UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

MAICON KIST

Radio and Baseband Unit Virtualization: Pushing the Boundaries of Future Mobile Networks

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Advisor: Prof. Dr. Juergen Rochol
Coadvisor: Prof. Dr. Cristiano Bonato Both

Porto Alegre
August 2020

CIP — CATALOGING-IN-PUBLICATION

Kist, Maicon

Radio and Baseband Unit Virtualization: Pushing the Boundaries of Future Mobile Networks / Maicon Kist. – Porto Alegre: PPGC da UFRGS, 2020.

112 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2020. Advisor: Juergen Rochol; Coadvisor: Cristiano Bonato Both.

1. Software defined radio. 2. Virtualization. 3. Baseband signal processing. 4. Functional split. 5. 5G. 6. Network function virtualization. I. Rochol, Juergen. II. Both, Cristiano Bonato. III. Título.

“Few can foresee wither their road will lead them, till they come to it’s end.”

— J.R.R. TOLKIEN

ABSTRACT

Existing mobile networks rely on closed and inflexible hardware-based architectures both at the radio frontend and in the core network. This hardware dependence significantly delays the adoption and deployment of new standards, impose significant challenges in implementing innovative new radio access technologies to maximize the network capacity and coverage, and prevent provisioning of truly-differentiated services that can adapt to uneven and highly variable traffic patterns. Because of such limitations, it is expected that future 5G mobile networks should be constructed based on the baseband centralization architecture. In such an architecture, the computational processing resources are centralized and software-defined implementations replace baseband unit hardware. Although baseband centralization architecture brings several opportunities for future 5G mobile networks, it is not free from challenges. Among the different challenges existent, we highlight the most relevant ones, (*i*) a single "one-size-fits-all" access technology, (*ii*) network coverage is limited, and (*iii*) high bandwidth requirements for fronthaul links. To address these challenges, we push the boundaries of mobile network architectures by presenting the AIRTIME. AIRTIME is a prototype system that integrates BBU and RRH virtualization to realize a flexible and adaptable solution in which the physical RAN infrastructure can be virtualized and specialized for any type of radio access technology. Our RRH virtualization layer enables multiple heterogeneous access technologies to coexist on top of the same radio front-end. It uses innovative baseband processing techniques to slice and abstract a radio front-end into multiple virtual radio front-ends. AIRTIME solves the challenges of current centralized baseband architectures while enabling an unprecedented control over any aspect of the access network. The methodology employed to show the feasibility of our proposal is through an experimental prototype, which was evaluated in an experimental 5G-like network in which one physical RAN is virtualized into two vRANs to provide connectivity services to service providers with vastly different requirements. We also have performed mathematical analysis to obtain the fronthaul bandwidth and processing resources required for different fine-grained vBBU distribution, as well performed simulations to obtain the maximum number of vBBUs that can operate over a constrained fronthaul. Our results show that AIRTIME is able to execute multiple virtual RAN, enabling multiple RANs tailored for particular services to share the same physical infrastructure. Moreover, AIRTIME increases the programmability, flexibility, and scalability that is lacking in future 5G mobile networks, while also catalyzing innovations in a range of areas, from the introduction of new access technologies specialized in specific services, to the management of data center and fronthaul resources.

Keywords: Software defined radio. virtualization. baseband signal processing. functional split. 5G. network function virtualization.

Virtualização de rádio e processamento de sinais: impulsionando os limites das futuras redes móveis

RESUMO

As redes móveis atuais são baseadas em uma arquitetura fechada e inflexível tanto no *front-end* de rádio quando no núcleo da rede. A dependência com o *hardware* dificulta o desenvolvimento de novos padrões, impõe desafios na implantação de novas tecnologias de acesso que maximizam a capacidade e cobertura da rede e impede o provisionamento de serviços que podem realmente se adaptar à diferentes tráfegos de rede. Devido a essas limitações, espera-se que a futura rede 5G seja baseada na arquitetura de centralização de banda-base. Nessa arquitetura, os recursos computacionais de processamento são centralizados e *hardwares* de processamento banda-base são substituídos por versões implementadas em software. Apesar da arquitetura de centralização de banda-base trazer muitas oportunidades para as futuras redes 5G, sua implantação não é livre de desafios. Dentre os diversos desafios existentes, destacam-se os mais importantes (*i*) uma tecnologia de acesso para todos os serviços, (*ii*) área de cobertura da rede é limitada, e (*iii*) grande largura de banda na rede de *fronthaul*. Para resolver esses desafios, expande-se as fronteiras das redes móveis com o AIRTIME. AIRTIME é um protótipo que integra virtualização de BBUs e RRHs para realizar uma solução flexível e adaptável na qual a infraestrutura física da rede de acesso pode ser virtualizada e especializada para qualquer tipo de tecnologia. A camada de virtualização de RRHs dessa proposta permite que várias tecnologias de acesso heterogêneas coexistam em cima da mesma RRH física. Nossa proposta utiliza técnicas de processamento de sinais avançadas para dividir e abstrair uma *front-end* de rádio em múltiplos *front-end* virtuais. AIRTIME resolve os desafios da centralização de banda-base ao mesmo que tempo em que habilita um controle sem precedentes de qualquer aspecto da rede de acesso. A metodologia empregada para mostrar a viabilidade dessa proposta é através de um protótipo que foi avaliado em uma rede experimental 5G em que uma RRH é virtualizada em duas vRRHs que provêm conectividade para serviços com requisitos diferentes. Nós também realizamos análises matemáticas para obter a largura de banda e processamento necessários para diferentes distribuições de vBBUs, assim como simulações para obter o número máximo de vBBUs que podem compartilhar um *fronthaul* com largura de banda limitada. Os resultados mostram que AIRTIME é capaz de multiplexar diversas redes de acesso virtuais em uma única rede física e permitindo que as redes virtualizadas sejam personalizadas para serviços específicos. Além disso, AIRTIME aumenta a programabilide, flexibilidade e a escalabilidade necessárias nas futuras redes móveis 5G, ao mesmo tempo que cataliza inovações em um número de áreas, da inclusão de tecnologias de acesso especializadas em determinados serviços, até o gerenciamento de *data centers* e recursos da rede de *fronthaul*.

Palavras-chave: radio definido por software, virtualização, processamento de sinais, divisão de funções.

5G	Fifth Generation of Mobile Networks
ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
API	Application Programming Interface
BBU	BaseBand Unit
BS	Base Station
CN	Core Network
CP	Ciclic Prefix
CPRI	Common Public Radio Interface
CU	Centralized Unit
DAC	Digital-to-Analog Converter
DSA	Dynamic Spectrum Access
DSP	Digital Signal Processor
DU	Distributed Unit
EE	Energy Efficiency
eMBBC	enhanced Mobile BroadBand Communications
FDD	Frequency Division Duplexing
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GPP	General Purpose Processor
HARQ	Hybrid Automatic Repeat-reQuest
HyDRA	HYpervisor for software-Defined RAdio
IQ	In-phase & Quadrature
IFFT	Inverse FFT

IoT	Internet-of-Things
LS-CMA	Large-Scale Cooperative Multiple Antenna Processing
LTE-A	Long Term Evolution-Advanced
MAC	Medium Access Control
MANO	Management & Orchestration
MCS	Modulation and Coding Scheme
MIMO	Multiple-Input Multiple-Output
mMTC	Massive Machine Type Communications
MSE	Mean Square Error
MVNO	Mobile Virtual Network Operator
NB-IoT	Narrow-Band IoT
NFV	Network Function Virtualization
NS	Network Service
NVS	Network Virtualization Substrate
OFDM	Orthogonal Frequency-Division Multiplexing
OSM	Open Source Mano
OSS/BSS	Operations Support System/Business Support System
PDCP	Packet Data Convergent Protocol
PRB	Physical Resource Block
QoS	Quality-of-Service
RAN	Radio Access Network
RANaaS	RAN-as-a-Service
RAT	Radio Access Technology
RE	Resource Element
RF	Radio Frequency

RLC	Radio Link Control
RoF	Radio-Over-Fiber
RRH	Remote Radio Head
RX	Receiver
SAU	Spectrum Allocation Unit
SCBE	Spectrum Configuration and Bandwidth Estimation
SDN	Software-Defined Network
SDR	Software-Defined Radio
SE	Spectral Efficiency
SIMD	Single Instruction Multiple Data
SINR	Signal-to-Interference-plus-Noise-Ratio
SP	Service Provider
SVL	Spectrum Virtualization Layer
TDD	Time-Division Duplexing
URLLC	Ultra-Reliable and Low-Latency Communication
USRP	Universal Software Radio Peripheral
vBBU	Virtual BaseBand Unit
VNF	Virtual Network Function
VR	Virtual Radio
vRAN	Virtual RAN
vRRH	Virtual Remote Radio Head
WiMAX	Worldwide Interoperability for Microwave Access

LIST OF FIGURES

LIST OF TABLES

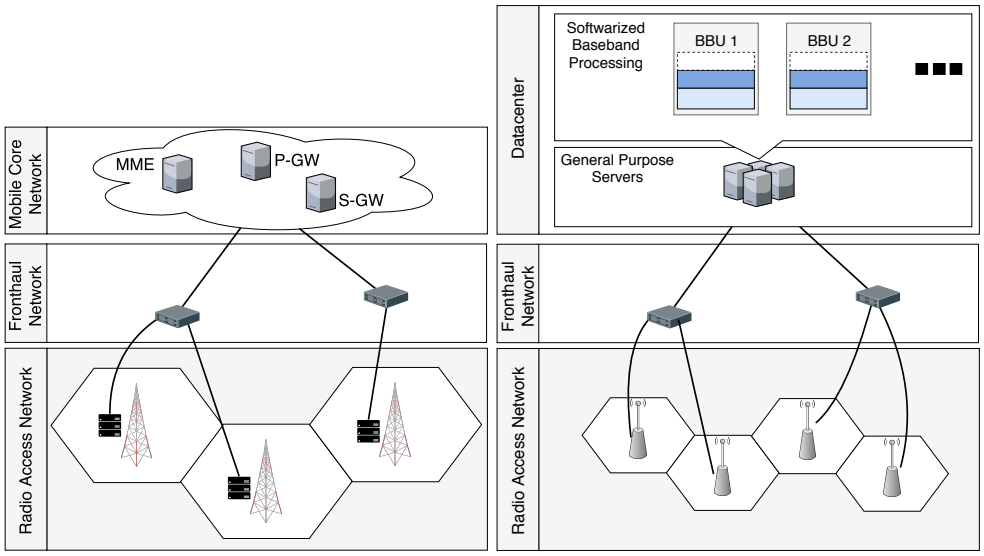
CONTENTS

1 INTRODUCTION

In recent years, wireless technology has emerged as one of the most significant trends in networking. Recent reports show that wireless broadband penetration has exceeded that of wired broadband networks. Moreover, recent advances in wireless communications and user-device processing capabilities have made it possible for wireless networks to provide a wide variety of multimedia applications and compelling wireless services. This trend is expected to continue in the future at a much faster growth rate. By 2021, the global mobile traffic will increase from 17 to 49 exabytes per month (CISCO, 2017). Addressing the expected exponential growth of rich multimedia applications and innovative services underscores the need to evolve mobile networks. To this end, the Fifth Generation of Mobile Networks (5G) is expected to support 1000x the aggregate data rate, 100x the user data rate, and 5x decrease in end-to-end latency, all while connecting 100x more devices (SEXTON et al., 2017). To meet with the expected three orders of magnitude capacity improvement and massive device connectivity, 5G centers its design objectives around programmability, flexibility, and scalability (QUINTANA-RAMIREZ et al., 2019).

The requirements for 5G can almost be considered contradictory in many ways. It is difficult to imagine a radio access technology optimized to provide data rates of Gbps to a virtual reality application, while also being optimized to provide connectivity to thousands of low data rate sensors. To settle this apparent contradiction, it is important to highlight that not every service requires each of the requirements mentioned above. Multimedia services, classified as enhanced Mobile BroadBand Communications (eMBBC), may demand access technologies optimized for high-data rates but may consist of only a few dozen of devices connected to the base station. On the other hand, a Internet-of-Things (IoT) service, classified as Massive Machine Type Communications (mMTC), is likely to consist of thousands of devices with low power consumption requirements, but not high data rates.

Recently, considerable effort has been devoted to investigating 5G mobile networks, and the drawback of the conventional mobile architecture has been realized (LIU et al., 2014a). In a conventional mobile network, shown in Figure 1.1(a), the physical infrastructure comprises geographically distributed Base Stations (BSs). These BSs are often proprietary devices that are highly integrated and optimized for a particular Radio Access Technology (RAT). Because of this, it is expected that future mobile networks, *i.e.*, 5G, will become increasingly software-defined and centralized. The baseband processing centralization architecture was proposed for Radio Access Networks (RANs), leveraging the cloud concept and technology. Figure 1.1(b) shows the major components of this architecture: (*i*) the Remote Radio Head (RRH), responsible for signal digitization, forwarding the digital signal to a central data center in the uplink or transmitting the digital signal received from the data center in the downlink, (*ii*) the data center, which provides the processing resources to run the software version of the BaseBand Unit (BBU), and (*iii*) the software implementing the full-blown BBU. The functionality split



(a) Conventional mobile network (b) Baseband centralized mobile network

Figure 1.1 – Evolution from a conventional mobile network to a softwarized and baseband centralized mobile network

allows mobile operators to expand the mobile network coverage and capacity quickly, simply by deploying RRHs, connecting them to the data center through a fronthaul network and allocating the processing resources for the BBU software.

While the baseband centralization is a significant improvement towards a mobile network that can provide connectivity services to a massive number of devices, it lacks the expected programmability, flexibility, and scalability. First, the current design of baseband centralization mandates a one-to-one mapping between RRHs and the BBU. Although intuitive, fixed mapping forces a RRH to provide only one access technology tailored to only one type of service. Second, all the signal processing is consolidated into the computing resource pool. As the central pool is likely to be far away from RRHs, fronthaul networking and statistical multiplexing may increase the total access latency, making it challenging to apply BBU centralization to some crucial scenarios such as extremely Ultra-Reliable and Low-Latency Communication (URLLC) (LARSEN; CHECKO; CHRISTIANSEN, 2019). Last but not least, BBU centralization requires the continuous exchange of raw baseband samples between RRHs and the central pool. Transportation of raw samples puts a tremendous bandwidth burden on the fronthaul network and may hinder the adoption of BBU centralization in networks with limited fronthaul resources.

The first drawback, the one-to-one mapping from RRH to BBU, forces the adoption of only one access technology. However, a single “one-size-fits-all” access technology is unable to address the diverging performance requirements imposed by future mobile services (OSSEIRAN et al., 2014; SANTOS et al., 2019). Radio virtualization is a promising solution to enable multiple access technologies on top of one physical antenna (TALEB et al., 2015), (LIU et al., 2014a), (ABDELWAHAB et al., 2016). Just as network virtualization makes multiple virtual

networks independent of the underlying physical network, radio virtualization targets to enable multiple isolated Virtual Radios (VRs) that are completely decoupled from the underlying physical antenna and can safely run on top of it. However, the state-of-the-art in radio virtualization is not able to provide this level of abstraction due to its design being focused in the previous mobile network architecture, *i.e.*, virtualizing an entire BS instead of the RRH. In other words, these works focused on adding a hypervisor to abstract the closed and technology-dependent BS hardware (SACHS; BAUCKE, 2008; ZAKI et al., 2010; KOKKU et al., 2012). Although they represented considerable advancements in radio virtualization, they can be considered archaic for 5G due to the migration from hardware-based to software-based implementations.

The second drawback, moving the entire BBU to a centralized data center, limits the maximum coverage that the network can provide to a radius of 40 Km of its location. This occurs because of the maximum latency allowed by some communication mechanisms in wireless communications, *e.g.*, the Hybrid Automatic Repeat-reQuest (HARQ) used in Long Term Evolution-Advanced (LTE-A) requires 3 ms round-trip latency (WUBBEN et al., 2014a). This latency includes the time to transport the raw signal from the RRH to the processing data center, the baseband processing at the data center, and the round-time trip back to the RRH. To mitigate the latency, recent architectural advancements consider moving some computational capabilities to RRHs located further away from the data center, as a type of "co-located small-capability data center (ZAIDI; FRIDERIKOS; IMRAN, 2015; REDANA et al., 2016), which could then execute the softwarized BBUs. Arguably, moving the entire BBU back to the co-located data center is a gimmicky solution, as it is a step back from the centralization envisioned for 5G.

The third drawback, the continuous exchange of raw baseband samples between RRHs and the central pool, limits the adoption of high bandwidth optical links in the fronthaul network. Transferring raw signal samples to the data center can require a constant fronthaul throughput of up to 5 Gbps for each RRH providing 100.8 Mbps (maximum capacity) of data rate for end-users. Obviously, the fronthaul network needs to be optical to cope with the high throughput and low-latency requirements. For instance, deploying one meter of optical fiber implies costs of up to $100 (in urban environments). In contrast, a microwave link with a range of a few tens of kilometers may be on the order of a few thousand dollars (BARTELT et al., 2015). The main effort to reduce the overall fronthaul requirements is to move only the baseband processing to the RRH, while the higher functions are performed in the centralized pool (CHECKO et al., 2015) (HUANG; CHIANG; LIAO, 2017). This partial centralization only needs to carry demodulated data, which is only a fraction of the original raw digital signal, but at the cost of sacrificing all cooperative signal processing achieved in the fully centralized architecture.

In this thesis, we introduce AIRTIME. AIRTIME is a prototype system that provides end-to-end virtual RAN slices while ensuring isolation and enabling flexible and adaptive provisioning of RAN resources to slices based on their requirements. Each slice in AIRTIME is a Virtual RAN (vRAN) instance running on top of shared physical network infrastructure, with customizable data and control planes based on the service requirements. Our system combines

the application of slicing in two levels: (*i*) the `RRH Hypervisor`, named HYpervisor for software-Defined RAdio (HyDRA), and (*ii*) the `BBU Hypervisor`. The first is a radio virtualization layer that enables multiple heterogeneous access technologies to coexist on top of the same RRH. It uses innovative baseband processing techniques to slice and abstract a RRH into multiple vRRH. The former is responsible for slicing the computing resources in isolated containers, enabling multiple Virtual BaseBand Units (vBBUs) to coexist on top of shared processing resources and realizing what we call fine-grained vBBU, *i.e.*, a flexible and adaptable solution in which the BBU functionalities are moved to multiple baseband processing containers (AKYILDIZ; WANG; LIN, 2015). Moreover, the design of AIRTIME allows vRANs to flexibly employ the RAT (a combination of vBBU and vRRH) that best fits any service requirement. AIRTIME achieve three core properties of 5G: (*i*) programmability, *i.e.*, vRANs can be reprogrammed on-demand to the RAT that best fits the service demands, (*ii*) flexibility, *i.e.*, resources allocated for vRANs can be changed on-demand to satisfy data center requirements, and (*iii*) scalability, *i.e.*, the fine-grained vBBUs enables any fronthaul network by moving specific radio functions closer to the vRRH according to the centralization gain desired.

The reasoning behind our proposal can be illustrated with an example: a RRH is just an antenna that converts radio signals from digital to analog domains in the downlink and from analog to digital in the uplink, while the BBU is responsible for implementing the access technology. The `RRH Hypervisor` is a virtualization layer between the RRH and the BBU that creates multiple vRRHs. It multiplexes the signal of vRRHs into a single combined signal that is transmitted by the underlying RRH. The software, as the uplink of a LTE-A BBU, encompasses operations, such as Fast Fourier Transform (FFT), a channel estimator, and a Resource Element (RE) de-mapper, which are sequentially applied to transform the received analog signal into user data. The `BBU Hypervisor` is responsible for executing each baseband processing function as a Virtual Network Function (VNF); the set of VNF are then aggregated to compose a LTE-A vBBU. This approach can reduce the fronthaul bandwidth and processing delay by moving part of baseband functions closer to the antenna; the centralization gains can be selected dynamically by moving VNFs closer to the vRRH. As current mobile network architectures focus on the centralization of processing resources, it is required a solution able to solve the challenges of a single "one-size-fits-all" access technology, limited coverage area, and high fronthaul bandwidth requirements, leading to the following fundamental question.

1.1 Fundamental Question, Hypothesis & Research Questions

Fundamental Question: **How to design a flexible solution that enables multi-RAT capabilities, high coverage area, and flexible fronthaul support?**

To overcome the limitations exposed in future mobile networks, especially the lack of multi-RAT capability, low coverage area, and high fronthaul requirements that demand optical links, and to answer the fundamental question, this thesis presents the following hypothesis:

Hypothesis: **a solution incorporating the concepts of radio virtualization and fine-grained baseband processing virtualization push the boundaries of future mobile networks with increased programmability, flexibility, and scalability.**

In order to guide the investigations conducted in this thesis, the following research questions (RQ) associated with the hypothesis are defined and presented.

RQ I. *Do vRRHs maintain the same transmission quality of its physical counterpart?*

RQ II. *Do fine-grained vBBUs maintain the performance required to cope with the latency requirements of 5G?*

RQ III. *What are the trade-offs by adopting fine-grained vBBUs when compared to atomic BBUs?*

The methodology employed to show the feasibility of the proposed solution is based on the development of a prototype of AIRTIME and experimental evaluation of its performance in a 5G-like use case. HyDRA is used to slice a RRH into vRRHs to provide connectivity services to constant-high-bandwidth mobile subscribers and burst-low-bandwidth sensors using two different access technologies: LTE-A and Narrow-Band IoT (NB-IoT). `BBU Hypervisor` is used to split the baseband processing of both technologies into a fine-grained vBBUs, that can have its functionality migrated between an Centralized Unit (CU) and Distributed Unit (DU) data centers. The DU data center provides small processing resources and located close to the physical RRH site, whereas the CU has high processing resources but located further.

1.2 Main Contributions

Throughout the development of this study, many contributions are expected, both in terms of conceptual advancements in the state-of-the-art of wireless virtualization in the context of 5G and in delivering tools for overcoming technological challenges. Some of these contributions are listed as follows:

- Rethinking design principles of radio virtualization to accommodate a wider range of access technologies;
- Adding more flexibility in radio virtualization to make it a more suitable solution for 5G networks provide multiple radios access technologies;
- Rethinking BBU functions using the virtualization paradigm to facilitate its adoption in current data center clouds.
- Enabling the dynamic distribution of baseband processing functions, which allows mobile networks to select the distribution option that best suits fronthaul and data center capabilities;

- Designing, prototyping, and evaluating a solution that integrates all previous concepts.

1.3 Thesis Roadmap

We give an overview of the remainder of this thesis with the objective of (*i*) guiding the reader over the process that answers the research questions and (*ii*) showing if our hypothesis holds. The best way to read this thesis is by following the sequence of chapters. However, readers familiarized with virtualization in wireless networks and the state-of-the-art in this field of research can skip Chapter 2 and 3, respectively, without losing context.

In Chapter 2, the essential background concepts and studies related to this thesis are reviewed. Initially, we give a brief overview of radio and baseband processing, detailing the main operations in a typical BBU and the points in which they can be split. In the sequence, we focus on the evolution of virtualization in wireless networks focusing mainly on the RAN. Afterward, discussions are presented about the requirements for RAN virtualization, and the different layers in which the RAN infrastructure can be virtualized.

In Chapter 3, the related research efforts in wireless virtualization are presented. We start this chapter presenting the taxonomy used to classify the most important published research that is strongly related to this thesis, *i.e.*, radio virtualization, and baseband processing virtualization. Afterward, we give a brief overview of all these research efforts based on top of which radio access technology they were designed.

In Chapter 4, this thesis contribution is presented. This section is organized to present a detailed view of AIRTIME, with focus on its two major contributions: the `BBU Hypervisor` and the `RRH Hypervisor`. We also present other two relevant components to our proposal that share functionalities with similar future mobile network architectures: the `vRAN Controller` and the `Cross-Layer Controller`

In Chapter 5 we present a prototype of AIRTIME. We dedicate a section to present the internal architecture of HyDRA, its algorithm to perform the slicing of a *physical* RF front-end into multiple virtual ones, the multiplexing process, and the configuration Application Programming Interface (API). We also dedicate a section to show the details of the `BBU Hypervisor`, which is build using standard open-source software for virtualization in data centers. Afterward, we present AIRTIME and its main components: *Access Network*, *CU data center*, *Fronthaul Network*, *DU data center*, *Virtualization Layer*, and *Virtual RANs*.

In Chapter 6, we show the evaluation of our prototype in a 5G-like network. The use case aims to show the deployment of vRANs to provide connectivity services to MBBC and mMTC Service Providers (SPs). We discuss the results obtained for the `BBU Hypervisor` and `RRH Hypervisor`.

In Chapter 7, we show additional evaluations for AIRTIME. Here we present analytical and simulated results for the fronthaul bandwidth requirements and latency, as well as CPU usage for different baseband processing VNFs distribution options.

In Chapter 8 some final remarks and conclusions are presented. Also, answers to all research questions are discussed and justified.

2 BACKGROUND

Although virtualization in the wired domain has received significant attention in the past years, wireless virtualization is still in its infant stage. The primary focus of wired network virtualization has been on designing an adaptive network substrate that can support multiple virtual networks running customized services. While solutions from the wired network and server domains can be used to virtualize the mobile core network, BS virtualization has to deal with problems specific to the radio transceiver architecture, *i.e.*, closed black-boxes that offers technology-specific configurations, and varying channel conditions that make it harder to virtualize the wireless resources across multiple entities. In this chapter, we dwell on the concepts to better understand the work presented in this thesis.

The remainder of this chapter is organized as follows. In Section 2.1, we detail the architecture of a radio transceiver and show how its evolution from hardware-based black-boxes to software-defined baseband processing impacts in the architecture of mobile networks. Section 2.2 presents the central concepts in RAN virtualization, the requirements that it must satisfy, and the different logical levels at which a wireless virtualization layer, *i.e.*, hypervisor, can operate on a radio transceiver. We close this section presenting some performance metrics necessary to evaluate the performance and quality of a virtualized wireless network. Finally, Section 2.3 reviews the main points presented in this chapter.

2.1 Architecture and Evolution of a Radio Transceiver Device

A radio transceiver can be split into two major components: the Radio Frequency (RF) front-end and the baseband processing. Modern radios perform the baseband processing in the digital domain, *i.e.*, with digital signal samples. At the same time, the RF front-end mainly contains analog radio circuitry. Thus, Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC) conversion form the natural interface between the baseband processing and the RF front-end, as shown in Figure 2.1.

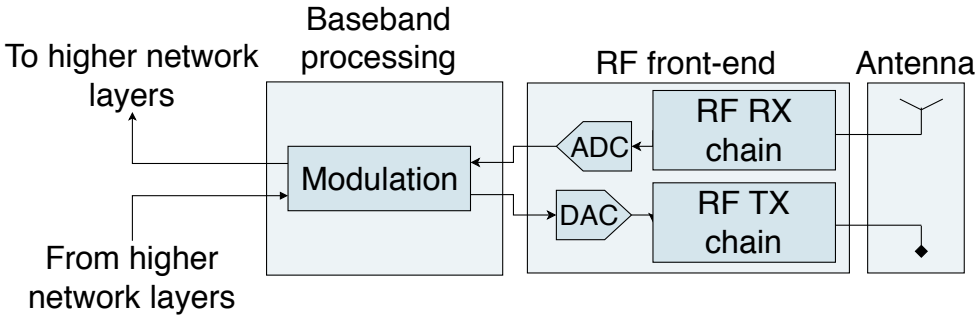


Figure 2.1 – Architecture of a radio transceiver

As shown in Figure 2.2, the baseband processing translates information bits into digital baseband waveforms, and vice versa. This function is also referred to as digital baseband modulation, or simply, digital modulation. Digital modulation maps a binary sequence to a segment of digital waveform samples, called symbols. At the receiver side, these symbols are demodulated to retrieve the embedded binary information. The baseband signals are not suitable to transmit directly. Thus, the RF front-end will convert the digital baseband samples into high-frequency analog radio signals. When receiving, the RF front-end selects the desired radio frequency signals, down-converts, and digitizes them to digital baseband samples. In current radio systems, the radio frequency is divided into fixed-size channels that equal to the baseband width at a pre-determined central frequency. The baseband signal may be mapped to one of pre-defined channels, but it cannot change the bandwidth or pick up an arbitrary central frequency.

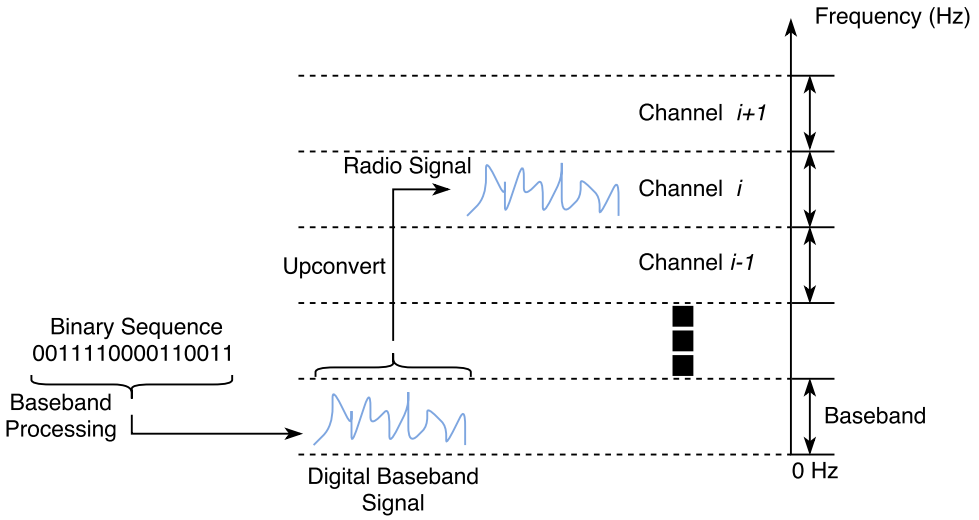


Figure 2.2 – Transformation from digital data to analog signal

In the traditional mobile network architecture, RF front-end and baseband processing functionalities are integrated inside a BS. The antenna module is located in the proximity (few meters) of the radio module, as the coaxial cables used to connect them exhibit high losses. This architecture was popular in 1G and 2G mobile networks. In recent years, the RF front-end was decoupled from the baseband processing; this enabled mobile operators to replace the baseband processing module when needed without too much of an effort. Arguably, the main architectural element of the RAN that offers opportunities for virtualization is the radio transceiver. And within the radio transceiver, virtualization can be performed in the baseband processing module, the RF front-end, or a combination thereof (we explore the virtualization of these later in this section).

2.1.1 Towards a Centralized Baseband Architecture

The high-bandwidth data exchange between the digital domain of the baseband processing and the analog domain of the RF front-end and antenna requires a high-bandwidth bus connecting these two domains. Initially, this requirement constrained engineers to design BSs within a single piece of hardware with a high-performance bus connecting the digital and analog domains. This constraint was removed with the advancements of fiber options, *i.e.*, a fiber cable could be used to connect the digital and analog domains, with the added benefit that both elements can be located further away. In the 3G and 4G era, the baseband processing was implemented on BBUs, specialized and dedicated hardware that implements a RAT, while a RRH integrates the RF front-end and the antenna. Figure 2.3(a) and 2.3(b) show the evolution from the 2G to the 3G/4G network architecture. In the downlink, *i.e.*, from BBU to RRH, the "data" generated by the BBU is a stream of IQ samples that represent the radio signal that must be transmitted by the RRH. Similarly, in the uplink, *i.e.*, from RRH to BBU, the "data" received by the BBU is the digitized version of the signal received in the RRH.

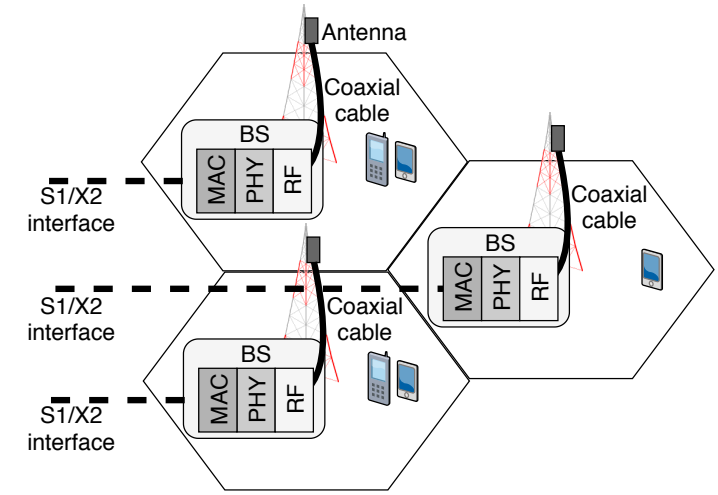
In current mobile networks, the RF front-end and the baseband processing have a much more apparent separation, as shown in Figure 2.3(b). The radio unit is called a RRH and provides the interface to the fiber and performs DAC and ADC, power amplification, and filtering. The baseband processing, now performed in the BBU, is isolated and independent from the radio unit. This architecture is considered to be "decentralized", as each BBU performs its operations independently of the others.

Recent advancements leverage the bandwidth of fiber-optic cables to incorporate cloud-based architectures into the mobile network. In this network architecture, the BBUs can be placed in a data center (thus the name "centralized baseband architecture"), enabling cost savings through centralized maintenance, cloud facility rental, and reduced environmental impact. A fronthaul network connects the data centers with RRHs, which can be geographically separated by up to 40 Km (the distance limitation comes from the delay constraints imposed by the mobile network protocol stack) (MAROTTA et al., 2018). The benefit of baseband centralization is multifold:

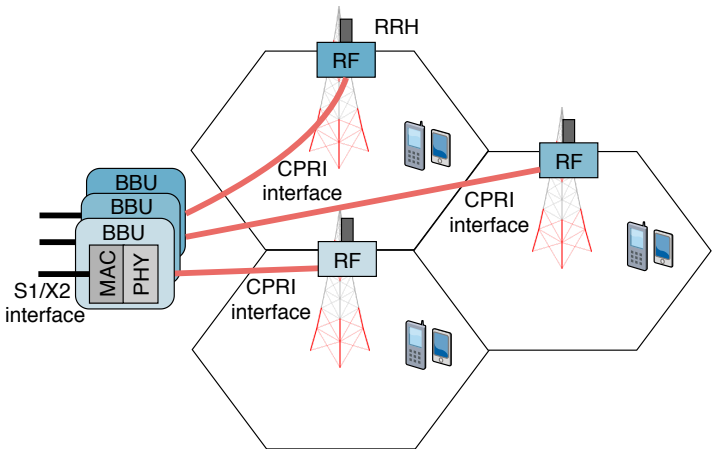
- The barriers against information exchange between BSs are primarily eliminated through centralization, enabling cooperation technologies, such as multi-user detection, and Large-Scale Cooperative Multiple Antenna Processing (LS-CMA).

- Computational resources can be flexibly provisioned from the pool to support regular or advanced wireless access-technologies (ZHAI et al., 2014).

- The utilization ratio of computational resources can be significantly improved through the statistical multiplexing of computational tasks (LIU et al., 2014b).



(a) Traditional Base Station



(b) Base Station with RRH and BBU

Figure 2.3 – Base Station architecture evolution

- The centralized computational resources can be virtualized and used to support software-defined BSs, greatly simplifying the development and maintenance of mobile networks.

Thus, by pooling BBUs in data centers, centralization gains are achieved. However, BBUs are hardware-based platforms utilizing specialized digital signal processors. As a long-term goal, it is beneficial to replace the hardware-based BBUs for software running on general-purpose hardware, *i.e.*, vBBU. It is worth noticing that replacing the hardware-based BBU for vBBUs running on top of standard data center hardware is currently considered as an important use case for Network Function Virtualization (NFV) (ROST et al., 2015; ADAMUZ-HINOJOSA et al., 2019).

Only fiber links are capable of supporting the necessary data rates between the RRHs and the data center. This constraint constitutes the main drawback of centralized baseband architectures, *i.e.*, it requires very high data rate links to the central data center. Wubben *et al.* (WUBBEN et al., 2014b) report a required fronthaul transmission rate of 10 Gbit/s for time-domain LTE-A

with eight receiver antennas and 20 MHz bandwidth. Centralized baseband is less flexible due to the use of optical fibers, as only spots with existing fiber access or with high revenue for fiber access may be chosen. It is expected that future 5G networks will deploy heterogeneous fronthaul solutions that are optimized for different scenarios. This mix of fronthaul characteristics will also imply a mix of centralized architecture that requires high-capacity links and decentralized solutions compatible with fronthaul solutions that introduce high latency and stronger throughput constraints. Recent architectural advancements explored centralized architectures while relaxing the requirements on the fronthaul network. The architectural advancement is the split of the BBU into two parts, one executed locally at the RRH, and one executed at a central processing unit. Depending on the chosen split, the fronthaul requirements are reduced, and a different degree of centralization gain is achieved. In the next subsection, we explore these split options.

2.1.2 Splitting the functionality between BBU and RRH

In this subsection, we introduce several functional split options of the baseband processing functionalities. These split options determine which operations are executed in the RRH and which ones are executed in the data center, directly influencing the required fronthaul data rate. We focus our discussion on the uplink of a LTE-A BBU since it is (*i*) one of the main access technologies for 5G, (*ii*) the most processing-intensive, and (*iii*) the uplink processing is significantly higher than the downlink processing (DOTSCH et al., 2013). Figure 2.4 gives an overview of the LTE-A baseband processing chain of the uplink and the possible functional split options.

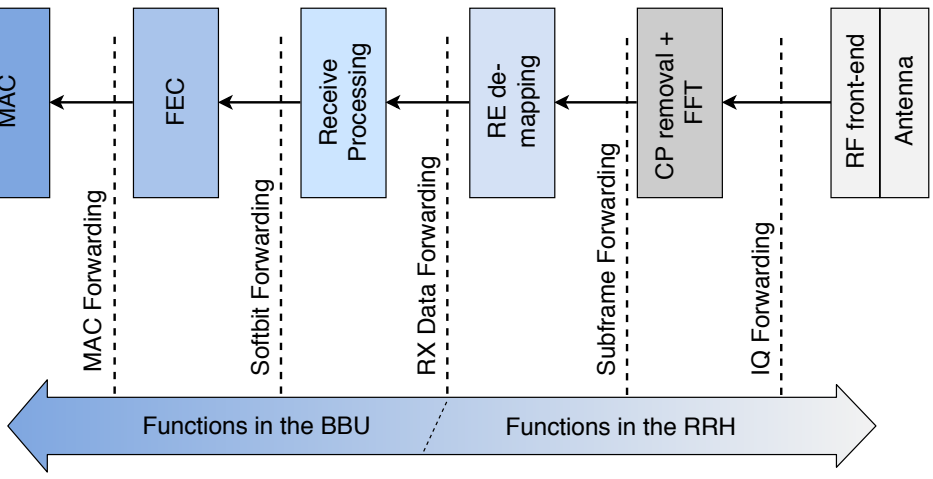


Figure 2.4 – Baseband processing splits in the uplink of a LTE-A BBU

2.1.2.1 *IQ Forwarding – Split A*

In IQ Forwarding, samples are transported over the fronthaul to the data center, which centralizes all the baseband processing. This approach is usually referred to as Radio-Over-Fiber (RoF) and is used in the Common Public Radio Interface (CPRI) standard. The main benefit of this split is that almost no digital processing is required at the RRH, potentially making them very small and cheap. Moreover, this split eases the adoption of LS-CMA because of the centralization of all In-phase & Quadrature (IQ) samples (PENG et al., 2014). This option is interesting only in the cases where the RRH is not connected to an edge data center or if the cost of fronthaul transport is low.

The fronthaul data rate required in this functional split is fixed; thus, mobile operators can determine beforehand whether it can handle the traffic of a RRH. This split does not allow for too much flexibility when considering load balancing: either the RRH is turned on forwarding its data rate through the fronthaul, or it is turned off and providing no signal coverage in its area.

2.1.2.2 *Subframe Forwarding – Split B*

In Subframe Forwarding, the Ciclic Prefix (CP) Removal and FFT are performed at the RRH, while the remaining functionalities are executed in the data center (THYAGATURU; ALHARBI; REISSLEIN, 2018). In this case, only the IQ samples of useful subcarriers are transported over the fronthaul, which in LTE-A represents roughly 60% of the total subcarriers. As the FFT can be implemented on dedicated hardware very efficiently, incorporating this function in the RRH is worthwhile when compared to pure IQ Forwarding.

Subframe forwarding becomes even more attractive close to 100% of the radio resources are being utilized because the fronthaul data rate required is always the same, while also enabling LS-CMA mechanisms. Also, in this functional split the baseband processing workload at the data center does not depend on the actual load of the RRH.

2.1.2.3 *RX Data Forwarding – Split C*

In RX Data Forwarding, the RE de-mapper is moved to the RRH site. In this split option, the data center receives the IQ samples of REs allocated to mobile subscribers, *i.e.*, 10% of 720 Mbps if 10% of REs are allocated (which is something that can change between every LTE-A frame). To allow for the joint processing of signals from multiple RRHs, it has to be ensured that only REs of mobile subscribers not considered for joint processing are removed, even if they are not (primarily) associated with the current RRH. This split option is particularly attractive when the RRH is under low usage.

2.1.2.4 *Softbit Forwarding – Split D*

In Softbit Forwarding, the RRH executes the Receiver (RX) processing, which consists of combining the signal of multiple antennas to perform channel equalization. Thus, moving this function to the RRH removes the dependency on the number of receiver antennas. LS-CMA is not possible at the central data center, but higher layer cooperations are still possible, such as joint bit decoding.

2.1.2.5 *MAC Forwarding – Split E*

MAC Forwarding is the one adopted by current mobile networks. The output of the RRH in this split option is pure Medium Access Control (MAC) payload. The resulting fronthaul rate depends mostly on the Modulation and Coding Scheme (MCS) used to communicate with mobile subscribers, *i.e.*, higher modulation orders will result in more bits at the MAC payload. Performing all the radio functions in the RRH terminates the possibility for joint physical layer processing in the data center, and only cooperation on higher layers, *e.g.*, joint scheduling, remains possible. As physical layer cooperation mainly revolves around interference mitigation, this option is beneficial in scenarios were RRHs are well separated, *e.g.*, for indoor deployments, or in narrow street canyons.

The move towards centralization and the utilization of general-purpose hardware enables the adoption of software implementing all the BBU functionalities running on top of high-performance General Purpose Processors (GPPs). Through software upgrades or re-configuration, programmable BBUs can provide any RAT (although constraints in the fronthaul bandwidth and latency may hinder the adoption of some). A natural evolution from the centralized RAN architecture is towards RAN virtualization. This evolution leverages the cloud data centers and BS virtualization (a combination of a virtual BBU and a virtual RRH (CHECKO et al., 2015)) to dynamically create vRANs on top of the physical infrastructure.

2.1.3 Towards RAN virtualization

RAN virtualization brings the ability to dynamically create, deploy, and manage vRANs, each one tailored to meet the requirements of a particular end-to-end service. Furthermore, by offering the capabilities to support multiple vRANs, RAN virtualization opens up new business models in which SPs can lease vRANs from the infrastructure providers (TALEB et al., 2015). In this scenario, the infrastructure provider controls all physical resources, comprising the radio spectrum, physical RRHs, hardware resources in data centers (*i.e.*, servers with processing, memory, and storage), and the fronthaul network. A SP enters into a contract with the infrastructure provider for one or more vRANs, which include at least one virtual BS, *i.e.*, a vRRH attached to a vBBU. Note that RAN virtualization is different from the legacy RAN sharing

model, in which the focus is only on the sharing of resources among mobile operators, *e.g.*, radio spectrum, network functions, and baseband processing.

Isolation, coexistence, programmability, and adaptability are the key elements for enabling end-to-end vRAN customization to accommodate the different services envisioned for 5G mobile networks. These three elements constitute the foundation for a multi-service and multi-tenant future mobile networks (FOUKAS et al., 2017), and it is realized by applying the principles of BS virtualization.

BS virtualization is the process of abstracting one physical BS and slicing it into virtual BSs holding certain corresponding functionalities and isolated from each other (LIANG; YU, 2015). In other words, it is the process of abstracting, slicing, isolating, and sharing a BS between multiple virtual BSs that hold all (or parts of) the functionalities of their physical counterpart. Slicing and virtualization can be implemented in a BBU to allow multiple vBBUs to run on top of the same physical hardware, or in a RRH to enable multiple vRRHs to run on top of physical RRHs. In this thesis, we argue for a combination of BBU and RRH slicing to instantiate complete end-to-end virtual RANs on top of the physical infrastructure.

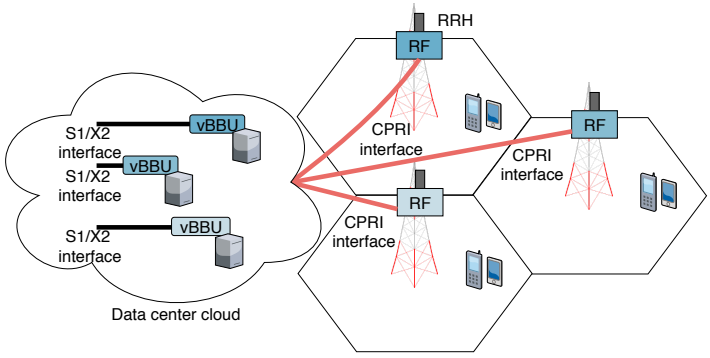
2.1.3.1 Virtual BBUs

A vBBU implements the signal processing operations related to transforming a sequence of bits, *e.g.*, carrying user data, into IQ samples that represent the radio signal, which must be transmitted (exactly like the physical hardware-based version). A vBBU pool can be executed on a GPPs, leveraging highly-optimized signal processing libraries, as well as taking advantage of the ever-increasing evolution of processors, such as higher processing power and energy efficiency. Figure 2.5(a) illustrates this architectural evolution.

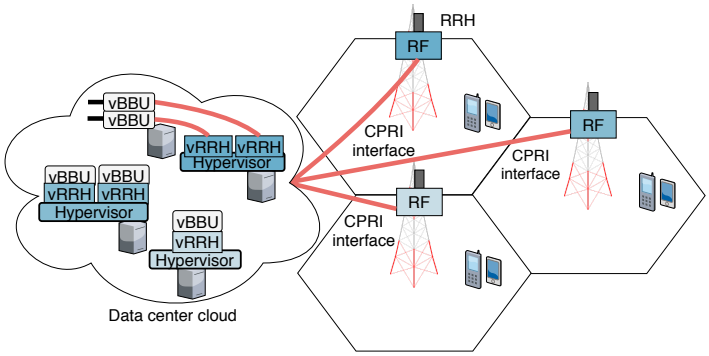
The first research efforts in BBU virtualization considered a static functional split between the vBBU and the RRH, in which the functions in the RRH are still implemented on specialized hardware, while the remaining functions are moved to the vBBU (KIST et al., 2019). Recently, the 3GPP RAN3 working group has considered a vBBU split consisting of two new entities, named DU and CU. The former can host time-critical functions of the physical layer, whereas the latter hosts non-time-critical features, such as the upper MAC layer. It is estimated that the DU can cover an area of up to 20 Km radius, while CU covers areas up to 200 Km (3GPP, 2017).

2.1.3.2 Virtual RRHs

Following our definition of BBU virtualization, we define RRH virtualization as a subset of BS virtualization in which a physical RRH is abstracted into vRRHs holding certain corresponding functionalities of their physical counterpart (KIST et al., 2017; SANTOS et al., 2019). As mentioned previously, a physical RRH is responsible for translating the stream of IQ sam-



(a) Base Station with RRH and vBBU.



(b) Base Station with vRRH and vBBU.

Figure 2.5 – Base Station architecture in RANaaS

ples generated by BBUs into over-the-air radio signals and converting over-the-air radio signals received by the antenna into a stream of IQ samples that are sent to the vBBU. In summary, a vRRH must operate exactly like its physical counterpart and a BBU should not be able to distinguish between a physical or virtual RRH.

In contrast to BBU virtualization, which can employ conventional cloud computing technologies, the virtualization of RRH is still in the early stages of development. Applying virtual machines or container-based solutions in this domain does not adequately address the problem as they do not deal with the additional dimension of virtualizing and isolating radio resources (spectrum and radio hardware). The existing RRH virtualization approaches that account for the slicing of radio resources fall into one of two categories:

- Slice and assign high-level radio resources between vRRHs instances by employing a common physical and lower MAC layers (FOUKAS et al., 2016; FOUKAS; MARINA; KONTOVASILIS, 2017).

- Slice and assign chunks of spectrum between vRRHs, which then interact with a (v)BBU (KIST et al., 2017).

The state-of-the-art in RRH virtualization uses a radio hypervisor for the slicing of low-level radio resources and for provisioning a dedicated chunk of radio resources for each vRRH,

which then interfaces with a (v)BBU, as shown in Figure 2.5(b). The next section presents the central concepts of virtualization in wireless networks.

2.2 RAN Virtualization: Requirements and Challenges

RAN virtualization can be performed at different levels, based on the type of resource that is being sliced: spectrum virtualization, access-technology virtualization, or link virtualization. Similar to wired network virtualization, in which the physical device is sliced and abstracted into multiple virtual counterparts, RAN virtualization needs physical wireless devices and radio resources to be sliced and abstracted into a number of virtual counterparts. In other words, virtualization, in both wired and wireless networks, can be considered as the process of splitting the entire network system (WANG; KRISHNAMURTHY; TIPPER, 2013), (LIANG; YU, 2015). However, the distinctive properties of the wireless environment, *e.g.*, attenuation, mobility, and broadcast, make the process of slicing and abstraction more complicated. Furthermore, RANs can provide connectivity services to a much wider range of access technologies than wired networks, which make the abstraction challenging to achieve.

In this thesis, we consider RAN virtualization as the technologies in which *physical wireless resources* and *physical radio resources* can be sliced and abstracted into virtual resources holding a subset of functionalities, and shared by multiple slices throughout isolation each other. Thus, virtualizing a RAN is to realize the process of slicing, isolating, and sharing the RAN infrastructure, *i.e.*, radio spectrum, RRHs, BBUs, and wireless links. Four essential requirements need to be met to implement RAN virtualization. We define them as follows:

Isolation: Isolation ensures that any configuration, customization, topology change, misconfiguration, and departure of any specific virtual resource will be not able to affect and interfere with other virtual resources. In other words, isolation means that any change in vRAN, such as the number of end-users, mobility of end-users, and fluctuating channel status, should not cause any change in resources allocated to other vRANs (KOKKU et al., 2012). Indeed, vRANs are transparent to each other, or we can say that they never know the existence of other vRANs. Since many virtual counterparts should coexist, isolation is the fundamental issue in RAN virtualization (and in virtualization in general) that guarantees fault tolerance, security, and privacy (CHOWDHURY; BOUTABA, 2009). Also, in wireless networks (and in especially the mobile networks) any change in one cell may introduce high interference to neighboring cells, and the mobility of end-users may create instability in a specific area (GESBERT et al., 2010). Therefore, isolation becomes more difficult and complicated in wireless networks compared to the wired counterpart, and even more complicated in mobile networks.

Coexistence: In wireless virtualization, the physical hardware should allow multiple independent virtual resources coexisting on the substrate physical network (BELBEKKOUCHE;

HASAN; KARMOUCH, 2012). In fact, the purpose of virtualization is to make multiple systems to run on the same physical network. Slices are different among themselves because they are created to satisfy the different requirements of each party. Thus, wireless virtualization needs to bear multiple slices who hold various Quality-of-Service (QoS) requirements, topology, services type, security level, user behavior, among others.

Flexibility: Flexibility in different layers of the network needs to be provided in RAN virtualization through the decoupling of control protocols from the underlying physical network and other coexisting virtual networks (CHOWDHURY; BOUTABA, 2009). However, flexibility depends on the level of virtualization, which in RAN networks can happen at spectrum level, protocol (or flow) level, or link-level (we will detail these levels Section 2.2.1). High-level virtualization may reduce the flexibility while better multiplexing resources across slices (and hence increased utilization with fluctuating traffic) and simplicity of implementation, but can reduce the efficacy of isolation and the flexibility of resource customization. In contrast, virtualization at a lower level leads to the reverse effects.

Programmability: Since vRANs are assigned to slice owners and the management of these elements are decoupled from the substrate networks, RAN virtualization needs to provide complete end-to-end control of the virtual resource to the slice owners. Slice owners can manage configuration, allocation of virtual networks, *e.g.*, routing table, virtual resource scheduling, admission, and even modifying access-technologies. Programmability needs to be integrated into RAN virtualization to help slice owners implement customized services, schedulers, RATs, among others. Thus it needs the virtualization layer to provide appropriate interfaces, programming language, and enabling a secure programming paradigm with a considerable level of flexibility (CHOWDHURY; BOUTABA, 2009).

In RAN virtualization, a resource can refer to the wireless equipment such as the entire BS, as well as low-level physical resources such as space-time-frequency slots. Ensuring the four requisites described in this section can be something simple or incredibly complex, depending on what resource is being abstracted. The abstraction of low-level resources provides the illusion that high-level resources (such as entire BSs) are being virtualized, although low-level resources must ultimately be partitioned to support that sharing. In the following subsection, we go further in the RAN virtualization by describing the main levels at which it can be performed.

2.2.1 Virtualization Depth

The depth of virtualization is the extent of penetration of slicing and partitioning on the wireless resource. It is tied to the granularity of virtualized resources and often dictates where the hypervisor is located inside the virtualization architecture. For example, in the access-technology-based virtualization (refer to Section 2.2.1.2), the hypervisor can be a network filter

sitting on top of the wireless networking stack. However, this is a "shallow"' virtualization and does not allow multiple access-technologies to share the same hardware. With "deeper" virtualization, the full RAT can eventually be sliced to enable each slice to adopt a customized RAT. In the following subsections, we present the three primary levels in which RAN virtualization can be performed.

2.2.1.1 Spectrum Virtualization

Spectrum level virtualization consists of abstracting the radio spectrum through frequency, time, space, and code dimensions, or a combination of those. It is crucial to differentiate spectrum virtualization from the standard techniques utilized to enable multiple wireless devices to access the wireless medium. More precisely, all wireless technologies use some sort of access technique to enable multiple wireless devices to perform transmissions in the wireless medium.

For example, 3G mobile systems used Frequency Division Duplexing (FDD), a frequency slicing method to enable all mobile subscribers to communicate with the BS, while at the same time using Time-Division Duplexing (TDD), a time-slicing method, to enable both downlink and uplink communications. In FDD, each mobile subscriber receives a small bandwidth channel to communicate with the BS for the duration of a call, whereas the TDD gives small portions of time during which all nodes perform the downlink or uplink communications. In this case, the mobile subscriber hardware is designed to operate within the frequency range and bandwidth of the channels that can be assigned to it. Figure 2.6 illustrates a scenario where three users are performing FDD to communicate with a BS, and TDD to receive data (downlink) and transmit data (uplink).

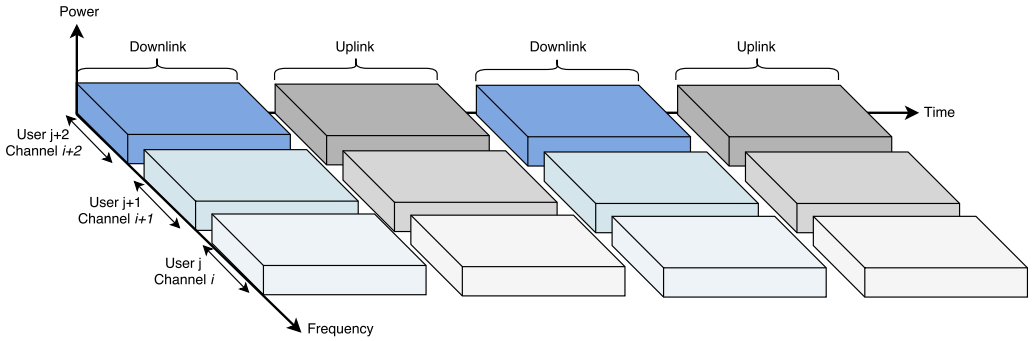


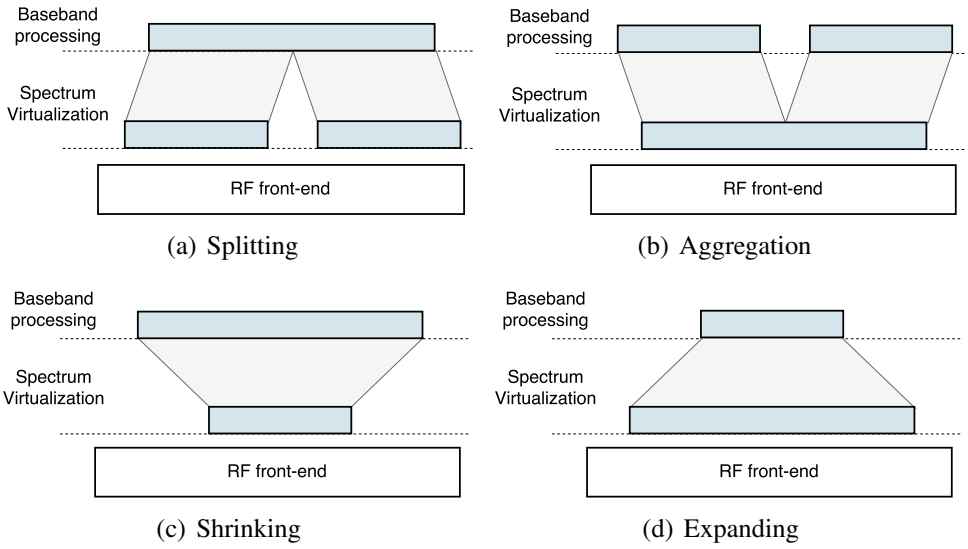
Figure 2.6 – Spectrum in the frequency division multiplexing technique

Spectrum level virtualization was first proposed by Paul and Seshan (PAUL; SESHAN, 2006) in a technical report for the GENI project. They conceptually discuss several methods to achieve spectrum abstraction, all of them based on the multiplexing of time, frequency, and space. In their case, spectrum virtualization would be used as a means to enable multiple concurrent experiments to run on top of the GENI project testbed using the same set of radio devices.

More precisely, the spectrum would be sliced, and each slice assigned to a given experiment. Thus, multiple experiments could run simultaneously on top of the same physical radio.

In this method, each slice owner will receive a time-slot to perform its transmission. Ensuring isolation at this level is a challenging task, as the virtualization layer needs to ensure that all devices are using the same slice simultaneously, which may require complex synchronization mechanisms across spectrum virtualization layers in different devices. Moreover, slice owners should be unaware of the time-slot synchronization mechanism and must see the spectrum as a continuous signal stream.

A spectrum virtualization layer adopting frequency slicing will assign a virtual spectrum band to each slice. Ideally, the mapping from real to virtual spectrum bands is the responsibility of the virtualization layer and should not be known by any of the slices. This opens up opportunities to perform four transformations from the real to the virtual spectrum, namely: splitting, aggregation, shrinking, and expanding. Figure 2.7 illustrates these transformations.



(a) Splitting (b) Aggregation

(c) Shrinking (d) Expanding

Figure 2.7 – Spectrum transformations possible within spectrum virtualization

Splitting: In this transformation, shown in Figure 2.7(a), a continuous virtual spectrum is mapped to two or more non-contiguous smaller bands of the physical spectrum. The total bandwidth of the virtual band and the sum of the smaller bands is the same. Also, theoretically, there is no limitation on the size of the gaps between each band of the physical spectrum, but in practice, it is limited by spectrum access policies and by technological limitations of the RF front-end.

Aggregation: In the aggregation transform, shown in Figure 2.7)(b), multiple non-contiguous virtual spectrum bands are mapped to a contiguous portion of the physical spectrum. Aggregation is particularly interesting to enable multiple PHYs to run on top of the same physical RF front-end. More precisely, by associating each virtual spectrum portion to a wireless PHY layer, the RF front-end can multiplex multiple RATs.

Shrinking: By doing shrinking, the spectrum virtualization layer reduces the bandwidth of the virtual spectrum when mapping to the physical spectrum by a given factor, as shown in Figure 2.7(c). Different from the previous transformations, this one can lead to loss of information or even destruction of the original signal to an irrecuperable form. To avoid such problems, shrinking can require that the baseband processing communicates with the virtualization layer so that it can specify the reduction factor.

Expanding: This transformation consists in increasing the virtual bandwidth by a given factor when doing the mapping to the physical spectrum, as illustrated in Figure 2.7(d). Different from shrinking, expanding consists in adding information to the virtual spectrum.

2.2.1.2 Protocol-level Virtualization

In this level of abstraction, the virtualization layer is placed between the MAC layer and upper layers of the networking stack (hence the name protocol virtualization), implemented as an overlay filter and software switch module over the existing radio hardware. This reduces the level of flexibility when compared to spectrum level virtualization by constraining vRANs to use the same access technology implemented in the physical device, *i.e.*, if we are considering the virtualization of a LTE-A BSs, all virtualized counterparts are LTE-A BSs as well.

This virtualization consists of adding a scheduler that allocates low-level resources required for a given slice. These resources include access control opportunities for the MAC layer and baseband processing modules for the PHY layer. Some flow virtualization functions, such as uplink flow scheduling, can also be integrated within the protocol scheduler. Because of its limitation, protocol virtualization is also known as "flow-based virtualization" in the literature. The placement of a protocol-level virtualization hypervisor is shown in Figure 2.8.

2.2.1.3 Link Virtualization

At this level, slices are interested in a link connecting two or more nodes in the network. The underlying virtualization system can use different access technologies and configurations to ensure that all nodes in the slice can communicate (WANG; KRISHNAMURTHY; TIPPER, 2013) (LIANG; YU, 2015). Thus, the focus of this level of virtualization is to abstract all the underlying aspects related to the wireless medium and provide a "line of communication between two nodes and with a specific QoS". Although appearing to be similar to resource sharing, link virtualization is fundamentally different, since resource sharing does not create independent resources. Also, virtualization allows the aggregation of links from multiple RATs into a single virtual link, which resource sharing does not.

Several complications exist in wireless link virtualization that do not exist in wired link virtualization due to the difference between the two mediums. The first difference is that it is

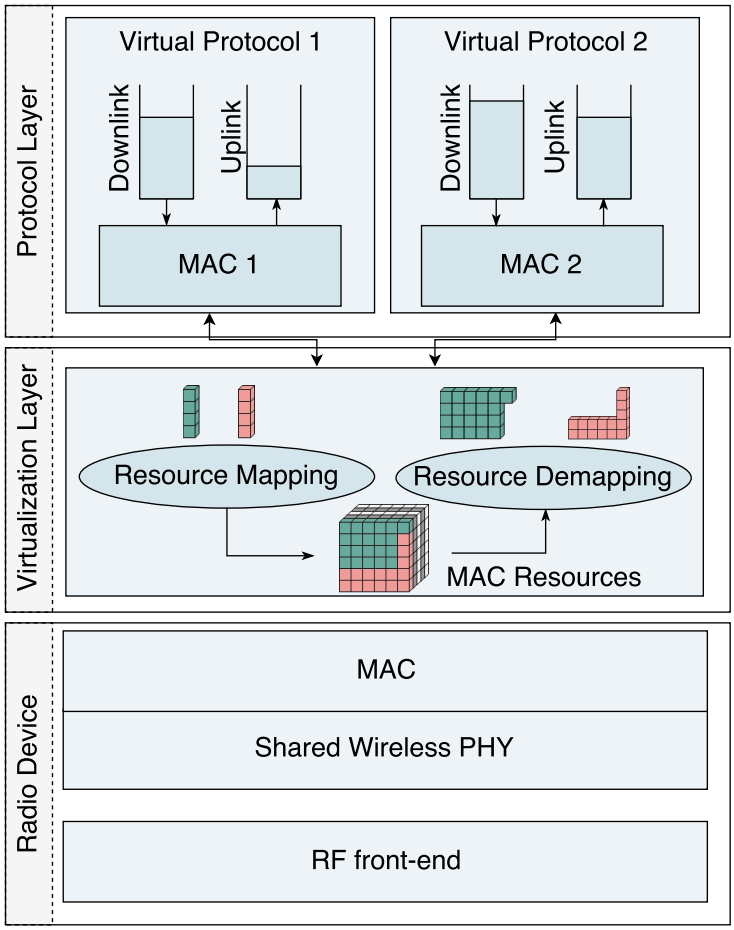


Figure 2.8 – Architecture of a hypervisor for protocol-level virtualization

not possible to predict the information throughput of wireless links in advance because of the inherent variation of the wireless channel over time. A second difference is that wireless links broadcast, and thus have the potential to interfere with any other wireless link, whereas in wired links this does not occur. Another complication for wireless links is that wireless nodes tend to be highly mobile, which means it is harder to predict and provision for the information transfer between nodes.

2.2.2 Performance Metrics of RAN virtualization

In this subsection, we present some performance metrics that are necessary to evaluate the performance and quality of a wireless hypervisor or for the vRAN as a whole. These performance metrics can be used to compare different virtualization mechanisms, architectures, resource allocation algorithms, or management systems. We classify the metrics into two categories, according to the requirements of wireless virtualization: performance metrics of traditional wireless networks and RAN-virtualization-specific metrics. These metrics are detailed in the next two subsections. Table 2.1 provides a summary of all metrics for quick reference.

Table 2.1 – Performance metrics for RAN virtualization

Types	Metrics	Units	Description
Metrics of Physical RANs	Coverage	m^3	3D area covered by a wireless technology
	Capacity	bps	Peak data rate for a certain area
	Spectrum Efficiency	$\frac{bps}{Hz}$	Capacity/bandwidth in a certain coverage area
	Energy Efficiency	$\frac{bps}{Joul}$	Capacity/energy consumption, in a certain coverage area
	Signal Latency	*seconds*	Packet delay and signaling delay
Metrics of Virtual RANs	Throughput between wireless entities	bps	Data rate achieved between virtual nodes
	Utilization and stress of physical resources	unitless	Used resources / available resources
	Delay and jitter between virtual nodes	*seconds*	Time for a packet to go from one virtual node to another virtual node
	Path length between virtual nodes	number of nodes	Number of hops in the physical network that connects two virtual nodes
	Isolation Level	unitless	Virtualized resource type

2.2.2.1 Metrics for Traditional RANs

- **Coverage and capacity:** Coverage refers to the whole geographical area where the physical RAN can cover. Capacity refers to the maximal aggregated peak rate (maximum theoretical throughput) for a particular area served by a RRH. Throughput usually refers to the data rate delivered to end-users during a specific time duration or area. Coverage, capacity, and throughput are the fundamental metrics in RAN design and optimization. In mobile networks, coverage and throughput are mainly related to channel bandwidth, transmit power, and RRH placement, which we briefly presented as follows.

  - Bandwidth: Is the total used radio spectrum in a certain area for serving end-users.
  - Transmit power: Is the power transmitted from the RRH to end-users as well as from end-users to the RRH.
  - RRH placement: Refers to the geographical distribution of RRHs to ensure that a new network or service meets the needs of end-users and operators. The deployment of small cells and relays will lead to higher throughput and broader coverage. Network sharing combining both virtualization and small cells deployment also can bring novel network planning strategies, which are different from traditional network planning.

- **Spectrum Efficiency:** Spectral Efficiency (SE) metric can be defined as the ratio between system throughput (or coverage) and bandwidth. SE has been widely accepted as an

important criterion for wireless network optimization, especially for mobile networks. To study the SE in a particular area, one can add the SE achieved by all the RRHs that use the same spectrum in it. Also, more detailed SE metrics can be used to evaluate the performance, such as the cell-edge-SE and the worst-5%-users-SE.

- **Energy Efficiency:** Energy Efficiency (EE) metric can be defined as the ratio between system throughput (or coverage) and energy consumption. It should be noted that the energy consumption is not limited to transmission energy consumption but should include the whole network energy consumption, including networking equipment and accessories, *e.g.*, air conditioners, lighting facilities, among others.

- **Signaling Latency:** Signaling latency refers to the delay of control signals among nodes that hold the responsibility of network management. Since RAN virtualization multiplies the number of nodes, *i.e.*, one physical RAN node can be sliced into multiple virtual nodes, the number signaling exchanges will be increased in the network, which can cause higher delay of signaling.

2.2.2.2 Metrics for Wireless Virtualization Solutions

In addition to the performance metrics of traditional RANs, some virtual-RAN-specific metrics can be used to measure the quality and performance of a RAN virtualization solution.

- **Throughput between wireless entities:** Different from traditional throughput, virtual-RAN-specific throughput is the average data rate achieved between virtual entities of the vRAN. This metric quantifies the connection performance between virtual nodes and is useful to evaluate the resource allocation algorithms and the management efficiency in virtualized nodes.

- **Utilization and stress of physical resources:** Since the physical resources are used to create virtual spectrum, virtual nodes, and virtual links, utilization is defined as the ratio between the used and available physical resources. For example, the utilization metric can be derived with a ratio of utilized and available bandwidth, power, time-slots, or signal processing. In addition, the stress metric quantifies the capability that the substrate physical resources have to bear a certain maximum number of virtual entities. For example, stress measures how many vBBUs can be mapped to a physical BBU. Utilization and stress can be used to evaluate resource allocation algorithms and virtualization mechanisms.

- **Delay and jitter between virtual entities:** Delay describes the amount of time needed for a packet to go from one node in the network to another node. Here, a node can be a virtual node or an end-user. The packet inter-arrival time can be measured by jitter,

which is inherent to physical networks. Jitter has greater effects on the performance of virtualized wireless nodes/resources than traditional nodes/networks by the same reason of signaling latency, *i.e.*, the increased number of nodes further increases the jitter. Delay and jitter can be used to evaluate virtualization mechanisms and management efficiency, since different mapping strategies and controller methods may greatly affect the network.

- **Path length between virtual entities:** Since some virtual nodes are connected by virtual links, which means that the direct physical link may not exist. The path length metric is the number of nodes between the physical nodes that represent the virtual ones. The path length will affect the delay and jitter due to that longer path length needs more physical nodes to forward the packets. Therefore, path length can be used to evaluate virtualization mechanisms and resource allocation algorithms.

- **Isolation level:** Isolation level can be used to measure the lowest virtualized physical resource level. For example, if the virtualization solution creates slices based on time-slots, the isolation level is time-slot.

2.3 Summary

We started this chapter by presenting the evolution of BS hardware as well as their association with the evolution of mobile networks. Initially, both the RF front-end and the BBU were coupled in a hardware box that implemented everything that the mobile access technology required. In recent years, the RF front-end was decoupled from the BBU; this enabled mobile operators to replace the BBU module when needed without too much of an effort. We showed that baseband centralization is the idea of moving all baseband processing boxes to a central data center and progressively replace the baseband processing hardware by vBBUs. Further, we present the concept of BBU functional split, which we will explore in our proposal in the next section. We showed that keeping some of the BBU functions at the RRH can significantly reduce the fronthaul bandwidth required, as well as alleviating some of the latency requirements.

We gave an overview of RAN virtualization, *i.e.*, virtualization applied to RAN and its associated devices. We presented the main requirements to realize RAN virtualization, *i.e.*, isolation, coexistence, flexibility, and programmability. We also presented the different levels at which a wireless hypervisor can be placed on a radio device to perform the slicing and abstraction of physical resources into virtual ones, and we showed that setting the hypervisor closer to the RF front-end gives the most of all requirements. Finally, we closed all background related topics by presenting performance metrics for virtualized wireless nodes and vRANs.

3 STATE-OF-THE-ART TECHNOLOGIES FOR VIRTUALIZATION IN RADIO ACCESS NETWORKS

In this chapter, we present the state-of-the-art in RANs virtualization and highlight the open challenges that serve as guidelines for our proposal. In Section 3.1, we present the methodology to classify the most influential research efforts in virtualization for RAN. In Section 3.2, we review research efforts that exploit the concept of virtualization in four different types of RANs for mobile devices. Finally, we identify the gaps in the state-of-the-art in Section 3.3.

3.1 Taxonomy

Since wireless networks include a variety of different technologies, it is difficult to classify the technologies for RANs virtualization by a particular property. Therefore, we will describe the following categorizing methodologies and use them as a taxonomy to classify the enabling technologies for virtualization in radio networks.

- **Radio Access Technology:** Unlike wired networks, the RAT adopted in each type of wireless network is different and often incompatible with each other. Most of the current virtualization efforts focus on IEEE 802.11 networks, cellular networks (including LTE and Worldwide Interoperability for Microwave Access (WiMAX) systems), heterogeneous networks, *i.e.*, a mix of multiple access technologies, among others.

- **Isolation Level:** RAN virtualization can also be classified according to the isolation level. Isolation level refers to the minimum resource units, which isolate the slices from each other. As we mentioned in the last chapter, RANs virtualization may be done at different levels, such as spectrum level, protocol level, or link level.

- **Virtual Resource Type:** In any virtualization technology, the physical resource that is being isolated (given by the isolation level) should be the same as the virtual resource assigned to each slice. This is not always true because the underlying physical hardware restricted some research efforts. Thus, virtual resource type refers to the type of resource assigned to each slice.

- **Purpose:** Originally, RAN virtualization was proposed for experimental purposes in the GENI project. In this project, multiple protocols needed to run simultaneously on the same infrastructure. Virtualization in the commercial market can be considered as the extension of this proposal. Thus, from the purpose's point of view, the research efforts can be classified into experimental and commercial.

3.2 Literature Overview

After defining the taxonomy for research efforts in RAN virtualization, we now discuss these according to different RATs. These enabling technologies are summarized in Table 3.1.

Table 3.1 – Comparison of RAN virtualization frameworks

Radio Access Technology	Authors	Isolation Level	Virtual Resource Type	Purpose	Contribution
IEEE 802.11 (WiFi)	(XIA et al., 2011), (NAGAI; SHIGENO, 2011)	MAC	MAC	Experimental	Enabling access point virtualization
	(BHANAGE et al., 2008), (SINGHAL et al., 2008), (LEIVADEAS et al., 2011)	Spectrum	Link	Testbed	Moving the testbed-as-a-service to the wireless environment
	(SMITH et al., 2007)	Spectrum	Time-slot	Testbed	Implementing virtualization mechanisms on large-scale 802.11 testbed
	(PEREZ; CABERO; MIGUEL, 2009), (BHANAGE et al., 2010), (NAKAUCHI; SHOJI; NISHINAGA, 2012)	Link	Time-slot	Testbed	Enabling link virtualization
IEEE 802.16 (WiMAX)	(BHANAGE et al., 2010a), (BHANAGE et al., 2010b)	Link	Link	Commercial	Introducing the virtual network traffic shaper
	(KOKKU et al., 2012), (KOKKU et al., 2013)	Flow	Flow	Commercial	Enabling simultaneous reservations of two class slices without modifying the MAC schedulers
	(LU et al., 2012), (LU; YANG; ZHANG, 2014)	Spectrum	Sub-carriers	Commercial	Enabling partially slicing and combination of sub-carriers and power allocation
LTE	(SACHS; BAUCKE, 2008), (ZAKI, 2012)	Spectrum	PRB	Commercial	Enabling BS virtualization
	(PHILIP; GOURHANT; ZEGHLACHE, 2012)	Spectrum	LTE Frames	Commercial	Using FlowVisor to slice the BS
	(YANG et al., 2013)	Flow	Link	Commercial	Enabling virtualization in C-RAN
	(GUDIPATI et al., 2013)	Flow	Link	Commercial	Abstracting the entire BS as multiple virtual BSs
Future Mobile Networks	(TAN et al., 2012)	Spectrum	Spectrum	Experimental	Enabling spectrum virtualization in SDRs
	(LIU et al., 2014a), (WANG et al., 2015)	Network	Network	Experimental	Centralized baseband architecture with coarse-grained BBU virtualization
	(AKYILDIZ; WANG; LIN, 2015)	Link	Network	Experimental	Considering the use hypervisors at three different levels in a wireless network
Any	AIRTIME (this thesis)	Spectrum	BBU and RRH	Testbed	Fine-grained BBU virtualization and creating multiple vRRHs to enable multiple independent access technologies on top of one physical RRH

3.2.1 Virtualization in IEEE 802.11 (WiFi)

Xia *et al.* (XIA et al., 2011) proposed a virtualization approach named "Virtual WiFi" to extend the virtual network embedding from wired networks to wireless networks. A kernel-based virtual machine system is used in Virtual WiFi to emulate the wireless devices. Each virtual machine has to establish its wireless connection through its virtual MAC layer. The multiplexing of multiple virtual MACs is performed by assigning a time slot to each one. Nagai and Shigeno (NAGAI; SHIGENO, 2011) explores another aspect of Virtual WiFi, which is the migration of virtual machines to other physical devices and the aggregation of multiple virtual access points for reasons such as reducing energy consumption and spectrum usage. Nagai and Shigeno propose a framework to realize the migration of virtual machines. In this framework, the connection between virtual machines and mobile users is maintained by migrating virtual machines and enabling them on the other physical access point.

Frequency slicing at the spectrum level is used by Bhanage *et al.* (BHANAGE et al., 2008), Singhal *et al.* (SINGHAL et al., 2008), and Leivadeas *et al.* (LEIVADEAS et al., 2011) to isolate transmissions in the wireless medium. Both Bhanage *et al.* and Singhal *et al.* chose the testbed ORBIT as the platform to implement their proposal. Bhanage *et al.* chose OpenVZ to run multiple operating systems on top of the physical device, while Singhal *et al.* uses the *User Mode* Linux operating system on top of the one used by the physical device. In both cases, the operating system in the physical device schedule the resources for the virtual ones. Extending their work, Leivadeas *et al.* propose a novel testbed-as-a-service architecture. Spectrum slicing is used to enable the co-existence of multiple experiments, each with a set of virtual access points.

Smith *et al.* (SMITH et al., 2007) investigate time slicing. By utilizing time slicing, the physical network is partitioned in the time domain across different virtual networks, such that each experiment or virtual access point is isolated. The virtualization mechanism is implemented on a large-scale WiFi testbed facility. Perez, Cabero, and Miguel (PEREZ; CABERO; MIGUEL, 2009) evaluates the time-slicing link virtualization from aspects of delay, jitter, and network utilization. Bhanage *et al.* (BHANAGE et al., 2010), present a similar work but focuses on the fairness issue of the uplink. In this work, the physical access point is allowed to allocate different uplink time quotas for individual virtual access points based on two proposed algorithms. Using these algorithms, the infrastructure can enforce fairness across slices, allowing the physical network to share its resources equitably. Nakauchi, Shoji, and Nishinaga (NAKAUCHI; SHOJI; NISHINAGA, 2012) argue that bandwidth schedule cannot achieve such high utilization when the static resource allocation radio to each slice is preset. Thus, a MAC layer time mechanism to control time quotas is proposed.

3.2.2 Virtualization in IEEE 802.16 (WiMAX)

Several virtualization approaches focus on the IEEE 802.16 standard, also known as WiMAX. Bhanage *et al.* (BHANAGE et al., 2010a) introduces a virtual network traffic scheduler for air interface fairness. Continuing their efforts in virtualization, they also propose a virtual base station architecture and a virtualization substrate (BHANAGE et al., 2010b).

We take the Network Virtualization Substrate (NVS) as an example to show WiMAX virtualization (KOKKU et al., 2012). NVS can be considered as a solution of virtualization not only for WiMAX but also for any RAT in which radio devices are black-box hardware that gives access only to the MAC and higher layers. It runs at the MAC layer of the radio device and operates by assigning entire MAC frames to slices. NVS main components are the *slice scheduler* and *flow scheduler*. To enable isolation, the slice scheduler allows simultaneous reservations of up to two slices, one of which must be bandwidth-based (requests a specific data rate) and one of which must be resource-based (requests a certain amount of spectrum or time-slots). For each MAC frame, the slice scheduler chooses a slice based on the criterion that the utility of the ra-

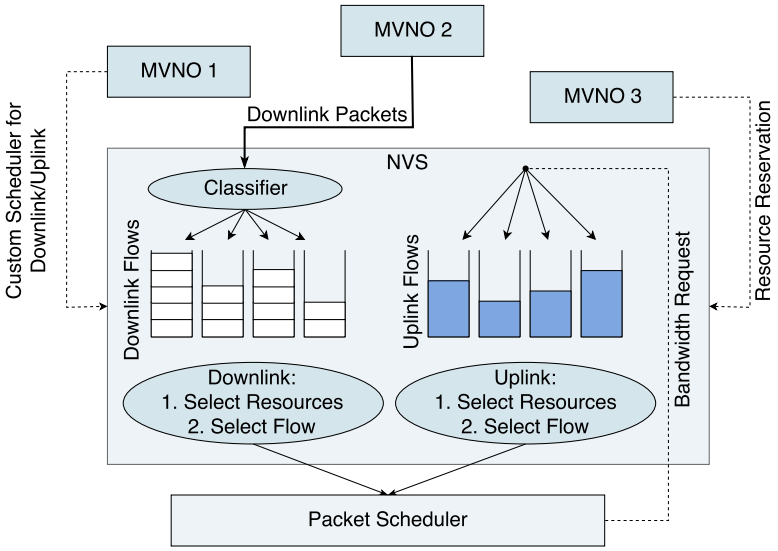


Figure 3.1 – Example of WiMAX virtualization

dio device is maximized. Functions calculate the utility agreed between the slice owner and the owner of the physical radio device in such a way that it can maximize the radio device revenue during its time of operation, while at the same time meeting the slice owner requirements.

After slice selection, the flow scheduler chooses a flow within the selected slice and ensure that each slice can employ custom flow scheduling policies. In other words, by building a generic flow scheduling framework, NVS allows each slice to determine the order in which packets are to be sent in the downlink and the resource slots that are allocated in the uplink. There are three modes in this framework: scheduler selection, model specification, and virtual time tagging. Scheduler selection and model specification provides the customized flow scheduling, whereas virtual time tagging delivers per-flow feedback to the slice by NVS. Each packet arriving into the per-flow queues will be tagged by NVS with a virtual timestamp. NVS can select packets based on this virtual time from the flow queues of each slice.

The authors continue the research efforts started in NVS by proposing a novel system named CellSlice (KOKKU et al., 2013). Firstly, CellSlice overcomes the deployment barrier of NVS, which modifies the MAC schedulers within the BS, by moving the slice scheduling to the gateway. Moreover, CellSlice dynamically adapts parameters of the flow shaping, enabling the following benefits: (*i*) isolation and slice requirements can be satisfied simultaneously, (*ii*) flows within a slice can experience a fair resource allocation, and (*iii*) the resource utilization of the physical BS can be maximized.

Rather than scheduling slices on a per-frame basis as done in NVS and *CellSlice*, Lu, Yang, and Zhang (LU; YANG; ZHANG, 2014) propose to schedule slices on a per sub-carrier basis, which goes one step closer to spectrum level virtualization. One exciting feature of this scheme is that only part of the sub-carriers can be sliced and virtualized. Thus, the authors propose a

slice assignment scheme to separate the carriers from/to the mobile operator owning the device and the SPs interested in the virtual carriers.

3.2.3 Virtualization in 3GPP LTE (Mobile)

The concept of virtualization in mobile networks can be traced back to Sachs and Baucke's *Virtual Radio*, a framework that presents the concept of virtual node and virtual radio (SACHS; BAUCKE, 2008). In Virtual Radio, the decoupling of the data plane and the control plane is defined such that different protocols and management strategies can run on different virtual nodes and links. A *virtualization manager* is responsible for managing the virtualized nodes and links. Although Sachs and Baucke do not provide a practical implementation of Virtual Radio, it lay down the foundations of virtualization in mobile networks.

We take the work of Zaki (ZAKI, 2012), which further develops the Virtual Radio framework, as an example to illustrate the implementation of virtualization in LTE-based mobile networks. This work is illustrated in Figure 3.2. A hypervisor is physically added to the LTE BS and logically allocated between the physical resources and the virtual BSs. The hypervisor takes the responsibility of abstracting the BS into multiple virtual BSs. There are two entities within the hypervisor acting in critical roles. The first one is the Spectrum Configuration and Bandwidth Estimation (SCBE), which is logically located on each virtual BS; the second one is the Spectrum Allocation Unit (SAU), which is located in the hypervisor. At frequent time intervals, the SCBE calculates the spectrum bandwidth required by the virtual BS and sent back this information to the SAU, which then allocates the adequate amount of Physical Resources Blocks (PRBs).

The PRB is the smallest unit that the hypervisor can allocate to virtual BS. The splitting of these resources is performed based on some criteria (*e.g.*, bandwidth, data rates, power, traffic load, or a combination of them). The SAU schedules the PRBs through a contract-based algorithm based on four types of pre-defined contracts: (*i*) fixed guarantees, where a fixed number of PRBs will be allocated, (*ii*) dynamic guarantees, where PRBs are allocated according to the value informed by SCBE and bounded by a maximum value, (*iii*) best effort with a minimum guarantee, where a minimum number of PRBs will be allocated and additional ones may be added in a best-effort manner, and (*iv*) best effort with no guarantees, where PRBs are allocated in a simple best-effort manner.

The concept of wireless virtualization is applied to centralized RANs by Yang *et al.* (YANG et al., 2013). This proposal is a software-defined RAN architecture containing three main parts: the wireless resource pool, the cloud resource pool, and the controller. In this architecture, the wireless resource pool virtualizes one physical RRH into multiple vRRHs with different LTE-based wireless protocols. The cloud resource pool is comprised of a large number of physical processors that run virtual BBUs. The controller takes the responsibility of the control plane of the several complete virtual RANs created by the integration of wireless and cloud

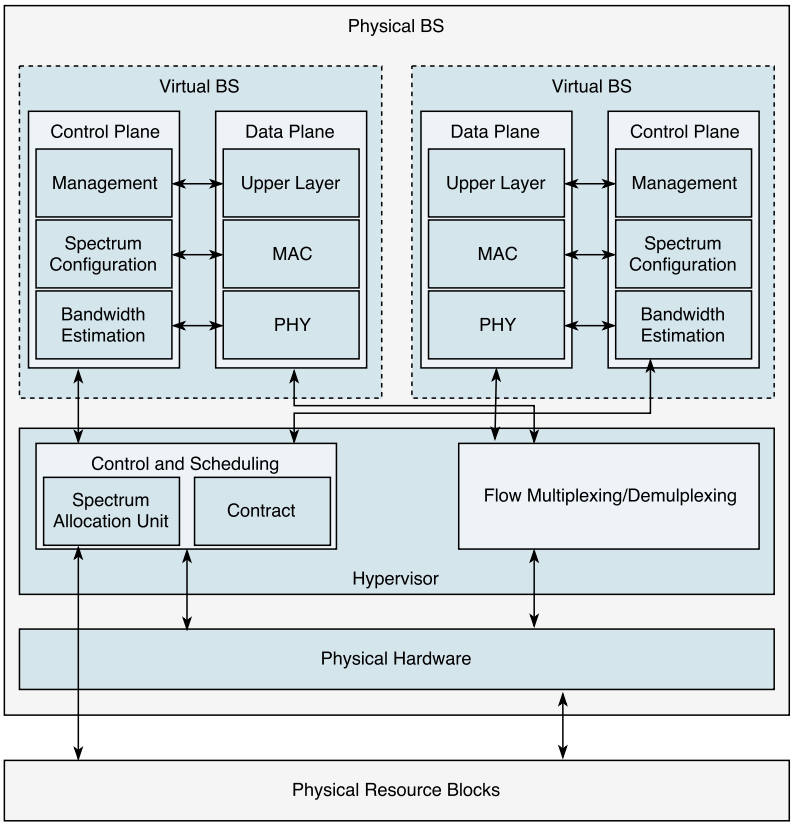


Figure 3.2 – Example of virtualization in LTE base stations

resources. Gudipati *et al.* (GUDIPATI et al., 2013) proposes a more general software-defined centralized RANs by considering all the BSs and RRHs in a geographical area as radio elements and abstracts them as virtual BSs.

3.2.4 Future mobile networks

Future mobile networks should be constructed with centralized physical resource placement and leverage the cloud technology to implement all baseband processing as software modules on top of standard data center infrastructure. Thus, in this subsection, we focus on wireless virtualization efforts based on this architecture.

Tan *et al.* (TAN et al., 2012) designs the Spectrum Virtualization Layer (SVL), a new software layer to facilitate Dynamic Spectrum Access (DSA) in future mobile networks. The goal of SVL is to bridge the gap of traditional RRHs, designed for a fixed frequency band with fixed bandwidth, and the band under DSA, which can have any time and space varying spectrum configuration. SVL is located between the RRH and the baseband processing, decoupling the tight connection between these two. The baseband processing generates digital waveforms, which are reshaped into different waveforms through aggregation, splitting, shrinking, and expanding (as presented in Section 2.2.1.1). Because SVL and the baseband processing are implemented

purely on software, it can enable the deployment of extremely flexible RRHs and can operate in a variety of spectrum bands and access technologies.

Liu *et al.* (LIU et al., 2014a) propose *CONCERT*, a converged edge infrastructure for future mobile communications and mobile computing services. Their architecture is constructed based on the concept of control/data plane decoupling. The data plane includes heterogeneous physical resources such as RRHs, computational resources, and software-defined switches. The *conductor*, a Software-Defined Network (SDN)-like controller, dynamically coordinates data plane physical resources to present them as virtual resources so that they can be flexibly utilized by both mobile communication and cloud computing services. Wang *et al.* (WANG et al., 2015) proposes a similar idea but with a focus on the 3G/4G network architecture, instead of the advanced centralized baseband architecture.

An architecture that truly adopts the "softwarization" paradigm across all layers of the network is proposed in Akyildiz, Wang, and Lin's *SoftAir* (AKYILDIZ; WANG; LIN, 2015). Resembling CONCERT, this architecture adopts control/data plane separation by using SDN. The control plane consists of network management and optimization tools and is implemented on the network servers.

The data plane consists of software-defined BSs in the RAN and software-defined switches in the mobile core network. The control logic for the physical, MAC, and network layers is implemented in software running on general-purpose computers and remote data centers. To isolate virtual networks, SoftAir implements three functions: network hypervisor for high-level virtualization, a wireless hypervisor for low-level virtualization of radio resources, and a switch hypervisor for low-level virtualization of wired resources, as shown in Figure 3.3.

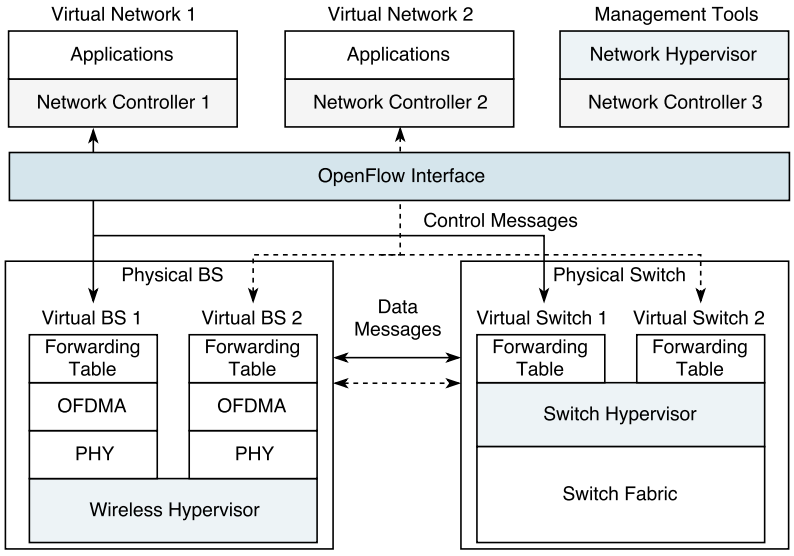


Figure 3.3 – Hypervisor placement in SoftAir architecture

- The network hypervisor is a high-level resource management framework, which allocates non-conflicting network resources to service providers or virtual network operators.

- The wireless hypervisor is a low-level resource scheduler running at the software-defined BSs. It enforces or executes the resource management policies determined by network hypervisor by employing a variety of wireless resource dimensioning schemes or wireless scheduling to guarantee isolation among virtual networks.

- The switch hypervisor focuses on bandwidth management in a single software-defined switch. In particular, bandwidth provisioning on switches offers bandwidth assurance for the designated virtual network.

Similar to previous virtualization efforts, these solutions adopt atomic vBBU functionalities and, consequently, cause bottlenecks in the fronthaul networks and the centralized processing pool. Different from our proposal, all of the above architectures lack a coherent framework that integrates RRH virtualization, in a fine-grained vBBU architecture. Also, RRH virtualization is a key building block for enabling the sharing of radio access networks, *i.e.*, RAN-as-a-Service (RANaaS).

3.3 Identified Gaps

Although different problems in centralized baseband processing are investigated and solutions based on virtualization are presented in state-of-the-art research efforts (SACHS; BAUCKE, 2008; ZAKI, 2012; TAN et al., 2012; LIU et al., 2014a; WANG et al., 2015; AKYILDIZ; WANG; LIN, 2015), we focus our attention to the following gaps identified in the literature:

- Emerging network services, such as eMBBC, mMTC, and URLLC, require highly differentiated access technologies to be integrated and deployed over the same infrastructure. However, current solutions are still designed to support a single "one-size-fits-all" RAN and RAT. Those solutions are not adequate for future mobile networks without a deeper re-design of their fundamental concepts considering this aspect.

- Virtualization solutions adopt an atomic vBBU approach, in which the entire baseband processing is realized in a centralized data center, while the RRHs incorporates only the RF front-end. This fixed distribution of functionalities significantly degrades wireless network coverage area, as the maximum distance between the data center and the RRH is limited due to latency constraints. A proper approach towards the way vBBU operates can mitigate this problem by allowing time-constrained radio functionalities to be closer to the RRH.

- The baseband centralization architecture puts a tremendous bandwidth requirement over the fronthaul network, constraining the fronthaul links to be composed solely of optical links. A proper solution that enables the adoption of heterogeneous fronthaul links can significantly accelerate the adoption of this architecture in future mobile networks.

We can see that there is a multitude of research challenges that must be addressed in the current state-of-art mobile network architectures so that the requirements of 5G networks are to be satisfied. However, what is the design of a flexible architecture that pushes the boundaries of future mobile by addressing these gaps?

To answer this question, we propose a system based on radio and fine-grained baseband processing virtualization to enable future mobile networks to provide connectivity services with multi-RATs, increase the coverage area by moving time-constrained functions closer to the RRH, and adopt flexible fronthauls by reducing the bandwidth required. Moreover, our solution is built to achieve three core principles of future mobile networks: (*i*) programmability, *i.e.*, vRANs can be reprogrammed on-demand to the RAT that best fits the service demands, (*ii*) flexibility, *i.e.*, resources allocated for vRANs can be changed on-demand to satisfy data center requirements, and (*iii*) scalability, *i.e.*, the fine-grained vBBUs enables any fronthaul network by moving specific radio functions closer to the vRRH according to the centralization gain desired. Our solution is presented in the next chapter.

3.4 Summary

This chapter presented the most important research efforts that developed the concept of virtual radio devices. We presented the taxonomy to classify each research effort according to four criteria: (*i*) the RAT that is being virtualized, (*ii*) the selected isolation level that dictates the hypervisor placement, (*iii*) the resource type that is being abstracted, and (*iv*) the purpose to which the research effort is directed.

Afterward, we presented the literature overview according to the underlying RATs that these research efforts used for transmission. Given the research efforts presented, it is clear that most of the frameworks for virtualization in wireless networks are constrained by the fact the BBU is a black-box with little-to-none flexibility. Thus, most of the frameworks are implemented at higher layers of the networking stack, *e.g.*, the MAC layer. We showed that the recent trend of implementing the BBU as a software opened opportunities for lower-level virtualization frameworks. Finally, we closed this chapter presenting the gaps in the current state-of-art frameworks and architectures for virtualization in wireless networks.

4 AIRTIME ARCHITECTURE

AIRTIME is the first prototype system that provides end-to-end vRAN while ensuring isolation and enabling flexible and adaptive provisioning of resources on the SPs service requirements. Therefore, our prototype can support running multiple vRANs on top of the physical infrastructure and tailoring each vRAN to a particular service. To this end, the design of AIRTIME explicitly separates the infrastructure provider and SPs, as shown in Figure 4.1. In the remainder of this chapter, we detail the operation of AIRTIME. We highlight that the two major contributions of this thesis are the fine-grained BBU virtualization and the RRH virtualization.

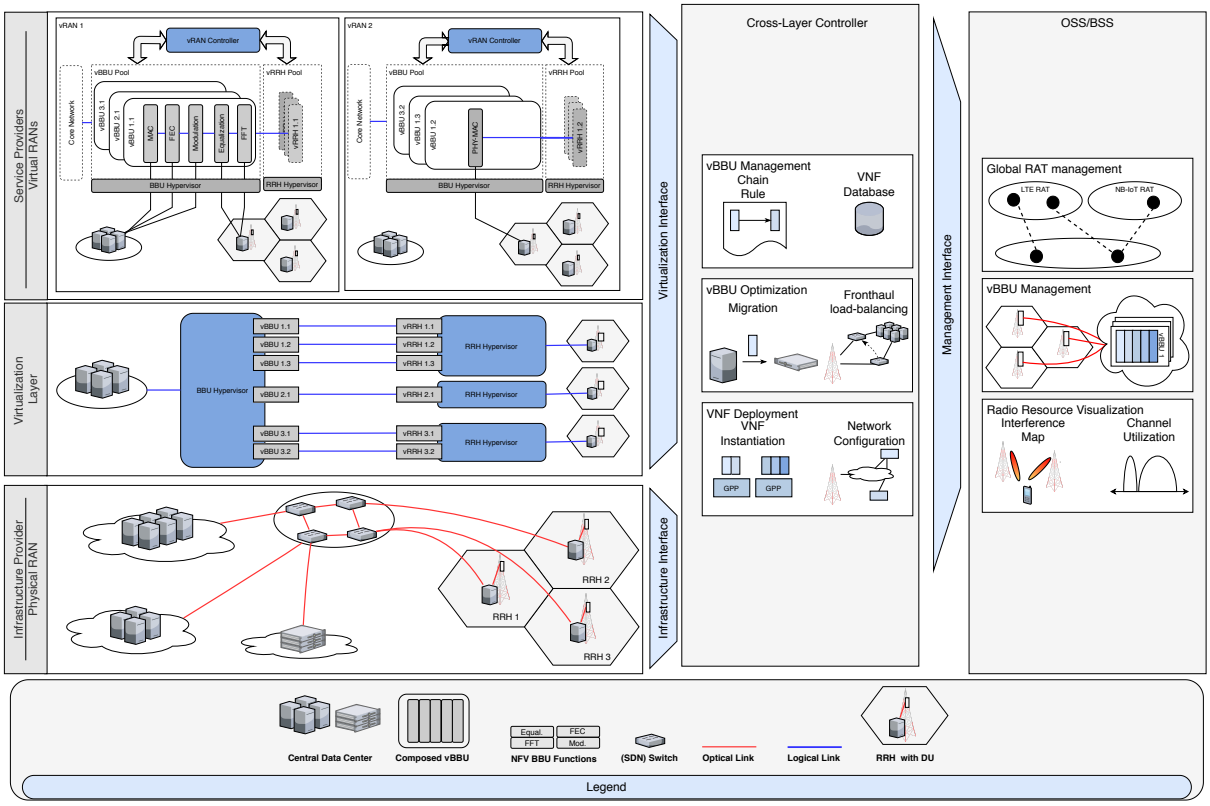


Figure 4.1 – High-level overview of AIRTIME

In AIRTIME, the RAT of each vRAN can be tailored to the particular requirements of the SP, such as Mobile Virtual Network Operators (MVNOs) and vertical markets (on-demand video streaming, internet-of-things for smart cities, and factory automation), by adjusting the radio resources in a vRRH, and especially the baseband functions in the vBBU. At the vRRH, the SP can tailor the radio spectrum bandwidth to serve its users better. It is in the vBBU that enormous opportunities for optimization occur in the form of changing baseband parameters or replacing the entire chain of baseband function VNFs to satisfy the service requirements.

The physical RAN encompasses the elements found in future mobile network architectures (CHECKO et al., 2015): (*i*) the high-capacity CU data centers for performing the non-time critical functions for all vBBUs, (*ii*) the low-capacity DU data center that hosts time-critical

functions of vBBUs that are bound to a physical RRH in a 10-20 Km radius, and (*iii*) a fronthaul network that interconnects CUs, DUs, and RRHs. Having the DU closer to the RRH supports moving baseband processing VNFs closer to the RRH to reduce bandwidth and latency requirements over the fronthaul network.

The virtualization layer is the pillar of AIRTIME's design, as it is where all our novelties are located: the `BBU Hypervisor` and the `RRH Hypervisor`. This layer is responsible for managing the vRAN slices, for ensuring complete isolation of radio, processing, and networking resources (radio resource isolation to guarantee that each slice can adopt a unique RAT, processing resource isolation to ensure performance stability, and networking isolation to ensure that changes in one slice configuration are self-contained). Primarily, the virtualization layer binds multiple isolated vRANs to the physical infrastructure and provides a virtual view of the underlying physical resources allocated to them. This layer must also offer a set of interfaces for managing the vRANs and a set of interfaces to map these changes to the physical resources.

SPs realize their vRAN through the creation of vBBUs and vRRHs over the virtualization layer. A vBBU is a chain of baseband processing VNFs bound to one vRRH, all of them managed by an independent `vRAN Controller`. In the next subsections, we present the main novelties of this thesis: the `BBU Hypervisor` and the `RRH Hypervisor`. Following our main novelties, we present the two architectural elements shared between several future mobile network designs: the `vRAN Controller`, and the `Cross-Layer Controller`.

4.1 BBU Hypervisor

The `BBU Hypervisor` provides the framework for the execution of baseband processing VNFs and their abstraction into a vBBU. The `BBU Hypervisor` follows the ETSI NFV model to abstract the physical infrastructure of the CU/DU data centers. This model already considers a virtualization architecture that comprises a set of interconnected VNFs that are deployed over the physical infrastructure. The different arrangements of the set of VNFs in the physical infrastructure give rise to different composition options, each one with its processing, storage, and networking requirements, *e.g.*, deploying the entire vBBU into a single physical machine requires more resources than deploying only a subset of it as VNFs. Moreover, the `BBU Hypervisor` maps baseband processing VNFs to dedicated virtualization containers (*e.g.*, virtual machine or Linux container) or even physical resources that do not support virtualization, such as Field Programmable Gate Arrays (FPGAs) or Digital Signal Processors (DSPs). The main benefit of this approach is the small virtualization footprint when using containers, while simultaneously enabling the execution of baseband functions in heterogeneous hardware devices.

The workload of the fine-grained vBBU is highly dependent on the access technology it is implementing. An LTE vBBU, for example, implements the encryption/decryption process

of the Packet Data Convergent Protocol (PDCP), the concatenation of the Radio Link Control (RLC) protocol, the radio scheduling and HARQ management handled by the MAC layer, and all the baseband processing functions of the PHY layer: Forward Error Correction (FEC), modulation/demodulation, equalization, FFT and CP addition. In the downlink, packets from the Core Network (CN) arrive at the fine-grained vBBU and are sequentially handled by each of the VNFs in the sequence before being presented in the form of IQ samples at the vRRH; and in the uplink, the same sequential handled occurs, but in the reverse direction. Both downlink and uplink can share the same VNF chain or use two independent chains, with the same or a different number of VNFs distributed in the same or different fashion. The run-time of each baseband processing VNFs is independent of each other and directly relates to the amount of data that comes to/from each RRH.

The VNFs implementing PHY baseband processing requires a large part of the computational capacity. Topping the list are FEC and Receive Processing VNFs, which together add up to more than half of the computational capacity required by an LTE vBBU, in both the downlink and uplink. Different from other PHY level baseband functions, the computational requirements of those two are highly dependent on the radio channel conditions. For example, a channel with low Signal-to-Interference-plus-Noise-Ratio (SINR) requires more data redundancy against errors, which translates into more computational time to perform the encoding and decoding tasks.

Running vBBUs as a composition of baseband processing VNFs enables a higher level of programmability. With this, SPs can tailor any aspect of vBBU by changing the flowgraph of VNFs. For example, by adding a baseband processing VNF that performs carrier aggregation to the chain of functions that build vBBUs. Computationally intensive baseband processing VNFs can be migrated to high-performance FPGAs or DSPs to achieve higher data throughput, at the cost of less flexibility and non-virtualization capabilities. Similarly, different vRANs could employ different baseband processing VNFs setups, optimized for their particular service. Thus, the flexible placement of baseband processing VNFs, coupled with the RRH virtualization described in the next section, allows the creation of fully customizable vRANs that can be tailored to satisfy any service requirements.

4.2 RRH Hypervisor

The `RRH Hypervisor` is one of the main novelties of AIRTIME and is responsible for slicing radio resources. It is responsible for creating the virtualized version of the physical RRH, *i.e.*, the vRRHs, for ensuring their complete isolation, and for facilitating efficient sharing of the underlying physical radio resources. Essentially, the `RRH Hypervisor` for a future mobile network should: (*i*) abstract the physical RRH by adding a layer of indirection that maps physical radio resources to a vRRH, providing a virtual view of underlying radio resources, (*ii*) enable vRANs to adopt any vBBU (*i.e.*, any RAT), and (*iii*) apply changes in the vRRH

by mapping them to the physical RRH, ensuring that these changes do not interfere with other coexisting vRRHs. The `RRH Hypervisor` is an essential part of the infrastructure provider software framework for supporting the concept of RANaaS, *i.e.*, a system in which SPs can request an end-to-end vRAN slice to one or more infrastructure providers.

The `RRH Hypervisor` must be logically located between the physical RRH and vBBU and act as an intermediate layer between these two components. Here, the vBBU interacts solely with a vRRH and does not have any access to its physical counterpart. The vBBU operates as if it is connected to a physical RRH, *i.e.*, generating IQ samples in the downlink and receiving IQ samples in the uplink. In the downlink, the `RRH Hypervisor` intercepts the IQ samples from multiple vBBUs and then reshapes them into a stream of IQ samples that contain the multiplexed signal of all vBBUs. In the uplink, the hypervisor receives the IQ samples of a wideband chunk of spectrum that encompasses the spectrum of all vRRHs. The hypervisor then forwards to the vBBUs only the IQ samples of the portion of the radio spectrum corresponding to the respective slice. Moreover, vRRH must provide the same set of configuration options of the physical RRH, such as radio channel (or center frequency), bandwidth, and transmission/reception gains.

The `RRH Hypervisor` should be platform agnostic to enable vRANs to adopt any vBBU or combination of baseband processing VNFs. As we have seen in our state-of-the-art revision, adhering to such requirements is a challenging task, as different RATs have different abstractions for the underlying wireless spectrum. For example, some RATs split the full radio spectrum bandwidth into smaller chunks of transmission with a constant duration (such as LTE and WiMAX physical resources blocks), while others allocate the entire bandwidth to users and change the transmission interval according to the traffic demands (such as WiFi and LoRA transmission frames). The `RRH Hypervisor` should not use any of these abstractions; otherwise, it would make the slicing mechanism overly complicated, especially if it needs to operate with multiple abstractions simultaneously.

Configurations performed in the vRRH must be mapped to internal settings of the `RRH Hypervisor` or configurations in the physical RRH. The hypervisor should guarantee that any of such configurations do not cause any interference with other coexisting vRRHs. This guarantee is one of the fundamental features that any virtualization layer must adhere to, and the complexity of such a task is related to the internal slicing mechanism used by the `RRH Hypervisor`. The fact that the `RRH Hypervisor` handles only IQ samples significantly eases the fulfillment of this requirement because most, if not all, the configurations can be translated to modifications in the stream of IQ samples associated with the vRRH.

Based on the characteristics mentioned above, we can summarize the main requirements for our `RRH Hypervisor` as:

- **Coexistence:** the hypervisor must ensure that multiple vRRHs can coexist on the same physical RRH. Moreover, since vBBUs can implement different RATs, their requirements can be vastly different. The hypervisor must support multiple simultaneous vB-

BUs, implementing RATs with varying processing constraints, bandwidths, or channel access schemes.

- **Isolation:** the hypervisor must ensure that any configuration or misconfiguration in a vRRH will not be able to affect and interfere with other vRRHs. Since several vRRHs can coexist on top of the same physical RRH, isolation is the fundamental requirement that guarantees fault tolerance, security, and privacy (LIANG; YU, 2015).

- **Programmability:** vRRHs should have a level of programmability similar to standard physical RRHs. This means that a vRRH hold the same set of functionalities of its physical counterpart, *e.g.*, configuration of the center frequency, bandwidth, and transmission/reception gains, while fulfilling the previous two requirements.

4.3 vRAN Controller

The `vRAN Controller` is a logical entity that gives SPs the capability to manage and control their set of vBBUs and vRRHs in a way that best fits their application requirements. The `vRAN Controller` runs as a high-level orchestrator, responsible for tailoring the functionality and managing the allocation of resources to applications and users associated with the SP's vRANs as if it was operating using the physical infrastructure. In this sense, the `vRAN Controller` is similar to an SDN Controller; but with a focus on the optimization of the wireless interface between mobile users. It is important to highlight that this thesis does not focus on the heuristics or algorithms to optimize the assignment and use of physical resources to vRANs. However, AIRTIME offers all interfaces to perform such optimizations.

We can define three APIs for the `vRAN Controller`: (*i*) northbound API, (*ii*) southbound API, and (*iii*) eastbound API. The northbound API is used by SP's applications to request the tailoring of high-level requirements (such as throughput, latency, and packet loss) to a particular set of users. The southbound interface, in turn, is used by the `vRAN Controller` to interact with the vBBUs and vRRH to (*i*) tailor the virtual infrastructure so that the requirements of the application are met and (*ii*) react to events occurring in the physical infrastructure that propagate into changes in the virtual elements. The eastbound interface is used to interact with the infrastructure provider to manage the life-cycle of all virtual components within a vRAN, *e.g.*, creation, installation, migration of a particular baseband processing VNF, or entire vBBUs.

The `vRAN Controller` is also responsible for implementing the control protocols required for the communication and coordination of vBBUs and vRRH with the rest of the mobile infrastructure (*e.g.*, S1 and X2 interfaces in LTE). This communication means that all operations defined for a given RAN architecture can be supported as long as the appropriate interfaces and messages are implemented as part of the respective `vRAN Controller`. The communication of the `vRAN Controller` with the `BBU Hypervisor` and the `RRH Hypervisor` is

message-based and shares the physical network with the data traffic, *i.e.*, the data of baseband processing VNFs and vRRHs. This facilitates the `vRAN Controller` to be deployed as a VNF sharing the same physical infrastructure of the hypervisors or even in different CU/DU data centers. Moreover, it also gives the SPs flexibility to design their management system with varying levels of centralization based on their service needs and allows coordination among vRANs under the same SP ownership.

4.4 Cross-Layer Controller

The `Cross-Layer Controller` is a logical entity (which can be distributed into different physical boxes to improve adaptability and performance) responsible for the management and orchestration. The main functionalities of the `Cross-Layer Controller` are: (*i*) vBBU management, (*ii*) vBBU optimization, and (*iii*) VNF deployment. vBBU management covers all aspects related to the life-cycle of fine-grained vBBUs (such as creation, installation, and migration of baseband processing VNFs) according to the available physical resources, while at the same time ensuring that processing and latency requirements are fulfilled during the fine-grained vBBU operation. To this end, the `Cross-Layer Controller` must be aware of the resources required by each baseband processing VNF composing a vBBU chain. Baseband processing VNF deployment encompasses the steps to instantiate VNFs in the chosen processing resource.

Optimizing fine-grained vBBUs includes a broad range of adjustments, such as VNFs distribution, selection of hardware to execute VNFs aiming to increase the overall baseband processing performance or reduce the energy consumption, configuration of VNF distribution among the CU/DU data centers, as well as adjustments in the fronthaul network forwarding to reduce bottlenecks. Such optimizations should be located in the `Cross-Layer Controller`, as it has accurate and instantaneous information of all baseband processing VNFs. Due to the time-sharing nature of processing resources, different bandwidth and latency constraints, and heterogeneous processing capabilities in the data centers, such joint optimizations come with challenges because of its non-convex nature, making its implementation a research topic on its own (MAROTTA et al., 2018; PAPA et al., 2019).

The `Cross-Layer Controller` interacts with other components of our solution using three well-defined interfaces:

- A *Management Interface* with external applications allows gathering information regarding any aspect of the network from the `Cross-Layer Controller`. It enables the direct management of baseband processing VNFs, such as instantiation, distribution, and configuration.

- A *Virtualization Interface* with the Virtualization Layer provides the means to instantiate fine-grained vBBUs, and migrate and monitor VNFs.

- An *Infrastructure Interface* is used to configure the physical resources to match the expected behavior of fine-grained vBBUs, *e.g.*, the forwarding path. This interface must implement a trap system to notify the `Cross-Layer Controller` in cases of conflicting configurations or anomalous operations.

The Operations Support System/Business Support System (OSS/BSS) interacts with the `Cross-Layer Controller` to gather information from the network, *e.g.*, interference map, and channel and processing hardware utilization. Based on this, an operator can manage the RAN infrastructure with a global view of the vRANs and their associated resources.

4.5 Putting all together: vRAN Instantiation

The main interactions during the instantiation of a vRAN are illustrated in Figure 4.2. The network operator utilizes the OSS/BSS to specify the characteristics of the fine-grained vBBU, *e.g.*, RAT, center frequency, and channel bandwidth (1). The description of the fine-grained vBBU is sent to the `Cross-Layer Controller` through the Management Interface (2). After that, the `Cross-Layer Controller` must select physical processing resources that can fulfill processing and latency demands (3) and request the instantiation of the baseband processing VNF(s) through the Virtualization Interface (4). Next, the baseband processing VNFs are instantiated (5) and their data forwarding configured to compose the fine-grained vBBU (6). When the process is finished, the `Cross-Layer Controller` sends a notification message to the original application through the Management Interface (7). After that, the `Cross-Layer Controller` can optimize the distribution of the baseband processing VNFs (8).

Many challenges appear in this use case. For example, step (2) requires a comprehensive baseband processing VNF description language that can express rules to deploy them, and instructions to handle failures or to solve conflicting configurations. Selecting the appropriate set of processing resources to be used in step (3) requires up-to-date information on the network resources, which is hard to obtain because of fast fluctuations caused by mobile subscriber mobility, wireless channel characteristics, and even other vBBUs. Step (5) requires the development of a cross-platform system that abstracts the underlying processing resource, while at the same time making efficient use of different hardware capabilities. Finally, step (8) requires the development of algorithms to solve a non-convex optimization problem.

Fine-grained vBBU enables RRH densification, multi-RAT, and heterogeneous fronthauls, essential for future mobile network deployments. First, DU facilitates the densification of RRHs by moving interference mitigation techniques implemented in baseband processing VNFs closer to the RRH, allowing fast adaptation of transmission and reception parameters according to changes in the wireless environment. Second, the `RRH Hypervisor` presents unique opportunities to deploy multi-RAT, as its implementation should be technology agnostic. Moreover, RATs can be reconfigured by changing specific baseband processing VNF parameters, similarly

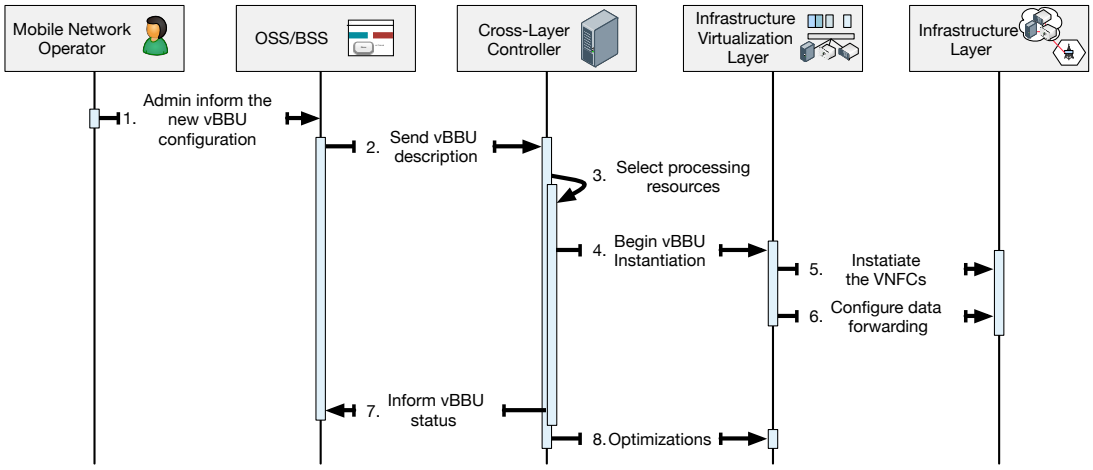


Figure 4.2 – High-level interactions of AIRTIME

to what is done in current baseband processing functions implemented in Software-Defined Radio (SDR) platforms (FONT-BACH et al., 2015), or by adding or removing functionalities, *e.g.*, by adding a baseband processing VNF to the fine-grained vBBU chain that performs carrier aggregation. Third, the dynamic and flexible distribution of VNFs enables heterogeneous fronthaul links. For example, all baseband processing VNFs can be moved to the regional data center if high bandwidth and low latency fronthaul links are connecting it to the physical RRHs. In the case of low bandwidth or high latency links, VNFs implementing physical layer functions can be moved to the DU data center, while the CU data center takes responsibility only for the MAC layer and higher layer functions. This flexibility is also necessary to deal with stringent latency requirements of mobile standards.

4.6 Summary

AIRTIME design enables multi-RAT capabilities, high coverage area, and flexible fronthaul support in future mobile networks. Our proposal introduces two significant innovations to realize this: (*i*) the `BBU Hypervisor` that enables the fine-grained distribution of baseband processing VNFs, and (*ii*) the `RRH Hypervisor` that enables the virtualization of the physical RRH. The integration of both innovations adds the programmability, flexibility, and scalability much needed in future mobile networks.

The BBU and RRH virtualization capabilities, allied with the distribution of baseband processing VNFs within the physical infrastructure, enable AIRTIME to offer a multi-service and multi-tenant RAN. SPs can tailor their vRAN to the requirements of their services by changing the baseband processing VNF according to their needs, *i.e.*, similarly to what is done in current baseband processing functions implemented in SDR platforms, or by adding or removing a baseband processing VNF from the fine-grained vBBU chain. The CU/DU data centers play a major role in services with latency constraints, such as URLLC, by reducing the fronthaul

bandwidth requirements so to allow the execution of baseband processing VNFs closer to the end-user. For example, all baseband processing VNFs can be moved to the DU center if low bandwidth or high latency links are connecting the DU and the CU. In the next chapter, we present the prototype of AIRTIME that implements most of the functionalities described in this chapter.

5 AIRTIME DESIGN AND IMPLEMENTATION

In this chapter, we present the implementation details of AIRTIME. We start with the `BBU Hypervisor`, focusing on the technologies used to realize our vision of flexible and programmable vBBUs. Afterward, we present the HyDRA (our `RRH Hypervisor` implementation) and how it enables multiple heterogeneous RATs to coexist on top of the same RRH. Finally, we present the main interactions of our prototype for instantiating a vRAN.

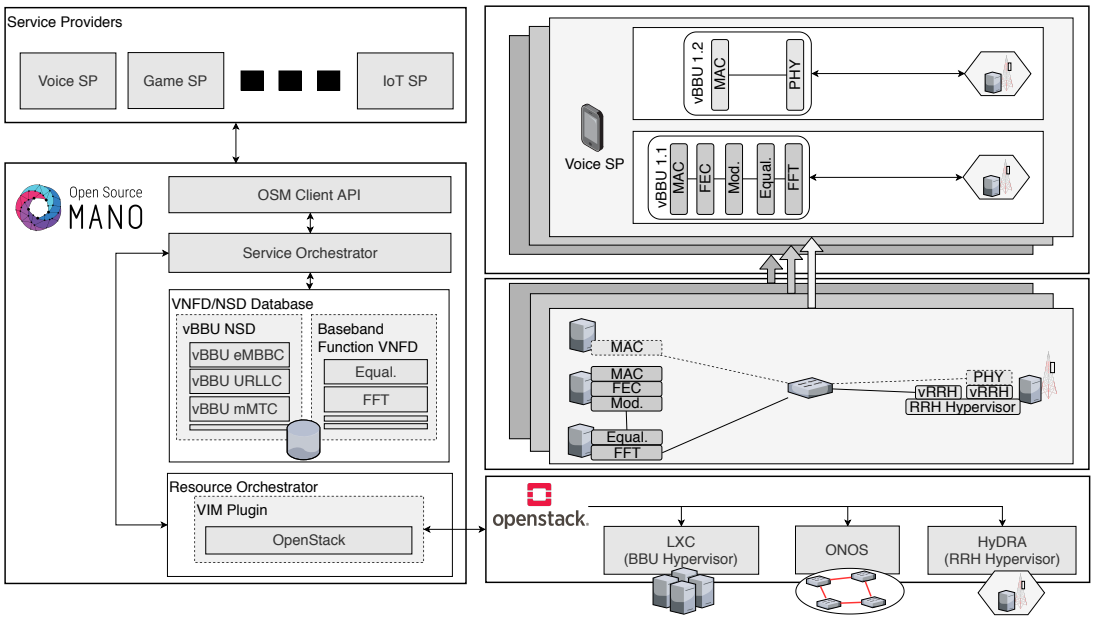


Figure 5.1 – Main components of AIRTIME

5.1 BBU Hypervisor and Cross-Layer Controller

In this section, we present the implementation of the `BBU Hypervisor`, responsible for realizing our view of flexible fine-grained vBBUs, and the `Cross-Layer Controller`. The software used to prototype AIRTIME combines the functionalities of these two components, and for this reason, we decided to describe them together.

The `BBU Hypervisor` comprises three top-level components: OpenStack, LXC, and ONOS. OpenStack is the virtual infrastructure manager that bridges Open Source Mano (OSM) with the three main physical resources of the data centers: RRHs, fine-grained vBBUs, and the SDN network interconnecting them. In the data centers, we use LXC and ONOS. The former is the light-weight container-based virtualization system that allows running baseband processing VNFs on the physical computers; and the latter is used to configure the flow tables of SDN switches, so that the high-level flowgraph of the fine-grained vBBU is embedded in the physical network.

OSM is the open-source and industry-leading software for managing VNFs in commercial data centers. OSM supports the deployment of VNFs as VMware virtual machines, LXC, or Docker containers. More importantly, OSM (Release 4 - R4) provides the functionality that enables: (*i*) providing a database of VNFs and vBBUs that can be deployed in the physical infrastructure, (*ii*) interfacing with OpenStack, ONOS, and the `RRH Hypervisor` to deploy an end-to-end vRAN over the physical infrastructure. (*iii*) requesting the creation or configuration of a vRAN using the OSM client API, and (*iv*) managing the life-cycle of all virtual resources, including vRAN, vBBU, and baseband processing VNFs. Here we can see the merge of functionalities of the `BBU Hypervisor` (items (*i*) and (*ii*)) and the `Cross-Layer Controller` (items (*iii*) and (*iv*)).

SPs request the creation or configuration of a vRAN by sending a JSON data object to the OSM client REST API. The JSON message specifies the physical RRH in which the SP wants over-the-air coverage (on top of which the vRRH will be created) and the vBBU. Specifying the vBBU can be done in one of two options: in the first option, the SP can use one of the available vBBUs in the OSM database, such as eMBBC, URLLC, or mMTC; in the second option, the SP can specify a set of baseband VNFs (also from the OSM database) that together form the desired vBBU functionality (in this case the chaining is done automatically by OSM with the chaining rules already set for each VNF). These two options allow for SPs that do not have the expertise in telecommunications to use readily available vBBUs customized for particular types of applications or for SPs with the know-how to build their vBBU from scratch.

Each type of vBBU is specified as a Network Service (NS) descriptor, which are JSON data objects with standardized fields that OSM can interpret, such as fields that specify the constituent VNFs of the vBBU and the connection between these VNFs. Similarly, the baseband VNFs descriptors are JSON data objects, but in this case, the main fields specify the computational resources, *i.e.*, network interfaces, storage, and processing, and the baseband processing task that VNF must execute after its initialization. All descriptors are saved in an internal database in OSM and are available to SPs through the OSM client API.

OSM also performs the life-cycle management of vBBUs and vRRHs. In our prototype, this management is limited only to the creation of vBBUs and vRRHs according to the available physical resources. The current release of OSM (version 4) lacks the capability to migrate baseband VNFs in response to events in the physical infrastructure that impact the vBBU performance. This functionality could be of great importance for commercial infrastructure providers that own multiple data centers and thousands of RRHs. However, we do not explore it in our prototype.

The last functionality of OSM to be described here is its integration with OpenStack and ONOS (which together are our `BBU Hypervisor`). OSM has the instructions on how to build a vBBU based on the NS and VNF descriptors, which are used to control OpenStack to instantiate the baseband processing VNFs in the physical infrastructure. After the instantiation

of all baseband processing VNFs, OSM interacts with ONOS to configure the SDN network in such a way that the VNFs follow the chain of baseband processing of the vBBU.

5.1.1 Fine-grained Baseband Processing VNF

Before moving to the `RRH Hypervisor`, we need to give an overview of how the fine-grained baseband processing VNFs of the vBBUs chained. Figure 5.2 illustrates how the chaining of VNFs is realized.

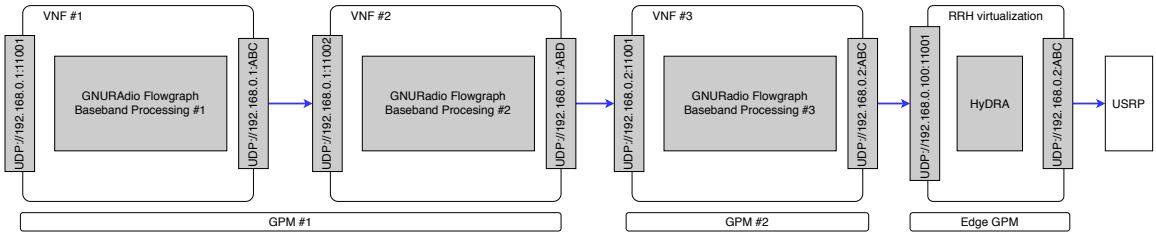


Figure 5.2 – Chaining of Fine-Grained Baseband Processing VNFs

Each VNF runs in a LXC container. This VNF has two major roles: receiving/sending data to/from the next/previous fine-grained VNF in the baseband processing chain and running the GNURadio flowgraph. The receiving/sending of data is performed using the UDP protocol. Each VNF that wants to receive data listens for data at a hard-coded UDP port, whereas the previous VNF in the chain needs to be configured with the IP and UDP port of the next VNF in the chain.

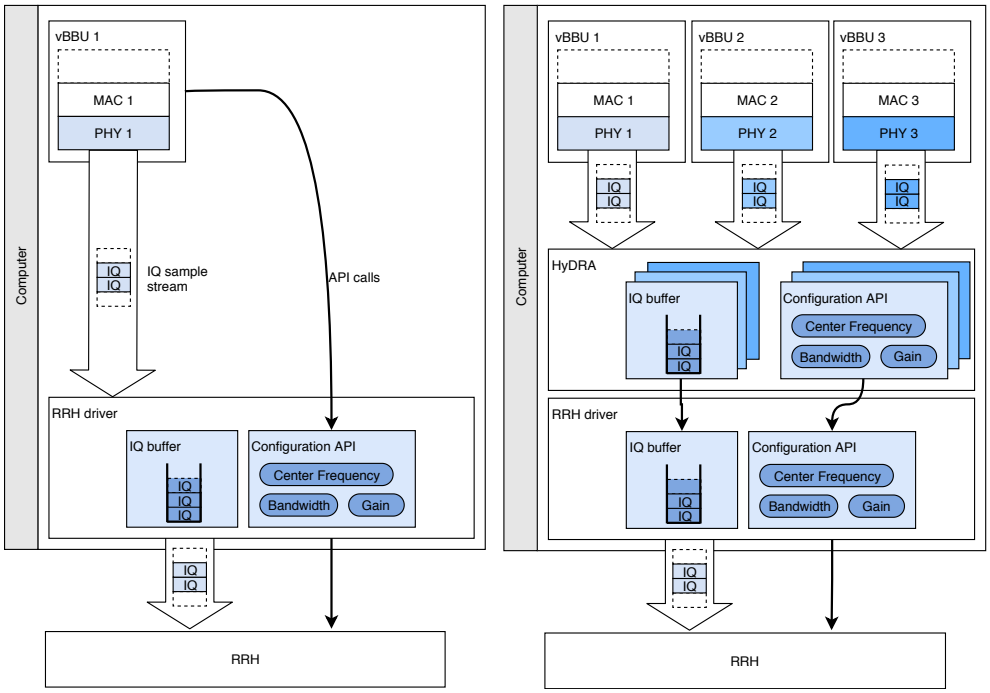
All data received by the VNF is forwarded to the GNURadio Flowgraph. The VNF expects the data received to be correct and representing the correct information, *e.g.*, IQ samples for low-PHY VNF, and user data for MAC VNF The GNURadio Flowgraph processes all data received and generates the output data, which is then forwarded to the next VNF in the baseband processing chain.

Finally, the end-point of all vBBUs is the vRRH. As we have seen, the vRRH are created by the `RRH Hypervisor`. The next section details the implementation of the `RRH Hypervisor`.

5.2 HyDRA– The Hypervisor for software-Defined RAdios

In this section, we present HyDRA, the implementation of the `RRH Hypervisor`. We will show that HyDRA, with the best of our knowledge, the most advanced RRH virtualization layer developed so far, due to the internal operations used to realize the RRH slicing, while adhering to the coexistence, isolation, and programmability requirements described in the Chapter 4.

RRH virtualization is "the process of abstracting a physical RRH and slicing it into vRRHs holding certain corresponding functionalities and isolating each other" (LIANG; YU, 2015). In other words, it is the process of abstracting, slicing, isolating, and sharing the radio hardware between multiple virtualized parts that hold all or part of the functionalities of their physical counterpart.



(a) Standard RRH platform. Only one vBBU can access the RRH

(b) Virtualized SDR platform. In this case the RRH is abstracted, sliced, and shared among multiple vBBU

Figure 5.3 – Comparison of a standard SDR platform and a virtualized SDR platform

Figure 5.3(a) illustrates a conventional RRH platform. It comprises a vBBU, the RRH driver, and the RRH. The vBBU implements the signal processing operations related to transforming a sequence of bits, *e.g.*, user data, into digitized IQ samples that represent the radio signal that must be transmitted (external applications usually perform the data acquisition). The vBBU can be executed on top of general-purpose processors, taking advantage of highly-optimized signal processing libraries. After performing its baseband operations, the vBBU then transfers the generated IQ samples to the RRH driver.

The hardware vendor of the RRH usually implements a driver that provides an API to enable vBBUs to recover and send digitized IQ samples from the RRH and to configure, among other parameters, the RRH center frequency, bandwidth, and transmission/reception gain. The RRH is responsible for translating the digitized IQ samples into/from radio signals that are transmitted/received by the antenna.

Each vRRH must have its own center frequency, bandwidth, and transmission/reception gain configuration. Such abstraction requires the design and implementation of the `RRH Hypervisor`

in line with the requirements described previously. Figure 5.3(b) illustrates where the `RRH Hypervisor` must be logically placed. In the next subsection, we present the design choices for HyDRA so that it complies with the requirements of coexistence, isolation, and programmability.

5.2.1 HyDRA Internal Architecture and Configuration API

The internal architecture of HyDRA is shown in Figure 5.4. HyDRA adds a layer of indirection between the physical RRH and the fine-grained vBBUs executing in the CU/DU data centers. vBBUs operate as if they were interfacing directly with a standard RRH by sending/receiving digitized IQ samples. HyDRA ensures isolation while allowing vBBUs to configure the central frequency, bandwidth, and transmission/reception gain on-the-fly. HyDRA makes use of a spectrum map to keep track of these configurations.

The spectrum map is flexible in HyDRA. For example, vBBUs can request any central frequency or bandwidth, as long as bands of operation of different vBBUs do not overlap. Usually, these configurations are set during the bootstrap of HyDRA, but our architecture allows for computational inexpensive on-the-fly changes, as they only trigger an update in the IQ mapping.

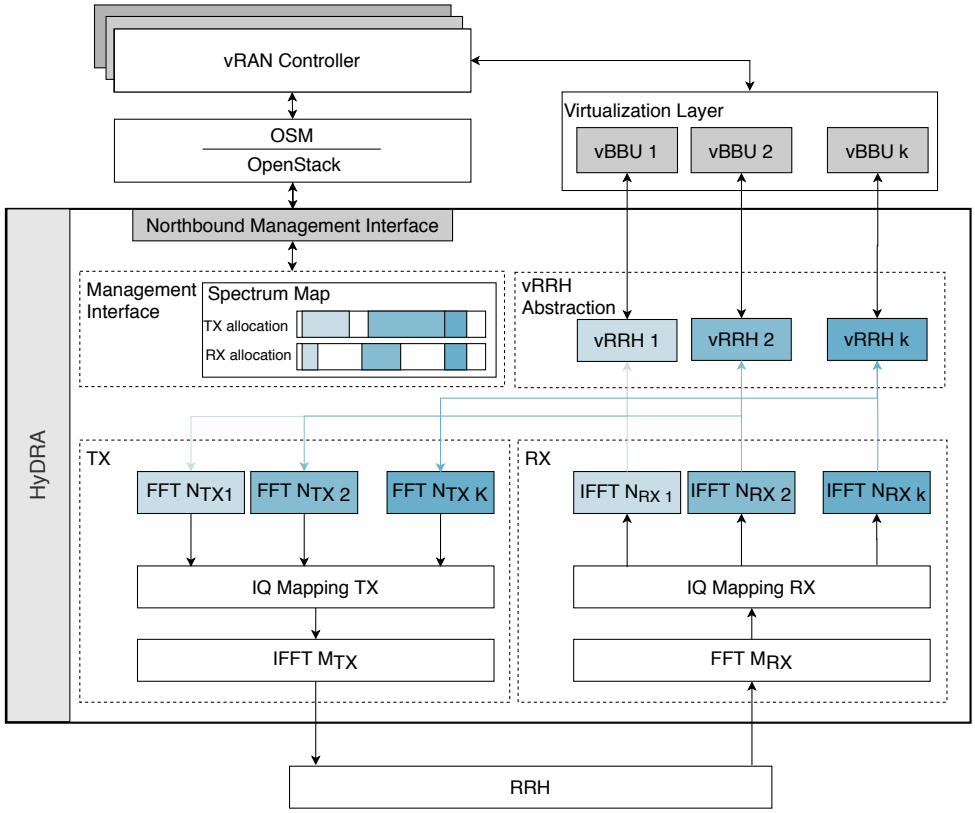


Figure 5.4 – Main architectural blocks of HyDRA

The core and challenging part in designing HyDRA was to multiplex the incoming digitized IQ samples of each vRRH into a single signal that is transmitted by the physical RRH.

Our multiplexing is based on FFT/Inverse FFT (IFFT) operations, as shown in Figure 5.5. The frequency-domain decomposition and recomposition using the FFT $n_{\text{TX i}}$ and IFFT M_{TX} operations retains the transparency property of the virtualization. It ensures that the frequency components generated by the vRRH are always mapped to the same component in the radio spectrum of the physical RRH. Therefore, the signal received by devices at the other end of the physical RRH would appear perfectly as if the vBBU is interfacing with a physical RRH. In the remainder of this subsection, we show how the multiplexing is performed in the downlink, *i.e.*, from vRRHs to end-users.

Initially, the incoming IQ samples from vRRH i are transformed from time to frequency domain in a FFT with $n_{\text{TX i}}$ points. $n_{\text{TX i}}$ is a function of the bandwidth of the vRRH and the sampling rate of the physical RRH. The resulting $n_{\text{TX i}}$ frequency components are mapped into the buffer of an IFFT with M_{TX} points following the Spectrum Map configuration. Thus, $n_{\text{TX i}}$ specifies the resolution used to multiplex the signal of the $i-th$ vRRH with the other slices coexisting in the physical RRH. HyDRA calculates $n_{\text{TX i}}$ (Equation 5.1) as a rate of the bandwidth $B_{\text{TX i}}$ with the total bandwidth B_{TX} and M_{TX}.

$$n_{\text{TX i}} = \left\lceil \frac{B_{\text{TX i}}}{B_{\text{TX}}} \times M_{\text{TX}} \right\rceil \quad \forall i \tag{5.1}$$

The value of the IFFT M_{TX} should be sufficiently large to cover the entire physical band used by the physical RRH with a frequency resolution that is equal or better than the frequency resolution of any instantiated vRRH. The value of M_{TX} is defined before during HyDRA's bootstrap and is manually defined. As a rule of thumb, its value is one of $[1024, 2048, 4098, 8196]$.

The mapping consists of moving the IQ samples of the FFT i to the correct bins of the IFFT M_{TX}. This process requires the definition the first and the last bin of the IFFT M_{TX} to which the $i-th$ vRRH is mapped, $m_{\text{TX i, 0}}$ and $m_{\text{TX i, 1}}$, respectively. The definition of these is a function of the center frequency F_{TX} and bandwidth B_{TX} of the RRH, and the center frequency $f_{\text{TX i}}$ and bandwidth $b_{\text{TX i}}$ of the $i-th$ vRRH, as shown in the Equation 5.2. After the mapping, we perform the IFFT to convert the frequency components of all vRRHs into the resulting multiplexed time-domain signal, which can be transmitted by the physical RRH.

$$\begin{aligned} m_{\text{TX i, 0}} &= \frac{f_{\text{TX i}} - F_{\text{TX}} - \frac{b_{\text{TX i}}}{2} + \frac{B_{\text{TX}}}{2}}{\frac{B_{\text{TX}}}{M_{\text{TX}}}} \\ m_{\text{TX i, 1}} &= m_{\text{TX i, 0}} + n_{\text{TX i}} \end{aligned} \tag{5.2}$$

The multiplexing process, based on FFT/IFFT operations, is a perfect fit for our virtualization objectives. First, FFT/IFFT operations are low-level baseband processing operations agnostic of the access technology implemented by the vBBU mapped to the vRRH; this enables

any vBBUs to be mapped to a vRRH, independently of the access technology it implements. Second, modifications in the original signal caused by the multiplexing are not distinguishable from well-known wireless disturbances, *e.g.*, path-loss, frequency shift, and phase distortion; this allows receiving devices to recover the data transmitted using conventional physical layer equalization mechanisms. Finally, the FFT/IFFT are computationally efficient operations, highly optimized for modern processors through Single Instruction Multiple Data (SIMD); this allows HyDRA to multiplex multiple vRRHs simultaneously while using only a fraction of the resources of modern processors.

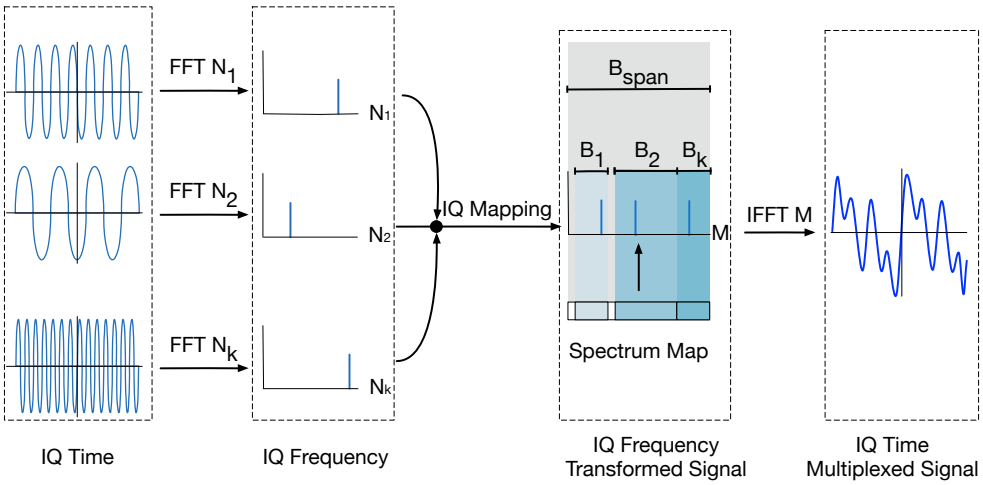


Figure 5.5 – Multiplexing process performed by HyDRA

5.2.2 HyDRA Configuration API

HyDRA defines a set of configuration APIs to enable the `vRAN Controller` or the `Cross-Layer Controller` to configure the vRRH. These APIs are summarized in Table 5.1.

The interface *create_rrh* can be used to create a new slice on the physical RRH and abstract the slice onto a vRRH, which is accessed in the same way that a standard RRH would be. The *bandwidth* parameter determines the spectrum bandwidth of the vRRH, and *center_frequency* specifies its center frequency. Before creating the vRRH, HyDRA verifies whether the bandwidth and the center frequency requested are valid, checking that the requested channels do not overlap with another vRRH and whether the physical RRH supports the required configurations. If accepted, HyDRA returns an object representing a standard RRH, which is used by the vBBU.

The remaining API provide the functionalities that a vBBU expects from a standard RRH. The *get_bandwidth*, *get_center_frequency* and *get_gain* interfaces communicate with the spectrum map and return the current bandwidth, center frequency or gain of the vRRH. Similarly,

Table 5.1 – Main HyDRA APIs

Return type	API name	Parameters
rrh*	create_rrh	(int bandwidth, float center_frequency
void	set_bandwidth	(int bandwidth)
int	get_bandwidth	()
void	set_center_frequency	(float center_frequency)
float	get_center_frequency	()
void	set_gain	(float decibels)
float	get_gain	()
void	send_samples	(IQ_samples *buffer, int size)
int	recv_samples	(IQ_samples *buffer, int buff_size)

their *set* versions allow a vBBU to change its bandwidth, center frequency, and gain, granted that the spectrum map accepts the requested configurations.

vBBUs call *send_samples* to forward a buffer of IQ samples to HyDRA. The parameters *buffer* and *size* specify the pointer and the number of digital samples to send, respectively. Similarly, vBBUs use *recv_samples* to receive all samples since the last call to this function. HyDRA fills *buffer* until all samples for the VR are transferred or until *buff_size* is reached.

HyDRA is designed to perform virtualization of the RRH. At this level, HyDRA provides the flexibility of spectrum level virtualization. This level of virtualization enables AIRTIME to provide independent RAT on top of the same RRH. In the next section, we present how the `BBU Hypervisor` and HyDRA come together to realize the creation of an end-to-end vRAN slice.

5.3 Putting all together: creating a vRAN slice

So far, we have described each of the main components of AIRTIME's prototype. We now address how the main components of our prototype are integrated for the creation of a virtual base station within a vRAN slice. Starting with the request from the `vRAN Controller`, OSM must deploy all baseband processing VNFs of vBBU for the selected RAT and request the creation of a vRRH slice with HyDRA on top of the chosen physical RRH. After the successful deployment, ONOS automatically installs correct flows on the SDN switches so that all virtual elements are part of the same virtual network as the vRAN Controller. While the vRAN slice is active, the `vRAN Controller` can interact with the baseband processing VNFs and vRRHs to optimize any aspect of a virtual base station. The `vRAN Controller` can also interact with OSM to request the destruction of the virtual resources.

HyDRA can be turned into a VNF or be executed as standard software on top of the physical machine. As a rule of thumb, we want all HyDRA instances to be hosted in DUs data centers, *i.e.*, close to the physical RRH, while the baseband processing VNFs can be distributed between the same DU and CU data centers (there are of course some rare configurations to which this rule does not apply). Because HyDRA is responsible for the RRH slicing, its execution

must outlive all baseband VNFs and should not be interrupted during the entire operation of AIRTIME. HyDRA runs on any computer with GNURadio. It is publicly available online at GitHub (https://github.com/maiconkist/gr-hydra). The complete source code is accompanied by several examples with configurable parameters such as the number of vRRHs, central frequency, and bandwidth of vRRH, and GUI elements such as spectrum waterfall and plotters to visualize data transmitted and received.

5.4 Summary

This chapter presented the design choices and implementation of the two main components of AIRTIME: the `BBU Hypervisor` and the `RRH Hypervisor`. We started with the `BBU Hypervisor`, showing that its implementation is based on open-source software widely used in commercial data centers and testbeds for mobile networks. The prototype `BBU Hypervisor` has its functionalities "merged" with the `Cross-Layer Controller`; it assumes the Management & Orchestration (MANO), *e.g.*, fine-grained vBBU management and optimization and baseband processing VNF deployment, and the interfaces used to interact with the physical and virtual infrastructure.

In the sequence, we presented the internal architecture of HyDRA (our `RRH Hypervisor`) and how it uses advanced baseband processing algorithms to slice and abstract the RRH into multiple vRRHs.

We closed this chapter exploring the baseband functions of a LTE-A BBU as an example to illustrate how a fine-grained vBBU can be instantiated by aggregating a set of baseband processing VNFs to implement this access technology.

6 EXPERIMENTAL EVALUATION

In this chapter, we show the evaluation of the prototype described previously. Our goal is to validate the RRH and BBU virtualization capabilities of AIRTIME, with regards to performance, isolation, and scalability. Furthermore, we prove that AIRTIME addresses the three important challenges highlighted in this thesis: vRAN vRAN programmability, vRAN flexibility, and vRAN scalability. In Section 6.2, we evaluate the performance of the `BBU Hypervisor` in terms of instantiation and migration time, fronthaul requirements, and throughput and latency experienced by the vRAN end-users. Afterward, in Section 6.3, we evaluate the performance of HyDRA in terms of signal modifications introduced due to the multiplexing operation (isolation), and processing and memory footprint (scalability). Finally, in Section 6.4, we present a qualitative comparison of AIRTIME with other state-of-the-art proposals.

6.1 Experimental Setup

The experimental evaluation setup is illustrated in Figure 6.1. Acting as CU and DU data centers, we have two laptops with an Intel i5-6440HQ processor, 8 GBs of memory, using the operational system Ubuntu 18.04 with the latest updates installed. These laptops are connected to a Dell SDN switch model S4048T-ON, which acts as the fronthaul network. An Ettus Universal Software Radio Peripheral (USRP) model B210 operates as a RRH. This USRP can tune in any frequency from 70 MHz to 6000 MHz, with a maximum instantaneous bandwidth of 56 MHz. In practice, the maximum instantaneous bandwidth represents the bandwidth available to HyDRA to create vRRHs. More precisely, the bandwidth of all vRRHs multiplexed in the same USRP cannot exceed this value, and all vRRHs' channels must be located inside the instantaneous bandwidth range.

We have installed LXC in CU and DU data centers, so both can run the baseband processing VNFs. We selected the laptop acting as CU to install OSM and OpenStack, so it can assume the responsibilities of the `BBU Hypervisor`, managing all baseband processing VNFs in the CU and the DU data center. Two fine-grained vBBU NSs are installed in OSM. The first one is an LTE-A vBBU that communicates with mobile subscribers. This vBBU is built with the aggregation of three baseband processing VNFs, namely the low-PHY VNF, the high-PHY VNF, and the MAC VNF. The second is a NB-IoT vBBU that comprises only one baseband processing VNF, which implements the entire RAT to communicate with IoT sensors. The `RRH Hypervisor`, *i.e.*, HyDRA, is installed only in the DU data center. We emphasize that our experimental setup is restricted to the RAN, thus not comprising the CN of the SPs. In this case, we do not have a full stack for LTE and NarrowBand-IoT because we aim to validate the RRH and BBU virtualization capabilities of AIRTIME. Moreover, the CU and DU are separated from each other by 10 meters and connected through a 1 Gbps Ethernet link with an average network

latency of 4 ms. Table 6.1 summarizes the main radio parameters for the LTE-A and NB-IoT vBBUs, as well as for HyDRA.

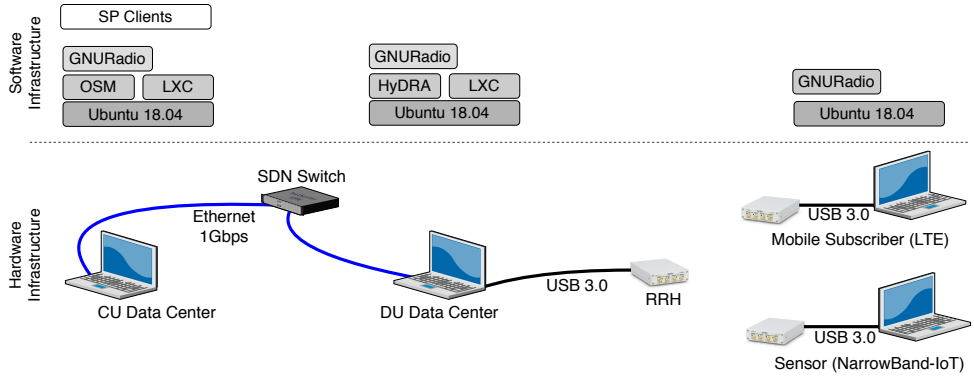


Figure 6.1 – Experimental setup

Finally, we have two mobile users for LTE-A and NB-IoT RATs. Each user comprises an Ettus USRP B200 and a laptop running Ubuntu 18.04 with the latest updates and same hardware configuration as the DU. The distance between the RRH and the user's antenna is precisely 5 meters. The users do not run the `BBU` and `RRH Hypervisors`; our solution is designed to work with legacy end-user devices that do not need to be aware of the virtualization occurring in the network.

We envisioned a scenario whereby an "MBBC SP" and an "IoT SP" interface with OSM to request the creation of vRANs (or vBBUs and vRRHs within a particular vRAN) based on their particular application requirements. The `vRAN Controller` of each SP interacts with the OSM to perform the following changes, based on timestamps:

- Step ① (start at t = 0s): MBBC SP requests the creation of a vRAN with an LTE-A coverage. All three baseband processing VNFs of the LTE-A vBBU are instantiated in the CU data center.
- Step ② (start at t = 0s): The IoT SP requests the creation of a vRAN with a NarrowBand-IoT coverage. The baseband processing VNF for this RAT is instantiated in CU.
- Step ③ (start at t = 100s): The low-PHY baseband processing VNF of the LTE-A vBBU is moved from CU to DU.
- Step ④ (start at t = 200s): The high-PHY baseband processing VNF of the LTE-A vBBU is moved from CU to DU.
- Step ⑤ (start at t = 200s): The NarrowBand-IoT baseband processing VNF is moved from CU to DU.
- Step ⑥ (start at t = 300s): The MAC VNF of the LTE-A vBBU is moved from CU to DU.

This experimental setup also shows that AIRTIME addresses two important challenges of future mobile networks: (*i*) programmability, by having SPs with vRANs tailored to their particular service requirements, *i.e.*, a customized RAT, and (*ii*) flexibility, as SPs can request the spectrum and baseband processing resources tailored to the service requirements, and (*iii*) scalability, by having the baseband processing VNFs being migrated to increase resource usage efficiency. However, it is expected that in a real-life scenario, the `vRAN Controller` acts based on events occurring in the vRAN, instead of working with the pre-determined timestamps of our experimental setup.

Table 6.1 – MBB SP, IoT SP, and HyDRA configurations

SPs	vBBU	VNF(s)	Parameters
MBB	LTE-A (GNU Radio based)	VNF 1: Low-PHY VNF 2: High-PHY VNF 3: MAC	**CF**: 947 MHz, **BW**: 1.4 MHz, **FFT**:128, **CP**: 7 symbols (short), **MOD**:QPSK
IoT	NB-IoT (GNURadio based)	VNF 1: IoT (PHY & MAC)	**CF**: 951MHz, **BW**: 200 KHz, **FFT**:64, **CP**: 7 symbols (short), **MOD**:BPSK
–	–	HyDRA	**CF TX/RX**: 950 MHz, **BW TX/RX**: 8 MHz, **FFT M TX/RX**: 4096

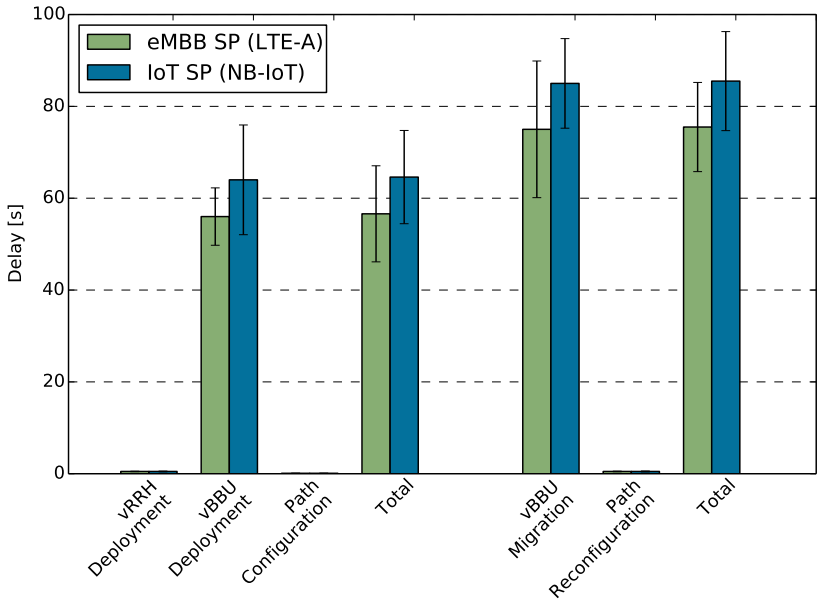
6.2 BBU Hypervisor: Instantiation and Migration of Fine-Grained vBBU

In this section, we show the benefits and impact of the `BBU Hypervisor` through two use cases. First, we evaluate the capabilities of the `BBU Hypervisor` to best fit the available fronthaul network bandwidth by exploring different fine-grained vBBU distribution options. Finally, we evaluate the impact of different fine-grained vBBU distribution options on the latency to transport IQ samples between the CU and DU data centers.

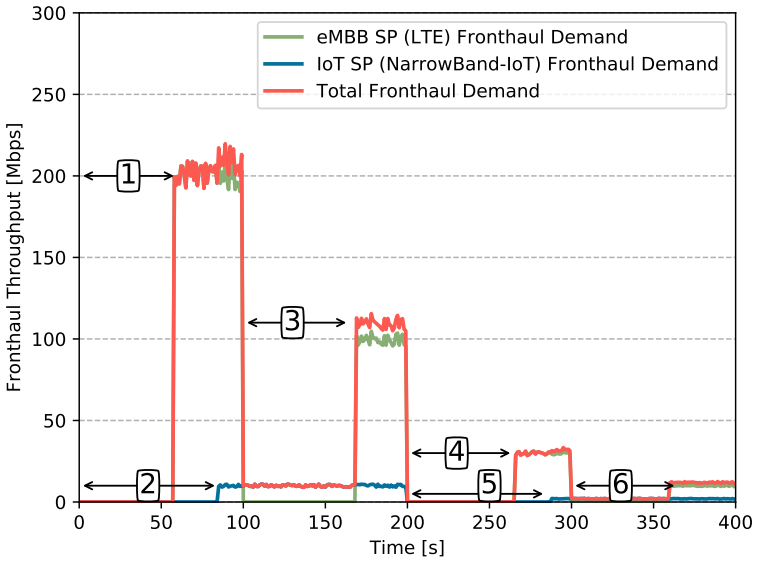
We split our performance analysis in two categories: (*i*) infrastructure level (subsection 6.2.1), and (*ii*) service level (Subsection 6.2.2). At the infrastructure level, we are interested in obtaining insights about the time required to create a vRAN and the fronthaul throughput required between the CU and DU for different baseband processing VNF distributions for this vRAN. At the service level, we analyze the throughput and delay experienced by the SPs to communicate with end-users.

6.2.1 Infrastructure level performance

The deployment of the LTE-A and NB-IoT vRANs start in time t = 0s. Figure 6.2(a) provides insights regarding the time taken, considering the three main operations performed: vRRH deployment, vBBU deployment, and path configuration. These results are shown with a 95% confidence interval. We can see that deploying vBBUs is a time demanding operation, taking up to 60s for LTE-A and 85s for NB-IoT. The vRRH deployment and path configuration operations take less than 1s, thus accounting for only a fraction of the total time required to



(a) Average time required to perform vRAN deployment and migration.



(b) Fronthaul throughput required for different VNF distributions

Figure 6.2 – Infrastructure level evaluation of AIRTIME

deploy the vRANs. In the same figure, we also show the average time needed for migrating the fine-grained vBBUs, together with the time required for the path reconfiguration. We can see that fine-grained vBBU migration takes a longer time than its initial deployment. This behavior occurs because the migration requires the additional transfer of the baseband processing VNF state, *i.e.*, processes running, files opened, and network connections.

The throughput required for each vRAN between the CU and the DU is shown in Figure 6.2(b) (the values shown here were obtained in only one of our experiment executions). The

circled numbers indicate when the steps described in the experimental setup is completed. From time t = 0s until Step ① completes, we can see that no traffic goes through the fronthaul, followed by a spike in the traffic when the LTE IQ traffic starts (approximately 200 Mbps). When both vBBUs are running in the CU (after Step ② completion), the IQ transfer requires approximately 230 Mbps. The migration of the low-PHY LTE VNF starts at t = 100s, which reflects in the LTE IQ traffic halting until it was completed at t = 160s (Step ③). The same behavior can be observed in the subsequent migration steps. From this evaluation, we can observe that offloading the baseband processing VNFs to DU supports the adoption of fronthaul links with low bandwidth capacity.

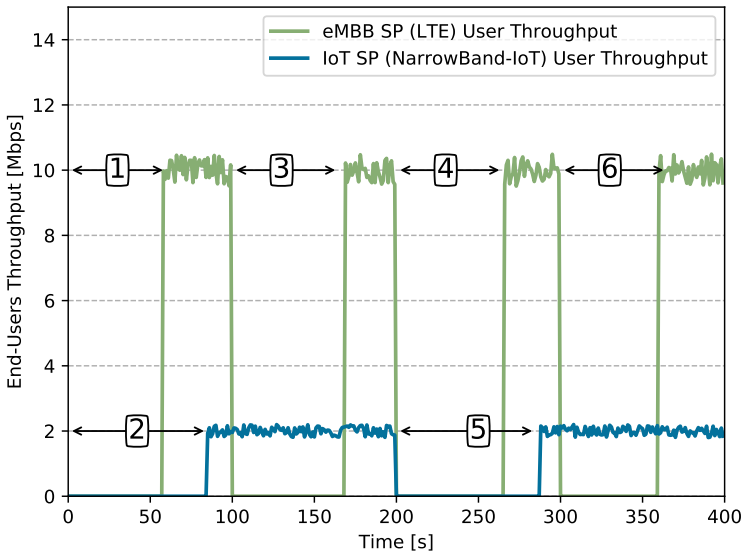
6.2.2 Service level performance

Next, we look at the throughput and latency experienced by end-users, shown in Figure 6.3 (these values were obtained from the same experiment execution mentioned previously). The throughput in Figure 6.3(a) is almost constant for both vRANs for all baseband processing VNF distributions. Let us take LTE-A mobile subscriber as an example: the throughput is constant when all baseband processing VNFs are running in CU (from Step ① completion to t = 100s) and when it everything is running in DU (from Step ⑥ completion to t = 400s). This behavior is because the bottleneck is not the fronthaul link but rather the LTE-A and NB-IoT RATs, which with the configuration we used can achieve only 10 Mbps and 2 Mbps, respectively.

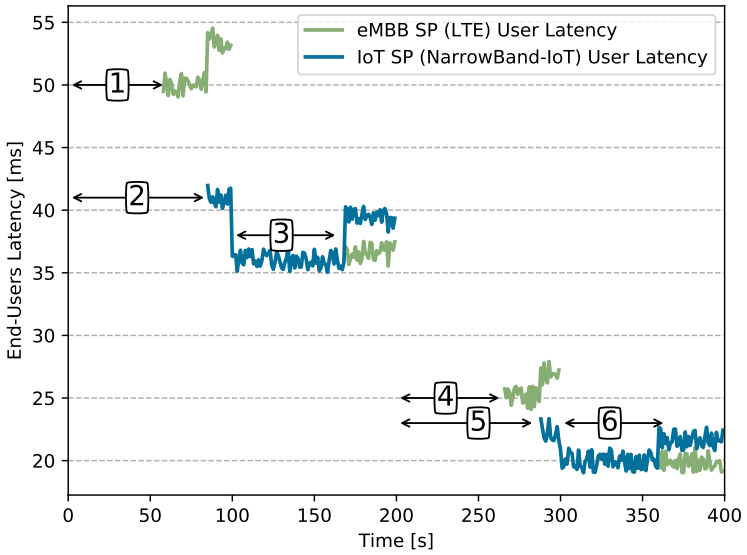
The latency experienced by the end-users is shown in Figure 6.3(b). Migrating the baseband processing VNFs to DU significantly reduces the latency. In this case, let us take the IoT sensor as an example: we have a latency of approximately 48 ms when the unique baseband processing VNF is running in CU together with all LTE-A VNFs. This latency drops significantly when LTE-A vBBU stops completely (at t = 100s) and then increases again when LTE-A vBBU restarts in step ③ (but now with the low-PHY VNF executing in DU). This evaluation also shows another benefit of offloading the baseband processing VNFs to DU: it can enable URLLC vRANs due to the proximity with end-users. However, the caveat when the baseband processing VNFs are moved from CU to DU is the loss of advanced LS-CMA mechanisms, which we do not explore in AIRTIME's prototype due to their high implementation complexity. We do not make further explorations of this trade-off in AIRTIME's prototype due to its high implementation complexity.

6.3 HyDRA: Multiplexing of LTE-A and NB-IoT

In this section, we analyze the capabilities of HyDRA in multiplexing the RATs used by the MBB SP and the IoT SP. In Subsection 6.3.1, we evaluate signal degradation that occurs due to the multiplexing operation used by HyDRA. Then, in Subsection 6.3.2, we show the CPU usage of the HyDRA VNF as we increase the number of vRRHs sharing the same physical RRH.



(a) Throughput experienced by end-users



(b) Latency experienced by end-users

Figure 6.3 – Service level evaluation of AIRTIME

6.3.1 Isolation

In this analysis, we are interested in evaluating the remaining challenge in future mobile networks: vRAN isolation, *i.e.*, the isolation between vRRHs. As HyDRA multiplexes the vRRHs in the frequency domain, the best way to measure their isolation is by considering different frequency separations, *i.e.*, guard-bands, among them. More precisely, we measure the SINR at the LTE-A mobile subscriber and the IoT sensor receiver, for different guard-bands

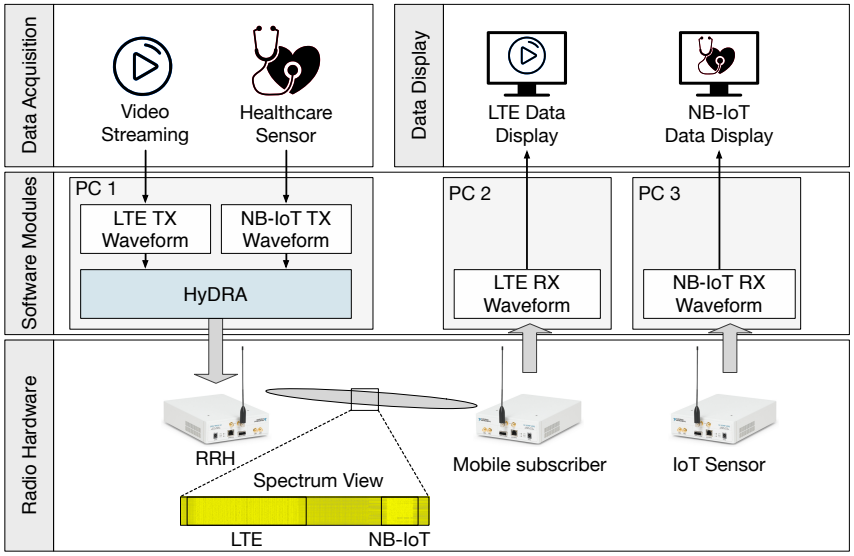
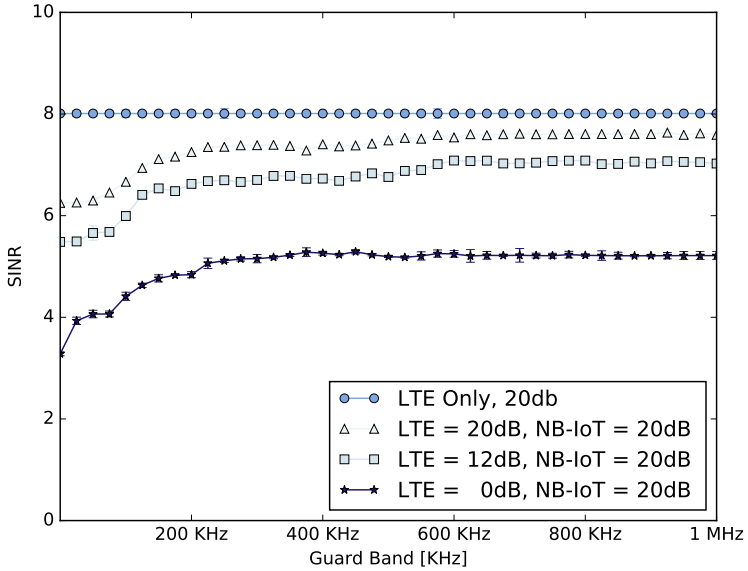


Figure 6.4 – Experimental scenario used to evaluate HyDRA

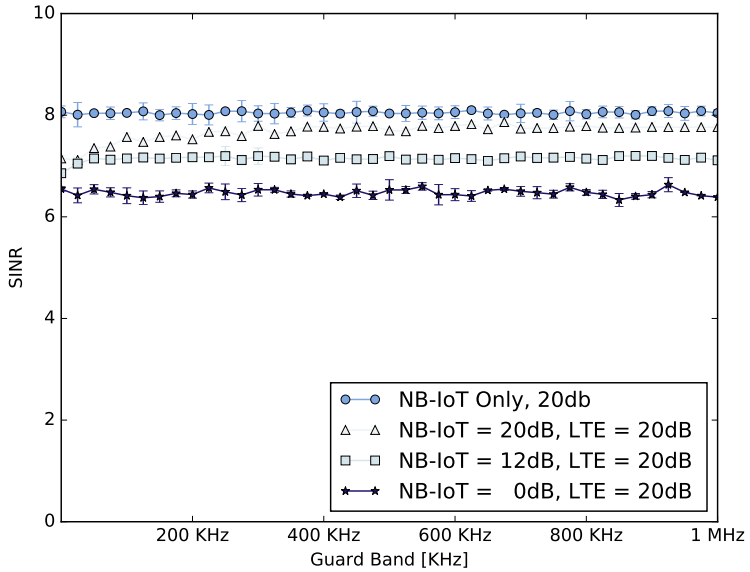
between the vRRHs associated to the LTE-A and NB-IoT vBBUs, respectively. In addition to the guard-band, we are also interested in analyzing the impact of gain differences in the multiplexing process, *i.e.*, when one of the vRRH utilizes a signal gain higher than the others.

The results obtained are shown in Figure 6.5(a) for the LTE-A mobile subscriber and 6.5(b) for the NB-IoT receiver. In Figure 6.5(a), the curve labeled *LTE-A Only* shows the SINR at the mobile subscriber receiver when using the physical RRH without virtualization. We compare against the case where HyDRA is used to simultaneously support LTE-A and NB-IoT vRRHs, in three different configurations; in each one, we reduce the gain of the LTE-A vRRH while maintaining the gain of the NB-IoT constant. We measured the SINR at the mobile subscriber when the LTE-A vRRH is set to 20 dB gain, when the vRRH assigned to the NB-IoT transmitter uses a 12 dB gain, and when the LTE-A uses a gain of 0 dB. In all curves, we can see that guard-bands larger than 200 KHz present a SINR degradation of 1 dB. The same reasoning applies for Figure 6.5(b), which shows the SINR at the NB-IoT receiver. In this case, the NB-IoT signal presented a minimal reduction in the SINR for all signal gains and guard bands.

These results show that HyDRA the isolation of vRRHs has some limitations. HyDRA uses a sequence of FFT and IFFT operations to multiplex the signals of multiple vRRHs. The spectral leakage of these two operations can introduce offset errors in the amplitude or phase of the original signal. To quantify such errors, we evaluate the multiplexing operation of HyDRA considering different sizes for the size of the FFT assigned to each virtual vRRH and the IFFT, *i.e.*, the values of N_{LTE-A}, N_{NB-IoT}, and M. For each value of M, both LTE-A and NB-IoT signals are multiplexed by HyDRA, transmitted over-the-air and then immediately demultiplexed. Then, we calculate the Mean Square Error (MSE) between the original and demultiplexed signals.



(a) SINR observed at the mobile subscriber as a function of the guard-band between the vRRHs



(b) SINR observed at the NarrowBand-IoT receiver as a function of the guard-band between the vRRHs

Figure 6.5 – SINR observed at the end-users for different guard-bands and gains for each vRAN

Table 6.2 shows the MSE values obtained. The row "Without Virtualization" presents the MSE when the IQ samples were transmitted over-the-air (in a channel without any additional source of noise) without being multiplexed by HyDRA. In this case, the MSE is due only to the over-the-air degradation. The remaining rows show the values of the MSE after the signals are multiplexed and transmitted by HyDRA.

Table 6.2 – MSE of the multiplexing process considering different IFFT sizes

	LTE-A vRRH	NB-IoT vRRH
Without Virtualization	2.1 dB	1.1 dB
$M = 1024$	11.2 dB	7.9 dB
$M = 2048$	6.1 dB	4.1 dB
$M = 4096$	3.4 dB	2.2 dB
$M = 8192$	2.4 dB	1.1 dB

The MSE decreases as the number of IFFT bins increases. LTE-A presents a higher MSE due to its original signal being generated from a more complex Orthogonal Frequency-Division Multiplexing (OFDM) configuration with a higher number of carriers and constellation symbols. When HyDRA uses a large enough number of IFFT bins (8192), the resulting MSE, for both the LTE-A and NB-IoT cases, is comparable to the MSE measured in the absence of HyDRA. These results show it is possible to eliminate the multiplexing error by using sufficiently large IFFTs.

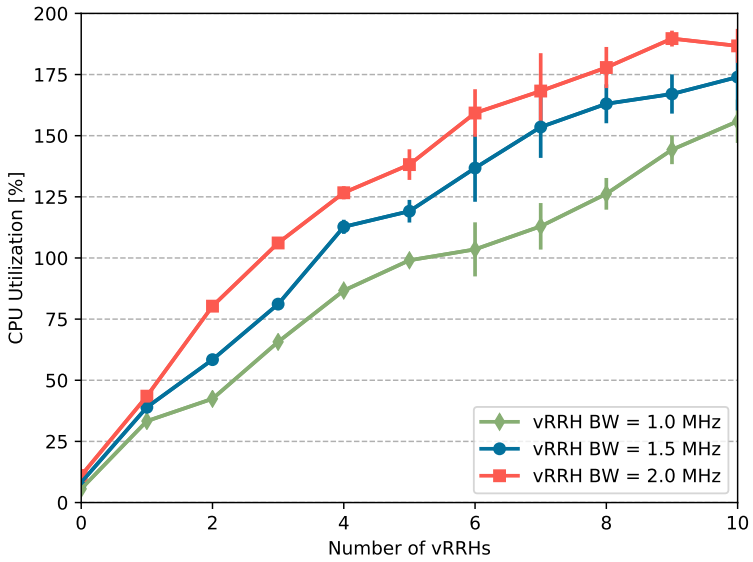
6.3.2 Scalability

In our third experimental evaluation, we quantify how AIRTIME scales in terms of processing and memory requirements, *i.e.*, the increase of resources used as vRANs are deployed. Arguably, the bottleneck in AIRTIME is the `RRH Hypervisor` VNF instance because: (*i*) only one instance is bound to a physical RRH. Thus, this single VNF must cope with the processing demands of multiple vRRHs; and (*ii*) this instance must multiplex the IQ samples from multiple vRRH using only the processing resources of one physical machine (different from fine-grained vBBUs that can be distributed among multiple data centers and physical machines). For these reasons, we focus only on the analysis of the resources used by HyDRA. Moreover, two main parameters impact the performance of HyDRA and that we will explore: (*i*) the number of vRRHs and (*ii*) the bandwidth used by each vRRH.

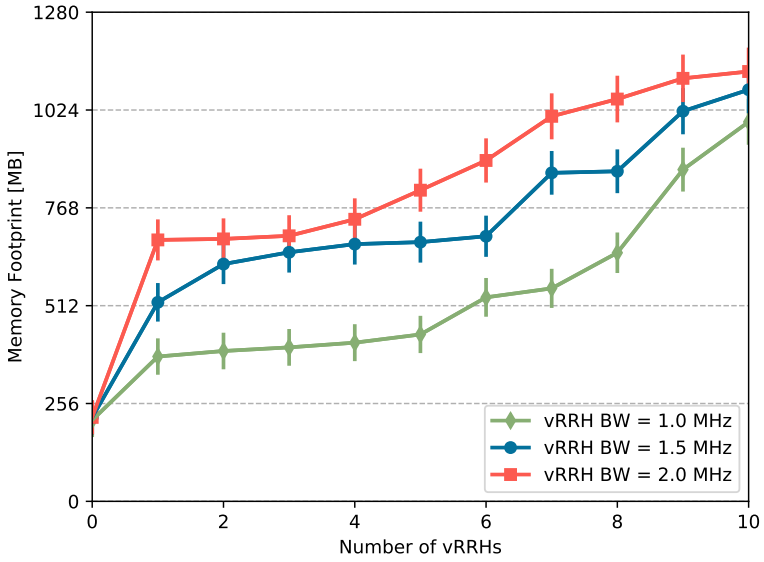
To assess the overhead incurred for the virtualization operations performed by HyDRA, we used a setup where we start from zero vRANs and request the creation of one vRAN up to a total of ten vRANs, all using the same bandwidth [1 MHz, 1.5 MHz, and 2 MHz]. We saturate the traffic of each vRAN with TCP traffic (similar to the previous setup) and measure the CPU usage (using the `mpstat` tool) and memory footprint (using the `vmstat` tool) of the VNF running HyDRA.

In Figure 6.6(a) we show the results for the CPU usage as a percentage of one core in the Intel i5-6440HQ processor, whereas Figure 6.6(b) shows the total memory used in MB. Apart from the initial overhead associated with the creation of the first vRRH, the creation of extra vRRHs incurs in small and almost constant increments in both CPU and memory. The

initial increase is due to HyDRA internal operations that are inactive when there are no vRRH (construction of the spectrum map of vRRHs, FFT, IFFT, and IQ mapping).



(a) HyDRA VNF CPU utilization



(b) HyDRA VNF memory footprint

Figure 6.6 – CPU and memory footprint of the HyDRA VNF for varying number of vRRHs and vRRH bandwidths

The almost constant rise in CPU utilization after the first vRRH is a result of the additional FFT and internal buffers required for each new vRRH. As we deploy vRANs, the CPU utilization increases to the point that the baseband processing VNFs cannot share the physical machine of HyDRA without scaling issues. For example, in the commodity machine used in

our experiment, if we deploy five vRANs with 1 MHz each, the hypervisor requires up to one full core of the processor, leaving only the other core for the baseband processing of the vBBUs. The actual number of vRRHs that can be supported depends both on the resources available for HyDRA and the capabilities of the physical RRH.

6.4 Qualitative Benefits

In this section, we summarize the main qualitative benefits of the virtualization design adopted in AIRTIME.

- **RRH virtualization:** AIRTIME enables the virtualization of the RRH using the novel `RRH Hypervisor`. Our approach for RRH virtualization is state-of-the-art. RRH virtualization is a major step towards future multi-tenancy networks, as being introduced by 3GPP (SAMDANIS; COSTA-PEREZ; SCIANCALEPORE, 2016).
- **Multi-radio access networks:** AIRTIME simplifies the integration and operation of multi-RATs on top of one physical RAN infrastructure. In current centralized baseband architectures, a RRHs is mapped to only one (v)BBU and, therefore, can only operate one access technology. AIRTIME enables RRHs to connect to multiple vBBUs and operate in any technology simply by creating a new vRRH.
- **Flexible multi-tenant access networks:** Current multi-tenant access networks are restricted by the underlying RAT of the physical RAN. In contrast, AIRTIME enables multiple SPs to run a full-blown vRAN tailored to their service requirements on top of the physical RAN.
- **CAPEX reduction:** AIRTIME can run on top of the infrastructure of centralized baseband architectures, while at the same time enabling multiple vRANs to share the same RAN. As a consequence, SPs can deploy tailored RATs simply by creating a new vRRH-vBBU pair and multiplex it through HyDRA.

6.5 Summary

In this chapter, we have shown the experimental results obtained using the prototype of AIRTIME. We measured the performance of the `BBU Hypervisor` and `RRH Hypervisor` in a 5G-like scenario in which a "MBB SP" and a "IoT SP" request the creation of vRANs tailored to their service requirements. For the `BBU Hypervisor`, we measured its performance in terms of the time required to complete the vBBU instantiation and migration, as well as the fronthaul network requirements and delay and throughput experienced by the SPs. For the `RRH Hypervisor`, we measured its capabilities to isolate multiple vRRHs and its scalability in terms of CPU and memory usage. With the evaluation conducted in this chapter, we have

shown that our prototype can support the deployment of multiple heterogeneous vRAN, satisfying the main requisites for future mobile networks: programmability, flexibility, and scalability.

7 MATHEMATICAL ANALYSIS AND EMULATION

In this chapter, we show additional evaluations of AIRTIME. Different from the previous experimental evaluation, here we focus on mathematical analysis and emulation of scenario with a constrained fronthaul. In Section 7.1, we calculate the fronthaul requirements for a LTE-A vBBU for different baseband processing VNF distribution options. In Section 7.2 we use srsLTE to emulate an LTE-A vBBU and measure the CPU usage for each baseband processing VNF. Afterward, in Section 7.3, we emulate AIRTIME with a constrained and shared fronthaul link to measure the latency experienced by vBBUs.

7.1 Mathematical Analysis of Bandwidth Requirements

Contrary to traditional centralized baseband architectures, fine-grained vBBU allows the flexible distribution of baseband functions according to fronthaul link constraints or data center requirements. We exploit the bandwidth requirements of the five most common split options between the CU and DU data centers considering possible baseband processing VNFs in a LTE-A vBBU, as illustrated in Figure 7.1.

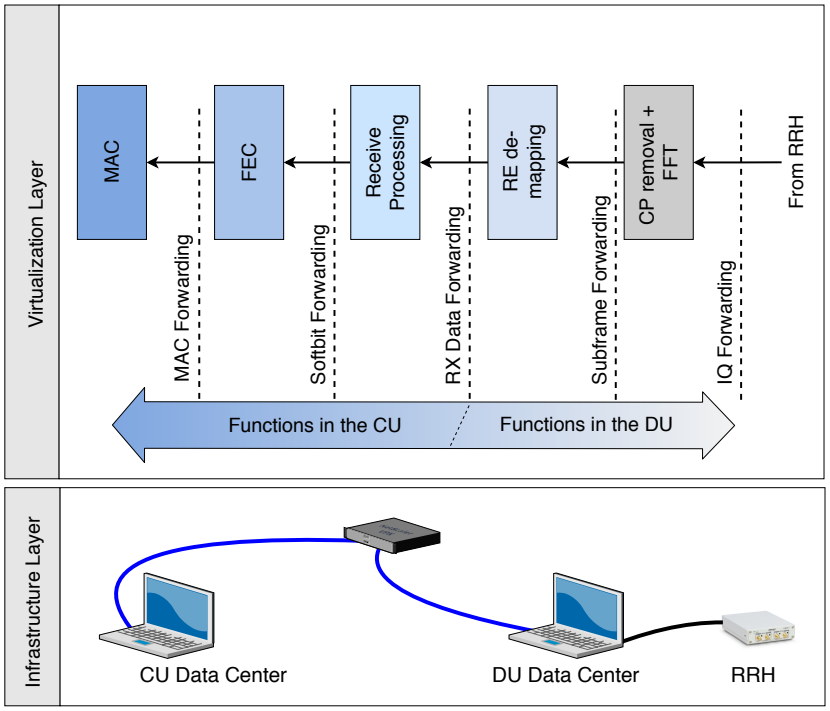


Figure 7.1 – Fine-grained split options for a LTE-A vBBU

We assume that the CU data center is connected to one vRRH using a sampling rate (f_S) of 30.72 MHz (for a 20 MHz channel) with a Multiple-Input Multiple-Output (MIMO) configuration (N_R) of 2x2 and oversampling factor (N_O) of 2. In this configuration, the LTE-A vBBU uses 1.200 subcarriers (N_{SC}) and a symbol duration (T_S) of 66μs. We considered a vRRH

with 10%, 50%, and 100% utilization rate of REs (η), representing different levels of mobile subscriber load. The analog signal to IQ samples conversion (N_Q) is set to 10 bits/sample. Table 7.1 summarizes the parameters and values used for this analysis and the emulation in the next subsection.

Symbol	Description	Value Used	Impacts in
f_S	Sampling Rate	30.72 MHz	Bandwidth
N_R	Number of Antennas	2	Bandwidth
N_O	Oversampling Factor	2	Bandwidth
N_{SC}	Number of Used Subcarriers	1.200	Bandwidth
T_S	Symbol Duration	66.6 μs	Bandwidth
η	Fraction of RE used	[0.1, 0.5, 1.0]	Bandwidth
N_Q	Quantization Bits per IQ	10	Bandwidth
F_{BW}	Fronthaul Capacity	10 Gbps	Latency
F_L	Distance CU-DU	15 Km	Latency

Table 7.1 – Parameters used in the analytical and simulated scenarios

The bar chart in Figure 7.2 shows the required fronthaul bandwidth for each fine-grained vBBU split option. In IQ Forwarding, samples are transported over the fronthaul to the CU data center, which centralizes all baseband processing VNFs. The fronthaul data rate required in this distribution is fixed. Thus mobile operators can determine beforehand whether it can handle the traffic of a vRRH. The main benefit of this distribution is that almost no digital processing is required at the DU data center. Moreover, this distribution eases the adoption of LS-CMA because of the centralization of all IQ samples. This option is attractive only in the cases where the DU data center is already overloaded with other baseband processing VNFs or if the cost of fronthaul transport is low. The fronthaul demand when using this distribution option is given by:

$$\begin{aligned} B_{FH}^{IQ} =& N_O \cdot f_S \cdot 2 \cdot N_Q \cdot N_R \\ & 2 \cdot 30.72\text{MHz} \cdot 2 \cdot 10\text{bits} \cdot 2 = 2.46\text{Gbps} \end{aligned}$$

In Subframe Forwarding, the baseband processing VNF implementing the CP Removal and FFT is moved to the DU data center. In this case, only the IQ samples of useful subcarriers are transported over the fronthaul, representing roughly 60% of the total subcarriers in our configuration. Eliminating this overhead reduces the bandwidth required to 720 Mbps. Subframe Forwarding is attractive when 100% of the wireless resources are being utilized because the fronthaul data rate required is always the same, while at the same time enabling LS-CMA mechanisms. Also, the baseband processing workload does not depend on the actual load of the vRRH. The fronthaul demand when using this distribution option is given by:

$$B_{FH}^{SF} = N_{SC} \cdot T_S^{-1} \cdot 2 \cdot N_Q \cdot N_R$$
$$1.200 \cdot (66\mu\text{s})^{-1} \cdot 2 \cdot 10\text{bits} \cdot 2 = 720\text{Mpbs}$$

In RX Data Forwarding, the baseband processing VNF implementing the RE de-mapper is moved closer to the (v)RRH. In this distribution option, the CU data center receives the IQ samples of REs allocated to mobile subscribers, *i.e.*, 10% of 720 Mbps if 10% of REs are allocated (which is something that can change in each LTE-A frame). Because of this, the fronthaul data rate required is not constant. Based on fine-grained vBBU this distribution option can be selected on-the-fly when less than 50% of the wireless resources of a vRRH are being allocated, significantly reducing the overhead in the fronthaul network. This fronthaul demand when using this distribution can be calculated using the factor of REs allocated and B_{FH}^{SF}:

$$B_{FH}^{RX} = B_{FH}^{SF} \cdot \eta = 720\text{Mbps} \cdot [0.1, 0.5, 1.0] = [72, 360, 720]\text{Mbps}$$

In SoftBit Forwarding, the DU data center executes all VNFs required to recover bits from the received radio signal, which includes both user data and higher layer control data, such as MAC headers. This distribution option reduces the fronthaul data rate required to a fraction of the standard IQ Forwarding adopted in atomic vBBUs, but at the cost of disabling LS-CMA at the regional data center. However, fine-grained vBBU allows LS-CMA to be performed between all vRRHs connected to the same DU data center. The fronthaul required is given by:

$$B_{FH}^{SB} = B_{FH}^{RX} / N_R = [72, 360, 720]\text{Mbps}/2 = [36, 180, 360]\text{Mbps}$$

MAC Forwarding is the approach used in 4G mobile networks, in which MAC packet data units are transported over the fronthaul. Although the burden of the fronthaul is significantly reduced, the processing demand at the DU data center becomes the major bottleneck, as the baseband processing VNF implementing the FEC requires considerable computational capacity. This bandwidth required for this split is given by:

$$B_{FH}^{MAC} = N_{SC} \cdot T_S^{-1} \cdot \eta \cdot S$$
$$1.200 \cdot (66\mu s)^{-1} \cdot [0.1, 0.5, 1] \cdot 3\text{bit/cu} = [5.4, 27, 54]\text{Mbps}$$

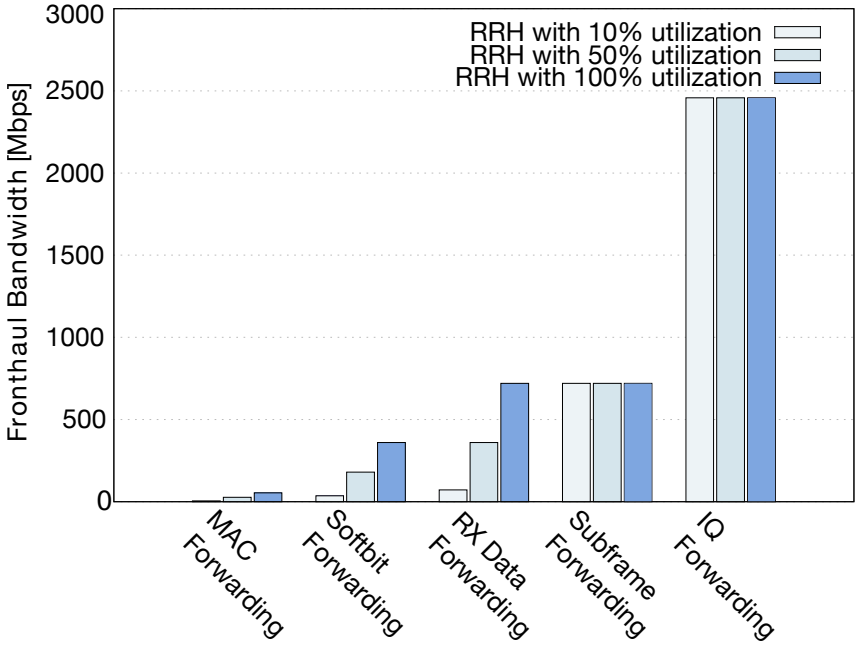


Figure 7.2 – Fronthaul bandwidth for each fine-grained vBBU distribution option

7.2 vBBU Distribution Options and Processing Requirements

We also wanted to understand the processing requirements of each baseband processing VNF of a LTE-A vBBU. For this, we run the widely used srsLTE software, which is an implementation of the LTE-A base station. We measure the CPU usage in an Intel Core i5-4250U2 1.3 GHz in the w-iLab-t testbed. We highlight that srsLTE implements the LTE-A vBBU as a single piece of software, *i.e.*, atomic vBBU. However, it runs multiple threads, each of which based on a particular task of the baseband processing. We used a simple bash script to measure the CPU usage of each thread. To obtain the CPU usage for each of the five baseband processing VNFs shown in Figure 7.1, we grouped the threads by the operations performed in each one of them in such a way that matches the operations in the VNFs. However, this grouping is not exact, as srsLTE were not designed to match the operations in each split options. Table 7.2 show the obtained results for each baseband processing VNF and for all six standardized LTE channel bandwidths. First, we can note that the same baseband processing VNF, *e.g.*, CP removal + FFT, requires more processing time as the channel bandwidth increases. The CPU usage increase is a side effect of higher channel bandwidths due to the high number of digitized IQ samples being processed.

We highlight that increasing the channel bandwidth does not correlate to the same increase in the CPU usage, *e.g.*, doubling the channel bandwidth does not incur double CPU usage. This happens because several baseband processing operations take advantage of modern processor instructions, such as SIMD. Moreover, we can see that the FEC and Receive Processing are by far the most CPU intensive functions (with the first being slightly more intensive than the

		LTE Channel Bandwidth					
		1.4 MHZ	**3 MHZ**	**5 MHZ**	**10 MHz**	**15 MHz**	**20 MHz**
VNFC	**CP Removal + FFT**	8.9	10.7	10.7	10.9	13.4	16.1
	RE demapping	3.4	4.7	8.7	13.1	15.4	16.1
	Receive Processing	10.2	12.0	18	32.8	40.3	46.9
	FEC	10.2	11.4	16.7	32.1	41.6	47.6
	MAC	6.8	6.7	8.0	8.5	10.2	12.8

Table 7.2 – CPU usage for each fine-grained LTE-A vBBU function

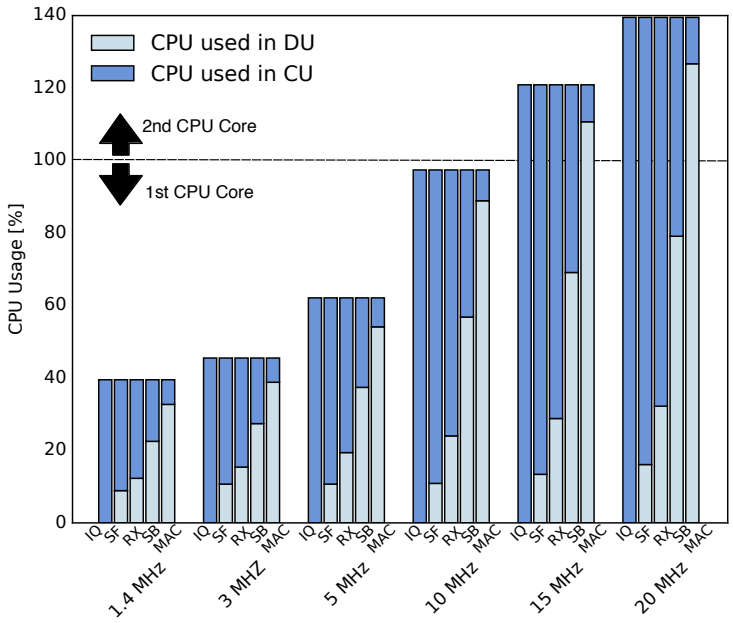


Figure 7.3 – Total CPU usage in CU and DU data center for each distribution options

latter). Both functions together require up to half of the total CPU consumed by the vBBU. Finally, the CPU usage is reduced drastically as the vBBU functions shift from the low physical operations, which encompass all operations except the MAC, to the higher MAC layer.

We further explore the fine-grained vBBU possibilities by measuring the CPU usage in the CU and DU data centers, as shown in Figure 7.3. We highlight the fact that usage below 100% uses only one processing core of the CPU, whereas above it uses two CPU cores. As expected, all CPU usage is concentrated in the CU data center when adopting the IQ Forwarding option, but at the cost of substantial fronthaul bandwidth, as we mentioned earlier. For comparison, adopting the Subframe Forwarding option moves only a fraction of the total processing required to the DU, while at the same time reducing the fronthaul demand from 2.46 Gbps to 720 Mbps.

7.3 Latency of fine-grained vBBU distribution using Mininet

We also wanted to understand how distributing the baseband processing VNFs between the CU and DU data centers affects the latency experienced by end-users subscribers. Therefore, we emulated the infrastructure shown in Figure 7.4 in the Mininet network emulator. We varied the number of vBBU and measured the latency given a fronthaul link with capacity limited to 10 Gbps (F_{BW}) and length (F_L) of 15 Km. As the number of vBBU increases, it is expected that the competition for the shared and limited capacity of the fronthaul link increases the overall latency. We leveraged the virtual hosts in Mininet to act as vBBU (and this is the only reason for using Mininet in this analysis) by generating the traffic at the data rates according to the results presented in the previous section, considering a 50% utilization rate, *i.e.*, 2.46 Gbps for IQ Forwarding, 720 Mbps for Subframe Forwarding, 360 Mbps for RX Data Forwarding, 180 Mbps for Softbit Forwarding, and 27 Mbps for MAC Forwarding. The latencies obtained are shown in Figure 7.5.

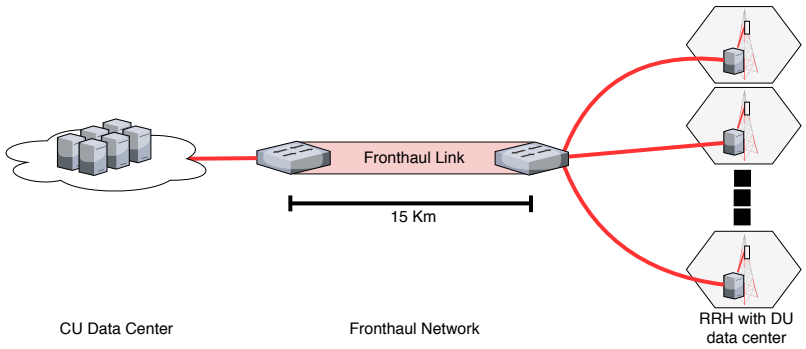


Figure 7.4 – AIRTIME infrastructure with constrained fronthaul

Considering that an LTE-A vBBU needs to generate an ACK/NACK response in 3 ms to stay compliant with the 3GPP LTE-A HARQ timing, we have an estimate of the maximum number of fine-grained vBBUs that can be executed simultaneously if all of them adopt the same distribution option. This number was as follows in our emulated network: 4 for IQ Forwarding, 13 SubFrame Forwarding, 17 for RX Data Forwarding, 33 for SoftBit Forwarding, and 49 for MAC Forwarding. We highlight that the processing at the CU data center must compensate for the latencies in the fronthaul network. For example, 14 vBBUs adopting the SubFrame Forwarding distribution lead to an average latency of 2.77 ms, the CU data center only 0.23 ms to perform the processing to generate the HARQ message.

AIRTIME avoids the drawbacks of centralized baseband architectures by allowing fine-grained vBBU functions to be distributed according to the fronthaul and CU and DU data center constraints. We also demonstrated the benefits and impact of AIRTIME by (*i*) exploring different distribution options to analyze the required fronthaul network bandwidth, and (*ii*) analyzing the latency experienced by VNFs between CU and DU data centers in a fronthaul network with limited bandwidth. However, as fine-grained vBBUs have strict real-time, low-latency, and

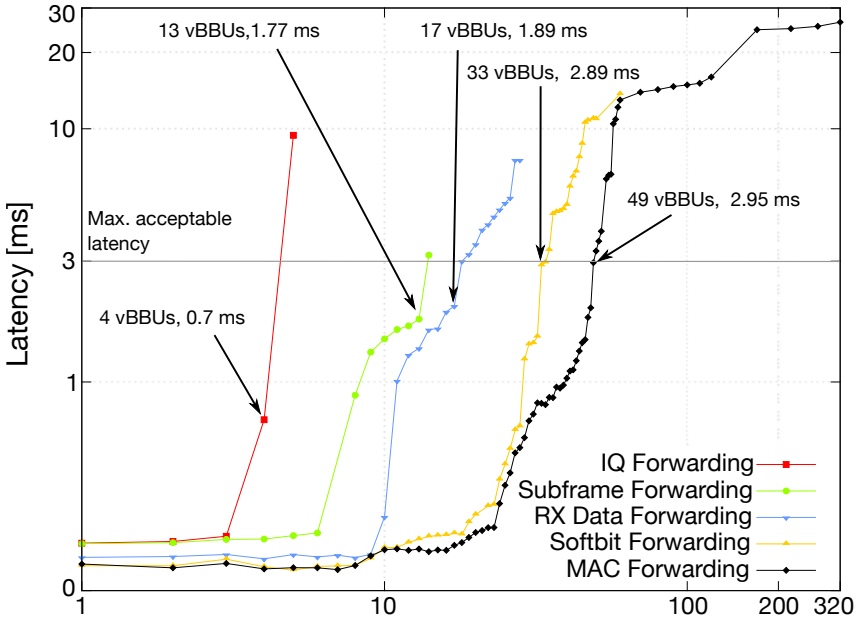


Figure 7.5 – Average latency as a function of the number of fine-grained vBBUs

high-performance requirements, to match the performance of the traditional bare-metal implementation, some challenges must be overcome:

- Advanced algorithms for real-time signal processing in baseband processing VNFs with close-to-bare-metal performance to facilitate the adoption of fine-grained vBBUs in standard data center hardware, *i.e.*, GPPs.
- Efficient and flexible real-time operating systems to achieve a dynamic allocation of physical processing resources for baseband processing VNFs and to ensure processing latency and jitter control.
- Programmable fronthaul network with native support to time-critical packet switching to enable fast and reliable on-the-fly migration of VNFs between data centers.

We expect that AIRTIME can catalyze mobile network innovations in a range of areas, from the introduction of new RATs specialized in specific services, to management of data center and fronthaul resources. Fine-grained BBU virtualization solves the challenges of current centralized baseband architectures while enabling unprecedented control over any aspect of the virtualized RANs.

7.4 Summary

In this chapter, we further evaluate our proposal with the help of mathematical analysis and an emulation software. First, we calculated the fronthaul bandwidth requirements for a LTE-A vBBU with different baseband processing VNFs distributions. Then, we used the fronthaul

bandwidths required for each distribution option to simulate the AIRTIME architecture with a fronthaul network with a constrained link. In this emulation, we were interested in measuring how many vBBUs our architecture can run simultaneously in each distribution option, considering that the maximum acceptable latency is 3 ms. The results show AIRTIME can enable the infrastructure provider to adapt the fine-grained vBBU distribution options according to the fronthaul network requirements. Moreover, similar to the results obtained in our prototype, moving the baseband processing VNFs to the DU (closer to the physical RRH) significantly reduces the latency experienced by the end-users, which is of utmost importance for to adopt low-latency services, such as URLLC.

8 CONCLUSIONS

In this thesis, we discussed the challenges of the baseband centralization architecture that is being considered for the 5G network, *i.e.*, (*i*) a single "one-size-fits-all" access technology, (*ii*) limited coverage area due to latency requirements, and (*iii*) high fronthaul bandwidth requirements to transport digitized signal samples. To address these challenges, we proposed AIRTIME, an architecture that integrates fine-grained BBU virtualization and with RRH virtualization (a radio virtualization layer that enables multiple access technologies to coexist on top of the same antenna). Together, these two solutions enable multiple heterogeneous vRANs coexisting on top of a shared physical infrastructure and allowing SPs to fully customize their slice to widely different service requirements.

Given the challenges identified and the proposal of a solution that integrates virtualization to lowest level mobile network architecture, *i.e.*, the radio device, extensive research, development, and experimentation have been conducted aiming to verify the following hypothesis.

***Hypothesis:* incorporating the concepts of radio virtualization and fine-grained baseband processing virtualization push the boundaries of future mobile networks with increased programmability, flexibility, and scalability.**

The BBU virtualization layer increases the scalability that is lacking in centralized baseband architectures by allowing vBBU functions to be distributed in the processing resources according to the fronthaul and data center constraints. The RRH virtualization layer is as a mechanism to provide flexible and programmable connectivity services in the next generation of mobile networks. The decouple of BBU and RRH adds a layer of indirection. It provides a virtual RRH abstraction to vBBUs, which operates as if it is interfacing with a standard RRH.

AIRTIME integrates both the BBU and the RRH virtualization layer in a seamless architecture. We evaluate a prototype of AIRTIME in a specific 5G-like scenario to demonstrate its capability to enable multi-radio access networks by slicing the physical RAN into multiple virtualized RANs tailored to specific services. We also explored different fine-grained vBBU distribution options to analyze the required fronthaul network bandwidth, latency, and processing resource requirements.

It is now possible to answer the three Research Questions (RQs) associated with the hypothesis that has been posed to guide the research conducted in this thesis. The answers to each question are detailed as follows.

RQ I – Do vRRHs maintain the same transmission quality of its physical counterpart?

Answer: We evaluated the transmission characteristics of physical and virtual RRHs in a 5G-like network. We presented experimental results that explore RRH virtualization performance in terms of guard-bands and SINR. The results show that vRRH multiplexing is feasible with some impact on the transmission quality.

RQ II – Do fine-grained vBBUs maintain the performance required to cope with the latency requirements of 5G?

Answer: The reasoning behind proposing fine-grained vBBU is to adapt the required bandwidth and latency according to the fronthaul characteristics and data center capabilities. Our results show that fine-grained vBBU can significantly reduce the bandwidth and latency required if the baseband functions are moved closer to the physical RRH at the cost of reduced advanced cooperation opportunities; on the other hand, full cooperation can be achieved if the fronthaul is less bandwidth restricted.

RQ III – What are the trade-offs by adopting fine-grained vBBUs when compared to atomic vBBUs?

Answer: Fine-grained vBBUs further enhances the atomic vBBU approach by enabling bandwidth and latency restricted fronthauls; it also increases scalability, as baseband processing functions can be executed in specialized hardware, and flexibility, as the vBBU chain can be configured on-the-fly according high-level service requirements.

Based on the studies conducted, it is possible to identify several open challenges that remain after this thesis. We discuss these challenges in the following subsection, as they can be subject to future work

8.1 Improvements and Open Challenges

- **OpenAirInterface 5G:** The prototype developed in this thesis uses LXC containers running GNURadio flowgraphs to perform the baseband processing. This was done because the author had some expertise with this software stack. However, this approach is not LTE-A or NB-IoT compliant. A much better solution would be to use OpenAirInterface 5G, as it implements the full LTE-A stack, and it is compliant to 3GPP standards. This can be an exciting topic for a master's student.

- **Container optimizations:** The deployment and migration time of the containers in AIRTIME's prototype are higher than expected. One could borrow the optimization techniques from the state-of-the-art container virtualization and apply them to the prototype of this thesis to achieve much better deployment and migration times.

- **Container placement algorithms:** This thesis does not use any container placement algorithm to decide in which data center and machine of the data center the containers should be deployed or migrated. Future research efforts can explore different placement algorithms considering the peculiarities of baseband processing VNFs (some VNFs have require constant data rate in the input and/or generate a constant data rate output, others

can have its input or output data rate estimated based on adjacent VNFs or in the RRH workload) fronthaul and data center capabilities, together with additional infrastructure costs such as electricity (the cost of using processing resources in one region or another).

- **Heterogeneous RATs with HyDRA:** The prototype developed in this thesis uses only OFDM-based RATs, *i.e.*, LTE-A and NB-IoT. Measuring HyDRA's performance, *i.e.*, isolation, memory, CPU usage, with non-OFDM is an interesting final year project for undergrad students.

- **Latency introduced by HyDRA:** This thesis did not present any results regarding the latency introduced by the hypervisors when compared to non-virtualized solutions. Measuring the additional delay introduced by HyDRA is an interesting result and should have been explored in this thesis. This can be measured by creating an end-user application that generates data packets containing a timestamp of when the packet was generated. This packet is then sent the input of a vBBU to be transmitted over-the-air. At the reception side, another end-user application receives this packet and estimates the latency (after going through the vBBU). One can measure the delay without using HyDRA and with different vBBUs, and with HyDRA considering different number of vRRHs and combination of vBBUs. This is an interesting project for a master's student.

- **Automatic Gain-Control for HyDRA:** Our prototype `RRH Hypervisor` requires all vRRH to have a similar transmission gain, *i.e.*, the IQ samples received by all vRRHs. One interesting improvement would be to implement Automatic Gain Control (AGC) so that HyDRA can adjust the gains of all vRRHs automatically.

REFERENCES

3GPP. *Summary of RAN3 status on CUDU split option 2 and option 3, and questions/issues for RAN2*. [S.l.], 2017.

ABDELWAHAB, S. et al. Network function virtualization in 5G. IEEE Communications Magazine, vol. 54, no. 4, p. 84–91, 2016.

ADAMUZ-HINOJOSA, O. et al. Harmonizing 3gpp and nfv description models: Providing customized ran slices in 5g networks. IEEE Vehicular Technology Magazine, vol. 14, no. 4, p. 64–75, Dec 2019.

AKYILDIZ, I. F.; WANG, P.; LIN, S.-C. SoftAir: A software defined networking architecture for 5G wireless systems. Computer Networks, Elsevier, vol. 85, p. 1–18, Jul 2015.

BARTELT, J. et al. Fronthaul and backhaul requirements of flexibly centralized radio access networks. IEEE Wireless Communications, vol. 22, no. 5, p. 105–111, Oct 2015.

BELBEKKOUCHE, A.; HASAN, M. M.; KARMOUCH, A. Resource Discovery and Allocation in Network Virtualization. IEEE Communications Surveys & Tutorials, vol. 14, no. 4, p. 1114–1128, 2012.

BHANAGE, G. et al. VNTS: A Virtual Network Traffic Shaper for Air Time Fairness in 802.16e Systems. In: EEE International Conference on Communications, [S.l.]: IEEE, 2010. p. 1–6.

BHANAGE, G. et al. Virtual basestation: Architecture for an open shared wimax framework. In: Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures - VISA '10, New York, New York, USA: ACM Press, 2010. p. 1.

BHANAGE, G. et al. Evaluation of openvz based wireless testbed virtualization. Technical Report WINLAB-TR-331, Rutgers University, 2008.

BHANAGE, G. et al. SplitAP: Leveraging Wireless Network Virtualization for Flexible Sharing of WLANs. In: EEE Global Telecommunications Conference, [S.l.]: IEEE, 2010. p. 1–6.

CHECKO, A. et al. Cloud RAN for Mobile Networks - Technology Overview. IEEE Communications Surveys & Tutorials, vol. 17, no. 1, p. 405–426, 2015.

CHOWDHURY, N.; BOUTABA, R. Network Virtualization: State of the Art and Research Challenges. IEEE Communications Magazine, vol. 47, no. 7, p. 20–26, Jul 2009.

CISCO. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021. 2017. Available from Internet: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf.

DOTSCH, U. et al. Quantitative Analysis of Split Base Station Processing and Determination of Advantageous Architectures for LTE. Bell Labs Technical Journal, vol. 18, no. 1, p. 105–128, Jun 2013.

FONT-BACH, O. et al. When SDR meets a 5G candidate waveform: Agile use of fragmented spectrum and interference protection in PMR networks. IEEE Wireless Communications, vol. 22, no. 6, p. 56–66, Dec 2015.

FOUKAS, X.; MARINA, M. K.; KONTOVASILIS, K. Orion: RAN Slicing for a Flexible and Cost-Effective Multi-Service Mobile Network Architecture. In: International Conference on Mobile Computing and Networking, [S.l.: s.n.], 2017. p. 127–140.

FOUKAS, X. et al. FlexRAN: A Flexible and Programmable Platform for Software Defined Radio Access Networks. In: Emerging Networking EXperiments and Technologies, [S.l.: s.n.], 2016. p. 427–441.

FOUKAS, X. et al. Network Slicing in 5G: Survey and Challenges. IEEE Communications Magazine, vol. 55, no. 5, p. 94–100, 2017. ISSN 0163-6804.

GESBERT, D. et al. Multi-Cell MIMO Cooperative Networks: A New Look at Interference. IEEE Journal on Selected Areas in Communications, vol. 28, no. 9, p. 1380–1408, Dec 2010.

GUDIPATI, A. et al. SoftRAN: Software Defined Radio Access Network. In: ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, New York, USA: ACM Press, 2013. p. 25.

HUANG, B.-S.; CHIANG, Y.-H.; LIAO, W. Remote radio head (RRH) deployment in flexible C-RAN under limited fronthaul capacity. IEEE International Conference on Communications (ICC), IEEE, p. 1–6, May 2017.

KIST, M. et al. HyDRA: A hypervisor for software defined radios to enable radio virtualization in mobile networks. In: IEEE Conference on Computer Communications Workshops, [S.l.: s.n.], 2017. p. 960–961.

KIST, M. et al. Flexible Fine-Grained Baseband Processing with Network Functions Virtualization: Benefits and Impacts. Elsevier Computer Networks, vol. 151, p. 158–165, 2019.

KOKKU, R. et al. NVS: A Substrate for Virtualizing Wireless Resources in Cellular Networks. IEEE/ACM Transactions on Networking, vol. 20, no. 5, p. 1333–1346, Oct 2012.

KOKKU, R. et al. CellSlice: Cellular wireless resource slicing for active RAN sharing. International Conference on Communication Systems and Networks, IEEE, p. 1–10, Jan 2013.

LARSEN, L. M.; CHECKO, A.; CHRISTIANSEN, H. L. A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks. IEEE Communications Surveys and Tutorials, IEEE, vol. 21, no. 1, p. 146–172, 2019.

LEIVADEAS, A. et al. An Architecture for Virtual Network Embedding in Wireless Systems. International Symposium on Network Cloud Computing and Applications, IEEE, p. 62–68, Nov 2011.

LIANG, C.; YU, F. R. Wireless Network Virtualization: A Survey, Some Research Issues and Challenges. IEEE Communications Surveys & Tutorials, vol. 17, no. 1, p. 358–380, 2015.

LIU, J. et al. CONCERT: a Cloud-based Architecture for Next-generation Cellular Systems. IEEE Wireless Communications, vol. 21, no. 6, p. 14–22, Dec 2014.

LIU, J. et al. On the statistical multiplexing gain of virtual base station pools. EEE Global Communications Conference, IEEE, p. 2283–2288, Dec 2014.

LU, X. et al. An elastic resource allocation algorithm enabling wireless network virtualization. Wireless Communications and Mobile Computing, p. 295–308, Dec 2012.

LU, X.; YANG, K.; ZHANG, H. An Elastic Sub-carrier and Power Allocation Algorithm Enabling Wireless Network Virtualization. Wireless Personal Communications, vol. 75, no. 4, p. 1827–1849, Apr 2014.

MAROTTA, M. A. et al. Characterizing the Relation Between Processing Power and Distance Between BBU and RRH in a Cloud RAN. IEEE Wireless Communications Letters, vol. 7, no. 3, p. 472–475, 2018. ISSN 2162-2337.

NAGAI, T.; SHIGENO, H. A Framework of AP Aggregation Using Virtualization for High Density WLANs. International Conference on Intelligent Networking and Collaborative Systems, IEEE, p. 350–355, Nov 2011.

NAKAUCHI, K.; SHOJI, Y.; NISHINAGA, N. Airtime-based resource control in wireless LANs for wireless network virtualization. In: 2012 Fourth International Conference on Ubiquitous and Future Networks (ICUFN), [S.l.]: IEEE, 2012. p. 166–169.

OSSEIRAN, A. et al. Scenarios for 5G mobile and wireless communications: the vision of the METIS project. IEEE Communications Magazine, vol. 52, no. 5, p. 26–35, May 2014.

PAPA, A. et al. Optimizing dynamic ran slicing in programmable 5g networks. In: IEEE International Conference on Communications (ICC), [S.l.: s.n.], 2019. p. 1–7.

PAUL, S.; SESHAN, S. Technical Document on Wireless Virtualization. GENI: Global Environment for Network Innovations, Technical Report, 2006.

PENG, M. et al. Heterogeneous Cloud Radio Access Networks: a New Perspective for Enhancing Spectral and Energy Efficiencies. IEEE Wireless Communications, vol. 21, no. 6, p. 126–135, 2014.

PEREZ, S.; CABERO, J. M.; MIGUEL, E. Virtualization of the Wireless Medium: A Simulation-Based Study. IEEE ehicular Technology Conference, IEEE, p. 1–5, Apr 2009.

PHILIP, V. D.; GOURHANT, Y.; ZEGHLACHE, D. OpenFlow as an Architecture for e-NodeB Virtualization. e-Infrastructure and e-Services for Developing Countries, p. 49–63, 2012.

QUINTANA-RAMIREZ, I. et al. The Making of 5G: Building an End-to-End 5G-Enabled System. IEEE Communications Standards Magazine, IEEE, vol. 2, no. 4, p. 88–96, 2019.

REDANA, S. et al. View on 5G Architecture. White paper of the 5G-PPP architecture Working Group, 2016.

ROST, P. et al. Benefits and challenges of virtualization in 5G radio access networks. IEEE Communications Magazine, vol. 53, no. 12, p. 75–82, Dec 2015.

SACHS, J.; BAUCKE, S. Virtual radio: a framework for configurable radio networks. International Conference on Wireless Internet, p. 1–7, 2008.

SAMDANIS, K.; COSTA-PEREZ, X.; SCIANCALEPORE, V. From network sharing to multi-tenancy: The 5G network slice broker. IEEE Communications Magazine, vol. 54, no. 7, p. 32–39, Jul 2016.

SANTOS, J. et al. Towards Enabling RAN as a Service - The Extensible Virtualisation Layer. IEEE International Conference on Communications (ICC), IEEE, Shanghai, China, 2019.

SEXTON, C. et al. 5G: Adaptable Networks Enabled by Versatile Radio Access Technologies. IEEE Communications Surveys & Tutorials, vol. 19, no. 2, p. 688–720, 2017.

SINGHAL, S. et al. Evaluation of UML Based Wireless Network Virtualization. Next Generation Internet Networks, IEEE, p. 223–230, Apr 2008.

SMITH, G. et al. Wireless virtualization on commodity 802.11 hardware. ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, ACM Press, New York, USA, p. 75, 2007.

TALEB, T. et al. EASE: EPC As a Service to Ease Mobile Core Network Deployment Over Cloud. IEEE Network, vol. 29, no. 2, p. 78–88, 2015.

TAN, K. et al. Enable Flexible Spectrum Access with Spectrum Virtualization. IEEE International Symposium on Dynamic Spectrum Access Networks, p. 47–58, Oct 2012.

THYAGATURU, A. S.; ALHARBI, Z.; REISSLEIN, M. R-fft: Function split at ifft/fft in unified lte cran and cable access network. IEEE Transactions on Broadcasting, vol. 64, no. 3, p. 648–665, Sep 2018.

WANG, H. et al. SoftNet: A Software Defined Decentralized Mobile Network Architecture Toward 5G. IEEE Network, vol. 29, no. 2, p. 16–22, 2015.

WANG, X.; KRISHNAMURTHY, P.; TIPPER, D. Wireless network virtualization. International Conference on Computing, Networking and Communications, IEEE, p. 818–822, Jan 2013.

WUBBEN, D. et al. Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN. IEEE Signal Processing Magazine, vol. 31, no. 6, p. 35–44, Nov 2014.

WUBBEN, D. et al. Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN. IEEE Signal Processing Magazine, vol. 31, no. 6, p. 35–44, 2014.

XIA, L. et al. Virtual WiFi: Bring Virtualization from Wired to Wireless. ACM Special Interest Group on Programming Languages (SIGPLAN), vol. 46, no. 7, p. 181, Jul 2011.

YANG, M. et al. OpenRAN: a software-defined ran architecture via virtualization. ACM Special Interest Group on Data Communication (SIGCOMM), ACM Press, New York, USA, p. 549, 2013.

ZAIDI, Z.; FRIDERIKOS, V.; IMRAN, M. A. Future ran architecture: Sd-ran through a general-purpose processing platform. IEEE Vehicular Technology Magazine, vol. 10, no. 1, p. 52–60, March 2015.

ZAKI, Y. Future Mobile Communications: LTE Optimization and Mobile Network Virtualization. Springer Science & Business Media, 2012.

ZAKI, Y. et al. LTE Wireless Virtualization and Spectrum Management. Wireless and Mobile Networking Conference (WMNC), IEEE, Budapest, Hungary, p. 1–6, Oct 2010.

ZHAI, G. et al. Load Diversity Based Optimal Processing Resource Allocation for Super Base Stations in Centralized Radio Access Networks. Science China Information Sciences, vol. 57, no. 4, p. 1–12, Apr 2014.

APPENDIX A AUTHORED ARTICLES

This appendix presents the articles published since the beginning of the doctorate until this thesis proposal. Theses articles are papers resulted from the investigations about virtualization and this proposal. These papers also present the proposed architecture that was shown in details in this thesis proposal. Finally, these articles also present the performance evaluation of this proposal when compared to other research efforts.

Submitted and In Review Articles

- Title: AIRTIME: End-to-End Virtualization Layer for RAN-as-a-Service in Future Multi-Service Mobile Networks
  - Authors: KIST, M.; SANTOS J.F.; COLLINS, D.; ROCHOL, J.; DASILVA, L.; BOTH, C. B..
  - Journal: IEEE Transactions on Mobile Computing.
  - DOI/URL: N.A.
  - Date: N.A
- Title: Virtual Radios, Real Services: Enabling RANaaS Through Radio Virtualization
  - Authors: SANTOS J.F.; KIST, M.; ROCHOL, J.; DASILVA, L..
  - Journal: IEEE Transactions on Network and Service Management
  - DOI/URL: N.A.
  - Date: N.A

Authored Published Articles

- Title: Flexible Fine-Grained Baseband Processing with Network Functions Virtualization: Benefits and Impacts.
  - Authors: KIST, M.; WICKBOLDT, J. A.; GRANVILLE, L. Z.; ROCHOL, J.; DASILVA, L.; BOTH, C. B..
  - Journal: Elsevier Computer Networks (ComNet).
  - DOI/URL: https://doi.org/10.1016/j.comnet.2019.01.021
  - Date: March, 2019.
- Title: SDR Virtualization in Future Mobile Networks: Enabling Multi-Programmable Air-Interfaces.
  - Authors: KIST, M.; ROCHOL, J.; DASILVA, L.; BOTH, C. B..

  - Conference: IEEE International Conference on Communications (ICC) .
  - DOI/URL: https://doi.org/10.1109/ICC.2018.8422643
  - Date: May, 2018.

- Title: HyDRA: A Hypervisor for Software Defined Radios to Enable Radio Virtualization in Mobile Networks.
  - Authors: KIST, M.; ROCHOL, J.; DASILVA, L.; BOTH, C. B..
  - Conference: IEEE Conference on Computer Communications Poster and Demo (INFOCOM Poster/Demo).
  - DOI/URL: https://doi.org/10.1109/INFCOMW.2017.8116510
  - Date: May, 2017.
  - Location: Atlanta, GA, USA

- Title: Adaptive Threshold Architecture for Spectrum Sensing in Public Safety Radio Channels.
  - Authors: KIST, M.; FAGANELLO, L. R.; BONDAN, L.; MAROTTA, M. A.; BOTH, C. B.; GRANVILLE, L. Z; ROCHOL, J..
  - Conference: IEEE Wireless Communications and Networking Conference (WCNC).
  - DOI/URL: https://doi.org/10.1109/WCNC.2015.7127484.
  - Date: May 2014.
  - Location: New Orleans, LA, USA.

APPENDIX B CO-AUTHORED PUBLISHED ARTICLES

This appendix presents articles that were co-authored during the doctorate. These articles could be divided into (*i*) articles related to this thesis proposal and (*ii*) marginal results from cooperation with other researchers. This division is shown next.

Co-authored articles related to this thesis proposal

- Title: Towards Enabling RAN as a Service - The Extensible Virtualisation Layer.
  - Authors: SANTOS, J. F.; KIST, M.; BELT J. V.; ROCHOL, J.; DASILVA L. A..
  - Conference: IEEE International Conference on Communications (ICC).
  - DOI/URL: https://doi.org/10.1109/ICC.2019.8761679.
  - Date: July 2019.
- Title: Integrating Dynamic Spectrum Access and Device-to-Device via Cloud Radio Access Networks and Cognitive Radio.
  - Authors: MAROTTA, M. A.; FAGANELLO, L. R.; KIST, M.; BONDAN, L.; WICKBOLDT, J. A.; GRANVILLE, L. Z.; ROCHOL, J.; BOTH, C. B..
  - Journal: International Journal of Communication Systems (ICC).
  - DOI/URL: https://doi.org/10.1002/dac.3698
  - Date: May, 2018.
- Title: Adaptive Monte Carlo Algorithm to Global Radio Resources Optimization in H-CRAN.
  - Authors: SCHIMUNECK, M.; KIST, M.; ROCHOL, J.; TEIXEIRA, A. R.; BOTH, C. B..
  - Conference: IEEE International Conference on Communications (ICC).
  - DOI/URL: https://doi.org/10.1109/ICC.2017.7996788
  - Date: May, 2017.
  - Location: Paris, France
- Title: Design Considerations for Software-Defined Wireless Networking in Heterogeneous Cloud Radio Access Networks.
  - Authors: MAROTTA, M. A.; KIST, M.; WICKBOLDT, J. A.; GRANVILLE, L. Z.; ROCHOL, J.; BOTH, C. B..
  - Journal: Springer Journal of Internet Services and Applications (JISA).

  - DOI/URL: https://doi.org/10.1186/S13174-017-0068-X.
  - Date: November, 2017.

Co-authored articles not directly related to this thesis proposal

- Title: Towards Low-Complexity Wireless Technology Classification Across Multiple Environments
  - Authors: FONTAINE, J.; FONSECA, E.; SHAHID, A.; KIST, M.; DASILVA, L. A.; MOERMAN, I.; POORTER, E. D..
  - Journal: Elsevier Ad-Hod Networks (AdHoc).
  - DOI/URL: https://doi.org/10.1016/j.adhoc.2019.101881
  - Date: May, 2019
- Title: Context-Aware Cognitive Radio Using Deep Learning.
  - Authors: PAISANA, F.; SELIM, A.; KIST, M.; ALVARES, P.; TALLON, J.; BLUEMM, C.; PUSCHMANN, A.; DASILVA, L..
  - Conference: IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN).
  - DOI/URL: https://doi.org/10.1109/DySPAN.2017.7920784.
  - Date: March, 2017.
  - Location: Baltimore, MD, USA.
- Title: An Experimental Assessment of Channel Selection in Cognitive Radio Networks
  - Authors: UMBERT, A.; SALLENT, O.; ROMERO, J. P.; GONZALES, J. S.; COLLINS, D.; KIST, M. .
  - Workshop: 5G - Putting Intelligence to the Network Edge (5G-PINE).
  - DOI/URL: Not Available Yet.
  - Date: May, 2018.
  - Location: Rhodes, Greece.

APPENDIX C HYDRA IN INTERNATIONAL PROJECTS

This appendix presents the *Wireless Network Slicing Functionality for 5G* (WINS_5G), an international project that was written exploring AIRTIME as an innovative networking slicing tool for testbed. The project is was aprooved in the 5GInFire Horizon 2020 programmee (grant agreement number 732497), as under he 1st Open Call for "new infrastructure functionalities and additional infrastructures". Futher details of WINS_5G are as follows:

- Abstract: WINS_5G will add the capability to instantiate new network function virtualisation (NFV) experimental vertical instances (EVIs) (*i*) to the radio access network supported by the radio slicing and virtualisation tool called Hypervisor for Software Defined Radios (HyDRA), developed by Trinity College Dublin researchers, (*ii*) which will be supported by the FUTEBOL Control Framework for the instantiation and orchestration of NFV experimentation scenarios in wireless, packet and optical networks. HyDRA as a VNF supported by Open Source MANO (OSM) will be available not only in the Iris Testbed, but also in other 5GINFIRE testbeds equipped with Universal Software Radio Peripheral (USRP). These elements will enhance the 5GINFIRE ecosystem by offering the opportunity for experimenters to test and evaluate advanced 5G use case scenarios, such as massive eHealth communications in the Internet-of-Things, high-definition multimedia services in mobile broadband, and ultra-low latency communications for industry automation. WINS_5G is well aligned with the 5GINFIRE vision for 5G and general enough to support different Experimental Vertical Instances (EVIs).
- Authors: KIST, M.; COLLINS, D..
- Period: From May/2018 to December/2020
- Organization: Trinity College Dublin
- Funding Origin: 5GInFire (H2020 grant number 732497) – 1st Open Call, Category 2 (new infrastructure functionalities and additional infrastructures)
- Project URL: https://5ginfire.eu/
  https://5ginfire.eu/wins

APPENDIX D RESUMO EXTENDIDO

As redes móveis atuais são baseadas em uma arquitetura fechada e inflexível tanto no *front-end* de rádio quando no núcleo da rede. A dependência com o *hardware* dificulta o desenvolvimento de novos padrões, impõe desafios na implantação de novas tecnologias de acesso que maximizam a capacidade e cobertura da rede e impede o provisionamento de serviços que podem realmente se adaptar à diferentes tráfegos de rede. Devido a essas limitações, espera-se que a futura rede 5G seja baseada na arquitetura de centralização de banda-base. Nessa arquitetura, os recursos computacionais de processamento são centralizados e *hardwares* de processamento banda-base são substituídos por versões implementadas em software. Apesar da arquitetura de centralização de banda-base trazer muitas oportunidades para as futuras redes 5G, sua implantação não é livre de desafios. Dentre os diversos desafios existentes, destacam-se os mais importantes (*i*) uma tecnologia de acesso para todos os serviços, (*ii*) área de cobertura da rede é limitada, e (*iii*) grande largura de banda na rede de *fronthaul*. Para resolver esses desafios, expande-se as fronteiras das redes móveis com o AIRTIME. AIRTIME é um protótipo que integra virtualização de BBUs e RRHs para realizar uma solução flexível e adaptável na qual a infraestrutura física da rede de acesso pode ser virtualizada e especializada para qualquer tipo de tecnologia. A camada de virtualização de RRHs dessa proposta permite que várias tecnologias de acesso heterogêneas coexistam em cima da mesma RRH física. Nossa proposta utiliza técnicas de processamento de sinais avançadas para dividir e abstrair uma *front-end* de rádio em múltiplos *front-end* virtuais. AIRTIME resolve os desafios da centralização de banda-base ao mesmo que tempo em que habilita um controle sem precedentes de qualquer aspecto da rede de acesso.

Ao longo desta tese foram conduzidos diversos experimentos com o objetivo de responder a hipótese da tese e as perguntas de pesquisa. Vamos focar nesse aspecto a partir de agora.

Hipótese: **incorporar conceitos de virtualização de rádio e virtualização de processamento de sinais em grão-fino expandem as fronteiras das futuras redes móveis com maior programabilidade, flexibilidade e escalabilidade.**

A camada de virtualização de BBU aumenta a escalabilidade por habilitar que as funções de processamento de sinais das vBBUs sejam distribuídas de acordo com as limitações da rede de *fronthaul* e do recursos de processamento dos *data centers*. A camada de virtualização de RRH é um mecanismo para prover, de forma flexível e programável, conectividade aos serviços da próxima geração de redes móveis. O desacoplamento entre BBU e RRH adiciona um nível de indireção. Esse nível prove uma RRH virtual para a vBBU, que opera como que se estivesse interfaceando com uma RRH física.

AIRTIME une as camadas de virtualização de BBU e RRH em uma arquitetura integrada. A metodologia empregada para mostrar a viabilidade dessa proposta é através de um protótipo que foi avaliado em uma rede experimental 5G em que uma RRH é virtualizada em duas vRRHs

que provêm conectividade para serviços com requisitos diferentes. Nós também realizamos análises matemáticas para obter a largura de banda e processamento necessários para diferentes distribuições de vBBUs, assim como simulações para obter o número máximo de vBBUs que podem compartilhar um *fronthaul* com largura de banda limitada.

Podemos responder as três Pergundas de Pesquisa (PP) que foram colocadas para guiar a pesquisa dessa tese com base nos resultados obtidos. As PPs e as respectivas respostas são:

PP I – As vRRHs mantém a mesma qualidade de transmissão quando comparadas à versão física?

Resposta: Nós avaliamos as características de transmissão das RRHs físicas e virtuais em uma rede similar à 5G. Nós apresentamos resultados experimentais que exploram o desempenho da virtualização de RRHs em termos de bandas de guarda e SINR. Os resultados mostram que a multiplexação de vRRHs é possível mas com impacto negativo na qualidade da transmissão.

PP II – As vBBUs de grão-fino mantém o desempenho para atender aos requisitos de latencia do 5G?

Resposta: A proposta de virtualizar as vBBUs em um grão-fino se baseia na adaptação da largura de banda e da latência de acordo com as características do *fronthaul* e das capacidades dos *data centers*. Os resultados demonstram que a virtualização em grão-fino ajuda a reduzir a largura de banda e latência quando as funções de processamento de sinais são migradas para próximo da RRH física. Contudo, isso custa uma redução na capacidade das BBUs adotarem mecanismos de cooperação.

PP III – Quais são os *trade-offs* por adotar uma vBBU de grão-fino invés de vBBUs atômica?

Resposta: vBBU de grão-fino são uma melhora da versão atômica pois habilitam *fronthauls* com largura de banda limitada ou latência elevada. Elas também melhoram a escalabilidade, já que as funções de processamento de sinais podem ser executadas em *hardwares* dedicados; e flexibilidade, visto que a cadeia de processamento da vBBU pode ser configurada em tempo de execução de acordo com os requisitos dos serviços.

Com base nos estudos realizados, é possível identificar diversas questões que ficaram em aberto nesta tese. Essas questões podem ser objeto de trabalhos futuros e são discutidas na subseção a seguir.

D.1 Melhorias and Desafios em Aberto

- **OpenAirInterface 5G:** o protótipo desenvolvido nesta tese usa *containers* LXC rodando *flowgraphs* do GNURadio para realizar o processamento de sinais. Foi feito assim porque o autor tinha alguma experiência com esses *softwares*. No entanto, esta abordagem não chega a ser compatível com LTE-A ou NB-IoT. Uma solução muito melhor seria utilizar o OpenAirInterface 5G, já que este implementa a pilha de protocolo LTE-A completa e é compatível com os padrões 3GPP. Este pode ser um tópico interessante para um aluno de mestrado.

- ***Container* optimizations:** o tempo de implantação e migração dos *containers* no protótipo do AIRTIME são maiores do que o esperado. Pode-se tomar emprestadas as técnicas de otimização da virtualização de *containers* de última geração e aplicá-las ao protótipo desta tese para obter tempos de implantação e migração muito melhores.

- **Algortimos de colocacão de *containers*:** esta tese não usa nenhum algoritmo de colocação de *containers* para decidir em qual *data center* e máquina os *containers* devem ser implantados ou migrados. Futuras pesquisas podem explorar diferentes algoritmos de posicionamento considerando as peculiaridades das VNFs de processamento de sinais (algumas VNFs requerem taxa de dados constante de *input* e/ou geram um *output* em taxa de dados constante, outros podem ter sua taxa de dados estimada. Tudo isso, somado aos custos de infraestrutura, como eletricidade (o custo de usar recursos de processamento em uma região ou outra) podem ser utilizados na decisão de colocação dos *containers*.

- **RATs heterogêneas com HyDRA:** o protótipo desenvolvido nesta tese usa apenas tecnologias de acesso baseadas em OFDM, *i.e.*, LTE-A e NB-IoT. Medir o desempenho do HyDRA, ou seja, isolamento, memória e uso de CPU com tecnologias não-OFDM é um projeto de último ano interessante para alunos de graduação.

- **Latência introduzida pelo HyDRA:** esta tese de doutorado não apresentou resultados em relação à latência introduzida pelos *hypervisor* quando comparada às soluções não virtualizadas. Medir o atraso adicional introduzido por HyDRA é um resultado interessante e deveria ter sido explorado nesta tese. Isso pode ser medido criando um aplicativo de usuário-final que gera pacotes de dados contendo um *timestamp* de quando o pacote foi gerado. Este pacote é então enviado para a vBBU para ser irradiado pela antena. No lado da recepção, outro aplicativo do usuário-final recebe esse pacote e estima a latência (após passar pela vBBU). Pode-se medir o atraso sem usar HyDRA e com diferentes vBBUs, e com HyDRA considerando diferentes números de vRRHs e diferentes tipos de vBBUs. Este é um projeto interessante para um aluno de mestrado.

- ***Automatic Gain-Control* for HyDRA:** o protótipo do `RRH Hypervisor` requer que todas as vRRHs tenham um ganho de transmissão similar. Uma melhoria interessante

seria implementar um mecanismo de AGC para que o HyDRA possa ajustar os ganhos de todas as vRRHs automaticamente.