uniform resource locator (URL). Thus, the "`/`" is used to delimit hierarchical components. And each component is translated in characters according to the Unicode transformation format (UTF). Additionally, "`%`" is used to mark numbers in hexadecimal that represent bytes which are not representable by UTF.

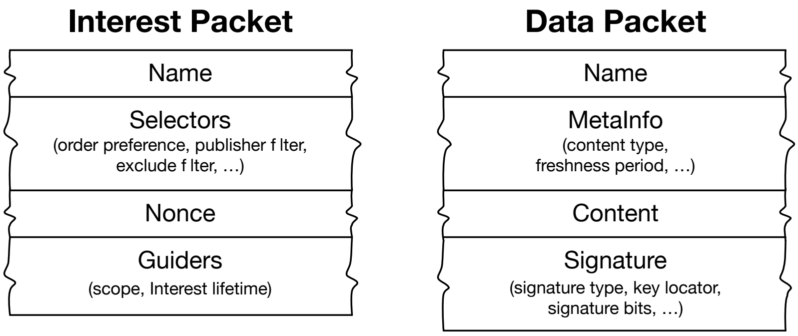
As an example, an application might use the name "`/br/ufpa/videos/aula1.mp4`" to indicate the video of a class from the Federal University of Pará (UFPA). In this example, we can observe the hierarchical structure because the videos are distributed by UFPA, which is managed by another entity, Brazil (`br`).

One suggestion of NDN is to mark special components of the name. For the previous example, assuming the video is bigger than the maximum NDN packet size, the data could be partitioned in multiple packets. As a consequence, a last component is added to each packet to identify individual segments. To identify that a component is a segment number, its first byte is "`%00`" which is then followed by the segment number. For this case, the legible name of the second segment of the video would be "`/br/ufpa/videos/aula1.mp4/%00%02`".

[2] http://www.sigcomm.org/content/acm-conference-information-centric-networking

Figure 2 illustrates the two NDN packets: the interest and data packet. The former is composed of the name and attributes that enable the routing. The data packet is composed of the name of the content it represents, the content itself, and the signature that is used to verify its authenticity.

Figure 2 – Named-data networking packets: the *interest* and the *data*.



Source: Zhang *et al.* [4].

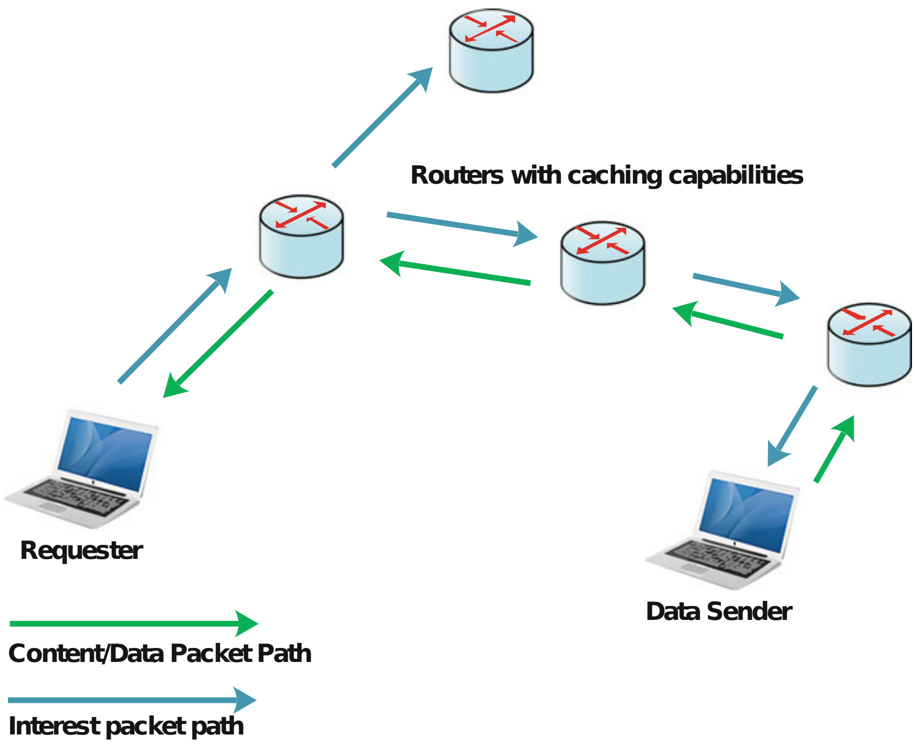
Figure 3 illustrates the data acquisition in ICN. To acquire a data packet, the consumer shall send an interest packet. The network is in charge of routing this packet to a node that is capable of providing the content. This node might either have the content cached, or is capable of generating it, in the latter case, called producer. Once the content is found, it is routed back using the inverse path, reaching the consumer. This data acquisition happens accordingly the main NDN component, the NDN forward daemon (NFD), which will be discussed next.

The NDN router maintains three main data structures: the content store (CS), the pending interest table (PIT), and the forward information base (FIB), wherein the first is used for caching and the others for routing. The CS stores, according to the caching strategy, certain data packets that went through the routers so that it can be used to satisfy future requests. The PIT maintains a list of interest packets that were forwarded by the routers so that it can forward data packets accordingly. Lastly, the FIB is the table that maps the interfaces that can satisfy specific interests.

Figure 4 depicts the primary packet processing for NDN. At the reception of an interest packet, a node first checks if the requested content is present at the CS, according to interest name and other packet fields if present. If the content is in the CS, it is stored in the cache of this node. Therefore, the node can respond directly with the proper data packet.

When the requested data is not in the CS, it is inserted in the PIT, and it is verified if other entries for the same content are already in the PIT. If no request was in the PIT, the interest is forwarded according to FIB. Otherwise, no further actions are performed.

Figure 3 – Data acquisition in an ICN network.



Source: Ahmed *et al.* [42].

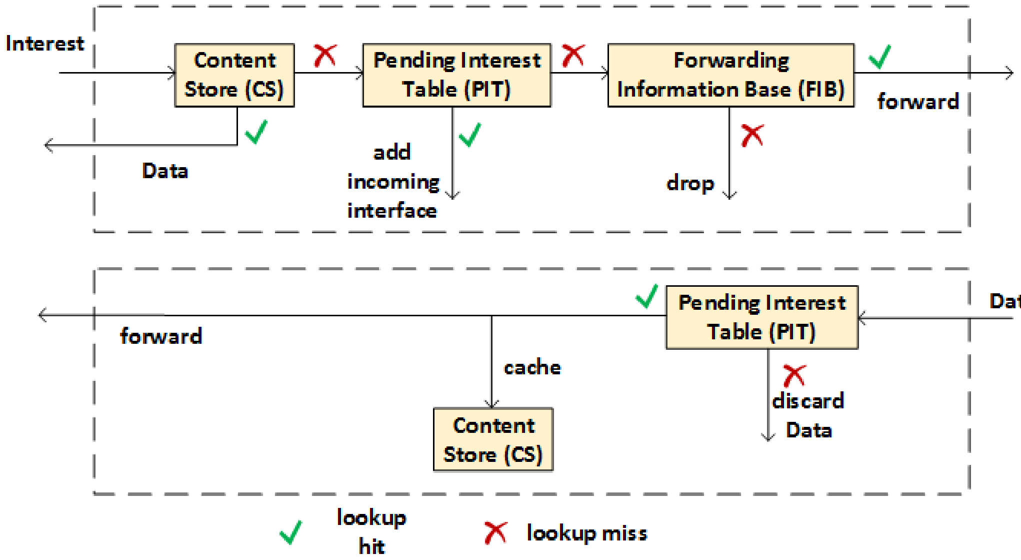
At a data packet reception, if the data matches a request present in the PIT, it is cached in the CS according to the caching strategy, and forwarded to the interface from which the interest arrived, which is given by the PIT. If an interest was not sent for the received data packet, the data is discarded and no further action is performed.

The described NDN architecture enables native caching, given by the CS, and traffic aggregation, by joining requests in the PIT. Those are important properties that might enhance the performance of mobile networks. Section 3.3 highlights these applications.

2.2 Fourth Generation (4G) Mobile Networks

The most common mobile network today are the fourth generation mobile networks (4G) [2]. When planned, they had as requirements for the radio interface a scalable transmission band of up to 50 MHz, latency on the user plane of less than 10 ms, mobility of up to 350 km/h among others [43]. Two of the leading technologies developed to fulfill those requirements were the WiMax and the EPS. We chose to use the EPS for this work because it is, currently, the most widespread mobile network technology [2]. For concreteness, the discussion in this work focuses on 4G, as done by Ravindran *et al.* [15]. In many aspects, the 5G mobile network is an evolution of its predecessor, and the presented discussion can be extended from 4G to 5G given their evolutionary path.

Figure 4 – Representation of the NDN packet processing.



Souce: author [30].

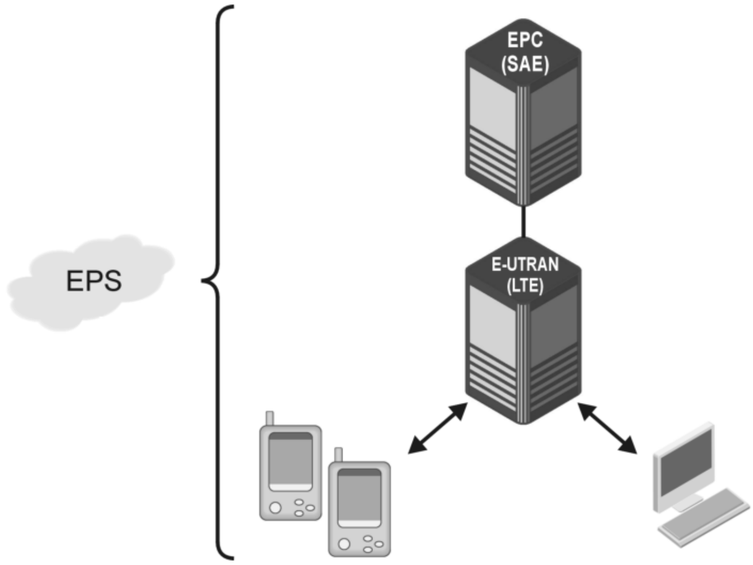
The EPS is a specification standardized by the 3rd Generation Partnership Project (3GPP) representing the evolution of the universal mobile telecommunications service (UMTS). However, this does not mean that the UMTS is not developed anymore; instead, the two standards coexist and are improved together. That is true also for the predecessor of UMTS, the general packet radio service (GPRS).

The coexistence of multiple technologies requires additional elements to interconnect the systems. For simplicity, this work describes only the components required for the EPS network without considering the elements which enable the communication with other technologies.

Multiple names are used to describe the 4G networks, to be specific, in the 3GPP context, the network is basically divided between *access* and *core*. Strictly, LTE comprehends the access network, also called evolved universal terrestrial radio access network (E-UTRAN). In the network core, called EPC, the system architecture evolution (SAE) is standardized. Therefore, the term EPS can be used to encompass the whole system that 3GPP standardizes to fulfill the 4G requirements. Figure 5 illustrates the associations between those definitions.

Figure 6 depicts the main protocols and components of the EPS. The communication is split between *data* and *user* planes, represented by the line and block arrows, respectively. The cylinders represent bidirectional communication channels, except for the tunnels between UE and E-UTRAN node B (eNodeB), where the channels are unidirectional and shared, besides being used in both control and user plane. The anchor component for the system configuration is the mobility management entity (MME). Accordingly, it has

Figure 5 – Illustration of acronyms associations of the 3rd Generation Partnership Project fourth generation mobile network.



Source: Kreher and Gaenger [44].

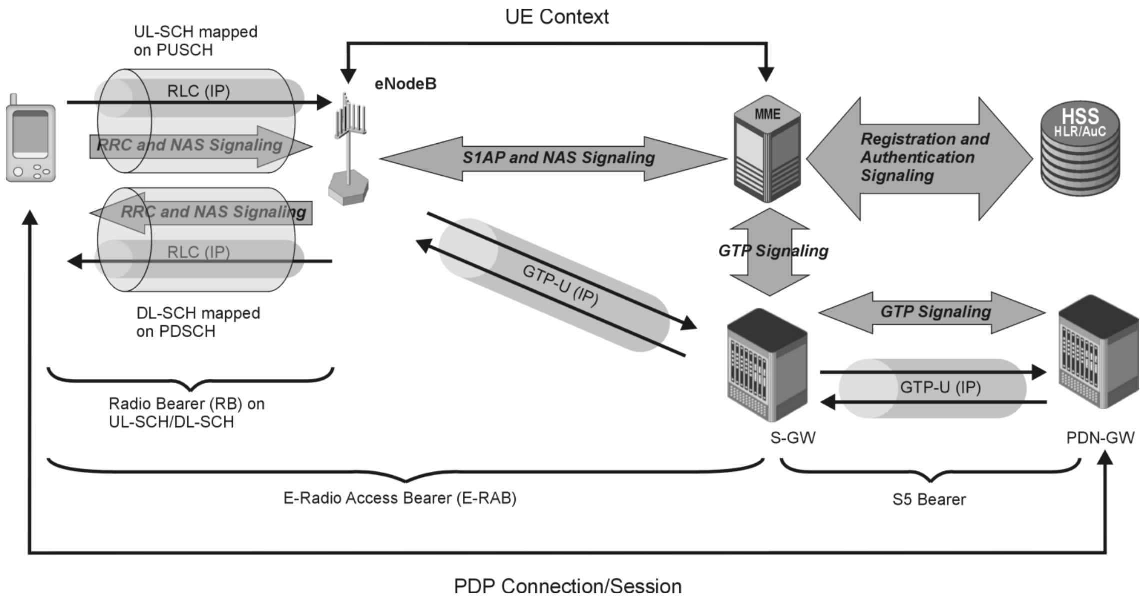
communication channels with almost all the components in the user plane. One example of MME responsibility is the setup of the GPRS tunneling protocol (GTP) tunnel in the user plane (GTP-U) between serving gateway (S-GW) and the eNodeB. The home subscriber server (HSS) is also used only in the user plane. It is responsible for controlling the subscriber access. The S-GW and the packet data network gateway (PDN-GW) are the gateways of the network in charge of processing packets sent and received by the UE in the user plane.

We proceed with the description of the functionalities of each of those components. Moreover, to exemplify a use-case of their responsibilities, the process of user attachment is described in details.

The eNodeB is the access point of the user in the LTE network. Consequently, it is responsible for the transmission and reception of data in the radio interface. One of the crucial responsibility of the eNodeB is the scheduling of the radio channels. The eNodeB uses various parameters to multiplex those channels, such as link quality and user priority and capabilities. In the user plane, the eNodeB actuates as a router of the transport layer. In the control plane, it is responsible for selecting the MME which receives the non-access stratum (NAS) messages. It is important to highlight that security services and fragmentation are implemented at the eNodeB given that the radio channel is insecure.

The MME is responsible for the UE NAS connection. The MME authorizes the UE attachment and stores the user location in the network, which makes the MME responsible for informing the UE location when paging is requested. During the UE attachment, the MME is also responsible for choosing the best S-GW to establish the GTP-U tunnel

Figure 6 – Protocols and components of the evolved packet system (EPS) network.



Source: Kreher and Gaenger [44].

between eNodeB and S-GW. The MME also manages the bearers. Thus, when needed, the UE requests to it the creation of dedicated bearers.

The S-GW operates as an anchor for the user in the network, in other words, when the user moves among eNodeBs, the S-GW buffers the data addressed to the UE until the connection with the new eNodeB is established. The buffering also happens when the UE is in the sleep state given that the packages are only delivered to the user after the paging.

The PDN-GW is the interface between the EPS network and the services offered by the external packet network. The external network might be public (internet) or private. To provide connectivity to different private networks, the UE might be connected to more than one PDN-GW. It is also in the PDN-GW that NAT, billing functions, legal traffic interception and quality of service (QoS) might be implemented.

The HSS maintains a database with the profiles of the authorized subscribers in its network. Thus, it is responsible to provide subscriber authentication services.

The EPS is implemented using two data traffic planes. One for network control and configuration, the control plane. And the other for carrying the subscriber application data, the user plane. Therefore, there are a different protocol stack and elements for each plane, in the next section, these planes and its protocol stacks will be detailed.

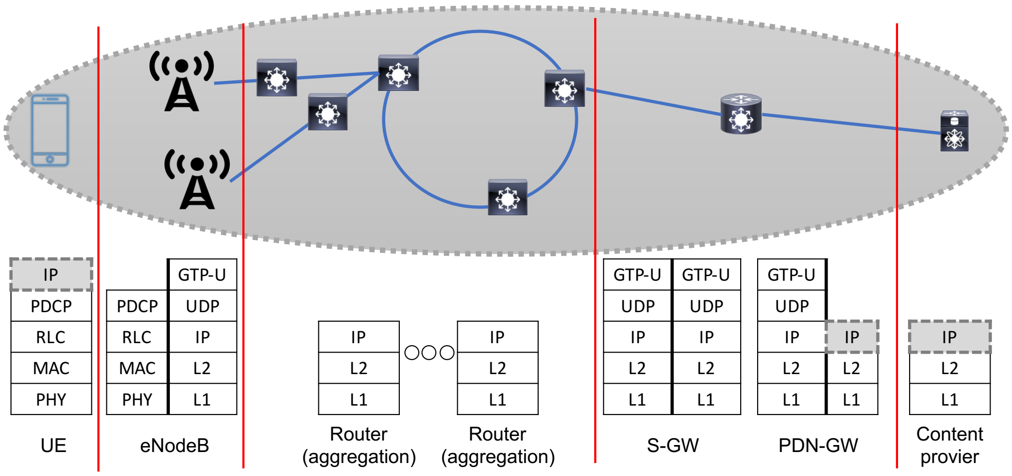
2.2.1 User Plane

The user plane has a goal of providing the communication channel between the UE and some application outside the EPS. Figure 7 depicts a simplified view of the current user plane LTE stack, which works over IP, consisting of a UE, eNodeB, aggregation nodes, S-GW and PDN-GW. The protocols used for data delivery are also exposed.

The communication between UE and eNodeB implements the LTE stack to carry the IP packets (main layer of this work) and contains the LTE physical layer (PHY), medium access control (MAC), radio link control (RLC) and packet data convergence protocol (PDCP). The eNodeB translates the LTE stack to the backhaul's. In the backhaul, the traffic is transported using GTP-U to facilitate carrying IP traffic encapsulated over IP. This tunnel allows, for example, the user to move among eNodeBs while maintaining its original IP.

Multiple technologies can be used in the backhaul, so layer 1 (L1) and layer 2 (L2) in Figure 7 represent any protocol implied by the communication technology. A requirement is that L2 must transport IP. The eNodeB encapsulates the user's IP packet in a GTP-U tunnel carried over the user datagram protocol (UDP). The backhaul contains multiple aggregation nodes that are capable of aggregating traffic, even of multiple operators. The S-GW relays traffic between two GTP tunnels. The PDN-GW finally processes the user's IP packet and delivers it to the content provider. The user's IP packet is represented in Figure 7 with a gray dashed border to emphasize that it is different from the IP in other components. For instance, it is different in the aggregation routers, where the IP is transporting the user's IP encapsulated in a GTP tunnel. These and other GTP features will be described next.

Figure 7 – EPS user plane IP-based architecture.



Source: author [30].

2.2.1.1 General Packet Radio Service Tunneling Protocol

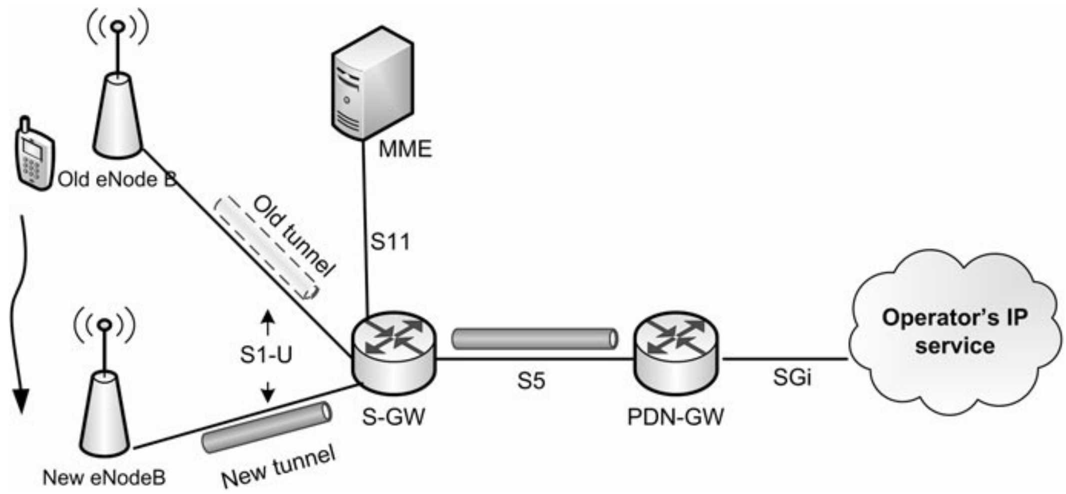
The GPRS tunneling protocol, as the name suggests, arose from GPRS needs, e.g., the management of bearers and user traffic tunneling [45]. The GTP is divided into two components, the control GTP (GTP-C) and the user GTP.

The GTP-C is used to control and manage the GTP-U tunnels, creating the path in which the user data can flow. The GTP-U does the proper transport of user traffic, encapsulating the user IP packet inside another IP packet. Achieving the goal of forwarding the internal IP packet throughout a generic IP network, which the internal IP is not part of, until its destination network.

One of the main GTP properties is the capacity of dividing traffic of one single interface into multiple flows, which might belong to more than one user. With the goal of implementing this functionality, the GTP identifies the flow on the tunnel endpoints by the tunnel endpoint identity (TEID).

Using this strategy, it is possible to prioritize flows and change routes modifying only the tunnels TEID, this management is done by the GTP-C. This kind of management is illustrated by Figure 8, it shows that the UE changed the eNodeB to which it is attached to. The MME is responsible for adapting the network to this change. So, it creates a TEID in the new eNodeB, informing the S-GW TEID. Additionally, it updates the S-GW about the new TEID in the new eNodeB. This way, the UE maintains its configurations and connections of the network layer.

Figure 8 – EPS user mobility example in which the UE IP is maintained, to this end, the network modifies only the tunnel between S-GW and eNodeB.



Source: Ali-Yahiya [46].

The GTP-U header is shown in Figure 9, it can be seen that its structure is simple and that its main component is the destination TEID. Consequently, the GTP-U can, basically, be interpreted as a flow identifier.

It is interesting to note that the IP addressing made by the UE is not used during the forwarding throughout the transport network. What matters in this path are only the bearers. This statement is illustrated in Figure 10. When sending a packet, the UE chooses a bearer to be used. According to it, at the eNodeB, the packet is encapsulated

Figure 9 – GTP-U header. Note: * non-used bit, [1,2,3,4] optional fields.

Bits

Octets	8	7	6	5	4	3	2	1
1	Version			PT	(*)	E	S	PN
2	Message Type							
3	Length (1st Octet)							
4	Length (2nd Octet)							
5	Tunnel Endpoint Identifier (1st Octet)							
6	Tnnel Endpoint Identifier (2nd Octet)							
7	Tunnel Endpoint Identifier (3rd Octet)							
8	Tunnel Endpoint Identifier (4th Octet)							
9	Sequence Number (1st Octet)[1, 4]							
10	Sequence Number (2nd Octet)[1, 4]							
11	N-PDU Number[2, 4]							
12	Next Extension Header Type[3, 4]							

Source: Olsson *et al.* [45].

in a GTP-U tunnel of the interface between eNodeB and S-GW (S1-U). Encapsulated in the GTP-U tunnel, the UE packet becomes a payload. Therefore, in the transport network, its address is not checked anymore, instead, the GTP-U is, which reveals the S-GW address. At the S-GW, the GTP-U of the S1-U is opened and, again, encapsulated in a GTP tunnel, this time for the interface between S-GW and PDN-GW (S5). Similarly, the internal IP is not checked and the traffic is routed according to the bearers. Finally, at the PDN-GW, the packet is opened, the authorization of the flow at the bearer is checked, and the communication with the application server is fulfilled. The path of the packet coming from the internet is similar to the one just described. However, it is executed in the opposite direction. This process is also illustrated in Figure 10. Finishing the overall transmission of user packets in the EPS.

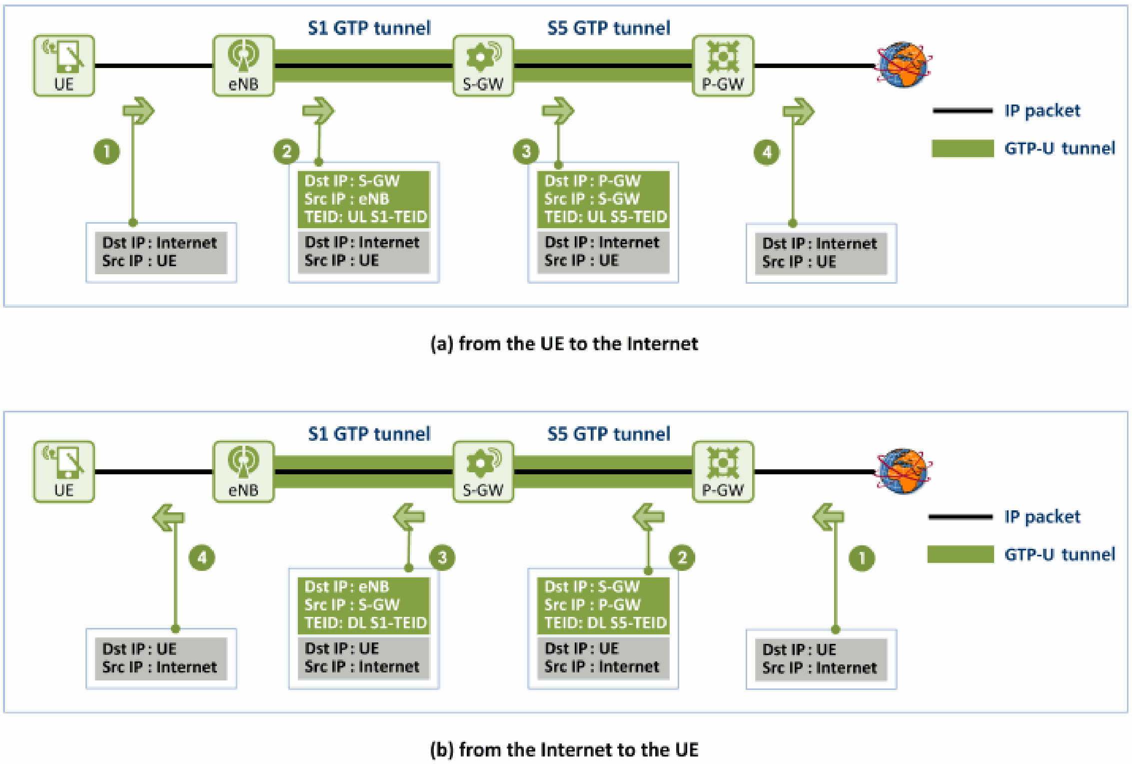
2.2.1.2 Quality of Service

With the goal of guaranteeing a quality of service for different data flows, the EPS defines the concept of bearers. The bearers are defined in layers for each communication interface. Therefore, it possible to decompose the requirements in parameters for each connection. Figure 11 shows one illustration of EPS bearers. The idea is that flows with different characteristics are mapped into different bearers. For example, web browsing should be mapped in a non-guaranteed bitrate (non-GBR) bearer. However, for a voice over IP (VoIP) a bearer with a guaranteed bitrate (GBR) should be used.

2.2.2 Control Plane

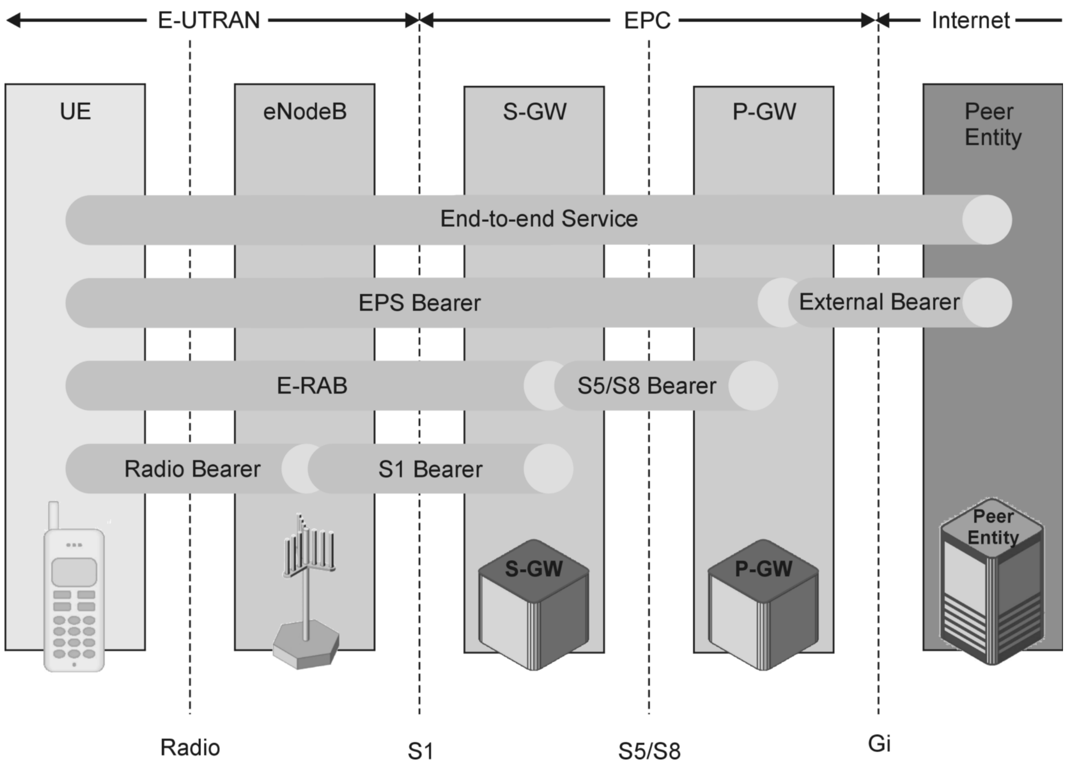
Figure 12 presents the protocols used in the EPS control plane. The control plane manages the user, executing tasks such as attachment of the user to the network and

Figure 10 – Illustration of the addresses used for routing at the EPS user plane.



Source: Cmyoo [47].

Figure 11 – EPS bearer types to configure quality of service in the network.

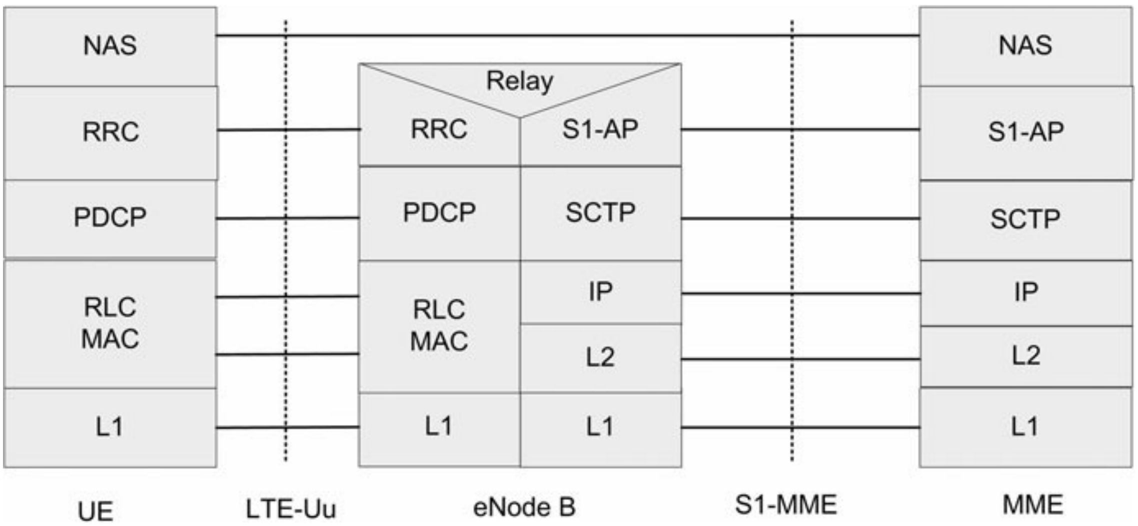


Source: Kreher and Gaenger [44].

configuration of dedicated bearers.

One of the main protocols in the control plane is the non-access stratum. Through it, the UE can communicate with the MME and negotiate various network parameters, from authentication to mobility management. Another important protocol for communication is the radio resource control (RRC), which has, among its responsibilities, the establishment of radio bearers in the E-UTRAN.

Figure 12 – EPS control plane protocol stack.



Souce: Ali-Yahiya [46].

2.2.3 Attachment of a User Equipment to the Evolved Packet System Network

The attachment of a UE to the EPS network implies the use of all the components described in Section 2.2. Thus, describing this procedure can help comprehending the functions of each of those network elements, as well as the establishment of user plane bearers.

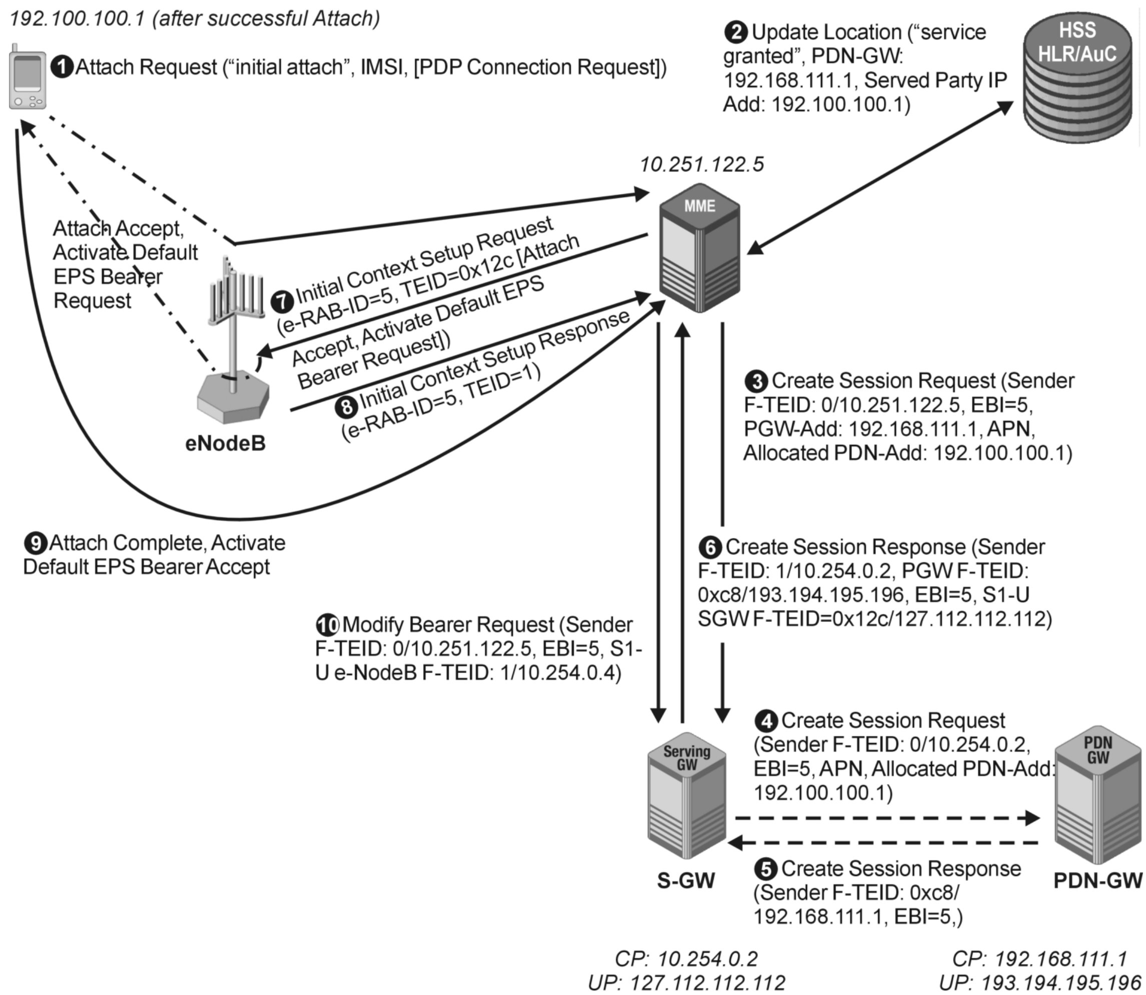
The attachment process starts with the user powering on the UE. This will trigger the attachment procedure at the network. After its completion, the user should be able to access the services offered by the bearer.

During the registering process, done in the control plane, all the network will be prepared to receive the subscriber traffic. Thus, the default bearer will be established, which includes the radio, S1, and S5 bearers. The configuration of those will be concluded before the end of the attachment procedure, delivering a low latency to the service access. The access to the network and the type of quality of service available for the subscriber are controlled by the HSS.

Figure 13 illustrates, step by step, the messages exchanged among the EPS nodes to attach the UE to the network. We detail and discuss each of those steps next.

The attachment procedure, in its complete form, would consider when was the last time that the subscriber used the network; if it would prefer to use other types of access network; if it would be a roaming subscriber; among other options. However, because the objective of this section is to use the attachment to exemplify the use of the network elements, the simplest form of attachment is discussed. This simplification implies the omission of some details, which indicates that the attachment procedure could have additional cases in practice.

Figure 13 – Procedure of UE attachment to the network.



Souce: Kreher and Gaenger [44].

1º step The first message exchanged is the NAS attach request. This message is sent, through the UE and eNodeB, from the universal subscriber identification module (USIM) to the MME. The main information contained in this message are: the identification of the subscriber given as international mobile subscriber identity (IMSI), the packet data network (PDN) connectivity request, and the UE capabilities. The IMSI enables the verification if the subscriber is allowed or not in the network. The PDN connectivity request indicates the intent of exchanging data and will cause the creation of the default bearer. And the UE capabilities announces, among other information, the cryptographic and integrity algorithms supported by the UE.

During this step, the message exchange is not secured. Consequently, information such as subscriber identity and user and password for PDN access are insecurely transmitted and might be intercepted by an intruder.

It is also worth noting that to forward the NAS messages to the MME, the eNodeB informs an identifier, the e-NodeB-UE-S1AP-ID, which identifies the subscriber in the eNodeB. Its counterpart is contained in the first NAS response message sent by the MME, the MME-UE-S1AP-ID, which identifies the subscriber in the MME. These identifiers are used in all the future communication in the interface between eNodeB and MME (S1AP) to refer to the subscriber.

2º step In the second step, the HSS is contacted by the MME to authorize and define parameters of the subscriber attachment. The message exchanged is called update location because the MME stores, at the HSS, the list of eNodeB in which the user might be found (to enable paging).

The HSS also decides which PDN-GW is used and which IP should be assigned to the UE; this information is contained in the MME response. So when this information arrives, the UE is able to configure its gateway immediately.

To define user plane bearers, the HSS informs two maximum throughputs, one should be allocated for the default bearer and the other is used for the PDN. The later has a higher value, it is the sum of the throughput of all the allocated bearers.

3º step In this step, the MME initiates the resource allocation at the user plane. The message to be sent to the S-GW contains the EPS bearer identifier (EBI). The EBI is used to identify the path composed of many tunnels, through which default bearer packets flow.

This procedure creates a tunnel GTP-C between MME and S-GW. This tunnel has as identifier, in the MME side, a fully qualified TEID (F-TEID).

4º step After receiving the message requesting a session creation, the S-GW creates its GTP tunnels. Two GTP-Cs to communicate with the MME and PDN-GW. Plus two GTP-Us to communicate with the eNodeB and PDN-GW, which are used to carry user plane packets to and from the UE.

5º step In this step, the PDN-GW creates two tunnels to communicate with the S-GW. One GTP-U and one GTP-C, in the control and user plane, respectively.

The PDN-GW also confirms the reception of the user data and inform the F-TEID of the tunnel just created to the S-GW.

6º step The message in the sixth step concludes the creation of the user plane tunnels in the EPC. For this, in this sixth message, many F-TEIDs are informed to the MME. The GTP-C, for communication between S-GW and MME, in the control plane. Also, two in the user plane, one for communication between eNodeB and S-GW and another for communication between S-GW and PDN-GW.

After receiving all the identifiers, the MME holds the freedom do freely modify the network elements for the user plane. Hence, for example, in case of congestion, the MME might change the S-GW to which the user is attached.

In the F-TEID of the S-GW and PDN-GW, two IPs are used, one for the control network and another for the user network. Therefore, the two traffic are dissociated.

7º step Once the EPC is prepared to receive the subscriber traffic, the remaining configuration is in the user plane of the E-UTRAN. This step performs this configuration. The message sent to the eNodeB contains the F-TEID of the S-GW to which the eNodeB should forward subscriber data in the user plane. Consequently, user data can now go through the EPS network and go out in the PDN-GW.

Another essential information contained in this step message is the global unique temporary identifier (GUTI) which refers to the UE. With the objective of making the communication private, the GUTI might be used as an identifier, which avoids the transmission of the IMSI in insecure channels. Given that the GUTI, contrary to the IMSI, is a temporary identifier, the user cannot be identified outside of context.

With this information, the eNodeB configures a radio bearer with the UE, sending to it the IP, gateway, and other attributes to be used. The message of default bearer activation indicates that the UE can use the user plane.

8º step After creating the user context and making its configuration, the eNodeB informs, in this eighth step, the MME that the configuration was accepted. Also, the eNodeB sends to the MME the F-TEID that is used, in the user plane, by the S-GW to forward to the eNodeB data destined to the UE.

9º step The message in the ninth step informs the MME that the UE accepted the configuration, and that it is ready to carry user plane data.

10º step After the UE is configured, only one end of the tunnels is open: the address of the eNodeB endpoint in the S1-U. In other words, the S-GW is not informed where to forward messages addressed to the user. Therefore, to close the tunnel, the MME informs, the S-GW, the TEID of the eNodeB to be used. This TEID was informed to the MME at step 8.

This step concludes the configuration of the user plane in the EPS and the user attachment is completed.

2.2.4 Identifiers and Tunnels of the Subscriber Attached to the Network

Once the subscribed is attached to the network according to the procedure described in Section 2.2.3, the system is configured as Figure 14 shows. It illustrates the components and identifiers used to carry subscriber data both at the control and user planes.

For the communication between UE and MME, the subscriber is identified using the GUTI; between MME and HSS using the IMSI; between the eNodeB and MME the UE-S1AP-ID. The remaining communication, in the EPC control plane, between MME, S-GW and PDN-GW, which employs GTP-C tunnels, use the F-TEID.

For the EPC user plane, only GTP-U is used. Thus, all the identifiers are F-TEIDs, which are mapped by the mediation of the bearer associated with the subscriber (EBI).

In the E-UTRAN, the communication between UE and eNodeB uses only one identifier for both user and control plane, the cell radio network temporary identifier (C-RNTI) in the MAC layer. In this case, the division between user and control plane is made by the upper layers.

2.2.5 Evolved Multimedia Broadcast Multicast Service

The evolved multicast broadcast service (E-MBMS) has as objective in the EPS implement the packet delivery from one origin to multiple destinations [48]. This service enables the sharing of network resource among multiple subscribers when they are interested in the same content. The E-MBMS makes available two types of delivery: the broadcast, in which all the users receive the content; and the multicast when users must indicate that they want to receive the content.

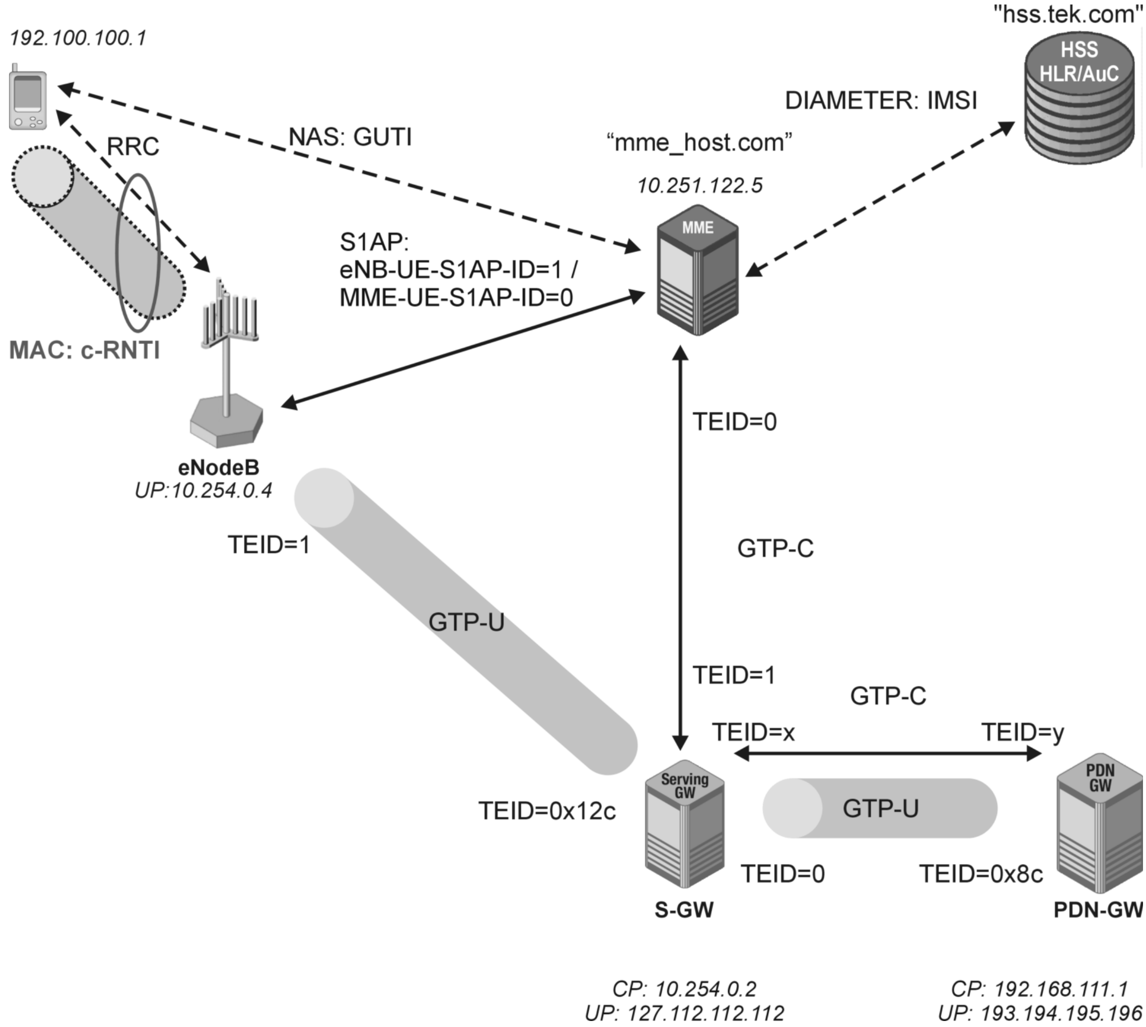
Figure 15 shows the E-MBMS architecture. The components multi-cell/multicast coordination entity (MCE) and multimedia broadcast multicast service gateway (MBMS-GW) are logical entities. Thus, they might or might not be a single physical entity.

The broadcast service is, usually, available in multiple eNodeB [48]. Therefore, resource coordination at the eNodeB is necessary, so, for example, they can transmit at the same frequency and time. This coordination is the kind of responsibility assigned to the MCE.

The MBMS-GW is responsible for sending data flow, in the user plane, for the eNodeB. Furthermore, the broadcast multicast service center (BM-SC) controls the authentication and the network flows.

It is important to highlight that finite resources are shared among users in the E-MBMS, e.g., in the E-UTRAN, the electromagnetic spectrum is shared. Therefore, the same content can be distributed at the same time for as many users as necessary, similarly to what happens in digital television. This broadcast strategy decrease costs and is highly

Figure 14 – User and control planes connection identifiers when the subscriber is attached to the EPS.



Source: Kreher and Gaenger [44].

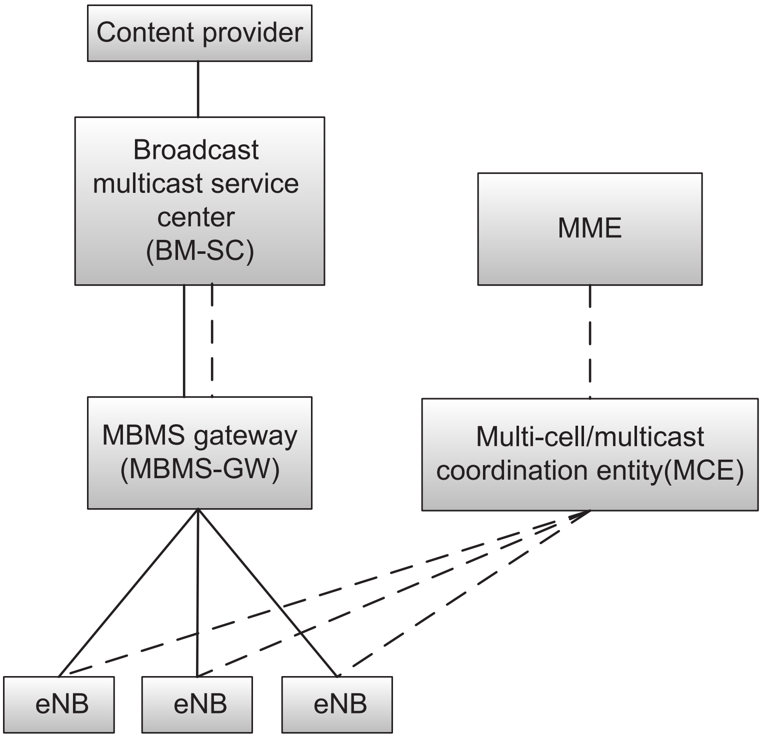
scalable. However, E-MBMS requires coordination between service and network provider and is not yet widely deployed.

The described EPS components and architecture are not a holistic view of the system, which would be outside of the scope of this work. However, the presented material contains the important information for the development of the testbed and the integration of ICN into mobile networks, which will be discussed in Section 3.

2.3 Reinforcement Learning

Machine learning (ML) has been successful in many applications, including network management, for example, in traffic prediction [49]. Many of those applications are in the class of supervised learning, i.e., where a dataset is needed, from which the ML algorithm learns from.

Figure 15 – Evolved multimedia broadcast multicast service (E-MBMS) architecture.



Souce: Ahmadi [48].

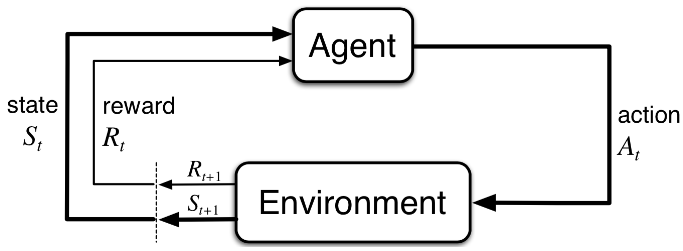
Another class of ML that has increasing popularity is reinforcement learning (RL). RL has raised much interest because it is a learning method which does not rely on labeled data, it can, instead, interact with an environment and learn directly from practice, or trial and error. RL is particularly interesting for network management where the network, an environment, might be too complex and require human experts to operate, which are expensive and not highly scalable. In this section, we describe the RL algorithm principles that were applied to the slice admission problem in Section 4.

2.3.1 Agent and Environment Interaction

Figure 16 shows a diagram of the agent environment interaction. The objective is to train an agent to control the environment. We set the objective of the agent by the reward signal, i.e., the agent should take actions that maximize its reward collection. The agent receives as input a state S_t and based on that it should decide on an action A_t, which then takes the environment to a new state S_{t+1} and the environment produces a reward $R_t + 1$. We refer to the sets of possible states and possible actions as $\mathcal{S}$ and $\mathcal{A}$, respectively.

A naive example might help to illustrate this concept. Let us assume that an InP has a link with capacity 2 Mbps, and tenants request slices that impose traffic of 1 Mbps over this link. If the InP accepts the slice in the network and provides enough capacity for

Figure 16 – Agent and environment interaction signals for reinforcement learning, or Markov decision process.



Souce: Sutton and Barto [50].

it to run over its lifetime, a revenue of 1 currency is earned. However, if for some reason the resources were not available, a penalty of 1 currency is paid by the InP (-1 revenue). The slices live in the network for 11 hours. Moreover, tenants request new slices every 4 hours and the agent is evaluated based on the revenue achieved by the end of a day, which is called an episode. The problem then is to design an agent for the InP that is capable of deciding which slices to accept so that the revenue is maximized.

Modeling this example with RL might lead to an agent that can receive as state the number of slices inside the system and can take two decisions, either accept or reject the new slice. Table 1 presents the recorded track (trajectory) of an agent and environment interaction. For this interaction the agent decisions were taken randomly, with the same probability of accepting or rejecting the slice. The trajectory shows that the agent accepted the slice at $t = 0$ when there was no slice at the system. At $t = 1$ another slice arrived, which was also accepted by the system, which now has two slices. At $t = 2$, another slice arrived and was accepted. At time $t = 3$, a slice was rejected, and a reward $R_3 = 1$ was received because the first slice left between $t = 2$ and $t = 3$. At $t = 4$ a slice is rejected, and the second slice left between $t = 3$ and $t = 4$, thus $R_4 = 1$. The slice received at $t = 5$ is rejected, and $R_5 = -1$ given by the leave of the third slice. This reward is a penalty because between real-time 8 and 11 the first and second slices were using all the resources, leaving none for the third slice.

Based on trajectories like the one presented in Table 1, reinforcement learning algorithms can understand what actions maximize rewards. We proceed defining concepts that are used for some of such algorithms, and then present one of them.

2.3.2 Reinforcement Learning Components

RL algorithms rely on some concepts to learn the task. We present those concepts while using the slice admission problem to illustrate. One is the *policy*, which is responsible for deciding which action to take given a state. It is represented by $\pi(a|s)$ where a is an action and s is a state, i.e., the probability of taking action a in state s. This function

Table 1 – Example trajectory of agent interaction with the naive slice admission environment using a random policy.

Real time	Agent time (t)	S_t	A_t	R_t
0	0	0	Accept	-
4	1	1	Accept	0
8	2	2	Accept	0
12	3	2	Reject	1
16	4	1	Reject	1
20	5	0	Reject	-1
-	6	-	-	0

Source: author.

might be estimated by a parameterized function. For example, an artificial neural network (ANN). Therefore, to represent the parameters explicitly, we can use $\pi(a|s, \boldsymbol{\theta})$ so that $\boldsymbol{\theta}$ is the parameter vector.

For the defined slice admission problem, the optimal policy π_*, obtained empirically, is $\pi(\text{Reject}|s) = 1$ for $s > 2$ and $\pi(\text{Reject}|s) = 0$ otherwise which implies $\pi(\text{Accept}|s) = 0$ for $s > 2$ and $\pi(\text{Accept}|s) = 1$ otherwise. For this trivial example that should be ease to observe because it is rentable to accept slices whenever there is enough resources, but they result in a loss of revenue when there is no resource.

Another concept is the *return*, defined as $G_t = \sum_{k=t+1}^{T} R_k$. G_0 represents the performance of an agent in an episode. The return can be used to compare consequences of actions, R_t measure the immediate reward after taking action A_{t-1}. However, the consequences of A_{t-1} might propagate further, for example, the slices admitted in the naive environment stay for multiple time steps. Table 1 shows that action A_0 was responsible for the reward at R_3 and used resources that could not be used by the third slice. Consequently, the third slice caused the penalty at R_5. Thus, when evaluating actions, long-term consequences have to be measured with G_t, instead of R_t.

To measure how good it is to be in a state, the *value function*, $v_\pi(s) = \mathbb{E}_\pi\left[G_t|S_t = s\right]$, is defined. It gives us what is the expected return if the agent is in state s. However, it is important to note that it is defined for a specific policy π because the rewards received in the future depend on future decisions. The value function allow the agent to decide where it is better to be in, though, it does not say anything about how to go there. Hence, another function is defined, the *action-value function*, $q_\pi(s, a) = \mathbb{E}_\pi\left[G_t|S_t = s, A_t = a\right]$, which gives the expected value of the return when at state s take action a and thereafter follow policy π.

With those definitions, we can have an agent that controls the environment to the best of its ability. It just needs a policy to decide the actions. A common one is the greedy policy, which is defined based on the action-value function, i.e., the best action to take is the one that maximizes the expected return $\pi(s) = \arg\max_a q_\pi(s, a)$.

We can then understand reinforcement learning algorithms as different forms of learning a policy, or, in some cases, learning the action-value function from which we can derive the best policy. For example, *Q-learning* estimates the action-value function [50]. And, in Q-learning, when the action-value function is represented by an ANN, the algorithm is known as *deep Q-learning* (DQN) [51]. *Policy gradient* methods apply gradient ascent algorithms to optimize the policy functions. The intuition is that it observes past actions and increase the probability of actions that led to good rewards and decrease the probability of ones that led to bad rewards [52]. Furthermore, some policy gradient methods use an estimation of value function to help defining what was a good action, those are the *actor-critic methods* [53]. This categorization is broad and is not a comprehensive list of recent techniques. However, RL is a hot area, and the literature has been growing considerably; still, this represents the basics of recent advances.

2.3.3 First-Visit Monte Carlo Control Algorithm

To exemplify how a policy estimation might work, we describe the first-visit Monte Carlo control [50] algorithm (Algorithm 1). For this algorithm, we use the ϵ-soft policy. It is defined as a policy which prefers the greedy action with probability $1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|}$ and each other options is given probability $\frac{\epsilon}{|\mathcal{A}|}$. We use this policy because the algorithm estimates the action-value function by averaging all the returns obtained when the same action and state were visited. Thus, using a non-deterministic non-greedy policy guarantees that every possibility has the change of being tried.

In summary, in each iteration, Algorithm 1 generates a trajectory (agent and environment interaction until the end of an episode), and from the collected trajectories the agent estimates an action-value function, and using the estimated action-value function it updates its policy by increasing the chance of taking decisions that increase the expected return.

To exemplify, we apply the algorithm to estimate an agent for the naive slice admission environment defined earlier. We start by setting $\epsilon = 0.3$, $Q(s, a) = 0$, $\pi(\text{Accept}|s) = 0.85$ and $\pi(\text{Reject}|s) = 0.15$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. We also arbitrarily decided that in case of the estimation of *Accept* and *Reject* have the same value, we set higher probability to *Accept*. Then we collect a trajectory and show in Table 2. From which we generate the $Returns(s, a)$ list (Table 3), estimate the new action-value function shown in Table 4 and the new policy in Table 5. As indicated by Algorithm 1.

As might be expected, not much conclusion can be drawn from one iteration of the algorithm. However, we can start seeing how the action-value is starting to be estimated, and how important the samples in the $Returns(S_t, A_t)$ list are. Luckily, at one of the most

Algorithm 1 First visit Monte Carlo control algorithm for estimating an optimal policy.

```
π ← an arbitrary ε-soft policy
Q(s, a) ∈ ℝ (arbitrarily)
Returns(s, a) ← empty list
loop
    Generate an episode following π: S_0, A_0, R_1, ..., S_{T−1}, A_{T−1}, R_T
    G ← 0
    for t ← T − 1, T − 2, ..., 0 do                          ▷ Each step of the episode
        G ← G + R_{t+1}
        if the pair S_t, A_t do not appear in S_0, A_0, S_1, A_1, ..., S_{t−1}, A_{t−1} then
            Append G to Returns(S_t, A_t)
            Q(S_t, A_t) ← average(Returns(S_t, A_t))
            A* ← arg max_a Q(S_t, a)
            for a ∈ 𝒜 do
                if a = A* then
                    π(a|S_t) ← 1 − ε + ε/|𝒜|
                else
                    π(a|S_t) ← ε/|𝒜|
                end if
            end for
        end if
    end for
end loop
```

Source: adapted from Sutton and Barto [50].

critical states, at $t = 3$, besides having a higher probability of accepting the request, the agent rejected it, which was a decision in line with the optimal policy. Therefore, this is reflected in the estimated action-value function, that now predicts a higher value for $S_t = 2$ and $A_t =$ Reject, changing the starting policy, which preferred accepting in that case.

Table 2 – Interaction with the naive slice admission environment using the policy $\pi(\text{Accept}|s) = 0.85$ and $\pi(\text{Reject}|s) = 0.15$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ for the first iteration of Algorithm 1.

Real time	Agent time (t)	S_t	A_t	R_t
0	0	0	Reject	-
4	1	0	Accept	0
8	2	1	Accept	0
12	3	2	Reject	0
16	4	1	Accept	1
20	5	1	Accept	1
-	6	-	-	2

Source: author.

We show the process for another iteration of the algorithm. Table 6 shows the

Table 3 – $Returns(S_t, A_t)$ list in the first iteration (Table 2) of Algorithm 1 for the naive slice admission environment.

S_t	A_t	$Returns(S_t, A_t)$
0	Reject	4
0	Accept	4
1	Accept	4
2	Reject	4

Source: author.

Table 4 – Action-value function estimated in the first iteration of Algorithm 1 for the naive slice admission environment. The estimation is based on Table 3.

S_t	A_t	$Q(S_t, A_t)$
0	Reject	4
0	Accept	4
1	Accept	4
2	Reject	4
Otherwise		0

Source: author.

Table 5 – Policy estimated in the first iteration of Algorithm 1 for the naive slice admission environment (based on the action-value function shown in Table 4).

S_t	A_t	$\pi(A_t \vert S_t)$
0	Reject	0.15
0	Accept	0.85
1	Accept	0.85
1	Reject	0.15
2	Reject	0.85
2	Accept	0.15

Source: author.

new interaction with the environment, which is now using the new version of the policy, i.e., it is less likely to accept a slice at state two. The new version of the $Returns(S_t, A_t)$ is shown at Table 7. We can observe that as the algorithm runs, the agent collects new experiences and being a stochastic policy, the agent explore new state and action pairs. For instance, rejecting at state one, which has low probability because it is not expected to give high returns, but is still tried. Another observation is that taking the same decision at the same state does not cause the same return, as the return is dependent on other decisions. From this data, a new version of the action-value function estimation is collected and shown in Table 8. It accounts for the new experiences and now predicts the same return for accepting or rejecting at state one. This draw could have changed the policy. However, because we defined that in case of a tie we choose accept, the policy stays the same as in Table 5 for the next iteration.

Table 6 – Interaction with the naive slice admission environment using the policy shown in Table 5 for the second iteration of Algorithm 1.

Real time	Agent time (t)	S_t	A_t	R_t
0	0	0	Accept	-
4	1	1	Reject	0
8	2	1	Accept	0
12	3	1	Accept	1
16	4	2	Reject	0
20	5	1	Accept	1
-	6	-	-	2

Source: author.

Table 7 – $Returns(S_t, A_t)$ list in the second iteration of Algorithm 1 for the naive slice admission environment. Collected from Tables 2 and 6.

S_t	A_t	$Returns(S_t, A_t)$
0	Reject	4
0	Accept	4
1	Accept	4
2	Reject	4
1	Accept	4
2	Reject	3
1	Reject	4
0	Accept	4

Source: author.

Table 8 – Action-value function estimated in the second iteration of Algorithm 1 for the naive slice admission environment (based on action-value function shown in Table 8).

S_t	A_t	$Q(S_t, A_t)$
0	Reject	4
0	Accept	4
1	Accept	4
2	Reject	3.5
1	Reject	4
Otherwise		0

Source: author.

Following the algorithm should be straight forward. We show the action-value function estimation obtained at the tenth iteration in Table 9. We can observe that the estimation is becoming more fine-grained, for example, rejecting at states zero and one lead to low expected returns. We also see that the agent, at least once, tried to accept a slice at state two, which is a decision that leads to penalty. This penalty is reflected in the action-value function. Even though this is a bad decision, the agent cannot know that beforehand. Therefore, it is good that it tried it and now sets a low priority to such decision.

The policy derived from the action-value function in Table 9 is still the same presented in Table 5, and, as we presented earlier, it is not the optimal policy. The reason for that is that the reinforcement learning agent behavior might be divided between training and testing (or deploying). So, during training, we do not want to set a determinist policy, otherwise the agent would never observe some actions. Thus, not know if they are good or bad. However, during testing (or deploying) the policy might be adjusted, and, if we want the agent to take only the best decisions it knows, we can set $\epsilon = 0$ so that the agent always take the greedy action and maximizes its reward (leading to the optimal policy in this case). This training and testing division is valid if we know that the environment dynamics is not changing. If it is, the agent might have to continue exploring and training, and sometimes perform suboptimally, so that it captures the new environment dynamics.

Table 9 – Action-value function estimated in the tenth iteration of Algorithm 1 for the naive slice admission environment.

S_t	A_t	$Q(S_t, A_t)$
0	Reject	2
0	Accept	3.56
1	Accept	3.56
2	Reject	3.57
2	Accept	1
1	Reject	3
Otherwise		0

Source: author.

The examples and algorithms described provide an overview and intuition on how reinforcement learning algorithms work. However, as we stated, this is a naive environment, and that is why we were able to provide a simple algorithm to solve it. In practice many adaptations have to be done. The same slice admission problem is studied in Chapter 4. However, the slice admission environment used for the simulations are more complex because it becomes closer to reality. Therefore, it does not have only one resource, but instead it has multiple resources, of different types, which are located at different places. Furthermore, the slices do not arrive at a determinist rate, instead, they have stochastic arrival time, as well as stochastic duration and resource utilization. These characteristics increase the complexity, for example, in representing the state of the system, which if done in tables as described in this section it would increase so much as to become impracticable. Modern reinforcement learning algorithms overcome this by replacing the functions representations from tables to ANNs. However, the mechanisms and intuition for more generic techniques are the same, and having this basic knowledge helps when configuring the reinforcement learning agents for the specific application.

3 ICN for Radio Access Network

The chapter defines ICN in the radio access network, discuss its advantages and associated challenges. Previously presented deployments are contrasted with the one adopted for the implemented testbed. Furthermore, LTE features are described regarding issues in LTE that are of importance to ICN and this work. The discussion covers proposals to overcome those issues based on IP, but most of them are less efficient than ICN-RAN in 5G scenarios, being it because of scalability of deployment or processing. Moreover, ICN-RAN issues and challenges are highlighted.

3.1 Content Delivery in Current Mobile Networks

It is useful to describe how content delivery in mobile networks happens currently to understand how ICN can improve it. So we present a quick summary of some properties of LTE content delivery. In the LTE architecture, the UE maintains the IP assigned to it until it detaches or moves to another network [12]. This aspect will be discussed given its importance in this work. The network here is interpreted as an access point name (APN) connection, and the UE has an IP for each APN it is connected to. To carry the traffic, each APN connection is made through a default bearer [54]. Dedicated bearers can be set up on each APN to prioritize traffic. However, no additional IP is needed because the dedicated bearer is associated with the default bearer and IP filtering does the traffic prioritization. The bearer to each flow is selected according to the traffic flow template (filter) specified during the dedicated bearer establishment.

During the attachment, the UE is provided with a connection to one APN. Standard applications use this APN, yet additional APN connections or dedicated bearers are set up by specialized applications. Hence, for this discussion, we are assuming that the IP maintained by the UE is the one corresponding to its APN connection at the attach time. It is also important to note that an APN can contain multiple eNodeBs. Therefore, as the UE moves among eNodeBs, it does not necessarily change its IP address.

However, when the UE IP address does have to change, anchoring and tunneling functions must be activated to manage the IP address. For instance, mobile IP solution relies on tunneling between the UE and a home network node, which serve as an anchor point to reach the UE. These mobility functions for IP connections imply in reduced flexibility and needs a dedicated control infrastructure [55]. Moreover, regardless of mobility, the IP connection assigned to the UE is carried using GTP tunnels in the LTE backhaul. One consequence of using GTP tunnels to carry UE traffic from the eNodeB to the PDN-GW is that the UE packet is not processed until it reaches the S-GW. Such encapsulation challenges the deployment of edge applications, such as caching near the access network.

Another consequence of the GTP tunnel structure is the unicast nature of information flow in the backhaul. Unicast is especially inefficient for popular content, i.e., when the users access the same data, redundant streams of this content travel through the entire backhaul, even if the users accessed the data at the same time. In this case, the backhaul usage scales linearly with the user's data streams, even if the streams are the same [56]. Still, this inefficiency accompanies a protocol overhead as high as 53% [12].

Another fact against unicast delivery of popular content in mobile networks is that, typically, an operator's core network occupies a large area and can be located, e.g., 350 km away from the UE [57]. Along this path, data from multiple sites share the infrastructure, even from different operators, but data streams are still unicast. Using this architecture, typically, the closest to the user that a potential service provider can position the content is in the core network.

Depending on their business models, the deployment of applications and content in the core network may be a good solution from the perspective of some companies, given the core network reaches many users. For example, content providers currently deploy multiple servers close to core networks (PDN-GW) around the world in the so-called content delivery networks (CDNs) [58], [59]. However, especially for network operators, it is particularly beneficial to have efficient alternatives to GTP tunneling that support more efficient media distribution. The ICN deployment presented in this work is one promising alternative.

Other solutions to intercept GTP traffic and deploy services closer to the users having been proposed in the literature. For example, Garcia-Perez and Merino [60] developed a device that intercepts packets in the GTP tunnel and routes them locally for defined applications. Nevertheless, this solution is not straightforward to implement when the network operator carries traffic using IP security (IPsec), which is often the case given the shared transport infrastructure [12]. Furthermore, transmission control protocol (TCP) and IP protocols do not enable cache functionality in the transport (network) layer, which leads to the use of sophisticated application traffic optimizers [61].

Even if the solution reroutes the GTP tunnel, as in Garcia-Perez and Merino [60], and deploys CDN servers closer to the user, as in Ren *et al.* [58], the connection-oriented nature of content distribution still implies in redundancy [17], i.e., for CDN servers to acquire the content and also when multiple users are receiving the content from the same server.

Problems with this model also arise when user mobility is introduced. Although transparent to the content provider, mobility support comes with significant costs for the network operator, given the connection-oriented approach of the IP-based solution. Such approach requires the reconfiguration of network tunnels in the control plane [62] even if CDN servers are located at the network edge. The control overhead is manageable in current networks, but in future networks, with the massive introduction of small cells, the overhead to control mobility can increase significantly.

The massive deployment of CDN servers is also a challenge, and the management of them is complicated, which could be affordable only by big content providers. Besides, routing is based on domain name system (DNS) queries, which is not optimal and not always select the server closer to the client [61]. From the security perspective, large-scale deployment of servers requires a larger trust chain, which could be hard to maintain, and a breach of security keys could expose user's data [61].

Those are some of the difficulties of current mobile networks (the literature provides more details of those and other difficulties [17], [56], [63]) which would be amended by the adoption of ICN-RAN. The details of ICN-RAN and its main benefits are discussed in Section 3.3.

3.2 Related Work

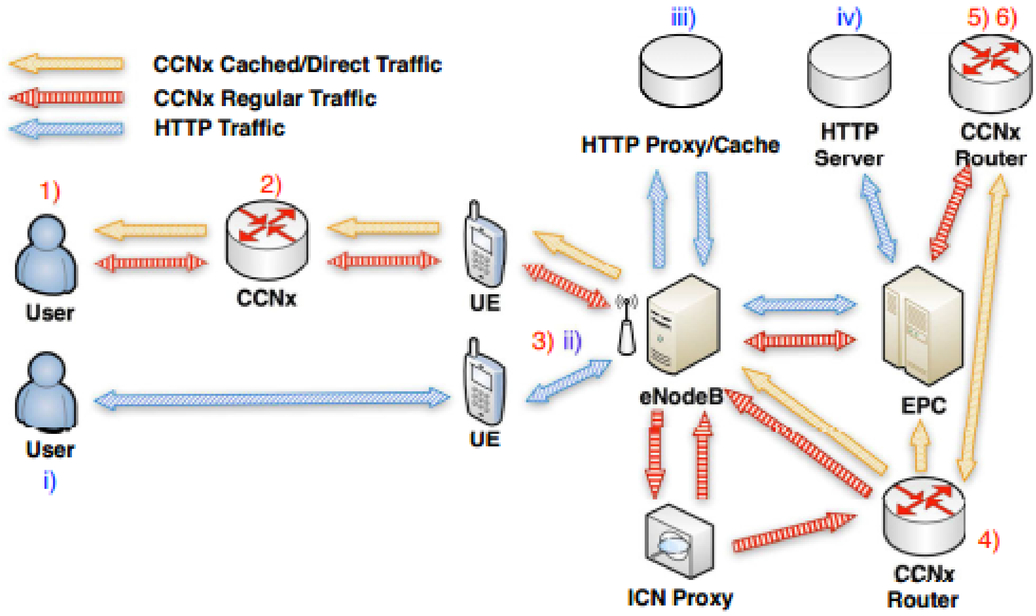
ICN and EPS integration has been an active topic in the literature. Hence, this section gives an introduction of related work. Also, Section 3.5 gives more details of the architecture of some of them, specially about their connection to this thesis.

Gomes and Braun [13] proposes an architecture in which the eNodeB intercepts IP packets and forward them to an ICN proxy, Figure 17 represents this. It shows that depending on the type of packet the proxy might process it if it is an ICN packet, or convert it to an ICN packet when it is a hypertext transmission protocol (HTTP) request. After this process, the proxy might answer locally if the content is available in its cache, or forward the interest to be fulfilled. For complete compatibility with the EPS standard, additional interfaces have to be defined. Therefore, communication with the EPC is enabled, to allow, for example, billing and legal traffic interception, as in Cisco [64].

Figure 17 also exemplify the flows of the proposal, the red and orange arrows illustrate the native ICN flow, and the blue illustrates the HTTP/IP. In the eNodeB, the interest might be satisfied in three ways: the ICN proxy forwards it to the eNodeB, where it finds one ICN router after going through the S-GW. In the second case, where there exists a parallel ICN network, the proxy forwards the packet without going through the EPC. In the last case, the interest is satisfied in the proxy's cache.

One drawback of the Gomes and Braun [13] proposal is that the cache is HTTP, and the strategy is not compatible with secure HTTP (HTTPS). This non-compatibility makes the solution not readily usable because, when compared with HTTP, the HTTPS

Figure 17 – Data flow in the proposal from Gomes and Braun [13].



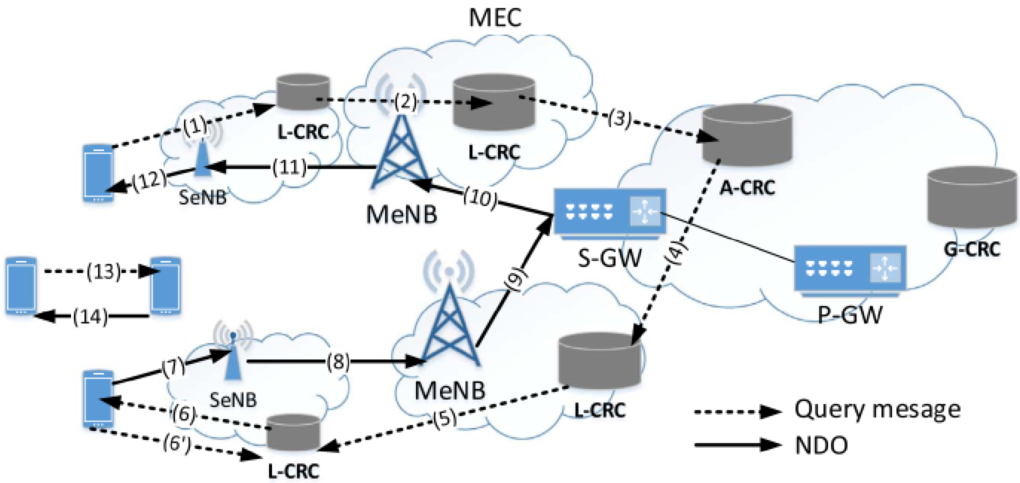
Source: Gomes and Braun [13].

traffic corresponds to 50% of the total. Moreover, HTTPS adoption is predicted to grow because it offers more security when compared to HTTP [65].

Kim [14] discusses the integration of ICN in the EPS network in the context of the UE communication. The author argue that one of the main challenges is the maintenance of routes when the UE acting as a content producer moves among eNodeBs. Kim [14] proposes a new EPS network component to handle this problem, the content routing controller (CRC), responsible for maintaining the routes and forwarding the interests to the content producer. A new signaling radio bearer (SRB) realizes the communication between UE and CRC.

Figure 18 shows two examples of CRC interactions, the user plane components of a standard EPS network are represented in blue, and the newly proposed components in gray. The CRCs are divided into local (L-CRC), aggregation (A-CRC) and global (G-CRC). The simplest communication happens in the device to device (D2D) case, in which the producer publishes its content directly to the consumer (13)-(14). In the second example, when D2D is not possible, the consumer makes the request to the L-CRC (1), which forward the interest until the A-CRC (2)-(3), that then direct the interest to the local producer network (4)-(6). A confirmation is sent to the L-CRC (6') to guarantee that the producer is in the right eNodeB. Finally, the content is sent to the consumer through the user plane of the standard EPS network (7)-(12), finishing the exchange. For the described example, the G-CRC was not used, but it is required in cases when the communicating UE are not associated with the same A-CRC.

Figure 18 – Data flow in the proposal made by Kim [14].



Source: Kim [14].

Suthar and Stolic [12] discusses many aspects of the integration between ICN and EPS. Their proposal is the inclusion of a forward agent in the UE, enabling, for the application, the option to use either ICN or IP, Figure 19 illustrates this idea. Given that ICN is enabled at the UE by the forwarding agent, Suthar and Stolic [12] discuss the consequences of using ICN routers in each EPS node. Figure 19 highlights the case in which the ICN packet is de-encapsulated only in the P-GW. However, the authors argue that the ICN packet might be processed as close to the user as in the eNodeB.

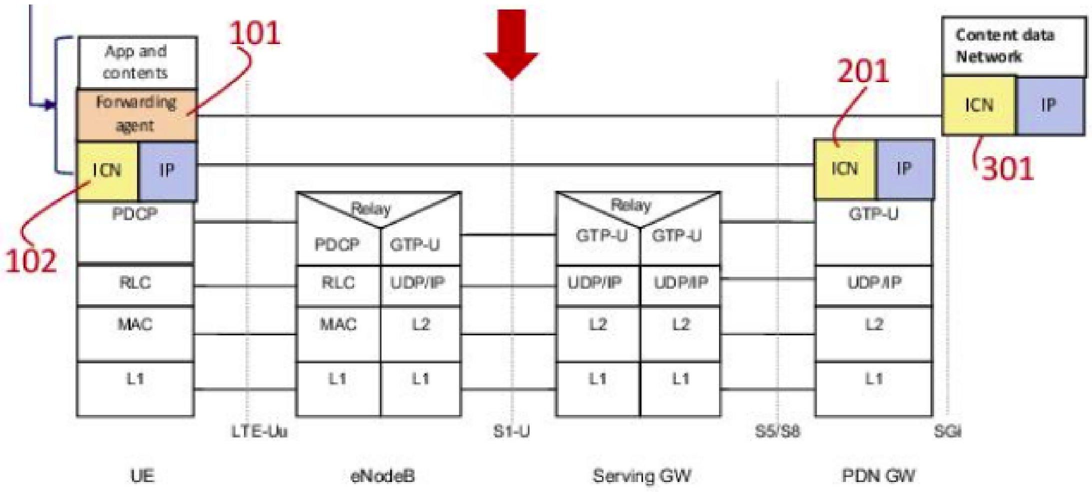
In Suthar and Stolic [12], the implementation is not discussed. Instead, they discuss the implications of using ICN in mobile networks. For example, they suggest that ICN should not be used to replace IP, but instead coexist with it. Another conclusion was that IP should not be replaced in the control plane because the messages are unique. Thus, the use of ICN is not necessary since its main benefits, such as caching and aggregation, would not be used.

3.3 Advantages of ICN with Focus on ICN-RAN

As discussed, the use of name-based data packet forwarding, PIT, and CS represent relevant aspects of the ICN paradigm. Name-based data packet forwarding does not need to keep information from end-users to forward data. It only needs to know the interface from which the interest arrived. This operation simplifies content delivery since there is no need for end-to-end connections between nodes, making mobility, at first, easily performed. For instance, a user in handover, from the NDN layer perspective, would only need to send interest packets to its new attached cell, without any need for connections [11].

Furthermore, one of the uses for the PIT is *interest aggregation*, i.e., when a request

Figure 19 – Components and protocols of the EPS user plane proposed by Suthar and Stolic [12], the new elements are highlighted.



Source: Suthar and Stolic [12].

for a content was already made, and the node is waiting for the data packet, new requests for the same content during this period do not need to be forwarded, but only inserted in the PIT. When the proper data packet arrives, it is sent to all the recorded interfaces in the PIT, working in a broadcast fashion natively. This feature can reduce bottlenecks and be more supportive of flashing crowd events and other scenarios with a large number of requests for the same content [55]. In summary, in broadcast events, instead of the usual setup of additional LTE components [66], same interests would simply be aggregated, and the data packet delivered to the proper users.

Lastly, the inclusion of CS in packet processing implies that every node in the network is able to cache contents, which can reduce delay for content retrieval. Furthermore, it also enables the coupling with other promising features of future mobile networks, such as MEC [11], [67]. MEC is a relatively new paradigm that provides ultra-low latency applications to users, implementing cache and computing power as close to the user as possible.

Regarding Figure 7, ICN in RAN concerns the UE and the eNodeB. This scheme enables improvement in network performance with the use of forwarding strategies, which in ICN context, gain new dimension as content replicas may be obtained from distinct paths [68]. In this manner, many forwarding strategies are proposed, e.g., to reduce service delay for end users [69] or benefit from content replicas to choose the shortest request path [70]. In RAN, forwarding strategies can be used to arrange load balancing between wireless links, resulting in better content delivery and resource usage [21].

Another important ICN-RAN feature is the possibility to cache contents at the eNodeB or the UE itself. With cache at the UE, not only the proper user could benefit from it, but in a scenario with D2D communication, other users could also be served, alleviating the RAN load. Cache at the eNodeB can also be an advantage, but mostly with proactive caching. Elayoubi and Roberts [71] carefully assessed that the trade-off between the cache memory and bit rate led to the conclusion that a minor impact of the reactive cache can be obtained at the network edge. The results suggest further investigation of proactive cache scenarios for base stations. In this manner, most of the gains from cache are actually obtained when adopting ICN in the whole network. Carofiglio *et al.* [55] states that according to traffic measurements from mobile networks, 50% of HTTP requests are cacheable, leading to traffic reductions from 60% to 95%. Additionally, the use of ICN can reduce around 40% of backhaul usage in mobile networks [55], if enabled from the UE to core elements.

Interest aggregation can also lead to traffic reductions in the backhaul, avoiding sending requests for the same content along the network. Furthermore, the possibility of having identical requests increases with the number of requests [72], which means that, in effect, more gains from interest aggregation can be obtained for higher-level nodes of the network, i.e., core elements.

From the traffic offload perspective, ICN in RAN permits a more straightforward content retrieval than current solutions based on multipath TCP (MTCP), which can exploit, e.g., both Wi-Fi and LTE radio technologies. Even if MTCP permits the seamless usage of the air interface for higher layers, the transport layer still needs to treat each interface as a different flow connection. This restriction is not needed in ICN, since contents can be dynamically requested, regardless of the interface.

Those are some of the main advantages of adopting ICN in the RAN. Section 3.4 deals with some issues that need to be circumvented for successful ICN-RAN deployments.

3.4 Main Issues of ICN-RAN

The issues for the adoption of ICN are naturally not only technical. Some of the technical issues arise because of the policies currently adopted. Future 5G and network technologies, in general, will fit distinct business models. Stakeholders such as operators and content providers may have distinct positions not only in aspects such as network neutrality, but also technologies. For example, the benefits of caching are largely favoring streaming service providers such as Netflix, which have sophisticated logistics for CDN. Only when one looks at caching from an operator's perspective, the full potential of *edge* ICN deployment is realized. It allows a more efficient media distribution for the owner of the transport network.

However, these schemes might present challenges regarding policy and charge for users. Traffic aggregation along with caching strategies implies that a request might be attended by any node on the network, and satisfy different groups of users. Therefore, a PDN-GW could not correctly apply policy and shaping schemes on a flow basis, neither charging operations for each user. For instance, network operators charge user per data consumption, and in IP-based networks, this makes sense because the data flow belongs to a single user. However, in an ICN network, the same traffic could potentially satisfy multiple users. So, who should be charged and how, is an open question. In addition, the data do not always come from the same place. Popular data will potentially cost more (from the network perspective) for the early access than for the later ones (because the data would be cached).

Nevertheless, in the case of a cloud RAN (C-RAN) scheme, core functions might be co-located with eNodeB features. Thus, gateway functions, such as policy and charging, could be realized without losing track of user data, since the C-RAN would be the first hop of communication. Nonetheless, details regarding bearer establishment and other core control messages should be adapted to be compliant with ICN. Concerning the eNodeB protocols, PDCP would have to be modified to be coupled with ICN mainly regarding header compression and no need for integrity protection.

Another technical issue introduced by policies is the priority used in current networks, i.e., some users have more priority in its requests. However, in ICN, the interest of a high priority user could be aggregated (in the PIT) to the interest of an early low priority user interest. Therefore, upstream routers would not differentiate the distinct users and not apply the appropriate priority. Alternatively, a low priority request for popular content could lead to a better quality of experience than a high priority request for unpopular content.

Still, in this context, ICN access control is needed. Content providers currently deliver their content in a unicast channel, which provides a straightforward way of charging the user access. In contrast, ICN, by nature, stores the content in the network and does not provide feedback on how many, or which users accessed the content. This lack of information introduces challenges to both operator and content provider when charging the user, and also in the per-user content penalization.

On 5G scenarios, mobility plays a significant role, especially with the adoption of a massive number of small cells. In this scenario, ICN is a good alternative for the content consumption. However, when the content producer is mobile, ICN raises concerns [11]: how to update many big routing tables in ICN routers.

Another dimension of the ICN-RAN deployment is how to reuse the current IP network. Although ICN does not require or even maintain compatibility with IP, some believe it should coexist with the deployed infrastructure. The need for reusing current infrastructure is one of the assumptions that guided the ICN-RAN deployment and testbed presented in this work.

3.5 Previously Proposed ICN-RAN Deployments

The literature has discussed different ICN deployments in mobile networks [11]–[13], [15]. 5G Americas [11] makes considerations about ICN deployment in LTE, e.g., discussing dedicated radio bearers, local breakouts and network slices for ICN. [12] approaches the deployment stack itself and layers that should be added. The discussion presented by Ravindran *et al.* [15] adequately captures previous work and formalizes three deployments called: *overlay*, *integrated* and *flat*, which are represented here in Figures 20, 21 and 22, respectively.

In the former, using current mobile networks, ICN is deployed on top of IP. The integrated deploys ICN over IP only in the communication between the UE and PDN-GW. While the flat deployment deploys ICN in the RAN without IP, i.e., the UE is capable of sending ICN packets directly over the transport layer [15].

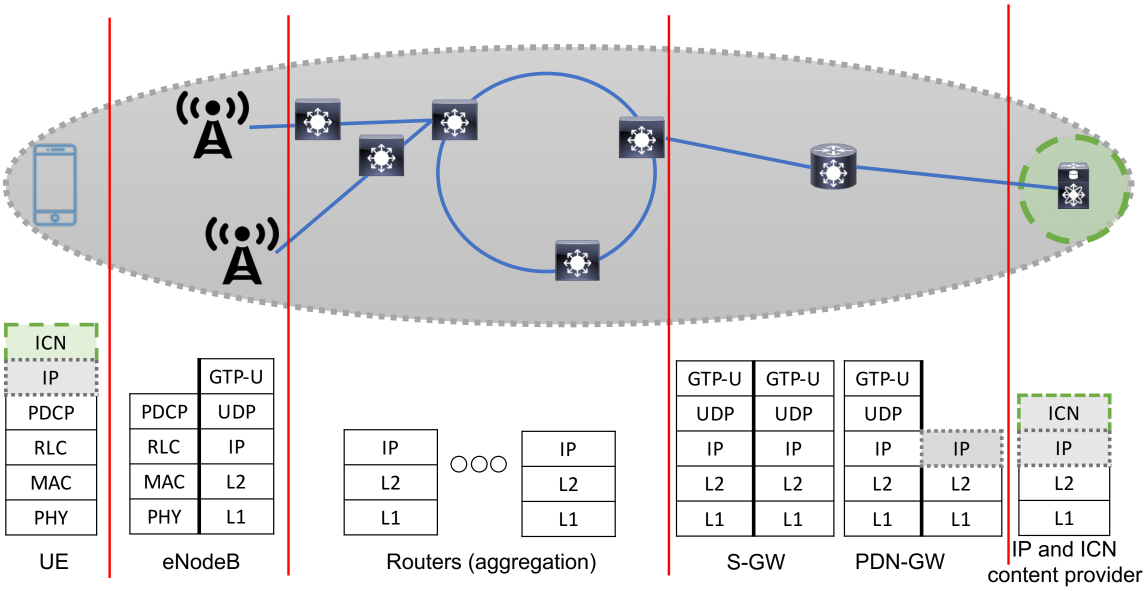
In the overlay deployment using current mobile networks, ICN is deployed on top of IP (Figure 20). This deployment is characterized by the encapsulation of the ICN packet in an IP packet by the UE, and it is only de-encapsulated by the content provider, i.e., from the mobile network point of view, ICN is not different from any other application running in the network. As such, the mobile network does not benefit from its deployment. The overlay deployment can benefit the content provider easing the deployment of cache in its CDN, but this discussion is out of the scope of this work.

In the integrated deployment, ICN is deployed over IP only in the communication between the UE and PDN-GW. As Figure 21 shows, the RAN is similar to the overlay's, but differs in the aspect that the ICN packet is de-encapsulated inside the mobile network, in the PDN-GW. Thus, ICN can now provide benefits for the mobile network, such as an economy in the interface with the content provider (interface between PDN-GW and external network, or Gi), caching content inside the mobile core network.

Figure 22 presents a deployment similar to the flat proposed in Ravindran *et al.* [15], and because of this is called here flat*. It can be observed that the RAN in flat* now deploys ICN directly over the PDCP layer, and does not depend on IP. The UE, eNodeB, and aggregation routers are enabled to support a dual-stack, deploying ICN and IP over a transport layer (L2). In contrast to the other two discussed deployments, the mobile network using flat* can now exploit all the advantages discussed in Section 3.3.

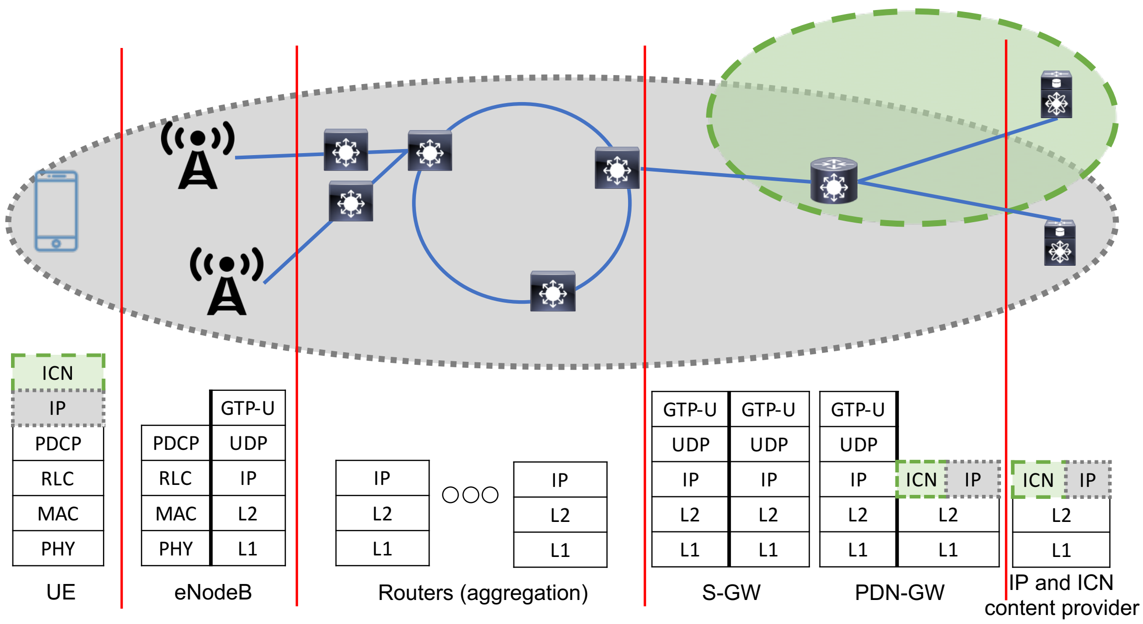
The flat deployment, proposed by Ravindran *et al.* [15] can be viewed as a specialized flat* deployment. The difference is that in flat* any protocol can be used for ICN transport (L2), but the flat deployment specializes flat* to deploy ICN in the aggregation nodes using IP for transport (L2). It is convenient to present the flat* in detriment of flat, due to the stronger connection that can be done between flat* and the deployment adopted in the proposed testbed.

Figure 20 – ICN on the radio access network (IRN-RAN) *overlay* deployment from Ravindran *et al.* [15].



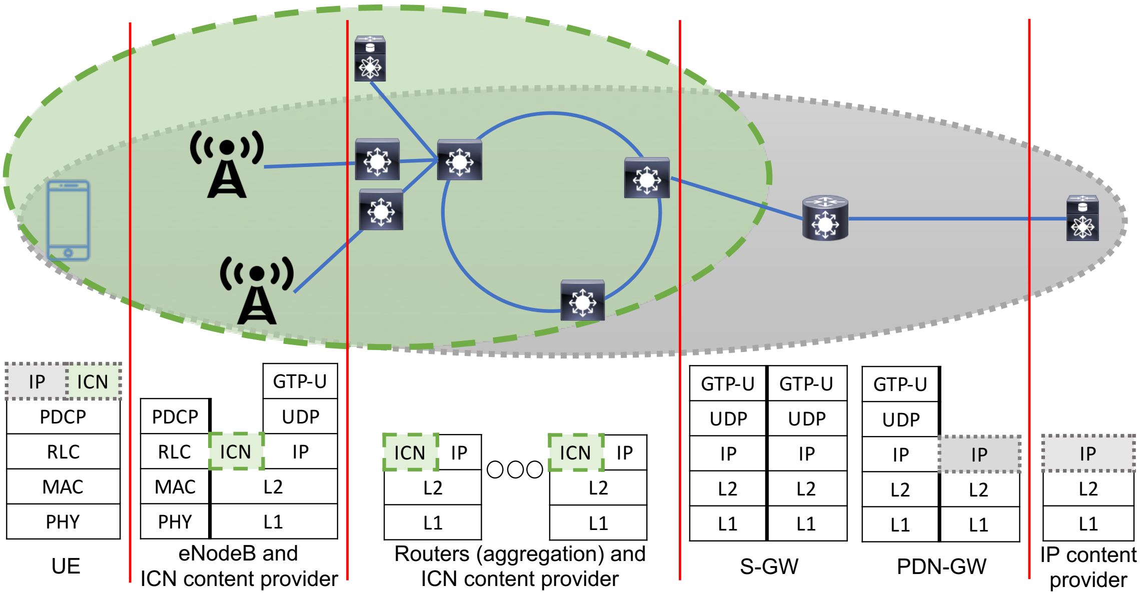
Source: author [30].

Figure 21 – ICN-RAN *integrated* deployment from Ravindran *et al.* [15].



Source: author [30].

Figure 22 – ICN-RAN *flat** deployment (a generalized version of the flat described in Ravindran *et al.* [15]).



Source: author [30].

3.6 Developed Testbed

In order to test ICN on 5G, it is necessary to rely on some 4G technologies until 5G equipment is available. One of the restrictions in the implementation of the proposed testbed is the adoption of COTS LTE UE. Hence, only the application layer of the UE can be modified, which requires working over IP. This limitation is imposed because LTE lower layers are implemented in hardware and firmware, which cannot be efficiently modified without the manufacturer's support. Even so, this scheme permits ICN layer to request content with minor impact on the current behavior of LTE networks. For the eNodeB and EPC used in the testbed, a software implementation using software-defined radio (SDR) is used. In this case, access to all the code is possible, and therefore modifications can be implemented.

3.6.1 Adopted ICN *Edge* Deployment

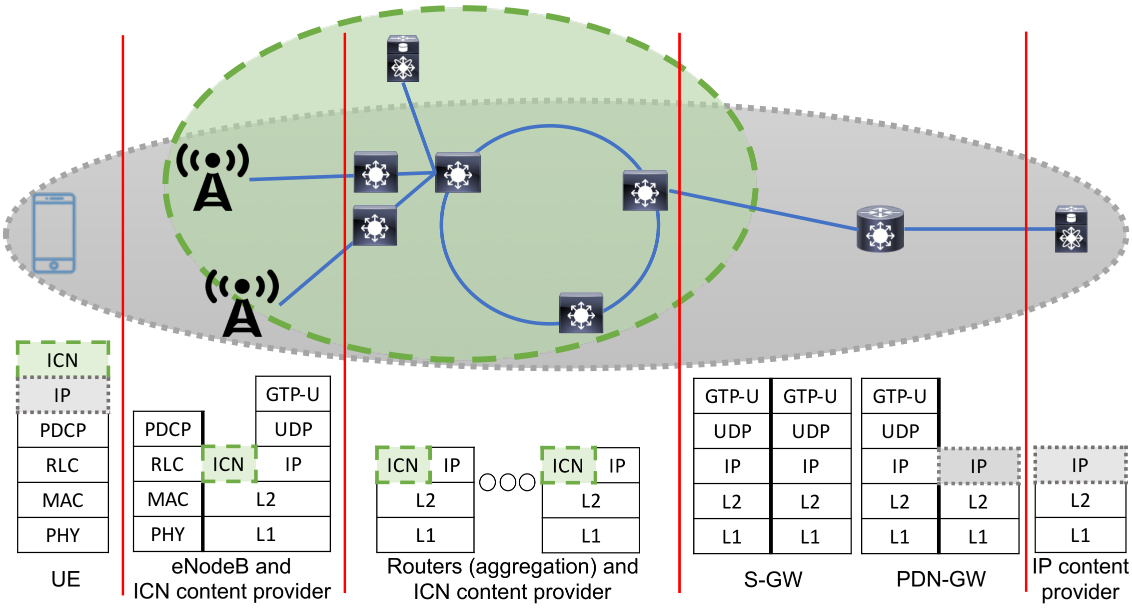
While seeking a suitable deployment, we identified an alternative not formalized in Ravindran *et al.* [15] and called here *edge* deployment. The edge deployment is similar to a more general model of deploying non-IP protocols in 5G networks [73]. Although not formalized according to the aspects emphasized in this work, edge was used in Gomes and Braun [13] and discussed in Suthar and Stolic [12].

The edge deployment is characterized by partially deploying ICN in the RAN and is depicted in Figure 23. The edge deployment is an intermediate step before a flat* deployment can be implemented. Its main advantage is to obtain some ICN benefits, such as aggregation in the backhaul while maintaining compatibility with the 4G UE. The UE eNodeB communication transports ICN over IP (as an application in a pure LTE network, as discussed in Section 2.2). This implies that the user still needs an anchor and IP configuration, which the flat* deployment does not require.

It should be noticed about edge that, in spite of requiring IP, the eNodeB is capable of filtering and forwarding ICN packets natively. Accordingly, the benefits of ICN in the mobile network can be observed. In the backhaul and core network, the current infrastructure can be reused, and ICN can be deployed in dual-stack routers.

The drawbacks of using edge instead of flat* are the overhead of using IP for transport which involves additional signaling and unused IP headers. Nonetheless, the network takes advantage of multicast routing and caching in the backhaul and core network, just like in the flat* deployment. Though, mobility is not fully available with the edge deployment because the core network still has to configure the IP of the user. Thus, increasing the setup time of cell association (for example in handover) and requiring more signaling than what would be needed for a pure ICN association.

Figure 23 – ICN-RAN *edge* deployment adopted in our testbed.



Source: author [30].

3.6.2 Main Software Used in the Testbed

In this work, we chose to use free and open-source software as much as possible. So system cost could be reduced, and we could achieve a high degree of tools customization.

The two main tools needed to enable the development of this testbed are one for the EPS network and one for ICN. For the former, we chose OpenAirInterface (OAI) and for the latter NDN.

OAI was created as an experimentation environment with the goal of prototyping 5G. The initiative of producing such a tool arose because besides the software simulation tools have evolved significantly, those are not yet enough to try all the 5G hypothesis. Thus, a tool to generate more real and generic tests and validation was necessary.

Without OAI, 4G prototyping would depend on commercial tools, which have elevated costs and are not sufficiently flexible, mainly for commercial reasons. Therefore, OAI fills this gap to develop a prototyped mobile network [74].

One of the main technologies that enabled the development of OAI was the software-defined radio [75]. Before SDR, the processing power required to develop radio systems made prototyping too expensive, thus unpopular.

With OAI, using an SDR board and a COTS computer, it is possible to offer an EPS network to which a COTS LTE UE can connect. The OAI implements both E-UTRAN and EPC. Hence, we can obtain a full testbed for the prototyping of new mobile network technologies, such as the one proposed by this thesis.

The main reason for choosing NDN, as discussed in Section 2.1, is its active development community and its vast and stable toolset. The NDN group publishes the NDN router, or NFD, which might be installed in different systems, from COTS computers to embedded systems and Android smartphones.

The NDN ecosystem contains a great toolkit for developing NDN application in many programming languages, including Python, C/C++, JavaScript, and Java. Other tools include support for network monitoring, data repository (NDN file server), artificial traffic generator, and network simulator. All of them are open source, which facilitates its adoption and evolution.

3.6.3 Testbed Implementation Details

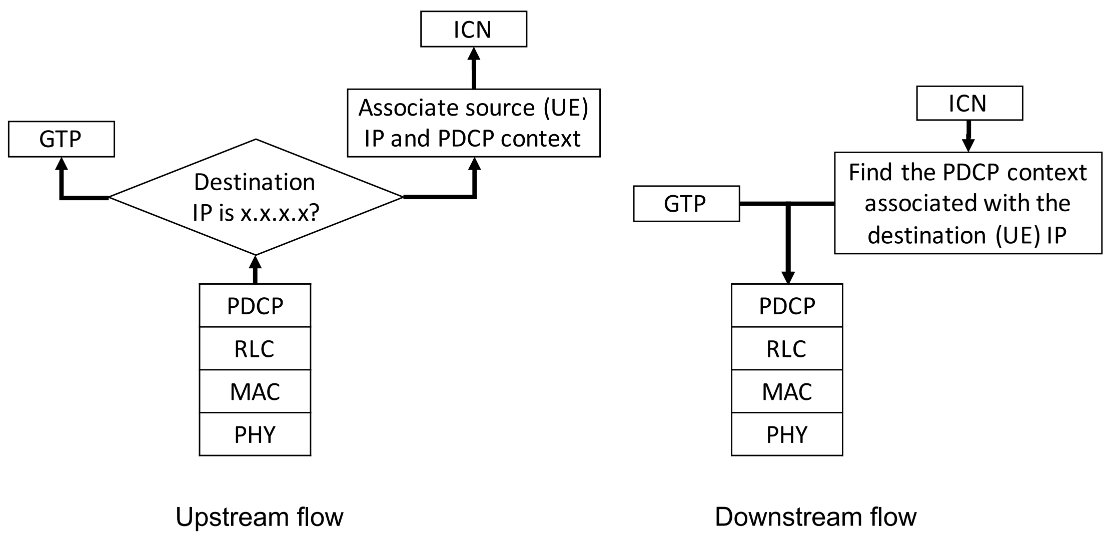
According to the LTE standard [76], LTE packets coming from the UE would pass through the PDCP layer in the eNodeB to the user plane GTP tunnel without any processing, modification nor routing. Inside the GTP-U tunnel, the packet travels over an insecure Ethernet access network and is processed by the S-GW.

In the developed testbed, the PDCP layer of the eNodeB was modified to implement a local breakout that redirects traffic addressed to the NDN router. That is done without the need to traverse all the access network nor encapsulate the packet in a GTP-U tunnel. Figure 24 illustrates the data flow at the eNodeB in the testbed implementation.

The NDN router is abstracted as an eNodeB local process using virtualized IP for transport between router and UE. For communication with the core network, any transport can be used. To reuse current infrastructure (as illustrated in Figure 23), the router communicates over Ethernet using the same access network [73] as the IP traffic encapsulated in GTP-U. So, ICN routers can be placed anywhere in the access network, implementing simple functions such as traffic prioritization, caching or any edge function. All of this while being already protected by the security features of ICN.

As illustrated in Figure 24, in the uplink flow, the eNodeB filters packets sent by the users based on IP destination address. Not selected IP addresses follow the usual flow from PDCP to GTP layer, to be forwarded to the EPC. ICN packets (matched through IP destination address) are delivered to a local process to record the sender PDCP entity. Then they are forwarded to an ICN network over Ethernet for content not cached at the eNodeB. In downlink flow, ICN packets sent from the ICN network are routed to the PDCP entity previously mapped, which sends the packet to the user that requested the content.

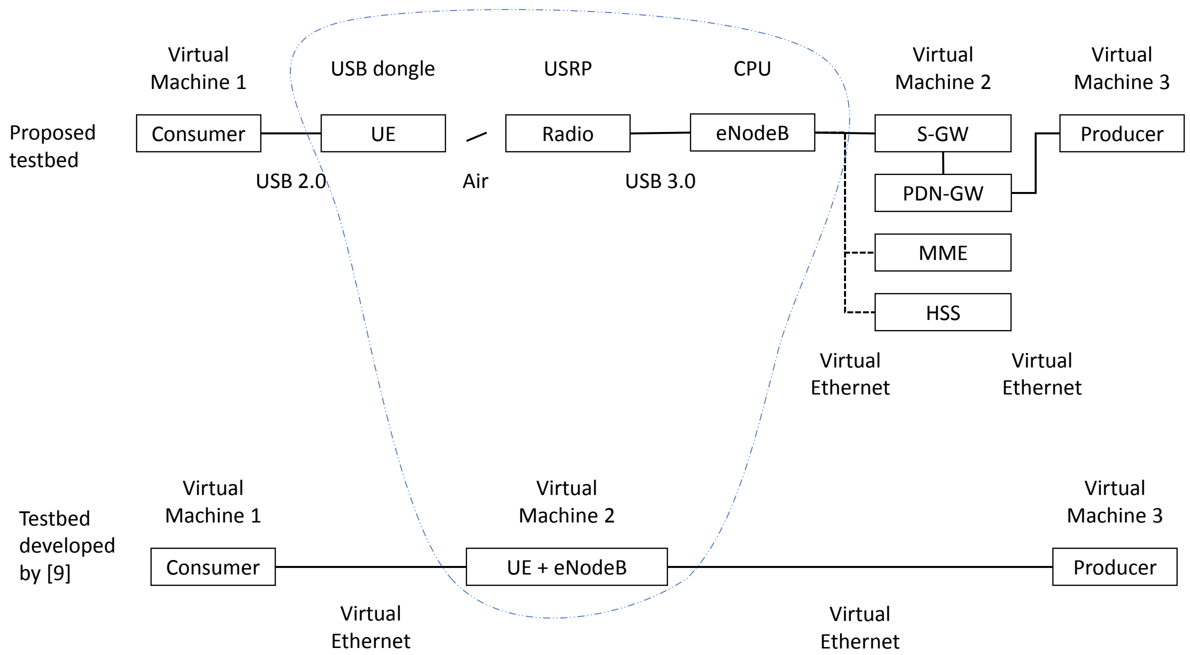
Figure 24 – Data flow of the breakout implemented above the PDCP layer for the edge deployment.



Source: author [30].

The illustration presented in Figure 25 shows a diagram to clarify the setup. The diagram presents the platform that executes each functional block, as well as the interfaces connecting them. According to the edge deployment, multiple aggregation routers could be present between eNodeB and S-GW. Nonetheless, since we are using only one eNodeB no aggregation is needed. Thus, to simplify, we did not use any aggregation routers. A connection between the proposed platform and the one described by Gomes and Braun [13] is also shown. The comparison shows that our proposal extends from Gomes and Braun [13], i.e., we are implementing the components, including the radio interface between eNodeB and UE (LTE-Uu), instead of emulating them. Furthermore, we also included the core network which is not present in Gomes and Braun [13].

Figure 25 – Diagram illustrating the components of the proposed testbed in comparison with previous work. The traced curve illustrates the equivalent components that were simulated by Gomes and Braun [13]. Their study does not include the core network.

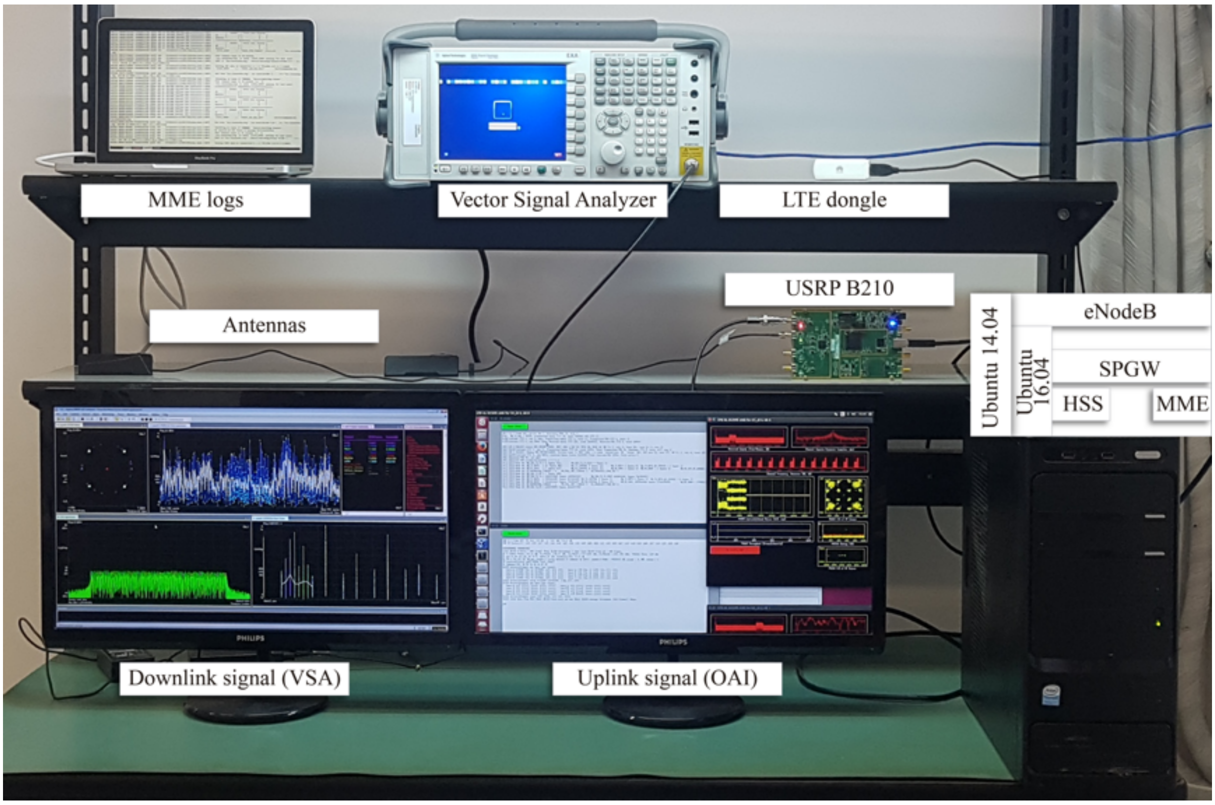


Source: author [30].

Figure 26 shows a picture of the testbed. The hardware for the LTE network is composed of a personal computer (i7-5930K processor), a SDR (universal software radio peripheral, or USRP, B210), a LTE dongle (E3272), a Motorola Moto G3 2015 smartphone (not shown in the picture) and antennas. A vector signal analyzer (N9010A) is used for measurements. The LTE eNodeB needs Ubuntu 14.04 and is installed on the personal computer (PC). Also on the PC, there is a virtual machine (VirtualBox) running Ubuntu 16.04 where the EPC, composed of S-GW, PDN-GW, MME and HSS, is available. The reason for having two distinct Ubuntu versions is that the eNodeB requires a special real-time kernel. The PC is connected to the USRP using a universal serial bus (USB) 3.0 cable.

This setup provides an LTE network over the air, to which the LTE dongle and smartphone are connected. Another virtual machine running on the PC connects to the dongle using a USB 2.0 cable to run the applications over the LTE network. An artificial delay of 11.5 ms was imposed on the link connecting the eNodeB and EPC to emulate a typical real setup [57].

Figure 26 – Picture of the testbed used to generate the prototyped EPS network (4G) and experimenting with ICN.



Source: author [30].

3.7 Proofs of Concepts With the Testbed

This section presents proofs of concepts for ICN over mobile networks using three scenarios. The goal is to show the potential of the testbed for evaluating performance, benchmarking algorithms, and testing new ideas. The first PoC demonstrates the reduced latency of accessing the content in the eNodeB. The second one uses ICN cache and interest aggregation features to decrease latency and backhaul usage in a live LTE network. And the third evaluates the performance of state of the art NDN software, exposing the need for optimized applications.

3.7.1 Latency for Accessing Content at the Network Edge

To measure the latency-wise performance of the edge deployment compared with legacy networks (ICN overlay deployment), the round trip time (RTT) was measured in two scenarios. In the first (overlay), the LTE network is used in the standard form. As discussed, in this case, ICN is an application with its server running on the EPC. In the second scenario (edge), ICN is running at the eNodeB through the local breakout proposed in Section 3.6. The latency is expected to be lower in the edge deployment because the packet does not have to transverse the backhaul as in the overlay deployment.

Table 10 shows the estimated RTT times, measured during 25 s, with a sampling period of 500 ms, for each scenario. It is observed that the edge deployment outperforms the overlay in all the calculated criteria. This result indicates that when low latency is required, ICN in the eNodeB is a good option. It is important to note that this is a best-case scenario for overlay deployments because, in a real-world scenario, the backhaul is subject to congestion, which is not happening in this test.

Table 10 – Results comparing the latency in milliseconds when the first ICN router is available in the eNodeB (edge) and when it is available in the EPC (overlay). Values are round trip times. Min and Max stand for the minimum and maximum values. Stddev stands for standard deviation.

Architecture	Min	Mean	Max	Stddev
Overlay (ICN in EPC)	27	40.02	144	15.32
Edge (ICN in eNodeB)	16	27.34	124	14.81

Source: author [30].

3.7.2 Multimedia Delivery Over ICN-RAN

To evaluate video transmission, in this work, we developed an ICN video player in JavaScript. This allows us to take advantages of hypertext markup language 5 (HTML5) media source extensions and the NDNjs library. The player is compatible with any recent HTML5 compatible Web Browser. For the presented tests, we used Google Chrome.

The JavaScript code processes the HTML5 `video` element, in which a new attribute called `ndnsrc` was added. The `ndnsrc` attribute indicates the name of the video content that will be requested using NDN. The video player requests, sequentially, the content segments that represent the video. Then provides each segment to the media source extensions which decodes them and show the video to the user.

This experiment considers a scenario where two users are streaming the same video to evaluate ICN cache and interest aggregation functionality in the access network. Although the two users see the same video, their viewing time is not synchronized, i.e., user 1 joins the stream later than user 2. In the edge deployment, we expect that the video does not transverse the backhaul more than once. This happens as a result of the ICN cache properties, which saves recently accessed content directly into eNodeB memory. Thus, saving backhaul traffic.

Figure 27 shows the traffic in the eNodeB UE interface (LTE-Uu), and in the backhaul, i.e., between eNodeB and EPC (S1). For convenience, the traffic in the LTE-Uu interface for each user is also shown.

We divided the experiment into three steps. The idea is to reproduce situations where each user is reproducing the video alone and together. Specifically: in the first step, only the user 2 is requesting the video; in the second, the two users; and in the third, only the user 1.

It is possible to observe that when a local breakout is used (Figure 27b) the traffic in the backhaul is not affected by how many users are requesting the video, i.e., the video is not transmitted more than once in the backhaul. The video is transmitted only once because on step 2, when user 1 starts to request the video, user 2 had already obtained it in an earlier step. Therefore, the eNodeB stored it. On step 3, at approximately 98 seconds, it is observed that the eNodeB does not use the backhaul anymore. This happens because the users do not request new data since user 2 stopped and user 1 is requesting data previously accessed by user 2. The backhaul is only used again when user 1 requests data not viewed earlier by user 2, at around 118 seconds. This behavior of backhaul utilization does not happen on Figure 27a because, in this scenario, all the data are obtained by the users from the EPC. As a consequence, the inclusion of a new user directly affects the backhaul usage, being it, approximately, the total traffic of the two users.

This PoC highlights one of the leading benefits of using ICN in the network edge: the efficient use of the backhaul. Moreover, when dealing with popular content, the system bottleneck becomes the content delivery at the network edge since the content is not redundantly transmitted in the backhaul.

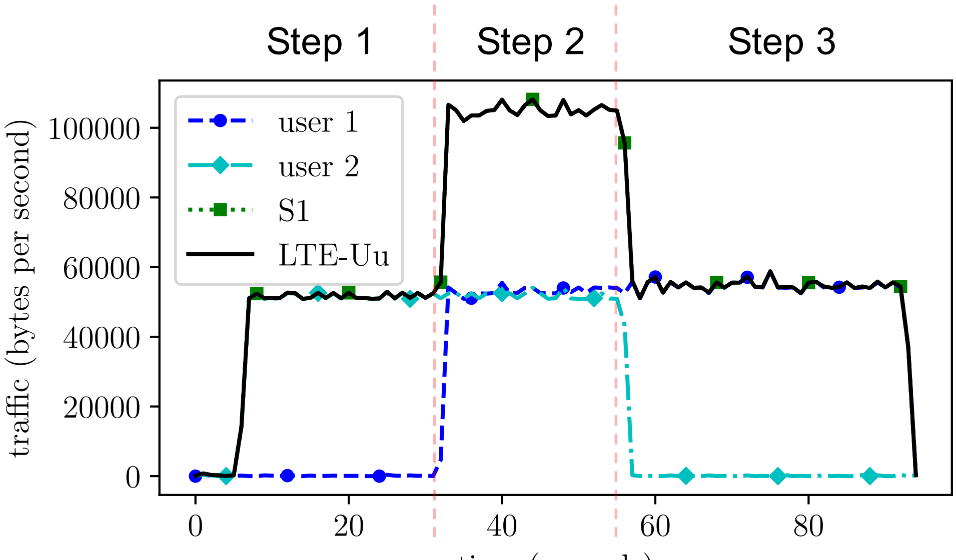
3.7.3 Real-Time ICN-RAN

This section evaluates the performance of the system regarding throughput. As discussed in Section 3.6.1, ICN traffic is encapsulated in the IP protocol when the edge deployment is used. Thus we first measured the upper bound throughput for the system, i.e., the throughput achieved using only IP, and then discuss the performance of ICN, emphasizing why the upper bound was not achieved.

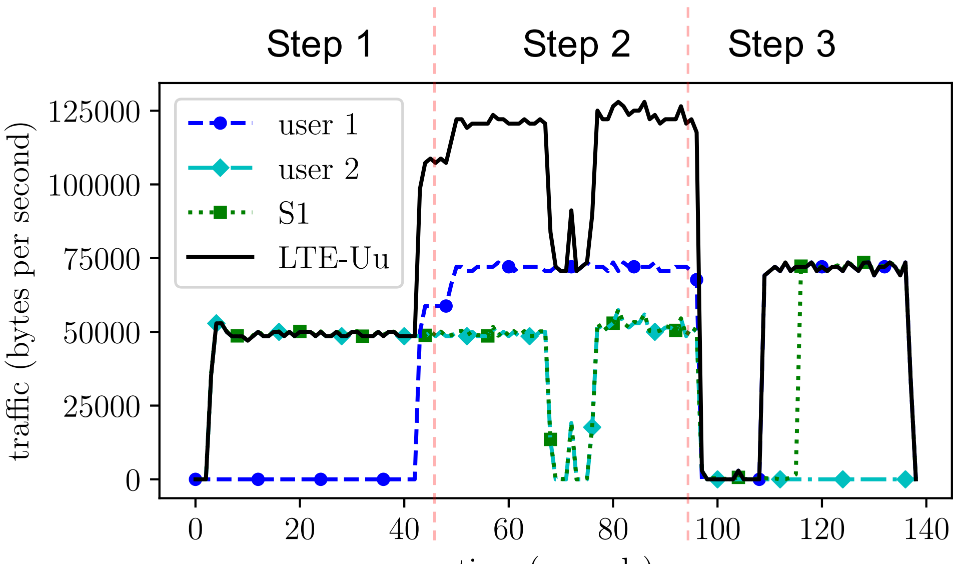
A commercial smartphone is used (Motorola Moto G3 2015) in this test to evaluate the capabilities of current NDN software. Firstly, to validate the connection and establish an upper bound for the ICN evaluation, we used Iperf3 and measured the LTE network (IP) bandwidth. The values obtained were 1,554 and 1,329 KB/s in the downlink and uplink, respectively. Then, to test the ICN bandwidth, a Java application was written for Android using the open source jndn library. The application uses data packets of 1,378 bytes to maintain compatibility with the common 1500 bytes of Ethernet. The application sends interests according to a window (the size of the window, in this case, refers to the number of packets allowed to be in transit) and computes traffic statistics. The implementation of cryptographic functions by software is computationally heavy, and ICN needs them to sign and encrypt the content. Thus this software implementation can be a bottleneck in

Figure 27 – S1 (*backhaul*) and LTE-Uu traffic when two users are requesting a video in the overlay or edge deployment.

(a) Overlay deployment. The curves of S1 and LTE-Uu are superimposed.



(b) Edge deployment.



Source: author [30].

the performance of ICN applications in systems with limited computational power.

The upper bound bandwidth, measured by the Iperf3, is achieved by ICN if $\frac{1{,}554{,}000}{1{,}378} = 1,127.71$ ICN packets per second are transmitted. Considering that the RTT is 0.02334 seconds, as measured in Section 3.7.1, a window of one packet transmits $\frac{1}{0.02334} = 42.84$ packets per second. Thus, to achieve the maximum throughput, a window of $\frac{1{,}127.71}{42.84} = 27.34$ packets is needed.

Our evaluation showed that there is no benefit from a window larger than 3 packets. The NDN library imposed this limit because it could not process that many packets in real-time. Therefore the maximum throughput is not being determined by the network, but by the software, which cannot process the window of 27.37 packets in real-time.

At best, the system was capable of processing 45.2 ICN packets per second, measured in sessions with a duration of 40 seconds and a window of 3 packets. This processing translates to a throughput of 63.8 KB/s, considering that each ICN packet is composed of 1,378 bytes. However, 378 bytes of each packet are ICN headers, which leads to 50.71 KB/s of information. This throughput cannot be directly compared with the IP one, since NDN adds new functions to the network, such as security [4].

These results showed that the NDN libraries would benefit from additional optimization. Otherwise, applications that demand high bandwidth utilization are limited when executing in a real environment. One example is the video player developed in Section 3.7.2, which can only reproduce low-resolution videos in real-time.

4 Reinforcement Learning-Based Slice Admission

Virtualization is at the core of 5G network architecture, where cloud platform spans across the different network segments, diverging from the traditional centralized cloud architecture. A cloud-based network infrastructure inherently supports multi-tenancy and resource sharing. In such a context, there is a need for intelligent resource management to deploy network slices on the shared infrastructure. As motivated in Section 1.2, this thesis proposes a reinforcement learning agent to optimize the network resource utilization. The present section describes this contribution. We start by describing some network slice management studies that have been considered in the literature, followed by our approach to tackle the slice admission problem.

4.1 Slice Admission Related Work

Samdanis *et al.* [26] proposes a 5G slice network management architecture. It is focused on having many players interacting over the network and the interfaces for communication among them. The architecture assumes a shared radio access network, which is divided into multiple domains. Each domain is controlled by a domain manager, which are themselves managed by a higher-level network manager, called 5G network slice broker. The 5G network slice broker is the entity that communicate with the tenants. To operate the network, the authors explicitly cite a set of metrics that are important for slice management. These metrics include the amount of resources allocated to a network slice, such as physical resources or data rate; timing, such as starting time, duration or periodicity of a request and time window; the type of resources; and quality of QoS parameters, such as radio/core bearer type, prioritization, delay jitter, and loss. These metrics are important for understanding what the general service requirements at a high level are, such as service mobility, data offloading and disruption tolerance. Thus, the slice management entity can be estimate if the current load on the system can fit the new slice.

Sciancalepore *et al.* [77] developed an admission control module for slice admission in a mobile network. Their model assumes that the bottleneck of the network is the physical resource (spectrum), which is shared among the network tenants. The information provided by the tenant to InP at the slice request includes maximum resource utilization, duration of the slice, and traffic class. In this context, traffic class specifies some behavior of the traffic, i.e., delay tolerance, and if the bit rate should be guaranteed or not. The work considers a total of six traffic classes. Once deployed in the system, tenants request resources according to a Poisson process, and the InP must provide them. Otherwise, a SLA violation penalty is incurred. The solution proposed to solve this problem applies a prediction of the traffic load of the requested slice. Based on this and the predictions of the previously admitted slices, the admission module can evaluate if the new slice can be placed into the system. The combination of all the possible slices in the system is modeled as a geometric knapsack problem. When the slice leaves the system, the prediction module (as part of the admission) is informed of the actual behavior of the slice. Therefore, it can evaluate how accurate its prediction was and update its knowledge with the new experience.

Another system for slice admission is studied by Bega *et al.* [78]. Their model of mobile network also has the physical resource (e.g., spectrum) as the bottleneck, and they have two types of traffic classes: elastic and inelastic. Inelastic users are characterized by having an SLA which specifies that all the requested resources must be provided when needed. Elastic users, on the other hand, do not require a specific number of resources and can cope with a variation of the number of allocated resources. Also, their SLA is specified by average resource availability. A slice request is composed of the slice duration, the traffic type, and the slice size (in number of clients). The admission problem is modeled as a Markov decision process (MDP). The states are the number of elastic and inelastic users present in the system. The actions are: accept or reject the slices, while the objective is to admit as many slices as possible, whereas guaranteeing the tenants requested SLA. Bega *et al.* [78] proposes the use of Q-learning to solve the problem. They compare their solution with two heuristics and an analytical algorithm. They show that their proposed solution can adapt if the system does not behave as modeled and provide better decisions than their baseline.

Apart from the research works targeting new approaches to network slicing and resource management, there is significant work being done on developing platforms. These platforms can be used to integrate such solutions in real networks. The seminal work on the platform side was started with the European Telecommunications Standards Institute (ETSI) NFV group. They released a whitepaper outlining how network infrastructure made of physical nodes would be transformed into a software system running on general-purpose servers [79]. Subsequently, the group presented the NFV Management and Orchestration Framework (MANO) that has been the reference architecture for many platforms currently being developed for virtualized network management [80]. The reference implementation of the ETSI MANO architecture is called Open Source MANO (OSM) and is actively maintained by the open source community. Similar initiatives were started by vendors and commercial entities to produce carrier-grade options resulting in platforms like Open Orchestrator Project (OPEN-O) and Enhanced Control, Orchestration, Management & Policy (ECOMP). And, more recently, their converged realization, Open Network Automation Platform (ONAP) [81].

Work on the development of management and orchestration platforms is ongoing together with standardization on the architecture, interfaces, and functionality. Nevertheless, the need for developing intelligent solutions for core features such as network slicing remains with the platforms providing a more straightforward path towards integrating the solutions in a realistic environment. In this work, we address the intelligent network management, and present a policy agent for slice admission control in virtualized 5G networks.

We proceed with an overview of a high-level network management loop that can be applied at multiple levels of the network, especially to control the admission of new slices.

4.2 The Mobile Network Control Loop

Network management operations in many levels of the network such as service admission or orchestration can be modeled as a closed-loop operation, as depicted in Figure 28.

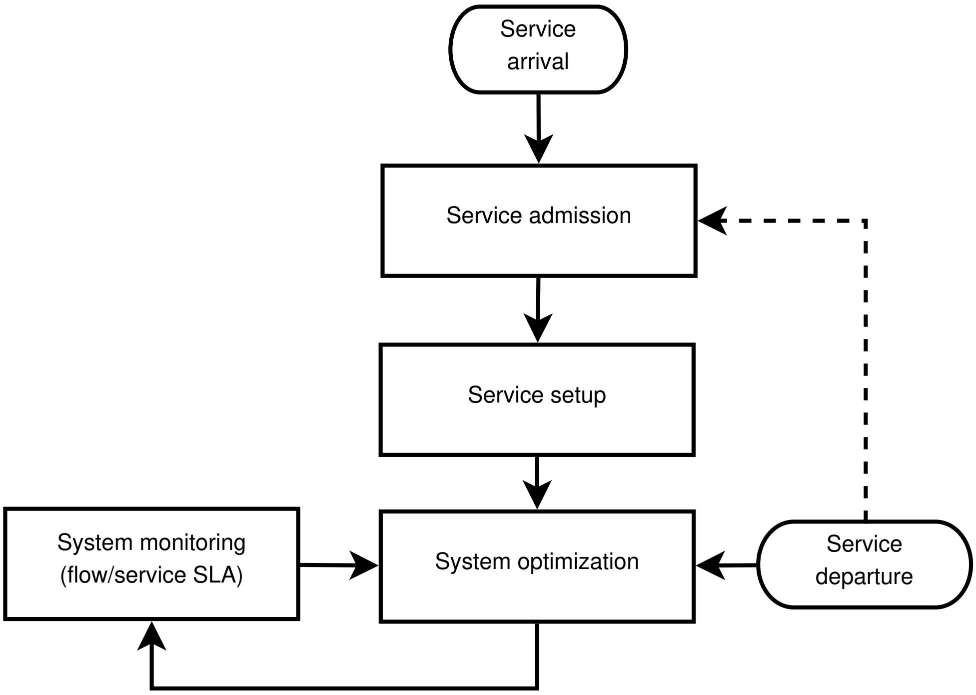
In general, a service arrives at some part in the network and then goes through an admission policy. The policy decides whether it is in the interest of the network operator or infrastructure provider to accept the service or reject it.

Services that are admitted in the network require some setup. For example, the service might consume some infrastructure resources. And the network may have multiple resource pools from which the resources could be provided. Consequently, the decision of which pool to use is taken in this step.

Once the new service is deployed, the system enters in a general control loop where not only the deployed service instances but also the network, is continuously monitored and optimized. The closed-loop monitoring and control mechanism ensures that if a deployed service instance needs more or fewer resources than it did at the deployment time, those additional resource requirements can be accommodated. Or, if they are not needed anymore, they can be allocated to another slice in the network. Additionally, in the case of service completion or departure from the network, the closed-loop operation ensures that not only the resources are taken away from the departing service but also the state of remaining services is optimized.

For an intelligent admission policy, the SLA parameters observed by the departing service during its lifetime are essential, since positive or negative feedback can be used by the admission policy to optimize its performance for future decisions. The management and control loop depicted in Figure 28 can be adapted to operate in various contexts or other parts of the network. To demonstrate this general management concept, we apply it to our proposal where we use the closed-loop approach to a high-level network operation.

Figure 28 – Flowchart representing an overview of the general network management loop. The dashed line represents the exchange of information, for example, reporting the overall satisfaction experienced by the service during its lifetime.



Source: author Batista *et al.* [31].

In this network operation, slices are deployed onto a shared network infrastructure.

In this network sharing context, there are three distinct roles comprising the InP, entity that owns the infrastructure on which the slices are executed. The tenant, entity that requests resources from the InP to run its services. Along with a user, which consumes the services from the tenant.

A slice deployment request made from the tenant contains an SLA requirement, and the InP must provide enough resources to fulfill the SLA. Examples of SLA include network coverage over a certain geographical area and minimum network bandwidth. The InP has limited resources, therefore, in some situations, cannot fulfill the SLA for all the tenants requesting slices.

The limited resources motivate the existence of the slice admission module. Its objective is to admit as many slices as possible into the system, with the objective of maximizing resource utilization, consequently increasing the revenue for the infrastructure provider. The constraint is that it should not allow slices that would have their SLA violated, or cause SLA violation for the other deployed services.

A network slice may require resources in multiple parts of the network, for example, processing power in base stations or connectivity in the backhaul. Certain decisions must be made to setup those resources. For instance, there are usually multiple paths connecting the base station to the core network. Therefore, the decision of which of those paths will provide the required connectivity for a particular slice is made by the service setup module.

Once the slice is admitted in the network, it becomes operational and the system enters in its main control loop with respect to that service. Resources in the system are optimized continuously so that their allocation to each slice matches the real-time service needs.

The system is also constantly monitored so that the SLA observed by all the deployed slices is recorded and evaluated for compliance. The last significant change in the system happens when the lifetime of a slice ends, and it must leave the network. This leave triggers an optimization of the system so that it can optimize its resources to the slices currently deployed. The departure of a slice is also reported to the admission control. Thus, it can evaluate the consequences of the admission of other slices as well as the slice behavior. The latter enables the admission control to learn how to make better admission decisions in the future.

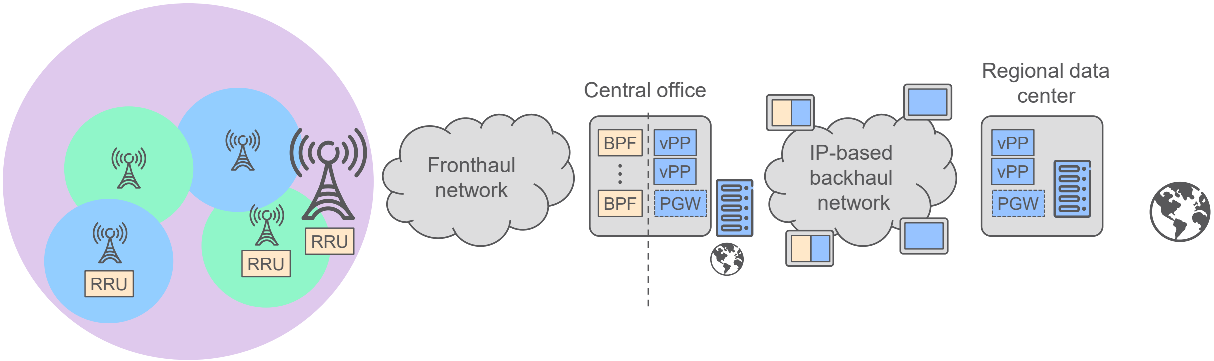
4.2.1 System Model for Slice Admission

In this section, we present the details of the considered system model for the proposed reinforcement learning agent for slice admission control. Our system model considers not only physical resources but also the high-level components of the network such as virtual network function (VNF) and computational and connectivity resources. Furthermore, any of those resources might be a bottleneck for the different slice classes in the network. As the main objective is to evaluate the policy agent for slice admission against other approaches, our system model is based on the concepts presented by Raza *et al.* [82].

The overall network architecture is presented in Figure 29. Its main components, modeling a metropolitan area, are a couple of regional data centers (RDCs), a few dozes of central offices (COs), and hundreds of remote radio units (RRUs). The RDCs provide connectivity to external networks and have general-purpose processors (GPPs). The COs have both special (radio) purpose processors (SPPs) and GPPs. Those resources at COs are more expensive than in the cloud but they can deliver a lower latency. The RRUs provides the radio access to end-users.

At the assumed abstraction level, there are some components that can be deployed by slices. The edge (CO) and core (RDC) both provide GPP, which can be used by slices to execute general processing functions such as a virtual package processor (vPP); mobile network functions, such as package gateway (PDN-GW); and slice specific applications. The other resource is connectivity, which enables communication between edge and core. Each CO $c \in \mathcal{C}$ has a capacity of g_c GPPs; each RDC $r \in \mathcal{R}$ has a capacity of g_r GPPs;

Figure 29 – Overall slice resources architecture in a flexible mobile network. Slice require functions that can be placed and consume resources in different parts of the network.



Source: author [31].

and each link $l \in \mathcal{L}$ has a capacity of d_l link capacity units. In the assumed model, all the resources have integer units.

The described resources are consumed by slices that deploy the presented components in the network. The maximum number of GPP that a slice can request at each CO is k_c, and at any RDC is k_s. Moreover, the maximum number of connectivity resource between them is k_m. Deployed slices have dynamic resource requirements over time, and if the InP can not provide the requested resources, an SLA violation occurs.

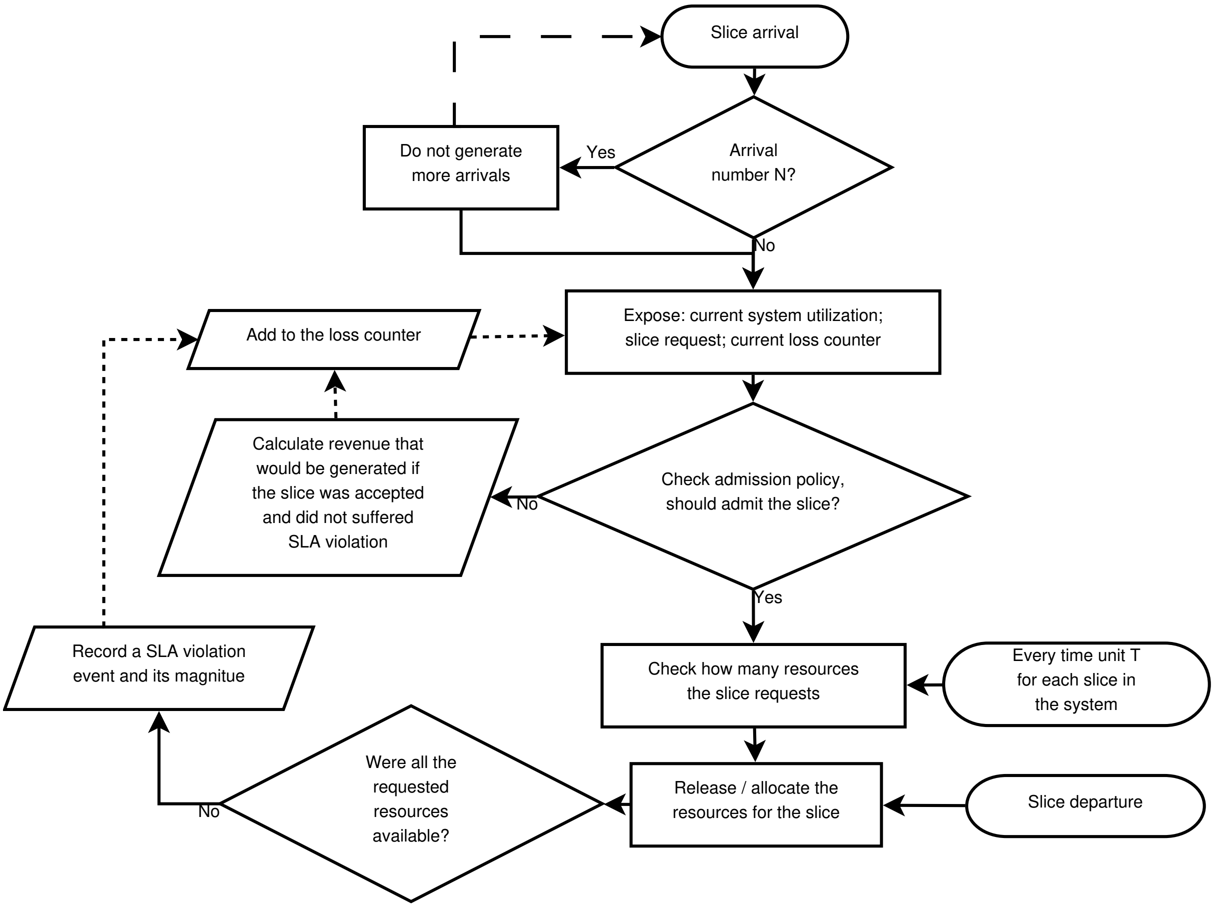
Tenant t requesting a network slice must inform the InP of their immediately requested GPPs at each CO j_c (which indicates the region it wants to have coverage in), number of GPPs at a RDC j_s, connectivity between them j_m, duration j_e, and priority j_p.

If an InP reserves enough resources for the admitted slice, it is guaranteed that no SLA penalty will be imposed and the tenant will remain satisfied. However, the tenant does not always use the maximum number of allocated resources, and reserving them leads to resource underutilization. Therefore, the InP can try to understand the behavior of its tenants and sometimes oversubscribe the system by deploying additional slices. Hence achieving higher revenue with a managed risk of causing SLA violation.

When the slice ends its service life cycle, the InP receives its revenue for hosting the service. This revenue is a fixed amount agreed at the time of admission based on slice parameters. Any SLA violation causes a decrease in this value, which is proportional to the magnitude of the violation.

The described environment model allows for the study of slice admission and its consequences in the system. In the next section, it is used to study new techniques for training reinforcement learning slice admission agents.

Figure 30 – Overview of the experience gathering in the network slicing system.



Source: author [31].

4.2.2 Reinforcement Learning Agent for Slice Admission

In the network slicing scenario, the objective of the InP is to increase revenue. In the general management loop presented in Section 4.2, one of the decisions that can be optimized is the slice admission. Assuming that all the other procedures (e.g., setup and scaling) are established, we study an RL agent that learns when to accept or reject a slice into the system to maximize profit. Such an agent would have to learn how the system behaves. Thus, it can consider the current load on the system, understand the risk of causing an SLA violation, and decide on the slice admission.

The overall modeling of the experience acquisition in the network slicing system is presented in Figure 30. The diagram emphasizes the data that is generated and on which data the agent learns from. In the system, there are three events: slice arrival, slice departure, and a periodic check of slice health (requested resources).

The system is modeled in an episodic fashion to make the learning suitable for RL. Each episode is determined by the arrival of N slices. The objective is to accept as many of those as possible while avoiding SLA violation.

Upon a slice arrival, the agent is consulted to make a decision. It is made aware of the current system utilization, the slice request parameters, and the accumulated loss suffered by the system until the time of arrival. Based on that, it either accepts or rejects the slice. If rejected, the InP interprets that some revenue could be obtained if the system had resources to admit the slice. Thus, the rejection is interpreted as a loss, and implies in a negative reward. Accepted slices are allocated the number of resources requested, if those are available. Yet, if only a fraction of the resources is available, those are allocated, and an SLA violation is recorded.

The slice scaling event revisits all deployed slices in the system and checks how many resources the slices are requesting and adjusts the resource allocation. As before, if only a fraction of the requested resources is available, the fraction is allocated, and an SLA violation is recorded. The scaling event is fired periodically with period T.

When the slice finishes its execution, a slice departure event is generated, which releases resources allocated to the slice immediately. When all the N slices arrive, and the accepted ones finish their execution, an episode is finished.

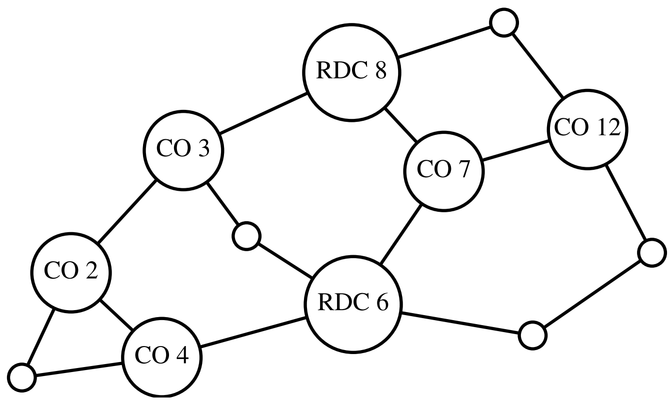
Aligned with previous work [82], [83], our RL agent is using a policy network, which has a configurable number of inputs, hidden layers and two outputs. The two outputs represents the probability of accepting and the probability of rejecting the slice. The input to the neural network is: the system utilization, the amount of each resource requested by the slice, its duration, priority and tenant. This information is encoded in a binary representation according to $\mathcal{A}_b(x)$, which is defined as a bit field of b bits with the first x bits equal to one and the others $b-x$ bits equal to zero. The state vector s, when slice j is requesting to enter the system, is then created by concatenating the bit fields: $\mathcal{A}_{g_c}(n_c)$, $\mathcal{A}_{k_c}(j_c)$, $\mathcal{A}_{g_r}(n_r)$, $\mathcal{A}_{k_s}(j_s)$, $\mathcal{A}_{d_l}(n_l)$, $\mathcal{A}_{k_m}(j_m)$, $\mathcal{A}_{k_e}(\min(j_e, k_e))$, $\mathcal{A}_1(j_p)$ and $\mathcal{A}_{n_t}(t)$, where n_c, n_r and n_l are the number of busy resources at c, r and l, respectively, $c \in \mathcal{C}$, $r \in \mathcal{R}$, $l \in \mathcal{L}$, and k_e represents the maximum requested duration observable by the agent. From this information, we believe the RL agent can understand the slice requirement and system status to make the proper admission decision. This is evaluated in the next session.

4.3 Evaluation of the Reinforcement Learning Agent for Slice Admission

We evaluate the proposed method using the network topology shown in Figure 31 [84]. The nodes with a high degree of connectivity were selected as RDC, some with medium degree as CO, and some nodes are connection points that provide connectivity routes.

Each RDC has a capacity $g_r = 80$, each CO $g_c = 50$ and each link $d_l = 50$. The

Figure 31 – Network topology used to evaluate the system.



Source: author [31].

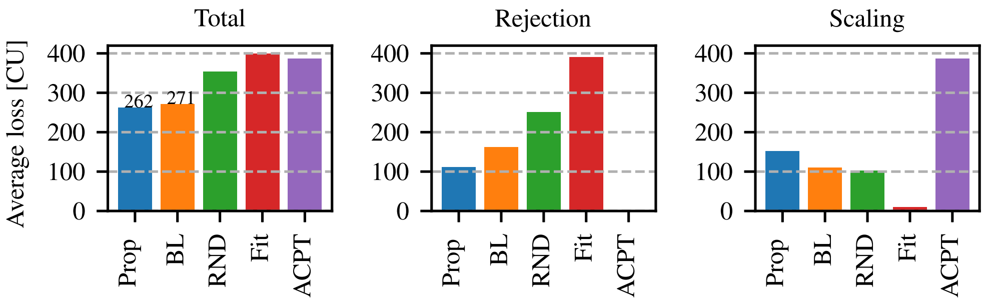
reference behavior of slices (profiles) are the ones reported by Raza *et al.* [85], i.e., the resource requirements of each slice are given by the time of the day and the type of the slice. Between 9:00 and 19:00 the high-priority slice requests 20 GPPs resources in its CO, 5 connectivity resources and 5 RDC GPPs. This is usually the busy hours at business districts where high-priority traffic is likely to occur. Other than at those hours, the high-priority slice requires only 15 GPPs in its CO. The low-priority slice, between 16:00 to 22:00, requires 10 GPPs in its CO, 10 connectivity resources and 10 RDC GPPs. This is likely to be the busy hour in residential districts, where low-priority slices are likely to exist. During other hours, low-priority slices require 5 GPPs in its CO, 5 connectivity resources and 5 RDC GPPs. This configuration reproduces a network that was projected to have as main bottleneck the resources at COs, which are the most expensive.

In contrast with previous work, which considered fixed slice profiles, we assume that the resources requests of a slice are non-deterministic. The current simulator accepts integer resource requests. Consequently, we chose to use a Binomial distribution with 5 trials and a tenant-dependent success probability (u_t) to generate noise. This noise is sampled every time unit and subtracted from the resource request of each resource type of the reference profile. If this subtraction leads to a negative resource request, it is understood as no resource needed (0). The slices impose penalties as discussed in Section 4.2.1 and the magnitude of those penalties are proportional to the tenant penalty weight (v_t).

We configured the system to run a simulation with two tenants, $t = 0$ and $t = 1$, and used $u_0 = 0.1$, $u_1 = 0.9$, $v_0 = 1$ and $v_1 = 0.1$. This setup makes tenant 0 have higher network usage, and his penalty higher than tenant 1's, which imposes a lower usage.

We consider as baseline an admission agent that is not aware of the tenant who is making the request [82]. The proposed strategy adds as input to the admission agent

Figure 32 – Overall results. Prop is the proposed policy. BL is the baseline. RND is random. Fit accepts a slice if there are enough resources at admission time. And ACPT is accept all the slices. The left graph shows the overall loss achieved by each policy, BL has a loss of 271, while Prop has 262. In the middle, only the loss incurred by rejecting the slice is shown and, on the right, only the loss incurred by scaling.



Source: author [31].

the identifier of the tenant who is requesting the slice, as described in Section 4.2.2. Both agents are using a policy network with 4 hidden layers with 40 neurons in each and a rectifier activation function, following the baseline.

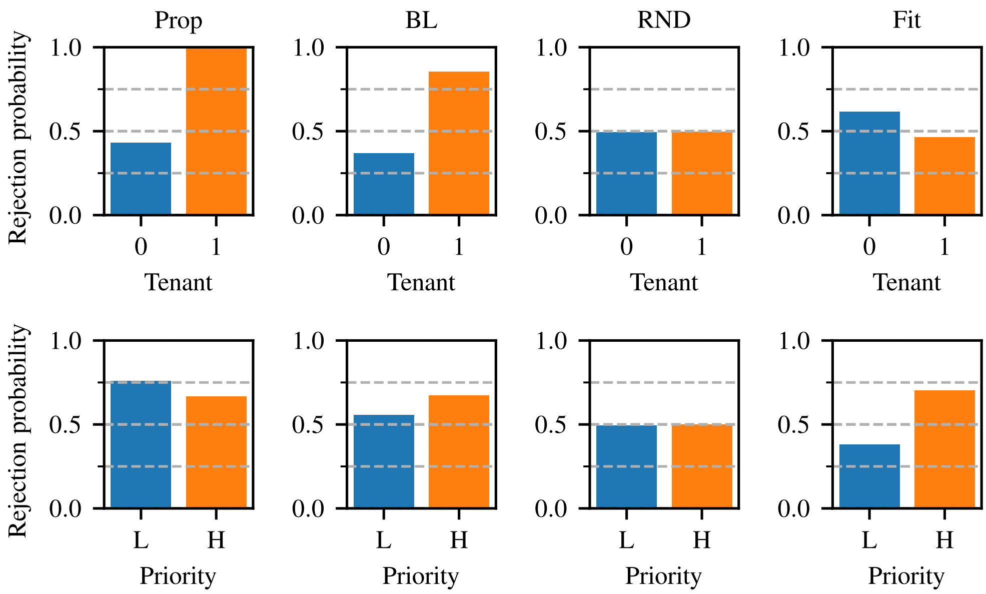
We trained the system using the REINFORCE with baseline algorithm [83] for 10,000 iterations, with 25 episodes per iteration, each episode with a load of 80 Erlangs and 600 arrivals. With the trained model, we ran the test for a new set of new 25 episodes with the same configuration.

Alongside our proposal (Prop), we show results for four other policies. BL is the baseline, which was adapted from Raza *et al.* [82]. With the exception of the tenant identifier, BL has the same input as Prop. RND is the random police, i.e., it chooses to accept or reject with equal probability. Fit accepts the slice if the InP has enough resources to fulfill its request at admission time. Those heuristics were defined and also used by Raza *et al.* [82].

Figure 32 summarizes the results. We can observe that accepting all the slices incurs a high scaling loss, which signals that some of the slices could have been rejected. Fit is too conservative and rejects too many slices causing resource underutilization. Randomly accepting the slices essentially accepts half of them. The baseline learns a better policy and achieves a balance between rejecting some slices and handling some scaling loss. However, it does not have information on which tenant is requesting the slice. Thus, it cannot achieve the performance of the proposed solution which can fine-tune the decision to the specific tenant, and consequently find a better balance.

We suppose that the baseline can still infer the tenant by the level of the usage. This level is indeed a function of the tenant (given u_t) and is present at its input. Nevertheless,

Figure 33 – Rejection probability for each policy (columns), marginalized by tenant (top line), or by priority (bottom line). ACPT policy is not shown, because it is zero for all cases.



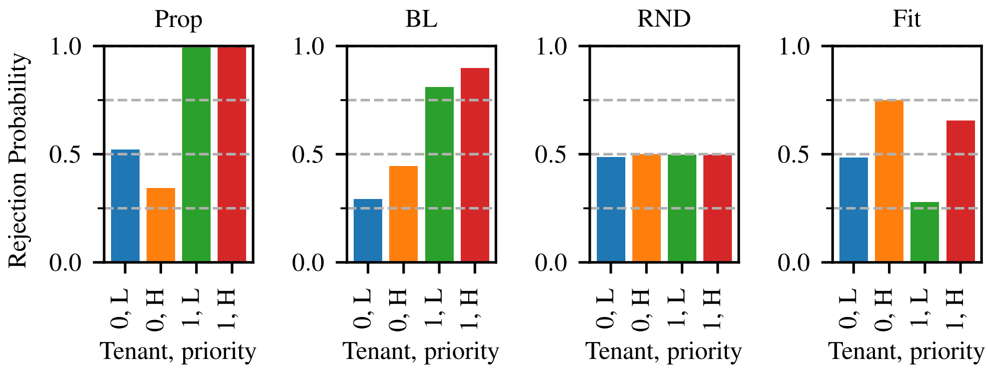
Source: author [31].

because the resource usage is noisy, it probably can not achieve the best possible information about the tenant, which is available for the proposed policy. To better understand which services the policies are choosing, we analyze the rejection probability for each class of slices.

We analyze which slices are being rejected in Figure 33. We can see that Fit rejects more slices of tenant 0. That happens because tenant 0 usually requests more resources when compared to tenant 1 (given that $u_0 < u_1$). Consequently, it is more probable that there are enough resources for its slices. Random accepts half in any marginalization by nature. The baseline accepts more slices from tenant 0. It should have learned that rejecting tenant 0 slices incurs a higher penalty ($v_0 > v_1$), but instead, it was probably inferred from the resource. However, the balance found by the baseline rejects more high-priority slices than low-priority ones. This is counter-intuitive, yet this can be due to no tenant awareness. For instance, the configuration of the system makes off-peak hours slices from tenant 0 and peak hours slices from tenant 1 have similar resource usage and priority, but different penalties. Finally, the proposed strategy seems to reject almost all the tenant 1 slices in exchange for a higher acceptance of tenant 0. It also manages to find a point where higher priority slices are more often accepted compared to lower priority ones.

Another way to investigate how slices are being classified is to examine the rejection probability marginalized by tenant and priority, shown in Figure 34. We can use the same

Figure 34 – Rejection probability for each policy (columns) marginalized by tenant and priority. Accept all is not shown since it is zero for all the cases.



Source: author [31].

rationale for Fit: it accepts more of the low priority from tenant 1, then low priority from tenant 0, then high priority from tenant 1 and high priority from tenant 0. That is precisely the expected sequence of slices if they are ordered by the expected magnitude of its resource request when compared to each other. Random once again accepted half of each class. For the baseline, we can see that it understands that some classes are more important than others. However, it seems that it can not identify that high-priority slices from tenant 0 are more valuable than low priority ones from the same tenant. That interpretation is one of the behaviors that the proposed method could learn. It also rejects most of the slices from tenant 1. That can happen because the load is high, and there are enough slices from tenant 0, so it can be lucrative enough. At lower loads the behavior can be different. Overall, we see that the proposed agent can improve the InP's revenue.

5 Final Considerations

This work had, in its two parts, a common characteristic: the application of multiple technologies and how its interworking might help to advance the field of mobile networks. The first part analyzed the integration of ICN into mobile networks, being both reasonably mature technologies, they allowed the development of a study prototype. The second part proposed the application of reinforcement learning for network management. Given the constraints of RL training, the low maturity of network slices and the need for diverse data, the work presented simulated results, which provided insights on what is important for this type of agent. Overall, those are valuable work for the academy, which is revealed by the accepted publications highlighted at Section 1.3.

This thesis presented the ICN deployment on the edge of a live mobile network and associated PoC. In contrast to the flat deployment of [15] that requires (new) ICN-enabled UE, the implemented testbed supports legacy UE. The thesis also contrasts how the different architectures might compare with current solutions, including CDN.

The results confirmed that ICN is a reasonable alternative to alleviate the redundant transmission in the backhaul. This is specially important because the backhaul is expected to be a bottleneck of the future mobile network, where the density of small cells can be very high.

The developed testbed presents an essential step in studying the implications of ICN adoption in mobile networks. Although multiple studies state that ICN adoption is beneficial, most of them do not analyze it in a prototyped network. A testbed helps to pinpoint aspects such as the limitation of current mobile hardware to execute ICN software, which did not provide enough throughput for the video transmission proof of concept.

Additionally, a prototyped network available for research enables the assessment of practical considerations of academical solutions. The presented testbed is a start point for such evaluation, for example, the addition of more elements (e.g., UE and eNodeB) would allow the validation, and potentially, improvement of work proposed in the literature for caching and mobility. In another dimension, the testbed also allows for new designs. For example, proposals of interlayer communication algorithms, such as using traffic information to optimize the scheduling in the radio layer.

The presented testbed also allows the evaluation of ICN-RAN concerning latency, caching, and software performance, as presented throughout Section 3.7. However, it still has limitations that call for improvement and future work. Specifically, the presented testbed does not account for multiple paths to the content, which would allow the evaluation of ICN's multicast routing benefits. Furthermore, the addition of more eNodeB would enable the assessment of implications incurred by mobility events.

Another expansion might include a flat architecture implementation, enabling the study of the consequences of implementing ICN directly over the PDCP layer. This development requires a software implementation of the UE so that both sides of the RAN have ICN incorporated.

On the second part of the thesis, the application of reinforcement learning for slice admission considered the tenant-specific penalties to InP when the InP attempts to oversubscribe its resources to make more revenue. The InP leverages the tenant-specific penalties to reduce the chance that of high SLA penalties.

The thesis modeled a reinforcement learning-based agent for slice admission control in virtualized 5G networks. We evaluated our proposal in a scenario where slice requests have a stochastic resource requirements footprint and admission control is non-trivial. Furthermore, we compared the performance against other related strategies. Our network model included the concept of different tenants that incur different revenues, consequently increasing the possible combinations of admission decisions.

In this context, we showed that the tenant-awareness contributes to a better admission policy and brings more revenue to the InP. This work is also important because it evaluates the network in a congested scenario, i.e., if the load is not high, the InP usually can accept all the slices. Hence, we compared all our results with the accept all policy, to show that at this load the InP can not accept all the slices. We do not envision that an InP would always operate at this load, it should be transient situations. For example, many slices might be requested because there might be a sports event in an area. Nevertheless, if the load stays high for an extended period of time, it is a signal that the InP should upgrade its infrastructure. Still, the upgrade is not a fast process, and while its capacity is not increased, this strategy allows the InP to explore the best possible revenue from its infrastructure.

Future work might include the evaluation of the policy agent in a more challenging system context where network and slices dynamics are increased, e.g., the proportion of tenants. The presented results considered a uniform number of requests for each tenant, but the number of slices deployed or requested for each tenant might change over time. We suppose that this creates some room in certain periods (when high-revenue tenant request is low) so that low-cost tenants can be admitted into the system. Those are characteristics that might be learned by the RL agent presented in this thesis. Thus, this continuous learning is one of the benefits of the RL agent, i.e., once modeled, it might be able to learn new system characteristics. However, the extent of this ability is yet to be evaluated.

Bibliography

[1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022", Cisco Systems, Inc., Feb. 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf (visited on 06/28/2019) (cit. on p. 1).

[2] Ericsson, "Ericsson Mobility Report June 2019", 2019, p. 36. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2019/ericsson-mobility-report-june-2019.pdf (visited on 06/28/2019) (cit. on pp. 1, 11).

[3] A. Morelli, M. Tortonesi, C. Stefanelli, and N. Suri, "Information-Centric Networking in next-generation communications scenarios", *Journal of Network and Computer Applications*, vol. 80, pp. 232–250, Feb. 15, 2017, ISSN: 1084-8045. DOI: 10.1016/j.jnca.2016.12.026. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516303423 (visited on 07/05/2019) (cit. on p. 1).

[4] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named Data Networking", *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014, ISSN: 0146-4833. DOI: 10.1145/2656877.2656887. [Online]. Available: http://doi.acm.org/10.1145/2656877.2656887 (visited on 06/28/2019) (cit. on pp. 1, 2, 9, 10, 53).

[5] J. Chen, M. Chiang, J. Erman, G. Li, K. K. Ramakrishnan, and R. K. Sinha, "Fair and optimal resource allocation for LTE multicast (eMBMS): Group partitioning and dynamics", in *2015 IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2015, pp. 1266–1274. DOI: 10.1109/INFOCOM.2015.7218502 (cit. on p. 1).

[6] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of Information-Centric Networking Research", *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1024–1049, Jul. 19, 2013, ISSN: 1553-877X. DOI: 10.1109/SURV.2013.070813.00063 (cit. on p. 1).

[7] D. Martín-Sacristán, J. F. Monserrat, J. Cabrejas-Peñuelas, D. Calabuig, S. Garrigas, and N. Cardona, "On the Way towards Fourth-Generation Mobile: 3GPP LTE and LTE-Advanced", *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 354 089, Dec. 2009, ISSN: 1687-1499. DOI: 10.1155/2009/354089. [Online]. Available: https://jwcn-eurasipjournals.springeropen.com/articles/10.1155/2009/354089 (visited on 06/28/2019) (cit. on p. 2).

[8] K. N. Patel and R. h. Jhaveri, “A Survey on Emulation Testbeds for Mobile Ad-hoc Networks”, *Procedia Computer Science*, International Conference on Advanced Computing Technologies and Applications (ICACTA), vol. 45, pp. 581–591, Jan. 1, 2015, ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.03.111. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915003476 (visited on 06/28/2019) (cit. on p. 2).

[9] M. Li, D. Pei, X. Zhang, B. Zhang, and K. Xu, “Interest-suppression-based NDN live video broadcasting over wireless LAN”, *Frontiers of Computer Science*, vol. 11, no. 4, pp. 675–687, Aug. 1, 2017, ISSN: 2095-2236. DOI: 10.1007/s11704-016-5563-x (cit. on p. 2).

[10] A. Ravi, P. Ramanathan, and K. M. Sivalingam, “Integrated network coding and caching in information-centric networks: Revisiting pervasive caching in the ICN framework”, *Photonic Network Communications*, vol. 30, no. 3, pp. 416–427, Dec. 1, 2015, ISSN: 1572-8188. DOI: 10.1007/s11107-015-0557-4 (cit. on p. 2).

[11] 5G Americas, “Understanding Information Centric Networking and Mobile Edge Computing”, 5G Americas, Dec. 2016. [Online]. Available: http://www.5gamericas.org/en/newsroom/press-releases/5g-americas-publishes-whitepaper-understanding-information-centric-networking-and-mobile-edge-computing/ (visited on 06/28/2019) (cit. on pp. 2, 38, 39, 41, 42).

[12] P. Suthar and M. Stolic, “Deploying information centric networking in LTE mobile networks”, in *2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, Sep. 2016, pp. 130–136. DOI: 10.1109/APWiMob.2016.7811432 (cit. on pp. 2, 34, 35, 38, 39, 42, 44).

[13] A. Gomes and T. Braun, “Feasibility of information-centric networking integration into LTE mobile networks”, ACM Press, 2015, pp. 627–633, ISBN: 978-1-4503-3196-8. DOI: 10.1145/2695664.2695790. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2695664.2695790 (visited on 06/13/2018) (cit. on pp. 2, 36, 37, 42, 44, 47, 48).

[14] W. Kim, “Beyond LTE-advance for information centric networking”, *Computer Standards & Interfaces*, vol. 49, pp. 59–66, Jan. 2017, ISSN: 09205489. DOI: 10.1016/j.csi.2016.08.007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0920548916300666 (visited on 06/13/2018) (cit. on pp. 2, 37, 38).

[15] R. Ravindran, A. Chakraborti, S. O. Amin, A. Azgin, and G. Wang, “5G-ICN: Delivering ICN Services over 5G Using Network Slicing”, *IEEE Communications Magazine*, vol. 55, no. 5, pp. 101–107, May 2017, ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600938 (cit. on pp. 2, 11, 42–44, 66).

[16] H. Jin, D. Xu, C. Zhao, and D. Liang, "Information-centric mobile caching network frameworks and caching optimization: A survey", *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, p. 33, Feb. 17, 2017, ISSN: 1687-1499. DOI: 10.1186/s13638-017-0806-6 (cit. on pp. 2, 3).

[17] Z. Zhang, C.-H. Lung, I. Lambadaris, and M. St-Hilaire, "When 5G meets ICN: An ICN-based caching approach for mobile video in 5G networks", *Computer Communications*, vol. 118, pp. 81–92, Mar. 1, 2018, ISSN: 0140-3664. DOI: 10.1016/j.comcom.2017.10.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366417305066 (visited on 06/28/2019) (cit. on pp. 2, 35, 36).

[18] A. S. Gomes, B. Sousa, D. Palma, V. Fonseca, Z. Zhao, E. Monteiro, T. Braun, P. Simoes, and L. Cordeiro, "Edge caching with mobility prediction in virtualized LTE mobile networks", *Future Generation Computer Systems*, vol. 70, pp. 148–162, May 1, 2017, ISSN: 0167-739X. DOI: 10.1016/j.future.2016.06.022. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16302072 (visited on 06/28/2019) (cit. on pp. 2, 3).

[19] G. Carofiglio, M. Gallo, and L. Muscariello, "Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation", *Computer Networks*, vol. 110, pp. 104–117, Dec. 9, 2016, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2016.09.012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616303024 (visited on 06/28/2019) (cit. on pp. 2, 3).

[20] Y. Nishiyama, M. Ishino, Y. Koizumi, T. Hasegawa, K. Sugiyama, and A. Tagami, "Proposal on routing-based mobility architecture for ICN-based cellular networks", in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 467–472. DOI: 10.1109/INFCOMW.2016.7562122 (cit. on pp. 2, 3).

[21] A. Gomes and T. Braun, "Load balancing in LTE mobile networks with Information-Centric Networking", in *2015 IEEE International Conference on Communication Workshop (ICCW)*, Jun. 2015, pp. 1845–1851. DOI: 10.1109/ICCW.2015.7247449 (cit. on pp. 2, 3, 39).

[22] A. V. Ventrella, G. Piro, and L. A. Grieco, "Publish-subscribe in mobile information centric networks: Modeling and performance evaluation", *Computer Networks*, vol. 127, pp. 317–339, Nov. 9, 2017, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2017.08.022. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128617303377 (visited on 06/28/2019) (cit. on pp. 2, 3).

[23] P. Suthar, M. Stolic, A. Jangam, and D. Trossen, "Native deployment of ICN in LTE, 4G mobile networks", IETF Secretariat, Internet-Draft draft-suthar-icnrg-

icn-lte-4g-04, Nov. 2017. [Online]. Available: http://www.ietf.org/internet-drafts/draft-suthar-icnrg-icn-lte-4g-04.txt (cit. on p. 3).

[24] P. Gomes, A. Vidal, and S. Lins. (Nov. 2, 2018). Next stop: Zero-touch automation standardization, [Online]. Available: https://www.ericsson.com/research-blog/next-stop-zero-touch-automation-standardization/ (visited on 12/07/2018) (cit. on p. 3).

[25] A. Oliva, X. Li, X. Costa-Perez, C. J. Bernardos, P. Bertin, P. Iovanna, T. Deiss, J. Mangues, A. Mourad, C. Casetti, J. E. Gonzalez, and A. Azcorra, “5G-TRANSFORMER: Slicing and Orchestrating Transport Networks for Industry Verticals”, *IEEE Communications Magazine*, vol. 56, no. 8, pp. 78–84, Aug. 2018, ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1700990 (cit. on p. 3).

[26] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, “From network sharing to multi-tenancy: The 5G network slice broker”, *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, Jul. 2016, ISSN: 0163-6804. DOI: 10.1109/MCOM.2016.7514161 (cit. on pp. 4, 54).

[27] P. Caballero, A. Banchs, G. de Veciana, X. Costa-Perez, and A. Azcorra, “Network Slicing for Guaranteed Rate Services: Admission Control and Resource Allocation Games”, *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6419–6432, Oct. 2018, ISSN: 1536-1276, 1558-2248. DOI: 10.1109/TWC.2018.2859918. [Online]. Available: https://ieeexplore.ieee.org/document/8424561/ (visited on 10/12/2018) (cit. on p. 4).

[28] J. Zheng, P. Caballero, G. de Veciana, S. J. Baek, and A. Banchs, “Statistical Multiplexing and Traffic Shaping Games for Network Slicing”, *IEEE/ACM Transactions on Networking*, pp. 1–14, 2018, ISSN: 1063-6692. DOI: 10.1109/TNET.2018.2870184 (cit. on p. 4).

[29] B. Han, D. Feng, and H. D. Schotten, “A Markov Model of Slice Admission Control”, *IEEE Networking Letters*, pp. 1–1, 2018, ISSN: 2576-3156. DOI: 10.1109/LNET.2018.2873978 (cit. on p. 4).

[30] P. Batista, I. Araújo, N. Linder, K. Laraqui, and A. Klautau, “Testbed for ICN media distribution over LTE radio access networks”, *Computer Networks*, vol. 150, pp. 70–80, Feb. 26, 2019, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2018.12.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S138912861831380X (visited on 07/05/2019) (cit. on pp. 5, 12, 15, 43–45, 47–50, 52).

[31] P. Batista, S. N. Khan, P. Öhlén, and A. Klautau, "Tenant-Aware Slice Admission Control Using Neural Networks-Based Policy Agent", in *Cognitive Radio-Oriented Wireless Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Poznan, Poland: Springer International Publishing, 2019, pp. 17–30, ISBN: 978-3-030-25748-4. DOI: 10.1007/978-3-030-25748-4_2 (cit. on pp. 5, 57, 59, 60, 62–65).

[32] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO Data for Machine Learning: Application to Beam-Selection Using Deep Learning", in *2018 Information Theory and Applications Workshop (ITA)*, Feb. 2018, pp. 1–9. DOI: 10.1109/ITA.2018.8503086 (cit. on p. 5).

[33] C. Natalino, M. R. Raza, P. Öhlén, P. Batista, M. Santos, L. Wosinska, and P. Monti, "Machine-Learning-Based Routing of QoS-Constrained Connectivity Services in Optical Transport Networks", in *Advanced Photonics 2018 (BGPP, IPR, NP, NOMA, Sensors, Networks, SPPCom, SOF) (2018), Paper NeW3F.5*, Optical Society of America, Jul. 2, 2018, NeW3F.5. DOI: 10.1364/NETWORKS.2018.NeW3F.5. [Online]. Available: https://www.osapublishing.org/abstract.cfm?uri=Networks-2018-NeW3F.5 (visited on 07/12/2019) (cit. on p. 5).

[34] P. Batista, L. Brito, E. Oliveira, M. Neto, and A. Klautau, "Pilot Projects for GSM Community Networks in Amazon", presented at the Workshop of ICT for Development (WTICp/D), Belém, Brazil: Brazilian Computer Society, 2017. [Online]. Available: https://sbrc2017.ufpa.br/wp-content/uploads/2017/08/proceedingsWTICpD2017v2.pdf (visited on 07/12/2019) (cit. on p. 5).

[35] B. Vilas Boas, M. Dias, P. Batista, A. Oliveira, and A. Klautau, "CELCOM Project: Engineering Practice via Community Networks in Amazon", *International Journal of Engineering Education (IJEE)*, to appear, Special Issue on Open Source & Collaborative PBL in Engineering Education 2019, accepted, to appear (cit. on p. 6).

[36] C. Batista, R. Cunha, P. Batista, A. Klautau, and N. Neto, "Utterance Copy in Formant-based Speech Synthesizers Using LSTM Neural Networks", presented at the Brazilian Conference on Intelligent Systems (BRACIS), Salvador, Brazil: Brazilian Computer Society, 2019, accepted, to appear (cit. on p. 6).

[37] M. Dias, J. Paulo, V. Soares, P. Batista, and A. Klautau, "Techno-economic study to connect Amazon using community cellular networks", presented at the Workshop of ICT for Development (WTICp/D), Belém, Brazil: Brazilian Computer Society, 2017. [Online]. Available: https://sbrc2017.ufpa.br/wp-content/uploads/2017/08/proceedingsWTICpD2017v2.pdf (visited on 07/12/2019) (cit. on p. 6).

[38] C. Nahum, J. Soares, P. Batista, and A. Klautau, "Emulation of 4G/5G Network Using OpenAirInterface", presented at the XXXV Brazilian Communications and Signal Processing Symposium, São Pedro, Brazil, 2017, ISBN: 978-85-66836-18-9. [Online]. Available: http://www.sbrt.org.br/sbrt2017/anais/anais_sbrt_2017.pdf (cit. on p. 6).

[39] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu, "Understanding Performance of Edge Content Caching for Mobile Video Streaming", *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1076–1089, May 2017, ISSN: 0733-8716. DOI: 10.1109/JSAC.2017.2680958 (cit. on p. 8).

[40] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content", in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, (Rome, Italy), ser. CoNEXT '09, New York, NY, USA: ACM, 2009, pp. 1–12, ISBN: 978-1-60558-636-6. DOI: 10.1145/1658939.1658941 (cit. on p. 8).

[41] B. Wissingh, C. Wood, A. Afanasyev, L. Zhang, D. Oran, and C. Tschudin, "Information-centric networking (ICN): CCN and NDN terminology", IETF Secretariat, Internet-Draft draft-wissingh-icnrg-terminology-03, Sep. 2017. [Online]. Available: http://www.ietf.org/internet-drafts/draft-wissingh-icnrg-terminology-03.txt (cit. on p. 8).

[42] S. H. Ahmed, S. H. Bouk, and D. Kim, "Content-Centric Networks (CCN)", in *Content-Centric Networks: An Overview, Applications and Research Challenges*, ser. SpringerBriefs in Electrical and Computer Engineering, S. H. Ahmed, S. H. Bouk, and D. Kim, Eds., Singapore: Springer Singapore, 2016, pp. 35–80, ISBN: 978-981-10-0066-9. DOI: 10.1007/978-981-10-0066-9_3 (cit. on p. 11).

[43] ITU-R, "Requirements related to technical performance for IMT-advanced radio interface(s)", Tech. Rep., Jan. 2008. [Online]. Available: http://www.itu.int/pub/R-REP-M.2134-2008/en (cit. on p. 11).

[44] R. Kreher and K. Gaenger, "LTE signaling, troubleshooting, and optimization", Jan. 2010. DOI: 10.1002/9780470977729 (cit. on pp. 13, 14, 18, 20, 24).

[45] M. Olsson, S. Sultana, S. Rommer, L. Frid, and C. Mulligan, "SAE and the evolved packet core", Jan. 2009. [Online]. Available: http://openlibrary.org/book/OL25402812M (cit. on pp. 15, 17).

[46] T. Ali-Yahiya, *Understanding LTE and its Performance*. Springer, Jan. 2011, ISBN: 978-1-4419-6457-1. DOI: 10.1007/978-1-4419-6457-1 (cit. on pp. 16, 19).

[47] Cmyoo, "LTE GTP tunnel i", Sep. 2013. [Online]. Available: https://www.netmanias.com/ko/post/blog/5836/eps-bearer-lte/lte-gtp-tunnel-i (cit. on p. 18).

[48] S. Ahmadi, *LTE-Advanced: A Practical Systems Approach to Understanding 3GPP LTE Releases 10 and 11 Radio Access Technologies*, ser. EngineeringPro collection. Elsevier Science, Oct. 2013, ISBN: 9780124051782. [Online]. Available: `https://books.google.com.br/books?id=DTsTAAAAQBAJ` (cit. on pp. 23, 25).

[49] R. Alvizu, S. Troia, G. Maier, and A. Pattavina, "Matheuristic with machine-learning-based prediction for software-defined mobile metro-core networks", *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 9, pp. D19–D30, Sep. 2017, ISSN: 1943-0620. DOI: `10.1364/JOCN.9.000D19` (cit. on p. 24).

[50] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second edition, ser. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018, 526 pp., ISBN: 978-0-262-03924-6 (cit. on pp. 26, 28, 29).

[51] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning", *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, ISSN: 1476-4687. DOI: `10.1038/nature14236`. [Online]. Available: `https://www.nature.com/articles/nature14236` (visited on 07/10/2019) (cit. on p. 28).

[52] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms", Jul. 19, 2017. arXiv: `1707.06347 [cs]`. [Online]. Available: `http://arxiv.org/abs/1707.06347` (visited on 07/10/2019) (cit. on p. 28).

[53] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning", in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, (New York, NY, USA), ser. ICML'16, JMLR.org, 2016, pp. 1928–1937. [Online]. Available: `http://dl.acm.org/citation.cfm?id=3045390.3045594` (visited on 07/10/2019) (cit. on p. 28).

[54] 3GPP, "Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS)", 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 24.301 V. 12.13.0, Jan. 2017 (cit. on p. 34).

[55] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Scalable mobile backhauling via information-centric networking", in *The 21st IEEE International Workshop on Local and Metropolitan Area Networks*, Apr. 2015, pp. 1–6. DOI: `10.1109/LANMAN.2015.7114719` (cit. on pp. 34, 39, 40).

[56] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, and M. Varvello, “From content delivery today to information centric networking”, *Computer Networks*, Information Centric Networking, vol. 57, no. 16, pp. 3116–3127, Nov. 13, 2013, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2013.07.002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613002156 (visited on 07/03/2019) (cit. on pp. 35, 36).

[57] U. Doetsch, N. Bayer, H. Droste, A. Roos, T. Rosowski, G. Zimmermann, P. Agyapong, N. F. Lindqvist, I. D. Silva, H. Eriksson, H. Tullberg, Ö. Bulakci, J. Eichinger, K. Dimtsa, M. Stamatelatos, V. Venkatkumar, M. Boldi, and I. Berberana, “Deliverable D6.4 final report on architecture”, Mobile and wireless communications enablers for the twenty-twenty information society (METIS), Tech. Rep., 2015 (cit. on pp. 35, 48).

[58] S. Ren, T. Lin, W. An, G. Zhang, D. Wu, L. N. Bhuyan, and Z. Xu, “Design and analysis of collaborative EPC and RAN caching for LTE mobile networks”, *Computer Networks*, vol. 93, pp. 80–95, Dec. 24, 2015, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2015.10.012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128615003850 (visited on 07/03/2019) (cit. on p. 35).

[59] M. Mangili, J. Elias, F. Martignon, and A. Capone, “Optimal planning of virtual content delivery networks under uncertain traffic demands”, *Computer Networks*, vol. 106, pp. 186–195, Sep. 4, 2016, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2016.06.035. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128616302158 (visited on 07/03/2019) (cit. on p. 35).

[60] C. A. Garcia-Perez and P. Merino, “Enabling Low Latency Services on LTE Networks”, in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, Sep. 2016, pp. 248–255. DOI: 10.1109/FAS-W.2016.59 (cit. on p. 35).

[61] D. Kutscher, “It’s the network: Towards better security and transport performance in 5G”, in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 656–661. DOI: 10.1109/INFCOMW.2016.7562158 (cit. on pp. 35, 36).

[62] J. Qiao, Y. He, and X. S. Shen, “Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks”, *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 7187–7198, Oct. 2016, ISSN: 1536-1276. DOI: 10.1109/TWC.2016.2598748 (cit. on p. 36).

[63] Z. Ming, M. Xu, and D. Wang, “InCan: In-network cache assisted eNodeB caching mechanism in 4G LTE networks”, *Computer Networks*, vol. 75, pp. 367–380, Dec. 24, 2014, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2014.10.021. [Online]. Available:

http://www.sciencedirect.com/science/article/pii/S1389128614003831 (visited on 07/03/2019) (cit. on p. 36).

[64] Cisco, "The cisco 5G strategy series: Packet core, transport, and identity management", 2016. [Online]. Available: https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/ultra-services-platform/5g-strategy-series.pdf (cit. on p. 36).

[65] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS adoption on the web", 2017 (cit. on p. 37).

[66] C. Casetti, C.-F. Chiasserini, F. Malandrino, and C. Borgiattino, "Area formation and content assignment for LTE broadcasting", *Computer Networks*, vol. 126, pp. 174–186, Oct. 24, 2017, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2017.07.006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128617302876 (visited on 07/04/2019) (cit. on p. 39).

[67] E. Ahmed and M. H. Rehmani, "Mobile Edge Computing: Opportunities, solutions, and challenges", *Future Generation Computer Systems*, vol. 70, pp. 59–63, May 1, 2017, ISSN: 0167-739X. DOI: 10.1016/j.future.2016.09.015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X16303260 (visited on 07/04/2019) (cit. on p. 39).

[68] A. Detti, C. Pisa, and N. B. Melazzi, "Modeling multipath forwarding strategies in Information Centric Networks", in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 324–329. DOI: 10.1109/INFCOMW.2015.7179405 (cit. on p. 39).

[69] G. Carofiglio, L. Mekinda, and L. Muscariello, "FOCAL: Forwarding and Caching with Latency Awareness in Information-Centric Networking", in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–7. DOI: 10.1109/GLOCOMW.2015.7413972 (cit. on p. 39).

[70] R. Chiocchetti, D. Perino, G. Carofiglio, D. Rossi, and G. Rossini, "INFORM: A Dynamic Interest Forwarding Mechanism for Information Centric Networking", in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, (Hong Kong, China), ser. ICN '13, New York, NY, USA: ACM, 2013, pp. 9–14, ISBN: 978-1-4503-2179-2. DOI: 10.1145/2491224.2491227. [Online]. Available: http://doi.acm.org/10.1145/2491224.2491227 (visited on 07/04/2019) (cit. on p. 39).

[71] S.-E. Elayoubi and J. Roberts, "Performance and Cost Effectiveness of Caching in Mobile Access Networks", in *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*, (San Francisco, California, USA), ser. ACM-ICN '15, New York, NY, USA: ACM, 2015, pp. 79–88, ISBN: 978-1-4503-3855-4. DOI: 10.1145/2810156.

2810168. [Online]. Available: http://doi.acm.org/10.1145/2810156.2810168 (visited on 07/04/2019) (cit. on p. 40).

[72] A. Dabirmoghaddam, M. Dehghan, and J. J. Garcia-Luna-Aceves, "Characterizing Interest aggregation in content-centric networks", in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, Vienna, Austria: IEEE, May 2016, pp. 449–457, ISBN: 978-3-901882-83-8. DOI: 10.1109/IFIPNetworking.2016.7497205. [Online]. Available: http://ieeexplore.ieee.org/document/7497205/ (visited on 07/04/2019) (cit. on p. 40).

[73] E. Pateromichelakis, J. Gebert, T. Mach, J. Belschner, W. Guo, N. P. Kuruvatti, V. Venkatasubramanian, and C. Kilinc, "Service-Tailored User-Plane Design Framework and Architecture Considerations in 5G Radio Access Networks", *IEEE Access*, vol. 5, pp. 17 089–17 105, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2736579 (cit. on pp. 44, 47).

[74] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet, "OpenAirInterface: A Flexible Platform for 5G Research", *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 33–38, Oct. 2014, ISSN: 0146-4833. DOI: 10.1145/2677046.2677053. [Online]. Available: http://doi.acm.org/10.1145/2677046.2677053 (visited on 07/05/2019) (cit. on p. 46).

[75] F. K. Jondral, "Software-Defined Radio—Basics and Evolution to Cognitive Radio", *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 3, p. 652 784, Aug. 1, 2005, ISSN: 1687-1499. DOI: 10.1155/WCN.2005.275 (cit. on p. 46).

[76] 3GPP, "General packet radio service (GPRS) enhancements for evolved universal terrestrial radio access network (e-UTRAN) access", 3rd Generation Partnership Project (3GPP), Tech. Rep. TS 23.401 V. 12.11.0, Apr. 2016 (cit. on p. 46).

[77] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization", in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9. DOI: 10.1109/INFOCOM.2017.8057230 (cit. on p. 54).

[78] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing", in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9. DOI: 10.1109/INFOCOM.2017.8057045 (cit. on p. 55).

[79] AT&T, BT, CenturyLink, C. Mobile, Colt, D. Telekom, KDDI, NTT, Orange, T. Italia, Telefonica, Telstra, and Verizon, "Network Functions Virtualisation. An Introduction, Benefits, Enablers, Challenges & Call for Action", SDN and OpenFlow

World Congress, Oct. 2012. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf (visited on 12/28/2018) (cit. on p. 55).

[80] S. Dahmen-Lhuissier. (Dec. 28, 2018). Open Source MANO, [Online]. Available: https://www.etsi.org/technologies-clusters/technologies/nfv/open-source-mano (visited on 12/28/2018) (cit. on p. 55).

[81] ONAP, "ONAP Architecture Overview", Dec. 27, 2018. [Online]. Available: https://www.onap.org/wp-content/uploads/sites/20/2018/06/ONAP_CaseSolution_Architecture_0618FNL.pdf (visited on 12/27/2018) (cit. on p. 55).

[82] M. R. Raza, C. Natalino, P. Öhlen, L. Wosinska, and P. Monti, "A Slice Admission Policy Based on Reinforcement Learning for a 5G Flexible RAN", in *2018 European Conference on Optical Communication (ECOC)*, Sep. 2018, pp. 1–3. DOI: 10.1109/ECOC.2018.8535483 (cit. on pp. 58, 61–63).

[83] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource Management with Deep Reinforcement Learning", ACM Press, 2016, pp. 50–56, ISBN: 978-1-4503-4661-0. DOI: 10.1145/3005745.3005750. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3005745.3005750 (visited on 03/12/2018) (cit. on pp. 61, 63).

[84] M. R. Raza, A. Rostami, L. Wosinska, and P. Monti, "Resource Orchestration Meets Big Data Analytics: The Dynamic Slicing Use Case", in *2018 European Conference on Optical Communication (ECOC)*, Rome: IEEE, Sep. 2018, pp. 1–3, ISBN: 978-1-5386-4862-9. DOI: 10.1109/ECOC.2018.8535581. [Online]. Available: https://ieeexplore.ieee.org/document/8535581/ (visited on 02/15/2019) (cit. on p. 61).

[85] M. R. Raza, A. Rostami, A. Vidal, M. A. S. Santos, L. Wosinska, and P. Monti, "Priority-Aware Service Orchestration Using Big Data Analytics for Dynamic Slicing in 5G Transport Networks", in *2017 European Conference on Optical Communication (ECOC)*, Sep. 2017, pp. 1–3. DOI: 10.1109/ECOC.2017.8346133 (cit. on p. 62).