Programa de Pós-Graduação em

Computação Aplicada

Mestrado/Doutorado Acadêmico

Luiz Felipe da Silva Cunha

ELASTIC5GC- Elasticidade Proativa no Core 5G para Melhorar a Utilização de Recursos e a Capacidade de Atendimento

São Leopoldo, 2021

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

LUIZ FELIPE DA SILVA CUNHA

ELASTIC5GC- ELASTICIDADE PROATIVA NO CORE 5G PARA MELHORAR A UTILIZAÇÃO DE RECURSOS E A CAPACIDADE DE ATENDIMENTO

SÃO LEOPOLDO
2021

Luiz Felipe da Silva Cunha

ELASTIC5GC- ELASTICIDADE PROATIVA NO CORE 5G PARA MELHORAR A UTILIZAÇÃO DE RECURSOS E A CAPACIDADE DE ATENDIMENTO

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Rodrigo R. Righi

Co-orientador:
Prof. Dr. Cristiano B. Both

São Leopoldo
2021

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Cunha, Luiz Felipe da Silva

ELASTIC5GC- Elasticidade Proativa no Core 5G para Melhorar a Utilização de Recursos e a Capacidade de Atendimento / Luiz Felipe da Silva Cunha — 2021.

75 f.: il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2021.

"Orientador: Prof. Dr. Rodrigo R. Righi, Unidade Acadêmica de Pesquisa e Pós-Graduação".

1. *Core*. 2. 5G. 3. Elasticidade. 4. SBA. 5. séries temporais. 6. redução de recursos. I. Título.

CDU 004.732

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecária responsável: Amanda Schuster — CRB 10/2517

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001 / This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

A minha esposa.

que me apoiou na decisão de realizar esta pós-graduação e que me apoia em todos os momentos de minha vida.

RESUMO

A nova geração de telecomunicações móveis (5G) está em vias de ser implantada em todo o mundo, contudo diversos aspectos da sua implementação estão em aberto. Essa nova geração tem como principal objetivo atender a realidade que está por vir, *i.e.*, alavancada pelos dispositivos IoT, com previsão para 2025 de mais de 79,4 zettabytes trafegados por ano e cerca de 41,6 bilhões de dispositivos conectados. Pensando nesses aspectos e visando estar preparado para essa nova realidade, o 3GPP lançou a *Release* 15. Um dos itens de adequação é a arquitetura baseada em serviços para o núcleo que, dentre outras características, desacopla os serviços de forma que cada um tenha uma responsabilidade específica, facilitando a multiplicação de serviços para atender os mais diversos e dinâmicos cenários. Nesse trabalho é apresentado um modelo para aumentar a capacidade de atendimento aos dispositivos e melhorar a utilização de recursos computacionais do próprio núcleo. Para isso, é proposta uma arquitetura que provê elasticidade horizontal proativa ao núcleo. Essa arquitetura tem como principais componentes (*i*) um balanceador de carga, que distribui todas as comunicações entre a rede de acesso e o núcleo que forem relacionadas aos equipamentos dos usuários para todas as funções de gerenciamento de mobilidade disponíveis e, (*ii*) um gerenciador de elasticidade, que realiza a alocação/desalocação dessas funções, utilizando a predição da carga de processamento das funções de gerenciamento de mobilidade. Essa predição é calculada com base na tendência das medições históricas, utilizando séries temporais, mais especificamente um modelo auto-regressivo integrado de médias móveis. Para evidenciar os resultados, este modelo foi submetido a 3 padrões de carga. Com a utilização desse modelo foi possível obter uma redução de até 38,28% no esforço computacional e um aumento de até 33,22% na capacidade de atendimento. Desta forma, foi possível evidenciar que com a replicação de funções de rede é viável o aumento na capacidade de atendimento e, através da utilização de elasticidade proativa, reduzir drasticamente o uso de recursos computacionais.

Palavras-chave: *Core*. 5G. Elasticidade. SBA. séries temporais. redução de recursos.

ABSTRACT

The next generation of mobile telecomunications (5G) is close to be deployed around the world, however, many aspects of its implementation are open, the main topic of this next generation is to attend the future reality, *i.e.*, that is leverage by IoT, with a prediction of more than 79.4 zettabytes of traffic per year and about 41.6 billions of connected devices. In this way, the 3GGP launched the release 15, one of presented items is a service based architeture to the core that, besides another characteristics, decouples services in a way where each service has a exclusive responsibility, making easily service multipliyng to attend most dynamic and several scenarios. This work shows a model to increase service capacity of the devices and improve the allocation of computational resources in the core. So, an architecture is proposed to provide proactive horizontal elasticity into the core. This architecture has as main components (i) a load balancer, that balancing all communications between the access network and the core related to user equipments for all mobility managing functions available and, (ii) a elasticity manager that does a allocation/deallocation of this functions, using processing load prediction of mobility managing functions. This prediction is calculated throught tendency of historical measurements using time series, more specifically a auto-regressive integrate moving average model. To show results this model was subjected to 3 load patterns. With use of this model was possible to reduct up to 38.38% in allocation of computational resources and to increase of up to 33.22% service capacity. In this way, was showed that networwing functions replications make feasible to increase service capacity and, using proactive elasticity, decrease drastically computational resource usage.

Keywords: Core. 5G. Elasticity. SBA. Time series. Resource reduction.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

3GPP	Third Generation Partnership Project
AMF	Access and Mobility Management Function
ARIMA	Auto-Regresivo Integrado de Médias Móveis
EPC	Evolved Packet Core
eNB	Evolved Node B
gNB	next Generation Node B
MME	Mobility Management Entity
NFVO	Network Function Virtualized Orchestrator
NGAP	Next Generation Application Protocol
NRF	Network Repository Function
OSI	Open Systems Interconnection Model
RAN	Radio Acess Network
SBA	Service Based Architecture
SMF	Session Manager Function
UE	User Equipement
UPF	User Plane Function
VIM	Virtualized Infrastructure Manager
VNFM	Virtualized Network Function Manager
VM	Virtual Machine
VNF	Virtualized Network Function
WS	Web Service

SUMÁRIO

1 INTRODUÇÃO

Vivemos em mundo totalmente conectado e a cada dia que passa mais dispositivos precisam ter conectividade com a Internet para atender as mais diversas demandas. Com o advento da Internet das coisas, a quantidade de dispositivos conectados a Internet tem crescido de forma exponencial, segundo FRAMINGHAM (2019) existirão cerca de 41,6 bilhões de dispositivos conectados, gerando cerca de 79,4 zettabytes em 2025, isso representa um crescimento anual de 28,7%. Dessa forma, as grandes entidades mundiais de padronização de telecomunicações móveis (*3rd Generation Partnership Project* (3GPP)[1] e *International Telecommunication Union* (ITU)[2] tem repensado seus padrões, tendo como principal foco a migração de funções de rede em *hardware* para *software*.

Um movimento importante ocorreu em 2011, com a introdução das redes definidas por software do inglês *Software-Defined Networks* (SDN), que é um paradigma que separa fisicamente o plano de controle do plano de dados, no intuito de mitigar os problemas da infraestrutura de rede como era conhecida. Essa separação pode reduzir o escopo dos equipamentos do plano de dados, passando a ter apenas a função de encaminhamento de dados. Além disso, centralizando o plano de controle em uma camada lógica, simplificando as mudanças e evoluções da rede (Kreutz et al., 2015). Esse paradigma introduziu mudanças importantes para as telecomunicações móveis, uma vez que na infraestrutura concebida com o 4G, o *Evolved Package System* (EPS), boa parte das funções de rede estão embarcadas em *hardware* específicos, de alto desempenho e com custos elevados, tanto para a aquisição, quanto a evolução e manutenção. Diversos trabalhos foram realizados nesse contexto, no intuito de aderir o paradigma SDN ao EPS (Knoll, 2015).

Em 2019, entendendo que o modelo atual de telecomunicação móvel não terá capacidade de atender a nova realidade de mobilidade do mundo, o 3GPP disponibilizou a *Release* 15, que define a nova geração de telecomunicação móvel, aplicando conceitos de SDN, tanto na estruturação da Rede de Acesso de Radio do inglês *Radio Access Network* (RAN), quanto do núcleo (3GPP, 2019a). No núcleo foi definida uma arquitetura baseada em serviços do inglês *Service Based Architecture* (SBA), seguindo os padrões do SDN. Além disso, essa arquitetura é subdividida em plano de controle e plano de dados, composta por serviços com responsabilidades bem definidas e todos definidos em nível lógico, ou seja, em *software* (3GPP, 2019a). Uma vez estando em *software*, esses serviços podem ser implementados em estruturas convencionais como *datacenters* com *hardware* genérico ou mesmo em *cloud computing*.

Essa migração apresenta uma relação custo-benefício implícita, pois de um lado temos que quanto mais perto do *hardware* menor é a utilização de recursos computacionais, porém com alto custo. Já no outro lado, quanto mais abstrato em *software* menor o custo, porém com maior consumo de recursos computacionais. Esse é um problema comumente resolvido através da

[1] https://www.3gpp.org/
[2] https://www.itu.int/

utilização de elasticidade, pois a alocação dos recursos é dinâmica e diretamente proporcional às necessidades do sistema em um determinado instante ou período.

1.1 Motivação

A aplicação de elasticidade para o plano de controle de redes de telecomunicação móvel vem sendo explorada ao longo dos últimos anos, e a segmentação dos serviços na arquitetura SBA simplifica esse trabalho. Algumas propostas de elasticidade no núcleo SBA foram realizadas (BUYAKAR et al., 2019; ALAWE et al., 2018). Entretanto, apenas foram utilizadas metodologias reativas e os testes foram aplicados, ou em modelos analíticos, ou em pseudo aplicações, sem que de fato fosse utilizada uma aplicação que implementasse todos os componentes e funções do núclo SBA.

Como o núcleo SBA é recente, as pesquisas utilizando EPC são mais robustas e propostas de elasticidade (BANERJEE et al., 2015; SALHAB; RAHIM; LANGAR, 2019) foram testadas em aplicações que possuem uma implementação completa. Algumas propostas (NGUYEN et al., 2018; AMOGH et al., 2017) apesar de apresentarem soluções que viabilizem a replicação de serviços, não aplicam técnicas de elasticidade. A utilização de elasticidade proativa é utilizada no núcleo EPC na proposta apresentada por Banerjee et al. (2015), porém, além de ser aplicado na quarta geração de telecomunicação móvel, as ações de elasticidade são baseadas em épocas, o que gera um intervalo de tempo em que o comportamento da rede pode ser alterado. Dessa forma, o sistema pode ficar sobrecarregado ou com pouca carga, mas nenhuma alocação/desalocação de recurso seria realizada.

Pensando em explorar as características do núcleo SBA, que provê funções com responsabilidades únicas e bem definidas, além do fato de serem totalmente em software, uma boa oportunidade de pesquisa é o aumento da capacidade de atendimento e a melhora do uso de recursos computacionais.

1.2 Questão de pesquisa

A questão de pesquisa que o modelo proposto busca responder é: *É possível implementar um modelo de elasticidade proativo para o core 5G definido na Release 15 (3GPP, 2019a), que seja transparente para a rede de acesso, melhorando a utilização de recursos computacionais?*

Hipótese: *A elasticidade horizontal proativa pode melhorar a utilização e consumo dos recursos computacionais e aumentar a capacidade de atendimento aos dispositivos de usuários.*

Esse trabalho propõe um modelo que define e executa ações de elasticidade para o núcleo SBA. Esse modelo foi denominado Elastic5GC e visa aumentar o desempenho da aplicação, reduzindo o tempo de resposta aos usuários (*User Equipment* - UEs) e tornando eficiente o uso de recursos computacionais. Dessa forma, os serviços devem se manter ativos, apenas quando sua utilização é necessária, para que, por exemplo, outros serviços que estejam concorrendo na

utilização de recursos possam utilizá-los. Nesse contexto, elasticidade proativa é aplicada para auxiliar na tomada de decisões antecipadas e assim prever quando e quais recursos devem ser alocados/desalocados, obtendo um desempenho melhor do que com a utilização de elasticidade reativa. Além disso, o modelo é transparente para a rede de acesso, ou seja, não deve ser necessário realizar quaisquer modificações nos UEs e RANs para que se conectem ao núcleo.

1.3 Contribuições e Resultados

Para endereçar a oportunidade de pesquisa foi proposto o modelo Elastic5GC. Esse modelo apresenta uma solução de elasticidade horizontal proativa para a camada de controle do núcleo 5G, sendo elaborado um gestor de elasticidade que, com base na utilização de CPU, aloca ou desaloca AMFs. A predição de carga é feita através da utilização de séries temporais, mais especificamente o modelo ARIMA. Para deixar transparente para a RAN foi incluído um balanceador de carga a frente dos AMFs, assim, não se faz necessária qualquer alteração na RAN ou nos AMFs.

Foram obtidos resultados expressivos tanto para redução de utilização de recursos, tendo como melhor resultado uma redução de 38,28%, quanto para capacidade de atendimento, tendo como melhor resultado um aumento de 33,22%. A implantação desse modelo converge para os objetivos do 5G, principalmente quanto ao tema de latência, logo, esse modelo contribui para a viabilização de soluções como cidades inteligentes, *cyber* medicina e tantas outras soluções que dependem de baixa latência.

1.4 Organização do Texto

Esta monografia está estruturada em sete capítulos. Após a introdução apresentada no Capítulo 1, os conceitos relacionados a esta monografia são apresentados no Capítulo 2, introduzindo a área de telecomunicação móvel, a elasticidade, as funções de redes virtualizadas e demais conceitos que auxiliam no entendimento do modelo. Em seguida, no Capítulo 3 são discutidos os trabalhos relacionados de maneira a apresentar o estado da arte envolvendo elasticidade para o núcleo de telecomunicação móvel. O Capítulo 4 apresenta a proposta do modelo de elasticidade para serviços do núcleo 5G, bem como as decisões de projeto e detalhes de implementação. No Capitulo 5, são apresentados a modelagem de testes e o protótipo. Já no Capitulo 6, são apresentados os resultados da aplicação do modelo no protótipo, bem como um comparativo com o estado da arte. Por fim no Capítulo 7, são apontadas as considerações finais sobre modelo, enfatizando as contribuições, limitações e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os itens fundamentais para a realização desse trabalho. Na Seção 2.1 são apresentados os conceitos referentes a telecomunicação móvel e as funções de rede virtualizadas. Enquanto, na Seção 2.2 são descritos conceitos referente a elasticidade e a predição de carga de trabalho.

2.1 Telecomunicação Móvel e Funções de Rede Virtualizadas

A telecomunicação móvel está passando por um momento de transição com a aplicação de conceitos de SDN e com o advento de Internet das Coisas. Desta forma, para viabilizar esta transição uma nova geração de telecomunicação móvel (5G) foi definida (3GPP, 2019a). Nesta seção, os principais conceitos aplicados pela nova geração de telecomunicação móvel são apresentados.

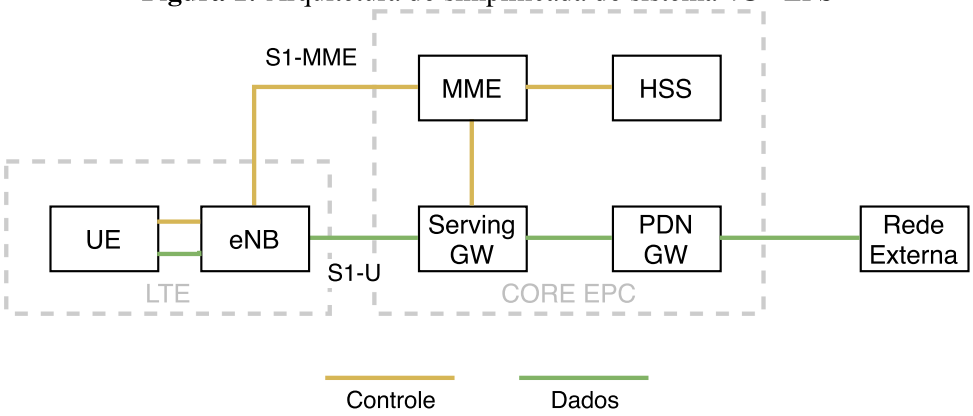
2.1.1 Rede Definida por *Software* e Função de Rede Virtualizada

Função de rede virtualizada do inglês *Network Function Virtualized* (NFV) é o desacoplamento do equipamento físico de rede das funções que operam sobre ele, ou seja, uma função de rede (*e.g.*, como um *firewall* ou *roteador*) pode ser definida como parte de um *software* (MIJUMBI et al., 2015). Uma vez que as funções podem ser aplicadas em *software* torna-se viável ter uma rede, onde o *hardware* tem menos responsabilidades. Dessa forma, isto permite que tenha-se uma rede definida por software, do inglês *Software-Defined Networking* (SDN), que é um paradigma/arquitetura de rede que separa a lógica de controle (plano de controle) dos roteadores e *switches*, fazendo o encaminhamento do tráfego (plano de dados). Nesta arquitetura, o plano de controle fica remoto e centralizado e gerencia o plano de dados de forma desacoplada, isto faz com que os *switches* passem a ser somente dispositivos de encaminhamento. Esta lógica de controle centralizada torna mais simples a aplicação de políticas e a reconfiguração e evolução da rede (Kreutz et al., 2015).

2.1.2 *Core* de Telecomunicação Móvel - 4ª Geração

A quarta geração de telecomunicação móvel (4G) utiliza o sistema o *Evolved Package System* (EPS) que estabelece uma arquitetura completa para uma rede telecomunicação móvel. O componente central deste sistema é o *core* da rede chama-se *Evolved Package Core* (EPC). Esse *core* tem a responsabilidade de prover serviços para os consumidores que estão conectados na rede de acesso. A Figura 1 apresenta uma versão simplificada da arquitetura do EPS. Neste sistema, existe uma rede de acesso *Long Term Evolution* (LTE) que possui 2 componentes básicos, UE, que é o equipamento ao qual os usuários se conectam a rede e, o *Evolved NodeB* que

Figura 1: Arquitetura do simplificada do sistema 4G - EPS



Fonte: Adaptado de (FIRMIN; 3GPP MCC, 2018)

é a estação de rádio base (BS) do LTE. Essa BS realiza, entre outras funções, a centralização da comunicação do UE com o *core* (3GPP, 2017; FIRMIN; 3GPP MCC, 2018). Além disto, o 3GPP (2017) define os componentes do *core* EPC, tendo os seguintes principais componentes:

***Mobile Management Entity* (MME):** componente central da camada de controle e possui como principais funções a seleção do PDN GW e do *Serving* GW, autenticação, sinalização e segurança, mobilidade e *roaming*.

***Serving Gateway* (Serving GW):** *gateway* da rede de acesso, ou seja, todas as informações do plano de dados oriundos da rede de acesso para o *core* tem como entrada este serviço e assim realizando roteamento e encaminhamento de pacotes.

***Package Data Network Gateway* (PDN GW):** este *gateway* é responsável, entre outras coisas, por prover conectividade para os usuários e alocar endereço de IP para o UE.

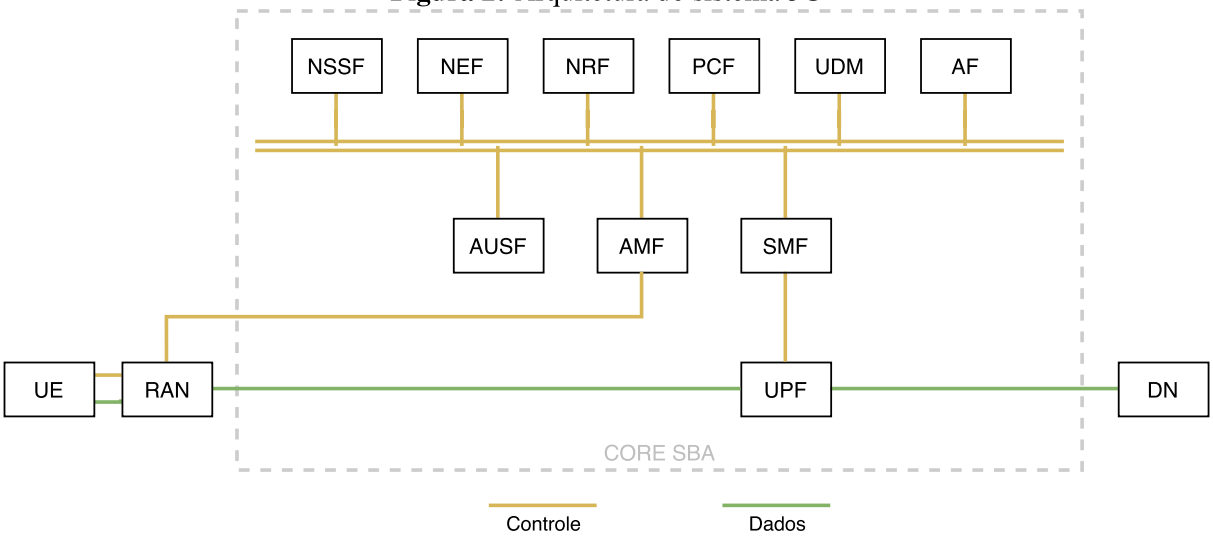
***Home Subscriber Server* (HSS):** é o banco de dados mestre dos dados de usuários, ele possui os dados de identificação (numeração e endereçamento), de segurança usados para autenticação e autorização, de localização e de perfil.

2.1.3 *core* de Telecomunicação Móvel - Nova Geração

A realidade atual das redes e as projeções de futuro das redes móveis (FRAMINGHAM, 2019) fizeram com que os requisitos dessas redes fossem redefinidos, tendo como base nos novos serviços e mercados. O 3GPP (2019b), com o lançamento da *Release* 15, descreve estes requisitos tendo em vista diferentes tipos de uso, conforme a seguir:

***Enhanced Mobile Broadband* (eMBB)** : alta densidade de conexão e tráfego, mobilidade e taxa de dados. Para cada instância o *downlink* deve ser de no mínimo 50 Mbps em locais

Figura 2: Arquitetura do sistema 5G



Fonte: Adaptado de (3GPP, 2019a)

abertos e no mínimo 1 Gbps em locais fechados (5GLAN) e metade destes valores para *uplink* (3GPP, 2019a).

***Critical Communications* (CC) e *Ultra Reliable and Low Latency Communications* (URLLC)** : baixa latência e alta disponibilidade na comunicação entre serviços. Para cada instância, deve possuir 99,9999% de confiabilidade no controle remoto para automação de processos com uma taxa mínima de 100 Mbps e uma latência ponta-a-ponta de no máximo 50 ms (3GPP, 2019a).

***Massive Internet of Things* (mIoT)** : muitos cenários farão com que o sistema 5G tenha que suportar uma alta densidade de dispositivos. Para estes casos é necessária a inclusão de aspectos operacionais que viabilizam a ampla variedade de dispositivos IoT (3GPP, 2019a).

Para viabilizar estes múltiplos cenários, o *core* 5G precisa ser altamente adaptável e para tal, foi definida uma arquitetura baseada em serviços (SBA - *Service-Based Architecture*) conforme pode ser visto na Figura 2. Nesta arquitetura, as funções de rede são separadas em serviços em uma granularidade que garante responsabilidade única. Todos os serviços SBA se comunicam através de protocolo de aplicação HTTP, utilizando RESTFul nível 2, na escala *Richardson Maturity Model* (3GPP, 2019a; FOWLER, 2010).

O 3GPP (2019a) define cada um destes serviços da seguinte forma:

AMF (*Access and Mobility management Function*) : responsável pelo controle de acesso, autenticação e autorização dos UEs, registro de área, controle de mobilidade, suporte para o *Network Slicing* e Seleção do SMF.

SMF (*Session Management Function*) : responsável pelo controle de sessão, definição dos endereços IP dos UEs, seleção e controle do UPF e configuração da direção do tráfego no UPF para rotear os dados para o destino correto.

UPF (*User Plane Function*) : responsável pelo roteamento e redirecionamento de pacotes, inspeção dos pacotes e parte da execução das políticas, reportar o uso de rede e tratamento da regras de *Quality of Service* (QoS) para o plano de usuário.

NRF (*Network Repository Function*) : responsável pelo gerenciamento dos NFs, incluindo registro, cancelamento de registro, autorização e descoberta.

NEF (*Network Exposure Function*) : expõe o que pode ser realizado pelos serviços da rede de forma externa.

UDM (*Unified Data Management*) : responsável pelo gerenciamento dos dados (UEs, Políticas, Sessões, etc.) de forma unificada.

AF (*Application Function*) : são funções de aplicação customizadas que podem ser incorporadas a rede.

NSSF (*Network Slice Selection Function*) : é uma função de rede dedicada para a seleção de *Network Slice*, ou seja, seleciona os serviços com características específicas para atender um determinado cenário de rede bem definido.

AUSF (*Authentication Server Function*) : serviço que viabiliza a autenticação unificada de acessos 3GPP e "não-3GGP".

PCF (*Policy Control Function*) : é o serviço responsável pela definição e entrega das políticas para os serviços da rede.

A comunicação entre o *core* e a rede de acesso ocorre através de duas interfaces: (*i*) a interface N2 que interconecta gNB e AMF para a comunicação do plano de controle e (*ii*) a N3 que interconecta gNB e UPF para a comunicação do plano de dados (3GPP, 2019a). Na interface N2 é utilizado o protocolo *Next Generation Application Protocol* (NGAP), neste protocolo são providos dois tipos de serviços:

- os associados aos UEs, que são todas as sinalizações que partem do UE como, por exemplo, o registro do dispositivo na rede, e;
- os não associados ao UEs, que são as comunicações do próprio eNB com o *core* como, por exemplo, o registro do eNB na rede (3GPP, 2018).

Este protocolo opera sob o protocolo de transporte *Stream Control Transmission Protocol* (SCTP) e sob o protocolo de rede *Non-Access stratum* (NAS), que provê gestão de mobilidade

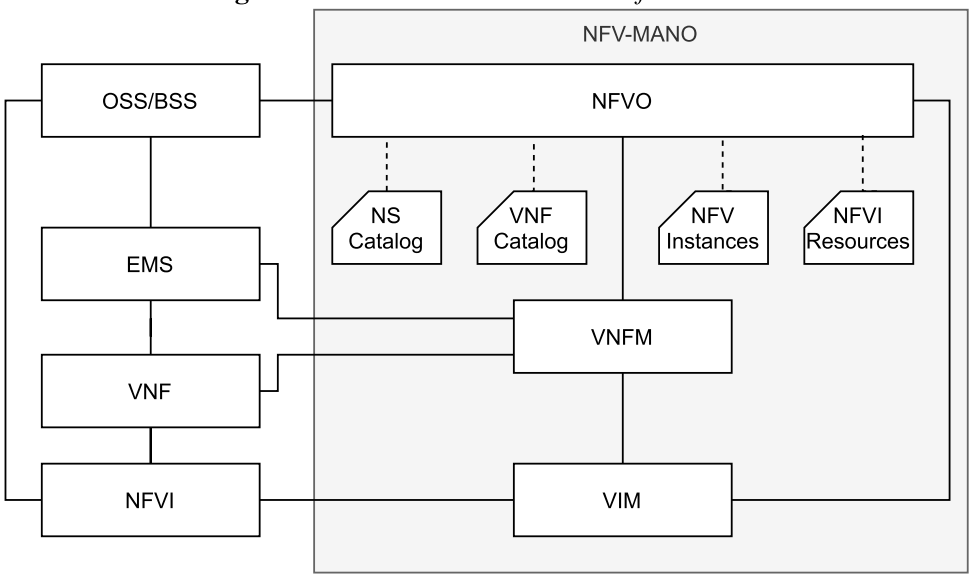
e de sessão entre o UE e o AMF (3GPP, 2018). Com a introdução desses serviços no *core* das redes móveis, surgiu a necessidade de modelos de gerenciamento dessas funções virtualizadas. Um exemplo, largamente utilizado, é o NFV-MANO, descrito na subseção seguinte.

2.1.4 *Network Functions Virtualization Management and Orchestration* - NFV-MANO

O NFV-MANO é um *framework* arquitetural que possui a responsabilidade de gerenciar a infraestrutura que suporta as redes virtualizadas, bem como orquestrar a alocação de recursos necessários para as funções de rede, tanto virtualizadas com não virtualizadas (ETSI, 2014). A Figura 3 apresenta esta arquitetura.

Dentro dos serviços presentes no NFV-MANO pode se destacar: (*i*) NFV *Orchestrator* (NFVO), (*ii*) VNF *Manager* (VNFM) e (*iii*) *Virtualized Infrastructure Manager* (VIM). NFVO é responsável pela orquestração das instâncias de funções de redes virtualizadas, mantendo um catálogo para permitir que demais instâncias tenham total conhecimento de funções disponíveis. Além disso, é esse serviço que realiza as chamadas para o VNFM e VIM para que realizem as alocações/desalocações de recursos e de funções de rede. VNFM é o responsável pela gestão das funções de rede virtualizadas, instanciando/removendo funções e coletando informações sobre as mesmas. VIM é o responsável pela gestão da infraestrutura que permite a virtualização de serviços, tendo como principal função a alocação/desalocação de recursos.

Figura 3: NFV-MANO *architetural framework*



Fonte: Adaptado de (ETSI, 2014)

2.2 Elasticidade e Previsão de Carga

Nesta seção são apresentado os conceitos de elasticidade e previsão de carga para buscar melhor desempenho no *core* 5G.

2.2.1 Elasticidade

Elasticidade é a capacidade de aumentar ou diminuir a quantidade de recursos disponíveis sem a interrupção do serviço e assim atender as variações de demanda (GALANTE; BONA, 2012). A elasticidade pode ser provida em três métodos e pode ser aplicada segundo dois modelos (COUTINHO et al., 2015). Os métodos de elasticidade podem ser horizontal, vertical ou migração. A Figura 4 exemplifica estes métodos e a seguir cada um é detalhado:

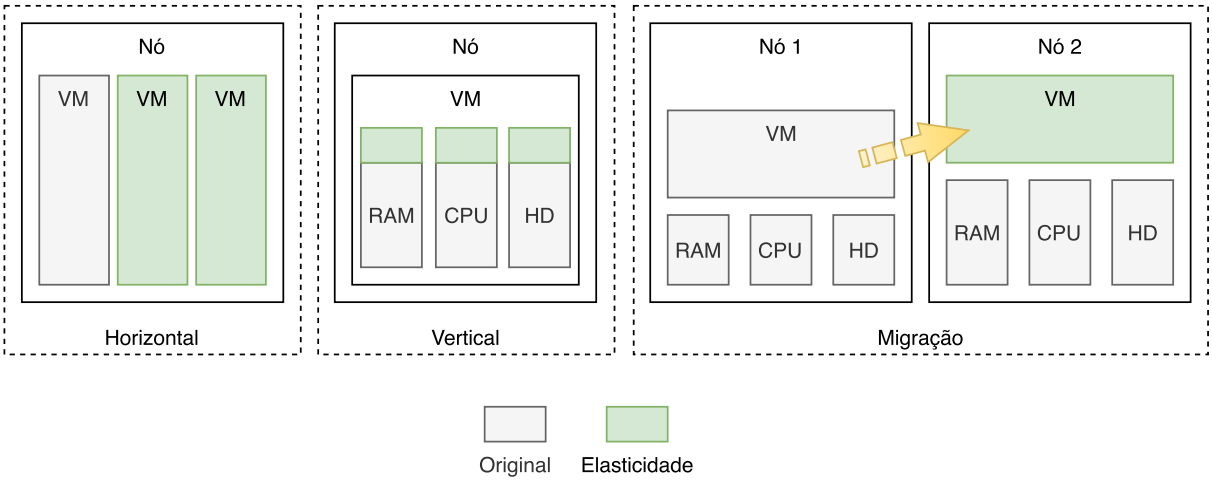
Horizontal: consiste na adição ou remoção de instâncias do ambiente virtual do usuário. Estas instâncias podem ser máquinas virtuais, contêineres ou módulos de aplicação que são replicados e disponibilizados pelo serviço de nuvem. Atualmente, a replicação é o método mais utilizado para prover elasticidade horizontal (GALANTE; BONA, 2012; COUTINHO et al., 2015).

Vertical: consiste na adição ou remoção de recursos de processamento, memória e armazenamento em uma instância de máquina virtual que esteja em execução. Desta forma, as máquinas virtuais tem seus atributos alterados e seus recursos físicos são aumentados ou diminuídos (GALANTE; BONA, 2012; COUTINHO et al., 2015).

Migração: consiste na transferência de uma máquina virtual ou contêiner que está executando em um servidor físico para outro servidor. A elasticidade pode ser implementada pela migração de uma virtualização para uma máquina física com melhores recursos para a execução da aplicação ou, na consolidação de um conjunto de virtualizações em um único servidor (GALANTE; BONA, 2012; COUTINHO et al., 2015).

Os modelos são divididos em reativo e proativo. O modelo reativo reage a carga atual e trabalha com limiares ou com violação de *Service Level Agreement* (SLAs), que determinam a necessidade de aumento de capacidade. No modelo proativo são usadas técnicas de projeção de carga para determinar a carga futura e, por consequência, quando esta carga excederá as capacidades atuais, algumas técnicas comumente usadas são *Fast Fourier Transform* (FFT), *Wavelets*, séries temporais e perfis (COUTINHO et al., 2015; ROSA RIGHI, 2013). A aplicação de um modelo proativo em relação ao reativo, permite a antecipação da reconfiguração de recursos, e assim, por exemplo, reduzir o tempo para concluir a aplicação (ROSA RIGHI, 2013)

Figura 4: Métodos de elasticidade. Horizontal: é a replicação de máquinas virtuais em um mesmo nó. Vertical: é o aumento de recursos de *hardware* de uma máquina virtual. Migração é a transferência de uma maquina virtual de um nó físico para outro que possua mais ou menos recursos.



Fonte: Elaborado pelo autor

2.2.2 Balanceamento de Carga

Em sistemas distribuídos, balanceamento de carga é um mecanismo que habilita a movimentação de serviços de um computador para outro, para deixar o serviço mais rápido ou minimizar o tempo de resposta (ALAKEEL et al., 2010; Citrix Systems, Inc., 2020; NGINX, 2020).

O balanceador de carga, no intuito de estar mais adequado a demanda do usuário, pode estar localizado nas camadas 4 ou 7 do modelo *Open System Interconnection* (OSI), sendo: (*i*) na camada 4 ou camada de transporte, onde o balanceador atua como um tradutor de endereço de rede sem inspecionar o conteúdo dos pacotes; (*ii*) na camada 7 ou camada de aplicação, onde o balanceador, diferentemente da camada 4, pode tomar as decisões de balanceamento com base nos conteúdos dos pacotes, incluindo certificados de segurança, sessões do usuário e quaisquer outros itens dado que está na camada mais alta do modelo OSI (Citrix Systems, Inc., 2020).

Para a realização do balanceamento de carga, são empregados algoritmos que variam dos mais simples aos mais complexos. Os algoritmos comumente empregados em sistemas distribuídos são: *Round Robin*, Menor quantidade de conexões, Menor tempo de resposta; Randômico, Menor consumo de rede e uma combinação de algoritmos (Citrix Systems, Inc., 2020; NGINX, 2020).

2.2.3 Estratégias de predição de carga futura

Em sistemas compartilhados, as aplicações estão em concorrência com cargas de trabalho produzidas por outras aplicações desconhecidas. As aplicações podem utilizar predições para adaptar seus comportamentos em resposta as mudanças do sistema para obter um melhor desem-

penho, estas estratégias são dividas em duas categorias homeostática e baseada em tendência (YANG; FOSTER; SCHOPF, 2003). A seguir, essas estratégias são descritas.

Homeostática: a estratégia de predição homeostática trabalha na premissa de que se o valor atual é maior (menor) do que a média dos valores históricos, o próximo valor, provavelmente, irá diminuir (aumentar).

Baseada em tendência: esta estratégia de predição baseada em tendência, i.e., trabalha na premissa de que o próximo valor segue a tendência de mudança de séries temporais. Esta abordagem assume que se o valor atual aumentou, o próximo valor também irá aumentar e se o valor atual diminuiu, o próximo valor também irá diminuir.

2.2.4 Séries Temporais

Segundo Ehlers (2007) uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo. O uso de séries temporais tem como principais objetivos: a descrição das propriedades da série, como por exemplo, o padrão de tendência; predizer valores futuros com base nos valores do passado e; controle de processos (EHLERS, 2007; BROCKWELL; DAVIS, 2016).

As séries temporais podem separadas em duas terminologias discretas e contínuas. As discretas tem intervalos discretos, ou seja, inteiros. Por exemplo, as vendas realizadas por uma empresa em um ano específico. Enquanto, as contínuas possuem intervalos de tempo contínuos. Por exemplo, o registro das chuvas a cada dia, uma série temporal contínua pode ser transformadas em discreta, avaliando um espaço de tempo determinado (EHLERS, 2007).

As séries temporais podem ser modeladas de várias formas de acordo com as suas características. Estes modelos são chamados de processos estocásticos (EHLERS, 2007). Os processos estocásticos podem ser subdivididos em classes, uma classes importante são os processos estacionários, onde as características do comportamento não variam no tempo, ou seja, é indiferente para esta classe escolher a origem dos tempos (EHLERS, 2007). Dentro dos processos estocásticos tem-se o método de médias. Este método suaviza a série removendo os valores discrepantes. Para a realização desta suavização, cada medição recebe um peso de acordo com o tempo em que foi realizada (EHLERS, 2007).

Existe também o processo Auto-Regressivo (AR), este processo é usado exclusivamente para predição, para o tal são realizados cálculos de coeficiente de auto-correlação e resolução de equações lineares para determinar os fatores de ponderação (EHLERS, 2007). Uma variação deste processo é o processo AR de Médias Móveis (ARMA), que é a junção do método de médias e do processo AR (EHLERS, 2007).

Finalmente, tem-se o processo AR integrado de Médias Móveis (ARIMA), diferentemente dos anteriormente citados, este processo se enquadra na classe de processos não estacionários, ou seja, existe a variação do comportamento da série de acordo com o tempo. Por este motivo,

é o mais indicados para a realização de predição, quando o comportamento não é bem definido (EHLERS, 2007; BROCKWELL; DAVIS, 2016).

2.3 Considerações Parciais

Este capítulo abordou os principais conceitos envolvidos no tema da presente proposta. Foram apresentados os conceitos básicos sobre telecomunicação móvel e funções de rede virtualizadas. Esses conceitos são necessários para um maior entendimento do contexto do modelo proposto. Após isso, foram detalhados pontos chaves que permitem a dinamicidade e balizam a tomada de decisão do modelo, sendo a elasticidade de recursos e a predição de carga.

3 TRABALHOS RELACIONADOS

Neste capítulo, são apresentados trabalhos relacionados ao tema, afim de apresentar a lacuna no estado da arte, referente a elasticidade e ao balanceamento de carga para os serviços do núcleo 5G. Na Seção 3.1, é apresentada a metodologia para a seleção de artigos para o embasamento do trabalho. Na Seção 3.2, são apresentados os trabalhos relacionados no intuito de evidenciar o estado da arte e as pesquisas realizadas na área. Por fim, na Seção 3.3, é apresentado um quadro comparativo entre os modelos apresentados, a fim de ressaltar os principais pontos de interesse para o desenvolvimento do presente trabalho.

3.1 Metodologia de seleção dos artigos

Para seleção dos artigos foram realizadas pesquisas nas bases IEEE, SCOPUS, ACM e Google Acadêmico. O primeiro passo foi buscar artigos que possuíssem menção a adaptabilidade, elasticidade ou balanceamento de carga no núcleo 5G, chegando a 4150 artigos, através da seguinte *string* de busca: (*5G OR (3GPP AND "release 15") OR ("mobile"AND "telecom"AND "next generation")) AND (core OR SBA OR "Service Based Architecture") AND (elasticity OR adaptability OR balancing OR orchestration)*.

Uma vez montada a base inicial, foram removidos os duplicados chegando a 3750 artigos. No intuito de eliminar pesquisas ultrapassadas e que não estejam mais aderentes ao *status quo* da área foram removidas as publicações com mais de 5 anos, chegando a um total de 2390. Posteriormente, foi realizado uma análise nos títulos por palavras chave, mantendo ou excluindo os artigos com base em 2 listas de palavras, chegando a um total de 315 artigos. Sendo que para a inclusão as palavras *elastic, balancing, balancer, dynamic, scalable, scaling, adaptability* e *orchestration*, já para exclusão apenas a palavra *survey*.

Após foram lidos os resumos dos 315 trabalhos mantendo os propusessem um modelo para elasticidade ou escalabilidade ou ainda para qualidade de serviço e de experiência do usuário tanto para o núcleo 4G quanto para o 5G, chegando a 41 artigos. Por fim, foram lidos estes artigos e selecionados os que apresentaram soluções que permitissem a dinamicidade dos serviços do núcleo 4G ou 5G, chegando a 7 artigos, que são apresentados na Seção 3.2.

3.2 Artigos selecionados

Nesta seção, são descritos os 7 trabalhos selecionados, de acordo com os critérios da seção anterior. Os trabalhos de Buyakar et al. (2019), Alawe et al. (2018), Banerjee et al. (2015), Salhab, Rahim e Langar (2019) e Arteaga et al. (2019) trazem uma solução de elasticidade para o núcleo, enquanto os trabalhos de Nguyen et al. (2018) e Amogh et al. (2017) trazem uma solução que permite a replicação de serviços do núcleo, mas sem a aplicação de elasticidade. Destes, apenas os trabalhos de Buyakar et al. (2019) e Alawe et al. (2018) são aplicados no

núcleo, definido na *Release* 15. A seguir, cada um destes trabalhos é apresentado, bem como seus resultados e forma de avaliação.

Buyakar et al. (2019) propõem um protótipo que implementa o SBA do núcleo 5G e o disponibiliza em um ambiente NFV. Este protótipo possui o serviço NRF, especificado na *Release* 15 do 3GPP, que efetua o registro e descoberta de serviços de rede. Além disto, os autores propõem um balanceador de carga no modelo *look-aside*, onde a rede de acesso de rádio realiza uma primeira requisição para o balanceador de carga que retorna qual será a AMF a ser utilizado. Posteriormente, a rede de acesso de rádio estabelece uma comunicação direta com este AMF. Para comunicação entre os serviços do núcleo é utilizado o *Google Remote Procedure Call* (gRPC) ao invés do HTTP REST especificado pelo 3GPP. A justificativa é o uso de CPU e o tempo de resposta que são consideravelmente menores. Para validação do protótipo foi definida uma métrica de latência do plano de controle, que, neste caso, é a soma dos tempos de *attach* e *detach* dos UEs em milissegundo. Foram realizados cenários de testes em um comparativo com 1, 2 e 3 AMFs e com 3 técnicas distintas de balanceamento de carga (*Round Robin*, randômica e com menor uso de CPU). A conclusão é de que a latência do plano de controle diminui a medida que é incrementado o número de AMFs. Além disso, a técnica de balanceamento com menor latência foi a que utilizou o menor uso de CPU.

Alawe et al. (2018) propõem um modelo analítico para escalar o serviço AMF, no intuito de reduzir o tempo de resposta, para isto é incluído um orquestrador de serviços ao SBA que realiza o escalonamento do AMF. Este modelo parte da premissa que a rede de acesso de rádio terá a capacidade de se comunicar com n AMFs e que todos os AMFs disponíveis serão propagados para a rede de acesso de rádio, através de *Domain Name System* (DNS). As decisões de escalabilidade são tomadas com base nas requisições de UEs e na quantidade destas requisições que cada AMF tem capacidade de atender. Para as ações de elasticidade, o orquestrador compara a carga geral do sistema com um limiar probabilístico fixo, se for superior, um novo AMF é alocado, caso contrário, o orquestrador verifica se o sistema com menos um AMF ficaria abaixo do limiar, se verdadeiro um AMF é desalocado. Para validação foram estabelecidos 4 cenários: Diferentes curvas de chegadas de UEs; Sistema com carga baixa; Sistema totalmente carregado, e; Sistema sobrecarregado. Para cada cenário, foi comparado o modelo proposto com um modelo de média móvel exponencialmente ponderadas (MMEP), que consiste no disparo randômico de requisições do UE para os AMFs ativos. Em todos os cenários, o modelo proposto foi mais satisfatório, tanto em latência quanto na distribuição de carga.

Nguyen et al. (2018) estabelecem um modelo de arquitetura que separa o plano de dados do plano de controle e adiciona balanceamento de carga de MMEs virtualizados, aplicando conceitos de VNF no núcleo EPC. Foram propostos 3 algoritmos de balanceamento de carga: uniformemente randomizado; *Round Robin* e; *Round Robin* com pesos. Para validar os algoritmos de balanceamento de carga, foram definidos 2 cenários, ambos com 4 MMEs virtuais, sendo o primeiro onde os MMEs possuem recursos de hardware iguais, mas com diferentes latências na comunicação com o balanceador de carga. O segundo com recursos diferentes e

latências iguais. A comparação foi feita com base na utilização de CPU e no tempo para a realização do *attach* dos UEs. Os testes realizados evidenciaram que a utilização do algoritmo *Round Robin* com pesos teve melhor desempenho que os demais.

Amogh et al. (2017) apresentam um modelo de segregação das funções do MME em 3 micro-serviços por tipo de requisição, sendo *attach*, *detach* e atualização de localização, utilizando técnicas de VNF e criando um *pool* para cada um destes serviços. Para manter a transparência para a rede de acesso de rádio, um balanceador de carga é colocado como *front-end* destes serviços que, além de balancear a carga para cada *pool*, realiza a interpretação do tipo de requisição para direcionar ao serviço correto. Para validação, foi realizada a comparação entre o modelo e uma versão do núcleo com balanceador de carga, mas como MME monolítico, sendo realizadas em 2 cenários. No primeiro, foi realizada uma avaliação da sobrecarga do balanceador de carga, já o segundo trata-se de uma avaliação da escalabilidade do MME. Em termos de capacidade de entrega, o balanceador de carga foi menos eficiente, devido a interpretação do pacote antes de direcionar ao serviço correto. Contudo, a segregação do MME em micro-serviços trouxe maior capacidade de entrega e menor uso de CPU, esse ganho em capacidade de entrega foi tão superior que sobrepôs a perda no balanceamento de carga.

Banerjee et al. (2015) apresentam um modelo chamado SCALE que provê elasticidade horizontal proativa para o MME do núcleo EPC, para o tal, foi adicionado um balanceador de carga em frente a um *pool* de MMEs virtualizados. Desta forma, a rede de acesso de rádio comunica com o balanceador de carga que decide qual MME virtual deverá processar e responder cada requisição. Este modelo realiza as ações de elasticidade em intervalos de tempo bem definidos (épocas), estas ações são definidas por projeções (processamento e memória) baseadas na época anterior. O custo de processamento é projetado utilizando médias móveis, tendo como entrada medições do processamento da época anterior, já para o custo de memória são projetados, também baseado na época anterior, a quantidade de UEs registrados e o número de sinalizações que serão efetuadas. Para os 2 custos é calculado o número de VMs necessários para suprir a necessidade, sendo utilizado para o provisionamento o número maior. Para validar este modelo foi construído um protótipo tendo como base a plataforma OpenEPC, para gerar o tráfego de UEs foi construída uma emulação do eNB. Como métricas, foram utilizadas (*i*) *delay* da comunicação fim-a-fim na percepção do UE, (*ii*) a quantidade de VMs necessárias no processamento das requisições para atingir a meta de latência do plano de controle e (*iii*) a média de carga de CPU das VMs. Nos testes, esse modelo foi comparado com uma versão do SCALE reativa e com o sistema comercial SIMPLE. Nas duas comparações e para todas as métricas, o modelo proativo foi mais eficiente.

Salhab, Rahim e Langar (2019) definem uma arquitetura de elasticidade horizontal reativa para o núcleo EPC no intuito de prover alta disponibilidade. A elasticidade é aplicada no núcleo como um todo, a comunicação com a rede de acesso de rádio é realizada através de um *proxy*, que realiza um balanceamento de carga para distribuir as requisições entre os *cores* existentes. As ações de elasticidade são tomadas com base na carga de CPU, que é comparada com limiares

de controle (superior e inferior), caso a carga seja superior, um novo núcleo é provisionado, caso a carga seja inferior, um núcleo é removido.

Finalmente, Arteaga et al. (2019) definem um modelo de elasticidade horizontal e vertical reativa, que realiza a elasticidade do MME, SGW e PGW. Para cada um dos serviços, foi adicionado um balanceador de carga, para que haja transparência para seus clientes. Nesse modelo, foram definidas três áreas subsequentes de desempenho com base na quantidade de usuários concorrentes, ou seja, a medida que a quantidade de usuários aumenta a estratégia se modifica. Na primeira área, não é realizada nenhuma ação de elasticidade, na segunda, a elasticidade é vertical e na terceira, a elasticidade é horizontal. Para que a elasticidade ocorra, além da verificação da área, é monitorado o número de registros por segundo. Para este indicador, foram definidos dois limiares. Assim somente ao ultrapassar o primeiro limiar, a elasticidade vertical é ativada e, somente ao ultrapassar o segundo limiar, a elasticidade horizontal é ativada. Ou seja, para que cada um dos métodos de elasticidade seja ativado, se faz necessária a combinação da quantidade de usuários concorrentes e o número de registros por segundo. Uma vez que tenha sido identificado um modelo específico de elasticidade, essa arquitetura realizada uma única ação de elasticidade com base nas configurações do administrador.

Para avaliar o modelo proposto por Arteaga et al. (2019), foram realizados diversos testes com diversas combinações de configurações de quantidade de recursos, para chegar a configuração com melhor desempenho em latência e uso de CPU para cada método de elasticidade. Para determinar os limiares e regiões, foram realizados testes com as configurações anteriormente selecionadas, verificando a capacidade máxima de usuários concorrentes com até 90% de uso de CPU. Por fim, foi realizado um teste com as configurações selecionadas comparando o modelo proposto com um sistema sem elasticidade, quando habilitada a elasticidade vertical houve uma redução de 70% em latência e cerca de 180% de aumento na quantidade de registros por segundo, já quando habilitada a elasticidade horizontal houve um aumento de 308% na quantidade de registros por segundo, porém houve um pequeno aumento na latência.

3.3 Comparativo e oportunidade de pesquisa

Conforme pode ser evidenciado na Tabela 1, existem trabalhos que exploram a utilização de replicação de serviços para permitir o paralelismo distribuído de ações do núcleo da telecomunicação móvel. Pensando no contexto atual, onde um volume bilionário de investimento já foi realizado na estruturação das redes de acesso, tendo milhares de pontos espalhados pelo mundo, um modelo que possua modificações na rede de acesso de rádio se torna complexo e financeiramente caro de ser aplicado. Desta forma, é imprescindível que a comunicação do núcleo com a rede de acesso seja totalmente transparente. Esta transparência também é válida, quando se fala da nova geração de redes de acesso, pois neste caso todos os softwares já desenvolvidos, bem como os que ainda serão, precisariam se adequar ao modelo, o que não é positivo, em termos de compatibilidade.

Tabela 1: Trabalhos Relacionados

Trabalho	Tecnologia de Virtualização	Transparente para a RAN	Versão núcleo	Implementa comunicação Fim-a-Fim	Modelo/Método de Elasticidade	Tipo da Proatividade
Buyakar et al. (2019)	Contêiner	Não	5G - SBA	Não	horizontal/reativa	-
Alawe et al. (2018)	-	Não	5G - SBA	Não	horizontal/ reativa	-
Nguyen et al. (2018)	Hypervisor	Sim	4G - EPC	Sim	sem elasticidade	-
Amogh et al. (2017)	Hypervisor	Sim	4G - EPC	Sim	sem elasticidade	-
Banerjee et al. (2015)	Hypervisor	Sim	4G - EPC	Sim	horizontal/proativa	baseada em épocas
Salhab, Rahim e Langar (2019)	Contêiner	Sim	4G - EPC	Sim	horizontal/reativa	-
Arteaga et al. (2019)	Hypervisor	Sim	4G - EPC	Sim	horizontal e vertical/reativa	-

Outro ponto extremamente importante é elasticidade dos serviços do núcleo, de tal forma que a utilização de recursos computacionais possa ser reduzida. Dentre os trabalhos em que houveram a aplicação de elasticidade, os trabalhos de Buyakar et al. (2019), Alawe et al. (2018), Salhab, Rahim e Langar (2019) e Arteaga et al. (2019) fizeram a utilização do modelo de elasticidade reativa que só realiza uma ação de elasticidade, quanto algum indicador é atingido. Entretanto, este modelo pode gerar ineficiência na utilização dos recursos, devido a poder realizar uma alocação ou desalocação tardia. Desta forma, entende-se que o modelo que introduz maior benefício é o proativo, que busca antever o comportamento da carga de trabalho e permite que a reconfiguração dos recursos possa ser realizada de forma antecipada e assim reduzindo a utilização de recursos.

O único trabalho que utiliza proatividade foi proposto por Banerjee et al. (2015). Entretanto, este modelo, além de utilizar o núcleo EPC, é baseado em épocas de tamanho fixo e a ação de elasticidade ocorre somente no início de cada época. Desta forma, se o comportamento se modificar no meio de uma época, o sistema apenas será redefinido na próxima época, podendo gerar ineficiência na utilização de recursos e na capacidade de atendimento. Uma forma de contingenciar este comportamento seria ter épocas de tamanho pequeno, porém, neste modelo o cálculo de predição só considera valores obtidos na última época, logo esta estratégia pode trazer perda de precisão na predição.

Finalmente, entende-se por oportunidade de pesquisa, um modelo de elasticidade horizontal proativo para o núcleo 5G previsto no *Release* 15. Esse modelo deve ser totalmente transparente para a rede de acesso e que, no intuito de melhorar a utilização de recursos, efetue as ações de elasticidade o mais próximo possível das necessidades do sistema.

3.4 Considerações parciais

Neste capítulo foi possível observar o estado da arte, onde vimos trabalhos que apresentam elasticidade reativa e proativa para os *cores* 4G e 5G. Além disso, foi possível identificar a oportunidade de pesquisa que foi o cerne para a definição do modelo Elastic5GC, o qual são apresentado no próximo capítulo.

4 O MODELO ELASTIC5GC

Este capítulo descreve o modelo Elastic5GC que provê elasticidade proativa para o core 5G. A Seção 4.1 apresenta as decisões de projeto para a realização deste modelo. A Seção 4.2 apresenta a arquitetura do modelo Elastic5GC.

4.1 Decisões de projeto

O modelo Elastic5GC tem por objetivo melhorar a utilização de recursos computacionais, de tal forma, que os recursos sejam alocados quando necessários e desalocados quando não mais necessários. Visando ter o menor tempo de provisionamento possível, foram utilizados contêineres para os serviços que são provisionados em tempo de execução. Além disto, foi adotada a elasticidade horizontal, devido ao fato de que, com elasticidade vertical, a quantidade de recursos fica limitada sempre aos recursos disponíveis em um único nó, enquanto na horizontal pode-se aproveitar os recursos, tanto de um nó quanto dos n nós disponíveis.

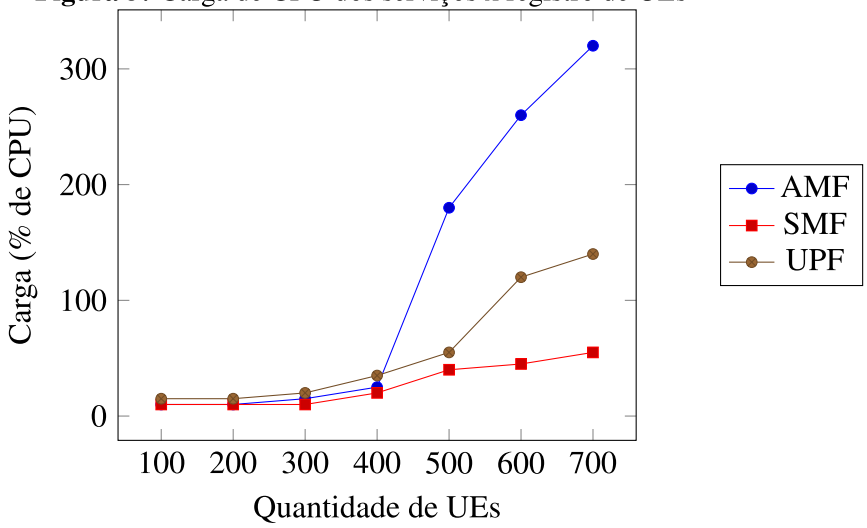
Para prover elasticidade proativa, foi utilizada uma estratégia baseada em tendência para determinar a carga futura, conforme apresentado na Seção 2.2.3. Dessa forma, considera-se um histórico de medições anteriores para a definição da tendência de carga (aumento ou redução). Esta projeção foi realizada utilizando o modelo ARIMA , por ser um modelo que representa uma série temporal baseada em tendência que, diferentemente de outras séries, leva em consideração a variação do comportamento ao longo do tempo. Deste modo, a utilização do modelo ARIMA é uma solução muito aderente ao cenários de predição de carga em sistemas distribuídos.

Optou-se por realizar a elasticidade apenas do serviço AMF, pois este é o serviço responsável pela comunicação com a rede de acesso. No que se refere ao plano de controle, é o serviço que intermedeia as comunicações da RAN com os demais serviços do núcleo e, por consequência, tem uma alta carga de trabalho, conforme apresentado por Buyakar et al. (2019) e ilustrado na Figura 5.

Para prover a utilização de mais de um AMF no núcleo SBA, foi adicionado um balanceador de carga na interface N2, interface de comunicação do plano de controle entre a rede de acesso e o núcleo (3GPP, 2019a), que realiza o balanceamento das comunicações relacionadas a UEs. Este balanceador opera na camada de aplicação (camada 7 do modelo OSI), pois a gNB estabelece uma conexão com o AMF e a mantém ativa, criando uma sessão por onde todos os UEs sinalizam para o núcleo. Logo, a segregação dos pacotes por UE ocorre apenas no protocolo de aplicação NGAP (3GPP, 2018). Desta forma, não seria viável utilizar algum balanceador de carga na camada de transporte (camada 4 do modelo OSI).

Para a realização do provisionamento de NFVs e de recursos computacionais foi utilizado o NFV-MANO *architectural framework* (ETSI, 2014), tanto por estabelecer padrões e técnicas que são extremamente aderentes ao contexto deste modelo, quanto por ser altamente difundido no universo de telecomunicações. Mais detalhes desta arquitetura são apresentados na seção

Figura 5: Carga de CPU dos serviços x registro de UEs



Fonte: Elaborada pelo autor com base em Buyakar et al. (2019)

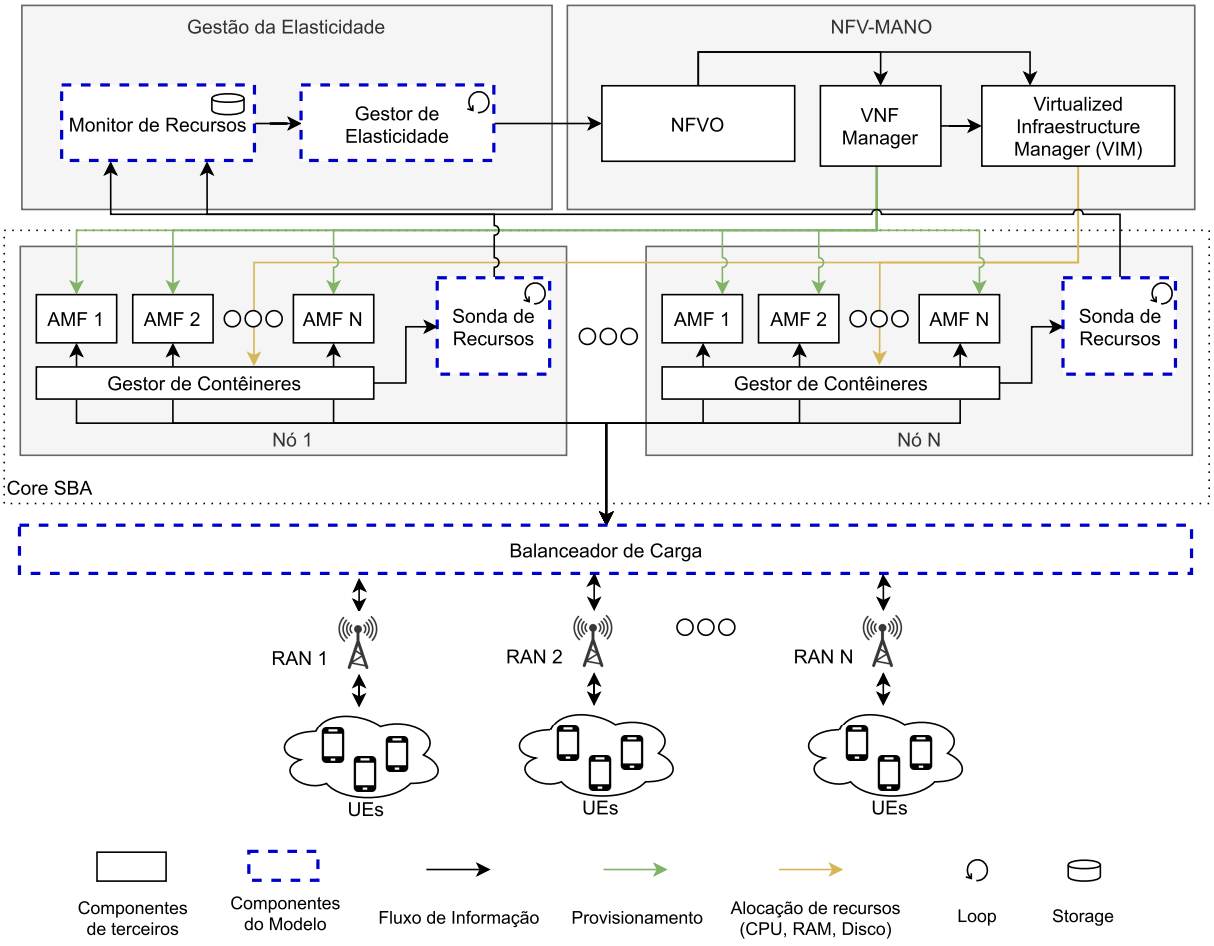
2.1.4. A principal função utilizada oriunda do NFV-MANO foi a alocação de recursos, logo, a definição de em que nó o serviço é alocado ou desalocado é realizada pelos componentes desta arquitetura, com a seguintes premissas: (*i*) a quantidade máxima de serviços AMF a serem alocados em um nó é a quantidade de núcleos de processamento disponíveis fisicamente; (*ii*) na desalocação o AMF a ser removido é sempre o último alocado.

4.2 Arquitetura Elastic5GC

Um dos objetivo do Elastic5GC é a melhora na utilização dos recursos computacionais do núcleo, de tal forma que *n* serviços AMF sejam executados para atender a demanda de sinalização de UEs. No intuito de obter maior maleabilidade no atendimento das demandas, aproveitando de forma eficiente os recursos computacionais e evitando a sobrecarga do sistema, esta arquitetura permite que os serviços possam ser instanciados em *n hosts*.

Nesta arquitetura, conforme apresentado na Figura 6, os principais componentes são o gerenciador de elasticidade, responsável pelas ações e pelas regras de execução de elasticidade e o balanceador de carga, que permite que mais de um AMF seja provisionado em um mesmo núcleo. Além disto, em cada nó existe uma sonda que envia as informações sobre os recursos utilizados por cada AMF instanciado para o monitor de recursos, que por sua vez armazena e centraliza as informações. Uma vez definida a necessidade da ação de elasticidade, o gestor de elasticidade informa para o NFV *Orchestrator* que solicita a alocação ou desalocação de recursos para o VIM e solicita a criação ou remoção de uma instância do AMF para o NFV *Manager*. Os serviços NFV *Orchestrator*, VNF *Manager* e VIM são parte da arquitetura NFV-MANO, detalhes de seu funcionamento podem ser vistos na Seção 2.1.4.

Figura 6: Arquitetura Elastic5GC

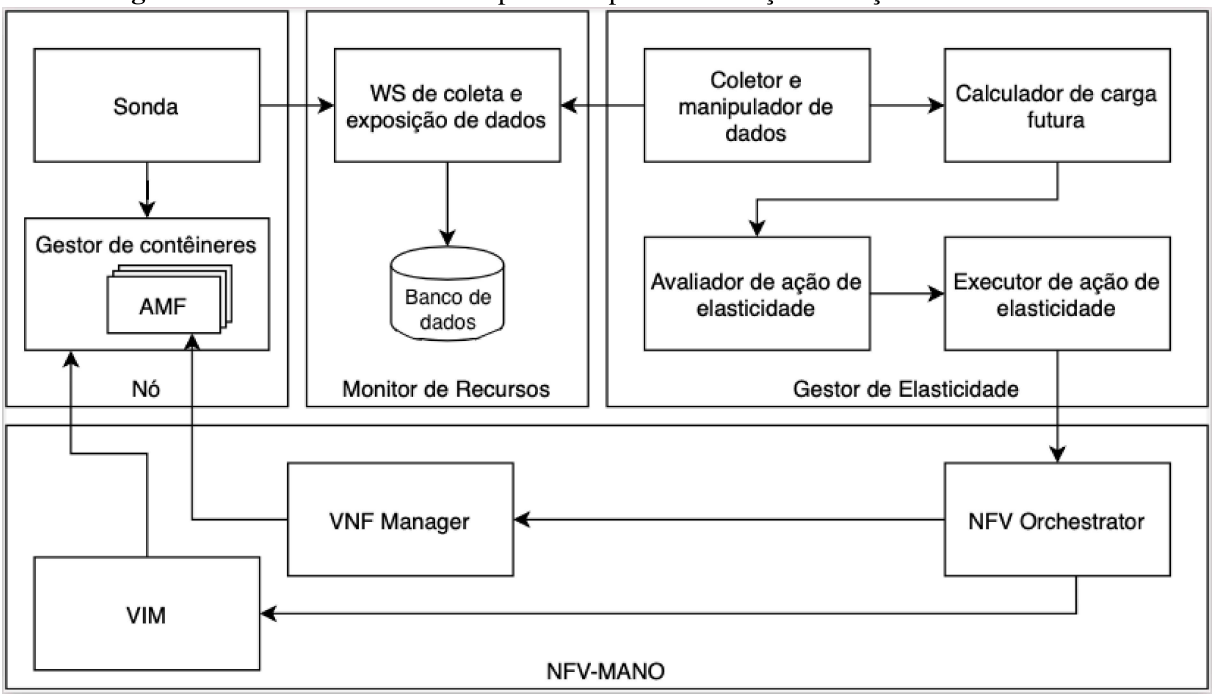


Fonte: Elaborada pelo autor

4.2.1 Gestor de Elasticidade

O gestor de elasticidade é o serviço central do modelo Elastic5GC, pois tem como responsabilidade definir em que momento devem ser realizadas ações de elasticidade. Para tomar esta decisão o gestor busca e sumariza as informações coletadas pelo monitor de recursos, projeta uma carga futura baseada na tendência das medições utilizando séries temporais, mais especificamente o ARIMA, avalia se esta carga futura carece de uma ação de elasticidade e em caso positivo a executa. Para cada uma destas etapas o gestor possui um componente de software (*i*) o coletor e manipulador de dados, (*ii*) o calculador de carga futura, (*iii*) o avaliador de ações de elasticidade (*iv*) o executor de ações de elasticidade. Estes componentes e como ele se conectam com os demais serviços do modelo podem ser visualizados na Figura 7. Além disso, na Figura 8 são apresentados os fluxos de captura de dados e de gestão da elasticidade. A seguir, estes componentes são apresentados, tanto com relação as suas funções, quanto em como se comunicam com os demais serviços.

Figura 7: Detalhamento dos componentes para a realização das ações de elasticidade.



Fonte: Elaborada pelo autor

4.2.1.1 Coletor e manipulador de dados

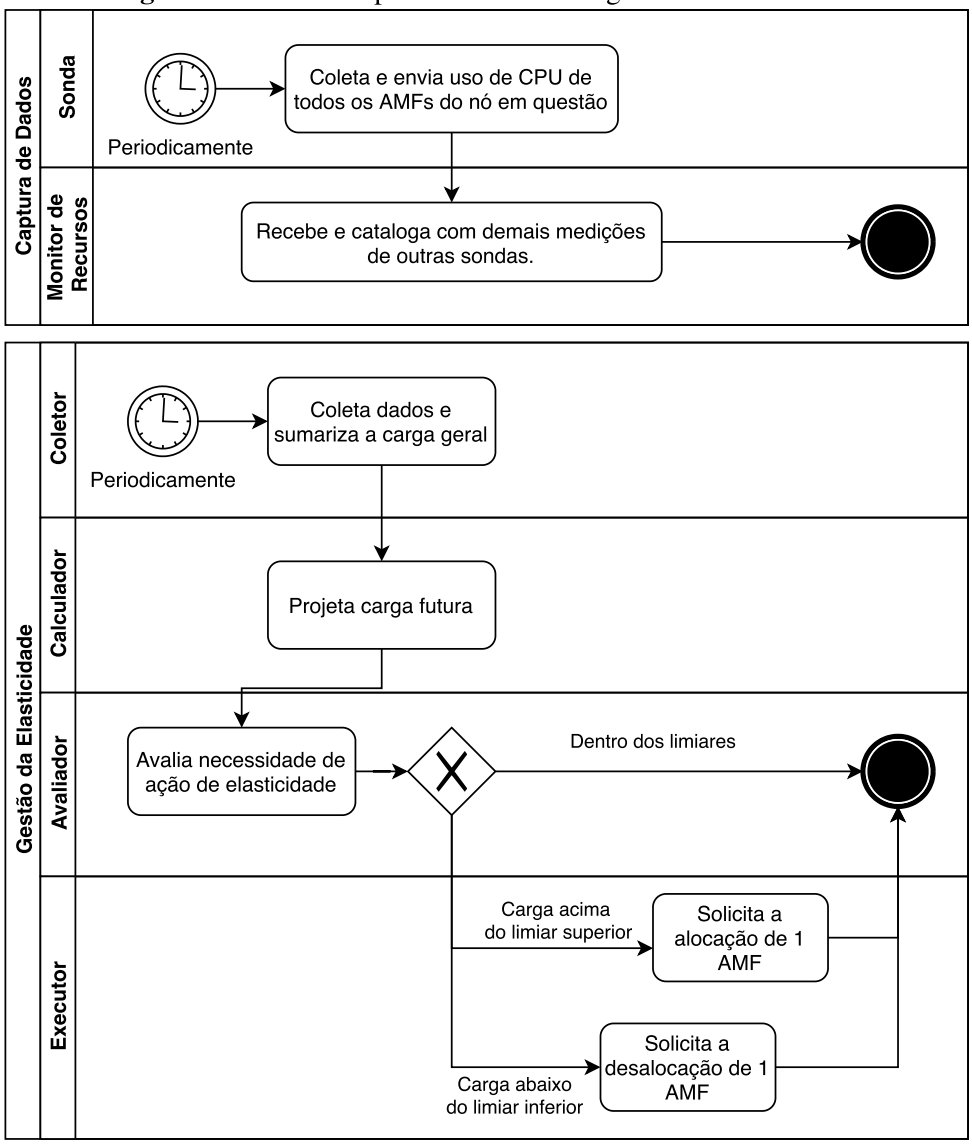
O coletor e manipulador de dados é o componente responsável pela comunicação com o *Web Service* do monitor de recursos para obter as medições de carga de CPU dos AMFs. Partindo da premissa que o balanceador de carga distribui a carga de trabalho, o mais uniforme possível entre os AMFs, o comportamento do sistema pode ser sumarizado em uma carga geral média, que é a média de todas as cargas dos AMFs, conforme Equação 4.1, onde n é o número total de AMFs instanciados no momento da medição, i é uma instância em específico e C_i é a carga de CPU medida para instância i. Sendo assim, para cada medição é calculada a carga geral média, gerando uma lista e enviando para o componente que calcula a carga futura.

$$CG = \frac{\sum_{i=1}^{n} C_i}{n} \tag{4.1}$$

4.2.1.2 Calculador de carga futura

Este componente realiza a projeção da carga futura conforme as medições de carga gerais coletadas. Duas características importantes deste componente são a utilização de séries temporais, para entender a tendência de carga, e o *lookahead*, que é o tempo futuro a ser realizada a projeção de carga. Para a análise de tendência foi usado o modelo ARIMA, pois é um modelo da classe de processos não estacionários, sendo assim, é aderente a comportamentos que variam

Figura 8: Fluxo de captura de dados e de gestão de elasticidade.



Fonte: Elaborada pelo autor

de acordo com o tempo e é altamente indicado para projeção (EHLERS, 2007). A definição do *lookahead* é extremamente importante, pois se for utilizado um tempo pequeno a alocação de recursos pode ser tardia gerando ineficiência. Já, se o tempo for muito grande há um aumento na taxa de erros da predição, podendo inferir em uma ação de elasticidade equivocada. Para determinar o *lookahead* foi utilizado o tempo máximo em que um serviço AMF leva para ser provisionado e estar operacional, dividido pelo tempo de intervalo entre cada medição, conforme a Equação 4.2, onde tp é o tempo médio em que um AMF leva para ser provisionado, ti é o tempo médio que um AMF leva para ser inicializado e tm é o tempo de intervalo entre as medições.

$$lookahead = \frac{(tp + ti)}{tm} \tag{4.2}$$

4.2.1.3 Avaliador da ação de elasticidade

Uma vez calculada a carga futura é preciso interpretar se haverá a necessidade de uma ação de elasticidade e qual ação (alocação ou desalocação) deve ser realizada. Para o tal, foram definidos dois parâmetros de controle que determinam se o sistema está operando em carga baixa ou alta, sendo eles: (*i*) limiar inferior que indica que o sistema está operando em carga baixa e que serviços devem ser desalocados e; (*ii*) limiar superior que indica que o sistema está operando em carga alta e que serviços devem ser alocados. Com base nos artigos de Galante et al. (2016) e Rosa Righi et al. (2019) foram estipulados como limiares superior e inferior os valores de 80% e 20%, respectivamente. Este intervalo permite que a predição seja mais assertiva, visto que, pequenas flutuações nos valores previstos não são levadas em consideração para a tomada de decisão, diminuindo a ocorrência de ações de elasticidade equivocadas.

4.2.1.4 Executor de elasticidade

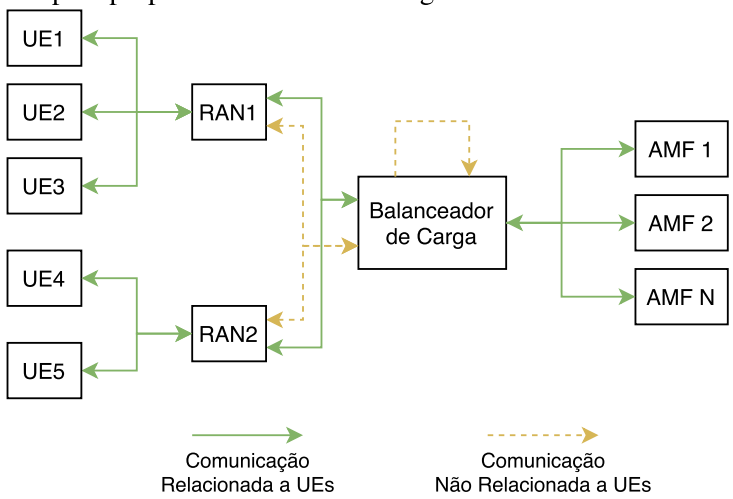
Com a utilização do NVF-MANO para gerenciamento da infraestrutura e disponibilização dos serviços, o executor de elasticidade tem apenas a função de fazer a *interface* com o *Web Service* do NFV *Orchestrator*. Desta forma, uma vez realizado o entendimento da necessidade da execução de uma ação de elasticidade, o executor de elasticidade comunica com o NFV *Orchestrator* informando que tal ação deve ser efetuada e assim alocando ou desalocando um AMF.

4.2.2 Balanceador de Carga

O balanceador de carga é parte importante deste modelo, pois ele é quem viabiliza a utilização de mais de um AMF de forma transparente para a rede de acesso de rádio. O balanceador é aplicado na camada de aplicação, mais especificamente sob protocolo NGAP (vide Seção 2.1.3). Primeiramente, é preciso ter em mente que no protocolo NGAP a comunicação entre a rede de acesso de rádio e AMF se divide em “relacionada a UEs“ e “não relacionadas a UEs“ que são as sinalizações dos UEs e do eNB, respectivamente (3GPP, 2018). Desta forma, o balanceador de carga fará o balanceamento da comunicação "relacionada a UEs", no entanto a comunicação "não relacionada a UEs"será tratada pelo próprio balanceador de carga. A Figura 9 ilustra este comportamento.

Neste balanceador foi utilizado o algoritmo *Round Robin*, por ser um algoritmo de baixo custo de processamento e assim minimizando o impacto no tempo de resposta para os UEs. Neste algoritmo, o balanceador possui uma lista circular de AMFs e quando a primeira requisição chega, esta é repassada para o AMF na primeira posição, a segunda requisição para o segundo e assim sucessivamente, até o último AMF da lista. Neste momento, ao chegar uma nova requisição, a lista circular volta para a primeira posição e o ciclo se repete.

Figura 9: Fluxo e comunicação do balanceador de carga. As comunicações relacionadas a UEs são balanceadas e entregues aos AMFs virtualizados. As requisições não relacionadas a UEs não são balanceadas sendo resolvidas pelo próprio balanceador de carga.



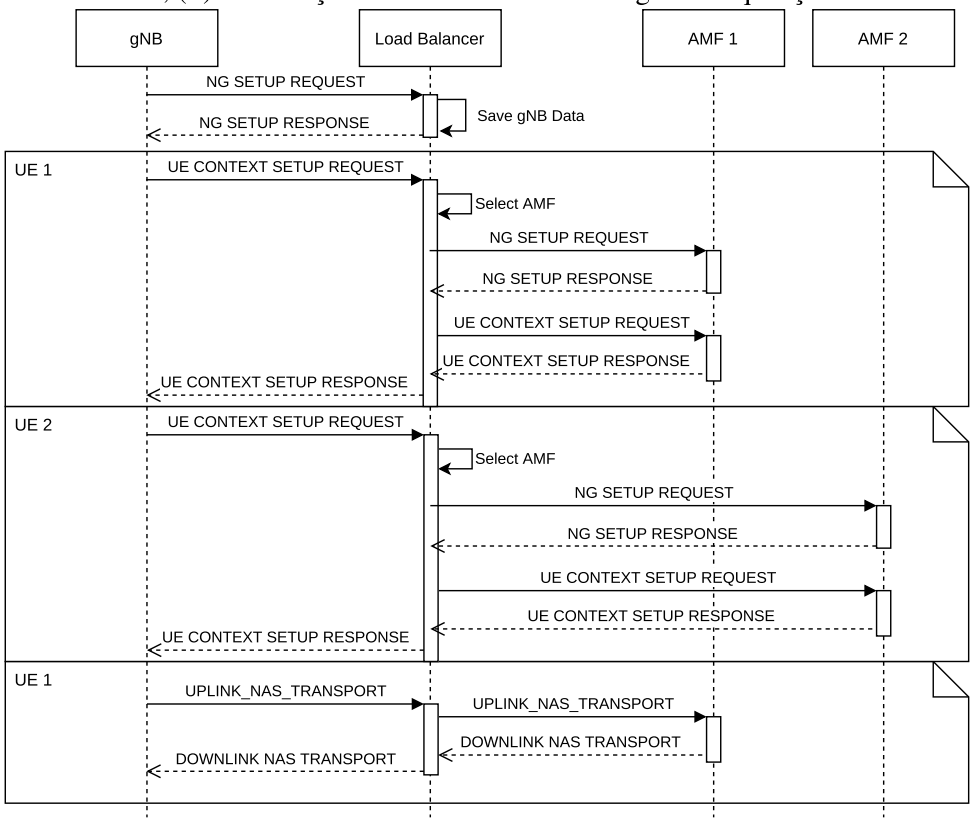
Fonte: Elaborada pelo autor

A Figura 10 apresenta o comportamento do balanceador de carga frente as demandas dos gNBs e UEs. Podemos observar, no topo da figura, o registro de um gNB através da operação NG_SETUP_REQUEST, para este tipo de requisição (não relacionada a UEs) o balanceador de carga faz o tratamento e monta a resposta sem a necessidade de encaminhar para um AMF. Em seguida, na chegada do "UE 1", o balanceador de carga realiza a seleção do AMF, neste caso o "AMF 1", uma vez selecionado e se for a primeira conexão para o gNB em questão, o balanceador de carga envia o registro do gNB que origina a chamada, através da operação NG_SETUP_REQUEST, em seguida envia a requisição do UE, neste caso a operação UE_CONTEXT_SETUP. Na chegada do "UE 2", o processo se repete em sua totalidade, pois se trata da primeira comunicação oriunda do gNB para o "AMF 2". Por fim, na chegada do "UE 3"o AMF selecionado ("AMF 1") já possui o registro do gNB, logo a não há a necessidade de registro do gNB e assim é enviada apenas a operação do UE.

Outro ponto importante do balanceador de carga é forma de conhecimento de novos AMFs ou ainda que AMFs nâo estão mais disponíveis. Para o tal, o balanceador faz o registro de uma função de retorno (*callback*) no repositório de funções de rede (NRF) para que todos os eventos de registro ou remoção de registro de AMFs sejam notificadas para o balanceador de carga. Contudo, a remoção do registro do AMF só ocorre após a conclusão de todos os registros, logo faz-se necessária a comunicação entre o VNF Manager e o balanceador de carga, para que requisições de chegada referente a novos UEs não sejam mais encaminhadas ao AMF que está em processo de desligamento.

A Figura 11 apresenta o comportamento do balanceador de carga de acordo com os registros de remoção de registros dos AMFs. Ao inicializar o balanceador de carga envia um registro de uma função de retorno (*callback*) ao NRF, para que quaisquer mudanças nos AMFs o mesmo seja notificado. Na alocação de um AMF, executado pelo VNF Manager, este AMF realiza

Figura 10: Diagrama de sequência apresentado o comportamento do Balanceador de carga com a chegada de requisições. Cabem salientar 2 pontos: (*i*) o registro da gNB que o balanceador realiza sem comunicar com o core e; (*ii*) não seleção de AMF na cheda da segunda requisição do UE1.



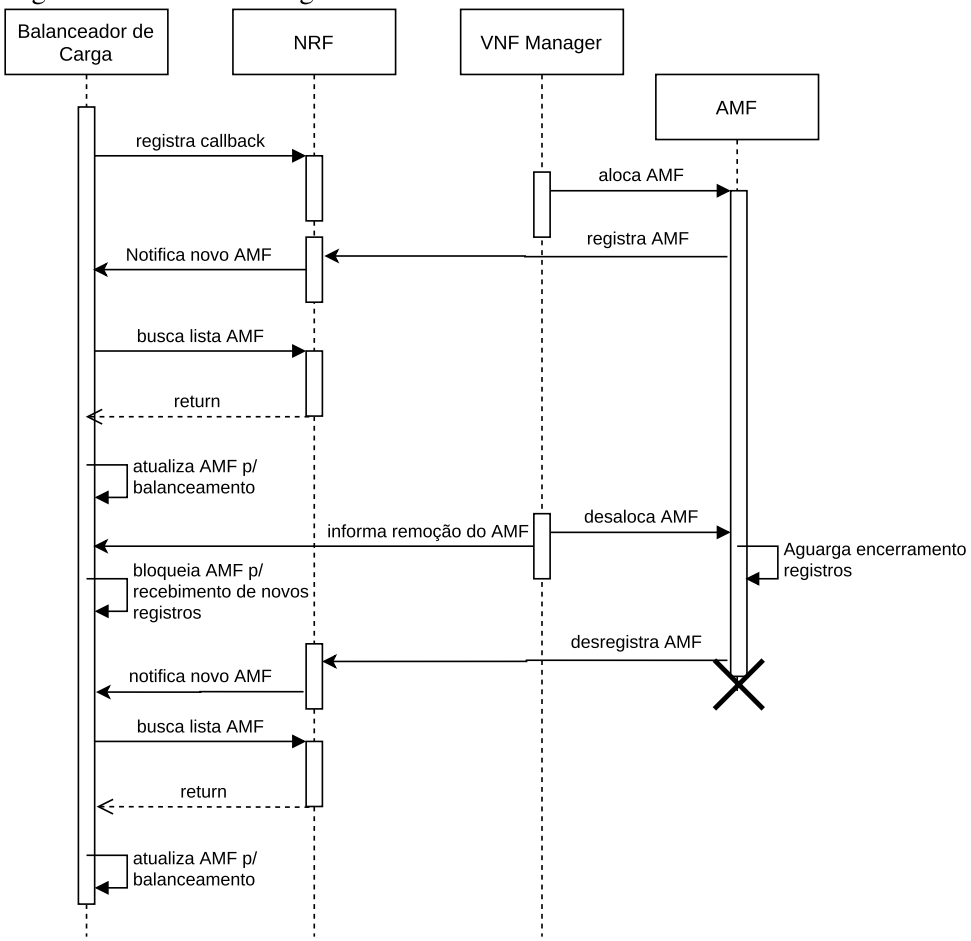
Fonte: Elaborado pelo autor

o auto registro junto ao NRF, que automaticamente envia um sinal ao balanceador de carga indicando que houve alguma alteração, o balanceador de carga, por sua vez, busca, junto ao NRF, a lista atualizada de AMFs, assim passando a enviar requisições a este novo AMF. Uma vez que alguma regra de negócio defina que há a necessidade de remoção de um AMF o VNF Manager realiza a desalocação do AMF e envia um sinal para que o balanceador de carga pare de enviar novas requisições para este AMF, assim, evitando que mensagens sejam enviadas para um AMF em processo de desligamento.

4.3 Considerações parciais

Neste capítulo foi apresentado de forma detalhada o modelo Elastic5GC, com todas as decisões de projeto, arquitetura e os detalhes de cada componente que o compõe. No capítulo seguinte é apresentado o método de avaliação do modelo e a estruturação do protótipo.

Figura 11: Diagrama de sequência apresentado o comportamento do Balanceador de carga quando AMFs são registrados ou tem seu registro removido.



Fonte: Elaborado pelo autor

5 METODOLOGIA DE AVALIAÇÃO

Este capítulo apresenta as métricas de avaliação que podem ser visualizadas na Seção 5.1. Além disso, a Seção 5.2 apresenta a metodologia de desenvolvimento e avaliação do modelo Elastic5GC.

5.1 Métricas de Avaliação

A avaliação deste modelo ocorreu de forma quantitativa utilizando métricas para a verificação de capacidade de atendimento e de esforço computacional. A seguir são descritas as métricas que foram utilizadas. Além disso, é descrita uma discussão sobre o conceito de cada métrica para auxiliar em sua compreensão.

5.1.1 Capacidade de atendimento

Uma das expectativas de melhoria deste modelo é a capacidade de atendimento do *core*. A capacidade de atendimento pode ser medida pela quantidade de requisições de UEs que podem ser atendidas, que pode ser calculada através do somatório dos tempos das requisições dividido pelo total de requisições realizadas, conforme pode ser observado na Equação 5.1, onde n é o número total de operações realizadas, i é uma requisição específica e t_i é o tempo que o *core* gasta para responder esta requisição.

$$TMR = \frac{\sum_i^n t_i}{n} \tag{5.1}$$

5.1.2 Esforço computacional

Outra faceta importante do modelo é a melhora de uso dos recursos computacionais, ou seja, quando necessário os recursos computacionais são utilizados em sua plenitude, mas quando não necessários eles são liberados. Para avaliar esta característica foi avaliado o esforço computacional utilizado pelo sistema, a premissa para esta métrica é que os recursos computacionais são alocados especificamente para os serviços, ou seja, não são compartilhados. Desta forma, mesmo que não haja nenhuma execução em um serviço, o tempo em que esteve ativo contabiliza, pois os recursos não poderiam ser utilizado por outro serviço. Sendo assim, esta métrica é o somatório do tempo em que cada serviço esteve ativo, conforme a Equação 5.2 onde, i é uma instância de um serviço e t_i é o tempo em que esta instância esteve ativa.

$$EC = \sum_i^n t_i \tag{5.2}$$

5.2 Metodologia de Desenvolvimento e Avaliação

Este modelo tem como principais componentes o gestor de elasticidade e o balanceador de carga. Estes dois componentes são altamente responsáveis pela adaptabilidade do modelo. O protótipo é apresentado na Subseção 5.2.1 e os testes são detalhados na Subseção 5.2.2.

5.2.1 Protótipo do Elastic5GC

Como o intuito deste trabalho é a proposição de um modelo de elasticidade e não a criação do *core* 5G, foram realizadas buscas nos repositórios de código fonte aberto Github[1], GitLab[2] e Bitbucket[3], por ferramentas que implementassem o *core* 5G SBA que permitisse uma comunicação ponta-a-ponta e que possuísse código fonte aberto. A única ferramenta encontrada (até a finalização deste trabalho) foi o free5GC (FREE5GC.ORG, 2019).

O free5GC é um projeto de código aberto para o *core* da quinta geração (5G) de telecomunicação móvel (FREE5GC.ORG, 2019). Este projeto possui três *stages*, sendo o primeiro deles implementa protocolo de comunicação 4G com UE 4G e *Base Station* (eNB) 4G. O *stage* 2 é um modelo simplificado que permite testes em um *core* 5G SBA, porém sem algumas funcionalidades como, por exemplo, a orquestração através do repositório de funções de rede (NRF). Por fim, o *stage* 3 implementa um *core* 5G com a maioria das funcionalidades definidas pela *Release* 15 do 3GPP.

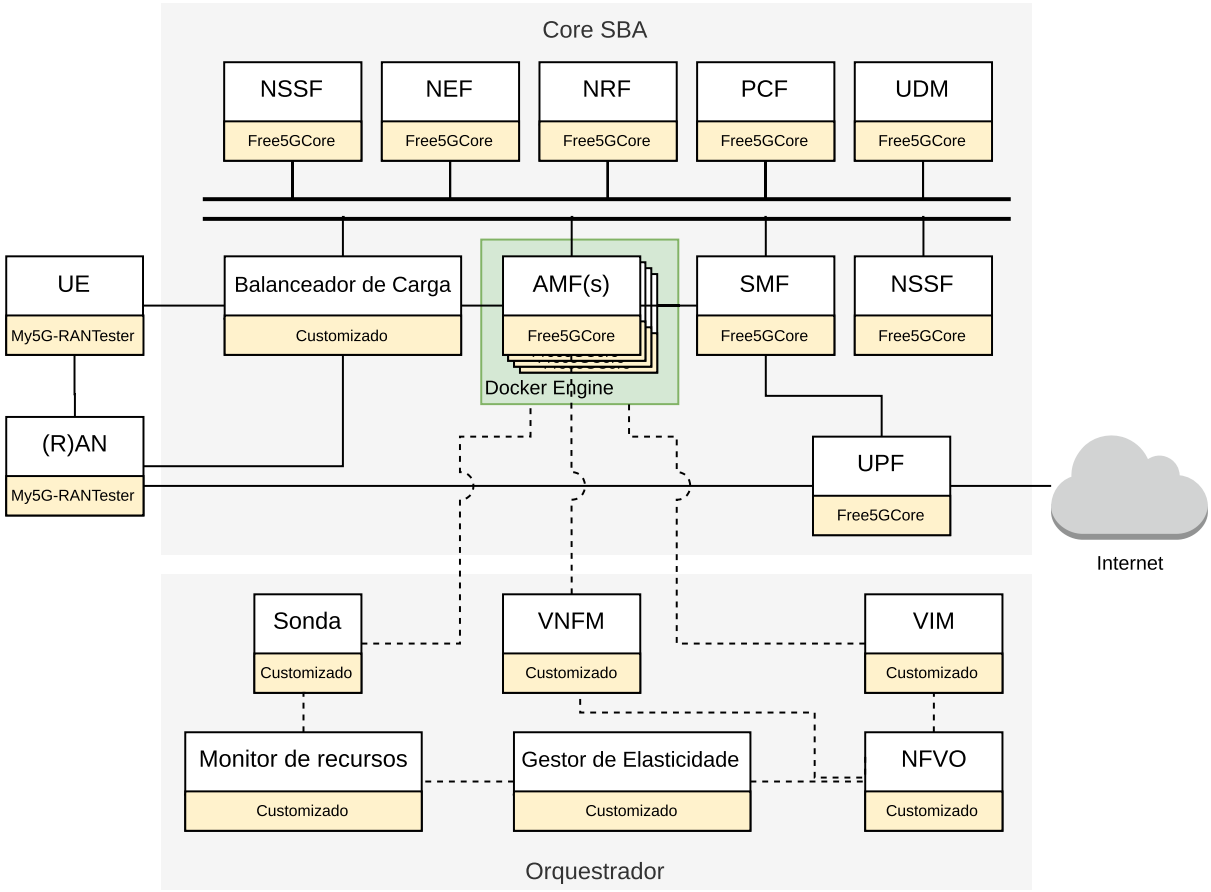
O protótipo utilizado para a validação deste modelo foi feito a partir do *core* do free5GC na *stage* 3. Para que fosse possível implementar o modelo de elasticidade Elastic5GC foi adicionado um balanceado de carga que centraliza o acesso aos multiplos AMFs disponíveis. Este balanceador de carga é responsável, tanto pelo balanceamento dos AMFs, quanto pelas comunicações não relacionadas aos UEs. A Figura 12 apresenta a implementação deste protótipo.

Primeiramente, foi criada a estrutura inicial com a conteinerização do free5GC[4], bem como a criação de um ambiente de testes utilizando o projeto my5G-RANTester[5], seguido das configurações para conexão entre os dois projetos. Com a estrutura inicial configurada, foi realizada a criação do balanceador de carga[6], utilizando a linguagem Go[7], que além de ser naturalmente paralelizável, permitiu que as bibliotecas usadas pelo projeto free5GC fossem aproveitadas.

O balanceador de carga implementa o algoritmo *Rouding Robin*, pela sua simplicidade e pelo vasto uso em projetos. O Algoritmo 1 demonstra o funcionamento geral do balanceador de carga, onde, na inicialização há o registro do *callback* no NRF, seguido da busca da lista de AMFs disponíveis e a inicialização do servidor SCTP/NGAP. A cada recebimento de requisição

[1] https://github.com
[2] https://gitlab.com
[3] https://bitcuket.com
[4] https://github.com/lfelipecunha/free5gc-Compose
[5] https://github.com/lfelipecunha/my5G-RANTester
[6] https://github.com/lfelipecunha/elastic5gc-amf-load-balancer
[7] https://golang.org/

Figura 12: Protótipo do *core*. Cada caixa representa o serviço e na parte inferior de cada uma, é indicado *software* utilizado



Fonte: Elaborado pelo autor

é identificado o tipo de mensagem, relacionada ou não relacionada a UEs, para a realização do tratamento da mensagem. Se não relacionada a UEs o balanceador de carga faz o tratamento e responde, porém, se relacionada a UEs, seleciona o AMF e encaminha a requisição para o tal.

Tendo a estrutura do balanceador de carga funcional, desenvolveu-se os componentes do orquestrador. Todos estes componentes foram desenvolvidos em Python na versão 3.6, sendo eles: Sonda[8]; Monitor de Recursos[9]; Gestor de Elasticidade, NFVO, VNFM e VIM[10]. Para a realização da geração de carga foi utilizado a biblioteca python statsmodel[11]. Uma premissa adotada foi que a conexão de um UE fica designada para um AMF até o fim de seu registro, ou seja, não há troca de contexto do UE entre AMFs, isto foi motivado por não existir tal funcionalidade no free5GC.

[8] https://github.com/lfelipecunha/elastic5gc-probe
[9] https://github.com/lfelipecunha/elastic5gc-monitor
[10] https://github.com/lfelipecunha/elastic5gc-elastic_manager
[11] https://www.statsmodels.org/stable/index.html

Algorithm 1 Balanceador de carga

Entrada: Requisição NGAP.
Saída: Resposta NGAP.

```
1: registra callback no NRF()
2: busca lisa de AMFs disponíveis no NRF()
3: inicia thread com LISTENANDSERVE(sctpSocket)
4: function LISTENANDSERVE(sctpSocket)
5:     recebe nova conexão; abre nova thread
6:     if requisição está relacionada a UEs then
7:         UE ← NGAP_ID
8:         AMF ← SELECIONAAMF(UE)
9:         encaminha requisição para AMF
10:    else
11:        trata a requisição não relacionada a UEs
12:    end if
13: end function
14: function SELECIONAAMF(UE)
15:    if UE já está relacionada a um AMF then
16:        return AMF já relacionado ao UE
17:    else
18:        return próximo AMF da lista circular que não esteja bloqueado
19:    end if
20: end function
```

5.2.2 Cenários de teste

Para a validar o modelo Elastic5GC foram realizados cenários de testes para simular as conexões de UEs na rede, comparando-o com o free5GC sem elasticidade, sendo eles:

Cenário 1: realiza-se um ciclo de conexões começando com um número mínimo de UEs e a cada etapa do ciclo aumenta-se a quantidade de conexões a uma taxa polinomial. Este ciclo ocorre até que todos os UEs estejam conectados. Neste cenário serão criadas conexões de registros de UEs em uma taxa crescente de 10 UEs por etapa, executado em 19 etapas com intervalo de 3 segundos entre etapas, sendo a primeira com 10 UEs e a última com 190 UEs, em um total de 1900 UEs.

Cenário 2: realiza-se um ciclo de conexões começando com um número máximo de UEs e a cada etapa do ciclo diminui-se a quantidade de conexões a uma taxa polinomial. Este ciclo ocorre até que todos os UEs estejam conectados. Neste cenário serão criadas conexões de registros de UEs em uma taxa decrescente de 10 UEs por etapa, executado em 19 etapas com intervalo de 3 segundos entre etapas, sendo a primeira com 190 UEs e a última com 10 UEs, em um total de 1900 UEs.

Cenário 3: realiza-se um ciclo de conexões em que cada etapa o número de UEs é calculado com base em uma distribuição de Poisson. Neste cenário serão criadas conexões de re-

gistros de UEs, usando valores aleatórios com base na distribuição de Poisson. Para isso, foi utilizada uma média de 50 UEs por etapa, executado em 18 etapas com intervalo de 3 segundos entre etapas, em um total de 878 UEs.

Para executar estes cenários de teste foram utilizadas simulações tanto para a rede de acesso de rádio, quanto para as UEs, para o tal foi utilizado o projeto *open-source* my5G-RANTester, que faz parte da my5G Initiative[12]. Este projeto permite a criação de *templates* para executar registros de UEs e testes de transmissão de dados em um *core* 5G.

5.3 Considerações Parciais

Neste capítulo foi apresentado o modelo de avaliação, com as métricas de Capacidade de atendimento e Esforço computacional, bem como a apresentação do protótipo construído para validação do modelo Elastic5GC. No próximo capítulo são apresentados os resultados obtidos de acordo com itens supracitados.

[12]https://my5g.github.io/

6 RESULTADOS

Neste capítulo são apresentados os resultados da aplicação do modelo Elastic5GC. Na Seção 6.1 é apresentado o formato da avaliação do motor de predição, de tal forma a obter os valores de entrada para o ARIMA. Já na Seção 6.2 são apresentados os resultados do modelo seguindo os cenários supracitados e comparando-os com o estado da arte.

6.1 Avaliação do Motor de Predição

Como apresentado no Capítulo 4, o modelo Elastic5GC utiliza séries temporais para a realização da predição de carga, mais especificamente o ARIMA. Os três parâmetros de entrada p, d e q, conforme apresentado na Seção 2.2.4, descrevem o comportamento da série temporal e, por consequência, a projeção da carga. Para definir estes parâmetros foram realizados testes, primeiramente, com cargas sintéticas e para refinamento foram testados com carga capturadas a partir do free5GC rodando sem modificações ou balanceamento de carga.

Para a avaliação de desempenho foram geradas 3 cargas sintéticas, crescente, decrescente e usando a distribuição de Poisson. Na carga crescente foi projetado um crescimento de 2% a cada etapa, iniciando em 2% e terminando em 100%. Já para a carga decrescente foi projeto um decréscimo de 2% a cada etapa, iniciando em 100% e finalizando em 2%. Para a carga baseada na distribuição de Poisson foram gerados 50 valores aleatórios com base na distribuição, tendo como média 80%.

Para verificar a eficiência dos valores, foram comparados os resultados obtidos pela projeção e o dado de entrada. Para cada teste foram calculados o valor Médio de Erros (ME) e o Desvio Médio Absoluto (DMA). O valor ME é a média das diferenças entre o valor projetado e o de entrada. O valor esperado para este indicador é o mais próximo de zero, contudo, este valor sozinho não é confiável pois, por exemplo, se uma medição for 30 acima e outra 30 abaixo a média será zero. Desta forma, o indicador de DMA apresenta o desvio médio das diferenças entre a carga projetada e o dado de entrada. O valor esperado para este indicador é o mais próximo de zero. Dessa forma, este indicador complementa o ME, por exemplo, no caso anterior o desvio médio padrão será 30, logo apesar da média de erros ser zero o desvio é muito alto, logo a eficiência da projeção é ruim, desta forma, o que se busca é a proximidade de zero tanto para ME, quanto para o DMA, o algoritmo utilizado para a criação destes testes esta disponibilizado em repositório público [1].

[1] https://gist.github.com/lfelipecunha/e68dc8f2c74ba568816533ae6619be97

Tabela 2: Valores de erros dos algoritmos. Foram avaliados o EM, DMA e EDMG. Foram testados os parâmetros do ARIMA de 0 até 2, pois são as formas mais comuns de utilização do algoritmo (BROCKWELL; DAVIS, 2016). Os valores em verde são os melhores resultados em cada uma das colunas. As linhas destacadas são os resultados escolhidos para serem analisados.

	EDMG	Poisson		Crescente		Decrescente	
		ME	DMA	ME	DMA	ME	DMA
Arima(0,0,0)	58,07	-1,90	6,52	-30,00	7,24	30,00	7,24
Arima(0,0,1)	58,28	-2,07	6,37	-30,19	7,18	30,19	7,18
Arima(0,0,2)	49,52	-1,01	7,59	-21,04	9,66	21,04	9,66
Arima(0,1,0)	24,21	-0,39	7,81	-11,52	0,48	11,52	0,48
Arima(0,1,1)	22,78	-0,17	7,88	-10,57	0,47	10,57	0,47
Arima(0,1,2)	45,20	0,87	30,98	-5,97	4,04	5,97	4,04
Arima(0,2,0)	36,14	1,42	32,05	-2,00	0,00	2,00	0,00
Arima(0,2,1)	19,44	1,47	15,30	-2,00	0,00	2,00	0,00
Arima(0,2,2)	30,44	-0,37	12,85	-7,51	5,41	7,51	5,41
Arima(1,0,0)	25,77	-2,01	6,52	-12,31	0,62	12,31	0,62
Arima(1,0,1)	27,39	-3,28	7,34	-11,58	0,95	9,82	2,89
Arima(1,0,2)	25,57	-2,16	8,53	-6,68	4,68	4,93	5,64
Arima(1,1,0)	10,84	-0,13	8,03	-2,00	0,00	2,00	0,00
Arima(1,1,1)	15,24	-1,74	8,99	-2,49	0,95	0,73	2,51
Arima(1,1,2)	28,27	-1,48	22,28	-2,49	0,95	0,73	2,51
Arima(1,2,0)	27,24	1,37	23,20	-2,00	0,00	2,00	0,00
Arima(1,2,1)	21,24	1,00	15,82	-2,52	1,02	0,76	2,44
Arima(1,2,2)	18,81	-1,27	12,87	-2,56	1,04	0,80	2,44
Arima(2,0,0)	11,00	-1,74	6,43	-2,06	0,06	2,06	0,06
Arima(2,0,1)	14,98	-2,94	7,59	-2,47	0,88	0,71	2,58
Arima(2,0,2)	15,32	-2,04	8,85	-2,46	0,88	0,70	2,58
Arima(2,1,0)	11,09	0,27	8,15	-2,00	0,00	2,00	0,00
Arima(2,1,1)	14,35	-1,08	8,76	-2,48	0,95	0,72	2,51
Arima(2,1,2)	22,48	-0,62	17,35	-2,48	0,95	0,72	2,51
Arima(2,2,0)	23,36	0,79	19,91	-2,00	0,00	2,00	0,00
Arima(2,2,1)	26,26	-3,07	18,58	-2,52	1,02	0,76	2,44
Arima(2,2,2)	35,34	-6,51	22,04	-3,03	2,00	-0,48	4,79

Para verificar a eficiência de forma geral, foi calculado o indicador de Erros e Desvio Médio Absoluto Global (EMDA). Este indicador faz uma média global da eficiência dos parâmetros em todas as cargas de teste, conforme a Equação 6.1, onde P_{ME} e P_{DMA} são o EM e o DMA para carga com distribuição de Poisson, respectivamente. C_{ME} e C_{DMA} são o EM e o DMA para carga crescente respectivamente e, D_{ME} e D_{DMA} são o EM e o DMA para carga decrescente respectivamente.

$$EDMG = \frac{\|P_{ME}\| + P_{DMA} + \|C_{ME}\| + C_{DMA} + \|D_{ME}\| + D_{DMA}}{3} \tag{6.1}$$

A Tabela 2 apresenta resultados dos testes com as cargas sintéticas, com base nos resultados foram realizadas duas avaliações, primeiramente os itens melhor resultado (itens em verde na tabela) e em segundo lugar as combinações que obtiveram os menores EDMG (linhas em amarelo). Pode-se observar que para houve certa dispersão nos resultados, contudo 3 combinações de parâmetros chamam a atenção. Primeiramente, a combinação $Arima(1, 1, 0)$, neste caso houveram os melhores EDMG e ME e DMA para as cargas crescente e decrescente, mesmo que não tenham sido os melhores valores para a distribuição de Poisson, a distância para os melhores valor é de 0,04 e 0,05 para ME e DMA, respectivamente. Outro resultado relevante foi para a combinação $Arima(2, 0, 0)$ neste caso nenhum valor foi o melhor, porém todos os valores estão muito próximos dos melhores. Por último temos a combinação $Arima(2, 1, 0)$, neste caso para as cargas crescentes e decrescentes foram obtidos os melhores valores, já o EDMG e os ME e DMA da distribuição de Poisson foram próximos aos melhores.

Para identificar dentre estas 3 combinações qual a utilizar, foi realizado um comparativo com capturas de carga do sistema executando com um AMF sem elasticidade e sem balanceamento de carga, para este comparativo foram utilizados os mesmos indicadores utilizados nas cargas sintéticas. Além disso, foram geradas 3 cargas, uma crescente, uma decrescente e outra utilizando a distribuição de Poisson. Nestes casos, as cargas foram geradas pela chegada de UEs, ou seja, para a carga crescente foi estipulada uma chegada inicial de dois UEs por segundo com crescimento gradual até 200 UEs por segundo, para a carga decrescente foi estipulada uma carga inicial de 200 UEs por segundo com um decréscimo gradual até dois UEs por segundo, já para a carga com distribuição de Poisson foi estipulada uma chegada média, seguindo a distribuição, de 80 UEs por segundo.

Como pode ser observado na Tabela 3, a combinação que trouxe melhor resultado foi a $Arima(1, 1, 0)$, dado que o EDMG foi o mais baixo dentre os testados. Sendo assim, para os experimentos de predição de carga e elasticidade foram utilizados estes valores no algoritmo do ARIMA.

Tabela 3: Valores de erros dos algoritmos. Foram avaliados a ME, DMA e EDMG. Foram testados as 3 combinações selecionadas pelas cargas sintéticas com cargas capturadas.

	EDMG	Poisson		Crescente		Decrescente	
		ME	DMA	ME	DMA	ME	DMA
Arima(1,1,0)	-0,83	11,71	24,94	-4,67	4,52	4,72	7,27
Arima(2,0,0)	-6,44	12,02	53,37	-12,39	3,47	19,73	8,33
Arima(2,1,0)	-3,16	12,92	27,87	-5,13	5,28	3,13	7,94

6.2 Avaliação do Elastic5GC

Nesta seção são apresentados os resultados do modelo Elastic5GC nos três cenários apresentados na Seção 5.2.2. Cada cenário foi executado em três ambientes, sendo eles: (i) *Baseline*: utilizado o *core* com apenas um AMF fixo; (ii) Balanceador com dois AMFs: utilizando o *core* com dois AMFs fixos e com o balanceador de carga e; (iii) Elastic5GC: utilizando o *core* com um AMF inicial e com o protótipo do modelo Elastic5GC. Para cada cenário e ambiente foram calculados a capacidade de atendimento, através da métrica Tempo Médio de Resposta (vide Seção 5.1.1, e o esforço computacional (vide Seção 5.1.2). Para medir tempo de resposta de cada UE foi mensurado o tempo entre o momento em que a RAN enviou o primeiro pacote (referente ao UE) para o núcleo e, o momento da chegada do pacote confirmando o registro.

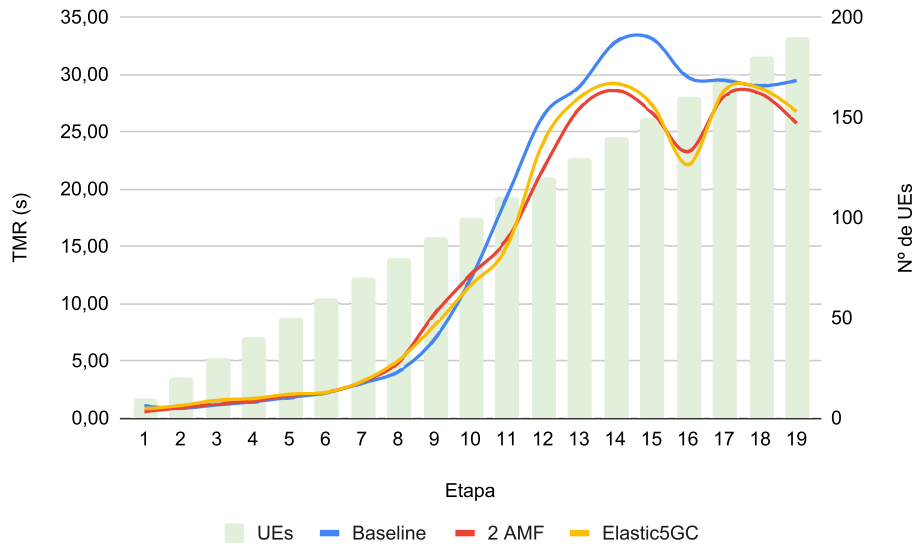
Estes testes foram executados utilizando dois nós físicos com a seguinte configuração de *hardware*: processador Octa Core Intel(R) Atom(TM) 2.40GHz; 8GB de memória RAM e; disco rígido de 250GB. A ferramenta de testes e os componentes do orquestrador foram alocados em um nó, enquanto o *core*, o balanceador de carga e a sonda foram alocados no outro nó. O intervalo entre as medições de carga foi de 2 segundos e para o *lookahead* foram utilizados 5 intervalos, calculados conforme descrito na seção 4.2.1.2, dado que o tempo para alocação do *container* foi de 3,5 segundos e o tempo médio para inicialização foi de 6,5 segundos.

6.2.1 Cenário 1 - Carga Crescente

O primeiro cenário executado foi o de carga crescente conforme descrito na seção 5.2.2. A figura 13 apresenta a chegada de UEs por etapa e o comportamento do Tempo Médio de Resposta por etapa para cada ambiente. Um ponto a salientar que até a etapa dez, ou seja os primeiros 550 UEs, o tempo médio de resposta nos três cenários é muito parecido. A partir da etapa 11 há um descolamento do ambiente *Baseline* dos demais ambientes, sendo assim, fica evidenciado que ao passo que há um incremento de UEs se fazem necessários mais recursos computacionais para atender a demanda de chegada. Outro ponto importante é um decréscimo abrupto no tempo de médio de resposta na etapa 15, este comportamento deve-se a simultanei-

dade das requisições, pois há o aumento do tempo de resposta e por consequência as conexões ficam ativas por mais tempo. Neste caso em específico, o que ocorreu foi o término de um número considerável de requisições e, assim, reduzindo a simultaneidade que afeta diretamente na capacidade de atendimento do *core*.

Figura 13: Comportamento do tempo médio de resposta e quantidade de UEs para cada etapa para o cenário 1 (vide seção 5.2.2) utilizando o Elastic5GC.



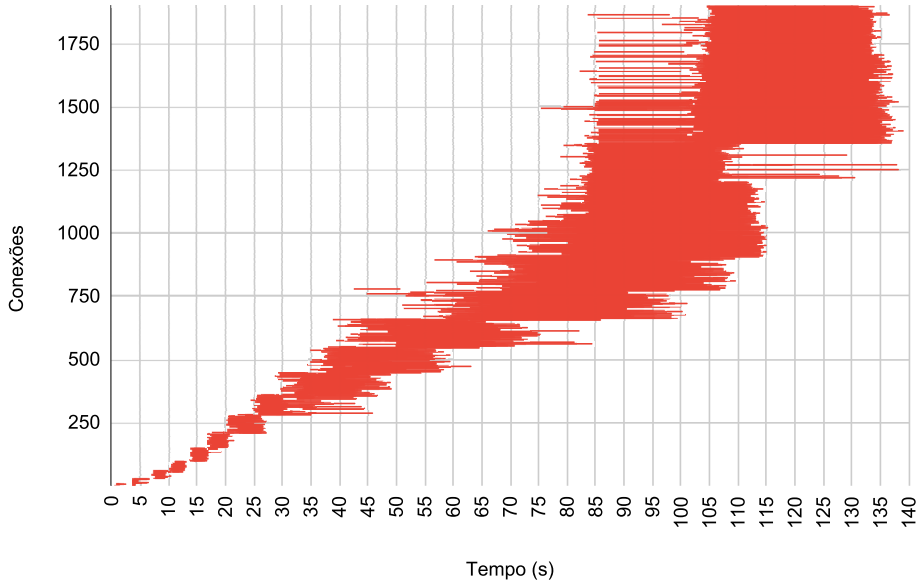
Fonte: Elaborado pelo autor

Pode-se ver na figura 14 o comportamento da simultaneidade para a carga crescente. As primeiras etapas (primeiros 20 segundos de execução) tem um espaçamento entre a finalização do registros e o início da próxima etapa, isto se dá pelo baixo volume de UEs chegando e pelo intervalo te tempo entre cada etapa. No entanto, a partir do segundo 20 o tempo de resposta aumenta devido ao aumento na quantidade de UEs, o efeito é que as etapas passam a se sobrepor, ou seja, uma nova rajada de UEs é lançada enquanto o *core* ainda está processando a anterior. Pode se observar que entre os segundos 85 e 110 há a maior quantidade de UEs sendo registrados simultaneamente, porém no segundo 110 há um encerramento em massa, fazendo com que a simultaneidade nos segundos subsequentes seja reduzida em cerca de 60%.

Na figura 15 pode-se visualizar o comportamento da elasticidade frente a carga crescente. Um ponto importante é que a carga não cresce regularmente, ou seja, existem picos de subida e queda, isto ocorre devido ao comportamento de chegada dos UEs, pois os UEs chegam em rajadas e vão sendo processados e liberados pelo AMF, logo há um pico de subida na chegada e um pico de descida logo antes de subir novamente. Este comportamento dificulta a predição de carga, gerando ineficiência na alocação sendo registrados 3 pontos acima do limiar superior.

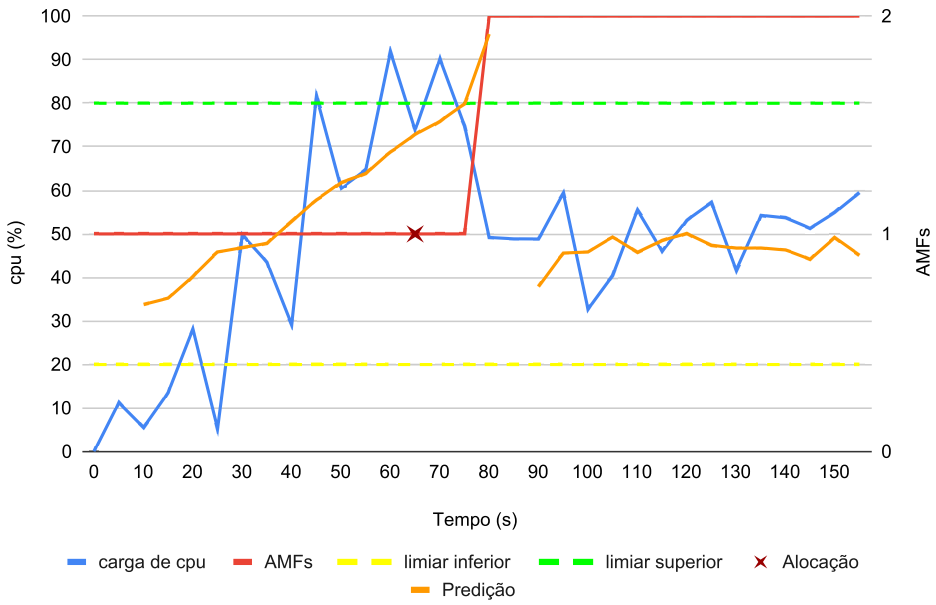
Para provar os benefícios do modelo Elastic5GC, foram realizados comparativos gerais da execução, ou seja, considerando todas as etapas e comparando estas execuções entre si, os resultados são apresentados na Tabela 4. Com relação ao tempo médio de resposta quem obteve

Figura 14: Tempo de vida de cada conexão para registro dos UEs e suas ocorrências na linha do tempo para o ambiente do modelo Elastic5GC. Quanto maior a área coberta por uma sessão vertical de largura x, maior é a quantidade de conexões simultâneas.



Fonte: Elaborado pelo autor

Figura 15: Comportamento da alocação de AMFs e da carga de CPU medida para o cenário 1 (vide Seção 5.2.2) utilizando o Elastic5GC. Um ponto importante é que há um desvio na predição da alocação do segundo AMF, pois o carga de CPU ultrapassa o limiar superior antes da alocação do mesmo.



Fonte: Elaborado pelo autor

o melhor resultado foi o ambiente com dois AMFs, sendo 10,88% mais eficiente que no ambiente *Baseline* e reduzindo em 14,41% o desvio padrão das respostas. O modelo Elastic5GC apesar de não ter o menor tempo médio de resposta, sendo 9,47% mais eficiente que o *Baseline* e reduzindo 17,55% o desvio padrão das respostas, foi apenas 1,58% menos eficiente que o ambiente com dois AMFs. Esta pequena diferença se da pelo desvio na precisão do ARIMA para as cargas crescentes como foi evidenciado no teste apresentado na Tabela 3 e a Figura 15 apresenta este desvio de forma detalhada.

Tabela 4: Comparativo das métricas de Capacidade de Atendimento e Esforço Computacional entre os três ambientes no cenário com carga crescente.

		Capacidade de Atendimento		**Esforço Computacional**
		TMR	**Desvio Padrão**	
Baseline	execução	22,06s	12,42s	113,29s
2 AMF	execução	19,66s	10,63s	318,82s
	comparativo *Baseline*	-10,88%	-14,41%	181,42%
Elastic5GC	execução	19,97s	10,24s	196,76s
	comparativo *Baseline*	-9,47%	-17,55%	73,68%
	comparativo 2 AMFs	1,58%	-3,67%	-38,28%

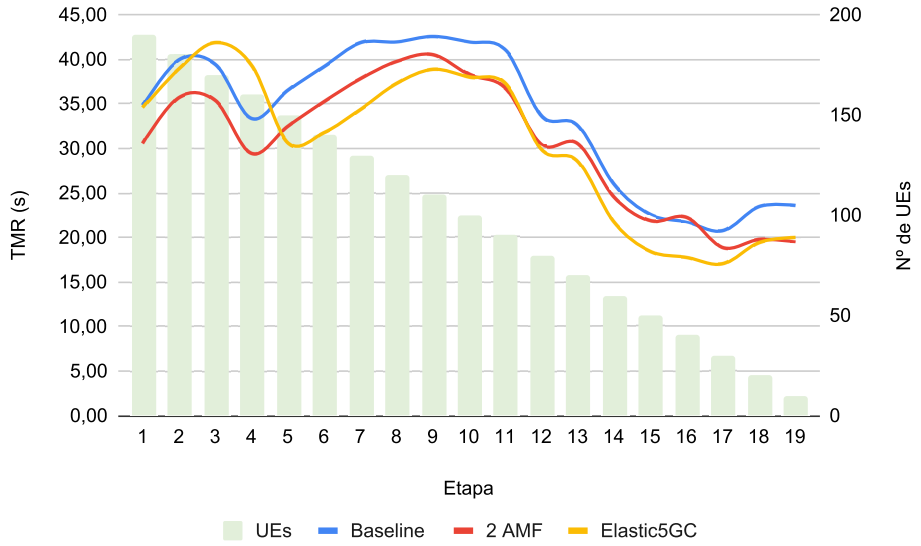
Com relação ao esforço computacional é esperado que o ambiente *Baseline* seja mais eficiente, pois há apenas um AMF durante todo o tempo de execução, sendo assim, a comparação relevante é entre ambiente com dois AMFs e o ambiente Elastic5GC, pois assim pode-se validar a eficiência da elasticidade. Neste caso, o Elastic5GC foi 38,28% mais eficiente, isto ocorre pela adaptabilidade que este modelo agrega, pois, como pode ser evidenciado na Figura 15, a alocação do segundo AMF só ocorre quando o sistema se encontra em sobrecarregado.

6.2.2 Cenário 2 - Carga Decrescente

O segundo cenário executado foi o de carga decrescente conforme descrito na Seção 5.2.2. A Figura 16 apresenta a chegada de UEs por etapa e o comportamento do Tempo Médio de Resposta por etapa para cada ambiente, um ponto a salientar que até a etapa 5, o tempo médio de resposta do ambiente Elastic5GC é maior que os demais, isto ocorre por dois motivos. Primeiramente, se comparado ao ambiente com dois AMFs, há um intervalo para que o algoritmo de predição identifique que há a necessidade de alocação (evidenciado na Figura 17). Além disto, quando comparado ao ambiente com *Baseline* existe um elemento de rede a mais, pois, no caso do Elastic5GC, existe o balanceador de carga entre a gNB e o AMF que acrescenta cerca de 100 ms em cada requisição. Contudo, após a etapa 5 o tempo médio do Elastic5GC se aproxima dos demais, chegando, em algumas etapas, a ser o mais eficiente.

Na Figura 17 pode-se observar o comportamento da elasticidade frente a carga decrescente. Primeiramente há um desvio na primeira ação de alocação, isto ocorreu devido ao pico inicial e por existirem poucas medições para a predição, mesmo assim somente 2 registros acima do

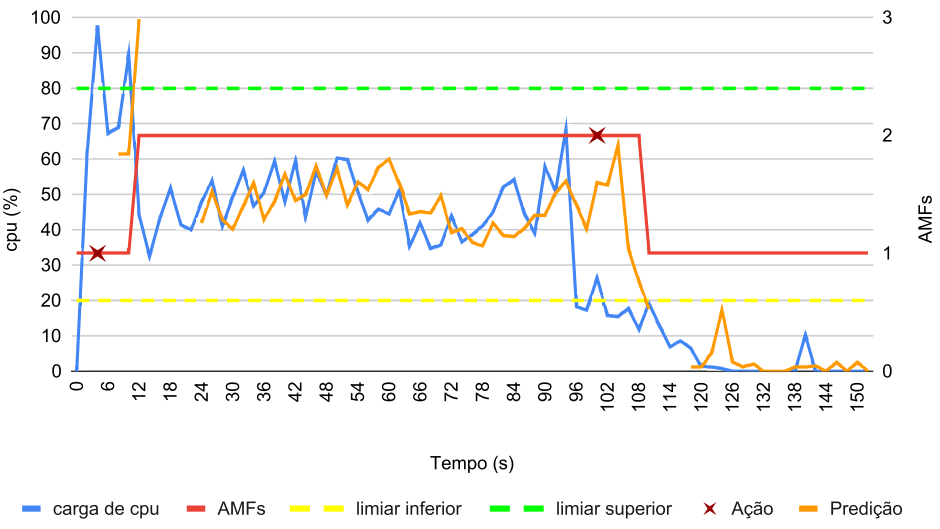
Figura 16: Comportamento do tempo médio de resposta e quantidade de UEs para cada etapa para o cenário 2 (vide Seção 5.2.2).



Fonte: Elaborado pelo autor

limiar foram registrados antes da alocação. Outro ponto que ocorreu foi a identificação de 4 registros abaixo do limiar inferior antes da desalocação, isto ocorreu pois houve uma queda abrupta da carga devido ao encerramento de muitos UEs simultaneamente.

Figura 17: Comportamento da alocação de AMFs e da carga de CPU medida para o cenário dois (vide Seção 5.2.2) utilizando o Elastic5GC. Dois pontos importantes: *(i)* há um desvio na identificação da subida carga e por consequência na alocação do segundo AMF; *(ii)* também há um desvio na identificação da queda de carga e por consequência na remoção do AMF



Fonte: Elaborado pelo autor

Para provar os benefícios do modelo Elastic5GC, foram realizados comparativos gerais da execução, ou seja, considerando todas as etapas e comparando estas execuções entre si, os resultados são apresentados na Tabela 5. Com relação ao tempo médio de resposta quem obteve o melhor resultado foi o ambiente com dois AMFs, sendo 9,27% mais eficiente que no ambiente *Baseline*. O modelo Elastic5GC apesar de não ter o menor tempo médio de resposta, sendo 6,95% mais eficiente que o *Baseline*, foi apenas 1,58% menos eficiente que o ambiente com dois AMFs. Além disto, Elastic5GC reduz 12,25% o desvio padrão das respostas se comparado ao *Baseline*, o que mostra consistência do resultado obtido. Esta pequena diferença se da pelo desvio na precisão do ARIMA para as cargas decrescentes como foi evidenciado no teste apresentado na Tabela 3. A Figura 17 apresenta este desvio de forma detalhada.

Tabela 5: Comparativo das métricas de Capacidade de Atendimento e Esforço Computacional entre os três ambientes no cenário com carga decrescente.

		Capacidade de Atendimento		**Esforço Computacional**
		TMR	**Desvio Padrão**	
Baseline	execução	36,67s	8,16s	97,54s
2 AMF	execução	33,27s	8,33s	197,09s
	comparativo *Baseline*	-9,27%	2,08%	102,06%
Elastic5GC	execução	34,12s	7,16s	137,62s
	comparativo *Baseline*	-6,95%	-12,25%	41,09%
	comparativo 2 AMFs	2,55%	-14,05%	-30,17%

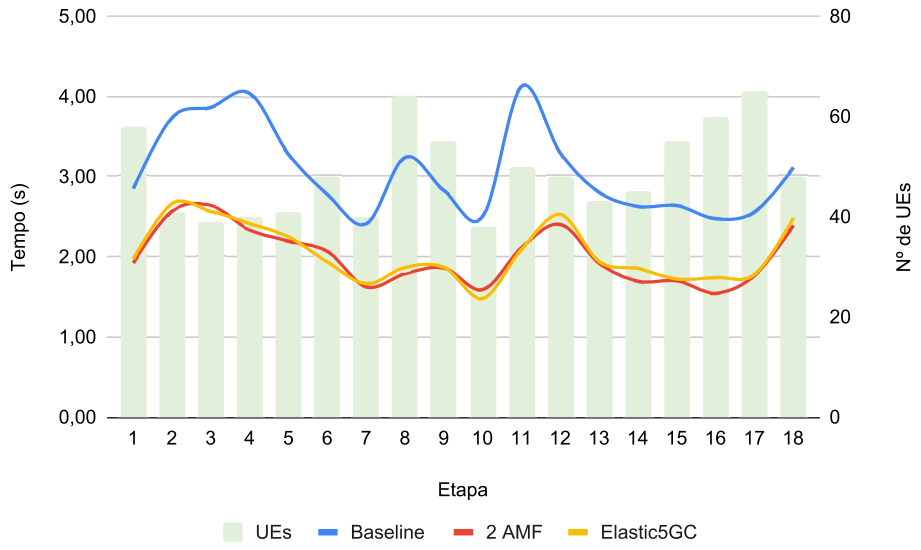
Com relação ao esforço computacional é esperado que o ambiente *Baseline* seja mais eficiente, pois há apenas um AMF durante todo o tempo de execução. Sendo assim, a comparação relevante é entre o ambiente com dois AMFs e o ambiente Elastic5GC, pois assim pode-se validar a eficiência da elasticidade. Neste caso, o Elastic5GC foi 30,17% mais eficiente, isto ocorre pela adaptabilidade que este modelo agrega, pois, como pode ser evidenciado na Figura 17, quando há uma tendência de queda de carga evidenciando a não necessidade da utilização dos recursos computacionais atuais, um AMF é desalocado.

6.2.3 Cenário 3 - Distribuição de Poisson

O terceiro cenário executado foi o de carga com base na distribuicão de Poisson conforme descrito na Seção 5.2.2. A Figura 18 apresenta a chegada de UEs por etapa e o comportamento do Tempo Médio de Resposta por etapa para cada ambiente, um ponto a salientar que desde a primeira etapa, o tempo médio de resposta do ambiente *Baseline* é superior ao tempo de resposta dos demais ambientes. Esse comportamento deve-se ao fato de que para atender a demanda de chegadas de UEs são necessários mais recursos computacionais do que um AMF possui. Percebe-se também que os ambientes com dois AMFs e Elastic5GC tem comportamento muito parecido, isto ocorre, pois o algoritmo de predição de carga foi bastante eficiente, fazendo a alocação do segundo AMF, logo antes do sistema entrar em sobrecarga, o que fica evidenciado

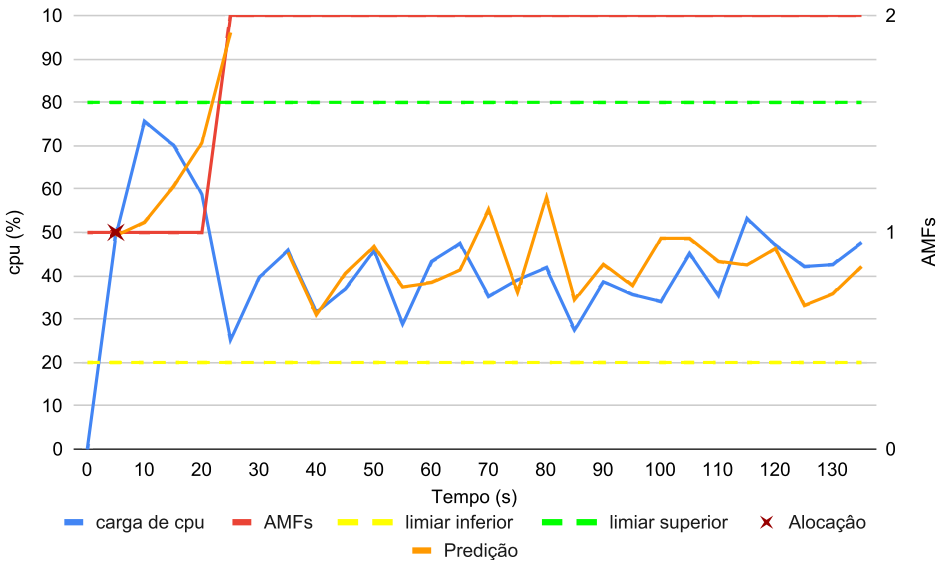
na Figura 19.

Figura 18: Comportamento do tempo médio de resposta e quantidade de UEs para cada etapa para o cenário três (vide Seção 5.2.2).



Fonte: Elaborado pelo autor

Figura 19: Comportamento da alocação de AMFs e da carga de CPU medida para o cenário três (vide Seção 5.2.2) utilizando o Elastic5GC. Neste caso a predição de carga foi adequada, pois a alocação do segundo AMF ocorre antes que o sistema entre em sobrecarga.



Fonte: Elaborado pelo autor

Para provar os benefícios do modelo Elastic5GC, foram realizados comparativos gerais da execução, ou seja, considerando todas as etapas e comparando estas execuções entre si, os resultados são apresentados na Tabela 6. Com relação ao tempo médio de resposta quem obteve

o melhor resultado foi o ambiente com dois AMFs, sendo 34,87% mais eficiente que no ambiente *Baseline* tendo uma redução de 53,84% no desvio padrão do tempo de resposta. O modelo Elastic5GC apesar de não ter o menor tempo médio de resposta, sendo 33,22% mais eficiente que o *Baseline*, foi apenas 2,53% menos eficiente que o ambiente com dois AMFs. Além disso, Elastic5GC reduz em 54,84% o desvio padrão das respostas se comparado ao *Baseline*, o que mostra consistência do resultado obtido. Esta pequena diferença se da pelo fato de que no ambiente Elastic5GC durante os primeiros 20 segundos opera apenas com um AMF, por mais que o ambiente não apresente sobrecarga há concorrência maior do que se comparado com com o ambiente com dois AMFs. Entretanto, este indicador é pouco comprometido, pois a predição da carga futura foi muito eficiente, dado que a alocação do segundo AMF ocorre antes da sobrecarga do sistema, conforme apresentado na Figura 19.

Tabela 6: Comparativo das métricas de Capacidade de Atendimento e Esforço Computacional entre os três ambientes no cenário com carga baseada na distribuição de Poisson.

		Capacidade de Atendimento		**Esforço Computacional**
		TMR	**Desvio Padrão**	
Baseline	execução	3,04s	1,24s	54,42s
2 AMF	execução	1,98s	0,58s	108,65s
	comparativo Baseline	-34,87%	-53,84%	99,65%
Elastic5GC	execução	2,03s	0,56s	96,62s
	comparativo Baseline	-33,22%	-54,84%	77,55%
	comparativo 2 AMFs	2,53%	-3,45%	-11,07%

Com relação ao esforço computacional, assim como nas demais cargas, é esperado que o ambiente *Baseline* seja mais eficiente, pois há apenas um AMF durante todo o tempo de execução. Dessa forma, a comparação relevante é entre o ambiente com dois AMFs e o ambiente Elastic5GC, pois assim pode-se validar a eficiência da elasticidade. Neste caso, o Elastic5GC foi 11,07% mais eficiente, isto ocorre pela adaptabilidade que este modelo agrega, pois, como pode ser evidenciado na Figura 19, a alocação do AMF somente ocorre quando existe uma tendência de aumento de carga evidenciando uma sobrecarga futura.

6.2.4 Discussão sobre estado da arte

Após apresentar os resultados do protótipo Elastic5GC, essa seção destina-se a comparar o modelo com o estado da arte. O modelo buscou solucionar o gerenciamento de funções de rede do *core* 5G, mais especificamente do AMF, visando a redução do tempo de resposta e a redução do uso de recursos computacionais. Em termos do tempo de resposta foi possível atingir uma redução do tempo médio de resposta de 33,22% para carga com distribuição de Poisson. Já em termos de redução do uso de recursos o modelo chegou a reduzir em 38,28% para o caso de carga crescente.

Como foi demonstrado nos trabalhos relacionados, apenas um trabalho utiliza proatividade

no intuito de reduzir o tempo de resposta para *core* (BANERJEE et al., 2015). O trabalho de Banerjee et al. (2015) foi implementado para o core 4G utilizando uma abordagem de predição baseada em épocas, esta abordagem pode não ser precisa, pois não é adaptativa ao comportamento dinâmico da rede, dado que as épocas são de tamanho estático e a ação de elasticidade só ocorre ao fim de cada época, ou seja, mesmo que o sistema entre em sobrecarga ou em ociosidade é preciso que a época seja encerrada para que ação ocorra. Uma forma de suprimir este problema seria reduzir o intervalo de tempo (tamanho) da época, porém o cálculo de predição seria prejudicado, dado que, neste modelo, somente os dados capturados no intervalo da época são utilizados como entrada para a predição, não sendo considerados dados de épocas anteriores.

Já o Elastic5GC utiliza o ARIMA para a realização da predição, tendo como entrada todas as medições a partir da última ação de elasticidade (ou da inicialização do sistema), garantindo assim uma melhor projeção de carga. Já as ações de elasticidade podem ocorrer a qualquer momento, tendo como limitação o tempo mínimo de intervalo entre as medições. Desta forma, a alocação ou desalocação acontece o mais próximo possível da necessidade do sistema.

7 CONCLUSÃO

A área de telecomunicações móveis está em um momento de grande mudança, a publicação da *Release* 15 feita pelo 3GPP (3GPP, 2019b) definiu uma nova direção para as pesquisas e abriu um novo horizonte de possibilidades. Mesmo que algumas desta possibilidades já tenham sido observadas dentro do *core* EPC, o *core* SBA ainda tem diversos pontos para exploração. O Elastic5GC explora o ponto em aberto que trata da dinamicidade dos dispositivos móveis e como se dará a relação com o *core*. Neste contexto, um modelo de elasticidade horizontal proativa do serviço AMF foi apresentada, no intuito de aumentar a capacidade de atendimento e a utilização de recursos computacionais no atendimento das inúmeras requisições feitas pelos UEs, buscando prever o comportamento da aplicação e assim alocar os recursos necessários no *core*.

A Seção 1.2 apresentou a seguinte questão de pesquisa: *Como deve ser o modelo de elasticidade proativo para o core 5G definido, na Release 15, que seja transparente para a rede de acesso, reduzindo a utilização de recursos computacionais?* Para respondê-la foi criado o modelo Elastic5GC que possui como elementos chave o Gestor de elasticidade e o Balanceador de carga. O Gestor de elasticidade é responsável pela alocação da quantidade de AMFs necessárias para garantir o menor tempo de resposta das requisições dos UEs, bem como, desalocar AMFs para reduzir o uso de recursos computacionais mantendo o menor tempo de resposta. Já o balanceador de carga é o viabilizador da transparência, pois as redes de acesso não precisam ter conhecimento de quantos ou quais AMFs existem para pode se comunicar com o *core*.

Após a definição do modelo, foram definidos os testes a serem realizados para validar o modelo. O Capitulo 5 apresenta como os testes foram modelos, bem como, a ferramenta de testes para simulação da rede de acesso e UE e como o protótipo foi construído. Além disso, foram modeladas três cargas de trabalhos para os testes: crescente, decrescente e com base na distribuição de Poisson. Essas cargas de trabalho serviram para simular ambientes reais de utilização. Os testes comparam execuções dessas cargas de trabalho sendo gerenciadas pelo Elastic5GC com execuções sem gerenciamento de elasticidade e com número fixo de AMFs. Para a avaliação, foram utilizadas duas métricas: tempo médio de resposta e esforço computacional.

Conforme apresentado na Seção 6.2, o Elastic5GC reduziu o tempo de resposta se comparado ao ambiente com apenas um AMF e reduziu o esforço computacional se comparado ao ambiente com dois AMFs, estes resultados foram obtidos para as três cargas testadas. Para a carga crescente reduziu em 9,47% o tempo de resposta e 38,28% o esforço computacional. Para a carga decrescente reduziu 6,95% o tempo de resposta e 30,17% o esforço computacional. Já para carga com distribuição de Poisson reduziu 33,22% o tempo de resposta e 11,07% o esforço computacional. Apesar de não ter mantido o menor tempo de resposta se comparado ao ambiente com dois AMFs, o desvio foi pequeno sendo o maior de 2,55% na carga decrescente. Entretanto, a redução do esforço computacional foi expressiva chegando a 38,28% na carga

crescente, logo pode-se concluir que o Elastic5GC consegue reduzir efetivamente o esforço computacional mantendo o tempo de resposta muito próximo ao ótimo.

7.1 Contribuições

O Elastic5GC busca estabelecer uma arquitetura que permita a redução do uso de recursos do serviço AMF do *core* 5G SBA, utilizando como base a utilização de CPU. Tendo como principal característica:

1. A definição de um modelo de elasticidade proativa de serviços no *core* 5G, o que é um precedente importante visto que não existem outras soluções que realizem tal implementação. Definindo assim, heurísticas para determinar as ações de elasticidade no intuito de melhorar o uso de recursos computacionais e aumentar a capacidade de atendimento.

Em comparação com os trabalhos relacionados apresentados no Capítulo 3, pode-se destacar que o Elastic5GC apresenta um modelo de gerenciamento automático que não exige modificações nas gNBs ou mesmo no *core* 5G. Comparando-o com o trabalho de Banerjee et al. (2015), que também apresenta um modelo proativo que visa melhorar o tempo de resposta das requisições de UES, pode-se perceber que o Elastic5GC é mais adaptativo a dinamicidade do comportamento das chegadas dos UEs, pois, realiza as ações de elasticidade o mais próximo possível das necessidades, diferente do trabalho de Banerjee et al. (2015) que tem as ações limitadas a épocas de intervalo de tempo (tamanho) estático.

Além das contribuições acadêmicas para o estado da arte, Elastic5GC apresenta uma grande contribuição para a sociedade. Uma vez que estamos vivenciando a transformação das redes telecomunicação móvel, se faz necessário modelos que busquem eficiência no esforço computacional, pois, para as empresas de telecomunicação esta redução do esforço pode refletir em redução de custos operacionais. Já sob a óptica da população em geral a garantia do menor tempo de resposta viabiliza novas experiências como cidades inteligentes, rodovias inteligentes, *cyber* medicina e tantos outras oportunidades em que latência é crucial.

7.2 Limitações e Trabalhos Futuros

Nesta seção são elencadas algumas das limitações encontradas durante o desenvolvimento e implementação do protótipo. Uma limitação encontrada foi com relação ao algoritmo de predição de carga, ele gera alguns desvios principalmente nas cargas crescentes e decrescentes, conforme apresentado na Figura 15 e na Figura 17. Apesar dos desvios o impacto é pequeno se comparado ao ganho em redução de esforço computacional.

Uma segunda limitação é que este modelo realiza as predições de carga de CPU e baseia a alocação de AMFs com base neste recurso. Contudo, podem haver outros recursos como memória ou taxa de transferência de rede que impactem na capacidade de atendimento e na redução do uso dos recursos computacionais.

Algumas limitações do modelo Elastic5GC podem ser vistas como oportunidades para trabalhos futuros:

- Elasticidade de demais funções de rede do *core*: O modelo Elastic5GC realiza a elasticidade do AMF, porém, esta função de rede tem dependência de demais funções para seu funcionamento. Dessa forma, o tempo de resposta para as requisições poderiam ser melhorados se houve a replicação das demais funções.

- Posicionamento das funções de rede: Algo a ser explorado é a localização das funções de rede que devem ser instanciadas considerando os recursos em *cloud*, *edge* e *fog*, para buscar uma redução ainda maior no tempo de resposta.

- Explorar o plano de dados: O Elastic5GC faz a elasticidade e balanceamento apenas do plano de controle, logo, verificar se o modelo também entrega resultado na plano de dados é importante para as tecnologias emergentes.

REFERÊNCIAS

3GPP. **3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Network architecture (Release 14)**. 2017. n. TS 23.002. V14.1.0.

3GPP. **5G; NG-RAN; NG Application Protocol (NGAP) (3GPP TS 38.413 version 15.0.0 Release 15)**. 2018. n. TS 138 413. V15.0.0.

3GPP. **3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Release 15 Description; Summary of Rel-15 Work Items (Release 15)**. 2019. n. TR21.915. V15.0.0.

3GPP. **3rd Generation Partnership Project;Technical Specification Group Services and System Aspects;Service requirements for the 5G system;Stage 1 (Release 15)**. 2019. n. TS 22.261. V15.8.0.

ALAKEEL, A. M. et al. A guide to dynamic load balancing in distributed computer systems. **International Journal of Computer Science and Information Security**, Saudi Arabia, v. 10, n. 6, p. 153–160, 2010.

ALAWE, I.; HADJADJ-AOUL, Y.; KSENTINI, A.; BERTIN, P.; DARCHE, D. On the scalability of 5G Core network: the amf case. In: IEEE ANNUAL CONSUMER COMMUNICATIONS & NETWORKING CONFERENCE (CCNC), 2018., 2018, França. **Anais...** IEEE, 2018. p. 1–6.

AMOGH, P.; VEERAMACHANENI, G.; RANGISETTI, A. K.; TAMMA, B. R.; FRANKLIN, A. A. A cloud native solution for dynamic auto scaling of mme in lte. In: IEEE 28TH ANNUAL INTERNATIONAL SYMPOSIUM ON PERSONAL, INDOOR, AND MOBILE RADIO COMMUNICATIONS (PIMRC), 2017., 2017, Canadá. **Anais...** IEEE, 2017. p. 1–7.

ARTEAGA, C. H. T.; ANACONA, F. B.; ORTEGA, K. T. T.; RENDON, O. M. C. A scaling mechanism for an evolved packet core based on network functions virtualization. **IEEE Transactions on Network and Service Management**, Colômbia, v. 17, n. 2, p. 779–792, 2019.

BANERJEE, A.; MAHINDRA, R.; SUNDARESAN, K.; KASERA, S.; MERWE, K. Van der; RANGARAJAN, S. Scaling the LTE control-plane for future mobile access. In: ACM CONFERENCE ON EMERGING NETWORKING EXPERIMENTS AND TECHNOLOGIES, 11., 2015, USA. **Proceedings...** ACM, 2015. p. 1–13.

BROCKWELL, P. J.; DAVIS, R. A. **Introduction to time series and forecasting**. USA: springer, 2016.

BUYAKAR, T. V. K.; AGARWAL, H.; TAMMA, B. R. et al. Prototyping and Load Balancing the Service Based Architecture of 5G Core Using NFV. In: IEEE CONFERENCE ON NETWORK SOFTWARIZATION (NETSOFT), 2019., 2019, França. **Anais...** IEEE, 2019. p. 228–232.

Citrix Systems, Inc. **What is a Load Balancer? - Load Balancing Definition - Citrix**. Acessado em 15/06/2020, https://www.citrix.com/glossary/load-balancing.html.

COUTINHO, E. F.; CARVALHO SOUSA, F. R. de; REGO, P. A. L.; GOMES, D. G.; SOUZA, J. N. de. Elasticity in cloud computing: a survey. **annals of telecommunications-annales des télécommunications**, Brasil, v. 70, n. 7-8, p. 289–309, 2015.

EHLERS, R. S. Análise de séries temporais. **Universidade Federal do Paraná**, Brasil, 2007.

ETSI. **GS NFV-MAN 001 V1. 1.1 Network Function Virtualisation (NFV); Management and Orchestration**. 2014. n. NFV-MAN 001. V1.1.1.

FIRMIN, F.; 3GPP MCC. **The Evolved Packet Core**. USA: 3GPP, 2018.

FOWLER, M. Richardson maturity model. **URL: http://martinfowler. com/articles/richardsonMaturityModel. html**, USA, 2010.

FRAMINGHAM, M. **The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast**. [S.l.]: International Data Corporation (IDC), 2019.

FREE5GC.ORG. **open-source 5GC**. Acessado em 31/05/2020, http://www.free5gc.org/.

GALANTE, G.; BONA, L. C. E. de. A Survey on Cloud Computing Elasticity. **IEEE/ACM Fifth International Conference on Utility and Cloud Computing**, Washington, DC, 2012.

GALANTE, G.; DE BONA, L. C. E.; MURY, A. R.; SCHULZE, B.; ROSA RIGHI, R. da. An analysis of public clouds elasticity in the execution of scientific applications: a survey. **Journal of Grid Computing**, Brasil, v. 14, n. 2, p. 193–216, 2016.

Knoll, T. M. Life-cycle cost modelling for NFV/SDN based mobile networks. In: CONFERENCE OF TELECOMMUNICATION, MEDIA AND INTERNET TECHNO-ECONOMICS (CTTE), 2015., 2015, Germany. **Anais...** IEEE, 2015. p. 1–8.

Kreutz, D.; Ramos, F. M. V.; Veríssimo, P. E.; Rothenberg, C. E.; Azodolmolky, S.; Uhlig, S. Software-Defined Networking: a comprehensive survey. **Proceedings of the IEEE**, USA, v. 103, n. 1, p. 14–76, 2015.

MIJUMBI, R.; SERRAT, J.; GORRICHO, J.-L.; BOUTEN, N.; DE TURCK, F.; BOUTABA, R. Network function virtualization: state-of-the-art and research challenges. **IEEE Communications surveys & tutorials**, Spain, v. 18, n. 1, p. 236–262, 2015.

NGINX. **What Is Load Balancing? How Load Balancers Work**. Acessado em 15/06/2020, https://www.nginx.com/resources/glossary/load-balancing/.

NGUYEN, V.-G.; GRINNEMO, K.-J.; TAHERI, J.; BRUNSTROM, A. On Load Balancing for a Virtual and Distributed MME in the 5G Core. In: IEEE 29TH ANNUAL INTERNATIONAL SYMPOSIUM ON PERSONAL, INDOOR AND MOBILE RADIO COMMUNICATIONS (PIMRC), 2018., 2018, Itália. **Anais...** IEE, 2018. p. 1–7.

ROSA RIGHI, R. da. Elasticidade em cloud computing: conceito, estado da arte e novos desafios. **Revista Brasileira de Computação Aplicada**, Brasil, v. 5, n. 2, p. 2–17, 2013.

ROSA RIGHI, R. da; RODRIGUES, V. F.; DE NARDIN, I. F.; DA COSTA, C. A.; ALVES, M. A. Z.; PILLON, M. A. Towards providing middleware-level proactive resource reorganisation for elastic HPC applications in the cloud. **International Journal of Grid and Utility Computing**, Brasil, v. 10, n. 1, p. 76–92, 2019.

SALHAB, N.; RAHIM, R.; LANGAR, R. NFV Orchestration Platform for 5G over On-the-fly Provisioned Infrastructure. In: IEEE INFOCOM 2019-IEEE CONFERENCE ON COMPUTER COMMUNICATIONS WORKSHOPS (INFOCOM WKSHPS), 2019, França. **Anais...** IEEE, 2019. p. 971–972.

YANG, L.; FOSTER, I.; SCHOPF, J. M. Homeostatic and Tendency-based CPU Load Predictions. **IEEE - Parallel and Distributed Processing Symposium, 2003. Proceedings. International**, Chicago, 2003.