

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

ARIEL GALANTE DALLA COSTA

**Uma Solução de Orquestração Para  
Ambientes *Dynamic* C-RAN**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de Mestre em Ciência da  
Computação

Orientador: Prof. Dr. Lisandro Zambenedetti  
Granville

Co-orientador: Prof. Dr. Cristiano Bonato Both

Porto Alegre  
2018

## CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Galante Dalla Costa, Ariel

Uma Solução de Orquestração Para Ambientes *Dynamic C-RAN* / Ariel Galante Dalla Costa. – Porto Alegre: PPGC da UFRGS, 2018.

89 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2018. Orientador: Lisandro Zambenedetti Granville; Co-orientador: Cristiano Bonato Both.

1. Dynamic C-RAN. 2. Orquestração NFV. 3. Composição de VNF. I. Zambenedetti Granville, Lisandro. II. Bonato Both, Cristiano. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof<sup>a</sup>. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretora do Instituto de Informática: Prof<sup>a</sup>. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. João Luiz Dihl Comba

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“O sucesso é ir de fracasso em fracasso sem perder entusiasmo”*

— WINSTON CHURCHILL

## RESUMO

*Dynamic Cloud Radio Access Network (dynamic C-RAN)* é uma arquitetura de redes sem fio emergente que objetiva trazer benefícios como flexibilidade e agilidade para redes móveis de quinta geração (5G). Em *Dynamic C-RAN*, a carga de processamento das funções de rádio é distribuída ao longo de uma hierarquia de nuvens e, para tal, tem-se investigado o uso de *Network Function Virtualization (NFV)*. NFV tem potencial para oferecer a orquestração das funções de rádio, através da sua subdivisão em componentes de *Virtualized Network Function (VNF)*. Apesar da eficácia das soluções de orquestração NFV, tanto a academia quanto a indústria estão considerando as *Virtualized Network Functions (VNF)* como elementos atômicos, desconsiderando as vantagens que a divisão de VNFs em vários componentes diferentes. Com o objetivo de melhorar a orquestração VNF em cenários de *dynamic C-RAN*, neste trabalho são propostos Maestro e Orquestra. O primeiro é um orquestrador de NFV para ambientes sem fio orientado a composição de VNFs, a fim de estimar os potenciais ganhos ao se aplicar os conceitos de NFV em *dynamic C-RAN*. O segundo se refere a um orquestrador com a finalidade de operar em um ambiente experimental do tipo *dynamic C-RAN*, suportando os requisitos e atendendo as variações dos recursos computacionais e da rede. Ambos tiveram a avaliadas da efetividade e do custo de orquestração em termos de aderência aos requisitos de comunicação entre as funções de rádio e os posicionamentos dos componentes das VNFs na rede. Além disso, avaliou-se a relação entre tempo para realizar esses posicionamentos e a taxa de dados transmitida, considerando cenários de ocupação variável em *dynamic C-RAN*.

**Palavras-chave:** Dynamic C-RAN. Orquestração NFV. Composição de VNF.

## An Orchestration Solution For Dynamic C-RAN Environments

### ABSTRACT

Dynamic Cloud Radio Access Network (*dynamic* C-RAN) is an emerging wireless networking architecture that aims to bring benefits such as flexibility and agility to fifth generation (5G) mobile networks. In Dynamic C-RAN, the processing load of the radio functions is distributed along a cloud hierarchy and, for this purpose, the use of Network Function Virtualization (NFV) has been investigated. NFV has the potential to offer the orchestration of radio functions through its subdivision into Virtualized Network Function (VNF) components. Despite the effectiveness of NFV orchestration solutions, both academia and industry are considering VNFs (Virtualized Network Functions) as atomic elements, disregarding the advantages of splitting VNFs into several different components. In order to improve the VNF orchestration in scenarios of *dynamic* C-RAN, in this work are proposed Maestro and Orchestra. The first is an NFV orchestrator for VNF composition-oriented wireless environments in order to estimate potential gains by applying NFV concepts in *dynamic* C-RAN. The second refers to an orchestrator for the purpose of operating in an experimental environment of type *dynamic* C-RAN, supporting the requirements and taking into account the variations of computational resources and the network. Both Maestro and Orchestra evaluated the effectiveness and cost of orchestration in terms of adherence to the communication requirements between the radio functions and the positioning of the VNF components in the network. In addition, the relationship between time to perform these placements and the transmitted data rate was evaluated, considering scenarios of variable occupation in *dynamic* C-RAN.

**Keywords:** Dynamic C-RAN, NFV Orchestration, VNF Composition.

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BBU	Baseband Unit
BS	Base Station
BSS	Business Support System
CAPEX	CAPital EXpenditure
COPA	Container Orchestration and Provisioning Architecture
CPU	Central Process Unit
C-RAN	Cloud Radio Access Network
CP	Prefixo Cíclico
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
IaaS	Infrastructure as Service
MAC	Medium Access Control
MANO	Management and Orchestration
MEC	Multiaccess Edge Computing
NFV	Network Functions Virtualization
NFVI	Network Functions Virtualization Infrastructure
NFVO	Network Function Virtualized Orchestrator
NS	Network Service
OPEX	OPerational EXpenditure
OSS	Operation Support System
PNF	Physical Network Function
RRH	Radio Remote Head
sFlow	Simple Flow

SNMP	Simple Network Management Protocol
FFT	Transformada de Fourier Rápida
VDU	Virtualized Deployment Unit
VIM	Virtualized Infrastructure Manager
VM	Virtualized Machine
VNF	Virtualized Network Function
VNFC	Virtualized Network Function Component
VNFD	Virtualized Network Function Descriptor
VNFM	Virtualized Network Function Manager
USRP	Universal Software Radio Peripheral

## LISTA DE FIGURAS

Figura 2.1 Um exemplo de <i>dynamic C-RAN</i> .....	14
Figura 2.2 Arquitetura NFV com os blocos do NFV MANO detalhados.....	16
Figura 3.1 Arquitetura do NFVO Maestro .....	23
Figura 3.2 Arquitetura do NFVO Orquestra .....	29
Figura 4.1 Divisão de funções proposta por Wubben .....	37
Figura 4.2 Exemplo de implantação de funções de forma atômica e, divisão de múltiplas opções.....	38
Figura 4.3 Divisão de funções de rádio em VNFCs, mostrando <i>downlink</i> e <i>uplink</i> da BS .....	41
Figura 4.4 Cenário de VNFCs e USRP.....	42
Figura 4.5 Cenário de VNFCs e USRP emulado .....	43
Figura 4.6 Cenário experimental utilizando funções de radio sem fio.....	44
Figura 5.1 Consumo de taxa de dados agregada no <i>fronthaul</i> .....	47
Figura 5.2 Tempo para calcular o posicionamento dos VNFCs.....	48
Figura 5.3 Transmissão de dados no <i>fronthaul</i> .....	51
Figura 5.4 Tempo necessário para encontrar uma solução .....	52
Figura 5.5 Consumo de taxa de dados entre VNFCs em <i>download</i> (D) e <i>upload</i> (U) ...	54
Figura 5.6 Latência das VNFs fim-a-fim .....	56
Figura 5.7 Tempo de implantação das VNFs .....	57
Figura 5.8 Tempo de migração de funções entre uma nuvem e outra.....	58



## LISTA DE TABELAS

Tabela 2.1	Trabalhos relacionados ao posicionamento de VNFs e SFCs .....	18
Tabela 2.2	Características dos orquestradores .....	20
Tabela 3.1	Funções disponibilizadas pelo orquestrador ao operador de rede .....	32
Tabela 4.1	Quantidade de recursos disponíveis em cada VM (nuvem) .....	43
Tabela 5.1	Distribuição da ocupação das <i>macro</i> e <i>small cells</i> .....	49

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>11</b>
<b>2 FUNDAMENTAÇÃO E TRABALHOS RELACIONADOS.....</b>	<b>14</b>
<b>2.1 <i>Dynamic</i> C-RAN.....</b>	<b>14</b>
<b>2.2 Gerenciamento e orquestração em NFV .....</b>	<b>16</b>
<b>2.3 Soluções existentes .....</b>	<b>18</b>
<b>3 MAESTRO &amp; ORQUESTRA .....</b>	<b>22</b>
<b>3.1 Solução Conceitual e Maestro.....</b>	<b>22</b>
3.1.1 Arquitetura do orquestrador .....	23
3.1.2 Modelo de avaliação .....	25
<b>3.2 Solução Experimental e Orquestra .....</b>	<b>28</b>
3.2.1 Arquitetura do orquestrador .....	28
3.2.2 Exemplo de algoritmo de orquestração.....	32
<b>4 AVALIAÇÃO .....</b>	<b>37</b>
<b>4.1 Estudo de caso do orquestrador conceitual .....</b>	<b>37</b>
<b>4.2 Estudo de caso do orquestrador experimental.....</b>	<b>40</b>
<b>5 RESULTADOS .....</b>	<b>46</b>
<b>5.1 Resultados da solução conceitual.....</b>	<b>46</b>
5.1.1 Avaliação com demanda das BSs fixas .....	46
5.1.2 Avaliação com demanda das BSs variável.....	49
<b>5.2 Resultados da solução experimental.....</b>	<b>54</b>
<b>5.3 Discussões entre soluções.....</b>	<b>59</b>
<b>6 CONCLUSÕES .....</b>	<b>61</b>
<b>6.1 Trabalhos futuros .....</b>	<b>62</b>
<b>REFERÊNCIAS .....</b>	<b>63</b>
<b>APÊNDICE A — .....</b>	<b>66</b>
<b>APÊNDICE B — .....</b>	<b>75</b>

## 1 INTRODUÇÃO

*Cloud Radio Access Network* (C-RAN) é uma arquitetura de redes móveis sem fio que desacopla os componentes da Estação Base (BS) em unidades de rádio e de processamento, *i.e.*, *Radio Remote Head* (RRH) e *Base Band Unit* (BBU), respectivamente (TAN; YANG; WANG, 2017). A RRH fornece suporte para conexão entre a antena e a fibra, além de outras funcionalidades, como realizar processamento digital, conversão digital para analógica, conversão analógico para digital e amplificação de potência e filtragem. Muitas vezes conhecida como *Data Unity* (DU), a BBU é responsável pelo processamento do sinal. Em C-RAN, as RRHs são implantadas na RAN e a BBU pode ser implantada em uma infraestrutura central baseada em nuvem. Esta abordagem fornece compartilhamento de recursos entre sites em uma BBU *pool* virtualizada, resultando na redução dos custos operacionais da rede, potência e consumo de energia, que são menores quando comparadas com as abordagens RANs tradicionais (CHECKO et al., 2015).

Considerando os benefícios de C-RAN, *dynamic* C-RAN emergiu como uma evolução de C-RAN, dividindo funções de rádio *e.g.*, *Medium Access Control* (MAC) em uma infraestrutura hierárquica de *data centers* baseada em computação em nuvem. Essa abordagem visa a distribuição de processamento de funções de rádio ao longo da rede de acordo com suas funcionalidades (5G-PPP-WORKING-GROUP, a). Essa distribuição resulta em benefícios como adaptabilidade, balanceamento de carga, flexibilidade de implantação de serviços e melhor eficiência energética, as quais são algumas das características voltadas para redes móveis de quinta geração (5G) (5G-PPP-WORKING-GROUP, b). Em *dynamic* C-RAN, enquanto algumas funções precisam de suporte de hardware dedicado, devido às limitações de desempenho, como por exemplo, criptografia, filas de HARQ/usuários e FEC, outras funções podem ser executadas em um hardware comum de computador, permitindo que as funções sejam implementadas em software, como por exemplo a MAC. Por esta razão, as funções de rádio podem ser movidas dinamicamente na rede de acordo com os requisitos de serviço específicos, aproveitando as vantagens de computação em nuvem para prover o dinamismo objetivado por *dynamic* C-RAN (5G-PPP-WORKING-GROUP, a).

Embora o benefício do dinamismo oferecido pelos ambientes *dynamic* C-RAN, *i.e.*, mover as funções de rádio dinamicamente na rede, coordenar essas funções de rádio não é uma tarefa trivial. Por exemplo, a distribuição de funções de rádio em diferentes nuvens deve ser baseada em requisitos rigorosos, como baixa latência, *jitter* e menor taxa de

dados no *fronthaul*, dependendo da função de rádio (ABDELWAHAB et al., 2016). Nesse sentido, academia e indústria têm explorado conceitos de *Network Functions Virtualization* (NFV) para alocar e orquestrar funções de rádio virtualizadas, que são designadas como *Virtualized Network Functions (VNF)* (MIJUMBI et al., 2016). Desta forma, algumas funcionalidades de NFV, como por exemplo, serviços sob demanda e orquestração de recursos, podem ser usadas em contextos de *dynamic C-RAN* (ABDELWAHAB et al., 2016), permitindo adaptar, gerenciar e orquestrar funções de rádio virtualizadas (MIJUMBI et al., 2016). Ao aplicar os conceitos de NFV em *dynamic C-RAN*, as BBU's podem ser implementadas como VNFs (5G-PPP-WORKING-GROUP, a), que são compostas de *VNF Components (VNFC)*, este último que representa cada função de rádio da BBU, e são implantados em *Virtualized Deployment Units (VDU)* (DALLA-COSTA et al., 2017). Desta forma, um *NFV Orchestrator (NFVO)* é necessário para determinar o posicionamento de VNFCs e VDUs, considerando as diversas restrições das funções de rádio.

Plataformas de orquestrações tais como Cloud4NFV (SOARES et al., 2014), T-NOVA (XILOURIS et al., 2014) e OpenBaton (WANTAMANEE et al., 2016) surgiram considerando VNFs como funções atômicas, ou seja, elementos indivisíveis. No entanto, essas plataformas não aproveitam ao máximo os conceitos de orquestração NFV. Por exemplo, uma BBU virtualizada pode ser implantada de duas maneiras diferentes: *i)* totalmente implantada em um *front-end* de rádio (*i.e.*, perto da BS), com a finalidade de economizar recursos de rede (por exemplo, taxa de dados), ou *ii)* ter seu processamento dividido sobre uma infraestrutura de *dynamic C-RAN*, com a finalidade de equilibrar a carga de processamento entre as nuvens que possuem recursos computacionais disponíveis.

Desta forma, faz-se necessário seguir a pesquisa em duas etapas, com a finalidade de cobrir inteiramente o problema de pesquisa. A primeira, constitui em uma análise teórica da literatura e a proposição de uma solução conceitual, validando uma prova de conceito em que os ambientes do tipo *dynamic C-RAN* podem ter sua dinamicidade suportada através de NFV, reduzindo o consumo de taxa de dados e maximizando o uso dos recursos computacionais. Já na segunda etapa, uma implementação de *dynamic C-RAN* em um ambiente experimental, adaptando o ambiente através de uma solução de NFV, e demonstrando que a adaptabilidade do cenário é plausível em um tempo aceitável do ponto de vista do orquestrador. Por esta razão, o NFVO deve considerar a alocação e o gerenciamento do ciclo de vida das funções de rádio virtualizadas com diferentes restrições

de recursos. Portanto, este trabalho visa cobrir esta lacuna de orquestração, considerando a composição de VNFs para implantar VNFCs em infraestruturas hierárquicas de nuvem, com suporte para requisitos de funções de rádio de *dynamic C-RAN*.

Neste trabalho, houve contribuição na orquestração de NFV para *dynamic C-RAN* através de uma solução de orquestração dividida em duas etapas, e que expandiu-se em dois trabalhos: Maestro e Orquestra. O primeiro trabalho, chamado de Maestro é um orquestrador conceitual utilizando programação linear, que apresenta uma prova de conceito com foco nas características funcionais e técnicas de um ambiente *dynamic C-RAN* simulado. A segunda etapa consiste em um orquestrador experimental baseado em políticas, com suporte para algoritmos de orquestração personalizados, habilitando o gerenciamento e orquestração com diferentes composições (*i.e.*, divisões) de funções de rádio de redes sem fio, chamado de Orquestra. Nesta última etapa, a orquestração é analisada em um cenário de rede experimental, considerando o uso de funções de rádio virtualizadas reais.

O restante deste trabalho está organizado da seguinte forma: no Capítulo II, apresenta-se uma visão geral dos trabalhos relacionados com o *dynamic C-RAN* e a aplicabilidade dos conceitos do NFV em tais ambientes. No Capítulo III, apresenta-se as duas etapas da pesquisa, a primeira de solução conceitual e a segunda de solução experimental para ambientes *dynamic C-RAN*. No Capítulo IV, apresenta-se o modelo de avaliação para cada uma das etapas da seção III. No Capítulo V, avalia-se a solução conceitual, comparando com a solução experimental, avaliando e discutindo ambas soluções. Por fim, no Capítulo VI, apresenta-se conclusões e perspectivas finais para os trabalhos futuros.

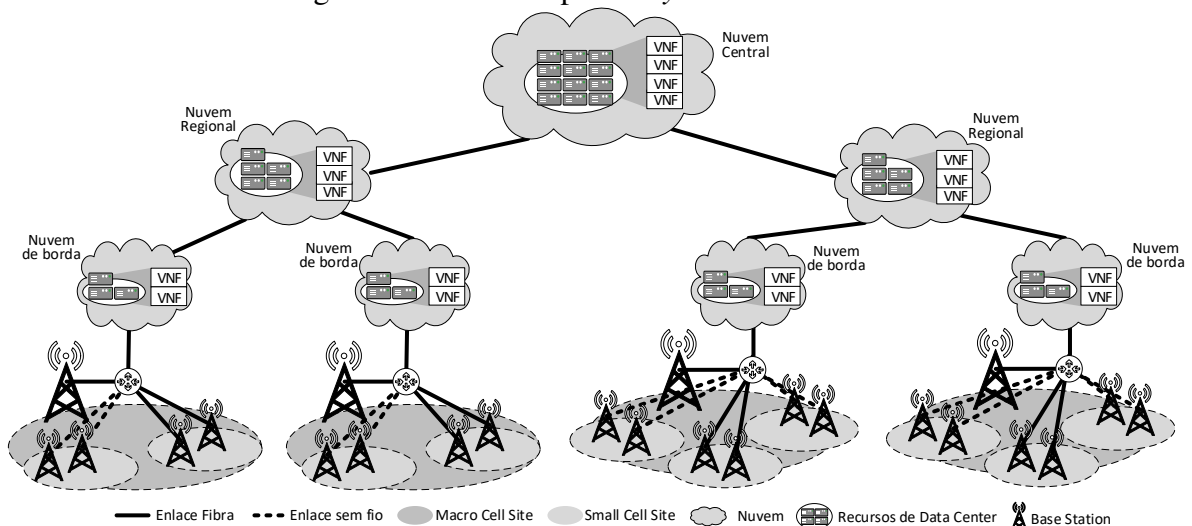
## 2 FUNDAMENTAÇÃO E TRABALHOS RELACIONADOS

Neste Capítulo, é fornecida uma visão geral sobre *dynamic* C-RAN e a aplicabilidade de conceitos de NFV nestes cenários. Primeiro, o ambiente *dynamic* C-RAN é explicado na Seção 2.1, apresentando e discutindo importantes esforços de pesquisa nessa área. Em seguida, é apresentada a aplicabilidade de NFV em ambientes *dynamic* C-RAN, que são discutidas na Seção 2.2. Por fim, na terceira e última parte deste Capítulo são apresentadas as soluções de orquestração existentes e os trabalhos relacionados, descritos na Seção 2.3.

### 2.1 *Dynamic* C-RAN

Enquanto em C-RAN todas as funções da BBU são movidas para uma nuvem centralizada (CHEN; DUAN, 2011; CHECKO et al., 2015), *dynamic* C-RAN pode ser vista como uma evolução de C-RAN, onde as funções de rádio são distribuídas em uma infraestrutura de nuvem hierárquica ao longo da rede. A principal vantagem da natureza hierárquica de *dynamic* C-RAN é a alocação dinâmica das funções de rádio, permitindo o provisionamento de serviços flexível (PENG et al., 2014).

Figura 2.1: Um exemplo de *dynamic* C-RAN



O cenário *dynamic* C-RAN, representado na Figura 2.1 é composto por *data centers* chamados de nuvem, que são organizados hierarquicamente (5G-PPP-WORKING-GROUP, a). Essas nuvens processam as *Physical Network Functions* (PNF), que são as

funções de rede legadas, e as funções de rede virtualizadas genéricas, que são conhecidas como VNFs. A nuvem central fornece o maior volume de recursos na infraestrutura, enquanto a nuvem regional e a de borda estão mais próximas do *front-end* da BS, contendo menos recursos computacionais do que a nuvem central. Considerando essa disposição hierárquica, uma BBU virtualizada pode ser implantada nas nuvens regionais ou centrais, para por exemplo, balanceamento de carga, considerando um número crescente de usuários ativos em uma borda específica (PENG et al., 2014). Além disso, diferentes políticas podem ser facilmente implantadas, como estratégias de balanceamento de carga personalizadas e políticas de economia de energia, dada a capacidade de distribuição dos serviços de rádio em diferentes nuvens (5G-PPP-WORKING-GROUP, a).

Quando comparado à abordagem centralizada de C-RAN, *dynamic C-RAN* apresenta muitas vantagens, como eficiência energética, economia de capital e gastos operacionais (CAPEX e OPEX), adaptabilidade de tempo de execução, flexibilidade de provisionamento de serviços e adaptabilidade de serviço. Por esta razão, as redes móveis 5G prevêem a operação em *dynamic C-RAN* (5G-PPP-WORKING-GROUP, a; 5G-PPP-WORKING-GROUP, b). Bartelt *et al.* (BARTELT et al., 2015) e Liu *et al.* (LIU et al., 2015) discutem as vantagens de dividir funções de rádio no processamento local e remoto, mostrando que a divisão de funções de rádio pode reduzir os requisitos de taxa de dados no *fronthaul*. Neste mesmo caminho, Wubben *et al.* (WUBBEN et al., 2014) analisa o impacto da taxa de dados do *fronthaul* por cada tipo de divisão de funções usando *Radio Access Point* (RAP), demonstrando a quantidade de taxa de dados consumida no *fronthaul* por cada uma das funções de rádio, tal como MAC, Soft- Bit, RX Data, Subframe e In-phase & Quadrature (I/Q). Embora estes trabalhos já mencionados destaquem a importância da divisão de funções de rádio, os autores não consideram questões relacionadas à gerenciamento e coordenação de funções de rádio divididas, de forma que possam ser otimizadas.

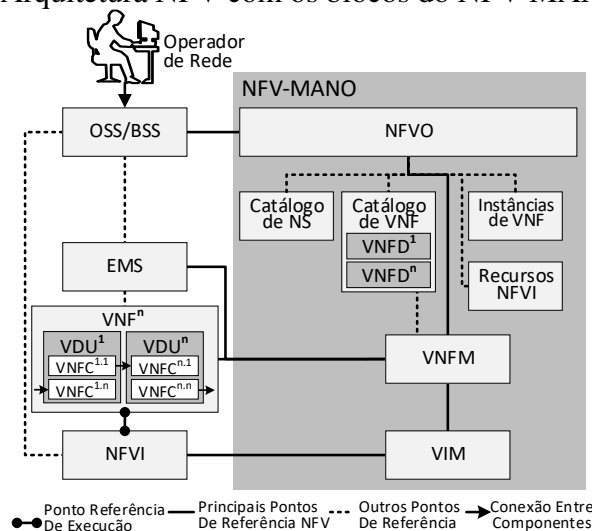
Coordenar as funções de rádio em *dynamic C-RAN* não é uma tarefa fácil, especialmente devido a restrições estritas, como a retransmissão de camadas de *Subframe*, que requer latência inferior a 1ms, enquanto o tempo do *Hybrid Automatic Repeat Request* (HARQ) não deve demorar mais do que 8ms (ORTÍN et al., 2014). Outro fator que pode alterar o bom funcionamento dos ambientes *dynamic C-RAN* é a mobilidade de *User Equipment* (UE) em toda a infraestrutura, exigindo que a rede seja adaptável de acordo com as variações das demandas dos usuários. Por esta razão, a academia e a indústria têm explorado conceitos de NFV para gerenciar e orquestrar funções de rádio virtuali-

zadas (LIU et al., 2015). Abdelwahab *et al.* (ABDELWAHAB et al., 2016), Liu *et al.* (LIU et al., 2015), Marotta *et al.* (MAROTTA et al., 2017) e, Riggio *et al.* (RIGGIO; RASHEED; NARAYANAN, 2015) advogam que NFV pode ter um papel importante no gerenciamento e orquestração de funções de rádio sem fio virtualizadas (MIJUMBI et al., 2016). Desta forma, *dynamic C-RAN* pode considerar BBUs virtualizadas como VNFs, com cada função de rádio tornando-se um VNFC (DALLA-COSTA et al., 2017). Contudo, o emprego de NFV em *dynamic C-RAN* conduz a novos desafios, especialmente (i) como implementar, distribuir e orquestrar VNFs em uma infraestrutura *dynamic C-RAN* e (ii) como garantir um desempenho satisfatório no posicionamento e gerenciamento do ciclo de vida das VNFs, suportando todos os requisitos dos ambientes *dynamic C-RAN*. Desta forma, os detalhes sobre a adoção do NFV em *dynamic C-RAN* são apresentados na próxima Seção, apresentando os principais conceitos do NFV e discutindo os desafios inerentes da combinação de NFV e *dynamic C-RAN*.

## 2.2 Gerenciamento e orquestração em NFV

Iniciado pelas principais operadoras de redes de telecomunicações do mundo, o *NFV Industry Specification Group (ISG)* foi criado sobre o *European Telecommunications Standards Institute (ETSI)*. Este grupo apresentou um consenso quanto à virtualização das funções de rede através da definição de conceitos de NFV (CHIOSI et al., 2012).

Figura 2.2: Arquitetura NFV com os blocos do NFV MANO detalhados



A medida que os conceitos de NFV evoluíram, o gerenciamento e a orquestração de VNFs ganharam importância crescente, levando o NFG ISG a criar o grupo *NFV Ma-*



*nagement and Orchestration* (NFV MANO) (QUITTEK et al., 2014). O NFV MANO prevê a criação de uma arquitetura para padronizar as soluções, provendo suporte para a indústria, e também, propondo mecanismos de gerenciamento e orquestração de VNFs. A arquitetura do NFV MANO é mostrada na Figura 2.2. Os elementos centrais da arquitetura de NFV são as VNFs, que são as implementações em softwares das funções de rede, implantadas sobre a *Network Function Virtualization Infrastructure* (NFVI). As VNFs são compostas por VNFCs, que são os menores elementos das VNFs. Os VNFCs são implantados em *containers* virtualizados, representados por VDUs, diretamente mapeados em um *container* virtualizado dedicado (e.g. máquina virtual ou Linux *container*) (CHIOSI et al., 2012).

Cada VNFC pode representar uma função de rádio virtualizada, como discutido por Abdelwhab *et al.* (ABDELWAHAB et al., 2016), que propõe uma arquitetura para NFV em redes 5G, discutindo o emprego de funções de rádio virtualizadas seguindo a arquitetura ETSI NFV MANO. Este trabalho aborda a importância de um NFVO para ambientes 5G, de como orquestrar recursos virtualizados e gerenciar dependências e fluxos de informações entre funções virtualizadas e não virtualizadas. Para implantar uma VNF, cada uma é descrita por um *VNF Descriptor* (VNFD), que detalha como cada VDU está conectado para compor cada VNF. Na arquitetura de NFV, todos os blocos do NFV MANO estão conectados, e são responsáveis pelo gerenciamento e orquestração de VNFs e serviços de rede. O *Operations/Business Support System* (OSS/BSS) é responsável por funções de controle e gerenciamento de regras de registro. Já o *Elemental Manager System* descreve individualmente como cada VNF é gerenciada em cada plataforma comum, ou seja, abstrai o hardware em relação a VNF. Informações sobre a composição, instanciação e operações do VNF são tratadas pelo NFVO, que é responsável pela inteligência no processo de provisionamento e composição de VNFs. O NFVO tem duas tarefas básicas: (i) orquestrar VNFs sobre múltiplos *Virtual Infrastructure Manager* (VIM), que são os gerenciadores de infraestrutura virtualizada, e (ii) gerenciar o ciclo de vida de VNFs e serviços de rede, interagindo com diferentes *VNF Managers* (VNFM), que são os gerenciadores das VNFs. Além disso, o NFVO também gerencia o catálogo de serviços de rede (NS), de VNFs, guarda as instâncias de VNFs e as instâncias de infraestrutura virtualizada.

## 2.3 Soluções existentes

As linhas de pesquisa na área de orquestração de NFV se dividem em trabalhos que focam no posicionamento e provisionamento formal de VNFs e *Service Function Chaining* (SFC), e outros que implementam NFV experimentalmente. Portanto, ambas as linhas de pesquisa serão explicadas a seguir.

Tabela 2.1: Trabalhos relacionados ao posicionamento de VNFs e SFCs

Autor	Tipo Posicionamento	Considera Funções Legadas	Considera Nodos Virtuais	Posiciona VNFCs	Considera Redes Sem Fio
(MOENS; TURCK, 2014)	VNFs	Sim	Não	Não	Não
(CLAYMAN et al., 2014)	VNFs	Não	Sim	Não	Não
(LUIZELLI et al., 2015)	VNFs e SFCs	Não	Sim	Não	Não
(BARI et al., 2016)	VNFs	Não	Não	Não	Não
(MECHTRI et al., 2017)	VNFs e SFC	Não	Não	Não	Não
(XU et al., 2017)	VNFs	Não	Sim	Não	Não
(JANG et al., 2017)	VNFs e SFCs	Não	Sim	Não	Não
(NAM; SONG; CHUNG, 2017)	VNFs e SFCs	Não	Não	Não	Não
(RIGGIO; RASHEED; NARAYANAN, 2015)	VNFs	Sim	Não	Não	Sim

A otimização no posicionamento de VNFs é um dos desafios de pesquisa enfrentados no NFV MANO. Alguns trabalhos cobrem o posicionamento formal de VNFs e SFCs. Esta forma de otimização fornece o provisionamento e posicionamento de VNFs, bem como SFCs na rede, com a finalidade de alcançar algum objetivo, como por exemplo, balanceamento de carga ou economia de taxa de dados. Desta forma, uma avaliação formal do posicionamento de VNFs foi proposta por Moens, com foco em VNFs e funções físicas legadas (MOENS; TURCK, 2014). As propostas focadas em SFC também são encontradas na literatura, incluindo o posicionamento autônomo de nós virtuais e alocação de serviços (CLAYMAN et al., 2014; LUIZELLI et al., 2015; LI; QIAN, 2016; BARI et al., 2016; MECHTRI et al., 2017; XU et al., 2017; JANG et al., 2017; MECHTRI

et al., 2017). Seguindo as abordagens de SFC, (NAM; SONG; CHUNG, 2017) propõe um esquema de NFV *Service Chaining* (cNSC), que calcula o número ótimo de clusters para minimizar o tempo de ponta-a-ponta dos serviços MEC. No contexto de redes sem fio, (RIGGIO; RASHEED; NARAYANAN, 2015) propôs um orquestrador para WLAN baseado em uma arquitetura parecida com o NFV MANO, porém, esta trata as VNFs de forma atômica, *i.e.*, não suportando divisão de VNFs em múltiplos VNFCs.

Através da Tabela 2.1 é possível observar as características dos trabalhos relacionados ao posicionamento de VNFs e VNFCs. As linhas da tabela representam os trabalhos relacionados e seus autores, enquanto que as colunas representam as características dos trabalhos. Através da tabela é possível observar que todos os trabalhos fazem o posicionamento de VNFs, e além disso, os trabalhos de (LUIZELLI et al., 2015), (MECHTRI et al., 2017), (JANG et al., 2017) e (NAM; SONG; CHUNG, 2017) fazem também SFC. Apenas o trabalho de (MOENS; TURCK, 2014) e (RIGGIO; RASHEED; NARAYANAN, 2015) consideram funções de rede legadas no processo de decisão. Já os trabalhos que consideram nodos virtuais são o de (CLAYMAN et al., 2014), (LUIZELLI et al., 2015), (XU et al., 2017) e (JANG et al., 2017). O único dos trabalhos que tem foco em orquestração de redes sem fio é o de (RIGGIO; RASHEED; NARAYANAN, 2015). Como pode-se observar na penúltima coluna a direita, nenhum dos trabalhos relacionados consideram o posicionamento de VNFCs, desta forma, não aproveitando toda a granularidade disponibilizada por NFV, e, necessária para a implementação dos ambientes do tipo *dynamic C-RAN*.

Do ponto de vista experimental, com base nas especificações ESTI NFV MANO, surgiram algumas propostas com foco em gerenciamento e orquestração de VNFs utilizando NFVOs. Uma solução, chamada de T-NOVA (XILOURIS et al., 2014) apresenta um NFVO focado em provisionar um *marketplace* de VNFs, suportando serviços ponta-a-ponta, *i.e.*, o NFVO suporta a compra, modificação e instalação (entre outras funcionalidades) de VNFs através de uma loja online. Da mesma forma, a Soares *et al.* (SOARES et al., 2014) também apresenta um NFVO para a orquestração de serviços de ponta-a-ponta, propondo o Cloud4NFV, uma plataforma que gerencia e orquestra VNFs de acordo com as especificações padrão ETSI MANO. Similarmente, o OpenStack Tacker é uma plataforma NFV para orquestração genérica de VNFs, fornecendo uma pilha funcional para orquestrar serviços de rede de ponta-a-ponta usando VNFs (CHEN et al., 2017). Da mesma forma, o Open Baton é uma solução NFVO de código aberto, que também é baseada na arquitetura ETSI NFV MANO, com foco no desempenho e portabilidade de VNFs,

utilizando EMS genéricos, e também, diferentes VNFMs genéricos (WANTAMANEE et al., 2016).

Com base em Open Baton, a Carella *et al.* (CARELLA et al., 2017) apresenta uma arquitetura para o NFV MANO, combinando diferentes tecnologias de infraestrutura para suportar a implantação de serviços de rede baseados em *containers* ao longo da borda da rede, com foco em 5G, utilizando *Multiaccess Edge Computing* (MEC). Considerando um cenário de 5G *Fog*, Lingen *et al.* (LINGEN et al., 2017) propôs uma arquitetura para prover funcionalidades do NFV MANO à infraestrutura do tipo *Fog*, mas sem considerar o uso de VNFCs para representar cada divisão de função de rádio.

Tabela 2.2: Características dos orquestradores

Recursos por NFVO	T-NOVA	Cloud4NFV	OpenBaton	Tacker
VIM	OpenStack e outros	OpenStack e outros	OpenStack e outros	OpenStack
Serviço fim a fim	Sim	Sim	Sim	Sim
Marketplace	Sim	Sim	Não	Não
Tipo de recursos	VNFs e SFCs	VNFs e SFCs	VNFs	VNFs e SFCs
Decisão baseada na composição de VNFs	Não	Não	Não	Não
Algoritmos de Otimização	Sim	Sim	Não	Não
Interface Gráfica	Sim	Sim	Sim	Não

Através da Tabela 2.2 é possível observar as características dos orquestradores presentes na literatura. As colunas da tabela representam os orquestradores, enquanto as linhas representam as características de cada um deles. Pode-se observar que os orquestradores TNOVA, Cloud4NFV, OpenBaton e Tacker suportam serviços fim a fim, e apenas T-NOVA e Cloud4NFV e OpenBaton possuem interface gráfica. De mesma forma, apenas T-NOVA e Cloud4NFV possuem módulos com algoritmos de otimização, que leva vantagem sobre os orquestradores que não possuem, pois, o posicionamento das funções pode ser feito de forma pré-definida, acelerando o processo de implantação. Outra característica importante são os tipos de recursos que estão disponíveis em cada orquestrador, todos possuem recursos de VNFs e SFCs, exceto o OpenBaton. Outra característica dos orquestradores, faz-se pelo VIM, que em todos os orquestradores são suportados por

OpenStack, e apenas o Tacker não suporta uma plataforma alternativa, ficando restrito a rodar em apenas uma plataforma.

Apesar de a divisão de funções sem fio gerar ganhos substanciais para redes sem fio (ABDELWAHAB et al., 2016) e a complexidade no gerenciamento de funções de rádio divididas, os esforços de pesquisa do NFV MANO não consideram a composição de VNFs na tomada de decisões para ambientes sem fio, bem como não elencam os desafios de orquestrar VNFs divididas. Em nenhum dos trabalhos é levado em consideração a composição das VNFs para a otimização, *i.e.*, utilizar a configuração de VNFCs para a tomada de decisão de onde implantar ou manter as VNFs, e desta forma, não atendendo os requisitos exigidos pelos ambientes do tipo *dynamic* C-RAN. Assim, considerando a falta de NFVOs com suporte na orquestração de VNFCs para ambientes *dynamic* C-RAN, neste trabalho é proposta uma solução de orquestração em NFV com a finalidade de orquestrar funções de rádio virtualizadas em ambientes *dynamic* C-RAN.

A solução foi dividida em duas etapas, originando dois trabalhos: Primeiramente, através de um orquestrador conceitual utilizando programação linear, chamado de Maestro, que tem uma prova de conceito apresentada focando nas características funcionais e técnicas de um ambiente *dynamic* C-RAN simulado. Na segunda etapa, um orquestrador de NFV experimental baseado em políticas, com suporte para algoritmos de orquestração personalizáveis, chamado de Orquestra. Nesta última etapa, o orquestrador é analisado em um cenário de rede experimental, considerando o uso de funções de rádio virtualizadas experimentais. No próximo capítulo, apresenta-se a solução de orquestração, suas etapas, e detalha-se sua arquitetura, também discute-se cada um dos seus blocos funcionais envolvidos em suas operações.

### 3 MAESTRO & ORQUESTRA

Neste capítulo é descrita a solução de orquestração para ambientes *dynamic* C-RAN dividida em duas etapas, que expandiu-se através de dois trabalhos. A primeira etapa originou o trabalho chamado de Maestro, um orquestrador de NFV conceitual baseado em um ambiente *dynamic* C-RAN simulado, com foco em características técnicas e funcionais deste tipo de ambiente. A segunda etapa, originou Orquestra, um orquestrador de NFV baseado em um ambiente *dynamic* C-RAN com foco em características experimentais desses tipos de ambiente.

A pesquisa foi conduzida em duas etapas com a finalidade de avaliar a viabilidade e depois realizar experimentação nos ambientes do tipo *dynamic* C-RAN através de NFV. Maestro propôs-se a avaliar a viabilidade da implantação de NFV nos ambientes *dynamic* C-RAN e descobrir as características, lacunas e limitações que esse tipo de ambiente possui ao ser habilitado por NFV, fornecendo uma prova de conceito desta viabilidade. Por outro lado, a segunda etapa, focou na experimentação e características experimentais, bem como nas limitações que esses tipos de ambiente *dynamic* C-RAN podem enfrentar, tanto em divisão de funções, como no ciclo de vida das VNFs implantadas experimentalmente sobre a hierarquia de nuvens.

O diferencial proposto por esta solução de orquestração em relação aos trabalhos anteriores, é a capacidade de posicionar VNFCs em diferentes nuvens, quando as funções virtualizadas possuem mais que uma opção de divisão. Sendo assim, Maestro e Orquestra podem posicionar os VNFCs de várias maneiras diferentes, possibilitando ganhos na transmissão de dados e na maximização do uso dos recursos computacionais nas nuvens.

#### 3.1 Solução Conceitual e Maestro

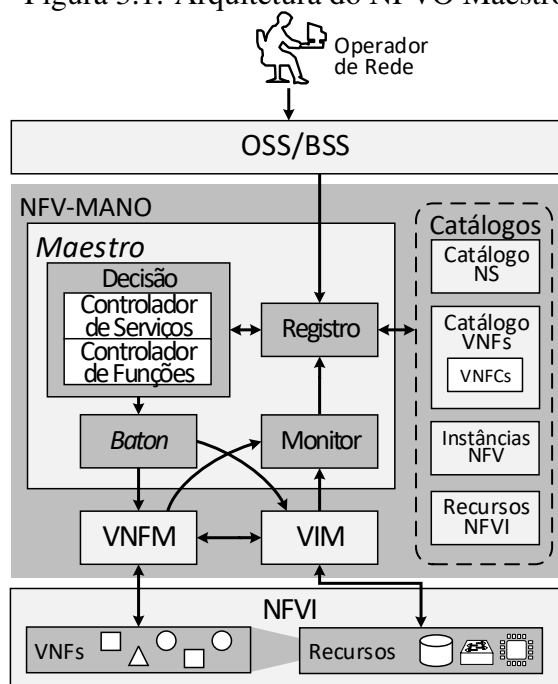
Nesta seção é descrita a primeira etapa da solução de orquestração, chamada de Maestro. Maestro é um orquestrador de NFV conceitual baseado em um ambiente *dynamic* C-RAN simulado, com foco em características técnicas e funcionais deste tipo de ambiente. O diferencial proposto pelo orquestrador Maestro em relação aos trabalhos anteriores, é a capacidade de posicionar VNFCs em diferentes nuvens, quando as funções virtualizadas possuem mais que uma opção de divisão. Sendo assim, o Maestro pode posicionar os VNFCs de várias maneiras diferentes, possibilitando ganhos na transmissão de dados e na maximização do uso dos recursos computacionais nas nuvens de borda.

Desta forma, o orquestrador recebe um conjunto de informações sobre a infraestrutura de rede, definidas pelo operador, e sua utilização, para determinar o posicionamento de cada função virtualizada para todas as BSs dentre as nuvens existentes. Na próxima subseção é descrita a arquitetura do orquestrador e posteriormente, é descrito o modelo de avaliação.

### 3.1.1 Arquitetura do orquestrador

Maestro foi projetado para atender a demanda de NFVOs que utilizam a composição de VNFs para a tomada de decisão de posicionamento, seguindo as recomendações do *framework* NFV MANO do ETSI (QUITTEK et al., 2014). Por esse motivo, Maestro pode operar em qualquer ambiente NFV que segue os padrões da arquitetura do ETSI. A Figura 3.1 descreve em detalhes todos os blocos funcionais que compõem a arquitetura Maestro e sua relação com os blocos funcionais de NFV definidos pelo ETSI.

Figura 3.1: Arquitetura do NFVO Maestro



Em um ambiente NFV, os operadores de rede são responsáveis por selecionar e configurar todos os serviços e funções da rede, respeitando as regras de controle de acesso aplicadas pelo OSS/BSS. Maestro se comunica com OSS/BSS, permitindo que operadores de rede configurem serviços, funções e NFVI. Além disso, alguns blocos funcionais do Maestro também podem ser configurados de acordo com as necessidades dos operadores de rede. Todos os blocos funcionais e de operação do Maestro são descritos a seguir.

**Registro:** De acordo com o framework NFV MANO, um NFVO tem acesso a diferentes catálogos em relação aos serviços e funções de rede disponíveis, como as instâncias atualmente em execução e os recursos disponíveis no NFVI. Esta informação pode ser usada pelo NFVO para realizar ações sobre a operação de serviços e funções. Na arquitetura Maestro, essas informações são tratadas pelo bloco de registro. Qualquer novo serviço ou VNF deve ser registrado nos catálogos para estar disponível para implantação no NFVI.

**Catálogos:** Armazena dados sobre os serviços, funções e recursos disponíveis na rede, como o estado atual das instâncias que operam nele, seguindo o modelo de informação proposto pelo ETSI NFV MANO (QUITTEK et al., 2014). Todos os catálogos estão conectados ao bloco Registro, que controla o acesso a esses dados pelos outros blocos funcionais.

**Decisão:** Determina quais ações serão realizadas na rede, como expandir/diminuir os recursos das VNFs, instanciar uma nova VNF ou alterar uma composição da VNF para outro tipo de divisão. O bloco de decisão é o elemento central da arquitetura Maestro, capaz de operar com diferentes mecanismos de decisão, que podem ser definidos antes de iniciar a operação do Maestro. Cada mecanismo pode considerar parâmetros diferentes para tomar sua decisão, como requisitos de taxa de dados, informações sobre recursos físicos (e.g., CPU e uso de memória), latência, tempo de atraso, tempo de resposta e vazão da rede. Independentemente do mecanismo de decisão utilizado, o bloco de decisão usa informações que residem nos catálogos para decidir as ações a serem tomadas. Essas ações podem afetar a rede em dois níveis diferentes: serviço ou função, por esse motivo, o bloco de decisão é subdividido em dois módulos:

- *Controlador de Serviços:* Responsável por gerenciar as *Service Function Chaining* (SFC), representadas na arquitetura de NFV como *VNF Forwarding Graphs* (VNFFG) (QUINN; GUICHARD, 2014; CHIOSI et al., 2012). Uma vez registrado no catálogo, um SFC pode ser instanciado e modificado pelo mecanismo de decisão implementado no Maestro. Este módulo decide a operação em relação ao ciclo de vida da SFC, como instanciação, validação do estudo de caso e ação, autorização, administração de integridade e visibilidade. Além disso, em um ambiente de NFV regido pelas políticas OSS/BSS sobre a operação da SFC, este módulo é responsável por garantir a aplicação dessas políticas;
- *Controlador de Funções (VNFs):* Responsável pelo gerenciamento das operações relacionadas à VNF e a tomada das decisões com base nas informações disponíveis



no mecanismo de decisão. Uma vez registrado no catálogo, uma VNF será considerada no processo de decisão, seja para a implantação de um novo serviço, ou para a mudança de uma VNF que já está executando no NFVI, de uma composição para uma diferente e que já está em execução.

**Baton:** Uma vez definidas as ações a serem realizadas em relação a serviços, VNFs ou recursos, elas devem ser enviadas para os respectivos blocos. O bloco *Baton* é responsável por receber essas ações do bloco de Decisão, traduzindo-o em configurações e encaminhando para o VIM e/ou VNFM. Esses dois blocos funcionais aplicam as configurações enviadas pelo bloco *Baton* de acordo com sua implementação.

**Monitor:** Monitora e coleta dados sobre operação de VNFs e recursos de NFVI, encaminhando todos os dados coletados para o bloco de registro. O bloco Monitor funciona de duas maneiras: *i*) orientado por eventos, como o mal funcionamento da VNF ou da VM; e *ii*) monitoramento por *polling*, realizando solicitações periódicas para VNFM e VIM obtendo informações como por exemplo, recursos disponíveis, consumo de taxa de dados e latência da rede. As informações relacionadas aos eventos são geradas pelo VNFM e VIM e não precisam da intervenção do bloco Monitor, enquanto o Monitor solicita periodicamente dados no modo de *polling*.

Ambos os blocos *Baton* e Monitor estão conectados aos blocos VIM e VNFM, que são da arquitetura de NFV do ETSI. Esses dois blocos são responsáveis pela operação das VNFs e dos recursos do NFVI, respectivamente. Maestro foi projetado para operar com qualquer implementação do VIM e VNFM que seguirem as recomendações do ETSI para implementações de NFV, tais como Cisco NSO (CISCO... , 2018), OpenStack (BRYANT et al., 2016) e Aurora (WICKBOLDT et al., 2014). Alterar de uma solução para outra, requer apenas a adaptação da camada de abstração nos blocos *Baton* e Monitor. Uma vez introduzida a arquitetura Maestro, na próxima subseção é descrito o modelo de avaliação, e no próximo capítulo descreve-se um estudo de caso e o modelo de avaliação, projetados como uma prova de conceito da operação do Maestro.

### 3.1.2 Modelo de avaliação

Considerando um ambiente *dynamic* C-RAN, contendo um conjunto de  $A$  BSs com um total de  $N$  nuvens (borda, regionais e centrais) e capacidade de  $F$  funções de

rádios virtualizadas, o total de dados transmitidos no *fronthaul* pode ser escrito como:

$$O_B(d) = \sum_{f=0}^F \sum_{n=0}^N \sum_{a=0}^A d_{f,n,a} D_B(f, n, a) \quad (3.1)$$

onde,  $f \in \{0, \dots, F\}$  corresponde as funções de rádio virtualizadas,  $n \in \{0, \dots, N\}$  representa as nuvens disponíveis para processamento dessas funções e  $a \in \{0, \dots, A\}$  simboliza as BSs correspondentes a uma *Dynamic C-RAN*. A matriz tridimensional  $d = [d_{f,n,a}]_{F \times N \times A}$  representa um posicionamento factível para as funções de rádio virtualizadas em cada BSs entre as nuvens. A matriz de posicionamento  $d_{f,n,a} \in \{0, 1\}$ , *i.e.*,  $d_{f,n,a} = 1$  indica que  $f$  foi posicionada sobre  $n$  para processar o sinal de  $a$ . Caso contrário, *i.e.*,  $d_{f,n,a} = 0$ , representa que  $f$  da BS  $a$  não está sendo processada em  $n$ . Finalmente,  $D_B(f, n, a)$  determina a taxa de dados necessária para processar  $f$  da BS  $a$  em uma determinada nuvem  $n$ . A equação  $D_B(f, n, a)$  considera o tipo de  $f$  a ser processada, a distância da nuvem  $n$  (*i.e.*, borda, regionais ou centrais) e a demanda de UEs na BS  $a$ , para determinar a quantidade de dados transmitidos no *fronthaul*. Maiores informações sobre o cálculo da quantidade de dados transmitidos, podem ser encontrados em Wubben, et al. (WUBBEN et al., 2014).

Baseado nas definições acima, o problema de minimização do total de dados transmitidos no *fronthaul*, através do posicionamento dinâmico das funções de rádio virtualizadas e suas restrições pode ser formulada como:

$$\min_{\{d\}} O_B(d) = \min_{\{d\}} \sum_{f,n,a} d_{f,n,a} D_B(f, n, a) \quad (3.2)$$

s.t.

$$\sum_n d_{f,n,a} = 1, \quad \forall f, \forall a, \quad (3.3)$$

$$\sum_{f,a} d_{f,n,a} D_V(f, n, a) \leq C_V^n, \quad \forall n, \quad (3.4)$$

$$\sum_{f,a} d_{f,n,a} D_B(f, n, a) \leq C_B^n, \quad \forall n, \quad (3.5)$$

onde,  $D_V(f, n, a)$  determina o consumo dos recursos de nuvem necessários para processar  $f$  da BS  $a$  em uma determinada nuvem  $n$ .  $C_V^n$  e  $C_B^n$  representam os limites de VDUs e de banda do *fronthaul* da nuvem  $n$ , respectivamente.

As restrições (3.3), (3.4) e (3.5) garantem a factibilidade da solução perante o ambiente de uma *Dynamic C-RAN*. A restrição (3.3) garante que para toda e qualquer BS

$a \in A$  exista obrigatoriamente uma função de rádio virtualizada  $f \in F$ . Além disso, a restrição garante que nenhuma  $f$  de uma mesma  $a$  seja duplicada dentre todas as nuvens possíveis  $n \in N$ . A restrição (3.4) garante que o posicionamento de uma função de rádio virtualizada  $f$  de uma BS  $a$  seja realizada somente em uma nuvem  $n$ , com capacidade de processamento, *i.e.*, a nuvem deve conter VDUs suficientes para atender todas as funções atribuídas. Por fim, a restrição (3.5) garante que o posicionamento de uma função  $f$  de uma BS  $a$  ocorra obrigatoriamente em uma nuvem  $n$  com capacidade de transmissão, *i.e.*, uma nuvem só pode receber  $f$  caso o *fronthaul* possua banda suficiente para atender os requisitos de transmissão de todas as funções virtualizadas nessa nuvem.

A principal função do *Maestro* é minimizar a taxa de dados transmitidos no *fronthaul*, adotando o modelo descrito na subseção 2.1. Para acelerar a resolução do problema linear, um novo elemento objetivo é inserido em (3.2), para maximizar a quantidade de recursos utilizados nas nuvens de borda, uma vez que estas nuvens não transmitem dados no *fronthaul* e devem ser priorizadas pelas soluções. Desta maneira, insere-se uma ponderação na função objetivo, afim de atingir os objetivos propostos. Tal minimização ponderada é descrita como:

$$\begin{aligned} \min_{\{\mathbf{d}\}} \quad & \omega \cdot \sum_{f,n,a} d_{f,n,a} D_B(f, n, a) + (\omega - 1) \cdot \sum_{f,a} d_{f,0,a} D_V(f, n, a) \quad (3.6) \\ \text{s.t.} \quad & (3)(4)(5) \end{aligned}$$

onde,  $0 \leq \omega \leq 1$  realiza a ponderação das funções objetivos para resolução do problema e  $d_{f,0,a} D_V(f, n, a)$  da prioridade das nuvens de borda no posicionamento das funções virtualizadas. A segunda parte de (3.6) tem por finalidade maximizar a ocupação de recursos nas nuvens de borda e acelerar a convergência para a solução. A maximização das nuvens de borda e a minimização da taxa de dados transmitidos no *fronthaul* fazem com que os algoritmos consigam obter um melhor desempenho quanto ao tempo, através do direcionamento das possibilidades de posicionamento das VNFs, durante sua execução. Nesta avaliação, foram utilizados 0,7 para o valor de  $\omega$ . Este valor foi definido empiricamente baseado no tempo de processamento do algoritmo e no tempo para a convergência da solução.

O *Maestro* garante que os componentes de funções de rádio não sejam implantados na mesma nuvem para a mesma BS, que não se exceda a capacidade computacional das nuvens e a capacidade de transmissão disponível no *fronthaul*, respeitando as restrições descritas anteriormente. O *Maestro* foi implementado utilizando o *IBM CPLEX* (IBM

Software, 2010), com a função objetivo descrita em (3.6).

A solução de orquestração foi dividida em duas etapas. Maestro é a primeira parte da solução de orquestração, avaliando os aspectos técnicos e funcionais de um ambiente simulado, como descrito anteriormente. Através deste trabalho, é possível observar a viabilidade da implantação de NFV por parte dos ambientes *dynamic* C-RAN, utilizando uma prova de conceito para validar o posicionamento das VNFs e VNFCs ao longo da infraestrutura, atendendo os requisitos exigidos por *dynamic* C-RAN. A segunda etapa, baseada nas avaliações efetuadas na primeira parte da solução, expandiu a pesquisa para um orquestrador NFV aplicado aos ambientes *dynamic* C-RAN experimentais, chamado de Orquestra, e que é descrito na próxima seção.

### 3.2 Solução Experimental e Orquestra

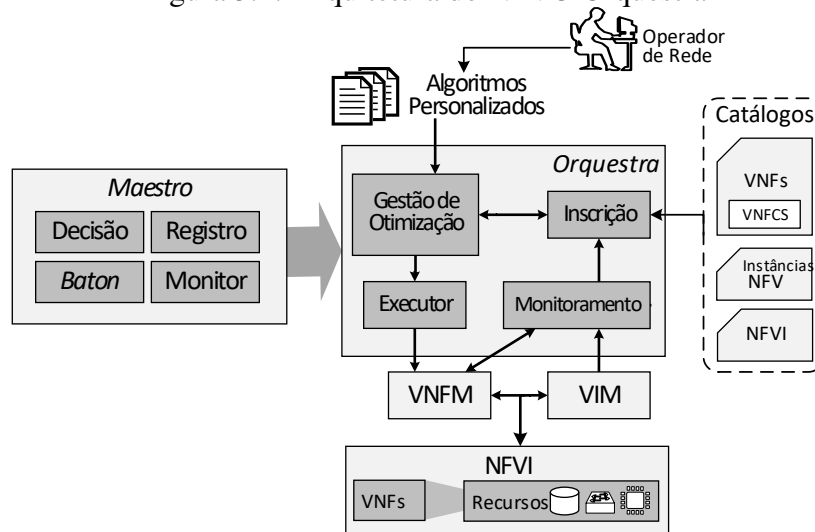
Nesta seção é descrito Orquestra, uma solução de orquestração de NFV baseada em um ambiente *dynamic* C-RAN experimental, com foco em personalização de algoritmos do operador da rede. Orquestra é uma evolução em relação ao trabalho anterior Maestro, que focava em um ambiente simulado, e Orquestra, foca em ambientes *dynamic* C-RAN experimental. Portanto, tanto arquitetura como estudos de casos, são conduzidos como uma evolução do Maestro para o Orquestra. O diferencial deste orquestrador para os trabalhos relacionados, é que permite a manipulação de VNFCs de forma personalizada, permitindo que possam ser posicionados em diferentes nuvens. Desta forma, é possível que as funções sejam gerenciadas e orquestradas para atingirem algum objetivo específico definido pelo operador da rede, tal como a redução da quantidade de taxa de dados no *fronthaul* ou balanceamento de carga entre as nuvens que processam os VNFCs. Na próxima seção é descrita a arquitetura do orquestrador, e posteriormente é descrito um exemplo de algoritmo de orquestração.

#### 3.2.1 Arquitetura do orquestrador

Orquestra foi proposto para cobrir a falta de NFVOs que permitem gerenciar VNFs através de sua composição, suportando os requisitos de orquestração dessas VNFs que podem ser divididas de várias formas em diversos VNFCs. A diferença do Orquestra para os trabalhos relacionados é que Orquestra possui a granularidade necessária para orques-

trar VNFCs, enquanto que os trabalhos relacionados, tratam as VNFs como elementos atômicos.

Figura 3.2: Arquitetura do NFVO Orquestra



A Figura 3.2 mostra em detalhes todos os blocos funcionais que compõem a arquitetura Orquestra e sua relação com os blocos da arquitetura de NFV definidos pelo ETSI. Partes da arquitetura do Orquestra foram herdadas do Maestro e tiveram algumas modificações para operar em ambiente experimental. Esses blocos foram reimplementados ou redefinidos de acordo com a necessidade de operação de cada um deles. Desta forma, alguns blocos tiveram também um incremento de funcionalidades e transformaram-se em outros. A viabilidade do Maestro habilitou o Orquestra a herdar algumas das funcionalidades e a lógica de modularização entre os módulos que compõem o orquestrador. Uma diferença em relação ao Maestro para o Orquestra é a inclusão dos algoritmos personalizados do operador de rede com a finalidade de otimizar ou apenas invocar uma funcionalidade do orquestrador.

Ao lado esquerdo da Figura pode-se notar os blocos disponíveis no Maestro, *i.e.*, Decisão, *Baton*, Registro e Montitor, e, no centro, os blocos disponíveis em orquestra, que são Gestão de otimização, Executor, Inscrição e Monitoramento. Desta forma, Orquestra possui as mesmas características do Maestro quanto a divisão de funções de cada bloco, porém, Orquestra é implementado de forma a viabilizar totalmente a experimentação, e esses blocos, são equivalentes para Maestro na etapa conceitual e Orquestra na etapa experimental .

O operador de rede interage com Orquestra usando OSS/BSS. Isso permite que o operador de rede desenvolva, crie e registre VNFs e serviços. Além disso, outra maneira

que o operador de rede interage com o Orquestra é escrevendo algoritmos personalizados, para o mecanismo de otimização. Desta forma, permite adaptar e escrever algoritmos otimizados para provisionamento e posicionamento de VNFs e serviços de rede, de acordo com o necessário em cada tipo de infraestrutura, neste caso, *dynamic* C-RAN. Esses algoritmos podem ser escritos na linguagem de programação Python. Com base nos blocos definidos anteriormente na Seção 3.1.1, todos os blocos funcionais do Orquestra estão descritos abaixo.

**Inscrição:** De mesma forma que presente em Maestro, a arquitetura possui um bloco que é focado na inscrição e no registro da orquestração NFV. O bloco inscrição acessa diferentes catálogos de serviços e funções de rede, objetivando principalmente manter a informação sobre o estado dos recursos disponíveis e das instâncias em execução, que estão rodando sobre o NFVI. Este bloco implementa um banco de dados para guardar todas as instâncias e todos os estados das VNFs. Outra função do bloco de inscrição é armazenar informação sobre o monitoramento, provendo uma interface entre dados e coletores, (*e.g.*, *sFlow collector*). Essas informações são utilizadas para implantar novas funções e serviços, bem como, gerenciar funções e serviços de rede já disponíveis.

**Catálogos:** Este módulo tem a finalidade de armazenar e disponibilizar os dados relativos ao catálogo dos serviços, catálogos de VNFs, encadeamento dos VNFCs, recursos NFVI disponíveis na rede, tal como o status dos catálogos. Além disso, também armazena os dados de instâncias de VNFs que estão em execução, podendo em uma eventual falha geral, fornecer informações que ajudem o operador de rede recuperar o funcionamento das VNFs. O catálogo está conectado ao bloco de inscrição, que fornece suporte para disponibilizar e acessar essas informações através de uma API.

**Gestão de otimização:** Diferente do que foi proposto no Maestro, este novo bloco tem por finalidade tomar decisões sobre os VNFCs e VNFs através dos algoritmos escritos pelo operador da rede. Além de decidir as ações, também tem por finalidade disponibilizar informações aos algoritmos do operador da rede (*e.g.*, CPU, uso de memória, uso de disco, latência, vazão e tempo de atraso da rede), que são do VNFM, VIM e NFVI. O bloco de gestão de otimização recebe como entrada algoritmos escritos pelo operador da rede ou fornecedores de software. Desta forma, este bloco executa ações dado as análises que podem ser personalizadas para atingir algum objetivo, definido pelo operador da rede. O bloco que era conhecido como decisão no Maestro, passa a ser chamado de gestão de otimização no Orquestra. No Maestro, este bloco representava um algoritmo linear, que tinha por finalidade fazer o posicionamento dos VNFCs ao longo da infraestrutura, e, no

Orquestra, este algoritmo é personalizável de acordo com o operador de rede.

A partir de uma API para a linguagem de programação *Python*, Orquestra suporta a delegação de ações a serem tomadas a partir dos algoritmos personalizados. Por exemplo, após o algoritmo do operador da rede encontrar um gargalo de uso de CPU em alguma das nuvens, solicita a API de Orquestra que faça uma migração de um VNFC, e neste caso, o orquestrador se encarrega de migrar e reposicionar o VNFC, migrando e reestabelecendo o serviço, e comunicação entre ambos VDUs. Cada mecanismo pode considerar parâmetros diferentes para tomar sua decisão, e desta forma, cabe ao operador da rede definir quais serão os parâmetros interessados pelo algoritmo.

Orquestra não implementa os dois controles diferentes para posicionamento de VNFs e SFC como realizado no Maestro. A partir dos algoritmos personalizados, o operador da rede pode decidir se é necessário orquestrar apenas um tipo de serviço, ou ambos. Esses algoritmos são focados em políticas definidas pelo operador da própria rede, como balanceamento de carga, ampliação/diminuição de recursos (*e.g.*, aumentar ou diminuir a quantidade de memória em determinada nuvem, com a finalidade de otimizar o desempenho ou economizar energia), e a alteração da composição das VNFs.

**Executor:** Tal como no Maestro, no Orquestra há um bloco responsável por traduzir os comandos para as plataformas das VNFs específicas, ou seja, ele tem por finalidade aplicar no VNFM e VIM os procedimentos que foram gerados no bloco de gestão de otimização. Além dessas ações realizadas e citadas anteriormente, ele também tem por finalidade executar configurações no VIM e VNFM. Assim que os procedimentos e ações forem definidas no bloco de gestão de otimização, este bloco é responsável por receber os procedimentos e ações e traduzir para a API nativa de cada plataforma do VIM e VNFM, com a finalidade de aplicar essas modificações neles.

**Monitoramento:** Monitora e coleta dados sobre VNFs, bem como recursos do NFVI, encaminhando todos os dados para o bloco de registro. Este bloco é implementado usando o *Simple Network Management Protocol* (SNMP) com a finalidade de monitorar o status das nuvens, como por exemplo, parâmetros de CPU, uso de memória e uso de disco. Já para monitoramento de rede, é utilizado *Simple Flow* (sFlow) e OpenFlow, que têm por finalidade entregar os fluxos que são gerados entre nuvens, entre os VNFCs além de outros parâmetros de rede, como vazão. No monitoramento dos *containers* Linux (LXD), as informações dos *containers*, o bloco monitor coleta dados usando a API do VNFM. O monitoramento é feito através de *polling*, realizando solicitações periódicas ao bloco aos VNFMs e ao VIM, com a finalidade de ter a maior quantidade de dados atualizados em

menor tempo, para que os dados sejam fornecidos ao mecanismo de otimização.

Assim como na arquitetura de Maestro, no Orquestra, ambos os blocos *baton* e monitor estão conectados ao VIM e VNFM, que são responsáveis pela operação das VNFs e dos recursos disponíveis do NFVI. Maestro pode operar quaisquer VIM e VNFM, desde que seja adaptada uma camada de abstração, que está presente nos blocos *baton* e monitor.

Orquestra tem a finalidade de controlar essas funções de rádio transformadas em VNFCs ao longo das nuvens, e como citado no modelo ESTI NFV MANO, a ligação entre esses VNFCs, são os VDUs. Para cada tipo de infraestrutura e rede, o operador pode escrever algoritmos personalizados, ou, obter esses algoritmos de fornecedores de software. Desta maneira, a próxima seção descreve dois algoritmos como prova de conceito da orquestração, desenvolvidos pelo operador da rede, um deles projetado para efetuar balanceamento de carga através de migrações, e outro para realizar a implantação dos VNFCs em nuvens com recursos disponíveis.

### 3.2.2 Exemplo de algoritmo de orquestração

Orquestra é um orquestrador de NFV com suporte a gerenciamento de divisão de VNFs através dos VNFCs. Uma das entradas efetuadas pelo operador da rede e também descritos na Figura 3.2 são os algoritmos personalizados com a finalidade de atingir objetivo definido pelo operador da rede ou fornecedores de software. Desta forma, o algoritmo pode adaptar a implantação, remoção ou realocação de VNFCs e conseqüentemente de VNFs, tal como realizar o SFC se necessário.

Tabela 3.1: Funções disponibilizadas pelo orquestrador ao operador de rede

Nome da função	Funcionalidade
IMPLANTA_VNF(nome_vnf, lista_vnfcs)	Implanta a VNF e VNFCs sobre múltiplas nuvens
EXCLUI_VNF(nome_vnf)	Remove completamente a VNF de todas as nuvens
MIGRA_VNFC(nuvem_origem, nome_vnfc, nuvem_destino)	Migra o VNFC da nuvem origem para destino
LISTA_NUVENS()	Retorna a lista de nuvens cadastradas na infraestrutura
INFORMA_NUVENS(tipo_recurso)	Retorna informações sobre cada uma das nuvens (e.g., memória, CPU)
INFORMA_VNFCs()	Retorna informações sobre onde está cada VNF/VNFC
INFORMA_FLUXOS()	Retorna informações referentes ao tráfego e taxa de dados entre VNFC e nuvens



---

**Algoritmo 1** Algoritmo de balanceamento de carga entre nuvem de borda, regional e central
 

---

```

1:  $limiar \leftarrow 95$ 
2:  $N \leftarrow LISTA\_NUVENS()$ 
3:  $CPU \leftarrow INFORMA\_NUVENS(cpu)$ 
4:  $VNFCs \leftarrow INFORMA\_VNFCs()$ 
5: para cada  $c$  em  $N$  faça
6:   se  $CPU[c] > limiar$  entao
7:      $disp[c] \leftarrow (100 \ominus CPU[c])$ 
8:     para cada  $s$  em  $disp$  faça
9:        $val \leftarrow disp[s]$ 
10:      se  $val > (100 - limiar)$  entao
11:         $origem \leftarrow NENHUM$ 
12:        se  $c = central$  entao
13:          se  $s = edge$  entao
14:             $origem \leftarrow regional$ 
15:          senao se  $s = regional$  entao
16:             $origem \leftarrow central$ 
17:          fim se
18:        senao se  $c = regional$  entao
19:          se  $s = central$  entao
20:             $origem \leftarrow borda$ 
21:          senao se  $s = borda$  entao
22:             $origem \leftarrow central$ 
23:          fim se
24:        senao se  $c = borda$  entao
25:          se  $s = regional$  entao
26:             $origem \leftarrow borda$ 
27:          senao se  $s = central$  entao
28:             $origem \leftarrow regional$ 
29:          fim se
30:        fim se
31:      se  $origem = NENHUM$  entao
32:        escreva “Nao existem nuvens disponíveis para migração”
33:        retorne falso
34:      fim se
35:       $org \leftarrow origem$ 
36:       $dst \leftarrow destino$ 
37:      para cada  $cnt$  em  $VNFCs[org]$  faça
38:        retorne  $MIGRAR\_VNFC(org, dst, cnt)$ 
39:      fim para
40:    fim se
41:  fim para
42: fim se
43: fim para

```

---

A Tabela 3.1 descreve os métodos e funções que são disponibilizadas pelo or-

questrador ao operador de rede na escrita dos algoritmos personalizados. Na coluna ao lado esquerdo, são mostrados os métodos e parâmetros de cada uma das funções e métodos. Ao lado direito, há uma descrição da funcionalidade que cada uma delas objetiva. Como por exemplo, na primeira linha da tabela a função `IMPLANTA_VNF` tem a finalidade de implantar uma VNF com VNFCs em uma infraestrutura de nuvem, podendo implantar cada VNFC em uma nuvem diferente ou todos eles na mesma, dependendo dos parâmetros passados ao chamar a função. Seguindo o ciclo de vida das VNFs, a função `EXCLUI_VNF` tem por finalidade excluir a VNF e todas suas VNFCs. Já a função `MIGRA_VNFC` tem por finalidade migrar os VNFCs *i.e.*, funções de rádio, de uma nuvem para outra. As funções `LISTA_NUVENS`, `INFORMA_NUVENS`, `INFORMA_VNFCs` e `INFORMA_FLUXOS` tem por finalidade retornar dados referentes ao bloco de registro e ao monitoramento de rede.

A primeira versão do orquestrador suporta diretamente uma API escrita em linguagem *Python*, suportando algoritmos também nesta mesma linguagem. Para implementar outras linguagens de programação, faz-se necessário escrever uma camada de abstração da linguagem *Python* para as outras, *i.e.*, qualquer outra ferramenta desenvolvida em outra linguagem de programação poderia utilizar o orquestrador.

Outra possibilidade oferecida pelo Orquestra é mover os algoritmos diretamente para a máquina local do operador de rede, e, importar as bibliotecas que o Orquestra disponibiliza, também na linguagem *Python*. Essa opção permite que o operador da rede execute os algoritmos diretamente de sua máquina local, consumindo os recursos das bibliotecas disponibilizadas pelo Orquestra. Neste caso, Orquestra torna-se uma camada de abstração entre o VIM e VNFM e, a partir das chamadas feitas pelo algoritmo do operador da rede, faz a aplicação na infraestrutura gerenciada por ele, abstraindo todo processo do operador de rede. Esta última opção não contempla o carregamento automático de algoritmos, como descrita anteriormente

O Pseudocódigo 1 exemplifica um simples algoritmo focado em migração de VNFCs quando uma nuvem estiver sobrecarregada em uso de CPU, com decisão baseada em limiar. Este limiar definido pelo operador da rede. Na parte superior, no início do algoritmo, na primeira linha, define-se o *limiar*, que tem por finalidade suportar o processo de decisão e definir o quão necessário é a tomada de uma decisão. Este limiar representa a quantidade máxima de ocupação de CPU que é aceita em cada nuvem, e desta forma, define se um servidor está com ocupação de CPU alta ou não, e a partir disso, desencadeia a execução de ações subsequentes. Também são definidos na segunda e terceira linhas os

conjuntos  $N$ , que é a lista de nuvens, e  $CPU$ , que armazena as informações sobre o uso de CPU das nuvens respectivamente.

Seguindo o pseudocódigo, na quarta linha,  $VNFCs$  armazena informações sobre os VNFCs, como por exemplo o consumo de CPU e memória em cada um deles. Na quinta linha é percorrida a lista de nuvens disponíveis  $N$ . A partir disso, na sexta linha é verificado para cada nuvem se a ocupação do CPU dela é maior que o limiar. Se for maior que o limiar, então a sétima linha calcula a disponibilidade de processamento por nuvem, e desta forma, em caso das nuvens percorre-se uma análise de todas as nuvens já analisadas, incluindo a atual, e verifica-se se há disponibilidade em alguma delas, descrito na nona linha por  $val$ . Na décima linha, é verificado se o valor disponível é maior do que a diferença do limiar para o total, *i.e.*, se há disponibilidade na nuvem analisada, e caso tenha, então a partir da décima primeira linha é analisado de qual origem e para qual destino os VNFCs serão migrados, baseado no tipo da origem e tipo de nuvem que está acima do limiar. O código para verificação se estende até a trigésima linha. Então, se a origem tiver recursos disponíveis, os VNFCs são migrados da nuvem origem para a nuvem destino, em ordem de instanciação.

No caso de não haver recursos disponíveis, uma mensagem é mostrada ao operador da rede, que deve tomar uma decisão e migrá-los manualmente, assim como descrito da trigésima primeira para a trigésima quarta linha. De mesma forma, para continuar mantendo o serviço, o algoritmo poderia observar dados de fluxos e, conseqüentemente, tomar decisões diferentes, como parar um VNFC ou a própria VNF.

Para exemplificar a implantação das VNFs ao longo da infraestrutura, o Pseudocódigo 2 descreve uma implantação de VNFs baseado na quantidade de  $CPU$  disponível em cada nuvem. Este pseudocódigo visa maximizar o processamento das nuvens, hospedando a maior quantidade de VNFCs distribuídas. Para isto, o algoritmo recebe como entrada a quantidade de BSs a serem implantadas juntamente com o *limiar*, que define a quantidade máxima de processamento que cada nuvem pode atingir. Na terceira linha é efetuado um laço para que todas as BSs sejam implantadas, e desta forma, nas próximas linhas verificam-se se alguma das nuvens suporta a implantação de VNFCs. Esta suportando, então são implantados os VNFCs nas nuvens, podendo ser implantados em 1, 2 ou 3 nuvens diferentes, conforme a disponibilidade de processamento. Finalmente, na penúltima linha é efetuado de fato a implantação da VNF, chamando a função `IMPLANTA_VNF` e passando de parâmetros a configuração de VNFC para cada nuvem, descritas anteriormente.

---

**Algoritmo 2** Algoritmo exemplo de implantação baseado em limiar
 

---

```

1:  $limiar \leftarrow 99$ 
2:  $totalBS \leftarrow 12$ 
3:  $i \leftarrow 0$ 
4: enquanto  $i < totalBS$  faça
5:    $parametros \leftarrow \emptyset$ 
6:    $CPU \leftarrow \text{INFORMA\_NUVENS}(cpu)$ 
7:   se  $CPU[borda] \leq limiar$  entao
8:      $parametros \leftarrow parametros + borda : VNFC1$ 
9:      $parametros \leftarrow parametros + borda : VNFC2$ 
10:     $parametros \leftarrow parametros + borda : VNFC3$ 
11:     $parametros \leftarrow parametros + regional : VNFC4$ 
12:    senao se  $CPU[regional] \leq limiar$  entao
13:       $parametros \leftarrow parametros + regional : VNFC1$ 
14:       $parametros \leftarrow parametros + regional : VNFC2$ 
15:       $parametros \leftarrow parametros + regional : VNFC3$ 
16:       $parametros \leftarrow parametros + regional : VNFC4$ 
17:    senao
18:       $parametros \leftarrow parametros + central : VNFC1$ 
19:       $parametros \leftarrow parametros + central : VNFC2$ 
20:       $parametros \leftarrow parametros + central : VNFC3$ 
21:       $parametros \leftarrow parametros + central : VNFC4$ 
22:    fim se
23:     $parametros \leftarrow parametros + central : VNFC5$ 
24:     $\text{IMPLANTA\_VNF}(BS' + i, parametros)$ 
25: fim enquanto

```

---

Para avaliar o desempenho da solução de orquestração proposta, no próximo capítulo são descritos os estudos de caso para os ambientes simulados e experimentais, Maestro e Orquestra respectivamente. Desta forma, o estudo de caso contempla a primeira fase do trabalho, a simulada, e posteriormente, a fase experimental, discutindo os resultados encontrados em cada uma dessas fases.

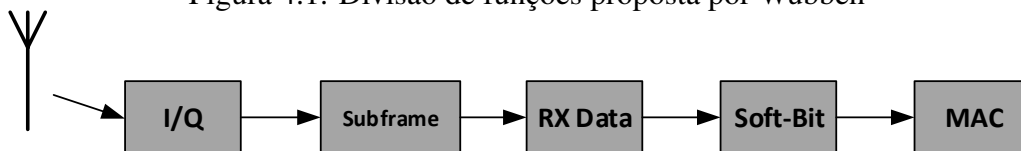
## 4 AVALIAÇÃO

Neste capítulo são descritos os estudos de caso utilizados para avaliação dos orquestradores Maestro e Orquestra, ambos conduzidos em ambientes do tipo *dynamic C-RAN*. Os estudos de caso foram conduzidos em duas etapas, sendo a primeira em um ambiente simulado com foco nas características técnicas e funcionais de *dynamic C-RAN*, e a segunda, em um ambiente experimental. Ambos os estudos de caso são descritos a seguir.

### 4.1 Estudo de caso do orquestrador conceitual

Este estudo de caso é baseado na divisão de funções de rádio sem fio apresentada por Wubben *et al.* (WUBBEN et al., 2014), aplicada ao contexto de *dynamic C-RAN*. Esta forma de divisão é interessante neste contexto, devido a variabilidade da demanda de taxa de dados no *fronthaul*, imposta por cada opção de divisão, sendo dividida em 5 funções: *Media Access Control* (MAC), *Soft-Bit*, *RX Data*, *Subframe* e *In-phase and Quadrature* (I/Q).

Figura 4.1: Divisão de funções proposta por Wubben

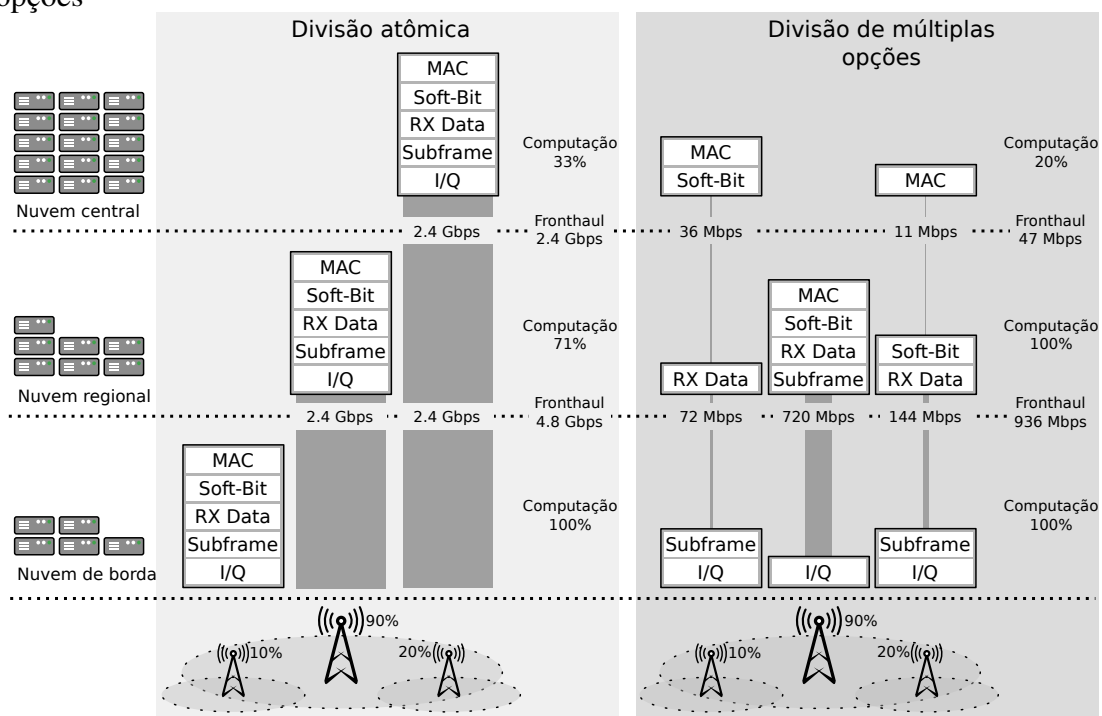


A Figura 4.1 esboça a conexão entre as 5 funções propostas por Wubben através do no *uplink* da BS. Ao lado esquerdo, o sinal é recebido em tempo domínio da RRH e transformado de analógico para digital, como mostrado ao lado esquerdo da Figura. Então, I/Q tem a finalidade de encaminhar esse sinal para a BBU, incluindo o recebimento do *frame* de Prefixo Cíclico (CP). Após, a função de Subframe remove o CP e transforma o sinal Rx para o domínio frequência, usando a Transformada de Fourier Rápida (FFT). RX Data é somente uma parte utilizada pelos User Equipments (UE) e tem por finalidade balancear a carga de transferência de dados para ser conduzida até a nuvem. Na Soft-Bit, como o processamento por usuário consiste na equalização da frequência domínio, este bloco tem por finalidade combinar o sinal de múltiplas antenas durante a equalização do canal, abstraindo a quantidade de antenas do canal e removendo a dependência no números de antenas recebidas pela nuvem. Já durante a decodificação do *Forward Error*

*Correction* (FEC) são removidos os símbolos redundantes, resultando em um pacote com apenas a carga útil, chamado de MAC, que são dados como saídas do decodificador. É um processo complexo e que depende da modulação em que foi escrito. Comumente pode ser encontrado em implementação de hardware, mas atualmente há uma alternativa para a transformação dele em software, e por isso, está presente neste trabalho (WUBBEN et al., 2014).

Este estudo de caso nos permite demonstrar o uso de elementos do NFV MANO (VNFs, VDUs e VNFCs) na orquestração de *dynamic* C-RAN, e como diferentes divisões podem afetar o desempenho da rede e o uso de recursos. Por esse motivo, as divisões utilizadas são baseadas nos requisitos de comunicação e computação das funções que compõem as funções de rádio sem fio.

Figura 4.2: Exemplo de implantação de funções de forma atômica e, divisão de múltiplas opções



A Figura 4.2 retrata o estudo de caso, mostrando os benefícios da implantação de funções de forma dividida, contra a divisão atômica, conforme proposto por Wubben *et al.* (WUBBEN et al., 2014). Além de propor as divisões, os autores definiram requisitos de comunicação (*i.e.*, taxa de dados) entre cada uma das funções, e que, no nosso caso, o NFVO deve considerar no processo de decisão. As BSs são retratadas na parte inferior da Figura 4.2, cada uma com ocupação diferente. A macro célula (*b*) apresenta uma ocupação de 90%, enquanto a célula pequena (*a* e *c*) são 10% e 20% ocupados, res-

pectivamente. A carga de trabalho das BSs é distribuída em três níveis de nuvem: (i) de borda, perto das BSs; (ii) regional, um pouco longe das BSs; e (iii) central, ainda mais longe das BSs, geralmente nas instalações do provedor de serviços. Cada nível de nuvem tem uma capacidade de processamento diferente: quanto mais longe das BSs, maior é a capacidade de computação. Neste estudo de caso, a capacidade computacional da nuvem é medida em relação às unidades de computação. A nuvem de borda fornece 5 unidades de computação, enquanto as nuvens regionais e centrais fornecem 7 e 15 unidades de computação, respectivamente. Como o objetivo deste estudo de caso é analisar o cumprimento dos requisitos de rede, simplificou-se a análise computacional, considerando que cada componente virtualizado (*i.e.*, VNFC) consome uma unidade de computação, apenas. Isso significa que as VNFCs com maior requisitos de processador preencherão uma unidade de computação inteira, enquanto as VNFCs com menos requisitos de processador também terão uma unidade de computação dedicada, que será subutilizada.

Assumindo o exemplo em que o objetivo é reduzir a taxa de dados *fronthaul*, o NFVO não pode melhorar a alocação mostrada no lado esquerdo da Figura 4.2, uma vez que os 5 VNFCs devem ser movidos e alocados em conjunto na mesma nuvem. Nesse tipo de implantação, o consumo de taxa de dados agregado é de 4.8 Gbps entre a nuvem de borda e a nuvem regional. A implantação atômica também é subótima neste caso, pois acaba subutilizando os recursos computacionais disponíveis em relação às unidades de computação em uso, onde apenas 5 das 7 unidades de computação disponíveis na nuvem regional são alocados (*i.e.*, 71% de uso de recursos de computacionais). Sem mais opções de divisão de VNFs, ou seja, VDUs, não há nenhum VNFC capaz de usar as 2 unidades de computação restantes.

Na implantação com divisão de múltiplas opções, no entanto, as alocações mudam drasticamente. Para otimizar a taxa de dados no *fronthaul*, o NFVO pode priorizar combinações de divisões com menor consumo de taxa de dados agregado no *fronthaul*, considerando a capacidade computacional máxima das nuvens de borda e regional. Uma vantagem desta abordagem é o uso de todos os recursos computacionais na nuvem regional, o que não foi possível na implantação atômica, devido à indivisibilidade dos componentes da mesma VNF. A otimização do consumo de taxa de dados *fronthaul* é um exemplo do objetivo do NFVO. Usando Maestro, os operadores de rede podem definir algoritmos de decisão com objetivos diferentes.

No estudo de caso apresentado, considera-se que a ocupação BS influencia o requisito de consumo de taxa de dados de funções de *RX Data* para acima, conforme descrito

por Wubben *et al.* (WUBBEN et al., 2014). Como consequência, as VDUs que contêm funções de *Subframe* e *I/Q* foram implantadas mais próximas das BS, uma vez que os requisitos de taxa de dados são mais elevados. Por exemplo, o NFVO pode selecionar qualquer função a ser implantada na nuvem central. No entanto, atribuir as funções de BS  $b$  na nuvem central consumirá muito mais taxa de dados, uma vez que a BS  $b$  apresenta maior ocupação do que as outras, de modo que o requisito de taxa de dados também é superior a  $a$  e  $c$ .

Em resumo, o estudo de caso discutido nesta subseção permite observar duas vantagens principais da implantação de divisão de múltiplas opções quando comparada a implantação atômica: *i*) a fragmentação de recursos de computação é reduzida por causa da alocação flexível de diferentes composições e, *ii*) o consumo de taxa de dados no *fronthaul* é reduzido quando são considerados os requisitos específicos de cada divisão, embora a mesma quantidade de carga de trabalho de processamento de sinal (mesma ocupação de BS) tenha sido usada em ambos os cenários de implantação. Com base nessas vantagens, e como primeira parte da solução de orquestração, foi conduzida uma avaliação como um problema de programação linear, explicado em detalhes no próximo capítulo. Na próxima subseção é descrito a evolução da primeira parte do estudo de caso, contemplando a segunda parte, onde a solução de orquestração foi implantada em um ambiente experimental.

## 4.2 Estudo de caso do orquestrador experimental

Neste estudo de caso, baseia-se nas funções de rádio sem fio disponibilizadas pelo projeto wishful (WISHFUL, 2018; KIRST et al., 2018), aplicando-se ao contexto de *dynamic* C-RAN. Essas funções permitem que um cenário experimental *dynamic* C-RAN seja projetado, e cada função de rádio passa a ser um VNFC, implementado sobre um *container* Linux do tipo LXD<sup>1</sup>. A etapa conceitual e experimental possuem uma diferença na forma da divisão das VNFCs e propriamente as funções que compõem as BSs devido a limitações tecnológicas da parte experimental. Desta forma, a etapa experimental trabalha com funções disponibilizadas experimentalmente, mas que podem por ora não refletir a granularidade que é objetivada e que pode ser alcançada de fato por *dynamic* C-RAN.

A Figura 4.3 mostra como são implementadas as funções de rádio através dos VNFCs. Para *downlink* da BS, as funções de rádio são divididas em três VNFCs, sendo

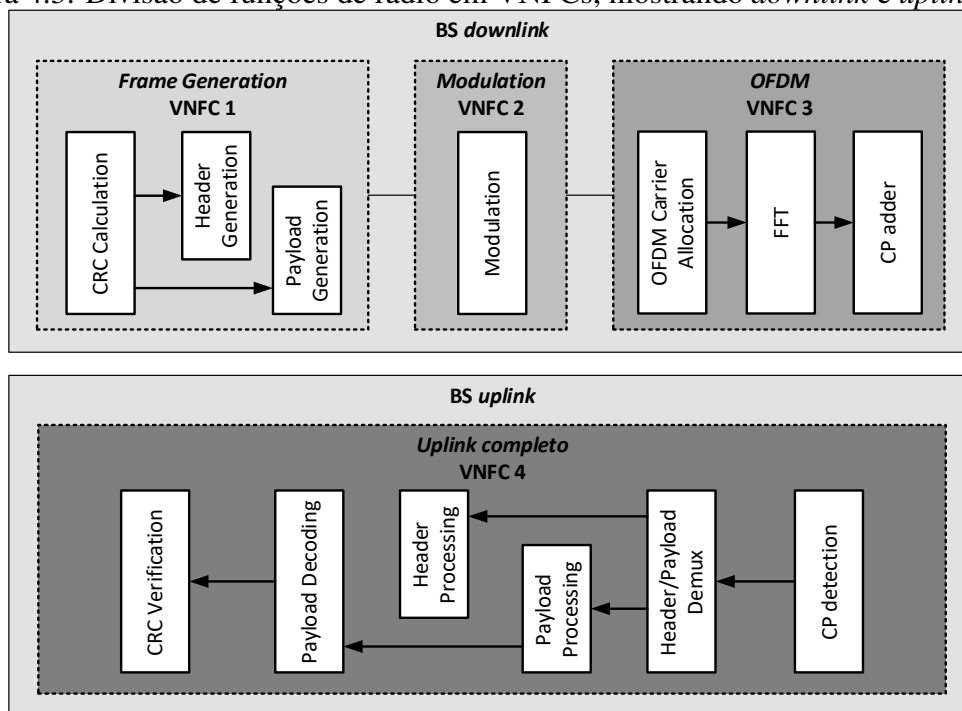
---

<sup>1</sup><https://linuxcontainers.org/lxd/>



eles: i) *Frame Generation*: que realiza do processamento de *CRC Calculation*, *Header Generation* e *Payload Generation*, ii) *Modulation*: que realiza o processamento de *Modulation*, e finalmente, iii) *OFDM* que realiza o processamento de *OFDM Carrier Allocation*, *FFT* e *CP adder*. Já para *uplink* da BS, a divisão das funções é feita de forma atômica, em apenas um VNFC, que realiza o processamento (da direita para esquerda) *CP decision*, *Header/Payload Demux*, *Payload Processing*, *Header Processing*, *Payload Decoding*, *CRC Verification*. Desta forma, uma BS é processada completamente, tanto para *downlink*, como para *uplink*.

Figura 4.3: Divisão de funções de rádio em VNFCs, mostrando *downlink* e *uplink* da BS

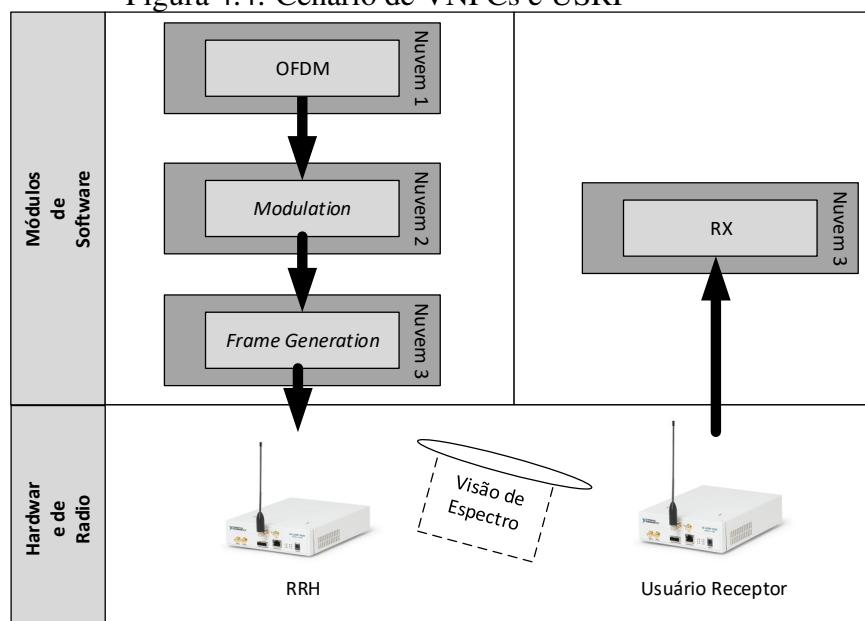


Há mais um VNFC acoplado ao final da cadeia das 4 VNFCs que tem a finalidade de emular um *Universal Software Radio Peripheral (USRP)*. O USRP é uma família de rádios definidos por software que pode ser programável, e que tem a função de receber as amostras de sinais geradas nas VNFCs anteriores e realizar o encaminhamento dessas amostras. A emulação do USRP foi realizada utilizando os blocos de processamento de sinais disponíveis no GNU Radio. O GNU Radio é um *framework* para desenvolvimento de softwares especializados em processamento de sinais.

Antes de ser descrito como a emulação do USRP funciona, será descrito como a sequência de operação das VNFCs ocorre na presença de um USRP. O USRP recebe um *stream* de amostras de sinais digitais do VNFC conectado ao USRP. O USRP então realiza a conversão dessas amostras para um sinal de radiofrequência que é transmitido pelo meio

sem fio. No lado da recepção, um equipamento USRP receptor, configurado para receber os sinais sem fio na mesma frequência do transmissor, captura o sinal analógico e realiza o processo de digitalização, o que dá origem a um *stream* de sinais digitais. Esse *stream* é enviado para um software que implementa o equipamento receptor. O sinal digital original (gerado pelo VNFC) e o sinal gerado pelo USRP receptor devem ser idênticos em um meio de comunicação perfeito, isto é, sem presença de qualquer ruído ou outro efeito de degradação. A seguir, a Figura 4.4 exemplifica o fluxo das amostras de sinais efetuados por um USRP real, onde há uma RRH e um usuário receptor, como descritos anteriormente.

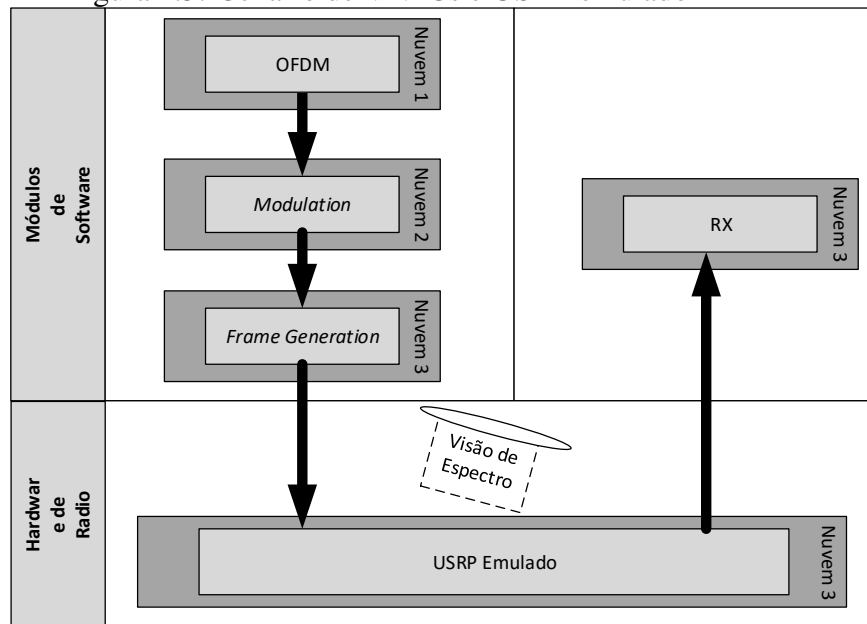
Figura 4.4: Cenário de VNFCs e USRP



Porém, como um meio de comunicação sem fio não pode ser obtido na prática, decidiu-se utilizar o emulador de USRP. A Figura 4.5 descreve a alteração na parte inferior, que, em comparação com a Figura 4.4 teve a inclusão do emulador de USRP. Nesse caso, a transmissão pelo meio sem fio não ocorre e o sinal digital é transferido diretamente pela rede cabeada. Para tal, o emulador USRP apenas encaminha as amostras de sinais digitais diretamente para o software que implementa o equipamento receptor. Em suma, a conversão de digital para analógico no transmissor, e de analógico para digital no receptor é removida.

O cenário utilizado para experimentação é mostrado na Figura 4.6. Para o gerenciamento e controle da infraestrutura virtualizada, *i.e.*, NFVI e VIM respectivamente, utilizou-se o *Aurora Cloud Manager* (WICKBOLDT et al., 2014), descritos na parte superior da figura.

Figura 4.5: Cenário de VNFCs e USRP emulado



Aurora é uma plataforma de nuvem de Infraestrutura como Serviço (IaaS) que permite o gerenciamento flexível de recursos por meio da capacidade de programação. Neste trabalho, Aurora suporta a virtualização das máquinas virtuais utilizadas, bem como, dá suporte à instanciação da rede OpenFlow entre elas. Desta plataforma, 3 máquinas virtuais são instanciadas com a finalidade de atender um cenário de *dynamic* C-RAN, cada uma representando um nível de nuvem. Na parte superior da figura, são mostradas as máquinas virtuais *VM1*, *VM2* e *VM3*, que são nuvens (respectivamente borda, regional e central), que possuem o sistema LXD instalado, e suportam a implantação das VNFs através deste sistema de *containers* (CANONICAL, 2018). O sistema de *containers* LXD é um gerenciador de *containers* que oferece uma experiência de usuário semelhante às máquinas virtuais, usando *containers* Linux. A instanciação das imagens de LXD é baseada em imagens pré-criadas disponíveis em um repositório.

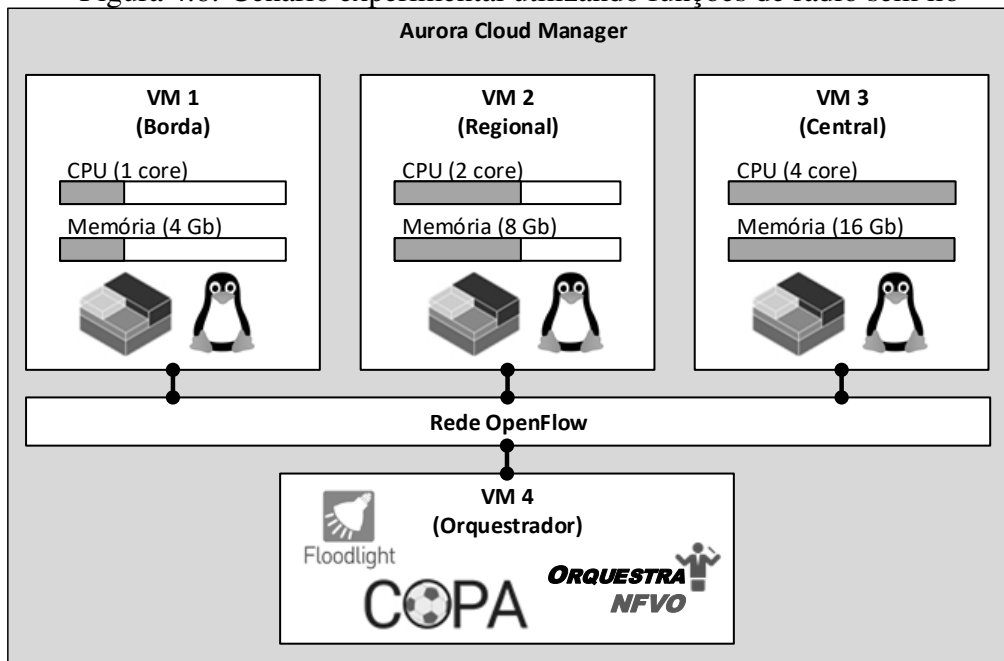
Tabela 4.1: Quantidade de recursos disponíveis em cada VM (nuvem)

Configuração por VM	VM 1 (borda)	VM 2 (regional)	VM 3 (central)	VM 4 (orquestrador)
CPU (cores)	1	2	4	2
Memória RAM (em GB)	4	8	16	2
Disco (em GB)	200	200	200	50

A Tabela 4.1 expressa a distribuição de recursos por cada uma das nuvens. A nuvem de borda possui menos recursos que as outras, tendo 1 *core* de CPU e 4Gb de

memória RAM. A nuvem regional, com mais recursos que a de borda, mas com menos do que a central, possui 2 *cores* de CPU e 8Gb de memória RAM. Já a nuvem central, sendo a nuvem com maior capacidade, possui 4 *cores* de CPU e 16Gb de memória RAM. Ambas as nuvens são virtualizadas sobre um processador Intel(R) Xeon de 1.90GHz, e com 6 *cores*/6 segmentos. Ambas as nuvens possuem 200Gb de espaço em disco. Como já descrito anteriormente, os VNFCs são suportados por *containers* Linux do tipo LXD, e a cadeia de processamento deles forma uma VNF, que neste caso, é a BS, tanto em *downlink*, como *uplink*. As funções de rádio descritas na Figura 4.3 são posicionadas nas VM1, VM2 e VM3, de acordo com os algoritmos escritos pelo operador da rede, para atingir algum objetivo específico. Nestas VMs, também são incluídos agentes SNMP, que tem por finalidade monitorar os recursos disponíveis nas nuvens.

Figura 4.6: Cenário experimental utilizando funções de radio sem fio



Na parte central da Figura 4.6 são mostradas as conexões entre as VMs, conectadas por uma rede SDN OpenFlow, com um controlador Floodlight, hospedado na VM4. Além da rede OpenFlow, é utilizado (*Simple Flow*) sFlow para captura dos fluxos entre os *containers* e as nuvens, com a finalidade de ter o consumo de taxa de dados no *fronthaul*, e fornecer estas informações aos algoritmos do operador da rede, para que possam ser tomadas as decisões. Na parte inferior da Figura 4.6, há a VM4, que é a VM responsável por gerenciar e orquestrar as funções de rádio virtualizadas. A separação de uma VM somente para orquestração e gerenciamento se deve ao objetivo de não afetar o desempenho das nuvens que processam as VNFs, tendo que forçá-las a migrar funções por conta

de desempenho consumido pelo gerenciamento e orquestração. Esta VM possui 2Gb de memória RAM, 2 cores de CPU, e 50Gb de espaço de disco. A VM4 também hospeda o *Container Orchestration and Provisioning Architecture* (COPA), provido pelo projeto FUTEBOL (FUTEBOL, 2018).

COPA é um gerenciador de *containers* do tipo LXD que fornece uma camada de abstração entre os *containers* e as suas respectivas APIs, *i.e.*, para utilizar o sistema de *containers*, o COPA abstrai os processos necessários para a manutenção do ciclo de vida das VNFs, (*e.g.*, instanciar, destruir, migrar VNFCs entre nuvens, VMs ou locais de rede, integrar com outras linguagens de programação). Além de fornecer suporte via API, ele também suporta gerenciamento via OSS/BSS, fornecendo também administração via interface gráfica. Neste trabalho, o COPA é utilizado como VNFM, suportando o ciclo de vida dos VNFCs e fornecendo suporte para a orquestração das VNFs, de forma que quando as ações decididas pelos algoritmos do operador da rede, então o Orquestra comunica as decisões ao COPA e aguarda que elas sejam realizadas (*e.g.*, a migração de um ou mais VNFCs de uma nuvem para outra). Outro software que está incluso também na VM4 é o Orquestra, que tem por finalidade orquestrar e gerenciar os VNFCs e as VNFs.

Orquestra tem a finalidade de controlar essas funções de rádio transformadas em VNFCs ao longo das nuvens, e como citado no modelo ESTI NFV MANO, a ligação entre esses VNFCs, são os VDUs. Para cada tipo de infraestrutura e rede, o operador da rede pode escrever algoritmos personalizados, ou, obter esses algoritmos de fornecedores de software. Desta forma, no próximo capítulo, são descritos os resultados obtidos por Maestro e Orquestra, junto com uma discussão de ambos os resultados.

## 5 RESULTADOS

Neste capítulo são descritos os resultados da avaliação de Maestro e Orquestra. Como já descrito anteriormente, ambos fazem parte de uma solução de orquestração que foi conduzida em duas etapas. Enquanto maestro é a primeira parte do trabalho, que de forma simulada foca em aspectos técnicos e funcionais de *dynamic* C-RAN, Orquestra, a segunda parte, foca em um ambiente de orquestração experimental de também *dynamic* C-RAN. As avaliações de cada uma das etapas são descritas a seguir.

### 5.1 Resultados da solução conceitual

A avaliação simulada foi feita em duas etapas. Primeiramente, utilizando demanda fixa de consumo das BS, e posteriormente, suportando a demanda variável de cada uma. Ambas as avaliações foram realizadas utilizando as funções de rádio descritas por Wubben et al. (WUBBEN et al., 2014), conforme descritas na Figura 4.2.

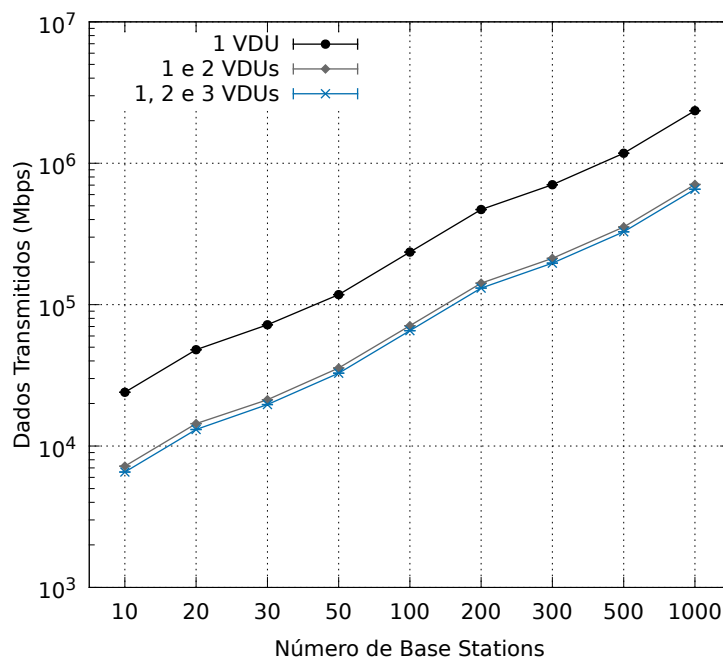
#### 5.1.1 Avaliação com demanda das BSs fixas

Implementou-se o modelo usando o *IBM CPLEX Optimizer* (IBM Software, 2010). A simulação foi executada em um computador com processador Intel Core i7 2.4 GHz com 8 GB de memória RAM, sobre um sistema operacional Linux de 64 bits. Decidiu-se variar o número de antenas em 9 cenários diferentes, entre 10 e 1000 BSs, com a finalidade de entender o impacto que isso representaria tanto para o consumo de taxa de dados no *fronthaul*, como no tempo necessário para calcular o posicionamento dos VNFCs. Para todos os cenários, considerou-se três nuvens (*i.e.*, borda, regional e central) com capacidade combinada para acomodar a carga de trabalho de processamento de sinal das BSs, de forma proporcional aos exemplos da Figura 4.2 com ocupação das BSs fixadas em 50%. A ocupação das BSs influencia a taxa de dados transmitidos para as funções de *MAC*, *Soft-Bit*, *RX Data*, tal como é descrito por Wubben et al. (WUBBEN et al., 2014). Todos os cenários foram repetidos 50 vezes, e o tempo medido pelo software CPLEX para calcular o posicionamento de VNFC varia em cada execução.

A Figura 5.1 demonstra o consumo de taxa de dados agregado no *fronthaul* para diferentes formas de divisão das VNFs. O eixo x indica o número de BS em cada cenário

de experimentação e o eixo y (em escala logarítmica) demonstra a quantidade de taxa de dados agregada em Gbit/s, dado o posicionamento dos VNFCs calculado pelo algoritmo de solução linear. A linha 1 VDU, que é a divisão atômica, representa a taxa de dados consumida quando apenas um VDU é usado para implantar cada VNF, *i.e.*, cada antena terá seus cinco VNFCs posicionados na mesma nuvem. A linha 1 e 2 VDUs, demonstra o consumo de taxa de dados de até dois VDUs por VNF. Finalmente, a linha divisão de 1, 2 e 3 VDUs demonstra a quantidade de taxa de dados agregada *fronthaul* de até 3 VDUs por VNF (semelhante aos exemplos na Figura 4.2, na forma da divisão de múltiplas opções).

Figura 5.1: Consumo de taxa de dados agregada no *fronthaul*

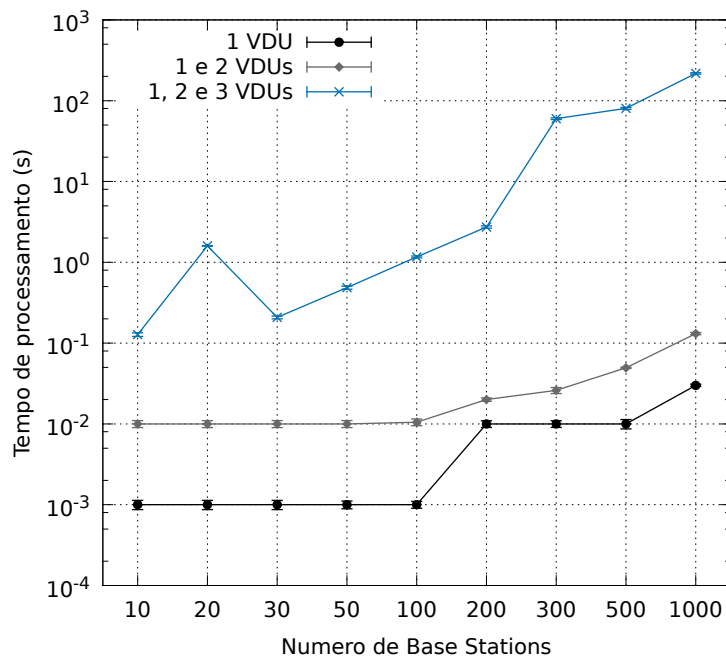


Analisando a Figura 5.1, é possível visualizar que existe uma redução importante no consumo de taxa de dados no *fronthaul* ao comparar a linha divisão atômica com qualquer uma das outras duas linhas para qualquer cenário de experimentação. Esta redução representa uma clara vantagem da divisão de múltiplas opções, que representa uma redução constante de três vezes no consumo de taxa de dados no *fronthaul* a medida que a quantidade de VDUs por VNF aumenta. Por exemplo, no cenário 1000 antenas, a divisão atômica consumiria 2,35 Tbit/s, enquanto usando 2 e 3 VDUs por VNF esses números diminuíram para 708,12 Gbit/s e 655,2 Gbit/s, respectivamente. O ganho obtido não é tão significativo ao comparar as linhas divisão 1 e 2 VDUs e divisão 1, 2 e 3 VDUs. Esta pequena redução no consumo de taxa de dados, é possível devido à distribuição de VNFCs maximizando o processamento na nuvem de borda. O aumento no número de VDUs disponíveis para implantação é útil para economizar a taxa de dados no *fronthaul*, mas

também apresenta complexidade no processo de posicionamento dos VNFCs, tal como será discutido em seguida.

A Figura 5.2 ilustra o tempo médio decorrido para calcular o posicionamento de cada VNFC em todas as nuvens disponíveis. O tempo medido pelo CPLEX varia em cada execução, de modo que as barras de erro são mostradas no gráfico e representam um intervalo de confiança de 99% para o tempo médio. O eixo x mostra o número de BS em cada cenário de experimentação, enquanto o eixo y (em escala logarítmica) representa o tempo em segundos necessário para calcular o posicionamento dos VNFCs. Tal como na Figura 5.1, as linhas representam respetivamente a VNF com 1 VDU apenas, (*i.e.*, divisão atômica), e subsequente, até 2 VDUs e até 3 VDUs como divisão de múltiplas opções.

Figura 5.2: Tempo para calcular o posicionamento dos VNFCs



Na Figura 5.2 é possível visualizar que, a medida que o número de VDUs aumenta, o tempo também aumenta, em razão do aumento da complexidade do posicionamento dos VNFCs. Para a divisão atômica e divisão de 1 e 2 VDUs, embora haja um aumento de 100 BS, o tempo total de cálculo é quase insignificante até 1000 BS. O aumento de tempo entre a divisão de 1 e 2 VDUs e a divisão de 1, 2 e 3 VDUs, por outro lado, é substancial. Um grande aumento na carga de processamento pode ser justificado pelo número de possíveis combinações de VDUs que podem ser usadas para implantar cada VNF.

Comparando ambos os gráficos das Figuras 5.1 e 5.2, é possível notar que no primeiro há ganho em relação ao consumo de taxa de dados ao considerar divisões de até 3 VDUs. No entanto, requer muito mais tempo de processamento para um grande número



de antenas do que até 2 VDUs (217 para 3 VDUs contra 0, 131 segundos para 2 VDUs em 1000 BSs), o que pode ser um obstáculo em cenários onde o recurso as alocações mudam dinamicamente.

### 5.1.2 Avaliação com demanda das BSs variável

Sabendo que os dados transmitidos no *fronthaul* variam de acordo com a ocupação das BSs, conduziu-se a avaliação também em cenários que variam a demanda das BSs. Para realizar os experimentos foi necessário estimar os percentuais de ocupação com base em uma simulação utilizando as recomendações do *3rd Generation Partnership Project* (3GPP), do documento *Further advancements for E-UTRA physical layer aspects* (3GPP, 2010). Esse percentual de ocupação foi calculado através do método de *Monte Carlo*, em uma simulação do tráfego gerado por *User Equipments* (UEs) associados a uma infraestrutura com 7 *macro cells* e 15 *small cells* distribuídas dentro da área de cobertura de cada *macro cell*, totalizando 112 BSs. Em média, 60 usuários foram distribuídos aleatoriamente dentro da área de cobertura de cada *macro cell* e associados a BS mais próxima de cada usuário (seja uma *macro* ou *small cell*). A distribuição das BSs e UEs foi realizada em uma área de 2000x2000 metros, seguindo as normas para simulação definidas no documento *Small cell enhancements for E-UTRA and E-UTRAN - Physical layer aspects* do 3GPP (3GPP, 2013) para uma rede celular, considerando a existência de áreas com diferentes densidades de BS e UEs *i.e.*, zonas rurais e urbanas.

Tabela 5.1: Distribuição da ocupação das *macro* e *small cells*

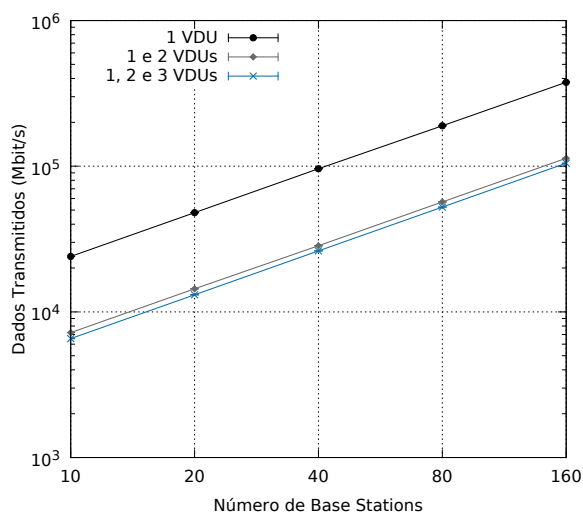
Índice de Ocupação	<i>Macro cells</i>		<i>Small cells</i>	
	Carga de Ocupação	Probabilidade de Ocupação	Carga de Ocupação	Probabilidade de Ocupação
0	15%	0,42%	2,5%	56,97%
1	25%	1,42%	12,5%	31,59%
2	35%	5,14%	22,5%	9,28%
3	45%	10,28%	37,5%	1,88%
4	65%	15,71%	47,5%	0,28%
5	75%	19,17%	-	-
6	85%	17,29%	-	-
7	95%	30,57%	-	-
	Total	100%	-	100%

Na Tabela 5.1 pode-se encontrar a ocupação das BSs utilizadas na simulação, expressa através de uma distribuição discreta de probabilidades, servindo como entrada para o modelo linear detalhado na Seção 3.1.1. A ocupação das BSs influencia a taxa de dados

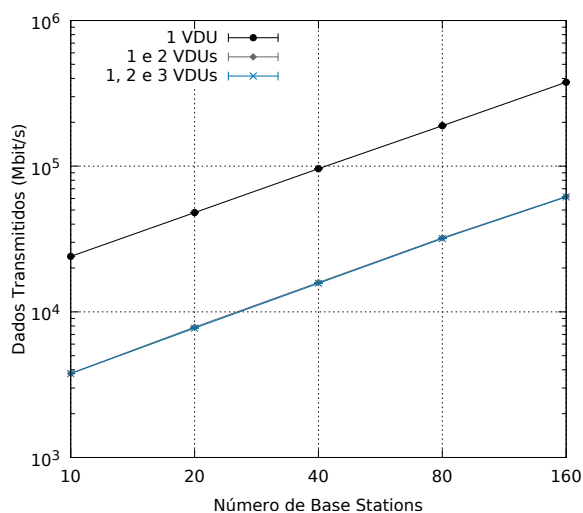
transmitidos para as funções de *MAC*, *Soft-Bit*, *RX Data*, tal como é descrito por Wubben et al. (WUBBEN et al., 2014). A quantidade de *macro cells* consideradas na simulação corresponde a 33,3% da infraestrutura, enquanto que as *small cells* representam 66,7%. Essa medida se deve à quantidade de UEs distribuídos ao longo da região geográfica, sendo maior para as *macro cells* e menor para as *small cells*. Os cenários de avaliação possuem 10, 20, 40, 80 e 160 BSs ao longo da infraestrutura.

Assim como no modelo de avaliação com demanda fixa, o modelo de programação linear foi implementado com *IBM CPLEX Optimizer* (IBM Software, 2010). A simulação foi executada em um computador com um processador Intel i7 de 2.4 GHz, com 8 GB de memória RAM, executando sobre um sistema operacional Linux de 64 bits. Para todos os cenários, considerou-se três nuvens (*i.e.*, de borda, regional e central), com capacidade proporcional para acomodar a carga de trabalho nos diferentes cenários. Todos os experimentos foram repetidos 50 vezes e o tempo para posicionar os VNFCs ao longo dos VDUs, foi medido pelo *IBM CPLEX*. Considerou-se tanto a ocupação fixa das BSs no posicionamento das funções de rádio, quanto ocupação variável (conforme a Tabela 5.1), diferentemente da avaliação de demanda fixa. Para avaliar os ganhos do posicionamento na ocupação variável, comparou-se os mesmos cenários gerados em ocupação (*i.e.*, fixa e variável) de 10 a 160 BS. Desta forma, é possível comparar a quantidade de tempo necessária para efetuar o posicionamento das funções de rádio e a taxa de dados consumida no *fronthaul* em ambos os casos, comparadas ao mesmo cenário com a mesma quantidade de BSs.

Pode-se observar na Figura 5.3(a) a variação total de dados transmitidos no *fronthaul* para diferentes opções de divisão de funções de rádio, considerando que as BSs tenham ocupação percentual fixa em 50%. A Figura 5.3(b) apresenta o mesmo resultado, porém considerando ocupação variável das BSs (aleatoriamente atribuídos conforme a Tabela 5.1). O eixo x denota a quantidade de BSs em cada experimento, enquanto o eixo y (em escala logarítmica) denota os dados transmitidos no *fronthaul* em Mbit/s ocasionado pelas opções de divisão selecionadas pela função de posicionamento. A curva 1 VDU (*i.e.*, divisão atômica) representa a transmissão dos dados utilizando apenas 1 VDU contendo todos os VNFCs. Já a curva 1 e 2 VDUs expressa a transmissão dos dados quando VNFs são posicionadas em um ou dois VDUs (*i.e.*, podendo se utilizar uma opção de divisão para separar VNFCs em até dois VDUs). Por fim, a curva 1, 2 e 3 VDUs representa os dados transmitidos de uma VNF implantada em até 3 VDUs, podendo usar simultaneamente até duas opções de divisão para separar os VNFCs de uma mesma VNF).

Figura 5.3: Transmissão de dados no *fronthaul*

(a) Percentual de ocupação fixa das BSs



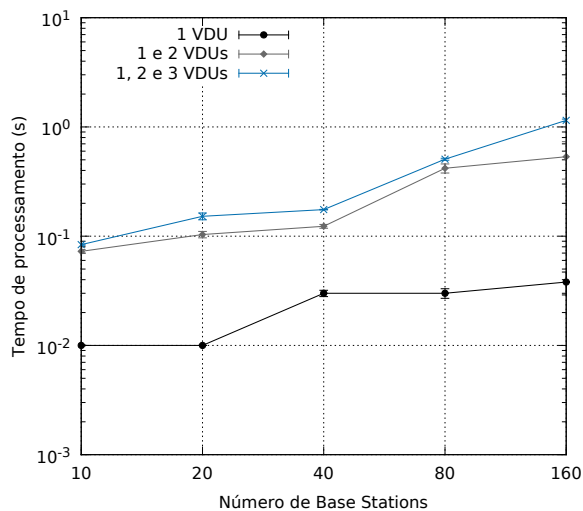
(b) Percentual de ocupação variável das BSs

Observando o gráfico de dados transmitidos com ocupação fixa, para 10 BSs a quantidade de dados transmitido utilizando apenas 1 VDU foi de 24 Gbit/s, enquanto que até 2 e até 3 VDUs esse valor é reduzido para 7,2 Gbit/s e 6,5 Gbit/s, respectivamente. Percentualmente, houve uma redução de 70% entre os cenários utilizando 1 VDU versus 2 VDUs e de 72,7% comparando 1 e 3 VDUs. Entre os cenários de até 2 e até 3 VDUs a diferença foi apenas de 9%. Já no caso mais extremo, considerando 160 BSs, a banda consumida utilizando 1 VDU atingiu 376,8 Gbit/s, enquanto que para até 2 VDUs foi de 113,4 Gbit/s e para até 3 VDUs foi de 104,83 Gbit/s. Assim, manteve-se aproximadamente os mesmos preceituais de redução entre as curvas, enfatizando a importância da composição de VNFs no uso racional dos recursos do fronthaul.

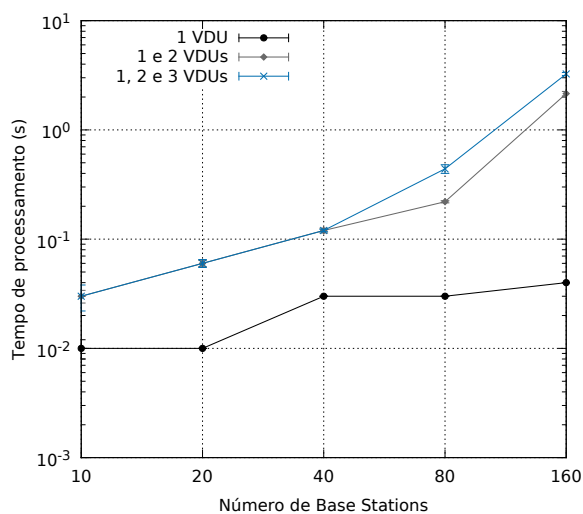
Ao observar o gráfico de dados transmitidos com ocupação variável para 10 BSs,

a quantidade de dados transmitidos para 1 VDU foi dos mesmos 24 Gbit/s obtidos com ocupação fixa. Isso se deve ao fato de que no posicionamento de 1 VDU pode-se utilizar apenas a opção de divisão da função de rádio I/Q, que requer uma taxa de transferência fixa independente da ocupação da BS. Ainda considerando 10 BSs, as curvas de até 2 VDUs e até 3 VDUs obtiveram taxas de transmissão de dados na ordem de 3,8 Gbit/s e 3,7 Gbit/s, respectivamente, atingindo reduções de aproximadamente 83% em comparação com a curva de 1 VDU. Na comparação dos resultados entre as curvas de 2 VDUs e 3 VDUs, os valores se mantiveram sempre muito similares para todas as quantidades de BSs simuladas com diferenças sempre abaixo de 1%.

Figura 5.4: Tempo necessário para encontrar uma solução



(a) Solução na ocupação fixa



(b) Solução na ocupação variável

Pode-se observar que tanto na ocupação fixa como na variável, o orquestrador é capaz de minimizar a transmissão de dados no *fronthaul*, quando existir a possibilidade

de selecionar as divisões de funções de rádio com 2 ou 3 VDUs. Essa redução é ainda mais evidente quando há demanda variável nas BSs, o que se deve principalmente a dois fatores: (i) existe uma diferença de ocupação média significativa entre *macro* e *small cells*, o que fornece ao orquestrador uma gama de opções de VDUs com transmissão de dados diferentes para posicionar; e (ii) apesar da alta ocupação nas *macro cells* (em média 75%), há um grande número de *small cells* com baixa ocupação (em média 8%), o que reduz a ocupação média por BS para a casa dos 30%. A utilização de até 3 VDUs, ainda que acarrete ganhos em relação a até 2 VDUs, traz vantagens muito tímidas no cenário de ocupação variável, que podem fazer com que o esforço extra para calcular o posicionamento não compense, pois obtém ganhos pouco significativos em relação a 2 VDUs, e possui maior complexidade no cálculo.

As Figuras 5.4(a) e 5.4(b) ilustram o tempo médio para computar o posicionamento de todos os VNFCs sobre a infraestrutura de nuvens disponível, considerando as BSs com ocupação fixa e variável. As barras de erro representam um intervalo de confiança de 99%, obtido por meio de 50 repetições para cada experimento. O eixo x expressa o número de BSs no cenário, enquanto o eixo y denota o tempo médio em segundos necessário para calcular o posicionamento dos VNFCs.

Os gráficos de tempo demonstram que para até 80 BSs, o tempo necessário para realizar o posicionamento com ocupação fixa é superior ao tempo de ocupação variável. Isso se justifica pois no cenário de ocupação fixa os recursos se esgotam mais rapidamente, de forma que o orquestrador testa mais possibilidades para alocar os VDUs de forma eficiente. Entretanto, para 160 BSs o cenário de ocupação variável consome mais tempo (3,19s para até 3 VDUs, contra 1,12s da ocupação fixa) do que a ocupação fixa. Fato este que pode ser explicado pela grande quantidade de combinações de VDUs com demandas de banda diferentes disponíveis para posicionamento pelo orquestrador.

No cenário de ocupação variável, para realizar o posicionamento de até 40 BSs utilizando 2 ou 3 VDUs, os tempos de cálculo são extremamente similares. Isso torna viável a utilização de divisões de funções de rádio em até 3 VDUs, uma vez que isso leva a uma redução, ainda que pequena, no consumo de recursos do *fronthaul*. Já para 80 e 160 BSs, o tempo necessário para realizar o posicionamento utilizando até 3 VDUs, ocasiona um aumento de mais de aproximadamente 40% no tempo médio de cálculo em relação a até 2 VDUs.

Desta forma, pode-se concluir que em um ambiente com uma grande quantidade de BSs, onde a variação da ocupação for frequente, a utilização de até 2 VDUs pode ser

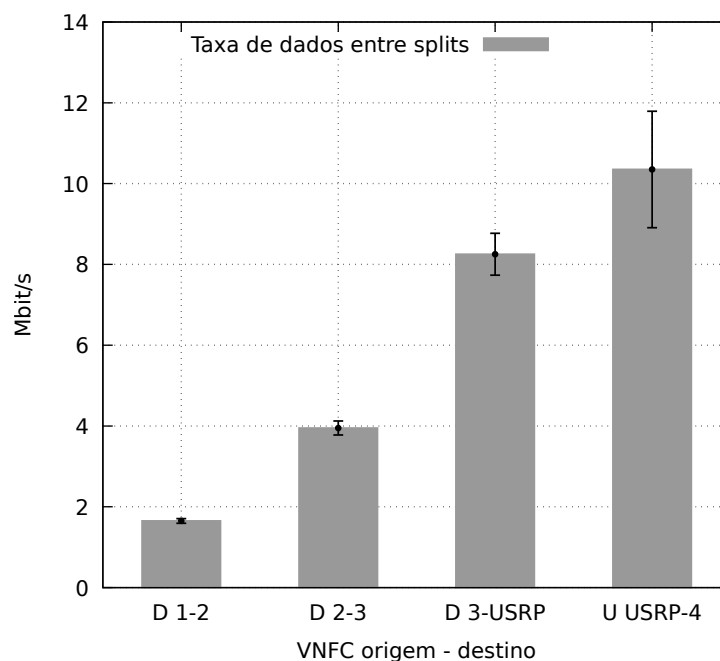
mais adequada para permitir que o orquestrador responda mais rapidamente às mudanças do ambiente sem prejudicar significativamente o consumo de recursos.

Finalmente, vale ressaltar que não está sendo considerado o tempo necessário para a implantação/migração dos componentes de VNFs entre as nuvens, o que, em uma infraestrutura real, deverá influenciar no tempo total necessário para que a nova configuração calculada pelo orquestrador entre em vigor. Ademais, os tempos de resposta da modelagem linear sempre são dados até a execução total do posicionamento dos VDUs, de acordo com a função objetivo definida na Seção 3.1.2. Nesse caso, se essa função for modificada, os tempos de posicionamento serão diferentes.

## 5.2 Resultados da solução experimental

O ambiente de experimentação foi hospedado por um computador com processador Intel(R) Xeon(R) de 1.90GHz e 32 Gb de memória RAM, sobre um sistema operacional Linux de 64 *bits*. Tal como descrito na Seção 4.2, neste experimento, uma VNF é composta de 5 VNFCs, que podem ser implantados em diferentes nuvens.

Figura 5.5: Consumo de taxa de dados entre VNFCs em *download* (D) e *upload* (U)



O algoritmo utilizado para o posicionamento dos VNFCs, descrito no capítulo anterior (Pseudocódigo 2) objetivou a maximização do uso dos recursos computacionais das nuvens, e implantou 4 VNFs em 3 nuvens (*i.e.*, VNFCs na nuvem borda, regional e

central até que a de borda ser completada), 5 VNFs em 2 nuvens (*i.e.*, VNFCs na nuvem regional e central até a borda ser completada) e 3 VNFs em apenas uma nuvem (*i.e.*, todos os VNFCs na nuvem central, até que seu processamento atingisse 100%), totalizando 12 BS completas e 60 *containers* implantados na infraestrutura. Ambos os testes descritos a seguir foram repetidos 10 vezes e foram avaliados através de uma taxa de confiança de 99%. Todos os gráficos também possuem barras de erro no eixo y, que auxiliam na observação da variação entre os experimentos. As medições de taxa de dados foram realizadas utilizando sFlow apoiado pela ferramenta gráfica NFSen e já no controlador OpenFlow, foi utilizado FoodLight. O monitoramento de recursos das VMs foi feito através de SNMP e do COPA VNFM.

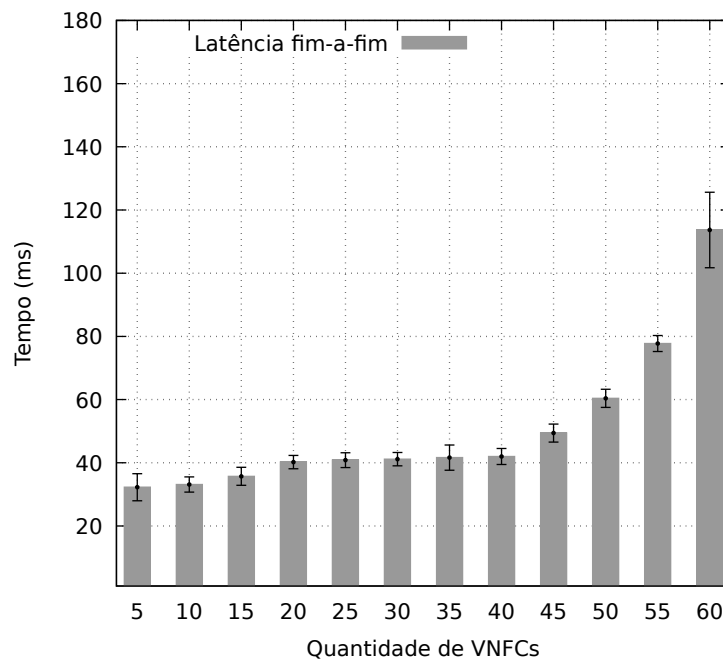
A Figura 5.5 demonstra a quantidade média de taxa de dados consumida entre os VNFCs durante o tempo de experimentação. No eixo y é descrita a quantidade média de Mbit/s consumida entre os VNFCs, e no eixo x, a relação de medição entre as divisões de VNFCs. Para diferenciar o fluxo dos dados, as legendas com D representam tráfego de *downlink*, e as legendas com U, representam tráfego de direção de *uplink*. Como é possível notar no gráfico, para o fluxo de *downlink* entre o VNFC 1 e 2, houve um consumo médio de 1,65 Mbit/s. Entre o VNFC 2 e 3 houve o consumo médio de 3,95 Mbit/s, e entre o VNFC 3 e USRP houve um consumo médio de 8,25 Mbit/s. Já no sentido de *uplink*, houve um consumo médio de 10,35 Mbit/s entre o USRP e o VNFC 4. A avaliação não considera os dados utilizados nas respostas do protocolo TCP (*i.e.*, *acks*), e apenas é considerada no sentido de um fluxo de dados (*i.e.*, *downlink* e *uplink*). Desta forma, é possível notar que a maior quantidade de dados consumida é entre o USRP e VNFC 4, já que, apenas um VNFC é utilizado para realizar todo o processamento no *uplink*, diferente do que é feito para o *downlink*, que é dividido em 3 VNFCs. Outro aspecto do comportamento que pode ser visto, é que a taxa de dados entre VNFCs aumenta gradativamente no sentido de *downlink* (perto do fator 2, como por exemplo para VNFC 1 e 2 e VNFC 2 e 3). Este aumento se deve a natureza da comunicação entre as funções de rádio.

Os próximos resultados são descritos através de uma quantidade de BS agregada em cada experimento. A cada valor no eixo x, é representado um incremento de 1 BSs (*i.e.*, 5 *containers*), e desta forma, o eixo x representa uma quantidade agregada de instâncias de BSs em números de VNFCs, (*e.g.*, 5, 10, 15).

A Figura 5.6 demonstra a latência fim-a-fim das VNFs, expressa em razão da quantidade de VNFCs. No eixo y, é mostrada a latência em milissegundos, e o no eixo x, a quantidade de VNFCs utilizadas na experimentação. Para 5 VNFCs, a latência teve

um tempo médio de 32 ms, enquanto que no maior pico, de 60 VNFCs, a latência média foi de 113 ms. Através do gráfico é possível notar que a latência aumenta em relação que a quantidade de processamento passa a ser escassa, tal como descrito no gráfico nos 60 e nos 20 VNFCs, quando a nuvem central e a de borda tem seu processamento perto de 100%, respectivamente. Outro fator que contribui para o incremento da latência é a quantidade de tráfego passando entre o *fronthaul*, requisitando maior processamento dos comutadores e das nuvens.

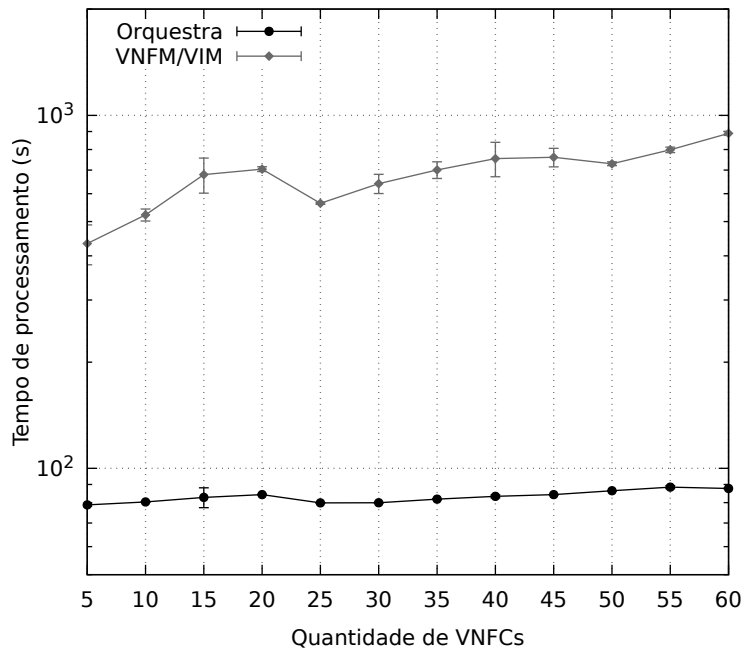
Figura 5.6: Latência das VNFs fim-a-fim



A Figura 5.7 demonstra a média de tempo necessária para implantar um conjunto de VNFCs, em determinado tempo da rede. No eixo y (em escala logarítmica) é descrita a quantidade de tempo média necessária para implantar uma VNF (5 VNFCs), dado a quantidade de VNFCs já implantadas, e descritas no eixo x. O gráfico separa o tempo de implantação do VNFM (*i.e.*, COPA e LXD), do tempo de implantação do orquestrador Orquestra. O tempo de implantação do COPA e LXD competem ao tempo em que as VNFs e seus VNFCs (*i.e.*, *containers*) são implantadas nas nuvens até o ponto em que as funções fiquem disponíveis para manipulação, porém ainda não possuem a configuração necessária para sua total execução. Já o tempo de implantação do orquestrador, é o tempo necessário para estabelecer conexão e configurar os VNFCs e começar a gerar tráfego entre as funções. O tempo total de implantação das funções, é a soma dos tempos de implantação do VNFM e do Orquestra.



Figura 5.7: Tempo de implantação das VNFs

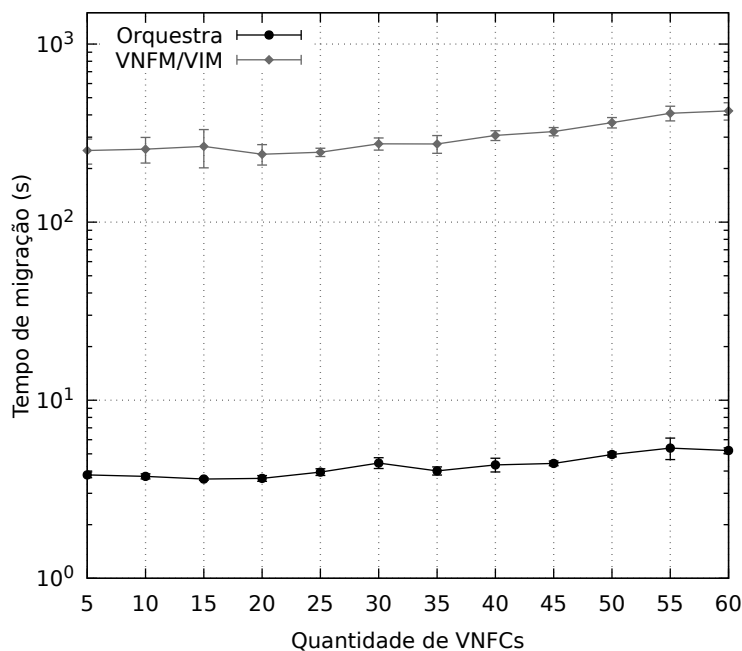


Para a implantação de 5 VNFs, o VNFM demorou 433,56 segundos e para a implantação de 60 VNFs, o VNFM demorou 888,96 segundos, mostrando que o tempo aumenta exponencialmente conforme a quantidade de VNFs implantados. Já para o tempo de configuração consumido pelo Orquestra, para 5 VNFs foi de 78 segundos, e para 60 VNFs, foi de 88 segundos. Através do gráfico é possível notar que conforme a nuvem tem seu processamento perto de 100%, a quantidade de tempo necessária para o VNFM implantar os VNFs, aumenta, tal como é mostrado na Figura 5.7 no eixo x em 20 e 45. Após a modificação nas nuvens que recebem os VNFs, o tempo de implantação do VNFM diminuiu, tal como é mostrado no eixo x nos valores 25 e 50. Já para o tempo do Orquestra, os valores de variação foram menores, mostrando que o aumento se dá predominantemente no tempo de implantação do VNFM, e não na configuração dos VNFs, ou seja, mesmo para maior quantidade de VNFs, apesar do tempo aumentar, o do Orquestra tem menor variação que o VNFM.

Por fim, a Figura 5.8 mostra a média de tempo (em segundos) necessária para migrar uma função de uma nuvem para outra, e para migrá-la de volta a mesma nuvem. O eixo y (em escala logarítmica) mostra a quantidade de tempo necessária para a migração, e o eixo x, mostra a quantidade de VNFs implantados no *fronthaul* quando ocorreram as migrações. O gráfico se divide em tempo de migração, que é realizado pelo VIM/VNFM, e tempo de reconfiguração de funções desempenhado pelo Orquestra. Para 5 VNFs, a média de tempo de migração do VNFM/VIM foi de 252 segundos, enquanto que para 60

VNFCs, foi de aproximadamente 421 segundos. Já para a reconfiguração desempenhada pelo Orquestra, para 5 VNFCs a média foi de 3,8 segundos, enquanto que para 60 VNFCs, foi de 5,21 segundos. Através do gráfico, é possível notar um aumento na quantidade de tempo necessária para migrar as funções conforme a quantidade de VNFCs é incrementada. Este mesmo comportamento se repete no processo de reconfiguração das funções pelo Orquestra, que também aumenta conforme a quantidade de VNFCs for incrementada. Desta forma, é possível concluir que, tal como a latência, a quantidade de tempo ocupada na migração tem relação com a quantidade de processamento ocupada pelas nuvens, ou seja, quanto mais escasso o processamento, maior é a quantidade de tempo para migrar.

Figura 5.8: Tempo de migração de funções entre uma nuvem e outra



Através dos gráficos é possível observar que há relação entre o tempo de migração e a latência com a quantidade de VNFCs implantados. Conforme há incremento na quantidade de VNFCs, maior é o tempo necessário para migração e o tempo de latência. Já para o tempo de implantação, o aumento do tempo se dá quando o processamento das nuvens está perto dos 100%, e desta forma, quando há mudança na forma de divisão dos VNFCs, o tempo diminuiu. O consumo de taxa de dados entre VNFCs se mostrou entre apenas uma divisão de VNFCs, pois, para o *uplink*, há apenas um VNFC disponível.

### 5.3 Discussões entre soluções

Nesta seção faz-se uma discussão entre as duas etapas que compunham a solução de orquestração, a conceitual e a experimental. Como já descrito anteriormente, a primeira parte do trabalho, a conceitual, buscou avaliar a viabilidade da utilização de NFV em ambientes do tipo *dynamic* C-RAN, avaliando os aspectos técnicos e funcionais deste tipo de ambiente, enquanto que a segunda e última parte, a experimental, focou em desenvolver *dynamic* C-RAN em um ambiente de experimentação com funções reais.

Através da avaliação conceitual, é possível notar a viabilidade do ambiente *dynamic* C-RAN e a redução na taxa de dados no *fronthaul*, bem como a maximização dos recursos nas nuvens de borda que o orquestrador proporciona, *i.e.*, não considerando o tempo de implantação e migração das funções, torna-se uma solução bastante atrativa.

Do ponto de vista experimental, implementar *dynamic* C-RAN através de NFV ainda pode esbarrar em algumas limitações no quesito de migrações, principalmente por parte do VIM e VNFM. O tempo para migrar e reestabelecer o serviço pode não oferecer a dinamicidade necessária para estes tipos de ambiente. Pode-se tentar melhorar o tempo de migração do serviço não migrando os VNFCs, mas sim, realizando a implantação de novos VNFCs em outras nuvens, o que provavelmente agilizaria o processo de reestabelecimento do serviço, dado a comparação entre o tempo de migração e implantação.

Outra possibilidade não levantada no trabalho mas que é plausível, é a utilização do bloco de decisão do Maestro pelo bloco de gestão de otimização do Orquestra, o que faria o algoritmo linear como entrada no algoritmo do operador da rede. Desta forma, o operador de rede pode utilizar o modelo de programação linear para posicionar os VNFCs nas nuvens, fazendo os algoritmos convergirem em um tempo menor e tendo a minimização de taxa de dados no *fronthaul*.

Nesse trabalho a solução de orquestração experimental permite que o operador de rede escreva algoritmos personalizados para orquestração, o que pode adaptar o orquestrador em vários cenários diferentes, tanto para uma rede *dynamic* C-RAN, como para uma testbed de experimentação, dado que a abstração feita pelo VIM e pelo VNFM conseguem portar o Orquestra para vários outros tipos de ambientes de execução, adaptando-o de acordo com o ambiente a lógica de operação da rede, permitindo que o orquestrador possa também ser utilizado para fins comerciais, tal como uma operadora de rede de celulares.

Algumas limitações da etapa conceitual são a adaptação a outros tipos de rede

sem hierarquia, como por exemplo redes do tipo anel, que neste caso não são contemplados neste estudo, mas podem ter ganhos significativos na implantação de funções com múltiplas divisões, quando necessário. Outra limitação é a execução do algoritmo linear periodicamente, que, em grandes infraestruturas pode inviabilizar muitas execuções simultâneas. Também outra limitação é que o algoritmo linear necessita ser executado para construir o posicionamento da infraestrutura do zero, desta forma não considera as BSs já alocadas. Uma forma de agilizar o processo e melhorar esse tempo, é obtendo os custos de migração e definindo um custo para a alocação já existente.

Pelo lado experimental, nos cenários apresentados, os resultados demonstraram que o orquestrador não é o gargalo em relação a quantidade de tempo das operações, mas vários resultados podem ser característicos do ambiente proposto para experimentação, e podem ser alterados em uma infraestrutura distribuída em que as máquinas virtuais não são implantadas na mesma máquina física, como é o caso da latência. Outra limitação é a operação do orquestrador em modo texto, sem interface gráfica. Adotando interface gráfica, pode facilitar a manipulação das informações e melhorar a usabilidade do orquestrador pelo operador de rede.

## 6 CONCLUSÕES

Neste trabalho houve avanço na orquestração de NFV para ambientes *dynamic* C-RAN através de duas propostas. Uma delas, chamada de Maestro, baseada em programação linear com foco em um ambiente *dynamic* C-RAN simulado, e outra, chamada de Orquestra, baseada em um ambiente *dynamic* C-RAN experimental.

A solução simulada baseia-se nos componentes de funções virtualizadas para realizar o posicionamento de VNFs, baseando-se na modelagem de posicionamento de VNFs e em programação linear dos recursos em *dynamic* C-RAN. A avaliação foi realizada baseada em simulação de um caso de uso, considerando as especificações de uma rede celular do 3GPP. O *Maestro* cumpre com o objetivo proposto, reduzindo a taxa de dados transmitida no *fronthaul* e maximizando a quantidade de recursos utilizados nas nuvens de borda e regionais, tentando maximizar o uso dos recursos computacionais nas nuvens de borda. Ademais, o *Maestro* fornece uma visão geral dos ganhos que podem ser adquiridos através da divisão de funções de rádio em diferentes nuvens e VDUs.

Além disso, pode-se concluir que se o mecanismo de decisão necessita ser executado a cada 3,19 segundos (pior tempo para 160 BSs), consegue-se atender os requisitos do ambiente de uma rede *Dynamic* C-RAN, encontrando uma solução de posicionamento de todos os componentes. Vale ressaltar que quando implantado em uma rede física, faz-se necessário considerar outros parâmetros, tal como a quantidade de tempo para a migração de uma VNF de um VDU para outro, e desta forma, o mecanismo de decisão pode ter seu tempo para calcular o posicionamento, alterado.

Na solução experimental, foi proposta e avaliada uma solução de NFV experimental para ambientes *dynamic* C-RAN com suporte a divisão das VNFs em diferentes VNFCs. A solução baseia-se em políticas do operador da rede, que pode escrever algoritmos com alguma finalidade específica. Na condução da avaliação da solução experimental, foi proposto um algoritmo com a finalidade de maximizar o aproveitamento do processamento nas nuvens de borda, regional e central.

Pode-se concluir, que a solução experimental se mostra viável para os ambientes *dynamic* C-RAN no quesito de implantação de VNFs, suporte a migração das funções e da troca dos VNFCs de lugar, como previsto por esse tipo de ambiente. Outra conclusão que pode-se obter é quanto a latência. Nos cenários propostos na avaliação, em que todas as VMs ficaram hospedadas na mesma máquina física, conforme a quantidade de recursos computacionais vão ficando mais escassos, a latência fim-a-fim aumenta, *i.e.*, para dimi-

nuir a latência fim-a-fim, é necessário um incremento na quantidade de processamento em todas as nuvens. Outra alternativa, é dividir os VNFCs em mais nuvens, com a finalidade de paralelizar o processamento e garantir mais recursos simultâneos. Por outra via, os testes mostraram que os maiores gargalos estão no VIM e VNFM, e não no orquestrador. Os gargalos do VIM e VNFM são predominantemente o tempo na implantação e na migração dos VNFCs. Vale reassaltar que essa condição pode ser bastante ligada a característica do cenário e, pode-se alterar para ambientes em que as VMs não estiverem na mesma máquina física.

## **6.1 Trabalhos futuros**

Como trabalhos futuros, pretende-se avaliar mais extensamente o Orquestra em um ambiente experimental, modificando a infraestrutura em tempo de execução (como por exemplo modificar a capacidade das nuvens ou a latência da rede) e comparar diferentes algoritmos do operador da rede para a mesma solução.

Além disso, pretende-se implantar Maestro e Orquestra para outros tipos de rede, como por exemplo, redes óticas. Esses trabalhos podem ser conduzidos em testbeds, onde os parâmetros de rede e infraestrutura podem ser controlados, desta forma o operador da rede pode escrever algoritmos diferentes e avaliar os impactos que isso causa nos experimentos.

## REFERÊNCIAS

- 3GPP. *Further advancements for E-UTRA physical layer aspects; Evolved Universal Terrestrial Radio Access - TR 36.814*, 2010. Disponível em: <[www.3gpp.org/Specs/36814-900.pdf](http://www.3gpp.org/Specs/36814-900.pdf)>.
- 3GPP. *Small cell enhancements for E-UTRA and E-UTRAN - Physical layer aspects; Technical Specification Group Radio Access Network - TR 36.872*, 2013. Disponível em: <[www.3gpp.org/ftp/Specs/archive/36\\_series/36.872/36872-c10.zip](http://www.3gpp.org/ftp/Specs/archive/36_series/36.872/36872-c10.zip)>.
- 5G-PPP-WORKING-GROUP. *View on 5G Architecture*. Disponível em: <<https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-5G-Architecture-WP-July-2016.pdf>>.
- 5G-PPP-WORKING-GROUP. *View on 5G Architecture (Version 2.0)*. Disponível em: <[https://5g-ppp.eu/wp-content/uploads/2017/07/5G-PPP-5G-Architecture-White-Paper-2-Summer-2017\\_For-Public-Consultation.pdf](https://5g-ppp.eu/wp-content/uploads/2017/07/5G-PPP-5G-Architecture-White-Paper-2-Summer-2017_For-Public-Consultation.pdf)>.
- ABDELWAHAB, S. et al. Network Function virtualization in 5G. **IEEE Communications Magazine**, v. 54, n. 4, p. 84–91, Abril 2016. ISSN 0163-6804.
- BARI, F. et al. Orchestrating virtualized network functions. **IEEE Transactions on Network and Service Management**, v. 13, n. 4, p. 725–739, Dezembro 2016. ISSN 1932-4537.
- BARTELT, J. et al. Fronthaul and backhaul requirements of flexibly centralized radio access networks. **IEEE Wireless Communications**, v. 22, n. 5, p. 105–111, Outubro 2015. ISSN 1536-1284.
- BRYANT, R. et al. **Accelerating NFV Delivery with OpenStack. White Paper**, 2016. 1–24 p.
- CANONICAL. **Linux Containers - LXD**. 2018. Disponível em: <<https://linuxcontainers.org/lxd/>>.
- CARELLA, G. A. et al. Prototyping NFV-based multi-access edge computing in 5G ready networks with open baton. In: **2017 IEEE Conference on Network Softwarization (NetSoft)**, 2017. p. 1–4.
- CHECKO, A. et al. Cloud RAN for Mobile Networks - A Technology Overview. **IEEE Communications Surveys Tutorials**, v. 17, n. 1, p. 405–426, Abril 2015. ISSN 1553-877X.
- CHEN, J. et al. Implementing NFV system with openstack. In: **IEEE Conference on Dependable and Secure Computing**, 2017. p. 188–194.
- CHEN, K.; DUAN, R. **C-RAN The Road Towards Green RAN. White Paper.**, 2011. 1–48 p.
- CHIOSI, M. et al. **Network Functions Virtualisation (NFV)**, 2012. 1–16 p.
- CISCO Network Services Orchestrator Solutions. 2018. Disponível em: <<http://www.cisco.com/c/en/us/solutions/service-provider/solutions-cloud-providers/network-services-orchestrator-solutions.html>>.

CLAYMAN, S. et al. The dynamic placement of virtual network functions. In: **IEEE Network Operations and Management Symposium (NOMS)**, 2014. p. 1–9. ISSN 1542-1201.

DALLA-COSTA, A. G. et al. Maestro: An NFV orchestrator for wireless environments aware of VNF internal compositions. In: **2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)**, 2017. p. 484–491. ISSN 1550-445X.

FUTEBOL, C. **D4.3: Documented control framework**, 2018.

IBM Software. **Efficient modeling with the IBM ILOG CPLEX Optimization Studio**. Somers, NY, 2010. White Paper. Disponível em: <<https://www.ibm.com/software/commerce/optimization/cplex-optimizer/>>.

JANG, I. et al. Joint optimization of service function placement and flow distribution for service function chaining. **IEEE Journal on Selected Areas in Communications**, PP, n. 99, p. 1–1, 2017. ISSN 0733-8716.

KIRST, M. et al. SDR virtualization in future mobile networks: Enabling multi-programmable air-interfaces. In: **IEEE International Conference on Communications**, 2018. v. 1, p. 1–6.

LI, X.; QIAN, C. A survey of network function placement. In: **2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)**, 2016. p. 948–953.

LINGEN, F. van et al. The unavoidable convergence of NFV, 5G, and fog: A model-driven approach to bridge cloud and edge. **IEEE Communications Magazine**, v. 55, n. 8, p. 28–35, 2017. ISSN 0163-6804.

LIU, J. et al. Graph-based framework for flexible baseband function splitting and placement in C-RAN. In: **IEEE International Conference on Communications (ICC)**, 2015. p. 1958–1963. ISSN 1550-3607.

LUIZELLI, M. C. et al. Piecing Together the NFV Provisioning Puzzle: Efficient Placement and Chaining of Virtual Network Functions. In: **IFIP/IEEE International Symposium on Integrated Network Management (IM)**, 2015. p. 98–106.

MAROTTA, A. et al. Impact of CoMP VNF placement on 5G coordinated scheduling performance. In: **2017 European Conference on Networks and Communications (EuCNC)**, 2017. p. 1–6.

MECHTRI, M. et al. NFV Orchestration Framework Addressing SFC Challenges. **IEEE Communications Magazine**, v. 55, n. 6, p. 16–23, 2017. ISSN 0163-6804.

MIJUMBI, R. et al. Management and orchestration challenges in network functions virtualization. **IEEE Communications Magazine**, v. 54, n. 1, p. 98–105, January 2016. ISSN 0163-6804.

MOENS, H.; TURCK, F. D. VNF-P: A model for efficient placement of virtualized network functions. In: **10th International Conference on Network and Service Management (CNSM) and Workshop**, 2014. p. 418–423. ISSN 2165-9605.



NAM, Y.; SONG, S.; CHUNG, J. M. Clustered nfv service chaining optimization in mobile edge clouds. **IEEE Communications Letters**, v. 21, n. 2, p. 350–353, Fevereiro 2017. ISSN 1089-7798.

ORTÍN, J. et al. **D5.2 Final Definition of iJOIN Requirements and Scenarios**, 2014. Disponível em: <<http://www.ict-ijoin.eu/wp-content/uploads/2012/10/D5.2.pdf>>.

PENG, M. et al. Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies. **IEEE Wireless Communications**, v. 21, n. 6, p. 126–135, December 2014. ISSN 1536-1284.

QUINN, P.; GUICHARD, J. Service Function Chaining: Creating a Service Plane via Network Service Headers. **Computer**, v. 47, n. 11, p. 38–44, Nov 2014. ISSN 0018-9162.

QUITTEK, J. et al. **Network Functions Virtualisation (NFV) - Management and Orchestration**, 2014. v. 1, 1–21 p.

RIGGIO, R.; RASHEED, T.; NARAYANAN, R. Virtual network functions orchestration in enterprise WLANs. In: **IFIP/IEEE International Symposium on Integrated Network Management (IM)**, 2015. p. 1220–1225. ISSN 1573-0077.

SOARES, J. et al. Cloud4NFV: A platform for Virtual Network Functions. In: **IEEE International Conference on Cloud Networking (CloudNet)**, 2014. p. 288–293.

TAN, Z.; YANG, C.; WANG, Z. Energy consume analysis for ring-topology TWDM-PON front-haul enabled cloud ran. **Journal of Lightwave Technology**, v. 35, n. 20, p. 4526–4534, Outubro 2017. ISSN 0733-8724.

WANTAMANEE, A. et al. Virtualising machine to machine (M2M) application using open baton as NFV-compliant framework for building energy management system. In: **2016 11th International Conference on Computer Science Education (ICCSE)**, 2016. p. 199–204.

WICKBOLDT, J. A. et al. Resource management in IaaS cloud platforms made flexible through programmability. **Elsevier Computer Networks**, v. 68, p. 54 – 70, 2014. ISSN 1389-1286.

WISHFUL. **Wishful**. 2018. Disponível em: <<http://www.wishful-project.eu/>>.

WUBBEN, D. et al. Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN. **IEEE Signal Processing Magazine**, v. 31, n. 6, p. 35–44, Novembro 2014. ISSN 1053-5888.

XILOURIS, G. et al. T-NOVA: A marketplace for virtualized network functions. In: **European Conference on Networks and Communications (EuCNC)**, 2014. p. 1–5.

XU, Z. et al. Throughput maximization and resource optimization in NFV-enabled networks. In: **2017 IEEE International Conference on Communications (ICC)**, 2017. p. 1–7.

## APÊNDICE A — PAPER PUBLICADO – AINA 2017

*Dynamic Cloud Radio Access Network (Dynamic C-RAN)* é uma emergente arquitetura de redes sem fio que objetiva flexibilidade, agilidade nos negócios, adaptabilidade, além de outros benefícios. Em *dynamic C-RAN*, as funcionalidades das redes sem fio podem ser divididas em pequenos componentes e distribuídas ao longo de uma infraestrutura hierárquica de nuvens. Conceitos de *Network Functions Virtualization (NFV)* tem sido recentemente investigados para facilitar o gerenciamento dessas funções de redes sem fio. Embora as várias vantagens avogadas em favor da divisão das funções, e a efetividade nas soluções de orquestração de NFV, academia e indústria estão considerando *Virtualized Network Functions (VNFs)* como elementos atômicos, não aproveitando as potenciais vantagens da divisão de VNFs em diferentes componentes. Objetivando a melhorar a orquestração de VNFs em cenários de *dynamic C-RAN*, neste trabalho foi proposto Maestro: um orquestrador para ambientes sem fio que é capaz de decidir através de muitas possibilidades de divisão de VNFs qual é a melhor escolha para cada situação. Maestro é desenhado para operar usando diferentes mecanismos de decisão, que podem ser definidos baseados nas necessidades do operador de rede. Também foi avaliada a efetividade da proposta através da modelagem do mecanismo de decisão do orquestrador como um problema de programação linear. Desta forma, foi mostrado como o consumo de taxa de dados no *fronthaul* pode ser reduzido quando comparadas diferentes composições de VNFs ao invés de utilizar uma abordagem de VNFs atômica.

- **Título** – *Maestro: An NFV Orchestrator for Wireless Environments Aware of VNF Internal Composition*
- **Conferência** – The 31st IEEE International Conference on Advanced Information Networking and Applications (AINA-2017)
- **Tipo** – *Main track (full-paper)*
- **Qualis** – A2
- **Data** – 27 a 29 de Março de 2017
- **Realizada em** – Taipei, Taiwan

# Maestro: An NFV Orchestrator for Wireless Environments Aware of VNF Internal Compositions

Ariel Galante Dalla-Costa\*, Lucas Bondan\*, Juliano Araujo Wickboldt\*,  
Cristiano Bonato Both†, Lisandro Zambenedetti Granville\*

\*Institute of Informatics – Federal University of Rio Grande do Sul

†Department of Applied Mathematical and Social Sciences – Federal University of Health Sciences of Porto Alegre  
Email: {agdcosta, lbondan, jwickboldt, granville}@inf.ufrgs.br, cbboth@ufcspa.edu.br

**Abstract**—Dynamic Cloud Radio Access Network (Dynamic C-RAN) is an emerging wireless architecture that aims for flexibility, business agility, adaptability, among other benefits. In a Dynamic C-RAN, wireless functionalities can be split into smaller components and distributed along a hierarchical cloud infrastructure. Network Functions Virtualization (NFV) concepts have been recently investigated to facilitate management-related operations of these wireless functionalities. Despite the many advocated advantages of the function’s splitting and the effectiveness of NFV orchestration solutions, both academia and industry are considering Virtualized Network Functions (VNF) as atomic elements, disregarding the potential advantages of splitting VNFs into several different components. Aiming to improve VNF orchestration in Dynamic C-RAN scenarios, in this paper we propose *Maestro*: an NFV orchestrator for wireless environments that is able to decide among several possible VNF compositions which are more suitable for each situation. *Maestro* is designed to operate using different decision mechanisms that can be defined based on network operators’ needs. We evaluate the effectiveness of our proposal by modeling the orchestrator’s decision mechanism as a linear programming problem. Thus, we show how fronthaul bandwidth consumption can be reduced threefold considering different VNF compositions against atomic VNF placement.

**Index Terms**—Dynamic C-RAN, NFV orchestration, VNF composition

## I. INTRODUCTION

Dynamic Cloud Radio Access Networks (Dynamic C-RAN) move functions from the radio front-end to the network core in a cloud-based infrastructure, providing more flexibility for service provisioning [1]. The dynamic deployment of radio functions in Dynamic C-RAN reduces Capital and Operational Expenditures (CAPEX and OPEX), at the same time it turns the management of such functions easier than the management of traditional dedicated devices. In Dynamic C-RAN, Base Station (BS) processes may be split according to their functionalities and placed in different locations of the infrastructure, which results in benefits and opportunities, such as adaptability, load balancing, service deployment flexibility, and energy efficiency [2].

Despite the benefits of Dynamic C-RAN, coordinating radio functionalities operation is not an easy task, since the path along these functions must support strict requirements, such as low delay and minimum data rate [3] [4]. For this reason, both academia and industry have been exploiting the employment of Network Functions Virtualization (NFV) concepts to

manage and orchestrate virtualized radio functionalities in next generation wireless networks [5]. The NFV architecture allows network operators to design real-time automated orchestration solutions, easing operations such as network functions monitoring, upgrading, and maintenance, when compared to traditional network functions management [6].

NFV platforms such as Cloud4NFV [7] and T-NOVA [8] employ NFV Orchestrators (NFVO) for Virtualized Network Functions (VNF) provisioning. These platforms consider VNFs as atomic elements, *i.e.*, indivisible functions. Despite useful, this approach does not take full advantage of the NFV architecture, whereas VNFs can be presented in different compositions split in different Virtual Deployment Units (VDU), enabling the creation of VNFs that perform the same function, but with different internal composition. For example, a VNF of a Baseband Unit (BBU) may be placed in two different ways: totally deployed in the radio front-end or spread along a Dynamic C-RAN infrastructure, depending on its composition.

Taking into account the lack of NFVO aware of VNF internal compositions, we propose *Maestro*: an NFVO for wireless networks focused on the decision process regarding VNF deployment. *Maestro* is designed to adapt VNF operation based on a catalog with different VNF implementations. This catalog consists of VNFs that perform the same network function but are represented by different VDUs. VNFs with different compositions lead to a decision problem, where the NFVO must decide which is the best VNF composition to deploy, considering the resources available on the Dynamic C-RAN and monitored information regarding VNFs operation.

We advance the state of this research topic through the following contributions: (i) proposing *Maestro*, an NFV orchestrator aware of internal VNF compositions, (ii) detailing the NFV management and orchestration definitions found in the literature and applying these definitions to Dynamic C-RAN, and (iii) presenting a case study and modeling the orchestrator’s decision mechanism as a linear programming problem showing *Maestro*’s advantages in comparison with standard NFV orchestration in a Dynamic C-RAN environment. The results obtained show that, when comparing the use of atomic VNFs and VNFs with multiple VDUs, there is a significant decrease of data rate consumed in the fronthaul network in the order of 70% and, on the other hand, an increase in the VDU placement calculation time.

The remaining of this paper is organized as follows. In Section II, we present the background and related work on Dynamic C-RAN and NFV Management and Orchestration (MANO). In Section III we present *Maestro*, our solution for NFV orchestration in wireless environments. Based on the proposed solution, a case study and an evaluation model based on linear programming are presented in Section IV. In Section V we present and discuss the results obtained during regarding the proposed evaluation model. Finally, in Section VI we present final remarks and perspectives for future work.

## II. BACKGROUND AND RELATED WORK

In this section, we provide a background on Dynamic C-RAN, that is the targeted scenario of our proposal. Next, we discuss important works in this area and motivate our research by discussing the applicability of NFV in Dynamic C-RAN environments.

### A. Dynamic C-RAN

Dynamic C-RAN is a dynamic implementation of Cloud Radio Access Networks (C-RAN) that is a centralized architecture for wireless networks [9]. Different from C-RAN, where all radio-related functions are moved to the cloud, in Dynamic C-RANs baseband functions are usually kept close to the BSs, while higher level functions are deployed in distributed clouds. This distribution has many advantages when compared to C-RAN, such as energy efficiency, cost saving on Capital and Operational Expenses (CAPEX and OPEX), capacity improvement, more flexibility due to the dynamic deployment of functions, and service adaptability. For this reason, 5<sup>th</sup> Generation (5G) Mobile Networks foresees operating in Dynamic C-RANs [1].

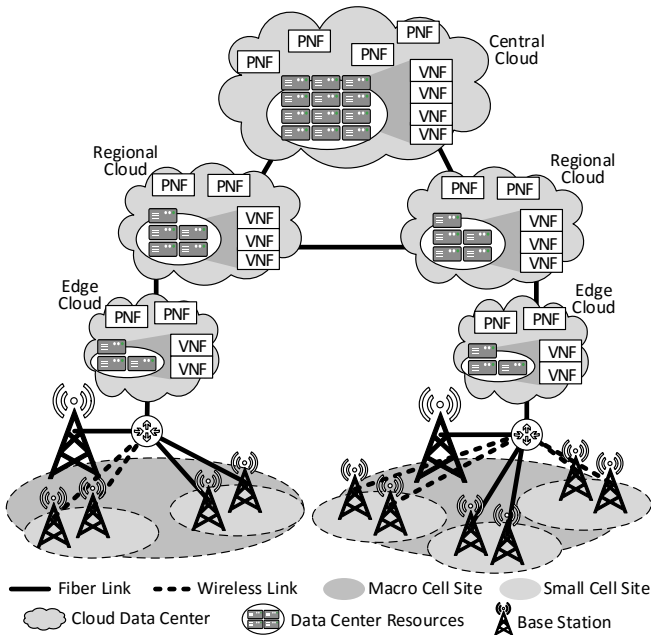


Fig. 1. Example of Dynamic C-RAN scenario

Our Dynamic C-RAN scenario is composed of cloud data centers hierarchically organized, similar to the geographical distribution of clouds in Fog computing scenarios [10], as depicted in Fig. 1. These clouds are composed of both legacy Physical Network Functions (PNF) and generic services for hosting VNFs. The Central Cloud is usually at the operator's central office and provides the highest resource volume in the infrastructure. Regional and Edge clouds are closer to the radio front-end, usually providing fewer resources than the Central Cloud. The hierarchical strategy enables the dynamic placement of virtualized functions, turning service provisioning more flexible [11]. For example, a new virtualized Baseband Unit (BBU) can be deployed on a Regional Cloud to accommodate the increasing number of users in a specific Edge Cloud. Moreover, enforcing load balancing and energy saving policies become easier, due to the flexibility for migrating VNFs across data centers.

Coordinating Dynamic C-RAN with virtualized BBU functions is not an easy task. The proper division of these functions is an important research problem currently explored in this area. In their work, Abdelwahab *et al.*[3] and Liu *et al.*[4] present splitting proposals and strategies for BBU functions. These works recognize that NFV can play an important role in the management and orchestration of virtualized wireless functions [5], [12]. However, using NFV in this kind of network leads to another research problem regarding how to orchestrate virtualized wireless functions across the Dynamic C-RAN infrastructure properly.

In this paper, we abstract the problem of dividing BBU functions, assuming that any validated division can be considered in the orchestration process, such as the proposals of Abdelwahab *et al.* [3] and Liu *et al.* [4]. Our focus resides on the orchestration of VNFs representing BBU functions, considering different compositions for the same VNF. In the next subsection, we provide a detailed explanation of the NFV architecture, focusing on the functional blocks responsible for MANO operations, presenting some works found in the literature regarding NFV orchestration, and discussing their applicability in Dynamic C-RAN environments.

### B. NFV Management and Orchestration

Initiated by the world's leading telecommunications network operators, the NFV Industry Specification Group (ISG) created under the European Telecommunications Standards Institute (ETSI) coordination is working on a consensus regarding the virtualization of network functions, focusing on providing integration among NFV solutions [13]. Initially, the NFV ISG created an architecture where the central elements are the VNFs, with complementary functional blocks involved in their operation. As the NFV concept evolved, the management and orchestration of VNFs became an important issue for the proper operation of NFV environments, leading the NFV ISG to create the NFV MANO group, focused on specifying the operation of these mechanisms [14]. In Fig. 2 we present the NFV architecture proposed by ETSI, detailing the functional blocks regarding NFV MANO.

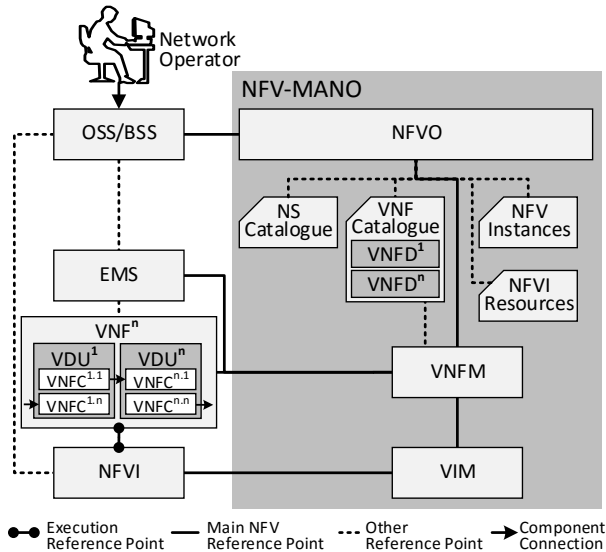


Fig. 2. NFV architecture with detailed MANO blocks

The central elements of the NFV architecture are the VNFs that are software implementations of NFs deployed in the Network Function Virtualization Infrastructure (NFVI). A VNF is composed of VDUs, which are the smallest elements of a VNF, directly mapped on a dedicated virtualization container (e.g., virtual machine, Linux container) [14]. Likewise, each VNF is described by a VNF Descriptor (VNFD) that details how VDUs are connected to compose the VNF. Each instance of a deployed VDU is represented by a VNF Component (VNFC), which also handles the connections points between VDU instances. In other words, a VNF could be represented by different VDUs (i.e., the image of a VM/container for each VNF composition), while each instance of a given VDU is represented by a VNFC (i.e., the VM/container deployed). All these elements must be considered by NFV orchestration solutions designed to operate with VNFs presenting different internal compositions.

All functional blocks of the NFV architecture are connected to NFV MANO elements, responsible for managing and orchestrating services, functions, and resources. Services and functions should be registered by network operators, respecting access control rules enforced by Operations/Business Support Systems (OSS/BSS). Information regarding VNF composition, instantiation, and operation are handled by the NFV Orchestrator (NFVO), which is the NFV element responsible for bringing intelligence to service provisioning and composition processes. For this reason, the NFVO is responsible for two main tasks: (i) resource orchestration across multiple Virtual Infrastructure Managers (VIM), fulfilling resource orchestration requirements, and (ii) life-cycle management of network services composed by VNFs [14], interacting directly with different VNF Managers (VNFM).

NFVO has access to the Network Services (NS) and VNF Catalogs, that maintain information regarding available ser-

vices and functions. Furthermore, the NFVO also has access to NFV instances operating in the NFVI, and to physical and virtual resources of the NFVI. All this information can be used by the NFVO to take decisions about network services and functions operation, performing actions such as scale up or down VNF resources and migrating VNFs across the NFVI. For this reason, the orchestration of NFV services and VNFs is a key process for the proper operation of NFV-based environments, since it has a direct impact on the network performance [15].

Taking into account the important role played by the NFVO in the NFV architecture, some proposals have emerged in the literature. For example, OpenMANO is an open-source based on an implementation of ETSI NFV MANO framework focused on performance and portability [5]. Moreover, NFV platforms, such as Cloud4NFV [7] and T-NOVA [8], propose NFVOs for VNF provisioning focused on end-to-end services. Munoz *et al.* [16] propose an orchestrator for hybrid Software-Defined Networking (SDN) fiber data center environments based on the migration of virtual SDN controllers across heterogeneous infrastructures. All solutions cited above are designed to operate on any network, but they do not provide any directions for the orchestration of wireless-based virtualized functions.

The splitting of wireless functions can substantial important gains for wireless networks. Proposals like Abdelwahab *et al.* [3] and Liu *et al.* [4] highlight the advantages and challenges in splitting BBU's functions, but the orchestration of VNFs composed by different elements remains an open question. At the best of our knowledge, *Maestro* is the first NFV orchestrator able to handle VNFs considering this granularity. In the next section, we present in details the functional blocks of *Maestro*, explaining its operation.

### III. NFV ORCHESTRATOR

*Maestro* was designed to fulfill the lack of NFVOs aware of VNF composition, following the recommendations of ETSI's NFV MANO framework [14]. For this reason, *Maestro* can operate in any NFV environment that follows ETSI's architecture. Fig. 3 depicts in details all functional blocks that compose *Maestro* architecture and their relationship with the NFV functional blocks defined by ETSI.

In an NFV environment, network operators are responsible for selecting and configuring all network services and functions, respecting access control rules enforced by OSS/BSS. *Maestro* communicates with OSS/BSS, enabling network operators to configure services, functions, and the NFVI. Moreover, some functional blocks of *Maestro* can also be configured according to network operators needs. All functional blocks designed in *Maestro* operation are described as follows.

**Register:** According to NFV MANO framework, an NFVO has access to different catalogs regarding available network services and functions, as the instances currently running and resources available in the NFVI. This information may be used by the NFVO to perform actions regarding the operation of

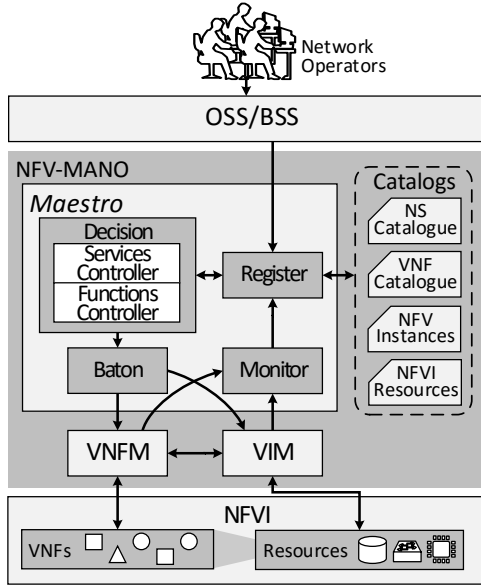


Fig. 3. *Maestro* Architecture

services and functions. In *Maestro* architecture, this information are handled by the Register block. Any new service or VNF must be registered in the catalogs to be available for instantiation in the NFVI.

**Catalogs:** Store data regarding available services, functions, and resources in the network, as the current status of the instances operating on it, following the information model proposed by ETSI NFV MANO [14]. All catalogs are connected to the Register block, which controls the access to this data by the other functional blocks.

**Decision:** Determine which actions will be performed in the network, such as scaling up/down VNF's resources, instantiate a new VNF, or change a VNF composition for another one. The Decision block is the central element of *Maestro* architecture, capable of operating with different decision mechanisms, which could be defined before starting *Maestro*'s operation. Each mechanism can consider different parameters to take its decision, such as data rate requirements, information about physical resources (e.g., CPU and memory usage), latency, delay, response time, and network throughput. Regardless the decision mechanism used, the Decision block uses information residing in the catalogs to decide the actions to be taken. These actions can affect the network at two different levels: service or function. For this reason, the Decision block is subdivided into two modules:

- **Services Controller:** Responsible for managing the Service Functions Chainings (SFC), represented as VNF Forwarding Graphs (VNFFG) in the NFV architecture [17], [13]. Once registered in the catalog, an SFC can be instantiated and modified by the decision mechanism implemented in *Maestro*. This module decides the operation regarding the SFC life-cycle, such as instantiation, validation, authorization, integrity administration,

and visibility. Moreover, in an NFV environment ruled by OSS/BSS policies regarding the SFC operation, this module is responsible for guarantee the enforcement of these policies;

- **Functions Controller:** Responsible for handling VNF-related operations and taking the decisions based on the available information and the decision mechanism implemented. Once registered in the catalog, a VNF will be considered in the decision process, either for deploying a new service or changing a VNF already running in the NFVI for a new one with a different composition.

**Baton:** Once defined the actions to be performed regarding services, VNFs, or resources, they must be sent to the respective blocks. The Baton block is responsible for receiving these actions from the Decision block, translating it into configurations, and forwarding it to the VIM and/or VNFM. These two functional blocks will apply the configurations sent by the Baton block according to their implementation.

**Monitor:** Monitors and collects data regarding VNFs operation and NFVI resources, forwarding all data collected to the Register block. The Monitor block operates in two ways: (i) oriented by events, such as VNF or VM malfunctioning; and (ii) monitoring by polling, performing periodical requests to VNFMs and VIM attempting information such as available resources, data rate consumption, and network latency. Event-related information is generated by VNFM and VIM and does not need the intervention of the Monitor block, while the Monitor periodically requests data in the polling mode.

Both Baton and Monitoring blocks are connected to VIM and VNFM blocks from the ETSI NFV architecture. These two blocks are responsible for handle VNF operation and the resources of the NFVI, respectively. *Maestro* was designed to operate with any implementation of VIM and VNFM that follow the ETSI's recommendations for NFV implementations, such as Cisco NSO [18], OpenStack [19], and Aurora [20]. Changing from one solution to another requires only adapting the abstraction layer present in both Baton and Monitoring blocks of *Maestro*. Once introduced *Maestro* architecture, in the next section we describe a case study and the evaluation modeling designed as a proof-of-concept of *Maestro* operation.

#### IV. CASE STUDY AND EVALUATION MODEL

We defined a case study to demonstrate the efficiency of *Maestro* in selecting an appropriate VNF deployment. Based on this case study, we modeled *Maestro* operation to evaluate the resulting set of VDUs selected and their efficiency in fulfilling a set of predefined requirements. Both the case study and the evaluation modeling are described in details in the next subsections.

##### A. Case Study

Our case study is based on wireless radio functions split presented by Wubben *et al.* [21], applied to the context of Dynamic C-RAN. This way of splitting is interesting in this context due to the variability of the fronthaul data rate demand imposed by each split option, being divided into 5 functions:

MAC, Soft-Bit, RX Data, Subframe, and I/Q. More details about the operation of each function can be found in the work of Wubben *et al.* [21]. This case study allows us to demonstrate the use of NFV MANO elements (VNFs, VDUs, and VNFCs) in the orchestration of Dynamic C-RAN and how different splits may impact in the network performance and resources usage. For this reason, the splits used are based on the communication and computing requirements of the functions that compose the wireless radio functions. Fig. 4 depicts our case study, showing benefits of the split deployment against the atomic split, as proposed by Wubben *et al.* [21]. Besides proposing the splits, the authors defined communication requirements among each of the functions that must be considered by the NFVO in the decision process.

The BSs are depicted at the bottom of Fig. 4, each one with a different occupation. The macro cell (*b*) presents an occupation of 90%, while small cell (*a* and *c*) are 10% and 20% occupied, respectively. The workload of the BSs is distributed into three cloud levels: (i) Edge, close to the BSs; (ii) Regional, a little bit far from the BSs; and (iii) Central, even farther from the BSs, usually on the service provider premises. Each cloud level has a different processing capacity: the farther from the BSs, the greater the computing capacity of it. In our case study, the computational capacity of the cloud is measured regarding of computing unities. The Edge cloud provides 5 computing unities, while Regional and Central clouds provide 7 and 15 computing unities, respectively. Since our objective with this case study is analyzing the fulfillment of network requirements, we simplified the computational analysis considering that each virtualized component (*i.e.*, VNFC) consumes one computing unity. It means that the most computing intensive VNFCs will fill an entire computing unit, whereas less intensive VNFCs will also have one dedicated computing unity, that will be underused.

Assuming as an example that the objective is set for saving fronthaul data rate, the NFVO cannot improve further the allocation shown at the left side of Fig. 4, since the 5 VNFCs must be moved and allocated together based on the same VDU. In this deployment, the aggregated data rate consumption is 4.8 Gbps between Edge and Regional clouds. The atomic deployment is also suboptimal in this case because it creates a fragmented situation regarding computing unities in use, where only 5 of the 7 computing unities available in the Regional cloud are allocated (*i.e.*, 71% of computing resources usage). With no more options to split VNFs, *i.e.*, VDUs, there is no VNFC able to use the remaining 2 computing unities. In the Split-aware deployment, however, the allocations drastically change. For optimizing data rate in the fronthaul, the NFVO can prioritize combinations of splits with smaller aggregated data rate consumption to be allocated, considering the maximal computational capacity of the Edge and Regional clouds. An advantage of this approach is the usage of all computational resources on the Regional cloud, which was not possible in the Atomic deployment due to the indivisibility of the components of the same VNF. The fronthaul data rate consumption optimization is an example of

NFVO objective. Using *Maestro*, network operators can define decision algorithms with different goals.

In the case study presented we consider the BS occupation influences the data rate consumption requirement of functions from RX Data and above, as described by Wubben *et al.* [21]. As a consequence, VDUs containing Subframe and I/Q functions were deployed closer to the BSs, since their data rate requirements are higher. For example, the NFVO could select any function to be deployed at the Central cloud. However, allocating the functions of BS *b* in the Central cloud will consume much more data rate, since the BS *b* presents higher occupation than the others, so the data rate requirement is also higher than *a* and *c*.

In summary, the case study discussed in this subsection allows us to observe two main advantages of the split-aware deployment compared to the atomic deployment: (i) computing resource fragmentation is reduced because of the flexible allocation of different VDU compositions and (ii) the aggregated fronthaul data rate consumption is reduced considering the specific requirements of each split even though the same amount of signal processing workload (same BS occupation) was used in both deployment scenarios. Based on this advantages, we modeled an evaluation as a linear programming problem, explained in details in the next subsection.

### B. Evaluation Model

To evaluate *Maestro* architecture, we defined a decision mechanism based on the case study presented in the previous subsections. The decision mechanism was modeled as a linear programming problem, based on the solution proposed by Luizelli *et al.* [22] for the VNF placement problem. Our decision solution, however, considers computing capacity and fronthaul data rate requirements for choosing the best VDU combination in different datacenters, while Luizelli *et al.* only considered the placement of atomic VNFs in one datacenter. The parameters used in our model are presented in Table I.

TABLE I  
EVALUATION PARAMETERS

Parameter	Description
$V$	VDU list
$H$	Host list
$A$	Base Station list
$DB$	Fronthaul data rate cost
$DV$	VDU computational cost
$CH$	Host computational capacity
$CB$	Host fronthaul data rate capacity
$x$	Tuple $(v, h, a)$ , where $\forall v, h, a \in (V, H, A)$
$\omega$	Data rate decision weight

Our model receives as input the lists of VDUs, hosts, and BSs ( $V$ ,  $H$ , and  $A$ , respectively). For the sake of simplicity, in this experiment we consider one host with a given capacity to represent a whole datacenter, *i.e.*, we disregard the internal organization or topology of a datacenter when calculating fronthaul data rate requirements.  $DB$  is the cost of the VNFC traffic to be transported from the BS to the place where it will be allocated. Similarly,  $DV$  represents the amount of

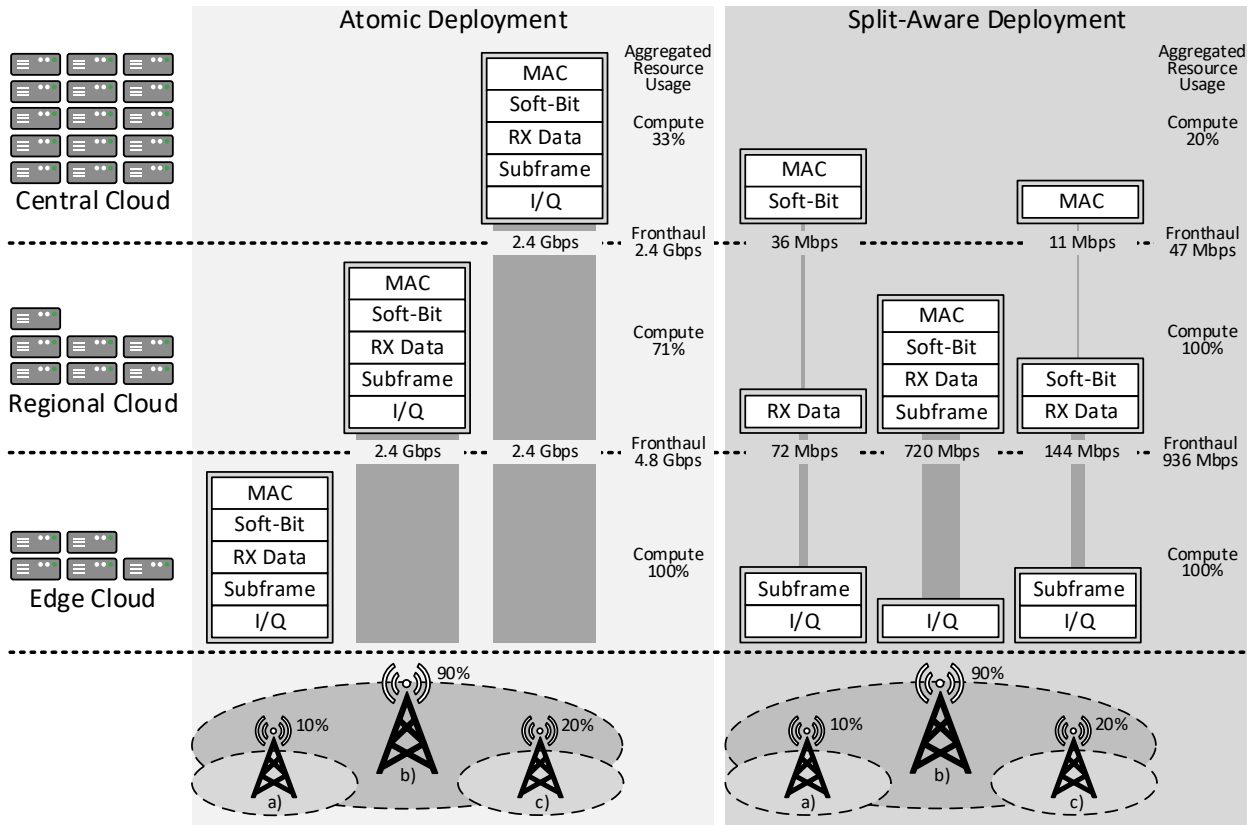


Fig. 4. Atomic and split-aware orchestration examples

computing resources the VDU requires from the hosts.  $CH$  represents the number of computing unities of a given hosts. In the same way,  $CB$  is the maximum data rate that can be consumed in a given host. The combination of VNFC, host, and BS is represented by  $x$ . Finally, we defined a weight for the decision process regarding the importance of the data rate ( $\omega$ ). Once set the parameters of our evaluation, we designed the fitness function of evaluation model as follows.

$$\min \left( \omega \sum_{v=0}^V \sum_{a=0}^A \sum_{h=0}^H x(v, h, a) DB(v)h + (\omega - 1) \sum_{v=0}^V \sum_{a=0}^A x(v, 0, a) DV(v) \right) \quad (1)$$

The objective is minimizing the fronthaul data rate consumption, maximizing the resource usage in the clouds close to the BSs (Edge and Regional). The fitness function must obey two main restrictions: (i) not exceeding the computational capacity of the hosts and (ii) not exceeding the fronthaul capacity. These two restrictions are represented by Equations 2 and 3, respectively.

$$\forall h \in H, \sum_{a=0}^A \sum_{v=0}^V x(v, h, a) DV(v) \leq CH(h) \quad (2)$$

$$\forall h \in H, \sum_{v=0}^V \sum_{a=0}^A x(v, h, a) DB(v) \leq CB(h) \quad (3)$$

Once defined the parameters and the model to be evaluated, in the next, we present and discuss the results obtained during the evaluation of the proposed model.

## V. RESULTS & DISCUSSION

We have implemented the model using IBM's CPLEX Optimizer [23]. The simulation was run on a computer with Intel Core i7 processor 2.4 GHz with 8 GB RAM, running a Linux operating system. We decided to vary the number of antennas in 9 scenarios between 10 and 1000 to understand the impact it would pose to both the fronthaul data rate consumption and the time to calculate VNFC placement. For all scenarios, we considered three datacenters (*i.e.*, Edge, Regional, and Central) with just combined capacity to accommodate the signal processing workload of the antennas proportional to the examples in Fig. 4 and antenna occupation fixed in 50%. All scenarios have been repeated 20 times since the time measured by the CPLEX software to calculate VNFC placement varies in each execution.

Fig. 5 shows the aggregate data rate consumption in the fronthaul network for different forms of VNF split. The x-axis indicates the number of antennas in each experimentation



scenario and the y-axis (log-scaled) shows the amount of aggregate fronthaul data rate consumed in Gbps given the placement of VNFCs calculated by the solver. The *Atomic split* line represents the data rate consumed when only one VDU is used to deploy each VNF, *i.e.*, each antenna will have its five VNFCs positioned in the same cloud. The line *1 and 2 VDU split* shows data rate consumption in the situation in which the solver can use single VDU deployments (like the *Atomic split* line) as well as two VDUs per VNF. Finally, the *1, 2, and 3 VDU split* line show the amount of data rate consumed in the fronthaul by deployments up to 3 VDUs per VNF (similar to the examples in Fig. 4 *Split-Aware Deployment*).

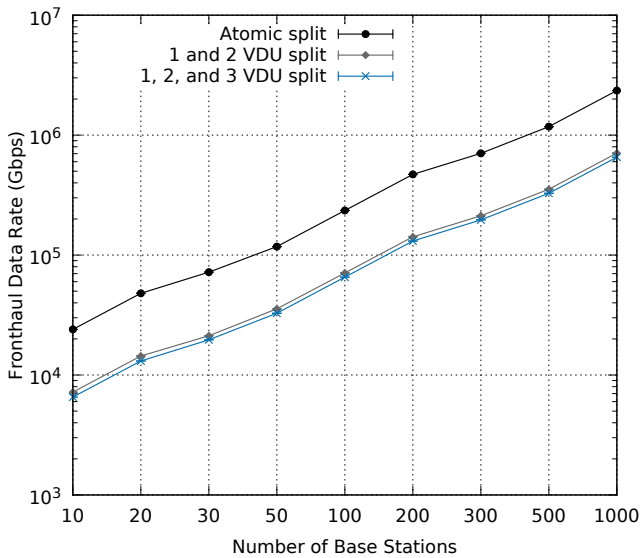


Fig. 5. Aggregate fronthaul data rate consumption

Analyzing Fig. 5, it is possible to visualize that there is a major reduction in data rate consumption when comparing the *Atomic split* line with any of the other two lines for any experimentation scenario. This reduction represents a clear advantage of the *Split-Aware Deployment*, which accounts for a consistent threefold reduction of fronthaul consumption as the amount of VDUs per VNF increases. For example, in the 1000 antennas scenario, the atomic split would consume up to 2.35 Tbps, whereas using 2 and 3 VDUs per VNF these numbers decrease to 708.12 Gbps and 655.2 Gbps, respectively. The gain obtained is not so beneficial when comparing *1 and 2 VDU split* and *1, 2, and 3 VDU split* lines. This small reduction in terms of data rate consumption is possible due to the distribution of VNFCs maximizing the processing in the Edge cloud. The increase in the number of VDUs available for deployment is useful to save fronthaul data rate, but also introduces complexity in the VNFCs' placement process as we discuss next.

Fig. 6 illustrates the average time elapsed to compute the placement of each VNFC in all available clouds. The time measured by CPLEX varies in each execution, so error bars

are shown in the plot representing a confidence interval of 99% for each average time measure. The x-axis shows the number of antennas in each experimentation scenario, while the y-axis (log-scaled) represents the time in seconds required to calculate the placements of VNFCs. As in Fig. 5, the lines represent the atomic VNFs, up to 2 VDU, and up to 3 VDUs split options.

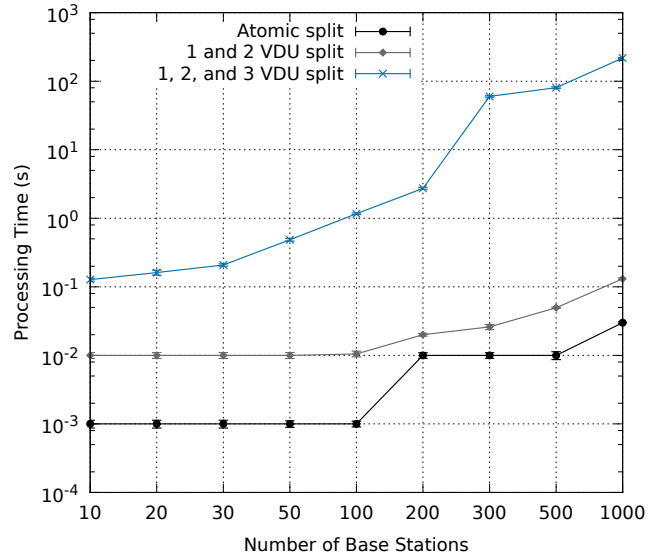


Fig. 6. Time to calculate VNFC placement

From Fig. 6 it is possible to visualize that, as the number of VDUs increases, so does the time for processing the placement of the VNFCs. For the *Atomic split* and *1 and 2 VDU split*, although there is an increase from 100 antennas on, the overall calculation time is almost negligible up to 1000 antennas. The increase in time between *1 and 2 VDU split* and *1, 2, and 3 VDU split*, on the other hand, is substantial. A large increase in processing load can be justified by the number of possible combinations of VDUs that can be used to deploy each VNF.

Comparing both charts of Figs. 5 and 6, it is possible to note that in the former there is gain regarding data rate consumption when considering 3 VDU splits. However, it requires way more processing time for a large number of antennas than 2 VDU splits (217 seconds for 3 VDUs against 0.131 seconds for 2 VDUs for 1000 antennas), which can be an obstacle in scenarios where resource allocations change dynamically.

## VI. CONCLUSION

In this paper, we proposed *Maestro*, an NFV orchestrator for wireless environments aware of VNF internal compositions. The central element of *Maestro* architecture is the decision mechanism, designed for different network operators objectives. Moreover, *Maestro* was designed to operate with any VIM and VNFM implementations following the recommendations of ETSI NFV MANO. As proof-of-concept, we modeled the decision mechanism as a linear programming problem, showing the benefits of VNF composition-aware orchestration.

The results show that *Maestro* has reduced the amount of fronthaul data rate consumed for 1000 antennas from 2.35 Tbps using an atomic split to 708.12 Gbps considering 2 VDUs, and 655.2 Gbps considering 3 VDUs. These results represent a reduction of approximately 69.9% of the data rate consumption for 2 VDUs and 72.1% for 3 VDUs. The reductions of data rate consumption between 2 to 3 VDU splits were in the order of 7.4%. It is also possible to observe an increase of the average time used to calculate the placement of 2 and 3 VNFCs of their respective VDU, increasing from 0.03 seconds in the atomic split to 0.13 seconds in 2 VDUs and 217.2 seconds with 3 VDUs. Through these results, we observe that the improvement in data rate is about the increase in processing time for placement between 2 and 3 VDU splits. This increase in processing time hinders network adaptability facing dynamic resource allocation scenarios.

There are several avenues for future research. First, we plan to improve the experimental modeling by refining the computational capacity analysis considering different needs for each VDU. We also intend to extend *Maestro* operation further by implementing new decision mechanisms based on different network environments. Moreover, *Maestro* can be improved regarding functionalities through new mechanisms to enhance network operation. For example, *Maestro* can consider network policies or indents during the decision process, enforcing network operation according to predefined rules.

#### ACKNOWLEDGMENT

The research leading to these results received funding from the European Commission H2020 programme under grant agreement no. 688941 (FUTEBOL), as well from the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC. The authors would also like to thank Anderson Santos da Silva, Maicon Kist, and Marcelo Luizelli for their important contributions during the elaboration and implementation of this work.

#### REFERENCES

- [1] G. P. W. Group, "View on 5G Architecture," 5G PPP, Tech. Rep., 2016.
- [2] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis, "Fronthaul and backhaul requirements of flexibly centralized radio access networks," *IEEE Wireless Communications*, vol. 22, no. 5, pp. 105–111, October 2015.
- [3] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 84–91, April 2016.
- [4] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "Graph-based framework for flexible baseband function splitting and placement in C-RAN," in *IEEE International Conference on Communications (ICC)*, June 2015, pp. 1958–1963.
- [5] R. Mijumbi, J. Serrat, J. I. Gorricho, S. Latre, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, January 2016.
- [6] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2015.
- [7] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento, "Cloud4NFV: A platform for Virtual Network Functions," in *IEEE International Conference on Cloud Networking (CloudNet)*, Oct 2014, pp. 288–293.
- [8] G. Xilouris, E. Trouva, F. Lobillo, J. M. Soares, J. Carapinha, M. J. McGrath, G. Gardikis, P. Paglierani, E. Pallis, L. Zuccaro, Y. Rebahi, and A. Kourtis, "T-nova: A marketplace for virtualized network functions," in *European Conference on Networks and Communications (EuCNC)*, June 2014, pp. 1–5.
- [9] K. Chen and R. Duan, "C-RAN The Road Towards Green RAN. White Paper." China Mobile Research Institute, Tech. Rep., 2011.
- [10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *SIGCOMM Workshop on Mobile Cloud Computing (MCC)*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16.
- [11] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 126–135, December 2014.
- [12] R. Riggio, T. Rasheed, and R. Narayanan, "Virtual network functions orchestration in enterprise WLANs," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 1220–1225.
- [13] M. Chiosi *et al.*, "Network Functions Virtualisation (NFV)," ETSI NFV ISG, White Paper 1, 2012, available at: [https://portal.etsi.org/NFV/NFV\\_White\\_Paper.pdf](https://portal.etsi.org/NFV/NFV_White_Paper.pdf).
- [14] J. Quittek *et al.*, "Network Functions Virtualisation (NFV) - Management and Orchestration," ETSI NFV ISG, White Paper, 2014.
- [15] L. Bondan, C. R. P. dos Santos, and L. Z. Granville, "Comparing Virtualization Solutions for NFV Deployment: A Network Management Perspective," in *IEEE Symposium on Computers and Communication (ISCC)*, June 2016, pp. 669–674.
- [16] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, T. Szyrkowicz, A. Autenrieth, V. Lopez, and D. Lopez, "Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B62–B70, November 2015.
- [17] P. Quinn and J. Guichard, "Service Function Chaining: Creating a Service Plane via Network Service Headers," *Computer*, vol. 47, no. 11, pp. 38–44, Nov 2014.
- [18] Cisco - Network Services Orchestrator Solutions. Available at: <http://www.cisco.com/c/en/us/solutions/service-provider/solutions-cloud-providers/network-services-orchestrator-solutions.html>. Accessed September, 2016.
- [19] R. Bryant *et al.*, "Accelerating NFV Delivery with OpenStack. White Paper," OpenStack Foundation, Tech. Rep., 2016.
- [20] J. A. Wickboldt, R. P. Esteves, M. B. de Carvalho, and L. Z. Granville, "Resource management in IaaS cloud platforms made flexible through programmability," *Elsevier Computer Networks*, vol. 68, pp. 54 – 70, 2014.
- [21] D. Wubben, P. Rost, J. S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, "Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 35–44, Nov 2014.
- [22] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspar, "Piecing Together the NFV Provisioning Puzzle: Efficient Placement and Chaining of Virtual Network Functions," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 98–106.
- [23] IBM Software, "Efficient modeling with the IBM ILOG CPLEX Optimization Studio," ILOG Optimization and Analytical Decision Support Solutions, Somers, NY, Tech. Rep. WSW14059-USEN-02, August 2010, White Paper. [Online]. Available: <https://www.ibm.com/software/commerce/optimization/cplex-optimizer/>

## APÊNDICE B — PAPER PUBLICADO – SBRC 2017

*Dynamic Cloud Radio Access Network (Dynamic C-RAN)* é uma arquitetura de redes sem fio emergente que objetiva trazer benefícios como flexibilidade e agilidade para redes móveis de quinta geração (5G). Em *dynamic C-RAN*, a carga de processamento das funções de rádio é distribuída ao longo de uma hierarquia de nuvens e, para tal, tem-se investigado o uso de *Network Function Virtualization (NFV)*. NFV tem potencial para oferecer a orquestração das funções de rádio, através da sua subdivisão em componentes de *Virtualized Network Function (VNF)*. Neste trabalho propõe-se uma avaliação detalhada do Maestro – um orquestrador de NFV para ambientes sem fio orientado a composição de VNFs, a fim de estimar os potenciais ganhos ao se aplicar os conceitos de NFV em *dynamic C-RAN*. Através de um modelo baseado em programação linear, avaliou-se a efetividade e custo de orquestração em termos de aderência aos requisitos de comunicação entre as funções de rádio e os posicionamentos dos componentes das VNFs na rede. Além disso, avaliou-se a relação entre tempo para realizar esses posicionamentos e a taxa de dados transmitida, considerando cenários de ocupação variável em *dynamic C-RAN*.

- **Título** – *NFV em Redes 5G: Avaliando o Desempenho de Composição de Funções Virtualizadas via Maestro*
- **Conferência** – XXXV Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (XXXV SBRC 2017)
- **Tipo** – *Main track (full-paper)*
- **Qualis** – B2
- **Data** – 15 a 19 de Maio de 2017
- **Realizada em** – Belém, Pará - Brasil

## NFV em Redes 5G: Avaliando o Desempenho de Composição de Funções Virtualizadas via Maestro

Ariel Galante Dalla-Costa<sup>1</sup>, Matias A. K. Schimunek<sup>1</sup>,  
Juliano Araujo Wickboldt<sup>1</sup>, Cristiano Bonato Both<sup>2</sup>,  
Luciano Paschoal Gaspary<sup>1</sup>, Lisandro Zambenedetti Granville<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul

<sup>2</sup>Departamento de Matemática Aplicada e Ciências Sociais  
Universidade Federal das Ciências da Saúde de Porto Alegre

{agdcosta, makschimunek, jwickboldt}@inf.ufrgs.br

cbboth@ufcspa.edu.br, {paschoal, granville}@inf.ufrgs.br

**Abstract.** *Dynamic Cloud Radio Access Network (Dynamic C-RAN) is an architecture wireless that aims benefits such as flexibility and agility to new five generation (5G) network. In Dynamic C-RAN, the radio functions processing is distributed along of a hierarchical of clouds. Network Functions Virtualization has been investigated the facilities to employ this wireless functions. NFV offers orchestration of split of radio functions into Virtualized Network Functions (VNF). The orchestrator needs to consider strict requirements of C-RAN between distributed components of VNF to get real earnings. In this work, we propose a detailed evaluation of Maestro - an NFV orchestrator for wireless environments aware VNF composition. We evaluate Maestro through of linear model programming, considering the effectiveness and orchestration overhead for deciding the best place to VNF components. In addition, we evaluate the trade-off between network data rate and time to generate the placement of components.*

**Resumo.** *Dynamic Cloud Radio Access Network (Dynamic C-RAN) é uma arquitetura de redes sem fio emergente que objetiva trazer benefícios como flexibilidade e agilidade para redes móveis de quinta geração (5G). Em Dynamic C-RAN, a carga de processamento das funções de rádio é distribuída ao longo de uma hierarquia de nuvens e, para tal, tem-se investigado o uso de Network Function Virtualization (NFV). NFV tem potencial para oferecer a orquestração das funções de rádio, através da sua subdivisão em componentes de Virtualized Network Function (VNF). Neste trabalho propõe-se uma avaliação detalhada do Maestro – um orquestrador de NFV para ambientes sem fio orientado a composição de VNFs, a fim de estimar os potenciais ganhos ao se aplicar os conceitos de NFV em Dynamic C-RAN. Através de um modelo baseado em programação linear, avaliou-se a efetividade e custo de orquestração em termos de aderência aos requisitos de comunicação entre as funções de rádio e os posicionamentos dos componentes das VNFs na rede. Além disso, avaliou-se a relação entre tempo para realizar esses posicionamentos e a taxa de dados transmitida, considerando cenários de ocupação variável em Dynamic C-RAN.*

## 1. Introdução

*Cloud Radio Access Network* (C-RAN) é uma arquitetura de redes sem fio que separa os componentes das *Base Stations* (BS) em *Radio Remote Head* (RRH) e *Baseband Unit* (BBU). Uma RRH é uma unidade de rádio, enquanto uma BBU é uma unidade de processamento de sinais [Checko et al. 2015]. As RRHs estão localizadas na RAN e as amostras de sinais são transportadas através de enlaces de fibra óptica até as BBUs, que são centralizadas em uma infraestrutura de nuvem, também conhecidas como *BBU pool*. *Dynamic C-RAN* é uma evolução do conceito original de C-RAN, que prevê a distribuição das funções de rádio sobre uma infraestrutura de nuvem, como por exemplo, uma nuvem de borda que processa parte das funções de rádio, e uma nuvem regional que realiza o processamento das demais funções [5G-PPP-Working-Group 2016]. Em *Dynamic C-RAN*, o processamento das funções de rádio pode ser dividido de acordo com as suas funcionalidades, sendo distribuídas em diferentes locais da infraestrutura da rede, resultando em benefícios, tais como adaptabilidade, balanceamento de carga, flexibilidade na implantação de serviços e eficiência energética [Bartelt et al. 2015].

Apesar dos benefícios de *Dynamic C-RAN*, coordenar as funções de rádio não é uma tarefa trivial, por exemplo distribuir as funções de rádio ao longo de diferentes níveis da rede necessita de um algoritmo especializado. Esse algoritmo de distribuição deve respeitar requisitos restritos, tais como taxa mínima de dados transmitidos e atraso mínimo. Por esta razão, tanto academia quanto indústria vêm explorando conceitos de *Network Functions Virtualization* (NFV), na coordenação e distribuição das funções de rede. Utilizar NFV em *Dynamic C-RAN*, torna possível mover as funções de rádio para uma nuvem distribuída, virtualizando essas funções, chamadas de *Virtualized Network Functions* (VNF). Desta forma, as características de NFV, como por exemplo, escalabilidade, migração e orquestração, podem ser aproveitadas em cenários dinâmicos, como em *Dynamic C-RAN* [Heideker e Kamienski 2016], [Abdelwahab et al. 2016].

Os conceitos de NFV devem ser adaptados para gerenciar e orquestrar as funcionalidades de rádio virtualizadas na próxima geração de redes sem fio [Mijumbi et al. 2016], permitindo adaptação e dinamicidade ao ambiente 5G. Para tal, faz-se necessário calcular o posicionamento das funções, atendendo aos requisitos restritos do ambiente *Dynamic C-RAN*. Mais especificamente, um orquestrador deve determinar a localização dos componentes de funções virtualizadas, chamados de *Virtualized Network Functions Components* (VNFCs) ao longo das *Virtualized Deployment Units* (VDUs), considerando as restrições temporais das aplicações. Esse processo, conhecido como posicionamento de funções, é de responsabilidade do orquestrador, pois uma mesma função pode ser dividida em VNFs de diferentes formas pré-estabelecidas.

O orquestrador, chamado *Maestro* [Dalla-Costa et al. 2017], foi proposto em um trabalho anterior, que considera a composição de funções virtualizadas na implantação de VNFCs ao longo da rede. Naquele trabalho, conduziu-se uma prova de conceito focando nas características funcionais e na simulação de ambientes *Dynamic C-RAN*. Neste artigo, realiza-se uma avaliação detalhada do *Maestro*, utilizando diferentes composições de VNFs, buscando caracterizar a capacidade deste orquestrador em satisfazer requisitos temporais restritos, impostos pelo ambiente dinâmico de rede de acesso sem fio. Ademais, avalia-se a relação entre o tempo consumido para o cálculo de posicionamento dos VNFCs e a redução da quantidade de dados transmitidos (ganho de até 83,68%) no *fronthaul*. As-

sim, a principal contribuição deste artigo é mostrar o posicionamento dos VNFCs em um ambiente *Dynamic C-RAN*, devido a ocupação das BSs, observando quais cenários e quais características de ocupação influenciam no tempo para realizar tal posicionamento.

O restante deste artigo está organizado da seguinte maneira. Na Seção 2 são discutidos os trabalhos relacionados. Na Seção 3 é descrito e exemplificado um cenário de *Dynamic C-RAN*. Na Seção 4, propõe-se um estudo de caso, descrevendo e discutindo o desempenho do orquestrador. Finalmente, na Seção 5 são apresentadas as conclusões e os trabalhos futuros.

## 2. Trabalhos relacionados

Nessa seção são apresentados os trabalhos relacionados sobre *Dynamic C-RAN* e, posteriormente, discute-se o estado da arte sobre orquestração e posicionamento de funções virtualizadas, através de NFV.

### 2.1. *Dynamic C-RAN*

Diferente de C-RAN, em que o processamento de funções de rádio é movido para a nuvem centralizada, em *Dynamic C-RAN* funções de rádio podem ser distribuídas entre nuvens organizadas hierarquicamente. Essa distribuição possui diversas vantagens potenciais quando comparada a C-RAN, como maior eficiência energética, diminuição da latência, maior escalabilidade, flexibilidade e adaptabilidade de serviços. Por essa razão, redes 5G estão sendo idealizadas em ambientes *Dynamic C-RAN* [5G-PPP-Working-Group 2016]. Trabalhos encontrados na literatura especializada em redes sem fio [Bartelt et al. 2015, Liu et al. 2015] demonstram quais são as possibilidades e as vantagens de dividir as funções de rádio, por exemplo através da divisão do processamento local e remoto em diferentes funções, tais como MAC, Soft-Bit, RX Data, *Subframe* e *In-phase & Quadrature (I/Q)*. Além disso, estudos mostram como essas subdivisões de funções de rádio implicam na taxa de dados transmitida no *fronthaul* [Wubben et al. 2014].

Coordenar *Dynamic C-RAN* por meio de funções de rádio virtualizadas é um tema de pesquisa que está sendo bastante investigado [Abdelwahab et al. 2016, Liu et al. 2015]. Existem trabalhos na literatura que advogam que NFV pode desempenhar um papel importante no gerenciamento e na orquestração de funções de rede sem fio virtualizadas [Mijumbi et al. 2016, Riggio et al. 2015]. Desta forma, *Dynamic C-RAN* pode considerar a virtualização da BBU, utilizando componentes para cada camada de rádio. Entretanto, ao empregar NFV nesse tipo de rede, depara-se com outros dois principais desafios de pesquisa. O primeiro refere-se a como posicionar e orquestrar funções virtualizadas em uma infraestrutura do tipo *Dynamic C-RAN*. O segundo consiste em assegurar desempenho satisfatório para o cálculo do posicionamento de VNFs, de modo a não prejudicar a dinamicidade do ambiente como um todo. Por exemplo, as funções de rádio de retransmissão da camada *Subframe* não suportam latências superiores a 1ms, enquanto na cada MAC o tempo do *Hybrid Automatic Repeat Request (HARQ)* deve ser menor ou igual a 8ms [Ortín et al. 2014]. Outro fator que pode alterar o funcionamento adequado do ambiente *Dynamic C-RAN* é a mobilidade dos *User Equipments (UEs)*, ao longo da infraestrutura, exigindo que o orquestrador gere a rede de acordo com as variações das demandas do usuário. Na Subseção 2.2 são apresentados os trabalhos relacionados sobre orquestração em um ambiente *Dynamic C-RAN* utilizando NFV.

## 2.2. Orquestração baseada em NFV

Dando um passo adiante na área de *Dynamic C-RAN*, foi proposta uma arquitetura para NFV em redes 5G [Abdelwahab et al. 2016], discutindo a implantação de funções de rádio virtualizadas. Essa proposta se assemelha a arquitetura *Management and Orchestration* (MANO) do *European Telecommunications Standards Institute* (ETSI) [Quittek et al. 2014]. A arquitetura do ETSI visa padronizar as implementações, fornecer suporte à indústria e prover gerenciamento e orquestração de funções virtualizadas, definindo um *Network Function Virtualization Orchestrator* (NFVO). Segundo a especificação do ETSI, uma VNF é composta por VNFCs, que são os menores elementos da VNF e são implantados em *containers* virtualizados, ou VDUs. Todas essas VNFs são descritas por um *VNF Descriptor* (VNFD), que representa como os VNFCs são distribuídos e implantados ao longo dos VDUs.

Na área de orquestração em NFV, destacam-se, inicialmente, dois trabalhos relacionados a solução de orquestração. O primeiro refere-se a uma plataforma aberta para NFV, baseado em nuvem, que suporta uma API e outros componentes provenientes do *framework* ETSI MANO [Makaya et al. 2015]. O segundo propõe uma arquitetura modular para NFV, que permite o gerenciamento baseado em políticas de VNFs [Giotis et al. 2015]. Além disso, pode-se destacar outros trabalhos que abordam o posicionamento de funções virtualizadas, tais como T-NOVA [Xilouris et al. 2014] e Cloud4NFV [Soares et al. 2014]. Entretanto, nenhum desses trabalhos considera a composição das funções para a tomada de decisão em um ambiente sem fio, sendo assim, não atendem os requisitos necessários para orquestrar ambientes *Dynamic C-RAN*.

Um orquestrador específico para WLAN foi proposto [Riggio et al. 2015], sem considerar a divisão de componentes de funções, ou seja, não sendo adaptável ao cenário de *Dynamic C-RAN*. Além disso, uma avaliação formal de posicionamento de recursos de funções virtualizadas foi proposta [Moens e Turck 2014], focando em cenários híbridos entre VNFs e funções físicas legadas. Em relação as propostas de posicionamento automático dos nós virtuais e alocação de serviços [Clayman et al. 2014], pode-se destacar a abordagem baseada em heurísticas de *Service Function Chaining* (SFC) [Luizelli et al. 2015, Li e Qian 2016]. Essa abordagem, apesar de considerar o posicionamento de VNFs, não leva em consideração os componentes na tomada de decisão, assim, os tempos para efetuar o posicionamento não são diretamente comparáveis. Desta forma, com a finalidade de avaliar a dinamicidade da orquestração em *Dynamic C-RAN*, neste trabalho avalia-se a arquitetura do orquestrador Maestro, proposto anteriormente [Dalla-Costa et al. 2017].

## 3. Orquestração em *Dynamic C-RAN*

O cenário *Dynamic C-RAN* é composto de uma hierarquia em nuvens, aproveitando a distribuição geográfica das RANs como em ambientes FOG [Bonomi et al. 2012]. A nuvem central geralmente possui a maior quantidade de recursos disponíveis, enquanto as nuvens regionais e de borda são posicionadas próximas ao *front-end*, possuindo geralmente menos recursos. Essa organização hierárquica permite a implantação de funções virtualizadas de forma dinâmica, tornando o provisionamento do serviço mais flexível [Peng et al. 2014]. Por exemplo, uma nova unidade de processamento de sinal virtualizado pode ser implantada em uma nuvem regional para suprir uma determinada demanda,

devido ao aumento do número de usuários na RAN. Além disso, algumas técnicas de balanceamento de carga e de economia de energia tornam-se facilmente aplicáveis nesses cenários, uma vez que existe flexibilidade na migração de VNFs entre os diferentes níveis de nuvem. Assim, dividir e posicionar as funções de rádio que demandam maior transmissão de dados, faz-se necessário através de um orquestrador. Desta forma, na Subseção 3.1 apresenta-se um modelo de posicionamento de VNFs e sua implementação, no orquestrador *Maestro* através de programação linear, é discutida na Subseção 3.2.

### 3.1. Modelo de posicionamento de VNFs em *Dynamic C-RAN*

Considerando um ambiente *Dynamic C-RAN*, contendo um conjunto de  $A$  BSs com um total de  $N$  nuvens (borda, regionais e centrais) e capacidade de  $F$  funções de rádios virtualizadas, o total de dados transmitidos no *fronthaul* pode ser escrito como:

$$O_B(d) = \sum_{f=0}^F \sum_{n=0}^N \sum_{a=0}^A d_{f,n,a} D_B(f, n, a) \quad (1)$$

onde,  $f \in \{0, \dots, F\}$  corresponde as funções de rádio virtualizadas,  $n \in \{0, \dots, N\}$  representa as nuvens disponíveis para processamento dessas funções e  $a \in \{0, \dots, A\}$  simboliza as BSs correspondentes a uma *Dynamic C-RAN*. A matriz tridimensional  $d = [d_{f,n,a}]_{F \times N \times A}$  representa um posicionamento factível para as funções de rádio virtualizadas em cada BSs entre as nuvens. A matriz de posicionamento  $d_{f,n,a} \in \{0, 1\}$ , *i.e.*,  $d_{f,n,a} = 1$  indica que  $f$  foi posicionada sobre  $n$  para processar o sinal de  $a$ . Caso contrário, *i.e.*,  $d_{f,n,a} = 0$ , representa que  $f$  da BS  $a$  não está sendo processada em  $n$ . Finalmente,  $D_B(f, n, a)$  determina a taxa de dados necessária para processar  $f$  da BS  $a$  em uma determinada nuvem  $n$ . A equação  $D_B(f, n, a)$  considera o tipo de  $f$  a ser processada, a distância da nuvem  $n$  (*i.e.*, borda, regionais ou centrais) e a demanda de UEs na BS  $a$ , para determinar a quantidade de dados transmitidos no *fronthaul*. Maiores informações sobre o cálculo da quantidade de dados transmitidos, podem ser encontrados em Wubben, et al. [Wubben et al. 2014].

Baseado nas definições acima, o problema de minimização do total de dados transmitidos no *fronthaul*, através do posicionamento dinâmico das funções de rádio virtualizadas e suas restrições pode ser formulada como:

$$\min_{\{d\}} O_B(d) = \min_{\{d\}} \sum_{f,n,a} d_{f,n,a} D_B(f, n, a) \quad (2)$$

s.t.

$$\sum_n d_{f,n,a} = 1, \quad \forall f, \forall a, \quad (3)$$

$$\sum_{f,a} d_{f,n,a} D_V(f, n, a) \leq C_V^n, \quad \forall n, \quad (4)$$

$$\sum_{f,a} d_{f,n,a} D_B(f, n, a) \leq C_B^n, \quad \forall n, \quad (5)$$

onde,  $D_V(f, n, a)$  determina o consumo dos recursos de nuvem necessários para processar  $f$  da BS  $a$  em uma determinada nuvem  $n$ .  $C_V^n$  e  $C_B^n$  representam os limites de VDUs e de banda do *fronthaul* da nuvem  $n$ , respectivamente.



As restrições (3), (4) e (5) garantem a factibilidade da solução perante o ambiente de uma *Dynamic C-RAN*. A restrição (3) garante que para toda e qualquer BS  $a \in A$  exista obrigatoriamente uma função de rádio virtualizada  $f \in F$ . Além disso, a restrição garante que nenhuma  $f$  de uma mesma  $a$  seja duplicada dentre todas as nuvens possíveis  $n \in N$ . A restrição (4) garante que o posicionamento de uma função de rádio virtualizada  $f$  de uma BS  $a$  seja realizada somente em uma nuvem  $n$ , com capacidade de processamento, *i.e.*, a nuvem deve conter VDUs suficientes para atender todas as funções atribuídas. Por fim, a restrição (5) garante que o posicionamento de uma função  $f$  de uma BS  $a$  ocorra obrigatoriamente em uma nuvem  $n$  com capacidade de transmissão, *i.e.*, uma nuvem só pode receber  $f$  caso o *fronthaul* possua banda suficiente para atender os requisitos de transmissão de todas as funções virtualizadas nessa nuvem.

### 3.2. Programação linear dos recursos em *Dynamic C-RAN* utilizando *Maestro*

O diferencial proposto pelo orquestrador *Maestro* em relação aos trabalhos anteriores é a capacidade de posicionar VNFCs em diferentes VDUs, sobre diferentes nuvens, quando as funções virtualizadas possuem mais que uma opção de divisão. Sendo assim, o *Maestro* pode posicionar os VNFCs de várias maneiras diferentes, possibilitando ganhos na transmissão de dados e na maximização do uso dos recursos computacionais nas nuvens de borda. Desta forma, o orquestrador recebe um conjunto de informações sobre a infraestrutura de rede, definidas pelo operador, e sua utilização, para determinar o posicionamento de cada função virtualizada para todas as BSs dentre as nuvens existentes.

A principal função do *Maestro* é minimizar a taxa de dados transmitidos no *fronthaul*, adotando o modelo descrito na Subseção 3.1. Para acelerar a resolução do problema linear, um novo elemento objetivo é inserido em (2), para maximizar a quantidade de recursos utilizados nas nuvens de borda, uma vez que estas nuvens não transmitem dados no *fronthaul* e devem ser priorizadas pelas soluções. Desta maneira, insere-se uma ponderação na função objetivo, afim de atingir os objetivos propostos. Tal minimização ponderada é descrita como:

$$\begin{aligned} \min_{\{\mathbf{d}\}} \quad & \omega \cdot \sum_{f,n,a} d_{f,n,a} D_B(f, n, a) + (\omega - 1) \cdot \sum_{f,a} d_{f,0,a} D_V(f, n, a) \quad (6) \\ \text{s.t.} \quad & (3)(4)(5) \end{aligned}$$

onde,  $0 \leq \omega \leq 1$  realiza a ponderação das funções objetivos para resolução do problema e  $d_{f,0,a} D_V(f, n, a)$  da prioridade das nuvens de borda no posicionamento das funções virtualizadas. A segunda parte de (6) tem por finalidade maximizar a ocupação de recursos nas nuvens de borda e acelerar a convergência para a solução. A maximização das nuvens de borda e a minimização da taxa de dados transmitidos no *fronthaul* fazem com que os algoritmos consigam obter um melhor desempenho quanto ao tempo, através do direcionamento das possibilidades de posicionamento das VNFs, durante sua execução.

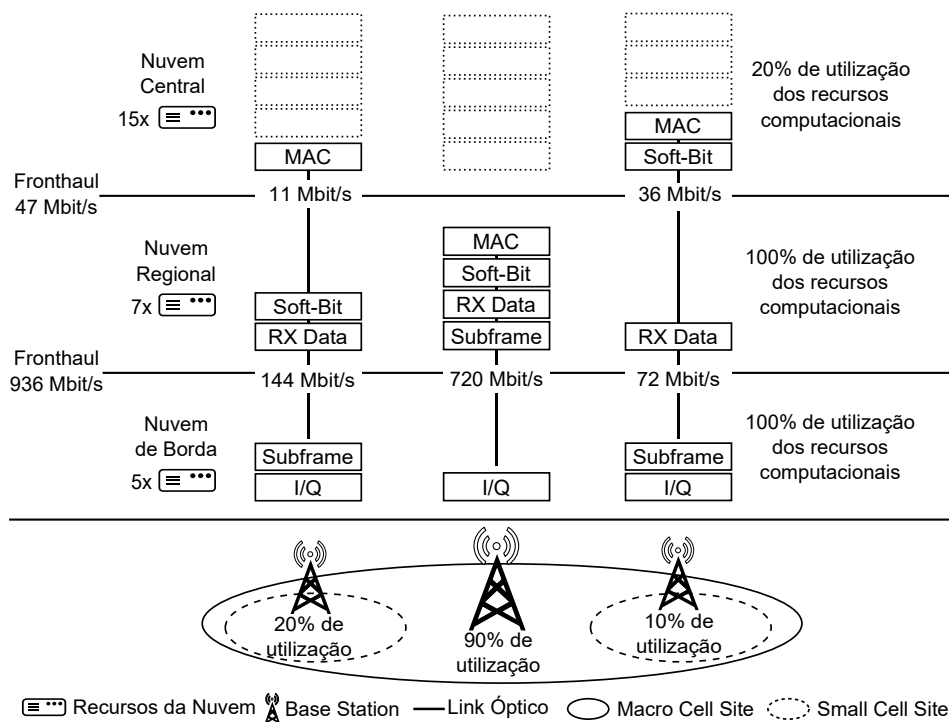
O *Maestro* garante que os componentes de funções de rádio não sejam implantados na mesma nuvem para a mesma BS, que não se exceda a capacidade computacional das nuvens e a capacidade de transmissão disponível no *fronthaul*, respeitando as restrições descritas anteriormente. O *Maestro* foi implementado utilizando o *IBM CPLEX* [IBM Software 2010], com a função objetivo descrita em (6). Na Seção 4 é apresentado o estudo de caso e os experimentos utilizados para avaliação do *Maestro*.

## 4. Estudo de Caso e Experimentos

O estudo de caso e os experimentos apresentados nesta seção visam demonstrar o desempenho do orquestrador *Maestro* em posicionar as composições de funções mais apropriadas para a implantação dos componentes de VNFs. Baseado em uma infraestrutura hierárquica, modelou-se um conjunto de VDUs, juntamente com um conjunto de BSs, que são descritos a seguir.

### 4.1. Estudo de Caso

Este estudo de caso é baseado nas opções de divisão de funções de rádio, descritas por Wubben, et al. [Wubben et al. 2014] e aplicadas no ambiente de *Dynamic C-RAN*. A abordagem proposta por Wubben, et al. prevê a execução distribuída de cinco funções de rádio, a saber: *MAC*, *Soft-Bit*, *RX Data*, *Subframe* e *I/Q*. Essa abordagem objetiva permitir um posicionamento mais flexível dos recursos de computação e de rede necessários para processar o sinal de rádio de forma distribuída, através do uso de virtualização. Maiores detalhes sobre a operação de cada divisão e das próprias funções de rádio são descritas em Wubben, et al. [Wubben et al. 2014]. A decisão acerca da utilização de cada opção de divisão gera uma variabilidade na taxa de dados transmitidos no *fronthaul* e de recursos de computação nas nuvens, o que torna essa abordagem interessante para este estudo de caso. Considerando essa abordagem, é possível mostrar como a orquestração de componentes de VNFs causa impacto no desempenho da rede e na utilização de recursos.



**Figura 1. Exemplo de divisões de funções de rádio implantadas como VNFs**

Na Figura 1 pode-se observar um exemplo de posicionamento das funções de rádio em componentes de VNFs e as suas respectivas distribuições ao longo dos níveis de nuvens disponíveis (*i.e.*, borda, regional e central). No exemplo, cada VNF implementa uma BS, sendo que cada uma das cinco funções de rádio são encapsuladas em um VNFC,

que por sua vez são implantados em até 3 VDUs. As BSs podem ser visualizadas na parte inferior da Figura 1, onde cada uma delas possui uma porcentagem de ocupação diferente em função da capacidade total de transmissão. Essa variação de ocupação influencia o impacto das opções de divisão selecionadas em termos de dados transmitidos no *fronthaul*. No exemplo, a *macro cell* na parte inferior central, tendo ocupados 90% de sua capacidade, foi implantada com 2 VDUs posicionados na borda e na nuvem regional, consumindo uma taxa fixa de 720Mbit/s para comunicação entre as funções de rádio I/Q e *Subframe*. Já para as *small cells*, tendo ocupações de apenas 10% e 20%, optou-se por utilizar 3 VDUs posicionando as funções acima do RX Data – que têm seus requisitos de banda minimizados pelas baixas taxa de ocupação – nas nuvens regional e central.

A lógica por trás do exemplo da Figura 1 é de tentar posicionar os VDUs de forma a minimizar o total de dados transmitidos no *fronthaul*. Consequentemente, a nuvem de borda, que fica mais próxima da RAN, é prioritariamente ocupada com funções de rádio de nível mais baixo que são aquelas que apresentam requisitos mais estritos de comunicação. A nuvem regional, por sua vez, possui maior quantidade de recursos computacionais em relação a nuvem de borda, porém encontra-se mais distante da RAN. Sendo assim, as funções de rádio implantadas na nuvem regional enfrentam atrasos de comunicação para transporte dos dados até esse local. Finalmente, a nuvem central é a nuvem que possui maior quantidade de recursos disponíveis, porém é também aquela que está mais distante da RAN. Essa nuvem se torna mais atraente para o processamento de funções de nível mais alto (*e.g.*, Soft-Bit ou MAC), principalmente para BSs com menor ocupação. Vale salientar que, por medida de simplificação, nesse exemplo considera-se que cada função de rádio encapsulada em um VNFC consome uma unidade de processamento de uma nuvem onde é posicionada. Em um ambiente real, a demanda de processamento potencialmente seria variável em função de fatores como a ocupação das BSs, por exemplo.

Sabendo que os dados transmitidos no *fronthaul* variam de acordo com a ocupação das BSs, para realizar os experimentos foi necessário estimar os percentuais de ocupação com base em uma simulação utilizando as recomendações do *3rd Generation Partnership Project* (3GPP), do documento *Further advancements for E-UTRA physical layer aspects* [3GPP 2010]. Esse percentual de ocupação foi calculado através do método de *Monte Carlo*, em uma simulação do tráfego gerado por UEs associados a uma infraestrutura com 7 *macro cells* e 15 *small cells* distribuídas dentro da área de cobertura de cada *macro cell*, totalizando 112 BSs. Em média, 60 usuários foram distribuídos aleatoriamente dentro da área de cobertura de cada *macro cell* e associados a BS mais próxima de cada usuário (seja uma *macro* ou *small cell*). A distribuição das BSs e UEs foi realizada em uma área de 2000x2000 metros, seguindo as normas para simulação definidas no documento *Small cell enhancements for E-UTRA and E-UTRAN - Physical layer aspects* do 3GPP [3GPP 2013] para uma rede celular, considerando a existência de áreas com diferentes densidades de BS e UEs *i.e.*, zonas rurais e urbanas.

Na Tabela 1 pode-se encontrar a ocupação das BSs utilizadas na simulação, expressa através de uma distribuição discreta de probabilidades, servindo como entrada para o modelo linear detalhado na Seção 3. A ocupação das BSs influencia a taxa de dados transmitidos para as funções de *MAC*, *Soft-Bit*, *RX Data*, tal como é descrito por Wubben et al. [Wubben et al. 2014]. A quantidade de *macro cells* consideradas na simulação cor-

**Tabela 1. Distribuição da Ocupação das *macro* e *small cells***

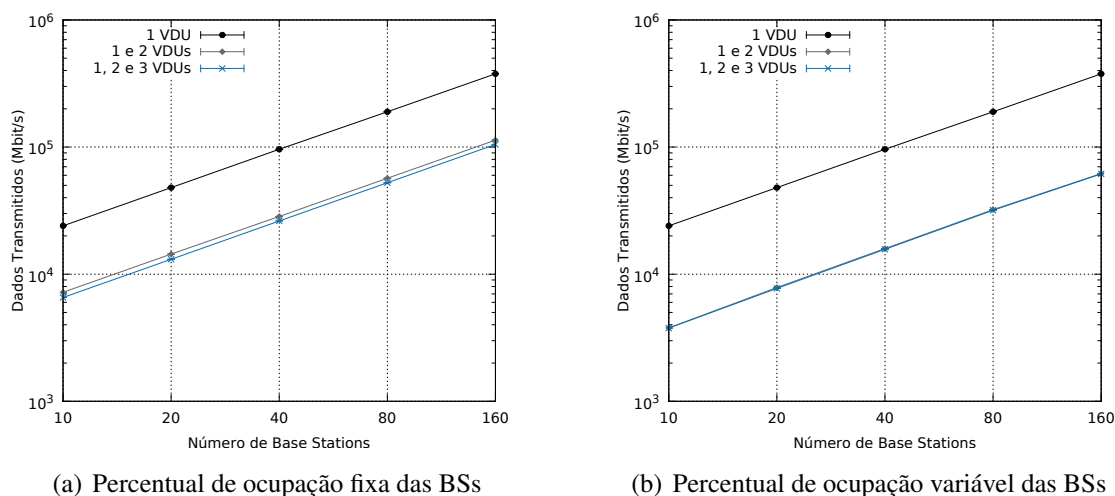
Índice de Ocupação	<i>Macro cells</i>		<i>Small cells</i>	
	Carga de Ocupação	Probabilidade de Ocupação	Carga de Ocupação	Probabilidade de Ocupação
0	15%	0,42%	2,5%	56,97%
1	25%	1,42%	12,5%	31,59%
2	35%	5,14%	22,5%	9,28%
3	45%	10,28%	37,5%	1,88%
4	65%	15,71%	47,5%	0,28%
5	75%	19,17%	-	-
6	85%	17,29%	-	-
7	95%	30,57%	-	-
	Total	100%	-	100%

responde a 33,3% da infraestrutura, enquanto que as *small cells* representam 66,7%. Essa medida se deve à quantidade de UEs distribuídos ao longo da região geográfica, sendo maior para as *macro cells* e menor para as *small cells*. Os cenários de avaliação possuem 10, 20, 40, 80 e 160 BSs ao longo da infraestrutura. Para esse trabalho, considerou-se que cada um dos três níveis hierárquicos de nuvem possui respectivamente 5, 7 e 15 unidades de processamento, como pode ser observado na Figura 1, aumentando essas unidades proporcionalmente à quantidade de BSs em cada cenário. Baseado os cenários de simulação apresentados, na Subseção 4.2 os resultados obtidos são descritos e analisados.

#### 4.2. Resultados

O modelo de programação linear foi implementado com *IBM CPLEX Optimizer* [IBM Software 2010], utilizando 8 *threads* e o mesmo algoritmo *Dual Simplex* para todos os cenários. A simulação foi executada em um computador com um processador Intel i7 de 3.1 GHz, com 16 GB de memória RAM, executando sobre um sistema operacional Linux de 64 *bits*. Para todos os cenários, considerou-se três nuvens (*i.e.*, de borda, regional e central), com capacidade proporcional para acomodar a carga de trabalho nos diferentes cenários, tal como descrito na Subseção 4.1. Todos os experimentos foram repetidos 50 vezes e o tempo para posicionar os VNFCs ao longo dos VDUs, foi medido pelo *IBM CPLEX*. Neste artigo, considerou-se tanto a ocupação fixa das BSs no posicionamento das funções de rádio, quanto ocupação variável (conforme a Tabela 1), diferentemente da avaliação realizada no trabalho anterior [Dalla-Costa et al. 2017]. Desta forma, é possível comparar a quantidade de tempo necessária para efetuar o posicionamento das funções de rádio em ambos os casos.

Pode-se observar na Figura 2(a) a variação total de dados transmitidos no *fronthaul* para diferentes opções de divisão de funções de rádio, considerando que as BSs tenham ocupação percentual fixa em 50%. A Figura 2(b) apresenta o mesmo resultado, porém considerando ocupação variável das BSs (aleatoriamente atribuídos conforme a Tabela 1). O eixo x denota a quantidade de BSs em cada experimento, enquanto o eixo y (em escala logarítmica) denota os dados transmitidos no *fronthaul* em Mbit/s ocasionado pelas opções de divisão selecionadas pela função de posicionamento. A curva *1 VDU* representa a transmissão dos dados considerando uma VNF atômica, utilizando apenas um VDU contendo todos os VNFCs. Já a curva *1 e 2 VDUs* expressa a transmissão dos dados



**Figura 2. Transmissão de dados no *fronthaul***

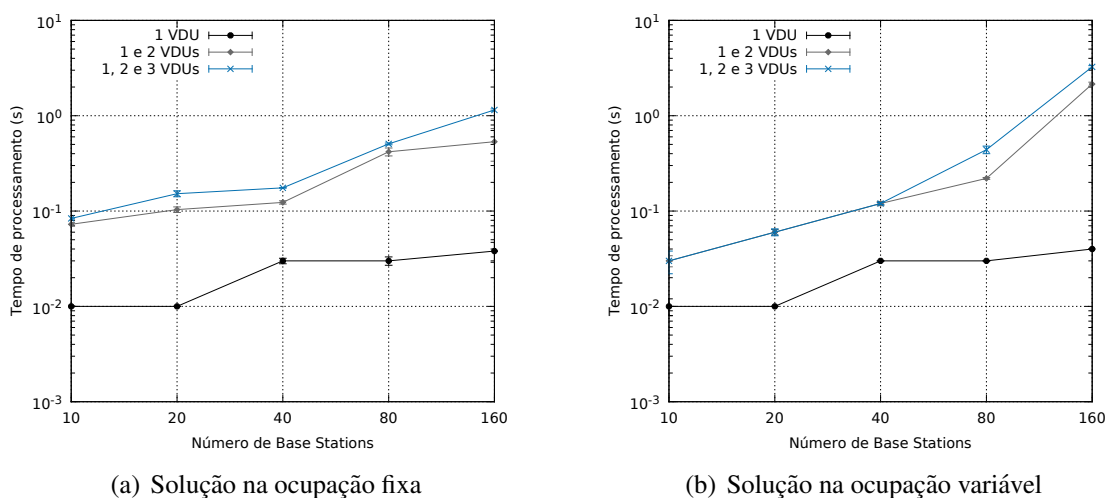
quando VNFs são posicionadas em um ou dois VDUs (*i.e.*, podendo se utilizar uma opção de divisão para separar VNFCs em até dois VDUs). Por fim, a curva *1, 2 e 3 VDUs* representa os dados transmitidos de uma VNF implantada em até 3 VDUs, similar ao que é demonstrado na Figura 1 (*i.e.*, podendo usar simultaneamente até duas opções de divisão para separar os VNFCs de uma mesma VNF).

Observando o gráfico de dados transmitidos com ocupação fixa, para 10 BSs a quantidade de dados transmitido utilizando apenas 1 VDU foi de 24 Gbit/s, enquanto que até 2 e até 3 VDUs esse valor é reduzido para 7,2 Gbit/s e 6,5 Gbit/s, respectivamente. Percentualmente, houve uma redução de 70% entre os cenários utilizando 1 VDU versus 2 VDUs e de 72,7% comparando 1 e 3 VDUs. Entre os cenários de até 2 e até 3 VDUs a diferença foi apenas de 9%. Já no caso mais extremo, considerando 160 BSs, a banda consumida utilizando 1 VDU atingiu 376,8 Gbit/s, enquanto que para até 2 VDUs foi de 113,4 Gbit/s e para até 3 VDUs foi de 104,83 Gbit/s. Assim, manteve-se aproximadamente os mesmos percentuais de redução entre as curvas, enfatizando a importância da composição de VNFs no uso racional dos recursos do *fronthaul*.

Ao observar o gráfico de dados transmitidos com ocupação variável para 10 BSs, a quantidade de dados transmitidos para 1 VDU foi dos mesmos 24 Gbit/s obtidos com ocupação fixa. Isso se deve ao fato de que no posicionamento de 1 VDU pode-se utilizar apenas a opção de divisão da função de rádio I/Q, que requer uma taxa de transferência fixa independente da ocupação da BS. Ainda considerando 10 BSs, as curvas de até 2 VDUs e até 3 VDUs obtiveram taxas de transmissão de dados na ordem de 3,8 Gbit/s e 3,7 Gbit/s, respectivamente, atingindo reduções de aproximadamente 83% em comparação com a curva de 1 VDU. Na comparação dos resultados entre as curvas de 2 VDUs e 3 VDUS, os valores se mantiveram sempre muito similares para todas as quantidade de BSs simuladas com diferenças sempre abaixo de 1%.

Neste trabalho, diferentemente dos trabalhos anteriores, compara-se o desempenho do posicionamento de BSs considerando ocupação fixa e variável em ambientes de *Dynamic C-RAN*. Pode-se observar que tanto na ocupação fixa como na variável, o orquestrador é capaz de minimizar a transmissão de dados no *fronthaul*, quando existir a

possibilidade de seleccionar as divisões de funções de rádio com 2 ou 3 VDUs. Essa redução é ainda mais evidente quando há demanda variável nas BSs, o que se deve principalmente a dois fatores: (i) existe uma diferença de ocupação média significativa entre *macro* e *small cells*, o que fornece ao orquestrador uma gama de opções de VDUs com transmissão de dados diferentes para posicionar; e (ii) apesar da alta ocupação nas *macro cells* (em média 75%), há um grande número de *small cells* com baixa ocupação (em média 8%), o que reduz a ocupação média por BS para a casa dos 30%. A utilização de até 3 VDUs, ainda que acarrete ganhos em relação a até 2 VDUs, traz vantagens muito tímidas no cenário de ocupação variável, o que coloca em dúvida se de fato o vale esforço extra para calcular posicionamentos mais complexos para obter ganhos pouco significativos.



**Figura 3. Tempo necessário para encontrar uma solução**

As Figuras 3(a) e 3(b) ilustram o tempo médio para computar o posicionamento de todos os VNFCs sobre a infraestrutura de nuvens disponível, considerando as BSs com ocupação fixa e variável. As barras de erro representam um intervalo de confiança de 99%, obtido por meio de 50 repetições para cada experimento. O eixo x expressa o número de BSs no cenário, enquanto o eixo y denota o tempo médio em segundos necessário para calcular o posicionamento dos VNFCs.

Os gráficos de tempo demonstram que para até 80 BSs, o tempo necessário para realizar o posicionamento com ocupação fixa é superior ao tempo de ocupação variável. Isso se justifica pois no cenário de ocupação fixa os recursos se esgotam mais rapidamente, de forma que o orquestrador testa mais possibilidades para alocar os VDUs de forma eficiente. Entretanto, para 160 BSs o cenário de ocupação variável consome mais tempo (3,19s para até 3 VDUs, contra 1,12s da ocupação fixa) do que a ocupação fixa. Fato este que pode ser explicado pela grande quantidade de combinações de VDUs com demandas de banda diferentes disponíveis para posicionamento pelo orquestrador.

No cenário de ocupação variável, para realizar o posicionamento de até 40 BSs utilizando 2 ou 3 VDUs, os tempos de cálculo são extremamente similares. Isso torna viável a utilização de divisões de funções de rádio em até 3 VDUs, uma vez que isso leva a uma redução, ainda que pequena, no consumo de recursos do *fronthaul*. Já para 80 e 160

BSs, o tempo necessário para realizar o posicionamento utilizando até 3 VDUs, ocasiona um aumento de mais de aproximadamente 40% no tempo médio de cálculo em relação a até 2 VDUs. Desta forma, pode-se concluir que em um ambiente com uma grande quantidade de BSs, onde a variação da ocupação for frequente, a utilização de até 2 VDUs pode ser mais adequada para permitir que o orquestrador responda mais rapidamente às mudanças do ambiente sem prejudicar significativamente o consumo de recursos.

Finalmente, vale ressaltar que não está sendo considerado neste trabalho o tempo necessário para a implantação/migração dos componentes de VNFs entre as nuvens, o que, em uma infraestrutura real, deverá influenciar no tempo total necessário para que a nova configuração calculada pelo orquestrador entre em vigor. Ademais, os tempos de resposta da modelagem linear sempre são dados até a execução total do posicionamento dos VDUs, de acordo com a função objetivo definida na Seção 3.2. Nesse caso, se essa função for modificada, os tempos de posicionamento serão diferentes.

## 5. Conclusão e Trabalhos Futuros

Neste trabalho, foi realizada uma avaliação do *Maestro*, um orquestrador para ambientes *Dynamic C-RAN*, que baseia-se nos componentes de funções virtualizadas para realizar o posicionamento das VNFs. Seu mecanismo de decisão foi baseado na modelagem de posicionamento de VNFs e em programação linear dos recursos em *Dynamic C-RAN*. A avaliação foi realizada baseada em simulação de um caso de uso, considerando as especificações de uma rede celular do 3GPP. O *Maestro* cumpre com o objetivo proposto, reduzindo a taxa de dados transmitida no *fronthaul* e maximizando a quantidade de recursos utilizados nas nuvens de borda e regionais, tentando maximizar o uso dos recursos computacionais nas nuvens de borda. Ademais, o *Maestro* fornece uma visão geral dos ganhos que podem ser adquiridos através da divisão de funções de rádio em diferentes nuvens e VDUs.

Além disso, pode-se concluir que se o mecanismo de decisão necessita ser executado a cada 3,19 segundos (pior tempo para 160 BSs), consegue-se atender os requisitos do ambiente de uma rede *Dynamic C-RAN*, encontrando uma solução de posicionamento de todos os componentes. Vale ressaltar que quando implantado em uma rede física, faz-se necessário considerar outros parâmetros, tal como a quantidade de tempo para a migração de uma VNF de um VDU para outro, e desta forma, o mecanismo de decisão pode ter seu tempo para calcular o posicionamento, alterado.

Como trabalho futuro pretende-se implantar o *Maestro* em uma infraestrutura real, conduzindo experimentação de forma real, utilizando elementos do *Maestro* para realizar o posicionamento dos componentes ao longo da rede. Além disso, pretende-se adaptar o orquestrador *Maestro* para outros tipos de rede, como por exemplo, para redes ópticas. Outro caminho a seguir, é que o *Maestro* possa considerar fatores (*i.e.* políticas) durante o processo de decisão, de acordo com regras pré-definidas por operadores de rede.

## Agradecimentos

Este trabalho foi financiado pelo programa Horizon 2020 da União Europeia para pesquisa, desenvolvimento tecnológico e demonstração no âmbito do acordo n°. 688941 (FUTEBOL), bem como pelo Ministério da Ciência, Tecnologia, Inovação e Comunicação (MCTIC) através da RNP/CTIC.

## Referências

- 3GPP (2010). Further advancements for E-UTRA physical layer aspects; Evolved Universal Terrestrial Radio Access - TR 36.814. Technical report. Disponível em: [www.qtc.jp/3GPP/Specs/36814-900.pdf](http://www.qtc.jp/3GPP/Specs/36814-900.pdf).
- 3GPP (2013). Small cell enhancements for E-UTRA and E-UTRAN - Physical layer aspects; Technical Specification Group Radio Access Network - TR 36.872. Technical report. Disponível em: [www.3gpp.org/ftp/Specs/archive/36\\_series/36.872/36872-c10.zip](http://www.3gpp.org/ftp/Specs/archive/36_series/36.872/36872-c10.zip).
- 5G-PPP-Working-Group (2016). View on 5G Architecture. Technical report, 5G PPP.
- Abdelwahab, S., Hamdaoui, B., Guizani, M., e Znati, T. (2016). Network function virtualization in 5G. *IEEE Communications Magazine*, 54(4):84–91.
- Bartelt, J., Rost, P., Wubben, D., Lessmann, J., Melis, B., e Fettweis, G. (2015). Fronthaul and backhaul requirements of flexibly centralized radio access networks. *IEEE Wireless Communications*, 22(5):105–111.
- Bonomi, F., Milito, R., Zhu, J., e Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. In *SIGCOMM Workshop on Mobile Cloud Computing (MCC)*, MCC '12, pages 13–16, New York. ACM.
- Checko, A., Christiansen, H. L., Yan, Y., Scolari, L., Kardaras, G., Berger, M. S., e Dittmann, L. (2015). Cloud RAN for Mobile Networks - A Technology Overview. *IEEE Communications Surveys Tutorials*, 17(1):405–426.
- Clayman, S., Maini, E., Galis, A., Manzalini, A., e Mazzocca, N. (2014). The dynamic placement of virtual network functions. In *IEEE Network Operations and Management Symposium (NOMS)*, pages 1–9.
- Dalla-Costa, A. G., Bondan, L., Wickboldt, J. A., Both, C. B., e Granville, L. Z. (2017). Maestro: An nfv orchestrator for wireless environments aware of vnf internal compositions (to appear). *IEEE International Conference on. Advanced Information Networking and Applications*.
- Giotis, K., Kryftis, Y., e Maglaris, V. (2015). Policy-based orchestration of NFV services in Software-Defined Networks. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–5.
- Heideker, A. e Kamienski, C. (2016). Gerenciamento flexível de infraestrutura de acesso público à internet com nfv. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*.
- IBM Software (2010). Efficient modeling with the IBM ILOG CPLEX Optimization Studio. Technical Report WSW14059-USEN-02, ILOG Optimization and Analytical Decision Support Solutions, Somers, NY. White Paper.
- Li, X. e Qian, C. (2016). A survey of network function placement. In *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 948–953.
- Liu, J., Zhou, S., Gong, J., Niu, Z., e Xu, S. (2015). Graph-based framework for flexible baseband function splitting and placement in C-RAN. In *IEEE International Conference on Communications (ICC)*, pages 1958–1963.



- Luizelli, M. C., Bays, L. R., Buriol, L. S., Barcellos, M. P., e Gasparly, L. P. (2015). Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 98–106.
- Makaya, C., Freimuth, D., Wood, D., e Calo, S. (2015). Policy-based NFV management and orchestration. In *Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on*, pages 128–134.
- Mijumbi, R., Serrat, J., I. Gorricho, J., Latre, S., Charalambides, M., e Lopez, D. (2016). Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*, 54(1):98–105.
- Moens, H. e Turck, F. D. (2014). VNF-P: A model for efficient placement of virtualized network functions. In *10th International Conference on Network and Service Management (CNSM) and Workshop*, pages 418–423.
- Ortín, J., Caballero, P., IMDEA, Rost, P., e NEC (2014). D5.2 final definition of ijoin requirements and scenarios. Technical report, INFISO-ICT-317941 iJOIN, <http://www.ict-ijoin.eu/wp-content/uploads/2012/10/D5.2.pdf>.
- Peng, M., Li, Y., Jiang, J., Li, J., e Wang, C. (2014). Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies. *IEEE Wireless Communications*, 21(6):126–135.
- Quittek, J. et al. (2014). Network Functions Virtualisation (NFV) - Management and Orchestration. White paper, ETSI NFV ISG.
- Riggio, R., Rasheed, T., e Narayanan, R. (2015). Virtual network functions orchestration in enterprise WLANs. In *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 1220–1225.
- Soares, J., Dias, M., Carapinha, J., Parreira, B., e Sargento, S. (2014). Cloud4NFV: A platform for Virtual Network Functions. In *IEEE International Conference on Cloud Networking (CloudNet)*, pages 288–293.
- Wubben, D., Rost, P., Bartelt, J. S., Lalam, M., Savin, V., Gorgoglione, M., Dekorsy, A., e Fettweis, G. (2014). Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN. *IEEE Signal Processing Magazine*, 31(6):35–44.
- Xilouris, G., Trouva, E., Lobillo, F., Soares, J. M., Carapinha, J., McGrath, M. J., Gardikis, G., Paglierani, P., Pallis, E., Zuccaro, L., Rebahi, Y., e Kourtis, A. (2014). T-NOVA: A marketplace for virtualized network functions. In *European Conference on Networks and Communications (EuCNC)*, pages 1–5.